Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Organizace předmětu Internet věcí

Diplomová práce

Autor: Bc. Patrik Urbaník
Studijní obor: Informační management (im2-p)

Vedoucí práce: Ing. Pavel Blažek, Ph.D.

Hradec Králové srpen 2021

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 16.8.2021 Patrik Urbaník

Poděkování:

Děkuji vedoucímu diplomové práce panu Ing. Pavlu Blažkovi, Ph.D. za jeho čas, metodické vedení, za cenné a odborné rady, kterými přispěl k vypracování této práce.

Anotace

Předmětem diplomové práce je organizace a vypracování výukového materiálu nového předmětu Internet věcí (IOT) na Fakultě informatiky a managementu UHK. Obsahem dokumentu je podklad pro teoretickou část výuky, který je tvořen ze dvou tematických celků. Ty odpovídají dvěma online kurzům společnosti Cisco, které byly předlohou pro strukturu předmětu. První celek, 4. až 9. kapitola (1.-6. týden), vychází z kurzu IoT Fundamentals: Connecting Things, který je úvodem do problematiky Internetu věcí. Druhý tematický okruh, 10. až 15. kapitola (7.-12. týden), je založen na kurzu IoT Fundamentals: IoT Security, jehož zaměřením je kybernetická bezpečnost a hrozby IoT. Ty byly doplněny a výrazně rozšířeny o informace z odborných publikací a oficiálních dokumentací. Obsahem dokumentu je tedy dvanáct logicky i obsahově navazujících témat, které studenty seznámí se základními pojmy a strukturami IoT systémů, s metodikou tvorby odborného řešení a s bezpečností a hrozbami Internetu věcí obecně.

Annotation

Title: Organization of the subject Internet of Things (IoT)

The subject of the diploma thesis is organization and elaboration of study materials for a new subject Internet of Things (IOT) at the Faculty of Informatics and Management UHK. The content of this document is the basis for the theoretical part of the course, which consists of two thematic units. These correspond to the two online Cisco educational courses that were the model for the course structure. The first unit, chapters 4 to 9 (weeks 1 to 6), is based on the IoT Fundamentals: Connectings Things course, which is an introduction to the Internet of Things. The second unit, chapters 10 to 15 (weeks 7 to 12), is based on the IoT Fundamentals: IoT Security course, which focuses on cyber security and IoT threats. These were supplemented and significantly expanded with information from professional publications and official documentation. Therefore the content of the document is twelve logically arranged topics that will introduce students to the basic concepts and structures of IoT systems, with the methodology of creating professional solutions and the security and threats of the Internet of Things in general.

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Internet věcí (IoT) je v současnosti globálním trendem a možná nejrychleji se rozvíjející technologií, která ovlivňuje vývoj všech průmyslových i neprůmyslových odvětví a život každého z nás. Jejím účelem je, jako u většiny nových technologií, zefektivnit, usnadnit, optimalizovat procesy a aktivity, ať už ve firmách, nebo v domácnostech. Stále více lidí tuto technologii využívá, aniž by si to uvědomovali, např. virtuální asistenti, chytré spotřebiče, kamerové systémy apod. V průmyslu je tento rozvoj daleko rychlejší a stále více firem implementuje nová řešení různých složitostí a velikostí. Může se jednat o relativně jednoduché systémy řízení vytápění, osvětlení a bezpečnostní systémy, nebo o celé autonomní výrobní linky. Ovšem, jak už to bývá, žádná nová technologie nepřináší jen výhody. Rozvoj této technologie je dokonce tak rychlý, že s ním ostatní technologie a přístupy nedokážou držet krok. Nasvědčuje tomu především to, že, ještě i v roce 2021, technologie IoT, až na několik iniciativ různých velkých společností, nepodléhá téměř žádným standardům. To vede podniky k vývoji vlastních proprietárních řešení. Výsledkem je velká technologická a infrastrukturní rozmanitost a nízká úroveň interoperability aplikací různých společností. Hlavním problémem a nejdiskutovanějším tématem jsou však hrozby a nebezpečí, které s sebou IoT přináší. Podstata IoT spočívá ve sběru, zpracování a analýze nepředstavitelného objemu dat, který každým dnem roste. Právě tato data, na kterých může záviset fungování celého nemocničního oddělení nebo výrobní linky, jsou hlavní hrozbou, pokud dojde k jejich zneužití. Hrozby se ovšem netýkají jen pokročilého technologického vybavení, ale i běžných domácích spotřebičů (lednice, pračky, kávovary, termostaty a další), chytrých doplňků (hodinky, sluchátka a další) apod.

2 Cíl práce

Cílem diplomové práce je vypracovat výukový materiál pro předmět Internet věcí určený pro studenty prezenční i kombinované formy studia. Součástí výukové části budou teoretický a praktický blok, které budou doplněné o prvky zpětné vazby pro lektory i studenty, z nichž bude patrná úroveň dosažené znalosti probíraného tématu. Součástí materiálu bude také souhrnné závěrečné ověření znalostí. Obsah předmětu bude postavený na Cisco kurzech „IoT Fundamentals: Connecting Things“ a „IoT Fundamentals: IoT Security“ doplněných o další zdroje vhodně rozšiřující pohled na tématiku s využitím odborných zdrojů a dokumentací.

3 Metodika zpracování

Technologie Internetu věcí je jednoznačně jedním z nejdiskutovanějších a nejaktuálnějších témat v oblasti výpočetní techniky. Účelem předmětu Internet věcí (IOT) je studenty seznámit s tímto, stále se rozvíjejícím, trendem a poukázat nejen na jeho důležitost v různých aplikačních oblastech, jako je průmysl, zdravotnictví, energetika, informační technologie a spotřebitelské technologie, ale i upozornit na různé bezpečnostní hrozby, které s sebou přináší.

Věcný obsah Internetu věcí, a tím pádem i předmětu, je velmi rozsáhlý. Aplikují a dávají se v něm do kontextu dosud získané znalosti hned z několika oborů (Informační management, Aplikované informatika) a předmětů (Bezpečnost dat, Programování, Počítačové sítě, Principy počítačů a další) vyučovaných na VŠ, ale i z učiva SŠ (Elektronika, Hardwarové prvky). Z tohoto důvodu se v teoretické části předpokládá alespoň základní znalost problematiky z těchto oblastí. Související témata nebudou v průběhu výuky detailně probírána, ale jen zmíněna pro připomenutí.

Předpokládá se, že předmět bude určen pro studenty, jak z prezenční, tak i z kombinované formy studia pro obory Informační management (bakalářský i magisterský) a Aplikovaná informatika (bakalářský i magisterský). To, hlavně pokud studenti bakalářských oborů nenavštěvovali školu s technickým zaměřením, s sebou přináší hned několik požadavků nejen na tvorbu materiálů, pro výklad a pro výuku obecně, ale i na průběžné a závěrečné testování, z důvodu rozdílných počátečních znalostí studentů.

Časová dotace výuky je stanovena na jedno cvičení o délce tří vyučovacích hodin týdně. Cvičení je rozděleno na teoretickou část, u které je délka stanovena maximálně na první vyučovací hodinu (tj. 40-45 minut), a na praktickou část ve zbytku cvičení (tj. 80-90 minut).

Struktura teoretické části předmětu vychází, ale ne zcela kopíruje, dva online odborné kurzy společnosti Cisco. Jedná se o IoT Fundamentals: Connecting Things, který je úplným úvodem do problematiky Internetu věcí a dává do souvislosti výše zmíněné oblasti, a IoT Fundamentals: IoT Security zabývající se výhradně kybernetickou bezpečností a hrozbami souvisejícími s IoT. Oba kurzy jsou kvalitně

zpracované, a kromě teoretických podkladů, obsahují i praktická cvičení, která jsou ovšem v drtivé většině případů zaměřená na prostředí Packet Tracer, anebo určená pouze pro testovací/laboratorní prostředí z důvodu bezpečnosti a podstaty některých aktivit, především v druhém kurzu. Lze je oba absolvovat v anglickém jazyce, což je velkým přínosem z pohledu IT praxe. Za jejich úspěšné absolvování je možné získat celkem dva certifikáty od společnosti Cisco, které jsou mezinárodně velmi uznávané. Samotná výuka je však vedena v jazyce českém s využitím rozšířených materiálů v českém jazyce.

I když se může zdát, že na sebe kurzy navazují, neplatí to zcela. V některých oblastech se překrývají, či dokonce opakují. Dalším nedostatkem je jejich přílišná stručnost v některých částech a často představují pouze vlastní aplikace a řešení preferované společností Cisco. Posledním problémem, s nímž je nutno při vytváření výukových materiálů počítat, je velmi dynamický vývoj v oblasti IoT (a IT obecně). Obecně platný základ bude nutné periodicky doplňovat a obměňovat o aktuálně platné poznatky. Z tohoto důvodu byly použity Cisco materiály pouze jako základní zdroj obecných, a z jiných předmětů známých, informací. Lze tak konstatovat, že se zdaleka nejedná o pouhý překlad do českého jazyka. Doplňující informace byly čerpány z odborné literatury, článků a příspěvků na webu a z oficiálních dokumentací a technických specifikací (především standardů a protokolů).

Jako základ je definováno celkem 12 témat s tím, že 1. až 6. týden je seznámením s oblastí IoT (odpovídá IoT Fundamentals: Connecting Things) a 7. až 12. týden se věnuje výhradně kybernetické bezpečnosti IoT řešení (odpovídá IoT Fundamentals: IoT Security). Ke každému tématu je vytvořena přednášková prezentace, které jsou studentům po příslušném týdnu zpřístupněny v e-learningovém portálu oliva.uhk.cz.

Praktická část cvičení obsahuje ukázky a demonstrace podle probrané tématiky v teoretické části a laboratorní praktická cvičení studentů. Ukázky jsou buď vlastní nebo založené na cvičeních z kurzů (např. Cisco laby). Studenti pracují buď v simulačním prostředí Cisco Packet Tracer, nebo na fyzických zařízeních dostupných v laboratoři. Na základě vzešlé nepříznivé globální epidemiologické situace a nařízené distanční výuce, se ve snaze alespoň částečně vynahradit praktická cvičení, využívá i webový simulátor Arduina TinkerCAD, protože

prostředí Packet Tracer není příliš vhodné pro tuto formu vzdělávání a neumožňuje požadovanou úroveň virtualizace komponent.

Vzhledem k cíli a rozšířením výuky není absolvování závěrečných online testů v kurzech Cisco dostačující pro ověření dosažených znalostí, a tím pádem pro uzavření závěrečného hodnocení. Ověření získaných znalostí probíhá závěrečným teoretickým testem v prostředí blackboard a vypracováním individuálního nebo skupinového projektu na libovolné související téma. Zahrnuje návrh řešení a vytvoření dokumentace podle doporučeného postupu v kapitole 9. Tvorba IoT řešení.

4 Věci a jejich propojení

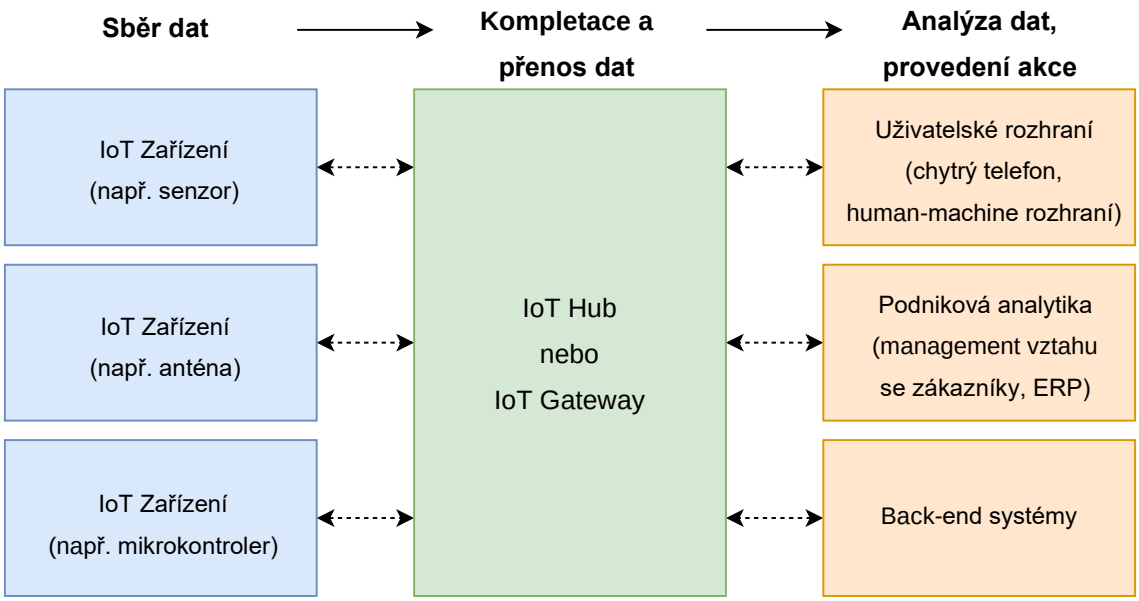
Informace v kapitole Věci a jejich propojení jsou převzaty z online kurzu Cisco [1], z části Chapter 1: Things and Connections, pokud není uvedeno jinak.

Technologie stále více ovládají svět a stávají se součástí každodenních činností většiny z nás. Je tu ovšem jedna technologie, která v posledních letech zaznamenala obzvlášť velký pokrok jak v průmyslu, tak ve spotřebitelské sféře, a to je Internet věcí.

4.1 Internet věcí

Internet věcí (IoT) je globální síť fyzických věcí připojených k internetu. IoT je spojením internetu spolu se sběrem, zpracováním a analýzou dat. Každé připojené zařízení má vlastní unikátní ID a mohou mezi sebou komunikovat bez potřeby interakce člověka, a z tohoto důvodu se označují jako „smart“. Součástí IoT může být v současnosti prakticky cokoli. Od elektrických spotřebičů, jako lednice, pračka, konvice, mikrovlnná trouba, přes mobilní telefony, hodinky, automobily, až po letadla nebo medicínské přístroje, či dokonce zvířata, pokud jsou opatřena biočipem, a lidi v případě, že používají lékařská zařízení, jako kardiostimulátor, pumpy pro diabetiky a podobná zařízení[2]. Data z takovýchto zařízení jsou analyzována a použita k interakci a k vylepšení jejich funkcionality. V současnosti se tento výraz pojí hlavně ve spojení se smart city a smart home.

V principu se jedná o síť tzv. web-enabled chytrých zařízení, což jsou zařízení, která jsou postavená na službě World Wide Web a je k nim možné přistoupit přes webový prohlížeč. Tato zařízení využívají vestavěné procesory, senzory, komunikační a měřící prvky, ke sběru a zasílání získaných dat. Zařízení mezi sebou data sdílejí přes IoT gateway nebo další okrajová zařízení, odkud jsou dále zaslána do cloudu k analýze, anebo jsou alternativně analyzována lokálně, již na místě vzniku. Jednotlivé prvky mezi sebou mohou komunikovat i na přímo a na základě těchto dat vykonávat různé akce. Zařízení dokážou pracovat téměř bez lidského zásahu, pokud nepočítáme jejich nastavení a předání základních instrukcí [2].



Obrázek 1 Struktura IoT systému
Zdroj: vlastní zpracování (podle [2])

4.1.1 Historie IoT

I když je myšlenka IoT vcelku nedávnou záležitostí, nápad obohatit běžné objekty o senzory a formu inteligence se zrodil již v letech 1980-1990. Ovšem s úrovní technologií, které byly v této době dostupné, byl pokrok velice pomalý a až na pár výjimek se vizi příliš realizovat nedařilo. Čipy byly příliš velké a neexistovala efektivní cesta, jak by mezi sebou zařízení mohla komunikovat. Prvním člověkem, který termín internet věcí zmínil, byl Kevin Ashton v roce 1999. Ovšem trvalo téměř další desetiletí, než se tato vize začala stávat skutečností. První aplikací IoT bylo přidávání RFID čipů do zařízení za účelem jejich lokalizace. Postupně začali náklady, na výrobu a integraci senzorů do zařízení, spolu s připojením k internetu klesat a IoT se pomalu začalo blížit k podobě, jakou známe ze současnosti. Největší zájem o tyto technologie byl hlavně ze strany výrobních podniků v podobě machine-to-machine (M2M) aplikací [3].

4.2 Jaký přínos má technologie IoT?

IoT pomáhá lidem žít a pracovat chytřeji a získat plnou kontrolu nad jejich životy. Ovšem hlavní přínos IoT není v automatizaci domácností, ale v oblasti podniků a výroby, kde jsou IoT prvky téměř nepostradatelnou součástí. Podnikům poskytuje

náhled v reálném čase do firemních procesů, ať už se jedná o výrobní roboty a přístroje nebo logistické operace. Automatizace firemních procesů snižuje nároky na lidskou práci, zefektivňuje a zlepšuje výrobní postupy a umožňuje lepší kontrolu a monitoring. Čím dál více podniků si všímá potenciálu IoT a stále častěji je tato technologie základem všech operací a procesů ve firmě [3]. Největší zastoupení má IoT u podniků z odvětví výroby a transportu, dále ho lze najít též i v zemědělství, stavebnictví a energetice. Využití této technologie v průmyslu se označuje pojmem Industrial IoT (IIoT), které je specifickou aplikací obecného konceptu IoT. Ovšem jako veškeré technologie a vylepšení sebou IoT kromě výhod přináší i nevýhody [2]:

Výhody:

- Informace přístupné kdekoli, kdykoli z čehokoli.
- Zlepšuje komunikaci mezi zařízeními.
- Šetří čas a peníze.
- Automatizace procesů vede ke zlepšení poskytovaných služeb a snižuje potřebu interakce člověka se systémem.

Nevýhody:

- S rostoucím počtem připojených zařízení a množstvím dat se zvyšuje riziko krádeže dat.
- Pokud bude automatizace postupovat i do dalších činností ve firmě, podniky budou muset spravovat tisíce, desetitisíce, až statisíce IoT zařízení.
- Pokud dojde k problému v systému, je šance, že to ovlivní fungování celého IoT systém.
- Neexistence jednotného standardu ztěžuje komunikaci mezi zařízeními od různých výrobců.

V reálném světě se můžeme s využitím IoT setkat téměř na každém kroku. Příklady zařízení a aplikací IoT [4]:

- Chytré domácnosti (Smart Homes) – spotřebiče, osvětlení, vytápění, bezpečnostní systémy atd.

- Chytré doplňky (Smart Wearables) – hodinky, náramky, ale už i třeba náušnice a prsteny
- Automobily – palubní počítače, navigace, autopilot, asistent parkování
- Průmyslové využití IoT (IIoT) – logistika, výroba, monitoring
- Chytrá města (Smart Cities) – veřejné osvětlení, odpadkové koše, semafory, radary, chytrá parkoviště
- Chytré zemědělství (Smart Farming) – senzory vlhkosti, měření srážek
- Chytré nakupování (Smart Retail) – čtečky čárových kódů, obchody bez obsluhy
- Chytrá energetika (Smart Grid) – efektivnější řízení a distribuce elektřiny, zlepšení dostupnosti
- Chytré zdravotnictví (Smart Healthcare) – monitorování zdravotního stavu pacientů v i mimo zdravotnická zařízení, měření tlaku, cukru, kyslíku, váhy

4.2.1 Pilíře IoT podle Cisco

Rapidní nasazování nových IoT řešení napříč všemi oblastmi s sebou přináší spoustu otázek a výzev. Jak integrovat miliony zařízení od různých výrobců? Jak integrovat nová zařízení do již existujících sítí a řešení? Jak tato nová zařízení zabezpečit? Za účelem pomoci s překonáním těchto překážek společnost Cisco představila svůj vlastní koncept Cisco IoT systému.

Cisco IoT systém definuje šest základních pilířů, které by měly pomoci s nasazením nových IoT systémů:

- **Síťové připojení** (Network Connectivity)
- **Fog Computing**
- **Kybernetické a fyzické zabezpečení** (Cybersecurity and Physical Security)
- **Analytika dat** (Data Analytics)
- **Řízení a automatizace** (Management a Automation)
- **Aplikační platforma** (Application Enablement Platform)

Obrázek 2 Pilíře Cisco IoT systému
Zdroj: [1]

4.3 IoT systém

V současnosti existuje nespočet IoT kompatibilních zařízení, ale jejich základ je v podstatě stejný. Většina z nich totiž využívá senzory, kontrolery a aktuátory k provádění akcí. Spojení více různých zařízení do jednoho funkčního celku poté tvoří řízený systém (controlled system). Ten je ovládán řídícím systémem (kontrolerem), který řídí, kontroluje, reguluje a usměrňuje chování ostatních zařízení v řízeném systému k dosažení požadovaného výstupu. V posledních letech se takovýto systém stal základem většiny spotřebičů a zařízení, které používáme v každodenním životě, a je nedílnou součástí, stále se rozrůstající, automatizace napříč všemi odvětvími. Hlavním rysem systému je jasný vztah mezi vstupem a výstupem, který lze matematicky popsat.

4.3.1 Senzory

Senzor je zařízení, jehož hlavní funkcí je detekovat události nebo změny prostředí ve fyzickém světě. Měřenou veličinou může být světlo, vlhkost, pohyb, tlak, teplota a podobně. Senzor je vždy použit v kombinaci s jiným elektrickým zařízením. Takto naměřená data poté posílá do dalších zařízení, typicky do kontroleru, ke kterému může být senzor připojen přímo nebo bezdrátově. K propojení se používá, jak analogových, tak digitálních obvodů. Kontroler poté data vyhodnocuje a podle zjištěných informací ovlivňuje chování aktuátoru.

4.3.2 Aktuátory

Aktuátor je zařízení, většinou mechanické povahy, které se chová nebo vykonává úkony podle instrukcí obdržených od kontroleru. V IoT se typicky rozlišují tři typy aktuátorů:

- Elektrické – Využívají elektrickou energii k provádění mechanických operací.
- Hydraulické – Využívají tlaku kapaliny k provádění mechanických operací.
- Pneumatické – Využívají stlačený vzduch k provádění mechanických operací.

Aktuátory mohou být zodpovědné například za transformaci elektrického signálu na fyzický výstup. Výstup může poskytnout uživateli informace prostřednictvím signálů LED, změnou v prostředí nebo ovlivněním chování jiného zařízení.

4.3.3 Kontrolery

Kontroler tvoří spojení mezi senzorem a aktuátorem. Může být součástí zařízení, stejné sítě nebo umístěný v tzv. fogu. Jeho hlavní funkcí je sběr dat od senzorů a poskytnutí připojení k internetu nebo do sítě s ostatními zařízeními. Kontrolery mají možnost se rozhodovat na základě analýzy dat nebo mohou data odeslat výkonnějším počítačům, které se nacházejí ve stejné síti nebo jsou dostupné přes internet v cloudu.

Asi nejpoužívanější kontrolery využívající uzavřenou smyčku, jsou Arduino a RaspberryPi (dále jen RasPi). Oba dokážou fungovat bez internetu a jediný zásadní rozdíl mezi nimi je fyzická velikost, výpočetní výkon a operační systém. V praxi se Arduino typicky využívá pro příjem analogových vstupů, ale není to zásadou. Běžně se totiž obě zařízení využívají spolu, kdy Arduino může být zodpovědné za sběr dat a RasPi za jejich zpracování a případně i analýzu.

4.4 Průběh procesu v IoT systému

Jak bylo již zmíněno, nejjednodušší systém tvoří senzory připojené, pomocí drátu nebo bezdrátově, k aktuátorům a kontrolerům. Jako v případě ostatních systémů, kde jedno zařízení má více funkcí, tak v IoT systému tomu není jinak a konkrétně se jedná o kontroler. V první řadě dokáže autonomně sbírat data od senzorů. Dále

může působit v roli gateway pro lokální sítě, což znamená, že v případě zpracování dat na okraji sítě, zprostředkovává spojení mezi tímto místem a místy vzniku dat. Takové zapojení nazýváme fog computingem. Druhou možností je, že může zajišťovat komunikaci mezi zařízeními a aplikací v cloudu pro případ, kdy je potřeba uložit a analyzovat data z více míst a zpřístupnit je všem ostatním zařízením. Data mohou být v cloudu zpracována, analyzována i uložena. K jejich zpracování však může dojít už na úrovni fog computingu a v cloudu být jen uskladněna.

Proces je tedy posloupnost kroků nebo akcí, využívajících hodnot vstupu k dosažení požadovaného výstupu uživatele procesu. Systém poté funguje jako nadřazený prvek a soubor pravidel, která určují kroky procesu. Jednoduchá struktura procesu by se tedy dala popsat: Vstup -> Akce -> Výstup, např. současná rychlost auta -> brzdi -> snížení rychlosti.

V procesech se často využívá zpětné vazby. To je situace, kdy hodnoty výstupu procesu ovlivňují budoucí vstup. Zpětná vazba se často realizuje ve smyčce, čímž dojde k nepřetržitému ovlivňování vstupu a dochází k využití principu příčina-následek. Většina systémů má nějakou formu zpětné vazby a rozlišují se dva základní druhy. První je pozitivní zpětná vazba, která posiluje původní vstup. To znamená, že urychluje transformaci výstupu stejným směrem, jako předchozí naměřená hodnota. Druhým typem je negativní zpětná vazba. Ta ruší nebo působí proti původnímu vstupu, což vede k postupné stabilizaci systému na úrovni přibližné rovnováhy.

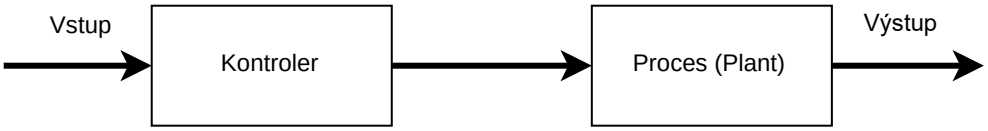
4.4.1 Řídící systémy

Řídící systém běžně zahrnuje kontroler, který využívá vstupu a výstupu k řízení a regulaci chování systému, za účelem dosažení požadovaného stavu. Vstup specifikuje podobu výstupu v průběhu celého procesu. Kontroler určuje jaké specifické změny je potřeba provést. V teorii řízení se tato řízená část, proces a aktuátor, často označují pojmem „plant“. Teorie řízení je základní strategií pro volbu správného vstupu, aby došlo ke generování požadovaného výstupu. Tato teorie je aplikovatelná ve všech typech zařízení. Například u automobilu řidič plní roli kontroleru a auto plní roli plant [5].

Systém s otevřenou smyčkou (Open-loop)

Ve světe IoT se využívají především dva typy systémů. Jsou to systémy s otevřenou smyčkou (Open loop) a systémy s uzavřenou smyčkou (Closed loop).

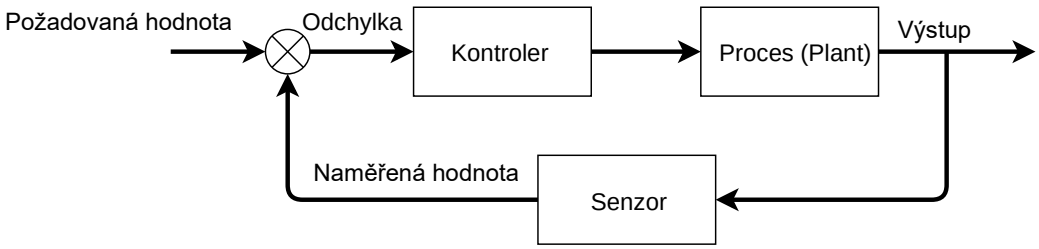
V systému s otevřenou smyčkou je akce řídící jednotky zcela nezávislá na předchozím výstupu systému. Příkladem jsou třeba elektrické sušiče rukou, kde bude teplý vzduch foukat do té doby, dokud pod ním budou ruce bez ohledu na to, jestli jsou již suché. Dále se s nimi setkáme i u většiny kuchyňských spotřebičů, které fungují na základě nastavených programů a po dobu daného času. To znamená, že otevřené systému nepřijímají žádnou zpětnou vazbu. Výhodou takových systému je jejich jednoduchost na výrobu a design, nenáročnost na údržbu, stabilita a většinou jednoduché použití. Nevýhodou je nepřesnost a nespolehlivost [6].



Obrázek 3 Systém s otevřenou smyčkou (open-loop)
Zdroj: vlastní zpracování (podle [6])

Systém s uzavřenou smyčkou (Closed-loop)

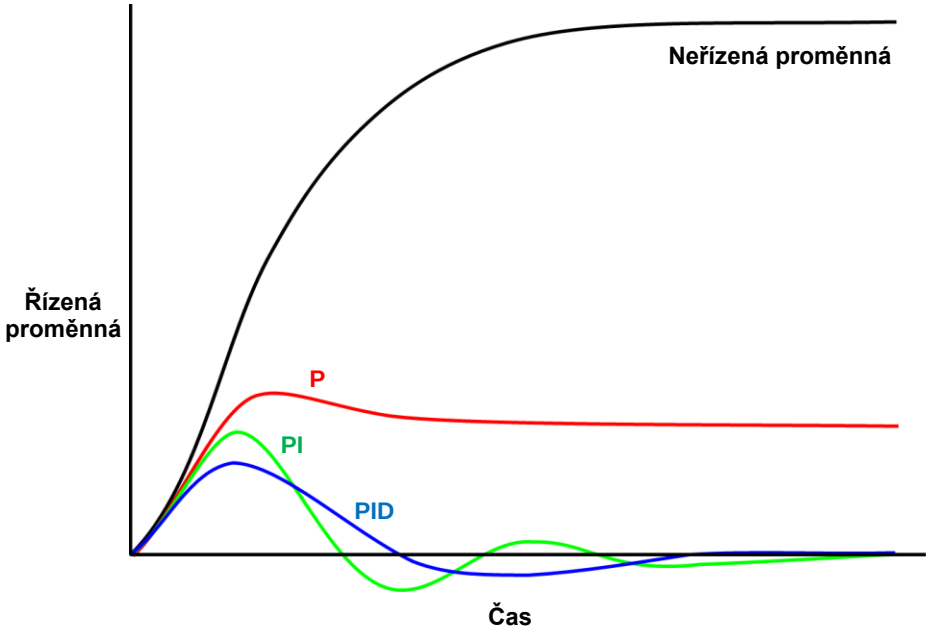
Oproti tomu v systému s uzavřenou smyčkou lze vstup ovlivnit s využitím zpětné vazby za účelem dosažení požadovaného stavu. Takový systém se již nazývá automatický řídící systém. Zařízení, která toto využívají lze opět najít okolo nás. Například klimatizace nebo topení v automobilu, kde obojí funguje s ohledem na teplotu v místnosti nebo autě. Výhodou využívání zpětné vazby je vyšší přesnost a spolehlivost, jelikož dokážou automaticky reagovat na nepřesnosti. Nevýhodou je větší nákladnost výroby, složitost, vyžadují pravidelnou údržbu a nižší stabilita [6].



Obrázek 4 Systém s uzavřenou smyčkou (closed-loop)
Zdroj: vlastní zpracování (podle [6])

Kontrolerů využívajících uzavřenou smyčku existuje hned několik typů. Efektivní způsob, jak v systému implementovat smyčku pro zpětnou vazbu představují PID kontrolery. PID je zkratka pro proporční (proportional – P), integrální (integral – I) a derivační (derivative – D) což označuje tři způsoby, jakými kontroler zpracovává chybu/odchylku (viz Obrázek 5).

- **P** – Kontroler se soustředí specificky na rozdíl mezi zamýšleným výstupem a naměřenou hodnotu výstupu a na základě toho upraví budoucí vstup, kde změna odpovídá naměřenému rozdílu. Typem P mohou být kontrolery pneumatické, analogové a elektronické.
- **PI** – Kontrolery využívají historická data k určení, jak dlouho je systém odchýlen od požadovaného výstupu. Čím déle tím více se změní budoucí vstup.
- **PID** – Tyto kontrolery měří naopak to, jak rychle se k požadované hodnotě výstupu blíží a upravují vstup tak, aby požadovaný výstup byl dosažen co nejpřesněji. Typy PI a PID mohou reprezentovat pouze digitální kontrolery.

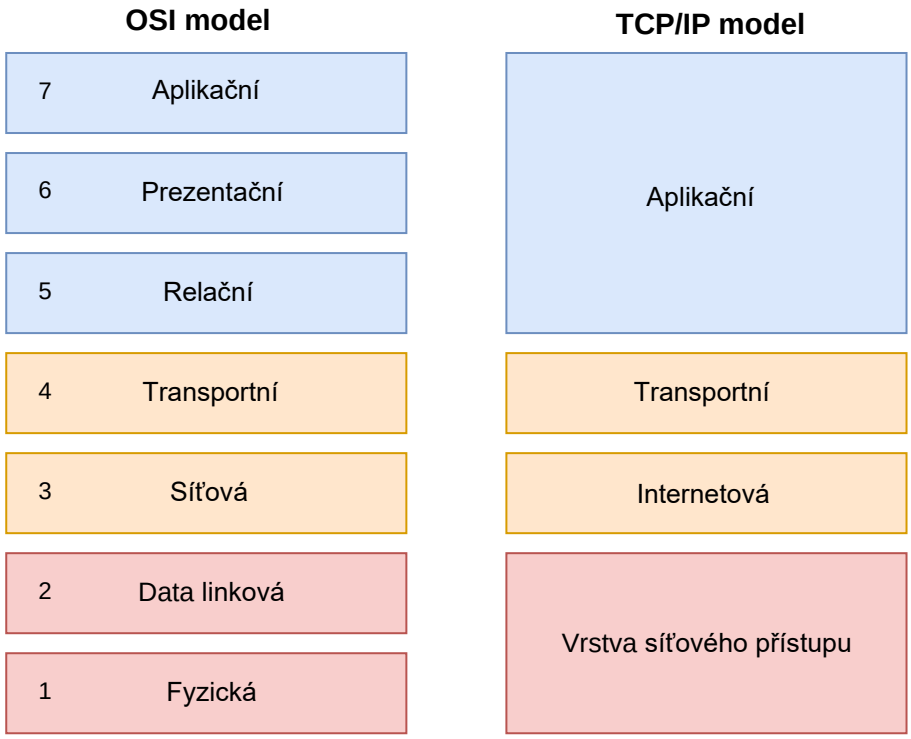


Obrázek 5 Graf průběhu PID kontrolerů

Zdroj: vlastní zpracování (podle https://instrumentationtools.com/what-is-pid-controller/)

4.5 Síťová komunikace

Ke znázornění fungování sítě se často používají vrstvené síťové modely. Výhodou takových modelů je, že zjednodušují popis a návrh fungování sítě a síťových protokolů. Nejčastěji používanými modely jsou OSI a TCP/IP (viz Obrázek 6). Starší TCP/IP model definuje čtyři funkční oblasti (vrstvy), které musí nastat, aby bylo docíleno komunikace. Druhý referenční model OSI poskytuje rozšířený seznam funkcí a služeb, která se mohou objevit na jednotlivých vrstvách, a navíc popisuje i to, jak spolu vrstvy spolupracují. V počítačových sítích se stále využívají oba modely, proto je dobré je znát. Často se na vrstvy TCP/IP modelu odkazuje jejich názvem a na vrstvy OSI modelu jejich číslem.



Obrázek 6 Síťové modely OSI a TCP/IP
Zdroj: vlastní zpracování (podle [1])

Aplikační vrstva modelu TCP/IP představuje data uživateli, obstarává kódování a řídí dialog. Transportní vrstva podporuje komunikaci mezi různými zařízeními napříč sítí. Internetová vrstva určuje nejlepší cestu, kterou data putují sítí. Vrstva síťového přístupu řídí samotná fyzická zařízení a přenosová média, která tvoří síť.

Podobně je tomu i v případě OSI modelu. 1. Fyzická vrstva popisuje mechanické, elektrické, funkcionální a procedurální způsoby k aktivaci, udržení a deaktivaci fyzických propojení pro přenosu dat ve formě bitů do a ze síťového zařízení. 2. Data linková vrstva udává metody pro výměnu datových rámců mezi zařízeními. 3. Síťová vrstva poskytuje služby k výměně paketů skrz síť mezi určenými koncovými zařízeními. 4. Transportní vrstva definuje funkce pro segmentaci, přenos a složení dat pro každou navázanou komunikaci. 5. Relační vrstva poskytuje služby vyšší 6. Prezentační vrstvě k řízení dialogů a výměny dat. Kromě toho 6. vrstva umožňuje i základní reprezentaci dat, přenesených mezi službami aplikační vrstvy. A poslední 7. Aplikační vrstva obsahuje protokoly použité pro komunikaci mezi samotnými procesy.

Standardizace

K zajištění efektivní komunikace mezi rozmanitými IoT systémy je vyžadováno použití konzistentních, bezpečných a známých technologií a standardů. Standard stanovuje důležité vlastnosti materiálu, výrobku nebo pracovního postupu, což vede ke standardizaci. Mezi nejznámější soustavy technických standardů patří ISO a IEC. Za účelem standardizace a strukturace připojení byly vytvořeny dva zmíněné modely. S obrovským rozvoje IoT je také důležité zajistit to, aby zařízení mezi sebou mohla komunikovat bez ohledu na výrobce a technologii. Mnoho organizací, jako např. Industrial Internet Consortium (IIC), OpenFog Consortium a Open Connectivity Foundation (OCF) spolupracují s lidmi z oblasti průmyslu, vlády a vzdělaní s cílem podpořit tvorbu a osvojení nových IoT technologií. Navíc pomáhají s tvorbou standardizovaných architektur a frameworků, které umožní zařízením spolehlivě a bezpečně komunikovat. Například sdružení IIC představila standardizovanou architekturu IIRA pro průmyslové využití IoT. Také organizace OCF pracuje na tvorbě několika řešení, které mají být součástí jedné interoperabilní specifikace. Podílejí se i na financování projektu IoTivity, což je open source framework, který umožňuje komunikaci zařízení bez ohledu na výrobce a operační systém.

Propojení zařízení

Pro komunikaci mezi IoT zařízeními se, podobně jako v klasických počítačových sítích, používají tři základní typy médií:

- **Měděný kabel** – Měděné médium je levné, nenáročné na instalaci a disponuje nízkým elektrickým odporem. Jejich nevýhodou je ovšem omezená přenosová vzdálenost a podléhají rušení.
- **Optické vlákno** – Optika dokáže přenášet na výrazně delší vzdálenosti a nepodléhá rušení. Signál je vytvářen světelnými zdroji, jako laser a LED. Přijímací zařízení detekuje signál pomocí světelných senzorů a převádí je na napětí. Základní klasifikace optický kabelů je single-mode (SMF) a multi-mode (MMF). SMF se vyznačuje velmi malým jádrem a použitím drahého laseru k zaslání signálu na velké vzdálenosti, až stovky kilometrů. MMF obsahuje větší jádro a využívá levné LED k zasílání světelných signálů. Často se využívá v LAN sítích, protože podporuje rychlost 10 Gb/s až na 550 metrů.
- **Bezdrátové** – Tento způsob je možné realizovat širokou škálou způsobů, například elektromagnetické signály, rádiové a mikrovlnné frekvence a satelitní spojením, ale také pomocí infračerveného paprsku.

4.6 Dopad IoT na soukromí

Podle dat společnosti Cisco bylo v roce 2019 připojeno k internetu zhruba 20 miliard zařízení a odhadovaný počet v roce 2023 je téměř 30 miliard zařízení. Největší podíl v tomto počtu mají chytré telefony a M2M (IoT). Jenže s narůstajícím počtem připojeným zařízení roste i množství dat, která se nacházejí na internetu. S ním pak narůstá nebezpečí jejich kompromitace. V posledních letech se proto firmy čím dál více zabývají zabezpečením dat a ochranou soukromí. Jak lze vlastně chápat soukromí? Jeden pohled na soukromí může být takový, že člověk má možnost určit, jaká data chce sdílet, jak budou sbírána, kým budou sbírána a jak budou využita. Nebo je to schopnost kontrolovat, do jaké míry je člověk identifikovatelný při využívání online nebo offline služeb. Poslední v řadě to může být i právo se úplně izolovat od ostatních [7].

Nejvíce dat uživatelé sdílejí právě prostřednictvím internetu. Nejčastěji se jedná o věk, pohlaví, příjem a geografickou lokaci, velmi často jde též o historii navštívených stránek a výčet by mohl pokračovat. S rozmachem IoT přibývají další kategorie dat a mnohonásobně roste jejich množství, jelikož IoT poskytuje více možností jejich získávání. Díky specifickým zařízením lze sbírat i data o lidských emocích, aktivitách a zdraví. Dalším příkladem jsou populární domácí asistenti (Ciri, Google), kteří mají mikrofon stále zapnutý a dokážou tak poslouchat mimo příkazů dění s jejich funkcí nesouvisející. Ovšem tohle není otázka jen zařízení, která si lidé dobrovolně pořídí. Riziko představují i internetové bezpečností kamery, chytré billboardy, obchodní systémy a další veřejné technologie. Díky nim stále více lidí přichází o pocit soukromí a cítí se být sledováni či posloucháni a nemají možnost to nějak ovlivnit [8].

Většina IoT systémů je vytvářena právě za účelem monitorování lidí a prostředí, přičemž je generováno rozsáhlé množství tzv. metadat. Lidé musí věřit, že jejich soukromé informace (tzv. PII – Personal identifiable information) zůstávají opravdu soukromé a zabezpečené. Spousta nových IoT systémů by proto měla být navržena tak, aby tento cíl byl dosažen. Některá doporučení týkající se soukromí [1]:

- **Transparentnost** (Transparency) – Lidé by měli vědět, jaká data jsou o nich sbírána, proč jsou sbírána a kde budou uložena.
- **Sběr a využití dat** – Zařízení by měla sbírat a ukládat pouze taková data, která jsou relevantní účelu zařízení a mají vliv na jeho funkčnost.
- **Přístup k datům** – Před nasazením nových systémů je důležité určit kdo a za jakých podmínek bude mít přístup k sesbíraným datům.

Informace o tom, jaká data a za jakým účelem zařízení nebo software sbírají, se nacházejí v takzvaných TSA dokumentech (Terms of Service Agreement). Spousta uživatelů tyto dokumenty ovšem ignoruje, jelikož jsou dlouhé a plné „právnických“ pojmů, kterým většinou nerozumí. Tím souhlasí s podmínkami, které by se jim nemuseli zamlouvat, pokud by si dokument přečetli.

4.7 Metadata

Metadata by se jednoduše dala popsat jako data o datech. Mohou být součástí daného digitálního objektu (fotky, videa) nebo mohou být uložena separátně (cloud). Ve většině případů je běžný uživatel nevidí. Příkladem typických metadat je autor, datum vytvoření, datum změny, velikost souboru, místo vzniku, čas vzniku. V závislosti na způsobu využití a získání se taková data dají v určitých případech považovat za soukromá. Mohou být vytvořena manuálně nebo automaticky. Metadata jsou důležitá zejména na webových stránkách, kde je lze najít ve formě tzv. meta tags. Ty se dají chápat jako klíčová slova, popisující obsah stránky. Kromě toho mohou být použity marketingovými organizacemi, vládou a zdravotními institucemi pro „dobré“ účely. Na druhou stranu metadata mohou být zneužity k průniku do soukromí, k sledování, dokonce ke krádeži identity [9].

5 Senzory, aktuátory, mikrokontrolery

Informace v kapitole Senzory, aktuátory a mikrokontrolery jsou primárně převzaty z online kurzu Cisco [1], z části Chapter 2: Sensors, Actuators and Microcontrollers, pokud není uvedeno jinak.

5.1 Základy elektroniky

Každá látka je složena z atomů a každý atom je složen ze tří částic: protony, neurony a elektrony. Protony a neurony se nacházejí uvnitř jádra atomu a elektrony v okolí tohoto jádra. Pro lepší představu si atomové jádro můžeme představit jako slunce a elektrony jako planety, které ho obíhají. Dvě z těchto částic nesou náboj, elektrony záporný a protony kladný. Neuron je považován za neutrální nebo částici bez náboje. V praxi se lze setkat s pojmem protonové číslo (značíme Z), které určuje počet protonů v jádře. Tato hodnota je významná především v chemii, kde protonové číslo jednoznačně definuje chemický prvek [10]. V ideálním případě je uvnitř atomu stejný počet elektronů a protonů, a proto se jeho náboj poté rovná nule. V případě elektriky se částice se stejným nábojem odpuzují (++, --) a s rozdílným nábojem přitahují (+-). Atom se ve skutečnosti nevyskytuje sám, ale existuje jich obrovské množství vedle sebe, což má za následek odtržení jejich elektronů. Elektrony, které jsou od jádra nejdál se dostanou do meziatomárního prostoru, čímž se v daném

atomu poruší rovnováha a získá tak kladný náboj (počet protonů > počet elektronů). Odtržený elektron naopak nese náboj záporný. Tento volný elektron se po čase připojí k jinému atomu. Buď zaplní místo po jiném elektronu nebo navýší počet elektronů a vznikne atom se záporným nábojem. Tento proces se neustále opakuje. Některé látky mají volných elektronů v jeden okamžik více (kovy) a některé méně (sklo) [11]. Elektrický náboj je možné přenášet mezi povrchy těles. Látky, kde se elektrický náboj snadno přenáší, nazýváme vodiče a naopak látky, kde k přemístění náboje nedochází, nazýváme nevodiče (nebo izolanty)[12].

Elektrický náboj (Electric charge) označujeme symbolicky písmenem Q a jeho jednotkou je coulomb, který značíme jako C. Nejmenší elektrický náboj, který nelze dále dělit, nese označení elementární náboj, značí se písmenem *e* a jeho hodnota nabývá $e = \pm 1{,}602 \cdot 10^{-19}$ C. K měření elektrického náboje se používá buď elektroskop (pouze kvalitativně, nedokáže změřit typ ani velikost) nebo elektrometr (se stupnicí, dokáže měřit typ a velikost) [10].

5.2 Základní veličiny a vztahy

Pro připomenutí je zde uvedeno několik základních veličin a vztahů z oboru elektroniky, které jsou důležité do následujících částí zabývajících se zařízeními Arduino a RaspberryPi.

5.2.1 Elektrický proud

Elektrický proud (Electric current) představuje tok elektrického náboje ve formě volných elektronů. Měří se jako množství volných elektronů, které protečou daným místem za sekundu. Proud se označuje písmenem I a jeho jednotka je ampér označována symbolem A. Matematicky vyjádřený vztah mezi nábojem (Q) a proudem (I) je [10]:

$$I = \frac{Q}{t}$$

Elektrický proud se často plete s elektrickým napětím. Hlavním rozdílem je to, že proud teče, kdežto napětí prostě existuje. Důležité je ovšem to, že napětí může existovat bez proudu, ale naopak nikoliv, jelikož proud teče vodičem, který spojuje dva body s napětím. Další věc, která se často plete a zároveň je i nelogická, se týká

směru toku proudu. Směr pohybu elektronů je od záporného pólu ke kladnému, ale v elektrických schématech je standardizované značení směru toku proudu přesně naopak, tzn. od kladného k zápornému[11].

5.2.2 Elektrický potenciál a Elektrické napětí

Potenciál je veličina, která označuje schopnost určitého fyzikálního pole (v tomto případě elektrického) působit na hmotné body, popřípadě náboje v něm umístěné. Jeho hodnota je relativní a vztahuje se k místu s nulovým potenciálem, což u elektrického pole bývá povrch Země. Potenciální energii E_p má každé těleso nacházející se v potenciálovém poli nějaké síly. Změna potenciální energie je definována jako rozdíl potencionální energie původní a nové polohy v potenciálovém poli.[13]

Elektrický potenciál je druhem potenciálu, který popisuje potenciální energii náboje v elektrickém poli. Označuje množství potenciální energie potřebné pro přemístění náboje z daného bodu do bodu, kde se potenciál rovná nule. Elektrický potenciál značíme symbolem φ a jeho jednotkou je joule na coulomb ($J \cdot C^{-1}$) a platí, že 1 $J \cdot C^{-1}$ = 1 V. Potenciální energii lze vyjádřit pomocí práce a platí tedy vztah[11]:

$$\varphi = \frac{Ep}{Q} \wedge Ep = W \rightarrow \varphi = \frac{W}{Q}$$

Veličina W vyjadřuje množství práce potřebné k přemístění náboje Q. Jednotka práce je joule, označována symbolem J. Práce se počítá, jako množství elektrické síly F_e, měřené v newtonech N, vynásobené vzdáleností d v metrech nebo vynásobením intenzity elektrického pole E (měřeno ve $V \cdot m^{-1}$), náboje Q a vzdálenosti d [10].

$$W = F_e \cdot d = E \cdot Q \cdot d$$

Pro samotné elektrické napětí (Electric voltage) platí definice: Napětí 1 V je takové napětí, které je mezi konci vodiče, do kterého konstantní proud 1 A dodává výkon 1 W. Napětí se nejčastěji značí symbolem U (nebo V) a dá vyjádřit dvěma způsoby. První způsob vyjádření napětí je jako absolutní hodnota rozdílu elektrických potenciálů dvou bodů, což může vytvořit elektrický proud [10].

$$U = |\varphi_2 - \varphi_1|$$

Druhým způsobem je vyjádření pomocí podílu práce vykonané při přemístění elektrického náboje Q z bodu A do bodu B a velikosti přemístěného náboje [12].

$$U = \frac{W}{Q}$$

5.2.3 Elektrický odpor

Existují dva typy látek, které v elektronice rozlišujeme, a to vodiče a nevodiče. Rozdíl je v tom, kolik má látka k dispozici volných elektronů, čím více tím vodivější je. Toto se dá vyjádřit veličinou, která se nazývá vodivost. V elektronice se ovšem používá převrácená hodnota zvaná elektrický odpor (Electric resistance). Vodiče pak mají malý odpor a nevodiče vysoký odpor. Odpor se značí písmenem R a měří se v ohmech [Ω] [11]. Odpor ovlivňuje hned několik faktorů. Prvním faktorem je délka a to tak, že čím větší je délka materiálu, tím větší odpor bude mít. Dále je to průměr materiálu, který je ovšem nepřímo závislí na odporu, takže čím tlustší a širší materiál, tím menší bude jeho odpor, jelikož větší prostor dovolí téct více elektronům. Dalším faktorem je to, jak moc vodivý materiál je. Dobré vodiče mají malý odpor a naproti tomu materiály, které špatně vedou, mají velký odpor. Posledním důležitým faktorem je teplota materiálu. Tato vlastnost některým materiálům dovoluje dynamicky měnit odpor a využívají se pro výrobu elektrických součástek [10].

Velikost odporu materiálu se poté dá vyjádřit matematickým vztahem, jako podíl délky l a průřezu S, vynásobený měrným odporem ρ.

$$R = \rho \cdot \frac{l}{S}$$

Měrný odpor ρ [Ω·m] odpovídá odporu materiálu, jehož délka je jeden metr a průřez je jeden m^2. Závisí na materiálu a na teplotě. Hodnoty odporu pro konkrétní materiály lze najít ve fyzikálních tabulkách (např. měrný odpor hliníku je $\rho = 2{,}82 \cdot 10^{-8}\ \Omega \cdot m$) [12].

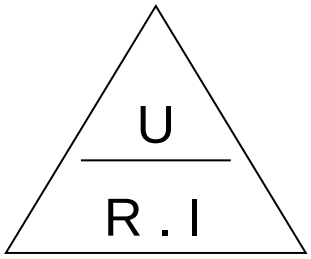
5.2.4 Ohmův zákon

Mezi výše uvedenými veličinami, tj. mezi napětím U, proudem I a odporem R, existuje vztah, který se nazývá Ohmův zákon. Podle něj je napětí přímo úměrné součinu proudu a odporu v obvodu. Je to jedno z nejzákladnějších a nejdůležitějších pravidel v elektrotechnice. Matematicky ho lze zapsat jako [11]:

$$U = I \cdot R \ \wedge \ I = \frac{U}{R} \ \wedge \ R = \frac{U}{I}$$

Tento vztah byl původně popsán pouze pro stejnosměrný proud, ale jeho zápis lze aplikovat i pro proud střídavý s tím, že veličiny U a I budou vyjádřeny jako komplexní čísla a místo odporu R bude použita impedance Z.

Pro snadnější zapamatování vztahu se někdy používá pyramidový zápis (viz Obrázek 7). Z toho lze přikrytím veličiny, kterou chceme počítat, získat její vzorec.



Obrázek 7 Ohmův zákon
Zdroj: vlastní zpracování

5.2.5 Elektrický výkon

Elektrický výkon (Electric power) je veličinou, která udává vykonanou elektrickou práci za jednotku času. Značí se písmenem P a jednotkou je Watt (W). Jeden watt je úměrný vykonané práci za jednu sekundu jedním voltem elektrického napětí při přenosu jednoho coulombu náboje po elektrickém obvodu. Jelikož jeden ampér se dá vyjádřit jako jeden coulomb za sekundu, elektrický výkon lze popsat vztahem $P = U \cdot I$ [10]. Pokud ovšem do rovnice dosadíme podle Ohmova zákona ($U = R \cdot I$), vznikne vztah $P = R \cdot I^2$, tzn. výkon je přímo úměrný odporu a druhé mocnině proudu. Celý vztah lze poté zapsat jako [11]:

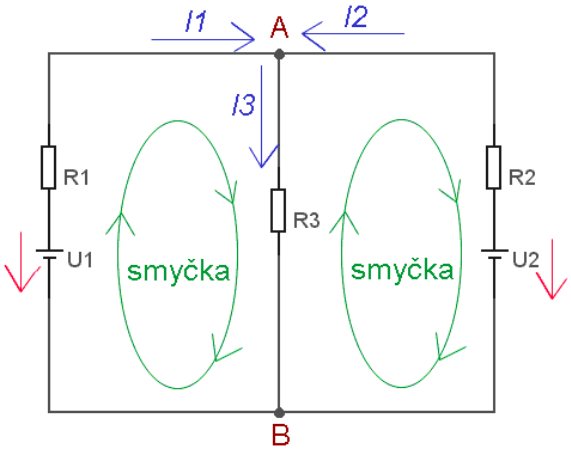
$$P = U \cdot I = \ I^2 \cdot R$$

Elektrický výkon se nejčastěji používá v souvislosti s komponentami odporu (rezistory) a jejich tzv. příkonem. Příkon označuje množství elektrické energie za

jednotku času. Pokud toto množství překročí maximální příkon součástky nebo spotřebiče, dojde k přeměně elektrické energie na teplo, a to může vést až k přehřátí součástky a jejímu poškození [11].

5.2.6 Kirchhoffovy zákony

Jako Kirchhoffovy zákony jsou označovány dvě pravidla o zachování náboje a energie v elektrických obvodech. První zákon se týká elektrického proudu a uzlů, druhý se zabývá smyčkami a napětím. Nejprve je ovšem potřeba uvést, co vlastně pojmy uzel a smyčka označují. Uzel je místo, ve kterém dochází ke spojení tří a více vodičů (viz Obrázek 8, body A a B) a smyčka je část obvodu mezi dvěma uzly [14].



Obrázek 8 Kirchhoffův zákon – uzel a smyčka
Zdroj: [14]

První Kirchhoffův zákon (o proudech, o uzlech)

První Kirchhoffův zákon se týká elektrického proudu a uzlů v obvodu. Konkrétně říká, že v každém uzlu se součet vstupujících proudů rovná součtu proudů, které uzel opouštějí. Matematicky lze vyjádřit pomocí algebraického součtu jednotlivých proudů I_k, přičemž proudy, které tečou do uzlu, jsou chápany jako záporné a proudy tekoucí ven z uzlu jako kladné [12].

$$\sum_{k=1}^{n} I_k = 0$$

Druhý Kirchhoffův zákon (o napětích, o smyčkách)

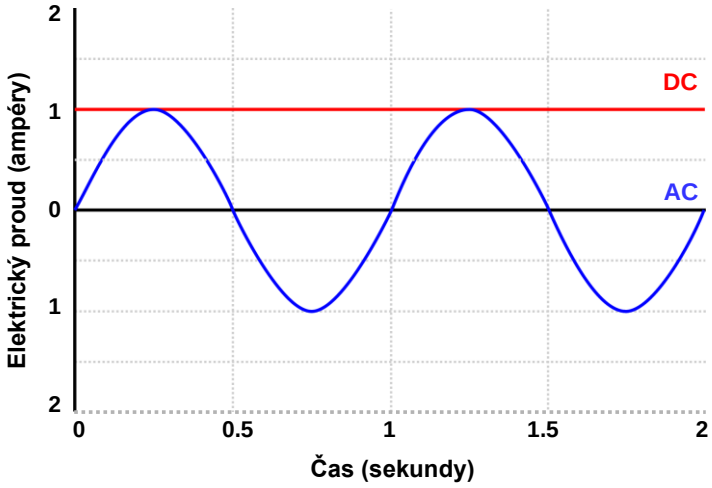
Druhý se Kirchhoffův zákon se zabývá napětím a smyčkami v obvodech. Přibližná definice je, součet napětí na zdrojích je roven součtu napětí na spotřebičích. Matematicky by zákon šlo opět vyjádřit, jako algebraický součet jednotlivých napětí U_k ve smyčkách [12].

$$\sum_{k=1}^{n} U_k = 0$$

5.3 Stejnosměrný a střídavý proud

Elektrický proud může téct dvěma různými směry. Proud, který obvodem teče stále stejným směrem (od plusu k mínusu), se označuje jako stejnosměrný proud (DC – direct current). Ve schématech se stejnosměrný proud označuje symbolem = nebo −. Typickými zdroji DC proudu jsou například galvanické články (tužkové baterie), fotovoltaické články, dynama [11].

Střídavý proud, na rozdíl od stejnosměrného, mění směr toku. Typickým grafickým znázorněním AC proudu je sinusoida, někdy přezdívaný jako harmonický střídavý proud. Křivka sinusoidy začíná v 0, následně roste až do kladného maxima a vrací se zpět do 0, zde se změní polarita (směr), klesá do minima a opět se vrací do 0, kde opět dojde ke změně polarity zpět na kladnou [10]. Tento průběh se neustále periodicky opakuje. Průměrná velikost proudu by po celou dobu měla být rovna 0. Na obrázku je vidět porovnání průběhů stejnosměrného a střídavého proudu [15].



Obrázek 9 Graf průběhu stejnosměrného a střídavého proudu
Zdroj: [15]

5.4 Elektronické součástky

Existuje spousta vlastností, podle kterých můžeme elektronické součástky rozdělit. Mezi kritéria dělení patří například složitost, spotřeba a jejich vliv na signál. Asi nejzákladnější dělení je podle spotřeby, a to na zdroje a spotřebiče. Dále podle složitosti na diskrétní a složené. Diskrétní součástky jsou takové, které mají jednu hlavní funkci a nelze je rozložit. Příkladem jsou například rezistory, potenciometry či LED diody. Složenými potom rozumíme ty, které, jak název napovídá, jsou složeny z více komponent a jejich chování je mnohokrát složitější. Podle posledního kritéria lze prvky rozdělit na pasivní a aktivní. Pasivní prvky do obvodu nevnášejí energii, ale naopak ji přeměňují na energii jiného druhu, tzn. chovají se jako spotřebiče. Aktivní naopak dokážou zvyšovat energii v obvodu díky napájení ze zdroje, tzn. chovají se jako zdroj.

5.4.1 Základní elektronické součástky

Elektronických součástek existuje nepřeberné množství. Pro potřeby této práce budou představeny jen ty nejzákladnější.

Rezistory

Rezistor je diskrétní pasivní součástka, která může mít mnoho barev, tvarů a velikostí pouzdra. Její účel je stále stejný, a to limitovat procházející proud. Dělí se na stálé a proměnné. U pevných rezistorů se hodnota odporu téměř nemění, pouze mírně vzhledem k teplotě a životnosti rezistoru nebo materiálu, ze kterého je vyrobený.

U proměnných rezistorů lze hodnotu odporu měnit. Existuje hned několik variant, například potenciometry a trimry (ty se nejčastěji používají v komerční elektronice k nastavení hlasitosti, světel, teploty) nebo rezistory na bázi senzorů, jako termistory (teplota), varistory (napětí), fotorezistory (světlo), které mění svůj odpor na základě jiné fyzikální veličiny [16].

Každý pevný rezistor má na pouzdru 4 nebo 5 pruhů (v současnosti i 6), které reprezentují takzvaný proužkový kód. Podle tohoto značení lze určit velikost elektrického odporu rezistoru. Co a jaká barva znamená je uvedeno v tabulce (viz Tabulka 1) [10].

Barva	Číslo	Násobitel	Tolerance
Černá	0	1	-
Hnědá	1	10	± 1 %
Červená	2	100	± 2 %
Oranžová	3	1000	-
Žlutá	4	10.000	-
Zelená	5	100.000	± 0.5 %
Modrá	6	1.000.000	± 0.25 %
Fialová	7	10.000.000	± 0.1 %
Šedá	8	-	-
Bílá	9	-	-
Zlatá	-	0.1	± 5 %
Stříbrná	-	0.01	± 10 %
Žádná	-	-	± 20 %

Tabulka 1 Barevný proužkový kód rezistorů

Zdroj: vlastní zpracování (podle https://cs.wikipedia.org/wiki/Rezistor)

Kód se čte zleva doprava a první tři pruhy vyjadřují hodnotu odporu. První pruh je první číslice, druhý pruh druhá číslice, třetí pruh je desítkový násobitel a čtvrtý pruh určuje toleranci. Pokud by rezistor měl 5 proužků, tak první tři určují hodnotu, čtvrtý je násobitel a poslední tolerance.

Kondenzátor

Kondenzátor je druhou základní součástkou v elektronických obvodech. Její hlavní charakteristikou je schopnost uchovat elektrický náboj i pod odpojení od zdroje. Takto nabitý kondenzátor, dokáže poté přenést uchovaný náboj do připojeného spotřebiče. Jeho specifickou vlastností je kapacita C. Ta označuje množství elektrického náboje, které dokáže kondenzátor udržet. Jednotkou kapacity je farad, značený písmenem F. V praxi se nejčastěji vyskytují kapacity v řádech mikrofaradů a pikofaradů. Existují ovšem i kondenzátory, které mají kapacitu ve stovkách faradů [11].

Nejjednodušší kondenzátory si lze představit, jako dvě vodivé destičky, které jsou odděleny vrstvou izolantu, tzv. dielektrikem. Vodivé plochy jsou nabité

nábojem s opačnou polaritou, což vede k tomu, že se začnou navzájem přitahovat. Pokud by mezi deskami došlo ke kontaktu, dojde k neutralizaci náboje, a tím k vybití. Z toho důvodu je mezi nimi umístěné zmíněné dielektrikum, které má za úkol kontaktu zabránit, a tím dojde k uchování náboje na destičkách [10].

Kondenzátory se dělí podle tvaru a podle použitého dielektrika. Tvarově lze rozlišit kondenzátory na deskové, válcové, kulové a svitkové. Podle dielektrika například na vzduchové, vakuové, plastové, papírové, elektrolytické apod.

Cívka

Cívka uzavírá trojici nejzákladnějších elektronických součástek, kterou tvoří spolu s rezistorem a kondenzátorem. Cívka je jednoduše vodič (nejčastěji měděný) navinutý na jádro z izolantu. Vinutí může být jednovrstvé, ale i dvouvrstvé (nutno použít křížové vinutí). Jako součástka má v obvodech dvě základní využití, jako elektromagnet, anebo jako induktor. Cívku v roli elektromagnetu lze najít například v elektromotorech, reproduktorech, v záznamových zařízeních apod [11]. V případě induktoru se o cívce mluví jako o nositeli indukčnosti. Indukčnost je fyzikální veličina, která vyjadřuje schopnost těles vytvářet ve svém okolí magnetické pole. Označuje se písmenem L a její jednotkou je henry, značeno H. V obvodech se magnetické pole využívá k blokování signálů o vyšší frekvenci a k blokování rychlých změn v toku proudu [10]. Cívky nalezneme také v další důležité součástce a to transformátoru.

Transformátor

Transformátor slouží k přenosu elektrické energie mezi dvěma obvody pomocí elektromagnetické indukce. V praxi je to nejdůležitější součástka k dopravě elektrické energie od výrobce ke spotřebiteli. Hlavní funkcí je snížení nebo zvýšení elektrického napětí ze zdroje. Najít ho můžeme například v nabíječce na mobil, kde se podílí na převodu 220 V ze zásuvky na 5 V pro mobilní telefon. Transformátor se typicky skládá ze dvou vinutí, primárního a sekundárního. Do primárního vinutí je přiveden proud ze zdroje a sekundární vinutí je připojeno ke spotřebiči. Obě vinutí jsou spojené jádrem, kterým žádný proud neprochází. To, jestli bude transformátor napětí zvyšovat nebo snižovat, záleží na počtu závitů na obou cívkách. V případě, že

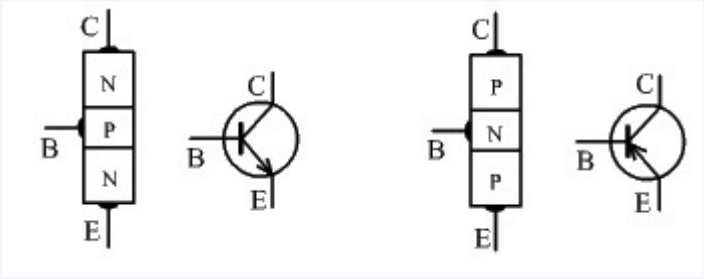
bude na primární cívce více závitů než na sekundární, do spotřebiče půjde menší napětí a naopak. Rozdíl v závitech je úměrný změně napětí [17].

Tranzistor

Tranzistor je polovodičová součástka, která se hojně využívá v číslicová technice ve formě logických hradel. Co ale znamená, že je polovodič? Vlastností polovodiče je, že proud vede jen za určitých podmínek. Znamená to, že může plnit funkci vodiče i izolantu. Důležitým (ne jediným) rozdělením polovodičů je podle nositele náboje na typy P a N. Rozhoduje, jaký prvek v látce zajišťuje vodivost. Jednoduše lze polovodič typu N popsat jako látku, která má v atomové struktuře přebytek elektronů a vodivost tedy zajišťují negativně nabité částice. Podobně je tomu i u typu P, což je látka, která má naopak v atomové struktuře elektronů nedostatek, a proto výsledná vodivost bude zajištěna pozitivně nabitými atomy. Známé polovodičové materiály jsou například Křemík (Si), Germanium (Ge) a Arsenid gallitý (GaAs), které vedou proud velmi špatně, ale dodáním vnější energie (světlo, teplo a další) lze ovlivnit vazby mezi atomy a tím dojde k uvolnění elektronů [18].

Spojením polovodiče typu P a typu N vznikne takzvaná oblast P-N přechodu, na jehož principu jsou založeny polovodičové součástky. Tou nejjednodušší je dioda. Zapojením do obvodu dojde k tomu, že přebytečné elektrony z polovodiče N, které jsou blízko spoje, přeskočí do oblasti P. To zapříčiní v bezprostředním okolí spoje vznik oblasti bez volných částic a ve zbytku obou polovodičů dojde ke změně náboje. Přechodem elektronů dojde k jejich převaze v polovodiči P a získá tak záporný náboj, a naopak polovodič N odchodem elektronů získá kladný náboj. Tím vzniká takzvané difúzní napětí. Připojením kladného napětí na polovodič N a záporného na polovodič P dojde k tomu, že kladné napětí k sobě ještě více přitáhne zbylé volné elektrony v N a záporné napětí k sobě přitáhne přebytečné kladné částice v P. Přechod je v závěrném směru, což znamená, že jeho odpor se blíží nekonečnu. Jiná situace nastane, pokud póly napětí otočíme. Kladné napětí na P začne přitahovat volné elektrony z N a záporně napětí na N začne přitahovat kladné částice z P a tím vznikne vodivý přechod a odpor součástky v obvodu se blíží nule. Můžeme tedy shrnout, že proud P-N přechodem prochází pouze jedním směrem [11].

