

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Dálkové ovládání domácností pomocí mikropočítače

Jan Čech

© 2018 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jan Čech

Informatika

Název práce

Dálkové ovládání domácností pomocí mikropočítače

Název anglicky

Remote control of households using a microcomputer

Cíle práce

Bakalářská práce je tematicky zaměřena na problematiku dálkové ovládání a řízení domácností pomocí mikropočítačů skrze mobilní zařízení.

Hlavním cílem práce je identifikovat možnosti využití mikropočítačů řízených prostřednictvím mobilních zařízení, pro vzdálené ovládání domácnosti a dále navržení a realizování konkrétní aplikace pro domácnost.

Díličí cíle jsou:

- vytvořit přehled možností využití mikropočítačů ovládaných dálkově z mobilních zařízení v domácnostech.
- vytvoření aplikace s mikropočítačem pro domácnost, která bude dálkově ovládaná z mobilního zařízení a následně celé toto řešení demonstrovat

Metodika

Metodika řešení problematiky bakalářské práce je založena na studiu odborných informačních zdrojů. Praktická část práce je zaměřena na vytvoření aplikace s mikropočítačem pro domácnost, která bude dálkově ovládaná z mobilního zařízení. Na základě syntézy teoretických poznatků a realizace aplikace pro domácnost budou formulovány závěry bakalářské práce.

Doporučený rozsah práce

35-40 stran

Klíčová slova

chytrá domácnost, mikropočítače, dálkové ovládání, Android, Arduino

Doporučené zdroje informací

Arduino Android Blueprints, Schwartz, Marco; Buttigieg, Stefan Packt Publishing 2014, ISBN: ISBN number:9781784390389, ISBN number:9781784391683

Arduino Home Automation Projects, Schwartz, Marco Packt Publishing 2014, ISBN: ISBN number:9781783986064, ISBN number:9781783986071

Beginning Android Application Development, Lee, Wei-Meng John Wiley & Sons, , Incorporated 2011, ISBN: ISBN number:9781118017111, ISBN number:9781118087299, EDICE: Wrox beginning guides.

Professional Android 4 Application Development, Meier, Reto Wiley 2012, ISBN: ISBN number:9781118102275, ISBN number:9781118223857

Předběžný termín obhajoby

2017/18 LS – PEF

Vedoucí práce

Ing. Jiří Brožek, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 23. 2. 2018

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 23. 2. 2018

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 01. 03. 2018

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Dálkové ovládání domácností pomocí mikropočítače" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 8.3.2018

Poděkování

Rád bych touto cestou poděkoval Panu Ing. Jiří Brožkovi, Ph.D. za vedení bakalářské práce.

Dálkové ovládání domácností pomocí mikropočítače

Abstrakt

Tato bakalářská práce se zabývá možnostmi využití mikropočítačů nebo také mikroprocesorů k ovládání zařízení (zapínání / vypínání) v domácnostech. Pro dálkové ovládání mikropočítače bylo využito mobilních zařízení (mobilní telefon, tablet) s operačním systémem Android, pro který byla navržena a naprogramována aplikace. Komunikace mezi mikropočítačem NodeMCU a mobilním zařízením probíhá prostřednictvím komunikačního protokolu IEEE 802.11 - WiFi. Součástí práce je i vytvořená ukázkové aplikace.

Klíčová slova: NodeMCU, WIFI, chytrá domácnost, dálkové ovládání, mikropočítače, Android, Arduino, mikrokontroler, ESP8266

Remote control of households using a microcomputer

Abstract

This bachelor thesis deals with the possibilities of using a microcontroller or microprocessors to control devices (switching them on / off) in households. Android OS Mobile device (phone or tablet) with custom designed and programmed app was used for controlling the micro PC device. The communication between the NodeMCU microcontroller and the mobile device is via the IEEE 802.11 - WiFi communication protocol. Creating of the sample application is also part of the work.

Keywords: NodeMCU, WIFI, smart home, remote control, microcomputer, Android, Arduino, microcontroler, ESP8266

Obsah

| | |
|--|-----------|
| 1. Úvod..... | 10 |
| 2. Cíl práce a metodika | 11 |
| 2.1. Cíl práce | 11 |
| 2.2. Metodika | 11 |
| 3. Teoretická východiska | 12 |
| 3.1. Dálkového ovládání v domácnostech..... | 12 |
| 3.2. Nasazení dálkového ovládání..... | 13 |
| 3.3. Technická omezení použití dálkového ovládání | 14 |
| 3.3.1. Dosah a oblast pokrytí signálem..... | 14 |
| 3.3.2. Umístění přijímacího zařízení..... | 15 |
| 3.4. Mikroprocesory | 19 |
| 3.4.1. Arduino | 21 |
| 3.4.2. NodeMCU..... | 23 |
| 3.4.3. Parallax Basic Stamp BS2-IC | 25 |
| 3.4.4. BX-24..... | 27 |
| 3.4.5. STM32 | 28 |
| 3.4.6. Raspberry Pi..... | 30 |
| 3.4.7. PLC | 32 |
| 3.4.8. Zhodnocení mikrokontrolerů | 34 |
| 3.5. Bezdrátová komunikace | 36 |
| 3.6. Popis NodeMCU | 38 |
| 3.6.1. NodeMCU Dev Kit v1.0..... | 38 |
| 3.6.2. Programovací prostředí pro NodeMCU | 44 |
| 3.7. Android | 47 |
| 4. Vlastní práce | 51 |
| 4.1. Dálkově ovládaný modul s mikroprocesorem..... | 51 |
| 4.1.1. Schéma zapojení | 52 |
| 4.2. Spínací relé..... | 56 |
| 4.3. Napájení NodeMCU a spínacích relé..... | 58 |
| 4.4. Řídící program pro mikrokontroler NodeMCU | 60 |
| 4.4.1. Struktura programu | 60 |
| 4.4.2. Inicializace základních prvků | 61 |
| 4.4.3. Obsluha požadavků..... | 62 |
| 4.4.4. Funkce na ovládání výstupů | 63 |
| 4.4.5. Spuštění WIFI..... | 63 |
| 4.5. Ovládací aplikace pro operační systém Android | 65 |

| | | |
|------------|---|-----------|
| 4.5.1. | Ovládací obrazovka | 65 |
| 4.5.2. | Design | 66 |
| 4.5.3. | Bezdrátová komunikace s NodeMCU | 67 |
| 4.5.4. | Odeslání požadavku na NodeMCU | 68 |
| 4.5.5. | Změna hesla mobilní aplikace | 69 |
| 4.5.6. | Struktura programu | 71 |
| 5. | Závěr..... | 72 |
| 6. | Seznam použitých zdrojů | 73 |
| 7. | Seznam obrázků | 78 |
| 8. | Seznam tabulek | 79 |
| 9. | Seznam použitých zkratk..... | 79 |
| 10. | Přílohy..... | 82 |
| | Příloha I. Zdrojový kód pro NodeMCU..... | 82 |
| | Příloha II. Zdrojový kód aplikace pro OS Android..... | 84 |

1. Úvod

Jedna z velmi častých činností v domácnostech je zapínání nebo vypínání různých zařízení a spotřebičů. Cílem této bakalářské práce je identifikace možností využití mikroprocesorů pro dálkové ovládání – zapínání / vypínání zařízení v domácnostech.

Pro ovládání jednotlivých zařízení domácnostech jsem vybral mikroprocesor NodeMCU, pro který jsem navrhl a naprogramoval software. Vlastní dálkové ovládání mikroprocesoru jsem navrhl a vytvořil v mobilní aplikaci běžící nad operačním systémem Android, který je současné době velice rozšířen v mobilních zařízeních – mobilní telefony a tablety. Aplikace pro mobilní zařízení bude komunikovat s NodeMCU prostřednictvím komunikačního protokolu WiFi, standard IEEE 802.11.

Součástí této bakalářské práce je i praktická část, která slouží k demonstrování navržených funkcionalit dálkového ovládání zařízení v domácnosti skrze mobilní aplikaci instalované ať už mobilních telefonech či tabletech.

2. Cíl práce a metodika

2.1. Cíl práce

Bakalářská práce je tematicky zaměřená na problematiku dálkového ovládání a řízení domácností pomocí mikropočítačů skrze mobilní zařízení.

Hlavním cílem práce je identifikovat možnosti využití mikropočítačů řízených prostřednictvím mobilních zařízení, pro vzdálené ovládání domácnosti a dále navržení a realizování konkrétní aplikace pro domácnost.

Dílčí cíle jsou:

- vytvořit přehled možností využití mikropočítačů ovládaných dálkově z mobilních zařízení v domácnostech.
- vytvoření aplikace s mikropočítačem pro domácnost, která bude dálkově ovládaná z mobilního zařízení a následně celé toto řešení demonstrovat.

2.2. Metodika

Metodika řešení problematiky bakalářské práce je založena na studiu odborných informačních zdrojů. Praktická část práce je zaměřena na vytvoření aplikace s mikropočítačem pro domácnost, která bude dálkově ovládaná z mobilního zařízení.

Na základě zhodnocení možností jednotlivých mikropočítačů – mikrokontrolerů bude vybrán ten, který nejlépe umožní realizovat dálkově ovládání spínání síťového napětí pro externí domácí zařízení. Při hodnocení vlastností mikropočítače bude důležité mimo jiné posoudit dostupnost vhodných periférií prostřednictvím, kterých bude možné ovládat, ale i regulovat externí zařízení v domácnosti.

Na základě syntézy teoretických poznatků a realizace aplikace pro domácnost budou formulovány závěry bakalářské práce.

3. Teoretická východiska

3.1. Dálkového ovládání v domácnostech

Automatizace domů je téma, které diskutováno mnoho let. Zahrnuje vše, co si můžete představit, abyste ovládali a automatizovali svůj domov. Nejrozšířenějším příkladem je alarmový systém vašeho domova. Existuje nespočet zařízení, které jsou k dispozici pro domácí automatizaci. Můžete si koupit kompletní domácí automatizační zařízení z mnoha obchodů a nechat nainstalovat ve vašem domě. Nicméně, mnoho z těchto systémů je velmi drahých a je nemožné je přizpůsobit vašim potřebám a také mohou mít zastaralé uživatelské rozhraní. Na druhou stranu máme platformy na bázi mikroprocesorů např. Arduino, NodeMCU a další, které nám umožní si vytvořit vlastní zařízení, kterým automatizujeme naši domácnost.¹

Naše domácnosti, domy, kanceláře jsou plně zařízení a spotřebičů, které se musejí velice často a opakovaně zapínat a vypínat. Člověka obtěžuje to, že se musí zvednout a dojít k zařízení, zásuvce, vypínači světel zapnout je nebo je vypnout a zase se vrátit na původní místo například na své pracoviště. Jedna z možností, jak se zbavit těchto často se opakujících a obtěžujících činností je instalace zařízení, které bude dálkově ovládáno, a které bude zapínat a vypínat požadovaná zařízení, osvětlení nebo spotřebiče.

Zde se nabízí, jako jedno z možných řešení použít mobilního zařízení, které má člověk skoro pořád u sebe, a kterému umožní jednoduché dálkové ovládání koncových zařízení. Tímto mobilním zařízením je mobilní telefon, a v domácnosti nebo v kanceláři je to také často tablet.

Ve většině mobilních telefonů a tabletech je nainstalován operační systém Android nebo operační systém iOS, který je používán v mobilních telefonech a tabletech společnosti

¹ SCHWARTZ Marko, Arduino Home Automation Projects, Packt Publishing Lth. 2014, 1 s.,[cit. 2018-1-02], ISDN: 9781783986071

Apple. Dle průzkumu společnosti Gartner měl v roce 2017 operační systém Android největší podíl na trhu a to 85,9 %, operační systém iOS má 15% podíl na trhu.²

Oba tyto systémy nám umožňují naprogramovat aplikace, které budou schopné prostřednictvím bezdrátové komunikace WiFi (standard IEEE 802.11)³ nebo prostřednictvím bezdrátové komunikace Bluetooth⁴ dálkově ovládat zařízení, která budou zapínat nebo vypínat osvětlení, zásuvky a zařízení.

Dálkově ovládané koncové zařízení, které bude ovládat prostřednictvím relé nebo spínacích obvodů (např. tranzistory, tyristory, triaky, atd.) k němu připojené spotřebiče, osvětlení, řídicí systémy a atd. (dále jen zařízení) může být navrženo a realizováno na základě různých řídicích systémů. Dálkově ovládané koncové zařízení může být mimo jiné realizováno za použití:

- mikroprocesoru, resp. mikrokontroleru, ke kterému jsou připojena komunikační a ovládací rozhraní (WiFi modul, Bluetooth modul, resp. relé),
- jednočipového počítače, ke kterému jsou připojena komunikační a ovládací rozhraní (WiFi modul, Bluetooth modul, resp. relé),
- nebo programovatelných logických automatů – PLC.

3.2. Nasazení dálkového ovládání

Dálkově ovládaným koncovým zařízením můžeme v bytě, domě nebo v kanceláři (dále jen v domácnosti) zapínat nebo vypínat (dále jen ovládat) mimo jiné následující zařízení:

- osvětlení – vnitřní, venkovní,
- zásuvky – jedno fázové i tři fázové,
- klimatizace, ventilace, topení,
- žaluzie, venkovní markýzy, rolety,

² GARTNER, *Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017* [online], Gartner Feb 22 2018 [cit. 2018-2-24], Dostupné z: <https://www.gartner.com/newsroom/id/3859963>

³ IEEE Standard Association, *IEEE Standard* [online], IEEE Standard Association 2016, [cit. 2018-2-02], dostupné z: <https://standards.ieee.org/findstds/standard/802.11-2016.html>

⁴ BLUETOOTH SIG Proprietary, *Bluetooth Core Specification v 5.0*, 166 s. Bluetooth SIG Proprietary Dec 06 2016, Dostupné z: <https://www.bluetooth.com/specifications/bluetooth-core-specification>

- vjezdová nebo garážová vrata,
- filtry bazénů a vířivek, čerpadla solárního ohřevu bazénů,
- zavlažování nebo osvětlení zahrady,
- v kancelářích je to zejména kávovar,
- atd.

Výše uvedený výčet oblastí, kde je možné dálkově zapínat a vypínat zařízení je jen orientační a dává čtenáři základní pohled na danou oblast.

V praktické části bude navržen a realizován dálkově ovládaný modul do něhož bude možné připojit spotřebič prostřednictvím jednofázového pohyblivého přívodu.

3.3. Technická omezení použití dálkového ovládání

Jedna z důležitých věcí, která musí být vzata v úvahu při instalaci dálkově ovládaného zařízení jsou technické limity navrženého řešení jako je dosah a pokrytí signálem WiFi nebo Bluetooth.

3.3.1. Dosah a oblast pokrytí signálem

Dosah a pokrytí signálem zaleží na typu zvolené antény na straně přijímacího zařízení, protože na straně mobilního zařízení je anténa, vyzařovací charakteristiky a vysílací výkon dán výrobcem daného zařízení – mobilního telefonu, tabletu.

Na straně přijímacího zařízení si můžeme sami vybrat, který komunikační modul použijeme, resp. jaký typ bezdrátové komunikace (WiFi, Bluetooth, GSM) použijeme pro naše zařízení a s jakým typem antény.

Antény dělíme na:

- všesměrové

Všesměrové antény jsou univerzální antény, u nichž dosah a oblast pokrytí signálem záleží na zisku antény (Obrázek 1). Čím je zisk antény větší, tím je větší její dosah v kolmém směru na anténu, ale snižuje se úhel vyzařování ve vertikálním směru.

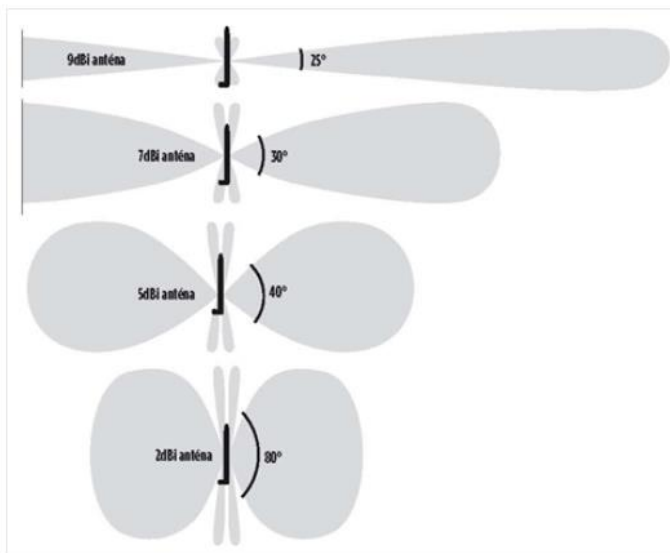
- sektorové

Anténa vyzařuje do určitého směru a dle typu antény je dán úhel vyzařování.

- směrové

Směrové antény vyzařují jen v úzkém úhlu několika stupňů.

Obrázek 1 Vyzařovací diagram všesměrové antény



Zdroj: ŽIVĚ, PROKOP Mirek, *Wi-Fi: Jak si zajistit velké pokrytí, rychlost a silný signál*, [online], ŽIVĚ 5. února 2014, [cit. 2018-1-20], Dostupné z: <https://www.zive.cz/clanky/wi-fi-jak-si-zajistit-velke-pokryti-rychlost-a-silny-signal/anteny-a-jejich-nastaveni-deformace-a-ztrata-signalu-mereni/sc-3-a-172347-ch-90933/><https://www.zive.cz/clanky/wi-fi-jak-si-zajistit-velke-pokryti-rychlost-a-silny-signal/anteny-a-jejich-nastaveni-deformace-a-ztrata-signalu-mereni/sc-3-a-172347-ch-90933/>

Pro dálkové ovládání koncových zařízení jsou nejvhodnější všesměrové antény, které nám pokryjí signálem požadovanou oblast bez toho, že bychom museli významněji řešit směr, odkud budeme chtít ovládat koncové zařízení.

3.3.2. Umístění přijímacího zařízení

Umístění přijímacího zařízení, které bude dálkově ovládat zařízení v domácnosti, je do značné míry ovlivněno tím:

- jaké zařízení budeme ovládat, je-li zařízení připojeno pohyblivým přívodem, je přenosné, nebo naopak je trvale instalováno na jednom místě,
- kde a jak je ovládané zařízení v domácnosti umístěno.

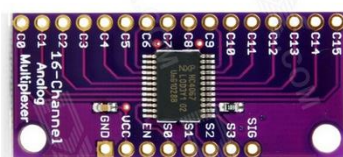
Přijímací zařízení s mikrokontrolerem nemusí být umístěno přímo u ovládaného zařízení, může být umístěno na vhodném místě, a to jak z pohledu šíření signálu nebo z pohledu připojení dálkově ovládaného mikrokontroleru k napájení.

