VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016 Dávid Vaľuš

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

SENZORICKÉ ZAŘÍZENÍ S WINDOWS 10 IOT CORE
SENSING DEVICE BASED ON WINDOWS 10 IOT CORE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Dávid Vaľuš
AUTHOR

VEDOUCÍ PRÁCE — Ing. Jiří Hošek, Ph.D.
SUPERVISOR

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Dávid Vaľuš ***ID:*** 164430
Ročník: 3 ***Akademický rok:*** 2015/16

NÁZEV TÉMATU:

Senzorické zařízení s Windows 10 IoT Core

POKYNY PRO VYPRACOVÁNÍ:

Platforma Windows 10 IoT Core představuje alternativu pro již zavedené Linuxové distribuce používané pro rychle se rozvíjející komunikaci v rámci Internetu věcí (Internet of Things). Cílem bakalářské práce proto bude vývoj komunikační jednotky zajišťující příjem dat od senzorů využívaných v IoT (senzory pohybu, teploty, vlhkosti, atd.). Řešení bude realizováno s využitím nízko-výkonového (embedded) zařízení Rasbperry Pi 2. Ukládání přijatých dat a jejich následné zobrazování bude provedeno s využitím dostupných nástrojů pro platformu Windows 10 IoT Core.

DOPORUČENÁ LITERATURA:

[1] HOWS, David a Peter MEMBREY. 2015. Learn Raspberry Pi 2 with Linux and Windows 10. 2nd Edition. New York: Apress. ISBN 978-1484211632.

[2] SCOTT, Robert. 2015. Raspberry Pi 2: Raspberry Pi 2 User Guide for Operating system, Programming, Projects and More!. 2nd edition. New York: CreateSpace Independent Publishing Platform. ISBN 978-1514369913.

Termín zadání: 1.2.2016 ***Termín odevzdání:*** 1.6.2016

Vedoucí práce: Ing. Jiří Hošek, Ph.D.
Konzultant bakalářské práce:

doc. Ing. Jiří Mišurec, CSc., *předseda oborové rady*

UPOZORNĚNÍ:
Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Táto práca sa zaoberá inštaláciou systému Windows 10 IoT Core na mikropočítač Raspberry Pi 2 a vytvorením AllJoyn aplikácie na meranie svetelnosti, teploty a vlhkosti vzduchu cez bluetooth rozhranie. V teoretickej časti sú popísané technológie ako M2M (Machine to Machine), IoT (Internet of Things), ich aktuálne nasadenie, vývoj do budúcnosti a platforma AllJoyn. Ďalej je v teoretickej častí porovnanie operačných systémov dostupných pre Raspberry Pi 2. V praktickej časti som sa venoval samotnej inštalácií systému Windows 10 IoT Core na dosku Raspberry Pi 2. Praktická časť venuje aj overeniu funkčnosti systému pomocou aplikácie na meranie teploty prostredníctvom SPI rozhrania a bluetooth. V praktickej časti sme sa pripojili k sieti AllJoyn a prostredníctvom nej zdieľali dáta z bluetooth senzoru.

KĽÚČOVÉ SLOVÁ

IoT, M2M, ARM, CPU, GPIO, Widnows 10 IoT, Raspberry Pi 2, AllJoyn, BLE, GATT

ABSTRACT

This bachelor thesis focuses on installation of system Windows 10 IoT Core on microcomputer Raspberry Pi 2 and creating AllJoyn application for measuring temperature, humidity and brightness via bluetooth interface. In theoretics part technologies like M2M (Machine To Machine) and IoT (Internet of Things) are described, actuall trends and the future and AllJoyn platform. There is also a comparsion of operating systems available for Raspberry Pi 2. Practical part focuses on the installation of system Windows 10 IoT Core on Raspberry Pi 2 board. It also deals with verification of the functionality of the system via testing aplication for temperature measurement via SPI interface and bluetooth. In practical part we joined the AllJoyn network and via this network we shared data from a Bluetooth sensor.

KEYWORDS

IoT, M2M, ARM, CPU, GPIO, Widnows 10 IoT, Raspberry Pi 2, AllJoyn, BLE, GATT

VAĽUŠ, Dávid *Senzorické zařízení s Windows 10 IoT Core*: bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 51 s. Vedúci práce bol Ing. Jiří Hošek, Ph.D.

PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Senzorické zařízení s Windows 10 IoT Core“ vypracoval(a) samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorském, o právoch súvisejúcich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhé, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

POĎAKOVANIE

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Jiřímu Hošekovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

POĎAKOVANIE

Výzkum popsaný v tejto bakalárskej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno ....

podpis autora(-ky)

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

OBSAH

Úvod **11**

1 M2M (Machine to machine) **12**
1.1 Vízia budúcnosti 13
1.2 Využitie M2M 13
1.3 Bezpečnosť 14
1.4 Porovnanie M2M a H2H(Human to Human) komunikácie 15

2 Windows 10 IoT **16**
2.1 Platformy Windows 10 IoT 16
2.2 Windows 10 IoT pre vstavané zariadenia 16
2.3 Technické riešenia IoT systému 17

3 Raspberry Pi 2 **18**
3.1 Popis 18
3.2 Operačné systémy pre Raspberry Pi 2 18
3.3 Zhodnotenie 22

4 Inštalácia systému Windows 10 IoT na Raspberry Pi 2 **23**
4.1 Komponenty potrebné k inštalácií Windows 10 IoT Core na Raspberry Pi 2 23
4.2 Nastavenie zariadenia 23
4.3 Inštalácia nástrojov Windows 10 IoT Core 24
4.4 Pripojenie Raspberry Pi 2 24
4.5 Užívateľské prostredie Windows 10 IoT Core 24

5 Meranie teploty pomocou MCP3208 **27**
5.1 Súpis použitých súčiastok 27
5.2 Zapojenie 27
5.3 Vytváranie projektu vo Visual Studio 2015 28
5.4 Aplikácia 29

6 AllJoyn **32**
6.1 AllSeen 32
6.2 AllJoyn 32
6.3 Microsoft a AllJoyn 33

7 Meranie teploty pomocou SensorTag CC2650 a vytvorenie AllJoyn aplikácie **34**
7.1 TI SensorTag v2 CC2650STK 34
7.2 Definovanie komunikácie 36
7.3 Vytváranie vlastnej knižnice AllJoyn rozhraní prostredníctvom AllJoyn CodeGen Tool 37
7.4 AllJoyn Aplikácia 38
7.5 Spúšťanie aplikácie 42
7.6 Zobrazovanie hodnôt 43

8 Záver **45**

Literatúra **46**

Zoznam symbolov, veličín a skratiek **48**

Zoznam príloh **50**

A Obsah priloženého CD **51**

ZOZNAM OBRÁZKOV

2.1 Topológia IoT systému 17
3.1 Doska Raspberry Pi 2 18
4.1 Predvolená aplikácia Windows 10 IoT Core 25
4.2 Webové rozhranie Windows 10 IoT Core 26
5.1 Zapojenie obvodu pre meranie teploty 27
5.2 GPIO Raspberry Pi 2 28
5.3 Vytváranie projektu 29
5.4 Nastavenie referencií 30
5.5 Prepnutie na vzdialené zariadenie 30
5.6 Zobrazovanie aktuálnej teploty, dátumu a času 31
7.1 Bloková schéma AllJoyn siete 34
7.2 TI SensorTag v2 CC2650STK 35
7.3 Profily, služby a charakteristiky v GATT 36
7.4 Pridávanie XML dokumentácie 38
7.5 Vygenerované súbory 39
7.6 Menné priestory 39
7.7 Menu Windows IoT Core Bluetooth Pairing Tool 42
7.8 Párovanie zariadení cez internetový prehliadač 43
7.9 Zoznam zariadení v AllJoyn sieti 44
7.10 Zobrazovanie hodnôt prostredníctvom AllJoyn 44

ZOZNAM TABULIEK

1.1 Porovnanie M2M a H2H komunikácie [4] 15

ÚVOD

Technológie M2M(Machine to Machine) a IoT(Internet of Things) zažívajú v posledných rokoch rozmach a v budúcnosti bude kvôli širokým možnostiam uplatnenia ešte rýchlejší. Využitie nachádzajú okrem iného v zdravotníctve, priemysle, doprave a obmedzené sú tvorivosťou samotných vývojárov.

AllJoyn je projekt s otvoreným zdrojovým kódom ktorý je vedený AllSeen alianciou pod hlavičkou Linux Foundation. Vďaka platforme AllJoyn spolu budú tieto zariadenia môcť komunikovať a vymieňať si informácie zariadenia rôznych výrobcov, komunikujúce rôznymi jazykmi. V kapitole 1.4 je porovnaná technológia M2M s H2H (Human to Human). Sú to rozličné technológie ktoré majú iné požiadavky na šírku pásma, oneskorenie, zabezpečenie, hardvér a iné. 1.1. Kapitola 6 sa venuje samotnému systému AllJoyn, AllSeen aliancii a prínosu spoločnosti Microsoft pre túto alianciu.

V bakalárskej práci je použitý systém Windows 10 IoT Core. Kapitola 2 sa venuje systému Windows 10 pre vložené zariadenia a popisuje jeho distribúcie, možnosti a minimálne hardvérové požiadavky. Systém Windows 10 IoT je ďalej podrobnejšie popísaný v kapitole 3.2 a porovnaný s podporovanými Linuxovými distribúciami systému pre vývojovú platformu Raspberry Pi 2. Ďalej sa v kapitole 3 sa špecifikujú parametre a rozmery Raspberry Pi 2.

Praktická časť sa venuje inštalácií systému Windows 10 IoT Core, nastaveniu počítača pre konfiguráciu Raspberry Pi 2 a príprave samotného zariadenia. Postup je podrobne popísaný v kapitole 4. V tejto kapitole je popísané tiež užívateľské prostredie Windows 10 IoT Core, možnosti vývoja aplikácií, predvolená aplikácia a pripájanie sa na zariadenie prostredníctvom siete.

Kapitola 5 sa venuje aplikácií na meranie teploty cez SPI rozhranie, pomocou ktorej sme overili funkčnosť systému a jeho plnú podporu Raspberry Pi 2. V kapitole 5 je aj popis použitých súčiastok, zapojenie obvodu a popis vytvárania projektu vo vývojom prostredí Microsoft Visual Studio. Meranie teploty sme prevádzali pomocou teplotného senzoru LM35 a ADP(Analógovo-digitálny prevodník) MCP3208.

Aplikácií na meranie teploty pomocou Bluetooth senzora TI CC2650 7.1 sa venuje kapitola 7. V tejto kapitole je tiež popísaná technológia GATT a BLE. Ďalej sa kapitola 7 venuje vytvoreniu AllJoyn aplikácie a zdieľaniu dát v sieti AllJoyn. Aplikáciou sme overili schopnosť Raspberry Pi 2 komunikovať cez Bluetooth a funkčnosť AllJoyn systému v kombinácií s Windows 10 IoT Core a Raspberry Pi 2.

1 M2M (MACHINE TO MACHINE)

V dnešnej dobe rastie trend využívania zariadení ktoré sú samočinné a nepotrebujú ku svojej obsluhe človeka. Aby bol využitý potenciál týchto zariadení, musia byť pripojené do jednej siete (Internet). Takto začali vznikať nové smery v komunikácií vstavaných zariadení (embedded zariadení) ako je napríklad IoT (Internet of Things), IoO (Internet of Objects) a tiež M2M.

M2M je predchodcom IoT. Rozdiel je, že tradičné M2M riešenia sa spoliehajú na P2P(Point to point) komunikáciu, kde využívajú špeciálne harvérové moduly, kde IoT zariadenia komunikujú prostredníctvom štandardizovanej IP siete. Je ich možné pripojiť oveľa viac a využívajú cloudové úložiska a middleware. Z toho vyplýva výhoda pre IoT, že dáta sú prístupné všetkým, bez nutnosti pridávania ďalšej kabeláže a pod. [1]

M2M je podmnožina IoT a ponúka vzdialený prístup k dátam, tieto dáta nie sú integrované v ostatnom softvéri a to je v porovnaní s IoT menej efektívne. Spojenie zariadení, senzorov, senzorových dát, analýz a iných je základnou myšlienkou IoT. IoT sa v porovnaní s M2M spolieha viac na softvér ako harvér aby bol ľahšie dostupný a aby sa dal prispôsobiť pre rôzných zákazníkov. To z môjho pohľadu predstavuje výhodu, keďže potom je systém jednoduchšie modifikovateľný a univerzálnejší. Takže ak je potrebný jednoduchý vzdialený prístup do stroja a aplikácie, pre správu služieb postačuje M2M, ale ak potrebujeme spracúvať veľké množstvo dát, alebo robiť rozsiahle analýzy a využívať iné prírastkové hodnoty založené na softvérových riešeniach, lepšou variantou je IoT [1].

Počet zariadení pripojených k Internetu rastie a dá sa očakávať, že bude rásť exponenciálne. Celosvetovo je už registrovaných viac ako 3 miliardy aktívnych mobilných telefónov [1]. No ľudia používajú viac zariadení ako napríklad laptopy, tablety a pod. Milióny nových zariadení je možné prepojiť priamo jedno k druhému (M2M). Tieto zariadenia komunikujú a ponúkajú služby prostredníctvom internetu bez zásahu človeka. Vytvárajú nové možnosti a inovácie z technologického a spoločenského hľadiska. Dokážu sa jednoducho pripojiť pomocou rôznych bezdrôtových sietí v reálnom čase a na ľubovoľnom mieste. Tento nárast v spojení s pripojiteľnými zariadeniami M2M a IoT (Internet of Things) rapídne rozvinie mnoho sektorov v nasledujúcich 5-10 rokoch [1].

