

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

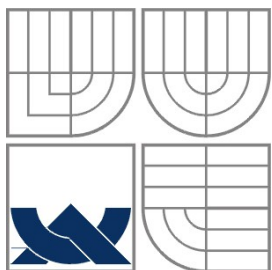
KOMUNIKACE MEZI SYSTÉMEM ANDROID
A PLATFORMOU ARDUINO

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

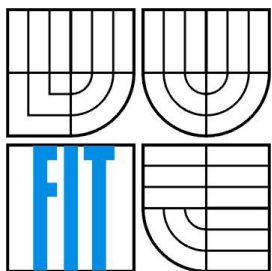
AUTOR PRÁCE
AUTHOR

Bc. Maksim Mychko

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

KOMUNIKACE MEZI SYSTÉMEM ANDROID A PLATFORMOU ARDUINO

COMMUNICATION BETWEEN THE ANDROID OS AND ARDUINO PLATFORM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MAKSIM MYCHKO

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN SAMEK, Ph.D.

BRNO 2015

Abstrakt

Diplomová práce se zabývá vytvořením knihovny pro OS Android, která umožňuje efektivní bezdrátovou komunikaci s platformou Arduino. Cílem práce je vytvoření aplikace pro OS Android, s využitím vytvořené knihovny, která umožňuje snímání dat z různých senzorů a řízení efektorů. V práci jsou probrána základní bezdrátová rozhraní OS Android a platformy Arduino, a také popsány jejich hlavní výhody i nevýhody a funkce nebo příkazy pro práci s nimi. Bylo provedeno srovnání spotřeby proudu moduly HM-10 a DIGI S1 a testování snímání dat z různých senzorů a řízení efektorů.

Abstract

This thesis deals with creating a library for the Android OS which enables efficient wireless communication with platform Arduino. The target is to create an application for Android OS, using a created library that allows you to capture data from various sensors and control effectors. The work describes basic wireless interfaces for Android OS and Arduino platform. The main benefits and drawbacks functions are also introduced. The comparison of current consumption for modules HM-10 and DIGI S1 has been carried out as well. Possibility of data acquisition and effector controlling exercises have been tested and verified.

Klíčová slova

Android, Arduino, IoT, Bluetooth low energy, BLE, HM-10, XBee, DigiMesh, OTG, Zigbee, NFC, GSM, WIFI.

Keywords

Android, Arduino, IoT, Bluetooth low energy, BLE, HM-10, XBee, DigiMesh, OTG, Zigbee, NFC, GSM, WIFI.

Citace

Mychko Maksim: Komunikace mezi systémem Android a platformou Arduino, diplomová práce, Brno, FIT VUT v Brně, 2015

Komunikace mezi systémem Android a platformou Arduino

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jana Samka, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Maksim Mychko
01.05.2015

Poděkování

Chtěl bych poděkovat vedoucímu této práce Ing. Janu Samkovi, Ph.D., za jeho pomoc při vytváření této práce.

© Maksim Mychko, 2015

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
1.1 Motivace.....	3
1.2 Cíl a struktura práce.....	3
2 Bezdrátová komunikace v OS Android.....	5
2.1 Bluetooth.....	5
2.1.1 Klasický Bluetooth.....	6
2.1.2 Bluetooth Low Energy.....	7
2.2 USB.....	9
2.3 NFC.....	12
2.4 WIFI.....	12
2.5 GMS/UMTS.....	13
3 Arduino.....	15
3.1 Bluetooth.....	16
3.1.1 Klasický Bluetooth.....	16
3.1.2 Bluetooth Low Energy.....	16
3.2 ZigBee.....	18
3.3 NFC.....	20
3.4 WIFI.....	22
3.5 GSM.....	22
4 Návrh knihovny funkcí pro OS Android.....	25
5 Implementace protokolu pro komunikace OS Android a Arduino.....	27
5.1 Seznam příkazů pro platformu Arduino.....	27
5.2 Struktura Android aplikace.....	32
5.2.1 Struktura modulu grafického rozhraní (Android).....	34
5.2.2 Struktura modulu pro komunikaci přes BLE (Android).....	35
5.2.3 Struktura modulu pro komunikaci přes UART (Android).....	36
6 Testování a měření spotřeby.....	37
7 Závěr.....	43
7.1 Možnosti rozšíření.....	43
7.2 Zhodnocení.....	43
Literatura.....	44
Seznam příloh.....	47
Příloha A.....	48

Příloha B.....	51
Příloha C.....	53
Příloha D.....	56
Příloha E.....	60

1 Úvod

1.1 Motivace

Jedny z nejperspektivnějších a nejrychleji rostoucích oblastí jsou inteligentní domácnost, nositelná elektronika a internet věcí [37].

Arduino platforma je určená pro rychlé vytvoření prototypu a je ideální pro vytvoření řídicí jednotky pro práci s efekty a/nebo snímání hodnot ze senzorů, která mohou být použita jako součást inteligentní domácnosti, nositelné elektroniky a internetu věcí. Tato platforma je jedna z nejvíce populárních, protože je open-source, zaměřená na neprofesionální uživatele a má velké množství dokumentace a příkladů.

Popularita zařízení s OS Android dělá tyto přístroje ideální pro jejich používání jako serveru pro řízení prvků z inteligentní domácnosti, nositelné elektroniky a internetu věcí.

V této diplomové práci budu řešit jeden z důležitých problémů – zjednodušení vytvoření zkušebního výrobku (prototypu) řídicí jednotky pro práci s efekty a/nebo snímání hodnot ze senzorů, která slouží jako základní prvek pro inteligentní domácnosti, nositelnou elektroniku a internet věcí.

Hlavní výhodou je, že programátor může začít dělat prototyp bez znalosti platformy Arduino. Všechny příkazy v protokolu jsou snadno čitelné a logické. Stačí jen připojit k Arduino, ve kterém už je nahrán „sketch“, senzory a/nebo efekty a bezdrátový modul, kterých je, vzhledem k popularitě platformy Arduino, na trhu obrovské množství za nízké ceny. Následně je možné v aplikaci pod OS Android vytvořit logiku práce se senzory a/nebo efekty a spustit program. Dále, když bude prototyp provádět veškeré plánované funkce, je možné převést část výpočtu do platformy Arduino.

Aktuálnost tohoto rozhodnutí může nepřímo potvrdit úspěch podobného projektu na kickstarter.com - The AirBoard, který překročil počáteční práh financování 6,5krát a dosáhl \$65,014 [4].

1.2 Cíl a struktura práce

Cílem této diplomové práce je navrhnout a realizovat knihovnu funkcí pro OS Android a platformu Arduino spolu s aplikací pro OS Android, která umožní snímání dat z různých senzorů a řízení efektorů připojených na platformu Arduino. Základem pro knihovnu pro OS Android bude sloužit sada příkazů pro práci s výstupy/vstupy platformy Arduino. Arduino poslouží ke snímání dat ze senzorů i řízení efektorů a za pomoci bezdrátového modulu BLE nebo DigiMesh bude odesílat data na zařízení s OS Android. Aplikace pod OS Android bude zobrazovat data a řídit platformu Arduino.

Na začátku této práce, v druhé a třetí kapitole se čtenář seznámí se základními bezdrátovými rozhraními OS Android a platformy Arduino. Budou popsány hlavní výhody a nevýhody bezdrátových rozhraní, a také funkce nebo příkazy pro práci s nimi. Dále, ve čtvrté a páté kapitole bude popsán návrh a implementace protokolu a knihovny. Na začátku kapitoly 5 jsou uvedeny základní příkazy protokolu pro práci s platformou Arduino. Dále v kapitole 5 bude probrána struktura a funkce knihovny OS Android. V kapitole 6 jsou popsány testy s různými senzory i efekty a srovnání aktuální spotřeby dvou bezdrátových modulů HM-10 (BLE) a DIGI S1 (DigiMesh). Kapitola 7 popisuje možnost rozšíření knihovny pro OS Android i Arduino a rovněž pro aplikace pro OS Android.

2 Bezdrátová komunikace v OS

Android

Android nabízí bohaté API (rozhraní pro programování aplikací) pro komunikaci přes Bluetooth, NFC, Wi-Fi, USB, P2P, SIP a standardní síťové připojení (radiové rozhraní).

2.1 Bluetooth

Technologie Bluetooth je definovaná standardem IEEE 802.15.1. Pracuje v pásmu 2400–2483,5 MHz (ISM – pásma pro rádiové vysílání v průmyslových, vědeckých a zdravotnických oborech. Provoz bez licenčních poplatků je dovolen). K přenosu využívá metody FHSS (frequency hopping spread spectrum).

Verze Bluetooth	Rychlost přenosu dat
Bluetooth v1.2	1 Mbit/s
Bluetooth v2.0 (ERD)	3 Mbit/s
Bluetooth v3.0 (HS)	24 Mbit/s
Bluetooth v4.0	1 Mbit/s

Tabulka 2.1: Maximální teoretické přenosové rychlosti podle standardů [1].

Je definováno několik výkonových úrovní (1 mW, 10 mW, 100 mW), s nimiž je umožněna komunikace do vzdálenosti cca 1–100 m [1].

Class	Maximální povolený výkon		Dosah (přibližný)
	mW	dBm	
Class 1	100	20	~100 metrů
Class 2	10	4	~10 metrů
Class 3	1	0	~1 metr

Tabulka 2.2: Třídy výkonnosti podle standardů [1].

Bluetooth podporuje jak dvoubodovou, tak mnohabodovou komunikaci. Při navázání spojení jedno zařízení vystupuje jako řídicí (master) a druhé jako podřízený (slave), což v průběhu spojení pak jako iniciátor (initiator) a inzerent (advertizer).

2.1.1 Klasický Bluetooth

Bezdrátová komunikace přes rozhraní Bluetooth je k dispozici od verze Android 1.0 (API level 1). OS Android obsahuje podporu automatického párování a stereo audio přenos od verze 1.5 (API level 3). Podpora Bluetooth verze 2.1 se objevila v Android 2.0 (API level 5).

Vlastnost	Bluetooth 2.0 (ERD)	Bluetooth 3.0 (HS – wifi mode)
Rychlost přenosu dat	1–3 Mbit/s	2,1–24 Mbit/s
Technologie pro přenos dat	Enhanced Data Rate (EDR)	AMP (Alternative MAC/PHY) — EDR pro přenášení malých objemů dat a IEEE 802.11 pro přenášení velkých objemů dat.
Unicast Connectionless dat – technologie pro přenos dat mezi dvěma zařízeními, bez vytvoření logického kanálu (L2CAP). Výrazně urychluje připojení a přenos malých objemů dat.	ne	ano
Bezpečnost.	AES-128	Stejná technologie (AES-128). Další vlastnosti – pozastavení (pauza) šifrování, periodická (každých 23,3 hodiny) změna šifrovacího klíče pro zařízení připojené dlouhou dobu.

Tabulka 2.3: Porovnání Bluetooth 2.0 a Bluetooth 3.0.

Všechny Bluetooth API jsou k dispozici v knihovně „android.bluetooth“. S Android Bluetooth API lze splnit čtyři hlavní úkoly pro komunikaci pomocí bluetooth: změna nastavení bluetooth adaptéru, hledání zařízení, která jsou buď párová, nebo dostupná v dané oblasti, připojování zařízení a přenos dat mezi zařízeními.

Třída	Popis činnosti
BluetoothAdapter	Struktura této třídy zahrnuje metody pro vyhledání dostupných zařízení, žádosti o seznam připojených zařízení, vytvoření instance třídy BluetoothDevice na základě známé MAC adresy a vytvoření BluetoothServerSocket pro čekání na požadavek na spojení z jiných přístrojů.

BluetoothDevice	Třída je spojena se vzdáleným zařízením Bluetooth. Instance této třídy se používá pro připojení přes BluetoothSocket nebo pro vyžádání informací o vzdáleném zařízení (jméno, adresa, třída, stát).
BluetoothServerSocket	Je to otevřený soket serveru, připravený ke zpracování přichozího požadavku. Aby bylo možné připojit dvě zařízení se systémem Android, jeden z nich by měl otevřít soket pomocí této třídy. Když vzdálené zařízení odešle požadavek na připojení, BluetoothServerSocket vrátí instanci BluetoothSocket.
BluetoothSocket	Rozhraní pro Bluetooth, podobně jako TCP soket. Je to bod spojení, který umožňuje komunikaci se vzdáleným zařízením pomocí InputStream a OutputStream.

Tabulka 2.4: Základní příkazy pro práci s Bluetooth [2].

2.1.2 Bluetooth Low Energy

Bluetooth low energy (Bluetooth Smart, nebo Bluetooth 4.0) používá méně energie než klasický Bluetooth. Technologie Bluetooth s nízkou spotřebou energie poskytuje dlouhodobé spojení malých zařízení, jako jsou senzory, a mobilních zařízení v rámci osobní sítě (PAN). Bluetooth moduly jsou dostupné ve dvou implementacích: single-mode (jednotlivý režim fungování) a dual-mode (duální režim fungování).

Single-mode čip podporuje komunikaci jen přes Bluetooth low energy (BLE), což zajišťuje nízkou spotřebu a kompaktní rozměry zařízení. Používá se hlavně na koncových zařízeních, jako jsou senzory, hodinky, klíčenky apod [36].

V čipu s duálním režimem fungování je BLE čip integrován do existujícího klasického Bluetooth čipu. Umožňuje používat všechny výhody různých verzí Bluetooth, jako je podpora velkého množství zařízení pracujících s technologií Bluetooth 2.0, nejvyšší rychlost přenosu v Bluetooth 3.0 a komunikace se senzory pomocí Bluetooth 4.0. Používají se především v zařízeních, která vykonávají funkce „master“, jako je telefon, tablet apod.

Technická specifikace	Klasický Bluetooth	Bluetooth Low Energy
Radio frekvence	2,4 GHz	2,4 GHz
Vzdálenost	100m	100m

Rychlost přenosu dat	1–3 Mbit/s	1 Mbit/s
Řízené jednotky (max.)	7	7
Čekací doba (z nespojeného stavu)	100 ms	6 ms
Topologie sítě	Scatternet	Star-bus
Spotřeba energie	1W	0,01–0,5W
Maximální odběr proudu	<30 mA	<15 mA

Tabulka 2.5: Porovnání technologií klasický Bluetooth a BLE [1].

Podpora Bluetooth Low Energy se objevila v Android 4.3 (API level 18). To umožňuje Android aplikaci komunikovat s BLE zařízení, jako jsou senzory, monitory tepové frekvence, fitness zařízení a tak dále.

Stejně jako v klasické verzi Bluetooth i v BLE při navázání spojení jedno zařízení vystupuje jako řídicí (master) a druhé jako podřízený (slave). „Master“ může mít mnoho připojených „slave“, ale „slave“ zařízení může mít jen JEDNO spojení – k „master“. V tom je rozdíl mezi BLE a klasickým Bluetooth.

Aby šetřil energii, „slave“ je obvykle v režimu „spánku“ a probouzí se pravidelně ke kontrole datových paketů z „masteru“. „Master“ zadává časové úseky, ve kterých se „slave“ probouzí, aby naslouchal na kanálu, což je časové rozdělení přístupu (TDMA – Time Division Multiple Access). „Master“ taky zadává schéma FHSS (Frequency-hopping Spread Spectrum) [2].

Činnost	Třída	Komentář
změna nastavení	BluetoothAdapter	Připojení k adaptéru, zapnutí/vypnutí adaptéru.
hledání zařízení	BluetoothAdapter.LeScanCallback	Nelze skenovat Bluetooth LE a klasické Bluetooth zařízení zároveň.
připojování	BluetoothGatt.connectGatt()	Nejprve musíme připojit k serveru GATT (peripheral/slave), aplikace bude fungovat jako klient GATT.
	BluetoothGattCallback	Je využívána k doručování výsledků klientovi, jako je například stav připojení, stejně jako jakýchkoli dalších klientských operací GATT.
přenos dat	BluetoothGattCharacteristic.getValue	Čte hodnotu Characteristic.