Mikrokontrolery a jejich externí rozhraní umožňují připojit např. spínací relé prostřednictvím kabelů v bezprostřední blízkosti ovládaného zařízení.

Pokud potřebujeme připojit a ovládat více zařízení, než je počet vstupně výstupních pinů mikroprocesoru, mikrokontroleru, pak máme následující možnosti:

- ovládaná relé připojíme prostřednictvím multiplexoru například pomocí analogového multiplexeru CD74HC4067⁵ (Obrázek 2). CD74HC4067 je integrovaný obvod, který umožňuje najednou připojit relé na 16 výstupů a přepínat mezi nimi pomocí čtyř ovládacích pinů. Na řídicím mikroprocesoru, tak ušetříme 11 pinů.

Obrázek 2 CD74HC4067 Analog / Digital MUX



Zdroj: DX DEALEXTREME, CD74HC4067 Analog / Digital MUX [online], [cit. 2018-2-14], Dostupné z: <http://www.dx.com/p/cd74hc4067-analog-digital-mux-breakout-for-arduino-385746>

- ovládaná zařízení připojíme prostřednictvím společné sběrnice např. I2C. I2C bus je zkratka, která vznikla z IIC bus, tedy Internal-Integrated-Circuit Bus. Sběrnice I2C umožňuje připojit, resp. adresovat až 128 zařízení.

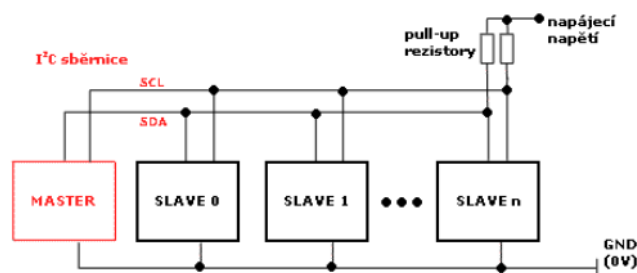
Hlavní výhodou je, že obousměrný přenos probíhá pouze po dvou vodičích - "data SDA (serial data)" a "hodiny SCL (serial clock)". To především u

⁵ TEXAX INSTRUMENT, *CD74HC4067 High Speed CMOS 16-Channel Analog Multiplexer/Demultiplexer* [online], Texas Instrument 2017, [cit.2018-2-17], Dostupné z: <http://www.ti.com/product/CD74HC4067?keyMatch=CD74hc4067&tisearch=Search--Everything>

mikrokontrolerů výrazně optimalizuje nároky na počet vstupně-výstupních pinů a celkově zjednodušuje výsledné zapojení. Na jednu sběrnici může být připojeno více integrovaných obvodů. V základní verzi jsou obvody adresovány 7bitově a v rozšířené verzi 10bitově. To umožňuje připojení 128 respektive 1024 čipů s různou adresou na jednu společnou sběrnici. Přenosová rychlost sběrnice je pro většinu aplikací dostatečná i v základní verzi, kde je frekvence hodin 100kHz. Ve vylepšených verzích to může být 400kHz nebo 1MHz.⁶

Možné propojení jednotlivých integrovaných obvodů ukazuje následující obrázek (Obrázek 3).

Obrázek 3 Možné propojení jednotlivých integrovaných obvodů na I2C sběrnici



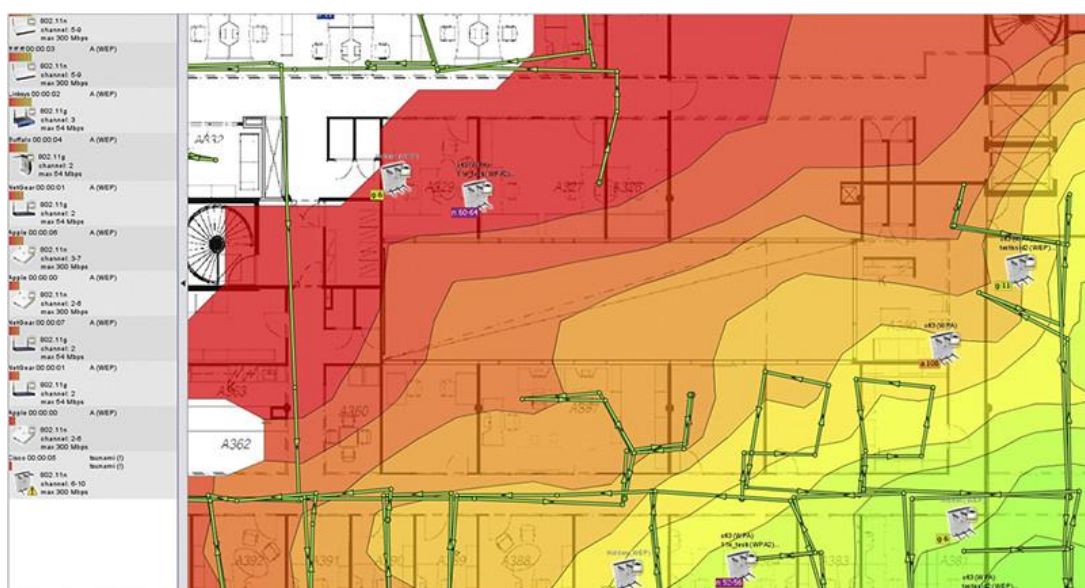
Zdroj: HW SERVER, *Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroleru PIC16F877* [online], HW server s.r.o. 2000, [cit. 20181-14], Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeeprom-24lc256>

Pro ověření vhodnosti umístění dálkově ovládaného přijímacího zařízení s mikroprocesorem je vhodné použít analyzátor úrovně signálu v domácnosti nebo zahradě.

⁶ HW SERVER, *Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroleru PIC16F877* [online], HW server s.r.o. 2000, [cit. 20181-14], Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeeprom-24lc256>

Jeden z možných analyzátorů měření úrovně signálu v prostoru je např. Ekahau Heat Mapper, který do předem připraveného plánu zanesete polohy dálkové ovládaného zařízení a poté polohy notebooku, na kterém probíhá měření signálu. Výsledkem je mapa s interpolovanými úrovněmi signálu ve všech místech požadovaném okolí. Po provedené analýze úrovně signálu se na výsledné mapě projeví fyzikální vlastnosti překážek, které jsou mezi dálkově ovládaným zařízením a mobilním zařízením. Jedná se zejména o vliv materiálů použitých na konstrukci zdí, přepážek domů a bytů.⁷

Obrázek 4 Mapu pokrytí Wi-Fi vytvořená pomocí programu Ekahau Heat Mapper



Zdroj: EKAHAU WIRELESS DESIGN, Ekahau HeatMapper - Map Wi-Fi Coverage and make it faster [online], Ekahau Wireless Design 2018, [cit. 20182-2], Dostupné z: <https://www.ekahau.com/products/heatmapper/overview/>

Při umístování dálkově ovládaných zařízení musíme brát v úvahu natočení antén a povahu prostředí, ve kterém bude signál šířen. Uvnitř budov jde hlavně o materiály, ze

⁷ ŽIVĚ, PROKOP Mirek, *Wi-Fi: Jak si zajistit velké pokrytí, rychlost a silný signál*, [online], ŽIVĚ 5. února 2014, [cit. 2018-1-20], Dostupné z: <https://www.zive.cz/clanky/wi-fi-jak-si-zajistit-velke-pokryti-rychlost-a-silny-signal/anteny-a-jejich-nastaveni-deformace-a-ztrata-signalu-mereni/sc-3-a-172347-ch-90933/>

kterých jsou složeny zdi a stropy či jiné elementy ve vnitřních prostorech, které by mohly ovlivnit šíření signálu. Venku jde zase o stromy, další domy, ale i oblačnost či mlhy.

Uvnitř budov může být rádiový signál zcela odražen (hladké neabsorbující povrchy), čímž je zcela změněn jeho směr. Nebo může být lomen, kdy je část signálu odražena a část projde materiálem dále, ale pod jiným úhlem. Dále může dojít k difrakci vln, které jsou deformovány, resp. ohnuty například předmětem v místnosti nebo budovou. Na nerovných površích se spoustou hran pak může dojít k odražení signálu do více směrů, a tím k jeho rozptýlení. A nakonec může signál narazit na takový materiál, který jej může zcela absorbovat a neefektivně přeměnit na teplo.⁸

3.4. Mikroprocesory

Pro navržení zařízení, kterým budeme moci dálkově ovládat – zapínat a vypínat externí zařízení, osvětlení, zásuvky a atd. máme řadu technických možností. Jedna z těchto možností je použití mikroprocesoru, který bude prostřednictvím vstupního komunikačního modulu dálkově komunikovat s mobilním zařízením (mobilním telefonem nebo tabletem), které bude prostřednictvím výstupních modulů (relé, spínací obvody) mikroprocesoru zapínat nebo vypínat zařízení v domácnosti.

Pro realizaci řídicího přijímače může použít mimo jiné následující platformy:

- mikropočítač - mikrokontroler s komunikačním a ovládacím rozhraním - WiFi modul resp. reléový modul,
- jednočipový počítač Raspberry pi, s komunikačním a ovládacím rozhraním - WiFi modul resp. relé,
- nebo programovatelný logický automat – PLC (Programmable Logic Controller).

⁸ ŽIVĚ, PROKOP Mirek, Wi-Fi: *Jak si zajistit velké pokrytí, rychlost a silný signál*, [online], ŽIVĚ 5. února 2014, [cit. 2018-1-20], Dostupné z: [https://www.zive.cz/clanky/wi-fi-jak-si-zajistit-velke-pokryti-rychlost-a-silny-signal/anteny-a-jejich-nastaveni-deformace-a-ztrata-signalu-mereni/sc-3-a-172347-ch-90933/](https://www.zive.cz/clanky/wi-fi-jak-si-zajistit-velke-pokryti-rychlost-a-silny-signal/anteny-a-jejich-nastaveni-deformace-a-ztrata-signalu-mereni/sc-3-a-172347-ch-90933/https://www.zive.cz/clanky/wi-fi-jak-si-zajistit-velke-pokryti-rychlost-a-silny-signal/anteny-a-jejich-nastaveni-deformace-a-ztrata-signalu-mereni/sc-3-a-172347-ch-90933/)

„Mikrokontroler je integrovaný obvod, který představuje kompletní mikropočítač. Mikrokontroler nebo také MCU je jednočipový počítač tvořený jednoduchým integrovaným obvodem. Vyznačuje se především dvěma hlavními vlastnostmi, a to kompaktností a spolehlivostí, který má jediný specifický úkol. Například se může jednat o řízení či regulaci. Jednočipové počítače neboli mikrokontrolery bývají mnohdy zakomponovány do takzvaných embedded systémů (vestavěný jednoúčelový systém), které řídí daný spotřebič např. automatickou pračku, myčku nádobí atd. v domácnosti.“⁹

Mikrokontroler potřebuje pro komunikaci s vnějším světem vstupně výstupní (I/O) zařízení jako jsou: ¹⁰

- sériové a paralelní porty
- sběrnice (I2C, CAN-BUS, Ethernet)
- analogově digitální převodníky (A/D)
- digitálně analogové převodníky (D/A)
- moduly pro řízení pulzně šířkové modulace pro řízení např. motorů
- snímače např. polohy, teploty, vlhkosti, proudu atd.
- LCD displeje.

Existuje celá řada platforem založených na mikroprocesorech, resp. mikrokontrolerech jako například Arduino, NodeMCU, Parallax Basic Stamp, BX-24 od Netmedia, STM32 a atd.

⁹ IT SLOVNÍK, *Mikrokontroler* [online], IT-Slovník 2018, [cit. 2018-1-6], Dostupné z: <https://it-slovník.cz/pojem/mikrokontroler>

¹⁰ UDAVASHANKARA V, Mallikarjunaswamy M S, *8051 Microcontroller, Hardware, Software and Applications* [online], New Delhi : Tata McGraw-Hill, ©2009., 7 s., [cit. 2018-1-15], ISBN: 9780070086814, Dostupné z: https://books.google.cz/books?id=iru1w4iRx_YC&pg=PA297&dq=8051+Microcontroller,+Hardware,+Software+and+Applications&hl=cs&sa=X&ved=0ahUKEwiW082PzsjZAhXFyKQKHZwLDwMQ6AEIKDAA#v=onepage&q=8051%20Microcontroller%2C%20Hardware%2C%20Software%20and%20Applications&f=false

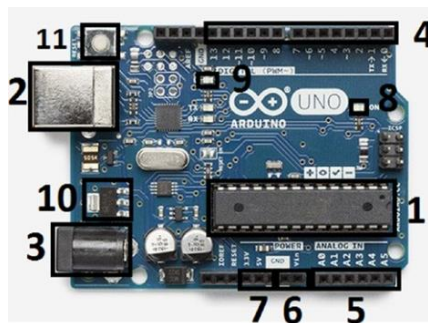
3.4.1. Arduino

Arduino je otevřená (open source) elektronická platforma založená na mikrokontrolerech ATmega od firmy Atmel.

Desky Arduino UNO jsou založeny na mikrokontroleru ATmega328P, jsou napájeny z USB nebo externího 5V zdroje. Mikrokontroler má 32KB flash paměť a 2KB paměť RAM. „Desky Arduino jsou nejlepší mikrokontrolery pro rychlé prototypování, protože k Arduino můžete připojit různé externí moduly se senzory a spínači.“¹¹

Obrázek (Obrázek 5) popisuje hlavní součásti desky Arduino UNO.

Obrázek 5 Deska Arduino UNO Rev3



Zdroj: PRADEEKA Senevirantne, *Builging Arduino PLCs, The essential techniques you need to develop Arduino-based PLCs*, Apress, 23 s., ISBN-13 (electronic): 978-1-4842-2632-2

Hlavní komponenty, označené výše na obrázku (Obrázek 5) čísly jsou následující:

1. Mikrokontroler ATmega328P
2. Konektor USB (typ B)
3. Napájecí konektor (5V až 9V)
4. Digitální piny
5. Analogové piny
6. Napájení (9V externí)

¹¹ PRADEEKA Senevirantne, *Builging Arduino PLCs, The essential techniques you need to develop Arduino-based PLCs*, Apress, 23 s., ISBN-13 (electronic): 978-1-4842-2632-2

7. Výstupní napětí OUT
8. Indikátor napájení
9. LED připojená k digitálním pinům 13
10. Regulátor napětí
11. Tlačítko Reset

Digitální piny lze použít pro propojení s různými senzory a spínači. Arduino UNO má 14 digitálních pinů a tyto piny mohou být konfigurovány jako vstupy nebo výstupy dle inicializace v programu vytvořeného například v Arduino IDE. Všechny digitální piny fungují na 5 V a mohou být zatíženy 20mA proudem, ale maximální proudová zátěž je však 40mA, při překročení tohoto proudu dojde k trvalému poškození mikrokontroleru. Výchozí stav digitálního pinu je vstupní a každý digitální pin má ve výchozím nastavení interní pull-up odpor (ve výchozím stavu odpojen) 20-50 kiloohmů.

Některé digitální piny mají speciální funkce:

- **Vysílání a příjem sériových dat:** Digitální piny 0 a 1 mohou být použity k provádění sériové komunikace, kde pin 0 může přijímat (RX) a pin 1 může vysílat (TX) sériová data TTL.
- **Modulace šířky impulsu:** Digitální piny 3, 5, 6, 9, 10 a 11 mohou být použity k poskytnutí 8bitového výstupu PWM.
- **Externí přerušení:** Digitální piny 2 a 3 mohou být konfigurovány tak, aby spouštěly přerušení na nízké hodnotě, stoupající nebo klesající hraně nebo změnu hodnoty. Tyto piny jsou velmi užitečné, když potřebujete ve svém řídicím procesu vytvářet alternativní cesty pomocí spínačů nebo senzorů a počítat impulsy ze snímačů.¹²

Technická specifikace Arduino UNO:

- | | |
|-------------------|------------|
| • Microcontroller | ATmega328P |
| • Provozní napětí | 5V |

¹² PRADEEKA Senevirantne, *Builging Arduino PLCs, The essential techniques you need to develop Arduino-based PLCs*, Apress, 24 s., ISBN-13 (electronic): 978-1-4842-2632-2

- Doporučení vstupní napětí 7-12V
- Doporučení vstupní napětí maximální 6-20V
- Digitální I/O Piny 14 (6 umí PWM)
- PWM Digital I/O Piny 6
- Analogové vstupní Piny 6
- DC proudové zatížení I/O Pinu 20 mA
- DC proudové zatížení 3.3V Piny 50 mA
- Flash paměť 32 KB (ATmega328P) z toho 0.5 KB alokováno pro bootloader
- SRAM 2 KB (ATmega328P)
- EEPROM 1 KB (ATmega328P)
- Clock Speed 16 MHz
- LED_BUILTIN 13

Přehled vyráběných desek Arduina naleznete na následujícím odkazu <https://www.arduino.cc/en/Main/Products> [cit. 2018-1-8]

3.4.2. NodeMCU

NodeMCU je open source platforma vhodná pro vytváření projektů v oblasti automatizace, které vyžadují komunikaci prostřednictvím WiFi. NodeMCU je založen na jedno jádrovém 32 bitovém procesoru XtensaLX 106 s instrukční sadou RISC a taktovací frekvencí 80MHz. Node MCU má paměť 128 Kbytes a využívá eXtendable Test Operating Systém (XTOS).

„Desky Node MCU první a druhé generace mají flash paměť o velikosti 4 MB. Firmware využívá skriptovací jazyk LUA. Node MCU může být programováno z programovacího prostředí Arduino IDE. Do prostředí Arduino IDE je potřeba nainstalovat ovladače desky ESP8266 pomocí správce nastavení nebo z webových zdrojů. Node MCU je velice levná IoT vývojová deska srovnatelná s Intel Galileo, Raspberry Pi, UDOO a dalšími

IoT vývojovými deskami. Na obrázku (Obrázek 6) je příklad vývojové desky NodeMCU Development Board/kit v 0.9, která je vhodná pro rychlý vývoj vlastních aplikací.¹³

NodeMCU obsahuje ESP8266 Wi-Fi modul, integrované GPIO, PWM, IIC a piny pro sériovou komunikaci. Umí pracovat s analogovými i digitálními signály.