V roku 2007 sa po prvý krát v histórii stalo, že viac ako 50% svetovej populácie žilo v mestách(UN-HABITAT, 2011). Do roku 2050 to má byť 70% svetovej populácie. Takže infraštruktúra miest a štátov sa bude musieť prispôsobiť od ciest, pouličného osvetlenia, sietí verejnej dopravy a mnohých ďalších (HM Treasury 2011). Väčšina z menovaných bude vybavená senzormi a pohonmi pre viac efektívnu správu a všetky tieto zariadenia, spojené v danej infraštruktúre, budú spojené do veľkej

siete dátových analýz a manažovania systému [2]. Všetky tieto možnosti efektívne pomôžu k rozvoju chytrých miest.

1.1 Vízia budúcnosti

Náš svet prešiel veľkou zmenou ktorá ovplyvnila život každého z nás. Ovplyvnila každého človeka, mesto, firmu a základy vecí na ktorých stojí naša spoločnosť a ekonomika. Internet zmenil ako komunikujeme, pracujeme, zabávame sa a aké máme možnosti. Priniesol veľkú technickú evolúciu. Staré a nové spoločnosti, mestá, ľudia sa museli adaptovať [1].

IoT zastrešuje komunikáciu medzi zariadeniami (či už chytré meracie prístroje, senzory, akuátory) čiže komunikáciu typu M2M. Tieto dve technológie spolu budú formovať základy nového sveta [2].

Všetko čo je v záujme a pod kontrolou ľudí, firiem a organizácií bude pripojené a poskytovať služby cez Internet. Pripojené bude všetko, čo by mohlo z pripojenia na sieť profitovať.

Výsledkom bude inteligencia všade okolo nás, takže sa čoraz častejšie budeme stretávať s pojmami ako inteligentné domy, elektrické siete či automobily.

Podľa odhadov prekročí počet zariadení používajúcich M2M do roku 2020 dve miliardy kusov. Je to z dôvodu lepšieho pokrytia siete LTE(Long Term Evolution), dokonalejšími rádiovými technológiami, priaznivejším prostredím a rozmachom technológií. Využívať sa budú hlavne v zdravotníctve, doprave a spotrebnom priemysle. K rastúcemu trendu bude prispievať aj klesajúca cena a nízka energetická náročnosť [1].

M2M komunikácia zatiaľ predstavuje len malý podiel na celkovej dátovej prevádzke, približne 0,1%. Tento podiel však bude stúpať s aplikáciami podporujúcimi malé oneskorenie a tiež rozmachom pripojiteľnej spotrebnej elektroniky, inováciami v automobilovom priemysle, logistike, reklame a neposlednom rade inteligentných domácností.

1.2 Využitie M2M

Technológia M2M nachádza využitie aj v domácnostiach. Napríklad pre zmenu teploty doma stačí zobrať do ruky mobilný telefón, alebo notebook a na diaľku teplotu upraviť [1]. Na diaľku sa dajú ovládať tiež napríklad kuchynské spotrebiče, môžeme zapnúť práčku, zatvoriť garážovú bránu a podobne.

V automobilovom priemysle sa s technickými inováciami M2M počíta tiež. Pomocou mobilu môžme auto naštartovať, nastaviť teplotu v interiéry, zapnúť vyhrievanie sedadiel, nastaviť GPS(Global Positioning System) navigáciu ba dokonca nechať auto autonómne zaparkovať a vyparkovať. V blízkej budúcnosti sa stane povinnou výbavou všetkých nových áut systém tiesňovej komunikácie e-call [1]. Kde sa napríklad pri vážnejšej nehode a po vystrelení airbagov automatický vytočí tiesňová linka a poloha auta sa odošle na riadiace stredisko. V budúcnosti sa počíta, že auta budú medzi sebou komunikovať a posielať si informácie napríklad o kolónach, o zlom stave vozovky a pod.

Výhody má aj pre zdravotníctvo. V západných krajinách sú už bežnou súčasťou lekárske prístroje monitorujúce stav pacienta, ktoré posielajú dáta do centrálneho systému, ktorý je prepojený s alarmom. Lekár môže napríklad merať tlak pacienta zo svojho domu a sledovať jeho celkový stav na diaľku [1].

Veľké využitie môže dosiahnuť v priemysle. Výroba sa bude dať kontrolovať a riadiť na diaľku. M2M naviac dokáže sledovať aj prevádzku firemných áut, alebo prenáša dáta v bankových termináloch.

Technológiu M2M využíva aj polícia. Vozidlo nahráva 360° video okolo vozidla a dokáže rozoznávať evidenčné čísla a automaticky ich porovnávať s databázou. Systém bez zásahu človeka zistí či automobil nie je kradnutý a či má platnú technickú kontrolu. V prípade overenia totožnosti nie je potrebné aby sa policajt spájal s dispečingom a zdĺhavo lustroval pomocou vysielačky, no dokáže to urobiť priamo z vozidla. Technológia využíva aj GPS a v prípade hovoru na tiesňovú linku 158, dispečer nahlási incident do systému a ten ho priradí najbližšej voľnej hliadke.

Využitie sa nachádza aj elektronickom mýtnom systéme, ktoré napríklad v Česku zatiaľ využíva pevnú mobilnú sieť a mýtne brány a na Slovensku to je realizované pomocou satelitného systému.

1.3 Bezpečnosť

Rozvoj fenoménu M2M tiež prináša bezpečnostné otázky. Je potreba zaistiť dostatočnú úroveň bezpečnosti a súkromia. Ako pri každej technológií sú benefity spojené aj s rizikom, takže väčšie využívanie M2M prinesie aj väčšiu atraktivitu pre útočníka. Komunikácia medzi strojmi bez dohľadu človeka naviac prináša riziko, že bude napadnutá bez povšimnutia človeka. Bezpečnostné riziko je väčšie aj o to, že technológia je bezdrôtová [2]. Dôležitou výzvou je pre to pracovať na bezpečnosti M2M prostredí, aby mohlo rásť a poskytovať očakávané výsledky bez zbytočných rizík.

1.4 Porovnanie M2M a H2H(Human to Human) komunikácie

Komunikácia M2M je vždy medzi dvoma zariadeniami (počítač, chytrý telefón, senzor) bez zásahu človeka. Komunikácia H2H je vždy medzi človekom a človekom prostredníctvom nejakého zariadenia. Za H2H komunikáciu považujeme napríklad telefonovanie, posielanie správ a iné. Služby H2H môžeme rozdeliť na 4 QoS (Quality of Service) triedy: konverzačná, vysielanie, interaktívna a na pozadí [3]. Porovnanie M2M a H2H technológie je v tabuľke 1.1.

Tab. 1.1: Porovnanie M2M a H2H komunikácie [4]

	M2M	**H2H**
Počet zariadení	Počet zariadení sa rýchlo zvyšuje, čo bol jeden z dôvodov zavedenia IPv6 (Internet Protocol v6).	Počet zariadení narastá, ale je obmedzený na schopnosť ľudí komunikovať naraz.
Objem dát	Väčšina dátovej prevádzky je upload, ktorý využíva malú šírku pásma napr. upload dát z teplomeru. Avšak napr. kamerový systém môže vyžadovať väčšiu šírku pásma.	Väčšinu dátového toku tvorí download, ktorý využíva značnú šírku pásma napr. pri sťahovaní dát, Web, či pozerania videa.
Batérie	Zariadeniu musí batéria vydržať omnoho viac ako u H2H	Batériu nie je problém vymeniť za novú.
Oneskorenie	Niektoré aplikácie sú určené pre riadenie v reálnom čase. Prioritné aplikácie majú nízku toleranciu.	Oneskorenie sa toleruje a to aj v hlasovej komunikácií.
Zabezpečenie	Je potrebné zabezpečiť, prípadne aj utajiť zariadenie z dôvodu náročnosti pravidelnej kontroly zariadenia.	Je väčšinou známe, že zariadenie bolo ukradnuté.
Dimenzovanie	Z dôvodu veľkého počtu pripojení v určitý čas je nutné riadne nadimenzovať mobilnú sieť, aby nedochádzalo k preťaženiu.	U H2H sa jedná o normálne dimenzovanie, ktorenie nepotrebuje špeciálny prístup.

2 WINDOWS 10 IOT

V bakalárskej práci je použitý Windows 10 IoT Core, pre to sa na tento systém a jeho distribúcie viac zameriame. Windows 10 IoT, je zameraný na širokú škálu inteligentných zariadení, od malých priemyselných brán do väčších zložitejších zariadení, ako sú predajné terminály a bankomaty. V kombinácií s vývojovými nástrojmi Microsoft a službou Azure IoT môže zhromažďovať, ukladať a spracúvať dáta.

2.1 Platformy Windows 10 IoT

Platformy pre Windows 10 IoT sa delia do 3 kategórií [5]:

- **Windows 10 IoT Enterprise** – je určený pre priemyselné zariadenia. Plná verzia Windows. Podporuje UWP aj klasické Windows aplikácie, pokročilú ochranu a plné grafické rozhranie. Minimálne hardvérové požiadavky sú 1GB RAM, 16GB flash X86/X64.
- **Windows 10 Mobile Enterprise** – verzia Windowsu pre mobilné priemyselné zariadenia s batériou. Implementovaná je podpora pre čítanie čiarových kódov. Umožňuje nastaviť viacerých užívateľov a ich práva. Mininmálne harvérové požiadavky sú 512MB RAM, 4GB flash a ARM/x86 architektúra procesora.
- **Windows 10 IoT Core** – alebo Windows embedded Compact. Navrhnutý pre drobné počítače, predajné automaty a napríklad aj chytré hodinky. Optimalizovaný pre drobné zariadenia. Podporuje UWP pre zjednodušenie vývoja aplikácií. Architektúra ARM/x86. Nemá grafické rozhranie.

2.2 Windows 10 IoT pre vstavané zariadenia

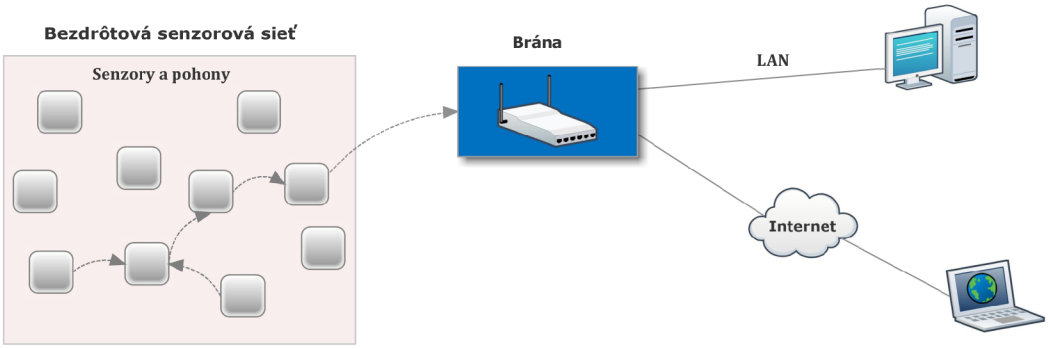
Windows 10 IoT Core sa používa na vstavaných zariadeniach, je to real time operačný systém. Má rovnakú funkcionalitu ako štandardný Windows, ale používa len tie moduly ktoré aplikácia potrebuje pre projekt [5]. Vstavané zariadenie je malé zariadenie ktoré dokáže prijímať a posielať informácie po sieti a jeho základ tvorí mikro počítač s nízkou spotrebou a výkonom, čo je označované ako embedded device. Windows 10 je možné spojiť s neskutočným množstvom vstavaných zariadení. Tieto zariadenia s Windows 10 IoT sú označované aj ako plytké vložené systémy. Postačuje im 256MB RAM ak zariadenie nemá displej a aspoň 512MB ak ho má v závislosti od rozlíšenia. Požiadavky na procesor sú aspoň 400MHz frekvencia, x86, x64 alebo ARM architektúra. Minimálne požiadavky na pamäťové úložisko sú 2GB [6].

Vstavané zariadenia sú napríklad spotrebná elektronika, ako nositeľná elektronika, roboti, pripojiteľné hračky, chladničky. Patria sem aj autonómne vozidlá, inteligentné domy, cesty, zdravotnícke pomôcky, priemyselné roboty a mnoho iných.

2.3 Technické riešenia IoT systému

Vďaka možnosti spustenia Windows 10 IoT o.i. na Rasperry Pi 2 sa nám otvárajú nové možnosti vytvárania vlastných vstavaných zariadení, limitovaných len našou fantáziou.

Požiadavky na procesor pre vstavané zariadenie sú malé rozmery a najmä nízka spotreba energie. Hlavnými uchádzačmi pre mikro procesor vstavaných zariadení sú Intel s jeho Intel Atom(výkonnejší, napríklad pre bránu) a Intel Quark(pre vložené systémy) a ARM(Acorn RISC Machine) s veľkým množstvom mikro procesorov ako napríklad Cortex M4 a Cortex-A pre brány, Cortex M0 pre vstavané systémy.