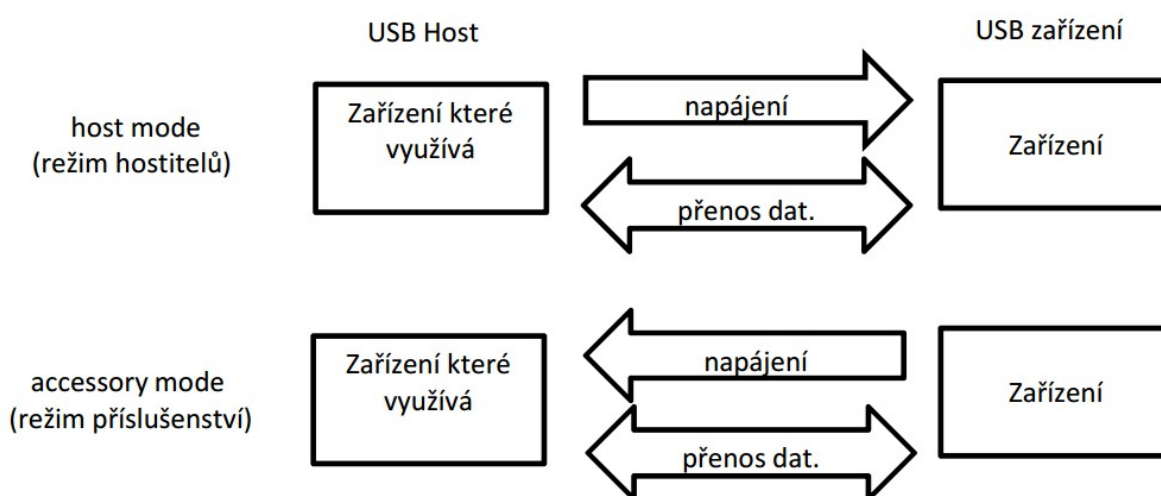
	BluetoothGattCharacteristic.setValue	Zapisuje hodnot Characteristic.
-	close()	Zavře klienta Bluetooth GATT.

Tabulka 2.6: Základní příkazy pro práci s Bluetooth [2].

2.2 USB

Operační systém Android umožňuje připojit přes USB velké množství různých periferních zařízení, které mohou zahrnovat bezdrátové moduly. Z tohoto důvodu zde bude popsán také tento princip komunikace.

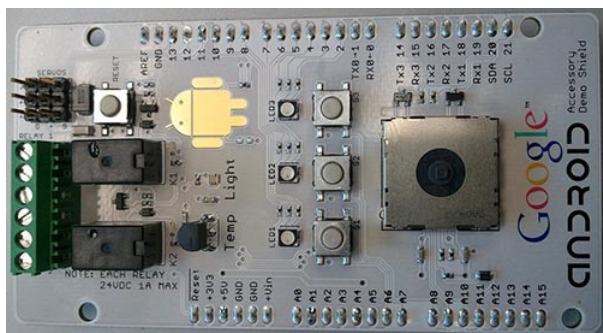
OS Android podporuje komunikaci po sběrnici USB ve dvou režimech: první – „accessory mode“ (režim příslušenství), druhý – „host mode“ (režim hostitelů).



Obrázek 2.1: Způsoby komunikaci po sběrnici USB.

Zařízení připojené v režimu příslušenství USB slouží jako USB hostitel. V tomto režimu dává napájení USB zařízení a musí dodržovat Android Open Accessory protokol popsany v dokumentaci Android Development Kit. Tento protokol otevírá mnoho možností pro vývojáře a radioamatéry: ovládání servo motorů, čtení stavu senzorů na Android telefonech nebo tabletech, jako je gyroskop, akcelerometr, kompas, GPS, přenos dat přes GPRS/HDSPA moduly a další.

Existují dvě oficiální verze zařízení: Google ADK 2011 Board a Google ADK 2012 Board.

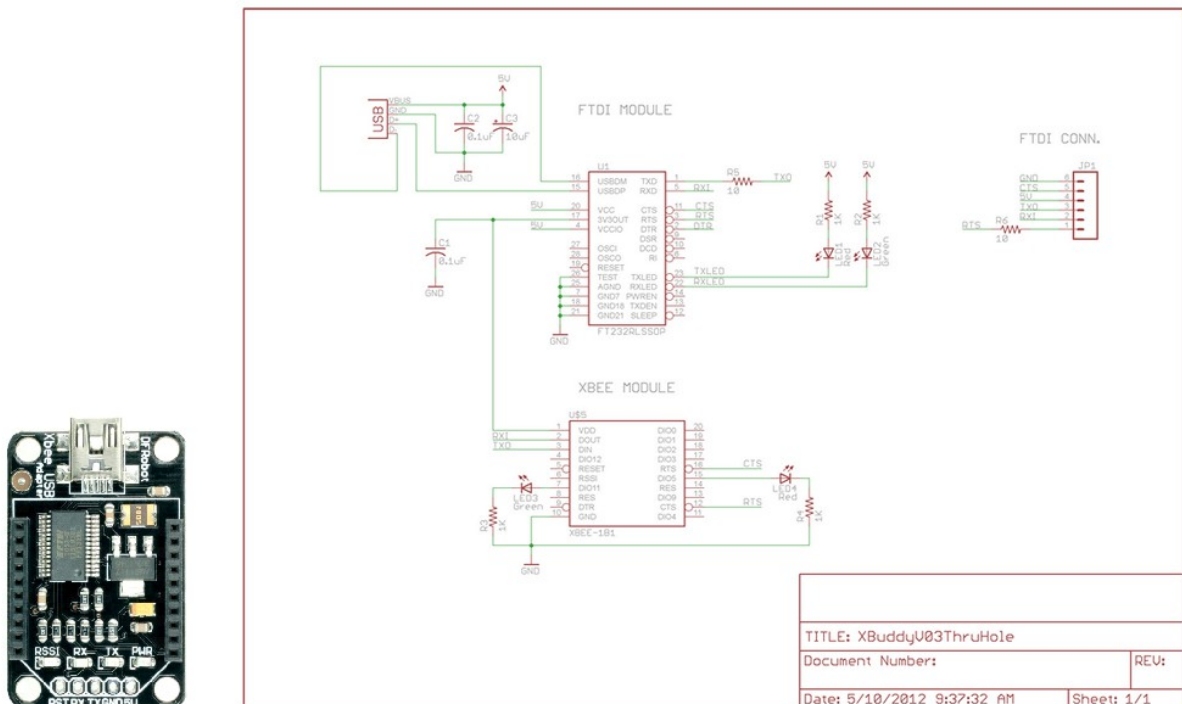


Obrázek 2.2: Google ADK Board 2011 (vlevo) a 2012 (vpravo).

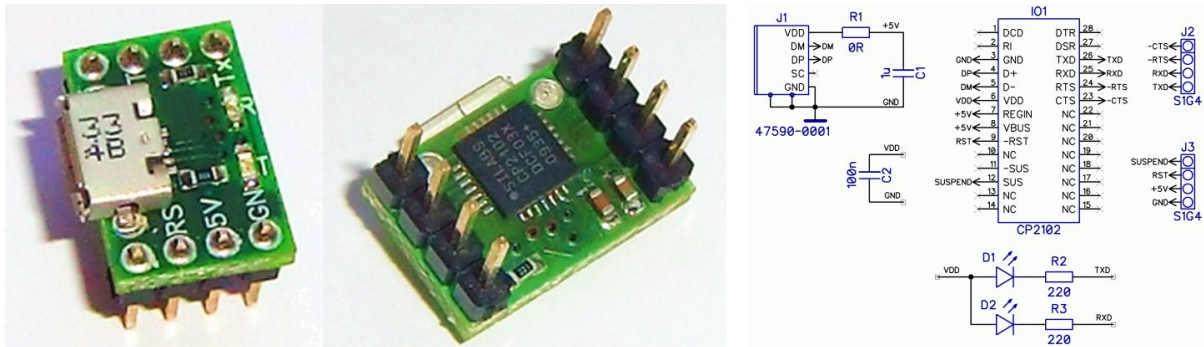
Google ADK 2011 Board: ATmega2560 (54 digitálních vstupů/výstupů (z nichž 14 může pracovat jako PWM výstupy) a 16 analogových vstupů (10 bit ADC)), 3 tlačítka, 3 RGB LED, joystick, senzor teploty, fotodioda, kapacitní snímač (android logo), 2 relé 24V/1A, tři konektory pro servomotory.

Google ADK 2012 Board: ARM 32-bit Cortex M3 mikroprocesor, externí SD slot, Bluetooth a NFC, akcelerometr, snímač atmosférického tlaku, vlhkosti a přiblížení, kapacitní posuvník, 6 sedmi-segmentových LED RGB matic, nízkofrekvenční zesilovač a reproduktor.

V režimu hostitelů lze připojit mnoho různých standardních zařízení, jako klávesnice, myš nebo USB flash disk. Napájení ve sběrnici zajišťuje Android zařízení. Velký zájem představuje komunikace mezi Android zařízeními a dalšími zařízeními pomocí virtuálních COM portů. Existuje velké množství čipů, které vykonávají tuto funkci, ale nejčastěji používané jsou tři – Prolific PL-2303, FTDI FT-232RL, Silicon Labs CP210x. Z vlastní praxe vím, že nejspolehlivější je čip FTDI, který je také nejdražší. Na druhém místě je CP210x. Při přenosu velkého množství dat nebo po dlouhé době bez komunikace čip PL2303 často nereaguje na příkazy a potřebuje restartovat.



Obrázek 2.3: XB-Buddy Kit (FTDI) (vlevo) a schéma zapojení (vpravo) [5].



Obrázek 2.4: USB-to-UART(CP2102) (vlevo a uprostřed) a schéma zapojení (vpravo) [6].

S pomocí virtuálního COM portu lze připojit libovolný modul pro bezdrátovou komunikaci na sériové lince UART, např. Zigbee UART Wireless Module, UART Bluetooth Module, UART Serial GPS Module, UART Serial WIFI Wireless Module, GSM SIEMENS TC35 SMS Module nebo NFC Reader UART.

Pro rychlý a snadný vývoj lze použít knihovnu `usb-serial-for-android` [7] v současné době podporující čip FT232R, CP2102. Nepotřebuje root přístup, ADK nebo speciální ovladače jádra; všechny ovladače jsou implementovány v jazyce Java. Pracuje jednoduše – `read()`, `write()` a pro další základní funkce používá vlastní protokoly. Další knihovna `usb-serial-for-android` [8] podporuje čip TI CC2531 ZigBee.

2.3 NFC

Technologie NFC (Near Field Communication) umožňuje zařízením spolu komunikovat na velmi krátkou vzdálenost – v řádu jednotek centimetrů. Podpora NFC se objevila v Android 2.3 (API level 9). NFC zařízení mohou být buď aktivní, nebo pasivní. V systému Android existuje možnost pracovat nejen s NFC ale i přenášet data (obrázky, video atd.) – Android Beam (od verze 4.0).

Metoda	Komentář
<code>NfcAdapter.ACTION_TAG_DISCOVERED</code>	Záměr zahájit činnost, pokud je TAG objeven.
<code>NfcAdapter.EXTRA_TAG</code>	Ukazatel nalezeného TAGu
<code>NfcAdapter.EXTRA_NDEF_MESSAGES</code>	Čtení
<code>writeNdefMessage</code>	Ukládání

Tabulka 2.6. Základní příkazy pro práci s NFC.

Výhodou NFC oproti jiným bezdrátovým technologiím je jednoduché párování a to, že pasivní zařízení nepotřebuje žádné napájení.

Nevýhody jsou malá přenosová vzdálenost a poměrně nízká přenosová rychlost (max. od 106 kbit/s do 424 kbit/s).

2.4 WIFI

WIFI je velmi populární bezdrátová technologie. Typicky WIFI síť obsahuje alespoň jeden přístupový bod (AP – Access Point) a alespoň jednoho klienta. Je také možné připojit dva klienty peer-to-peer ("Ad – hoc"), kdy jsou klienti připojeni přes síťové adaptéry „přímo“. Přístupový bod vysílá svůj identifikátor sítě (SSID) s použitím speciálních signalizačních paketů při rychlosti 0,1 Mbit/s za 100 ms. Proto je 0,1 Mbit/s nejnižší rychlostí přenosu dat pro Wi-Fi připojení. Když klient zná SSID sítě, může zjistit, zda je možné se k tomuto přístupovému bodu připojit.

Bezdrátová komunikace přes rozhraní WIFI je k dispozici od verze Android 1.0 (API level 1). WIFI hotspot (možnost vytvořit přístupový bod na zařízení s operačním systémem Android) je k dispozici od verze Android 2.2–2.2.3 (API level 8), WIFI Direct (Přímé spojení, jednoduchý přenos dat mezi zařízení, jako Bluetooth) – Android 4.0–4.0.2 (API level 14). Technologie Wireless display (Miracast) – bezdrátový přenos obrazu, jež funguje jako obyčejný HDMI kabel, ale vzduchem – se objevila v Android 4.2 (API level 17).

Výhody:

- Vysokorychlostní přenos dat.
- Mobilita.
- Možnost připojení více uživatelů současně.
- Záření Wi-Fi při přenosu dat ~10 krát méně než při přenosu dat přes GSM/UMTS .

Nevýhody:

- V rozmezí od 2,4 GHz provozuje několik standardů (Bluetooth, ZigBee atd.).
- Vysoká spotřeba energie.
- V režimu "Ad – hoc" rychlost jenom 11 Mbit/s (802.11b). Šifrování WPA2 není k dispozici, jenom WEP (klíče lze najít pomocí útoku).

2.5 GMS/UMTS

Operační systém Android podporuje video hovory, hlasové volání, SMS, MMS a datový přenos přes mobilní síť 2G/3G (LTE zařízení se teprve začínají objevovat na trhu). Tyto technologie jsou podrobně popsány v různé dokumentaci; z tohoto důvodu v tomto článku nebudu popisovat základní principy mobilních sítí. Dále budou popsány pouze základní příkazy pro rychlý vývoj aplikací v systému Android pro tři klíčové funkce: volání, zprávy a datový přenos přes mobilní síť a silné (výhody) a slabé (nevýhody) stránky mobilního připojení.

Pro volání můžeme použít knihovnu android.intent. Intent – popis akce, kterou by měla udělat aplikace. Touto akcí může být přechod do jiného okna, odchozí volání, otevření seznamu kontaktů

atd. V Intent existují dva základní atributy: akční funkce, která má být provedena (je definována jako konstantní: ACTION_VIEW, ACTION_DIAL atd.) a datový argument této funkce psaný jako URI. Například kombinace ACTION_DIAL, "tel:123456" zobrazí okno volání s vytočenými číslicemi 123456. Funkce volání startActivity(intent) spustí Intent. Příkaz pro volání je startActivity(new Intent(Intent.ACTION_CALL, "tel:123456")).

Pro posláni zprávy, můžeme použít třídu android.telephony.gsm.SmsManager. Odesláni SMS – sendTextMessage(). Pro přijímání musíme použít android.content.BroadcastReceiver – aby zachytil přichozí SMS. Při příjmu SMS bude provedena metoda onReceive. SMS zpráva je zabalena a připojena k objektu intent (druhý parametr metody onReceive). SMS je uložena v objektu intent v PDU formátu. Pro rozbalení SMS zprávy musíme použít metodu createFromPdu() třídy SmsMessage.