NodeMCU má následující vlastnosti:¹⁴

- Rozhraní: UART, Rx, Tx
- Počet GPIO: 10 (PWM, ADC, I2C, SPI, 1W)
- paměť flash 4096kB
- I/O piny 11
- analog piny 1
- PWM piny 10
- Ovládání: AT příkazy
- Protokol: TCP / IP
- Zabezpečení: WPA / WPA2
- Módy: AP, STA, AP + STA
- 802.11b/g/n
- WiFi výkon: 25dBm v režimu 802.11b
- Klidová spotřeba: <10μA
- Operační Napětí I/O 3,3V (5V tolerantní)
- proud na I/O pin 12mA

¹³ SATAPATHY, Suresh Chandra, Bhateja, Vikrant, Das, Swagatam, *Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016, Volume 2*, [online], Springer Singapore, 268 s., [cit. 2018-02-25]. Dostupné z:

<https://books.google.cz/books?id=6AxEDwAAQBAJ&pg=PA268&dq=NodeMcu+Lua+WI-FI+ESP8266&hl=cs&sa=X&ved=0ahUKEwj7uric-cDZAhWJh7QKHeEfCtAQ6AEIRDAE#v=onepage&q=NodeMcu%20Lua%20WI-FI%20ESP8266&f=false>

¹⁴ ESPRESSIF Systems IOT Team, *ESP8266EX Datasheet Version 4.3*, Espressif Systems 2015, 7 s., [cit. 2018-2-25], Dostupné z: http://www.santy.cz/data/file/schemata/SA0287/SA0287-ESP8266_2.pdf

Obrázek 6 NodeMCU Development Board/kit v 0.9



Zdroj: ELECTRONICWINGS, NodeMCU Development Kit/Board [online], ElectronicWings 2017, [cit. 2018-2-19] Dostupné z: <http://www.electronicwings.com/nodemcu/nodemcu-development-kitboard>

3.4.3. Parallax Basic Stamp BS2-IC

BASIC Stamp je vyráběn společností Parallax, Inc. Program pro mikrokontroler je vytvářen prostřednictvím programovacího jazyka PBASIC.

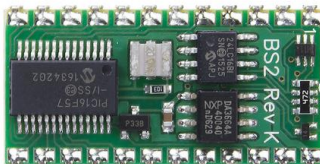
BASIC Stamp je konstrukčního hlediska malý plošný spoj obsahující osmibitový mikrořadič PIC16C57C, který je doplněn externím paměťovým modulem EEPROM s kapacitou 256 bajtů (paměť mikrokontroleru umožňuje držet až 500 instrukcí), jenž s mikrořadičem komunikuje pomocí sériové sběrnice typu I2C. Vývojový kit BS2-IC je možné připojit k osobnímu počítači, na němž probíhá vlastní vývoj software, s využitím sériového portu. K mikrořadiči lze přes jeho vstupně/výstupní piny připojovat různá periferní zařízení – LED, displeje, teplotní a optická čidla atd. Na základě BASIC Stamp jsou stavena různá ovládání například. topení v domě, jednoduché roboty.¹⁵

¹⁵ KUHNEL Claus, *BASIC Stamp: An Introduction to Microcontrollers* [online], 34 s. [cit. 2018-2-09], Newnes; 2 edition (August 4, 2000), ISBN-13: 978-0750672450, dostupné z: <https://books.google.cz/books?id=sQ9AigIIOKgC&pg=PA34&dq=BS2->

Základní specifikace: ¹⁶

- Rychlost procesoru: 20 MHz; ~4,000 PBASIC instrukce/sec
- PBASIC řídicí instrukce: 42
- 24-pin DIP
- I/O piny: 16 + 2 seriové
- RAM velikost: 32 Bytes (6 I/O, 26 proměnných)
- EEPROM (Program) velikost: 2 KBytes; ~500 PBASIC instrukcí
- napájení: 5.5 to 15 VDC (Vin), or 5 VDC (Vdd)
- proudový odběr: 3 mA Run, 50 μ A v klidovém režimu
- proudové zatížení I/O: 20 mA / 25 mA
- proudové zatížení na vybraných pinech: 40 mA / 50 mA na 8 I/O pinech
- Sériová komunikace (9600 baud při programování)
- rozměr: 1.20 x 0.63 x 0.15 in (30.0 x 16.0 x 3.81 mm)
- rozsah provozních teplot: -40 to +185 °F (-40 to +85 °C)

Obrázek 7 Mikrokontroler BS2-IC



Zdroj: PARALAX INC, *BASIC Stamp 2 Microcontroller Module* [online], Parallax Inc 2018, [cit. 2018-123], Dostupné z: <https://www.parallax.com/product/bs2-ic>

¹⁶ PARALAX INC, *BASIC Stamp 2 Microcontroller Module* [online], Parallax Inc 2018, [cit. 2018-123], Dostupné z: <https://www.parallax.com/product/bs2-ic>

3.4.4. BX-24

BX-24 je jeden z nejvýkonnějších programovatelných mikrokontrolerů. Základní modul BX24 je založen na 24-pinovém čipu mikrokontroleru Atmel, paměti EEPROM 32Kbytes, regulátoru napětí a RS232 portu. Programy jsou vytvářeny v počítači prostřednictvím rozhraní RS232 jsou přeneseny do modulu a v něm autonomně spuštěny. Na obrázku (Obrázek 8) je destička Mikrokontroler BasicX-24 hodná pro vývoj vlastních aplikací.¹⁷

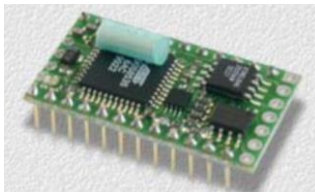
Základní specifikace BasicX-24:¹⁸

- Rychlost: 83 000 instrukcí za sekundu
- EEPROM 32K bajtů. Pro programy a data
- Maximální velikost programu Více než 8000 řádků základního kódu
- RAM 400 bajtů
- I / O piny 21 (16 normálních + 2 série P. + 3 k dispozici mimo oblast normálních pinů)
- Analogové vstupy (ADC) 8 (8 ze 16 pinů) může také fungovat jako 10bitové převodníky ADC při 6000 vzorků za sekundu.
- Rychlost sérií portů 1200 - 460,8 kB Baud
- Matematický procesor 32 bitů x 32 bitů plovoucího bodu
- Programovací rozhraní Vysokorychlostní řada
- Typ zapouzdření Modul 24 pinů typu DIP

¹⁷ BASIX, *BX24P* [online], Baxis, na následující odkazu jsou dostupné úplné specifikace mikrokontroleru BX24P [cit. 2018-1-4], Dostupné z: <http://www.basicx.com/Products/BX-24/bx24specs.htm>

¹⁸ BASIX, *BX24P* [online], Baxis, na následující odkazu jsou dostupné úplné specifikace mikrokontroleru BX24P [cit. 2018-1-4], Dostupné z: <http://www.basicx.com/Products/BX-24/bx24specs.htm>

Obrázek 8 Mikrokontroler BasicX-24



Zdroj: BASIX, BX24P [online], Baxis, [cit. 2018-1-4], Dostupné z: <http://www.basicx.com/Products/BX-24/bx24overview.htm>

3.4.5. STM32

STM32 resp. STM32F103C8T6 je výkonný mikrokontroler založený na ARM Cortex-M3 32bit v 48 pin LQFP pouzdře. Obsahuje vysoce výkonné RISC jádro s pracovní frekvencí 72MHz, vysokorychlostní integrované paměti, širokou řadu vyspělých I/O a periférií připojených ke dvěma APB sběrnicím. STM32F103C8T6 obsahuje 12bit ADC, časovače, PWM časovač, standardní a vyspělé komunikační rozhraní. STM32 několik režimů, kterému umožňují šetření proudem, a proto je vhodný pro návrhy nízko příkonových aplikací.¹⁹ Na obrázku (Obrázek 9) je modul Mikrokontroler STM 32.

Základní parametry jsou:²⁰

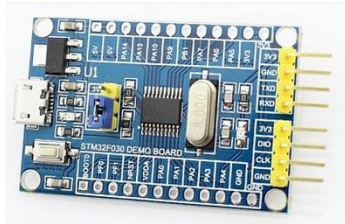
- Rozsah provozního napětí od 2V do 3.6V
- 64Kbytu flash paměti
- 20Kbytu SRAM
- CRC výpočetní jednotka, 96bitové unikátní ID
- Dva 12bitové, 1 μ s A/D převodníky (až 10 kanálů)
- 7 kanálový DMA kontrolér, 3 všeobecné časovače a 1 vyspělý řídicí časovač

¹⁹ ST MICROELECTRONIC, *STM32F0x1* [online], ST Microelectronic 2018, výrobce [cit. 2018-1-28], Dostupné z: http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-mainstream-mcus/stm32f0-series/stm32f0x1.html?querycriteria=productId=LN7

²⁰ FARNELL, *STM32F103C8T6 - ARM Mikrokontroler* [online], Farnell 2018, [cit. 2018-1-26], Dostupné z: <http://cz.farnell.com/stmicroelectronics/stm32f103c8t6/mcu-32bit-cortex-m3-72mhz-lqfp/dp/1447637>

- 37 rychlých I/O portů
- Serial wire debug (SWD) a JTAG rozhraní
- Dvě SPI, dvě I2C, tři USART, jedno USB a jedno CAN rozhraní
- Okolní provozní teplotní rozsah od -40 °C do 85 °C

Obrázek 9 Mikrokontroler STM 32



Zdroj: ROOT.CZ, STM32: mikrokontrolér vstřícný k amatérům [online], Root.cz 2017, [cit. 2018-2-2], Dostupné z: <https://www.root.cz/clanky/stm32-mikrokontroler-vstricny-k-amaterum/>

3.4.6. Raspberry Pi

Raspberry Pi je jednočipový počítač, který je srovnatelný se (slabším) stolním počítačem. Raspberry Pi je založen na mikroprocesoru ARM, desky s Raspberry Pi mají implementovaný HDMI konektor pro monitor, Ethernet konektor, konektor USB pro připojení klávesnice a myši. Na počítači Raspberry Pi je možné provozovat různé distribuce Linuxu, RISC OS, jakož i Microsoft Windows 10 IoT Core.

Na rozdíl od Arduina je možné Raspberry Pi použít nejen k ovládání různých zařízení (pomocí GPIO kontaktů), ale i k samotnému vývoji příslušných aplikací. Velmi často je používán jako multimediální přehrávač videa a hudby.²¹

Základní technické parametry jednodeskového počítače RASPBERRY Pi 3 Model B obrázek (Obrázek 10):²²

- procesor ARMv8 1.2GHz quad-core
- 4x USB 2.0
- 40x piny GPIO
- 1x HDMI s podporou FullHD (1080p 30fps)
- 1x ethernet port (10/100Mbit/s)
- 1x slot pro kameru (CSI)
- 1x slot pro displej (DSI)
- 1x 4-pinový 3,5mm audio jack + kompozitní video
- 1x napájecí micro USB port
- 1x slot pro micro SD(XC) kartu
- Wi-Fi 802.11n
- Bluetooth 4.1
- rozměry 86x56x20mm

²¹ MOLLOY Derek, *Exploring Raspberry Pi* [online], Wiley; 1 edition (June 13, 2016), 37 s., [cit. 2018-02-25], Dostupné z: <https://books.google.cz/books?id=K-phDAAAQBAJ&pg=PT704&lpg=PT703&focus=viewport&dq=nodemcu+description&hl=cs#v=onepage&q=nodemcu%20description&f=false>

²² TS BOHEMIA, Raspberry Pi 3 Model B [online], TS Bohemia 2018, [cit. 2018-2-11], Dohledatelné z: https://www.tsbohemia.cz/raspberry-pi-3-model-b_d239934.html

Obrázek 10 RASPBERRY Pi 3 Model B



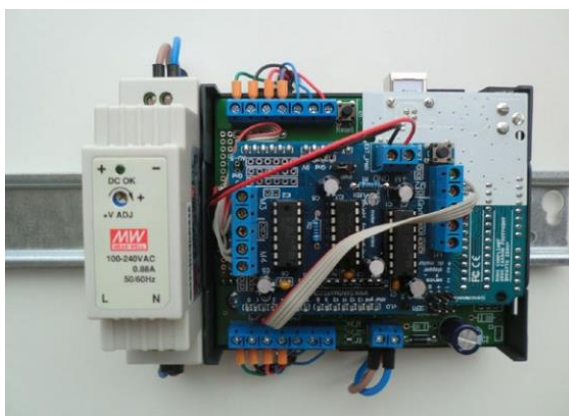
Zdroj: TS BOHEMIA, Raspberry Pi 3 Model B [online], TS Bohemia 2018, [cit. 2018-2-11],
Dohledatelné z: https://www.tsbohemia.cz/raspberry-pi-3-model-b_d239934.html

3.4.7. PLC

PLC je zařízení určené pro řízení technologických procesů v průmyslu. Řídicí modul PLC dostává vstupní data z externích periférií jako jsou senzory a přepínače a zpracovává je dle logiky, která odpovídá řízení daného technologického procesu. PLC řídí technologický proces prostřednictvím výstupních modulů zařízení, jako jsou relé a motory.

PLC je navrženo tak, aby zvládalo řízení průmyslové výroby a vytvářelo flexibilní, modulární a snadno automatizované systémy. Obrázek (Obrázek 11) znázorňuje základní PLC, který je namontován na DIN lištu s napájecím zdrojem.²³

Obrázek 11 PLC od firmy Hartmut Wendt Hard & Softwareentwicklung



Zdroj: PRADEEKA Senevirantne, *Builging Arduino PLCs, The essential techniques you need to develop Arduino-based PLCs*, Apress, 58 s., ISBN-13 (electronic): 978-1-4842-2632-2

PLC je řízeno mikrokontrolerem, který je schopen provozovat uživatelem definovaný software.

PLC na základě stavu vstupů ovládá výstupy tak, aby bylo dosaženo minimální odchylky od žádaného nebo zadaného stavu celého zařízení, řízeného procesu. Operační systém PLC má za úkol zajistit, aby byl program opakovaně (cyklicky) vykonáván. Vstupy

²³ PRADEEKA Senevirantne, *Builging Arduino PLCs, The essential techniques you need to develop Arduino-based PLCs*, Apress, 58, 59 s., ISBN-13 (electronic): 978-1-4842-2632-2

a výstupy jsou oddělené opto-členy, které ochrání mikrokontroler před vnějším rušením a oddělují jej od externích obvodů. Opto-izolované vstupy a výstupy jsou připojeny ke svorkovnici, přes kterou jsou připojena dálkově ovládané periférie, jako jsou senzory, přepínače a senzory pohony.

Napájecí zdroj je obvykle namontován na stejnou lištu DIN, na které je instalováno i vlastní PLC. K PLC lze připojit externí komunikační moduly, které mají rozhraní pro Ethernet nebo WiFi a lze přesně připojit PLC k internetu nebo k interní síti. Většina PLC má rozhraní USB pro programování paměti flash mikrokontroleru pomocí připojeného počítače.²⁴

PLC se obvykle skládá z následujících částí a příslušenství:

- Mikrokontroler (CPU)
- I / O moduly
- Digitální vstupy / výstupy
- Analogové vstupy
- Výstupy PWM
- Napájecí jednotka
- DIN lišta

PLC jsou prodávány v ucelených řadách jako například LOGO!8 od společnosti Siemens AG, který představuje univerzální logický modul. „LOGO!8 nabízí řešení pro aplikace v domácnostech např. osvětlení chodeb, exteriérů a teras, stahování rolet, otvírání vrat, řízení ventilačních systémů, řízení vytápění v domech atd.“²⁵

Oproti mikrokontrolerům, jako je například Arduino jsou PLC navrženy pro řízení technologických procesů, kde je důležité robustnost a spolehlivost. Pokud jsou PLC nasazena v domácnostech, pak jsou nejčastěji používány pro řízení a regulaci vytápění domu.

²⁴ PRADEEKA Senevirantne, *Builging Arduino PLCs, The essential techniques you need to develop Arduino-based PLCs*, Apress, 58, 59 s., ISBN-13 (electronic): 978-1-4842-2632-2

²⁵ SIEMENS AG, *SIMATIC LOGO! Manual - Industry Support Siemens* [online], Siemens AG 2001-2003, [cit. 2018-2-20], Dostupné z: https://cache.industry.siemens.com/dl/files/461/16527461/att_82564/v1/Logo_e.pdf

3.4.8. Zhodnocení mikrokontrolerů

V té to kapitole jsou souhrnně uvedeny základní výhody a nevýhody výše uvedených mikroprocesorů z pohledu jejich použití pro dálkové ovládání spotřebičů v domácnosti.

Arduino:

a) Výhody:

Ucelená řada výkonných mikroprocesorů na bázi open source, velký sortiment desek, velké množství různých specializovaných externích modulů, jednoduché programování v prostředí Arduino IDE, nízká pořizovací cena desek i externích modulů, snadné vytváření prototypů zařízení.

b) Nevýhody:

Na deskách není v základu instalována přímá podpora komunikace prostřednictvím WiFi nebo Bluetooth.

NodeMCU

a) Výhody:

Desky jsou na bázi open source, v základu je na desce WiFi modul, 4MB paměť, podpora zabezpečení WiFi pře WPA/WPA2, rozměr desky, jednoduché programování v prostředí Arduino IDE, nízká pořizovací cena desek i externích modulů, snadné vytváření prototypů zařízení, možnost použití externích modulů určených pro Arduino.

b) Nevýhody:

Z pohledu uvažovaného nasazení v praktické části nejsou zřejmě žádné nevýhody.

BS2-IC

a) Výhody:

Možnost použití externích modulů určených pro Arduino, nízká cena, nízký proudový odběr, proudové zatížení vstupně výstupních pinů 40mA.

b) Nevýhody:

EEPROM má velikost proze pro 500 instrukcí, k programování je pout vlatní programovací jazyk PBASIC, který není kompatibilní s programovacím prostředím Arduino IDE, Na deskách není v základu instalována přímá podpora komunikace prostřednictvím WiFi nebo Bluetooth.

BX-24

a) Výhody:

Založen na stejném čipu Atmel jako Arduino, velké množství různých specializovaných externích modulů, jednoduché programování v prostředí Arduino IDE, nízká pořizovací cena desek i externích modulů, snadné vytváření prototypů zařízení.

b) Nevýhody:

Na deskách není v základu instalována přímá podpora komunikace prostřednictvím WiFi nebo Bluetooth.

STM32

a) Výhody:

Podporuje několik režimů pro šetření proudem, proto je vhodný pro nízkopříkonová řešení, výkon procesoru, dva 12 bitové A/D převodníky, podpora CAN Bus rozhraní.

b) Nevýhody:

c) Na deskách není v základu instalována přímá podpora komunikace prostřednictvím WiFi nebo Bluetooth.

Raspberry Pi

a) Výhody:

jednočipový počítač, který je srovnatelný se stolním počítačem, podpora distribucí operačních systémů např. Linuxu a Microsoft Windows 10 IoT Core, vývoj aplikací přímo na desce.

b) Nevýhody:

Z pohledu uvažovaného nasazení v praktické části je to příliš robustní řešení, kde by došlo jen k minimálnímu využití výkonu počítače, cena v porovnání s mikrokontrolery.