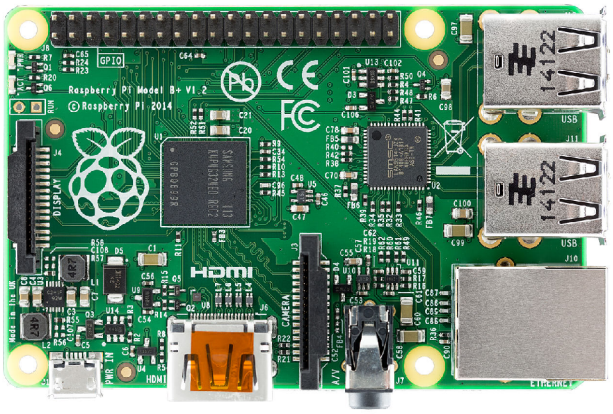


Obr. 2.1: Topológia IoT systému

V automatizovanej výrobe, má každý robot mikro počítač ktorý je pripojený ku sieti. Tieto počítače vzájomne komunikujú a zdieľajú informácie. Obsluha môže v sieti LAN(Local Area Network) riadiť výrobný proces. Do siete sa je možné pripojiť aj prostredníctvom siete Internet, ale napríklad s obmedzenými právomocami kvôli bezpečnosti. Obr. 2.1

3 RASPBERRY PI 2

V tejto práci je použité Raspberry Pi 2. Je to jednodoskový počítač s doskou veľkosti zhruba platobnej karty, prípadne ide o niečo menšiu doštičku (výpočtový modul). Vyvíja ho britská nadácia Raspberry Pi Foundantion s cieľom podporiť výučbu informatiky na školách.



Obr. 3.1: Doska Raspberry Pi 2

3.1 Popis

Základom Raspberry Pi 2 je verzia SoC BCM2836 firmy Broadcom, ktorý obsahuje štyri spätne kompatibilné CPU (Computer Processor Unit) s taktom 900 MHz, jednotkou SIMD a 1 GB pamäte RAM [7].

Raspberry Pi 2 obsahujú grafický procesor VideoCore IV kompatibilný s OpenGL ES 2.0. Naopak neobsahujú žiadne rozhranie pre pevný disk alebo SSD - pre zavedenie systému a trvalé uchovanie dát je určený slot na SD kartu [7].

K Raspberry Pi 2 je možné pripojiť externý monitor prostredníctvom HDMI rozhrania a takisto pripojenie k sieti pomocou Ethernet portu. Pripojenie periférií zabezpečujú 4 USB porty a Micro-USB port pre napájanie.

3.2 Operačné systémy pre Raspberry Pi 2

Raspberry Pi 2 podporuje mnoho OS (operačných systémov). Väčšina z nich sú na báze Linuxu a jeden na základoch Windowsu. Operačné systémy na jadre Linuxu sú napríklad Raspbian OS, Arch Linux ARM, OSMC, OpenELEC, Snappy Ubuntu Core, Ubuntu MATE, Debian Jessie a Android. Jediný Windows operačný systém je Windows 10 IoT Core, ktorý je novinka medzi vyššie spomínanými [8], [9].

Niektoré z týchto operačných systémov je možné nainštalovať aj na iné vložené zariadenia ako napr. Intel Galileo, MinnowBoard Max a DragonBoard410c.

- **Raspbian OS**
  Je systém jednoduchý pre používanie, najznámejší a má veľkú podporu po celom svete. Je to najlepšia platforma na učenie začínajúcich programátorov na prácu s Raspberry Pi. Raspbian OS pravidelne dostáva najnovšie dostupné aktualizácie. Raspbian OS tiež zahŕňa matematický koprocesor, ktorý chýba na špecifikáciách ostatných operačných systémov, ako aj Debianu, ktorý je považovaný za otca Raspbianu.
  Medzi funkcie Raspbian OS patria aj grafické balíky ako predinštalovaný softvér a ak potrebujeme viac pre Raspberry, tak balíčky ako napríklad internetový prehliadač, posielanie správ, alebo kancelárske balíčky sa dajú jednoducho pridať. Raspbian využíva GUI (grafické rozhranie) LXDE, ktoré je možné nahradiť iným balíčkom podľa vlastného výberu. Raspbian OS má podporu veľkej komunity čo zaručuje informácie a podporu na veľmi vysokej úrovni [8], [9].
- **Arch Linux ARM**
  Je jeden z najjednoduchších systémov dostupných pre Raspberry Pi 2. Tento operačný systém je na základoch Arch Linux. Jeho jednoduchosť ma za cieľ, aby mal užívateľ plnú kontrolu nad systémom a z toho dôvodu prichádza bez grafického prostredia. Táto jednoduchosť niekedy prináša benefity, ale niekedy aj problémy. Pre užívateľov ktorý nikdy nepracovali s príkazovým riadkom je tu možnosť nainštalovať grafické prostredie až po tom čo sa naučia o právach administrátora a konfigurácii. Táto verzia je odporúčaná pre užívateľov ktorý už majú skúsenosti s Linuxom. Má tiež podporu veľkej komunity a na internete je dostupnej veľa dokumentácie s vysvetleniami a dôležitými informáciami [8].
- **OSMC**
  Celým názvom Open Source Media Center. Systém OSMC je nástupcom Raspbmc a je navrhnutý tak, aby podporoval rovnaké funkcie ako Raspbmc. Raspberry je skvelá platforma pre multimediálne aplikácie a na to je potrebný operačný systém ako OSMC. OS umožňuje sledovať televíziu pomocou tuneru, prehrávanie hudby, alebo prehrávanie vide pomocou PlayOn Media Server z prémiových služieb ako je Hulu, Netflix a Amazon. Okrem toho môže prehrávať video súbory z USB kľúčov, alebo prechádzať knižnicu obrázkov.
  Nainštalovať tento operačný systém je jednoduché. Je ho ľahké stiahnuť a prispôsobiť s preferovanými doplnkami. Inštaluje sa pomocou inštalátora, ktorý beží na Windowse, Linuxe aj Mac OS. OS je možné prispôsobiť stovkami užívateľmi vytvorenými doplnkami pre zvýšenie schopnosti Raspberry Pi prehrávať hudbu, video či zobrazovať fotky [8], [9].

- **Open ELEC**
  Možnosťou ako je z Raspberry Pi 2 možné spraviť multimediálne centrum je aj systém Open ELEC. Je dobrý na prehrávanie videa, pozeranie televízie a prehrávanie hudby. Open ELEC sa líši od ostatných operačných systémov ktoré bežia na Raspberry Pi 2. Tento operačný systém bol vybudovaný pre spustenie aplikácie s otvoreným zdrojovým kódom na zábavné mediálne centrum Kodi (predtým známy ako XBMC) [8].
- **Snappy Ubuntu Core**
  Raspberry Pi 2 s 1GB operačnej pamäte plne podporuje Ubuntu OS. Pre Raspberry Pi 2 je ale prispôsobená verzia Snappy Ubuntu Core. Na rozvoji Snappy Ubuntu Core sa podieľajú Ubuntu community, The Canonical Ltd. a Raspberry Pi Foundation. Hlavnou myšlienkou je vybudovať silný operačný systém pre pripojiteľné zariadenia a ľahko použiteľnú platformu pre robotov a automatizáciu. Ubuntu sľubuje, že bude možné systém si absolútne prispôsobiť(plný prístup k jadru) a dá užívateľom celkový výkon. Operačný systém je aj napriek tomu schopný prijímať aktualizáce a to najmä kvôli bezpečnosti.
  Pre všetkých užívateľov je Snappy skvelý spôsob ako sa naučiť o robotoch, chytrých zariadeniach a celkovo internete vecí. Môžu sa vyvíjať, simulovať a spúšťať aplikácie a s minimálnymi prostriedkami. Naviac je Ubuntu oceňovaná ako bezpečná platforma, ktorá umožňuje aj zabezpečené pripojenie v cloude.
  Cieľom Snappy je priviesť všetky znalosti z Ubuntu Touch (OS pre chytré telefóny) na dosky Raspberry Pi 2. Obsahuje AppArmor nástroj, ktorý udržuje aplikácie izolované jednu od druhej, čo má pomôcť užívateľom pri práci. Snappy prilákal veľa vývojárov zapojených do sofistikovaných projektov ako autonómne autá, robotika a iné chytré veci [8], [9].
- **Ubuntu MATE**
  Je založený na Ubuntu 15.04, ktorý ponúka zaujímavé vylepšenia v jadre v oblasti videa s vysokým rozlíšením. Grafické rozhranie sa nazýva MATE Desktop Enviroment. Základ je s aplikáciami ktoré pripomínajú Raspbian OS. Vydáva sa inou cestou ako Snappy Ubuntu Core. Zahŕňa dobrý výber aplikácií ako Shotwell prehliadač obrázkov a organizátor, Mozilla Thunderbird a Mozilla Firefox, Libre Office, audio prehrávač a tiež Transmission torrent downloader [8].
- **Debian Jessie**
  Verzia pre Raspberry Pi 2 vychádzajúca z operačného systérmu Debian 8.0. V závislosti na tejto verzii vývojari z Debian OS plánujú vydať dve nové verzie. Debian 9.0 s označením Stretch, ktorá sa už testuje a Debian 10 s označením Buster.
  Debian Jessie zahŕňa nástroje viacerých grafických prostredí pre moderný stolový počítač. Takže nechýba ani internetový prehliadač, kancelárske programy.

Jednoducho povedané Jessie zmení Raspberry Pi 2 na moderný stolový počítač. Dostupných, ako aj v ostatných Linuxových operačných systémoch je veľmi veľa balíčkov pre inštaláciu [8].

- **Android**
  Android nie je podporovaný Raspberry Pi Foundation. S daným výkonom sa Raspberry Pi 2 radí do kategórie porovnateľnej s chytrým telefónom strednej triedy. Kvôli tomuto faktu Raspberry Pi 2 vytvára záujem nainštalovať si tento operačný systém u mnohých užívateľoch systému od Googlu. Android predstavuje obmedzenie, pretože nepodporuje myš a klávesnicu. Pre to je najlepšia možnosť ako a získať výhody tohto operačného systému je použiť dotykovú obrazovku pripojenú k Pi 2. Android je dobre vyzerajúci systém s jednoduchými animáciami, veľkou ponukou tém a jednoduchých aplikácií [8].
- **Windows 10 IoT Core**
  Raspberry Pi platforma sa z používania len Linuxových distribúcií rozširuje na školy kde je vyžadovaný softvér kompatibilný s Windowsom a je zdarma. Systém Windows 10 IoT není navrhnutý ako klasická desktopová verzia pre spúšťanie programov ako MS Office, Photoshop, ale dokáže ovládať roboty, automatizované systémy a iné chytré zariadenia. Sytém umožňuje aj komunikáciu pomocou Bluetooth a WiFi. Inžinieri spoločnosti Microsoft chcú rozšíriť komunitu Windows užívateľov najmä v oblasti internetu vecí. Túto verziu systému Windows je možné spustiť aj na jednodoskových počítačoch MinnowBoard Max a Intel Galileo [8].
  Najzásadnejšiu zmenu, ktorú prináša Windows 10 z hľadiska vývoja aplikácií je, Uiversal Windows Platform. Zjednodušene povedané, jedna aplikácia pobeží na rôznom hardvéri a rôznom vydaní Windows 10 (PC, telefón, Xbox, HoloLens, Raspberry Pi 2, ...).
  Pre niektorý hardware, ako napríklad Raspberry Pi 2, bude treba aplikáciu preložiť pre odpovedajúcu procesorovú architektúru. V princípe pôjde stále o tú istú aplikáciu s rovnakým zdrojovým kódom.
  Takmer všetko čo poznáme z Windowsu (menu, ikony, znaky) zmizlo. Windows 10 IoT Core nemá užívateľské rozhranie. Spravuje sa cez webové rozhranie alebo Powershell, alebo na ňom beží aplikácia, ktorá užívateľské rozhranie môže mať.
  Windows 10 IoT Core je možné spustiť v dvoch modoch. Hlavičkový a bezhlavičkový. Rozdiel medzi týmito dvoma režimami je prítomnosť, alebo neprítomnosť akejkoľvek formy užívateľského rozhrania. V predvolenom nastavení je Windows 10 IoT Core v hlavičkovom režime a spúšťa aplikácie ktoré zobrazujú informácie meno počítača a IP adresu. Navyše v hlavičkovom režime štandardného UI (užívateľského rozhrania) UWP (Universal Windows Plat-

form) sú dostupné aj interaktívne aplikácie. Zariadenia ktoré nepotrebujú UI je možné nastaviť do bezhlavičkového režimu. UI je zakázané, aplikácie už nie sú interaktívne a tak je možné redukovať množstvo systémových prostriedkov. Pre vývoj aplikácií potrebujeme Windows 10 a Visual Studio 2015. Pokiaľ chceme vytvoriť aplikáciu, ktorá ma UI a nepotrebuje GPIO(General-purpose input/output), tak vytvoríme WUP aplikáciu, ktorú môžeme komplet odladiť na stolovom počítači.
Najväčší rozdiel medzi Windows 10 IoT Core a Linuxovými distribúciami pre Raspberry Pi 2 je chýbajúce grafické rozhranie. Čo je jednoduchšie a stabilnejšie pre dlhodobú prevádzku.
V budúcnosti pravdepodobne Windows 10 IoT Core nahradí Windows XP, ktorý je stále používaný vo veľkom počte bankomatov a platobných terminálov. Windows XP je už zastaraný a Microsoft sa nechce s nikým deliť o miesto na trhu.
Windows 10 IoT Core je možné nainštalovať aj na iné zariadenia ako Raspberry Pi 2. Microsoft podporuje inštaláciu aj na MinnowBoard Max a DragonBoard410c. Je možné ho nainštalovať aj na ďalšie zariadenia napríklad ako Intel Galileo, ale kvôli nízkemu výkonu nefunguje dobre a Microsoft skončil s podporou pre túto platformu [10].