Pro datový přenos přes mobilní síť v systému Android existuje mnoho možností, například můžeme použít POST požadavky na vzdálený server a přijímat informace z odpovědi ze serveru. Tuto metodu komunikace lze použít také při připojení k internetu přes WIFI. Pro odesláni žádosti musíme vytvořit objekt org.apache.http.client.HttpClient, org.apache.http.client.methods.HttpPost (ve kterém zadáme adresu URL, která bude přijímat data, a spojí ji s daty ve formátu proměnná:hodnota) a spustit dotaz (požadavek) httpClient.execute(httpPost). Pro získání dat potřebujeme zpracovat odpověď ze serveru, kterou dostaneme po dotazu (HttpResponse response=httpclient.execute(httpPost)), dále dostaneme objekt HttpEntity responseEntity=response.getEntity() a převedeme na řetězec responseString=EntityUtils.toString(responseEntity).

Výhody mobilních komunikací (2G/3G):

- Vysoká mobilita (velké pokrytí, má výhodu při velkých vzdálenostech nebo jako havarijní možnost komunikace).
- Schopnost přenášet hlas, video, zprávy.
- Zprávy dosáhnou destinace, i když je účastnická stanice vypnutá nebo mimo dosah sítě.

Nevýhody mobilní komunikace (2G/3G):

- Vysoká spotřeba energie.
- Omezená velikost zprávy (SMS, MMS).
- Některé moduly umí přenášet pouze data (tablety, 3G modemy).
- Cena.

3 Arduino

Arduino je velmi jednoduchý, ale velmi funkční kontrolér pro rychlé prototypování a realizaci nejrůznějších nápadů [3]. Tato mikrokontrolerová vývojová deska dává příležitost naučit se základy práce s mikroprocesory a realizovat své nápady v HW, často i bez pájení. Arduino získalo velkou popularitu díky jednoduchosti a otevřenosti. Základy práce s Arduino se můžete naučit za pár hodin. Je tomu tak díky tisícům publikací, učebnic, poznámek na internetu a velké sérii video tutoriálů na Arduino.



Obrázek 3.1. R2 – D2 ovládaní pomocí Arduino [10].

Arduino se skládá z mikro kontroléru Atmel AVR a prvků pro programování a integraci s dalšími systémy. Na mnohých deskách je lineární regulátor napětí +5V nebo +3.3V a křemenný krystalový oscilátor 16MHz nebo 8MHz. V mikro kontroléru je nahrán bootloader (kód, který se po spuštění postará o start základních funkcí mikro kontroléru), takže nepotřebuje externí programátor.

Vývojové prostředí Arduino (Arduino IDE) – je více-platformní aplikace napsaná v programovacím jazyce Java, která obsahuje editor kódu, kompilátor a modul přenosu firmware v mikro kontroléru.

Programy pro Arduino jsou napsány v jazyce velmi podobném C++, který je doplněný jednoduchými a jasnými funkcemi pro ovládání I/O kontaktů.

Malý příklad (UART přijmout a poslat text.):

```
byte byteRead;
void setup() {Serial.begin(9600);}
void loop() {
  if (Serial.available()) { /* zkontroluj, zda byla odeslána data z počítače:*/
    byteRead = Serial.read(); /* přečti nejnovější byte */
    Serial.write(byteRead); /* odešli zpátky načtenou hodnotu*/
  }
}
```

3.1 Bluetooth

Technologie Bluetooth je popsána v kapitole 2 této práce. Základní moduly připojené k Arduino pro komunikaci přes Bluetooth jsou "Bluetooth Serial" moduly. Komunikace s moduly jde přes RX a TX (UART). Konfigurace se provádí pomocí AT příkazů.

3.1.1 Klasický Bluetooth

Nejčastěji používané moduly klasického bluetooth jsou HC-03 ... HC-06. Moduly jsou založeny na čipu BC417 od Cambridge Silicon Radio, který poskytuje hardwarovou podporu Bluetooth 2.0+EDR (Enhanced Data Rate) a flash paměti ES29LV800DB – 70WGI od Excel Semiconductor 8 Mbit (1 MB), která obsahuje firmware a nastavení.

S původním čínským firmware může modul pracovat ve dvou režimech: jednoduchý „rádio prodlužovač UART“ a řízení s pomocí AT-příkazů. V prvním případě je vše jednoduché – zapínáme modul, připojujeme se k modulu z počítače nebo jiného zařízení, které je schopné ustavit COM-port pro Bluetooth, a odesíláme data na tento port. V druhém režimu, lze modul ovládat pomocí AT příkazů ve formě „AT+PŘÍKAZ“, například příkaz „AT+NAME?“ vytiskne název modulu.

Modul může pracovat v režimu „master“ nebo „slave“, podporuje přenosovou rychlost od 2400 do 1382400bps. Napájený je napětím 3,3V, spotřeba proudu je kolem 50mA.

Moduly HC-03/HC-05 jsou více funkční než HC-04/HC-06. Moduly HC-03/HC-05 mohou pracovat v režimu „master“ nebo „slave“ (režim je možné měnit pomocí AT-příkazu), HC-04/HC-06 modul může pracovat pouze v „master“ nebo „slave“ režimu (režim nelze měnit pomocí AT-příkazu). Moduly HC-03/HC-04 jsou používány v průmyslu (Industrial level) a moduly HC-05/HC-06 na civilní úrovni (Civil level). Základní příkazy modulu HC-05 se nachází v příloze D (tabulka D.1).

Ve většině případů moduly HC-03...HC-06 nepodporují přenos zvuku (omezení firmwaru). Pokud budeme potřebovat přenos hlasu anebo větší dosah, můžeme použít modul Bluegiga WT11. Tento modul se používá v ArduinoBT [13]. Výhody – má dobrou dokumentaci, vysoký výkon (1. třída – vzdálenost ~100m), vysokou kvalitu výroby. Nevýhodou je, že cena je asi 10 krát vyšší než u HC-05 (\$ 50 a \$ 5) [12].

3.1.2 Bluetooth Low Energy

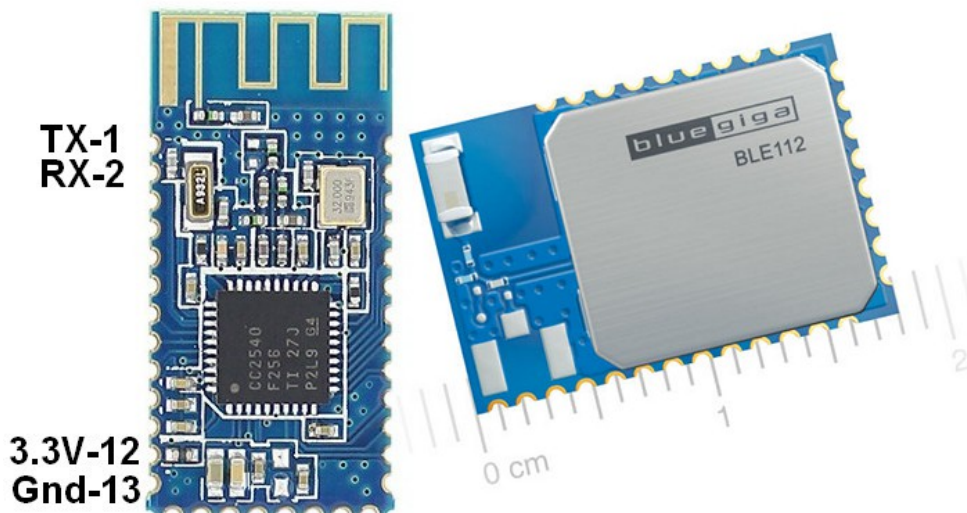
V současné době je většina modulů založena na čipu CC2540 (CC2541) Single-Chip BLE.

Parametr	Min.	Typ.	Max.	Jednotka
Provozní podmínky				
Frekvenční pásmo	2402	-	1280	MHz

Propustnost (kapacita)	-	1000	-	kBd
Napájecí napětí	2	-	3,6	V
Teplota prostředí	-40	-	+85	Celsia
Výstupní výkon	-20	-	+4	dBm
Přijem				
Citlivost přijímače	-	-93	-	dBm
Útlum v blízkosti kanálu, +1MHz	-	5	-	dB
Útlum v blízkosti kanálu, -1MHz	-	5	-	dB
Útlum v blízkosti kanálu, +2MHz	-	30	-	dB
Útlum v blízkosti kanálu, -2MHz	-	30	-	dB
Spotřeba proudu				
Spotřeba proudu, příjem	-	19,6	-	mA
Spotřeba proudu, vysílání, +5dBm	-	31,6	-	mA
Spotřeba proudu, vysílání, 0dBm	-	27	-	mA
Spotřeba proudu, režim "sleep"	-	0,4	-	mA

Tabulka 3.1. Základní charakteristiky čipu CC2540 [14].

Na trhu je velké množství modulů založených na tomto čipu, prakticky všechny jsou velmi malé velikosti. Nejznámější z nich je modul NM-10 (průměrná kvalita výroby, nízká kvalita podpory, 2 verze konfigurace modulu, nízká cena ~\$ 6) a Bluegiga BLE112 (dobrá kvalita výroby, mnoho vývojových nástrojů, stabilní podpora, různé možnosti konfigurace, cena ~ \$ 21) [13].



Obrázek 3.2: Modul HM-10 (levý) a modul Bluegiga BLE112 (pravý).

Stejně jako klasický bluetooth, BLE moduly jsou řízeny pomocí AT-příkazů. Základní příkazy modulu HM-10/HM-11 se nachází v příloze D (tabulka D.2).

Bluegiga Bluetooth Smart SDK podporuje modul Bluegiga BLE112. Tento SDK zahrnuje: Bluetooth Smart stack (with L2CAP), ATT, GATT, Security Manager a také:

BGAPITM binární protokol pro aplikace s odděleným hostem (MCU);

BGScript skriptovací jazyk pro samostatné aplikace (hodnoty ze snímače mohou být zpracovány bez použití externího mikro kontroléru);

Profil Toolkit pro vývoj profilů Bluetooth low energy.

Nejvíce zajímavý je BGScript a Profil Toolkit, protože umožňují výrazně rozšířit použití tohoto modulu. Nevýhodou je, že pro režim bezdrátového COM portu („rádio prodlužovač UART“) je třeba nahrávat FW speciálním programátorem (cena \$ 49) [16]. Proto se o tomto modulu nebudu rozepisovat do detailů.

3.2 ZigBee

Pro komunikaci protokolem ZigBee jsou často používány moduly XBee S1, S2 (firmy MaxStream nebo Digi International). Tyto moduly se používají v originálním Arduino Wireless SD Shieldu [17].



Obrázek 3.3: Moduly XBee (vlevo) i Wireless SD Shield (vpravo).

	XBee Series 1	XBee Series 2
Dosah uvnitř budovy	(30m)	(40m)
Dosah za přímé viditelnosti	(100m)	(120m)
Vysílací výkon	1 mW (0dbm)	2 mW (+3dbm)

RF Přenosová rychlost	250 Kbps	250 Kbps
Citlivost přijímače	-92dbm (1% PER)	-98dbm (1% PER)
Napájecí napětí	2.8 - 3,4 V	2.8 - 3,6 V
Vysílací proud (typický)	45 mA (@ 3.3 V)	40 mA (@ 3.3 V)
Idle/Přijem proud (typický)	50 mA (@ 3.3 V)	40 mA (@ 3.3 V)
„Sleep mode“ proud	10 uA	1 uA
Frekvence	ISM 2.4 GHz	ISM 2.4 GHz
Rozměry	0.0960" x 1.087"	0.0960" x 1.087"
Provozní teplota	od -40 do 85 C	od -40 do 85 C
Typy antén	PCB, Integrated Whip, U. FL, RPSMA	PCB, Integrated Whip, U. FL, RPSMA
Síťová topologie	Point to point, Star, Mesh (DigiMesh firmware)	Point to point, Star, Mesh
Počet kanálů	16 Direct Sequence Channels	16 Direct Sequence Channels
Možnosti filtrace	PAN ID, Channel & Source/Destination	PAN ID, Channel & Source/Destination

Tabulka 3.2: Rozdíly mezi moduly S1 a S2 [17].

Konfigurace modulů se provádí pomocí USB-to-XBee desky (nebo Arduino s prázdným sketch souborem) a programem X-CTU. USB-to-XBee jsou obyčejné USB-UART adaptéry, které jsou popsány v kapitole 1.3 tohoto dokumentu. X-CTU servisní program, pomocí kterého můžeme nahrát novou verzi firmware, je na internetových stránkách výrobce.

Veškerý firmware, který je k dispozici, může být v X-CTU klasifikován podle následujících kritérií:

Rodina: Zigbee , ZNet 2.5 a více jiných. Definiuje sadu dostupných možností. Rodina je důležitá k vytvoření komplexní sítě s konkrétními požadavky. Moduly XBee Series 2 kompatibilní s rodinou Zigbee jsou označeny jako XB24-ZB a ZNet 2.5 a XB24-B.

Role: Koordinátor, Router, EndDevice. Definiuje účel zařízení v síti. Jak již bylo uvedeno, v síti musí být jeden koordinátor a libovolný počet směrovačů a koncových zařízení.

Způsob, jak komunikovat s modulem: AT, API, PH, analogové IO, digitální IO. To určuje, jak se posílá příkaz modulu: je možné použít standardní AT-příkazy nebo můžeme použít svůj vlastní protokol (API). Tato metoda je důležitá, pokud potřebujeme změnit nastavení modulu za běhu: změnit svou vlastní adresu, cílovou adresu, režim spotřeby energie atd. AT-příkazy lze jednoduše používat, mohou být zaslány standardní metodou; API-příkazy jsou rychlejší, ale vyžadují vytvoření protokolu. Základní AT-příkazy moduly XBee S1, S2 se nachází v příloze D (tabulka D.3).

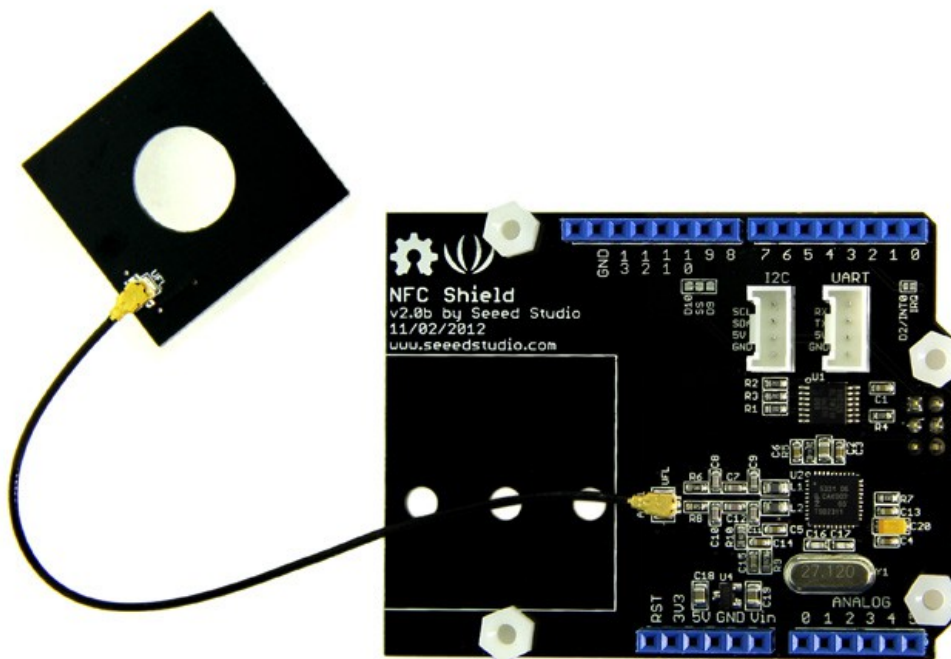
Hlavní výhoda modulů XBee S1, S2 – snadnost použití, hlavní nevýhoda – vysoká cena ~\$ 30-60.

Mnohé moduly ZigBee jsou založeny na čipu CC2530 (kittybee ZigBee shield for arduino). [20] Existují dvě verze ZigBee stacku pro ten čip – standardní TI Z-Stack [21] a ZBOSS Stack [22]. Tento čip obsahuje mikroprocesor 8051, který může být naprogramován pro jednoduché úkoly. Hlavní výhoda – univerzálnost a cena (modul – ~\$ 10), hlavní nevýhoda – složitost použití, pro programování potřebuje externí programátor (cena ~\$ 10).