A konečně již samotné tranzistory. Těch existuje několik typů, např. NPN, PNP, FET, JFET, MOSFET. Není cílem jednotlivé varianty podrobně popsat, budou zmíněny jen základní typy NPN a PNP. Tyto typy tranzistorů jsou praktickou aplikací přechodu P-N jen místo jednoho přechodu mají hned dva (zkratky NPN a PNP označují uspořádání polovodičů) a pouzdro má pak tři vývody. Elektrody nesou označení C – Kolektor, B – Báze a E – Emitor (viz Obrázek 10). Připojením kladného napětí na bázi tranzistoru NPN dojde k otevření a proud teče od kolektoru k emitoru. Pro PNP je obdobné, akorát aby se tranzistor otevřel, musí být na bázi přivedeno napětí záporné a proud teče od emitoru ke kolektoru. Směr, jakým proud tranzistorem protéká, je na schématické značce vyznačen šipkou. V analogových obvodech se tranzistory využívají nejčastěji jako zesilovače. V oblasti logických obvodů vykonávají funkci spínačů, tzn., že hraje roli hlavně to, jestli proud tranzistorem protéká či nikoliv [18].



Obrázek 10 Tranzistory NPN a PNP

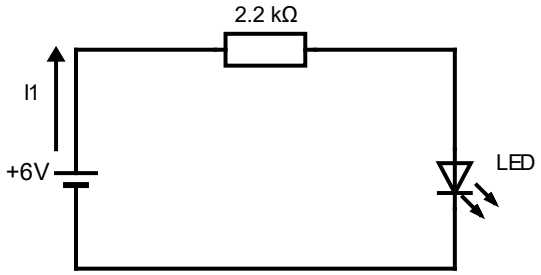
Zdroj: https://www.itnetwork.cz/programovani/nezarazene/elektronika-tranzistory

5.5 Základy elektrických obvodů

Spojením elektrických součástek vzniká tzv. elektrický obvod. Obvod je uzavřené vodivé spojení, které dovoluje elektronům proudit mezi komponenty a vzniká tak elektrický proud. K tomu, aby obvodem mohl téct proud, musí obsahovat zdroj elektrické energie (např. baterie). Druhou podmínkou je, že obvod musí být uzavřený, ovšem ne permanentně. Existují i takzvané otevřené obvody, kde je tok proudu v určitém místě přerušen například pomocí vypínačů (on/off), přes které proud poteče jen ve chvíli jejich sepnutí. Jelikož proud teče cestou nejmenšího odporu, může dojít k tzv. zkratu. Zkrat je nezamýšlené spojení mezi dvěma body, mimo určený obvod, což může způsobit tok příliš velkého proudu a tím pádem

přetížení součástek, což se může projevit přehřátím/roztavením vodičů a součástek a následně selháním spotřebičů, v nejhorším případě i požárem.

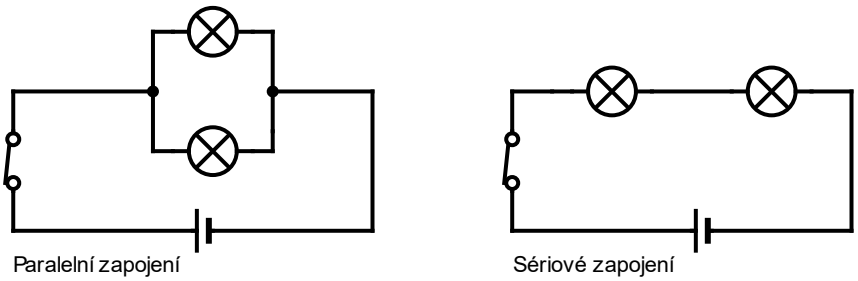
Obvody se graficky reprezentují pomocí schémat, kde má každá součástka vlastní značení. Příklad jednoduchého obvodu se třemi základními součástkami, konkrétně se zdrojem napětí o velikosti 6 V, rezistorem s odporem 2200Ω a LED diodou.



Obrázek 11 Ukázka elektrického obvodu
Zdroj: vlastní zpracování (podle [1])

5.5.1 Sériové a paralelní obvody

V obvodech existují dva způsoby zapojení: sériové a paralelní. V sériovém obvodu jsou součástky zapojeny za sebou na stejné cestě mezi kladným a záporným pólem zdroje. Proud takovým obvodem prochází lineárně všemi komponenty. Z toho plyne, že pokud jedna z komponent přestane fungovat, dojde k přerušení obvodu, čímž dojde k vypnutí všech ostatních prvků. Typickým příkladem sériového zapojení je vánoční osvětlení. V paralelním obvodu dochází k rozdělení toku proudu do dvou a více větví. Každá komponenta čerpá z vlastního toku proudu, z čehož plyne hlavní výhoda paralelního zapojení, že v případě poškození některé ze součástek nedojde k přerušení toku proudu. Příklad obou způsobů zapojení lze vidět na obrázku (viz Obrázek 12):

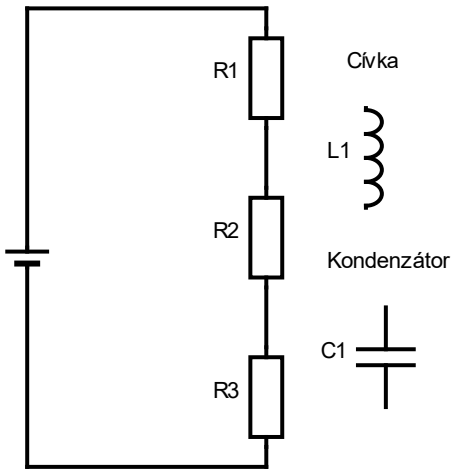


Obrázek 12 Sériové a paralelní zapojení

Zdroj: vlastní zpracovaní (podle [1])

Sériové a paralelní zapojení má největší význam při využívání součástek, jako rezistory, kondenzátory a cívky. Podle typu zapojení se poté liší i výpočet výsledné hodnoty odporu, kapacity a indukce a samozřejmě napětí a proudu. Příklad toho, jak takové zapojení jednotlivých součástek může vypadat a jak se vypočítá výsledná hodnota dané veličiny, lze vidět na obrázcích (viz Obrázek 13 a 14). Pro ukázku jsou v obvodu použity rezistory, ale pro cívky a kondenzátory, je zapojení naprosto identické.

V případě sériového zapojení je výpočet pro odpor a indukčnost stejný. Součet všech hodnot rezistorů nebo cívek se rovná celkovému odporu nebo indukčnosti. Pro celkovou kapacitu je to velmi podobné. Jediným rozdílem je to, že kapacity kondenzátorů umístíme do jmenovatele zlomku s čitatelem 1, je to tedy součet všech hodnot kapacit, umocněných na -1.



Pro výpočet odporu:

$$\mathbf{R_T = R_1 + R_2 + R_3 + . . + R_k}$$

Pro výpočet indukce:

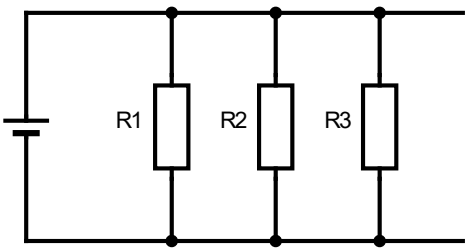
$$\mathbf{L_T = L_1 + L_2 + L_3 + . . + L_k}$$

Pro výpočet kapacity:

$$\mathbf{\frac{1}{C_T} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + . . + \frac{1}{C_k}}$$

Obrázek 13 Sériové zapojení rezistorů
Zdroj: vlastní zpracování (podle [10])

U paralelního zapojení je to v podstatě totéž. Celková hodnota odporu nebo indukce se počítá, jako součet všech hodnot rezistorů nebo cívek umocněných na -1 a naopak kapacita je obyčejný algebraický součet.



Pro výpočet odporu:

$$\frac{\mathbf{1}}{\mathbf{R_T}} = \frac{\mathbf{1}}{\mathbf{R_1}} + \frac{\mathbf{1}}{\mathbf{R_2}} + \frac{\mathbf{1}}{\mathbf{R_3}} + . . + \frac{\mathbf{1}}{\mathbf{R_k}}$$

Pro výpočet indukce:

$$\frac{\mathbf{1}}{\mathbf{L_T}} = \frac{\mathbf{1}}{\mathbf{L_1}} + \frac{\mathbf{1}}{\mathbf{L_2}} + \frac{\mathbf{1}}{\mathbf{L_3}} + . . + \frac{\mathbf{1}}{\mathbf{L_k}}$$

Pro výpočet kapacity

$$\mathbf{C_T} = \mathbf{C_1} + \mathbf{C_2} + \mathbf{C_3} + . . + \mathbf{C_k}$$

Obrázek 14 Paralelní zapojení rezistorů

Zdroj: vlastní zpracování (podle [10])

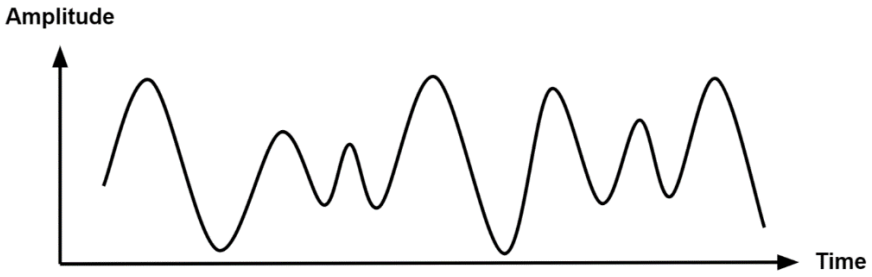
Jedinou výjimku tvoří paralelní zapojení právě dvou rezistoru a cívek. V takovém případě je možné použít jednodušší vzorec pro výpočet celkové hodnoty, a to:

$$R_T = \frac{R_1 \cdot R_2}{R_1 + R_2};\ L_T = \frac{L_1 \cdot L_2}{L_1 + L_2}$$

5.5.2 Analogové a digitální obvody

V praxi se elektrické obvody dělí ještě na analogové, digitální a jejich kombinaci (smíšené obvody). **Analogová elektronika** se zabývá spojitým proměnným signálem. Signálem se rozumí různé množství v čase proměnného napětí nebo proudu. Analogové obvody mohou být definovány, jako spojení elektronických komponent (rezistorů, kondenzátorů apod.) o různé složitosti. Takové obvody dokážou signál zesílit, upravit, ztlumit, rušit, a dokonce konvertovat na digitální signál. V dnešní době je velmi těžké najít obvody, které by byly čistě analogové. I když se může zdát, že analogový signál je pojem minulosti, i dnes přináší některé výhody [19]:

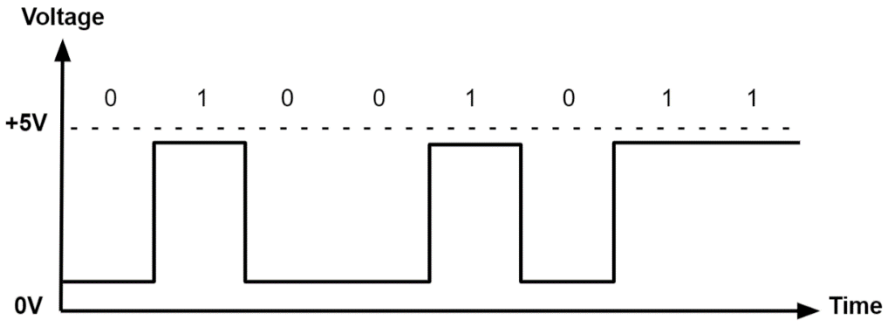
- Jsou jednodušší na zpracování
- Stále jsou nejlepší volbou pro přenos zvuku a obrazu
- Nabízejí daleko větší hustotu a mohou tak poskytovat přesnější informace
- Přesněji zobrazují změny ve fyzickém světě, jako zvuk, světlo, teplo, pozice, tlak
- Analogové systémy jsou méně citlivé z hlediska elektrické tolerance



Obrázek 15 Analogový signál
Zdroj: [19]

Oproti tomu **digitální obvody** pracují pouze s dvěma diskrétními hodnotami, které mohou být reprezentovány jako 1/0, on/off, high/low atd., a které odpovídají velikosti napětí. Většinou používá značení, kdy typicky kladná hodnota napětí odpovídá výrazu binární 1 (popř. on, high) a hodnota blížící se nebo rovna 0 V odpovídá binární 0 (popř. off, low). Hlavní součástkou digitální obvodů jsou tranzistory, které jsou zapojené jako takzvaná logická hradla vykonávající funkce z Booleovy algebry, jako AND, NAND, OR, NOR, XOR a jejich kombinace. Digitální obvody mohou poskytovat logiku a zároveň i paměť (příkladem je paměť RAM). Návrh digitálních obvodů se zcela liší od obvodů analogových, jelikož pracují s binárním signálem, a proto není potřeba řešit zkreslení nebo úbytek napětí. V praxi se hojně využívají i smíšené obvody (kombinace analogového a digitálního), například převodníku signálu (A/D a D/A) a většina moderních rádiových a komunikačních obvodů. Výhody digitálního signálu jsou [19]:

- Méně podléhá hluku, zkreslení a rušení.
- Digitální obvody mohou být reprodukovány masově s nízkými náklady.
- Zpracování digitálního signálu je více flexibilní s využitím programovatelných systému.
- Je bezpečnější, jelikož je snadněji zašifrovatelný.
- Lze ho přenášet na velké vzdálenosti.



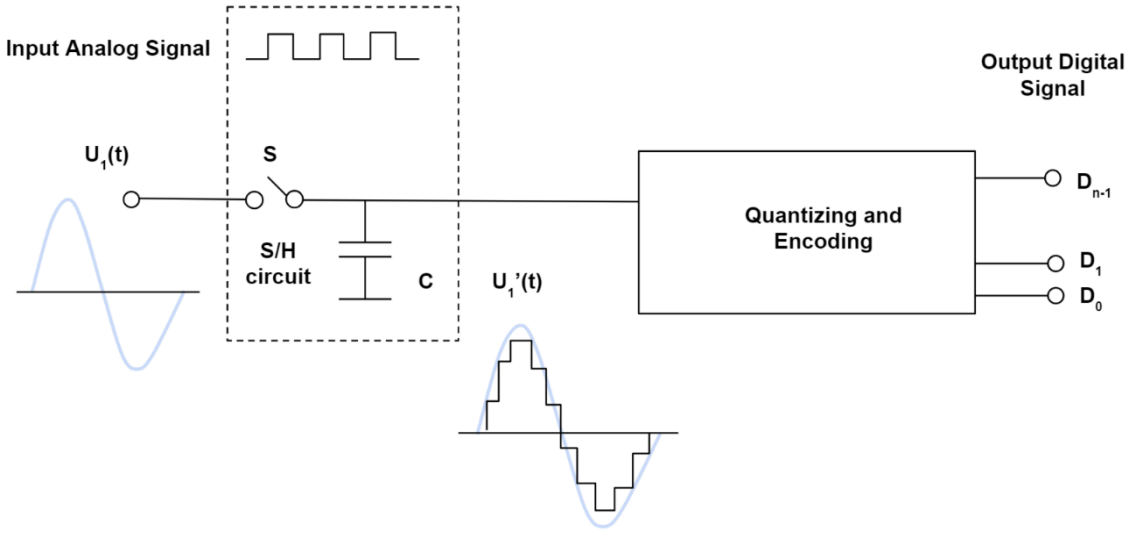
Obrázek 16 Digitální signál
Zdroj: [19]

5.5.3 Převod signálu

Jak bylo zmíněno, většina zařízení a systémů používá jak analogový, tak digitální signál. Taková řešení vyžadují způsob transformace informací mezi oběma typy.

Z analogového na digitální (A/D)

V tomto případě je předpokladem to, že vstupem je analogový signál. Ten prochází skrz „vzorkovač“ (sample-hold), kde se vytvoří odhadovaná digitální reprezentace. Vzorkovač může být tvořen jednoduchým obvodem, tvořeným kondenzátorem a spínačem. Tím vzniknou kvantované diskrétní hodnoty reprezentující vstupní signál [19]. Velikost jednotlivých dílků změny napětí závisí na rozlišení vzorkovače a platí, že čím vyšší tím přesněji dokáže převést analogový signál. Typicky se používají rozlišení převodníků o velikosti 1bit, 2bity, 4bity a 16bit, s tím že 1bit představuje signál jako na předchozím obrázku (viz Obrázek 16). Například pokud by rozlišení převodníku bylo 10bitů (1024 úrovní) a vstupní napětí by bylo 5V, každý dílek by reprezentoval změnu napětí o 4,88 mV (5/1024). Posledním krokem je zakódování digitalizovaného signálu do binárního čísla (velikost odpovídá rozlišení), které reprezentuje amplitudu vstupního signálu [20].

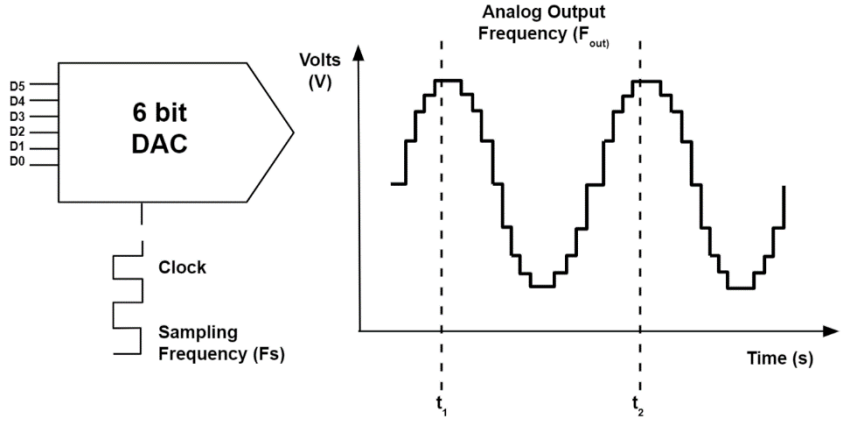


Obrázek 17 Schéma pro A/D převod
Zdroj: [19]

Z digitálního na analogový (D/A)

Proces převodu digitálního signálu na analogový má vlastně opačný průběh. Vstup má tentokrát digitální podobu, tzn. binární hodnotu, a výstupem bude diskrétní hodnota, která je odhadem analogového signálu. Opět platí, že s rozlišením roste i přesnost reprezentace analogového signálu. V případě rozlišení 4,88 mV a vstupního binárního řetězce $(0000011111)_2 = (31)_{10}$ by výsledné napětí dosahovalo hodnoty 31·0,00488 = 0,15 V. Celý proces je trochu složitější a nebude tu podrobně popsán. Často se v zařízeních pro D/A převod používá filtr, který výstupní analogový signál ještě více vyhladí. Jedním typem převodníku D/A je PWM, který bude představen v následujících kapitolách [19].

Time	D5	D4	D3	D2	D1	D0
1/Fs	0	1	1	1	1	1
2/Fs	1	0	1	0	1	0
3/Fs	1	1	1	0	0	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
N-2/Fs	0	1	0	1	0	1
N-1/Fs	0	0	0	1	1	1
N/Fs	1	1	0	1	0	1



Obrázek 18 Schéma pro D/A převod
Zdroj: [19]

5.6 Arduino

Arduino je open-source elektronická platforma zaměřující se na jednoduchost použití hardwaru i softwaru. V posledních letech se produkty značky Arduino staly velmi oblíbenými v oblasti IoT. Platforma je založena na principu komunity, takže na vývoji a inovacích se podílí celá řada uživatelů různých úrovní (studenti, programátoři, profesionálové, hobby uživatelé atd.). Zaměřená je především na studenty a netechnicky orientované lidi s nízkými zkušenostmi s programováním hardwaru. Pro Arduino jsou výjimečné, faktory, jako nízká pořizovací cena, podpora cross platform (Windows, MacOS, Linux), jednoduché programovací prostředí (Arduino IDE), open-source software i hardware s možností rozšíření obojího [21].

Produkty Arduino mají podobu takzvaného jednodeskového počítače (single-board computer – SBC) založeného na mikrokontroleru ATmega. V moderním pojetí se však pojmem mikrokontroler (MCU board) označují celé kompaktní integrované obvody, navržené plnit určitou roli ve složitějších systémech. Základními částmi mikrokontroleru jsou procesor, paměť a rozhraní vstupu a výstupu (I/O). Kromě toho může obsahovat i převodníky analogového signálu na digitální (ADC) a naopak (DAC), sériové porty a další. Důležité je to, že všechny tyto elementy jsou dostupné na jednom čipu/desce. Na produkty Arduino, až na výjimku jakou je Arduino Yún, není možné nainstalovat operační systém. Mikrokontrolery lze najít v automobilech, robotech, zdravotnických zařízeních atd. Kromě Arduina existují ještě alternativy, jako Adafruit, Sparkfun, a velké množství klonů Arduino produktů, jako Seeeduino, FreeDuino [22]. Složitějším a výkonnějším jednodeskovým počítačem je například Raspberry Pi, které bude zmíněno později.

5.6.1 Rozhraní mikrokontrolerů

Předtím, než budou představeny desky Arduino, je potřeba se seznámit s rozhraními, které lze na nich nalézt. Netýká se to pouze produktů Arduino, ale i ostatních mikrokontrolerů/počítačů. Jsou to například [23]:

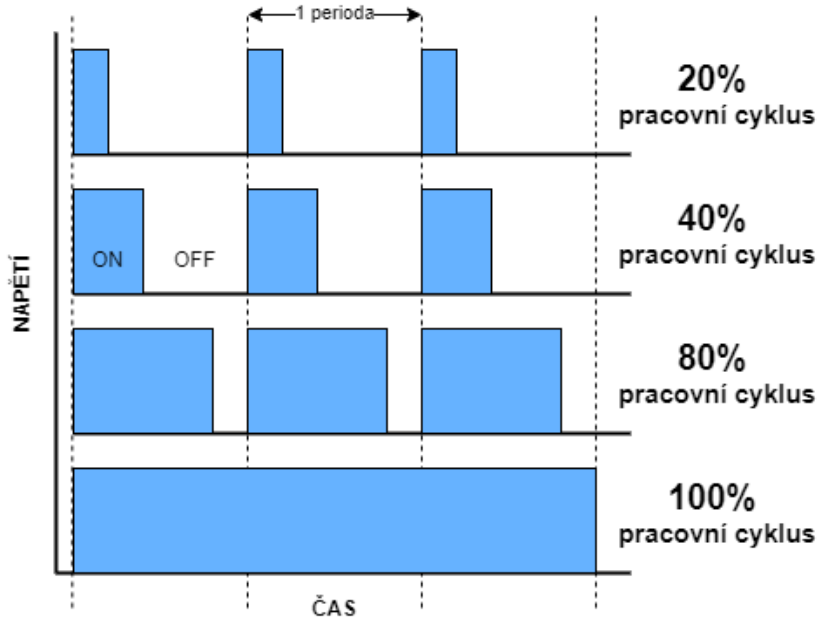
- **GPIO piny** (General-purpose input/output) jsou piny, které slouží jako vstup a zároveň jako výstup. Fungují na principu logické 0 (pro 0 V) a 1 (pro 5 V), ale lze použít i napětí o velikosti 3,3V. V praxi se používají pro připojení senzorů,

rozšiřujících desek, LED, tlačítek atd. Desku je tedy možné napájet buď napětím o velikosti 5 V nebo 3,3 V. To hraje důležitou roli právě u rozšiřujících komponent. Pokud k obvodu, který je napájen 3,3V, připojíme součástku, která je napájena 5 V, dojde k poškození mikrokontroleru.

- **Sběrnice** dovoluje mikrokontroleru propojení s dalšími obvody, jako senzory, moduly GSM, GPS, displeji, anebo pro propojení mikrokontrolerů mezi sebou.
- **A/D a D/A převodníky**, které převádějí vstupní napětí/proud na digitální hodnotu reprezentující velikost dané vstupní veličiny, nebo naopak digitální vstup na výstupní proud/napětí. Různé mikrokontrolery mají různý počet bitů.
- **Univerzální asynchronní přijímač/vysílač** (Universal asynchronous receiver-transmitter – UART) slouží k odesílání nebo přijímání dat. Komunikaci konkrétně obstarávají dva piny s označením TX (transceiver – vysílač) a RX (receiver – přijímač) a jedná se o asynchronní a plně duplexní komunikaci. Aby byla komunikace možná obě zařízení musí mít stejně nastavené parametry jako, přenosovou rychlost, velikost datového slova, paritu, stop bit a řízení toku. Přenosová rychlost se udává v baudech, v tomto případě 1 baud odpovídá rychlosti přenosu 1 bit/s, a pohybuje se od 600 baudů až do 921600 baudů. V praxi se propojení provádí tak, že vysílač prvního zařízení TX1 se připojí na přijímač druhého zařízení RX2 a TX2 se připojí na RX1.
- **Sériové periferní rozhraní** (Serial Peripheral Interface – SPI) slouží k propojení komunikujících prvků. Komunikace probíhá stylem nadřízený (Master) a podřízený (Slave), s tím, že komunikace je plně duplexní, jelikož Master disponuje generátorem hodinového signálu. Sběrnice obsahuje čtyři základní vodiče, a to: SCK (hodiny), MISO (Master in, Slave out), MOSI (Master out, Slave In) a SS (Slave Select). Nevýhodou sběrnice je právě použití hodin. To vyžaduje plnou synchronizace všech zařízení (stejné zpoždění), z čehož vyplývá, že komunikace není možná na velké vzdálenosti. Kromě hodinového signálu je to i neexistence způsobu ověření doručení dat.
- **Sériová sběrnice I2C** (Inter-Integrated Circuit) je sběrnice podobná SPI, ale komunikace na ní probíhá half-duplexem. Sběrnice obsahuje dva vodiče: datový SDA a SCL, který zasílá hodinový signál ze zařízení Master. Oproti SPI je teda jeho použití složitější, a to hlavně z důvodu, že je potřeba na SDA přepínat vstup a

výstup. Každé zařízení, které je na I2C sběrnici připojené, je přiřazeno identifikační číslo o velikosti 7 bitů (do 128 zařízení) nebo 10 bitů (do 1024 zařízení). Podobně jako v počítačových sítích jsou nějaké adresy rezervované, a proto nelze připojit přesně 128 a 1024 zařízení.

- **Sériová sběrnice I2S** (Inter-IC Sound) je sériová sběrnice primárně určená pro propojení digitálních zvukových zařízení. Zajišťuje přenos PCM dat mezi integrovanými obvody. PCM je označení pro pulzně kódovanou modulaci, což by se jednoduše dalo vysvětlit, jako metoda pro převod analogového zvukového signálu na digitální zvukový signál.
- **Digitální pin s funkcí PWM** (Pulse Width Modulation) na desce nahrazuje chybějící piny pro analogový výstup a jsou využívány při použití funkce analogWrite(). Pin dokáže velice rychlým spínáním vytvořit obdélníkový signál, kde periodická délka kladného signálu se poté rovná výstupnímu napětí na daném pinu. Hodnota se počítá podle vzorce $U = \frac{VCC}{2^8} \cdot x$, kde VCC je napájecí napětí (v případě Arduino 5 V), x je hodnota předaná funkcí analogWrite() s tím, že číselný rozsah parametru je 0 až 255. Obrázek 11 znázorňuje, jak takový signál vypadá pro různé hodnoty. Pojem pracovní cyklus (duty cycle) vyjadřuje procentuální část jedné periody, kdy signál dosahoval úrovně sepnutí. Procenta vyjadřují hodnotu předanou ve funkci analogWrite(), což například znamená, že 20 % odpovídá analogWrite(51) a 80 % odpovídá analogWrite(204). Nejlepším příkladem, jak si pracovní cyklus ukázat, je pomocí spínání LED diody. Velikost cyklu se dá reprezentovat pomocí intenzity svícení diody s tím, že se intenzita zvyšuje s velikostí cyklu (a tedy i parametru analogWrite()).



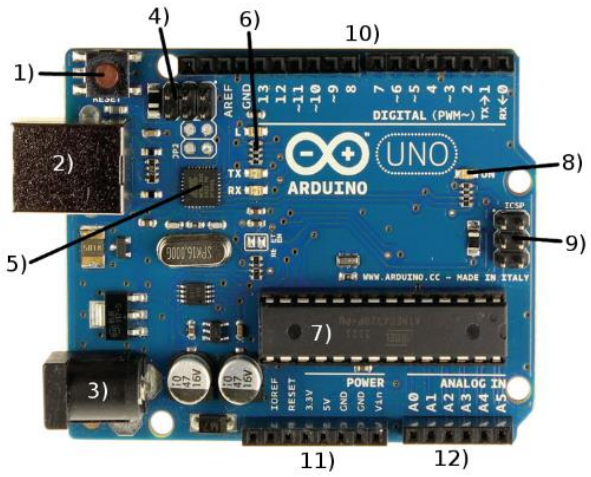
Obrázek 19 Duty cycle PWM pinu

Zdroj: vlastní zpracování (podle https://create.arduino.cc/projecthub/muhammad-aqib/arduino-pwm-tutorial-ae9d71)

5.6.2 Desky Arduino

Arduino nabízí nemalé množství různých modelů a verzí, lišící se hlavně velikostí a výbavou. Většina z nich využívá procesor ATmega od firmy Atmel a typické je pro ně grafické zpracování s dominantní modrou barvou. Příklady konkrétních modelů jsou Mini, Nano, Micro, LilyPad, Fio, Uno, Leonardo, Mega2560, Due a další.

Nejpoužívanější a nejoblíbenější model je Arduino Uno, které bude představeno podrobněji. Kromě výše zmíněných rozhraní obsahuje již klasické USB a případně lze funkcionalitu rozšířit pomocí takzvaných Arduino shieldů. Z modelu UNO se později vyvinuly další dva modely: Ethernet, který místo USB má konektor pro připojení k síti, a Bluetooth, kde je USB nahrazeno Bluetooth modulem pro bezdrátovou komunikaci. Oba tyto modely se v současné době již nevyrábí a přidání funkcionality je tak možné pouze pomocí shieldů. Stručný přehled toho, co lze na Arduino Uno desce najít (nemusí odpovídat každé desce Uno) [24]:



Obrázek 20 Arduino Uno

Zdroj: https://www.distrelec.cz/cs/deska-mikroovladace-uno-arduino-a000066/p/11038919

1) Resetovací tlačítko, které se používá pro znovuspuštění nahraného programu. U různých modelů bude mít různé umístění, vždy je ale jednoduše identifikovatelné pomocí nápisu RESET.
2) USB konektor typu B. Jiné verze mohou místo USB obsahovat sériový port, micro USB, popřípadě může být zcela nahrazen konektorem Ethernet pro připojení do sítě, nebo modulem Bluetooth.
3) Napájecí konektor lze využít v případě, že pro napájení není použito USB.
4) ICSP pro externí programování USB-sériového převodníku. Využíváno pouze profesionálními uživateli. Některé modely převodník nemají nebo je obsažen v čipu, takže tam tento konektor není.
5) USB Serial převodník má na starosti komunikaci mezi hlavním čipem a PC.
6) LED diody L, RX, TX. L dioda může být použita pro testování místo externí LED, ovšem některé modely ji nemají. Diody RX a TX signalizují probíhající komunikaci přes sériovou linku.
7) Hlavní čip (mikrokontroler) celé desky.
8) LED dioda pro stav ON.
9) Sériové rozhraní ICSP pro programování hlavního čipu.
10) Digitální piny. Piny označené vlnovkou podporují PWM modulaci.
11) Napájecí výstupy Arduina.
12) Analogové vstupy. Lze je využít i jako digitální vstupy a výstupy.

5.7 Prostředí Arduino IDE

Arduino IDE (vývojové prostředí) je open-source software, který umožňuje nahrání kódu do paměti Arduina. Základ IDE odpovídá grafickému výukovému prostředí Processing, kam byly přidány některé funkce, a hlavně podpora programovacího jazyka Wiring (přesněji se jedná o knihovnu jazyka C++). Je možné ho stáhnout přímo z oficiálních stránek Arduina (arduino.cc) a podporuje operační systémy Windows, Linux i MacOS. Program pro Arduino je možné tvořit i ve známých prostředích, jako Visual Studio. Stačí jen nainstalovat příslušný plugin [24].

V případě Arduina se program označuje pojmem skica nebo sketch. Sketch je rozdělen na dvě základní části: void setup() a void loop(). Ještě by se dalo hovořit o třetí části, která odpovídá deklaraci globálních proměnných většinou na začátku skici mimo setup i loop. Void setup() se provádí pouze při prvním spuštění (nebo restartu). Součástí setupu bývají například definice vstupních a výstupních pinů. Funkce Void loop() se provádí až po setupu a již takto definovaná, je procházena cyklicky do nekonečna. To znamená, že po provedení posledního řádku kódu se spustí znovu od začátku. Samotné nahrání programu do paměti Arduina se provádí, buď klávesovou zkratkou Ctrl + U, anebo možností Verify & Upload [25].

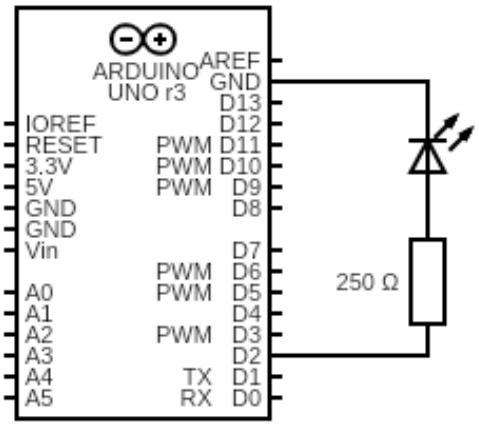
5.7.1 Knihovna Wiring

Jak bylo zmíněno, Wiring je knihovnou programovacího jazyka C++, a proto spousta vlastností a prvků vychází právě z něj. Základy jazyka Wiring, jako datové typy, proměnné, programovací struktury (cykly, pole apod.), zde nejsou zmíněny, jelikož se od ostatních jazyků liší jen zápisem, ale podstata je stále stejná. Za zmínku stojí pouze práce se vstupy a výstupy (piny), která tvoří velmi podstatnou část vývoje pro desky Arduino. Kompletní představení vlastností a prvků jazyka Wiring, včetně ukázek, je možné najít v publikaci od týmu HW Kitchen [24] nebo od Matúše Seleckého [25].

Prostředí IDE a knihovna Wiring nabízejí několik základních funkcí, které slouží k obsluze GPIO pinů. Mezi ty základní patří [25]:

- **pinMode()** – Slouží k nastavení konkrétního pinu na výstup nebo vstup (pin nemůže být definován jako vstup a výstup zároveň). Např.: *pinMode(2, INPUT);* nastaví pin s číslem 2 jako vstup.
- **digitalWrite()** – Pomocí této funkce se digitální pin nastavuje na danou úroveň, a to HIGH (logická 1) nebo LOW (logická 0). Např.: *digitalWrite(4, HIGH);* nastaví pin s číslem 4 na hodnotu HIGH, čímž by došlo například k rozsvícení diody.
- **digitalRead()** – Slouží ke čtení z digitálních pinů pod podmínkou, že čtený pin je v režimu INPUT. Hodnoty lze následně uložit do proměnné.
- **analogReference()** – K nastavení referenčního rozsahu napětí pro analogový vstup (tzn. maximální hodnota vstupního rozsahu). U desky Arduino Uno existují tři možné parametry této funkce: (1) DEFAULT – základní reference pro 5 V nebo 3,3 V, (2) INTERNAL – odpovídá hodnotě 1,1 V nebo 2,56 V v závislosti na procesoru, (3) EXTERNAL – hodnota napětí na pinu AREF (0-5 V). Při použití reference EXTERNAL by dolní hranice neměla být menší než 0 V a horní hranice větší než 5 V, jinak by mohlo dojít k poškození zařízení.
- **analogRead()** – Reprezentuje funkci A/D převodníku. Konkrétně model Arduino Uno disponuje 10bitovým převodníkem, takže referenční hodnotu 5 V rozdělí na 1024 (2^{10}) hodnot (5/1024 = 4,8 mV/dílek). To by znamenalo, že při každé změně napětí o 4,8 mV se změní inkrementuje i hodnota proměnné, reprezentující tento analogový vstup v digitální podobě.
- **analogWrite()** – Funkce sloužící k obsluze digitálních pinů s funkcí PWM, které nahrazují analogový výstup. Fungování PWM bylo popsáno v jedné z předcházejících kapitol (viz 5.7.1 Rozhraní mikrokontroleru).
- **Serial.println()** – K výpisu proměnných nebo řetězců do konzolového okna.

První program



Obrázek 21 Arduino první program
Zdroj: vlastní zpracování (podle [25])

Je dáno schéma, na kterém je vyobrazeno jednoduché zapojení rezistoru a LED diody na piny mikrokontroleru Arduino UNO. Předpokládané napětí zdroje (v tomto případě desky) je 5 V. Při rozhodování, jak velký odpor bude v obvodu zapojen, je nutné si uvědomit následující věci. Maximální proud, který běžnou luminiscenční diodou může protékat je zhruba 10-25 mA, ale pro tuto úlohu bude použita hodnota 20 mA. Druhou věcí je to, že barva diody určuje i úbytek napětí. Pro tuto chvíli bude předpokládaný úbytek 1,7 V. Výpočet odporu by tedy vypadal: (5-1,7)/0,02 = 165 Ω. Z toho plyne, že minimální velikost odporu musí být 165 Ω. Nyní už stačí jen ze schématu vyčíst na jakém portu je obvod připojený a může se přejít ke kódu.

Výstupem je pin s označením D2, a proto první krok bude jeho nastavení na OUTPUT v části void setup(). Pak už následuje samotné rozsvícení LED nastavením pinu na hodnotu HIGH a její zhasnutí hodnotou LOW.

```
int led = 2;    //uložení čísla použitého pinu do proměnné
void setup() {
     pinMode(led, OUTPUT);      //nastavení pinu na výstupní
}
void loop() {
     digitalWrite(led, HIGH);   //rozsvícení LED
```

```
    delay(1000);                //pauza 1000ms
    digitalWrite(led, LOW);     //zhasnutí LED
    delay(1000);                //pauza 1000ms

}
```

6 Softwarové vybavení

Informace v kapitole Softwarové vybavení jsou převzaty z online kurzu Cisco [1], z části Chapter 3: Software is Everywhere, pokud není uvedeno jinak.

V předchozích kapitolách byly představeny komponenty, které nám umožní propojit fyzické prostředí s výpočetním systémem (senzory a elektronická zařízení) a následný sběr dat. Takto získaná data se dále zpracovávají, lze je uložit, přenést nebo zpracovat. A právě zpracování dat má na starosti software nebo program. Pojem program zpravidla označuje nějakou posloupnost instrukcí (algoritmus), která má pomoci ke splnění daného úkol. Algoritmus se nemusí týkat pouze počítačových programů, ale i každodenních činností. Příkladem je například recept na vaření, pracovní postup, návody apod.

Konkrétně počítačový program je tvořen programátorem, který vytváří zmíněný algoritmus, v programovacím jazyce. Pojem software je potom obecné označení pro programové vybavení počítače, ale v dnešní době se to netýká už jen počítačů, ale prakticky každého zařízení (pračka, lednice, konvice, trouba, auto atd.). Podle zařízení a úlohy programu se poté rozdělují do několika skupin např.: operační systémy, firmware a aplikace. Operační systém je program, který zajišťuje komunikaci uživatele s počítačem (Windows, Linux, iOS). Pojmem Firmware se označují programy, které vykonávají specifickou činnost. Najdeme je v široké škále zařízení, např. v pevných discích, chytrých hodinkách, systémech automobilů (ABS) a samozřejmě v IoT zařízeních. A poslední skupinou jsou aplikace, jejichž účelem je pomoci uživateli s vykonáváním konkrétního úkonu (Word, PowerPoint, Mozilla, Visual Studio atd.).

Skutečnost, že v dnešním světě je téměř každé zařízení řízeno nějakým programem, poptávka po lidech, kteří tyto programy dokážou tvořit, stále roste. Ať už se jedná o programátory v oblasti firmwaru, mobilních aplikací, ovladačů zařízení

nebo webů. S rozvojem IoT se tato poptávka ještě zvýšila, jelikož tato technologie nabízí nové příležitosti a tím pádem zcela nová zaměření.

Počítačový program je tvořen výrazy, funkcemi a logickými strukturami. Výrazy mohou mít formu proměnné, konstanty nebo operátorů. Funkce jsou definované posloupnosti, které se aplikují na přijímané parametry, vracející hodnotu získanou provedením této funkce. Typickým příkladem prvků, které lze najít téměř v každém kódu programu je struktura IF-THEN, která větví program na část TRUE a FALSE v závislosti na definované podmínce. Dále jsou to cykly typu FOR a WHILE. Cyklus znamená, že se danou částí kódu projde tolikrát, dokud nebude platit ukončovací podmínka. U cyklu FOR je počet opakování předem známý, zatímco u WHILE se sada instrukcí zpravidla vykonává tak dlouho, dokud výraz v podmínce je TRUE. Tyto struktury jsou v každém programovacím jazyce velmi podobné. To, co se liší, je jejich zápis. Proto je důležité před psaním kódu se seznámit s takzvanou syntaxí, což v programování označuje souhrn pravidel pro zápis kódu v daném jazyce.

Počítačové programy (i programovací jazyky) je možné rozdělit do dvou hlavních kategorií: interprety a kompilátory. Interpret je program, který spouští instrukce přímo v napsaném jazyce, bez překladu na strojový kód. Nevýhodou je, že interpretovaný program běží pomaleji než program kompilovaný, ovšem na druhou stranu je rychlejší kód interpretovat než kompilovat [1]. Interpret používá tři různé typy spouštění kódu. První typ je, že spouští přímo zdrojový kód, v druhém případě překládá zdrojový kód do efektivnějšího mezikódu a následně spustí a poslední typ je, že přímo spustí předem předkompilovaný mezikód. Například programovací jazyk BASIC je interpretem 1. typu, Python 2. typu, Java 3. typu. Obecné výhody interpretace jsou rychlejší vývoj, jednoduchost, kompatibilita, laditelnost [26].

Oproti tomu kompilátor překládá kód napsaný v jednom jazyce do jiného (chtěného) před samotným spuštěním. Kompilátory typicky překládají kód psaný v jazyce vyšší úrovně (zdrojový kód) do jazyka nižší úrovně, jako assembler nebo strojový kód [26]. Kompilovaný kód se typicky provede rychleji, jelikož běží přímo na procesoru. Každou změnu je nutné provádět ve zdrojovém kódu. Typickým příkladem kompilovaných jazyků jsou C a C++.

6.1 Programovací jazyky

Podobně, jako v lidské řeči, i v programování existuje velké množství programovacích jazyků. Některé skupiny programovacích jazyků jsou lepší než ostatní v určitém typu úloh. Například Javascript je velmi oblíbeným jazykem pro tvorbu webových aplikací a C, C# a C++ jsou jazyky hojně používané při tvorbě komplexních programů (např. Linux je napsán v jazyce C).

Konkrétně pro oblast IoT je v současnosti nejpopulárnějším jazykem Python. Jeho přední výhodou je jednoduchost a univerzalita v poměru s tím, co dokáže. Python podporuje procedurálně, objektově i funkcionálně orientované programovací postupy. Primárním cílem Pythonu ovšem není excelovat v nějaké oblasti, ale pomoci programátorům psát kód efektivněji a rychleji, což je jeden z důvodů proč často bývá doporučován jako jazyk, který je dobré se naučit první. V této práci mu bude věnována větší pozornost v následujících kapitolách.

Dalším jazykem, který bude pro zajímavost představen je Blockly. Od ostatních jazyků se liší tím, že nevyžaduje psaní kódu, ale skládání bloků s předem určenou funkcionalitou. Z toho důvodu se Blockly označuje jako vizuální programovací jazyk. Obsahuje bloky, které reprezentují základní operace (podmínky, cykly atd.). Pro náročnější uživatele nabízí možnost si nové bloky naprogramovat a vizualizovat v takzvané *Block factory*. Blockly se spouští ve webovém prohlížeči a ze vzniklého programu, dokáže vygenerovat kód v jazyce JavaScript, PHP a Python.

6.2 Programy v IoT

Běžným využitím IoT systémů je sběr dat. Tato data ovšem pochází z mnoha různých zdrojů a často jsou různých formátů. Proto je důležité data zpracovat a očistit, než budou uložena a použita. Zpracována mohou být, buď na místě sběru a poté poslána, nebo poslána a zpracována na jednom místě i s daty z ostatních zdrojů. Právě počítačové programy mají zjednodušit a zefektivnit proces získávání, zpracování, ukládání a využívání dat.

Další důležitou funkcí IoT systému je učinit rozhodnutí. Spousta IoT zařízení je navržena za účelem vykonaní určité činnosti při splnění podmínek. Například odemčení dveří při přiložení klíče, regulace teploty v místnosti v závislosti na

současné teplotě, varování při hrozícím nebezpečí apod. Funkce učinit rozhodnutí může být do zařízení přidána právě nahráním napsaného programu.

Aby IoT systém mohl fungovat je potřeba zajistit i komunikaci mezi programy, s webem nebo se zařízeními. Tuto komunikaci zajišťuje takzvané API (Application Programming Interface). API je rozhraní, které nabízí možné interakce se softwarovou komponentou. Jednoduše by se dalo říct, že jsou to informace o tom co, jak a za jakých podmínek daná komponenta dokáže [27]. Díky tomu mohou aplikace mezi sebou komunikovat, sdílet data a nabízet služby. API existují v několika podobách, a to například API operačního systému, aplikační API a webová API. IoT aplikace toto využívají ke komunikaci mezi sebou nebo ke komunikaci se službami umístěnými v cloudu.

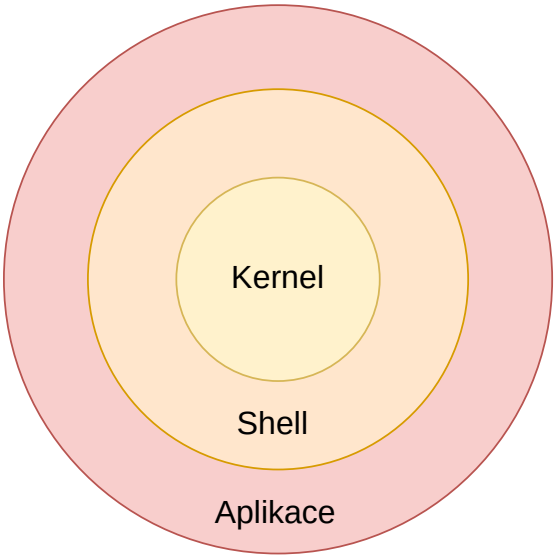
API, které ulehčuje komunikaci přes internet je tzv. REST API (Representational State Transfer). Komunikace v REST je založena na protokolu HTTP. Dříve se webové zdroje označovaly exkluzivně pomocí URL, což dnes již neplatí a každá entita může být pojmenována, adresována nebo označena pomocí ID. Takovou entitou mohou být i data z různých aplikací nebo nastavení různých zařízení (nastavení teploty v domě, srdeční tep naměřený chytrými hodinkami atd.). Každý takový element má nyní přiřazený URI (Unique Resource Identifier) a pomocí těchto identifikátorů poté API žádá konkrétní data. Jelikož je využit protokol http, je možné použít i jeho operace, jako GET, POST, PUT, DELETE. IoT systémy ve většině případů spoléhají právě na REST API pro komunikaci s cloudovými službami.

6.3 Linux

Linux je operační systém vytvořený komunitou programátorů v roce 1991. Systém je zcela pod open source licencí, vysoce upravitelný, díky čemuž ho najdeme na všech typech zařízení od hodinek až po superpočítače. Ke svému fungování nevyžaduje velký výkon hardwaru, a proto se stal i velmi oblíbenou volbou v oblasti IoT. Jeho však asi největší výhodou je to, že je navržený na to, aby fungoval na síti, a proto je velmi často používán pro správu sítí. Díky tomu, že naprosto celý kód Linuxu je dostupný komukoliv, uživatelé ho můžou upravovat, rozšiřovat, zmenšovat podle svého, a to dalo za vznik takzvaným distribucím Linux. Distribuce označují různé balíčky nebo „verze“ vytvořené uživateli na jádru Linux a dostupné opět zdarma.

Příklady známých distribucí jsou Debian, Ubuntu, Red Hat, Slackware a pro následující kapitoly důležitý Raspbian.

Strukturu Linux lze rozdělit na tři základní části: Kernel, Shell a Aplikace (viz Obrázek 22). Kernel je srdcem operačního systému a řídí veškeré operace. Zároveň slouží jako prostředník pro komunikaci shellu s hardwarem. Shell (terminál, konzole, CLI) funguje jako překladatel příkazů. Jedná se tedy o rozhraní, které uživateli dovoluje interagovat se systémem a předávat mu příkazy. Shell se spouští ihned po úspěšném přihlášení uživatele do systému. I přesto, že nové distribuce Linuxu nabízejí velmi pěkné a přívětivé grafické rozhraní, shell, jakožto textově založené rozhraní, je stále velmi populární nástroj pro práci se systémem. Celá interakce by se tedy dala popsat jako uživatel-shell-kernel.



Obrázek 22 Struktura systému Linux
Zdroj: vlastní zpracování (podle https://beginnersforum.net/blog/2014/11/08/structure-of-linux/)

Souborový systém FHS

Důležitou věcí pro práci se systémy založenými na Linuxu je pochopit strukturu jejich souborového systému označovaného jako FHS (Filesystem Hierarchy Standard). Ten definuje základní hierarchii adresářů a jejich obsah. FHS se řídí třemi základními zásadami. (1) Kořenový adresář má být co nejmenší, aby bylo možné systém spustit i z diskety. (2) Statické části systému souborů by měli být oddělený od dynamických. (3) Části obsahující soubory pro jediný počítač, soubory sdílené se skupinou počítačů se stejnou architekturou a s rozdílnou architekturou by měli být

odděleny, aby mohly být sdíleny po síti. Souborového systému FHS se drží většina distribucí Linux, ale některé mají strukturu pozměněnou. Nejvýše je v hierarchii umístěn tzv. kořenový adresář (/root). Ten obsahuje všechny systémové i uživatelské adresáře a soubory, a to i v případě, že jsou umístěné na jiném fyzickém nebo virtuálním zařízení. K zápisu do adresáře je potřeba oprávnění root uživatele (někdy nazývaný jako superuser) [28]. O úroveň níže se nacházejí například adresáře [29]:

- **/bin –** Obsahuje základní uživatelské příkazy.
- **/boot –** Bootovací soubory s nastavením spouštění systému.
- **/dev –** Soubory připojených zařízení.
- **/etc –** Konfigurační soubory používané všemi programy.
- **/home –** Domovský adresář uživatele, obsahující uložené soubory, osobní nastavení apod.
- **/lib –** Systémové knihovny.
- **/media –** Dočasné adresáře pro odnímatelná zařízení (CD, DVD apod.)
- **/mnt –** Adresář používaný k dočasnému namapování jiného souborového systému umístěného na nějaké úložném médiu.
- **/opt –** Adresář pro uživatelské aplikace.
- **/sbin –** Podobně jako /bin obsahuje uživatelské příkazy. Ovšem příkazy v této složce jsou většinou dostupné pouze správci a jiným pověřeným osobám.
- **/srv –** Síťové soubory pro protokoly ftp, www, cvs apod.
- **/tmp –** Dočasné soubory, které se mažou s každým vypnutím systému.
- **/usr –** Sekundární souborový systém (hierarchie) pro uživatelská data.
- **/proc –** Virtuální souborový systém obsahující informace o kernelu a procesech.

Základní příkazy v Linux

Díky GUI v moderních distribucích je pohyb mezi adresáři velmi ulehčen, ale znalost základních příkazů v shellu může přijít vhod v některých situacích. Předností shellu je i to, že nabízí lepší zobrazení struktury souborového systému, a proto je nejlepší se s ní seznámit prostřednictvím této cesty. K opravdu těm nejzákladnějším textovým příkazům pro práci se systémy Linux patří [28]:

- *man* – zobrazí dokumentaci k danému příkazu
- *cd* – přesune uživatele zpět do je domovského adresáře v /home
- *cd adresář* – přesune uživatele do parametrem specifikovaného adresáře
- *cd ..* – vrátí uživatele o úroveň výše
- *pwd* – vrátí aktuální adresář, ve kterém se uživatel nachází (print working directory)
- *ls adresář* – vypíše obsah parametrem určeného adresáře
- *ls* – vypíše obsah aktuálního adresáře
- *mkdir název adresáře* – vytvoří nový se zadaným názvem
- *rmdir název adresáře* – smaže adresář s odpovídajícím názvem
- *rm název souboru* – smaže zvolený soubor
- *mv* – slouží k přesunu (*mv soubor cíl*) nebo k přejmenování souboru (*mv název název*)
- *cp zdroj cíl* – zkopíruje zdrojový soubor do cílového adresáře
- *grep řetězec* – používá se pro vyhledání v parametru předaného řetězce znaků v adresáři
- *ipconfig* – k zobrazení aktuální konfigurace sítě
- *iwconfig* – zobrazí aktuální informace o konfiguraci bezdrátové sítě
- *apt-get* – k získání, nastavení nebo smazání systémových balíčků

Kromě těchto výše zmíněných ještě stojí za zmínku tři příkazy týkající se správy běžících procesů. Proces označuje právě spuštěný program, kterému systém přidělil paměť a další prostředky. Každý proces je označený Process ID (PID), které kernel využívá k identifikaci. K zobrazení spuštěných procesů slouží příkazy *ps* a *top*. Rozdíl mezi nimi je v tom, že *ps* zobrazí statický seznam spuštěných procesů v čase vykonání příkazu a *top* vrátí dynamický seznam. K ukončení příkazu *top* stačí napsat *q*. K ovlivnění běžících procesů je potom příkaz *kill.* Název už napovídá, k čemu zřejmě slouží. *Kill* ovšem kromě ukončení běžícího procesu, dokáže na základě předaných parametrů proces i restartovat nebo pozastavit. K určení cílového procesu se používá právě PID zjištěné příkazy *ps* nebo *top*.

Oprávnění v systémech Linux

Oprávnění přístupu k souborům a složkám jsou v systémech Linux základním prvkem zabezpečení dat. Oprávnění jsou podobně jako v OS Windows členěna na čtení (read), zápis (write), spouštění (execute). Tyto oprávnění se přiřazují třem typům uživatelů, a to uživatel (user), skupina (group) a ostatní (other). V následující tabulce (viz Tabulka 3) je vidět struktura přiděleného oprávnění, kterou dostaneme spuštěním příkazu *ls -l,* který vrátí základní informace. Ve sloupci *Typ* je určen typ zkoumané entity. Znak – znamená, že se jedná o soubor a *d* značí adresář. Poté následuje řetězec šesti znaků odpovídající nastavenému oprávnění. První tři znaky určují oprávnění uživatele, prostřední tři oprávnění skupiny a poslední tři oprávnění pro ostatní.

Typ	Uživatel			Skupina			Ostatní		
- nebo d	r	w	x	r	w	X	r	w	x

Tabulka 2 Struktura souborového oprávnění
Zdroj: vlastní zpracování (podle [1])

Příklad výstupu *ls -l* by pak mohl vypadat následovně:

-rwxrw-r-- 1 rod staff 1108485 Aug 14 7:34 My_Awesome_File

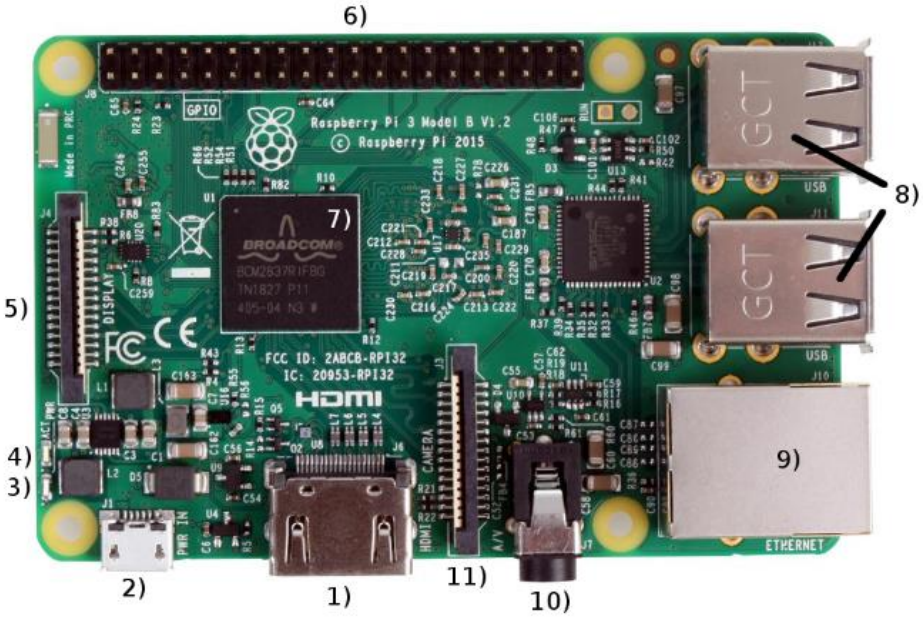
Takovýto výstup sděluje následující informace. Zkoumaná entita je soubor, na kterém jsou nastavené oprávnění čtení, zápis, spouštění pro uživatele (rwx), čtení, zápis pro skupinu (rw-), která soubor vlastní a pouze čtení pro ostatní uživatele (r--). Číslo 1 určuje počet hard linků, což momentálně není podstatné. Další dva sloupce určují jméno uživatele (rod) a jméno skupiny (staff). Následuje velikost souboru v bytech, datum a čas poslední úpravy a název souboru nebo adresáře.

Souborové oprávnění je základní částí Linuxu a nemůže být obejito. Uživatel může v souboru/adresáři dělat jen to, co mu oprávnění dovoluje. Jediný uživatel, který má právo oprávnění přepsat, je root uživatel (tzv. super user), což mu dává plnou kontrolu nad celým systémem.

6.4 Raspberry Pi 3 Model B

Raspberry Pi je dalším populárním zařízením používaným v oblasti IoT a robotiky. Jedná se o malý plnohodnotný počítač na jedné desce (SBC), ke kterému lze připojit další externí periferie (klávesnice, myš, monitor atd.), stejně jako ke klasickému stolnímu PC. Z kompaktnosti vyplývá, že výkonem se desktopům a notebookům nevyrovná, ale i tak spolehlivě vykoná základní funkce. Rozdílem proti Arduinu je to, že RaspberryPi již nabízí plnohodnotné GUI rozhraní díky operačnímu systému a kód je tak možné tvořit přímo na zařízení, kdežto na Arduino se nahrává již hotový program a vykonává poté pouze funkce v něm definované. Oficiálním operačním systémem je Raspbian OS, který vznikl z distribuce Debian a je zcela zdarma. Kromě dalších Linuxových systémů, lze na zařízení nainstalovat i OS od Microsoftu, konkrétně W10 IoT Core. Raspbian je ovšem optimalizovaný přímo pro Raspberry a poskytuje uživatelům přizpůsobené grafické rozhraní (GUI) se základními nástroji. Jako médium pro operační systém a zároveň jako fyzická paměť pro Pi slouží SD karta, s doporučenou velikosti minimálně 8 GB. Lze ale konstatovat, že Raspberry Pi je něco více něž počítač, jelikož umožňuje přístup k hardwaru na desce, a to konkrétně ke GPIO pinům, které slouží k připojení dalších zařízení. Podobně jako Arduino deska obsahuje i SPI, I2C, I2S a UART (viz 5.7.1 Rozhraní) [23].

I Raspberry má několik různých modelů (Pi 1, Pi 2, Zero), ale nejpoužívanějším je Raspberry Pi 3. Pi 3 oproti svému předchůdci podporuje i připojení pomocí WiFi, Bluetooth 4.1 a Bluetooth Low Energy (BLE, viz 7.5.2 Bluetooth LE), což ho dělá skvělou volbou pro IoT systémy. Oproti Arduinu má ovšem stále větší spotřebu, a proto se používá spíše jako řídící jednotka v rámci IoT systému (např. MQTT broker, viz 7.6.2 MQTT). Základní rozhraní, která Pi 3 obsahuje, tedy jsou [23, 30]:



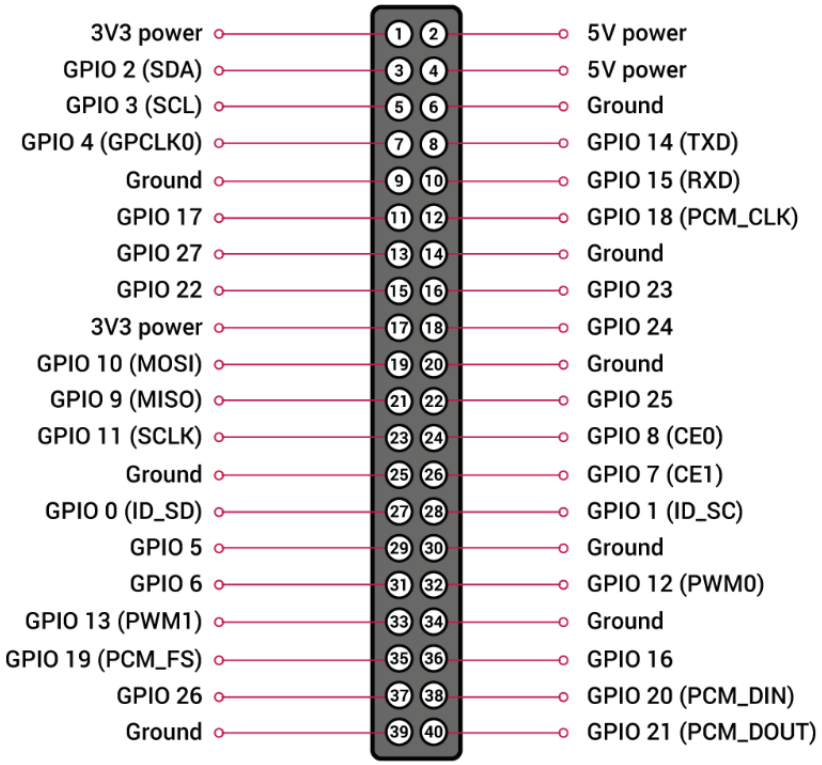
Obrázek 23 Raspberry Pi 3

Zdroj: https://sg.element14.com/raspberry-pi/raspberrypi3-modb-1gb/sbc-raspberry-pi-3-mod-b-1gb-ram/dp/2525226

1) HDMI pro připojení RasPi k monitoru nebo televizi.
2) Micro USB napájecí port.
3) Napájecí LED nepřetržitě svítí červeně pokud je RasPi napájeno. V případě poklesu napětí pod 4,63 V dioda začne blikat.
4) LED ACT PWR je zeleně svítící dioda signalizující činnost SD karty.
5) DSI (Display Serial Interface) je sériové rozhraní pro připojení LCD displeje pomocí 15 pinového plochého kabelu.
6) 40 pinová hlavice obsahující GPIO (vstupně/výstupní) piny.
7) Procesor Broadcom BCM2837.
8) 4x USB porty pro připojení dalších zařízení.
9) Ethernet/Lan port při připojení k síti (10/100Mbps).
10) Port pro obrazový/zvukový výstup.
11) CSI (Camera Serial Interface) pro připojení speciální Pi kamery.

GPIO hlavice Raspberry Pi 3 obsahuje celkem 40 pinů (označeno číslem 6), ale ne všechny jsou uživatelsky programovatelné. Konkrétní rozdělení pinů je znázorněno na obrázku (viz Obrázek 24). Je důležité si neplést fyzické číslování (1-40), které je popořadě a užitkové číslování, které je podle funkce pinu. Kromě

běžných GPIO pinů, jsou zde i piny s funkcí PWM (GPIO12, GPIO13, GPIO18 a GPIO19), ale softwarově lze PWM aktivovat na všech pinech. Dále sériové piny TX a RX, I2C piny SDA a SCL, dvě sady SPI pinů MOSI a MISO a samozřejmě dva 5 V, dva 3 V a 8x Ground, které jako jediné nejsou programovatelné. Grafický náhled rozdělení pinů je dostupný i přímo z OS po zadání příkazu *pinout* v terminálu [31].



Obrázek 24 Raspberry Pi 3 číslování GPIO pinů
Zdroj: [31]

V současnosti již existuje novější verze Raspberry Pi 4, která vyšla v roce 2019. Jedná se spíše o upgrade a update předchozího Pi 3, který žádné zásadní novinky nepřinesl. Raspberry Pi 4 primárně přináší vyšší výkon, Bluetooth verze 5.0, dva porty USB 3.0 (a dva USB 2.0) a dlouho očekávaný Gigabit Ethernet port. Dále došlo ke změně napájecího portu na typ USB-C a podpory připojení dvou monitorů přes mikro HDMI konektory.

6.4.1 PL-App

Cest, jak zařízení Raspberry Pi nakonfigurovat a používat, existuje mnoho. V předcházejících kapitolách bylo zmíněno, že Raspberry Pi je velmi podobné klasickému počítači. Proto jednou z možností je jednoduše připojit monitor (pouze HDMI) a externí periferie, jako klávesnice a myš (pouze USB) a pracovat s ním jako s PC. Ale aby bylo možné použít RasPi jako IoT zařízení, je třeba k němu přistupovat vzdáleně přes síť. V případě RasPi se používá takzvaný „bezhlavý" (headless) přístup. Tento pojem označuje zařízení, které je konfigurované, aby pracovalo bez monitoru (proto bezhlavé), klávesnice a myši a je typicky řízené z jiného PC skrz vzdálený přístup. Tento přístup se často používá pro provoz serverů. Pro vzdálené připojení je možné použít řadu různých nástrojů, ale v tomto kurzu bude využita aplikace zvaná Cisco PL-App. Ta zajistí připojení k RasPi, které tak bude možné spravovat na dálku a bez přítomnosti externích periferií přímo na zařízení. PL-App podporuje pro vzdálený přístup jak GUI, tak textově založené ovládání. Dostupná je ke stažení v odpovídajícím kurzu společnosti Cisco.

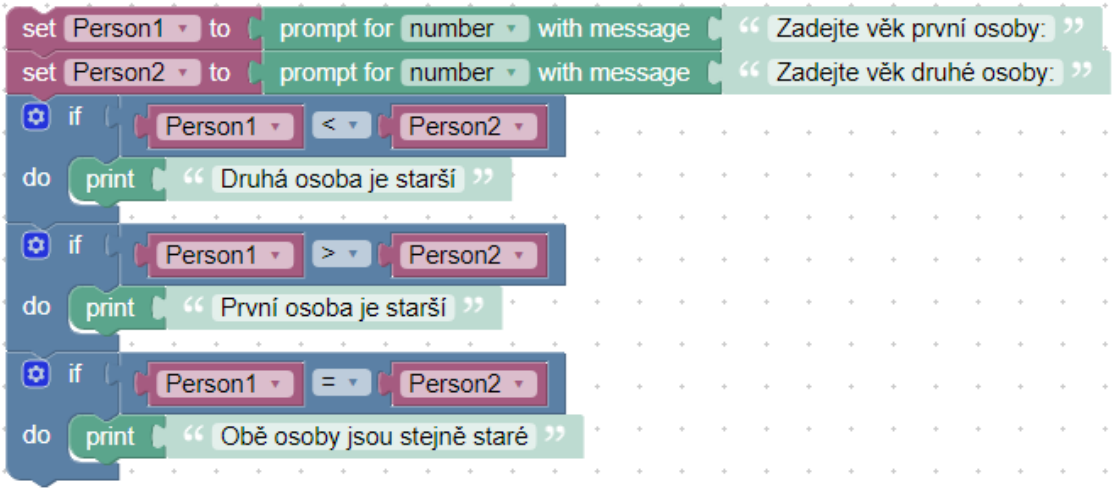
Kromě bezdrátového připojení se dá nástroj použít i k instalaci operačního systému. K tomu, aby bylo možné na RasPi nainstalovat operační systém, je potřeba mít jeho image umístěn na micro SD kartě. Image označuje jediný soubor, který obsahuje přesnou kopii celého systému a může tak být použit prakticky k jeho zkopírování na jiné zařízení. Současné paměťové karty nabízejí dostatečné uložiště, takže nehrozí problém s nedostatkem místa. Běžnými formáty image souboru jsou .img nebo .iso. PL-App lze tedy použít i k přesunu image souboru na SD kartu. Společnost Cisco si pro účely kurzu vytvořilo vlastní upravenou verzi Raspbian s označením PL-App Image (dostupný v kurzu), ale na způsobu instalace to nic nemění.