3.3 Zhodnotenie

Vylepšený hardvér Raspberry Pi 2 priniesol veľa nových možností pre vývojárov a demonštruje ako veľký krok spravila táto platforma od 1. generácie. To môžeme vidieť aj na širokej škále podporovaných operačných systémov, ktoré ponúkajú veľa možností využitia.

Linuxové distribúcie systému pre Raspberry Pi 2 ponúkajú mnoho variant. Niektoré majú bohaté grafické rozhranie, niektoré zjednodušené a iné ho nemajú. Rozdielne sú aj možnosti ich využitia (robot, plne funkčný počítač, multimediálne centrum ...). Užívateľ si môže vybrať tú variantu ktorá mu najviac vyhovuje a vďaka veľkému množstvu balíčkov, ktoré su voľne na stiahnutie, prispôsobiť. Výhodou Linuxových distribúcií je podpora veľkej komunity a možnosť stiahnuť si už veľa vytvorených programov.

Hlavné využitie systému Windows vidím v robotike a vo vstavaných zariadeniach, kde zariadenia budú spĺňať svoju úlohu bez zásahu človeka a tým pádom nebudú potrebovať grafické rozhranie ktoré by zbytočne zaťažovalo hardvér. Veľkú výhodu vidím v Univerzálnej Windows Platforme, ktorá uľahčuje tvorbu a odladzovanie aplikácií a tiež v cloudovej službe Azure.

4 INŠTALÁCIA SYSTÉMU WINDOWS 10 IOT NA RASPBERRY PI 2

V tejto kapitole sa popisuje inštalácia systému Windows 10 IoT Core na Raspberry Pi 2. Návod je zo stránok spoločnosti Microsoft [11].

4.1 Komponenty potrebné k inštalácií Windows 10 IoT Core na Raspberry Pi 2

1. Počítač so systémom Windows 10
2. 5V Micro USB napájací zdroj
3. Micro SD kartu s kapacitou 8 alebo 16GB, odporúča sa class 10
4. HDMI kábel a monitor
5. Ethernet kábel
6. Čítačku pamäťových kariet SD

4.2 Nastavenie zariadenia

1. Na počítači sa odporúča mať OS Windows 10 (verziu 10.0.10240), alebo novší.
2. Inštalácia programu Visual Studio 2015.
3. Nastavenie Visual Studio 2015
   - Odporúča sa Visual Studio Community Edition, ale použiť môžeme aj vyššie verzie ako Visual Studio Professional 2015, alebo Visual Studio Enterprise 2015.
   - Pri inštalácii Visual Studio zvolíme Custom inštaláciu a vyberieme Universal Windows App Development Tools -> Tools and Windows SDK.
   - V prípade, že už máme Visual Studio nainštalovaný, je nutné si k nemu doinštalovať potrebné nástroje.
4. Inštalácia Windows IoT Core Project Templates. Tieto šablóny je možné stiahnuť cez internetový prehliadač, alebo priamo v programe Visual Studio (Tools -> Extensions and Updates -> Online).
5. Povolenie vývojárskeho módu. V nastaveniach systému Windows sa klikne na Aktualizácie a bezpečnosť, z lišty vľavo sa vyberie možnosť pre vývojárov a vyberie sa políčko vývojársky mód.

4.3 Inštalácia nástrojov Windows 10 IoT Core

1. Stiahne sa Windows 10 IoT vo formáte ISO.
2. Otvorí sa ISO súbor a nainštaluje sa `Windows_10_IoT_Core_RPi2.msi`.
3. Do čítačky kariet sa vloží micro SD karta.
4. Spustí sa program IoTCoreImageHelper.exe.
5. V programe sa zvolí daná SD karta a súbor flash.ffu. Cesta k súboru je: `C:\Program Files (x86)\Microsoft IoT\FFU\RaspberryPi2\flash.ffu`. Karta bude naformatovaná a nakopírujú sa na ňu súbory potrebné k inštalácii Windows 10 IoT Core.

4.4 Pripojenie Raspberry Pi 2

1. Vloží sa Micro SD karta ktorú sme si pripravili.
2. Pripojí sa sieťový kábel do siete LAN (Local Area Network) a uistíme sa, že počítač je pripojený v rovnakej sieti.
3. Pripojí sa HDMI monitor.
4. Pripojí sa napájací zdroj.
5. Windows 10 IoT Core sa zavedie automaticky po pripojení napájacieho zdroja.
6. Zvolí sa jazyk zariadenia a potvrdíme tlačidlom Next.
7. Po zavedení systému sa spustí predvolená aplikácia. Viď. Obr. 4.1

4.5 Užívateľské prostredie Windows 10 IoT Core

Užívateľské prostredie je veľmi jednoduché. Po zavedení systému sa spustí predvolená aplikácia s jednoduchým menu. Windows 10 IoT Core dokáže rozoznať pripojenú myš a pomocou nej prebieha aj obsluha systému. Obr. 4.1

V karte device info sa zobrazujú základné informácie o zariadení ako je meno a IP adresa. Zobrazuje sa aj obrázok Raspberry Pi 2, pripojené USB zariadenia a druh pripojenia k sieti. Nedá sa nič editovať ani inak meniť.

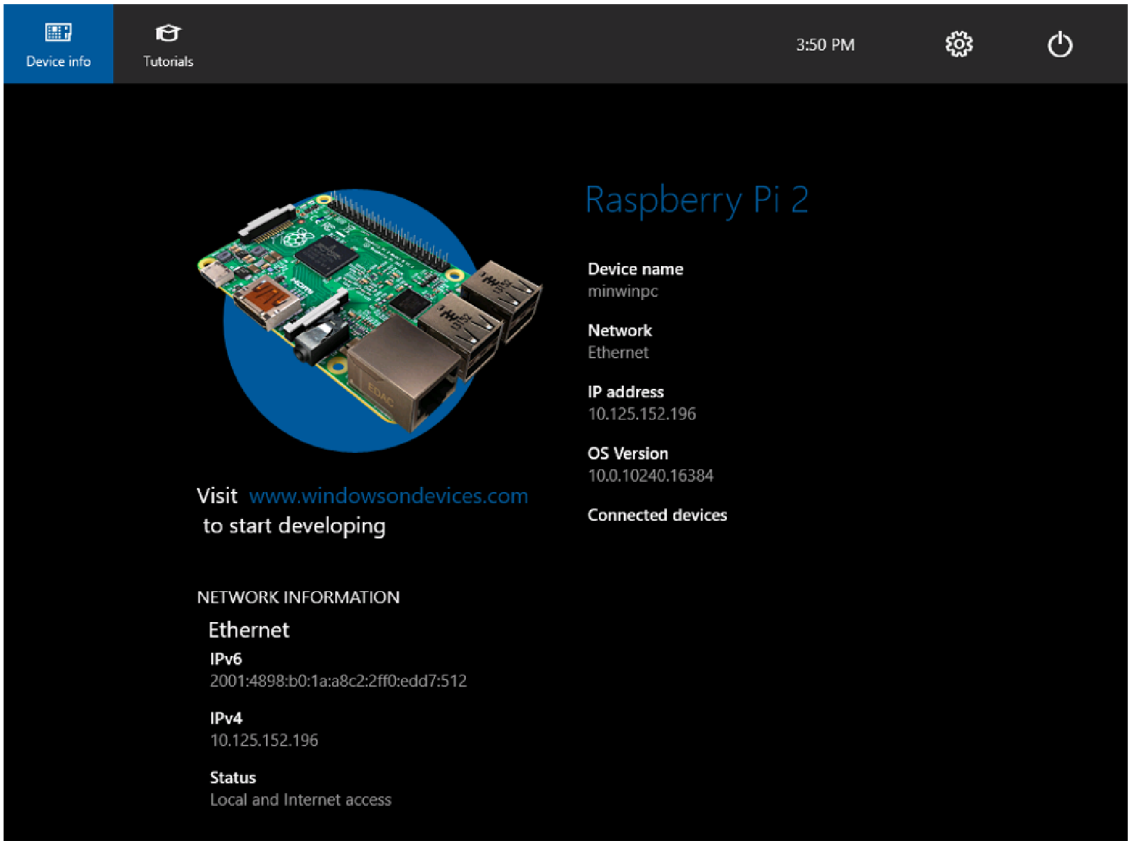
V karte tutorials sú jednoduché návody pre začiatočníkov s Rasppberry Pi 2 ako napríklad blikanie LED(Light Emitting Diode).

V hornej lište sa zobrazuje aj čas, ktorý nie je možné zmeniť cez predvolenú aplikáciu.

V karte nastavenia môžeme zmeniť jazyk, alebo v prípade problémov s pripojením do siete požiadať o novú IP adresu.

Karta s logom napájania ponúka vypnutie a reštart zariadenia.

Základné nastavenie ako meno zariadenia, heslo a časová zóna prebieha textovými príkazmi pomocou Windows PowerShell, alebo pomocou SSH. Prostredníctvom Windows PowerShell, alebo pomocou SSH je tiež možné vzdialene ovládať zariadenie.

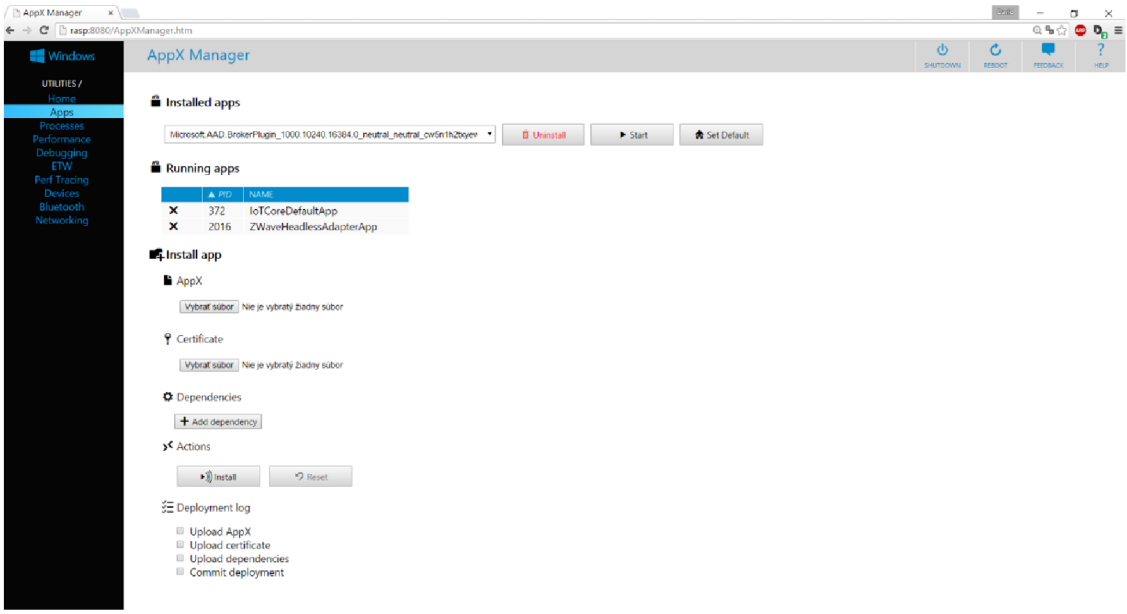


Obr. 4.1: Predvolená aplikácia Windows 10 IoT Core

Windws 10 IoT Core podporuje aj vlastný webový server. Pripájame sa k nemu cez prehliadač zadaním `http://<menozariadenia>:8080`. Po pripojení sa je nutné sa autentifikovať. Cez webové rozhranie môžeme napríklad nahrávať naše aplikácie, odladzovať ich a sledovať stav zariadenia a procesov. V ľavej časti obrazovky je menu s položkami: Viď. Obr. 4.2

- Home zobrazuje základné informácie o zariadení.
- Apps poskytuje inštaláciu, mazanie a iné akcie s aplikáciami.
- Processes umožňuje nám vidieť a spravovať procesy ktoré práve bežia a v akej miere využívajú pamäť a procesor.
- Performance zobrazuje grafy v reálnom čase a to využívanie procesora a dáta prenášané cez I/O(Input/Output). Zobrazuje aj celkovú, používanú, voľnú a vyhradenú pamäť.
- Debugging pomáha identifikovať problémy s aplikáciou.

- ETW(Event Tracing for Windows) sledovanie udalostí.
- Performance tracing umožňuje nahrávanie stopy v rôznych nastaveniach systému.
- Device zoznam práve pripojených zariadení.
- Networking zobrazuje IP adresu a popis pripojenia.



Obr. 4.2: Webové rozhranie Windows 10 IoT Core

Výhodou Windows 10 IoT Core je, že v porovnaní s Raspbian podporuje UWP (Universal Windows Platform), to znamená, že jednu aplikáciu môžeme spustiť na všetkých Windows 10 zariadeniach(počítač, chytrý telefón, tablet a vložené zariadenia ako Raspberry Pi 2). Výhodou je aj, že kód môžeme písať a odladzovať na počítači a spúšťať na Raspberry Pi 2. Windows 10 IoT Core má dobre spravenú a zabezpečenú cloudovú službu Azure s ktorou Raspberry Pi 2 spolupracuje a vďaka tomu sa môžeme pripájať a prezerať dáta. Základným rozdielom medzi Raspbian OS a Windows 10 IoT je, že Windows má len jednoduché grafické rozhranie. Výhodou Windows je jednoduchosť, väčšia spoľahlivosť, rýchlejšia odozva systému a menšie hardvérové nároky . Nevýhodou je, že k Raspberry musíme byť pripojený prostredníctvom siete a tiež, že aj jednoduché nastavenia ako meno zariadenia a čas musíme nastavovať pomocou príkazového riadku.