K dispozici jsou také moduly na čipu ATmega128RFA1 (deska zigduino (Arduino + ATmega128RFA1)), pro tento čip existuje speciální knihovna pro Arduino – Zigduino Radio Arduino Library. Výhody – snadné použití, knihovna pro Arduino. Nevýhody – exkluzivnost a vysoká cena \$ 69,9 [23] [24].

3.3 NFC

Většina modulů je založena na čipu NXP Semiconductors PN532. Nejznámější jsou NFC Shield V2.0 (Seed Studio, stojí \$ 29,50) a Adafruit PN532 NFC/RFID Controller Shield PN532 shield (stojí \$ 39). V tomto článku se zabývám modulem NFC Shield V2.0 od Seed Studio, protože má nižší cenu a vyšší kvalitu dokumentace a podpory.



Obrázek 3.4: SeedStudio NFC Shield V2.0.

Klíčové vlastnosti:

- 5cm max. efektivní dosah,
- podpora rozhraní: I2C (výchozí), SPI, UART (HSU),
- slouží k bezkontaktní komunikaci na 13,56 MHz
- podpora P2P komunikace,
- podpora protokolu ISO14443 typu A a typu B.

Pro rychlý a snadný vývoj je možné použít knihovny NFC library for Arduino, které používá knihovnu NDEF library - tato knihovna pracuje s moduly Adafruit NFC Shield, Seeed Studio NFC Shield v2.0, Seeed Studio Xadow NFC a Elechouse PN532 NFC RFID [25] [26]. Možno komunikovat přes SPI (knihovna PN532_SPI), I2C (knihovna PN532_I2C), UART (knihovna PN532_HSU). S pomocí této knihovny můžeme: číst/zapisovat Mifare Classic Card, komunikovat s operačním systémem Android 4.0+ (Android Beam) a dělat emulace karty (NFC Type 4 tag).

Čtení zpráv z tagu	Psaní zpráv v tagu
<pre>if (nfc.tagPresent()) { NfcTag tag = nfc.read(); tag.print(); }</pre>	<pre>if (nfc.tagPresent()) { NdefMessage message = NdefMessage(); message.addTextRecord("Hello, Arduino!"); nfc.write(message); }</pre>

Tabulka 3.3: Příklad psaní a čtení textu.

3.4 WIFI

Pro připojení WI – FI rozhraní pro Arduino lze použít Arduino WiFi Shield (stojí € 69) nebo Arduino Yun (stojí € 69).

Klíčové vlastnosti Arduino WiFi Shield :

- Připojení k desce Arduino (např. UNO nebo Mega2560),
- Pracovní napájení 5V (napájeno z desky Arduino),
- Podporované formáty bezdrátového připojení: 802.11b/g,
- Formáty šifrování připojení: WEP a WPA2 Personal,
- Připojení k Arduino přes SPI sběrnici,
- Slot pro micro SD karty,
- ICSP konektor,
- Možnost připojení přes FTDI k desce Wi-Fi pro odladění,
- Mini-USB port pro upgrade firmwaru Wi-Fi desky.

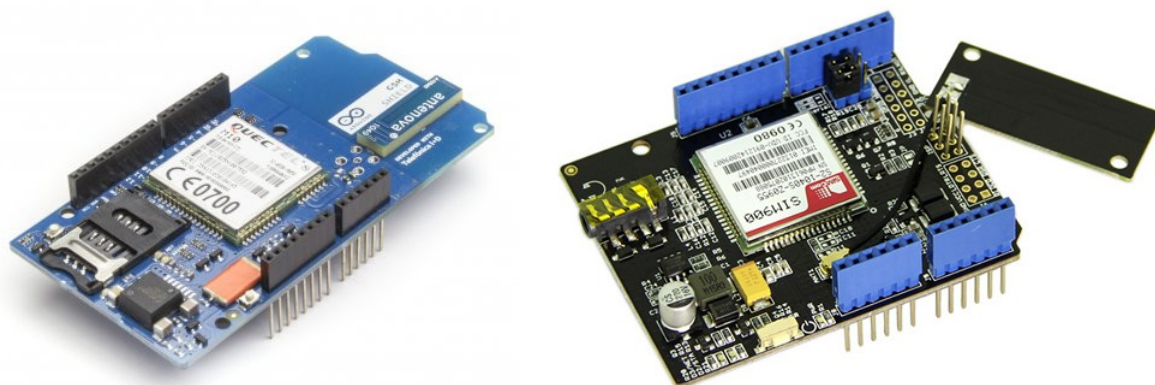
V Arduino Yun WIFI rozhraní postaveném na čipu Atheros AR9331 s operačním systémem na SoC Atheros AR9331 je OpenWrt distribuce Linino. 16 MB flash paměti obsahuje linuxový image a plný správce python balíčků, 64 MB DDR2 musí umožnit běh úkolů náročných na CPU, jako je například spuštění webového serveru nebo videa z webové kamery. Pokud je to nutné, 16 MB vestavěné paměti lze rozšířit pomocí SD karty nebo USB Flash.

Výhody Arduino Wi-Fi Shield a Arduino Yun: velká všestrannost, vysoce výkonné čipy, standardní knihovny. Nevýhody: vysoká spotřeba energie čipu modulů, vysoká cena.

Existuje mnoho jiných modulů. Optimální vlastnosti a přijatelnou cenu má modul HI-LINK HLK-RM04 (stojí \$ 14) [27].

3.5 GSM

Nejznámější GSM moduly – Arduino GSM Shield, jsou založeny na modemu M10 od Quectel (stojí € 69) a Seeed Studio GPRS Shield V2.0 (založeny na modemu SIMCom SIM900) (stojí \$ 59,90).



Obrázek 3.5: Arduino GSM Shield (integrována anténa) (vlevo) a Seeed Studio GPRS Shield V2.0 (vpravo).

Budu popisovat podrobněji modul Seeed Studio GPRS Shield V2.0, protože je levnější a bezpečnější (Arduino GSM Shield se může spontánně vznítit [29]). Na GPRS Shield V2.0 je postaveno mnoho projektů, zatímco modul Arduino GSM Shield má dobrou dokumentaci (což se dá překonat) a SIMCom vyrábí pin-kompatibilní moduly pro multi-band WCDMA/GSM/GPRS/EDGE (v blízké budoucnosti může být uvolněna nová verze modulu) .

Shield je založen na modulu SIMCOM SIM900, na němž se rovněž nachází: SIM-karta, standardní 3,5 mm jack pro audio vstup a výstup a konektor pro externí anténu.

Komunikace s deskou je přes sériový port s pomocí AT-příkazů. Úplný seznam příkazů je uveden v manuálu k SIM900 [30]. S pomocí jumperů můžeme nastavit kontakty pro komunikaci:

hardware pin0 a pin1, software serial-emulace pin7 a pin8. Přes tyto kanály pak budou předány AT-příkazy. Základní AT-příkazy modulu GPRS Shield V2.0 se nachází v příloze D (tabulka D.4).

Stačí připojit sluchátka a mikrofon skrze 3,5 mm jack, pak po obdržení výzvy od účastníka můžeme komunikovat stejným způsobem jako v obyčejném telefonu.

Ovládat GPRS zařízení přímo pomocí AT-příkazů není vždy vhodné. Tyto příkazy je třeba neustále mít v paměti. Některé funkce byly implementovány v SIM900_SIM908 Arduino's Shield Library. Knihovna podporuje schopnost posílat a přijímat hovory a SMS, dostávat o nich informace, kontrolovat, zda je nová zpráva. Všechny AT-komunikace jsou skryté v detailech implementace tříd a metod, které je možné použít [31].

Knihovna GSM je součástí Arduino IDE 1.0.4 a novějších.

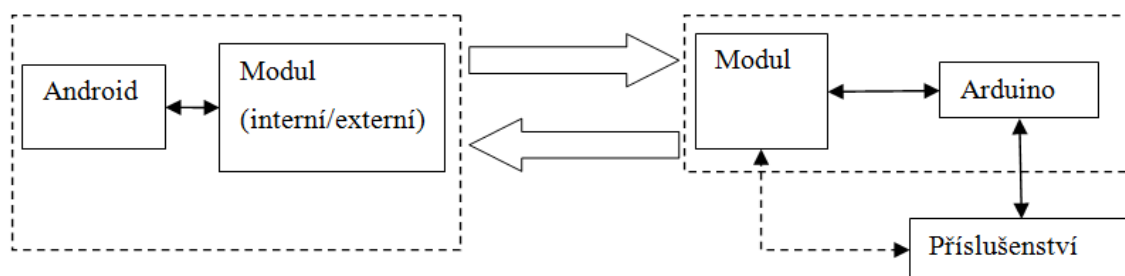
Třída	Popis
GSMVoiceCall class	Umožňuje hlasovou komunikaci přes modem. Musí být přidán pro plné využití mikrofonu a reproduktoru.
GSM_SMS class	Umožňuje odesílání a přijímání krátkých textových zpráv (SMS) zpráv.
GPRS class	Tato třída je zodpovědná za zahrnutí komunikace TCP.
GSMClient class	Třída vytvoření klientů, kteří se mohou připojit k serverům a odesílat a přijímat data.
GSMServer class	Třída vytvoření serverů, které mohou odesílat data a přijímat data z připojených klientů (programy běžící na jiných počítačích nebo zařízeních).
GSMModem class	Třída usnadňující diagnostickou komunikaci s modemem.
GSMScanner class	Třída poskytující diagnostické informace o síti.
GSMPIN class	Třída nástrojů pro komunikaci se SIM kartou.
GSMBand class	Třída poskytující informace o kmitočtovém pásmu. Obsahuje také metody pro stanovení typu sítě.

Tabulka 3.4: Základní třídy Arduino GSM knihovny [32].

4 Návrh knihovny funkcí pro OS Android

Na začátku této kapitoly popíšu schéma komunikačních zařízení z OS Android a Arduino.

Modul bezdrátové komunikace zařízení s OS Android může být součástí zařízení (čip je umístěný na základní desce zařízení) nebo externím modulem (připojuje se k zařízení přes port). Modul bezdrátové komunikace v Arduino může být také součástí zařízení (čip/modul je umístěný na základní desce) nebo externím modulem (připojuje se k desce), ale na rozdíl od OS Android můžeme pro práci s externím nebo interním modulem používat stejnou knihovnu. Například v OS Android pro interní BLE modul potřebujeme používat standardní knihovnu Bluetooth; pokud modul připojíme přes USB OTG – musíme použít jinou knihovnu, protože komunikace s modulem bude probíhat přes virtuální UART port.



Obrázek 4.1: Blokové schéma komunikace zařízení s OS Android a Arduino.

Nejvhodnější metody komunikace s ohledem na spotřebu energie jsou Bluetooth low energy nebo ZigBee. Podpora Bluetooth Low Energy se objevila v Android 4.3, tato verze je zcela nová (červenec 2013). Na trhu je zatím velmi málo zařízení, OS podporuje pouze čipy Broadcom. ZigBee standardní knihovna neexistuje.

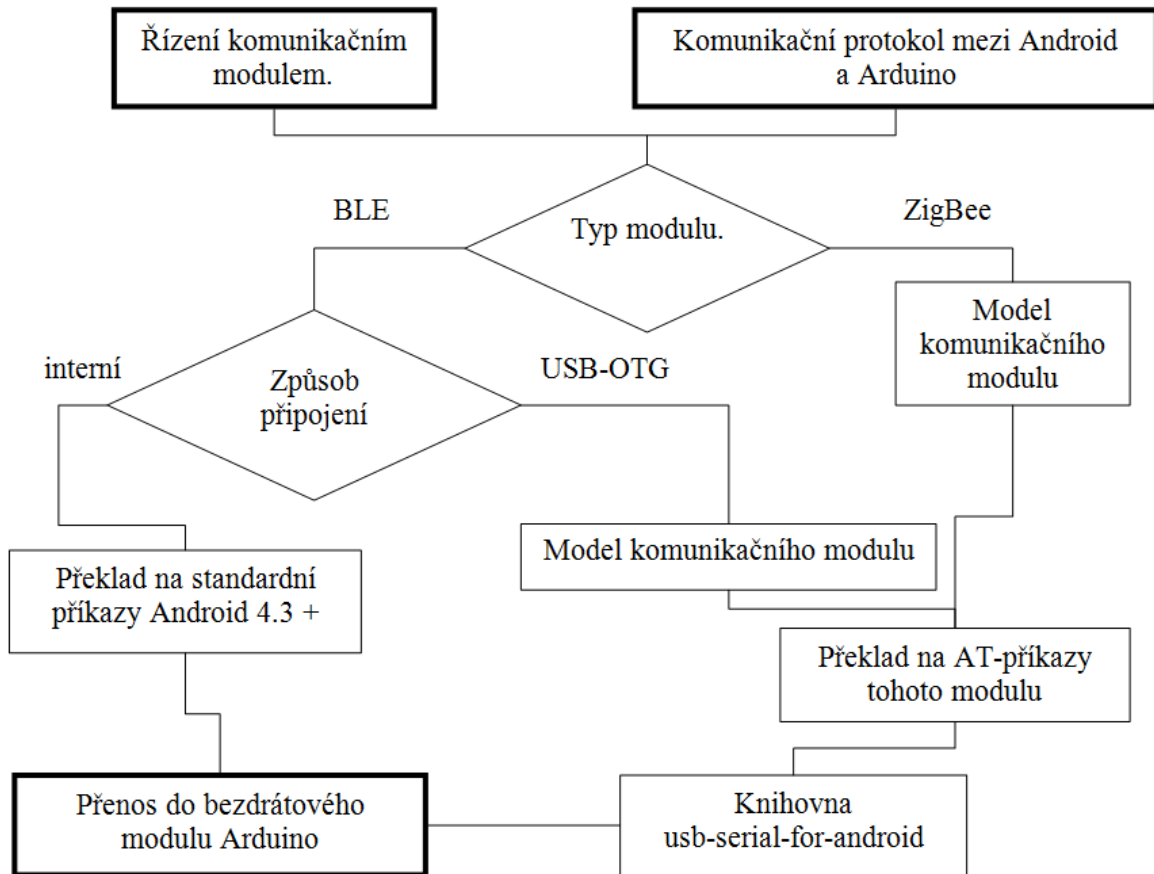
Z tohoto důvodu by knihovna měla plnit následující funkce:

1. Podpora základních modulů UART-to-{ Bluetooth low energy, ZigBee} (změna nastavení modulu, odesílání/přijímání a zpracování dat).
2. Pokud má telefon interní modul Bluetooth low energy – překládání příkazů knihovny na standardní příkazy Android.
3. Implementace protokolu Android-to-Arduino

Klíčové vlastnosti:

- Jednoduché a intuitivní příkazy.
- Základní příkazy pro práci s populárními moduly:
 - ◆ senzory (teplota, vlhkost, vzdálenost, pohyb atd.),

- ◆ motory (servo, běžné motory, atd.),
- ◆ bezdrátové moduly (dálkové ovládání energie, rychlost, atd.).
- Obyčejná, cyklická nebo plánovaná komunikace s Arduino (např. pro kontrolu teploty, atd.).
- Zpracování vstupních dat (pokud bude Arduino přenášet data ze senzoru bez zpracování)
- V případě teplotního senzoru – konverze teploty ve stupních Celsia na Kelviny.



Obrázek 4.2: Blokové schéma knihovny.

5 Implementace protokolu pro komunikace OS Android a Arduino

Při implementaci protokolu byly hlavní priority univerzálnost, snadnost použití příkazů a rychlé rozšíření knihovny jinými uživateli.