6.5 Programování Raspberry Pi 3

6.5.1 Blockly

Nástroj Blockly zde bude uveden čistě pro zajímavost, jelikož může začátečníkům velice pomoci s pochopením, jak vlastně takový program funguje a zasvětit je do programování. Blockly je vizuální programovací nástroj, který k tvorbě programu

využívá bloky s předem určenou funkcionalitou. To znamená, že uživatel nemusí napsat jediný řádek kódu. Bloky obsahují volitelné sloty a prázdná místa, ve kterých je možná specifikovat a předat požadované parametry. Samotný proces tvorby se tedy skládá pouze z přetahování a skládání bloků do požadované posloupnosti. V základu jsou v Blockly k dispozici všechny základní struktury, jako cykly, podmínky, funkce a deklarace proměnných. Každý blok má na sobě různá zkosení a výběžky (podobně jako puzzle), což naznačuje možnost připojení dalších bloků a zároveň uživateli pomáhá k tvorbě smyslného programu. Kromě klasický programovacích struktur nabízí i bloky, které jsou specificky navržené pro senzory a aktuátory. Z takto vytvořeného programu dokáže poté vygenerovat kód v několika programovacích jazycích, jako Javascript a hlavně Python. Jak Blockly vypadá, ukazuje následující obrázek (viz Obrázek 25), kde je sestaven jednoduchý program na porovnání stáří dvou osob.



Obrázek 25 Ukázka programu v Blockly
Zdroj: vlastní zpracování

6.5.2 Python

Python je interpretovaný programovací jazyk, takže je potřeba interpret ke spuštění kódu. Kód v pythonu je možné napsat v jakémkoli textovém editoru a následně spustit v jednom z mnoha interpretů, které jsou dostupné pro většinu operačních systémů. To dělá tento jazyk jedním z nejuniverzálnějších, co se platformy týče. Existují i nástroje, Py2exe a Pyinstaller, které dokážou kód zabalit do spustitelného souboru, čímž potřeba interpretu odpadá. Na systémech Linux se python interpret

typicky nachází v /usr/bin/python. Pro jeho spuštění stačí do shellu napsat příkaz *python* (automaticky spustí nejnovější verzi). Že se spuštění povedlo a interpret je v interaktivním módu se pozná podle značky „>>>“ na začátku řádku.

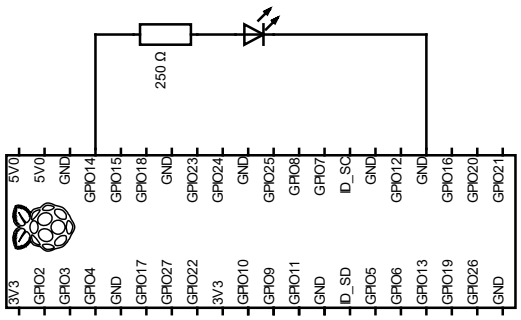
V interaktivním módu lze spouštět jednoduché výrazy tvořené proměnnými, čísly, operátory a i funkcemi. Zajímavostí je speciální proměnná značená “_“ (podtržítkem), která v sobě ukládá výsledek posledního spuštěného výrazu. V tomto jednoduchém prostředí je ale možné tvořit i složitější struktury, jako cykly, podmínky, pole (n-tice, listy, sety, slovníky) apod. Ukončením interpretu dochází ke ztrátě všech proměnných a funkcí, a proto je samozřejmostí, že pro komplexnější programy je nutné využít textový editor. Kód vytvořený v takovém editoru se poté označuje jako skript. Skripty se ale mohou stát velmi složitými a dlouhými a nejspíš je bude nutné rozdělit do menších bloků. Tím vznikne takzvaný hlavní skript, do kterého je poté možné ostatní části (moduly) naimportovat. Moduly mohou být funkce, které plní určitou úlohu nebo celé prvky výsledného programu. Vytvořením těchto modulů se program stává nejen přehlednějším, ale navíc jsou moduly snadno znovupoužitelné v jiných skriptech. V případě IoT jsou velmi typické moduly pro zpracování dat a self-testy (úroveň baterie apod.), bezpečnostní moduly (hashe, šifry) a server-klient moduly (FTP, NTP apod.).

Jazyk Python opět disponuje obecně známými prvky a strukturami, a proto zde nebudou popsány. Navíc je, oproti Wiringu, daleko komplexnější kvůli velkému počtu knihoven, které se i pro stejné problémy liší v závislosti na situaci. Opět je tak možné využít široký výběr literatury nebo online materiálů a kurzů nejen čistě pro Python, ale také pro práci s RaspberryPi. Pro zajímavost je tu uvedena pouze knihovna pro ovládání GPIO pinů dostupných na deskách RaspberryPi.

Python knihovna RPi.GPIO

Funkce, se kterými je možné přistupovat ke GPIO pinům na desce, jsou součástí knihovny RPi.GPIO. Ta obsahuje příkazy velmi podobné jazyku Wiring. První a velmi důležitou věcí je výběr způsobu číslování pinů. Knihovna nabízí dva režimy: BOARD a BCM. BOARD reprezentuje fyzické značení pinů a bude k nim možná přistupovat pomocí čísel 1 až 40. Výhodou toho číslování je jednodušší orientace a větší pravděpodobnost, že skript bude fungovat i na dalších revizích

RasPi. Oproti tomu BCM reprezentuje číslování s příznakem GPIO (viz Obrázek 24). To znamená, že například v režimu BOARD by se pin s číslem 36 rovnal pinu číslo 16 (GPIO16) v režimu BCM. Nastavení režimu se pythonu provádí funkcí GPIO.setmode(GPIO.BOARD nebo GPIO.BCM). Dále je potřeba, stejně jako u Arduina, nastavit piny jako výstup nebo vstup. K tomu slouží funkce GPIO.setup(pin, IN/OUT). Z takto nakonfigurovaných pinů je poté možné číst, funkcí GPIO.input(pin), nebo jim předat požadovaný stav pomocí GPIO.output(pin, stav). Stav může být zapsán jako 0/1, false/true a GPIO.LOW/GPIO.HIGH. Na konci každého programu je doporučeno „uklidit" příkazem GPIO.cleanup(). Ten smaže veškerou konfiguraci nastavenou programem (režimy pinů, režim číslování, konzoly apod.). Těchto pár znalostí stačí již k realizaci prvního jednoduchého programu. Zadání bude stejné, jako u prvního programu na Arduino, to znamená zapojení a rozsvícení LED diody. Ještě předtím je ale potřeba nainstalovat danou knihovnu RPi.GPIO. K tomu bude využit Package installer for Python, ve zkratce pip. Instalace se na systému Raspbian provede příkazem sudo apt-get install python-dev python-pip. Instalace balíčku je obdobná, a to příkazem pip install RPi.GPIO. Nyní již k samotnému programu [32]:



Obrázek 26 Raspberry Pi první program
Zdroj: vlastní zpracování

```
#import modulů RPi.GPIO a time do programu
import RPi.GPIO as GPIO
import time

LED = 8                          #vytvoření proměnné s číslem pinu
GPIO.setmode(GPIO.BOARD)         #nastavení režimu číslování na BOARD
GPIO.setup(LED, GPIO.OUT)        #nastavení pinu na výstup
GPIO.output(LED, GPIO.HIGH)      #rozsvícení LED
Time.sleep(10)                   #čekat 10 vteřin
GPIO.output(LED, GPIO.LOW)       #zhasnutí LED
GPIO.cleaneup()                  #vyčistění konfigurací
```

I když je Raspberry Pi vcelku výkonný a flexibilní mikropočítač, existují aplikace, na které se nehodí. Jeho hlavní nevýhodou je absence analogových pinů a tím pádem na něj nelze připojit analogové senzory. Často se tento problém řeší předřazením Arduino mikrokontroleru, který tyto piny má a je na něj možné senzory připojit. Jinými slovy, jelikož je Arduino možné ovládat z klasického PC, je to možné realizovat i z RasPi, které pak může využít jeho analogové piny. Jediná nevýhoda toho spojení je, že sketche psané pro Arduino jsou často na bázi jazyka C a skripty běžící na RasPi v jazyce Python. Propojení těchto dvou zařízení lze zrealizovat obyčejným USB kabelem (sériově). V této kooperaci může Arduino fungovat jako sběrač dat, která následně zašle na RasPi, kde dojde k jejich zpracování. Dalším běžným využitím je tvorba programů na RasPi a jejich testováni na Arduino. Až v době, kdy program funguje, jak má, se Arduino umístí na zamýšlené místo.

7 Počítačové sítě, Fog a Cloud computing

Informace v kapitole Počítačové sítě, Fog a Cloud Computing jsou převzaty z online kurzu Cisco [1], z části Chapter 4: Networks, Fog and Cloud Computing, pokud není uvedeno jinak.

Stěžejní věcí pro fungování IoT systému je existence počítačových sítí. Účel počítačové sítě je zajištění komunikace mezi připojenými zařízeními. Sítě se typicky rozdělují na LAN (Local Area Network) a WAN (Wide Area Network). Ovšem s rozvojem IoT se objevily nové pojmy, a to LPWAN (Low Power Wide Area Network) a PAN (Personal Area Network). Počítačová síť obsahuje tři hlavní části: zařízení, médium, služby. Zařízení a médium jsou fyzické prvky sítě a představují počítače, switche, routery, access pointy a kabely, kterými jsou zařízení propojené. Služby označují emailové hostingy a webové hostingy.

LAN a PAN typicky označují síť, která se geograficky nachází v malé oblasti. LAN se často vyskytuje v domovech, školách, firmách a kancelářích. Často je spravována jedním člověkem nebo organizací a veškeré bezpečnostní politiky platí na úrovni pouze této sítě. Poměrně novou aplikací LAN je spojení průmyslových strojů v továrnách, v souvislosti s pojmem Industrial Internet. Tento pojem označuje zařízení, které jsou schopné pracovat v průmyslovém prostředí, jehož

charakteristickou vlastností je velká úroveň prachu, hluku, teploty a vibrací. PAN je typem lokální sítě, která označuje blízké okolí člověka a užití bezdrátové technologie. Příkladem PAN sítě je propojení mobilního telefonu s chytrými hodinkami nebo se sluchátky pomocí Bluetooth.

Pod pojmem WAN se rozumí síť, která se rozprostírá na geograficky velkém území, často celá města, územní celky, státy, a dokonce i kontinenty. Často je vlastní velké organizace nebo poskytovatelé internetových služeb. Tradiční WAN je v drtivé většině propojen kabely a připojen na síť elektrické energie. Ovšem IoT WAN často využívá ke komunikaci bezdrátové technologie a zařízení jsou napájeny pomocí baterií, a proto se začaly označovat pojmem LPWAN.

Síťová zařízení a média

Síťová zařízení jsou zařízení, která mezi sebou komunikují skrz počítačovou síť. Mezi ně patří takzvaná koncová zařízení (počítače, mobily, tiskárny atd.). V prostředí IoT sítě to může být téměř cokoliv (pračky, myčky, lednice, topení, světla, zámky atd.). Koncové zařízení může být, buď zdrojem zprávy, nebo jejím příjemcem. K tomu, aby bylo možné zařízení v síti rozlišit, slouží jedinečná síťová adresa. Dalším využitím jsou zprostředkovatelská (intermediary) zařízení. Ta mají na starost připojení jednotlivých koncových zařízení do sítě a obstarávají tok dat v síti. Příkladem takových zařízení jsou gatewaye, routery, switche, access pointy.

Komunikace skrz síť je vedena po médiích. Typicky se využívají tři typy médií: metalické kabely, skleněná nebo plastová vlákna a bezdrátová komunikace. Každé médium má své vlastnosti a charakteristiky, a ne vždycky se všechna média dají využít v dané situaci. Například pro propojení senzorů, se moc nehodí použít kabely, ale existují řešení, která takovou aplikaci vyžadují. Pokud by senzory byly propojeny bezdrátově je možné dosáhnout přenosové vzdálenosti od několika centimetrů do několika kilometrů, ovšem v závislosti na prostředí, kde rozhoduje více faktorů. Výhodou je ovšem nízká spotřeba a uspokojivá rychlost přenosu dat.

Síťové protokoly

Pouze výběr správného média pro komunikaci není dostatečný k tomu, aby komunikace v síti fungovala. Je potřeba nastavit jasná pravidla, která musejí zařízení dodržovat. Těmto pravidlům se říká síťové protokoly. Protokoly mají za úkol zajistit

to, že zprávy budou odeslány a doručeny a v požadované podobě. Nejznámějšími skupinami protokolů jsou Ethernet a TCP/IP. Ethernet protokoly a technologie mají na starost hlavně zprostředkování komunikace mezi lokálními zařízeními. Oproti tomu TCP/IP protokoly zajišťují propojení zařízení z různých sítí skrz internet. Ve světě IoT se objevují nové protokoly, které využívají TCP/IP, ale jsou navrženy tak, aby vyhovovaly specifickým požadavkům. Pro IoT sítě vznikla i nová označení, a to Low Power and Lossy Networks (LLNs). Takové sítě se vyznačují menší přenosovou rychlostí a větší ztrátovostí paketů.

Routování

Internet je v podstatě spojením velkého množství LAN a následných WAN, které následně tvoří celosvětovou síť známou jako internet („síť sítí“). LAN a WAN poté plní roli cest pro přenos packetů. Procesu směrování paketů k jejich cíli, co nejkratší cestou, se říká routování. Routování, ať už uvnitř LAN nebo mezi LAN, obstarává síťový prvek, kterému se říká router. Jelikož mezi zdrojem a destinací se může nacházet velké množství routerů, jedno zařízení není schopné směrovat daný paket celou cestu, a proto každý router řeší pouze jeden krok. Podle ISO/OSI modelu routování probíhá na síťové vrstvě. Mezi nejpoužívanější směrovací protokoly patří RIP, pro menší sítě, a OSPF, pro velké sítě.

V LAN síti router obvykle zastává roli takzvané výchozí brány (default gateway). Ta slouží jako bod, přes který lokální zařízení mohou přistupovat k internetu. Ovšem aby komunikace na internetu a celkově v síti mohla fungovat, každé zařízení musí mít unikátní identifikátor (v rámci sítě), takzvanou IP adresu. Pokud budeme hovořit o IPv4 protokolu, pak veřejnou IP adresu pro přístup k internetu propůjčuje zařízením s privátní IP adresou lokální sítě právě výchozí brána. To znamená, že každé zařízení v LAN bude na internetu vystupovat pod IP adresou routeru, který funguje jako přechod do internetu. Pokud brána na zařízení nastavena nebude, stále je možné komunikovat s ostatními zařízeními v síti, ale ne se zařízeními mimo síť. Veškerá odchozí i příchozí komunikace do/z internetu musí projít přes výchozí bránu. V případě použití protokolu IPv6 nejsme omezení privátními adresami.

7.1 Komunikace v IoT

IoT zařízení mají většinou k dispozici malý paměťový prostor a jsou limitované napájením. Kromě toho jsou běžně umístěné v ne zrovna optimálních podmínkách, které stěžují, jak komunikaci, tak i napájení. Kvůli těmto překážkám se vyvíjejí nové přístupy a technologie, specificky navržené pro IoT. Komunikace mezi zařízeními většinou probíhá pomocí rádiových vln, a to obousměrně (přijímání/odesílání) nebo jen jednosměrně (odesílání). V roce 2015 výbor Internet Architecture Board (IAB) vydal dokument o chytrých zařízeních s označením RFC 7452 [33], ve kterém popisuje čtyři základní komunikační modely využívané IoT zařízeními: device-to-device, device-to-cloud, device-to-gateway a back-end data-sharing.

První model **device-to-device** představuje spojení dvou a více zařízení, které mezi sebou komunikují přímo. Komunikovat mohou přes internet, často ovšem využívají protokoly, jako Bluetooth nebo ZigBee. Tento model se převážně aplikuje v domácích IoT sítích, kde jsou propojeny malá zařízení (žárovky, termostaty, zámky, vypínače atd.). Pro spotřebitele je ovšem nevýhodou jistá nekompatibilnost mezi zařízeními, přesněji mezi protokoly. Například zařízení, která používají protokol ZigBee, nejsou v základu kompatibilní se zařízeními, která využívají protokol Z-Wave.

Dalším modelem je **device-to-cloud** představující spojení IoT zařízení přes internet přímo na cloudovou službu, která může zastávat roli poskytovatele aplikace k výměně dat nebo ke kontrole a řízení provozu. Tento přístup využívá k připojení ke službě již existujících propojení přes Ethernet nebo WiFi. Na využití modelu lze narazit opět u některých domácích zařízení, konkrétně zařízení, kde má smysl například ukládat data, nebo ke kterým chce mít vzdálený přístup. Jedná se tedy o SmartTV, kamery, termostaty, zvonky s kamerou, alarmy atd. Podobně, jako v předchozím modelu, je zde problém s kompatibilitou, a to konkrétně mezi zařízením a cloudovou službou. Každý výrobce má vlastní službu, na kterou je možné zařízení připojit, což limituje nebo zcela znemožňuje použití alternativ. Často se to označuje jako „vendor lock-in“.

Předposledním modelem je **device-to-gateway**, nebo někdy označovaný jako device-to-application-layer-gateway (ALG) model. ALG neboli brána aplikační

vrstvy vykonává funkci prostředníka mezi internetem a aplikačním serverem a rozhoduje o tom, zda bude či nebude umožněna komunikace s aplikačním serverem. Model je tedy založen na tom, že zařízení využívá právě ALG k připojení ke cloudové službě. Často jako ALG vystupuje chytrý telefon, na kterém běží aplikace a ta komunikuje s cloudem. Příkladem jsou klasické fitness aplikace. Ty k měření většinou využívají různá zařízení, jako hodinky, hrudní pásy atd. Takové zařízení ovšem není schopné se samo připojit k službě, a proto využívá aplikaci v mobilním telefonu právě jako ALG. Další možné využití je opět v domácnostech, kdy jedno zařízení, připojené na cloud, funguje jako hub pro ostatní zařízení.

Posledním modelem je **back-end data-sharing**. Tento model umožňuje uživateli exportovat a analyzovat data z chytrých zařízení v kombinaci s daty z dalších zdrojů. V praxi to znamená, že je možná získat data z různých typů zařízení. Jedná se o rozšíření modelu device-to-cloud, kde jsou data shromažďována odděleně podle typu zařízení a typu služby. Efektivní fungování tohoto modelu závisí i na architektuře celého IoT systému.

7.2 Přenosové technologie

Zařízení mezi sebou mohou komunikovat prostřednictvím různých komunikačních technologií a protokolů. Kromě těch typických, jako Bluetooth (v tomto případě nová verze Bluetooth LE), které už v současnosti jsou v IoT spíše samozřejmost, to budou i technologie vyvinuté speciálně pro IoT, jako ZigBee, LoRa, SigFox a další. Každá z nich je vhodná pro určité využití. Rozhodujícími faktory jsou přenosová rychlost, spotřeba energie, dosah a frekvence. Je důležité zmínit, že následující technologie nebo protokoly naleží nižším vrstvám OSI modelu (fyzické, data linkové, síťové). To znamená, že jsou sice zodpovědné za samotný přenos dat mezi zařízeními, ale nespecifikují význam a využití přenášených dat [34]. Protokolům, které pracují na vyšších vrstvách je věnována následující kapitola.

7.2.1 WiFi

Zkratka WiFi je v počítačových sítích označení souboru standardů pro bezdrátovou komunikaci (IEEE 802.11). Od ostatních bezdrátových technologií se ovšem liší v několika věcech a konkrétně pro IoT zrovna vhodná není. WiFi přenáší na

frekvenci 2,4 GHz nebo 5 GHz, což je pro porovnání mnohonásobně vyšší než v celulární síti. S velkou frekvencí přichází i možnost přenosu většího množství dat. Ovšem IoT technologie cílí hlavně na nízkou spotřebu a vyšší dosah. WiFi je pravý opak. Vysoká spotřeba a malý dosah. Využití najde pouze u zařízení, kterým nevadí vyšší spotřeba a přenášejí víc dat na menší vzdálenost (zabezpečení domácnosti). Standardů pro technologii WiFi existuje několik a jsou poplatné době jejich vzniku: 802.11a, 802.11b, 802.11g, 802.11n a 802.11ac. Každý ze standardů má svoje výhody a nevýhody, hlavně v souvislosti s přenosovou rychlostí, náklady, rušením apod. Zkrátka WiFi jako taková není vhodná technologie pro použití v IoT systému. Ovšem v současnosti jsou známé dva nové standardy, které cílí i na IoT [35].

Prvním a starším standardem je 802.11ah, přezdívaný WiFi HaLow, který byl představen v roce 2017. HaLow k přenosu využívá nižší frekvenci, konkrétně 900MHz, nabízí provoz s nízkou spotřebou a možnost tvořit velké skupiny zařízení (okolo tisíce zařízení na jeden přístupový bod). Uvádí se, že dosah HaLow je dokonce větší než u většiny IoT technologií a vyčnívá zejména v prostředích, kde je potřeba aby signál prošel několika překážkami. Jedná se tak o WiFi standard s nejnižší pracovní frekvencí [36].

Druhým standardem, který je zatím stále ve vývoji a jeho představení se očekává v roce 2021, je 802.11ax, někdy označovaný zkratkou WiFi HEW (High Efficiency Wireless) nebo WiFi 6. V tomto případě se nejedná o standard speciálně pro IoT, ale o nástupce předchozí WiFi 5, což bylo označení pro 802.11ac. WiFi 6 je navržena tak, aby byla schopná pracovat na frekvencích mezi 1 a 6 GHz. Oproti předchozí verzi by mělo dojít ke čtyřnásobnému nárustu v přenosové rychlosti a k poklesu zpoždění (latency) až o 75 %. Díky velkému frekvenčnímu rozpětí by WiFi 6 mělo být vhodné jak pro využití v počítačových sítích, tak i v IoT zařízeních [37].

7.2.2 Bluetooth LE

Bluetooth je bezdrátový protokol, definovaný standardem IEEE 802.15.1, používaný v komunikaci na krátké vzdálenosti, v síti PAN. V současnosti je podporována většinou mobilních zařízení, především v oblasti přenosu zvuku. V IoT se ovšem využívá jeho nová verze, a to Bluetooth Low-Energy (někdy označováno Bluetooth Smart), která vznikla ze standardu Wibree společnosti Nokia. BLE jako takové není

samostatnou technologií, ale jedním ze tří režimů, které byly implementovány již v Bluetooth 4.0 [23].

Samotný LE režim ještě nabízí dva pracovní režimy, a to centrální a periferní. Pokud zařízení pracuje v centrálním režimu, má větší spotřebu energie, protože dochází navíc ke zpracování přijímaných dat. Oproti tomu periferní režim je úspornější, jelikož pouze předává data okolním zařízením. Rozdíl mezi Bluetooth a BLE je i v rozdělení pásma. Klasické Bluetooth pracuje na frekvenci 2 402 až 2 480 MHz, což dělá celkem 79 vysílacích kanálů po 1 MHz. BLE má pouze 40 kanálů po 2 MHz, z toho kanály 1 až 37 jsou určeny pro komunikaci mezi zařízeními a 38 až 40 pro technickou komunikaci. Zajímavostí je to, že poslední tři kanály byly vybrány záměrně, jelikož nekolidují s kanály WiFi a nedochází tak k vzájemnému rušení [23].

V současnosti již existuje Bluetooth verze 5. To oproti předchozí verzi nabízí až čtyřnásobný dosah a dvojnásobnou rychlost přenosu v úsporném režimu. Maximální přenosový rychlost vyrostla na 2 Mbps, oproti 1 Mbps předchozí verze. Dosah se oproti verzi 4, která měla 50 metrů ve volném prostranství a 10 metrů v uzavřených prostorech vyšplhal až na 200 metrů v otevřeném prostoru a 40 metrů ve vnitřních prostorech.

7.2.3 ZigBee

ZigBee je pojem označující bezdrátový protokol vyznačující se nízkou výkonností, spotřebou energie a přenosovou kapacitou. V současnosti se na vývoji podílí přes 60 prestižních firem, mezi nimi například Siemens, Samsung, Motorola a další. Nejčastěji se používá, podobně jako Bluetooth, v osobních sítích (PAN) v domácnostech, zdravotnických zařízeních a další. Konkrétními aplikacemi ZigBee jsou například vypínače světel, zobrazovací displeje, zařízení pro řízení dopravy apod. Technologie ZigBee je postavena na standardu 802.15.4 a je navržena tak, aby levnější a jednodušší než obdobné technologie, jako WiFi a Bluetooth. Nejčastěji se používá v zařízeních, které mají vysokou výdrž baterie. Přenosová rychlost se pohybuje od 20 kbps do maximálních 250 kbps a dokáže data přenášet až na vzdálenost 75 metrů [23].

Zařízení dokážou pracovat ve třech různých režimech. Prvním režimem je periodické odesílání dat, nejčastěji ze senzorů. Druhým režimem je odesílání dat při

reakci na nějaký podnět, jako spuštění alarmu, stisk zvonku u dveří, změna teploty. Třetím a posledním režimem je opakující se přenos dat s nějakým zpožděním. Technologie využívá protokolu CSMA/CA, který zabraňuje kolizím při komunikaci [23].

Zigbee tvoří sítě podle dvou bezpečnostních modelů, a to s centralizovaným a distribuovaným zabezpečením. V centralizovaném modelu může být síť vytvořena pouze Zigbee koordinátorem, a naopak distribuované sítě mají na starost routery. Uzly se mohou připojit do kterékoliv sítě a té se následně adaptovat. Proces připojení uzlu do sítě se označuje pojmem *network steering*[38]. Každá Zigbee síť je zabezpečená 128bit standardem AES a má přidělené PAN ID. Stejně tak i každé zařízení má vlastní 64bit (nebo zkrácené 16bit) identifikační číslo [23].

Zařízení v síti se někdy rozdělují i podle rozsahu činností, kterou vykonávají, na zařízení s plnou funkčností (FFD – Full Functional Device) a limitovanou funkčností (RFD – Reduced Functional Device). Jak už název napovídá, zařízení FFD mají k dispozici všechny funkce sítě ZigBee, oproti tomu RFD mají pouze ty nejdůležitější funkce a komunikují pouze s koordinátorem. RFD často bývají koncová zařízení, která se starají o monitoring a sběr dat [23].

7.2.4 LoRa

LoRa (Long Range) nebo LoRaWAN je technologie navržená speciálně pro bezdrátovou komunikaci na velké vzdálenosti. Komunikace zpravidla probíhá obousměrně mezi koncovými zařízeními pro sběr dat, jako čidla a senzory, a gateway zařízením (tzv. koncentrátorem). Sítě využívající tuto technologie mají často topologii rozšířené hvězdy (star-of-stars), nebo long range star. Komunikace probíhá na celkem 9 kanálech v pásmu 868 MHz, z toho prvních 8 je pro komunikaci gateway s koncovými zařízeními a poslední pro komunikaci ze sítě ke koncovému zařízení [23].

Základny přijímají data ze všech koncových zařízení v dosahu, což znamená, že stejná data mohou být odeslány na více základen najednou. Gateway je poté předá řídícímu serveru, ke kterému je připojena, buď dedikovaným spojem nebo zabezpečeným připojením přes internet. Řídící server má na starosti komunikaci a směrování zpráv mezi cloudem a koncovým zařízením, monitoruje síť a loguje

veškerý provoz. Server z takto obdržených dat vyfiltruje duplicitní záznamy pomocí sekvenčního čísla zprávy. Komunikace opačným směrem opět prochází přes gateway [39].

Kvůli různorodým koncovým zařízením a požadavkům standard LoRaWAN definuje tři třídy zařízení A, B, C , především podle rychlosti komunikace a životnosti baterie. Charakteristické pro jednotlivé třídy je [39]:

- Ve třídě **A** komunikaci začíná koncové zařízení a to tak, že buď periodicky odesílá data, anebo v závislosti na podnětu (podobně jako pracovní režimy ZigBee). Po odeslání dat, jsou vyhrazeny dva intervaly, kde se čeká na doručení zprávy.
- Ve třídě **B** komunikace probíhá opačným směrem, tj. ze sítě ke koncovému zařízení. Princip spočívá v tom, že se na koncových zařízeních v pravidelných intervalech zapíná přijímací modul a pouze v tuto chvíli je schopné přijmout zprávy. K tomu, aby byly zprávy odesílány ve správnou chvíli, je potřeba synchronizovat zařízení v síti. To se realizuje synchronizačními impulsy, které jsou do sítě pouštěny každých 128 sekund, a podle těchto signálů se srovnávají vnitřní časovače koncových zařízení.
- Třída **C** se podobá třídě B, ale rozdíl je v tom, že zde je přijímač zapnutý po celou dobu.

LoRaWAN aplikuje dvě vrstvy zabezpečení: jednu pro síť a jednu pro aplikace. Vrstva zabezpečení sítě zajišťuje, že všechny uzly v síti jsou autentické a vrstva bezpečnosti aplikace chrání data koncových uživatelů. Každá úroveň používá vlastní, na druhém nezávislý, AES 128bit klíč. Podle prvního klíče, označovaného jako NwkSkey, se šifruje komunikace pouze uvnitř sítě a zároveň brání připojení nechtěných zařízení. Podle druhého klíče, označovaného jako AppSkey, se šifruje komunikace směrem ke cloudu. Každé koncové zařízení má svůj 64bitový identifikátor, tzv. EUI64 [39].

7.2.5 SigFox

SigFox je, podobně jako LoRaWAN, bezdrátová komunikační technologie velkého dosahu s malou přenosovou kapacitou pro data, typicky ze senzorů a dalších měřících zařízení. Využití SigFoxu lze najít v Industry 4.0, SmartCity, bezpečnostních systémech, logistice, při odečítání vody, plynu apod. Vlastnosti sítě SigFox jsou výdrž baterie, která může dosahovat až 15 let, pořizovací cena a jednoduchost implementace. Ve velkém je síť SigFox používána ve Francii, Španělsku a ve Velké Británii. Dosah technologie se při přímé viditelnosti udává až na 200 km, v krajině na 50 km a ve městech v rozmezí 3-5 km. Stejně jako LoRa pracuje v pásmu 868 MHz (v Evropě). V současnosti již nabízí obousměrnou komunikaci mezi cloudem a koncovými zařízeními [40].

SigFox byl záměrně vytvořen pro přenášení zpráv malého objemu, jelikož menší objem dat je roven menší spotřebě a větší výdrži baterie. Celý přenášený rámec je tvořen 26 byty, z toho až 12 bytů je určeno pro data (tzv. Payload). Pro zpětnou komunikaci je pak obsah dat omezen na maximálně 8 bytů a pouze 4 zprávy denně. Původně zpětnou komunikaci SigFox ani neumožňoval. V porovnání s IP rámcem je tento rámec velice úsporný, jelikož při přenosu dat o objemu 12 bytů by IP rámec i tak dosahoval velikosti 40 bytů [40].

Typickou topologií SigFox sítě je hvězda. Podobně jako LoRa využívá základen, které pokrývají danou oblast, stejně jako LoRa. Každý stát má svého SigFox operátora, v ČR je to společnost SimpleCell Networks, který přijímá místní zprávy a následně je zasílá přímo do cloudu SigFoxu, kde se zprávy třídí a zpracovávají. SigFox zajišťuje veškeré funkce související s komunikací, od identifikace, autentizace zařízení až po uložení dat a přístupu k datům. Cena za připojení zařízení do sítě se pohybuje od jednotek po desítky korun za měsíc v závislosti na počtu přenesených zpráv a množstevní slevě [23].

7.2.6 4G/5G

Celulární (mobilní) síť je technologie, která našla využití především v telekomunikacích, ovšem její použití je možné i v IoT. To nejvíce benefituje především z velkého pokrytí této technologie, což umožňuje zařízením komunikovat na velké vzdálenosti. První spuštění celulární sítě proběhlo v 80.

letech minulého století, kdy byla představena takzvaná první generace (1G). Od té doby, zhruba každých deset let, přichází nová generace, až po současné 4G, a dokonce už i 5G.

4G je současně používaná technologie pro přenos dat. Mezinárodní standard pro telekomunikace (IMT-Advanced) udává přenosovou rychlost pro každý systém využívající 4G na 100 Mbps pro více pohyblivá zařízení (automobily, vlaky apod.) a 1 Gbps pro málo pohyblivá nebo stacionární zařízení. Kromě v dnešní době již samozřejmých funkcí, mezi které patří hovory, video hovory a mobilní internet, podporuje i cloudové služby, mobilní televizi, mobilní 3D televizi, herní služby apod. Nejpopulárnější 4G systémy jsou LTE a WiMAX. Především standard LTE se v posledních letech stal zajímavých pro IoT. Poslední verze LTE Advanced Pro (někdy označována jako 4.5G, 4.9G, Pre-5G) s sebou přinesla spoustu nových technologií a mezi nimi i LTE IoT zahrnující standard Narrowband IoT (NB-IoT). NB-IoT je rádiová technologie pro LPWAN sítě pro připojení velkého množství různých zařízení. Zaměřuje se především na pokrytí ve vnitřních prostorech, nízké náklady, velkou výdrž baterie a velkou hustotu připojených zařízení. NB-IoT je výhodná především pro aplikace a zařízení, které vyžadují častější a méně pravidelnou komunikaci [41]. V roce 2019 tuto technologii nabízelo již 142 operátorů po celém světě a konkrétně i v ČR je 100% pokrytí [Vodafone].

Další generací je 5G, jejíž nasazování začalo v roce 2019 a odhaduje se, že v roce 2025 by mělo 5G využívat 1,7 miliardy uživatelů. Primárním cílem nebudou pouze mobilní telefony, ale očekává se velký přínos i pro oblast IoT a M2M. V současnosti se již vyrábějí mobilní telefony podporující právě 5G a konkrétně v ČR již všichni velcí operátoři začali s testováním a postupným pokrýváním území. Lze tedy očekávat ostré spuštění 5G sítě během pár let.

7.3 Komunikační standardy vyšších vrstev

V předchozí kapitole byly představeny nejznámější přenosové technologie a standardy, které ale přísluší nižším vrstvám OSI modelu. Ovšem nelze zapomenout i na standardy vyšší aplikační vrstvy. Ty mají totiž taky velmi důležitou úlohu. Existence standardizovaných komunikačních protokolů zajišťuje, že bude možné propojit mezi sebou více zařízení od různých výrobců a značek. To znamená, že i

když budou všechny zařízení operovat na LoRaWAN, není zajištěno jejich vzájemné snadné propojení, jelikož každé zařízení stále může data vyžadovat, odesílat nebo chápat zcela jinak. Tím se rozumí i maličkosti, jako různé formáty čísel. Mezi aktuálně nejpoužívanější standardy patří CoAP, MQTT a XMPP [34].

7.3.1 Protokol CoAP

Protokol CoAP (Constrained Application Protocol) velmi připomíná HTTP. Je založený na REST modelu, což znamená, že server zpřístupňuje data pod adresou URL a klienti k nim přistupují za pomoci známých metod GET, POST, PUT, DELETE. Komunikace je tedy realizována asynchronně na principu klient-server. Hlavní využití má v komunikačním modelu device-to-device a byl navržen speciálně pro IoT, tak aby využíval minimální zdroje na zařízení i na síti. Zároveň však nabízí velmi dobrou spolupráci s HTTP. Pro komunikaci využívá UDP protokol, který je, oproti TCP, považován za nespolehlivý, ale to není tak úplně pravda. Spolehlivost je v CoAP řešena pomocí značení zpráv. Standard definuje celkem čtyři typy (značky) zpráv: potvrditelné (confirmable – CON), nepotvrditelné (non-confirmable – NON), potvrzení (acknowledgement – ACK) a reset (RST). Zpráva odeslaná s příznakem CON poté funguje podobně, jako by byla odeslané protokolem TCP. Dokonce dochází k opětovnému zasílání do doby, než odesílatel obdrží od příjemce zprávu s příznakem ACK a se stejným ID. Pokud cílové zařízení není schopné zprávu zpracovat, a ani na ni odpovědět chybovou hláškou, místo ACK zašle zprávu RST. Pokud zpráva nevyžaduje potvrzení, přidá se k ní příznak NON. Takže i přesto, že využívá protokol UDP, CoAP stále dokáže zajistit spolehlivou komunikaci na principu požadavek-odpověď [42].

7.3.2 Protokol MQTT

MQTT (Message Queueing Telemetry Transport) ke komunikaci mezi klienty využívá takzvaných brokerů. Broker označuje centrální řídící prvek, který působí jako prostředník pro výměnu zpráv mezi klienty. MQTT je velmi používaný protokol především v zařízeních s malou spotřebou energie. Stejně jako CoAP využívá model klient-server, kde server je právě broker a klient je koncové zařízení, často typu senzor nebo čidlo. Ke komunikaci mezi serverem a klientem se používá protokol

TCP. Komunikační model MQTT je označován jako publish-subscribe. Klienti mohou publikovat, anebo odebírat data od jiných klientů. Broker má na starosti sběr těchto publikovaných dat, roztřídění podle témat (topic) a zaslání dat klientům (odběratelům). Jeden klient může být v některých tématech publisher a v jiných subscriber. MQTT nemá nijak předepsaný obsah zprávy („payload agnostic"), prostě přenáší binární data, nejčastěji ve formátech JSON, BSON, text, ale přenést může cokoli. I přesto, že maximální velikost zprávy je až 256 MB, skutečná velikost běžných zpráv je mnohonásobně menší. Pro zpětnou vazbu při komunikaci má MQTT definované tři úrovně QoS (Quality of Service). První je úroveň at-most-once, která nezaručuje potvrzení odeslání ani doručení. Druhá je at-least-once a ta zaručuje alespoň jedno doručení zprávy. Nejvyšší úrovní je just-once, kde je každá zpráva doručena právě jednou. Klient může a nemusí podporovat všechny tři úrovně QoS [43].

7.3.3 Protokol XMPP

Extensible Messaging and Presence Protocol (XMPP) je rozšiřitelný protokol pro zasílání zpráv, zjištění stavu, komunikaci více účastníků, hlasové hovory, videohovory apod. Jedná se o implementaci jazyka XML, do kterého zapouzdřuje zprávy. Původně vznikl exkluzivně pro síť Jabber, ale později se ukázalo, že ho kromě okamžitého zasílání zpráv lze využít i pro komunikaci mezi programy nebo pro ovládání botů. Dnes je již oficiálním internetovým standardem uvedeným v dokumentech organizace IETF RFC6120 [44] a RFC6121. XMPP je zcela otevřené a dostupné všem. Servery běžně využívají k přenosu protokol TCP. Síť, která operuje na XMPP je decentralizovaná, a tak uživatelé mohou provozovat vlastní servery. Výhodou vlastních serverů je to, že díky využití XML, je možná customizace nad rámec základních funkcí.

I XMPP využívá architekturu klient-server, ale v tomto případě se ze strany klientů málokdy jedná o přímou komunikaci. Decentralizovaná struktura by se dala přirovnat k e-mailu. Uživatel si může vybrat server, kterému důvěřuje, ale zároveň ho bude moc kdykoli změnit. Samozřejmě, že komunikace s uživateli jiných serverů je stále možná, právě jako u e-mailu. Uživatel má vlastní JID (Jabber ID), které obsahuje název serveru a uživatelské jméno, např. uživatel@server.cz. Zaslání

zprávy pak probíhá tak, že nejdříve se zpráva odešle na server odesílatele. Tento server otevře spojení se serverem příjemce (pokud je to možné) a zprávu na něj zašle. Odtud již zpráva putuje k příjemci. Většinou je možné se k uživatelskému účtu přihlásit z několika různých klientů, podobně jako e-mailový klient Outlook od Microsoftu umožňuje přidání několika účtů najednou.

7.4 Cloud a Fog computing

Zařízení připojených do sítě stále přibývá a v případě IoT to platí dvojnásobně. S každým dalším senzorem a čidlem se zvětšuje i objem dat, který je potřeba sesbírat, zpracovat, uložit, analyzovat a vyhodnotit. Pro většinu uživatelů je toto nereálné již provádět lokálně (on-premise), a proto stále častěji využívají takzvané cloud a fog služby.

Cloud computing lze chápat jako poskytování aplikací, služeb a zdrojů uživatelům, kteří k nim přistupují vzdáleně. Uživatel si může pronajmout software nebo i výpočetní výkon (hardware), které jsou umístěny mimo jeho lokalitu (off-premise). Tímto uživateli odpadá potřeba vlastního serveru, což výrazně sníží náklady na údržbu a nasazení nového softwaru. Vlastnosti cloud computingu jsou [45]:

- **Více nájmovost** (Multitenancy) – konkrétní fyzické zdroje jsou sdíleny mezi více uživateli
- **Průběžný systém placení** (Pay as you go) – uživatel platí za to, co reálně využije
- **Aktuálnost** (Up to date) – veškeré aktualizace a upgrady provádí poskytovatel
- **Škálovatelnost a výkonnostní elasticita** (Scalability and performance elasticicty) – uživatelům dovoluje jednoduše změnit zdroje a služby

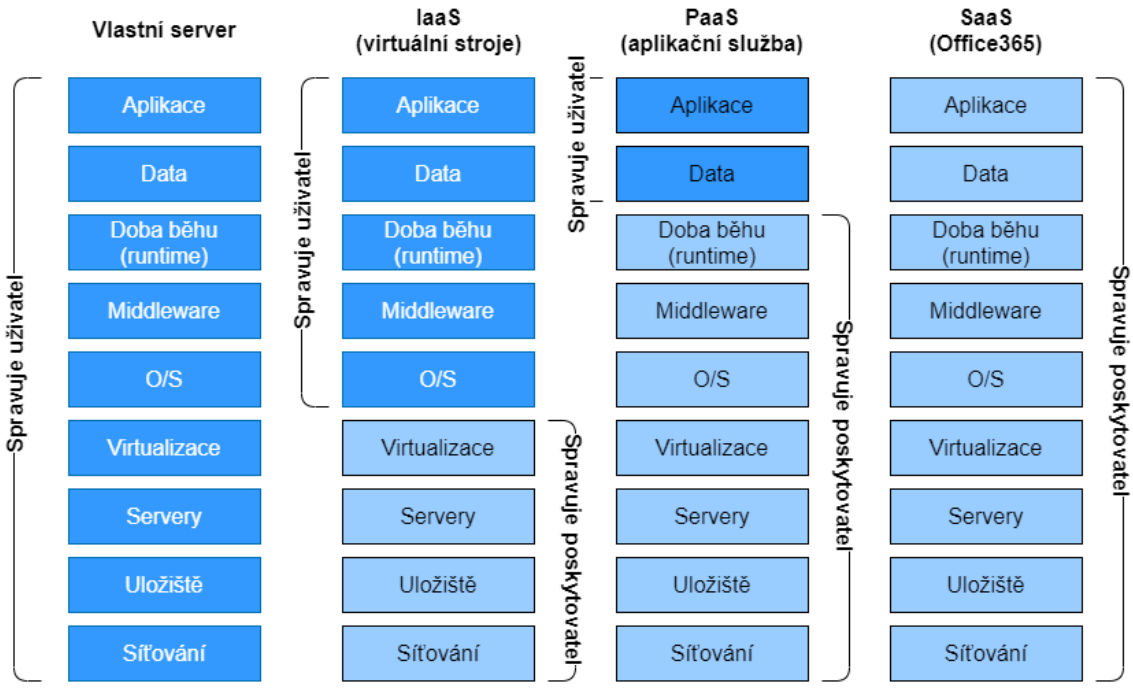
Služby poskytované v cloudu se často označují příznakem „as a service“ (jako služba). Existují tři základní modely poskytování služeb: Infrastructure as a service (IaaS), Platform as a service (PaaS) a Software as a service (SaaS). Hlavním rozdílem modelů je zejména v tom, co spravuje poskytoval a co uživatel.

Součástí modelu **IaaS** je pouze poskytnutí hardwarových zdrojů, jako síťová zařízení, úložného prostoru, výpočetního výkonu apod., uživateli, který na něm poté provozuje vlastní software a provádí operace. Poskytované zdroje je možné kdykoli jednoduše změnit (on-demand – na požádání). Poskytovatel je současně zodpovědný za údržbu poskytovaných zdrojů a za jejich funkčnost [46].

Model **PaaS** je rozšířením IaaS o určitou softwarovou úroveň. Uživateli jsou poskytnuty všechny potřebné prostředky pro tvorbu aplikací a služeb, jako návrh, testování, implementace, hostování, integrace webových služeb, integrace databází a uložišť. Nevýhodou tohoto modelu je nízká interoperabilita, tj. přenos aplikace k jinému poskytovateli nemusí být možný, anebo velmi náročný. V případě poskytovaní služby pro vývoj a provoz mobilních aplikací se někdy hovoří o modelu mPaaS [46].

V posledním modelu **SaaS** je uživateli poskytnuta samotná aplikace i hardware, na kterém běží. Uživatelé k ní přistupují vzdáleně pomocí internetu, takže je dostupná kdekoli a kdykoli, pokud má uživatel k internetu přístup. Poskytoval provádí veškeré updaty aplikací a upgrady hardwaru. Tímto způsobem jsou často poskytovány aplikace s nízkou úrovní interakce. Příkladem konkrétní SaaS je služba Office 365 od Microsoftu [46].

Na následujícím obrázku (viz Obrázek 27) je znázorněno, za co v jakém modelu zodpovídá poskytovatel a za co uživatel. Pro porovnání je zde model On Premise, což znamená, že veškerý hardware a aplikace vlastní uživatel a jsou umístěny lokálně. SaaS je tedy opakem On Premise modelu.



Obrázek 27 Porovnání cloudových modelů

Zdroj: vlastní zpracování (podle http://cloudonmove.com/iaas-paas-saas-what-do-they-mean/)

Cloudové platformy

Díky vysoké dostupnosti a jednoduché škálovatelnosti se cloudové služby staly skvělou cestou, jak rozšířit funkčnost IoT systému. Cloud je v IoT využíván zejména na zpracování dat a následné ukládání, místo toho, aby to vykonávala samotná zařízení. Výhodou je to, že data v cloudu jsou poté dostupná jakémukoli zařízení v síti, dokud je připojené k internetu. Pro IoT systémy, které by z těchto cloudových služeb mohly těžit, existuje nespočet možností, co se nabídky týče. Největšími cloudovými platformami jsou AWS od Amazonu, Azure od Microsoftu a Google Cloud. Tyto platformy nabízejí obrovské množství služeb pro různá využití včetně IoT. Za zmínku ovšem stojí i platformy jako IFTTT, Zapier a Built.io, které jsou svojí nabídkou více zaměřené na IoT. Všechny tři platformy se snaží o propojení aplikací, zařízení a služeb od různých značek za účelem automatizace v daných aplikacích, zařízeních a službách.

IFTTT („If This Then That“) se zaměřuje především na tvorbu takzvaných appletů. Applet označuje programovou komponentu, která je schopna běžet v jiné aplikaci za účelem konkrétní činnosti. Důležité je to, že daná činnost je plně automatizovaná a spouští se při zadaném podnětu. Na webu IFTTT je v tuto chvíli

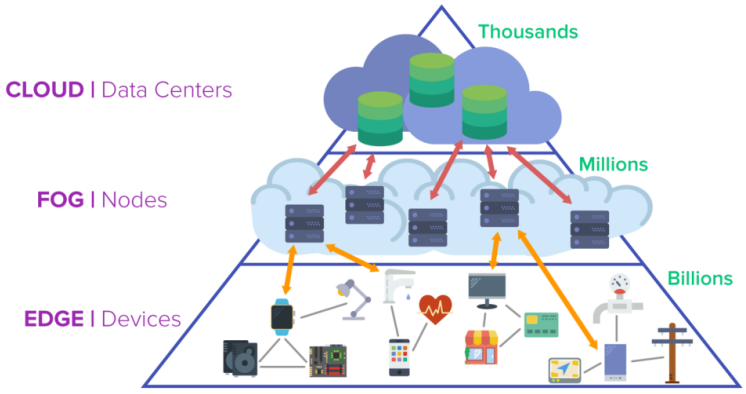
k dostání velké množství appletů od různých vývojářů a pro různé aplikace. Příklad takového appletu je například to, že uživatel dostane SMS, kdykoli pohybový sensor zaznamená u jeho domu pohyb. Normálně by takový úkol nebyl zrovna jednoduchý na realizaci, protože by zařízení muselo komunikovat skrze internet a mobilní síť (celulární síť) zároveň. IFTTT tento proces zjednodušuje tím, že činnost odeslání SMS je spojená s přístupem ke konkrétní URL adrese. Pokud tedy senzor zaznamená pohyb, stačí mu pouze vstoupit na danou URL a applet se postará o odeslání SMS. Dalším příkladem je zápis do sešitu excel na základě hlasových příkazů pro domácího asistenta Alexa. IFTTT nabízí omezený přístup zdarma a například vývojářská úroveň vyjde uživatele zhruba na 199$ ročně (cca 4500 Kč) [47].

Zapier je v hodně věcech velice podobný IFTTT. Jedná se o další platformu zabývající se automatizací úkonů napříč různými aplikace, v případě Zapier hlavně webovými aplikacemi. Oproti IFTTT se celkově více zaměřuje spíše na využití v podnikatelské sféře. I Zapier nabízí přístup zdarma omezený počtem akcí, které jsou zde nazvány pojmem zaps, a to na maximálně pět zapů. Při přesáhnutí tohoto počtu je vyžadován placený účet.

Ani platforma **Built.io** se od předchozích dvou v ničem zásadním neliší. Jediným rozdílem je to, že se více zaměřuje na samotné vývojáře a snaží se jim co nejvíce proces tvorby aplikací usnadnit tím, že nabízí pokročilejší funkce a nástroje v porovnání se Zapier a IFTTT.

Fog computing

Pojem fog computing pochází od společnosti Cisco a označuje alternativní řešení ke cloud computingu. Fog je definován jako decentralizovaná výpočetní vrstva mezi cloudem a zařízeními, která produkují data (viz Obrázek 28). Cílem je nabídnout uživateli určité služby blíže k okraji sítě, kde data vznikají (označeno pojmem edge), čímž se snižuje potřebná vzdálenost přenosu dat, zvyšuje se efektivnost sítě a snižuje zátěž na cloudu.



Obrázek 28 Edge, Fog, Cloud
Zdroj: https://www.omnisci.com/technical-glossary/fog-computing

Fog computing je významný zejména v oblasti IoT, kde různá zařízení generují velký objem dat a často je potřeba je co nejrychleji zpracovat a vyhodnotit. Tím, že se výpočetní služby a struktura, jednotně označeno jako uzel (angl. Node), přesunou blíže k těmto zařízením, se výrazně sníží odezva/zpoždění (latency) a lze tak dosáhnout zpracování téměř v reálném čase. Uzly, které jsou k edge nejblíže, sbírají data z routerů a modemů a následně je přenáší na nejvhodnější místo pro analýzu. Fog computing se často zaměňuje s dalším podobným pojmem, a to edge computing, což označuje proces generování, zpracování a uložení dat prakticky lokálně. Fog poté obsahuje edge computing, a navíc poskytuje potřebnou infrastrukturu a síťová propojení k přenosu dat. Mezi základní výhody fogu patří [48]:

- **Minimalizace odezvy** – Zpracování dat blíže k zařízením, která je získávají, umožňuje vyhodnocení dat v reálném čase a tím rychlejší reakce na různé podněty (varovné systémy, bezpečnostní zařízení apod.).
- **Snížení objemu přenesených dat** – Fog computing snižuje objem přenesených dat do cloudu, čímž uvolňuje místo důležitějším činnostem.
- **Snížení provozních nákladů** – Zpracování dat lokálně šetří síť, a tím se snižují operativní náklady.
- **Zvýšení bezpečnosti** – Jednotlivé nody ve fogu rozdělují síť do několika segmentů, na které je možné aplikovat samostatné bezpečnostní politiky a firewally.

- **Větší spolehlivost** – Souvisí předchozími výhodami. Snížením odezvy a objemu přenesených dat poskytuje větší spolehlivost v nepříznivých podmínkách nebo ve stavu nouze.
- **Větší flexibilita v podnikání** – Díky fogu lze rychleji a jednodušeji nasazovat nový služeb, které více odpovídají požadavkům klienta.

7.5 Big data

Data je možná popsat dvěma pojmy, data in motion (data v pohybu) a data at rest (data v klidu). Data in motion označuje proces získávání informací z dat v momentě, kdy jsou získávána, a následně mohou a nemusí být uloženy. Dá se to představit, jako popis nějaké události, která se právě děje. Oproti tomu data at rest se průběžně ukládají. Zisk informací z dat předtím, než jsou uložena, je významný především ve zdravotnictví, výrobě, energetice, veřejném sektoru a u poskytovatelů služeb. Pro spoustu situací jsou data in motion lepší volbou, protože je proces rychlejší a odpadá potřeba jejich uložení. V současnosti každý den vzniká tak velký objem dat, který již nelze rozumně ukládat v centralizovaných datových skladech. S rostoucím počtem senzorů a čidel je stále jasnější, že rozhodnutí a akce je nutné provádět na okraji sítě (edge), ve které dochází ke vzniku těchto dat. Řešením je právě fog computing, který tyto služby přináší blíže k místu vzniku a dochází tak k real-time analýze, a tím pádem i k zisku informací.

Konkrétně senzory, čidla a obecně rozvoj IoT technologií jsou jedním z hlavních důvodů exponenciálního růstu objemu dat. S rostoucím počtem měřících zařízení se zvětšují i požadavky na ostatní zařízení, které řídí jejich přenos. Existují již i mobilní routery, které jsou nasazovány například v letadlech, a dokonce i v některých osobních automobilech. Počet zařízení tedy není to jediné, co roste, ale jsou to i samotné hranice IoT. Internet je nyní dostupný na požádání téměř kdekoli, kdykoli, jakkoli.

Tento obrovský nárust dat dal za vznik novému termínu, a to big data. Tímto pojmem se označují především data, která jsou tak různorodá a složitá, že je velmi náročné je uložit, zpracovat a vyhodnotit s využitím tradičním postupů. Podstata práce s takovými soubory je stále stejná, jako se soubory dat menších objemů, ovšem rychlost zpracování a rychlost přenosu big data je mnohonásobně vyšší, a to

ztěžuje i získávání informací. Za tímto účelem vznikly tzv. big data systémy, které pomáhají se zpracováním velkého množství dat. K popisu big data se používá 3V charakteristika (Volume, Velocity, Variety) [49]:

- **Objem** (Volume) – Čistý objem dat, je jednou z charakteristik, kterou lze big data systém popsat. Týká se to i toho, že pro zpracování takového množství často nestačí jeden počítač, a tak je potřeba dedikovaného výpočetního prostoru.
- **Rychlost** (Velocity) – S velikostí souvisí i rychlost, jakou se data pohybují v systému. Data do systému proudí z několika zdrojů a často je za potřebí jejich zpracování v reálném čase. Opět toto nelze provádět na jednom zařízení, a proto je za potřebí robustních systémů.
- **Rozmanitost** (Variety) – Poslední hlavní charakteristikou je rozdílnost dat v big data systému. Tím, že se v systému potkávají data z různých zdrojů a tím pádem různých formátů a struktur (video, audio, logy, texty) je potřeba data třídit a transformovat.

Ovšem s dalším vývojem se objevily návrhy na další charakteristiky, kterými lze big data popsat a v současnosti se tedy mluví až o 6 V. Spíše než o charakteristiky, se ale jedná o problémy, které s big data systémy souvisí. Přidané charakteristiky jsou:

- **Pravdivost** (Veracity) – Velké množství zdrojů dat a složitost jejich zpracování, může ohrozit pravdivost (kvalitu) dat.
- **Variabilita** (Variability) – Velká rozmanitost dat vede k rozdílům v jejich kvalitě.
- **Význam** (Value) – Nejobtížnější úloha systému je datům přiřadit význam nebo hodnotu.

Data se stávají čím dál důležitějším prvkem napříč všemi odvětvími a je to jedna z hlavních hnacích sil technologického pokroku. V současnosti již existuje velké množství systémů a přístupů, které se problematikou big data zabývají. Příkladem open source řešení, které se různým způsobem zabývají ukládáním, zpracováním a přenosem big data, jsou například Apache Hadoop, Webex Teams, Apache Cassandra a Apache Kafka a často tyto služby bývají i v nabídce poskytovatelů cloudových služeb, jako Azure, AWS, Google Cloud apod.

8 Digitalizace podnikání

Informace v kapitole Digitalizace podnikání jsou převzaty z online kurzu Cisco [1], z části Chapter 5: IoT Applications in Business, pokud není uvedeno jinak.

Internet věcí je o připojení dosud nepřipojitelných zařízení a udělat je přístupnými přes internet. V současnosti je možné k síti připojit cokoliv a odhaduje se, že v roce 2025 by měl počet připojených zařízení dosáhnout počtu 30 miliard (v roce 2020 je to zhruba 20 miliard). Velký podíl na tom mají domácnosti, kde kromě běžných zařízení (počítače, mobilní telefony, notebooky apod.) lze do sítě připojit i dříve nepřipojitelná zařízení (pračky, myčky, alarmy, zvonky apod.). Druhou velkou oblastí, kde prudce narůstá počet připojených zařízení, je průmyslové a komerční prostředí. Oproti domácím systémům se od těchto aplikací očekává mnohonásobně vyšší spolehlivost a úroveň automatizace, jelikož mohou vyžadovat okamžité reakce na probíhající události, ale i tak některé systémy stále vyžadují interakci člověka. Selhání systému v takovém prostředí může mít nedozírné následky.

I přes obrovské množství již připojených zařízení se jedná pouze o jednotky procent celkového počtu zařízení a lze tak očekávat ještě větší rozvoj IoT. Spousta současně připojených zařízení je často připojena k síti se specifickým využitím, což ztěžuje proces integrace těchto zařízení do IoT systému. Příkladem jsou komerční a obytné komplexy, kde současně existuje spousta separátních sítí a kontrolních systémů pro vytápění, ventilaci, zabezpečení, osvětlení, telekomunikační služby. Podobně je to i v průmyslových objektech, ale v ještě větším měřítku. Takové systémy se začaly označovat jednotným pojmem „operační technologie“ (OT – Operational Technology). Sjednocení těchto rozdílných technologií a protokolů do jedné konvergované sítě by spoustu věcí nejen zjednodušilo, ale došlo by i ke znatelnému ušetření zdrojů. Takto rozsáhlá infrastruktura by přinesla obrovské možnosti v oblasti zabezpečení, analýzy a managementu.

8.1 IoT v průmyslu

Snaha o vznik „smart“ továren, v podobě Průmyslu 4.0, průmyslového internetu (the Industrial Internet) a evropské iniciativy pro továrny budoucnosti, nastartovala využití IoT v průmyslových odvětvích s cílem zvýšení flexibility a produktivity a

zároveň nižšími výrobními náklady. Tento koncept se označuje pojmem Industrial IoT (IIoT) a stal se hlavní součástí evoluce celého IoT. Taková řešení se ovšem potýkají s nemálo překážkami, které jsou unikátní a rozdílné od ostatních IoT systémů. To vše je především kvůli potřebě speciálních technologií, jako integrovaných programovatelných logických kontrolerů (PLC) a systémů pro dozor, řízení a sběr dat (SCADA). Obě tyto technologie společně se související průmyslovou sítí, která je propojuje, tvoří hlavní infrastrukturu zmíněných operačních technologií (OT). Zde nastala situace, že OT technologie se vyvíjejí samostatně a nezávisle na technologiích IT, především kvůli potřebám systémů v odvětvích průmyslové výroby, produkce energie, distribuce energie apod. Na takové systémy jsou kladeny nároky týkající se nepřetržitého provozu, bezpečnosti, operací v reálném čase a další. Nástup IIoT technologií představuje výzvy především ve spojení stávajících IT systémů, jako systémy pro plánování podnikových zdrojů (ERP), a OT technologií. V takovém případě je nutné ERP systémy rozšířit o funkcionalitu týkající se produkce, které jsou většinou řízeny výrobními exekučními systémy (MES). S jistotou lze konstatovat, že systém, který by propojil řízení od business procesů až po samotné senzory, by nabídl nutně potřebou flexibilitu. Ovšem to neznamená, že průmyslové technologie jsou součástí pouze továren a výrobních podniků. Přednosti, které OT nabízí najdou využití i v dalších odvětvích. Významnými oblastmi v evoluci IIoT je i energetický sektor, který vlastně tvoří tu kritickou část veškeré infrastruktury. Kvůli neustále rostoucí populaci a z toho důvodu rostoucí poptávce po energiích se řízení distribuce stalo velmi důležitou oblastí. Mimo energetiku se IIoT adoptuje i ve vodohospodářství, logistice, zemědělství apod.

Proces konvergence IT a OT je jen část problému, se kterým se vzestup IIoT potýká. Mimo to je potřeba vytvořit odpovídající architektury, které pomůžou s efektivní tvorbou a správou IIoT systémů, dále je vyžadován vývoj technologií pro řízení kyberfyzických (cyber-physical – CPS) systémů, senzorů a sítí a samozřejmě musí být bráno v úvahu zabezpečení a bezpečnost. Všechny tyto části musí být navrženy sjednoceně v kontextu IIoT. Jaký je rozdíl mezi zabezpečením a bezpečností? Odborná literatura naznačuje, že bezpečnost (safety) byla hlavní starostí v oblasti OT a zabezpečení (security) se řešilo především v IT. Pokusy o jejich spojení přinesly poznatek, že bezpečnost je nedosažitelná bez zabezpečení. To

je důvodem velké pozornosti věnované problémům zabezpečení některých průmyslových systémů. V současnosti již existuje několik standardů zabývajících se tímto problémem (např. ISO/IEC 27000 a ISA/IEC 62443), ale k tomu, aby bylo možné sjednotit veškeré systémy pod IIoT, je ještě daleko.

8.2 Cisco IoT systém

Společnost Cisco představila koncept Cisco IoT systému, který by měl organizacím pomoci se zavedením IoT řešení. Specificky snižuje složitost digitalizace v odvětvích výroby, energetiky, vodohospodářství, pohonných hmot, logistiky, těžařského průmyslu apod. Tento systém byl představen již v první kapitole této práce (viz kapitola 4.2.2), jako šest pilířů Cisco IoT systému. V části digitalizace průmyslu a podnikání ovšem opravdu zapadá do kontextu a nabírá na relevantnosti. Jedná se tedy o soubor nových a existujících produktů a technologií a konkrétně se jedná o:

- **Síťové připojení** – spolehlivé, škálovatelné, výkonné síťové řešení vhodné do jakéhokoli prostředí
- **Fog computing** – software a hardware, který poskytuje IoT aplikace na okraji sítě a umožňuje efektivně zpracovávat a řídit data v momentě jejich vzniku
- **Zabezpečení** (kybernetické i fyzické) – zabezpečení cloudu až po fog, které pokrývá celé kontinuum útoku, tzn. před, během a po
- **Analýza dat** – distribuovaná infrastruktura, na které běží specifický software
- **Řízení a automatizace** – zjednodušené řízení velkých IoT sítí, které dovoluje spojení OT a IT
- **Platforma podporující aplikace** – platforma, která umožňuje vývoj a nasazení cloudově založených aplikací do fogu

Existuje spousta druhů sítí podle využití, rozsahu a umístění. Jedno mají však společné, a to potřebu zařízení pro síťové připojení, které už se ale liší podle typu sítě. Například je jasné, že v domácí síti nebudou zařízení stejná, jako v průmyslové síti. To, jaká zařízení mohou být použita právě v průmyslu, popisuje první pilíř Cisco

systému. Cisco, které taková zařízení i vyrábí a poskytuje, je označuje jednoduše, a to průmyslové routery, průmyslové switche, průmyslové přístupové body (AP) a vestavěné sítě. Taková zařízení podporují celou škálu komunikačních rozhraní, jako Ethernet, sériové, celulární, WiMAX, LoRa a další IoT protokoly.

Síťová zařízení odesílají a přijímají mnoho proudů dat a tyto proudy je možné rozdělit do tří kategorií: datový provoz (data traffic), řídící provoz (control traffic), správní provoz (management traffic). Každá kategorie má rozdílnou funkci, a z toho důvodu se uvádí, že i každé zařízení obsahuje tři úrovně (planes):

- **Datová** (data plane) – Reprezentuje všechny aktivity, které zařízení provádí za účelem obdržení a odeslání dat dalším zařízením.
- **Řídící** (control plane) – Na této úrovni dochází k rozhodnutí, jakou cestu zvolit k nejefektivnějšímu doručení dat.
- **Správní** (management plane) – Slouží k přístupu přímo k zařízení, ke konfigurace a updatu.

Správa IoT uzlů je často realizována standartně vzdáleným přístupem skrz Telnet, SSH nebo webové rozhraní. Tato forma není ovšem vhodná pro velký počet zařízení v IoT sítí, a proto je potřeba lepší a automatizovanější způsob. V poslední době někteří výrobci zařízení nevěnovali příliš velkou pozornost zabezpečení komunikačního kanálu směrem ke koncovým zařízením. Použití velmi známých a jednoduchých hesel ve spojení s nezabezpečenými způsoby vzdáleného přístupu se stovky tisíc zařízení (konkrétně IP kamer) staly oporou při největším DoS útoku vůbec (Mirai botnet). Proto je velmi důležité zabezpečení nejen přenosu dat, ale i samotné správní úrovně zařízení, aby se předešlo jejich nevhodnému použití. Toto se ovšem netýká jen průmyslových zařízení, ale i spotřebitelských zařízení v domácnostech. Zde jsou v sázce bankovní informace, zdravotní záznamy a další osobní informace, které se skrz nezabezpečené zařízení mohou dostat do špatných rukou.

Správní úroveň je většinou zabezpečena pouhou aktualizací firmwaru na zařízení, ale zrovna tato věc je často přehlížena. I na první pohled obyčejné zařízení, jako chytrý kávovar, může být odrazovým můstkem k útoku většího rozsahu.

Opatření, která zvládnout i běžní uživatelé a spotřebitelé, a která pomohou minimalizovat toto riziko jsou: (1) Ujistit se, že zařízení může být jednoduše aktualizováno, (2) Nakupovat pouze u prověřených prodejců, (3) Připojit zařízení k samostatnému segmentu sítě nebo využít virtuální LAN (VLAN), (4) Pravidelně kontrolovat a aplikovat aktualizace, (5) Ujistit se, že všechna výchozí uživatelská jména a hesla byla změněna, (6) Ujistit se, že přístup k zařízení je pouze od důvěryhodného zdroje v síti, a poslední (7) Vypnout všechny nepotřebné funkce.

8.2.1 Horizontální a vertikální trh

Horizontální trh splňuje společné nebo obdobné potřeby pro velký rozsah různorodých odvětví, jako zdravotnictví, výroba, energetika a veřejnost. Například společnosti zabývající se zabezpečením, informačními technologiemi a finančními službami operují na horizontálním trhu. Produkty horizontálního trhu musí být propagovány více různorodým společnostem a bývají méně specializované. Například systém pro chytré osvětlení je využitelný v každé společnosti nehledě na odvětví.