PLC

a) Výhody:

PLC jsou navržena pro průmyslová řešení, robustnost jejich konstrukce, spolehlivost,

b) Nevýhody:

Cena v porovnání s mikrokontrolery, drahá řešení, nutnost použití modulu od stejného výrobce jako je řídicí modul PLC, nároky na prostor při instalaci.

Vyhodnocení:

Z pohledu uvažovaného nasazení v praktické části nejlépe vychází použití NodeMCU, Výhodou této desky, že její součástí je Wi-Fi modul ESP8622 (ESP-12E), který obsahuje čip CP2102 TTL a podporuje protokoly 802.11b/g/n a má zabezpečení WPA/WPA2, není tedy nutný žádný přídavný WiFi modul, který je potřeba u dalších mikrokontrolerů. Dále je pořizovací cena desek a možnost k programování použít Arduino IDE programovací prostředí.

3.5. Bezdrátová komunikace

Mikropočítače potřebují komunikovat se svým okolím, k tomuto mimo jiné mohou využít bezdrátovou komunikaci. Na trhu je velké množství bezdrátových komunikačních modulů využívající standardy WiFi (IEEE802.11x), Bluetooth, ZigBee (IEEE 802.15.4) a Near Field Communication (NFC).

Nejde jednoznačně říci, který bezdrátový komunikační standard je nejlepší, každý standard má své výhody a nevýhody:

- **Bluetooth** je populární standard pro propojení s počítačovými periferiemi, audio zařízeními, osobními sítěmi (PAN) a mobilními zařízeními. Tento standard je používán u aplikací, kde rychlost dat není rozhodujícím faktorem. Má nízkonákladový a nízko příkonový profil, což je zvláště vhodné pro zařízení s bateriemi. Je zavedena i technologie Bluetooth s nízkou spotřebou energie (LE), která má za cíl podporovat aplikace, u kterých je vyžadována velmi nízká spotřeba energie, při zachování srovnatelných komunikačních rozsahů.
- **Wi-Fi** komunikace je vhodnější než Bluetooth pro rozsáhlé síťové aplikace, u nichž je vysoká přenosová rychlost kritická, proto je oblíbená u zařízení připojených k internetu a přenosných počítačů. WiFi je definováno standardem IEEE 802.11x (písmeno x je označení verze), funguje ve stejném frekvenčním pásmu jako Bluetooth tedy bezlicenční frekvenční pásmo od 2,4GHz do 5,9GHz dle použitého protokolu nebo technologie.
- **ZigBee** standard (IEEE 802.15.4) je navržen tak, aby měl nízký výkonový profil a přesto mohl komunikovat na vzdálenosti do 100 m a mohl vytvářet síťové uspořádání pro další rozšiřování rozsahu sítě. ZigBee má omezenou

maximální rychlost dat (20, 40, 250 kbit/s) což je v porovnání s Bluetooth a Wi-Fi velká nevýhoda. ZigBee má nízkou komunikační latenci znamená, což je výhodné pro řízení v reálném čase.

- **NFC** je standard pro radiokomunikaci krátkého dosahu, který staví na komunikaci pomocí rádiových frekvencí (RFID). Podporuje komunikační rozsah až 20 cm a umožňuje velmi vysoké přenosové rychlosti, když se zařízení téměř dotýkají (tj. vzdálenost je menší než 5 cm). NFC podporuje komunikaci s nepohyblivými zařízeními pomocí indukčního spoje.²⁶

Obecné charakteristiky uvedených bezdrátových standardů jsou shrnuty v tabulce (Tabulka 1) Porovnání různých bezdrátových standardů. Je zřejmé, že rychlost a rozsah komunikace jsou velmi důležitými faktory při výběru modulu, který bude instalován do dálkově ovládaného zařízení. Z uvedených bezdrátových standardů jsou Bluetooth, WiFi a NFC běžně podporovány mobilními telefony a tablety.

²⁶ MOLLOY Derek, *Exploring Raspberry Pi* [online], Wiley; 1 edition (June 13, 2016), 536 s., [cit. 2018-02-25], Dostupné z: <https://books.google.cz/books?id=K-phDAAAQBAJ&pg=PT704&lpg=PT703&focus=viewport&dq=nodemcu+description&hl=cs#v=onepage&q=nodemcu%20description&f=false>

Tabulka 1 Porovnání různých bezdrátových standardů

| | BLUETOOTH | WI-FI | ZIGBEE | NFC/RFID |
|--------------|---------------|----------------|-----------------|----------------|
| Standard | IEEE 802.15.1 | IEEE 802.11 | IEEE 802.15.4 | ISO/IEC |
| Range | 10m to 100m | 50m to 100m | 30m to 100m+ | <20cm |
| Power | Low | High | Very Low | Very Low |
| Data Rate | <2.1Mb/s | 10 to 300Mb/s | <250kb/s | Up to 20Mb/s |
| Topology | Star | Star | Mesh/Star | Point-to-point |
| Organization | Bluetooth SIG | Wi-Fi Alliance | ZigBee Alliance | NFC Forum |

Zdroj: MOLLOY Derek, *Exploring Raspberry Pi* [online], Wiley; 1 edition (June 13, 2016), 536 s., [cit. 2018-02-25], Dostupné z: <https://books.google.cz/books?id=K-phDAAAQBAJ&pg=PT704&lpg=PT703&focus=viewport&dq=nodemcu+description&hl=cs#v=onepage&q=nodemcu%20description&f=false>

Z pohledu dosahu a přenosové rychlosti je nejvýhodnější použít WiFi komunikační protokol pro komunikaci mikrokontroleru s mobilním zařízením při vytvoření praktické ukázky ovládání zařízení v domácnosti.

3.6. Popis NodeMCU

Pro vytvoření praktické ukázky ovládání zařízení v domácnosti jsme zvolili vývojovou desku NodeMCU jak již bylo zdůvodněno v kapitole 3.4.8 Zhodnocení mikrokontrolerů. Pro vlastní řešení byl použit NodeMCU Dev Kit v1.0.

3.6.1. NodeMCU Dev Kit v1.0

NodeMCU má vstupně výstupní rozhraní - piny, kterým lze přiřadit různé funkce prostřednictvím programování příslušných registrů.

Piny jsou rozděleny na: ²⁷

- GPIO (General Purpose Input Output)
- ADC (Analog to Digital Converter)

²⁷ ELECTRONICWINGS, *NodeMCU Development Kit/Board* [online], ElectronicWings 2017, [cit. 2018-2-19] Dostupné z: <http://www.electronicwings.com/nodemcu/nodemcu-development-kitboard>

- SPI (Seriál Peripheral Interface)
- I2C (Inter - Integrated Circuitice)
- UART (Univerzal Asynchronous Receiver Transmitter)

GPIO piny

Na obrázku (Obrázek 12) bíle označené piny: GPIO0 – GPIO16

NodeMCU má obecné vstupně / výstupní piny na své desce, jak je znázorněno na obrázku (Obrázek 12). Piny můžeme programově nastavit na úroveň High / Low (1/0, 1 je napětí 5V) a přímo jimi řídit například LED diodu, relé připojené přes optočlen nebo spínat tranzistor, který chrání pin před proudovým přetížením, a spíná navazující obvody. Na pinech GPIO3, GPIO1, GPIO9 a GPIO10 můžeme generovat PWM signál (Pulse Width Modulation), kterým můžeme například řídit otáčky malého motorku.²⁸

Kanál ADC

Na obrázku (Obrázek 12) fialově označený pin: A0

NodeMCU má na desce jeden ADC kanál / pin a to na pinu označeném A0. Analogově/digitální převodník (ADC) se používá k převodu analogového signálu do digitální podoby. Čip ESP8266 má zabudovaný 10bitový ADC převodník pouze s jedním ADC kanálem, tj. má pouze jeden ADC vstupní pin pro čtení analogového napětí od externího zařízení / snímače. Kanál ADC je multiplexován s napětím baterie, proto jej můžeme nastavit tak, aby měřilo buď napájecí napětí, nebo externí analogové napětí. Rozsah vstupního napětí pro pin ADC je 0-1,0V při čtení externího analogového napětí.²⁹

Piny SPI

Na obrázku (Obrázek 12) červeně označené piny: SD1, CMD, SD0, CLK a SDIO_DATA0 – SDIO_DATA3, SDIO_CMD, SDIO_CLK

NodeMCU založená na ESP8266 podporuje Hardware SPI (HSPI) komunikaci na čtyřech pinech, podporuje také komunikaci Quad-SPI. Pomocí rozhraní SPI můžeme k

²⁸ ELECTRONICWINGS, *NodeMCU Development Kit/Board, GPIO* [online], ElectronicWings 2017, [cit. 2018-2-19] Dostupné z <http://www.electronicwings.com/nodemcu/nodemcu-gpio-with-esplorer-ide>

²⁹ ELECTRONICWINGS, *NodeMCU Development Kit/Board, IDE* [online], ElectronicWings 2017, [cit. 2018-2-19] Dostupné z <http://www.electronicwings.com/nodemcu/nodemcu-adc-with-esplorer-ide>

NodeMCU připojit libovolné zařízení podporující SPI a umožnit komunikaci s ním. Sériové periferní rozhraní (SPI) je rozhraní čtyř vodičové sběrnice, které používá sériový komunikační protokol. SPI komunikuje v režimu master duplex nebo plný duplex, což znamená, že na sběrnici může současně komunikovat pouze jeden master a jediný slave. Zařízení s funkcí SPI pracují ve dvou základních režimech SPI, tj. SPI Master Mode a SPI Slave Mode, kdy hlavní zařízení odpovídá za zahájení komunikace a generuje hodinové impulzy pro synchronní přenos dat. Master zařízení může obsluhovat více slave zařízení, které jsou připojeny ke sběrnici a to jedno po druhém.

SPI piny (SD1, CMD, SD0, CLK), jsou používány výlučně pro komunikaci Quad-SPI s flash pamětí na ESP-12E, a proto nemohou být použity pro aplikace SPI.³⁰

Piny I2C

Na obrázku (Obrázek 12) bíle označené piny: GPIO0 – GPIO16

NodeMCU podporuje sběrnici I2C na čipu ESP8266 GPIO. I2C (Inter Integrated Circuit) je protokol připojení k sériové sběrnici. I2C je také nazýváno jako TWI (dvoukanálové rozhraní), protože používá pouze dva vodiče pro komunikaci. Tyto dva vodiče jsou SDA (sériová data) a SCL (sériové hodiny).

I2C je komunikační protokol založený na potvrzení, tj. vysílač čeká na potvrzení od přijímače po přenosu dat, aby zjistil, zda jsou data přijata úspěšně přijímačem. NodeMCU má podporu funkčnosti I2C na všech pinech GPIO.

I2C pracuje ve dvou režimech,

- Hlavní režim
- Slave režim

SDA (sériová data) se používá pro výměnu dat mezi hlavním a podřízeným zařízením. SCL (sériové hodiny) se používají jako synchronní hodiny mezi hlavním a podřízeným zařízením.

³⁰ ELECTRONICWINGS, *NodeMCU Development Kit/Board, SPI* [online], ElectronicWings 2017, [cit. 2018-2-19] Dostupné z <http://www.electronicwings.com/nodemcu/nodemcu-spi-with-arduino-ide>

Hlavní zařízení – Master iniciuje komunikaci s podřízeným zařízením. Master vyžaduje adresu od slave zařízení, aby mohl zahájit konverzaci s tímto podřízeným zařízením. Slave zařízení reaguje na hlavní zařízení, pokud je adresováno hlavním zařízením.³¹

UART

Na obrázku (Obrázek 12) světle šedě označené piny: SD_DATA_1(RXD1), GPIO2(TXD1), GPIO17(TXD2), GPIO1(TXD0), GPIO16(RXD2), GPIO3(RXD0)

NodeMCU založená ESP8266 má dvě rozhraní UART, UART0 a UART1. Vzhledem k tomu, že UART0 (RXD0 & TXD0) se používá k nahrání firmwaru / kódů do desky, nemůžeme je používat v aplikacích při nahrávání firmwaru / kódů. UART je sériový komunikační protokol, ve kterém jsou data přenášena sériově po jednotlivých bitech. Asynchronní sériová komunikace je široce používána pro byte orientovaný přenos. V Asynchronní sériové komunikaci se bajt dat přenáší najednou. Sériový komunikační protokol UART používá definovanou strukturu rámců pro své datové bajty. Struktura rámce v asynchronní komunikaci je:

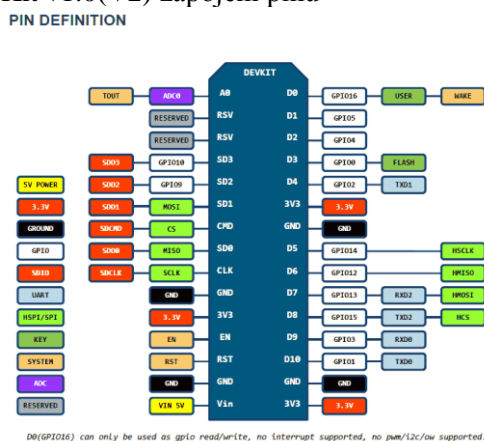
- **START bit:** je bit, kterým je zahajovaná komunikace a je to bit LOW tedy úroveň 0V,
- **Paket bitů:** Datové bity mohou být pakety 5 až 9 bitů. Normálně používáme 8bitový datový paket, který je vždy odeslán po START bit.
- **Bit STOP:** Obvykle je to jeden nebo dva bity. Je odeslán po paketu datových bitů, který označuje konec rámce. Stop bit je vždy logicky vysoký High, tedy 5V.

Asynchronní sériový komunikační rámec se obvykle skládá ze START bitu (1 bit), následovaného datovým bajtem (8 bitů) a pak z bitu STOP (1 bit), který tvoří 10bitový snímek. Rámec může také sestávat ze dvou bitů STOP namísto jediného bitu a po bit STOP může být také bit PARITY.

³¹ ELECTRONICWINGS, *NodeMCU Development Kit/Board, I2C* [online], ElectronicWings 2017, [cit. 2018-2-19] Dostupné z <http://www.electronicwings.com/nodemcu/nodemcu-i2c-with-esplorer-ide>

Rychlost přenosu dat ESP8266 přes rozhraní UART může dosáhnout 40x 115200, tj. 4,5 Mb / s. Ve výchozím nastavení je přenosová rychlost UART0 115200 pro frekvenci 40MHz, tato frekvence může být změněna na uživatelem definovanou hodnotu podle potřeby aplikace.³²

Obrázek 12 NodeMCU Dev Kit v1.0(V2) zapojení pinů



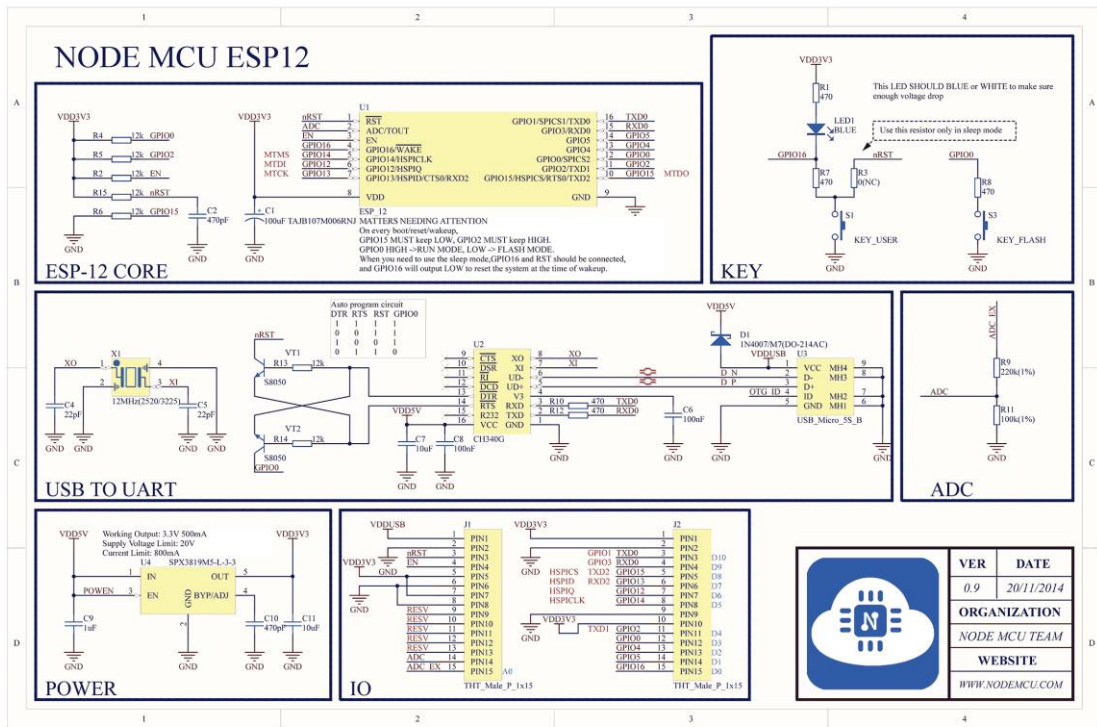
Zdroj: ELECTRONICWINGS, *NodeMCU Development Kit/Board* [online], ElectronicWings 2017, [cit. 2018-2-19] Dostupné z: <http://www.electronicwings.com/nodemcu/nodemcu-development-kitboard>

³² ELECTRONICWINGS, *NodeMCU Development Kit/Board, UART* [online], ElectronicWings 2017, [cit. 2018-2-19] Dostupné z <http://www.electronicwings.com/nodemcu/nodemcu-uart-with-esplorer-ide>

Hardware specifikace

Na obrázku (Obrázek 13) je vidět hardware NodeMCU Dev Kit v0.9 technická specifikace a design. Design NodeMCU je otevřený, je pod licenci open source.³³

Obrázek 13 NodeMCU Dev Kit v0.9 technická specifikace a design



Zdroj: ELECTRONICWINGS, *NodeMCU Development Kit/Board* [online], ElectronicWings 2017, [cit. 2018-2-19] Dostupné z: <http://www.electronicwings.com/nodemcu/nodemcu-development-kitboard>

³³ ELECTRONICWINGS, *NodeMCU Development Kit/Board* [online], ElectronicWings 2017, [cit. 2018-2-19] Dostupné z: <http://www.electronicwings.com/nodemcu/nodemcu-development-kitboard>

3.6.2. Programovací prostředí pro NodeMCU

Pro psaní programu je možné využít programovací prostředí Arduino IDE, který umožňuje snadné psaní kódu a následné nahrání programu na desku NodeMCU. Programovací prostředí Arduino IDE lze nainstalovat na operační systém Windows, Max OS X a Linux.