Při implementaci protokolu bylo rozhodnuto nedodávat AT příkazy, protože jsou jen pro počáteční konfiguraci modulu, což lze jednoduše udělat s pomocí software od výrobce. Protokol se skládá z čtyř částí (jedna v Arduino a tři v OS Android). Tato struktura je vybraná z důvodu možnosti rychlé změny parametrů protokolu:

- knihovna pro platformu Arduino,
- modul grafického rozhraní (knihovna pro OS Android).
- modul pro komunikaci přes BLE (Android),
- modul pro komunikaci přes UART (Android).

Arduino je univerzální platforma z různých modulů, od jednoduchých senzorů a efektorů na práci s paměťovými čipy po protokoly I2C nebo ISP. Aby se zachovala jednoduchost, bude Arduino používat interní funkce pro práci s jednoduchými senzory, ale pro práci s komplexními senzory bude používat externí knihovny. To zajistí snadnou výměnu kódu knihovny pro další senzory a efekty.

Stále více a více moderních zařízení podporuje BLE. To dělá využívání modulu HM-10 velmi efektivním, protože pro komunikaci potřebujeme jen jeden modul HM-10 a Android zařízení s podporou BLE. Zároveň se při komunikaci dvou platformami Arduino velmi často se používá modul DIGI S1.

Pro připojování k zařízení s Android OS musí být použit OTG kabel a UART adaptér, který umožňuje připojit jakýkoliv modul s UART rozhraním do zařízení se systémem Android.

5.1 Seznam příkazů pro platformu Arduino

Nejdůležitější částí protokolu jsou příkazy pro práci ze vstupů/výstupů platformy Arduino.

V protokolu se používá symbol začátku věty na výstupu příkazu a symbol konce věty. Seznam specifických symbolů:

- '#' - začátek věty, která bude odeslána přes bezdrátovou komunikaci; věta pro digitální vstup/výstup (lze definovat jako „digital_separator“).
- '!' - začátek věty, která bude odeslána přes bezdrátovou komunikaci; věta pro analogový vstup/výstup (lze definovat jako „analog_separator“).
- '@' - začátek věty, která bude odeslána přes bezdrátovou komunikaci; informační věta (lze definovat jako „error_separator“).

- ';' - konec libovolné věty výstupu příkazu nebo libovolného příkazu, který dostává Arduino přes bezdrátovou komunikaci (lze definovat jako „endAnsw_separator“).

Dále budou uvedeny základní příkazy, jejich parametry vstupu a výstupu a teoretické možnosti použití. Seznam servisních příkazů:

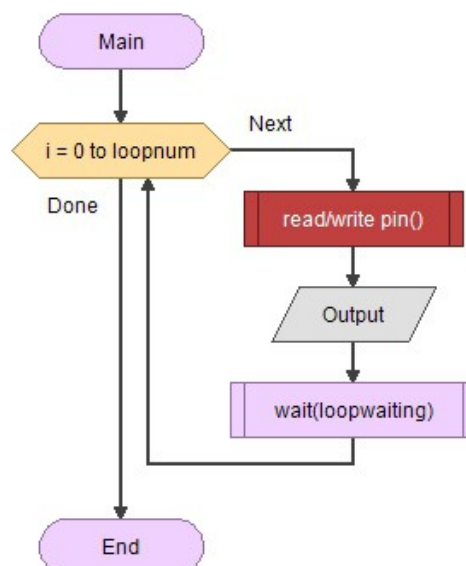
1. Příkaz „PING“.

- k čemu slouží: ověření funkčnosti platformy Arduino,
- vstup: žádný,
- výstup: „@READY“;
- možnost použití: při zahájení komunikace aplikace na OS Android; může zjistit, že platforma Arduino přijímá příkazy a taky pro ověření, že se platforma Arduino nezasekla.

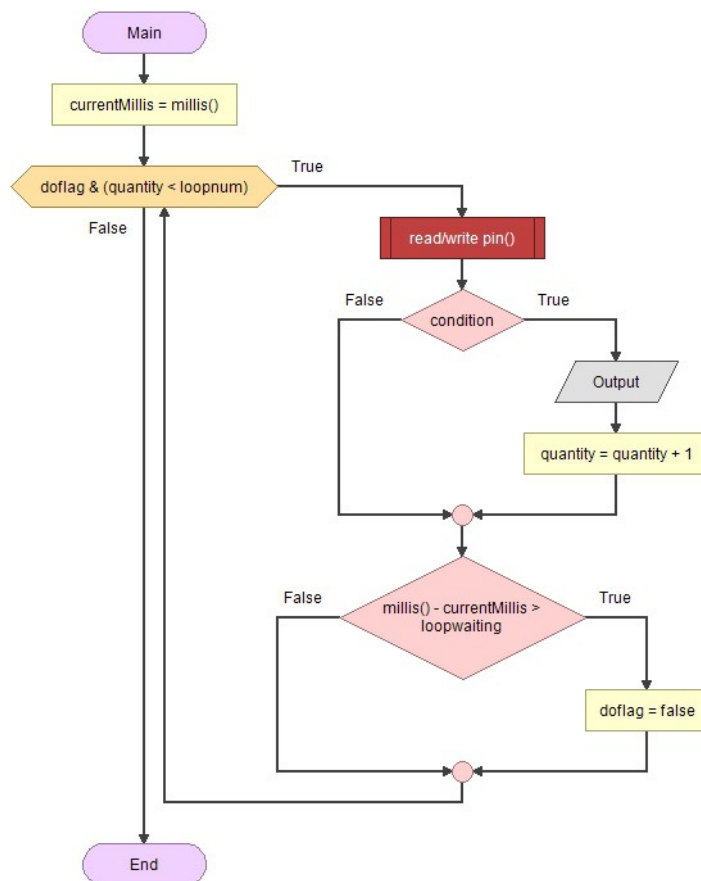
2. Příkaz „PARAM“.

- k čemu slouží: nastavení počtu cyklů (v příkazech bez delay – maximální počet vzorků) a délky čekání mezi cykly (v příkazech bez delay – maximální čas čekání),
- vstup: 1 argument – počet cyklů, typu unsigned int (1..65 535), 2 argument – délka čekání v ms, typu unsigned long (1..4 294 967 295),
- výstup: „@L=“ + počet cyklů + „;“ + „@W=“ + délka čekání + „;“;
- možnost použití: hlavně se používá k vytvoření periodických činností.

Tento příkaz je velmi důležitý, protože s parametry „loop_num“ (počet cyklů/maximální počet vzorků) a „loop_waiting“ (délka čekání/maximální čas čekání) pracuje většina příkazů pro práci s vstupy/výstupy Arduino. Na obrázku 5.1 je uveden algoritmus příkazu pro práci s vstupy/výstupy Arduino s funkcí „delay“. Na obrázku 5.2 je zobrazen algoritmus příkazu bez funkce „delay“.



Obrázek 5.1: Příkaz s funkcí „delay“.



Obrázek 5.2: Příkaz bez funkce „delay“.

Seznam příkazů pro práci s vstupy/výstupy Arduino:

1. Příkaz „RD_AN“.

- k čemu slouží: čtení analogového vstupu platformy Arduino (používá se funkce „delay“ piny A0...A7),
- vstup: 1. argument – číslo pinu, typu int(0..7), 2. argument – formát výstupní hodnoty typu String („V“, any_other),
- výstup: “!” + číslo pinu + kdy 1. argument se rovná „V“, hodnota bude ve voltech, typu String (0.00v...5.00v) (pro Arduino s napětím 3,3V je potřeba změnit hodnotu „arduino_volt“), jinak bude typu String (0..1023) + „;“,
- možnost použití: libovolná čidla s analogovým výstupem, například flame sensor*, gas (ozone) sensor, photo resistor, potentiometer, hall effect sensor (analog out), microphone sound detection sensor (analog out) atd. (*dále jsou jména senzorů uvedena v angličtině, poněvadž při testování bude použit velký „Sensor KIT“, který má jména senzorů jen anglicky).

2. Příkaz „RD_DG“.

- k čemu slouží: čtení digitálního vstupu platformy Arduino (používá se funkce „delay“, piny D2...D13),
 - vstup: 1. argument – číslo pinu, typu int (2..13),
 - výstup: „#“ + číslo pinu + String („HI“, „LO“) + „;“. (lze definovat jako „HI“ a „LO“) + „;“,
 - možnost použití: libovolná čidla s výstupem 1 nebo 0, například: PIR motion sensor, reed sensor, mercury sensor, hall effect sensor (digital out), microphone sound detection sensor (digital out), IR distance sensor, IR Line Tracking sensor, IR photo interrupter sensor apod.
3. Příkaz „WR_DG“.
- k čemu slouží: zápis hodnoty do digitálního výstupu platformy Arduino (píše jen jedenkrát piny D2...D13),
 - vstup: 1. argument – číslo pinu, typu int(2..13), 2. argument – hodnota pro zápis, typu String („ON“, „OFF“), (lze definovat jako „ON“ a „OFF“),
 - výstup: žádný,
 - možnost použití: lze řídit libovolným aktuátorem se vstupem 1 nebo 0, například: relay, led, laser a atd.
4. Příkaz „WRPWM“.
- k čemu slouží: zápis PWM hodnoty do digitálního výstupu platformy Arduino (píše jen jedenkrát, piny D3, D5, D6, D9, D10, D11),
 - vstup: 1. argument – číslo pinu, int(3,5,6,9,10,11), 2. argument – hodnota pro zápis, typu int (0..255),
 - výstup: žádný,
 - možnost použití: lze řídit libovolným aktuátorem, který podporuje PWM se vstupem 1 nebo 0, například: led, motor a atd.
5. Příkaz „JOYST“.
- k čemu slouží: čtení hodnot z výstupu „PS2 joystick sensor“. Slouží jako ukázka čtení z mnoha pinů za jeden cyklus (používá se funkce „delay“, piny A1(x), A2(y), D4(button)),
 - vstup: žádný,
 - výstup: „#02“ + x_pin(0..1024) + „;“ + y_pin(0..1024) + „;“ + b_pin(„HI“, „LO“) + „;“,
 - možnost použití: rozhraní člověk-zařízení apod.
6. Příkaz „LED_3“.
- k čemu slouží: zápis 3. PWM hodnoty; slouží jako ukázka zápisu mnoho pinů (píše jen jedenkrát, piny D3(led_1), D6(led_2), D11(led_3)),

- vstup: 1 argument – hodnota pro zápis “led_1”, typu int (0..255), 2 argument – hodnota pro zápis “led_2”, typu int (0..255), 3 argument – hodnota pro zápis “led_3”, typu int (0..255),
 - výstup: žádný,
 - možnost použití: ovládání RGB led, RGB led Strip (RGB led pásek) apod.
7. Příkaz „IRRCV“.
- k čemu slouží: čtení a dekodování signálu z IR dálkového ovladače (nepoužívá se funkce „delay“, piny D2..D13),
 - vstup: 1. argument – číslo pinu, typu int(2..13),
 - výstup: “#“ + číslo pinu + protokol kódování (N – NEC, S – SONY, 5 – RC5, 6 – RC6, U – Unknown, C – Other protocol) + „ “ + délka kódu v bitech + „ “ + hodnota v HEX formátu + „,“,
 - možnost použití: ovládání stroje pomocí IR signálu a taky přijímání a dekodování IR signálu.
8. Příkaz „SWTCH“.
- k čemu slouží: čtení digitálního vstupu s funkcí „Debounce“ (ošetření zákmitů), se zachováním vnitřního stavu (nepoužívá se funkce „delay“, piny D2..D13),
 - vstup: 1. argument – číslo pinu, typu int(2..13), 2. argument – čas zdržení pro funkci „Debounce“, typu unsigned long (0..4 294 967 295),
 - výstup: “#“ + číslo pinu + vnitřní stav, typu String(“|”, “0”) (stav se mění, když se změní hodnota pinu z HIGH na LOW) + „,“,
 - možnost použití: libovolná čidla s výstupem 1 nebo 0, jejichž stav se rychle mění, například vibration sensor, knock sensor (čidlo klepání) apod.
9. Příkaz „BUTTN“.
- k čemu slouží: čtení digitálního vstupu s funkcí „Debounce“ (ošetření zákmitů), reaguje jen na změnu hodnoty pinu (nepoužívá se funkce „delay“, piny D2..D13),
 - vstup: 1. argument – číslo pinu, typu int(2..13), 2. argument – čas zdržení pro funkci „Debounce“, typu unsigned long (0..4 294 967 295),
 - výstup: “#“ + číslo pinu + hodnota pinu, typu String(“HI”, “LO”) + „,“,
 - možnost použití: libovolná čidla s výstupem 1 nebo 0, například button (tlačítko) apod.
10. Příkaz „DHTSW“.
- k čemu slouží: čtení hodnoty z „DHT-11 single-wire humidity and temperature sensor”. Slouží jako ukázka práce se „single-wire” sběrnici (používá se funkce „delay“, piny D2..D13),
 - vstup: 1. argument – číslo pinu, typu int(2..13),

- výstup: “#“ + číslo pinu + hodnota vlhkosti, typu int (0..100) + „%,“ + hodnota teploty, typu int (0..100) + „'C;“,
- možnost použití: měření vlhkosti a teploty.

11. Příkaz „TMPOW“.

- k čemu slouží: čtení hodnoty z „DS18B20 1-Wire temperature sensor“. Slouží jako ukázka práce s „1-Wire“ sběrnici (nepoužívá se funkce „delay“, piny D2..D13),
- vstup: 1. argument – číslo pinu, typu int(2..13), 2. argument – jednotka teploty, typu String (“F”, any_other). Pokud se 2. argument rovná „F“, potom bude hodnota ve stupních Fahrenheita, jinak ve stupních Celsia,
- výstup: “#“ + číslo pinu + hodnota teploty, typu int (-55..257) + („'C;“, „'F;“),
- možnost použití: měření teploty.

12. Příkaz „I2CLS“.

- k čemu slouží: čtení hodnoty z „Si1145 UV/IR/Visible Light sensor“. Slouží jako ukázka práce s „I2C“ sběrnici (nepoužívá se funkce „delay“, piny A4(sda), A5(scl)),
- vstup: žádný,
- výstup: „!04“ + „VS:“ + hodnota viditelného světla, typu int (0..1023) + „, IR:“ + hodnota infračerveného záření, typu int (0..1023) + „, UV:“ + UV index, typu float (0.00..11.00) + „,“,
- možnost použití: měření viditelného světla, IR záření a UV indexu.

13. Příkaz „TMPAN“.

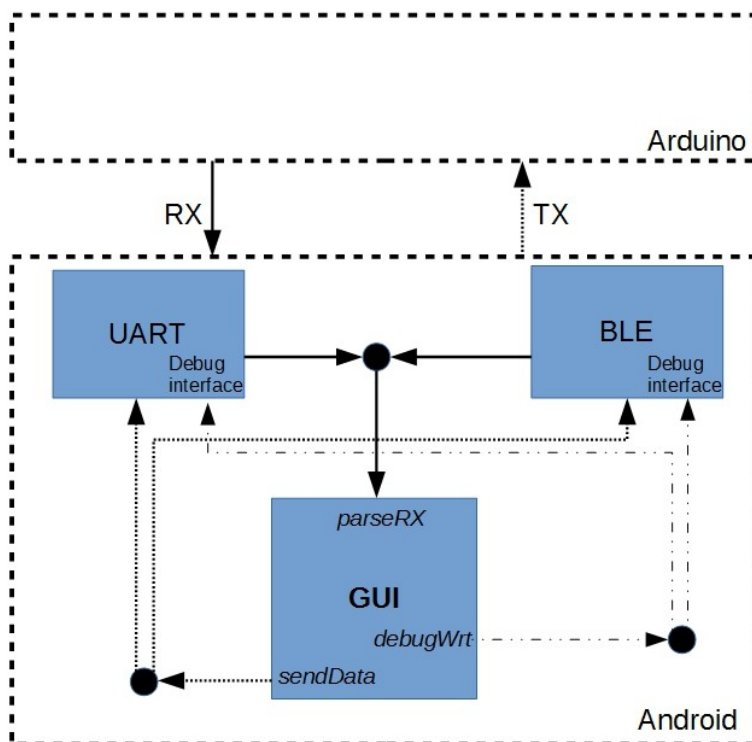
- k čemu slouží: měření teploty s pomocí termistoru a čtení z analogového vstupu platformy Arduino (používá se funkce „delay“, piny A0..A7),
- vstup: 1. argument – číslo pinu, typu int(0..7), 2. argument – jednotka teploty, typu String (“F”, any_other). Pokud se 2. argument rovná „F“, potom bude hodnota ve stupních Fahrenheita, jinak ve stupních Celsia,
- výstup: “#“ + číslo pinu + hodnota teploty, typu int (-55..257) + („'C;“, „'F;“),
- možnost použití: měření teploty.