Druhým typem je trh vertikální. Ten se skládá ze společností, které nabízejí služby a produkty konkrétní skupině klientů. Běžnými vertikálními trhy jsou automobilový průmysl, bankovnictví, vzdělávání, zdravotnictví apod. Je tedy vidět, že jedno odvětví může figurovat v obou trzích. Je to způsobeno tím, že trh nerozdělují samotná odvětví, ale konkrétní produkty a služby. Zmíněné osvětlení je možné implementovat v továrně i v nemocnici, ale systém pro uchování zdravotnických záznamů v továrně asi uplatnění nenajde.

Nehledě na odvětví, IoT představuje nové příležitosti pro interakci mezi různými zařízeními. To nabízí nejen nové podnikatelské příležitosti, ale i nové možnosti v oblasti tzv. customer experience, což by se volně dalo přeložit jako pocit, který si návštěvník odnese. Takové systémy se například používají na sportovních a kulturních akcích a cílí především na zapojení fanoušku a diváků. Není to tedy jen o integraci IoT zařízení, ale i tom, jak se daná zařízení využijí

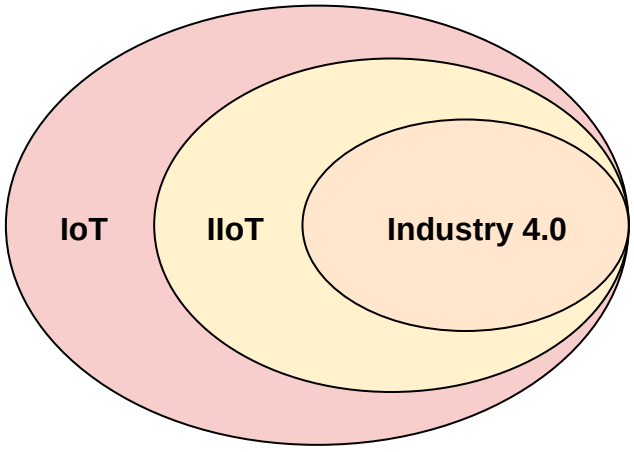
8.3 Industry 4.0

Termín Industry 4.0, někdy označovaný jako 4. průmyslová revoluce, byl poprvé představen v Německu v roce 2011. Podstata Industry 4.0 je rozsáhlé začleňování výpočetních a komunikačních systémů, které by vedly k vysoké úrovni automatizace a zároveň k masové výrobě a produkci vysoce přizpůsobených služeb, díky vysoké flexibilitě jednoduše programovatelných, konfigurovatelných a řízených výrobních linek. Poslední roky bylo ve znamení znatelného pokroku ve výkonných, ale nízko energetických procesorech, pamětech a komunikačních zařízeních. Ty s sebou přinesli obrovské výpočetní možnosti pro obrovské množství spotřebitelských zařízení. Téměř každá spotřebitelská elektronika dnes může obsahovat přívlastek „smart“. To posloužilo jako základ evoluce vestavěných systémů a služeb až na úroveň internetu věcí, dat a služeb. Dnes se příklady tohoto vývoje dají najít téměř v každé činnosti. To vše nasvědčuje tomu, že vývoj Industry 4.0 (tedy nástup 4. průmyslové revoluce) je přirozeným evolučním krokem v době, kdy průmyslové technologie ovlivňují a jsou ovlivňovány pokrokem spotřebitelských technologií v oblasti IoT.

Průmysl nejvíce ovlivnili pokroky především v oblasti měřících systémů v kombinaci s vestavěnými systémy a komunikačními sítěmi. Důležité je především to, že díky senzorů byla zaplněna mezera mezi fyzickým a digitálním světem, čímž poskytuje mnohonásobně bohatší informace a podporuje tak inteligentní řídící systémy. Ztělesněním iniciativy Industry 4.0 je „smart“ továrna. Tento koncept je založen na hierarchickém uspořádání zmíněných kyberfyzických systémů, ve kterém je chytrá produkce propojena na více úrovních za účelem dosažení vysokého stupně automatizace, flexibility, autonomie, efektivity, odolnosti, bezpečnosti a nízkých nákladů. Celá vize chytré továrny vypadá tak, že stroje budou propojeny v chytrých dílnách (smart plant), materiály a zdroje budou efektivně řízeny, výrobní proces bude řízen v reálném čase s minimálním využitím zdrojů, rekonfigurace a reorganizace výrobních procesů a customizace produktů bude realizovatelná v reálném čase a zákazníci budou moci sledovat průběh výroby svého produktu [38].

8.4 Industrial IoT (IIoT)

Samotná oblast IIoT by se dala nazvat obecným konceptem aplikace IoT v průmyslovém sektoru. Může se zdát, že pojmy IIoT a Industry 4.0 znamenají prakticky to samé, ale není tomu tak. IIoT je často chápáno jako zobecnění Industry 4.0 a zaměřuje se, kromě efektivity výroby, i na správu, údržbu majetku a další související operace. Industry 4.0 by se tedy dalo považovat za podmnožinu IIoT, a to je podmnožinou celého IoT (viz Obrázek 29). Čím se ale IIoT liší od IoT? Jsou to především technologie a požadavky, které v oblasti průmyslu vedou ke specializovaným a náročným řešením, které často vyžadují netradiční a originální přístupy. To a spousta dalších faktorů vedly k tomu, že se začaly vyvíjet samostatné koncepty, strategie, technologie a řešení, od úrovně zařízení, až po samotné služby. Z toho důvodu je potřeba proces vývoje oblasti IIoT koordinovat a monitorovat. Tohoto nelehkého úkolu se zhostila takzvaná konsorcia podnikatelů (sdružení podnikatelů za účelem dosažení konkrétního cíle), například Industrial Internet Consortium (IIC), která plní vůdčí roli v oblasti IIoT. Společnost General Electric v roce 2012 představila pojem Industrial Internet a společně s tím označili technologie a procesy, jako machine-to-machine (M2M), SCADA, human-machine interface (HMI), průmyslovou analýzu dat a kyberbezpečnost za nejdůležitější složky vize IIoT [38].



Obrázek 29 Vztah IoT, IIoT a Industry 4.0
Zdroj: vlastní zpracování (podle [38])

8.5 Architektury IIoT

S neustálým rozvojem IIoT systémů a služeb roste i potřeba architektur, které kromě efektivního využití zajistí i interoperabilitu služeb, velkého množství zúčastněných stran, kyberfyzických systémů, komunikačních systému, poskytovatelů služeb a dalších. To znamená, že je potřeba velké úsilí k tomu, aby byly vyvinuty standardy a referenční architektury, které budou zúčastněnými stranami přijaty. V roce 2012 agentura The International Telecommunication Union (ITU) vydala doporučení s označením ITU-T Y.2060 [50], které obsahovalo referenční architekturu pro IoT obecně, včetně aplikací, které spadají do kontextu IIoT (smart grid, e-health apod.). Kromě ITU i IIC konsorcium pracovalo na vlastní architektuře s názvem Industrial Internet Reference Architecture (IIRA). Na architekturu IIRA se nahlíží, jako na detailnější a specializovanější verzi ITU, jelikož je primárně zaměřená na oblast IIoT, a ne na IoT obecně [38].

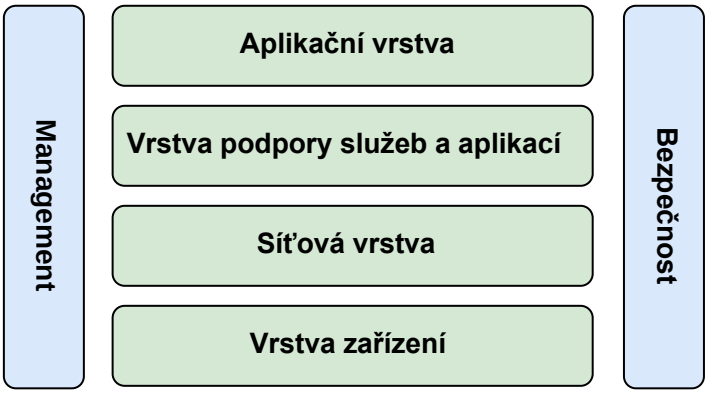
8.5.1 ITU architektura

Následující kapitola vychází z oficiálního dokumentu od unie ITU Overview of the Internet of things s číslem Y.2060 [50].

Architektura ITU je pojata velmi obecně a zabývá se především základními charakteristikami IoT, které se následně snaží rozšířit a obohatit. První takovou snahou je zahrnout koncept komunikace všeho (any THING) spolu s komunikací kdykoli (any TIME) a kdekoli (any PLACE). Jako „věci“ jsou v ITU označeny fyzické a virtuální objekty, které jsou identifikovatelné a dokážou se připojit do komunikační sítě a zároveň disponují informacemi. Důležitou částí ITU je komunikační model, který znázorňuje celkem tři komunikační situace mezi zařízeními, a to: (1) Komunikace skrz komunikační síť s využitím gateway, (2) Komunikace přímo skrz komunikační síť bez gateway, (3) Komunikace na přímo bez komunikační sítě a gateway. Je možná kombinace situací 1 a 3, 2 a 3.

Jak bylo zmíněno, ITU je v základu postavené na obecných charakteristikách IoT, které vedou k požadavkům, které musí referenční architektura splňovat. Hlavními požadavky zmíněnými v ITU jsou interoperabilita, připojení založené na identifikaci, autonomie služeb a sítě, využití lokalizačních služeb, zabezpečení, soukromí a schopnost spravovat zařízení a služby. Na základě těchto požadavků byl

představen několika úrovňový ITU model (viz Obrázek 30). Ten se skládá ze čtyř hierarchicky uspořádaných vrstev a dvou vertikálních, které zasahují do všech úrovní.



Obrázek 30 ITU referenční model
Zdroj: vlastní zpracování (podle [50])

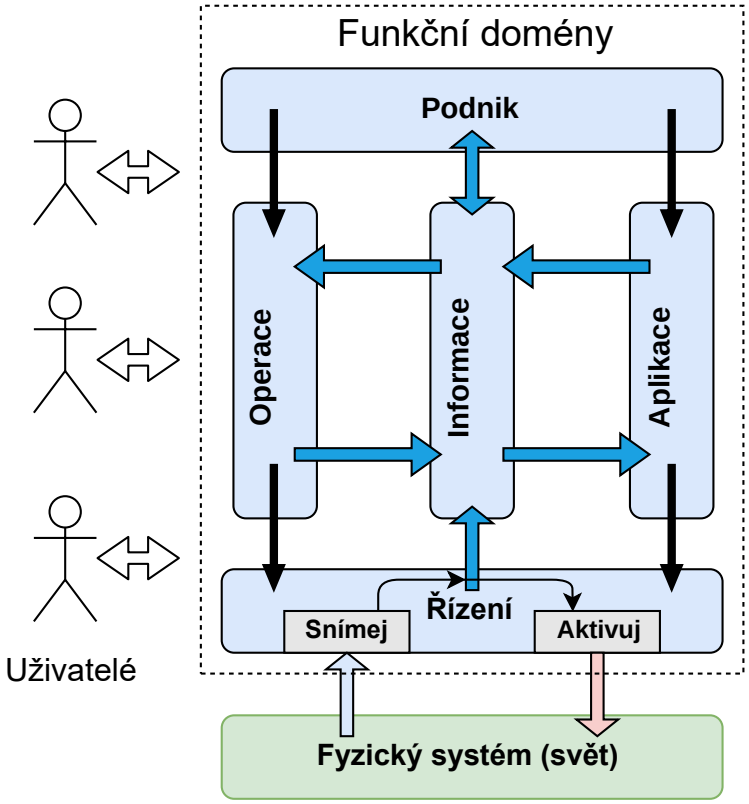
Nejnižší vrstvou model je **vrstva zařízení** a komunikačních bran. Model definuje tři základní typy zařízení podle jejich možností, které vycházejí z komunikačních modelů uvedených výše. Kromě zařízení tato vrstva obsahuje i všechny související komunikační technologie a konverzi mezi nimi, což zajišťuje vysokou interoperabilitu. **Síťová vrstva** poskytuje funkcionalitu odpovídající síťové a transportní vrstvě OSI modelu. To znamená vhodné funkce pro síťové připojení, jako pro řízení přístupu, správu mobility, autentizaci, autorizaci a správu účtů (AAA). Z transportní vrstvy poté funkce zaměřující se na doručení IoT služeb a pro ně relevantních dat a přenos řídících a správních informací. **Vrstva podpory služeb a aplikací** obsahuje běžnou i specifickou funkcionalitu pro IoT aplikace a služby. Tím se rozumí například zpracování a uložení dat, spolu se speciálními funkcemi s ohledem na požadavky jednotlivých aplikací a služeb. A poslední horizontální je **aplikační vrstva**, která obsahuje samotné aplikace a služby. První vertikální vrstvou je **vrstva managementu**. Ta obsahuje běžnou i specifikou funkcionalitu. Jedná se například o řízení konfigurace, topologie, zdrojů, výkonu, chyb, zabezpečení a účtů. Podobně je to i s celkově poslední **vrstvou zabezpečení**. Ta také zahrnuje běžné funkce, jako autentizace, autorizace, integrita, důvěrnost, soukromí, bezpečné routování, kontrola přístupu apod. Tyto funkce implementuje na všech horizontálních vrstvách.

S ohledem na množství účastníků a jejich různých cílů a zájmů je v dokumentu prezentováno i pět základních modelů z pohledu pěti rolí, které mohou účastníci zastávat. Tyto role jsou: poskytoval zařízení, poskytovatel sítě, poskytovatel platformy, poskytovatel aplikace a uživatel. Podle názvu je již zřejmé, co jaká role obstarává. Zmíněných pět modelů se poté odvíjí od počtu účastníků, kteří dané role plní. První model je založen na tom, že jedna organizace plní roli poskytovatele zařízení, sítě, platformy, aplikace a druhý účastník pouze roli uživatele. S každým modelem roste počet účastníků až do bodu, kdy pátý model obsahuje pět účastníků a každý z nich vykonává jednu roli.

8.5.2 IIRA architektura

Referenční model IIRA se více zaměřuje na funkční architekturu IIoT systémů, především na integraci stávající průmyslových řídících systémů (ICS) spolu s informačními technologiemi do jednoho sjednoceného modelu tak, aby vyhovoval požadavkům všech účastníků. Spojení ICS a IT s sebou ale přináší několik výzev, jelikož systémy OT byly vyvinuty jinou evoluční cestou než technologie IT. Jenže nástup pokročilejších senzorů, aktuátorů, procesorů a pamětí umožnilo průmyslovým řídícím systémům vykonávat mnohem komplexnější operace, které by dříve vyžadovaly speciálně navržené IT systémy. Na druhou stranu se komplexnější ICS systémy staly více zranitelnými vůči selháním a útokům, což vede k dalším požadavkům na jejich funkcionalitu.

Právě integraci IT a OT systémů do jednoho modelu se architektura IIRA snaží znázornit. Tento funkční přístup rozděluje IIoT systémy do pěti hlavních domén, z nichž každá reprezentuje skupinu specifických funkcí. Těmito pěti doménami jsou: řízení, operace, informace, aplikace a podnik. Uspořádání domén spolu se směry toku dat a řízení jsou znázorněny na obrázku (viz Obrázek 31), modře vyplněné šipky reprezentují tok dat a černé šipky značí řízení [38].



Obrázek 31 IIRA referenční model
Zdroj: vlastní zpracování (podle [51])

Doména řízení reprezentuje celou řídící smyčku (control loop) realizovanou ICS systémem. Konkrétně zahrnuje senzory, logiku, akce a to představuje tzv. plant realizovanou několika ICS systémy. Pojem „plant" v teorii řízení označuje věc, která je řízena. Ovšem přesný význam tohoto termínu není specifikován a „plant" může zahrnovat celý systém kromě nebo včetně částí, které implementují řízení. V kontextu této kapitoly pojem „plant" označuje kompletní proces implementované řídící smyčky. Jednoduchá řídící smyčka by se dala popsat, jako zařízení se senzorem odesílá data do řídícího centra, které zasílá příkazy na vykonaní akce aktuátoru na daném zařízení. **Operační sféra** obsahuje funkce, které jsou potřeba k vykonání akcí kontrolního systému v doméně řízení. Kromě toho je součástí i monitorování a management systému spolu s optimalizací k dosažení maximální účinnosti. **Informační doména** plní roli sběrače dat ze všech ostatních částí systému a následně v ní dochází k analýze za účelem podpory rozhodování na vysoké úrovni. **Aplikační vrstva** se skládá z funkcí, které jsou přímo závislé na konkrétní aplikaci, včetně API a uživatelských rozhraní, tak aby ostatní aplikace a uživatelé mohli

funkce využít s maximální efektivitou. Poslední, **vrstva podniku**, zahrnuje systémy a funkce, které se podílejí na řízení a rozhodování na podnikové úrovni, jako ERP a MES systémy [51].

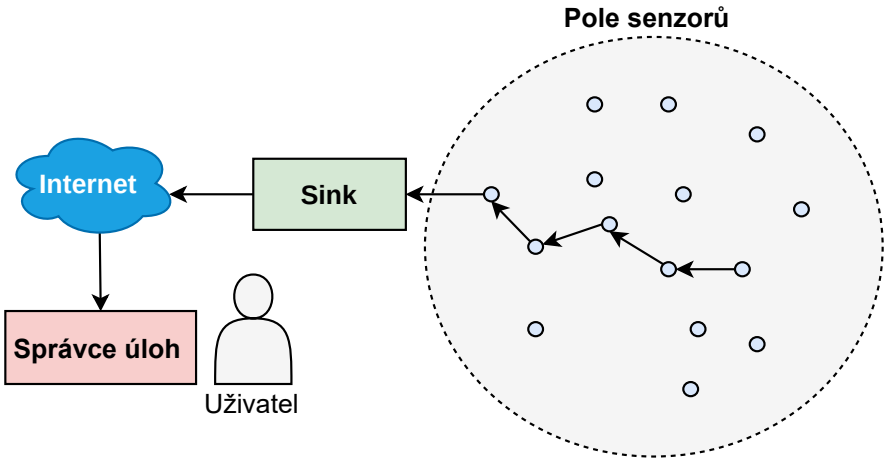
Je třeba poznamenat, že architektura IIRA se soustředí na koncept control plant. To znamená, že se zabývá všemi hledisky okolo řídící smyčky, která realizuje konkrétní plant. Jelikož řídící smyčka může být jednoduchá a skládat se z jednoho systému, nebo komplexnější, kdy je několik systémů uspořádáno v hierarchii, je možné dekompozici IIRA modelu na funkční domény použít na všech úrovních. Takže rozdělení IIoT systému na domény není ekvivalentní s přístupem architektury ITU, kdy dochází k vrstvení systému, ale jedná se o logické rozdělení funkcionality v jedné vrstvě nebo napříč více vrstvami systému. Z toho důvodu IIRA definuje průřezové funkce (cross-cutting), které jsou nezbytné pro vývoj kompletní IIoT aplikace. Mezi tuto funkce se řadí připojení, data management, analytika, inteligentní a stabilní řízení apod. Například připojení musí být realizováno hierarchicky s ohledem na standardy a praktiky týkající se propojení komponent uvnitř jednoho ICS nebo napříč více různými systémy. U těchto průřezových funkcí je poté možné vysledovat, že tvoří vrstvenou architekturu podobnou ITU modelu. Lze tedy považovat oba přístupy za komplementární (doplňují se navzájem) a IIRA architekturu za generalizaci ITU. Tato souvislost a komplementarita je více zřejmá z pohledu implementace, jelikož zahrnuje veškeré potřebné technické a technologické detaily pro kompletní IIoT systém, včetně funkcí systému, komunikačních a síťových protokolů, typů rozhraní a další [38].

8.5.3 Bezdrátové senzorové sítě (WSN)

Informace převzaty z publikace Overview of Wireless Sensor Network od autorů Matin a Islam, který představuje jednu z možných infrastruktur pro senzorové sítě [52].

Bezdrátové senzorové sítě (Wireless Sensor Networks – WSN) by se daly definovat jako sítě, které jsou samo konfigurovatelné (self-configured) a bez infrastruktury (infrastructure-less), k monitorování fyzikálních nebo enviromentálních veličin, jako teplota, zvuk, vibrace, tlak, pohyb a kooperativně dopravit získaná data skrz síť do základny, nebo do místa označeného pojmem „sink

node“, kde budou data zpracována a analyzována. Sink node nebo základna plní roli rozhraní mezi uživateli a sítí. Získání informací ze sítě je možné zadáním dotazů a sbíráním výsledků ze sink node. Typicky WSN síť obsahuje až statisíce senzorů, které mezi sebou komunikují pomocí rádiových signálů. Součástí snímacího zařízení bývá většinou senzor, výpočetní zařízení a rádio přijímač. Po nasazení jsou senzory zodpovědné za sebeorganizaci a vhodnou síťovou infrastrukturu, která je nejčastěji tvořena na principu multi-hop komunikace. Pracovní režim zařízení může být buď nepřetržitý, nebo závislý na událostech. Někdy jsou snímací zařízení vybavena i aktuátory a v takovém případě lze hovořit o Wireless Sensor and Actuator Network (WSAN). Návrh WSN sítí vyžaduje nekonvenční přístupy z důvodu některých omezení těchto řešení. Vzhledem k tomu, že hlavní požadavek na WSN je nízká úroveň složitosti zařízení společně s nízkou spotřebou, je velmi důležité najít správnou rovnováhu mezi komunikační schopností a schopností zpracovat signál a data. S cílem navrhnout pro WSN odpovídající protokol a algoritmus bylo provedeno několik výzkumů a bylo předloženo několik různých návrhů. Nedostatkem těchto prvotních studií ovšem bylo to, že předpokládaly všechna snímací zařízení za homogenní. V posledním desetiletí se nové výzkumy provádějí již s předpokladem heterogenních zařízení. Spolu s novými síťovými architekturami a pokroky v měřících technologiích se limity WSN posunuly a nabízejí tak nové možné a širší spektrum aplikace. To, jak může vypadat jednoduchá struktura WSN, je znázorněno na následujícím obrázku (viz Obrázek 32).



Obrázek 32 Struktura bezdrátové senzorové sítě
Zdroj: vlastní zpracování (podle [52])

Díky své flexibilitě při řešení problémů se WSN sítě staly velmi populární hned v několika oblastech. Příklady využití jsou:

- **Vojenské aplikace** – WSN může být součástí vojenského velení, řízení, komunikace, výpočetní techniky, zpravodajství, průzkumných a zaměřovacích systémů apod.
- **Monitorování oblastí** – Senzory pokrývají určitou oblast, kde je cílem sledování konkrétní události. Když jeden ze senzorů podnět detekuje, nahlásí to základně a ta provede odpovídající akce.
- **Transport** – WSN lze využít pro získávání dopravních informací v reálném čase, které jsou následně využity v dopravních modelech nebo k varování řidičů o zácpě či jiných dopravních problémech.
- **Zdravotnické aplikace** – Některé aplikace ve zdravotnictví obsahují WSN k monitorování pacientů, diagnóze, pro řízení podávání léků v nemocnici, monitorování fyziologických dat a ke sledování doktorů.
- **Snímání prostředí** – S touto aplikací se více pojí pojem Enviromental Sensor Network (ESN), který je aplikací WSN a v této oblasti se takové sítě používají pro detekci lesních požárů, měření znečištění ovzduší, detekci sesuvů půdy a pro sledování podmínek ve sklenících.
- **Monitorování struktur** – Využití nacházejí i pro monitorování pohybu ve strukturách (pohyb samotných objektů, ne lidský pohyb), jako mosty, nadjezdy, náspy, tunely atd., což technikům zjednodušuje pravidelné kontroly bez nutnosti návštěvy daného místa.
- **Průmyslové monitorování** – V průmyslu našli zásadní využití v údržbě na základě stavu stroje (condition-based maintenance – CBM), čímž výrazně snižují náklady na údržbu. V kabelových systémech je počet senzorů limitován především cenou samotných kabelů.

Struktura WSN sítě

Struktura WSN sítě může mít podobu několika známých topologií. První typickou topologií je obyčejná hvězda. Hvězda označuje uspořádání, kdy jedna základna může posílat nebo přijímat zprávy z od několika uzlů/senzorů. Senzory

nemají povoleno komunikovat mezi sebou. Výhodou této topologie je její jednoduchost, minimální spotřeba uzlů a velmi malou prodlevu při komunikaci se základnou. Nevýhodou je to, že základna musí být v dosahu všech závislých uzlů a síť s takovým uspořádáním není příliš robustní, jelikož komunikace v určité oblasti je závislá na jednom uzlu. Další běžnou topologií je mesh. V taková síti mohou všechny uzly komunikovat navzájem pokud jsou v dosahu. Právě tato topologie umožňuje multi-hop komunikaci. Výhodou mesh je škálovatelnost a redundance. Termín redundance znamená, že pokud nějaký uzel selže, zpráva může do cíle putovat přes jiné, pokud jsou v dosahu odesílatele. Nevýhodou je vyšší spotřeba energie při komunikaci multi-hop a delší čas doručení zprávy, který roste s velikostí sítě. Poslední topologií je kombinace mesh a hvězdy, která nese označení hybrid star mesh. Tato topologie poskytuje robustnost topologie mesh a zároveň nízkou spotřebu hvězdy. V tomto uspořádání senzory s nejnižším výkonem nemají schopnost přeposílat zprávy, ale ostatní uzly stále fungují na principu multi-hop, což umožňuje směrovat zprávy, ze zmíněných senzorů, jiným uzlům v síti. Uzly, které mají schopnost přeposílat zprávy často disponují větším výkonem a tím pádem vyšší spotřebou, a proto se běžně připojují přímo do elektrické sítě. Tuto topologii například využívá známý protokol ZigBee.

Jak bylo znázorněno na obrázku (viz Obrázek 32), senzory jsou umístěny v senzorovém poli a každý z nich má schopnost data sbírat a směrovat do sink node a ke koncovým uživatelům. Sink node komunikuje se správcem úloh skrz internet nebo satelitní připojení. Komunikaci v síti a celkově její fungování lze znázornit pomocí modelu, který obsahuje pět horizontálních a tři vertikální vrstvy, přesněji planes. Konkrétně se jedná o vrstvu aplikační, transportní, síťovou, data linkovou a fyzickou, které jsou uspořádány horizontálně v tomto pořadí od nejvyšší, a o vertikální úroveň správce úloh, řízení mobility a řízení spotřeby. Na aplikační vrstvě můžou být různé typy softwaru v závislosti na úloze sítě. Transportní vrstva pomáhá udržet tok dat v senzorové síti. Síťová vrstva má na starost směrování dat z transportní vrstvy a specifikuje komunikační protokoly mezi senzory a sink node. Data linková vrstva je zodpovědná za multiplexování datových proudů, detekci rámců, řízení přístupu k médiím (MAC) a kontrolu chyb. Fyzická vrstva poskytuje metody k modulaci, výběru frekvence, šifrování dat a samotnému odesílání a

přijímání dat. Tři vertikální úrovně pomáhají monitorovat výkon, pohyb a rozdělení úkolů ve WSN.

Komunikační protokoly a algoritmy ve WSN

Hlavním posláním senzoru v takové síti je získat data a zaslat je základně v prostředí multi-hop komunikace, kde základem je určení této trasy. K tomu existuje obrovské množství protokolů a strategií, které lze ve WSN sítích využít, a které berou v potaz limitovaný výkon a zdroje, s časem se měnící kvalitu přenosu a možnost ztráty paketů.

K těm nejjednodušším patří **flooding** (zaplavení). Jedná se o často používanou techniku pro vyhledávání cest a šíření informací v kabelových i bezdrátových sítích. Tato technika nevyžaduje nákladné údržby, složité topologie a algoritmy. Princip flooding je jednoduchý. Každý uzel pošle paket všem svým sousedním uzlům, které udělají to samé a paket tímto způsobem eventuelně dojde k cíli. Problémem této techniky je to, že může dojít k nekonečné replikaci paketů. Podobným způsobem funguje i další přístup zvaný gossiping (povídání). Funguje na téměř stejném principu jako flooding, ale rozdílem je to, že uzel neposílá data všem sousedním, ale pouze jednomu náhodně vybranému uzlu. Tento proces je opakován na každém uzlu, dokud zpráva nedojde do cíle, nebo není dosaženo maximálního počtu hopů. Skupina WSN protokolů, které využívají flooding a gossiping k šíření informací po síti, se označují zkratkou Sensor Protocols for Information via Negotiation (SPIN).

Dalším routovacím algoritmem pro senzorové sítě je **Low-Energy Adaptive Clustering Hierarchy** (LEACH). Je navržen pro sběr a doručení dat do sink node. Hlavním cílem LEACH je prodloužení života sítě, snížení spotřeby energie jednotlivých uzlů a s využitím agregace dat snížit počet přenášených zpráv. Za tímto účelem LEACH využívá hierarchický přístup a síť rozděluje do několika částí (clusterů). Každý cluster je řízen hlavou clusteru (cluster head), která zodpovídá za několik úkolů. Prvním úkolem je sběr dat od uzlů v daném clusteru a agregovat je do jedné zprávy. Druhým úkolem je přenést agregovaná data základně jediným hopem. Třetí je tvorba Time Division Multiple Access (TDMA) rozvrhu, což znamená, že každý uzel má přiřazený čas, ve kterém může vysílat. Tento rozvrh poté vysílá všem

členským uzlům v daném clusteru. Obecně by se základní operace LEACH daly rozdělit do dvou fází. První přípravná fáze, ve které dochází k rozdělení na clustery a volbě hlav clusterů a druhá ustálená fáze, kde se provádí sběr, agregace a doručení dat základně.

A posledním příkladem protokolů jsou dva velmi podobné hierarchické protokoly **Threshold-sensitive Energy Efficient Network** (TEEN) a **Adaptive Periodic Threshold-sensitive Energy Efficient Network** (APTEEN), které se využívají především v aplikacích, kde hlavní roli hraje čas. V TEEN senzor snímá nepřetržitě, ale data jsou přenášena méně častěji. Hlava clusteru nastaví pro cluster pevnou prahovou hodnotu (hard threshold), která určuje požadovanou hodnotu měřené veličiny, a měkkou prahovou hodnotu (soft threshold), která reprezentuje změnu v naměřené hodnotě a aktivuje vysílač na senzoru. Hard threshold tedy výrazně snižuje počet přenášených zpráv tím, že uzly data přenášejí jen pokud se naměřené hodnoty nachází v požadovaném rozsahu. Soft threshold snižuje počet přenosů ještě více tím, že k přenosu nedochází, pokud v měření došlo k žádné nebo velmi malé změně. Hlavní nevýhodou tohoto protokolu je to, že pokud senzory neobdrží prahové hodnoty, nebudou vůbec vysílat a nebude z nich možné získat data.

Kromě těchto zmíněných existuje ještě několik dalších známých a často používaných protokolů, jako Power-Efficient Gathering in Sensor Information Systems (PEGASIS), Geographic Adaptive Fidelity (GAF) a Directed Diffusion, které již nebudou v této práci představeny, jelikož se nijak výrazně neliší od předchozích protokolů.

8.6 Aplikace IIoT

IIoT aplikace lze najít ve velkém množství IoT oblastí. OT systémy se staly standartní výpočetní platformou pro řízení a fungování těch nejdůležitějších infrastruktur. Spojení vysokých výpočetních a úložných kapacit PLC a SCADA systémů, které umožňují řízení aplikací v reálném čase a vysokou dostupnost, je dělá velmi populárními při stavbě infrastruktur i mimo produkční odvětví, pro které byly původně vytvořeny. V současnosti je velká část infrastruktur založena právě na systémech ICS, z nichž nejvýznamnější je energetický sektor. Produkce a zpracování

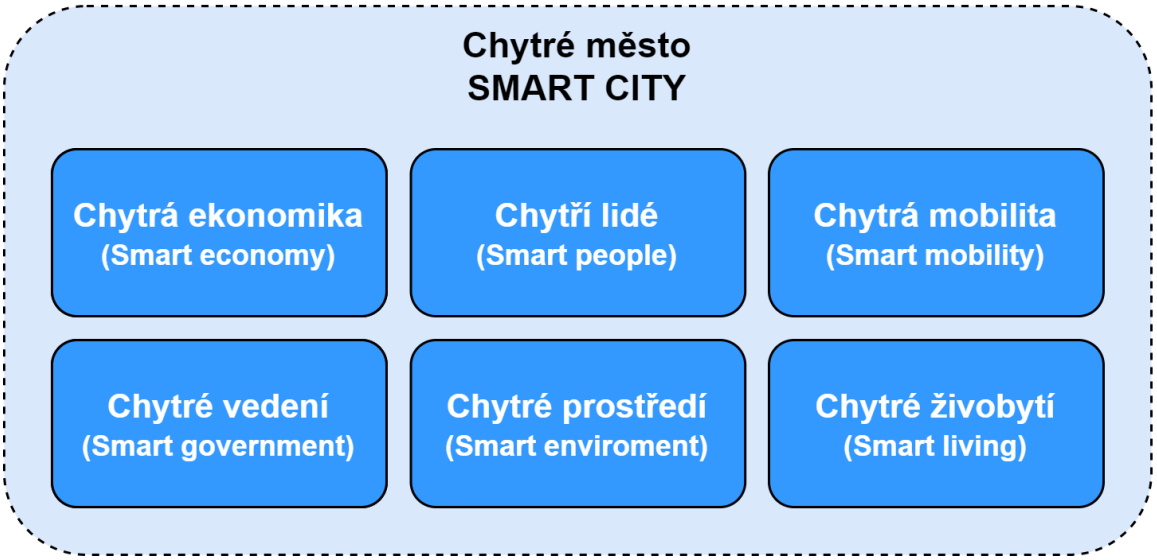
energie je součástí těžkého průmyslu země, a proto se přirozeně hodí pro jejich využití. I ostatní distribuční sítě využívají tento model založený na ICS, konkrétně vodohospodářství, ropné a plynařské společnosti a v neposlední řadě i přepravní firmy [38]. Významné aplikace IoT a IIoT lze najít i v konceptu Smart City, který bude dále podrobněji popsán.

Všechny tyto oblasti vyžadují dodatečné nasazení a osvojení komponent, především kyberfyzických systémů a ICS, aby bylo možné dosáhnout představované podoby služeb ve velkém měřítku. Revoluce v IIoT je stále na počátku a sektory, které již v současnosti spoléhají plně na využívání ICS budou zastávat vedoucí roli v dalším vývoji. Obecně instalace IoT systémů ve velkém měřítku představuje nemálo výzev hlavně z důvodu vysoké složitosti a heterogenity použitých systémů. Ty jsou prohloubeny ještě více, pokud se vezme v potaz i omezení bezdrátových systémů, přísné požadavky na zabezpečení a požadavky na monitoring parametrů, které jsou vyžadovány rozhodovacími procesy. V porovnání s řešeními v malém měřítku, například v domácím prostředí, jsou i velice náchylné k chybám. Charakteristickým příkladem formalizované, ale chybové instalace, je metoda „outside-in“, kdy senzory, aktuátory a kontrolery jsou nainstalovány ještě před vytvořením samotné sítě a infrastruktury ve firmě. Pokud jsou ovšem tyto výzvy překonány, výsledkem je velice flexibilní IIoT systém, který zvyšuje úroveň autonomie uvnitř podniku [38].

8.7 Chytré město

V současnosti pro chytré město (smart city) neexistuje žádná veřejně uznaná definice. Většina definic ale poukazuje na využití informačních komunikačních technologií (ICT) ve veřejné správě a městských službách a na to, že smart city je řízené daty, které umožňují městským službám výrazně zlepšit kvalitu života občanů v oblastech, jako bezpečnost, transport, zdravotnictví, vzdělání, energetika, vodohospodářství a dalších [53]. Důležité je ovšem to, že chytré město nedělají jen využité technologie, ale je třeba začlenit i občany. Tento nápad je založen na moderním trendu decentralizace řízení s využitím podpůrných mechanismů. Tato vize byla dále významně ovlivněna čtvrtou průmyslovou revolucí, tzn. Industry 4.0.

Koncept chytrého města je založen na využití technologií IoT, automatizace produkce, automatizace manažerského rozhodování a na analýze velkého objemu dat (big data). Ovšem implementace tohoto konceptu je ještě složitější, než vypadá. Problémy nejsou spojené jen s inovacemi městské infrastruktury a dalších zmíněných oblastí, ale především s transformací celého správního systému města na všech úrovních [54].



Obrázek 33 Koncept Smart City
Zdroj: vlastní zpracování (podle [54])

Chytré město by se tedy dalo popsat jako spojení několika „smart“ oblastí a následné rozšíření o další aplikace. Mezi nejzákladnější indikátory patří chytrá ekonomika (Smart Economy), chytré vedení (Smart Government), chytré prostředí (Smart Environment), chytré živobytí (Smart Living), chytrá mobilita (Smart Mobility) a chytří lidé (Smart People) [54, 55]. Je třeba zdůraznit, že konceptů a pojetí smart city existuje spousta a tohle je příklad jednoho z nich.

Smart Economy zahrnuje veškerá opatření zaměřená na transformaci a posílení ekonomiky města. Součástí toho je zlepšování podnikatelského klimatu, atraktivity města pro start-upy a investory a celkově růstu ekonomiky inovační a udržitelnou cestou. Využití digitálních nebo chytrých technologií vede k ekonomické prosperitě, která poté generuje stabilní a příznivé podmínky pro všechny účastníky.

Hlavním cílem **Smart Government** je posílení interakce mezi vedením města a všemi zúčastněnými stranami, tzn. občany, podniky a ostatními společenskými organizacemi, ve městě. Vedení města musí přehodnotit kvalitu, měřítko a rozsah poskytovaných služeb pro občany a podniky, směrem k využití nových metod, jako spolutvorba (co-creation), což znamená, že názor spotřebitelů ovlivňuje výsledný produkt po celou dobu procesu výroby, a crowdsourcing, který označuje nový způsob dělby práce, kdy je ke spolupráci přizvána široká veřejnost, a ne vybraná skupina (outsourcing). Pomoci může i implementace nových technologií a inovací (pro občany, služby, řídící infrastrukturu). V posledních letech se v souvislosti s těmito metodami začal objevovat pojem „město jako služba“ (city as a service).

Smart Enviroment popisuje, jak vedení města pečuje o zastavěné oblasti a přírodní prostředí za účelem zlepšení pobytu občanů a návštěvníků města. Nejvíce diskutovanými tématy jsou snížení produkce odpadu, znečištění, emisí, dále energetika a s tím související přechod na tzv. lokální energii (energie je vytvářena lokálně blízko místa spotřeby). V současnosti se tyto řešení zahrnují již do územního plánování, a to výrazně zjednodušuje a urychluje jejich případnou implementaci.

Smart Living míří především na zvýšení kvality života pomocí inkluzivních strategií, napříč všemi věkovými a demografickými skupinami. Zaměřuje se na začlenění sociální a digitálních věcí, jako elektronické služby, připojení a sociální platformy, zlepšení zdravotnictví a zdravotní péče pro starší občany (eHealth, Ambient Assisted Living), bezpečnost, bydlení a chytré budovy. K lepší dostupnosti a zlepšení pocitu občanů (citizen experience) pomáhají nové metody pro angažování společnosti a technologie založené na IoT (LPWAN a WiFi).

Předposledním indikátorem je **Smart Mobility**. Tato oblast cílí především na zlepšení kvality služeb městského dopravního systému společně s osvojením nových dopravních řešení. Dosažení levnější, rychlejší a k prostředí šetrnější dopravy společně s integrovaným multimodálním systémem, je pro města největší výzvou. Patří sem i podpora nových způsobů dopravy, například elektrická vozidla, hydrogenem poháněná vozidla, autonomní vozidla, systémy sdílení kol a automobilů (bike sharing a carpooling), které jsou důležitým aspektem pro město vydávající se cestou smart.

A poslední jsou **Smart People**. Jak bylo v úvodu kapitoly zmíněno, chytré město nezahrnuje pouze technologie, ale i samotné občany. Konkrétně se tato oblast zaměřuje na transformaci způsobů, jakými občané interagují s veřejným a soukromým sektorem. Pro efektivnější poskytování informací a služeb je důležité vytvořit sociální a digitální rovnost. Kromě toho je zde kladen důraz i na vzdělaní s cílem zlepšení volby zaměstnaní a pracovních příležitostí. Toho se týká i rozvoj talentů, který může mít dlouhodobě kladný vliv na celou ekonomiku.

Stovky měst po celém světě již koncept smart city testují. Jedná se o města, jako New York, Tokyo, Shanghai, Barcelona, Tel Aviv, Stockholm, Amsterdam a každým rokem se k nim přidávají další. Největší počet projektů zaměřených na smart city má s přehledem Rusko. Aktuálně je možné zmínit i Egyptský projekt Nové Káhiry, která má být v budoucnu jejich novým hlavním městem, započatý před několika lety a konečně to vypadá, že město přivítá své první obyvatele Z pohledu postavení firem a značek na technologickém trhu jsou na prvních příčkách giganti, jako IBM, Cisco, Siemens, Microsoft, Toshiba, Huawei a další společnosti [56].

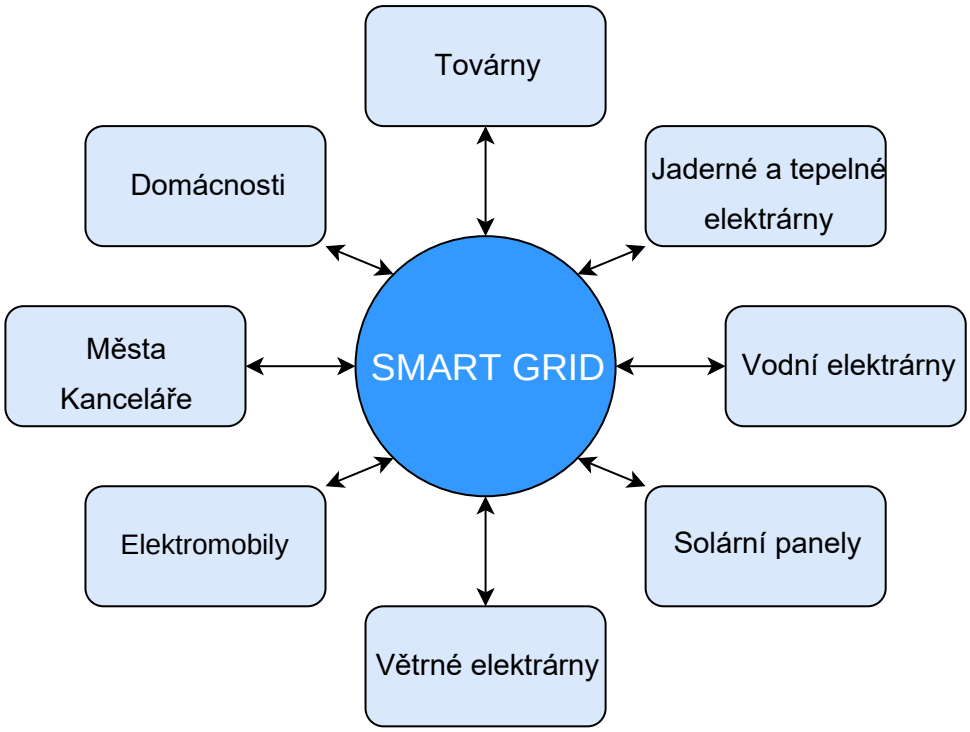
Jelikož osvojování IoT technologií a podobných principů probíhá ve všech oblastech velmi podobně, pro představu zde budou podrobněji představeny dvě velmi významné oblasti, a to energetika a zdravotnictví.

8.8 Chytrá energetika

Smart city je integrací vertikálních a horizontálních infrastruktur a systémů, a proto se někdy označuje „systémem systémů“. Ale systémem, který všem ostatním umožňuje fungovat, je chytrá rozvodná síť (Smart Grid). Ministerstvo energetiky Spojených států amerických Smart Grid popsalo jako automatizovanou a široce distribuovanou síť pro dodávku elektrické energie. Elektrická infrastruktura je jedním z nejdůležitějších a nepostradatelných nástrojů každého města. Nedostupnost elektrické energie totiž znamená nedostupnost všech ostatních služeb a systémů. Základní filozofií tohoto nového přístupu je osvojení dlouhodobých a efektivních strategií z ekonomického hlediska a zároveň plnit požadavky v otázce ochrany prostředí [56]. Hlavním rozdílem oproti stávající klasické elektrické síti je to, že Smart Grid využívá digitální technologie, a díky tomu je umožněn obousměrný provoz/přenos. Směrem od poskytovatele ke spotřebiteli

proudí elektrická energie. Opačným směrem, od spotřebitele k poskytovateli, proudí informace. Informacemi se rozumí sesbíraná data, z různých zařízení umístěné různě po sítí, například o spotřebě. Znázornění jednoduché struktury Smart Grid je vidět na obrázku (viz Obrázek 34). Typická struktura Smart Grid obsahuje objekty, které energii generují, objekty, které energii spotřebovávají a všechny jsou propojeny sítí. Mezi hlavní benefity Smart Grid patří například [57]:

a) Efektivnější přenos elektrické energie
b) Rychlejší obnovení dodávky energie v případě výpadku
c) Navýšení rozsáhlých systémů obnovitelné energie
d) Snížení nákladů na operace a správu
e) Zlepšení zabezpečení
f) Zlepšení systémů pro výrobu energie u spotřebitele (customer-owned)
g) Díky optimalizaci dodávky by mohlo dojít ke snížení sazeb za elektřinu



Obrázek 34 Koncept Smart Grid
Zdroj: vlastní zpracování (podle [57])

V praxi proces rozvoje sítě Smart Grid ve městě spoléhá na pět základních směrů [56]. První z nich je **podpora čistých (obnovitelných) zdrojů energie** (RES). Chytrá města jsou, mimo jiné, hodnocena i z pohledu užívání energie, která je šetrná k životnímu prostředí a Smart Grid řešení je základem pro podporu rozvoje obnovitelných zdroj a kogeneračních elektráren (společná výroba elektrické energie a tepla). S tím souvisí i integrace distribuovaných zdrojů, které umožňují spolehlivou a flexibilní dodávku energie.

Dalším směrem je **chytré měření**. Zde hraje největší roli právě obousměrná komunikace. Pokročilá měřící infrastruktura (AMI) umožňuje aktivní participaci spotřebitelů při kritických situacích, kdy je potřeba regulovat zátěž a při implementaci dynamických tarifů za účelem stimulace integrace RES a elektromobilů. Kromě toho AMI slouží jako zdroj informací při tvorbě předpovědí.

Třetí věcí je **efektivní veřejné osvětlení**. Zájem obcí o osvojení udržitelných řešení výrazně roste, především z důvodu optimalizace a účinnější spotřeby energie. Používají se především LED lampy a senzory, které automaticky vypínají a zapínají světla, když je potřeba, a to i uvnitř budov.

Předposledním je **integrace elektromobilů**. Největším problémem současnosti je znečištění ovzduší ve velkých městech, a právě využití elektrických automobilů se nabízí jako řešení. Jejich rozsáhlé nasazení ovšem vyžaduje inteligentní řešení pro elektrické sítě. Největší problém představuje řízení nabíjení baterií v těchto autech, jelikož by snadno mohlo dojít k přetížení elektrické sítě. Zároveň je to ale možnost využití přebytečné energie z obnovitelných zdrojů, především přes noc.

A poslední je **aktivní účast spotřebitelů**. Řešení Smart Grid je z většiny orientované na samotného spotřebitele. V této souvislosti se objevil nový pojem, a to prosumer (provider-consumer). Jedná se o člověka, který spotřebovává a zároveň generuje elektrickou energii. Spotřebitel se prosumerem může stát již ve chvíli, kdy například svůj elektromobil připojí do sítě, čímž se z něj stane malý zdroj energie, který může pomoci ve špičkách s regulací toku energie na síti.

Aby bylo možné těchto cílů dosáhnout, Smart Grid musí být vytvořen ve třech úrovních: Fyzická, ICT infrastruktura a standardy [56].

(1) **Fyzická infrastruktura** – Smart Grid je nezbytnou průmyslovou a ekonomickou revolucí s přihlédnutím na velmi zastaralé prvky, které je možné najít v rozvodnách a elektrické síti. Pokrok ve výpočetních technologiích a inovace v elektrotechnických materiálech umožňují vytvoření elektrického vybavení, pokročilé automatizace, ochrany a řídících systémů. V současnosti je většina těchto technologií již dostupných a je možné tedy učinit první kroky k tvorbě Smart Grid. Využití automatizace, komunikace a IT technologií v určitých úrovních dokáže zajistit vysokou spolehlivost distribučních sítí ve městech. Modernizace rozvoden a elektrické sítě je samozřejmostí, aby situace, jako izolace a lokalizace výpadku a zároveň její odstranění a obnovení dodávky, byly provedeny v co nejkratším čase.

(2) **ICT infrastruktura** – Inteligence je elektrické síti dodána díky softwarovým aplikacím a komunikační infrastruktuře, které spojuje operátora s komponenty. Pokrok v oblasti komunikačních technologií ICT je velmi prospěšný při tvorbě Smart Grid. Nasazení ICT v rozsáhlém měřítku umožňuje vzdálený přístup k informacím, včetně přístupu k řídícím a kontrolním systémům a tím pádem vzniká i riziko neoprávněného přístupu. Tím se ochrana proti kyberfyzickým útokům stává kritickou. Jedním ze standardů, který prezentuje strategie pro zabezpečení řešení týkající se Smart Grid, je americký NISTIR 7628.

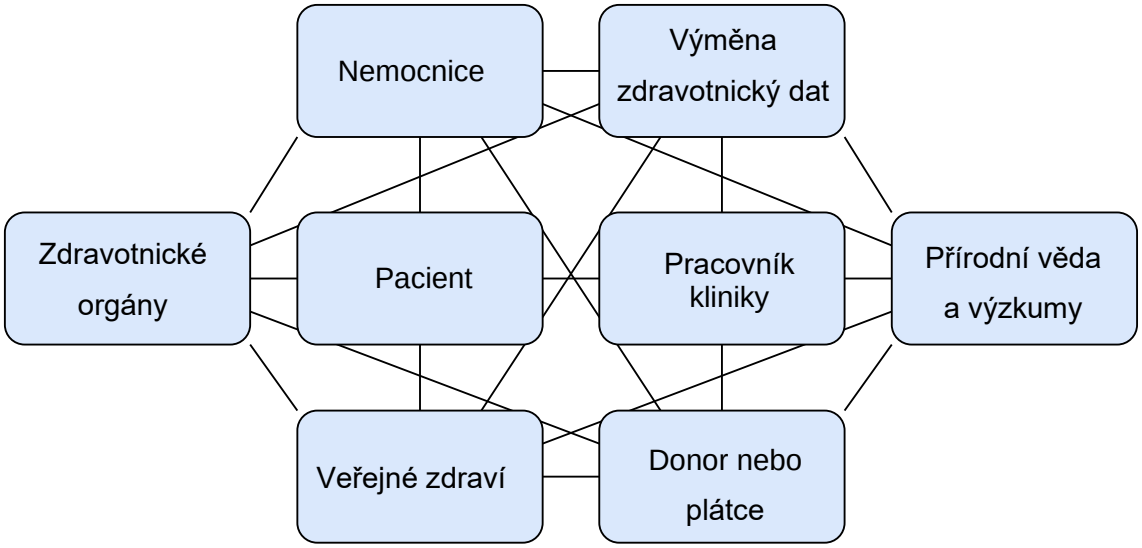
(3) **Standardizace** – Smart Grid, případně Smart City, kromě nových příležitostí představuje i bariéry kompatibility pro velké množství zařízení a vybavení od různých dodavatelů. Aby byla zajištěna hladká integrace všech prvků a zároveň co nejnižší operační náklady, je potřeba zajistit vhodnou standardizaci.

V Evropě se o standardizaci stará organizace The International Electrotechnical Comitee (IEC). Aby bylo možné co nejjednodušeji identifikovat vhodné standardy pro specifické aplikace, IEC vytvořila mapu standardů v doménách Smart Grid dostupnou na jejich webu [smartgridstandardsmap.com, 8.8.2021]. Podobně v USA poskytuje standardizace především The National Institute of Standards and Technology (NIST).

8.9 Chytré zdravotnictví

Tato kapitola vychází z publikace Smart Healthcare: Making medical care more intelligent od autorů Tian, Yan, Le Grange, Wang, Huang a Ye [58].

První návrhy na zapojení IoT technologií, superpočítačů a cloudu do zdravotnictví byly představeny v konceptu „Smart Planet“ v roce 2009 společností IBM. Smart Healthcare je zdravotnický systém, který využívá technologie, jako nositelné doplňky (wearables), IoT a mobilní internet, za účelem dynamického přístupu k informacím, spojení lidí, materiálů, zdravotnických institucí a aktivně řídit a reagovat na potřeby lékařského ekosystému. Chytré zdravotnictví se skládá z více účastníků, například doktorů, pacientů, nemocnic, výzkumných institucí a dalších. Znázornění struktury a propojení jednotlivých částí je naznačeno na obrázku (viz Obrázek 35). Dále by se dalo rozdělit i na dimenze, jako prevence onemocnění a monitorování, diagnóza a léčba, řízení nemocnic, rozhodování a zdravotnický výzkum. Základním kamenem chytrého zdravotnictví je kombinace IT (IoT, mobilní internet, cloud, big data, AI a další) a moderní biotechnologie. Z pohledu pacienta se jedná hlavně o wearables, které mohou pomoci s monitorováním jejich zdravotního stavu a přivoláním lékařské pomoci (např. skrz domácí asistenty). Z pohledu doktorů a nemocnic se jedná především o systémy pro podporu rozhodování (DSS), kteří pomáhají například s určením diagnózy. Dále i integrované informační systémy pro správu zdravotnických materiálů, například laboratorní informační systém, systémy pro archivaci obrázků a pro komunikaci (Picture Archiving Communication System – PACS) a systém elektronických zdravotních záznamů. V nemocnicích najde využití i technologie rádio-frekvenční identifikace (RFID), která se často používá pro správu materiálu a pro podporu zásobování. Je tedy zřejmé, že využití různých technologií, dokáže snížit náklady a rizika spojené se zákroky, zefektivnit správu zdravotnického materiálu, zlepšit kooperaci a výměnu informací mezi regiony a urychlit tak vývoj telemedicíny (přenos informací od pacienta k doktorovi), která cílí až k personalizovaným zdravotním službám.



Obrázek 35 Koncept Smart healthcare
Zdroj: vlastní zpracování (podle [1])

Diagnóza a léčba

S aplikací umělé inteligence, chirurgických robotů a smíšené reality se oblast diagnózy a léčby stala „chytřejší“. Umělá inteligence se stala součástí, již úspěšně otestovaných, DSS systémů, např. diagnóza hepatitidy, rakoviny plic a rakoviny kůže. Přesnost a úspěšnost výsledků byla prokázána vyšší než u lékařů. Dále i systémy založené na strojovém učení našli své místo především v patologii a zobrazování. Nejvýznamnějším DSS systémem v oblasti zdravotnictví je Watson od společnosti IBM. Jedná se o kognitivní inteligentní systém, který skrze hloubkovou analýzu klinických dat a literatury poskytuje optimální řešení. Watson se stal užitečným především při určování rakoviny a cukrovky. Díky takovýmto systémům mohou doktoři poskytnout diagnózu podloženou expertním systémem, čímž se sníží šance nesprávné diagnózy, špatného vyložení symptomů a jejich přehlédnutí. Pro pacienta to znamená dřívější a cílenější léčba.

Management zdravotnictví

Nový zdravotnický model věnuje více pozornosti pacientům a konkrétně jejich samostatnosti (self-management). Zdůrazňuje především sebe monitorování v reálném čase, okamžitou zpětnou vazbu na naměřená data a včasný zásah zdravotnických služeb, pokud je to vyžadováno. Rozvoj nositelných a implantovatelných zařízení, chytrých domácností a platforem smart healthcare,

které jsou propojené IoT technologiemi, je dokonalým řešením pro tento model. Především poslední generace těchto zařízení kombinuje pokročilé senzory, mikroprocesory a bezdrátové moduly, které umožňují nepřetržitě snímat mnoho fyziologických indikátorů.

Příkladem aplikace modelu smart healthcare jsou chytré domácnosti. Ty ovšem v kontextu zdravotnictví označují speciální domy nebo apartmány, které obsahují senzory a aktuátory integrované do obytné infrastruktury, a ty monitorují fyzický stav a okolí člověka. Role takových domů se dělí do dvou základní oblastí: automatizace domácnosti a monitorování zdravotního stavu. Spojení těchto technologií dokáže poskytnou základní služby a zároveň sbírat data o zdravotním stavu, pomáhat lidem, kteří potřebují neustálou péči a snížit tak jejich závislost na pečovatelích a celkově zlepšit jejich kvalitu života.

Pacienti mohou sledovat svůj stav skrze mobilní aplikace a informační platformy. Například systém pro detekci stresu a jeho zmírnění využívá nositelný senzor, který nepřetržitě monitoruje jejich krevní tlak a automaticky pomáhají tělu snížit stres. Naměřená data je možné poté využít spolu s ostatními daty k předpovědi možných hrozeb, které mohou následovat a varovat pacienta touto cestou.

Virtuální asistenti

Virtuální asistent není objektem, ale algoritmem, který s uživatelem komunikuje skrz rozpoznávání řeči a při vykonávání činnosti na základě příkazů spoléhá na big data. Nejznámějšími domácími virtuálními asistenty jsou Microsoft Cortana, Google Assistant a Apple Siri. Většinou pomáhají uživatelům s různými úlohami od upomínek až po automatizaci domácnosti. V kontextu chytrého zdravotnictví na sebe berou roli spojovatele mezi pacienty, doktory a zdravotnickými institucemi. Pacientovi dokážou pomoci s převedením běžných výrazů do odborných termínů, a díky tomu zlepší poskytnutí následné zdravotnické služby. Doktorům můžou asistenti pomoci při komunikaci s pacienty, kdy asistent automaticky odpovídá na základě informací předaných pacientem. Ve zdravotnických institucích může aplikace asistentů ušetřit lidské i materiální zdroje. Existuje i speciální technologie

s označením Nuance, která může pomoci s komunikací mezi virtuálními asistenty, hlavně mezi běžnými a specializovanými.

Problémy a řešení

Tři představené oblasti, ve kterých má IoT nezanedbatelný vliv, jsou jen příkladem a případů využití této technologie ve zdravotnictví je mnohonásobně více. Od příchodu smart healthcare bylo představeno několik vyspělých konceptů a systémů. Jenže s příchodem nových technologií přicházejí i nové problémy. V současnosti chybí především vedení a programová dokumentace, což vede k nejasným cílům ve vývoji a k plýtvání zdroji. Instituce v různých regionech a zemích nemají jednotné standardy, a to je příčinou snížení integrity dat. Dalšími problémy je i množství dat, které stěžuje jejich sdílení a komunikaci, nekompatibilita různých zařízení a systémů, nedostatečné právní normy, tykající se především osobních údajů a využití naměřených dat.

K vyřešení těchto problémů je nutné se zaměřit na dva aspekty: technologie a regulace. V otázce technologií je potřeba uspíšit dospívání a stabilitu stávajících systémů skrze časté updaty a upgrady. Co se týče regulace, je potřeba sestavit jednotné technické standardy, které docílí maximální kompatibility mezi různými zařízeními a platformami, čímž se zlepší nejen integrita dat, ale odstraní se i bariéry pro jejich výměnu. Poslední důležitou věcí je zajištění bezpečnosti dat pomocí aplikace příslušných řešení a pomocí legislativy.

9 Tvorba IoT řešení

Informace v kapitole Tvorba IoT řešení jsou převzaty z online kurzu Cisco [1], z části Chapter 6: Create an IoT Solution, pokud není uvedeno jinak.

V předchozí kapitole bylo IoT představeno jako technologie, která dokáže transformovat každé průmyslové odvětví. Ty samé technologie byly představeny jako jedny z klíčových pro řešení globálních problémů týkajících se samotné planety. Těmito problémy se rozumí například spalování fosilních paliv, znečištění ovzduší, znečištění oceánů, změna klimatu, hlad, nemoci, nerovnost pohlaví a přístup k čisté vodě a hygienickým prostředkům a zařízením. Na hledání řešení se podílejí i velmi významné organizace a lidé s vlivem na globální úrovni. Jedním

z nich je, jeden ze zakladatelů společnosti Microsoft, Bill Gates se svojí manželkou Melindou, kteří skrze svoji nadaci Bill & Melinda Gates Foundation darovali miliardy dolarů na globální zdravotnické projekty a zemědělský rozvoj. Druhým je i Elon Musk, který pomocí nadace The Musk Foundation finančně podporuje výzkumy zabývající se obnovitelnými zdroji energie, výzkumy v oblasti pediatrie a vzdělávání v oblasti věd a techniky.

V roce 2000 se na setkání vůdců ze 189 zemí vytvořil seznam osmi cílů, které měli být splněny v rozmezí 15 let, a to (1) Vymýtit extrémní hlad a chudobu, (2) Dosáhnout univerzálního základního vzdělání (umožnit všem dětem školní docházku), (3) Podpora rovnosti pohlaví a postavení žen, (4) Snížit úmrtnost dětí, (5) Zlepšit mateřské zdraví, (6) Bojovat proti chorobách, jako HIV/AIDS, malárie apod., (7) Zajistit udržitelnost životního prostředí, (8) Rozvést globální partnerství pro podporu rozvoje. Tento soubor osmi cílů se označuje pojmem The Millennium Development Goals (MDGs). Vedoucí organizací, která pracuje na dosažení těchto cílů, je Rozvojový program OSN (UDNP). V roce 2015 vydala právě tato organizace zprávu o tom, do jaké míry byly tyto cíle naplněny.

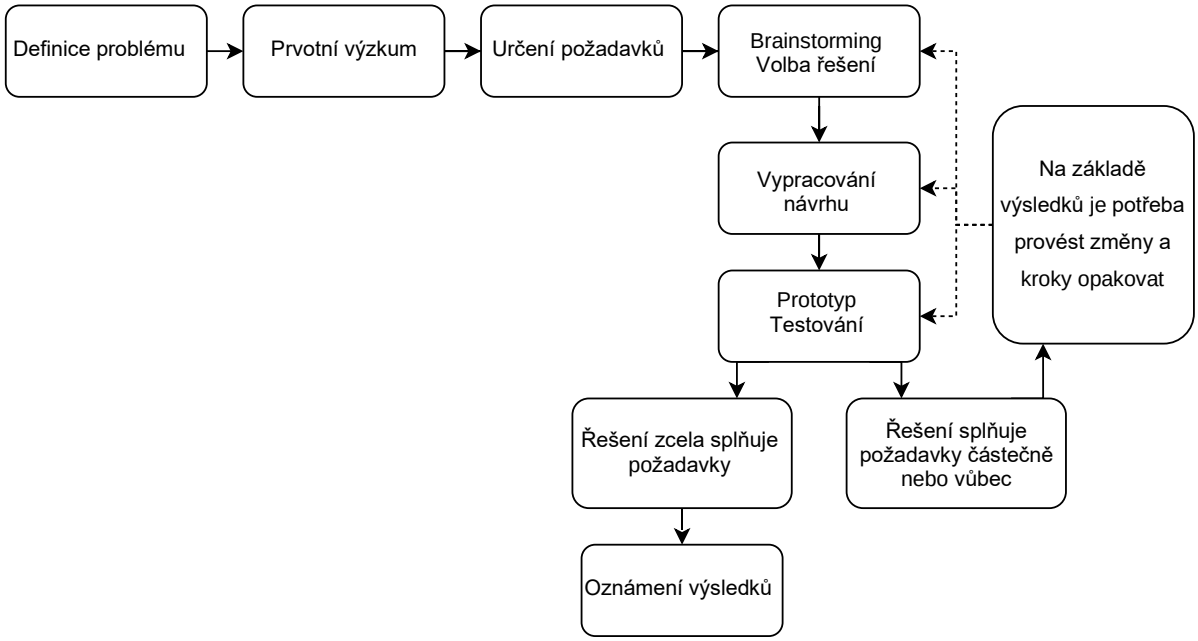
V téže roce se konalo nové setkání (opět 189 vůdců) s označením United Nations Sustainable Devolopment Summit, kde byly odhlasovány nové cíle udržitelného rozvoje (SDGs), které by měli být splněny do roku 2030. Tentokrát se jednalo o 17 cílů, které se problémy zabývaly podrobněji než předchozí MDGs, a kromě samotných problémů se snaží vymýtit i jejich příčiny.

V souvislosti s touto problematikou vydal v roce 2014 Institute of Transformative Technologies (ITT) studii, ve které představil „50 průlomů“. Jedná se o padesát klíčových technologií, které jsou potřebné pro globální rozvoj. Studie se zabývala celkem devíti hlavními oblastmi a jednou z nich byla digitalizace. Konkrétně průlom s číslem #42 se přímo zabývá technologiemi IoT. S řešením těchto globálních problémů může pomoci každý jednoduše tím, že se bude podílet na tvorbě nový nápadů a řešení.

9.1 Proces návrhu řešení

Kapitola Proces návrhu řešení je založena na procesu konstrukčního návrhu, který je představený na webu komunity Science Buddies [59].

Při návrhu nových řešení je nejlepší zvolit cestu ověřených metod a jednou z nich je proces konstrukčního návrhu (Enginnering Design Process), který je velmi oblíbený a používaný v široké škále především technických odvětví. Jedná se o sérii logicky uspořádaných kroků, které vedou k úspěšnému řešení. Celý proces návrh je silně iterativní a to znamená, že některé kroky je potřeba několikrát opakovat, než bude možné přejít k dalšímu kroku. Důležité je podotknout, že se v žádném případě nejedná o standardizovaný postup, ale pouze o obecný a teoretický popis procesu návrhu, a proto je možné v této souvislosti narazit na několik různých variant a termínů. I přesto, že se kroky i jejich počet mohou v ostatních případech lišit, jádro celého procesu zůstává téměř stejné. To, jak takový konkrétní proces návrhu může vypadat a jaké obsahuje kroky, znázorňuje následující obrázek (viz Obrázek 36).



Obrázek 36 Proces návrhu řešení
Zdroj: vlastní zpracování

9.1.1 Definice problému

Jak je vidět, v ideálním případě se jedná o posloupnost nejméně osmi kroků, ale v závislosti na podstatě řešeného problému se určité kroky mohou opakovat. Prvním krokem při návrhu, nejen IoT řešení, ale téměř jakéhokoli řešení, je definice problému, který má být řešen. V této fázi je důležité pokládat otázky typu: Jaký je problém nebo potřeba? Kdo má daný problém nebo potřebu? Proč je důležité daný problém řešit? Jednoduše je to tedy: Kdo potřebuje, co a proč? S nalezením nějakého problému, jehož řešení by pomohlo nejen autorovi, ale i ostatním lidem, mohou pomoci obyčejné věci, jako papír a tužka. Nedoporučuje se věnovat hned prvnímu problému, který přijde v úvahu, ale udělat seznam několika možných kandidátů a mezi nimi vybírat.

S výběrem, a samotnou definicí problému, může pomoci takzvaná myšlenková mapa. Jedná se o techniku, které pomáhá vyjádřit a generovat nové nápady a zájmové oblasti. K tvorbě myšlenkové mapy neexistuje žádný předepsaný postup a pravidla. Postup je jednoduchý, a to zapsat cokoliv, co člověka napadne v souvislosti s daným problémem. Je to tedy vizuální reprezentace myšlenek, proto myšlenková mapa, a její výsledná podoba by se dala nazvat organizovaným chaosem. Je to jednoduchá a rychlá metoda, jak přijít na nové možnosti a souvislosti. Myšlenková mapa není ale limitována pouze na tuto fázi a použít ji je možné prakticky v jakékoli fázi projektu. Jelikož samotná definice problému může někdy trvat v řádech týdnů, technici a návrháři často zaznamenávají veškeré informace a poznatky například do poznámkových bloků. Tím si po celou dobu udrží jakýsi řád, a navíc minimalizují šanci, že by finální definici problému na něco zapomněli. Kromě textu takové poznámky často obsahují i různé náčrty a schémata.