5 MERANIE TEPLOTY POMOCOU MCP3208

Cieľom bakalárskej práce bolo okrem iného otestovať systém Windows 10 IoT Core na doske Raspberry Pi 2. Zvolil som aplikáciu na meranie teploty, aby som vyskúšal zobrazovanie na monitore a komunikáciu Raspberry Pi 2 cez GPIO porty.

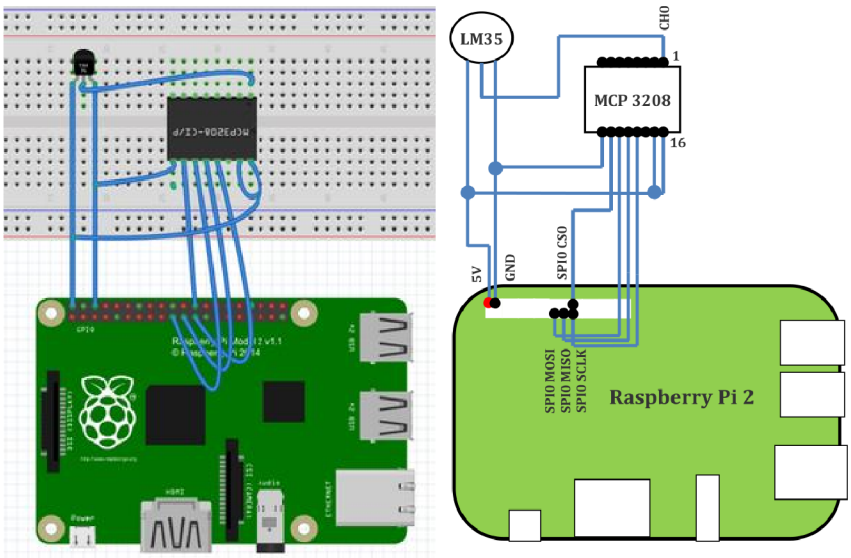
Meranie teploty je prevádzané pomocou SPI(Serial Peripheral Interface). Je to sériové periférne rozhranie, ktoré sa požíva pri komunikácií medzi mikroprocesormi a ostatnými integrovanými obvodmi. Teplotný senzor je pripojený do ADP(Analógovo-Digitálny prevodník) a ten je pripojený do Raspberry Pi 2 cez SPI rozhranie. Teplotný senzor je schopný rozoznávať teploty od 0 do 100°C s presnosťou 0,5°C.

5.1 Súpis použitých súčiastok

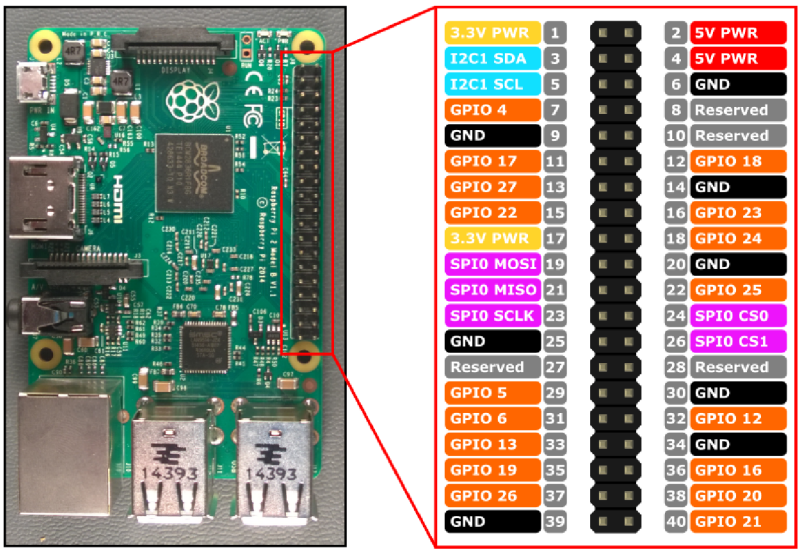
- Analógovo-digitálny prevodník MCP3208 CI/P.
- Teplotný senzor LM35. Rozsah 0-100°C.
- Nepájivé kontaktné pole
- HDMI Monitor

5.2 Zapojenie

- Obvod sa zapojí podľa obrázku 5.1. Raspberry Pi 2 má 40 GPIO(General-purpose input/output) pinov s rôznou funkciou. Viď obr. 5.2.
- K Raspberry Pi 2 sa pripojí HDMI monitor, sieťový kábel a napájanie.



Obr. 5.1: Zapojenie obvodu pre meranie teploty

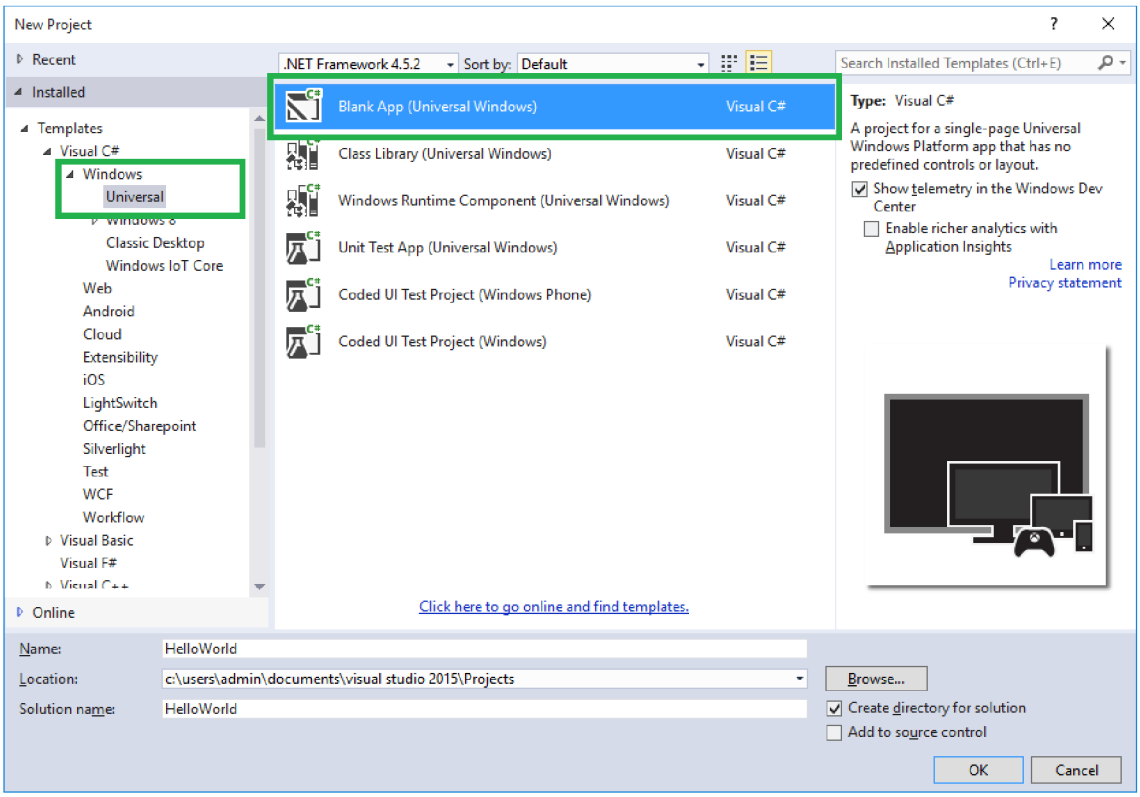


Obr. 5.2: GPIO Raspberry Pi 2

5.3 Vytváranie projektu vo Visual Studio 2015

1. Vo Visual Studio 2015 sa zvolí File/ New Project. Otvorí sa nám okno New Project a vyberieme Blank App(Universal Windows). Viď obr. 5.3
2. V menu projektu sa klikne pravým na References a zvolí sa Add Reference. Viď. obr. 5.4 Otvorí sa okno Reference Manager, v menu vpravo sa rozklikne Universal Windows -> Extensions a vyberie sa Windows IoT Extensions for the UWP.
3. Vo Visual studiu sa prepne z x86 na ARM a z Local Machine sa prepne na Remote Machine. Viď. obr. 5.5
4. Po kliknutí na Remote machine sa objaví okno Remote Connections kde zadáme meno, alebo IP adresu Raspberry Pi 2 a Authentication Mode sa zvolí None.

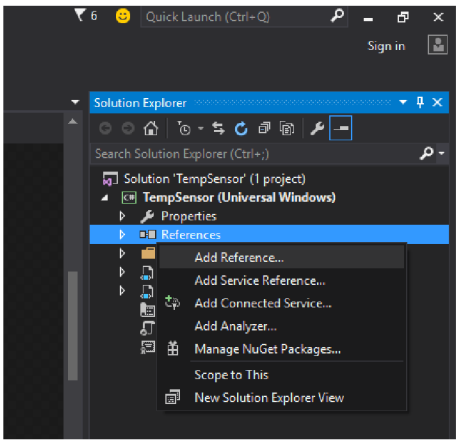
Týmto sa nastavilo, že naša aplikácia sa bude po Debug spúšťať priamo na Raspberry Pi 2.



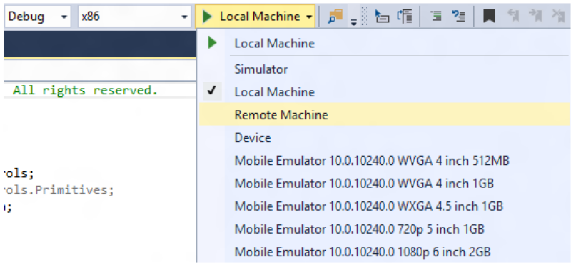
Obr. 5.3: Vytváranie projektu

5.4 Aplikácia

Na stránkach GitHub `https://github.com/ms-iot/samples/` sú dostupné vzorové aplikácie od spoločnosti Microsoft pre začínajúch vývojárov Windows 10 IoT Core. Zvolili sme si aplikáciu TempSensor/CS ktorú sme preložili do slovenčiny a upravili, aby zobrazovala teplotu v °C. Prispôsobili sme si jednoduché grafické rozhranie, ktoré zobrazuje aktuálnu teplotu, dátum a čas. Po kliknutí na debug sa aplikácia automaticky spustí na Raspberry Pi 2. Viď. obr. 7.2 Teplota sa zobrazuje na monitore pripojenom k Raspberry Pi 2.



Obr. 5.4: Nastavenie referencií

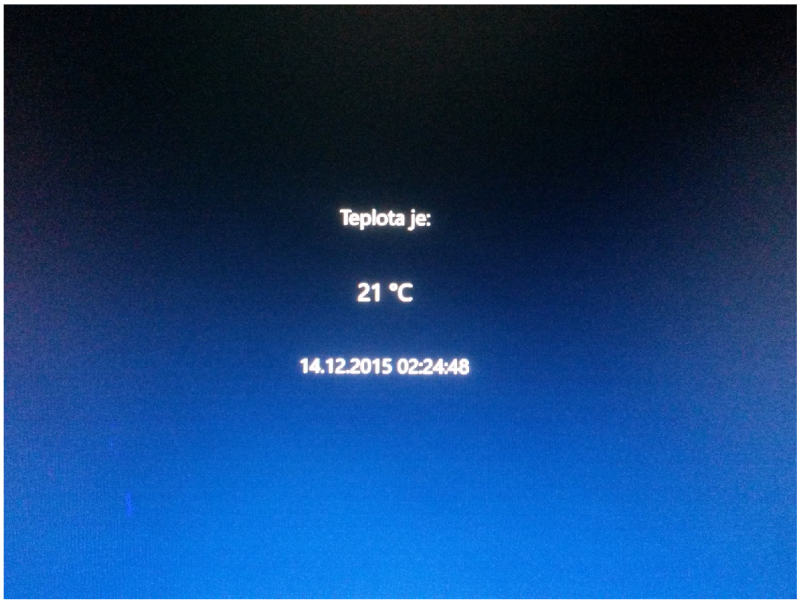


Obr. 5.5: Prepnutie na vzdialené zariadenie

```
private void Timer_Tick(object sender, object e)
{
    DisplayTextBoxContents();
}
        public void DisplayTextBoxContents()
        {
          SpiDisplay.TransferFullDuplex(writeBuffer, readBuffer);
          res = convertToInt(readBuffer);
          textPlaceHolder.Text = String.Join(" ", res, "C");

          string datum = System.DateTime.Now.ToString("dd.MM.yyyy");
          string cas = System.DateTime.Now.ToString("HH:mm:ss");
          tex.Text = String.Join(" ", datum, cas);
         }
```

Do metódy `Timer_Tick`, ktorá v pravidelnom intervale aktualizuje zobrazované hodnoty na monitore sme vložili funkciu, ktorá vypisuje dátum a čas ako reťazec do štítku `tex`. Formát dátumu a času sme upravili na európsky. Pomocou tejto metódy sa aktualizuje aj aktuálna hodnota teploty, ktorá sa ako číselná premenná vkladá do štítku `textPlaceHolder`, kde sme pridali jednotku °C.



Obr. 5.6: Zobrazovanie aktuálnej teploty, dátumu a času

6 ALLJOYN

V tejto kapitole si popíšeme systém AllJoyn a alianciu AllSeen ktorá ho vyvíja. Zameriame sa na spoločnosť Microsoft a systém Windows 10 v spojení s AllJoyn.

6.1 AllSeen

AllJoyn je projekt s otvoreným zdrojovým kódom vedený AllSeen aliance, ktorá je pod hlavičkou Linux Foundation. Systém poskytuje sadu systémových služieb ktoré umožňujú interoperabilitu miliárd prístrojov, služieb a aplikácií pre internet vecí. Pre úspešné dosiahnutie tohoto cieľa bolo za potreby nielen širokej podpory jednotlivých systémových platforiem a procesorových architektúr, pre ktoré je AllJoyn vyvíjaný, ale tiež podpora firiem daného odvetvia, na ktorých závisí, či sa bude tento systém používať namiesto toho, aby každá z nich vyvíjala svoju vlastnú platformu [13].