5.2 Struktura Android aplikace

Aplikace se skládá ze tří částí nebo modulů. První modul je grafické uživatelské rozhraní, které bude odpovědné za komunikaci s uživatelem. Ostatní dva moduly budou zodpovědné za komunikaci s Arduino.

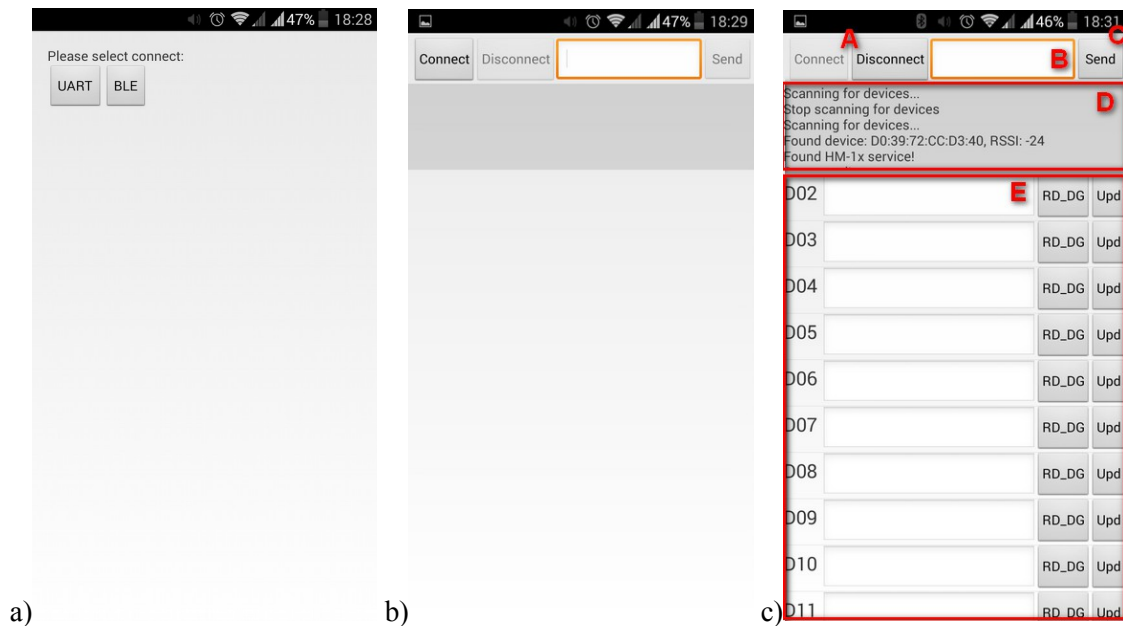
Metoda zpracování řetězců dat se nachází v modulu grafického uživatelského rozhraní, soubor „BoxAdapter.java“ (dále jen modul GUI). Když dostaneme libovolná data od Arduino v modulu

komunikace přes UART (soubor „UARTActivity.java“, dále jen modul UART) nebo v modulu komunikace přes Bluetooth low energy (soubor „BLEActivity.java“, dále jen modul BLE), tak provedeme drobné zpracování - příchozí zprávy jsou zapisovány do bufferu a rozdělují se symbolem konci zprávy. Po zpracování zavoláme metodu `Parse_RX(String rx_string)`, modul GUI, se vstupním řetězcem jako argumentem. Pokud potřebujeme předávat informace z modulu GUI do modulu UART nebo BLE, je třeba použít metod `debugWrt(String text)` pro vnitřní informace nebo metodu `sendData(String text)` pro data, která potřebujeme předat do Arduino. Na obrázku 5.3 je ukázaná struktura aplikace a komunikace mezi moduly.



Obrázek 5.3: Struktura aplikace a komunikace mezi moduly.

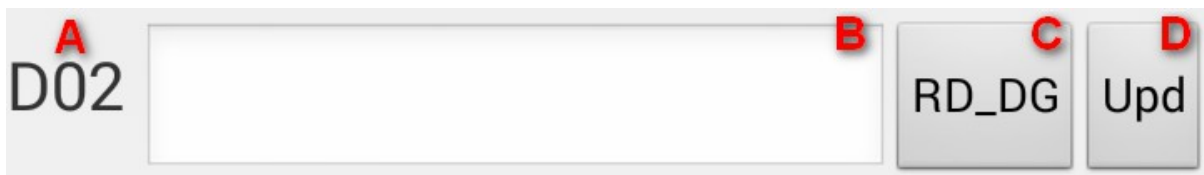
Aplikace obsahuje tři hlavní „Layout“ (pohledy): „activity_main.xml“ pro MainActivity.java, „activity_uart.xml“ (UART modul), „activity_ble.xml“ (BLE modul). Hlavní pohled „activity_main.xml“ (Obrázek 5.4 a) obsahuje jen dvě tlačítka, s pomocí kterých můžeme spustit UART modul nebo BLE modul (Obrázek 5.4 b) pohledy, které vypadají stejně. Po připojení (tlačítko „Connect“ z bloku A, Obrázek 5.4 c), budou k dispozici textová pole B a tlačko C, s pomocí kterých můžeme odeslat libovolný textový řetězec a také „Debug interface“ D, v kterém se zobrazují servisní informace. Hlavní blok pro práci s Arduino je ListView E, který vpluje dynamicky s pomocí adaptéru, jenž se nachází v GUI modulu.



Obrázek 5.4: Pohledy použité v aplikaci.

5.2.1 Struktura modulu grafického rozhraní (Android)

Základem modulu GUI je adaptér, který vyplňuje „View“ (pohled) v ListView pro každý pin Arduino. Adaptér bere pohled z „item.xml“ a mění data na základě hodnoty z array (řady) „objects“. Na obrázku 5.5 je uveden pohled jednoho elementu ListView, který je vytvořen na základě „item.xml“.



Obrázek 5.5: Pohled jednoho elementu ListView.

Element „View“ z ListView se vyplňuje na základě elementu z array „objects“ typu ArduinoItem (třída ArduinoItem z ArduinoItem.java). Datový typ ArduinoItem se skládá z hodnot:

- „name“ typu int, s pomocí této hodnoty bude vytvořen TAG pro tlačítko „Upd“ (Obrázek 5.5 D) a textové pole (Obrázek 5.5 B), které slouží pro spojování s potřebným elementem z ListView,
- „command_support“ typu int[] ukazuje, které příkazy podporuje daný pin Arduino,
- „command“ typu int ukazuje, který příkaz je vybrán, zobrazuje se na tlačítku příkaz (Obrázek 5.5 C),

- „pin_value“ typu String, obsahuje hodnotu z textového pole (Obrázek 1.5 B), při změně hodnoty „objects.pin_value“ se automaticky změní hodnota textového pole,
- „arduino_pin“ typu int, obsahuje číslo pinu v Arduino, bez písmenek A i D, jen číslo (0..13),
- „vsbl“ typu boolean, nepoužívá se.

Při vytvoření adaptéru se spustí metoda „fillData“ a vygeneruje array „objects“. S pomocí metody „getView“ se bude vyplňovat Element „View“ z ListView na základě elementu z array „objects“.

Modul GUI obsahuje metody pro parsování vstupního řetězce. Základem pro parsování jsou symboly na začátku řetězce a pozice těchto symbolů. Symboly na začátku řetězce musí být stejné jako symboly v knihovně pro platformu Arduino. Na obrázku 5.6 jsou všechny specifické symboly, které jsou potřeba pro parsování.

```
private static int separator_position = 0;
private static char analog_separator = '!';
private static char digital_separator = '#';
private static char error_separator = '@';
```

Obrázek 5.6: Specifické symboly pro GUI modul.

5.2.2 Struktura modulu pro komunikaci přes BLE (Android)

Modul byl vytvořen na základě příkladu „BluetoothLeGatt“ a také s pomocí praktického videa z Google I/O 2013 („Google I/O 2013 Best Practices for Bluetooth Development“) [33]. Při implementaci modulu BLE, pro zjednodušení, byl příklad „BluetoothLeGatt“ zredukován do jedné „Activity“.

Při vytvoření „Activity“ metody „onCreate“ potřebujeme získat objekt „BluetoothAdapter“, s pomocí tohoto objektu se bude provádět komunikace přes „radio Bluetooth“. Dále při zmáčknutí tlačítka „Connect“ spustíme hledání BLE zařízení s pomocí metody „startLeScan“, která dostane jako parametr metodu „scanCallback“. Metoda „scanCallback“ bude volána vždy při hledání BLE zařízení. Pro každé zařízení bude ověřena síla signálu a pokud bude větší než -60dBm, aplikace bude hledat číslo servisu HMSoft (proměna UART_UUID).

Po nalezení vhodného BLE zařízení, hledání bude ukončeno a s pomocí metody „connectGatt“ se aplikace připojí ke „GATT server“, který běží na BLE zařízení. Metoda „connectGatt“ dostává jako 3. argument odkaz na instanci třídy „callback“ „BluetoothGattCallback“. Instance obsahuje metody, které budou zavolány: při změně stavu připojení - „onConnectionStateChange“, při ukončení prohledávání charakteristik - „onServicesDiscovered“ a také při změně hodnoty charakteristiky - „onCharacteristicChanged“. Při změně hodnoty charakteristiky v metodu „onCharacteristicChanged“

bude zavolána metoda Parse_RX z GUI modulu, který dostane jako parametr data z charakteristiky. Data se odesílají pomocí metody „send_msg“.

HM-10 má RX a TX v jedné charakteristice, ale jiné BLE moduly ji mohou mít v různých charakteristikách. Aby se zjednodušilo pochopení a vytvořil univerzální kód pro RX a TX charakteristiky, byly zavedeny různé proměnné (RX_UUID a TX_UUID), ale se stejnou hodnotou.

5.2.3 Struktura modulu pro komunikaci přes UART (Android)

Modul byl vytvořen na základě knihovny „Physicaloid“ [34]. Při vytvoření „Activity“ metody „OnCreate“ se vytvoří objekt „mPhysicaloid“ třídy „Physicaloid“, s pomocí tohoto objektu se bude provádět komunikace s UART převodníkem. Metoda „open“ otevře UART port se standardním nastavením (9600bps-8-1-none-none). Dále bude s pomocí instance „addReadListener“ zavolána metoda „onRead“, která přečte data z bufferu UART převodníku. Odesílat data můžeme s pomocí metody „send_msg“.

Knihovna „Physicaloid“ je založena na knihovně „Java D2xx driver“ a má výhodu v tom, že obsahuje jednoduché metody a podporuje přijímání nahrávání FW do Arduino bez počítače [35]. Při implementaci měly obě knihovny stejnou možnost, jestli spojit RX a TX UART převodníku a dokola předávat jeden řetězec, například „1234567890“, občas na místo jednoho řetězce dostáváme dva, například „1234“ a „567890“. To znamená, že když budeme odesílat data a zároveň dostaneme řetězec od Arduino, tak řetězec může být špatně zpracován. Aby se odstranila separace řetězce, byl vytvořen buffer. Když se v řetězci objeví specifický symbol („!“, „#“, „@“), tak zbytek řetězce a další řetězce budou přepokopírované do bufferu, pokud neobjeví se symbol konce řetězce („;“). Po symbolu konce řetězce bude obsah bufferu odeslán v Parse_RX z GUI modulu. Začátek a zbytek řetězce nebo pokud řetězec neobsahuje žádný specifický symbol, tak bude odeslán v Parse_RX z GUI modulu bez čekání na další řetězec.

6 Testování a měření spotřeby

Používání komunikace přes BLE a DigiMesh zajišťuje nízkou spotřebu. Aby se vytvořila efektivní jednotka pro snímání dat ze senzorů a řízení efektorů, potřebujeme zjistit energetické nároky dvou modulů. Nastavení konfigurace modulů jsou uvedena v příloze C (pro DIGI S1 v programu XCTU a pro HM-10 v PCComAssistant) [15].

Zredukovat spotřebu energie lze pomocí automatického režimu spánku(AT+PWRM0), nastavení výkonu (AT+POWE[0..3]), zvětšením slave latency (AT+COLA[0..3]) a Link Layer connection interval(AT+COMI[0..9], AT+COMA[0..9]) pro BLE modul. Pro xBee S1 použitím režimu spánku, například s probuzením od externího přerušení (Sleep Mode = 1, PIN HIBERNATE).

Protože cílem testování bylo porovnat moduly, zjistit jejich spotřebu energie v obyčejných režimech práce (IDLE, Connect, Sending and receiving data) a porovnat naměřené hodnoty se spotřebou energie podle „datasheet“, byla použita standardní nastavení modulu BLE a DIGI S1. Spotřebu energie při různých režimech spánku je možné přibližně spočítat z obyčejných režimů práce, například modul Pro xBee S1 bude vypnut po 10 sekundách (spotřeba 10 μ A podle „datasheet“) a 1 sekundu bude v IDLE režimu (spotřeba 50 μ A podle „datasheet“) - tak dostaneme spotřebu 5mA za 11 sekund.

Pro měření byl použit stolní multimetr Fluke 8845A (Rozsah 100mA, Rozlišení 6½ Digits, přesnost podle manuálu \pm (0.05% měření + 0.005% z rozsahu)). Oba moduly mají pracovní napětí 3,3V a byly napájeny z laboratorního napájecího zdroje AGILENT E3632A (Rozlišení napětí 1 mV, stabilizace napětí nebyla použita). Pracovní místo pro měření proudu je ukázáno na obrázku 6.1.



Obrázek 6.1: Pracovní místo pro měření proudu.

Perioda testování byla nastavena na 60 sekund a pro názornost proudu dány do jedné tabulky 1.1. V „IDLE“ režimu bylo do modulu připojeno jen napětí 3,3V, „master“ zařízení bylo vypnuto (koordinátor pro DIGI S1, vypnuty „Bluetooth“ na zařízení s OS Android pro HM-10). V „Connected“ režimu bylo do modulu připojeno jen napětí 3,3V, a „master“ zařízení bylo zapnuto (koordinátor pro DIGI S1, navázáno spojení se zařízením s OS Android pro HM-10). V „Receive/Transmit data“ režimu bylo do modulu připojeno napětí 3,3V, piny RX a TX byli spojeni a z „master“ zařízení zasílán příkaz „RD_DG 02“ jedenkrát za 4 vteřiny (15 krát za test) a přijímán zpět. Režim „IDLE+Arduino“ je stejný jako „IDLE“, ale do modulu je připojen „Arduino Pro mini 3,3V“. Poslední režim „Receive/Transmit data from/to Arduino“ je shodný s „Receive/Transmit data“, ale připojeno je „Arduino Pro mini 3,3V“ a na zasílání příkazu „RD_DG 02“ jedenkrát za 5 vteřiny, „master“ dostává odpověď „#02 HI“ 5 krát. Diagramy odběru proudu na základě naměřených dat jsou v příloze A (Obrázky A1...A5).

Jméno režimu	Naměřený proud (Střední hodnota v mA)		Proud podle specifikace (mA)	
	HM-10	DIGI S1	HM-10	DIGI S1
IDLE	8,3947	52,0841	8,5	50
Connected	8,2508	52,2153	8,5	50
Receive/Transmit data	8,2499	51,9936	8,5	50/45

Sleep mode	-	-	0,4..1,5	<0.01
IDLE+Arduino	18,2825	61,9930	-	-
Receive/Transmit data from/to Arduino	18,2561	61,8923	-	-

Tabulka 6.1: Odběr proudu v různých režimech pro moduly HM-10 a DIGI S1.