9.1.2 Prvotní výzkum

Dalším krokem je výzkum v oblasti dané problematiky. Zvláště u projektů technické povahy je tento krok velice důležitý, protože je možné se učit ze zkušeností ostatních. Výzkum by měl zpravidla probíhat ve dvou základních směrech, a to uživatelé/zákazníci a existující řešení. I přesto, že řešení je navrhováno tak, aby fungovalo bez potřeby interakce uživatele, je stále důležité mít konkrétní představu o tom, kdo výsledné řešení koupí/bude používat. Cíloví uživatelé se dají dělit podle

různých kritérií (věk, zaměstnání, vzdělání, úroveň zkušenosti apod.). Dále je důležité udělat průzkum již existujících řešení, aby nedošlo k situaci, že na první pohled přelomový nápad mělo již desítky dalších lidí. Není třeba přinést zcela nové řešení, ale doručit nějakou přidanou hodnotu a něco, co ho udělá lepším oproti ostatním. Od existujících řešení je možné se i spoustu věcí přiučit. Například poskytnou podklad tomu, jak má vlastní řešení fungovat a jak je možné ho realizovat. Je naprosto zbytečné se dlouhé hodiny zabývat něčím, co už někdo udělal. A na druhou stranu mohou poskytnout i informace o tom, jakým chybám se při realizaci vyvarovat. Především v odborných oblastech je tato fáze velmi důležitá, a kromě výzkumu z pohledu návrhu, je dobré se seznámit i s oblastí, kde má řešení být nasazeno.

Výzkum je možné provést několika způsoby. První z nich prosté pozorování uživatelů, kteří používají podobné řešení nebo produkt, nebo kteří pracují v prostředí problému. Druhým způsobem je prozkoumat a analyzovat podobné produkty, jelikož mohou poskytnou cenné informace. Posledním typickým způsobem je výzkum pomocí literárních a internetových zdrojů (fóra, studie, výzkumy, zprávy apod.). Ale již ověřený způsob, jak dostat nejcennější rady a informace, je zeptat se lidí, kteří jsou experty v daném oboru, nebo lidí, kteří mají hodnotné zkušenosti s návrhy jiných řešení. Ale nemusejí to být jen profesionálové, kdo dokáže vnést do problému novou perspektivu a někdy pohled amatéra odhalí více než pohled profesionála. Tento proces se nazývá síťování (networking).

9.1.3 Určení požadavků

Třetím krokem je určení požadavků, které musí výsledné řešení splňovat, aby se dalo považovat za úspěšné. Počet požadavků se liší podle povahy a složitosti problému, například hobby nebo školní projekty typicky mají méně jak deset požadavků, ale komplexní a rozsáhlá řešení jich mohou mít tisíce, až dokonce desetitisíce. Povaha požadavků může být různá a mohou se týkat různých vlastností, jako velikost, cena, složitost použití, dopad na prostředí apod. Jeden z nejlepších způsobů, jak identifikovat požadavky pro vlastní projekt, je použít konkrétní existující řešení, které se co nejvíce podobá zamýšlenému výsledku. Důležité je prozkoumat každý detail produktu, jelikož každá vlastnost může reprezentovat

potenciální požadavek. Při analýze produktu je dobré vstřebávat i techniky, mechanismy a různé triky, které byly k výrobě použity. S každým dalším produktem se poté zásoba těchto znalostí rozšiřuje. I když se takový přístup může na první pohled zdát někomu nemorální, je to zcela běžné a praktikuje to spousta předních společností. Kromě analýzy produktu je důležité analyzovat i prostředí, pro které je výsledný produkt vytvářen. To pomůže se specifikací dalších požadavků na samotný produkt. Při specifikaci příliš velkého počtu požadavků může nastat situace, že samotný návrh a tvorba produktu se stane téměř nerealizovatelnou. Někdy nastane i situace, že se budou navzájem vylučovat, to znamená, že nelze dosáhnout všech určených požadavků a často bude potřeba najít nějaký kompromis, nebo požadavky změnit. To se označuje pojmem „trade-off". S rostoucím počtem požadavků poté roste i počet těchto kompromisů, což dělá tuto fázi příliš komplexní. Taková řešení se poté označují jako „over constrained" (příliš omezená), nebo naopak „under constrained" (málo omezená).

9.1.4 Brainstorming / Volba řešení

Při řešení návrhu je vždy několik dobrých řešení. Nedoporučuje se zaměřovat pouze na jednu možnost, jelikož existuje šance, že lepší řešení bude přehlédnuto. Při návrhu je vždy dobré přijít s co nejvíce možnostmi a až poté vybírat to, které se nejvíce hodí pro danou situaci. I na první pohled „šílené" řešení může pomoci identifikovat některé prvky, které mohou zlepšit výsledné řešení. Tomuto procesu se říká „ideation" (volně přeloženo jako „myšlení"). Při generování nápadů existuje několik klíčových pravidel, které se doporučuje dodržovat. Prvním pravidlem je „žádná omezení". V této fázi není žádné řešení nereálné, a proto není potřeba se omezovat na dvě na první pohled dobrá řešení. Čím více nápadů, tím více porovnání při finálním výběru bude k dispozici. Druhým a asi nejdůležitějším pravidlem je „nikdy nezůstat u prvního nápadu". Není horší situace než fixace na první nápad, jelikož to zastaví jakoukoli další kreativitu. I přesto, že nakonec toto řešení bude vybráno jako ideální, je zde velká šance, že bude možné aplikovat něco z dalších nápadů. Metod pro generování nápadů je několik. První metodou jsou, podobně jako u analýzy požadavků, již existující řešení. Jedná se o nejlepší zdroj nápadů při tvorbě alternativ, jelikož se jedná již o fungující a odzkoušená řešení. V této situaci je dobré

si položit například tyto otázky: (1) Je možné hlavní vlastnosti již existujících řešení zkombinovat novým způsobem? (2) Je možné celá řešené zkombinovat do jednoho? (3) Existují nějaké části, či vlastnosti, které danému řešení chybí? Druhou metodou je analogie. Jedná se o porovnání řešeného problému s kompletně jinou situací, při kterém se můžou objevit netradiční a originální nápady. Další metodou je velmi známý a oblíbený brainstorming. Skupinový brainstorming je skvělá cesta ke spoustě nápadů. Do tohoto procesu se může zapojit prakticky kdokoli a často tak vznikají velmi originální nápady. Při brainstormingu je
potřeba dodržovat jedno důležité pravidlo, a to „žádný nápad není špatný". Všechny návrhy je důležité sepsat, aby na konci mohla proběhnout diskuse. Ideation proces není záležitost jednoho dne. Naopak by se mělo jednat o jednu z nejdelších fází návrhu.

Následně je potřeba z těchto nápadů vybrat ten nejlepší. Prvním krokem je porovnat každé řešení s určenými požadavky. Zde dochází k prvnímu zúžení možností tím, že se vyřadí nápady, které požadavky splňují pouze v málo případech. Kromě požadavků, které jsou pro řešení mandatorní, lze uvažovat i o takzvaných „nice to have" vlastnostech. Některá řešení mohou takových vlastností mít více najednou, což je dělá lepšími. V potaz je potřeba vzít i univerzální kritéria, která se týkají téměř každého řešení. Je to například elegance, robustnost, estetické provedení, náklady, potřebné zdroje, potřebný čas, bezpečnost apod. Při porovnání jednotlivých řešení je možné využít jednoduchou rozhodovací matici (viz Tabulka 4). Každý řádek matice označuje jeden požadavek nebo kritérium a jednotlivé sloupce odpovídají navrženým možnostem řešení. V matici se hodnotí to, do jaké míry dané řešení splňuje každý určený požadavek. Typicky se používá tří úrovňové hodnocení vyjádřené váhou, a to: 0 = řešení daný požadavek nesplňuje vůbec, 1 = řešení splňuje požadavek jen z části, 2 = řešení zcela kompletně splňuje daný požadavek. Poté dojde k sečtení hodnot ve sloupcích jednotlivých řešení a největší výsledná hodnota určuje nejlepší řešení.

Požadavky a Kritéria	Řešení #1	Řešení #2	Řešení #3
Požadavek #1			
Požadavek #2			
Další + nice-to-have a univerzální			
Celkové hodnocení			

Tabulka 3 Rozhodovací matice
Zdroj: vlastní zpracování

Alternativou tohoto hodnocení je využití barev, kdy hodnotě 0 odpovídá červená, hodnotě 1 žlutá a hodnotě 2 zelená barva. To poskytne vizuální přehled o tom, jaké řešení splňuje, či nesplňuje dané požadavky. Počet řádků a sloupců matice není nijak omezený, jen je třeba mít na paměti, že všeho moc škodí a velký počet požadavků a možností může mít zcela opačný efekt a rozhodování může být tak těžší. Rozhodovací matice se většinou používá při tvorbě komplexnější a rozsáhlejších řešení. U menších projektů v rozsahu školních a hobby projektů, lze využít jednodušší přístup a to, výhody/pozitiva a nevýhody/negativa. Na první pohled jednoduchý přístup dokáže donutit se nad jednotlivými řešeními důkladněji zamyslet a pomůže vybrat to nejvhodnější.

9.1.5 Vypracování návrhu

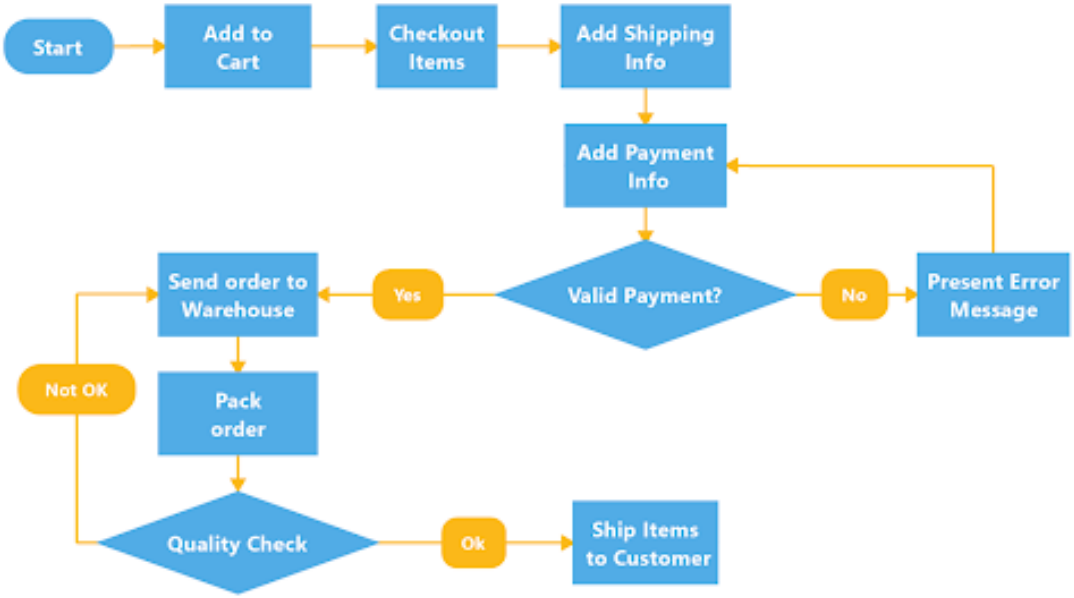
Po volbě vhodného řešení je na čase tento nápad rozvést do daleko detailnějšího návrhu. V některých situacích se s některými prvky návrhu, které by mohly do této části patřit, dá setkat již v dřívějších krocích, za účelem lépe ohodnotit různá potenciální řešení. Týká se to především nákladů, které se porovnávají velmi těžko bez určité představy finálního produktu. Fáze návrhu nemá žádný určený konec, jelikož rozvoj daného řešení probíhá i během fáze návrhu a často i potom, co je produkt nabídnut zákazníkům. Zároveň se v této fázi začínají eliminovat první potenciální rizika a optimalizovat požadavky řešení. Se snižováním rizika je dobré začít co nejdříve, než se rizika nahromadí a jejich odstranění bude náročnější. Často se totiž již během návrhu ukáže, že řešení takto fungovat nebude a bude potřeba

provést potřebné úpravy. Pro tuto situaci se používá rčení „Fail fast, fail early“ (Selži rychle, selži brzo). V případě, že řešení má definovaný velký počet požadavků, téměř jistě nastane situace, kdy se dva nebo více požadavků budou navzájem vylučovat a tuto situaci je nutné řešit. Optimalizace je tedy proces, kdy se hledá jistá rovnováha a nejlepší trade-off mezi různými požadavky.

V závislosti na podstatě řešeného problému se výstupy návrhové části budou lišit. V případě návrhu IoT řešení bude záležet i na rozsahu a složitosti daného řešení. Typicky využívané metody a přístupy jsou například náčrty a skeče, technické výkresy, modely, diagramy, prototypy a analýza. Návrháři využívají grafickou reprezentaci k záznamu důležitých poznatků a nápadů, ke komunikaci mezi sebou a dále slouží jako podklad pro konkrétnější návrhové dokumenty a modely. Vše začíná u jednoduchých náčrtů a skečů, které jsou typicky kreslené rukou a zjednodušené vystihují podobu produktu nebo jeho částí. Dále jsou takzvané technické výkresy, které jsou daleko podrobnější a přesně vystihují strukturu produktu. Často jsou tvořený ve speciálních softwarech (Computer-aided design – CAD) a lze je pak použít při sestavování výsledného produktu, jelikož detailně popisují vztahy mezi jednotlivými částmi. Dalším výstupem mohou být modely. Ty lze chápat buď jako fyzické objekty, označované makety, nebo počítačové modely používané při předpovědi a simulaci fungování produktu. U školního projektu se zcela jistě nepůjde do takových detailů, ale jistá struktura a návaznost zde bude zachována. Konkrétními výstupy by například mohlo být elektrické schéma zapojení, vývojový diagram, sekvenční diagram a z pohledu podnikaní třeba tzv. Business canvas.

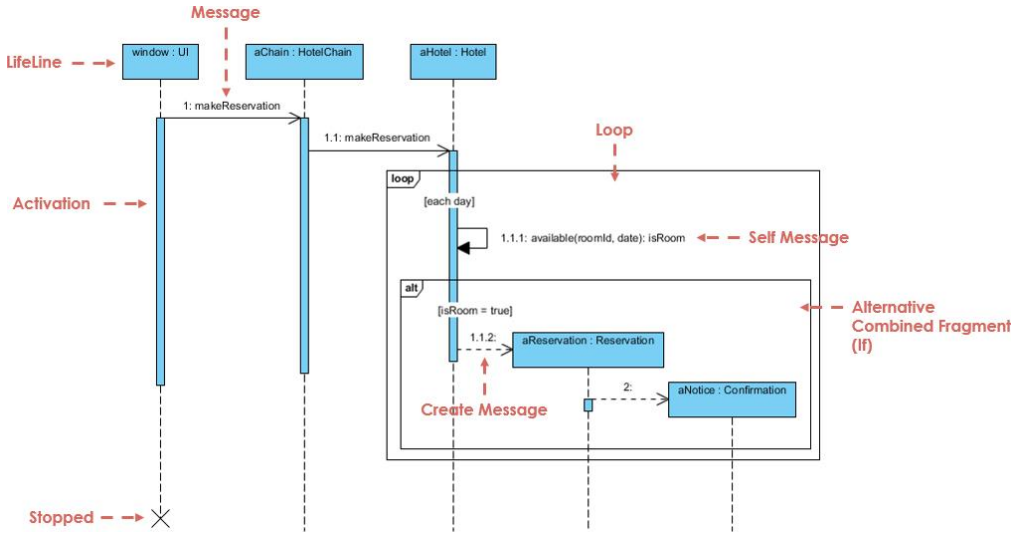
Schéma elektrického zapojení se používá ke grafickému znázornění elektrického obvodu. Znázorňuje konkrétní propojení využitých elektrických součástek a používají se při návrhu elektrických obvodů, konstrukci (např. plošných spojů) a údržbě zařízení a vybavení. Kromě elektrických součástek schémata často obsahují i logické součástky. Ke grafické reprezentaci se používá standardizované značení, a to konkrétně dva základní druhy: standard IEC a standard US. Pro tvorbu schémat v současnosti existuje nespočet nástrojů a simulátorů, které tento proces mohou udělat výrazně jednodušší a efektivnější.

Vývojový diagram (flowchart) je grafickou reprezentací algoritmu nebo průběhu nějakého procesu. Jedná se tedy o diagram týkající se funkční části řešení a nijak neovlivňuje ani neurčuje fyzickou strukturu řešení. Každý krok je v diagramu znázorněn obrazcem (čtverce, kosočtverce, obdélníky apod.) a ty jsou spojené tokem řízení (šipkami). Každý tvar má v diagramu daný význam, například obdélník definuje dílčí krok, kosočtverec větvení postupu podle určené podmínky atd. V informatice se vývojové diagramy využívají pro analýzu, návrh a dokumentaci, ale lze je využít pro reprezentaci procesů a pracovních postupů v jakémkoli odvětví.



Obrázek 37 Příklad vývojového diagramu
Zdroj: https://blog.mindmanager.com/blog/2020/01/use-flowcharts-document-work-processes/

Sekvenční diagram je jeden z mnoha diagramů jazyka UML (grafický jazyk pro vizualizaci, navrhování a dokumentaci systémů). Konkrétně se jedná o diagram interakcí, který znázorňuje časovou posloupnost a návaznost zasílání zpráv nejen mezi objekty, ale i aktéry nebo prvky uživatelského rozhraní, ke kterému dochází při vykonávaní funkce produktem. Sekvenční diagram se dělí na dvě dimenze, a to objektovou a časovou. Objektová dimenze se nachází na horizontální ose a obsahuje jednotlivé elementy zapojené v interakci spolu s jejich „životem“ (lifeline). Doporučuje se je řadit zleva doprava podle toho, kdy se do interakce zapojují. Časová dimenze reprezentuje vertikální osu a jak název napovídá, znázorňuje tok času postupující směrem dolů.

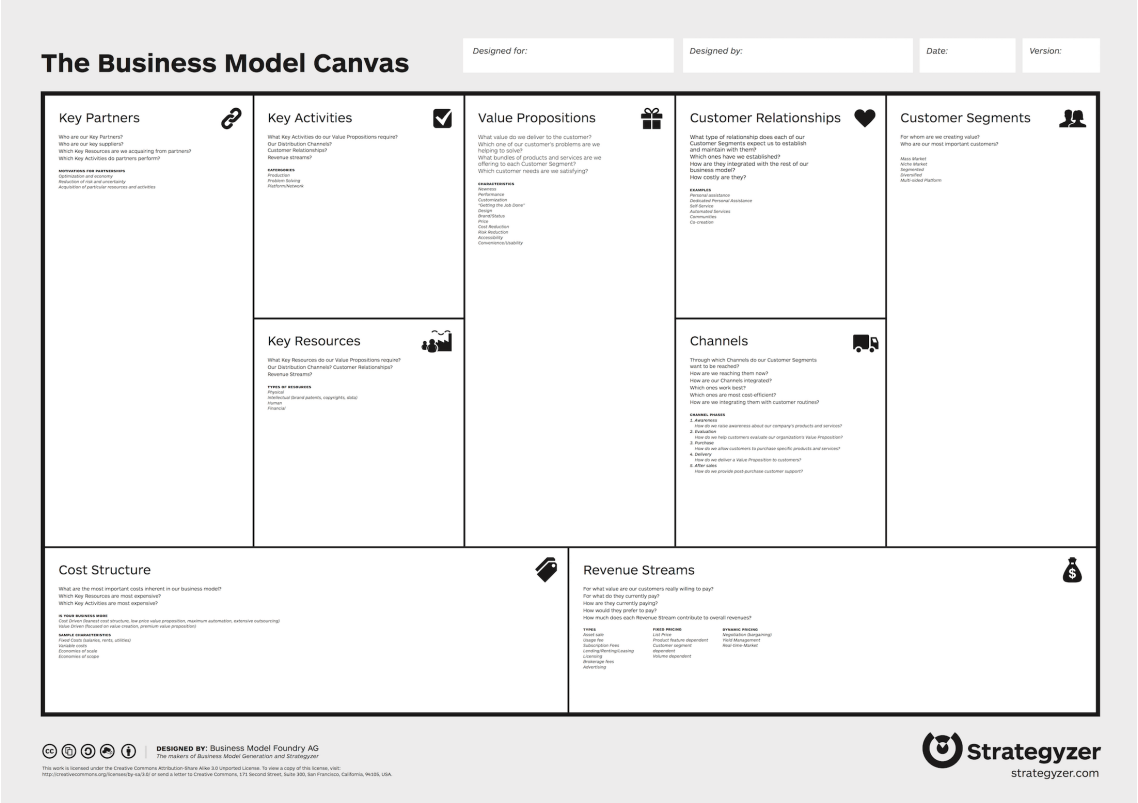


Obrázek 38 Příklad sekvenčního diagramu

Zdroj: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/

Business canvas model dokáže zobrazit základní stavební bloky, které je potřeba rozvíjet, spravovat, udržovat a prodávat na jednom jediném listu papíru. Model pomáhá se zamyšlením se nad strategicky nejdůležitějšími elementy produktu, na které je potřeba se zaměřit. Je možné ho využít pro konkrétní produkt nebo na celý podnik. Canvas se běžně rozděluje na devět kategorií, které je možné upravovat v závislosti na posuzované oblasti a produktu, ale jejich podstata bude stejná. Kategorie modelu jsou [60]:

- Hodnota nabízená produktem (Co to dělá a slibuje?)
- Zákaznický segment (Pro koho to je?)
- Klíčové aktivity (Kroky, které musí být provedeny k dosažení úspěchu)
- Klíčové zdroje (Personál, nástroje, rozpočet)
- Komunikační kanály (Marketing a prodej)
- Vztahy se zákazníky (Jak bude tým podporovat a pracovat s klienty?)
- Klíčový partneři (Zapojení třetích stran, kdo?, jak?)
- Struktura nákladů (Náklady na výrobu, prodej a podporu)
- Toky příjmů (Jak bude produkt vydělávat?)



Obrázek 39 Business canvas model
Zdroj: [60]

9.1.6 Prototyp a Testování

V případě již přijatelných podkladů a dokumentace je možné sestrojit prototyp produktu. Prototyp označuje takovou první testovací fungující verzi řešení a často se k jeho sestavení používají levnější materiály a komponenty. Někdy se stane, že takových prototypů může být sestaveno hned několik, jelikož se začnou objevovat problémy například v konstrukci předchozích prototypů. K sestavování prototypů elektrických zařízení existuje široká nabídka stavebnic a různých „kitů“. Nejčastěji se u takových prototypů využívá deska označována pojmem „breadboard“ (nebo „protoboard“). Je přímo určená pro prototypování a její hlavní výhodou je jednoduchá customizace zapojení, jelikož při zapojování není třeba nic pájet. Tato deska se často využívá i u různých softwarových a online simulátorů (TinkerCAD apod.). Kromě breadboardu tyto stavebnice obsahují i spoustu dalších komponent z oblasti IoT, jako senzory, kamery atd. Někteří návrháři kromě těchto speciálních kitů využívají i běžně dostupnou LEGO stavebnici. Už ale i značka LEGO nabízí stavebnice s označením LEGO Mindstorm a LEGO Technic, které obsahují a

podporují vše od senzorů, motorů, až po mini počítače. Od široké nabídky možností těchto stavebnic, se ale odvíjí i jejich cena, která se u některých součástek pohybuje v řádech tisíců.

Fáze testování by se dala rozdělit na dvě části, a to samotné testování a následný redesign na základě získaných poznatků. Testování je možné provádět několika způsoby, ale v podstatě se jedná o aplikaci různých scénářů uživatelské interakce s řešením. Je doporučeno, aby se testování účastnilo více uživatelů, jelikož každý z nich může na řešení mít jiný pohled a odhalit různé problémy. Neexistuje nic jako příliš mnoho testerů. Nahlášené problémy je ovšem třeba vyhodnotit a rozlišit, zda se nejedná o silně subjektivní problém (např. barva) a zda se vyplatí do něj investovat další čas. Dále také platí, že řešení by se mělo testovat v prostředí, pro které je zamýšleno, pokud je to možné. Čím více problému a nedostatků se v této fázi objeví, tím úspěšnější testování bylo. Testování by mělo přinést odpověď na následující otázky:

- Jsou uživatelé schopni překonat řešený problém využitím nebo interakcí s řešením?
- Potřebují se uživatelé ptát na některé věci při používání nebo interakcí s řešením?
- Používají nebo interagují uživatelé s řešením tak, jak bylo zamýšleno?
- Pokud existuje pro řešení měřitelné cíle, byly naplněny?

Odpovědi na tyto otázky a následné další otázky, jsou cenou zpětnou vazbou a základem pro další rozvoj řešení a fázi redesignu.

Redesign řešení se provádí na základě zjištěných poznatků z testování a Patří do toho oprava chyb, změny a takzvané QoL (Quality of Life) úpravy. Kromě řešení problémů je ale potřeba se zaměřit na další rozvíjení a „uhlazení“ vlastností a funkcí, které byly mezi uživateli úspěšné a byly hodnoceny kladně. Zde se k předchozím čtyřem otázkám pokládají další otázky, ale detailnější a cílené na konkrétní věc. Například pokud by odpověď na druhou otázku byla „Ano“, pak se nabízejí otázky jako:

- Z jakého důvodu se uživatel ptal?
- Byl uživatel při používání řešení zmatený?
- Jaká část řešení nebyla pochopitelná a proč?
- Jaké změny je potřeba udělat, aby další uživatelé nemuseli pokládat stejné otázky?

Takové a mnoho dalších otázek pomáhají řešení vylepšovat a připravovat na finální implementaci. Samozřejmě tím, že dojde k potřebným změnám a úpravám, tato fáze nekončí. Po redesignu je potřeba znovu provést testování. Testování a redesign se může v této smyčce opakovat tak dlouho, dokud bude potřeba. S každým opakováním dochází k vylepšování řešení a ke zvýšení šancí na úspěch. Samozřejmě může nastat situace a bohužel se tak stává, že během testování dojde k odhalení velmi závažné chyby nebo k jiným problémům, které nedovolí implementaci řešení. V takovém případě je potřeba se vrátit o několik kroků dozadu, například ke zpracování návrhu a krok zopakovat, nebo dokonce ještě dál k definici požadavků. Někdy bude stačit pouhá revize dostupných materiálů a dokumentace a někdy bude třeba drastičtějších zásahů.

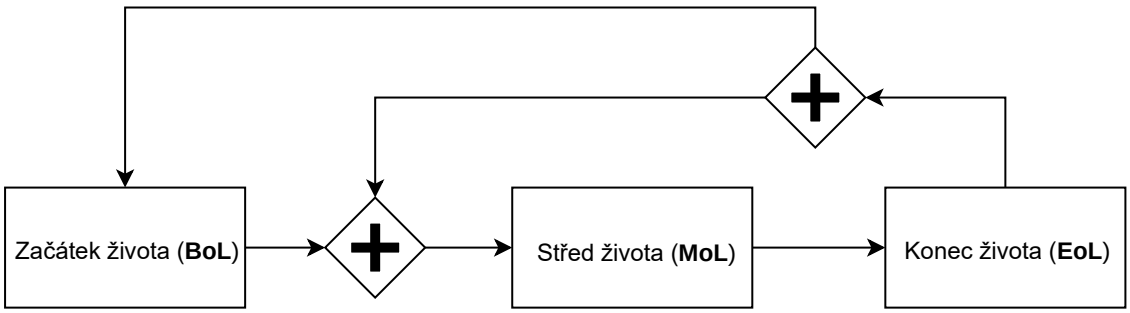
To odkazuje na jednu důležitou věc, kterou je potřeba připomenout. Jakýkoliv výstup testování (problémy, chyby, nedostatky apod.) a jakékoli změny je potřeba náležitě dokumentovat! Dobrá dokumentace je součást úspěchu a neplatí to jen v této fázi, ale ve všech krocích návrhového procesu a i po něm. Bez dobře vypracované a kompletní dokumentace nebude její revize možná a tím pádem nezbude nic jiného než některé kroky celé opakovat. Pozitivní věcí je ale to, že každá chyba je cenná zkušenost a drasticky snižuje šanci, že se stejná chyba bude opakovat v dalších projektech.

9.2 Management životního cyklu produktu

Kapitola vychází z publikace IoT Device Lifecycle – A Generic model autorů Soos, Janky, Kozma a Varga [61].

Podobně, jako u ostatních produktů a služeb, se i proces tvorby, nasazení, podpory a vyřazení IoT řešení dá popsat formou takzvaného životního cyklu. Jeho účelem je pomoci s definicí a následnou správou, včetně kontroly stavu,

zabezpečení, upgradů apod., připojených zařízení ve všech průmyslových i neprůmyslových doménách. Někdy se tato činnost nebo proces označuje pojmem „Management životního cyklu produktu“ (Product Lifecycle Management – PLM). Životní cyklus se objevuje i ve spojení s jinými pojmy, a to konkrétně například životní cyklus vývoje, projektu, dat apod. I v samotné oblasti IoT systémů a produktů je podob tohoto cyklu hned několik, ale jejich základ, ve většině případů, lze zobecnit do podoby, která rozlišuje tři základní fáze, a to: Začátek života (Beginning of Life – BoL), Střed života (Middle of Life – MoL) a Konec života (End of Life – EoL). Samotný životní cyklus není definovaný tak, že produkt musí projít všechny tři fáze a následně dojde k jeho vyřazením, ale, jak je znázorněno na obrázku (viz Obrázek 40, Pozn.: Značka + se v BPMN označuje jako Parallel gateway a znázorňuje, že dochází k paralelnímu rozdělení procesu na více větví), je reprezentován uzavřenou smyčkou, kde dochází k využití zpětné vazby k ovlivnění ostatních fází. Netýká se to jen jednoho konkrétního produktu, ale je třeba myslet na to, že informace získané v jednom cyklu života jednoho produktu, můžou, kromě předchozích a následujících fází, ovlivnit i nadřazené celky (řídící systémy apod.) a tím i ostatní produkty. Každá fáze tohoto životního cyklus by se dále dala rozložit na podprocesy, které více vystihují jejich podstatu a definují konkrétní operace.



Obrázek 40 Jednoduchý životní cyklus produktu
Zdroj: vlastní zpracování (podle [61])

První fáze, **začátek života** (BoL), zahrnuje proces návrhu a tvorby řešení. Ten je podrobně popsán v předchozí kapitole (viz 9.1 Proces návrh řešení) a nebude tedy znovu popisován.

Druhá fáze, **střed života** (MoL), je tou nejdůležitější fází celého cyklu. Zde se totiž odehrává vše týkající se nasazení a podpory řešení. Konkrétně se jedná o

proces nasazení (Provisioning), kde dochází k registraci produktu, připojení produktu a celkové přípravě na jeho spuštění. Následují procesy konfigurace (Configuration), aktualizace (Update) a údržby (Maintenance). Během procesu konfigurace zařízení nezískává žádné nové schopnosti, ale dochází ke změně jeho chování a jeho nastavení vzhledem k prostředí. Aktualizací se rozumí vylepšení nebo změna schopností produktu, které využívají k vykonávání úloh. U IoT zařízení se většinou provádí aktualizací firmwaru. I údržba je velmi důležitým procesem, jelikož bez ní by mohlo dojít k výraznému zkrácení produktů nebo by mohlo dojít k selháním a nežádoucím chybám v celé síti. Monitorování je posledním procesem této fáze a funguje jako křižovat, kde dochází k rozhodnutí, zda opustit danou fázi MoL a přejít do poslední konce života, anebo v ní setrvat. Monitorování slouží ke kontrole stavu a zdraví zařízení, což může pomoci s předvídáním možných selhání a včasným zásahem při eliminaci této možnosti. Vzdálené monitorování všech součástí systémů a produktů je v současnosti nepostradatelnou součástí v managementu a správě podobných řešení.

Poslední třetí fází je **konec života** (EoL). Pokud na zmíněné křižovatce dojde k rozhodnutí, že daný produkt či řešení v budoucnu nedokáže přinést požadované hodnoty, nebo by jeho oprava byla zbytečně nákladná, dojde k přechodu do této fáze, a to konkrétně do procesu odebrání zdrojů (Deprovisioning). Podstatou tohoto procesu je odebrání veškerých zdrojů, které byly v části nasazení přiděleny. Často je tento krok opomíjen a poté dochází k hromadění takzvaných „zombie“ zařízení. Toto označení se používá pro zařízení v síti, které již nejsou používány, nebo jsou dokonce zcela opuštěny, ale stále jsou pro ně rezervované zdroje. Dalo by se dokonce říct, že samotné odstranění fyzického zařízení z jeho umístění není tak důležité, jako odebrání jemu přiřazených zdrojů. A posledním procesem, nejen této fáze, ale celého cyklu, je vyřazení (Retire). Důvody k vyřazení zařízení mohou být různé, ale často to bývá za účelem jeho nahrazení nebo jednoduše z důvodu staří.

10 IoT pod útokem

Informace v kapitole Tvorba IoT řešení jsou převzaty z online kurzu Cisco [62], z části Chapter 1:The IoT Under Attack, pokud není uvedeno jinak.

V první polovině bylo IoT představeno jako jedna z hlavních příčin čtvrté průmyslové revoluce (označované Industry 4.0), a jako technologie, která dokáže jakékoli prostředí proměnit v „chytré“ prostředí, za pomoci senzorů, aktuátorů a podobných zařízení, které jsou následně připojeny k internetu. Ovšem, jak už to ve světe technologií bývá, žádná nová technologie s sebou nepřináší pouze výhody, ale i nespočet rizik a problémů. Za nejzávažnější potenciální riziko lze považovat to, že kromě nových možností pro uživatele nabízí IoT i nespočet nových možností pro útočníky a hackery.

10.1 Anatomie útoku

Hacking by se dal označit jako nepovolený a nežádoucí přístup do sítě nebo zařízení. Často se jejich cílem stávají uživatelská data nebo manipulace, či poškození daného systému. Hackeři se dají rozdělit do několika skupin, které jsou reprezentované „barvou čepice“, a to na: black hat, white hat, grey hat, blue hat, red hat, green hat. Poslední tři pojmy nejsou tak známé jako první tři skupiny, ale v posledních letech se tyto termíny objevují čím dál častěji. Ne ale všichni hackeři mají zlé úmysly a spousta organizací si hackery najímá na testování bezpečnosti firemních systémů. Takový hackeři se často pro odlišení označují pojmem etičtí hackeři (od toho Etický hacking) a z předchozího dělení je definuje skupina white hat.

Black hat

Pod pojmem Black hat je možné si představit typického hackera tak, jak ho popisují média. Jejich metody jsou velmi rozsáhle, ale nejčastějším cílem je finanční zisk [63]. Někteří začali jako nezkušení hackeři, kteří využívali koupené hackerské nástroje a někteří jsou vycvičení profesionálními hackery. Často jsou součástí větší skupiny nebo kriminální organizace. Hackeři (nebo celé organizace) si ve většině případů v průběhu času vybudují vlastní specializaci a zabývají se tak jednou konkrétní činností. Běžným příkladem specializované organizace jsou podvodná call centra. Oběť po telefonu přesvědčí, aby jim umožnila přístup do počítače z důvodu údajné

„opravy“ systému a často dojde k odcizení osobních dat a k podstrčení škodlivých softwarů. Největší hrozbou ze strany této hackerské skupiny je to, že black hat hacking je globální problém, a proto je velmi těžké ho zastavit. Dost často se stává, že mezi s sebou hackeři komunikují (např. na Dark web) a v případě odcizení osobních údajů uživatele dojde s největší pravděpodobností k jejich sdílení s ostatními hackery. Z toho důvodu jsou na seznamu podnikatelských rizik často umístěni na nejvyšších příčkách[64].

White hat

Etičtí nebo white hat hackeři jsou přesným opakem black hat hackerů. Etický hacker se považuje za experta, který ovládá a používá stejné postupy a metody jako black hat, ale za účelem odhalení slabin a zranitelností v systémech a následně pomáhají s jejich odstraněním. Jedná se tedy o hackera, který se do systému snaží nabourat, ale se svolením jeho majitele [63]. Etický hacking se v posledních letech stal velmi poptávanou a oblíbenou profesí. Certifikace je nepostradatelná pro vykonávání tohoto povolání a jedná se o takové potvrzení schopností hackera a ujištění organizace, že nehrozí žádné zneužití dat a systému. Velmi uznávanou a žádanou certifikací je CEH (Certified Ethical Hacker) od mexické společnosti The Electronic Commerce Council. CEH pokrývá rozsáhlou oblast v konceptech zabezpečení, nástrojů a útoků. Certifikát je akreditován americkým ministerstvem obrany, které ho dokonce na některých pozicích vyžaduje a Národní bezpečností agenturou (NSA). Kromě CEH existují ještě certifikace a kurzy, jako SANS GPEN, CREST, Foundstone Ultimate Hacking a další [65].

Grey hat

Spojením white hat a black hat vznikne grey hat. Hlavním cílem takových hackerů je především osobní potěšení. Grey hat má veškeré schopnosti black i white hat, ale rozdílem je to, že nemají zájem o to někoho poškodit, anebo někomu pomoci. Jednoduše je baví hledat mezery v zabezpečení a nabourávat se do systému a velmi zřídka se stane, že z jejich strany dojde k nějakému poškození, či odcizení. Zajímavostí je to, že majoritní část hackerské komunity tvoří právě tato skupina i přesto, že se o nich moc nehovoří[63].

Blue hat

Následující typy jsou poměrně nová označené pro některé skupiny hackerů, které spojuje podobný motiv útoku. První z nich je blue hat hacker, který se vyznačuje především nevšední agresí a nemilosrdností a nebojí se zajít do extrému. Jeho hlavním motivem je pomsta. Často využívají již existující kódy, malware a viry a pouze je upraví pro vlastní potřebu. Zaměřují se především na podniky a organizace, které je nějakým způsobem naštvali nebo z jejich pohledu poškodili. Může se jednat o zákazníky, dodavatele, zaměstnance apod [63].

Red hat

Termínem red hat se označují hackeři, kteří se vydávají za takové bojovníky spravedlnosti v hackerském prostředí. Jejich cílem je zastavit black hat hackery a je třeba říct, že je vůbec nešetří. Neváhají použít jakoukoli metodu a v podstatě tak používají jejich metody proti nim (DoS, viry, trojské koně atd.) a ničí je zevnitř. Pro běžného uživatele tito hackeři nepředstavují žádnou hrozbu a dali by se tak zařadit na stranu dobra společně s white hat. Jediným rozdílem je to, že o red hat hackerech cíl nejspíš nebude vědět [63].

Green hat

Pod poslední barvou se skrývají takový hackeři začátečníci, kteří jsou do světa scriptů, kódů a obecně hackování noví a nerealizují žádné útoky ani podezřelou aktivitu. Místo toho se pohybují v online prostředí a nabírají znalosti od zkušenějších hackerů. V tomto stádiu nepředstavuje hacker žádnou hrozbu, jelikož se zajímají více o to, jak takové útoky provádět než o jejich provedení. V dlouhodobém hledisku ale představují stejnou hrozbu jako například black hat [63].

Script kiddie

Script kiddie je označení pro útočníky, kteří se ne zcela dají označit za hackery, a tak trošku vybočují z řady. Jejich cílem je způsobit chaos a narušit fungování služeb. Nemají zájem o krádež ani přímé poškození. Nevyvíjejí vlastní malware, ale stahují již hotový škodlivý software a učí se ho používat podle videí. Jejich typickým útokem jsou DoS a DDoS útoky, které zahltí požadavky danou IP adresu a znepřístupní tak

službu ostatním uživatelům. I přesto, že nepředstavují přímé finanční ztráty a problémy s únikem dat, je velmi nepříjemné být příjemcem těchto typů útoků. Nepřístupná služba (např. e-shop) může způsobit ztrátu zákazníků a poškození reputace [63].

10.1.1 Databáze CVE

Zkratka CVE (Common Vulnerabilities and Exposures) je veřejná databáze známých bezpečnostních chyb počítačů a operačních systémů. Databázi často využívají experti a poradci v oblasti zabezpečení. CVE spravuje americká korporace MITRE s finanční podporou přímo od ministerstva. Záznamy v databázi jsou velmi stručné, neobsahují žádná data nebo informace o rizikách, dopadech a opravách. Každý záznam je v databázi označen unikátním CVE identifikátorem, pomocí kterého je poté možné přesně odkazovat na konkrétní chybu. Tyto ID jsou přiřazovány CVE autoritami (CNA – CVE Numbering Authority), které jsou reprezentovány předními IT obchodníky a organizacemi zabývající se bezpečností a výzkumem. CNA představuje rozsáhlý blok rezervovaných CVE identifikátorů pro danou společnost. Každý rok se zveřejňuje tisíce CVE záznamů. Samotné oznámení chyby ale může přijít odkudkoli, čímž upozorní některou z CNA autorit, které k dané chybě po ověření vydá záznam. Někteří dokonce za odhalení chyb nabízejí i tzv. bug bounties (odměny). Samozřejmě je nutné zdůraznit, že chyby zveřejňují až potom, co byla úspěšně nalezena oprava.

K hodnocení závažnosti objeveného bugu se používá například uznávaný standard CVSS (Common Vulnerability Scoring System). Rozsah hodnocení je od 0.0 do 10.0 (čím více, tím závažnější). Následující tabulka (viz Tabulka 4) znázorňuje, jak vypadá distribuce CVE podle stupně závažnosti [66]. Pro porovnání 20. března 2021 bylo v databázi celkem 123 454 záznamů s průměrným skóre 6,6.

CVSS	Počet zranitelností	%
0-1	838	0,5
1-2	1104	0,7
2-3	7140	4,5
3-4	7393	4,7
4-5	36959	23,3
5-6	30454	19,2
6-7	22825	14,4
7-8	32463	20,5
8-9	758	0,5
9-10	18572	11,7
Celkem: **158 506**, průměr CVSS: **6,5**		

Tabulka 4 Distribuce zranitelností podle CVSS (9.8.2021)
Zdroj: https://www.cvedetails.com/

10.1.2 Malware Mirai Botnet

Jedním z největších a nejzávažnějších útoků vůbec, který cílil na IoT zařízení, měl na svědomí malware s označením Mirai Botnet a poprvé byl objeven komunitou etických hackerů MalwareMustDie. Konkrétně se jednalo o zařízení s procesorem ARC, které malware přeměnil na vzdáleně řízené boty nebo „zombie“. Síť těchto botů se poté označuje jako botnet a nejčastěji se používá k provádění DDoS útoků. V roce 2016 byl pomocí Mirai proveden dosud největší DDoS útok, jehož cílem byl web velmi uznávaného experta v zabezpečení. Botnet síť obsahovala dohromady okolo 152 000 zařízení, z nichž většina byly bezpečností kamery. Takové množství botů dokázalo vytvořit konstantní provoz směrem na web o objemu 1Tb/s. Kromě toho existuje podezření, že i za dalším útokem v téže roce stojí opět Mirai. Po prvním útoku byl totiž zdrojový kód vypuštěn na internet, s cílem zakrýt původ softwaru jeho autorem. Tentokrát byl cílen na služby poskytovatele DNS služeb Dyn, což způsobilo výpadek připojení k internetu několika milionům lidí v USA a Evropě.

Překvapující je možná to, jak jednoduché bylo takové množství zařízení proměnit v botnet. Mirai neustále skenuje internet pro IoT zařízení využívající

procesor ARC, který běží na velmi ořezané verzi Linuxu. Následně testuje, zda se na zařízení dokáže přihlásit pomocí výchozí kombinace uživatelského jména a hesla, typicky: root/default, root/1111, admin/admin, tech/tech apod. Pokud ano, zařízení infikuje a připojí do botnetu. Druhou možností, jak malware dokáže zařízení proměnit v zombie, je phishing.

Autoři tohoto botnetu jsou Paras Jha a student univerzity Josiah White. Zajímavostí je to, že první zmíněný vlastnil firmu Protraf Solutions nabízející služby k zmírnění DDoS útoků. Jedná se tedy o klasický případ toho, že nabízel své služby firmám, které sám poškodil. Další útočníci, kteří byli v souvislosti s Mirai zadrženi, jsou tři Američané obviněni z plánování útoku na weby a hosting společností v USA a v zahraničí. Odsouzeni byly na deset let a byla jim udělena pokuta 250 000$.

Hrozba Mirai malware stále přetrvává. Jak bylo zmíněno, kód byl umístěn na internet, a tak se dostal do rukou dalších útočníků, kteří ho různě upravují. V současnosti již je známých několik mutací, jako Okiru, Satori, Masuta, PureMasuta apod. Další hrozbou jsou i nedávno objevené botnety IoTrooper a Reaper, kteří dokážou vyhledat, infikovat a ovládat zařízení daleko rychleji a v daleko větším měřítku [67].

Vyhledávač Shodan

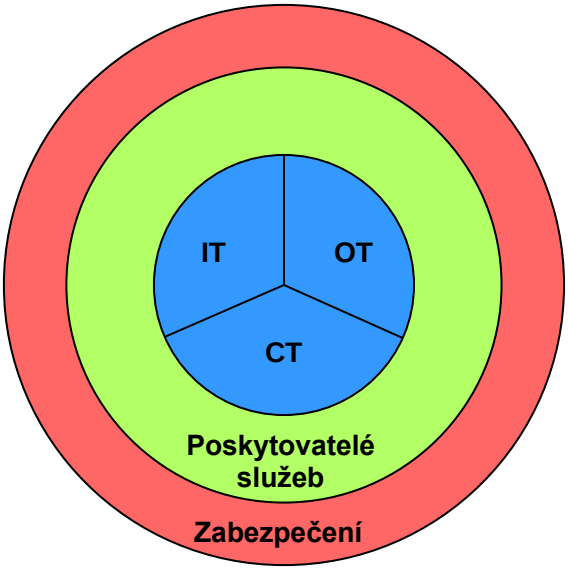
Vyhledávač Shodan by se dal označit za jeden z nejnebezpečnějších vyhledávačů z hlediska využívání. Spuštěn byl v roce 2009 a autorem je programátor John Matherly. Oproti ostatním totiž neprohledává pouze web, ale celý internet jako síť připojených zařízení. Každé zařízení je následně uloženo a lze ho kdykoli vyhledat. Konkrétně prohledává otevřené TCP porty, na které zašle dotaz a pokud objeví známou službu, k záznamu přidá zranitelnosti, screenshoty apod. Ze zařízení tu lze najít routery, kamery, IoT zařízení, rozhraní zařízení atd. Při vyhledávání je možné aplikovat filtry, např: město, zemi, souřadnice, hostname, operační systém apod. Jak bylo zmíněno Shodan neukládá pouhé záznamy. Výsledky je možné díky tagům zobrazit v interaktivní mapě nebo je možné prohledávat obrázky, které pořizuje během skenování, takže je možné si následně zobrazit otevřené plochy nebo kamery. Využívají ho především bezpečnostní experti, etičtí hackeři a výzkumníci. Samozřejmě se naskýtá i otázka, zda nemůže posloužit i útočníkům pro vyhledání

potenciálních cílů. Ano může, ale útočníci zcela jistě budou preferovat botnet, který odvede stejnou práci lépe a bez možnosti detekce [68].

10.2 Model zabezpečení IoT

V předchozích kapitolách bylo představeno rozdělení technologií na tři základní skupiny, a to Informační technologie (IT), Operační technologie (OT) a Spotřebitelské technologie (CT). V krátkosti tedy připomenutí, že IT označuje zařízení v datových centrech, v cloudu, senzory, aktuátory a podobně. Pod zkratkou OT se většinou rozumí ICS a SCADA systémy a všechna zařízení k těmto systémům připojena. Právě v technologiích IT a OT dochází v posledních letech ke snaze o jejich konvergenci. Právě to s sebou přináší nespočet výzev v otázce zabezpečení takto komplexních sítí. Z pohledu IT není to žádný problém nepředstavuje, jelikož v této oblasti se již dlouho dobu klade důraz především na jejich bezpečnost. Hlavní problém je na straně OT, které v minulosti operovali v izolovaných sítích bez připojení k internetu, nebo dokonce k IT sítím. Obecně nebyly OT systémy navrhovány tak, aby zabránily přístupu zvenčí. A poslední skupina CT, jak už název napovídá, reprezentuje skupinu spotřebních připojených zařízení v domácnostech, nositelná zařízení, chytrá auta apod.

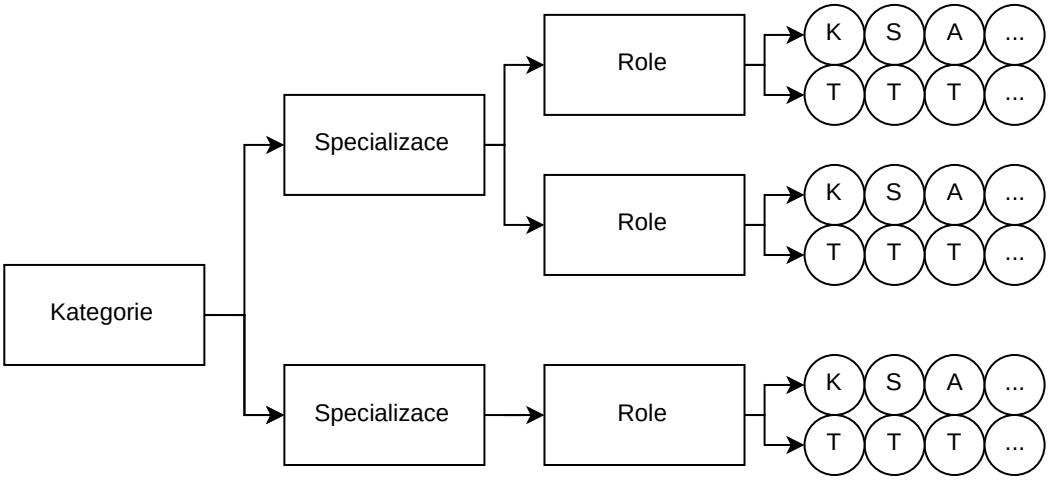
Ať už se v případě IoT jedná o zařízení IT, OT, CT nebo o jejich kombinaci, je vyžadováno silné zabezpečení. A právě poskytovatelé těchto služeb, kteří připojují zařízení k internetu, zaujímají v této problematice speciální postavení, jelikož odpovídají za to, aby služby, které klientům nabízejí, splňovaly jejich požadavky na zabezpečení. Na obrázku (viz Obrázek 41) je vidět konceptuální model zabezpečení IoT, kde je role poskytovatelů znázorněna.



Obrázek 41 Konceptuální model zabezpečení IoT
Zdroj: vlastní zpracování (podle [62])

10.2.1 NICE Framework

V roce 2017 publikoval Národní institut pro standardy a technologie (NIST) Národní iniciativu pro vzdělávání v kybernetické bezpečnosti (N.I.C.E.), jejíž součástí je i Rámec pracovních sil pro kybernetickou bezpečnost (CWF), který má pomoci s identifikací, náborem, rozvojem a udržením talentů v oboru kybernetické bezpečnosti. NICE v sobě sjednocuje kybernetickou bezpečnost a související práci. Publikace jasně definuje sedm základních kategorií, které jsou následně rozděleny na NICE specializace a ty obsahují konkrétní pracovní role. Pro účely kurzu jsou důležité především kategorie „Bezpečné poskytování“ (Securely Provision – SP) a „Chránit a bránit“ (Protect and Defend – PR), které se zabývají právě zabezpečením. Každá pracovní role má následně jasně definované potřebné znalosti, schopnosti a dovednosti (tzv. KSA) a úlohy (T), které ji přísluší. Následující obrázek (viz Obrázek 42) ukazuje vztah mezi těmito pojmy [62, 69].



Obrázek 42 Struktura CWF rámce
Zdroj: vlastní zpracování (podle [69])

Securely provision (SP)

Pracovní role (a specializace) v této kategorii odpovídají za konceptualizace, návrh, pořízení a implementaci bezpečných IT systémů. Konkrétně obsahuje sedm specializací a jedenáct pracovních rolí. Jednou ze specializací SP je Řízení rizik (RSK), která má definované dvě pracovní role: Schvalovací úředník a Posuzovatel kontroly bezpečnosti. Obecně je jejich posláním zajistit, aby existující i nové IT systémy splňovali požadavky organizace v oblasti zabezpečení a rizik. Právě role posuzovatele kontroly bezpečnosti je v IoT velmi důležitá, jelikož se jedná o zpravidla novou oblast v IT. Mezi úlohy pracovníků v této roli patří komplexní posouzení řídících, provozních a technických bezpečnostních kontrol s cílem určit jejich celkovou účinnost [62, 69].

Protect and Defend (PR)

Náplní pracovních rolí v kategorii PR je především identifikace, analýza a zmírnění hrozeb na IT systémy. Celkem obsahuje čtyři specializace a každá obsahuje jednu pracovní roli. Pro tento kurz je důležitá především specializace Posouzení a řízení zranitelností, která zahrnuje provádění hodnocení hrozeb a zranitelných míst, stanovení odchylek od přijatelných konfigurací a zásad, hodnocení úrovně rizika a vývoj nebo doporučení vhodných opatření. Konkrétně se jedná o roli Analytik hodnocení zranitelností. Lidé v této roli provádějí posudky IT systémů a identifikují,

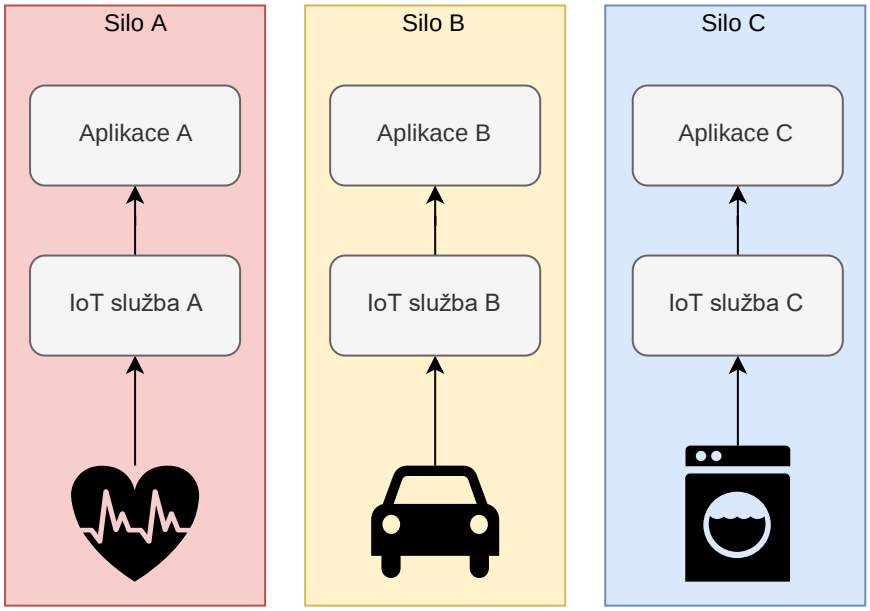
kde se odchylují od přijatelných konfigurací. Jsou také odpovědní za měření efektivity architektury zabezpečení proti známým zranitelnostem [62, 69].

11 IoT systémy a architektury

Informace v kapitole Tvorba IoT řešení jsou převzaty z online kurzu Cisco [62], z části Chapter 2: IoT Systems and Architectures, pokud není uvedeno jinak.

Jak již bylo zmíněno v první kapitole, nejčastějším formátem znázornění end-to-end komunikace v počítačových sítích jsou vrstvené modely. Použití těchto modelů s sebou přináší několik zásadních benefitů při popisování operací a protokolů, které se na síti vyskytují. Pomáhají s návrhem protokolů operujících na konkrétní vrstvě a s jejich následnou spoluprací s ostatními vrstvami. Podporují soutěživost, jelikož, právě díky protokolům, mezi sebou dokážou komunikovat i zařízení od různých výrobců. Zabraňují, aby se změny na jedné vrstvě projevily na ostatních vrstvách. A v neposlední řadě poskytují sjednocené termíny a pojmy k popisu fungování sítí. K těm nejznámějším patří modely OSI a TCP/IP. V případě modelů v IoT tomu není jinak a jejich účel je téměř identický [62].

Oproti počítačovým sítím v IoT existuje velké množství specifických řešení pro určité oblasti a pro ně jsou v mnoha případech vytvořeny specifické protokoly. Většina IoT řešení se označuje prefixem „Smart“, který vyjadřuje to, že do řešení byla aplikována nějaká forma inteligence. I přesto, že se jedná o využití potenciálu IoT, výsledkem jsou oblasti, které je možné si představit jako „sila“, která obsahují specifické technologie, specifické aplikace a specifická zařízení. Největší problém Internetu věcí je tedy téměř neexistující možnost interoperability. Dalším problémem, který díky těmto „silům“ vyvstává, je vznik spíše velkého množství Intranetů věcí a ne Internetu věcí. Této formy a největší úrovně inteligence těchto systémů může být dosaženo jen s plnou kolaborací všech vertikálních oblastí [70].



Obrázek 43 IoT sila

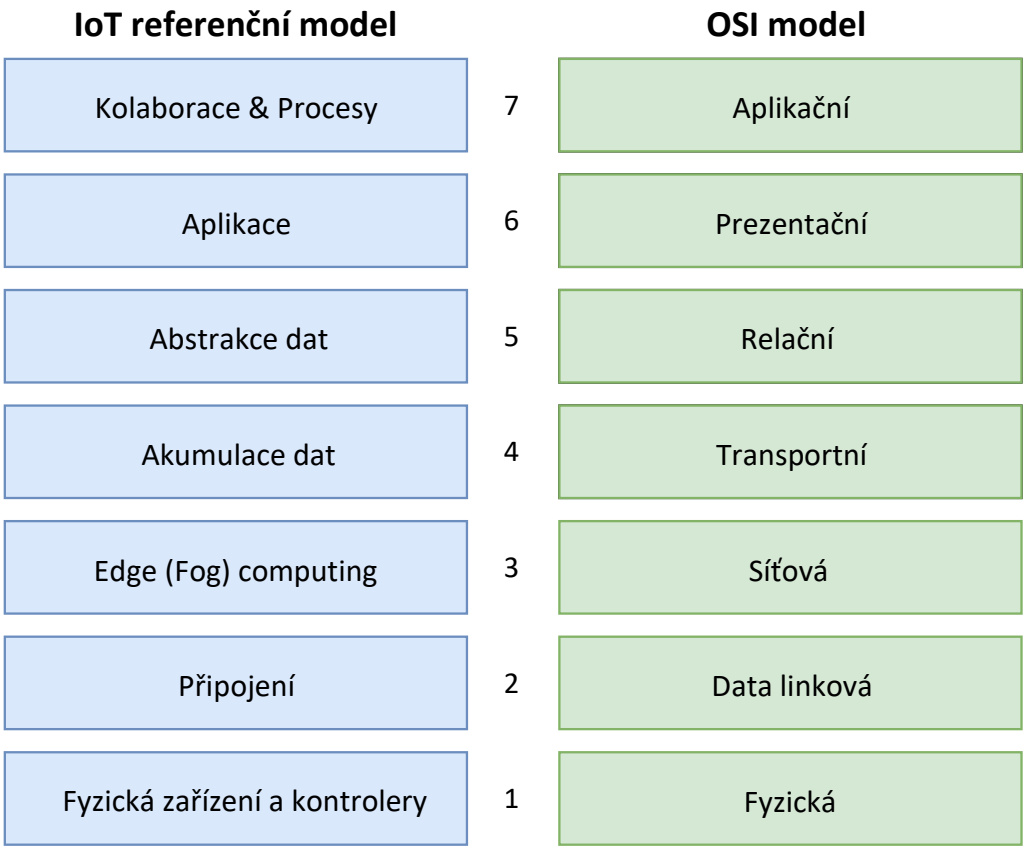
Zdroj: vlastní zpracování (podle [70])

To je příčinou vzniku různých referenčních modelů a architektur, které mají uživatelům poskytnou základní kognitivní pomoc při tvorbě IoT systému. Ty poskytují všem účastníkům společný jazyk, který pomáhá vést diskuzi, spojenou s návrhem. Dalším přínosem architektur je velká vzdělávací hodnota, díky vysoko úrovňovému pohledu na problematiku IoT. Dále poskytují projektovým managerům podklady nejen pro návrh struktury systému, ale i pro potřebnou práci, jelikož architektury zjednodušují identifikaci stavebních bloků IoT systému, což může pomoci s odpovědí na otázky v oblasti systémové modularity, architektury procesorů, potřeby účasti třetích stran, znovupoužití již vyvinutých komponent a podobně. Poskytují tedy společný podklad pro oblast IoT, na jehož základě poté vznikají konkrétní architektury v jednotlivých odvětvích a řešeních. To ale neznamená, že dva systémy navržené podle stejné architektury mají zaručenou interoperabilitu. Jedná se pouze o nástroj, který s tím má pomoci [70].

V kapitole zabývající se IIoT byly již dvě takové architektury představeny, a to ITU a IIRA. Kromě nich existují ještě například ETSI, Purdue, IoT-A, o kterých více dále.

11.1 IoT referenční model

V IoT systému jsou data generována různými typy zařízení, zpracována různými způsoby a přenášena na různá místa. Navržený model se, podobně jako síťový OSI model, skládá ze sedmi úrovní. IoT referenční model neomezuje rozsah ani umístění jeho částí a zároveň dovoluje zpracování na každé úrovni v jakékoli složitosti. Popisuje, jak by měli být řešeny úkoly na každé úrovni, aby byla zachována jednoduchost systému, možnost škálovatelnosti a zajištění podpory. A v neposlední řadě definuje i funkce potřebné k tomu, aby byl IoT systém kompletní. Na obrázku (viz Obrázek 44) jsou znázorněny jednotlivé vrstvy IoT referenčního modelu v porovnání s OSI modelem. Je nutné zmínit, že data v IoT proudí oběma směry, řídící informace od nejvyšší úrovně k nejnižší a z pohledu monitorování naopak.



Obrázek 44 Porovnání OSI a IoT referenčního modelu
Zdroj: vlastní zpracování (podle [62])

Kolaborace a procesy (Lidé a business procesy)

IoT systém a informace, které tvoří, nemají tak velkou hodnotu, dokud se od nich neodvíjejí nějaké akce. Ty často vyžadují procesy a lidi, kteří používají aplikace a příslušná data ke specifickým potřebám a často více uživatelů používá jednu aplikaci k více účelům. Tudíž základem této úrovně nejsou aplikace, ale samotná práce uživatelů s tím, že úroveň níže těmto uživatelům poskytuje správná data, ve správný čas. Tito uživatelé musí být schopni mezi sebou spolupracovat a komunikovat.

Aplikace (Reporty, analýza, řízení)

Software na této úrovni interaguje s daty (data at rest) na úrovni níže a následně je interpretuje do informací. IoT model nijak nedefinuje aplikace. Ty jsou závislé na vertikálním trhu, podstatě dat ze zařízení a podnikatelských potřebách. Některé aplikace se soustředí na monitorování, některé na řízení a to znamená, že složitost aplikací se bude výrazně lišit.

Abstrakce dat (Agregace a přístup)

IoT systémy bude potřeba škálovat do korporátních, mnohdy i globálních, rozměrů a budou vyžadovat několik skladovacích systému, k uskladnění dat ze všech zařízení a z podnikových systémů (ERP, HRMS, CRM). Úroveň abstrakce musí zajistit sladění více formátů dat z různých zdrojů, stálou sémantiku mezi daty napříč zdroji, ujistit se, že data jsou kompletní a připravená pro použití na vyšší úrovni, že jsou zabezpečená vhodnou autentizací a autorizací a podobně.

Akumulace dat (Skladování)

Před dosažením této úrovně se mluví o datech ve formě data in motion. Data se pohybují v síti rychlostí a v podobě určenou zařízeními, které je generují. Zařízení na první úrovni nezahrnují výpočetní kapacity, ale určité výpočetní úlohy se můžou objevit na druhé úrovni, jako překlad protokolů nebo aplikace síťových bezpečnostních zásad. Na třetí úrovni poté může docházet k inspekci paketů. V takovém případě je možné mluvit o fog computingu. Spousta aplikací nepotřebuje nebo nezvládne data zpracovávat ve stejné rychlosti, v jaké přicházejí. Na této úrovni tedy dochází k transformaci data in motion na data at rest, což umožňuje aplikacím k nim přistoupit, když potřebují.

Edge (Fog) computing (Analýza datových elementů a transformace)

Funkce této úrovně se odvíjí od potřeby transformovat datové proudy do podoby, které jsou vhodné pro uskladnění a zpracování na vyšších vrstvách. Soustředí se tedy na analýzu a transformaci velkých objemů dat. Senzor na první úrovni data generuje několikrát za sekundu, a proto bývá základních principem data zpracovávat co nejblíže místu vzniku. Jelikož jsou data přenášena na druhou úroveň v malých objemech, zpracování je běžně prováděno v podobě „paket po paketu".

Připojení (Komunikační a procesní jednotky)

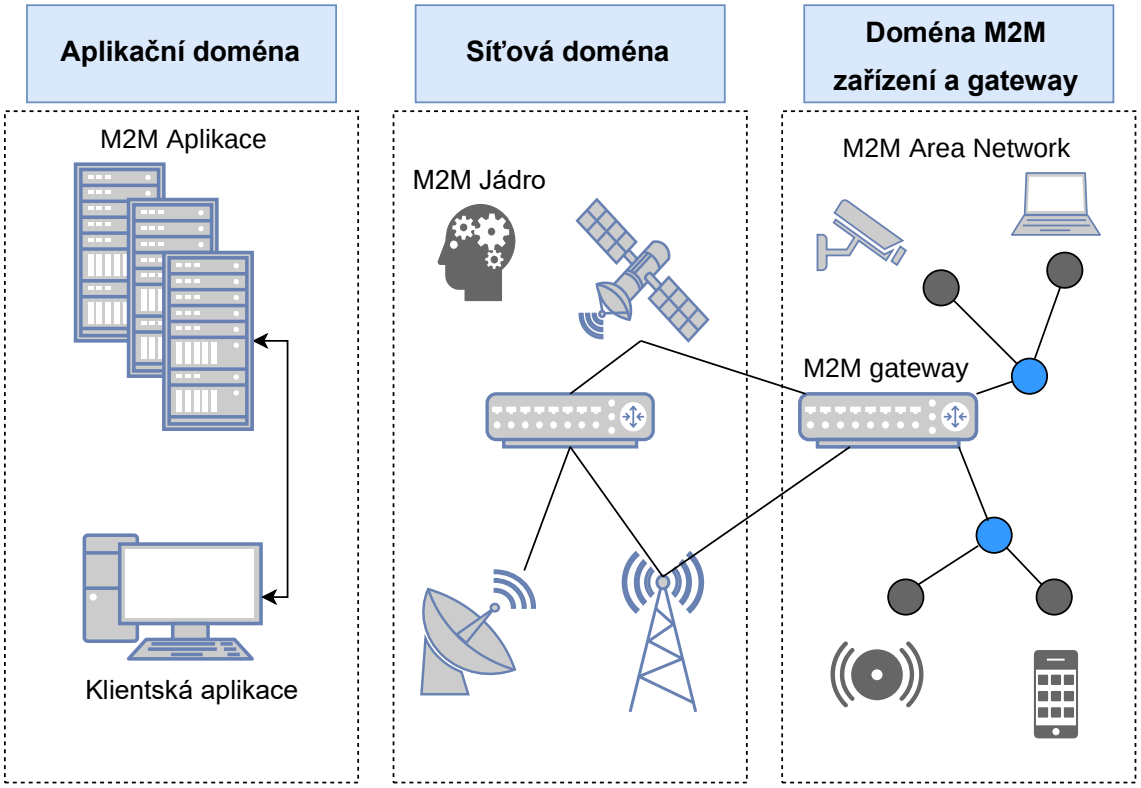
Komunikace a připojení je koncentrováno na této úrovni. Její hlavní funkcí je včasný a spolehlivý přenos dat. To zahrnuje přenos mezi zařízeními, uvnitř sítě a mezi sítěmi. Tento model počítá s tím, že komunikace a přenos dat bude zajištěna již existujícími sítěmi a není potřeba tvořit další sítě. Některá zařízení ovšem nepodporují IP, a proto bude potřeba do komunikace zavést různé prostředníky (gateway). Další zařízení budou potřebovat doplňkové moduly, aby byly vůbec schopné komunikovat. Druhá úroveň tedy propojuje první se třetí.