Programovací prostředí Arduino IDE je vedeno jako open source nástroj. Tento typ licence umožňuje rozšiřování a upravování jak samotného programovacího prostředí Arduino IDE, tak i jednotlivých knihoven, které umožňují pohodlné programování jednotlivých rozhraní pro input output zařízení a shieldů. Software Arduino IDE je volně dostupné ke stažení na následujícím odkazu <https://www.arduino.cc/en/Main/Software> [2018-2-20]

Následující obrázek (Obrázek 14) ukazuje programovací prostředí Android IDE.

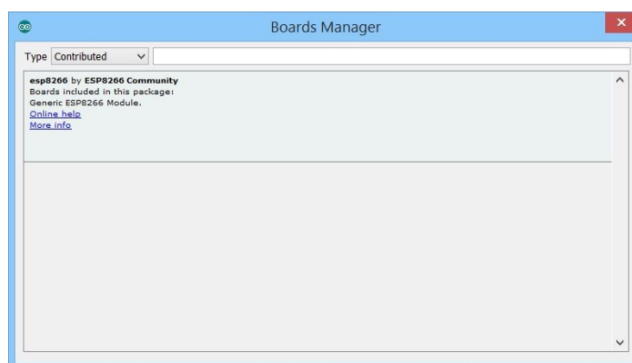
Obrázek 14 Programovací prostředí IDE



Zdroj: Autor

Aby bylo možné vytvářet programy pro desku NodeMCU v programovacím prostředí Arduino IDE, je nutné do prostředí doinstalovat patřičné knihovny. Obrázek (Obrázek 15) ukazuje instalaci knihovny ESP8266 (NodeMCU) desku do prostředí Arduino IDE³⁴. Knihovna pro desky NodeMCu je dostupná <https://github.com/esp8266/Arduino> [2018-2-20].

Obrázek 15 Instalace knihovny ESP8266 do prostředí Arduino IDE



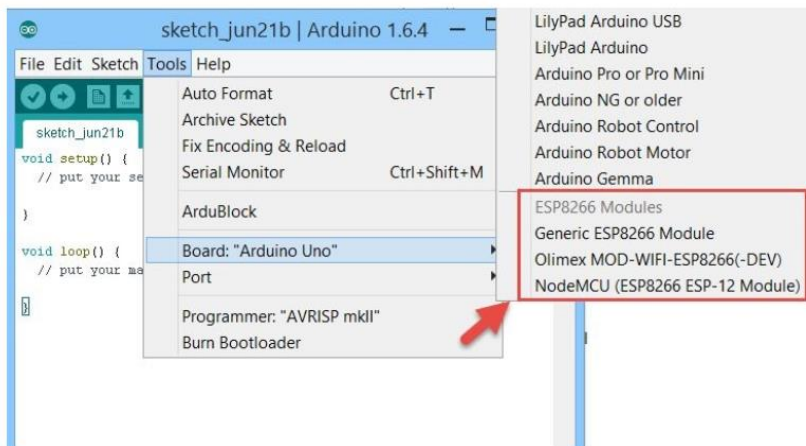
Zdroj: Kindle Edition, *NodeMCU Development Workshop* [online], PE Press; Auflage: 1 (31. Juli 2015), s 36, Dostupné z: <http://www.ebook777.com/nodemcu-development-workshop/>

Po nainstalování knihovny pro ESP8266 je nutné vybrat a nastavit vývojové prostředí pro práci s deskou NodeMCU.³⁵ To se provede výběrem desky viz obrázek (Obrázek16)

³⁴ KINDLE Edition, *NodeMCU Development Workshop* [online], PE Press; Auflage: 1 (31. Juli 2015), , s 36, Dostupné z: <http://www.ebook777.com/nodemcu-development-workshop/>

³⁵ KINDLE Edition, *NodeMCU Development Workshop* [online], PE Press; Auflage: 1 (31. Juli 2015), , s 37, Dostupné z: <http://www.ebook777.com/nodemcu-development-workshop/>

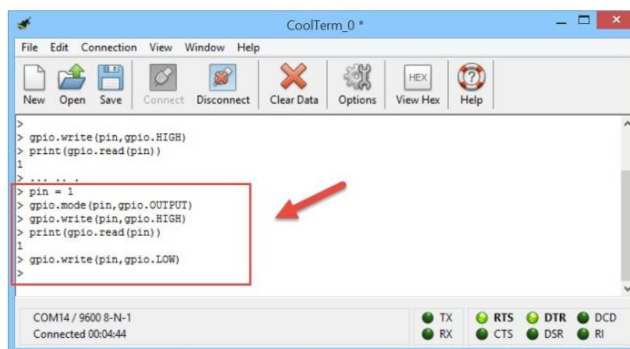
Obrázek 16 Nastavení NodeMCU desky v programovacím prostředí IDE



Zdroj: Kindle Edition, *NodeMCU Development Workshop* [online], PE Press; Auflage: 1 (31. Juli 2015), s 37, Dostupné z: <http://www.ebook777.com/nodemcu-development-workshop/>

Následující obrázek (Obrázek 17) ukazuje část kódu pro NodeMCU.³⁶

Obrázek 17 Arduino IDE s kódem



Zdroj: Kindle Edition, *NodeMCU Development Workshop* [online], PE Press; Auflage: 1 (31. Juli 2015), , 41 s., Dostupné z: <http://www.ebook777.com/nodemcu-development-workshop/>

³⁶ Kindle Edition, *NodeMCU Development Workshop* [online], PE Press; Auflage: 1 (31. Juli 2015), , s 41, Dostupné z: <http://www.ebook777.com/nodemcu-development-workshop/>

3.7. Android

Android je operační systém, který vytvořila společnost Google. Operační systém Android je vyvíjen jako otevřený, tedy jako open source, což znamená, že softwarový vývojáři mají volný přístup k jeho zdrojovému kódu a mohou ho modifikovat a vyvíjet, což vývoj aplikací a systému rychle posouvá ku předu.

Samotný OS Android je založen na Linuxovém jádře verze 2.6, které se stará mimo jiné o bezpečnost systému, obsluhu požadavků procesů a správu paměti. Použití Linuxového jádra neumožňuje aplikacím přímý přístup k jádru systému, ale musí přistupovat k jádru prostřednictvím API (Application Programming Interface), to přispívá k větší spolehlivosti, bezpečnosti a stabilitě systému.

Vývojové prostředí je navrženo tak, aby vývojářům umožnilo jednoduše pracovat se základními funkcemi telefonu (sestavení telefonního hovoru a jeho příjem, odeslání a přijímání SMS) a aby jim také umožnilo vytvářet graficky zajímavé aplikace, které využívají všech možností hardware mobilního zařízení.

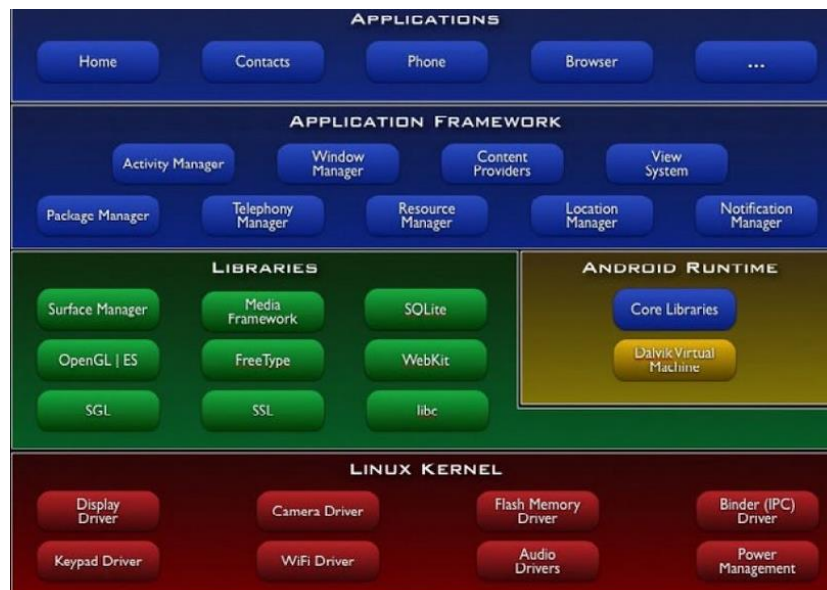
Architektura operačního systému Android je složena z následujících pěti vrstev:

- Aplikační,
- Aplikační Framework,
- Libraries (knihovny),
- Android runtime,
- Linux Kernel.

Aplikace, jak již bylo řečeno, přistupují k jádru prostřednictvím API a požadavků na nižší vrstvy, které požadavky upravují a postupují další vrstvě. Následující obrázek (Obrázek 18) ukazuje jednotlivé vrstvy OS Android. OS Android od samého počátku vyvíjen pro mobilní telefony a tablety, ale dnes již tento operační systém nalezneme i v televizích.³⁷

³⁷ UJBÁNYAI Miroslav, *Programujeme pro Android* [online], Grada Publishing a.s. 2012, 13 s. [cit. 2018-1-10], ISBN 9788024779836, Dostupné z: <https://books.google.cz/books?id=o6kHGXua8oQC&printsec=frontcover&dq=Android&hl=cs&sa=X&ved=0ahUKEwIU1PbA4MjZAhWBy6QKHeHcAhIQ6AEIKDAA#v=onepage&q=Android&f=false>

Obrázek 18 Architektura operačního systému Android



Zdroj: UJBÁNYAI Miroslav, *Programujeme pro Android* [online], Grada Publishing a.s. 2012, 17 s. [cit. 2018-1-10], ISBN 9788024779836, Dostupné z: <https://books.google.cz/books?id=o6kHGXua8oQC&printsec=frontcover&dq=Android&hl=cs&sa=X&ved=0ahUKEwIU1PbA4MjZAhWBy6QKHeHcAhIQ6AEIKDAA#v=onepage&q=Android&f=false>

Vývojové prostředí

Vývoj aplikací pro Android, je založeno na tzv. Host-Target programovacím a vývojovém prostředí. Vývoj programu v rámci Host-Target založen na tom, že vyvíjená aplikace se realizuje v rozdílném, prostředí než prostředí, ve kterém bude nakonec provozována. Aplikaci vyvíjíme na počítači s vývojovým prostředím vhodným pro vývoj programu, ale testování výsledného programu probíhá na vlastním mobilním zařízení.

Aplikace jsou testovány na reálném přístroji s operačním systémem Android. Pro testování je možné použít i emulátoru daného koncového mobilního zařízení. Konečné testování je ale vždy na reálném zařízení.³⁸

Ke psaní programu pro operační systém Android bylo použito programovací prostředí Android Studio. Android Studio je oficiální IDE pro operační systém Android od společnosti Google, které je postaveno na základech prostředí „IntelliJ IDEA“³⁹ od společnosti JetBrains. Android studio je dostupné pro operační systém Windows, Mac OS X a Linux.

Android studio je tool, který programátor může použít k vývoji aplikací pro OS Android.⁴⁰ Následující obrázek (Obrázek 19) ukazuje vývojové prostředí Android Studio.

³⁸ UJBÁNYAI Miroslav, *Programujeme pro Android* [online], Grada Publishing a.s. 2012, 23 s. [cit. 2018-1-10], ISBN 9788024779836, Dostupné z: <https://books.google.cz/books?id=o6kHGXua8oQC&printsec=frontcover&dq=Android&hl=cs&sa=X&ved=0ahUKEwIU1PbA4MjZAhWBy6QKHeHcAhIQ6AEIKDAA#v=onepage&q=Android&f=false>

³⁹ INTELLIJ IDEA, *IntelliJ IDEA* [online], *IntelliJ IDEA 22018*, [cit. 2018-01-09], Dostupné z: <https://www.jetbrains.com/idea/?fromMenu>

⁴⁰ ONUR Murat Yener, Dundar, *Expert Android Studio* [online], John Wiley & Sons 2016 , xxiv s., [cit. 2018-1-08], ISDN: 9781119110712, Dostupné z: <https://books.google.cz/books?id=mCjqDAAAQBAJ&pg=PA337&dq=Android+Studio+description&hl=cs&sa=X&ved=0ahUKEwjJkdvU6MjZAhWFyKQKHb7xDF8Q6AEIMTAB#v=onepage&q=Android%20Studio%20description&f=false>

4. Vlastní práce

4.1. Dálkově ovládaný modul s mikroprocesorem

V naší domácnosti používáme celkem tři notebooky zapojené do domácí sítě. Do domácí sítě je i zapojena i síťová laserová tiskárna, z která je z důvodu úspory elektrické energie stále vypnuta síťovým spínačem. V okamžiku, kdy někdo z domácnosti potřebuje tisknout, musí se zvednout od svého pracovního stolu a dojít si zapnout tiskárnu síťovým vypínačem. Neustálé chození zapínat tiskárnu nás obtěžovalo, proto jsem navrhl řešení, jak dálkově zapínat naši laserovou tiskárnu.

Moje řešení je založeno na mikrokontroleru NodeMCU, který je ovládán z našich mobilních telefonů a zapíná nebo vypíná naši tiskárnu prostřednictvím dvou relé, které zapínají nebo vypínají přívod elektrické energie do dvou zásuvek. Do zásuvek je možno připojit jakýkoli jednofázový spotřebič, který má maximální proudový odběr 10A při ~230V AC. (Při zatížení větším proudem, než 10A dojde ke zničení spínacího relé).

Pro ovládání spínacích relé byla zvolena deska NodeMCU. Výhodou této desky, že její součástí je Wi-Fi modul ESP8622 (ESP-12E), který obsahuje čip CP2102 TTL a podporuje protokoly 802.11b/g/n a má zabezpečení WPA/WPA2, není tedy nutný žádný přídavný WiFi modul, který je potřebný u jiných mikroprocesorů - mikrokontrolerů.

Aplikace pro mobilní zařízení (mobilní telefony) je navržena tak, že bude prostřednictvím komunikačního protokolu WiFi (standard IEEE 802.11b) dálkově komunikovat a řídit mikrokontroler NodeMCU.

Instalace aplikace do mobilního zařízení (tablet, mobilní telefon) s operačním systémem Android může být provedena dvěma způsoby:

- První způsob slouží pro vývojáře, kdy jde vytvořen pomocí IDE vytvořen SDK soubor. Tento soubor je následně nahrán do mobilního zařízení, kde je manuálně nainstalován potom, co je povolen mód pro vývojáře.
- Druhým způsobem je stažením z Google Play, kam může být aplikace umístěna po tom co vývojář zaplatí poplatky a projde registrací aplikace.⁴¹

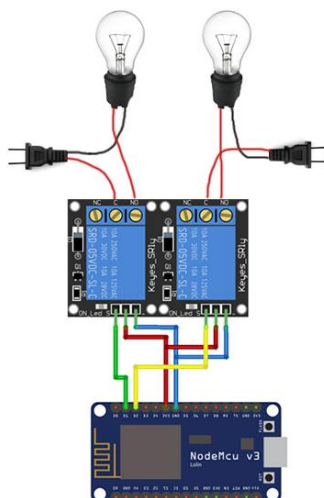
⁴¹ SCHWARTZ Marco, Buttigieg Stefan, Arduino Android Blueprints, Packt Publishing 2014, 30 s., ISBN 978-1-78439-038-9

4.1.1. Schéma zapojení

Na obrázku (Obrázek 20) je schématicky nakreslené napojení spotřebičů na ovládací relé (v našem případě jsou spotřebiče dvě žárovky). Modul relé je připojen k řídicí desce NodeMCU pomocí šesti vodičů:

- 2 červené vodiče vedou napájecí napětí 3,3V. Na desce NodeMCU je použit pin 3,3V a na modulu relé jsou připojeny na kontakty označené VCC.
- 2 modré vodiče jsou zemní (mínus baterie) a vedou z pinu GND. Na desce NodeMCU je použit pin GND a na modulu relé jsou připojeny na kontakty označené GND.
- 1 zelený vodič slouží k ovládní relé č1. Na desce NodeMCU je použit pin D2 (CPIO4) a na modulu relé je připojen na kontakt označení INN1.
- 1 žlutý vodič slouží k ovládní relé č2. Na desce NodeMCU je použit pin D3 (CPIO0) a na modulu relé je připojen na kontakt označení INN2.
- Deska NodeMCU je napájena z externího napájecího převodníku AC-DC ~230V na 5V, kde plus 5V je propojeno je pin 5V desky a mínus pól je připojen k pinu GND.

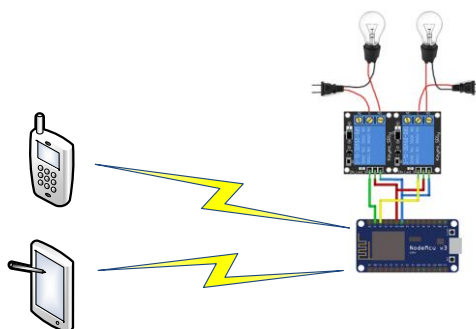
Obrázek 20 Schéma zapojení NodeMCU k relé a k výstupním zařízením



Zdroj: Autor

Na obrázku (Obrázek 21) je schematiky znázorněná komunikace mobilních zařízení s deskou NodeMCU, která spíná obvody relé.

Obrázek 21 Komunikace mobilních zařízení s deskou NodeMCU



Zdroj: Autor

Následující obrázek (Obrázek 22) ukazuje realizované dálkové spínání dvou zásuvek.

Obrázek 22 Realizace dvou zásuvek dálkově spínaných.



Zdroj: Autor

Použité komponenty

Pro realizaci dvou dálkově spínaných zásuvek byly použity následující komponenty:

- Instalační krabice o rozměrech 110x 140x 60 mm
- Dva kusy jedno zásuvek
- Tří žilový kabel, průměr žil 0,75 mm
- Plexisklová destička k přichycení desky NodeMCU, modulu relé a svorkovnic
- Svorkovnice
- Přívodní tří žilový kabel

Postup výroby

- Do víka instalační krabice byl odvrtán otvor pro osazení dvou zásuvek, které byly připevněny k víku vruty.
- Plexisklová destička byla zaříznuta na rozměr, který umožnil její vložení do vnitřku instalační krabice.
- Do plexisklové destičky byly vyvrtány otvory pro přichycení desky k instalační krabici a dále otvory pro přichycení svorkovnice, desky NodeMCU a modulu relé.
- Přes distanční podložky byl přichycen modul relé a deska NodeMCU, do které před vlastní instalací byl nahrán finál ovládací software.
- Na plexisklovou podložky byla přichycena svorkovnice .
- Do instalační krabice byl vyvrtán otvor pro přívodní kabel.
- Jednotlivé žíly přívodního kabelu byly připojeny ke svorkovnici.
- Ze svorkovnice byla přivedena fáze (hnědý vodič) na spínací kontakty relé a na napájecí vstup napájecího převodníku ~230V.
- Ze svorkovnice byla přivedena pracovní nula (modrý vodič) na napájecí vstup napájecího převodníku ~230V a do obou zásuvek (pracovní nula pravá zdířka zásuvek).
- Druhé spínací kontakty byly propojeny se zásuvkami černým vodičem (fáze v levé zdířce zásuvky).
- Do obou zásuvek byl přiveden ze svorkovnice zeleno žlutý vodič (nula) na zemnicí kolíky zásuvek.
- Deska NodeMCU byla připojena na napájení 5V dle schématu zapojení obrázek (Obrázek 23).
- Deska NodeMCU byla propojena s modulem relé dle schématu zapojení obrázek (Obrázek 24).
- Před uzavřením víka krabice se zásuvkami byla přezkoušena funkčnost zařízení.