Vďaka veľkému množstvu firiem, ktoré sa do AllSeen Alliance pripojili, je veľká nádej, že sa podarí víziu obecne prijatého a používaného rámca naplniť. Iniciátorom aliancie je spoločnosť Qualcomm, avšak dnes medzi ich členmi nájdeme radu ďalších, už viac ako 80 významných spoločností, ako napríklad Intel, IBM, Google, RedHat, Sony, Electrolux, Lenovo, Cisco a Microsoft [12].

AllSeen aliancia je v súčasnej dobe rozdelená do deviatich pracovných skupín, kde každá skupina má za úlohu návrh a vývoj určitej časti frameworku. Projektoví vývojári každej skupiny si volia svojho predsedu, ktorý ich zastupuje v technickom riadiacom výbore. Na to všetko dohliada správna rada. Väčšina skupín je inicializovaná jednou veľkou firmou, ktorá následne koordinuje túto časť vývoja a prácu celej skupiny [14].

6.2 AllJoyn

AllJoyn je systém, ktorý umožňuje zariadeniu komunikovať so zariadeniami navôkol. Príkladom by mohol byť pohybový senzor, ktorý dá žiarovke vedieť, že v miestnosti nikto nie je a môže zhasnúť [16].

Framework je navrhnutý aby bol čo najviac univerzálny a jednoducho rozšíriteľný pre nové architektúry, ale aj transportné štandardy. Preto dnes podporuje všetky najviac používane architektúry (x86, x64, ARM a MIPS). Celá platforma je navrhnutá tak, aby závislosť na architektúre bola minimálna a prípadne rozšírenia boli jednoduché. Transport dát je možný pomocou rôznych komunikačných štandardov, ako napríklad Bluetooth, Ethernet alebo Wi-Fi. AllJoyn sa v svojom návrhu snaží byť nezávislý na konkrétnom súčasnom, či prípadnom budúcom transportnom médiu

a protokole, preto je navrhnutý modulárne a to tak, že o vlastnú komunikáciu s fyzickou prenosovou vrstvou sa stará konkrétny modul, čím do budúcna zjednodušuje možnosti rozširovania systému [14].

Implementácia frameworkou je multiplatformová a je možné ju použiť na všetkých dnes bežne používaných operačných systémoch Linux, OS X a Windows. Framework je tak možné nainštalovať na osobné počítače a servery a tiež aj na smerovače založené na OperWRT, chytré telefóny so systémom Android a iOS, mini-počítače Raspberry Pi, systémy RTOS (Real-Time Operating System) ako napríklad ThreadX a podobne [14]. Opäť je kladený dôraz na lokalizáciu systémovo závislého kódu tak, aby prípadné rozšírenie na ďalšie OS bolo čo najjednoduchšie [16].

Podpora programovacích jazykov pre využitie AllJoyn platformy sa odvíja od použitého systému, procesorovej architektúry a subsystému AllJoyn. V súčasnej dobe je dostupné API rozhranie programovacích jazykov Java, C++, C#, C, Objective C a JavaScript [14].

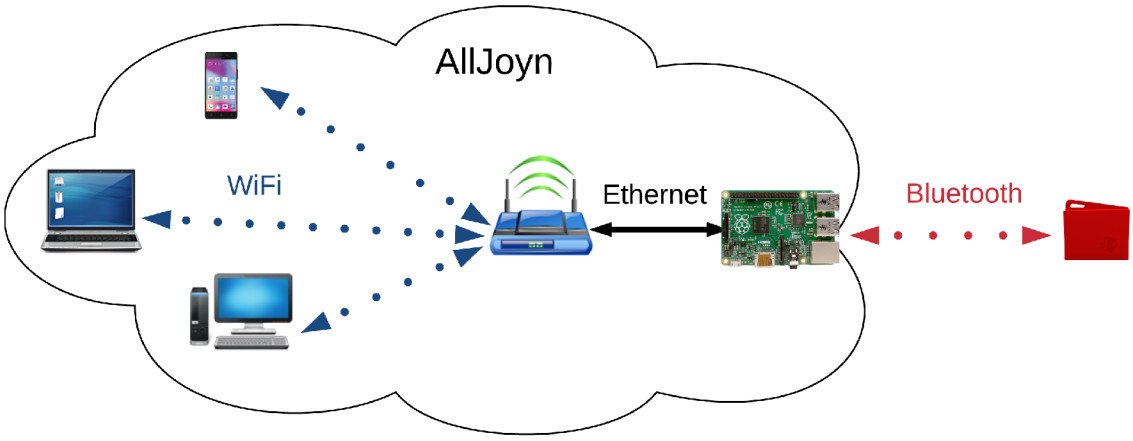
6.3 Microsoft a AllJoyn

Microsoft vstúpil do AllSeen aliancie v roku 2014 a pridal AllJoyn ako kľúčový prvok v systéme Windows 10. S vstavaným AllJoyn API, vývojári môžu slobodne písať AllJoyn aplikácie, ktoré bežia na zariadeniach so systémom Windows 10, vrátane PC, tabletov, telefónov, Xbox, ako aj na zariadeniach s Windows 10 IoT Core. K podpore platformy pre AllJoyn, Microsoft vydal AllJoyn Studio, jedná sa o rozšírenie Visual Studio, ktoré urýchľuje vývoj AllJoyn tým, že kombinuje generovanie kódu a API WinRT s predpripravenými šablónami aplikácií a automatickým riadením projektov. AllJoyn Studio umožňuje vývojárom plne využiť AllJoyn, bez toho aby sa museli obťažovať s nastavením a konfiguráciou [15].

- **Device System Bridge**
  Umožňuje prispôsobenie zariadení ktoré nepodporujú systém AllJoyn aby sa stali súčasťou systému a ich spoločným jazykom sa stáva AllJoyn. To znamená, že bez ohľadu na natívnu konektivitu sa zariadenie zobrazí a bude prístupné ako AllJoyn zariadenie. Vytvára virtuálne znázornenie pre nie AllJoyn domáce zariadenia ako ZigBee a Z-Wave a môže podporovať aj priemyselné systémy pre automatizáciu budov [15].
- **IoT Explorer for AllJoyn (AllJoyn Explorer)**
  Je Windows univerzálna aplikácia pre spojenie AllJoyn zariadení v lokálnej sieti. Umožňuje vidieť všetky dostupné zariadenia, kontrolovať ich rozhrania, nastavovať, prijímať signály a ovládať ich [15].

7 MERANIE TEPLOTY POMOCOU SENSORTAG CC2650 A VYTVORENIE ALLJOYN APLIKÁCIE

V tejto kapitole sa popisuje meranie svetelnosti, teploty a vlhkosti vzduchu pomocou senzoru TI SensorTag CC2650 komunikujúcemu s Raspberry Pi 2 cez Bluetooth. Popisuje sa tiež vytvorenie AllJoyn aplikácie a výsledné zobrazovanie hodnôt zdieľaných v sieti AllJoyn.Viď. Obr. 7.1



Obr. 7.1: Bloková schéma AllJoyn siete

7.1 TI SensorTag v2 CC2650STK

V tejto práci sme použili zariadenie SensorTag CC2650 ktorý komunikuje s Raspberry Pi 2 cez Bluetooth. Keďže Raspberry Pi 2 nemá bluetooth modul na doske, tak sme cez USB rozhranie pripojili bluetooth dongle, ktorý nebolo potrebné nijako nastavovať.

SensorTag v2 bol vyvinutý spoločnosťou Texas Instruments pre internet vecí, ako zariadenie s desiatimi nízko energetickými MEMS senzormi komunikujúcimi full-duplexne a v reálnom čase prostredníctvom BLE (Bluetooth Low Energy). Zariadenie umožňuje senzory spúšťať osobitne, čo ma pozitívny vplyv na výdrž batérie a v porovnaní s verziou 1 je o 75% úspornejšie. Senzory obsiahnuté v zariadení sú svetelný senzor, mikrofón, senzor vlhkosti vzduchu, kontaktný teplotný senzor, priestorový teplotný senzor, magnetický senzor, tlakový senzor, magnetometer, accelometer a gyroskop. Zariadenie je riadené procesorom ARM Cortex M3.

Pre zapnutie senzora bolo potrebné pracovať s GATT(Generic Attribute Profile) charakteristikami a adresami UUID (Universally Unique Identifier) získanými z TI

Obr. 7.2: TI SensorTag v2 CC2650STK

wiki <`http://processors.wiki.ti.com/index.php/CC2650_SensorTag_User's_Guide`>.

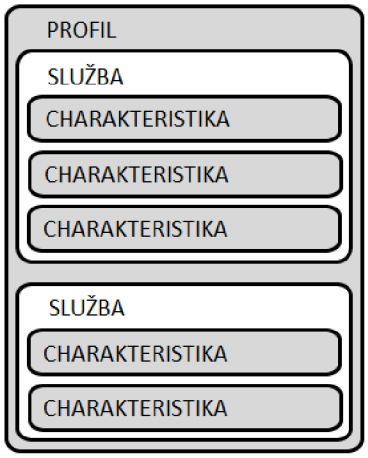
GATT definuje ako budú spolu komunikovať a prenášať dáta dve BLE zariadenia pomocou tzv. služieb a charakteristík. Protokol nazývaný ATT(Atribute protocol), ktorý poskytuje zoznam služieb, vlastností a dát požíva 16-bitové ID pre každú položku v zozname a druhé zariadenie sa pomocou adresy dopytuje priamo na danú službu. GATT spojenie je exkluzívne, to znamená, že BLE periférie je možné v čase pripojiť len do jedného centrálneho zariadenia(chytrý telefón, počítač, atď.). Akonáhle sa periférne zariadenie pripojí k centrálnemu zariadeniu, prestane sa inzertovať ako dostupné zariadenie a nebude ho viac možné vidieť, alebo sa k nemu pripojiť, dokiaľ nie je existujúce spojenie prerušené. K jednému centrálnemu zariadeniu môže byť ale pripojených perifférnych zariadení niekoľko [17].

Nadviazanie spojenia je jediný spôsob ako umožniť obojsmernú komunikáciu, kde periférne zariadenie odošle zmysluplné dáta a naopak [17].

V GATT sa využíva spojenie klient-server. Periférne zariadenie je GATT Server, ktorý má definície ATT vyhľadávaných dát, služieb a vlastností a centrálne zariadenie ktoré je GATT Klient posiela žiadosti tomuto serveru. Všetky prenosy začínajú GATT Klientom, ktorý dostáva odpovede od GATT Servera. Pri nadväzovaní spojenia, periférne zariadenie navrhne pripojovací interval centrálnemu zariadeniu a to sa pokúsi pripojiť každý interval a zistiť či sú k dispozícii nejaké nové dáta [17].

GATT prenosy v BLE sú založené na vnorených objektoch nazývaných profily, služby a charakteristiky.Viď. Obr 7.3.

- Profil v skutočnosti neexistuje na samotnom BLE perifférnom zariadení, je to jednoduchý preddefinovaný súbor služieb, ktorý bol zostavený návrhármi periférií.
- Služba sa používa na delenie dát do logických entít a obsahuje konkrétne bloky dát nazývané charakteristiky. Jedna služba môže mať jednú, alebo viac cha-



Obr. 7.3: Profily, služby a charakteristiky v GATT

rakteristík a každá služba sa odlišuje od ostatných prostredníctvom UUID (unikátneho číselného ID).

- Charakteristika je jeden dátový bod aj keď môže obsahovať rad súvisiacich dát, ako sú hodnoty X/Y/Z z accelerometra. Rovnako ako u služieb, sa charakteristika rozlišuje podľa preddefinovaného UUID.

7.2 Definovanie komunikácie

Najprv musíme zistiť aké je rozhranie medzi zariadením a aplikáciou ktoré bude kontrolovať a monitorovať zariadenie. V AllJoyn Frameworku je to prostredníctvom XML dokumentácie. V mojej práci som použil dokumentáciu ktorá bola dostupná ako vzorová na stránkach wiki AllSeen aliancie a doplnil podľa svojich potrieb.