Fyzická zařízení a kontrolery („Věci")

Nejnižší úroveň IoT referenčního modelu patří samotným zařízením a kontrolerům. Ty se v IoT označují jako „věci" a zahrnují velkou škálu koncových zařízení, které odesílají a přijímají informace. V současnosti již existuje tolik zařízení, že je téměř nemožné je vypsat do seznamu. Jelikož se zařízení výrazně liší a neexistují žádná pravidla na jejich velikost, lokaci ani původ. K tomu jsou vyráběna velkou škálou výrobců.

11.2 ETSI M2M architektura

ETSI je architektura představená již v roce 2008 Evropským institutem pro telekomunikační standardy. Jedná se o architekturu pro machine-to-machine (M2M) komunikaci, která v sobě zahrnuje i IoT zařízení. Architektura cílí především na efektivní end-to-end doručení M2M služeb, a za tímto účelem bylo definováno několik základních požadavků. Ty se zabývají především vlastnostmi týkající se zabezpečení a řízení komunikace. ETSI referenční architektura obsahuje tři domény: aplikační, síťovou a doménu zařízení (viz Obrázek 45) [71, 72].



Obrázek 45 ETSI M2M referenční architektura

Zdroj: vlastní zpracování (podle [62])

Aplikační doména se skládá z několika aplikací, které zahrnují spoustu heterogenních využití (use case) napříč různými odvětvími, jako energetika, automotive, zdravotnictví, doprava apod. Hlavní funkcí každé M2M aplikace je agregace naměřených dat z okolí, provedení výpočtů před rozhodnutím a následné zaslání příkazu, který odpovídá výsledků rozhodnutí.

Síťová doména, někdy označována jako M2M jádro, či základní síť, implementuje funkcionalitu k zajištění komunikace mezi doménou zařízení a aplikační doménou. Poskytuje funkce, jako management zařízení, dosažitelnost (reachability) a běžný komunikační mechanismus. Dodatečně zajišťuje i výměnu dat mezi zařízeními a aplikacemi v obou směrech. Jedním směrem agreguje data získané ze zařízení a předává je aplikacím. Druhým směrem zprostředkovává zařízením informace přijaté od aplikací (úpravy parametrů, ovládání).

Doména M2M zařízení obsahuje heterogenní koncové prvky, jako senzory, aktuátory a koncová zařízení, které jsou propojené pomocí různých technologií

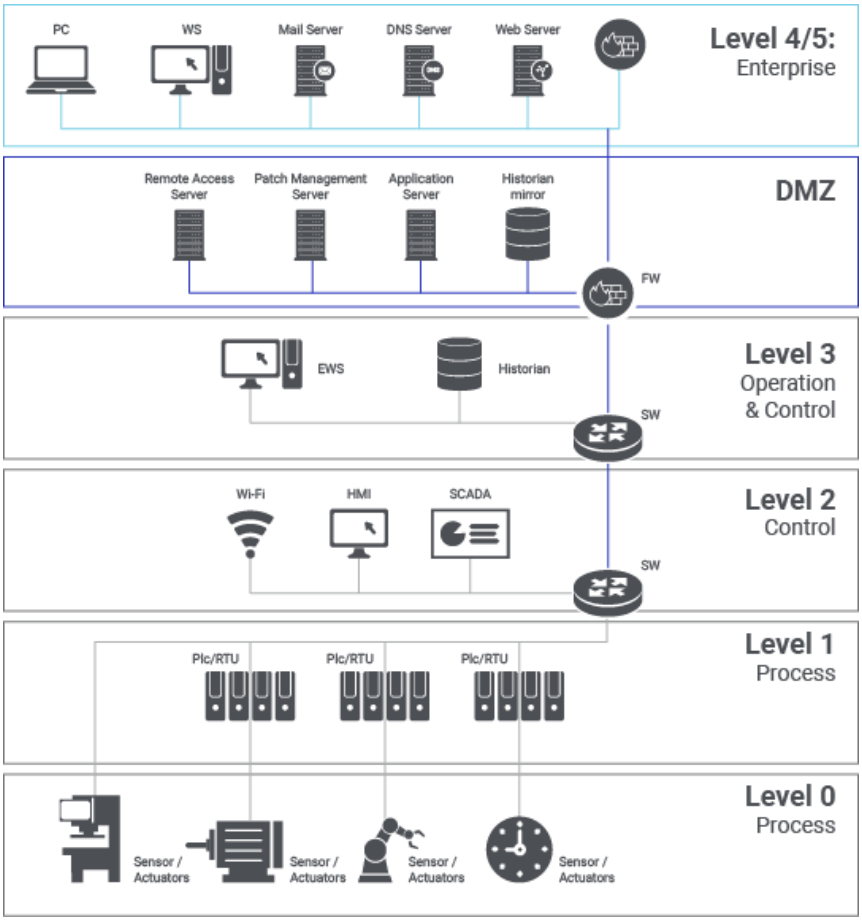
(ZigBee, Bluetooth apod.). Tuto část sítě uzavírá takzvaná M2M gateway, která schovává složitost sítě před zbytkem entit, které se účastní komunikace.

ETSI na druhou stranu neposkytuje žádnou specifikaci toho, jakou strukturu má mít síť M2M zařízení, ani základní síť. Místo toho umožňuje M2M aplikacím spouštět určité zásady na straně sítě nebo načítat informace ze sítě. Pomocí zásad lze definovat například činnost zpracování a přenosu dat [71, 72].

11.3 Purdue Model

Informace v této kapitole pocházejí z příspěvku na webu společnosti ZScaler, která nabízí služby v oblasti zabezpečení sítě a cloudových řešení [73].

Tento model byl představen již v 90. letech a jeho celý název je Purdue referenční podniková architektura (Purdue Enterprise Reference Architecture – PERA) a jak název napovídá, jeho cílem byly především podniky. PERA odvádí skvělou práci v definici jednotlivých vrstev, které jsou kritické v podnikové struktuře a metod jejich zabezpečení. Pokud byl tento model správně implementován, dalo se mluvit v té době o nadčasové architektuře, jelikož díky ní bylo možné vyplnit mezeru mezi průmyslovými řídícími systémy (ICS), operativními technologiemi (OT) a informačními technologiemi (IT). Na obrázku (viz Obrázek 46) jsou vidět jednotlivé vrstvy architektury a příklady jejich součástí.



Obrázek 46 Purdue model

Zdroj: [73]

Úroveň 4/5 – Podnik

Reprezentuje typickou IT vrstvu, kterou známe ze současnosti a kde se objevují primární podnikové funkce. Tato vrstva udává směr podniku a řídí výrobní operace. Zahrnuje ERP systémy, které řídí výrobní plány, využití materiálů, dodávky a inventář. Jakékoli narušení na této vrstvě se může projevit v několika denní odstávce, a to znamená výrazné finanční ztráty.

Úroveň 3.5 – Demilitarizovaná zóna (DMZ)

Demilitarizovaná zóna nebyla součástí modelu od začátku, ale byla představena v pozdějších úpravách. Tato vrstva zahrnuje bezpečnostní systémy, jako proxy a firewally. Zde dochází ke konvergenci IT a OT, což výrazně zvyšuje možnosti pro útok na OT systémy. Spousta výrobní podniků buď tuto úroveň nemá, nebo je její funkcionalita velmi omezená. Rozvoj automatizace navýšil potřebu

obousměrného toku dat mezi IT a OT systémy. Spojení těchto dvou světů zároveň představuje velkou konkurenční výhodu pro podniky, a to obecně zrychluje digitalizaci.

Úroveň 3 – Výrobní operační systémy

Na této úrovni dochází k řízení pracovního toku ve výrobě. Přizpůsobené systémy založené na běžných operačních systémech, jsou zde využívány k řízení výrobních várek, záznamu dat a ke správě výkonosti strojů a podniku obecně. Systémy na této úrovni se označují jako výrobní exekuční systémy (MES). Podoba MES je závislá na vyráběném produktu. Kromě těchto systémů se zde nacházejí i databáze pro záznam provozních dat. Komunikace mezi podnikovou úrovní a úrovní výroby je většinou realizována skrz síť k tomu určenou. Podobně jako na nejvyšší úrovni, jakékoli vyrušení může vést k obrovským prostojům ve výrobě.

Úroveň 2 – Řídící systémy

Systémy SCADA jsou zde používány pro dohled, monitorování a řízení fyzický procesů. Umožňují řízení systémů na velkou vzdálenost od fyzické lokace dílen a oproti tomu distribuované řídící systémy (DCS) a programovatelné logické kontrolery (PLC) jsou většinou nasazeny přímo v prostorách dílny. Základní řízení a monitorování DCS a PLC poskytuje takzvané stroj-člověk rozraní (HMI) a SCADA systémy data agregují a předávají vyšší vrstvě.

Úroveň 1 a 0 – Inteligentní zařízení a Fyzické procesy

Zahrnuje snímaní a manipulaci fyzických procesů, které se na této vrstvě provádějí, pomocí senzorů, analyzátorů, aktuátorů apod. Za účelem zvýšení efektivity senzory komunikují skrz mobilní síť přímo s monitorovacím softwarem výrobce v cloudu. A poslední nultá úroveň reprezentuje samotné fyzické procesy.

I přesto, že se Purdue model stal ikonickým standardem ve světě automatizace, v posledním letech se diskutuje jeho relevantnost ve stávajících IoT systémech. Tok dat v IoT systémech není totiž tak hierarchický, jak ho Purdue popisuje. Samotné senzory a aktuátory (Úroveň 0 a 1) mají často už vlastní formu inteligence, což je příčinou toho, že se hrozby objevují daleko níže, než tento model

předpokládal. V současnosti se edge computing stává běžným a velké množství dat, která jsou sbírána na nejnižších úrovních, jsou zde i do určité míry zpracována a následně zaslána přímo do cloudu. Tím dochází k obejití hierarchie, kterou Purdue popisuje. Ale i tak experti nabádají k tomu, aby se Purdue model zcela nezapomněl. Stálé totiž slouží požadavkům segmentace pro bezdrátové i drátové sítě a chrání OT síť od neoprávněného provozu a zneužití. To jsou klíčové věci, které je třeba zachovat k zajištění stabilní výroby a bezpečnosti obsluhujících zaměstnanců. Dále navrhují hybridní řešení, které spojuje segmentaci Purdue modelu a flexibilitu potřebnou pro průmyslové IoT systémy. Toto spojení je dosaženo přidáním nové vrstvy reprezentující průmyslový edge computing software. Tato vrstva může být umístěna na druhou nebo třetí úroveň a poskytnout možnost sběru dat z OT zařízení na nejnižších úrovních a zároveň usnadnit zisk dat z nejvyšších IT vrstev. To zajistí to, že hierarchie Purdue modelu může být přeskočena, když je třeba tím, že se data pošlou skrze tuto vrstvu a stále bude zajištěna jejich bezpečnost [74].

11.4 Architektura Zero Trust

Informace pocházejí z oficiální dokumentace vydané Národním institutem pro standardy a technologie od autorů Rose, Borchert, Mitchell a Connelly [75].

Zero Trust architektura je v kybernetické bezpečnosti vzor pro ochranu prostředků, založený na principu, že důvěra není udělována implicitně, ale vždy musí být ověřena. Jedná se o end-to-end přístup k bezpečnosti podnikových zdrojů a dat, který zahrnuje identitu (osobní i neosobní entity), pověření, správu přístupu a další.

Jednoduchý model přístupu ZTA se dělí na nedůvěryhodnou a důvěryhodnou zónu. V důvěryhodné zóně se nacházejí veškeré prostředky, ke kterým je následně přistupováno z nedůvěryhodné zóny. Na hranici mezi nimi se nachází bod bezpečnostního rozhodování (Policy decision point – PDP) a bod vymáhání zásad (Policy enforcement point – PEP). Zde systém ověřuje, zda žádající subjekt je autentický a požadavek je validní a následně rozhoduje o udělení či neudělení požadovaného přístupu. To naznačuje, že princip zero trust je aplikován především v autorizaci a autentizaci. PDP/PEP aplikuje takovou sadu pravidel, že veškerý provoz za tímto bodem má stejnou úroveň důvěry. Zároveň nemůže uplatňovat

žádné dodatečné zásady v této zóně. Proto aby byl bod PDP/PEP co nejpřesnější, musí být implicitní důvěryhodná zóna co nejmenší.

Základní zásady, které by měli být dodrženy při implementaci ZTA, jsou:

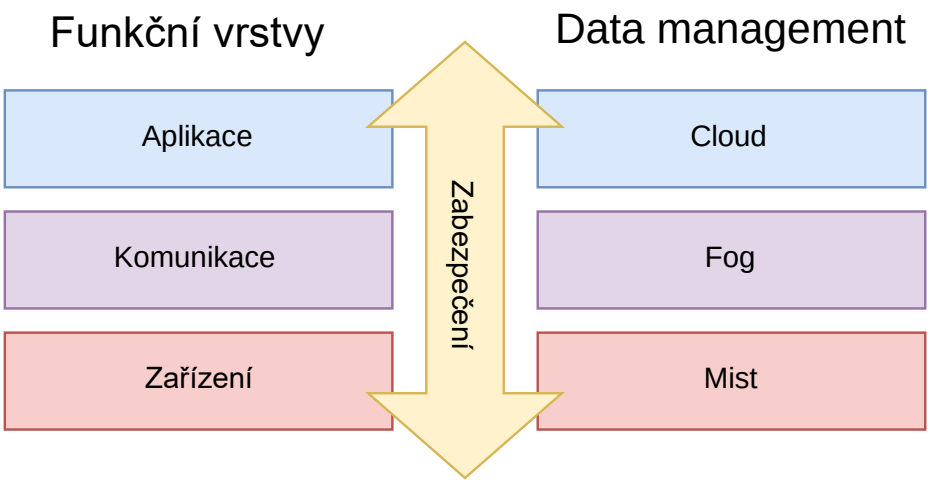
- Všechny zdroje dat a výpočetní služby jsou považovány za prostředky.
- Veškerá komunikace je zabezpečená bez ohledu na umístění sítě.
- Přístup k prostředkům podniku je přidělen při každé relaci (per-session basis).
- Přístup k prostředkům je určen dynamickou politikou, která zahrnuje pozorovatelné atributy, jako stav identity klienta, aplikace, služby a požadovaného prostředku a může zahrnovat i další atributy chování a prostředí.
- Podnik monitoruje a měří integritu a bezpečnostní postavení všech vlastněných a přidružených aktiv.
- Veškerá autentizace a autorizace je dynamická a přísně vymáhána před povolením přístupu.
- Podnik sbírá co nejvíce informací o současném stavu aktiv, sítě a komunikace a využívá je ke zlepšení bezpečnosti.

Zároveň ZTA i definuje několik předpokladů, kterých by se podniky měly držet v průběhu plánování a tvorby sítě. Týkají se jak infrastruktury sítě vlastněné podnikem, tak prostředků vlastněných podnikem, které operují na nepodnikové infrastruktuře (cloudové služby atd.). ZTA na síť nahlíží takto:

- Celá privátní síť podniku není považována za implicitní důvěryhodnou zónu.
- Zařízení na síti nemusejí být vlastněna ani spravována podnikem.
- Žádný prostředek/zdroj není v podstatě důvěryhodný.
- Ne všechny prostředky vlastněné podnikem jsou umístěny na podnikové infrastruktuře.
- Vzdálené podnikové subjekty a aktiva nemohou plně důvěřovat připojení lokální sítě.
- Aktiva a pracovní toky pohybující se mezi podnikovou a nepodnikovou infrastrukturou by měli mít důslednou a pevnou bezpečnostní politiku.

11.5 Jednoduchý IoT model

Již několikrát bylo zmíněno, že IoT systémy mohou být velmi rozsáhlé a komplexní. Tím se rozumí tisíce senzorů, aktuátoru a dalších zařízení. Ty se připojují k různým branám s využitím několika protokolů a ty se následně připojují k internetu a cloudovým aplikacím pomocí dalších protokolů. Aby bylo možné takový IoT systém zabezpečit, je důležité pochopit, kde v systému se zranitelnosti vyskytují. Zjednodušením systému na funkční oblasti může velmi pomoci s pochopením komplexních systémů. Na obrázku (viz Obrázek 47) je znázorněn jednoduchý IoT model, který se velmi podobá výše zmíněné architektuře ETSI M2M. Zde dochází ale k rozdělení funkčních částí a částí zabývajících se daty.



Obrázek 47 Zjednodušený IoT model
Zdroj: vlastní zpracování (podle [62])

Z pohledu funkčních vrstev se řeší především to, jak jsou věci připojené k síti. Například vrstva zařízení by u zavlažovacího systému obsahovala jednotlivé zavlažovací hlavy, senzory vlhkosti, senzory teploty a aktuátory. Na komunikační vrstvě jsou poté připojeny do lokálního řídícího systému, který monitoruje stav. Na aplikační vrstvě je řídící systém poté připojen ke vzdálenému datovému centru, kde se setkává s ostatními systémy.

Z pohledu managementu dat je naopak důležité to, kdy a kde jsou data zpracována. V případě Mist jsou data zpracována prakticky tím samým zařízením, které je generuje. Na příkladu zavlažovacího systému by to znamenalo, že jednotlivé

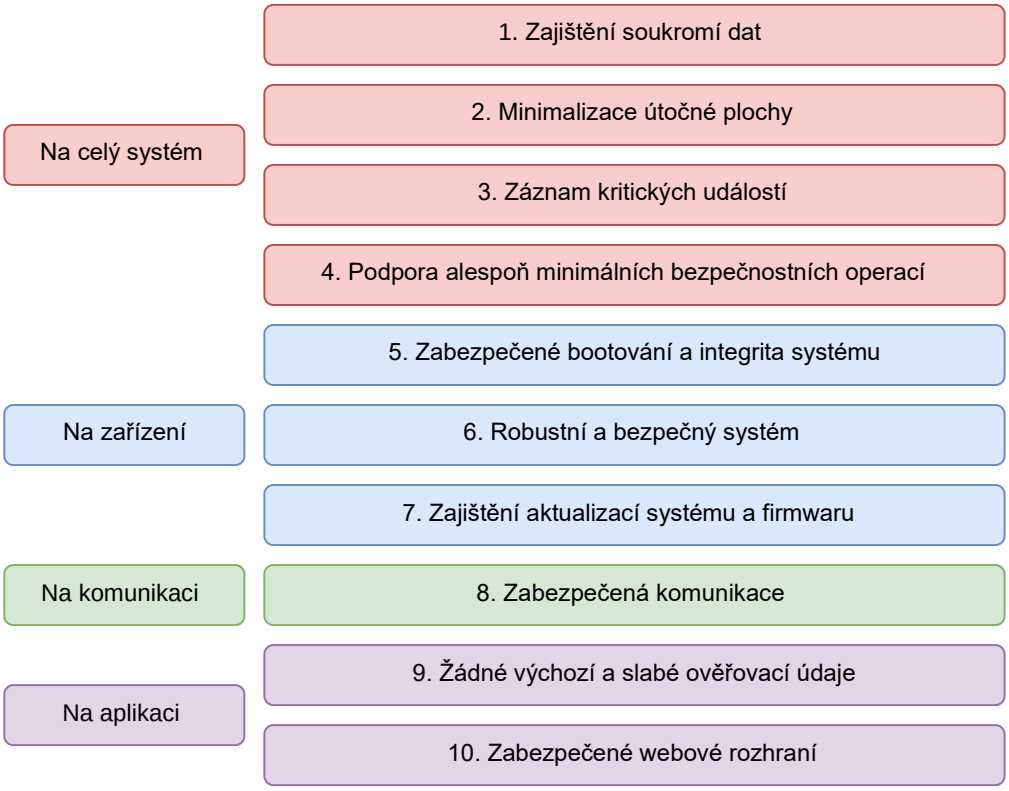
zavlažovací hlavy dokážou například měřit teplotu a vlhkost a na základě měření se autonomně rozhodovat. Pokud by docházelo ke zpracování dat až řídícím systémem, připadalo by to na vrstvu Fog. A pokud je možné ovlivňovat chování vzdáleně pomocí cloudové aplikace, mluví se o vrstvě Cloud. Bez ohledu na funkční aspekty nebo na management dat musí zabezpečení protínat všechny vrstvy.

11.6 Požadavky na zabezpečení IoT

Za základní požadavky, nejen na systémy IoT, ale obecně v otázce kybernetické bezpečnosti, se považuje takzvaná CIA triáda. Zkratka CIA označuje trojici těchto požadavků:

- **Důvěrnost** (Confidentality) – Kontrola nad přístupem k informacím a jejich zveřejňováním. Přenášená a uskladněná data jsou šifrována.
- **Integrita** (Integrity) – Zamezení nesprávné manipulaci s daty a informacemi, tzn. přidávání, úprava nebo jejich odstranění. Tím se rozumí i využití hashovacích funkcí k ochraně přenášených dat. S tím souvisí i řízení přístupy k uloženým datům.
- **Dostupnost** (Availability) – Zajištění přístupu k informacím a datům kdykoli je potřeba. V prostředí IoT to znamená, že zařízení mohou samostatně komunikovat na síti, aby mohla poskytovat data řídícím aplikacím.

Na základě CIA triády poté společnost Cisco představila sadu deseti kritických požadavků specifických pro systémy IoT. Ty je možné dále rozdělit podle povahy do čtyř funkčních vrstev, a to: požadavky na celý systém, na zařízení, na komunikaci a na aplikace (viz Obrázek 48).



Obrázek 48 Požadavky na zabezpečení IoT systému
Zdroj: vlastní zpracování (podle [62])

Požadavky na celý systém

- **Zajištění soukromí dat** – Jedná se o základní požadavek pro důvěrnost, tak jak bylo uvedeno v CIA. Zahrnuje veškerá podniková data, v pohybu i v úložištích. Soukromí dat zahrnuje minimálně ochranu osobních údajů (PII).
- **Minimalizace útočné plochy** – Útočná plocha označuje veškeré vstupní body do systému, které by mohly být zneužity k útoku. Tímto vstupním bodem může být i jedno nezabezpečené zařízení, skrze které se poté útočník dostane do celé sítě. Minimalizací útočné plochy se rozumí zabezpečení všech přístupových bodů a uzavření všech nepotřebných.
- **Záznam kritických událostí** – Tzv. logování označuje proces záznamu událostí v systému. Logování by mělo zahrnovat jak běžný provoz systému, tak především záznam neobvyklých jevů a situací.

- **Podpora alespoň minimálních bezpečnostních operací** – Tento požadavek se týká samotných zaměstnanců. Ti vyžadují řádné vedení a proškolení. Minimálně musejí být schopni monitorovat systémy pro bezpečnostní incidenty, zaznamenávat nově objevené slabiny a hrozby, prošetřovat narušení bezpečnosti systému.

Požadavky na zařízení

Tyto požadavky se týkají IoT zařízení, jako senzory, aktuátory, kontrolery apod. Všechna tato zařízení k fungování vyžadují operační systém, aplikace a data.

- **Zabezpečené bootovaní a integrita systému** – Opatření, která zajistí, že OS a software na zařízeních není napaden hackery nebo malwarem.
- **Robustní a bezpečný systém** – Zamezení spouštění nepotřebných síťových služeb operačním systémem. Takové služby mohou být zneužity jako potenciální přístupový bod útočníky.
- **Zajištění aktualizací systému a firmwaru** – Možnost aktualizovat firmware a operační systém na všech zařízeních, v případě nálezu nových bezpečnostních zranitelností, je klíčovým požadavkem. Jelikož je spousta zařízení nasazena v terénu, je vyžadován bezpečný mechanismus pro aktualizaci zařízení a jejich součástí ideálně skrz síť.

Požadavky na komunikaci

IoT komunikace je realizována oběma směry. Senzory zasílají data aplikacím a aplikace řídí aktuátory skrz síť. Přenášená data mohou být odcizena, poškozena, zničena, zmanipulována. Řídící komunikace může být změněna nebo kompletně padělána. Ať už nastane jakákoli situace, dopady mohou být nepředstavitelné.

- **Zabezpečená komunikace** – IoT systémy musejí mít implementována odpovídající opatření, která budou data chránit od napadení a manipulace. Dalším opatřením je i využití technik ověřování přijatých dat na to, zda pocházejí z autentických zdrojů.

Požadavky na aplikace

- **Žádné výchozí a slabé ověřovací údaje** – Implementace silného ověřovacího procesu na všech IoT zařízeních. Výchozí ověřovací údaje od výrobce musejí být změněna předtím, než dojde k nasazení zařízení. Délka a struktura hesla by minimálně měla odpovídat obecným zásadám pro tvorbu hesel.
- **Zabezpečené webové rozhraní** – IoT zařízení využívají API k přímé komunikaci s webovými aplikacemi. Ověřovací údaje použité k interakci zařízení a webové aplikace musí být chráněna před útoky.

11.7 Model pro analýzu hrozeb

Model pro analýzu hrozeb je nástroj primárně určený pro úlohy v oblasti řízení rizik a hodnocení zranitelností. Jedná se o strukturovaný přístup pro analýzu bezpečnosti a zranitelností systémů. Systémem může být hardware zařízení, software nebo komunikační sítě. Struktura, která je představena v této kapitole, je adaptací modelu od společnosti Microsoft přizpůsobená pro IoT systémy. Kromě celkového počtu kroků procesu analýzy hrozeb neobsahuje oproti původnímu modelu žádné zásadní změny. Jednotlivé kroky jsou:

1. Identifikace bezpečnostních cílů

Prvním krokem, od kterého se následné odvíjí zbytek procesu, je samotná identifikace a určení bezpečnostních cílů. K tomu je možné využít následující kategorie:

- **Identita** – Dokumentace řídících prvků, které zajišťují sběr důkazů o totožnosti uživatelů, kteří mají přístup k IoT systému. Uživatelem může být člověk nebo jiné zařízení.
- **Finance** – Dokumentace finančních rizik různých aspektů IoT systému, aby mohla být vedením stanovena přijatelná úroveň rizika (např. ztráta senzory vs. Ztráta síťového řadiče)

- **Reputace** – Dokumentace všech možných dopadů na reputaci společnosti v případě, že dojde k napadení systému. Například dopad na reputaci výrobce webových kamer, které byly použity pro rozsáhlý DDoS útok.
- **Soukromí a nařízení** – Dokumentace dopadů obav o soukromí a ostatních požadavků definovaných v různých nařízeních. Například data naměřená teploměrem budou mít méně požadavků na soukromí (samozřejmě v závislosti na podobě dat), než data naměřená fitness zařízením.
- **Garance dostupnosti** – Dokumentace očekáváné dostupnosti a garantované provozuschopnosti IoT systému. Například časová tolerance při výpadku průmyslového řídícího systému (ICS) bude pravděpodobně velmi malá a bude vyžadována implementace příslušných bezpečnostních opatření a redundance systému.
- **Bezpečí** – Dokumentace potenciálních dopadů na fyzické blaho lidí a fyzické poškození vybavení a výrobních prostor. Velmi důležití především v průmyslovém odvětví.

2. Dokumentace architektury systému

Druhý krok, jak už název napovídá, zahrnuje kompletní a detailní dokumentaci architektury a uspořádání zkoumaného IoT systému. Součástí dokumentace by měl být seznam všech komponentů IoT systému na všech funkčních vrstvách, tzn. aplikační, komunikační a na vrstvě zařízení. Dále znázornění veškerého datového toku v systému, například pomocí DFD (Data Flow Diagram) a přehled všech využitých technologií, protokolů a standardů.

3. Rozložení IoT systému

Zde dochází k detailnějšímu popisu individuálních součástí a vlastností systému, které byly definovány a zdokumentovány v předchozím kroku. Výstupem v této fázi by měl být kompletní profil zabezpečení, který se následné využije jako podpora a pomoc při identifikaci hrozeb a zranitelností v návrhu, implementaci a nasazení systému. K tvorbě výstupu se používají následující úkoly:

- Určení hranic důvěryhodnosti mezi důvěryhodnými a nedůvěryhodnými součástmi systému.

- Tvorba diagramu datového toku (DFD) mezi zařízeními, komunikačními sítěmi, aplikacemi.
- Dokumentace míst, kde dochází ke vstupu dat do systému.
- Určení citlivých dat v místech, kde dochází k jejich ukládání a manipulaci
- Rozšířit dokumentaci profilu zabezpečení o použité způsoby a pravidla k ověřování přístupu, autentizaci, autorizaci, konfiguraci atd.

4. Identifikace a ohodnocení hrozeb

V předposledním čtvrtém kroku dochází konečně k samotné identifikaci hrozeb a k určení jejich míry rizika. Hrozba je potenciální nebezpečí pro všechna aktiva, jako data nebo komponenty IoT systému. Útočníci (v anglické literatuře je často označují pojmem „threat actors“) jsou lidi nebo entity zneužívající zranitelnosti systému. Zranitelnosti jsou slabiny v IoT systému, které mohou být zneužity a představují právě zmíněnou hrozbu. Kombinace několika zranitelností poté tvoří takzvanou „útočnou plochu“, což není nic jiného než místa, kterými se útočník může do systému dostat a získat přístup k datům.

Pro pomoc s procesem identifikace a hodnocení hrozeb je možné využít několik nástrojů. Tento kurz konkrétně využívá nástroj STRIDE pro identifikaci hrozeb a DREAD pro určení jejich rizika. STRIDE je zkratka pro pět kategorií hrozeb, a to konkrétně pro falešnou identitu (**S**poofing identity), manipulaci a falšování dat (**T**ampering with data), zakrytí stop po vniknutí (**R**epudiation), únik dat (**I**nformation disclosure), znepřístupnění služby (**D**enial of service) a zvýšení oprávnění (**E**levation of privilege). DREAD je nástrojem krizového managementu a s jeho pomocí je možné kvantifikovat, porovnat a stanovit prioritu rizika, které odpovídá jednotlivým hrozbám nalezeným během procesu STRIDE. Riziko hrozeb lze určit sečtením proměnných poškození (**D**amage), reprodukovatelnost (**R**eproducibility), zneužitelnost (**E**xploitability), ovlivnění uživatelé (**A**ffected users), Viditelnost (**D**iscoverability) a vydělením součtu číslem 5. Oba modely budou podrobněji představeny v poslední kapitole, která se věnuje specificky procesu identifikace a hodnocení hrozeb.

5. Doporučení vhodných metod a technologií

Po identifikaci hrozeb a určení jejich rizika nezbývá nic jiného, než navrhnout a implementovat odpovídající opatření, s cílem eliminovat nebo alespoň snížit míru rizika. Během tohoto kroku je nutné i dbát na to, aby navržené metody dávaly smysl z obchodní perspektivy, a aby to bylo v souladu s existujícími pravidly a zásadami v organizaci.

12 Útok na IoT na úrovni zařízení

Informace v kapitole Tvorba IoT řešení jsou převzaty z online kurzu Cisco [62], z části Chapter 3: The IoT Device Layer Attack Surface, pokud není uvedeno jinak.

Oblasti IT a IoT jsou z pohledu komponent, komunikace, připojení a ověřovacích postupů velice podobné, ale existují oblasti specifické pro systémy IoT, kterým je nutné věnovat pozornost. OWASP (Open Web Application Security Project) je nezisková komunita, která publikuje články a různé studie zaměřené především na bezpečnost softwaru. Kromě publikací pořádají i vzdělávací konference a semináře. Právě OWASP sestavil seznam běžných a rozšířených hrozeb pro jednotlivé útočné plochy. První touto vrstvou je, jak název kapitoly napovídá, vrstva zařízení. Dalším dvěma vrstvám, konkrétně komunikační a aplikační vrstvě, budou věnovány následující kapitoly.

Na úrovni zařízení by se hrozby definované OWASP daly rozdělit do pěti hlavních kategorií. První kategorií jsou **hardwarové senzory**, čímž se rozumí fyzická zařízení používána ke snímaní a měření prostředí (senzory teploty, vlhkosti, světla apod.). Taková zařízení jsou často umístěna na místech, kde může velmi snadno dojít k ovlivnění naměřených dat, ať už úmyslně nebo neúmyslně, nebo obecně k neoprávněné manipulaci a s tím i k fyzickému poškození zařízení. Příkladem neúmyslného ovlivňování měření teploměru může být přítomnost větráku v jeho blízkosti. Pak v závislosti na tom, jak moc je hodnota teploty kritická pro fungování jiných zařízení, může tato situace mít buď téměř žádné, nebo i velmi vážné následky. Zřejmě nejjednodušší obranou proti manipulaci a poškození měřících zařízení, je instalace kamerových systémů a dalších podobných bezpečnostních opatření podle umístění zařízení.

Druhou kategorií je **paměť zařízení**, konkrétně neoprávněný přístup k ní. Jsem patří hrozby, jako výchozí přihlašovací údaje, citlivá data uložená v paměti, přihlašovací údaje uložené v nezašifrované podobě (plain-text) a uložené šifrovací klíče. Útočník by například mohl zneužít právě výchozí přihlašovací údaje a získat tak přístup k ostatním zmíněným informacím (citlivým datům). Takový útok na jedno zařízení by se pak záhy mohl proměnit v útok na celý systém.

Třetí kategorií, která i v některých ohledech souvisí také s pamětí, jsou **fyzická rozhraní zařízení**. Spousta IoT zařízení využívá vyměnitelná média, jako SD karty, k ukládání dat a operačního systému. V případě, že útočník získá přístup k zařízení, může dojít k vyjmutí karty a k jejímu odcizení nebo zkopírování, a navíc hrozí i odhalení unikátních identifikátorů zařízení jako je sériové číslo. Dále ke komunikaci a připojení k ostatním zařízením používají, kromě protokolu ethernet, i sériová rozhraní, jako UART, I2C, SPI, která byla představena v 5. kapitole (viz 5. kapitola Senzory, aktuátory a mikrokontroléry), a Joint Test Action Group (JTAG). JTAG není komunikační protokol, ale průmyslový standard určený pro ověřování návrhů a testování desek tištěných spojů (PCB). Připojením k těmto rozhraním může útočník získat přístup k terminálu zařízení a potenciálně odcizit nebo upravit data. Následně do zařízení může nainstalovat tzv. „zadní vrátka“, které mu poskytnou stálý přístup do zařízení. V případě rozhraní JTAG může navíc dojít k reverznímu inženýrství, což označuje proces, kdy je účelem zjištění fungování a konstrukce daného zařízení nebo předmětu.

Poslední dvě kategorie jsou **firmware zařízení** a **mechanismus aktualizace firmwaru**. Pokud není firmware dostatečné chráněn, může být ze zařízení extrahován a následně buď útočníkem upraven nebo zcela nahrazen. Potenciální hrozbou týkající se firmwaru je i obyčejná možnost zobrazit verzi firmwaru, jelikož následně útočník může využít zranitelnosti konkrétní verze. Proto udržovat verze všech aplikací, operačních systémů a firmwarů aktuální je kritické. To má vliv na celkovou bezpečnost zařízení a tím pádem i systému, protože, jak se často říká, bezpečnost systému je rovna bezpečnosti nejméně a nejhůře zabezpečeného prvku. Ovšem i při aktualizaci softwarových komponent je třeba dbát na to, aby update pocházel z ověřeného zdroje.

12.1 Omezená zařízení

Jak již bylo několikrát zmíněno, IoT se v drtivě většině skládá z „omezených“ zařízení, tzn. omezený výpočetní výkon, paměť. Sdružení Internet Engineering Task Force (IETF) publikovalo dokument s označením RFC 7228 [76], ve kterém definovalo celkem tři výkonnostní třídy těchto zařízení s označením C0, C1, C2. Nejnižší třída C0 zahrnuje zařízení, které podle definice mají operační paměť RAM menší než 10 kB a paměť flash menší než 100 kB. Do třídy C1 spadají zařízení s pamětí RAM okolo 10 kB a pamětí flash okolo 100 kB. Zařízení spadající do třídy C0 a C1 mají kvůli nízkému výkonu také omezené komunikační možnosti. Pokud se nějaké komunikace účastní, je velmi nepravděpodobné, že budou mít implementovány šifrovací mechanismy. Poslední třídou je C2, které zahrnuje zařízení, u kterých se paměť RAM pohybuje okolo 50 kB a paměť flash okolo 250 kB.

- **Třída 0 (C0)** – Do této třídy spadají zařízení velmi limitovaná výkonem. Většina z nich neobsahuje ani prostředky pro přímou a bezpečnou komunikaci s internetem. S tím jim pomáhají výkonnější zařízení, jako proxy, gatewaye nebo servery. Velmi obtížně se zabezpečují a spravují.
- **Třída 1 (C1)** – Ani zařízení v třídě C1 nedisponují dostatečným výkonem, aby mohla komunikovat s ostatními uzly na internetu s využitím protokolů HTTP a TLS. Dokážou však využít protokoly speciálně navržené pro takto omezená zařízení, jako CoAP (UDP), a účastnit se tak konverzací bez pomoci gatewaye.
- **Třída 2 (C2)** – Zařízení v třídě C2 mají již dostatečný výkon na to, aby dokázala implementovat stejné protokoly jako se používají na počítačích a serverech. I přesto se doporučuje využívat protokoly určené pro IoT, čímž dojde nejen k úspoře energie a přenosového pásma, ale zbude i dostatek místa pro aplikace.

Omezená zařízení, která se týkají tohoto kurzu, je možné rozdělit do tří kategorií: **chytré senzory, integrována zařízení, prototypová zařízení**. Senzory zde byly dávno před IoT. S příchodem IoT jim byla dána určitá úroveň inteligence a v současnosti již dokážou komunikovat s monitorovacím systémem a mají

schopnost sebe diagnózy. Integrovaná zařízení je často výpočetní systém navržený a sestrojený pro konkrétní účel. Operační systém takových zařízení často dokáže spouštět pouze jednu aplikaci. Pod prototypovými zařízeními si lze představit zařízení typu Raspberry Pi a Arduino a často slouží k tvorbě prototypů právě pro integrovaná zařízení [62].

12.1.1 Typy IoT procesorů

Majoritními typy procesorů používaných v IoT jsou ARM, MIPS a x86. Obecně se ale procesory dělí do dvou velkých kategorií: procesory s redukovanou instrukční sadou (RISC) a procesory s komplexní instrukční sadou (CISC).

RISC

Procesory RISC typicky obsahují méně tranzistorových jednotek než CISC, což znamená menší náklady na výrobu, méně spotřebované energie a méně produkovaného tepla. To je dělá skvělými kandidáty pro mobilní a IoT zařízení. Instrukční sada (ISA) těchto procesorů využívá jeden počítačový cyklus k vykonání jedné instrukce a obsahuje malý soubor jednoduchých instrukcí. Dvě hlavní architektury procesorů RISC jsou ARM a MIPS. ARM je architektura obecně licencovaná firmám k tvorbě vlastních procesorů a je dostupná v 32-bit a 64-bit. Například Raspberry Pi obsahuje procesor založený na architektuře ARM. MIPS procesory jsou využívány v integrovaných, síťových, mobilních a IoT zařízeních. Také jsou dostupné v 32-bit i 64-bit. V současnosti mají RISC procesory na trhu mobilních výpočetních zařízení drtivou převahu, a to především jednotky vyvinuté podle architektury ARM.

CISC

CISC procesory dokážou s jednou instrukcí vykonat několik operací. Obsahují více tranzistorů, což, v porovnání s RISC, znamená větší náklady, více spotřebované energie a více vyprodukovaného tepla. Výhodou je to, že komplexní instrukční sada dokáže výrazně snížit velikost programového kódu. To usnadňuje načítání a ukládání aplikací. Předními výrobci CISC procesorů jsou známé společnosti Intel a AMD. Obě společnosti v poslední letech projevují veliký zájem o oblast IoT a v současnosti již nabízejí hned několik alternativ procesorů vhodných pro použití

v IoT zařízeních. Intel například oživil stará jména a představil procesory s označením Atom, Celeron a Pentium. U AMD jsou to procesory s označením RYZEN Embedded specificky pro aplikace na úrovni edge a EPYC Embedded, který je specificky určen pro IoT gateway zařízení.

Heterogenní a big.LITTLE výpočetní technika

Tyto dva pojmy označují současné trendy v honbě za snižováním energetických požadavků a zvyšováním výkonu. Heterogenní přístup označuje systémy, které využívají více druhů procesorů nebo procesorových jader. Heterogenita, v případě použití dvou procesorů, je typicky řešena tak, že jeden procesor obsahuje nějakou instrukční sadu a druhý procesor sadu zcela odlišnou. Dříve dvě odlišné sady znamenaly i odlišný přístup k nim. Moderní heterogenní systémy používají jednu sadu s označením Heterogeneous System Architecture (HSA), která jim umožňuje zároveň používat dva různé typy procesorů (typicky CPU a GPU) na jednom integrovaném obvodu. V případě pouze jednoho procesoru se hovoří o heterogenní topologii. Taková jednotka používá jednu ISA, ale jádra mají různý výkon. Konkrétním příkladem jsou procesory typu ARM big.LITTLE. V současnosti již procesory s touto nebo podobnou strukturou existují. Například notebooky Macbook Air od společnosti Apple od roku 2020 používají procesory Apple M1, které obsahují čtyři jádra s vysokým výkonem a čtyři jádra s nízkým výkonem. Kromě toho je součástí M1 i hlavní GPU jednotka. I Intel v roce 2020 vydal hybridní big.LITTLE procesor pod označením Lakefield a zároveň plánuje koncem roku 2021 (Q4 2021) vydat novou řadu s označením Alder Lake (12. generace), ale tentokrát určenou pro desktopové počítače.

12.1.2 Typy paměti v IoT

IoT zařízení používají k ukládání dat, firmwaru a k výpočetním operacím několik typů pamětí. Běžnými typy jsou:

- **SD karta** – SD karty, dnes už microSD karty, jsou nejčastějším úložným médiem pro aplikace a naměřená data na IoT zařízeních. V současnosti se kapacita karet běžně pohybuje v řádech desítek GB. Díky tomu dokážou bez problému pojmout celý operační systém s konfiguračními soubory. Karty

musejí být chráněné proti vyjmutí, aby nedošlo k neoprávněné manipulaci a úniku dat.

- **Non-volatilní paměť** – Tento pojem označuje typy pamětí, které si uložené informace uchovají i poté, co dojde k odpojení od zdroje elektrické energie, což jsou například paměti EPROM a EEPROM. Non-volatilní, někdy semipermanentní, paměti se často používají k ukládání firmwaru a dalších informací důležitých k fungování zařízení. Speciálním typem, se kterým se v IoT uživatel může setkat, je paměť Embedded MultiMediaCard (eMMC), která je často napájena (integrována) přímo na desce, ale existují i odnímatelné verze. Je rychlá, s nízkou spotřebou energie a relativně levná na výrobu. Jelikož se jedná o integrovanou komponentu je lépe chráněna proti vyjmutí a krádeži než SD karty.
- **Volatilní paměť** – Opakem non-volatilních pamětí jsou volatilní, u kterých, při odpojení elektrické energie, dojde ke ztrátě uložených dat. Jedná se například o paměti typu SRAM a DRAM, které se používají k ukládání operačního kódu a jako dočasné uložiště při běhu zařízení.

12.1.3 Integrované systémy

Integrované systémy a zařízení jsou často navrženy tak, aby plnily specifickou funkci ve větších celcích. Příkladem jsou například zařízení domácího bezpečnostního systému. Všechny operace takového systému řídí mikrokontroler, který je naprogramován přímo pro tento účel. Senzory kontroleru poskytují data, který je vyhodnocuje a rozhoduje, zda alarm spustí nebo nikoliv. Mikrokontroler může být schopen informace zobrazovat na displeji nebo komunikovat s ostatním monitorovacím výpočetním vybavením. Některé integrované systémy však nepoužívají celý mikrokontroler, ale pouze mikroprocesor, který vyžaduje další externí komponenty jako paměť. Systém řízený mikrokontrolerem oproti tomu může obsahovat CPU, paměť flash, RAM, sériová komunikační rozhraní a další periferie v jednom integrovaném obvodu. Kromě toho takové systémy mohou obsahovat i integrovaný operační systém nebo mohou být přímo naprogramovány s použitím strojového kódu procesoru. V případě operačního systému se často jedná o upravené verze Linux.

Programování takových zařízení se liší od běžného programování. Počítačový software je často vyvíjen na procesorech, na kterých poté následně poběží. U integrovaných systému je software vyvíjen mimo prostředí, ve kterém bude fungovat, což komplikuje využívání programovacích nástrojů, a především debugging kódu (proces nalézání a redukce chyb). V takovém případě přichází na řadu výše zmíněný JTAG port.

12.1.4 Interpret a kompilátor

Co pojmy interpret a kompilátor označují a jaký je mezi nimi rozdíl, bylo zmíněno v úvodu 6. kapitoly (viz kapitola Softwarové vybavení), takže nyní jen krátce pro připomenutí. Při vývoji aplikačního software pro jakýkoliv systém má vývojář na výběr z velkého množství programovacích jazyků a vývojových prostředí. V případě, že je aplikace určena pro jiné zařízení, než je stolní počítač (např. chytrý telefon), je možné využít takzvaných emulátorů, které napodobují prostředí cílového přístroje. To samé platí i pro zařízení IoT. Kromě toho mají k dispozici i dva typy programovacích jazyků, a to kompilované a interpretované. Hlavní rozdíl mezi nimi je ten, že zdrojový kód psaný kompilovaným jazykem je čitelný v textovém editoru, ale musí být přeložen (kompilován) do strojového kódu, který je naopak čitelný a spustitelný procesorem. Pokud dojde ke změnám ve zdrojovém kódu, bude muset být znovu kompilován. Příkladem takových programovacích jazyků jsou C, C++, Rust, Visual Basic. Oproti tomu kód psaný interpretovaným jazykem je procesorem spouštěn instrukci po instrukci, které interpret překládá do strojového kódu. Pokud dojde k chybě, program se zastaví a můžou být provedeny změny. Nejpopulárnějším jazykem tohoto typu je bezpochyby Python, a další jsou například JavaScript, Perl, PHP. Podle OWASP programový kód představuje velké riziko. Jak interpretovaný, tak kompilovaný kód může být útočníkem pozměněn. U kompilovaného kódu je ale možnost kontroly digitální podpisu spustitelného souboru a zjistit tak, zda nedošlo ke změně.

12.1.5 Běžné IoT operační systémy

Jako většina běžných zařízení, tak i většina IoT zařízení potřebuje ke svému fungování operační systém. Výjimkou je například zařízení Arduino, které nemá žádný operační systém a spouští pouze uživatelem předaný kód. Jedinou překážkou je v tomto případě dostupný výkon a paměť. Z toho důvodu se pro zařízení IoT vyvíjejí speciální operační systémy. Často se jedná o vlastní „osekanou" verzi systému Linux, ale v současnosti již existuje spousta OS navržených specificky pro konkrétní zařízení. Příkladem jsou známé desky Raspberry Pi, která jako operační systém využívá Linux Raspbian, nebo Nvidia Jetson, který běží na systému s názvem Linux for Tegra. Tyto systémy by pro většinu IoT zařízení byly ještě stále příliš náročné a zbytečně komplexní. Ovšem i pro ty nejméně výkonná IoT zařízení existuje široká nabídka operačních systémů a důležité je poznamenat to, že většina z nich jsou open-source.

Jedna z velmi oblíbených možností je **Contiki**, který najde využití v podnikatelské i nepodnikatelské sféře. Velikost programového kódu (tzv. code footprint) se pohybuje okolo 100 kB a paměťová náročnost na RAM se dá omezit až na 10 kB. To umožňuje jeho použití u zařízení ze třídy C1 a výše. Je známý především díky své jednoduchosti v připojování malých a nízkoenergetických mikrokontrolerů k internetu. Reputaci získal především díky své jednoduchosti a užitečnosti při budování bezdrátových systémů v komplexech budov.

Další známou variantou je operační systém s označením **FreeRTOS**, který byl vyvinut společností Amazon a později vydán jako open source. Ke spouštění aplikací využívá cloudovou službu AWS IoT Core. Jeho předností je velmi malý objem programového kódu, který se pohybuje mezi 9-15 kB, což dělá vhodnou volbou i pro ty nejméně výkonná zařízení ze třídy C0.

Dalšími rozšířenými systémy jsou například Embedded Linux, TinyOS, RIOT, Mbed OS, které jsou všechny open-source a z komerčních alternativ například Windows 10 IoT. Určit nejlepší IoT operační systém je téměř nemožné. V podstatě záleží jen na použité desce, uživateli a v některých případech i na účelu [77].

12.2 Zranitelnost fyzických zařízení

Na základě uvedených vlastností je možné určit několik obecných hrozeb, které se omezených zařízení týkají. Ty základní souvisí zejména s tím, že taková zařízení jsou často na takových místech (veřejná místa, nezastavěné oblasti), kde je velmi těžké zajistit jejich fyzickou bezpečnost. Je to například krádež celého zařízení, fyzické poškození zařízení, ať už útočníkem nebo přírodními živly, vyřazení zařízení z provozu, znemožnění zařízení komunikovat apod. Prakticky kdykoli, kdy existuje fyzický přístup k zařízení, existuje i potenciální bezpečnostní riziko a netýká se to pouze omezených zařízení. I senzory mohou být neoprávněně přemístěny, čímž dojde ke ztrátě kalibrace nebo nemožnosti komunikovat s ostatními zařízeními. Spousta senzorů dnes již umí upozornit na nestandardní nastavení. Hrozbou je i manipulace s měřeným prostředím, čímž dojde ke zkreslení dat, a to může ovlivnit následné chování celého systému.

Tam, kde je to možné, by jako první linie obrany měl být instalován minimálně kamerový systém a zařízení by měla být umístěna v odolných obalech. Na obal se doporučuje aplikovat pečeť (např. samolepku), která poskytuje vizuální kontrolu, zda nedošlo k manipulaci se zařízením. Umístění zařízení ve vnitřních prostorech je obecně jednodušší, jelikož má provozovatel systému na daným prostředím kontrolu. Díky tomu lze implementovat dodatečná bezpečnostní opatření, jako zámky nebo alarmy a s tím i související kontrolu přístupu. K zabezpečení fyzického zařízení patří i zajištění záložního zdroje, například formou baterie, jelikož by výpadkem energie mohlo dojít ke ztrátě dat (volatilní paměť).

Další velmi běžné hrozby se týkají přímo hardwaru (sériová rozhraní, paměť apod.). Jedna známá kauza z roku 2017 se týkala paměťových eMMC čipů. Pomocí pěti drátů napájených k čipu a obyčejné čtečky SD karet bylo možné z paměti extrahovat firmware, operační systém a software, uložit je do PC a následně prohledávat na zranitelnosti. Tato hrozba se týkala stovek milionů zařízení a spousta výrobců ihned vydávala opravné aktualizace, ale i tak existuje spousta zařízení, kde je tato zranitelnost stále přítomna.

12.2.1 Zranitelnost firmwaru

Firmware je možné chápat jako integrovaný software zahrnující minimální operační systém a související programy k řízení zařízení, který mu umožňuje operovat. Jako ostatní software, tak i firmware může obsahovat bezpečnostní mezery, které jsou objeveny i několik let po jeho vydání. Hrozby u IoT zařízení týkající se firmwaru převládají a neustále se objevují nové, podobně jako v případě běžných stolních PC a operačních systémů. Konkrétní zranitelnosti jsou například:

- **Výchozí přihlašovací údaje** – I když to tak nevypadá, většina útoků na zařízení je kvůli výchozím přihlašovacím údajům od výrobce, respektive kvůli tomu, že nebyly uživatelem změněny. Je zcela běžné, že výrobci u zařízení používají zcela běžné údaje, jako admin/admin, root/root, admin/1234 apod. Díky tomu se útočník jednoduchým způsobem pokus omyl může na zařízení dostat. I prolomení slabého hesla bude útočníka stát více úsilí. Proto je důležité tyto výchozí údaje změnit ještě předtím, než dojde k připojení zařízení do internetu. I pár minut totiž může útočníkovi stačit k napadení zařízení. Důležité je i to, aby každé zařízení mělo unikátní heslo.
- **DDoS útok** – Není to tak, že by se IoT zařízení často stávala cílem DDoS útoku, ale jsou napadena zcela z opačného důvodu, tedy aby byla jeho součástí. Útočníci často využívají právě výchozí přihlašovací údaje, aby zařízení infikovali škodlivým softwarem. Takové zařízení je ve většině případu přidáno do takzvaného botnetu a slouží generátor požadavků při DDoS útoku (viz Mirai botnet).
- **Zastaralá verze firmwaru** – Pokud se útočník snaží napadnout nějaké zařízení, jedním z prvních kroků bude nejspíše snaha zjistit verzi firmwaru. Pokud firmware nebude aktuální, útočník může na internetu zjistit známé zranitelnosti a chyby dané verze a ty následně zneužít.
- **Útok přetečení vyrovnávací paměti** – Příčinou tohoto útoku je zranitelný software běžící na zařízení a většinou se jedná o neošetřený objem vstupu, který uživatel může zadat. Následkem mohou být poškozená data, výpadek služby nebo útočníkem nastrčený škodlivý kód.

- **Instalace zadních vrátek** – Poté, co útočník získá vzdálený přístup k zařízení, téměř vždy následuje instalace zadních vrátek, které mu usnadní a umožní další přístup. Někdy se stává, že výrobci IoT zařízení ponechají diagnostické a testovací nástroje součástí firmwaru, což znatelně zvýší riziko, které napadené zařízení přináší pro celý systém.

Udržovat firmware aktuální je kritickou součástí zabezpečení celého systému. Aktualizace jsou totiž často vydávány s cílem opravit chyby a zranitelnosti stávající verze. Pokud nedojde k okamžité aktualizaci všech zařízení, útočník bude mít výrazně usnadněnou práci. V případě, že předchozí verze obsahovala nějakou bezpečnostní mezeru, tak už bude pravděpodobně dohledatelná na internetu. Informace o zranitelnostech softwaru jsou totiž publikovány až v případě, že vyšla nová verze, kde je chyba opravena. Pokud tedy útočník zjistí verzi firmwaru na zařízení, stačí mu výše zmíněná databáze CVE, nebo jí podobné, a hned bude vědět, na co se zaměřit. Smutnou realitou je ovšem to, že obecně IoT bezpečnost se nestačí vyvíjet spolu s rozvojem IoT a ve spoustě případů se na opravné aktualizace čeká v řádech měsíců a některá zařízení se jich ani nedočkají. Na druhou stranu existuje i spousta zařízení, u kterých není možné firmware ani upgradovat, ani aktualizovat.

Při existenci jednotek i desítek zařízení by proces aktualizace firmwaru bylo možné provádět manuálně. Pokud ovšem v systému existuje stovky, tisíce, nebo dokonce desetitisíce zařízení je potřeba implementovat nějakou formu automatizovaného aktualizačního mechanismu. Aktualizace takového počtu zařízení představuje výzvu samo o sobě. Zde je několik věcí, které je třeba vzít v úvahu:

- Je firmware zařízení možné upgradovat nebo aktualizovat?
- Jak budou aktualizace řešeny pro velký počet zařízení?
- Jaká metoda aktualizace firmwaru bude použita?
- Může být proces aktualizace firmwaru automatizován?
- Jsou aktualizace podepsané a ověřené, tzn. z ověřených zdrojů?

Je důrazně doporučeno udržovat databázi všech zařízení v systému a informace o jejich firmwaru. Pokud vyjde nová verze firmwaru, aktualizaci zařízení je nutné provést co nejdříve. Důležité je tedy sestavit plán pravidelné kontroly webu výrobce. Dále se doporučuje odebírat další služby zabývající se zabezpečením, jako CVE, CERT a US-CERT. Někdy se stává, že informace o zranitelnosti zařízení jsou publikovány ještě předtím, než výrobce vydá příslušné opravy. I na takovou situaci je nutné mít předem připravený plán, s cílem ulehčit rozhodování v dané chvíli, zda nechat zařízení online, anebo je odpojit.

12.3 Modely řízení přístupu

Další linií zabezpečení, které je nutné aplikovat ve všech vrstvách systému, je řízení přístupu, k ochraně síťových zdrojů, zdrojů informačního systému a informací. Základní modely řízení přístupu, které musí znát každý bezpečností analytik, jsou dle [78]:

- **Mandatorní řízení přístupu (MAC)** – Aplikuje nejpřísnější řízení přístupu tím, že informacím přiřazuje štítky úrovně zabezpečení a následně povoluje přístup pouze uživatelům s odpovídajícím bezpečnostním pověřením. Kontroluje tedy bezpečnostní úroveň subjektu, který o přístup žádá, a objektu, ke kterému chce subjekt přistoupit. Přístupová práva jsou v MAC modelu přiřazována podle jasně stanovených pravidel centrální autoritou.
- **Nezávazné řízení přístupu (DAC)** – Dovoluje uživatelům řídit přístup k datům jako jejich vlastníkovi. DAC často využívá seznamy pro řízení přístupu (ACL) nebo jiné metody k definování jaký uživatel nebo skupina mají přístup k daným informacím. Přístup je tedy v tomto modelu řízen přístupovými právy, které stanoví uživatelé vystupující jako vlastníci.
- **Řízení přístupu na základě rolí (RBAC)** – Přístup k informacím se odvíjí od role (účetní, produktový manager, projektový manager apod.) a pověření uživatele, které má přidělené v rámci organizace. RBAC je obecně tvořeno vztahy role-oprávnění, uživatel-role a role-role. Poté platí, že subjekt může mít více rolí, role může mít více subjektů, role může mít více oprávnění, oprávnění může být u více rolí, operace může být přiřazena k více

oprávněním a oprávnění může být u více operací. Obecně tedy DAC implementuje principy „minimum oprávnění“ a „oddělení oprávnění“. Centrální autoritou v modelu DAC je pověřená osoba nebo skupina osob v rámci organizace.

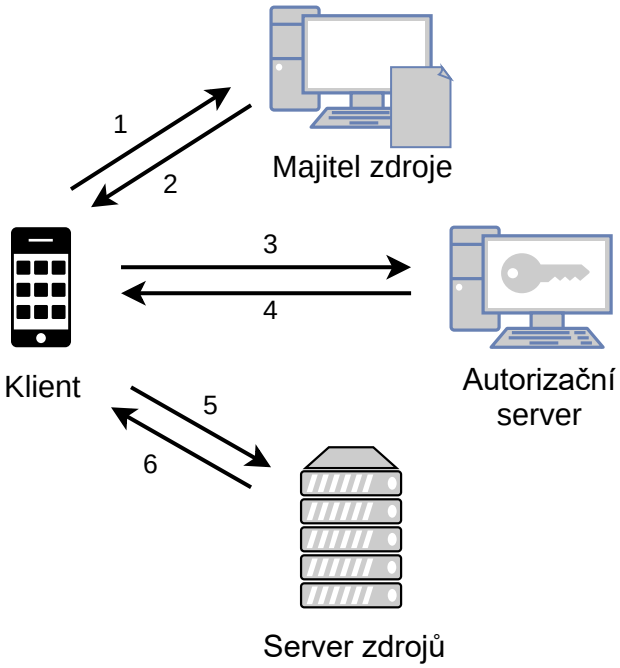
- **Řízení přístupu založené na atributech (ABAC)** – Z těchto čtyř je modelů je ABAC ten nejnovější. Přístup je řízen na základě atributů přistupovaného objektu, přistupujícího subjektu a faktoru týkajícího se prostředí, jako pozice, poloha, čas apod. Centrální autoritou v ABAC je, podobně jako v RBAC, organizací pověřené osoby.

12.3.1 OAuth 2.0 framework

Se stále rychleji rostoucím počtem IoT zařízení se řízení přístupu stává stále větším bezpečnostním problém v této oblasti. V minulosti se při výrobě těchto zařízení toto bezpečnostní hledisko nebralo příliš v potaz. To se ovšem výrazně změnilo a správa identity a přístupu (Identity and Access Management – IAM) se stal kritickou komponentou IoT řešení v organizacích. IAM je bezpečnostní přístup, který definuje, kdo má přístup, k jakým zdrojům a oprávnění, která mají, pokud je jim přístup umožněn. Autorizační framework s označením OAuth 2.0 je standardizovaným protokolem pro autentizaci a autorizaci v prostředí internetu a byl specifikován v dokumentu od skupiny IETF s identifikátorem RFC 6749 [79]. Použití tohoto přístupu pro správu řízení přístupu IoT zařízení je udělá celkově bezpečnější tím, že autorizační proces bude obstarávat server. Výhodné je to především v prostředí, kde existuje větší množství aplikací třetích stran, které vyžadují omezený přístup ke zdrojům. V opačném případě by zdroje museli sdílet pověření s třetí stranou. Tomu je možné se vyhnout tím, že dojde k delegování autorizace na zdroje pomocí těchto kroků:

1. Klient zašle požadavek na autorizaci majiteli zdroje. Požadavek může být zaslán i nepřímo na autorizační server jako na prostředníka. Majitelem je entita, která zdroj spravuje. Klient je zařízení, které žádá o přístup.
2. Majitel klientovi zašle povolení k autorizaci.

3. Klient zašle povolení k autorizaci na autorizační server, požádá o přístupový token a pokusí se autentizovat. Autorizační server je entita, která vydává a spravuje přístupové tokeny po autentizaci klientů.
4. V případě, že proces autentizace byl úspěšný, server ověří povolení k autorizaci a zašle přístupový token klientovi.
5. Klient zašle přístupový token na server zdrojů a zažádá o přístup ke zdroji. Server zdrojů je entita, která je hostitelem požadovaných zdrojů. Pozn.: Role autorizační server a server zdrojů může být reprezentován jednou entitou, ale proces se nezmění.
6. Po ověření přístupového tokenu server povolí přístup.



Obrázek 49 Ilustrace procesu autorizace OAuth 2.0
Zdroj: vlastní zpracování (podle [62])

Základní součástí jakékoliv IoT infrastruktury je, kromě autentizace, také identifikace IoT zařízení. Správa identit běžně označuje celý proces identifikace, autentizace a autorizace uživatelské přístupu k prostředkům, ale v IoT odkazuje i na identifikaci širokého rozsahu IoT zařízení a správu jejich přístupu k datům. V IoT dochází k situaci, kdy jsou velmi citlivá data přenášena mezi dvěma zařízeními, a proto je důležité mít nad přístupem k těmto datům kontrolu. S exponenciálně

rostoucím počtem zařízení roste i počet vztahů mezi nimi. Tradiční platformy pro správu identit a přístupu IAM nedokážou s tímto nárustem držet krok, jelikož ve velkých organizacích se může počet unikátních identit pohybovat v miliónech [62].

12.4 Bezpečnost dat

K zajištění důvěrnosti dat se v praxi využívají šifrovací mechanismy. Šifrování je algoritmus aplikovaný na uložená data, soubory nebo na veškerý síťový provoz, který data udělá nečitelná pro neautorizované subjekty. K dešifrování obsahu a udělat ho tak čitelným je potřeba tajný klíč. Hesla jsou často uložena v zašifrované podobě, tzn. ve formě hash (někdy označován jako otisk). K šifrování hesla dochází již během jeho přenosu. Vzniklý hash je porovnán s hashem uloženého hesla a tím dochází k ověření uživatele. Hashe jsou navrženy tak, aby bylo nemožné z nich získat původní zprávu. Oproti tomu v klasickém šifrování je potřeba, aby bylo možné původní zprávu převést zpět do čitelné podoby.

V IoT je šifrování dat nepostradatelnou součástí, jelikož může docházet k přenosu citlivých dat (zdravotní data pacienta apod.). Je velmi důležité, aby šifrovací algoritmy byly implementovány oběma směry, protože kromě naměřených důvěrných dat směrem ze zařízení, jsou důležité i řídící informace směrem k zařízení (dávkování léků apod.). Zachycení nebo manipulace s daty v jakémkoli směru, může mít velmi vážné následky. Netýká se to jen zdravotnictví, ale všech IoT systémů. Bohužel je v provozu stále velké množství starších IoT zařízení, které proces šifrování nepodporují a aby toho nebylo málo, v drtivé většině případů využívají bezdrátovou formu komunikace, která přenos dat ještě více ohrožuje. V současnosti si výrobci mohou vybrat z velkého množství standardů nabízejících alespoň nějakou úroveň ochrany, jako Zigbee, LoRa, LTE-M apod.

Běžná kryptografie je založena na tom, že odesílatel a příjemce znají a používají stejný tajný klíč. To znamená, že odesílatel zprávu zašifruje tím klíčem a příjemce ji pomocí stejného klíče dešifruje. Metoda, kde se k zašifrování i dešifrování zprávy používá stejný klíč, se označuje jako symetrická kryptografie. Odesílatel a příjemce se musejí dohodnout na jednotném tajném klíči, aniž by ho zjistil někdo další. Zajištění bezpečné zprávy klíčů, aby všechny klíče v kryptosystému zůstaly utajené, je jednou z několika výzev, kterou s sebou kryptografie přináší. S cílem

tento problém se správou klíčů vyřešit byl v roce 1976 představen koncept takzvaného veřejného klíče neboli asymetrického šifrování. Podle tohoto postupu, každý člověk obdrží pár klíčů, a to privátní a veřejný, které jsou na sobě matematicky závislé. Všechny veřejné klíče jsou publikovány a každý k nim přístup, zatímco soukromé klíče jsou utajeny a přístup k němu má pouze vlastník. Odpadá tak veškerá potřeba sdílení informací mezi odesílatelem a příjemcem, jelikož v komunikaci dochází pouze k přenosu veřejného klíče a nikdy k přenosu ani sdílení privátního klíče. Díky tomu může každý odeslat zašifrovanou zprávu použitím veřejného klíče, ale ta může být dešifrována pouze privátním klíčem zamýšleného příjemce. Asymetrická kryptografie neslouží pouze k šifrování obsahu (ochraně soukromí), ale také k ověření původu dat nebo softwaru. Implementace tohoto typu šifrování na IoT zařízení je vysoce doporučena k zajištění jejich bezpečnosti, ale v praxi to není v podstatě realizovatelné.

K podpoře rozsáhlé distribuce, identifikace a správy veřejných klíčů slouží takzvaná Infrastruktura veřejných klíčů (PKI) a Certifikační autority (CA). PKI je založeno na vysoce škálovatelném vztahu důvěry a využívá se k prokázání identity IoT zařízení. Zahrnuje hardware, software, lidi, opatření a procedury potřebné k vytváření, spravování, ukládání, distribuování a rušení digitální certifikátů. PKI certifikát obsahuje veřejný klíč vlastníka, jeho účel, autoritu, která certifikát vydala, jeho platnost a algoritmus použitý k tvorbě podpisu. Certifikáty vydané jednomu subjektu a jeho privátní klíče jsou uloženy lokálně na počítači v úložišti certifikátů (Certificate Store). Certifikační autorita je důvěryhodná třetí strana, která po ověření identity žadatele vydává certifikáty. Veškeré autoritou vydané certifikáty jsou uloženy v databázi certifikátů. Právě vydávání certifikátů obrovskému množství IoT zařízení představuje velkou výzvu, jelikož je to velmi časově náročné a téměř nemožné je spravovat. Jednou z možností je to, že organizace bude sama vydávat certifikáty svým zařízením. Existují i softwary, které umožňují automatické přiřazování certifikátů bez zásahu člověka [62].