Z tabulky je vidět, že modul HM-10 má menší odběr proudu, než DIGI S1. Existují omezení, která musíme brát v úvahu při projektování jednotky s komunikačním modulem HM-10 nebo DIGI S1. Modul HM-10 funguje jako „Slave Bluetooth“ zařízení a z toho důvodu nemůže komunikovat s „Master“ zařízením, pokud nebude iniciováno spojení od „Master“. Z opačné strany DIGI S1 může komunikovat kdykoliv, ale používá 6krát větší proud, než HM-10 s navázaným spojením. Při zapnutí „sleep mode“ modul HM-10 je viditelný (proud 0,4..1,5mA) a po připojení je aktivní (proud ~8,5mA). DIGI S1 v cyklickém „sleep mode“ není viditelný nebo aktivní, pokud spí, po probuzení může odesílat, přijímat nebo přesměrovat data. Pokud je použit acyklický „sleep mode“, nebude fungovat přesměrování dat (mesh network features).

Když není důležité používat mesh síť a brát v úvahu zvláštnosti používání HM-10, je nejlepší vybrat komunikační modul HM-10.

Testování obousměrného snímání a řízení bylo provedeno na „Sensor kit“ rozšířeném dalšími senzory a efektory [38]. Foto a seznam senzorů i efektorů se nachází v příloze B. V tabulce 6.2 je uvedeno jméno senzoru, příkaz, kterým byl senzor otestován, délka čekání i počet cyklů a také výsledky testování.

Jméno	Příkaz	Počet cyklů, délka čekání.	Výsledky.
Mercury switch	RD_DG 02;	1 loop, 1ms.	Zobrazila se správná hodnota digitálního signálu, na senzoru se rozsvítla dioda při nízké hodnotě na pinu. Příkaz se opakoval 5 krát.
Tilt switch	RD_DG 04;	5 loop, 100ms.	Zobrazila se správná hodnota digitálního signálu, ale senzor není příliš spolehlivý. Příkaz se opakoval 5 krát.
Hall sensor_s	RD_DG 07;	5 loop, 500ms.	Zobrazila se správná hodnota digitálního signálu, na senzoru se rozsvítla dioda při nízké hodnotě na pinu. Příkaz se opakoval 5 krát.
Opto interrupter	RD_DG 08;	5 loop, 1s.	Zobrazila se správná hodnota digitálního signálu. Pokud mezi vysílačem a přijímačem není překážka, na pinu bude nízká úroveň signálu. Příkaz se opakoval 5 krát.

FLAME sensor	BUTTN 12 0;	1 sample, 10s.	Zobrazila se správná hodnota digitálního signálu. Při detekci ohně na pinu byla vysoká úroveň signálu. Příkaz se opakoval 5 krát.
HC-SR501	BUTTN 12 5;	2 sample, 20s.	Zobrazila se správná hodnota digitálního signálu. Při detekci pohybu na pinu byla vysoká úroveň signálu. Příkaz se opakoval 5 krát.
Sound sensor	BUTTN 02 0;	1 sample, 20s.	Zobrazila se správná hodnota digitálního signálu. Při detekci šumů na pinu byla vysoká úroveň signálu. Příkaz se opakoval 5 krát.
Reed switch	BUTTN 03 5;	5 sample, 20s.	Zobrazila se správná hodnota digitálního signálu. Při detekci magnetického pole na pinu byla nízká úroveň signálu. Příkaz se opakoval 5 krát.
Obstacle sensor	BUTTN 5 99;	2 sample, 20s.	Zobrazila se správná hodnota digitálního signálu. Při detekci překážky na pinu byla nízká úroveň signálu. Příkaz se opakoval 5 krát.
Line sensor	BUTTN 05 9;	2 sample, 20s.	Zobrazila se správná hodnota digitálního signálu. Při detekci černé barvy na pinu byla vysoká úroveň signálu. Příkaz se opakoval 5 krát.
Knock sensor	SWTCH 6 1;	4 sample, 20s.	Zobrazila se správná hodnota digitálního signálu. Při detekci klepání na pinu byla nízká úroveň signálu. Příkaz se opakoval 5 krát.
Button	SWTCH 7 3;	5 sample, 30s.	Zobrazila se správná hodnota digitálního signálu. Při zmačknutí tlačítka se změnil vnitřní stav. Příkaz se opakoval 5 krát.
Touch sensor	SWTCH 8 1;	2 sample, 30s.	Zobrazila se správná hodnota digitálního signálu. Při dotyku se změnil vnitřní stav. Příkaz se opakoval 5 krát.
RGB led	LED_3 1 1 1;	-	LED dioda svítila na minimální úrovni.
RG led	LED_3 0 99;	-	LED dioda svítila pouze zeleně.
DHT11	DHTSW 02;	5 loop, 100ms.	Zobrazila se správná hodnota teploty(-0,8°C od referenčního zařízení) a vlhkosti (-4%) . Příkaz se opakoval 5 krát.
DS18B20	TMPOW 4 C;	5 loop, 500ms.	Zobrazila se správná hodnota teploty(+0,5°C od

			referenčního zařízení). Nefunguje na pinu D3. Příkaz se opakoval 5 krát.
Temperature sensor	TMPAN 0 C;	5 loop, 100ms.	Zobrazila se správná hodnota teploty(-0,15°C od referenčního zařízení). Příkaz se opakoval 5 krát.
ICSG017A	RD_AN 1 V;	10 loop, 99ms.	Zobrazila se správná hodnota napětí, při detekci plynu ~4,16V. Příkaz se opakoval 5 krát.
Photo resistor	RD_AN 2;	5 loop, 500ms	Zobrazila se správná analogová hodnota, tma ~20, denní světlo ~800. Příkaz se opakoval 5 krát.
YL-69	RD_AN 3;	5 loop, 1s	Zobrazila se správná analogová hodnota, minimální úroveň vlhkosti půdy ~800, maximální ~200. Příkaz se opakoval 5 krát.
Relay	WR_DG 8 ON;	-	Relé se zapnulo po příkazu.
Laser	WR_DG 9 ON;	-	Laser se rozsvítil po příkazu.
RS-360SH	WR_DG 10 ON;	-	Čerpadlo se zapnulo po příkazu.
RG led_5	WRPWM 3 255; WRPWM 6 255;	-	LED dioda svítila zeleně a červeně.
Adafruit SI1145	I2CLS;	5 loop, 1s.	Zobrazily se hodnoty osvětlení.
PS2 joystick	JOYST;	50 loop, 99ms.	Zobrazilo se 50 hodnot polohy.
IR receiver	IRRCV 3;	3 sample, 20s.	Zobrazily se 3 kódy NEC.

Tabulka 6.2: Výsledky testování senzorů a efektorů.

Jak vidíme, prakticky bylo ukázáno, že při použití této knihovny lze efektivně snímat data ze senzorů a řídit efekty.

Aplikace byla otestována na různých zařízeních s OS Android. Výsledky jsou v tabulce 6.3.

Zařízení	BLE modul	UART modul	Verze OS Android.
Sony xperia z1 compact	Funguje	Netestováno	v4.4
Prestigio PSP 5507	Funguje	Nefunguje	v4.4.2
Acer S520	Nepodporuje (OS)	Funguje	v4.2.2
Prestigio PMT5008	Funguje	Funguje	v4.4.2
TeXeT TM-9751HD	Nepodporuje (HW)	Funguje	v4.3, v4.4
LG Nexus 4	Nepodporuje (OS)	Nepodporuje (HW)	v4.2

Tabulka 6.2: Testování aplikace na zařízeních s OS Android.

Teoreticky by BLE modul měl fungovat na všech zařízeních s OS Android v4.3 a výše. Modul UART bych měl pracovat na většině tabletů a na malém počtu telefonů, a to z důvodu omezené podpory USB API.

7 Závěr

7.1 Možnosti rozšíření

Protokol lze do budoucna rozšířit o základní příkazy s Firmata protokolu. Tento protokol je velmi populární pro řízení platformou Arduino.

Dále je možné se zaměřit na předávání dat přes „Advertising“ vysílání při použití HM-10 ze „sensor firmware“ a také je možné řídit Arduinem v režimu spánku s pomocí pinu na HM-10, což sníží spotřebu energie.

Nejlépe je přidat další moduly pro bezdrátovou komunikaci, například ZigBee na čipu CC2530 (KittyBee Arduino ZigBee Shield) a funkci pro zpracování paketů podle požadavku jejich protokolu.

Do aplikace lze dodat funkci připojení podle MAC adresy a funkci vyhledávání zařízení (pro BLE a DIGI S1).

7.2 Zhodnocení

V rámci diplomové práce jsem zpracoval hlavní typy bezdrátových komunikačních prostředků mobilních zařízení využívajících OS Android a HW platformy Arduino.

Byl rozpracován protokol a knihovny umožňující vytvořit řídicí jednotku, která může být použita k snímání dat ze senzorů a ovládání efektorů s pomocí aplikací pod OS Android.

V současné době roste zájem o domácí automatizace a inteligentní domácnost. Vyrábí se mnoho senzorů od světově známých firem i od velmi malých společností. Na druhou stranu je těžké najít komplexní systém, který má inteligenci a vypracované chytré algoritmy analýzy dat ze senzorů a řízení efektorů.

Vývoj algoritmů pro systém řízení chování inteligentní domácnosti je dnes velmi perspektivní oblast. Pro vývoj logiky chování se nejlépe hodí jednotky založené na Arduino se zpracováním protokolu. Hlavní výhoda je v tom, že programátor, který používá knihovnu v OS Android, nemusí vědět nic o mikrokontrolerech. Stačí nahrát sketch v Arduino, připojit senzor nebo efektor a řídit nebo číst data v aplikaci pod OS Android.

Byly porovnány dva bezdrátové moduly DIGI S1 i HM-10 a byl vybrán BLE modul HM-10 z důvodu nízké spotřeby. Také bylo úspěšně otestováno množství senzorů a efektorů.

Byl vytvořen základ pro propojení libovolného modulu s UART rozhraním do Android zařízení pro komunikaci s platformou Arduino.

V rámci budoucího rozšíření by bylo vhodné dodat nové bezdrátové moduly a rozšířit protokol.

Literatura

- [1] Core Version 4.0 [online]. [cit. 2015-05-01]. Dostupný z WWW: <<https://www.bluetooth.org/en-us/specification/adopted-specifications>>
- [2] Android Connectivity API [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://developer.android.com/guide/topics/connectivity/index.html>>
- [3] Faludi, R.: Building Wireless Sensor Networks: With ZigBee, XBee, Arduino, and Processing. O'Reilly Series, O'Reilly Media, 2010, ISBN 9780596807733, 318 s.
- [4] Kickstarter, project The AirBoard [online]. [cit. 2015-05-01]. Dostupný z WWW: <<https://www.kickstarter.com/projects/223628811/the-airboard-sketch-internet-of-things-fast/description>>
- [5] XB-Buddy Kit [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.instructables.com/id/xB-Buddy-Kit/>>
- [6] USB-to-UART(CP2102) [online]. [cit. 2015-05-01]. Dostupný z WWW: <http://pandatron.cz/?2648&usb_modul_pu232c>
- [7] usb-serial-for-android [online]. [cit. 2015-05-01]. Dostupný z WWW: <<https://github.com/mik3y/usb-serial-for-android>>
- [8] usb-serial-for-android [online]. [cit. 2015-05-01]. Dostupný z WWW: <<https://github.com/alessandrelli/usb-serial-for-android>>
- [9] TI CC2531 ZigBee [online]. [cit. 2015-05-01]. Dostupný z WWW: <http://processors.wiki.ti.com/index.php/Android_ZigBee_Interface>
- [10] BigHappyDude R2D2 [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://bighappyduder2.blogspot.co.uk/>>
- [11] Itead Studio bluetooth module HC-05 [online]. [cit. 2015-05-01]. Dostupný z WWW: <ftp://imall.iteadstudio.com/Modules/IM120723009/DS_IM120723009.pdf>
- [12] Bluegiga WT11 [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.bluegiga.com/en-US/products/bluetooth-classic-modules/wt11i-bluetooth-class-1-module/>>
- [13] ArduinoBT [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.arduino.cc/en/Main/ArduinoBoardBT>>
- [14] TI: cc2540 [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.ti.com/product/cc2540>>
- [15] HM-10 [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.jnhuamao.cn/bluetooth.asp?ID=1>>
- [16] CC-Debugger [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.ti.com/tool/cc-debugger>>

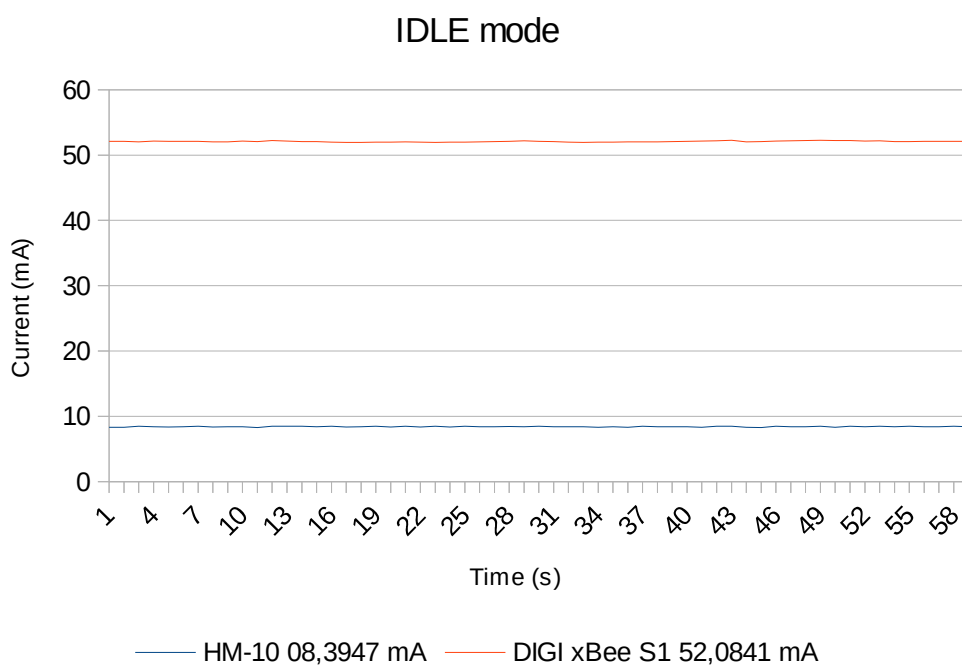
- [17] Arduino Wireless Shield [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://arduino.cc/en/Main/ArduinoWirelessShield>>
- [18] The Major Differences in the XBee Series 1 vs. the XBee Series 2 [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.digi.com/support/kbase/kbaseresultdetl?id=2213>>
- [19] Product Manual XBee/XBee-PRO 802.15.4 RF [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.digi.com/support/getasset?fn=90000982&tp=3>>
- [20] Kittybee ZigBee shield for arduino [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://code.google.com/p/kittybee/>>
- [21] TI Z-Stack [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.ti.com/tool/z-stack>>
- [22] ZBOSS Stack [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://zboss.dsr-wireless.com/downloads/index/zboss>>
- [23] Zigduino r2 [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.logos-electro.com/>>
- [24] Zigduino Radio Arduino Library [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://code.google.com/p/zigduino-radio/>>
- [25] NFC library for Arduino [online]. [cit. 2015-05-01]. Dostupný z WWW: <<https://github.com/Seeed-Studio/PN532>>
- [26] NDEF library [online]. [cit. 2015-05-01]. Dostupný z WWW: <<https://github.com/don/NDEF>>
- [27] HLK-RM04 [online]. [cit. 2015-05-01]. Dostupný z WWW: <http://www.hlktech.net/product_detail.php?ProId=39>
- [28] HLK_RM04_User_Manual.pdf [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.hlktech.net/inc/lib/download/download.php?DId=21>>
- [29] Lewis, J: Ouch. The Arduino GSM shield has a pretty serious design flaw, with its capacitors. [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.baldengineer.com/ouch-the-arduino-gsm-shield-has-a-pretty-serious-design-flaw-with-its-capacitors.html>>
- [30] SIM900 AT Command Manual V1.09 [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.simcom.ee/modules/gsm-gprs/sim900/>>
- [31] SIM900_SIM908 Arduino's Shield Library [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.gsmlib.org/download.html>>
- [32] GSM Library [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://arduino.cc/en/Reference/GSM>>