Moja XML dokumentácia:

```
1 <node xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/
      schemas/introspect.xsd">
2     <interface name="cz.davidvalus.SensorValues">
3         <description language="en">This interface provides
              capability to represent sensor values from SensorTag2.</
              description>
4         <annotation name="org.alljoyn.Bus.Secure" value="false"/>
5         <property name="Version" type="q" access="read">
6             <description language="en">The interface version</
                  description>
7             <annotation name="org.freedesktop.DBus.Property.
                  EmitsChangedSignal" value="true"/>
8         </property>
```

```
9         <property name="CurrentTemp" type="d" access="read">
10            <description language="en">Current temperature
                 expressed in Celsius</description>
11            <annotation name="org.freedesktop.DBus.Property.
                 EmitsChangedSignal" value="true"/>
12        </property>
13        <property name="CurrentLux" type="d" access="read">
14            <description language="en">Current ambient light
                 expressed in LUX</description>
15            <annotation name="org.freedesktop.DBus.Property.
                 EmitsChangedSignal" value="true"/>
16        </property>
17        <property name="CurrentHumidity" type="d" access="read">
18            <description language="en">Current humidity expressed
                 in % </description>
19            <annotation name="org.freedesktop.DBus.Property.
                 EmitsChangedSignal" value="true"/>
20        </property>
21    </interface>
22 </node>
```

7.3 Vytváranie vlastnej knižnice AllJoyn rozhraní prostredníctvom AllJoyn CodeGen Tool

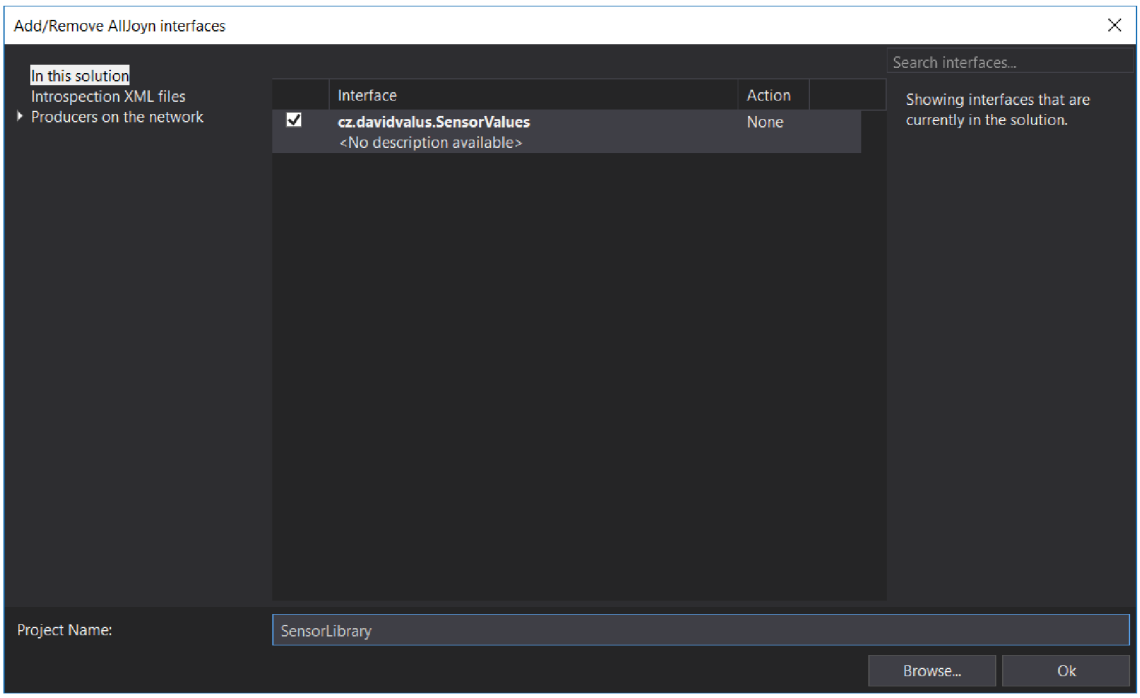
V tomto kroku budeme používať Visual Studio 2015 a AllJoyn Studio Extension na konvertovanie našej XML dokumentácie do knižnice ktorá odkazuje do UWP aplikácie.

Pred začatím je nutné si stiahnuť a nainštalovať program AllJoyn Studio. Inštaláciou sa pridá rozšírenie do Visual Stuio 2015.

1. Spustíme Visual Studio 2015 a z menu vyberieme `File -> New Project`.
2. Vyberieme `Windows -> Universal ->AllJoyn App`.
3. Pomenujeme project AllJoynSensorApp.
4. V menu Visual Studio vyberieme `AllJoyn-> Add/Remove Interfaces`.
5. Klikneme na tlačítko Browse a vyberieme náš xml súbor.
6. Zašktneme políčko a projekt si premenujeme. Viď. Obr. 7.4.
7. Klikneme OK.

Po dokončení týchto krokov sa vygenerovala knižnica, vďaka ktorej vieme komunikovať prostredníctvom AllJoyn. Viď. Obr. 7.5

Pre správne fungovanie AllJoyn použijeme menný priestor Windows.Devices.Alljoyn a názov knižnice ktorú chceme použiť. Viď. Obr. 7.6



Obr. 7.4: Pridávanie XML dokumentácie

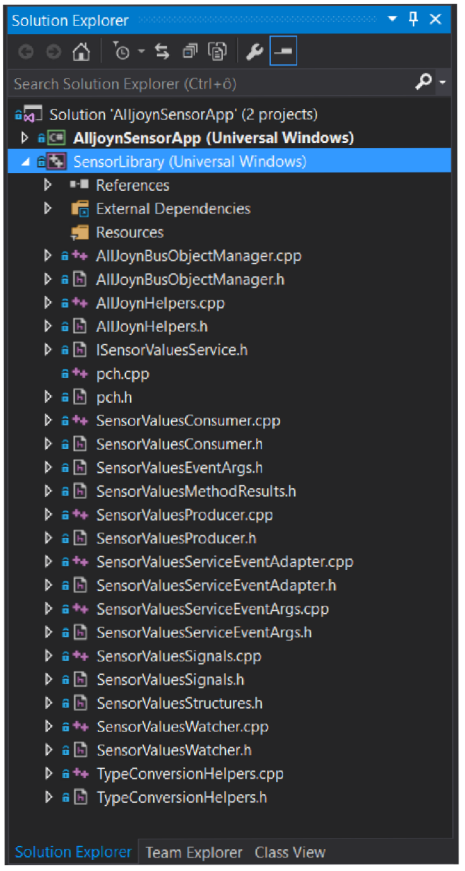
7.4 AllJoyn Aplikácia

Aplikáciu AllJoyn sme si vytvorili od základov a vytvorenie aplikácie sa skladlo z dvoch hlavných častí.

Prvou časťou bola komunikácia so SensorTag, z čoho vyplývalo zapínanie jednotlivých senzorov, práca s GATT a čítanie dát. Práca so senzormi bola zložitá a bolo nutné pracovať s dokumentáciiou z ktorej sme získali UUID služieb, popisy a veľkosti jednotlivých charakteristík a v neposlednom rade informácie o kalibrácii senzorov. [1]

Druhou časťou bolo zdieľanie dát v sieti AllJoyn a ich zobrazovanie prostredníctvom aplikácie IoT Explorer for AllJoyn. V tejto časti si bolo potrebné definovať rozhrania pomocou xml dokumentácie, ktorú sme získali editovaním a rozšírením vzorovej aplikácie dostupnej na stránkach AllSeen aliancie. Ďalej sme si pomocou nástroja AllJoyn Code Gen Tool vygenerovali vlastnú AllJoyn knižnicu ktorá sa odkazovala do UWP aplikácie a vďaka ktorej sme vedeli komunikovať s ostatnými AllJoyn zariadeniami v sieti.

[1] Dokumentácia SensorTag 2 je dostupná na stránkach Texas Instruments Wiki <`http://processors.wiki.ti.com/index.php/CC2650_SensorTag_User's_Guide`>.



Obr. 7.5: Vygenerované súbory

```
using cz.davidvalus.SensorValues;
using Windows.Devices.AllJoyn;
```

Obr. 7.6: Menné priestory

Nasledujúca časť kódu zobrazuje čítanie hodnoty z konfiguračnej charakteristiky senzoru a prácu s GATT charakteristikami a adresami UUID. Ak prečítaná hodnota je 1, znamená to, že senzor je zapnutý, ak je hodnota 0 senzor je vypnutý a zapnem ho zapísaním logickej 1 do jeho charakteristiky. Viď časť kódu 7.1

```
var config = service.GetCharacteristics(guid_config)[0];
            if (config != null)
            {
                GattReadResult sensorStatus = await config.
                    ReadValueAsync();
                Debug.WriteLine("Sensor config: " + sensorStatus.
                    Value.ToArray()[0]);
                //Ak nie je zapnuty senzor
                if (sensorStatus.Value.ToArray()[0] == 0x00)
                {
                    Debug.WriteLine("Sensor off, turning on sensor
                        ...");
                    DataWriter writer = new DataWriter();
                    writer.WriteByte(0x01);
                    await config.WriteValueAsync(writer.
                        DetachBuffer());

                    sensorStatus = await config.ReadValueAsync();
                    Debug.WriteLine("Sensor config: " +
                        sensorStatus.Value.ToArray()[0]);
                }
                else if (sensorStatus.Value.ToArray()[0] == 0x01)
                {
                    Debug.WriteLine("Sensor is on already.");
                }

            }
```

Listing 7.1: Zapínanie senzora

Zapínanie senzorov je asynchrónna operácia a to spôsobovalo problémy pri zapínaní oboch senzorov súčasne. Preto som zvolil riešenie, že sa senzory zapínajú s funkciou `Timer_Tick`. Svetelný senzor sa zapne až 5s po senzore vlhkosti vzduchu. Funkcia `producer.Emit` bola vygenerovaná pre producer v knižnici a aktualizuje hodnoty zdieľané v AllJoyn sieti každých 5s, tiež prostredníctvom funkcie `Timer_Tick`. Viď. časť kódu 7.2

```
private void Timer_Tick(object sender, object e)
        {
            if(sensorsEnabled >= 2)
            {
                allEnabled = true;
            }

            if(sensorsEnabled >= 1 && !allEnabled)
            {
                //Zapni dalsi senzor
                initBlueTooth("AA70", "AA71", "AA72");
            }

            //Ak su zapnute vsetky senzory
            if(allEnabled)
            {
                Debug.WriteLine("tick");
                readSensors();

                service.setTemp(sensor_hum_temperature);
                service.setHumidity(sensor_hum_humidity);
                service.setLight(sensor_light);

                producer.EmitCurrentTempChanged();
                producer.EmitCurrentHumidityChanged();
                producer.EmitCurrentLuxChanged();
            }
        }
```

Listing 7.2: Zapínanie druhého senzora

Nasledujúca funkcia bola implementovaná cez rozhranie vygenerované AllJoyn Code Gen. Doplnil som aby sa do funkcie vracala aktuálna hodnota vlhkosti vzduchu.

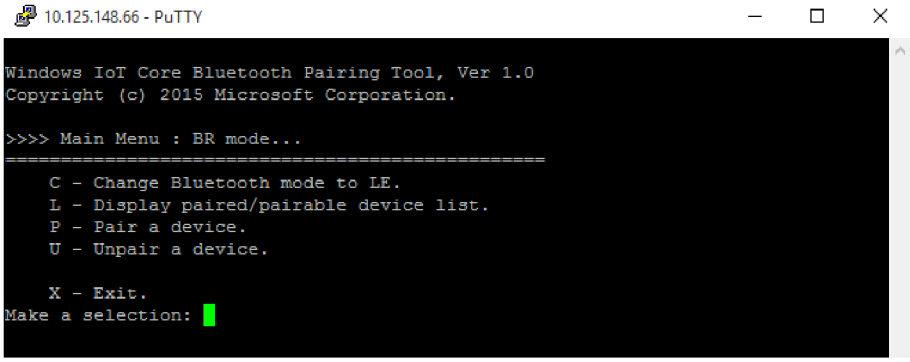
```
public IAsyncOperation<SensorValuesGetCurrentHumidityResult>
    GetCurrentHumidityAsync(AllJoynMessageInfo info)
        {
            Task<SensorValuesGetCurrentHumidityResult> task = new
                Task<SensorValuesGetCurrentHumidityResult>(() =>
            {
                return SensorValuesGetCurrentHumidityResult.
                    CreateSuccessResult(humidity);
            });

            task.Start();
            return task.AsAsyncOperation();
```

7.5 Spúšťanie aplikácie

Pred spustením aplikácie musíme spárovať Raspberry Pi 2 a SensorTag CC2650. Máme dve možnosti spárovania zariadení prostredníctvom bluetooth rozhrania. Jednou možnosťou je spárovať zariadenia prostredníctvom Windows IoT Core Bluetooth Pairing Tool.

1. Pripojíme sa k zariadeniu prostredníctvom SSH.
2. Spustíme príkaz `C:\Windows\System32\IoTBluetoothPairing.exe`. Viď. Obr. 7.7
3. Overíme si, či sme v móde LE(low energy), v ostatnom prípade nebude možné spárovať Raspberry Pi 2 s BLE zariadením. Módy bluetooth prepíname stlačením C.
4. Vyberieme možnosť `P - Pair a device`, zvolíme senzor a potvrdíme klávesou enter. Ak je proces párovania úspešný zadáme PIN 000000, ktorý je predvolený pre SensorTag.
5. Pre kontrolu procesu párovania zvolíme `L - Display paired/pairable device list`, ak sa tam nachádza senzor, párovanie bolo úspešné.

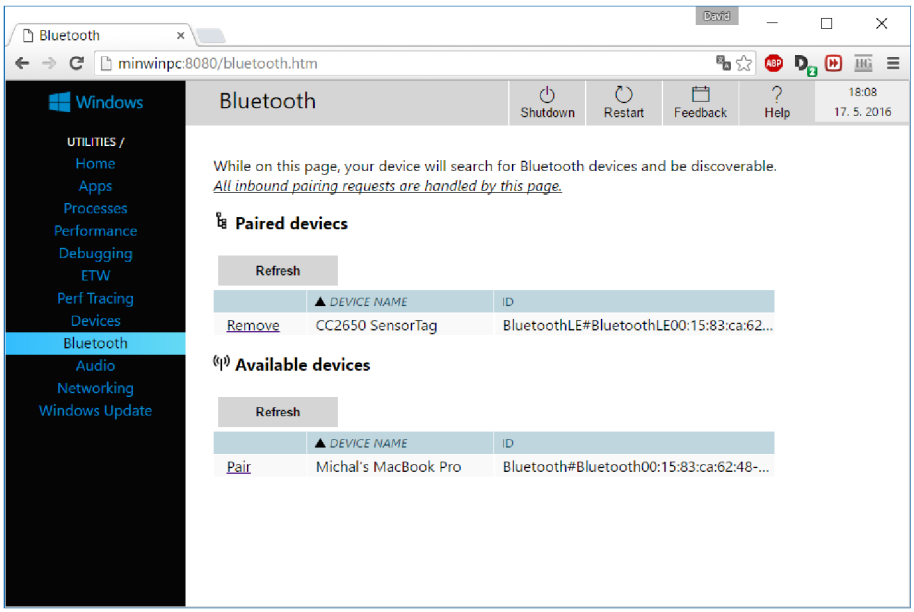


Obr. 7.7: Menu Windows IoT Core Bluetooth Pairing Tool

Druhou možnosťou je spárovať zariadenia cez internetový prehliadač.