Důvodem absence šifrovacích mechanismů je především povaha a velikost IoT zařízení a s tím související limitovaný výkon, viz. výše zmíněná omezená zařízení. Důsledkem nedostatečného výkonu a výpočetních zdrojů obecně je to, že tato zařízení zkrátka nezvládnou robustnější šifrovací algoritmy. V roce 2015

odstartoval institut NIST iniciativu za „odlehčenou“ kryptografii (lightweight cryptography). Cílem je vývoj standardizovaný kryptografický algoritmus vhodný i pro zařízení s minimálním výpočetním výkonem. Na začátku roku 2021 byly vyhlášeni finální kandidáti (celkem 10), tudíž se dá očekávat, že na přelomu roků 2021 a 2022 iniciativa svůj cíl splní. Několik „odlehčených“ algoritmů je již možné v literatuře najít. Jedná se především o algoritmy založené na symetrickém šifrování (konkrétně na Feistelově šifře), jelikož velikost a množství klíčů v asymetrickém šifrování není pro IoT vhodné. Příklady takových algoritmů jsou PRESENT, KATAN, Humming Bird a SIMON and SPECK [80].

13 Útok na IoT na komunikační vrstvě

Informace v kapitole Tvorba IoT řešení jsou převzaty z online kurzu Cisco [62], z části Chapter 4: IoT Communication Layer Attack Surface, pokud není uvedeno jinak.

Po úrovní zařízení je na řadě komunikační vrstva. Ta je zodpovědná za přenos dat mezi zařízeními, místy a aplikacemi. IoT aplikace, které fungují nejčastěji v cloudu, data ukládají, zpracovávají, analyzují a tvoří z nich výstupy a reporty. Na základě podmínek a pravidel řídí aktuátory, které následně plní určené pokyny. Takovéto smyčky se zpětnou vazbou jsou realizovatelné právě díky komunikačním kanálům. Samotný přenos dat a informací se také nejčastěji realizuje v cloudu. Zabezpečení sítě se netýká jen samotné komunikace, ale musí být uvažováno u všech elementů tvořících plochu pro útok na IoT systém. Jednou z hlavních podstat IoT systémů je především sběr dat. To znamená, že pokud dojde k jejich poškození, ať už ve formě dat v pohybu (data in motion), nebo dat v klidu (data at rest), může dojít k výpadku celého IoT systému.

V předchozí kapitole bylo zmíněno sdružení OWASP, které definovalo několik kategorií hrozeb podle úrovní systému. Pro úroveň komunikace jsou to hrozby týkající se síťových služeb zařízení a síťového provozu. První kategorie síťových služeb zařízení zahrnuje například únik dat, injekce, útok DoS, nezašifrované služby, nesprávně implementované šifrování, testovací/vývojové služby, zranitelné UDP služby apod.

Do kategorie síťového provozu patří provoz v sítích LAN, provoz z LAN do internetu, nestandardní protokoly, bezdrátová komunikace a manipulace s packety. Některé hrozby budou ještě zmíněny později v této kapitole.

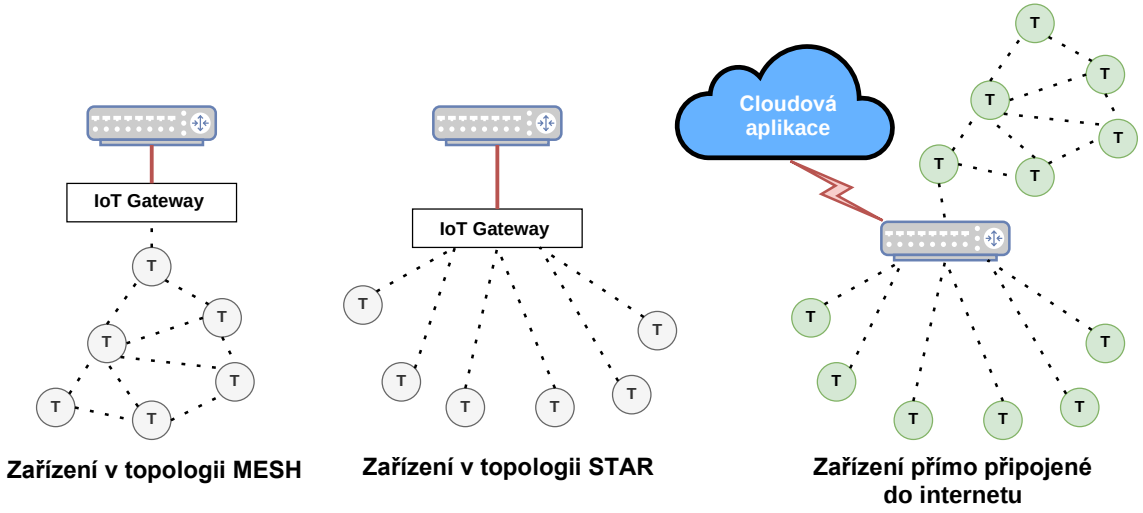
13.1 Komunikační topologie

IoT komunikace může nabývat několika forem v závislosti na aplikace a prostředí. Senzory mohou díky vlastním IPv6 adresám komunikovat s IoT aplikacemi skrz router s využitím různých LAN a WAN protokolů, nebo přímo do internetu přes služby mobilní sítě. Každá síť může používat různé protokoly pro komunikaci mezi zařízeními, mezi zařízením a gateway a mezi zařízeními a internetem/cloudem. Kromě známých označení, které již byly v dřívějších kapitolách zmíněny, jako LAN, WAN, WLAN, PAN a LPWAN, se objevují i specifičtější výrazy právě podle prostředí. Jsou to například:

- **Wireless Body Area Network (WBAN)** – Označení pro síť obsahující především zařízení, která jsou umístěna na těle nebo implantována uvnitř.
- **Wireless Personal Area Network (WPAN)** – Síť spojující zvuková zařízení, osobní fitness zařízení a chytré hodinky s mobilním telefon, který plní roli gateway (komunikuje s cloudovou aplikací). Často se k propojení používá Bluetooth.
- **Wireless Home Area Network (WHAN)** – Propojení domácích spotřebičů, alarmů, chytrých zvonků do gatewaye a následně internetu.
- **Wireless Field (Factory) Area Network (WFAN)** – Robustní síťově komponenty spojují senzory a aktuátory, které jsou náhodně rozmístěné v prostoru, převážně v průmyslových řešeních.
- **Wireless Neighborhood Area Network (WNAN)** - Síť s poměrně malou geografickou rozlohou, kterou obsluhuje router, často umístěný ve venkovním prostředí.
- A další…

Spousta IoT senzorových uzlů je tvořena omezenými zařízeními, a proto bude nutné v komunikaci použít gateway pro překlad přenosu WSN sítě na provoz protokolu IP pro komunikaci na tradičních datových sítích. WSN sítě obsahují stovky, tisíce i desetitisíce senzorů, které mají k dispozici velmi limitovaný výpočetní výkon a napájeny pouze z baterie. Kvůli tomu mohou pro komunikaci s ostatními uzly používat pouze rádiové vysílače s krátkým dosahem. I proto jsou ve WSN sítích použity specifické protokoly umožňující datům cestovat od uzlu k uzlu (multi-hop) do doby, dokud nedorazí ke gateway (více o WSN sítích v části 8.6.1 Bezdrátové senzorové sítě). Právě tyto komunikační kanály a protokoly tvoří plochu pro potenciální útok.

Existuje několik základních scénářů, podle které IoT bezdrátové protokoly operují. První dva scénáře se řídí, z počítačových sítí dobře známými, topologiemi Mesh a Star. Nutné podotknout, že se to týká především způsobu, jakým spolu komunikují IoT zařízení („Věci“). V topologii Mesh zařízení přeposílají data jiným zařízením za účelem dosažení gateway, která pro ně může být jinak mimo dosah. To umožňuje pokrytí mnohem větší oblasti, než kdyby s gateway museli komunikovat přímo. To je případ právě topologie Star (nebo hub-and-spoke), kde zařízení přeposílají data přímo na gateway a to znamená, že musejí být rozmístěna v jejím dosahu. V případě pokrytí větší oblasti by bylo nutné nasazení dodatečných gateway zařízení. Role gateway je v obou topologií zcela totožná. Převádí provoz z WSN lokálních sítí na Wi-Fi nebo Ethernet a zapouzdřuje data do IP packetů, která jsou tak připravena na přenos do internetu. Speciálním scénářem je situace, kdy nedochází k použití gateway a zařízení tak komunikují přímo s cloudovou aplikací. Každá „Věc“ disponuje vlastní IPv6 adresou a protokoly pro zasílání zpráv.



Obrázek 50 Scénáře komunikace v IoT
Zdroj: vlastní zpracování (podle [62])

13.2 Standard IEEE 802.15.4

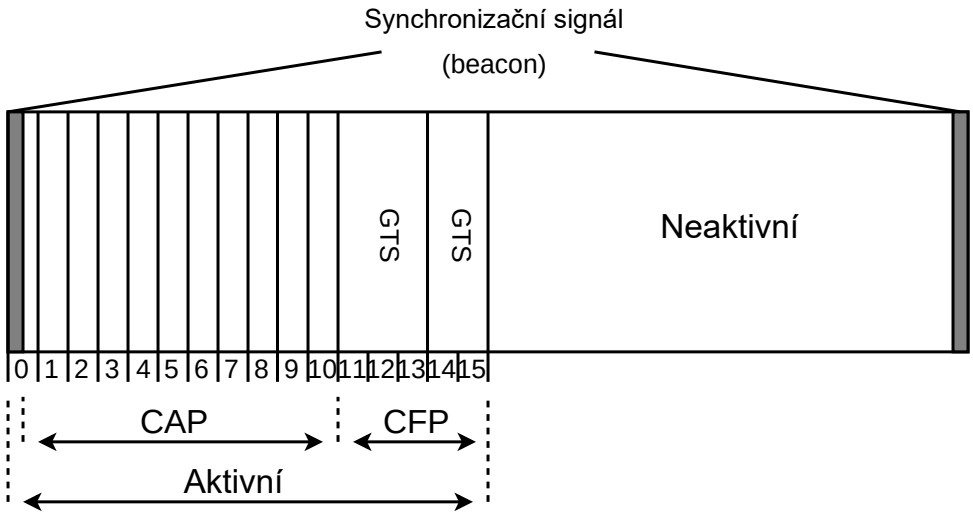
Omezený výkon a provozní schopnosti spousty IoT zařízení jsou důvodem vzniku nových protokolů umožňujících bezdrátovou komunikaci mezi nimi. Základním komunikačním modelem v IoT je M2M a většina dat je do sítě nahrávána v pravidelných interval, kde objem nahrávaných dat je velmi malý. Jedním z hlavních standardů, který byl vyvinut specificky pro účely takových zařízení je IEEE 802.15.4. I když byl původně určen hlavně pro sítě PAN, velmi rychle se stal oblíbeným i v jiných implementacích, včetně WSN. Klíčovými vlastnostmi standardu je práce v reálném čase, zamezení kolizím při komunikaci díky metodě CSMA/CA, integrovaná podpora zabezpečení komunikace, funkce správy napájení (rychlost/kvalita připojení), možnost přenášení dat ve třech frekvenčních pásmech (868/915/2450 MHz).

Standard definuje dvě síťové vrstvy, a to fyzickou (PHY) a úroveň správy přístupu (Medium Access Control – MAC). Nad úrovní MAC se nacházejí další vrstvy, jako síťová a aplikační, které už nejsou standardem 802.15.4 definovány, jelikož existuje několik možných implementací. Proto je označován za standard nižších vrstev. Vyšší vrstvy jsou poté doménou konkrétních protokolů, které jsou na tomto standardu postaveny, jako Zigbee, 6LoWPAN, WirelessHART apod. Fyzická vrstva zodpovídá za aktivaci/deaktivaci rádiového vysílače, přenos a příjem packetů po fyzickém médiu a výběr kanálu spolu s funkcemi pro správu energie a signálu.

Kromě toho zužitkovává i metodu pro zamezení kolizí CSMA/CA, indikátor síly přijatého signálu (RSSI) a indikátor kvality spojení (LQI), který určuje cenu cesty především v multi-hop komunikaci. Úroveň MAC umožňuje komunikaci mezi dvěma zařízeními pouze v jedné síti (v počítačových sítích označováno jako Layer 2 komunikace). Dále je zodpovědná za převod dat do rámců a kontrolu přijatých RX rámců, adresování zařízení, správu přístupu ke komunikačním kanálům a asociaci/disociaci zařízení.

Vrstva MAC také definuje dva typy zařízení (zmíněné již v části 7.5.3 Zigbee), a to zařízení s plnou funkcionalitou FFD a zařízení se sníženou funkcionalitou (RFD). FFD zařízení mohou v síti zastávat roli PAN koordinátora (v síti může existovat pouze jeden) nebo zařízen a mohou komunikovat s ostatními FFD i RFD zařízeními. RFD na druhou stranu mohou komunikovat pouze s FFD zařízeními a nikdy nemohou působit v roli koordinátora. Jsou určené pro velmi jednoduché operace, které spotřebují minimum výkonu a nedochází při nich k přenosu velkých objemů dat. PAN koordinátor může fungovat ve dvou módech: beacon-enabled mode a non-beacon-enabled mode. Beacon-enabled mód nabízí maximální úsporu energie, a proto se používá v situaci, kdy PAN koordinátor je napájen baterií. V módu non-beacon poté operují koordinátory napájené přímo ze sítě a úspora energie nehraje tak důležitou roli.

Úspora energie v módu beacon-enabled je dosažena především efektivní komunikací mezi zařízeními, a to díky takzvané struktuře superframe (viz Obrázek 51), což je vysílací formát definovaný PAN koordinátorem. Koordinátor v pravidelných intervalech všem zařízením vysílá synchronizační signály (beacon), kterými superframe začíná a končí. Dále se skládá z aktivní doby, kde dochází ke komunikaci mezi uzly s podporou metody CSMA/CA a neaktivní doby, kde zařízení přechází do spánku za účelem úspory energie. Aktivní část se dále dělí na contention access period (CAP) a contention free period (CFP). V CAP mohou komunikovat všechna zařízení a soupeří mezi sebou o přístup ke komunikačnímu kanálu. V CFP koordinátor přiřazuje zařízením exkluzivní časové sloty označované jako guaranteed time slot (GTS), během kterých může vysílat pouze dané zařízení [81].

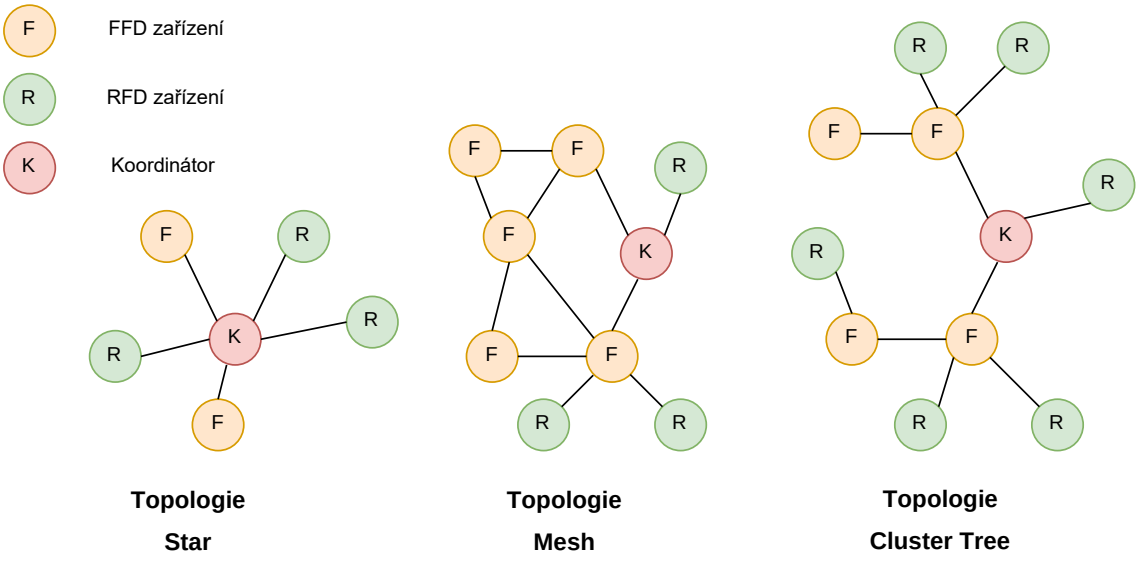


Obrázek 51 Struktura superframe
Zdroj: vlastní zpracování (podle [81])

Ze síťových topologií jsou standardem IEEE 802.15.4 podporovány tři: Star, Mesh, Cluster Tree (viz Obrázek 52). V topologii Star komunikace probíhá pouze mezi zařízeními a jediným centrálním koordinátorem, který je často napájen přímo ze sítě, kdežto ostatní zařízení jsou napájeny z baterie. Nejvíce benefitů tato topologie přináší aplikacím v automatizaci domácnosti, počítačovým periferiím, hračkám a video hrám. Nevýhodou je, že zařízení musejí být v dosahu koordinátora. V rozsáhle síti tak bude potřeba daleko víc koordinátorů než u ostatních topologií. Potom, co je zařízení FFD poprvé aktivováno, může vytvořit vlastní síť a stát se PAN koordinátorem. Každý koordinátor má identifikační číslo, které je unikátní v rádiovém dosahu. To umožňuje každé hvězdě pracovat nezávisle na ostatních sítích.

I v topologii Mesh se nachází pouze jeden PAN koordinátor. Rozdílem je ale to, že zařízení mohou komunikovat i mezi sebou. Největší využití nachází v průmyslových řídících a monitorovacích systémech, v sítích WSN a v aplikacích pro sledování aktiv a inventáře. Omezená RFD zařízení nemají schopnost předávat zprávy, takže musí být schopni se připojit k FFD nebo koordinátorovi.

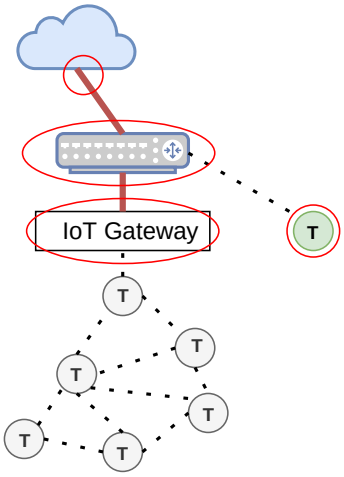
Cluster tree je speciálním případem implementace topologie Mesh, ve které převažují FFD zařízení tvořící „strom“ sítě. RFD zařízení se do sítě mohou připojit jako listy (leaf node) na konci větve. Jakékoliv FFD zařízení může být koordinátorem po své okolí, ale pouze jedno může být PAN koordinátorem sítě [62].



Obrázek 52 Topologie standardu IEEE 802.15.4
Zdroj: vlastní zpracování (podle [62])

13.3 Slabiny IoT na komunikační vrstvě

Spousta omezených zařízení nepodporuje plný balíček TCP/IP protokolů a rolí gateway zařízení je tento nedostatek kompenzovat. Ty zařízením poskytují služby spojené s IP a umožňují tak přenos senzorových dat nejčastěji do cloudové aplikace. Jenže právě přenos dat skrz IP sítě představuje vážné obavy. Plocha potenciálního útoku (attack surface) obsahuje v IoT čtyři základní elementy, a to síť senzorů, IoT gateway, firemní síť jako celek a horní spojení do internetu (viz Obrázek 53)



Obrázek 53 Attack surface IoT komunikační vrstvy
Zdroj: vlastní zpracování (podle [62])

13.3.1 Běžné zranitelnosti IP

Útoků cílených specificky na IP existuje hned několik. Mezi běžné útoky patří DoS, DDoS, útoky na protokol ICMP, Spoofing adres, man-in-the-middle (MITM) a převzetí kontroly nad relací (session hijacking).

Útok Denial of Service (DoS)

Útok nazvaný jako odmítnutí služby (Denial of Service) je typ kybernetického útoku, jehož cílem je znepřístupnit cílový počítač/server a jeho služby pro legitimní uživatele. Principem útoku DoS je zahlcení cíle tak velkým množstvím požadavků, až nebude moci zpracovávat normální uživatelský provoz a služby se tak stanou nedostupnými. Charakteristikou DoS je využití jediného počítače k provedení útoku. Tento útok se typicky dělí do dvou kategorií, konkrétně přetečení bufferu (buffer overflow) a záplavy (flood attacks) [82].

Buffer je oblast fyzické paměti, typicky součástí paměti RAM, která se používá k dočasnému ukládání dat při jejich přesunu. Příkladem jednoduchého využití bufferu z prostředí online služeb je načítání videí (zřejmě každý zná slovo „buffering“). Jednoduše se jedná o metodu streamování, která má minimalizovat přerušení při přehrávání, kdy dochází ke stažení určitého procenta videa do bufferu a z něj následně dochází k jeho přehrávání. Buffery jsou navrženy tak, aby drželi určité množství dat a pokud dojde k jeho přetečení, dochází k nahrazování a mazání aktuálně uložených dat. Principem útoku buffer overflow je tedy předávání velkého množství informací do doby, dokud nedojde k jeho přetečení, a to způsobí přepsaní dat ve vedlejších kontejnerech mimo oblast přidělenou bufferu. Útočník toho může využít k nahrání škodlivého kódu do paměti počítače nebo jednoduše jen k přepsání důležitých dat. V poslední řadě může dojít až ke spotřebování veškerých zdrojů, jako místo na HDD, paměti RAM a času procesoru, a to může vést ke zpomalení systému, pádu a dalšímu chování [83].

Útoky záplavou spočívají v tom, že útočník se snaží nasytit cílový server obrovským množstvím packetů, čímž dojde k znepřístupnění služby a někdy i k jejímu pádu. K tomu, aby tyto útoky byly úspěšné, útočník musí mít k dispozici větší šířku pásma (bandwidth) než cíl. Příkladem DoS útoku s využitím slabin ICMP protokolu je ICMP (ping) flood, který je popsán níže.

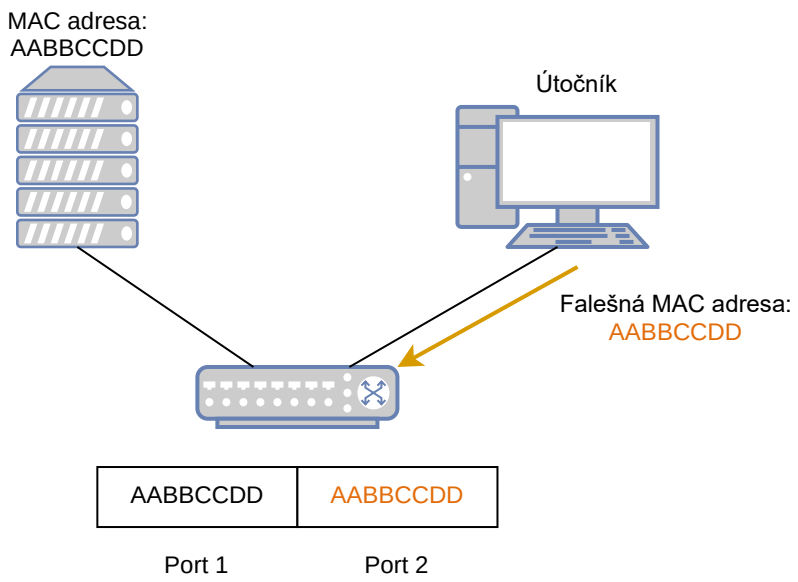
Útok Distributed Denial of Service (DDoS)

Cíl útoku distribuovaného odmítnutí služby je zcela stejný jako v předchozím případě. Jediným rozdílem je to, že DDoS k provedení útoku používá více zařízení najednou (stovky, tisíce i desetitisíce), které jsou napadeny útočníkem a může je ovládat na dálku. Takováto infikovaná+ zařízení se označují jako boti (nebo zombie) a síť těchto botů se označuje jako botnet. Obrana proti DoS útoku je v současnosti velmi jednoduchá, jelikož je velmi snadné rozpoznat a vyfiltrovat škodlivý provoz a zablokovat jeho zdroj. U útoku DDoS je filtrování provozu velmi obtížné, jelikož každé ze zařízení v botnetu je zároveň legitimním zařízením na internetu. Typů DDoS útoků existuje hned několik, například SYN flood, HTTP flood (týká se aplikační vrstvy a bude zmíněn až ve 14. kapitole), útok zesílením a odrazem a další [84].

Útok Address Spoofing

Spoofing IP adres je útok, kdy útočník vytvoří packety s falešnou IP adresou, díky kterým může schovat svoji identitu nebo působit jako jiný legitimní uživatel a získat tak přístup k zabezpečeným datům nebo obejít bezpečnostní opatření. Tento útok může být veden dvěma způsoby, a to non-blind a blind. Non-blind označuje situaci, kdy útočník vidí komunikaci mezi hostem a cílem. Spoofing se zde používá k průzkumu odpovědí od oběti, za účelem zisku informací o stavu firewallu, predikce TCP sekvenčního čísla nebo převzetí již autorizované relace. U blind spoofing útočník nevidí komunikaci mezi hostem a cílem a tato metoda se používá především pro provádění DoS útoků.

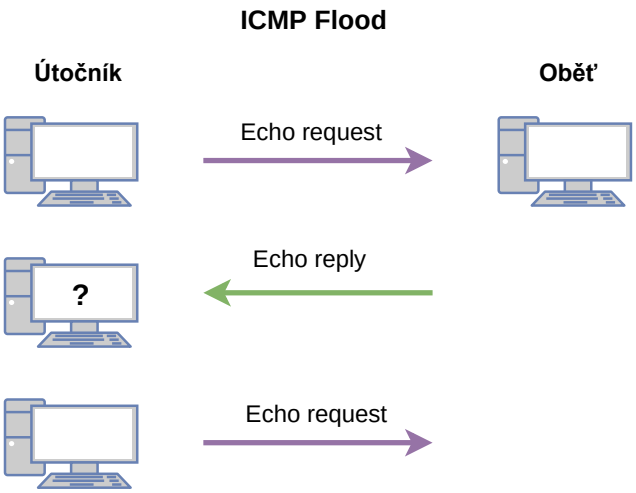
Útočník, kromě adresy IP, může zfalšovat i MAC adresu. Tento útok se používá v případě, že má útočník přístup do vnitřní sítě. Ten si upraví MAC adresu na vlastním zařízení tak, aby se shodovala s adresou cíle a zašle datový rámec do sítě. Switch prozkoumá zdrojovou MAC adresu rámce a přepíše záznam v tabulce, kde byla tato adresa přidělena portu 1, na port 2 (viz Obrázek 54). Nyní veškerá komunikace směřována na tuto MAC adresu půjde skrz port 2 směrem k útočníkovi.



Obrázek 54 Spoofing MAC adresy
Zdroj: vlastní zpracování (podle [62])

Útok ICMP Flood

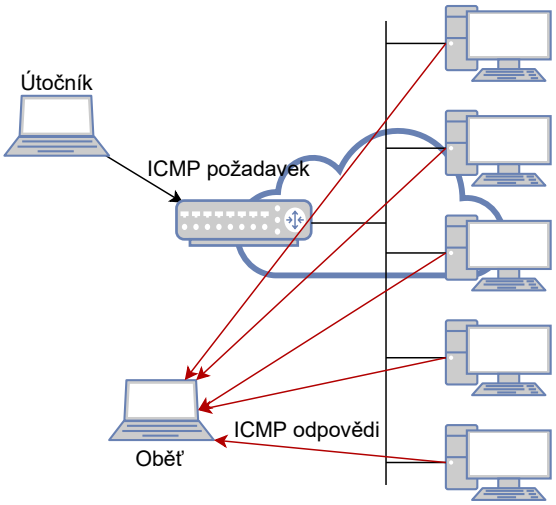
Internet Control Message Protocol (ICMP) byl vyvinut k přenosu diagnostických zpráv a chybových hlášení v případě, že trasy, hostitelé nebo porty nejsou k dispozici. Zprávy ICMP jsou generovány zařízením v případě chyby nebo výpadku sítě. Uživatelsky generovaný požadavek (echo request), k ověření dostupnosti destinace, je možné zaslat pomocí příkazu ping (popř. traceroute). Útočníci zneužívají protokol ICMP ke shromažďování informací s cílem zmapovat topologii sítě, určit aktivní (dosažitelné) hostitele, identifikovat operační systém a zjistit stav brány firewall. Často ho využívají i ke generování DoS útoky jako je ICMP (nebo ping) flood, kdy útočník zasílá velké množství echo request s falešnou adresou na cílové zařízení nebo server, které na ně následně musí odpovědět formou echo reply packetu (viz Obrázek 55). Další možností je využití botnetů, které dokážou generovat daleko větší provoz a zároveň není potřeba falšovat jejich IP. V tomto případě by se již jednalo o distribuovaný útok DDoS. Nejlepší obranou proti tomuto útoku je jednoduše vypnutí ICMP funkcionality, čímž dojde k eliminaci jak echo request, tak i echo reply packetů [85].



Obrázek 55 ICMP Flood útok
Zdroj: vlastní zpracování (podle [62])

Útoky zesílením a odrazem

Útoky typu zesílení a odraz (amplification and reflection) se často používají ke generování DoS/DDoS útoku. K jeho provedení je možné využít několik různý protokolů, jako DNS, NTP a zmíněné ICMP. Útok využívající ICMP protokol se v tomto případě označuje jako smurf attack. V první fázi, fáze zesílení, útočník zašle echo request, obsahující IP adresu oběti, velkému počtu hostitelů. Druhou fází je odraz, kde všichni oslovení hostitelé odpovídají zprávou echo reply na uvedenou adresu oběti (viz Obrázek 56). Dochází tedy k tvorbě DDoS útoku a k přehlcení cílového zařízení [62, 84].



Obrázek 56 Smurf Attack
Zdroj: vlastní zpracování (podle [62])

13.3.2 Slabiny protokolu TCP

Transmission Control Protocol (TCP) je protokolem transportní vrstvy a slouží především ke komunikaci v heterogenních sítích a je nezávislý na přenosovém médiu. Hlavní předností TCP je jeho spolehlivost. Na rozdíl od UDP, o kterém bude řeč později, vyžaduje potvrzení o přijetí zprávy od příjemce. V případě, že potvrzení nedorazí v určitém časovém intervalu, zprávu odešle znovu. Na druhou stranu to může komunikaci zdržet, což je v některých situacích nežádoucí. Dalším příkladem jeho spolehlivosti je to, že s přenosem dat musejí souhlasit obě strany. To je ošetřeno pomocí takzvaného „three-way handshake“ mechanismu. Příkladem protokolů aplikační vrstvy, které využívají spolehlivost TCP protokolu jsou HTTP, SSL/TLS, FTP apod.

Obecně protokoly transportní vrstvy, to znamená TCP a UDP, umožňují sledování více konverzací najednou a jejich spojení se správnou aplikací díky adresaci portů. Spojení mezi počítači je tak definováno cílovou a zdrojovou IP adresou a cílovým číslem portu. Prvních 1024 portů je pevně dáno a nesmějí být změněny. Každý port reprezentuje běžně používané a známé aplikace, např.: 21 FTP, 23 Telnet, 80 HTTP apod. Telnet je ovšem výjimkou a může mu být přidělen jakýkoliv otevřený port. Tato služba ovšem není bezpečná, a proto se nedoporučuje ji nechávat zapnutou na IoT zařízeních.

TCP protokol je zranitelný především proti skenování portů. Útočník může provést sken portů cílového zařízení a určit, jaké služby nabízí. Nástroje na skenování portů dokážou poskytnou velmi detailní informace o dostupných službách, které pak útočník může zneužít.

Útok TCP SYN Flood

Jedním z útoků, který zneužívá TCP protokol k provedení útoku, je TCP SYN Flood. Konkrétně se jedná o zneužití mechanismu „three-way handshake“ používaného pro navázání spojení, kde dochází k výměně tří paketů. Jako první klient zašle žádost o navázání spojení (SYN paket), hostitel zasílá zpět potvrzení (SYN-ACK paket) a čeká na potvrzení od klienta (ACK paket). SYN flood útok tedy spočívá v tom, že útočník neustále zasílá SYN pakety s falešnou IP adresou na cílový server. Ten zasílá odpovědi SYN-ACK na tyto falešné IP adresy a čeká na potvrzení ACK od klienta. To

ovšem nikdy nedorazí a zanedlouho tak nastane situace, že server bude zahlcen otevřenými TCP požadavky a začne tak odmítat žádosti od legitimních uživatelů. Jedná se tedy opět o typ DoS útoku [62, 84].

13.3.3 Slabiny protokolu UDP

Druhým protokolem transportní vrstvy je User Datagram Protocol (UDP), který si na rozdíl od TCP zakládá na jednoduchosti a rychlosti. UDP není orientován na spojení a nenabízí žádné funkce, jako garance doručení, opětovné zasílání zpráv apod. Příkladem protokolů využívajících UDP jsou DNS, TFTP, NFS, SNMP apod. Často se využívá i u aplikací na streamování videa nebo VoIP. To ovšem neznamená, že je UDP vždy nespolehlivé a zanedbatelné, jen je třeba funkce, které postrádá, aplikovat na jiných vrstvách. Je to velmi žádoucí protokol pro aplikace, které si zakládají na jednoduchých transakcích žádost-odpověď. Například pokud by služba DHCP využívala protokol TCP, došlo by ke vzniku zcela zbytečného provozu.

UDP ve výchozím stavu nepodporuje žádnou formu šifrování, ale je možné ho dodatečně implementovat. Toho může útočník využít ke sledování provozu, a dokonce ho pozměnit a teprve poté ho zaslat do destinace. Při změně obsahu zprávy dojde ke změně 16-bit kontrolního součtu, ale útočník může jednoduše vytvořit nový odpovídající změněnému obsahu a umístit ho do hlavičky. Cílové zařízení zkontroluje, zda součet odpovídá obsahu, ale nepozná, že zpráva byla změněna.

Útok UDP Flood

Nejběžnějším útokem zneužívajícím protokol UDP je UDP Flood. Opět se jedná o typ DoS útoku, kdy dochází k zasílání velkého množství UDP paketů na cílový server s cílem ho zahltit. Konkrétně se jedná o zneužití procesu, kterým server odpovídá na zaslané žádosti. Server při obdržení UDP paketu zkontroluje, zda některé programy aktuálně naslouchají na specifikovaném portu. Pokud na daném portu neběží žádná aplikace, server odpoví zprávou unreachable (ICMP echo reply). Útočníci často k tomuto útoku využívají nástroje, jako UDP Unicorn nebo Low Orbit Ion Cannon, které nejprve, buď pomocí skenování, nebo jednoduše pomocí zasílání žádostí, určí, které porty jsou na serveru zavřené. Následně začne neustále zasílat UDP pakety s falešnou adresou a s čísly těchto portů. Server se bude snažit na všechny žádosti

odpovědět formou unreachable. Kombinace velkého množství uzavřených portů a neustálého přívalu nových žádostí způsobí, že dojde k jeho zahlcení a nebude moct přijímat legitimní žádosti [86].

14 Útok na IoT na aplikační vrstvě

Informace v kapitole Tvorba IoT řešení jsou převzaty z online kurzu Cisco [62], z části Chapter 5: IoT Application Layer Attack Surface, pokud není uvedeno jinak.

Aplikační vrstva a její protokoly jsou základem celého IoT ekosystému, jelikož jsou veškerých interakcí mezi IoT zařízeními a mezi zařízeními a cloudovou/edge infrastrukturou. Typické funkce implementované protokoly na této vrstvě jsou zasílání zpráv (messaging) a objevování služeb (service discovery). Messaging protokoly, jako MQTT, CoAP, XMPP a AMQP, zajišťují sdílení a výměnu dat a service discovery protokoly, jako mDNS a SSDP, slouží pro detekci zařízení a služeb, které nabízejí. Protokoly se liší v mnoha aspektech, jako je architektura, model interakce a v transportním protokolu a výsledná volba tedy záleží na povaze IoT systému a jeho požadavcích. MQTT a CoAP jsou vhodnějšími kandidáty pro služby vyžadující sběr dat v omezeném prostředí. Oproti tomu AMQP, XMPP se lépe zaměřují na specifické systémové požadavky, jako zasílání zpráv v podniku, chatování, výměna dat v reálném čase a detekce přítomnosti v online prostředí.

Další věcí, ve které se protokoly významně liší, je jejich úroveň podporovaného zabezpečení. Služby týkající se zabezpečení jsou obecně brány jako volitelné a výrobci je tak často zanedbávají. Navíc šifrování end-to-end komunikace může být finančně nákladné, aby ho bylo možné implementovat na IoT zařízení [87].

Sdružení OWASP i pro tuto vrstvu definovalo několik běžných hrozeb týkajících se softwaru (firmware, operační systém, aplikace) na IoT zařízeních. Jsou to například [62]:

- **Výčet přihlašovacích jmen** – Útočník dokáže zjistit validní uživatelské jména pouhou interakcí s ověřovacím mechanismem.
- **Slabá hesla** – Útočník k přístupu do aplikace zneužije výchozí hesla, která nebyla změněna nebo běžná hesla.

- **Uzamčení účtu** – Příliš mnoho pokusů o autentizaci útočníkem může vést k uzamčení účtu.
- **Absence vícefázového autentizace** – Existence pouze jedné formy autentizace útočníkovi velmi usnadňuje práci.
- **Nezabezpečené komponenty třetích stran** – S objevováním nových zranitelností v aplikacích roste i jejich riziko zneužití, a proto je důležité udržovat aplikace aktuální.

14.1 Bezpečnost aplikací

Předtím, než se zařízení začala běžně připojovat k internetu, byla relativně chráněna od útoků, jelikož k nim útočníci museli mít fyzický přístup. Nyní mohou být detekovány na síti a napadeny na dálku. Populárními „lokálními" útoky jsou například nahrazení firmwaru, klonování zařízení a následné nahrazení legitimního zařízení, DoS a extrakce bezpečnostních parametrů (ověřovací informace, bezpečnostní klíče apod.).

Takové útoky jsou v současnosti ale velmi vzácné, jelikož útočník může k napadení zařízení využít komunikační protokoly a jejíž zranitelnosti. Prvním příkladem útoku je Man-in-the-middle (MITM), kde se útočník dostane do komunikace mezi dvěma zařízeními a může sbírat nebo modifikovat přenášená data předtím, než dojdou do cíle. Velmi podobným útokem je odposlouchávání. Útočník pasivně sleduje komunikaci v síti se snahou získat citlivé informace (identifikační čísla, citlivá data, osobní informace atd.). Oproti tomu útok označovaný jako SQL injekce se zaměřuje na zranitelnosti SQL aplikací. Důsledkem může být krádež, manipulace, neoprávněný přístup do souborového systému atd. Posledním běžným útokem je takzvaný routing attack, kdy útočník do sítě umístí vlastní směrovací zařízení, které bude veškeré packety zasílat do jím určeného cíle. Zde může vypustit buď vybrané packety (selective forwarding) nebo všechny obdržené (sinkhole).

Stejnou hrozbu představují i mobilní zařízení, a to především s nárustem využívání mobilní aplikací v posledních několika letech. V současnosti je možné nalézt mobilní aplikaci téměř na cokoliv. Používají se pro internetové bankovnictví, platby v obchodech, zasílání emailů a zpráv, správu a ukládání osobních souborů (fotek, videí, kontaktů, důležitých dokumentů apod.). A i přesto, že mobilní operační

systémy, jako Android a iOS jsou relativně bezpečné, jsou to právě tyto aplikace stažené z internetu, které zařízení, a s ním data, ohrožují. Hrozbu představují zejména nezabezpečené komunikační technologie a kanály, například při používání veřejných WiFi připojení, nezabezpečené úložiště dat, tj. aplikace mají přístup k paměti zařízení i přesto, že ho nepotřebují ke svému fungování, nezabezpečená autentizace (tj. správa relací), nesprávné využití platforem, jako TouchID, Keychain a Android Intents, a nedostatečné šifrování.

14.2 Bezpečnost cloudových a webových aplikací

Zranitelnosti webových a cloudových aplikací jsou jedním z hlavních zdrojů úniků dat. Každým dnem množství dat uložené v cloudu exponenciálně roste. Zabezpečení těchto aplikací, tak paradoxně vyžaduje použití jiných aplikací, které dokážou identifikovat a opravit případné zranitelnosti. Jsou to například webové skenery, aplikace pro monitorování webu a aktivity na něm a ochrana proti útokům na aplikační vrstvě v reálném čase. Pro webové a cloudové aplikace existují více specifické hrozby, jako útok injekcí, Cross-site scripting (XSS), externí XML entity (XXE), expozice citlivých dat a nesprávně nakonfigurovaná pravidla pro přístup.

14.2.1 Útok Cross-site scripting (XSS)

Cross-site scripting je typ útoku injekcí, při kterém dochází k umístění škodlivých scriptů na jinak bezpečné webové stránky. Útočník tedy využívá webovou aplikací k zaslání škodlivého kódu, často ve formě scriptu, který se spouští na straně prohlížeče, jinému koncovému uživateli. XSS může být použito na široké škále webů, které používají jakoukoli formu uživatelského vstupu pro generování dynamického výstupu (veřejná fóra, webové nástěnky apod.), aniž by došlo k jeho validaci a ověření. Prohlížeč nemá nejmenší šanci poznat, že se jedná o škodlivý script, jelikož pochází z věrohodného zdroje, a spustí ho. Ten následně může přistupovat k souborům cookies, tokenům relací a dalším citlivým datům uložených prohlížečem. Těmito scripty je dokonce možné přepsat HTML zobrazené u uživatele [88].

XSS je možné rozdělit na dva základní typy, a to stored XSS a reflected XSS. Stored XSS označuje útoky, kdy je script permanentně uložen na cílovém serveru (v databázi, ve fóru, v komentáři apod.). Ten se poté spustí v prohlížeči oběti v momentě, kdy si obsah zobrazí. Reflected XSS je útok, kdy je škodlivý script „odražen“ od webového serveru, například formou chybového hlášení, výsledku vyhledávání nebo jakoukoli další formou odpovědi. Typicky je oběti doručen prostřednictvím emailu nebo webové stránky. Ve většině případu se jedná o, na první pohled neškodný a legitimní, URL odkaz. Po kliknutí na odkaz se na server pošle požadavek, server ho zpracuje a výsledek zašle zpět oběti. Prohlížeč tento výsledek zobrazí a interpretuje i spolu se škodlivým kódem [89].

Základní obranou proti XSS je důkladná kontrola a čištění uživatelských vstupů předtím, než budou použity pro generování obsahu na stránce. Obecně existuje spousta frameworků, které slibují, že tento problém vyřeší, ale pouze za předpokladu jejich správné implementace. Dalším opatřením jsou pravidla s označením Content Security Policy (CSP). Pomocí nich lze definovat, jak má stránka nakládat s různými typy obsahu. Konkrétní politika obsahu může vypadat například tak, že externí scripty budou spouštěny pouze z vyjmenovaných domén, obrázky a videa z jakékoli domény a zbytek pouze z vlastní domény [89].

14.2.2 Útok SQL Injection

Dalším běžným útokem je SQL Injection, který má na svědomí velké množství závažných úniků dat. Útok spočívá ve vložení SQL dotazu skrz neošetřený uživatelský vstup (formuláře, vyhledávání apod.). V případě, že je injekce úspěšná, útočník může číst data z databáze, upravit záznamy, mazat záznamy, spouštět administrativní operace a další. I přesto, že je tento typ útoku známý již řadu let, stále patří mezi nejrozšířenější útoky a stále mu patří místo v Top 10 hrozbách společnosti OWASP [90]. Obrana proti SQL injekci je v celku jednoduchá a spočívá v tom, že hodnoty a samotný dotaz bude předán odděleně databázi, která si výsledný dotaz bezpečně složí sama. Místo proměnných se v dotazu použijí zástupné znaky a proměnné budou předány později v poli. Takto zpracované dotazy se pak označují jako parametrizované dotazy (Prepared statements) [91].

14.3 IoT protokoly pro zasílání zpráv

Messaging Protokoly pro zasílání zpráv (Messaging protocols) v IoT definují funkce a pravidla pro přenos zpráv mezi dvěma zařízeními. Pro komunikaci s webovými aplikacemi se typicky používá HTTP (popř. HTTPS) spolu s REST API. Využívání HTTP je ovšem spojeno s vysokou režií, a tak protokol není příliš vhodný pro IoT zařízení hned z několika důvodů. Prvním z nich je, že omezená zařízení ve většině případů nebudou moci nainstalovat HTTP služby. Dalším problém je absence publish-subscribe model v HTTP a s tím související nemožnost uchovávat nebo udržovat zprávy. Posledním zásadním nedostatkem je chybějící mechanismus, který by uživatele upozornil o tom, že zařízení již není dostupné.

Tato část je tedy zaměřena hlavně na tři, v IoT rozšířené, protokoly pro zasílání zpráv, a to MQTT, CoAP a XMPP. Ty již byly stručně představeny v předchozích kapitolách (viz kapitola 7.6 Komunikační standardy vyšších vrstev), a proto zde bude kladen důraz především na hrozby, které s sebou přinášejí.

14.3.1 Zabezpečení MQTT

Prvním protokolem je MQTT (Message Queueing Telemetry Transport) od společnosti IBM. Je založen na modelu publish-subscribe (client-server) a server, ke kterému se klienti připojují se nazývá broker.

MQTT podporuje několik různých ověřovacích a šifrovacích mechanismů, ale ani to není dostatečné pro ochranu zařízení využívajících MQTT, a především brokeru. Na konferenci DEF CON, zaměřenou na počítačovou bezpečnost, bylo demonstrováno, že právě broker, konkrétně jeho konfigurace, je, spolu se zranitelnostmi softwaru, původcem spousty bezpečnostních rizik MQTT. Ty je možné rozdělit do několika kategorií:

- **Autentizace** – MQTT broker nedostatečně kontroluje identitu publikujících (publisher) a odebírajících (subsriber) zařízení a neblokuje opakující se pokusy o autentizaci. Útočník by mohl získat kontrolu nad zařízením nebo zahltit broker.
- **Autorizace** – Broker nesprávně nastavuje oprávnění pro publikování a odběr. Útočník by mohl získat kontrolu nad daty a funkcemi MQTT zařízení.

- **Doručení zpráv** – Publisher zasílá zprávy, které nemohou být doručeny kvůli nedostupným nebo chybějícím subscriberům. Může zapříčinit výrazný pokles výkonu brokeru.
- **Ověřování zpráv** – Publisher zasílá zprávy obsahující nepovolené znaky, které mohou být nesprávně interpretovány. Riziko hned několika způsobů zneužití (např. výše zmíněné útoky injekcí a XSS).
- **Šifrování zpráv** – Data přenášená při komunikaci mezi server a klientem jsou v otevřené textové podobě (plaintext). Roste tak riziko MITM útoku.

V tuto chvíli (28.7.2021) je v databázi NVD celkem 55 CVE záznamů odpovídajících protokolu MQTT. Zajímavostí je to, že spousta z nich spadá do kategorie nesprávného ověřování zpráv.

U MQTT musejí být bezpečnostní opatření založená na možnostech klientů a brokerů. Ověřování klientů u brokera je dobrým začátkem. Ověření klienta může proběhnout třemi způsoby. Prvním z nich je pomocí klientského ID, kterým se klient může zapisovat na témata k odběru. Druhým způsobem je, že klient zašle brokerovi uživatelské jméno a heslo, aby mohl navázat spojení. Zde ovšem platí to, že tyto informace jsou přenášeny v plaintextu. Poslední možností jsou certifikáty přiřazené každému klientovi. Toto řešení sice nabízí dobrou úroveň zabezpečení, ale je velmi nepraktické v situaci, kdy existuje velký počet klientů.

Druhým krokem je implementace šifrovacích algoritmů. Pokud je vyžadována vysoká úroveň zabezpečení, šifrována musí být i samotná komunikace. Jednou z možností je technologie TLS (využívá se i u HTTP), která vytváří šifrované spojení mezi komunikujícími uzly. Na druhou stranu její nasazení může být v případě omezených zařízení problematické. Druhou možností je šifrování přímo přenášených dat (Payload) již na aplikační vrstvě. Dalo by se tedy konstatovat, že MQTT podporuje spoustu bezpečnostních služeb, které ale obecně neřeší rizika ovlivňující MQTT [87].

14.3.2 Zabezpečení CoAP

Druhým protokolem je CoAP, navržený specificky pro M2M komunikaci a je velmi podobný HTTP. CoAP je výbornou volbou, pokud je vyžadována nízká režie a jednoduchost, ale podobně, jako MQTT, není bezpečný. Hrozby, které přináší, je opět možné rozdělit do několika kategorií:

- **Parsování zpráv** – Parsery nesprávně zacházejí s příchozími zprávami. Může mít vliv na dostupnost uzlu vlivem jeho přetížení, a zároveň umožnit vzdáleně spouštění libovolného kódu.
- **Proxy a ukládání do mezipaměti (cache)** – Mechanismy správy přístupu u proxy a cache nejsou správně implementovány. Mohlo by dojít ke kompromitaci obsahu, a tím pádem ke ztrátě důvěryhodnosti a integrity zpráv.
- **Bootstrapping** – Nastavení a spuštění nových CoAP uzlů není správně implementováno. Neautorizovaná zařízení by tak mohla získat přístup do CoAP prostředí.
- **Generování klíčů** – Neposkytuje dostatečně robustní mechanismus generování kryptografických klíčů. Jejich použití ohrožuje uzly.
- **Spoofing IP adres** – Zfalšováním IP adresy některého z uzlů by útočník mohl vykonat několik různých útoků.
- **Výměny napříč protokoly** – Útočník zašle na CoAP uzel zprávu s falešnou IP adresou a číslem portu a může cílový uzel donutit k její interpretaci podle pravidel cílového protokolu.

Hlavním rozdílem oproti MQTT je skutečnost, že CoAP využívá protokol UDP, a tím pádem nepodporuje TLS. Místo toho využívá upravenou verzi s názvem Datagram Transport Layer Security (DTLS), a proto se někdy tento protokol označuje jako secured CoAP (CoAPs). Jelikož byl vyvinut tak, aby co nejvíce připomínal TLS, zabránilo se vzniku zcela nového protokolu. Původně byl DTLS určen pro výkonná zařízení, ale díky úpravám pro 6LoWPAN se výrazně snížila velikost packetu, fragmentace a spotřeba energie. V současnosti je pro integrovaná zařízení dostupná velmi zjednodušená verze s názvem tinydtls. Někteří protokol

DTLS označují za nejlepší pro IoT zařízení, protože dokáže ochránit data pomocí klíčů, zajistit výměnu klíčů a provést autentizaci (two-way handshake) [87]. K tomuto dni (28.7.2021) obsahuje databáze NVD celkem 16 CVE záznamů pro CoAP.

14.3.3 Zabezpečení XMPP

Protokol XMPP byl původně vytvořen specificky pro aplikaci Jabber. Je založen na XML a k přenosu dat využívá TCP. XMPP nabízí jednoduchý způsob adresování zařízení formou name@domain.com a užitečný je především v situacích, kde jsou data zasílána mezi vzdálenými uzly.

Z bezpečnostních služeb využívá metodu SASL, která se primárně stará o ověřování klientů na serverech a několikrát zmíněné TLS, pro zajištění důvěryhodnosti a integrity dat. Nemá ovšem plně implementované šifrování end-to-end komunikace, a to s sebou přináší různá rizika. Kromě toho spousta dalších hrozeb ovlivňuje produkty a služby používající XMPP. Obvyklými problémy jsou nedostatečná kontrola operací s pamětí, nevhodné ověřování certifikátů a přítomnost na pevno naprogramovaných (hard-coded) účtů. V současnosti (29.7.2021) je v databázi NVD 85 záznamů.

Za účelem zmírnit tyto hrozby, byla vytvořena série rozšíření, která se stále rozrůstá, s označením XEP series. Například část s číslem XEP-0178 je zaměřena specificky na správné nakládání s certifikáty v SASL autentizaci [87].

14.3.4 UPnP

Kromě uvedených komunikačních protokolů stojí za zmínku i funkce Universal Plug and Play (UPnP). Jedná se o sadu síťových protokolů, primárně určenou pro domácí sítě, která umožňuje síťovým zařízením, jako počítače, tiskárny, WiFi přístupové body a mobilní zařízení, navzájem objevovat vzájemnou přítomnost v síti a vytvářet funkční síťové služby. I přesto, že bylo UPnP navrženo s cílem ulehčit uživatelům připojit nová zařízení do sítě, přináší s sebou i několik vážných bezpečnostních mezer, které mohou útočníci zneužít k infiltrování sítě. Důrazně se doporučuje tuto funkci vypínat na všech zařízeních, pokud je to možné. Po jejím vypnutí je třeba

otestovat síťovou funckionalitu a zcela jistě bude potřeba některé služby manuálně nakonfigurovat.

Hlavní hrozbou UPnP je to, že se všechna lokální zařízení považuje za důvěryhodné a přátelské. Pokud se jedno z těchto zařízení stane obětí útoku a bude infikováno malwarem, mohlo by dojít k jeho šíření do ostatních periferií skrz router podporující tuto funkci.

15 Posouzení zranitelností a rizik v IoT systému

Informace v kapitole Tvorba IoT řešení jsou převzaty z online kurzu Cisco [62], z části Chapter 6: Vulnerability and Risk Assessment in an IoT System, pokud není uvedeno jinak.

Cílem procesu posouzení zranitelností a rizik je najít hrozby, které mohou být zneužity k potenciálnímu útoku na systém. Proceduru je možné provádět rutinně a pravidelně, často automatizovaně, nebo může být zaměřena na specifické části systému. Běžně se používají zdarma nebo komerčně dostupné nástroje. Velmi oblíbeným prostředím, které obsahuje velké množství těchto nástrojů (Nmap, Netcat, John the Ripper a další), je Kali Linux.

Průběh procesu by měl být důkladně naplánován. Plán obsahuje cíle a rozsah testu, použité nástroje a určení toho, kdo bude testy provádět. Testy mohou být prováděny zevnitř sítě nebo ze vzdálené lokace mimo síť. Veškeré zjištěné hrozby je nutné ohlásit a zaznamenat i spolu s jejich potenciální úrovní hrozby.

Proces posouzení se rozděluje na tři typy podle toho, kdo posouzení provádí a podle úrovně znalostí posuzovatele o zkoumané síti. Jsou to:

- **White box** – Posuzovatelé mají znalosti o zkoumaném systému. Často se při testování zaměřují na konkrétní aspekty. Posuzovatelé operují zevnitř organizace.
- **Black box** – Tento typ se nejvíce podobá skutečnému útoku. Posuzovatelé pracují pro třetí stranu a nemají žádné znalosti o zkoumaném systémů před provedením testů.
- **Gray box** – White box testeři identifikují hrozby a black box testeři jsou najati k testování jejich zneužití. Gray box testeři jsou kombinací obojího. Mají

částečnou znalost o zkoumaném systému, zahrnující přístup k dokumentaci o architektuře vnitřní sítě. Jejich cílem je ověřit existenci hrozeb, určit jejich snadnost zneužití a popsat eventuální dopady.

Jako součást posuzovacího procesu se doporučuje provádět takzvané penetrační testy. Jedná se o skutečné útoky, které mají odhalit potenciální dopady související se zneužitím známých hrozeb. Roli útočníka vykonávají etičtí hackeři a snaží se co nejvěrohodněji replikovat útok, jako by ho provedl skutečný hacker. Penetrační testování je součástí black box, jelikož útočník nemá žádné znalosti o struktuře cílového systému. Kromě ověřování existence hrozeb a jejich dopadů se penetrační testy dají využít i k potvrzení, že přijatá bezpečnostní opatření jsou účinná.

15.1 Posouzení rizika

Zranitelnosti IT systémů jsou typicky založeny na lidském faktoru, použitých protokolech a architektuře sítě. Zabezpečení OT se pro změnu týká hlavně bezpečnosti lidí a vybavení. IoT oproti tomu umožňuje vznik zcela novým rizikům. Asi zřejmě největším rizikem IoT jsou daleko rozsáhlejší útoky zapříčiněné velikostí IoT systémů (attack surface).

Každá zranitelnost softwaru nebo systému, posouzená v kontextu konkrétní organizace, představuje rozdílné riziko. To znamená, že stejná zranitelnost může mít pro jednu organizaci nepředstavitelný dopad a pro jinou zanedbatelný. Kromě toho záleží i na aktivech různých společností, které mohou mít pro útočníky větší hodnotu, čímž se pravděpodobnost zneužití nějaké zranitelnosti výrazně zvětšuje. Například stejná chyba ve webové aplikaci dovoluje útočníkovi získat z jedné společnosti osobní informace o klientech a jejich kreditních kartách a z jiné pouze inventář produktů a informace o nich.

Velikost rizika se odvíjí ze vztahu mezi hrozbami, zranitelnostmi a aktivy organizace. S jeho určením mohou pomoci odpovědi na následující otázky:

- Kdo jsou útočníci, kteří chtějí daný systém napadnout?
- Jaké slabiny může útočník v systému zneužít?
- Jaké dopady by měl úspěšný útok na systém/na organizaci?
- Jaká je pravděpodobnost, že dojde k různým útokům?

- Co může organizace udělat ke snížení rizika?

15.1.1 Hodnocení rizika pomocí Common Vulnerability Scoring System

Kapitola vychází z oficiální specifikace CVSS publikované na stránkách organizace FIRST.org [92].

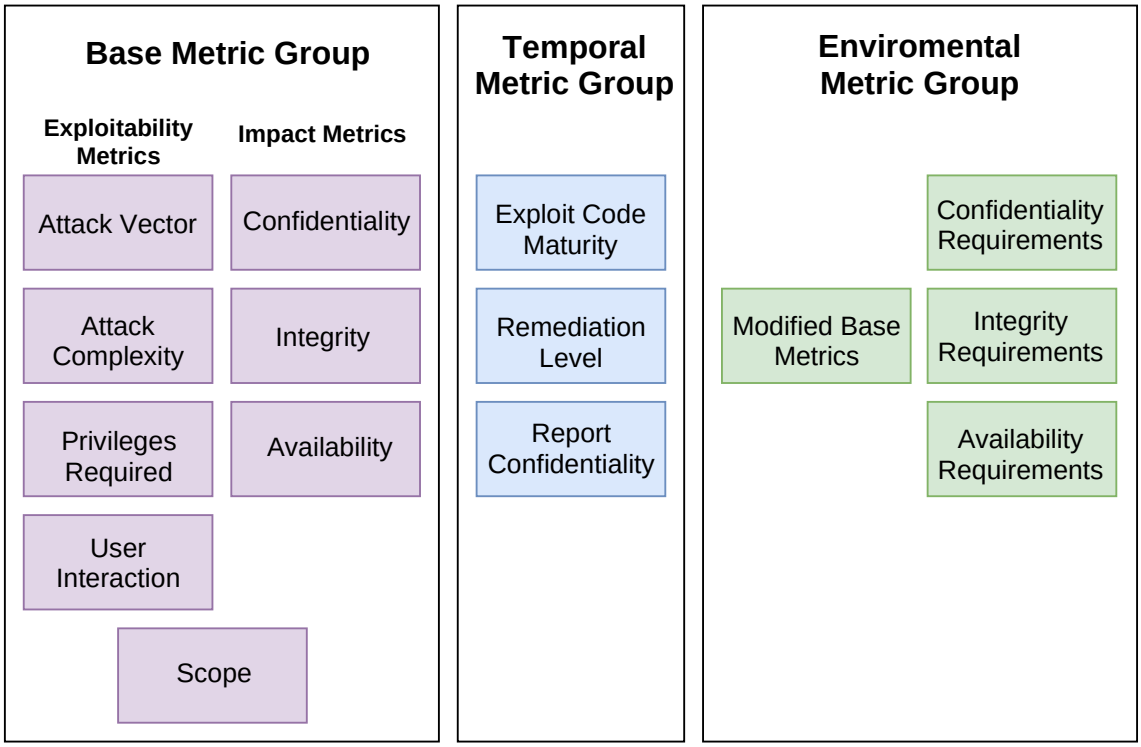
Common Vulnerability Scoring Systém (CVSS) je nástroj využívaný v procesu posuzovaní rizika pro vyjádření společných vlastností a závažnosti hrozeb hardwarových a softwarových systémů. Aktuální revize CVSS 3.1 (z 2019) je na dodavateli nezávislý (vendor-neutral) průmyslový standard pro vyjádření rizika pomocí různých kategorií metrik. Kombinace těchto vah poté slouží ke kvantifikaci rizika spojeného s danou zranitelností. Numerickou hodnotu je možné využít k určení závažnosti a priority pro její řešení. Zodpovědnost za správu a globální prosazení tohoto standardu má globální organizace s názvem The Forum of Incident Response and Security Teams (FIRST).

Současná verze CVSS není bohužel příliš vhodná pro použití v IoT systémech, jelikož nezohledňuje některé aspekty týkající se bezpečnosti pro ně charakteristické. I přesto je CVSS velmi užitečné pro pochopení, jakým způsobem je možné hodnotit zranitelnosti pomocí rizika. Očekává se, že připravovaná verze 4.0 by měla obsahovat několik novinek a změn pro oblast IoT.

Hodnocení pomocí CVSS vyžaduje výběr z několika daných hodnot celkem ve třech metrických skupinách pro každou identifikovanou zranitelnost. Zmíněnými třemi skupinami jsou (viz Obrázek 57):

- **Základní metrická skupina** (Base Metric Group) – Reprezentuje charakteristiky zranitelnosti, které jsou v čase neměnné a nezávislé na kontextu. Charakteristiky se dělí do dvou tříd, a to metriky zneužití zranitelnosti (Exploitability metrics) a metriky dopadů zranitelností (Impact Metrics), které odpovídají CIA triádě.
- **Dočasná metrická skupina** (Temporal Metric Group) - Měří charakteristiky, které se mohou v čase měnit. To znamená, že závažnost nově objevené zranitelnosti bude velká, ale bude klesat s příslušnými opravami a opatřeními.

- **Okolnostní metrická skupina** (Enviromental Metric Group) – Měří charakteristiky závisející na specifickém prostředí a okolnostech v organizaci.



Obrázek 57 Metrické skupiny CVSS
Zdroj: vlastní zpracování (podle [62])

Base Metric Group je nejdůležitější metrickou skupinou, jelikož tvoří základ výsledného skóre a zbývající dvě skupiny ho dále pouze upravují v závislosti na čase a prostředí. Metriky první skupiny jsou:

Metriky zneužití (Exploitability Metrics)

- **Attack Vector** – Vyjadřuje blízkost útočníka ke zranitelné komponentě. S rostoucí vzdáleností útočníka (logickou i fyzickou) poroste i skóre.
- **Attack Complexity** – Popisuje stav systému, který je mimo kontrolu útočníka, ale který musí existovat, aby bylo možné slabinu zneužít. Není zde zahrnuta žádná uživatelská interakce, pro kterou existuje samostatná metrika.

- **Privileges Required** – Vyjadřuje úroveň oprávnění, kterou útočník musí mít před provedením útoku. Čím méně oprávnění potřebuje, tím vyšší skóre bude.
- **User Interaction** – Zachycuje požadavek na účast uživatele, jiného než útočníka, k úspěšnému provedení útoku. Vyjadřuje tedy, zda zranitelnost může být zneužita čistě z vůle útočníka, nebo je potřeba účast někoho dalšího.
- **Scope** – Určuje, zda zranitelnost jedné komponenty má dopad na zdroje v jiném bezpečnostním segmentu (security scope).
    - **Security authority** – Mechanismus, který definuje a uplatňuje řízení přístupu ve smyslu, jak objekty/aktéři mohou přistupovat k jinak nepřístupným objektům a zdrojům.
    - **Security scope** – Všechny objekty a subjekty patřící do jurisdikce jedné autority.

Metriky dopadů zranitelností (Impact Metrics)

- **Confidentiality Impact** – Měří dopad na důvěryhodnost důsledkem úspěšného zneužití zranitelnosti.
- **Integrity Impact** – Měří dopad na integritu důsledkem úspěšného zneužití zranitelnosti.
- **Availability Impact** – Měří dopad na dostupnost důsledkem úspěšného zneužití zranitelnosti.

Proces výpočtu CVSS využívá online nástroj s názvem CVSS v3.1 Calculator, který je dostupný na oficiálních stránkách organizace FIRST (first.org). Kalkulačka má styl dotazníku, kde zvolené odpovědi reprezentují hodnoty v jednotlivých metrických skupinách pro danou zranitelnost. U každé metriky je možné zvolit pouze jednu hodnotu z předem dané množiny. Po vyplnění všech potřebných hodnot v základní skupině se vygeneruje skóre. Následně je možné vyplnit metriky i v časové a okolnostní skupině, čímž dojde k modifikaci celkového skóre. Závažnost se určuje na stupnici 0.0 až 10.0 a je rozdělená do pěti úrovní závažnosti:

- None – 0.0
- Low – 0.1 až 3.9
- Medium – 4.0 až 6.9
- High – 7.0 až 8.9
- Critical – 9.0 až 10.0

15.2 Modelování hrozeb podrobněji

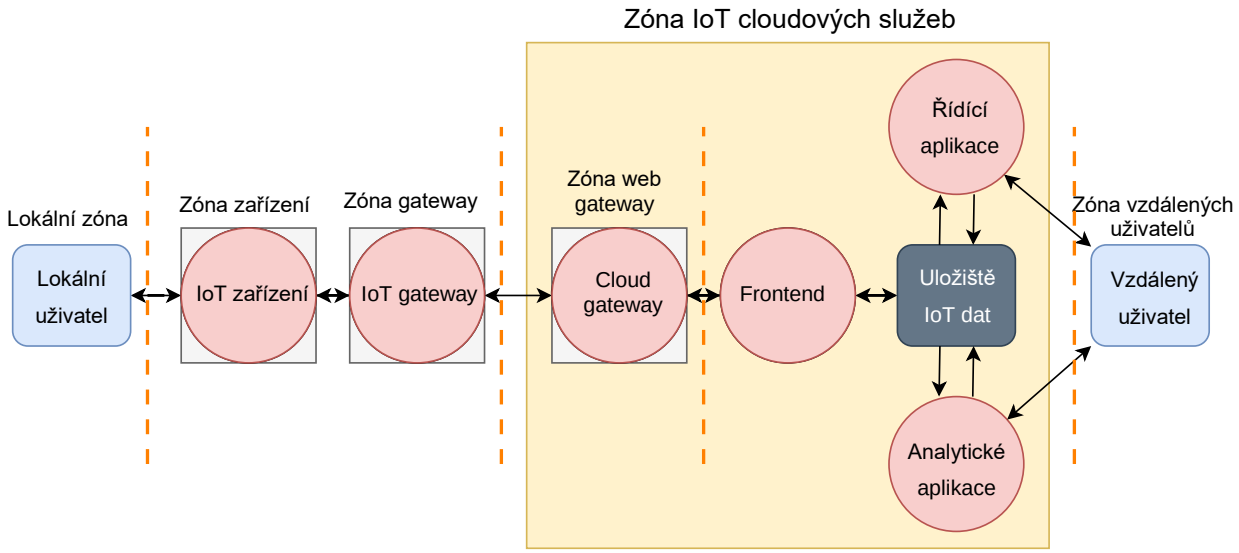
Modelování hrozeb je proaktivní přístup k posuzování bezpečnosti systémů. Existují tři přístupy, které se liší pohledem na zkoumaný systém. Prvním z nich je zaměřený na útok (attack-centric), a to znamená, že proces modelování probíhá z pohledu útočníka. Druhý se soustředí na obranu (Defense-centric)
Proces modelování hrozeb byl již stručně představen v předchozích kapitolách (viz 11.7 Model pro analýzu hrozeb), a proto se tato kapitola bude věnovat pouze doplnění a rozšíření jednotlivých kroků.

Nejdůležitějším dodatkem je zřejmě vypracování diagramu toku dat (DFD) ve druhém a třetím kroku. Potom, dojde k identifikaci bezpečnostních cílů v prvním kroku, je důležité vypracovat detailní dokumentaci celého systému, včetně veškerého datového toku. Následuje krok rozložení IoT systému, ve kterém by mělo dojít k určení hranic důvěryhodnosti a ke tvorbě právě DFD diagramu.