4.2. Spínací relé

Pro spínání externího zařízení se používají moduly relé, které jsou vyráběny v různých variantách, kdy na modulu je jedno, dvě, čtyři, osm a šestnáct relé. Na vlastním modulu relé se nachází optočlen/y (shodně s počtem relé), které nám galvanicky oddělují řídicí část od spínané části. Galvanické oddělení je důležité z důvodu odfiltrování rušivých signálů a oddělení nízkonapěťové části od části vysokonapěťové, která spíná ~230V.⁴²

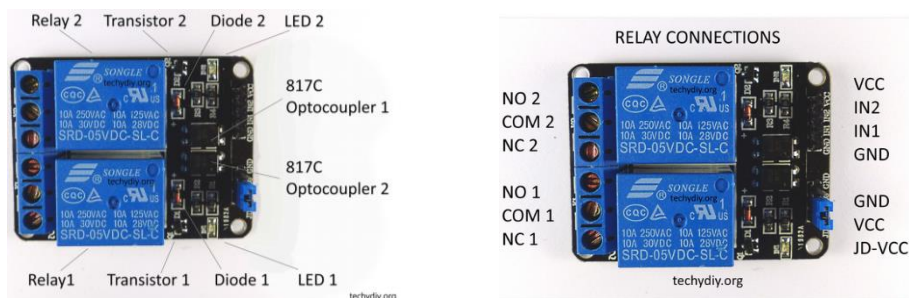
Relé, které jsou standardně používána v reléových modulech mohou spínat až 10 A při ~250V střídavého napětí nebo 10A a 30V stejnosměrného napětí. Spínací napětí optočlenu je 5V. Při jednom sepnutém relé je proudový odběr spínané strany 60mA při 5V.

Následující obrázek (Obrázek 23) ukazuje „spínací modul se dvěma relé Songle SRD-05VDC-SL-C a jejich popisem.“⁴³ Na obrázku (Obrázek 24) je schéma zapojení modulu jednoho relé. Schéma zapojení je indentické pro všechny relé nacházející se na modulu.

⁴² SCHWARTZ Marco, *Arduino Home Automation Projects*, Packt Publishing 2014, 36 s., ISBN 978-1-78398-606-4

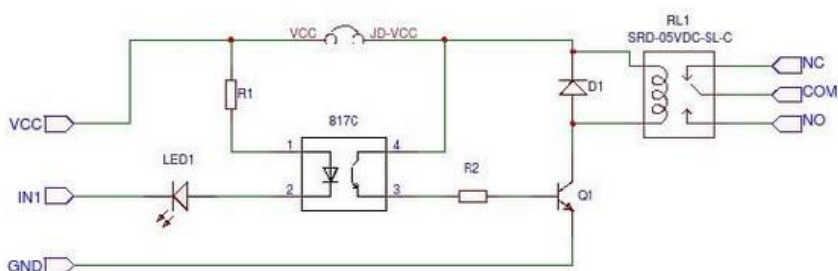
⁴³ GHI electronic, *Songle Reley*, [online], GHI electronics, 1, 2 s.[cit. 2018-2-11], Dostupné z: <https://www.ghielectronics.com/downloads/man/20084141716341001RelayX1.pdf>

Obrázek 23 Spínací modul s dvěmi relé Songle SRD-05VDC-SL-C



Zdroj: Techydiy, *Opto isolated dual relay module (Arduino)*, [cit. 2018-2-11], Dostupné z: <http://www.techydiy.org/opto-isolated-dual-relay-module-arduino/>

Obrázek 24 Schéma zapojení modulu s jedním relé



Zdroj: Techydiy, *Opto isolated dual relay module (Arduino)*, [cit. 2018-2-11], Dostupné z: <http://www.techydiy.org/opto-isolated-dual-relay-module-arduino/>

Technický list s popisem parametrů relé „Songle SRD-05VDC-SL-C“⁴⁴ je dostupný na internetových stránkách společnosti GHI electronics na odkaze:⁴⁵ <https://www.ghielectronics.com/downloads/man/20084141716341001RelayX1.pdf>

⁴⁴ GHI electronic, *Songle Reley*, [online], GHI electronics, 1, 2 s.[cit. 2018-2-11], Dostupné z: <https://www.ghielectronics.com/downloads/man/20084141716341001RelayX1.pdf>

⁴⁵ GHI electronic, *Songle Reley*, [online], GHI electronics, 1, 2 s.[cit. 2018-2-11], Dostupné z: <https://www.ghielectronics.com/downloads/man/20084141716341001RelayX1.pdf>

4.3. Napájení NodeMCU a spínacích relé

Mikrokontroler NodeMCu má napájecí napětí 5V DC, typická proudová spotřeba je udávána 80mA, ale maximální spotřeba je 350 mA. Proudové zatížení jednotlivých pinů je maximálně 40 mA.

Mikrokontroler NodeMCu můžeme napájet:

- připojením na baterii 5 až 12V, kterou propojíme k pinům VIN nebo 5V (označení dle typu použité desky) – plus pól a GND – minus pól,
- napájení přes USB port,
- připojením na napájecí převodník AC-DC 220V na 12V nebo 5V.

Pro realizaci dálkově ovládaného zařízení je vhodné použít napájecí převodník, protože jej instalujeme do modulu spolu s NodeMCU a spínacím relé. Převodník bude napájen ~230V, které budou i napájet spínané zařízení zásuvky. Jako jedno z možných řešení je použití napájecí modulu AC-DC ~230V na 12V a 5V, který dává při 5V až maximálně 1A. Výstupní proud 1A je dostatečný na to, aby nám pokryl proudový doběr NodeMCU a releového modulu tj. max proudový odběr NodeMCU 350mA + 2x 60mA releového modulu = 470 mA při maximálních hodnotách odběrů uvedených zařízení.

Pro proudový odběr 470mA při 5V můžeme například použít napájecí převodník AC-DC ~230V na 12V a 5V obrázek (obrázek 25), který je plně izolovaný s ochranou proti nadměrnému proudu a ochranou proti zkratu, napájení je v rozsahu AC ~85 ~ 265V. Převodník má dva výstupy stejnosměrného napětí DC 12V/1A a 5V/1A. Výstupní stejnosměrné napětí je regulováno, použit i filtrační obvod EMI. Rozměry jsou 7,5 x4.0 x 4.0 cm. Tabulka (Tabulka 2) ukazuje přehledně parametry napájecího převodníku AC-DC 220V na 12V a 5V

Obrázek 25 Napájecí převodník AC-DC 220V na 12V a 5V



Zdroj: Banggood, *12V 5V Fully Isolated Switching Power Supply AC-DC Module 220V to 12V* [online], [2018-2-25], Dostupné z: https://www.banggood.com/12V-5V-Fully-Isolated-Switching-Power-Supply-AC-DC-Module-220V-to-12V-p-1000431.html?rmmds=search&cur_warehouse=CN

Tabulka 2 Parametry napájecího převodníku AC-DC 220V na 12V a 5V

| Výstupní napětí | Minimální proud | Maximální proud | Proudová špička | Výstupní napětí - rozsah | Odběr |
|-----------------|-----------------|-----------------|-----------------|--------------------------|-------|
| +12V | 0A | 1A | 1.2A | 11.80~12.5V | 150mV |
| +5V | 0A | 1A | 1.2A | 4.90~5.1V | 80mV |

Zdroj: Banggood, *12V 5V Fully Isolated Switching Power Supply AC-DC Module 220V to 12V* [online], [2018-2-25], Dostupné z: https://www.banggood.com/12V-5V-Fully-Isolated-Switching-Power-Supply-AC-DC-Module-220V-to-12V-p-1000431.html?rmmds=search&cur_warehouse=CN

4.4. Řídicí program pro mikrokontroler NodeMCU

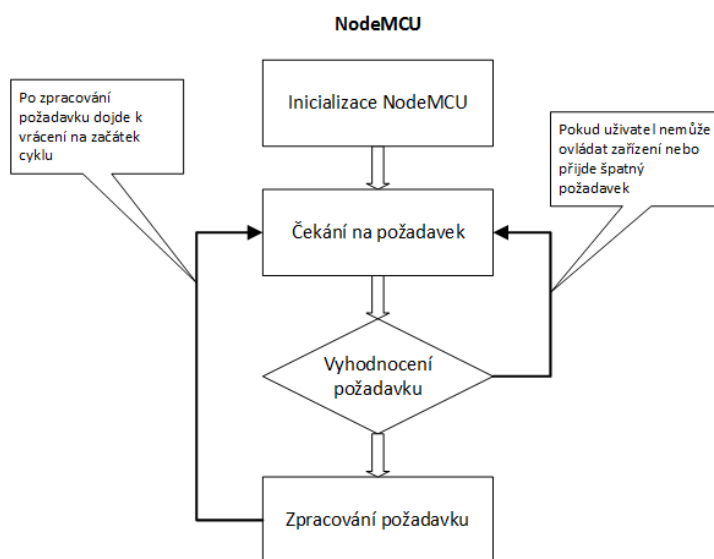
Program pro NodeMcu Lua, byl napsán ve Visual studio 2015 s rozšiřujícím doplňkem na psaní kódu pro Arduino Studio 1.6.13. Aby bylo umožněno vytvářet program v tomto prostředí pro desku NodeMCU je nezbytné nainstalovat knihovnu ESP8266WebServer. Tento doplněk lze ztáhnout skrze Arduino Studio nebo z webových stránek <https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WebServer> [cit. 2018-1-8]. Vlastní kód programu je uveden v příloze Příloha II. Zdrojový kód pro NodeMCU.

4.4.1. Struktura programu

Na obrázku (Obrázek 26) je znázorněn vývojový cyklus programu pro NodeMCU.

Z tohoto obrázku je vidět, jak jde vývoj aplikace od počítače inicializace až po zpracování příchozího požadavku, kdy po inicializaci NodeMCU program čeká ve smyčce – loop(), dokud neobdrží požadavek. Příchozí požadavek vyhodnotí, zdali je požadavek validní a zda uživatel má patřičná práva na požadování zpracování požadavku. Pokud vyhodnocení požadavku proběhne úspěšně je zpracován. Po zpracování požadavku zařízení se vrací na začátek a opět čeká na příchozí požadavek.

Obrázek 26 Struktura programu pro NodeMCU



Zdroj: Autor

4.4.2. Inicializace základních prvků

Na začátku programu je jako první vložena knihovna ESP8266WebServer.h pomocí příkazu `#include`. Tato knihovna obsahuje řadu metod, díky kterým je umožněno nastavit NodeMCU jako server a získat metody ke zpracování http požadavků přicházejících od uživatele.

```
#include <ESP8266WebServer.h>
```

Aby bylo možno využít funkce ESP8266WebServer je nezbytné deklarovat objekt, kde argument pro konstruktor slouží číslo portu. Číslo portu 80 je zvoleno záměrně, protože se jedná o základní hodnotu portu pro http.

```
ESP8266WebServer server(80);
```

Dalším krokem je nastavení základních parametrů WiFi, kterými jsou ssid a password. Tyto dva parametry jsou vloženy do stejnojmenných proměnných. Protože není žádoucí tyto proměnné přepisovat mají parametr `const` díky kterému jsou pouze ke čtení, avšak nikoliv k zápisu. Typ proměnných je `char`, díky kterému je možno uvést do proměnných znaky i čísla.

```
const char *ssid = "WIFI_NAME";  
const char *password = "heslo123";
```

Pro ovládání výstupních pinů je nezbytné nadefinovat jednotlivé piny, které mají být využívány, to provedeme pomocí příkazu `#define`. Tímto příkazem dojde k nadefinování proměnné patřičným pinem.

```
#define pin_1 4  
#define pin_2 5
```

Pro zpracování příchozích http požadavků je musíme zavolat metodu `HandleClient`. Tato metoda je obsažena v objektu `server`, jež můžeme používat díky předešlému vložení knihovny. Je potřeba vložit tuto funkci do funkce `loop`, která je hlavním cyklem programu.

Tato funkce je volána stále dokola, díky čemuž je program dokáže zpracovávat další přicházející http požadavky.

```
void loop() { server.handleClient(); }
```

4.4.3. Obsluha požadavků

NodeMCU dostává požadavky na provedení z mobilní aplikace v URL formátu. Na obsluhu těchto požadavků je zde připravená funkce `void handleRoot()`. Formát URL je následující `http://192.168.4.1:80/?req=vystup&key=klíč`. Jednotlivé parametry URL jsou následně rozděleny na samostatné segmenty a vloženy do proměnných, kdy parametr `key` nám nese klíč mobilní aplikace. Klíč je zde vložen záměrně kvůli zabezpečení. Pokud by se na zařízení připojil někdo, kdo by měl rovněž stejnou aplikaci, mohl by se snadno zmocnit daného zařízení. Díky přenesené hodnotě `key` může domácnost ovládat pouze uživatel, jenž má shodný klíč s klíčem, jež je vložen do zařízení proměnné `*key` v programu pro NodeMCU. Druhým parametrem, který je obsažený v URL, je `req`. Tento parametr nese požadavek na pin, jež má být obsloužen.

Každý požadavek, který je obdržén z mobilní aplikace je prvně uložen do proměnné. Tyto proměnné jsou zde dvě, a to `request` a `requestKey`. `request` nese požadavek na výstup, jenž má být spuštěný. `requestKey` nese klíč aplikace, ze které byl požadavek odeslán. Zachycení těchto parametrů se provádí pomocí metody `arg()`, jež získáme z objektu `server`, který je popsán v kapitole 4.4.2.

```
String request = server.arg("req");
```

```
String requestKey = server.arg("key");
```

Pro kontrolu zároveň upozornění uživatele, o tom, že se připojuje k jinému zařízení, než ke které má přístup díky unikátnímu klíči, který je vložený jak do programu NodeMCU, tak do mobilní aplikace. Pro tento účel je zde vložena podmínka, která kontroluje jednotlivé klíče proti sobě.

Pokud by se přijatý klíč neshodoval s klíčem aplikace, nedojde ke zpracování příchozího požadavku z aplikace. Pomocí metody `send`, která je získána z objektu `server`, je odeslaná chyba na zařízení o neprovedení požadavku.

```
if(requestKey != key) { server.send(200, "text/text", "Invalid key"); return; }
```

Pro obsluhu výstupu jsou zde podmínky, jenž hledají shodu. V případě shody hodnoty obsažené v proměnné request dojde k zavolání jedné ze dvou funkcí.

```
If(request == "1on") { turnOn(pin_1); }  
else if (request == "1off") { turnOff(pin_1); }
```

Pokud nedošlo k chybě a vše se provedlo, jak mělo, tak je pomocí metody send odeslán text o úspěšném provedení

```
server.send(200, "text/text", "Success");
```

4.4.4. Funkce na ovládání výstupů

Funkce jsou specifické pro každého uživatele dle typu využití prvků v domácnosti. V této práci tedy využiji k demonstrování funkcí napsaných pro spínání relátek.

Po přijetí požadavku na obsluhu výstupu a jeho validaci je stav výstupu změněn pomocí funkce digitalWrite(). Tato funkce má dva vstupní parametry. Prvním je výstupní pin a druhým stav, do kterého má být přepnut. V této aplikaci se využívají dva stavy, a to LOW pro sepnutí a HIGH pro vypnutí relátka.

```
turnOn(int pin) { digitalWrite(pin_1,LOW);}
```

4.4.5. Spuštění WIFI

Pro využití WIFI byla založena funkce startWifi(). V této funkci předáme parametry potřebné k spuštění WIFI v režimu access point, neboli přístupový bod, pomocí metody WiFi.softAP(ssid, password). Hodnoty ssid a password byli definovány v kapitole 4.4.2 a nyní tyto hodnoty jsou pouze předány.

```
WiFi.softAP(ssid, password);
```

Pro přistupování zařízení na vytvořený access point je potřeba rovněž definovat IP adresu. Předání IP adresy funkci `UPAdress` je provedeno prostřednictvím proměnné `ipAdress`, která nese základní hodnotu IP adresy, kterou dostane z metody `softAPIP()` a která se nachází v objektu `WiFi`. Základní IP adresa, kterou `softAPIP()` přidělí je 192.168.4.1.

```
IPAddress ipAddress = WiFi.softAPIP();
```

K zachycení jednotlivých parametrů řetězce je zde použita lambda funkce, která zpracuje jednotlivé parametry url a následně je předá do funkce `handleRoot()`

```
server.on("/", [= ](void) { return handleRoot(); });
```

Funkce `startWifi()` je následně inicializována ve funkci `setup()`. Je umístěn záměrně ve funkci `setup()`, protože funkce `setup()` se spouští implicitně pouze jednou na začátku při zapnutí zařízení a jeho inicializaci. Díky této vlastnosti funkce `setup()` jsou potřebné parametry pro spuštění access point inicializovány pouze při spuštění programu, tedy při zapnutí desky. Ve funkci `setup()` jsou dále umístěny definice pinů na výstup pomocí funkce `pinMode()`, kde `pinMode()` má dva vstupní parametry. Prvním parametrem je pin, pro který má platit definice a druhým parametrem je funkce pinu. Ta může být buďto `OUTPUT` pro výstup nebo `INPUT` pro vstup.

```
void setup() {  
    Serial.begin(9600);  
    pinMode(pin_1, OUTPUT);  
    pinMode(pin_2, OUTPUT);  
    startWifi();  
}
```


4.5. Ovládací aplikace pro operační systém Android

Program aplikace pro Android byl napsán v Android Studio 2.3.3, které obsahuje funkcionality, díky kterým je snadnější a rychlejší vytvářet samotný kód pro aplikaci a zároveň umožňuje vygenerovat a zaneš aplikaci do Google Play. Vlastní kód programu je uveden v příloze Příloha I. Zdrojový kód aplikace pro OS Android.