1. Ako adresu v internetovom prehliadači zadáme `http://<menozariadenia>:8080`.
2. Po pripojení je nutné sa autentifikovať.
3. Z ľavého menu zvolíme položku `Bluetooth`.
4. V zozname Available devices, vyberieme senzor a klikneme na `Pair`.
5. Po úspešnom spárovaní sa senzor presunie do zoznamu `Paired devices`. Viď. Obr. 7.8

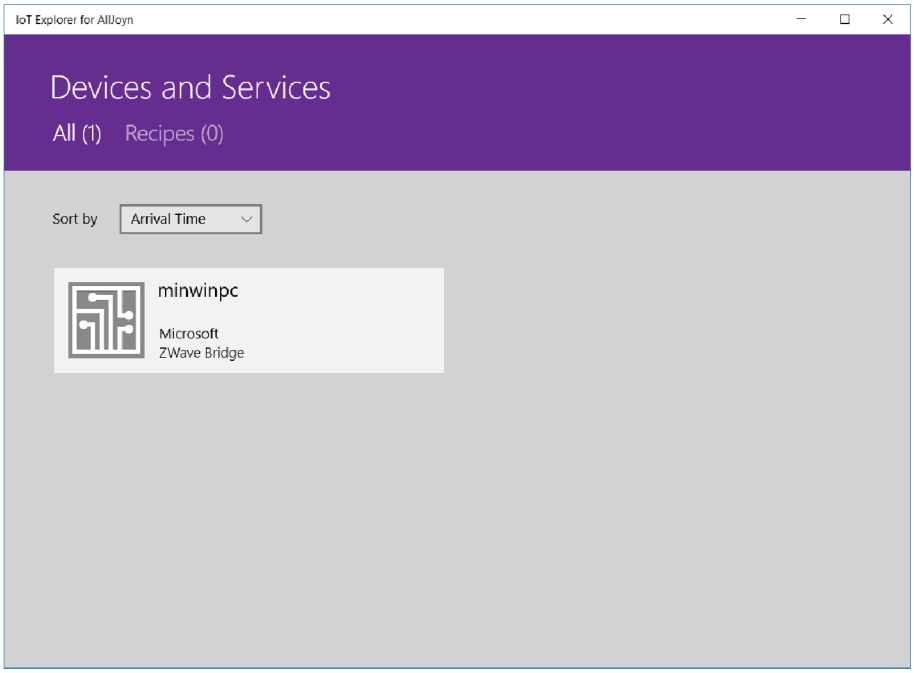
Aplikáciu môžeme spustiť cez Visual Studio 2015 v debugovacom móde, alebo cez internetový prehliadač, kde ju môžme nastaviť aj ako predvolenú.



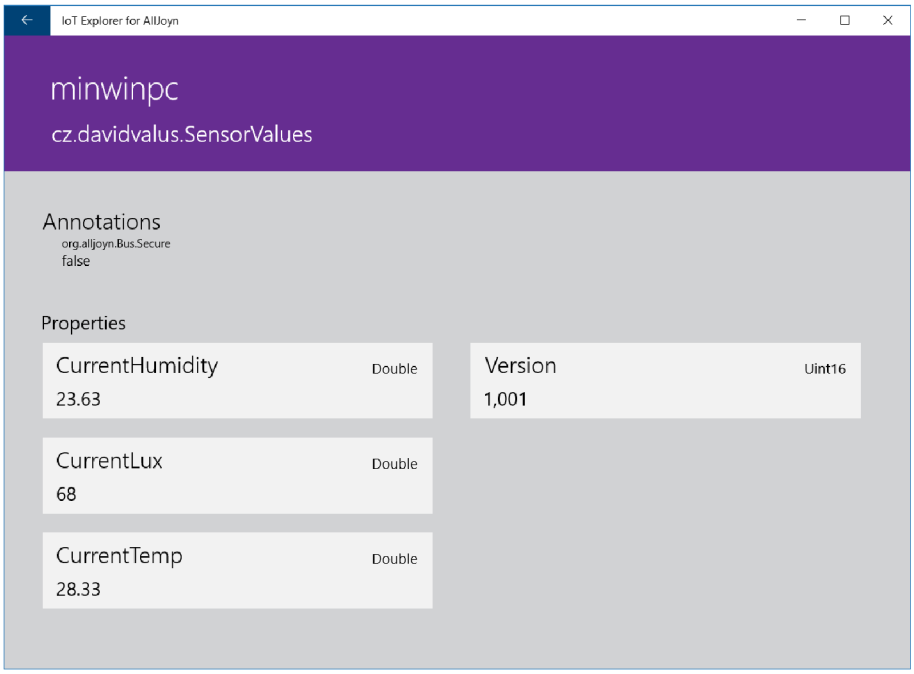
Obr. 7.8: Párovanie zariadení cez internetový prehliadač

7.6 Zobrazovanie hodnôt

Z Microsoft Obchodu si stiahneme a nainštalujeme aplikáciu IoT Explorer for AllJoyn. Po spustení aplikácie sa nám zobrazia dostupné zariadenia v sieti AllJoyn. 7.9 Klikneme na `minwinpc-> Service-> cz.davidvalus.SensorValues`. Po kliknutí sa nám zobrazuje svetelnosť, teplota a vlhkosť vzduchu, ktorých hodnoty sa v pravidelných intervaloch aktualizujú. 7.10



Obr. 7.9: Zoznam zariadení v AllJoyn sieti



Obr. 7.10: Zobrazovanie hodnôt prostredníctvom AllJoyn

8 ZÁVER

Zoznámili sme sa s technológiou M2M, IoT a platformou AllJoyn, ktoré zažívajú rozmach a v budúcnosti nájdu uplatnenie vo viacerých sektoroch. Pracovali sme so systémom Windows 10 IoT Core, ktorý je zdarma a predstavuje alternatívu voči mnohým Linuxovým distribúciám pre platofrmu Raspberry Pi 2, DragonBoard410c, MinnowBoard a iné. V teoretickej časti sme porovnali systém Windows 10 IoT Core s najrozšírenejšími systémami pre Raspberry Pi 2, popísali ich funkcie, využitie a zhrnuli sme si ich výhody.

Cieľom praktickej časti bakalárskej práce bolo overiť funkčnosť systému Windows 10 IoT Core na doske Raspberry Pi 2 a jeho podporu platformy AllJoyn. Praktická časť sa skladala z dvoch častí. V prvej sme si prispôsobili Windows 10 aplikáciu na meranie teploty, ktorá bola dostupná na stránkach Microsoft. Vytvorili sme si jednoduché grafické rozhranie pre aplikáciu kde sme spolu s aktuálnym dátumom a časom zobrazovali teplotu. Na meranie teploty sme použili teplotný senzor LM35 a A/D prevodník MCP3208, ktoré komunikovali s doskou Raspberry Pi 2 cez SPI rozhranie.

V druhej časti sme pracovali so systémom AllJoyn ktorý spája IoT zariadenia rôznych výrobcov a komunikujúce v rôznych jazykoch do jednej siete aby mohli spolu komunikovať a zdieľať informácie. Od základov sme si vytvorili Windows 10 AllJoyn aplikáciu na meranie svetelnosti, teploty a vlhkosti vzduchu pomocou senzora TI SensorTag CC2650, ktorý komunikoval s Raspberry Pi 2 cez Bluetooth rozhranie. Namerané hodnoty sme zdieľali v reálnom čase prostredníctvom AllJoyn siete.

V bakalárskej páci boli dosiahnuté všetky stanovené ciele. Prípadnému pokračovateľovi by som doporučil vytvorenie webovej aplikácie, kde by sa dáta zaznamenávali, prípadne vykresľovali grafy a boli prístupné cez Internet.

LITERATÚRA

[1] HOLLER, Jan. From machine-to-machine to the Internet of things: introduction to a new age of intelligence. xix, 331 pages. ISBN 9780080994017.

[2] BOSWARTHICK, David, Omar ELLOUMI a Olivier HERSENT. M2M communications: a systems approach. Hoboken, N.J.: Wiley, 2012, xxiii, 308 p.

[3] MAKRIS, Prodromos, Dimitrios N. SKOUTAS, Nikolaos NOMIKOS, Demosthenes VOUYIOUKAS a Charalabos SKIANIS. A Context-Aware Backhaul Management solution for combined H2H and M2M traffic. 2013 International Conference on Computer, Information and Telecommunication Systems (CITS). IEEE, 2013, s. 1-5. DOI: 10.1109/CITS.2013.6705719. Dostupné z: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6705719

[4] Juniper Networks, Inc., 2014 MACHINE-TO-MACHINE(M2M)-THE RISE OF THE MACHINES , 2011.

[5] Windows 10 for the Internet of Your Things [online]. 2015 [cit. 2015-12-01]. Dostupné z: https://www.microsoft.com/en-us/WindowsForBusiness/windows-iot

[6] Minimum hardware requirements. Msdn.microsoft.com [online]. 2015 [cit. 2015-12-13]. Dostupné z: https://msdn.microsoft.com/en-us/library/windows/hardware/dn915086(v=vs.85).aspx

[7] RASPBERRY PI 2 ON SALE NOW AT $35 [online]. 2015, 2015 [cit. 2015-12-01].Dostupné z URL: Dostupné z: <`https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/`>.

[8] CALIN, Dragos George. A sweet selection of the Raspberry Pi 2 OS. Intorobotics [online]. 2015 [cit. 2015-12-05]. Dostupné z: http://www.intorobotics.com/a-sweet-selection-of-the-raspberry-pi-2-os/

[9] DRAGICEVIC, Marko. Distributions for the Raspberry Pi at a glance. LINUX NEW MEDIA USA, LLC. Raspberry-pi-geek [online]. 2013 [cit. 2015-12-05]. Dostupné z: http://www.raspberry-pi-geek.com/Archive/2013/01/Distributions-for-the-Raspberry-Pi-at-a-glance

[10] Support for Windows on Intel Galileo Gen 1 and Gen 2 will end on November 30, 2015 [online]. 2015 [cit. 2015-12-01]. Dostupné z: https://ms-iot.github.io/content/en-US/win8/SetupGalileo.htm

[11] Get Started Install Windows 10 IoT Core [online]. 2015, 2015 [cit. 2015-12-01]. Dostupné z: https://ms-iot.github.io/content/en-US/GetStarted.htm

[12] Technology Leaders Establish the AllSeen Alliance to Advance the 'Internet of Everything'. Http://www.linuxfoundation.org/ [online]. San Francisco, 2013 [cit. 2016-05-17]. Dostupné z: http://www.linuxfoundation.org/news-media/announcements/2013/12/technology-leaders-establish-allseen-alliance-advance-

[13] Documentation. Www.allseenalliance.org [online]. 2016 [cit. 2016-05-17]. Dostupné z: https://allseenalliance.org/framework/documentation/learn

[14] Core Framework. Https://allseenalliance.org [online]. 2016 [cit. 2016-05-17]. Dostupné z: https://allseenalliance.org/framework/documentation/learn/core

[15] AllJoyn [online]. 2016 [cit. 2016-05-17]. Dostupné z: https://developer.microsoft.com/en-us/windows/iot/win10/AllJoyn.htm

[16] The AllJoyn™ Open Source Project. Www.linuxfoundation.org [online]. 2015 [cit. 2016-05-17]. Dostupné z: <`https://events.linuxfoundation.org/images/stories/pdf/lf_abs12_lioy.pdf`>

[17] GATT: Kevin Townsend. Https://learn.adafruit.com [online]. 2015 [cit. 2016-05-17]. Dostupné z: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

M2M	Machine to Machine
H2H	Human tu Human
IoT	Internet of Things
LTE	Long Term Evolution
GPS	Global Positioning System
QoS	Quality of Services
ARM	Acorn RISC Machine
LAN	Local Area Network
GB	Giga-Byte
CPU	Computer Processor Unit
SSD	Solid State disc
HDMI	High-Definition Multimedia Interface
GUI	Graphic User Interface
USB	Universal Serial Bus
OS	Operating system
UWP	Universal Windows Platform
IP	Internet Protocol
SSH	Secure Shell
I/O	Input/Output
ETW	Event Tracing for Windows
SPI	Serial Peripheral Interface
GPIO	General-purpose input/output
SPI	Serial Peripheral Interface
GATT	Generic Attribute Profile

BLE	Bluetooth Low Energy
UUID	Universally Unique Identifier

ZOZNAM PRÍLOH

A Obsah priloženého CD **51**

A OBSAH PRILOŽENÉHO CD

- AllJoynSensorApp
  Priečinok s aplikáciou AllJoyn ktorý obsahuje všetky súbory.

- IntrospectionSensorValues.xml
  XML súbor ktorým sme pomocou AllJoyn Code Gen vytvorili AllJoyn aplikáciu.

- BP-Valus.pdf
  Elektronická verzia bakalárskej práce.