DFD by měl zahrnovat komponenty, jako IoT zařízení, IoT gateway, lokální aplikace, edge zařízení, datové aplikace apod. Entity v DFD by měli splňovat tři základní pravidla. (1) Každý proces by měl mít alespoň jeden vstup a výstup, (2) Datová uložiště by měly mít toky pro čtení a zápis, (3) Uložená data v systému musejí projít alespoň jedním procesem. Ke grafické reprezentaci DFD existuje několik různých notací, ale jednou z nejznámějších je konvence Yourdon & Coad. Ta, stejně jako ostatní notace, definuje čtyři základní typy elementů, a to externí entitu, proces, uložiště dat a tok dat. Jediné, v čem se ostatní notace liší, je grafická reprezentace těchto elementů.

Dále je užitečné si v takto připraveném diagramu rozdělit entity do funkčních zón. Zóna označuje oblast systému, která vyžaduje rozdílnou autentizaci a autorizaci. Pomáhají i limitovat vystavení komponent systému zranitelnostem, které existují v jednotlivých zónách. Příkladem zón mohou být senzorová oblast,

webová aplikace, IP gateway apod. Po určení zón mohou být do diagramu přidány hranice důvěry, které vytvoří v síti sekce, kde úroveň důvěry mezi zařízeními na každém jejím konci bude rozdílná. Příklad DFD diagramu i s určenými funkčními oblastmi a vytyčenými hranicemi důvěry je vidět na obrázku (viz Obrázek 58).



Obrázek 58 DFD diagram s funkčními zónami a hranicemi důvěry
Zdroj: vlastní zpracování (podle [62])

15.2.1 Model pro identifikaci hrozeb STRIDE

Po vytvoření struktury systému může pokračovat proces identifikace hrozeb. V této části je možné aplikovat model STRIDE, který definuje šest kategorií hrozeb, včetně jejich definic a příkladů, spolu s jejich odpovídajícími bezpečnostními cíli. V 11. kapitole již bylo zmíněno, že zkratka STRIDE odpovídá anglickým pojmům Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service a Elevation of Privilege. Těmto hrozbám v daném pořadí odpovídají bezpečnostní cíle, a to konkrétně pravost (Authenticity), integrita (Integrity), nemožnost zahladit stopy (Non-repudiation), důvěrnost (Confidentiality), dostupnost (Availability) a autorizace (Authorization).

Ovšem ne všechny zmíněné hrozby se mohou vyskytovat u všech elementů DFD. Součástí STRIDE je proto i schéma, které určuje, jaká hrozba přísluší danému elementu. Nejvíce ohroženým elementem je proces, kterého se týkají všechny hrozby STRIDE. Následuje uložiště dat, kde potenciální hrozby jsou pouze TRID.

Datový tok je na tom podobně se třemi hrozbami TID a nejméně ohroženým elementem je překvapivě externí entita s hrozbami SR.

Je velmi důležité zdůraznit, že STRIDE není způsob posuzování zranitelností, ale pouze nástroj pro tvorbu přehledu potenciálních hrozeb u jednotlivých částí systému.

15.2.2 Model pro určení rizika DREAD

Pro každou hrozbu objevenou pomocí STRIDE je nutné určit její úroveň rizika pro organizaci. K tomu je možné využít model DREAD, který podobně jako CVSS dokáže riziko vyjádřit v numerické podobě. Výsledné hodnocení může být využito při stanovení priorit a v rozhodování o přijetí opatření. Z předchozích kapitol je známo, že DREAD je zkratkou pro pět anglických pojmů:

- **Damage potential** – Pokud dojde ke zneužití zranitelnosti, jaké dopady to může mít na systém?
- **Reproducibility** – Jaká je pravděpodobnost, že zranitelnosti využije více jak jeden útočník (Je útok reprodukovatelný?)?
- **Exploitability** – Jak jednoduché je danou zranitelnost zneužít?
- **Affected Users** – Jaké procento uživatelů je eventuálně ovlivněno, pokud dojde ke zneužití zranitelnosti?
- **Discoverability** – Jak jednoduché je zranitelnosti objevit? Jak známá je?

Následně jsou podle těchto kategorií hodnoceny všechny hrozby na stupnici, která je znázorněna v následující tabulce (viz Tabulka 5). Existuje i další stupnice společnosti OWASP s úrovněmi 0 (nízké riziko), 5 (střední riziko) a 10 (vysoké riziko), ale princip je stále stejný. Výsledná průměrná hodnota rizika se vypočítá jako součet všech udělených vah, který se vydělí číslem 5 (tzn. (D+R+E+A+D)/5).

Kategorie DREAD	3 - Vysoké riziko	2- Střední riziko	1 – Nízké riziko
Poškození ***D**amage Potential*	Systém nefunguje nebo je pod kontrolou útočníka; poškození uživatelů nebo zařízení.	Ztráta důležitých dat; dočasné výpadky dostupnosti systému.	Drobná nebo střední ztráta dat; Nízké dopady na systém.
Reprodukovatelnost ***R**eproducibility*	Každý pokus o útok bude úspěšný.	Zhruba polovina pokusů o útok bude úspěšná.	Velmi náročné na reprodukci; útok vyžaduje specifické podmínky nebo stav systému.
Zneužitelnost ***E**xploitability*	Útok je jednoduchý na provedení i nezkušenými útočníky.	Útok vyžaduje zkušenosti.	Útok vyžaduje velké množství zkušeností nebo organizovanou skupinu.
Ovlivnění uživatelé ***A**ffected Users*	Dostatek zařízení, aby to způsobilo vážný výpadek. Všichni uživatelé, kteří splňují standardy.	Některá zařízení, která nejsou aktualizována nebo nesplňují současné standardy.	Malé množství uživatelů a zařízení se specifickými konfiguracemi a rolemi.
Viditelnost ***D**iscoverability*	Velmi známé v hackerské komunitě. Představuje velkou hodnotu pro útočníky.	Méně známé a vyskytují se velmi málo, některé z nich benefitují útočníka.	Málo známé a pro útočníka téměř bezvýznamné.

Tabulka 5 Kategorie DREAD
Zdroj: vlastní zpracování (podle [62])

Někteří odborníci radí při hodnocení jednotlivých hrozeb trošku skeptický přístup a doporučují v kategorii Reprodukovatelnosti a Viditelnosti vždy dávat největší váhu. Důvodem je to, že není možné se před útočníky zcela schovat a vždy nějakou zranitelnost najdou.

15.3 Reakce na riziko

Po identifikaci hrozeb a určení jejich úrovně rizika je potřeba tyto informace využít a přijmout potřebná opatření. Řízení rizik (Risk Management) rozšiřuje proces posuzování hrozeb a rizik na komplexnější úroveň, která je následně udržována a sledována podnikem. Dokumentace procesu je běžně vyžadována především pro organizace podnikající ve vládním a finančním sektoru.

Běžně se uvádějí čtyři základní cesty, kterými se dá s rizikem vypořádat. Proces posouzení úrovně rizika může poté pomoci s rozhodováním, jakou z nich se vydat. Tyto přístupy se označují jako 4T a jsou to:

- **Vyhýbání se riziku** (Terminate) – Jedná se o okamžité zastavení aktivity, která dané riziko generuje. Při dalším zkoumání se může ukázat, že daná aktivita nepřináší organizaci benefit úměrný generovanému riziku, což může vést k permanentnímu přerušení.
- **Redukce rizika** (Treat) – Dochází ke snížení rizika přijmutím potřebných opatření. To zahrnuje implementaci přístupů zmíněných výše.
- **Sdílení rizika** (Transfer) – Přesunutí části rizika na ostatní účastníky. Toho lze docílit outsourcingem některých bezpečnostních služeb od třetí strany. Takové služby se označují jako Security as a Service (SECaaS, pozor neplést se SaaS). Dalším příkladem je zajištění pojištění podniku, a tím snížit případné finanční ztráty.
- **Udržování rizika** (Tolerate) – Přijmutí jak rizika, tak jeho eventuálních dopadů. Tato strategie je vhodná pouze pro situace, kdy potenciální dopady rizika nejsou nijak závažné, nebo náklady na požadovaná opatření by byly příliš velké. Existují ovšem i rizika, u kterých jiná volba neexistuje, jelikož je nereálné je vymýtit, snížit nebo sdílet.

16 Shrnutí

Podle struktury Cisco online kurzů (IoT Fundamentals: Connecting Things a IoT Fundamentals: IoT Security) byl vytvořen výukový materiál, který se dělí na dva velké tematické celky a celkem na dvanáct kapitol. První velký okruh, 1. – 6. kapitola, byl věnován seznámení s oblastí IoT, s charakteristikou a vlastnostmi IoT systémů, představení mikrokontroleru Arduino a jednodeskového počítače RaspberryPi, komunikaci v IoT systémech, tomu, jak IoT ovlivňuje průmysl (IIoT), a nakonec procesu tvorby vlastního IoT řešení. Druhý okruh, 7. – 12. kapitola, se zabýval výhradně bezpečností IoT systému a obsahoval informace o známých architekturách, útocích na IoT postupně na vrstvě zařízení, komunikace, aplikací, a nakonec byl představen postup posouzení hrozeb a určení jejich rizika. Materiály byly vypracovány s předpokladem, že studenti mají alespoň základní znalosti z programování, počítačových sítí a bezpečnosti dat. To znamená, že věci, které jsou obsahem jiných předmětů, byly v textu jen pro kontext zmíněny, ale nebyly podrobně popsány.

Obsah Cisco kurzů je ve spoustě případů příliš stručný a zaměřený na proprietární řešení. Obsah výuky byl proto doplněn o materiály a aktuální poznatky s využitím odborných publikací, standardů a protokolů a to přímo z oficiálních dokumentací a technických specifikací. Kurzy Cisco díky interpretaci základních informací nebývají často aktualizované, respektive nereagují operativně na nové trendy. Tento fakt byl při tvorbě zohledněný a studijní materiály byly podrobeny revizi a případné aktualizaci informací tak, aby odpovídali současné situaci a trendům v IoT. Tím bylo dosaženo stavu, že nejde o pouhý překlad Cisco materiálů do češtiny. Ke každému tématu vznikly prezentace určené pro podporu teoretické části každého cvičení. Tyto kromě teorie popsané v tomto textu byly doplněny i o zajímavosti a videa související s probíraným tématem.

Za pomoci uvedených zdrojů proběhla v letním semestru 2021 výuka. Všechny uvedené materiály a doporučená literatura byly pro studenty dostupné na e-learningovém portálu oliva.uhk.cz a alternativně ve skupině předmětu vytvořené v prostředí Microsoft Teams. Kvůli povinné distanční výuce nebylo možné realizovat praktická cvičení ve specializované laboratoři. Z toho důvodu bylo nutné

přistoupit k alternativním řešením a nalézt vhodné virtuální nástroje, které by ji alespoň částečně nahradily. Jako nejvhodnější bylo vybráno prostředí TinkerCAD, které nabízí tvorbu vlastního řešení formou nepájených obvodů s deskou Arduino a jejich následnou konfiguraci v jazyce Wiring. I přesto, že prostředí je velmi kvalitní a obsahuje velké množství komponent, má jeden hlavní nedostatek, a to, že neobsahuje moduly pro bezdrátovou komunikaci, které byly v předchozích verzích odstraněny z bezpečnostních důvodů. I tak bylo prostředí dostačující pro jednoduché ukázky a testování pro závěrečné práce studentů. Kromě toho byly ze strany vyučujících provedeny online ukázky vhodných a proveditelných zadání z Cisco kurzů. Vhodnými se myslí to, že spousta cvičení v kurzech, především ve druhém kurzu o bezpečnosti, je určena pro laboratorní prostředí, jelikož tam dochází k demonstracím různých síťových útoků a zranitelností. Dále byly vytvořeny i vlastní zadání, například pro seznámení a práci v prostředí TinkerCAD (viz Příloha 1).

Veškeré materiály byly s vyučujícími před výukou konzultovány a po výuce byly na základě zpětné vazby a poznatků provedeny vhodné korekce a úpravy v obsahu a struktuře. Od 6. týdnu semestru, který byl věnován kapitole Tvorba IoT řešení, byly studenti poučeni o vypracování závěrečné práce a byla jim poskytnuta šablona pro dokumentaci jejich řešení (viz Příloha 2), která kromě struktury dokumentu obsahuje i odkazy na užitečné nástroje a související kapitoly. Vzhledem k situaci, kdy nebylo možné využít fyzického vybavení v laboratoři, bylo pro závěrečné ověření znalostí studentů hodnocení postaveno na vypracování dokumentace vlastního IoT projektu. Toto by se dalo za stavu normálního průběhu výuky a při větším počtu studentů pojmout jako společná práce v definovaných týmech. K prověření individuálních znalostí je připravený test v systému oliva, který zahrnuje otázky z celého obsahu výuky.

17 Závěr

Cílem diplomové práce bylo zpracovat výukové materiály pro nový předmět, Internet věcí (IOT), jenž má studenty seznámit s problematikou tvorby a implementace IoT řešení a upozornit je na její potenciální slabiny a úskalí, která mohou nastat v průběhu životního cyklu.

Spojením, optimalizací a doplněním obsahu uvedených online kurzů od společnosti Cisco, je v obsahu předmětu dostatek informací na úrovni VŠ studia. Dodatečné informace byly čerpány z dostupné odborné literatury, oficiálních webů, dokumentací a technických specifikací. Pro samotnou výuku byly ke každému tématu vytvořeny přednáškové prezentace. V praktické části byla využita vlastní cvičení, která jsou uvedená jako Příloha 1 této práce. Dále bylo využito praktických úloh dostupných v použitých kurzech Cisco.

Studenti se v běhu realizované výuky seznámili se základy IoT problematiky, s tvorbou návrhu řešení, vývojem i realizací prototypu a tvorbou potřebné dokumentace. Celý proces byl veden s důrazem na bezpečnost řešení ve všech rovinách. Vzhledem k povinné distanční výuce v letním semestru 2020/2021 nebylo možné provádět laboratorní praktika. Aspoň částečně byla nahrazena úkoly ve virtuálním prostředí. Pro závěrečný praktický projekt byla, podle 9. kapitoly – Tvorba IoT řešení, vytvořena šablona dokumentace (viz Příloha 2) tak, aby studenti plně zúročili probranou problematiku ze všech kapitol.

Ze zájmu studentů o předmět je zřejmé, že jde o aktuální téma. Z jejich aktivního zapojení při výuce lze usoudit, že byl obsah přednesen srozumitelně. Obsah odevzdaných závěrečných prací měl adekvátní úroveň. Z tvorby podkladů a průběžného sledování vývoje IoT a technologií obecně je zřejmé, že bude nezbytné materiály v každém roce revidovat a aktualizovat, aby výuka byla pro studenty přínosem a aktuálním zdrojem informací.

18 Seznam použité literatury

[1] CISCO NETWORKING ACADEMY. IoT Fundamentals: Connecting Things: Internet of Things. *Cisco* [online vzdělávací kurz]. 2018 [cit. 2021-8-6]. Dostupné z: https://www.netacad.com/courses/iot/iot-fundamentals

[2] S. GILLIS, Alexander. Internet of Things (IoT). IoT Agenda [online]. 2020 [cit. 2021-8-6]. Dostupné z: https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

[3] RANGER, Steve. What is the IoT? Everything you need to know about the Internet of Things right now: Updated: The Internet of Things explained. What the IoT is, and where it's going next. ZDNet [online]. 2020, 3. únor [cit. 2021-8-6]. Dostupné z: https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/

[4] SOFTWARE TESTING HELP. 10 Powerful Internet Of Things (IoT) Examples Of 2021. STH [online]. 2021 [cit. 2021-8-6]. Dostupné z: https://www.softwaretestinghelp.com/best-iot-examples/

[5] TIM WESCOTT, 1 - The Basics, In Embedded Technology, Applied Control Theory for Embedded Systems, Newnes, 2006, str. 1-9, ISBN 9780750678391, Dostupné z: (https://www.sciencedirect.com/science/article/pii/B9780750678391500027)

[6] ELECTRICAL4U. Control Systems: What Are They?: Open-Loop & Closed-Loop Control System Examples. Electrical4U [online]. 2020 [cit. 2021-8-6]. Dostupné z: https://www.electrical4u.com/control-system-closed-loop-open-loop-control-system/

[7] IAPP.ORG. What does privacy mean? IAPP [online]. 2020 [cit. 2021-8-6]. Dostupné z: https://iapp.org/about/what-is-privacy/

[8] OVIC. Internet of Things and Privacy: Issues and Challenges [online]. 2021, duben [cit. 2021-8-6]. Dostupné z: https://ovic.vic.gov.au/privacy/internet-of-things-and-privacy-issues-and-challenges/

[9] HARE, Jason. OPENDATASOFT. What Is Metadata and Why Is It as Important as the Data Itself? Opendatasoft.com [online]. 2016, 25. srpen [cit. 2021-8-6]. Dostupné z: https://www.opendatasoft.com/blog/2016/08/25/what-is-metadata-and-why-is-it-important-data

[10] STORR, Wayne. Basic Electronics Tutorials: For Beginners and Beyond [online]. 2013 [cit. 2021-8-6]. Dostupné z: https://www.academia.edu/8221534/Basic_Electronics_Tutorials_for_beginners_and_beyond_by_Wayne_Storr.

[11] MALÝ, Martin. Hradla, volty, jednočipy: úvod do bastlení. Praha: CZ.NIC, z.s.p.o., 2017. CZ.NIC. ISBN 978-80-88168-23-2

[12] TESAŘ, Zdeněk a Iva PETŘÍKOVÁ. Základy elektroniky, součástky a obvody pro integrovanou výuku VUT a VŠB-TUO. Ostrava: Vysoká škola báňská – Technická univerzita Ostrava, 2014. ISBN 978-80248-3628-7.

[13] BEDNAŘÍK, Michal. Fyzika 1. 1. vydání. V Praze : České vysoké učení technické, 2011. ISBN 978-80-01-04834-4.

[14] KEKULE, Jaromír. Kirchhoffovy zákony. Elektross.gjn.cz [online]. 2003, 10. listopad [cit. 2021-8-7]. Dostupné z: http://elektross.gjn.cz/elektrina/el_proud/vedeni_proudu/kovy/kirch_zak.html

[15] VEICHI. What is the Difference Between AC and DC Power. Veichi.org [online]. 2013 [cit. 2021-8-7]. Dostupné z: https://www.veichi.org/solutions/related-articles/what-is-the-difference-between-ac-and-dc-power.html

[16] M. MIMS, Forest. Getting Started in Electronics [online]. 12. USA, 1994 [cit. 2021-8-7]. Dostupné z: https://www.academia.edu/9885504/Getting_Started_In_Electronics_Forrest_M_Mims

[17] MAJLING, Eduard. Transformátor – základní vlastnosti a dělení. Oenergetice.cz [online]. 2015, 5. duben [cit. 2021-8-7]. Dostupné z: https://oenergetice.cz/elektroenergetika/transformator-zakladni-vlastnosti-a-deleni

[18] PREDKO, Michael. Digitální elektronika bez předchozích znalostí. Brno: Computer Press, 2008. ISBN 978-80-251-2124-5.

[19]MPS. Analog Signals vs. Digital Signals. Monolithicpower.com [online]. [cit. 2021-8-7]. Dostupné z: https://www.monolithicpower.com/en/analog-vs-digital-signal

[20]GUDINO, Miguel. Engineering Resources: Basics of Analog-to-Digital Converters. Arrow.com [online]. 2018, 17. duben [cit. 2021-8-7]. Dostupné z: https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters

[21]ARDUINO. Introduction to Arduino: What is Arduino? Arduino.cc [online]. [cit. 2021-8-7]. Dostupné z: https://www.arduino.cc/en/guide/introduction

[22]LUTKEVICH, Ben. Microcontroller (MCU). IoT Agenda [online]. 2019 [cit. 2021-8-7]. Dostupné z: https://internetofthingsagenda.techtarget.com/definition/microcontroller

[23]ČERMÁK, Petr a Tomáš KRAMNÝ. Technologie IoT: Studijní opora pro kombinované studium [online]. Olomouc, 2018 [cit. 2021-8-7].

[24]VODA, Zbyšek a tým HW Kitchen. Průvodce světem Arduina [online]. 2. vydání. 2018 [cit. 2021-8-7]. Dostupné z: https://bastlirna.hwkitchen.cz

[25]SELECKÝ, Matúš a Martin HERODEK. Arduino: uživatelská příručka. Brno: Computer Press, 2016. ISBN 978-80-251-4840-2.

[26]TECHOPEDIA. Interpreter. Techopedia.com [online]. 2020, 12. srpen [cit. 2021-8-7]. Dostupné z: https://www.techopedia.com/definition/7793/interpreter

[27]FREEMAN, Jonathan. What is an API? Application programming interfaces explained. Infoworld.com [online]. 2019, 8. srpen [cit. 2021-8-7]. Dostupné z: https://www.infoworld.com/article/3269878/what-is-an-api-application-programming-interfaces-explained.html

[28]Linux: dokumentační projekt. 2., aktualiz. vyd. Praha: Computer Press, 2001. Operační systémy. ISBN 80-722-6503-2.

[29]BROWN, Paul. The Linux Filesystem Explained. Linux.com [online]. 2018, 18. duben [cit. 2021-8-7]. Dostupné z: https://www.linux.com/training-tutorials/linux-filesystem-explained/

[30]KASHIF, Shiekh. Raspberry Pi Introduction. Electronicwings.com [online]. 2019, 1. květen [cit. 2021-8-7]. Dostupné z: https://www.electronicwings.com/raspberry-pi/raspberry-pi-introduction

[31]RaspberryPI FOUNDATION. Oficiální dokumentace: GPIO. Raspberrypi.org [online]. [cit. 2021-8-7]. Dostupné z: https://www.raspberrypi.org/documentation/

[32]TRATNER, Jeff. Control Raspberry Pi GPIO Pins from Python. Ics.com [online]. 2019, 31. červenec [cit. 2021-8-7]. Dostupné z: https://www.ics.com/blog/control-raspberry-pi-gpio-pins-python

[33]ROSE, Karen, Scott ELDRIDGE a Lyman CHAPIN. The Internet of Things: An Overview: Understanding the Issues and Challenges of a More Connected World [online]. The Internet Society (ISOC), 2015 [cit. 2021-8-7]. Dostupné z: https://www.internetsociety.org/resources/doc/2015/iot-overview/

[34]VOJÁČEK, Antonín. Základní úvod do oblasti internetu věcí (IoT). Automatizace.hw.cz [online]. 2016, 16. září [cit. 2021-8-7]. Dostupné z: https://automatizace.hw.cz/zakladni-uvod-do-oblasti-internetu-veci-iot.html

[35]IOT FOR ALL a LEVEREGE. An Introduction to the Internet of Things [online]. Leverege, 2018 [cit. 2021-8-7]. Dostupné z: https://www.leverege.com/ebooks/iot-intro-ebook

[36]WI-FI ALLIANCE. Wi-Fi HaLow: Low power, long range Wi-Fi® for IoT. Wi-fi.org [online]. 2017 [cit. 2021-8-7]. Dostupné z: https://www.wi-fi.org/discover-wi-fi/wi-fi-halow

[37]CHRISTIANO, Marie. A Look at IEEE 802.11ax-2019, the New Wi-Fi Standard for HEW. Allaboutcircuits.com [online]. 2018, 30. leden [cit. 2021-8-7]. Dostupné z: https://www.allaboutcircuits.com/news/IEEE-802.11ax-2019-new-Wi-Fi-standard-hew-high-efficiency-WiFi/

[38]SERPANOS, Dimitrios a Marilyn WOLF. Internet of Things (IoT) Systems: Architectures, Algorithms, Methodologies [online]. Springer, 2018 [cit. 2021-8-7]. ISBN 978-3-319-69715-4. Dostupné z: https://link.springer.com/book/10.1007/978-3-319-69715-4

[39]LORA ALLIANCE. What is LoRaWAN®: A technical overview of LoRA and LoRaWAN. Lora-alliance.org [online]. 2015, November [cit. 2021-8-7]. Dostupné z: https://lora-alliance.org/about-lorawan/

[40]VOJÁČEK, Antonín. SIGFOX - princip, struktura, protokol, použití. Vyvoj.hw.cz [online]. 2017, 26. květen [cit. 2021-8-7]. Dostupné z: https://vyvoj.hw.cz/sigfox-princip-struktura-protokol-pouziti.html

[41]I-SCOOP. NB-IoT explained: a complete guide to Narrowband-IoT. I-scoop.eu [online]. [cit. 2021-8-7]. Dostupné z: https://www.i-scoop.eu/internet-of-things-guide/lpwan/nb-iot-narrowband-iot/

[42]SHELBY, Z., HARTKE, K., and C. BORMANN, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, 2014 červen, Dostupné z: <https://www.rfc-editor.org/info/rfc7252>.

[43]MALÝ, Martin. Protokol MQTT: komunikační standard pro IoT. Root.cz [online]. 2016, 29. červen [cit. 2021-8-8]. Dostupné z: https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/

[44]SAINT-ANDRE, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, 2011 březen, Dostupné z: <https://www.rfc-editor.org/info/rfc6120>.

[45]VELTE, Anthony T., Toby J. VELTE a Robert C. ELSENPETER. Cloud Computing: praktický průvodce. Brno: Computer Press, 2011. ISBN 978-80-251-3333-0.

[46]SORIANO, Miguel a Pavel BEZPALEC. Cloud Computing [online]. České vysoké učení technické v Praze Fakulta elektrotechnická, 2017 [cit. 2021-8-8]. ISBN 978-80-01-06211-1. Dostupné z: http://techpedia.fel.cvut.cz/single/?objectId=77

[47]MARTIN, James A. a Matthew FINNEGAN. What is IFTTT? How to use If This, Then That services. Computerworld.com [online]. 2020, 25. září [cit. 2021-8-8]. Dostupné z: https://www.computerworld.com/article/3239304/what-is-ifttt-how-to-use-if-this-then-that-services.html

[48]POSEY, Brien, SHEA, Sharon a Ivy WIGMORE, ed. Fog computing (fog networking, fogging). Internetofthingsagenda.com [online]. 2020, říjen [cit. 2021-8-8]. Dostupné z:

https://internetofthingsagenda.techtarget.com/definition/fog-computing-fogging

[49]ELLINGWOOD, Justin. An Introduction to Big Data Concepts and Terminology. Digitalocean.com [online]. 2016, 28. září [cit. 2021-8-8]. Dostupné z: https://www.digitalocean.com/community/tutorials/an-introduction-to-big-data-concepts-and-terminology

[50]ITU-T. Overview of the Internet of things. SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS: Next Generation Networks – Frameworks and functional architecture models [online]. 2012, 15. červen [cit. 2021-8-8]. Y. 2060. Dostupné z: https://www.itu.int/rec/T-REC-Y.2060-201206-I

[51]LIN, Shi-Wan, Bradford MILLER, Jacques DURAND, Graham BLEAKLEY, Amine CHIGANI, Robert MARTIN, Brett MURPHY a Mark CRAWFORD. The Industrial Internet of Things: Volume G1: Reference Architecture [online]. Industrial Internet Consortium, 2019, 19. červen [cit. 2021-8-8]. Dostupné z: https://www.iiconsortium.org/IIRA.htm

[52]MATIN, M.A. a M.M. ISLAM. Overview of Wireless Sensor Network [online]. intechopen.com, 2012 [cit. 2021-8-8]. Dostupné z: doi:10.5772/49376

[53]DUSTDAR, Schahram, Stefan NASTIĆ a Ognjen ŠĆEKIĆ. Introduction to Smart Cities and a Vision of Cyber-Human Cities [online]. Springer, 2017, 30. květen [cit. 2021-8-8]. Dostupné z: doi:https://doi.org/10.1007/978-3-319-60030-7_1

[54]LOM, Michal, Miroslav SVÍTEK a Ondřej PŘIBYL. Industry 4.0 as a Part of Smart Cities [online]. květen 2016 [cit. 2021-8-8]. Dostupné z: doi:10.1109/SCSP.2016.7501015

[55]BEE SMART CITY. Smart city indicators: BUILDING THE SMART CITY [online]. [cit. 2021-8-8]. Dostupné z: https://hub.beesmart.city/en/smart-city-indicators

[56]EREMIA, Mircea, Lucian TOMA a Mihai SANDULEAC. The Smart City Concept in the 21st Century. In: Procedia Engineering. 2017, s. 12-19. ISSN 1877-7058. Dostupné z: doi:https://doi.org/10.1016/j.proeng.2017.02.357.

[57]NIDHI N., PRASAD D., NATH V. (2019) Different Aspects of Smart Grid: An Overview. In: Nath V., Mandal J. (eds) Nanoelectronics, Circuits and Communication Systems. Lecture Notes in Electrical Engineering, vol 511. Springer, Singapore. Dostupné z: https://doi.org/10.1007/978-981-13-0776-8_41

[58]TIAN, Shuo, Wenbo YANG, Jehane Michael LE GRANGE, Wei HUANG a Zhewei YE. Smart healthcare: making medical care more intelligent. Global Health Journal [online]. 2019, Issue 3(Volume 3) [cit. 2021-8-8]. ISSN 2414-6447. Dostupné z: doi:https://doi.org/10.1016/j.glohj.2019.07.001

[59]SCIENCE BUDDIES. The Engineering Design Process. Science Buddies [online]. 2002 [cit. 2021-8-8]. Dostupné z: https://www.sciencebuddies.org/science-fair-projects/engineering-design-process/engineering-design-process-steps#problem

[60]PRODUCTPLAN. How to Build a Product Roadmap Based on a Business Model Canvas. Productplan.com [online]. [cit. 2021-8-8]. Dostupné z: https://www.productplan.com/learn/business-model-canvas/

[61]SOOS, Gabor, Daniel KOZMA, Ferenc Nandor JANKY a Pal VARGA. IoT Device Lifecycle – A Generic Model [online]. srpen 2018 [cit. 2021-8-8]. Dostupné z: doi:10.1109/FiCloud.2018.00033

[62]CISCO NETWORKING ACADEMY. IoT Fundamentals: IoT Security: Internet of Things. Cisco [online vzdělávací kurz]. 2018 [cit. 2021-8-6]. Dostupné z: https://www.netacad.com/courses/cybersecurity/iot-security

[63]BRIDEWELL CONSULTING. Different Types Of Hackers – And What They Mean For Your Business. Bridewellconsulting.com [online]. 1. říjen 2018 [cit. 2021-8-9]. Dostupné z: https://www.bridewellconsulting.com/different-types-of-hackers-and-what-they-mean-for-your-business

[64]KASPERSKY. Black hat, White hat, and Gray hat hackers – Definition and Explanation. Kaspersky.com [online]. [cit. 2021-8-9]. Dostupné z: https://www.kaspersky.com/resource-center/definitions/hacker-hat-types

[65]GRIMES, Roger A. What is ethical hacking? How to get paid to break into computers. Csoonline.com [online]. 27. únor 2019 [cit. 2021-8-9]. Dostupné z:

https://www.csoonline.com/article/3238128/what-is-ethical-hacking-and-how-to-become-an-ethical-hacker.html

[66]RED HAT, INC. What is a CVE? Redhat.com [online]. [cit. 2021-8-9]. Dostupné z: https://www.redhat.com/en/topics/security/what-is-cve

[67]CLOUDFLARE. What is the Mirai Botnet? Cloudflare.com [online]. [cit. 2021-8-9]. Dostupné z: https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/

[68]KRČMÁŘ, Petr. Shodan.io: užitečný vyhledávač internetových slabin. Root.cz [online]. 27. listopad 2018 [cit. 2021-8-9]. Dostupné z: https://www.root.cz/clanky/shodan-io-uzitecny-vyhledavac-internetovych-slabin/

[69]NEWHOUSE, William, Stephanie KEITH, Benjamin SCRIBNER a Greg WHITE. National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework. NIST Special Publication 800-181 [online]. National Institute of Standards and Technology, srpen 2017 [cit. 2021-8-9]. Dostupné z: doi:https://doi.org/10.6028/NIST.SP.800-181

[70]BAUER, Martin, Mathieu BOUSSARD, Nicola BUI, et al. Internet of Things Architecture IoT A: Deliverable D1.5 Final architectural reference model for the IoT v3.0 [online]. 2013 [cit. 2021-8-9]. Dostupné z: https://www.researchgate.net/publication/272814818_Internet_of_Things_-_Architecture_IoT-A_Deliverable_D15_-_Final_architectural_reference_model_for_the_IoT_v30

[71]ELMANGOUSH, A., A. CORICI, A. AL-HEZMI a T. MAGEDANZ. 11 - Mobility management for machine-to-machine (M2M) communications. Machine-to-machine (M2M) Communications [online]. Woodhead Publishing, 2015, , 187-206 [cit. 2021-8-9]. Dostupné z: doi:https://doi.org/10.1016/B978-1-78242-102-3.00011-3

[72]ETSI. Machine-to-Machine communications (M2M); Functional architecture: ETSI TS 102 690 V1.2.1 [online]. European Telecommunications Standards Institute, 6. květen 2014 [cit. 2021-8-9]. Dostupné z: https://www.etsi.org/deliver/etsi_ts/102600_102699/102690/02.01.01_60/ts_102690v020101p.pdf

[73]ZSCALER. What is the Purdue Model for ICS Security? Zscaler.com [online]. [cit. 2021-8-9]. Dostupné z: https://www.zscaler.com/resources/security-terms-glossary/what-is-purdue-model-ics-security

[74]GREENFIELD, David. Is the Purdue Model Still Relevant? Automationworld.com [online]. 12 květen 2020 [cit. 2021-8-9]. Dostupné z: https://www.automationworld.com/factory/iiot/article/21132891/is-the-purdue-model-still-relevant

[75]ROSE, Scott, Oliver BORCHERT, Stu MITCHELL a Sean CONNELLY. Zero Trust Architecture: NIST Special Publication 800-207 [online]. srpen 2020 [cit. 2021-8-9]. Dostupné z: doi:https://doi.org/10.6028/NIST.SP.800-207

[76]BORMANN, C., M. ERSUE a A. KERANEN. RFC 7228: Terminology for Constrained-Node Networks [online]. IETF, květen 2014 [cit. 2021-8-9]. Dostupné z: doi:10.17487/RFC7228

[77]PELAEZ, Agustin. 9 IoT Operating Systems To Use in 2021. Ubidots.com [online]. 26. květen 2021 [cit. 2021-8-9]. Dostupné z: https://ubidots.com/blog/iot-operating-systems/

[78]MARTIN, James A. Jak správně řídit přístup k datům. Computerworld.cz [online]. 20. říjen 2019 [cit. 2021-8-9]. Dostupné z: https://computerworld.cz/securityworld/jak-spravne-ridit-pristup-k-datum-55644

[79]HARDT, D. RFC 6749: The OAuth 2.0 Authorization Framework [online]. IETF, říjen 2012 [cit. 2021-8-9]. Dostupné z: doi:10.17487/RFC6749

[80]USMAN, Muhammad. Lightweight Encryption for the Low Powered IoT Devices [online]. Chosun University , Republic of Korea, listopad 2020 [cit. 2021-8-9]. Dostupné z: https://www.researchgate.net/publication/346554993_Lightweight_Encryption_for_the_Low_Powered_IoT_Devices

[81]KURUNATHAN H., SEVERINO R., KOUBAA A. and TOVAR E., "IEEE 802.15.4e in a Nutshell: Survey and Performance Evaluation," in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 1989-2010, thirdquarter 2018 [cit. 2021-8-9]. Dostupné z: doi: 10.1109/COMST.2018.2800898.

[82]CLOUDFLARE. What is a denial-of-service (DoS) attack? Cloudflare.com [online]. [cit. 2021-8-9]. Dostupné z: https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/

[83]CLOUDFLARE. What is buffer overflow? Cloudflare.com [online]. [cit. 2021-8-9]. Dostupné z: https://www.cloudflare.com/learning/security/threats/buffer-overflow/

[84]CLOUDFLARE. What is a DDoS attack? Cloudflare.com [online]. [cit. 2021-8-9]. Dostupné z: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/

[85]CLOUDFLARE. What is the Internet Control Message Protocol (ICMP)? Cloudflare.com [online]. [cit. 2021-8-9]. Dostupné z: https://www.cloudflare.com/learning/ddos/glossary/internet-control-message-protocol-icmp/

[86]CLOUDFLARE. UDP flood attack. Cloudflare.com [online]. [cit. 2021-8-9]. Dostupné z: https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/

[87]NASTASE L., "Security in the Internet of Things: A Survey on Application Layer Protocols," *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, 2017 [cit. 2021-8-9]. pp. 659-666, Dostupné z: doi: 10.1109/CSCS.2017.101.

[88]KIRSTENS. Cross Site Scripting (XSS). Owasp.org [online]. [cit. 2021-8-9]. Dostupné z: https://owasp.org/www-community/attacks/xss/

[89]FORMÁNEK, David. Zranitelnosti typu injekce: XSS aneb cross-site scripting. Root.cz [online]. 2. duben 2019 [cit. 2021-8-9]. Dostupné z: https://www.root.cz/clanky/zranitelnosti-typu-injekce-xss-aneb-cross-site-scripting/

[90]FORMÁNEK, David. Zranitelnosti typu injekce: SQL injekce. Root.cz [online]. 11. říjen 2018 [cit. 2021-8-9]. Dostupné z: https://www.root.cz/clanky/zranitelnosti-typu-injekce-sql-injekce/

[91]HRANICKÝ, Jan. Lekce 3 - Jak se bránit proti SQL injection. Itnetwork.cz [online]. [cit. 2021-8-9]. Dostupné z: https://www.itnetwork.cz/php/bezpecnost/jak-se-branit-proti-sql-injection

[92]FIRST.ORG. Common Vulnerability Scoring System version 3.1: Specification Document: CVSS Version 3.1 Release. First.org [online]. The Forum of Incident Response and Security Teams, červen 2019 [cit. 2021-8-9]. Dostupné z: https://www.first.org/cvss/specification-document

19 Seznam použitých zkratek

AAA Zkratka – Autentizace, autorizace a správa účtů (accounting)
ABAC Řízení přístupu na základě atributů
ABS Aktivní bezpečnost vozidla
ADC Převodník analog-digital
AES Symetrický šifrovací standard
ALG Gateway aplikační vrstvy
AMI Pokročilá měřící infrastruktura
AP Přístupový bod
API Aplikační programovací rozhraní
APTEEN Protokol WSN sítí
ASA Adaptivní bezpečnostní zařízení (Cisco ASA)
AWS Cloudová platforma Amazon Web Services
BLE Technologie Bluetooth Low Energy
CA Certifikační autorita
CAD Počítačem podporované projektování
CAP Období komunikace v IEEE 802.15.4
CBM Údržba na základě stavu
CEH Certifikát etického hackera
CFP Období komunikace v IEEE 802.15.4
CIA Důvěrnost, Integrita a Dostupnost
CISC Procesor s komplexní instrukční sadou
CLI Rozhraní s příkazovou řádkou
CNA CVE Autorita
CoAP IoT protokol pro zasílání zpráv
CSMA/CA Bezkolizní síťový protokol
CSI Kamerové sériové rozhraní
CSP Zásady zabezpečení obsahu
CT Spotřebitelské technologie
CVE Databáze známých hrozeb a zranitelností
CVSS Metoda posouzení rizika
CWF Rámec pracovní rolí pro kybernetickou bezpečnost
DAC Převodník digital-analog
DAC (model) Nezávazné řízení přístupu
DCS Distribuované řídící systémy
DDoS Distribuované odmítnutí služby
DFD Diagram datového toku
DHCP Síťový protokol dynamické konfigurace
DNS Systém/protokol doménových jmen
DoS Odmítnutí služby
DREAD Model pro určení rizika hrozeb
DSS Systémy pro podporu rozhodování

DTLS Protokol zabezpečení vycházející z TLS
eMMC Embedded MultiMediaCard standard
EPROM semipermanentní typ paměti typu ROM-RAM
EEPROM semipermanentní typ paměti typu ROM-RAM
ERP Plánování podnikových zdrojů
ESN Síť senzorů prostředí
ETSI Evropský ústav pro telekomunikační normy
EUI Rozšířený unikátní identifikátor
FFD Zařízení s plnou funkcionalitou
FHS Standard souborového systému Linux
FIRST The Forum of Incident Response and Security Teams
GPIO Univerzální vstupní/výstupní pin
GUI Grafické uživatelské rozhraní
HEW Standard bezdrátové komunikace (High Efficiency Wireless)
HMI Rozhraní člověk-stroj
HSA Heterogenní systémová architektura
HTTP Hypertextový přenosový protokol
I2C Sériové sběrnice
I2S Zvuková sériová sběrnice
IAB Výbor Internet Architecture Board
IAM Management identit a přístupu
IaaS Infrastruktura jako služba
ICMP Internet Control Message Protocol
ICS Průmyslové řídící systémy
ICSP Protokol pro sériové programování mikrokontrolerů
ICT Informační a komunikační technologie
IDE Vývojové prostředí
IDLE Vývojové a vzdělávací prostředí
IEC Mezinárodní elektronická komise
IEEE Institut pro elektrotechnické a elektronické inženýrství
IETF Komise pro technickou stránku internetu
IFTTT Cloudová platforma If This Then That
IIC Konsorcium pro průmyslový internet
IMT-A Požadavky pro technologii 4G od společnosti ITU
IoT Internet věcí
IIoT Průmyslový internet věcí
IIRA Referenční architektura průmyslového internetu od IIC
ISA Instrukční sada
ISO Mezinárodní organizace pro standardizaci
ISOC Organizace Internet Society

ITT Intitute of Transformative Technologies
ITU The International Telecommunication Union
JTAG Standard pro testování
KSA Zkratka – Znalosti, dovednosti a schopnosti
LAN Lokální síť
LEACH Komunikační protokol WSN sítí
LLNs Ztrátové sítě s nízkým výkonem
LPWAN Rozlehlá síť s nízkým výkonem
LQI Indikátor kvality spojení
M2M Technologie pro komunikaci stroj-stroj
MAC Řízení přístupu k médiu
MAC (model) Mandatorní řízení přístupu
MCU Jednočipový počítač
MES Výrobní exekuční systém
MITM Útok Man-in-the-middle
MMF Vícevidové optické vlákno
MQTT IoT protokol pro zasílání zpráv
N.I.C.E. National Institute of Computer Education
NIST The National Institute of Standards and Technology
NSA Národní bezpečnostní agentura USA
OCF OpenFog Consortium a Open Connectivity Foundation
OSI Standard počítačových sítí
OT Operační Technologie
OWASP Open Web Application Security Project
PAN Osobní síť
PACS Systém pro správu obrazové dokumentace
PaaS Platforma jako služba
PCB Deska plošných spojů
PCM Pulzně kódovaná modulace
PDP Bod bezpečnostního rozhodování
PEP Bod vymáhání zásad
PID Proporční, derivační a integrální kontrolery
PID (Linux) ID procesu
PII Osobní údaje
PKI Infrastruktura veřejných klíčů
PLC Programovatelné logické kontrolery
PLM Management životního cyklu produktu
PWM Pulzní šířková modulace
QoS Rezervace a řízení datových toků
RAM Operační paměť
RBAC Řízení přístupu na základě rolí
RES Obnovitelné zdroje energie
REPL Jednoduché programovací prostředí
REST Architektura API
RFC Série dokumentů od společnosti ISOC

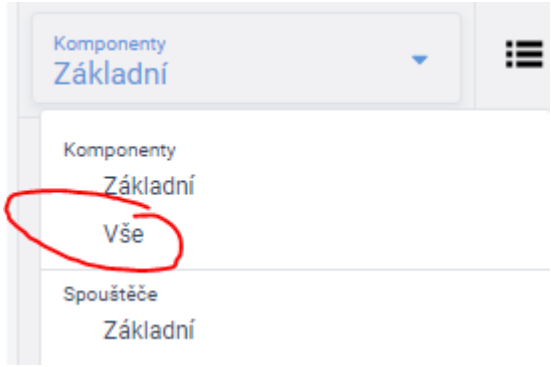
RFD Reduced Functional Device
RFID Identifikace na rádiové frekvenci
RISC Procesor s redukovanou instrukční sadou
ROM Paměť určená pouze ke čtení
RSA Šifra s veřejným klíčem
RSSI Indikátor síly přijatého signálu
SaaS Software jako služba
SCADA Systémy pro dozor, řízení a sběr dat
SCL (Arduino) Sériová hodinová linka
SDA (Arduino) Sériová datová linka
SECaaS Zabezpečení jako služba
SMF Jednovidové optické vlákno
SPI Sériové periferní rozhraní
SPIN Skupina protokolů ve WSN sítích
SQL Strukturovaný dotazovací jazyk
SSH Secure Shell protokol
SSL/TLS Protokol pro zabezpečení transportní vrstvy
STRIDE Model pro identifikaci hrozeb
TCP Přenosový protokol transportní vrstvy
TCP/IP Standard počítačových sítí
TDMA Přístup podle časového rozvrhu v protokolu LEACH
TEEN Komunikační protokol ve WSN sítích
TSA Dohoda o podmínkách poskytování služeb
UART Univerzální asynchronní přijímač/vysílač
UDP Síťový protokol transportní vrstvy
UML Grafický jazyk pro vizualizaci
UPnP Sada síťových protokolů pro automatickou konfiguraci zařízení v sítí
URL Jednotný lokátor zdroje
URI Jednotný identifikátor zdroje
USB Univerzální sériová sběrnice
VLAN Virtuální lokální síť
VPN Virtuální privátní síť
WAN Rozsáhlá počítačová síť
WSAN Bezdrátové senzorové a aktuátorové sítě
WSN Bezdrátové senzorové sítě
XML Rozšiřitelný značkovací jazyk
XMPP IoT protokol (Extensible Messaging and Presence Protocol)
XSS Útok Cross-site scripting
XXE Externí XML entita
ZTA Bezpečnostní architektura založená na nulové důvěře

Praktické zadání – Digitální teploměr

Komponenty

- 1x Arduino Uno R3
- 1x Potenciometr
- 1x Rezistor 330Ω
- 1x LCD 16x2
- 1x Teplotní senzor TMP36

V prostředí TinkerCAD nejsou v základu vidět všechny dostupné komponenty, což lze vyřešit jednoduše přepnutím do kategorie „**Vše**“.



Sériový monitor

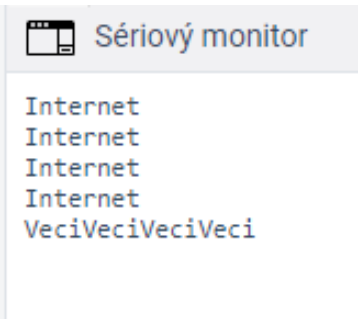
Než dojde na samotný teploměr, představíme si takzvaný sériový monitor. Sériový monitor je velice užitečná pomůcka při testování funkčnosti kódu. Lze ho využít pro zpětnou vazbu při provádění programu a taky pro sériovou komunikaci s ostatními zařízeními. Na fyzické desce Arduino mu odpovídají fyzické piny 0(RX) a 1(TX), které jsou monitorovány jiným čipem na desce Arduino, který je konvertuje pro přenos přes USB (pokud je připojené například do PC). V Arduino IDE monitoru odpovídá jediná ikona na pravé straně lišty. V prostředí TinkerCAD se sériový monitor nachází v dolní části okna s kódem.

Spuštění sériového monitoru se provádí příkazem **Serial.begin(9600);** (počáteční písmeno musí být velké!!!!). Hodnota 9600 je přenosová rychlost v baudech. Při použití fyzického zařízení a prostředí IDE se tato hodnota musí shodovat s hodnotou nastavenou v IDE, jinak se budou v monitoru ukazovat nesmysly. V TinkerCAD se přenosová hodnota na zařízení nenastavuje.

Základní příkazy pro zobrazení výstupu v monitoru jsou **Serial.println();** (malé L) a **Serial.print();**. Rozdíl je v tom, že Serial.println(); vypisuje vždy na nový

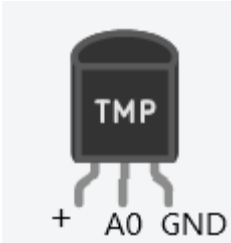
řádek a Serial.print(); pouze na jeden a za sebou. Výstupem může být string nebo například proměnná. Příklad výstupu sériového monitoru.

```
void setup()
{
  Serial.begin(9600); // zapnutí sériového monitoru
}
void loop()
{
     Serial.println("Internet");
     //Serial.print("Veci");
     delay(1000); // čekat 1 sek
}
```

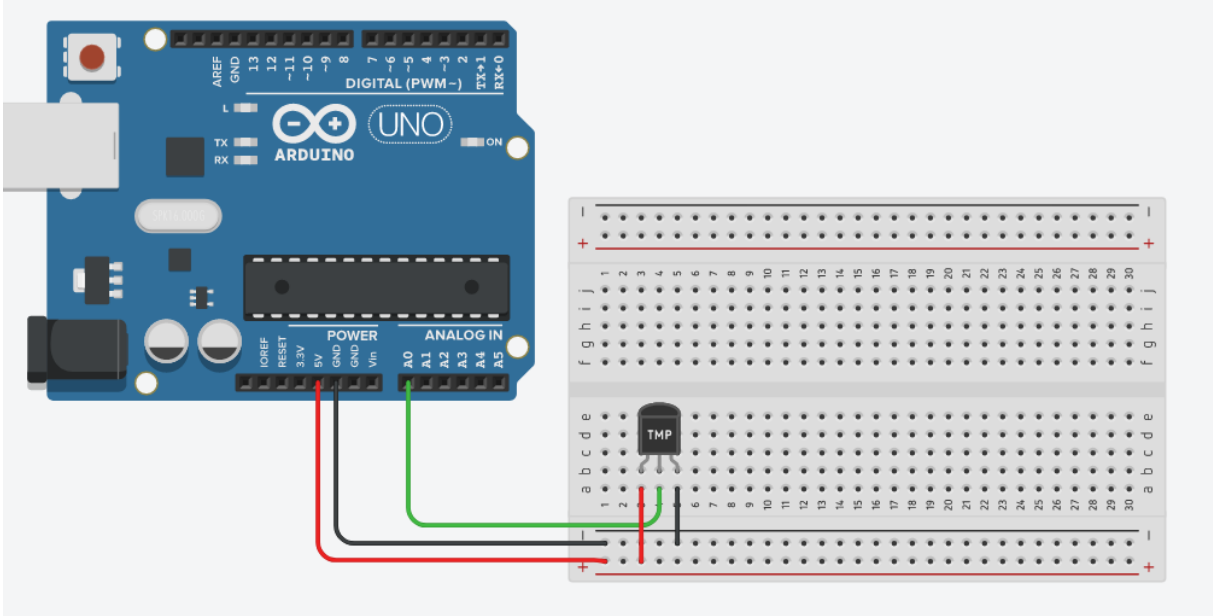


Měření teploty

K měření teploty je použit teplotní senzor s označením TMP36. Měřící rozsah tohoto teploměru je -40°C až 150°C (nebo -40°F až 302°F). Levý vývod je napájecí, z prostředního vývodu se získává naměřená hodnota a pravý vývod je pro uzemnění GND.



je Zapojení senzoru vypadá následovně:



Prvním krokem bude vytvoření proměnné kPinT, která bude držet označení pinu, na kterém je teploměr připojen, příkazem **const int kPinT = A0;** a aktivace sériového monitoru na 9600 baudů v části void.setup().

Druhým krokem bude samotné čtení hodnoty ze senzoru a její převod na teplotu ve stupních Celsia. Senzor vrací hodnoty v rozmezí **0-1023**, z čehož lze jednoduchým výpočtem získat hodnotu napětí, a to konkrétně tak, že získanou hodnotu vynásobíme **5.0** (referenční napětí 5 V) a vydělíme hodnotou **1024** (10 bit). V tabulce je vidět, že při teplotě 0°C senzor vrací napětí o velikosti **500 mV**, a že změna napětí o **10 mV** odpovídá změně teploty o 1°C.

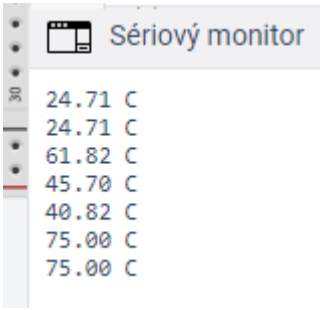
Teplota °C	Naměřené napětí
-10	400 mV
0	500 mV
10	600 mV
20	700 mV
30	800 mV
40	900 mV
50	1000 mV

Pro čtení hodnoty ze senzoru a její následný převod na teplotu ve °C proto vytvoříme jednoduchou funkci. Její zápis vypadá následovně:

```
float getTeplotaC()
// Deklarace funkce, vracet bude datový typ float
{
 int vstup = analogRead(kPinT);
// čtení z analogového pinu kPinT (A0) a uložení hodnoty (v
// rozmezí 0-102,) do proměnné vstup
 float napeti = (vstup * 5.0) / 1024;
// převod vstupní hodnoty na hodnoty napětí a vydělení
// hodnotou rozsahu 1024 (10 bitů), např. (153*5.0)/1024 =
// 0,747V (747 mV)-> mezi 20°C a 30°C
 return (napeti - 0.5) * 100;
// převod na °C odečtením 500mV (0°C = 0mV) a
// z rozlišení 10 mV/°C, (0,747 - 0,5)*100 = 24,7°C
 }
```

Teď už zbývá jen kód zkompletovat přidáním části **void loop()**, kde bude především samotný výpis naměřené teploty do sériového monitoru. Celý kód bude tedy vypadat následovně:

```
const int kPinT = A0;
void setup()
 {
 Serial.begin(9600);
 }
 void loop()
 {
      float teplotaC = getTeplotaC();
//Získání naměřené hodnoty pomocí volání funkce getTeplota();
      Serial.print(teplotaC);     //Výpis teploty do monitoru
      Serial.println(" C");       //Výpis stringu do monitoru
      delay(1000);                // čekání 1 sekundu (1000ms)
 }
float getTeplotaC()
{
 int vstup = analogRead(kPinT);
 float napeti = (vstup * 5.0) / 1024;
 return (napeti - 0.5) * 100;
 }
```



Pozn. V prostředí TinkerCAD lze změnu teploty simulovat tak, že klikneme na senzor TMP36 a posuvníkem se teplota manuálně nastaví.

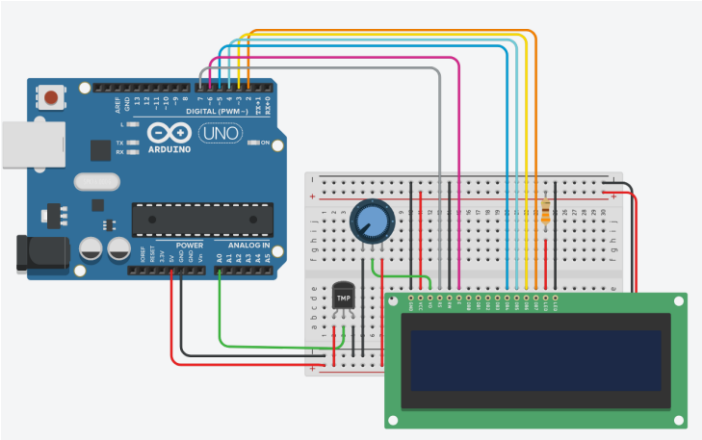
Zapojení LCD

Existuje několik typů LCD displejů, ale v tomto cvičení budeme využívat displej 16x2, který nabízí prostředí TinkerCAD. Označení 16x2 lze chápat jako 16 sloupců (0-15) a 2 řádky (0-1). LCD lze používat ve dvou režimech, a to v 4bitovém, který vyžaduje celkem 7 I/O pinů, nebo v 8bitovém, který vyžaduje 11 I/O pinů. Pro toto cvičení postačí 4bitový režim. V tomto cvičení si vystačíme se 6 piny na desce

Arduino, jelikož nebude nutné přepínat mezi režimy čtení a zápisu. Piny, které budou zapojené jsou:

- **+LED a -LED** – podsvícení displeje
- **VCC a GND** – vstup pro napájení a uzemnění
- **V0** – kontrast displeje (zapojený přes potenciometr)
- **RW** – v tomto cvičení připojený na zem
- **E** – umožní zápis na displej
- **DB7, DB6, DB5, DB4** nebo **DB3, DB2, DB1, DB0** – datové piny
- **RS** – slouží k výběru registru

Schéma zapojení může tedy vypadat následovně (červená barva reprezentuje 5V a černá GND):



K používání LCD displeje je potřeba knihovna s názvem **LiquidCrystal.h**. V prostředí TinkerCAD ji lze jednoduše přidat z nabídky (prostřední ikona v okně s kódem). Jako u ostatních komponent je nejdříve potřeba nakonfigurovat příslušené piny, které LCD displeji náleží. To se provádí příkazem ***LiquidCrystal <název_displeje>(čísla pinů)***. Dalším krokem je displej, podobně jako sériový monitor, potřeba aktivovat příkazem ***<název_displeje>.begin(počet_sloupců, počet_řádků)***. Následný výpis na LCD se provádí podobně, jako výpis do sériového monitoru, příkazem ***<název_displeje>.print()***. K nastavení pozice na displeji je možné použít příkaz ***<název_displeje>.setCursor(sloupec,řádek)***. Konkrétní kód pro náš teploměr by vypadal následovně:

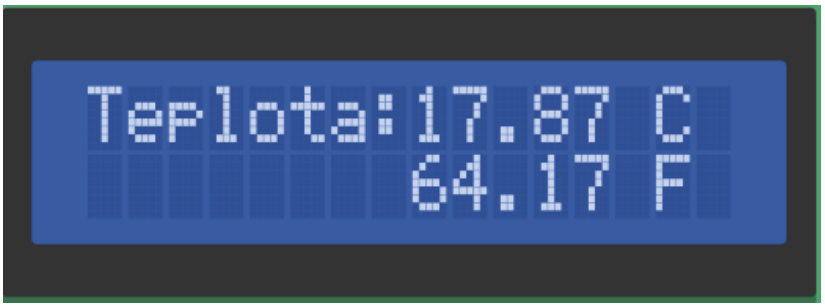
```
#include <LiquidCrystal.h> // knihovna pro LCD displej
const int kPinT = A0;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
// Inicializace pinů LCD displeje (RS, E, DB4, DB5, DB6, DB7)
void setup()
 {
      //Serial.begin(9600);
      lcd.begin(16,2); //zapnutí LCD displeje
 }

 void loop()
 {
     float teplotaC = getTeplotaC();
     //Serial.print(teplotaC);
     //Serial.println(" C");
     lcd.clear();               //vymaže obsah displeje
     lcd.setCursor(0,0);        //nastavení kurzoru na 1s1ř
     lcd.print(teplotaC);       //výpis na displej
     lcd.print(" C");
     delay(1000);
 }
float getTeplotaC()
{
 int vstup = analogRead(kPinT);
 float napeti = (vstup * 5.0) / 1024;
 return (napeti - 0.5) * 100;
 }
```

Rozšiřující úkoly

1. Rozšiřte kód o zobrazování teploty ve stupních Fahrenheita (viz obrázek).

   $$teplota\ °F = \frac{9}{5}(teplota\ °C) + 32$$

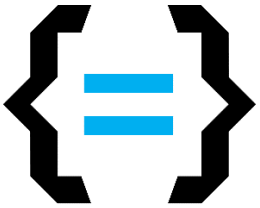


2. Rozšiřte kód tak, aby vypisoval hodnotu napětí v mV do sériového monitoru při každém měření teploty.
3. Rozšiřte kód o záznam nejvyšší a nejnižší naměřené teploty s výpisem změn do sériového monitoru.

Příloha č. 2 – Šablona pro dokumentaci

Univerzita Hradec Králové

Fakulta informatiky a managementu



Téma projektu

Internet věcí (IoT)

Autor: Jméno, příjmení
Studijní obor: Studijní obor

Hradec Králové měsíc rok

Obsah

Úvod a cíl práce

Stručně popsat řešený projekt a problém, kterého se týká + zamýšlená podoba a smysl řešení

Definice problému

Podrobnější popis řešeného problému. Odpověď na 3 základní otázky KDO?, CO? A PROČ?:

- Jaký je problém nebo potřeba?
- Kdo má daný problém nebo potřebu?
- Proč je důležité problém řešit?

Tipy pro tento krok:

- Nefixovat se k prvnímu nápadu
- Generovat nápady pomocí myšlenkové mapy

Prvotní výzkum – rešerše zdrojů

Sběr informací, průzkum již existujících řešení, rešerše souvisejících zdrojů apod.

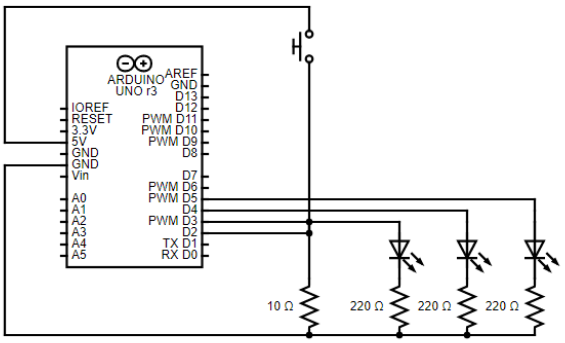
Určení požadavků

Definice KONKRÉTNÍCH požadavků na navrhované řešení. Lze využít poznatky z předchozí části, kdy došlo k průzkumu existujících řešení.

Vypracování návrhu řešení

Tato část bude obsahovat veškerý obsah, schémata, modely, diagramy týkající se návrhu řešení + krátký popis u každé části:

- **Skeče a náčrty (volitelné)**
  - Pokud někdo preferuje grafickou reprezentaci nápadů a myšlenek
- **Elektronické schéma zapojení**
  - Seznam použitých komponent + jednoduché schéma + krátký popis
  - Schéma bude obsahovat blok reprezentující desku Arduino/RasPi s výstupy, na kterých budou připojené příslušné periferie a komponenty (viz příklad)



  - K tvorbě je možné použít některý z dostupných editorů, např:
    - https://www.circuit-diagram.org/editor/

        - https://www.partsim.com/simulator
        - https://easyeda.com/page/download
- **Seznam operací**
    - Seznam kroků, které bude řešení vykonávat k provedení požadované funkce, např.
        - Nastavení hesla
        - Ověření zadaného hesla
        - Otevření dveří
        - Zaznamenání přístupu
        - Zamítnutí přístupu
        - Zablokování konzole / dalších akcí
        - Apod.
- **Vývojové diagramy**
    - Vývojový diagram, který bude znázorňovat fungování celého řešení + rozpad na jednotlivé procesy, které jsou nezbytné pro vykonání operace
    - Seznámení se se základy tvorby vývojových diagramů + příklady
        - https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial
    - K tvorbě diagramu je možné využít online nástroj Draw.io (nebo jiný)
        - https://app.diagrams.net/
- **Sekvenční diagramy**
    - Jednoduchý sekvenční diagram, který bude znázorňovat posloupnost komunikace jednotlivých prvků systému (vnějších i vnitřních)
    - Stručný a jednoduchý popis sekvenčního diagramu a jeho struktury
        - https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/
    - K tvorbě diagramu je možné využít následující online nástroj nebo software, jako Enterprise Architect a StarUML
        - https://online.visual-paradigm.com/app/diagrams/#diagram:proj=0&type=SequenceDiagram

- **Business model**
    - Zamyšlení se nad využitím a podobou produktu z pohledu podnikání
    - Tvorba tzv. Business canvas model
        - https://canvanizer.com/new/lean-canvas

Související laby v kurzu Cisco: Connected things -

- 6.3.1.4 – Práce s IFTTT
- 6.3.1.7 – Tvorba flowchartu
- 6.3.1.8 – Tvorba elektrického schématu zapojení

- 6.3.1.9 – Tvorba sekvenčního diagramu
- 6.3.2.3 – Prototyp a testování řešení
- 6.4.1.7 – Business model diagram

Prototyp a testování

- Tvorba prvního funkčního prototypu
    - Obrázek zapojení z prostředí TinkerCAD nebo podobného softwaru
    - PŘILOŽTE ODKAZ ŘEŠENÍ V PROSTŘEDÍ TINKERCAD (pokud používáte TinkerCAD)
- Dokumentace a otestování veškeré požadované funkcionality
- Dokumentace veškerých změn
- Možná vylepšení a „nice-to-have“ vlastnosti

Modelování a posouzení hrozeb

Této problematice se věnuje celá 2. polovina předmětu IoT (tj. kapitoly 7 až 12), kde jsou uvedeny příklady běžných a častých zranitelností/hrozeb a zároveň i proces, modely a nástroje, které by vám mohli pomoci. Jak je v přednáškách uvedeno, proces modelování hrozeb se skládá celkem z pěti kroků. V tomto dokumentu se zkuste zamyslet především nad těmito třemi kroky a krátce popište.

- Identifikace bezpečnostních cílů
- Identifikace hrozeb a zranitelností
    - Můžete si pomoci modely STRIDE a DREAD
- Návrh opatření

Závěr

Stručné shrnutí průběhu návrhu a tvorby řešení, popis výsledné podoby řešení, možná budoucí vylepšení.

Zdroje

Citace použitých zdrojů. Pokud se inspirujete projekty jiných uživatelů v prostředí TinkerCAD, uveďte odkaz.

Kód (i s komentáři)

UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Akademický rok: 2019/2020

Studijní program: Informační management
Forma studia: Prezenční
Specializace/kombinace: Informační management (im2-p)

Podklad pro zadání DIPLOMOVÉ práce studenta

Jméno a příjmení: **Bc. Patrik Urbaník**
Osobní číslo: **I1900319**
Adresa: K Hájku 1713, Nová Paka, 50901 Nová Paka, Česká republika

Téma práce: Program vzdělávání v oblasti Internetu věcí
Téma práce anglicky: The education program in the field of Internet of Things

Vedoucí práce: Ing. Pavel Blažek, Ph.D.
Katedra informačních technologií

Zásady pro vypracování:

Cíl práce:
Cílem diplomové práce je vypracovat materiály a praktické podklady, včetně kurzu v Blackboard, k předmětu Internet věcí (IoT). Obsahem práce bude teoretický základ z oblastí týkajících se IoT, jako základy elektroniky, počítačové sítě a programování. Praktická část se bude zabývat vypracováním plánu výuky na celý semester, tzn. rozdělení témat na jednotlivé týdny a vypracování praktických cvičení k danému tématu.
Osnova:
Úvod
Literární rešerše
Metodologie, způsob řešení
Analýza IoT technologií
Návrh osnovy předmětu a jeho modulů
Tvorba učebních a zkušebních textů s podporou výuky v prostředí LMS
Shrnutí výsledků
Závěry a doporučení

Seznam doporučené literatury:

1, Cisco kurz, IoT Fundamentals, (online)
2, Cisco kurz, IoT Fundamentals: Hackathon Playbook (online)
3, Malý, Martin. Hradla, volty, jednočipy: Úvod do bastlení. Praha: CZ.NIC, z. s. p. o., 2017. ISBN 978-80-88168-26-3
4, Massimo Banzi and Michael Shiloh; Getting Started with Arduino. Sebastopol: Maker Media 2015, ISBN 978-1-4493-6333-8
5, IoT fundamentals: networking technologies, protocols, and use cases for the internet of things / David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry Indianapolis: Cisco Press, [2017] ?2017, ISBN 978-1-58714-456-1 (brožováno)
6, GerryHowser, Computer Networks and the Internet: A Hands-On Approach; Springer 2020; ISBN 978-3-030-34495-5 (eBook: ISBN 978-3-030-34496-2)
7, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry; IoT fundamentals: networking technologies, protocols, and use cases for the internet of things; Indianapolis: Cisco Press 2017, ISBN 978-1-58714-456-1

8, Matúš Selecký; Arduino: Uživatelská příručka; Computer Press, 2016; ISBN: 978-80-251-4840-2

Podpis studenta: 

Datum: 16.8.2021

Podpis vedoucího práce: 

Datum: 16.8.2021