4.5.1. Ovládací obrazovka

Ovládací obrazovka slouží k vybírání požadavků, které mají být odeslány na NodeMCU. Je zde umístěno několik druhů tlačítek, které slouží k rozdílným požadavkům. Jelikož princip je pro všechna tlačítka stejný, budou jednotlivé funkce tlačítek podrobněji popsány v dílčích kapitolách. O zachycení požadavků jednotlivých tlačítek se stará metoda `listenerOnButton()`. Obsahuje definice jednotlivých tlačítek a jejich funkcionality. Metoda je inicializována při zapnutí aplikace ve v metodě `OnCreate()`.

Metoda `listenerOnButton()` vyčkává, dokud uživatel nezmáčkne libovolné tlačítko. Podle id tlačítka, které obdrží z aktivity, kde vznikl daný požadavek na obsluhu tlačítka je zavolána funkcionality pro provedení daného požadavku. Jednotlivé id jsou definované na začátku funkce `listenerOnButton()` a jsou přiděleny proměnným s parametrem o jaký vstup se jedná.⁴⁶

```
button1 = (Button) findViewById(R.id.button1);
```

Proměnné mají přiřazenou svojí metodu, jež má za účel kontrolovat, zda při zavolání není směřován požadavek právě na ní `setOnClickListener()`.

```
button1.setOnClickListener(new OnClickListener() {...})
```

⁴⁶ MEIER Reto, *Professional Android 4 Application Development*, John Wiley & Sons P&T 2012, 97 s., ISBN: 9781118223857

Pokud je volání směrováno právě na přidělené tlačítko, dojde ke vstupu do funkcionality. Uvnitř funkcionality se nachází další metoda `onClick()`. Ta hlídá, zda příchozí volání vzniklo zmáčknutím tlačítka. V takovémto případě jsou zpracovány dílčí požadavky vnitřku této metody

```
public void onClick(View v) {...}
```

4.5.2. Design

Vzhled aplikace obrázek (Obrázek 27) jsem vytvořil na základě uživatelské zkušenosti s obdobnými aplikacemi. Důraz byl kladený na jednoduchost a přehlednost aplikace.

Obrázek 27 Ovládací obrazovka na mobilním zařízení



Zdroj: Autor

4.5.3. Bezdrátová komunikace s NodeMCU

Ke zpracování a předání požadavků uživatele slouží třída `volleyConnection`. V této třídě je obsažena metoda `sendRequest`. Tato metoda slouží k sestavení URL s požadavky na zpracování a zařazení URL do fronty k odeslání. Metoda `sendRequest` má vstupní parametr typu `string`. Tento `string` obsahuje jednotlivé parametry, které jsou sestaveny a předány po stisknutí tlačítka uživatelem. Hodnota přijatá na vstupu je uložena do proměnné a následně předána metodě `stringRequest`. Metoda `stringRequest` je získána z knihovny `com.android.volley.toolbox.stringrequest` a má 4 vstupní parametry. Prvním parametrem je metoda. Pro účel této aplikace je použita metoda `GET`. Druhým parametrem je URL. URL je složené z IP adresy a portu, na kterém poslouchá zařízení `NodeMCU` a ze vstupního parametru metody `sendRequest`, který vložený v proměnné `endOfUrl`. IP adresa a port jsou nadefinovány do proměnných na začátku třídy `VolleyConnection` a následně jsou spojeny v proměnné URL. Třetím parametrem je odpověď zařízení a čtvrtým parametrem je chybová odpověď.

```
final private String ipAddress = "192.168.4.1"; //adresa arduino  
final private String portNumber = "80";  
final private String url = "http://" + ipAddress + ":" + portNumber + "/?";
```

```
StringRequest request = new StringRequest(Request.Method.GET, url+endOfUrl,  
new Response.Listener<String>() {  
    @Override  
    public void onResponse(String response) {  
        respondText.setText("Přepnuto");  
    }  
}, new Response.ErrorListener() { //pozdeji smazat  
    @Override  
    public void onErrorResponse(VolleyError error) {  
        respondText.setText("Chyba požadavku: " + error);  
    }  
});  
queue.add(request);
```

Metoda `stringRequest` je vložena do proměnné `request`. Proměnná `request` je následně vložena do `requestQueue` kde je zpracována v okamžiku, kdy se dostane na řadu. Fronta je získána z knihovny `import com.android.volley.RequestQueue`. Je využívána z toho důvodu, pokud by uživatel v jeden okamžik zmáčkl více tlačítek, tak aby došlo ke zpracování všech požadavků, jež uživatel požaduje. V případě, že by nebyla zde umístěna fronta, tak by došlo pouze ke zpracování posledního požadavku a ostatní by byly vynechány.

```
queue = Volley.newRequestQueue(activity);
```

4.5.4. Odeslání požadavku na NodeMCU

Odesílání požadavku na zařízení NodeMCU je provedeno potom, co je uživatelem stisknuto jedno ze šesti tlačítek na ovládací obrazovce. Tlačítka jsou zastoupena párově, tedy každé tlačítko, jež odesílá požadavek na NodeMCU o přepnutí stavu výstupu na On má rovněž párově k sobě tlačítko, jež slouží k přepnutí téhož tlačítka do stavu Off. Tlačítka jsou takto v páru z důvodu, pokud by se nepodařilo realizovat požadavek na zařízení NodeMCU, aby uživatel stále měl možnost zaslat požadavek, který se nepodařilo provést.

Při požadavku na přepnutí stavu výstupu na NodeMCU proběhne proces viz kapitola 4.5.1. V metodě `onClick` je pro tlačítka, jež přepínají stav NodeMCU na On nebo Off, umístěna funkce `handleButton()`.

Tato metoda má vstupní parametr typu `String`. Tento vstup v sobě nese řetězec, který obsahuje číslo a akci požadovanou uživatelem.

```
public void onClick(View v) {  
    handleButton("1on");}
```

Při zavolání metody `handleButton(String button)` dojde k převzetí vstupní řetězce a vložení této hodnoty do proměnné `button`. Tato proměnná je následně předána společně s proměnnou `apiKey` funkci `sendRequest()`, která je získána ze třídy `VolleyConnection`, pro jejíž zavolání je definována proměnná `connection`. Třída je popsána v kapitole 4.5.3.

```
public void handleButton(int button) {  
    String apiKey = sharedPreferences.getString(sharedPref, null);  
    connection.sendRequest("req="+button+"&key="+apiKey);}
```

Proměnná `apiKey` v sobě nese uložené heslo aplikace, jež je uloženo v daném zařízení. Heslo je vloženo do této proměnné pomocí metody `getString`. Pomocí sdílených preferencí je umožněno ukládat do zařízení různé údaje, u kterých je požadováno, aby zůstali zachováni i po vypnutí zařízení. Sdílené preference jsou získány a vloženy do proměnné `sharedPref` při zavolání metody `onCreate()` pomocí metody `getSharedPreferences`, jež je poskytnuta objektem `sharedPreferences`.⁴⁷

```
sharedPreferences = getSharedPreferences(sharedPref, Context.MODE_PRIVATE);
```

4.5.5. Změna hesla mobilní aplikace

V případě, že by se uživatel mobilní aplikace rozhodl změnit heslo, které je zasíláno společně s parametrem požadavku na obsluhu výstupu NodeMCU, je mu tak umožněno v při stisknutí tlačítka `Password`, jež se nachází záložce `Settings`. Po stisknutí tlačítka `Password` dochází ke skrytí tlačítka `Settings` a je zobrazeno textové pole, a dvě tlačítka `Save` a `Cancel`. Na jednotlivých tlačítkách jsou umístěny metody `setOnClickListener`. Pokud uživatel zmáčkne tlačítko `Cancel` dojde ke skrytí tlačítek `Save` a `Cancel`, ke skrytí vstupu pro vložení nového hesla a k opětovnému zobrazení tlačítka `Settings`. Zobrazení a skrytí tlačítek a vstupu dochází za použití metody `setVisibility`. Této metodě jsou předávány parametry `View.VISIBLE` pro zobrazení a `View.GONE` pro skrytí. V případě požadavku uživatele na uložení hesla je prvně vstup převeden na `string` pomocí metody `toString()`. Převedený vstup je vloženo do proměnné `pwd` typu `String`. Obsah této proměnné je následně uložen do mobilního zařízení pomocí použití metody `editor`, která je získána z objektu `SharedPreferences`. Po uložení hesla dochází k opětovnému zobrazení tlačítka `Settings` a ke skrytí tlačítek `Save` a `Cancel` a vstupu pro heslo.

```
String pwd = passwordField.getText().toString();  
SharedPreferences.Editor editor = sharedPreferences.edit();  
editor.putString(sharedPref, pwd);
```

⁴⁷ WEI-MENG Lee, Beginning Android Application Development, Lee, Wei-Meng John Wiley & Sons, , Incorporated 2011 EDICE: Wrox beginning guides., 229 s., ISBN: 9781118087299

```
editor.commit();
```

```
btnSettings.setVisibility(View.VISIBLE);
```

```
btnSubmit.setVisibility(View.GONE);
```

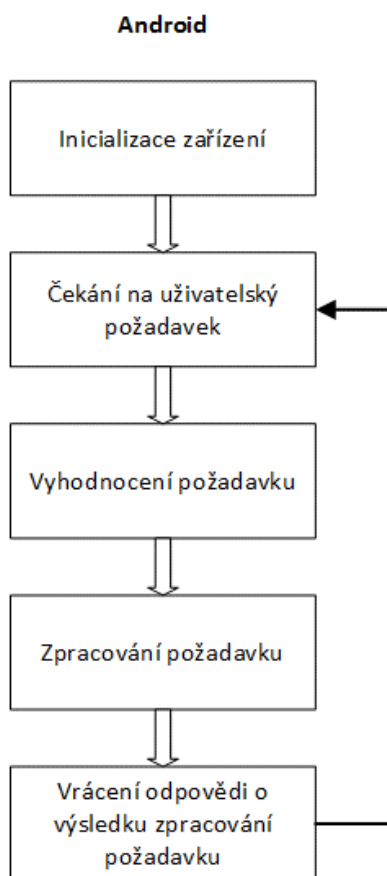
```
passwordField.setVisibility(View.GONE);
```

```
btnCancel.setVisibility(View.GONE);
```

4.5.6. Struktura programu

Na obrázku (Obrázek 28) je znázorněn vývojový cyklus mobilní aplikace pro operační systém Android. Je zde znázorněn cyklus aplikace po jejím spuštění, kdy dojde k inicializaci aplikace. Poté co je aplikace spuštěna vyčkává na uživatele než provede svůj požadavek kliknutím na některé tlačítko. Poté co klikne na některé z tlačítek, dojde k vyhodnocení, o jaké tlačítko se jedná. Na základě vybraného tlačítka je zpracován požadavek. Po zpracování požadavku dojde k vrácení odpovědi o úspěšném nebo neúspěšném zpracování požadavku a program se opět vrací na začátek a čeká na další požadavek.

Obrázek 28 Struktura mobilní aplikace



5. Závěr

Cílem této bakalářské práce bylo identifikovat možnosti využití mikropočítačů řízených prostřednictvím mobilních zařízení, pro vzdálené ovládání domácnosti a dále navržení a realizování konkrétní aplikace pro domácnost.

Práce se dělí na dvě části, a to teoretickou a praktickou. Teoretická část se zabývá potencionálním využitím dálkového ovládání domácností, ale i jiných míst. Následuje sestavení omezení, která jsou důležitá pro následný výběr vhodného zařízení, které bude sloužit k ovládání koncových elektronických zařízení. V kapitole Mikroprocesory byly představeny mikroprocesory, které je možné použít k ovládání spotřebičů v domácnosti. Jednalo se mikropočítače, resp. mikrokontrolery, jednočipové počítače a programovatelné logické automaty PLC, u kterých byly uvedeny jejich základní vlastnosti a technické parametry. Na základě těchto parametrů je vybrán vhodný mikropočítač, operační systém mobilního zařízení, pro který bude vytvořena aplikace pro uživatele, pomocí které bude mikropočítač ovládán a vhodný způsob přenosu dat z aplikace do mikropočítače. Vybraný mikropočítač a operační systém jsou důkladně popsány. Je vybráno vhodné prostředí pro vytvoření programu pro mikropočítač a pro vytvoření aplikace na mobilní zařízení.

V praktické části je prvně popsáno schéma zapojení mikropočítače k výstupním zařízením. Následně je popsán a schématicky znázorněn proces programu pro mikropočítač a pro mobilní aplikaci. Na závěr je program pro mikropočítač a mobilní aplikaci podrobně popsán a vysvětlen. Celé ovládání bylo sestaveno a odzkoušeno.

Praktická část obsahuje dvě části. První část vysvětluje jednotlivé části programu pro vybraný mikropočítač NodeMCU. Celý vývoj programu je znázorněn ve schématu. Druhá část praktické části vysvětluje jednotlivé části kódu pro mobilní aplikaci pro operační systém Android. Je zde schéma vývoje aplikace a náhled vytvořeného designu.

Výsledkem praktické části je sestavené a naprogramované zařízení NodeMCU, které je propojené pomocí WiFi s mobilním zařízením s operačním systémem Android pro, které byla naprogramována aplikace, pomocí které uživatel zadává požadavky na obsluhu do zařízení NodeMCU.

6. Seznam použitých zdrojů

Tištěné zdroje

GHI electronic, *Songle Reley*, [online], GHI electronics, 1, 2 s.[cit. 2018-2-11], Dostupné z:

<https://www.ghielectronics.com/downloads/man/20084141716341001RelayX1.pdf>

KINDLE Edition, *NodeMCU Development Workshop* [online], PE Press; Auflage: 1 (31. Juli 2015), 117 s, ASIN: B0135CUT3M, Dostupné z:

<http://www.ebook777.com/nodemcu-development-workshop/>

KUHNEL Claus, *BASIC Stamp: An Introduction to Microcontrollers* [online], 197 s. [cit. 2018-2-09], Newnes; 2 edition (August 4, 2000), ISBN-13: 978-0750672450, dostupné z: https://books.google.cz/books?id=sQ9AigIIOKgC&pg=PA34&dq=BS2-IC&hl=cs&sa=X&ved=0ahUKEwi6hKPOt8jZAhVE_aQKHf0eBFAQ6AEIKDAA#v=onepage&q=BS2-IC&f=false

MEIER Reto, *Professional Android 4 Application Development*, John Wiley & Sons P&T 2012, 868 s., ISBN: 9781118223857

MOLLOY Derek, *Exploring Raspberry Pi* [online], Wiley; 1 edition (June 13, 2016), 722 s., [cit. 2018-02-25], ISBN: 978-1-119-18868-1, Dostupné z:

<https://books.google.cz/books?id=K-phDAAAQBAJ&pg=PT704&lpg=PT703&focus=viewport&dq=nodemcu+description&hl=cs#v=onepage&q=nodemcu%20description&f=false>

ONUR Murat Yener, Dundar, *Expert Android Studio* [online], John Wiley & Sons 2016 , 459 s., [cit. 2018-1-08], ISDN: 9781119110712, Dostupné z:

<https://books.google.cz/books?id=mCjqDAAAQBAJ&pg=PA337&dq=Android+Studio+description&hl=cs&sa=X&ved=0ahUKEwjJkdvU6MjZAhWfYKQKHb7xDF8Q6AEIMTAB#v=onepage&q=Android%20Studio%20description&f=false>

PRADEEKA Senevirantne, *Builging Arduino PLCs, The essntial techniques you need to develop Arduino-based PLCs*, Apress, 183 s., ISBN-13 (electronic): 978-1-4842-2632-2

SATAPATHY, Suresh Chandra, Bhateja, Vikrant, Das, Swagatam, *Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016, Volume 2*, [online], Springer Singapore, 268 s., [cit. 2018-02-25]. Dostupné z:
<https://books.google.cz/books?id=6AxEDwAAQBAJ&pg=PA268&dq=NodeMcu+Lua+WI-FI+ESP8266&hl=cs&sa=X&ved=0ahUKEwj7uric-cDZAhWJh7QKHeEfCtAQ6AEIRDAE#v=onepage&q=NodeMcu%20Lua%20WI-FI%20ESP8266&f=false>

SCHWARTZ Marco, *Arduino Home Automation Projects*, Packt Publishing 2014, 132 s., ISBN 978-1-78398-606-4

SCHWARTZ Marco, Buttigieg Stefan, *Arduino Android Blueprints*, Packt Publishing 2014, 250 s., ISBN 978-1-78439-038-9

SCHWARTZ Marko, *Arduino Home Automatization Projects*, Packt Publishing Lth. 2014, 132 s., [cit. 2018-1-02], ISDN: 9781783986071

UDAVASHANKARA V, Mallikarjunaswamy M S, *8051 Microcontroller, Hardware, Software and Applications* [online], New Delhi : Tata McGraw-Hill, ©2009., 304 s., [cit. 2018-1-15], ISBN: 9780070086814, Dostupné z:
https://books.google.cz/books?id=iru1w4iRx_YC&pg=PA297&dq=8051+Microcontroller,+Hardware,+Software+and+Applications&hl=cs&sa=X&ved=0ahUKEwiW082PzsjZAhXFyKQKHZwLDwMQ6AEIKDAA#v=onepage&q=8051%20Microcontroller%2C%20Hardware%2C%20Software%20and%20Applications&f=false

UJBÁNYAI Miroslav, *Programujeme pro Android* [online], Grada Publishing a.s. 2012, 197 s. [cit. 2018-1-10], ISBN 9788024779836, Dostupné z:
<https://books.google.cz/books?id=o6kHGXua8oQC&printsec=frontcover&dq=Android&hl=cs&sa=X&ved=0ahUKEwui1PbA4MjZAhWBy6QKHeHcAhIQ6AEIKDAA#v=onepage&q=Android&f=false>

WEI-MENG Lee, *Beginning Android Application Development*, Lee, Wei-Meng John
Wiley & Sons, , Incorporated 2011 EDICE: Wrox beginning guides., 450 s., ISBN:
9781118087299

Elektronické zdroje:

BASIX, *BX24P* [online], Baxis, (na následující odkazu jsou dostupné úplné specifikace mikrokontroleru BX24P) [cit. 2018-1-4], Dostupné z:

<http://www.basicx.com/Products/BX-24/bx24specs.htm>

BLUETOOTH SIG Proprietary, *Bluetooth Core Specification v 5.0*, 2822 s. Bluetooth SIG
Proprietary Dec 06 2016, pdf dokument [stažen 2018-1-11], Dostupné z:

<https://www.bluetooth.com/specifications/bluetooth-core-specification>

ELECTRONICWINGS, *NodeMCU Development Kit/Board* [online], ElectronicWings
2017, [cit. 2018-2-19] Dostupné z:

<http://www.electronicwings.com/nodemcu/nodemcu-development-kitboard>

ELECTRONICWINGS, *NodeMCU Development Kit/Board, GPIO* [online],
ElectronicWings 2017, [cit. 2018-2-19] Dostupné z

<http://www.electronicwings.com/nodemcu/nodemcu-gpio-with-esplorer-ide>

ELECTRONICWINGS, *NodeMCU Development Kit/Board, I2C* [online],
ElectronicWings 2017, [cit. 2018-2-19] Dostupné z

<http://www.electronicwings.com/nodemcu/nodemcu-i2c-with-esplorer-ide>

ELECTRONICWINGS, *NodeMCU Development Kit/Board, IDE* [online],
ElectronicWings 2017, [cit. 2018-2-19] Dostupné z

<http://www.electronicwings.com/nodemcu/nodemcu-adc-with-esplorer-ide>

- ELECTRONICWINGS, *NodeMCU Development Kit/Board, SPI* [online],
ElectronicWings 2017, [cit. 2018-2-19] Dostupné z
<http://www.electronicwings.com/nodemcu/nodemcu-spi-with-arduino-ide>
- ELECTRONICWINGS, *NodeMCU Development Kit/Board, UART* [online],
ElectronicWings 2017, [cit. 2018-2-19] Dostupné z
<http://www.electronicwings.com/nodemcu/nodemcu-uart-with-esplorer-ide>
- ESPRESSIF IoT Team, *ESP8266EX Datasheet version 5.8 specification*, Espressif IoT
Team 2018, 28 s., pdf dokument [stažen 2018-2-1], Dostupný na:
https://www.espressif.com/en/support/download/documents?keys=&field_type_tid%5B%5D=14
- ESPRESSIF Systems IOT Team, *ESP8266EX Datasheet Version 4.3*, Espressif Systems
2015, 31 s., pdf dokument [stažen 2018-2-25], Dostupné z:
http://www.santy.cz/data/file/schemata/SA0287/SA0287-ESP8266_2.pdf
- FARNELL, *STM32F103C8T6 - ARM Mikrokontroler* [online], Farnell 2018, [cit. 2018-1-26], Dostupné z: <http://cz.farnell.com/stmicroelectronics/stm32f103c8t6/mcu-32bit-cortex-m3-72mhz-lqfp/dp/1447637>
- GARTNER, *Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017* [online], Gartner Feb 22 2018 [cit. 2018-2-24], Dostupné z: <https://www.gartner.com/newsroom/id/3859963>
- GHI electronic, *Songle Reley*, [online], GHI electronics, 2 s., pdf dokument [stažen. 2018-2-11], Dostupné z:
<https://www.ghielectronics.com/downloads/man/20084141716341001RelayX1.pdf>
- HW SERVER, *Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroleru PIC16F877* [online], HW server s.r.o. 2000, [cit. 2018-1-14], Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojenu-externi-EEPROM-24LC256>

IEEE Standard Association, *IEEE Standard* [online], IEEE Standard Association 2016, [cit. 2018-2-02], dostupné z: <https://standards.ieee.org/findstds/standard/802.11-2016.html>

INTELLIJ IDEA, *IntelliJ IDEA* [online], *IntelliJ IDEA 22018*, [cit. 2018-01-09], Dostupné z: <https://www.jetbrains.com/idea/?fromMenu>

IT SLOVNÍK, *Mikrokontroler* [online], IT-Slovník 2018, [cit. 2018-1-6], Dostupné z: <https://it-slovník.cz/pojem/mikrokontroler>

PARALAX INC, *BASIC Stamp 2 Microcontroller Module* [online], Parallax Inc 2018, [cit. 2018-123], Dostupné z: <https://www.parallax.com/product/bs2-ic>

SIEMENS AG, *SIMATIC LOGO! Manual - Industry Support Siemens* [online], Siemens AG 2001-2003, 27 s., pdf dokument [staženo 2018-2-20], Dostupné z: https://cache.industry.siemens.com/dl/files/461/16527461/att_82564/v1/Logo_e.pdf

ST MICROELECTRONIC, *STM32F0x1* [online], ST Microelectronic 2018, výrovice [cit. 2018-1-28], Dostupné z: http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-mainstream-mcus/stm32f0-series/stm32f0x1.html?querycriteria=productId=LN7

TEXAX INSTRUMENT, *CD74HC4067 High Speed CMOS 16-Channel Analog Multiplexer/Demultiplexer* [online], Texas Instrument 2017, [cit.2018-2-17], Dostupné z: <http://www.ti.com/product/CD74HC4067?keyMatch=CD74hc4067&tisearch=Search--Everything>

TS BOHEMIA, *Raspberry Pi 3 Model B* [online], TS Bohemia 2018, [cit. 2018-2-11], Dohledatelné z: https://www.tsbohemia.cz/raspberry-pi-3-model-b_d239934.html

ŽIVĚ, PROKOP Mirek, *Wi-Fi: Jak si zajistit velké pokrytí, rychlost a silný signál*, [online], ŽIVĚ 5. února 2014, [cit. 2018-1-20], Dostupné z:

<https://www.zive.cz/clanky/wi-fi-jak-si-zajistit-velke-pokryti-rychlost-a-silny-signal/anteny-a-jejich-nastaveni-deformace-a-ztrata-signalu-mereni/sc-3-a-172347-ch-90933/>

ŽIVĚ, PROKOP Mirek, *Wi-Fi: Jak si zajistit velké pokrytí, rychlost a silný signál*, [online], ŽIVĚ 5. února 2014, [cit. 2018-1-20], Dostupné z: <https://www.zive.cz/clanky/wi-fi-jak-si-zajistit-velke-pokryti-rychlost-a-silny-signal/anteny-a-jejich-nastaveni-deformace-a-ztrata-signalu-mereni/sc-3-a-172347-ch-90933/><https://www.zive.cz/clanky/wi-fi-jak-si-zajistit-velke-pokryti-rychlost-a-silny-signal/anteny-a-jejich-nastaveni-deformace-a-ztrata-signalu-mereni/sc-3-a-172347-ch-90933/>

7. Seznam obrázků

| | | |
|------------|---|----|
| Obrázek 1 | Vyzařovací diagram všesměrové antény | 15 |
| Obrázek 2 | CD74HC4067 Analog / Digital MUX | 16 |
| Obrázek 3 | Možné propojení jednotlivých integrovaných obvodů na I2C sběrnici.... | 17 |
| Obrázek 4 | Mapu pokrytí Wi-Fi vytvořená pomocí programu Ekahau Heat Mapped.. | 18 |
| Obrázek 5 | Deska Arduino UNO Rev3 | 21 |
| Obrázek 6 | NodeMCU Development Board/kit v 0.9 | 25 |
| Obrázek 7 | Mikrokontroler BS2-IC..... | 26 |
| Obrázek 8 | Mikrokontroler BasicX-24..... | 28 |
| Obrázek 9 | Mikrokontroler STM 32..... | 29 |
| Obrázek 10 | RASPBERRY Pi 3 Model B | 31 |
| Obrázek 11 | PLC od firmy Hartmut Wendt Hard & Softwareentwicklung | 32 |
| Obrázek 12 | NodeMCU Dev Kit v1.0(V2) zapojení pinů | 42 |
| Obrázek 13 | NodeMCU Dev Kit v0.9 technická specifikace a design..... | 43 |
| Obrázek 14 | Programovací prostředí IDE | 44 |
| Obrázek 15 | Instalace knihovny ESP8266 do prostředí Arduino IDE | 45 |
| Obrázek 16 | Nastavení NodeMCU desky v programovacím prostředí IDE | 46 |

| | | |
|------------|---|----|
| Obrázek 17 | Arduino IDE s kódem | 46 |
| Obrázek 18 | Architektura operačního systému Android | 48 |
| Obrázek 19 | Vývojové prostředí Android Studio | 50 |
| Obrázek 20 | Schéma zapojení NodeMCU k relé a k výstupním zařízení | 53 |
| Obrázek 21 | Komunikace mobilních zařízení s deskou NodeMCU..... | 53 |
| Obrázek 22 | Realizace dvou zásuvek dálkově spínaných. | 54 |
| Obrázek 23 | Spínací modul s dvěma relé Songle SRD-05VDC-SL-C | 57 |
| Obrázek 24 | Schéma zapojení modulu s jedním relé..... | 57 |
| Obrázek 25 | Napájecí převodník AC-DC 220V na 12V a 5V..... | 59 |
| Obrázek 26 | Struktura programu pro NodeMCU | 60 |
| Obrázek 27 | Ovládací obrazovka na mobilním zařízení..... | 66 |
| Obrázek 28 | Struktura mobilní aplikace | 71 |

8. Seznam tabulek

| | | |
|-----------|--|----|
| Tabulka 1 | Porovnání různých bezdrátových standardů | 38 |
| Tabulka 2 | Parametry napájecího převodníku AC-DC 220V na 12V a 5V | 59 |

9. Seznam použitých zkratk

| | |
|---------|-----------------------------------|
| μ A | Mikro Ampér |
| μ s | Mikro sekunda |
| A/D | Analogově digitální převodník |
| ADC | Analog to Digital Converter |
| AP | Access Point |
| API | Application Programming Interface |
| Atd. | a tak dále |
| CAN | Controller Area Network |
| CLK | CLOCK |
| CMD | Command |
| CRC | Cyclic Redundancy Check |

D/A Digitálně Analogový převodník
 DIN Označení lišty dle ČSN EN 60715
 DIP Dual Inline Package
 DMA Direct Memory Access
 DSI Digital Standard Interface
 EEPROM Electrically Erasable Programmable Read-Only Memory
 GPIO General Purpose Input/Output
 GSM Global System for Mobile Communications
 I/O Input / Output
 ID Identification
 IDE *Integrated Development Environment*
 IEEE Institute of Electrical and Electronics Engineers
 IIC Internal-Integrated-Circuit
 In Inch
 Inc Incorporated
 IP Internet Protocol
 JTAG Joint Test Action Group
 KB kilobit
 LCD Liquid Crystal Display
 Led Light-Emitting Diode
 mA Mili Ampér
 Mcu Micro Controller Unit
 MHz *Megahertz*
 Mm milimetry
 Např. například
 NFC Near Field Communication
 O/I Output / Input
 OS Operační Systém
 PAN personal access network
 PDA Personal Digital Assistant
 Plc Programmable Logic Controller
 PWM pulse width modulation
 Ram Random Access Memory

Resp respektive
RFID Radio Frequency Identification
RISC Reduced Instruction Set Computer
RX Receive
SCL Session Control layer
SMS Short message service
SPI Service Provider Interface
SRAM Static Random Access Memory
STA společná televizní anténa
SWD Serial Wire Debug
TCP/IP Transmission Control Protocol/Internet Protocol
TTL Transistor-Transistor-Logic
TX Transmit
UART Universal Asynchronous Receiver Transmitter
url Uniform Resource Locator
Usb Universal Serial Bus
VDC Virtual Design and Construction
WiFi Wireless Fidelity
WPA / WPA2 Wi-Fi Protected Access

10. Přílohy

Příloha I. Zdrojový kód pro NodeMCU

```
#include <ESP8266WebServer.h>

#define pin_1 4
#define pin_2 5

const char *ssid = "BP Jan Cech";
const char *password = "heslo123";
const char *key = "klic123";

ESP8266WebServer server(80);

void setup() {
  Serial.begin(9600);

  pinMode(pin_1, OUTPUT);
  pinMode(pin_2, OUTPUT);

  //vypnutí relatek
  digitalWrite(pin_1,HIGH);
  digitalWrite(pin_2,HIGH);

  startWifi();
}

void loop() {
  server.handleClient();
}

void startWifi(void)
{
  WiFi.softAP(ssid, password);
  IPAddress ipAddress = WiFi.softAPIP();

  server.on("/", [=](void) {
    return handleRoot();
  });

  server.begin();
}

void handleRoot() {
  String request = server.arg("req");
  String requestKey = server.arg("key");
```

```
if(requestKey != key) {
  server.send(200, "text/text", "Invalid key"); // code, type, string
  return;
}

if(request == "1on") {
  turnOn(pin_1);
}
else if(request == "1off") {
  turnOff(pin_1);
}
else if(request == "2on") {
  turnOn(pin_2);
}
else if(request == "2off") {
  turnOff(pin_2);
}

server.send(200, "text/text", "Success"); // code, type, string
}

void turnOn(int pin) {
  digitalWrite(pin,LOW);
}

void turnOff(int pin) {
  digitalWrite(pin,HIGH);
}
```

Příloha II. Zdrojový kód aplikace pro OS Android

MainActivity.java

```
package cech.jan.remotehomecontrol;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.content.Context;
import android.content.SharedPreferences;
import android.widget.PopupMenu;
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

    private VolleyConnection connection;
    private EditText passwordField;
    private Button btnSubmit, btnCancel, btnSettings, button1, button2,
button3, button4;
    public static final String sharedPref = "apiKey";
    SharedPreferences sharedPreferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        sharedPreferences = getSharedPreferences(sharedPref,
Context.MODE_PRIVATE);
        connection = new VolleyConnection(this);
        listenerOnButtons();
    }

    public void listenerOnButtons() {
        passwordField = (EditText) findViewById(R.id.txtPassword);
        btnSubmit = (Button) findViewById(R.id.btnSubmit);
        btnCancel = (Button) findViewById(R.id.cancel);
        btnSettings = (Button) findViewById(R.id.settings);
        button1 = (Button) findViewById(R.id.button1);
        button2 = (Button) findViewById(R.id.button2);
        button3 = (Button) findViewById(R.id.button3);
        button4 = (Button) findViewById(R.id.button4);

        btnSettings.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {

                PopupMenu popup = new PopupMenu(MainActivity.this,
btnSettings);
```

```

        popup.getMenuInflater().inflate(R.menu.popup_menu,
popup.getMenu());

        popup.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
            public boolean onOptionsItemSelected(MenuItem item) {
                btnSettings.setVisibility(View.GONE);
                btnSubmit.setVisibility(View.VISIBLE);
                passwordField.setVisibility(View.VISIBLE);
                btnCancel.setVisibility(View.VISIBLE);

                return true;
            }
        });
        popup.show();
    });

    btnCancel.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            btnSettings.setVisibility(View.VISIBLE);
            btnSubmit.setVisibility(View.GONE);
            passwordField.setVisibility(View.GONE);
            btnCancel.setVisibility(View.GONE);
        }
    });

    btnSubmit.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {

            String pwd = passwordField.getText().toString();

            SharedPreferences.Editor editor =
sharedpreferences.edit();
            editor.putString(sharedPref, pwd);
            editor.commit();

            btnSettings.setVisibility(View.VISIBLE);
            btnSubmit.setVisibility(View.GONE);
            passwordField.setVisibility(View.GONE);
            btnCancel.setVisibility(View.GONE);

            Toast.makeText(MainActivity.this, "Api password changed",
Toast.LENGTH_LONG).show();
        }
    });

    button1.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            handleButton("1on");
        }
    });

    button2.setOnClickListener(new OnClickListener() {
        @Override

```

```

        public void onClick(View v) {
            handleButton("1off");
        }
    });

    button3.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            handleButton("2on");
        }
    });

    button4.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            handleButton("2off");
        }
    });
}

public void handleButton(String button) {
    String apiKey = sharedPreferences.getString(sharedPref, null);

    connection.sendRequest("req=" + button + "&key=" + apiKey);
}
}

```

Activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="cech.jan.remotehomecontrol.MainActivity">

    <Button
        android:id="@+id/button1"
        android:layout_width="168dp"
        android:layout_height="85dp"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="136dp"
        android:text="Button On 1"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button2"
        android:layout_width="168dp"
        android:layout_height="85dp"
        android:layout_marginTop="136dp"
        android:text="Button Off 1"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toEndOf="@+id/button1"
        app:layout_constraintTop_toTopOf="parent" />

```

```

<EditText
    android:id="@+id/txtPassword"
    android:layout_width="138dp"
    android:layout_height="40dp"
    android:layout_marginEnd="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="16dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    android:visibility="gone"
    app:layout_constraintEnd_toStartOf="@+id/cancel"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintStart_toEndOf="@+id/btnSubmit"
    app:layout_constraintTop_toTopOf="parent">

    <requestFocus />
</EditText>

<Button
    android:id="@+id/btnSubmit"
    android:layout_width="92dp"
    android:layout_height="38dp"
    android:layout_marginLeft="16dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:text="@string/btn_submit"
    android:visibility="gone"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/settings"
    android:layout_width="92dp"
    android:layout_height="41dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="16dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="16dp"
    android:text="Settings"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/cancel"
    android:layout_width="92dp"
    android:layout_height="38dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginTop="16dp"
    android:text="Cancel"
    android:visibility="gone"

```

```
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintRight_toRightOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView  
    android:id="@+id/respondText"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="4dp"  
    android:layout_marginEnd="8dp"  
    android:layout_marginStart="8dp"  
    android:text="Respond"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.47"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintStart_toStartOf="parent" />
```

```
<Button  
    android:id="@+id/button3"  
    android:layout_width="165dp"  
    android:layout_height="84dp"  
    android:layout_marginBottom="8dp"  
    android:layout_marginLeft="16dp"  
    android:layout_marginTop="8dp"  
    android:text="Button On 2"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/button1"  
    app:layout_constraintVertical_bias="0.15" />
```

```
<Button  
    android:id="@+id/button4"  
    android:layout_width="168dp"  
    android:layout_height="84dp"  
    android:layout_marginBottom="8dp"  
    android:layout_marginEnd="8dp"  
    android:layout_marginRight="16dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="8dp"  
    android:text="Button Off 2"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.578"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintStart_toEndOf="@+id/button3"  
    app:layout_constraintTop_toBottomOf="@+id/button2"  
    app:layout_constraintVertical_bias="0.15" />
```

```
</android.support.constraint.ConstraintLayout>
```


Volley Connection.java

```
package cech.jan.remotehomecontrol;

import android.app.Activity;
import android.widget.TextView;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

public class VolleyConnection {
    final private String ipAddress = "192.168.4.1";
    final private String portNumber = "80";

    private RequestQueue queue;
    private TextView respondText;
    final private String url = "http://" + ipAddress + ":" + portNumber + "/?";

    VolleyConnection(Activity activity) {
        this.respondText = (TextView)
activity.findViewById(R.id.respondText);
        queue = Volley.newRequestQueue(activity);
    }

    void sendRequest(String endOfUrl) {
        StringRequest request = new StringRequest(Request.Method.GET,
url+endOfUrl,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    respondText.setText("success");
                }
            }, new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    respondText.setText("Chyba požadavku: " + error);
                }
            }
        ));
        queue.add(request);
    }
}
```