- [33] Sample BluetoothLeGatt [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://developer.android.com/samples/BluetoothLeGatt/index.html>>
- [34] PhysicaloidLibrary [online]. [cit. 2015-05-01]. Dostupný z WWW: <<https://github.com/ksksue/PhysicaloidLibrary>>
- [35] FTDI's Android Support Page [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://www.ftdichip.com/Android.htm>>
- [36] Townsend, K; Cufi, C.; Akiba; Davidson, R.: Getting Started with Bluetooth Low Energy. O'Reilly Media, 2014, ISBN 9781491949511, 180 s.
- [37] Adler, E.: Here's Why 'The Internet Of Things' Will Be Huge, And Drive Tremendous Value For People And Businesses [online]. Poslední modifikace: 2014-08-19 [cit. 2015-05-01]. Dostupný z WWW: <<http://www.businessinsider.com/growth-in-the-internet-of-things-market-2014-1>>
- [38] 37 in 1 box Sensor Kit For Arduino Starters [online]. [cit. 2015-05-01]. Dostupný z WWW: <<http://en.keyes-robot.com/products/show.aspx?id=152>>

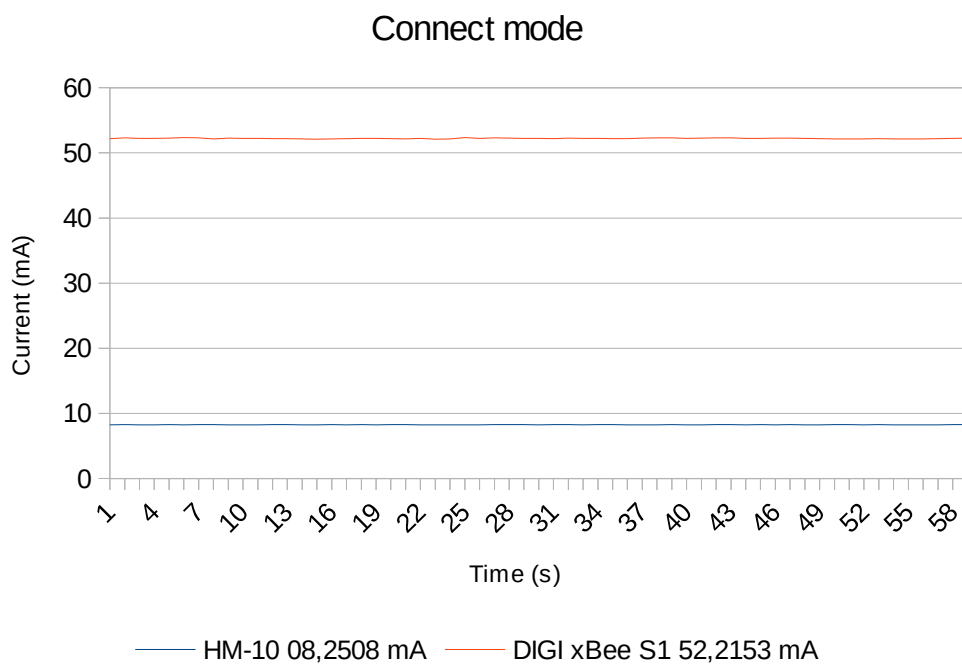
Seznam příloh

- Příloha A. Diagramy odběru proudu.
- Příloha B. Seznam senzorů a efektorů.
- Příloha C. Konfigurace modulů.
- Příloha D. AT-příkazy.
- Příloha E. CD/DVD.

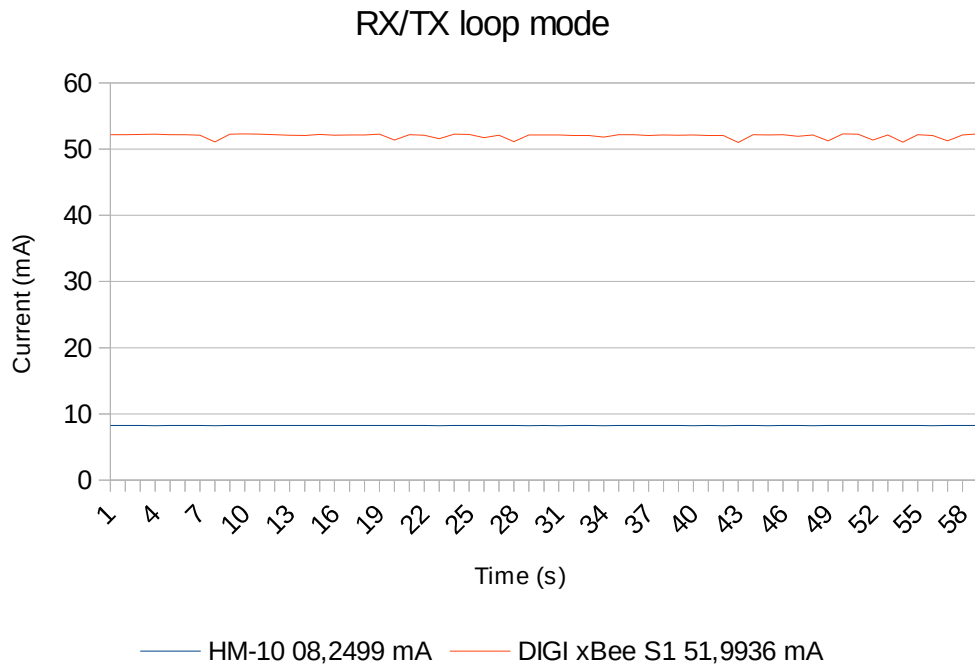
Příloha A.



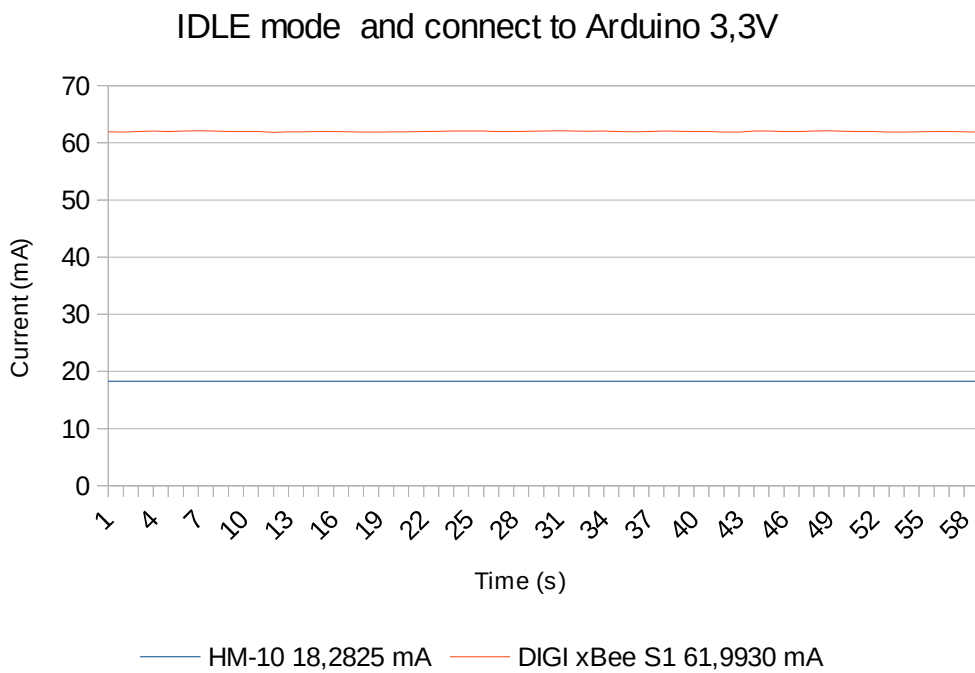
Obrázek A.1: IDLE režim bez zapnutí režimu spánku.



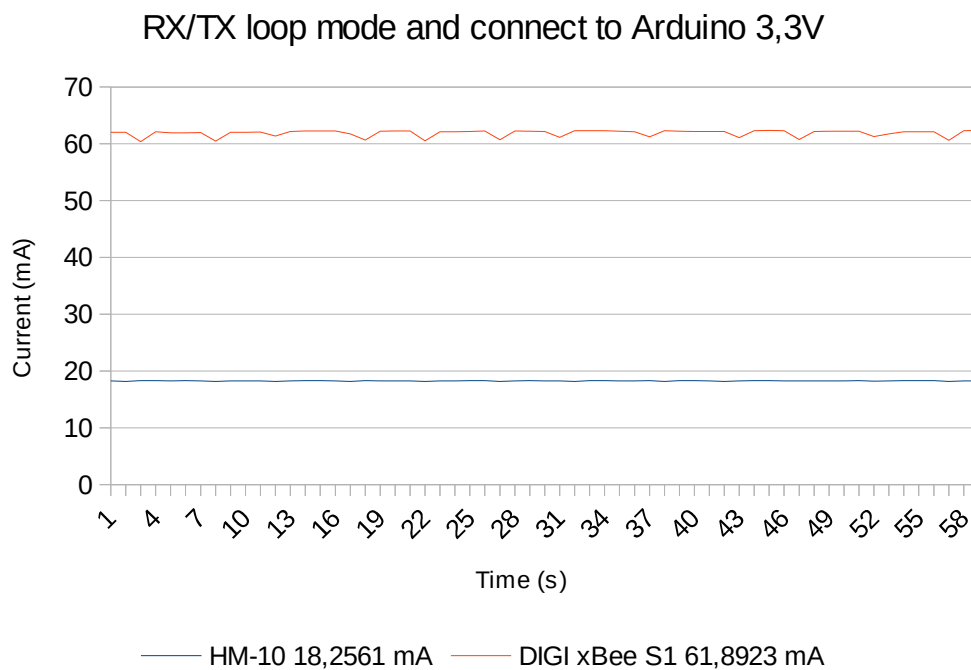
Obrázek A.2: Modul je připojen bez zapnutí režimu spánku.



Obrázek A.3: Režim odesílání a přijímání dat bez zapnutí režimu spánku.



Obrázek A.4: IDLE režim s připojeným Arduino bez zapnutí režimu spánku.



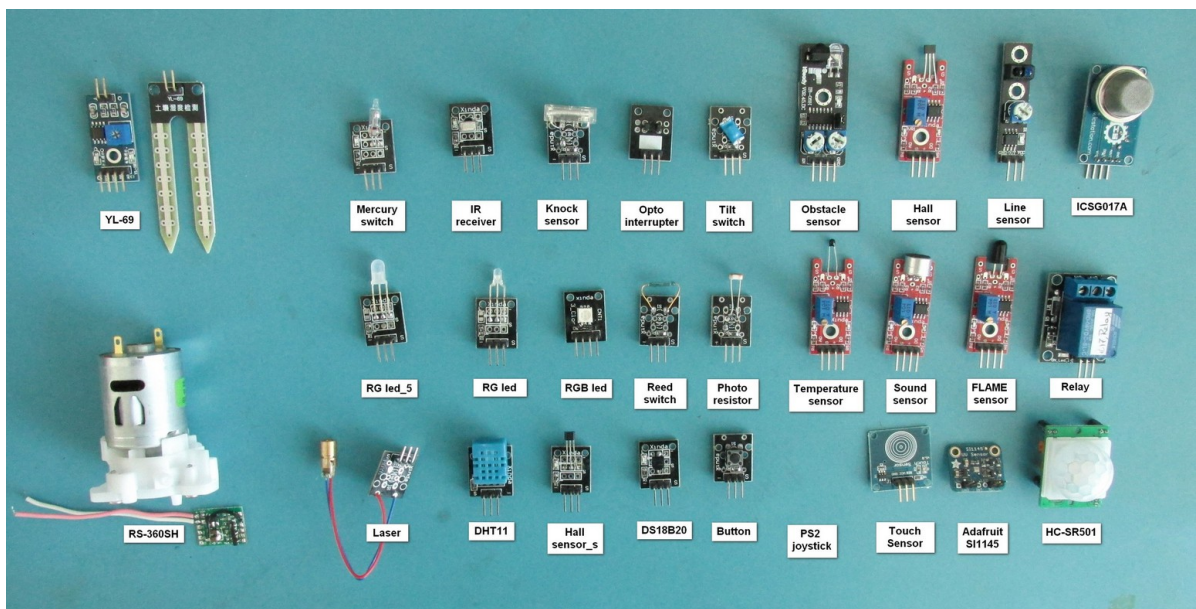
Obrázek A.5: Režim odesílání a přijímání dat s připojeným Arduino bez zapnutí režimu spánku.

Příloha B

Jméno	Jméno v Sensor kitu	Popis.
Adafruit SI1145	-	Senzor osvětlení ve viditelné oblasti (400nm-800nm), blízké IR (550nm-1000nm) oblasti a UV indexu.
DS18B20	KY-001	Číslicový teploměr Dallas Semiconductor, s typickou přesností $\pm 0.5^{\circ}\text{C}$ (-10°C to $+85^{\circ}\text{C}$), rozsah -55 do $+125^{\circ}\text{C}$.
DHT11	KY-015	Senzor teploty a vlhkosti vzduchu DHT-11. Rozsah měření vlhkosti 20% - 80%, přesnost $\pm 5\%$. Rozsah měření teploty $0-50^{\circ}\text{C}$, přesnost $> \pm 2^{\circ}\text{C}$.
FLAME sensor	KY-026	Modul detekce plamene (základ - fototranzistor YG1006). Detekce vlnové délky 760–1100nm. Úhel detekce 60 stupňů. Nastavitelná citlivost potenciometrem.
ICSG017A	-	Modul detektoru plynů a kvality vzduchu se senzorem MQ-135 (CO ₂ , NO _x , NH ₃ , alkohol, benzen).
IR receiver	KY-022	Infračervený přijímač. Použitý senzor: HX1838 s vysokou citlivostí. Dosah: 5 až 8m.
PS2 joystick	KY-023	PS2 ovladač. Použitý senzor: P3 America 802.
Relay	KY-019	Relé modul 5V 1-kanálové rozhraní, potřebuje 15-20mA. AC250V 10A; DC30V 10A. Základ: SRD-5VDC-SL-C.
Temperature sensor	KY-028	Analogový snímač teploty. Použitý senzor: NTC termistor. Rozsah $-55^{\circ}\text{C}/+125^{\circ}\text{C}$, přesnost $\pm 0.5^{\circ}\text{C}$. Použitý senzor: TTC3A103 34D.
Photo resistor	KY-018	Světelné Čidlo. Použitý senzor: fotorezistor.
Laser	KY-008	Laser modul červený 650nm 5mW 5V. Základ: laser dioda.
RGB led	KY-009	SMD 5050 RGB led dioda.
Opto interrupter	KY-010	Použitý senzor: infračervená světelná závora vysílač/přijímač.
Reed switch	KY-021	Jazyčkový magnetický senzor. Použitý senzor: jazyčkový kontakt.
Mercury switch	KY-017	Snímač polohy, náklonu. Použitý senzor: rtuťový spínač.
Tilt switch	KY-020	Snímač polohy.
Knock sensor	KY-031	Čidlo klepání.
Obstacle sensor	KY-032	Senzor pro detekci překážek. Pracuje na principu snímání odraženého IR paprsku.

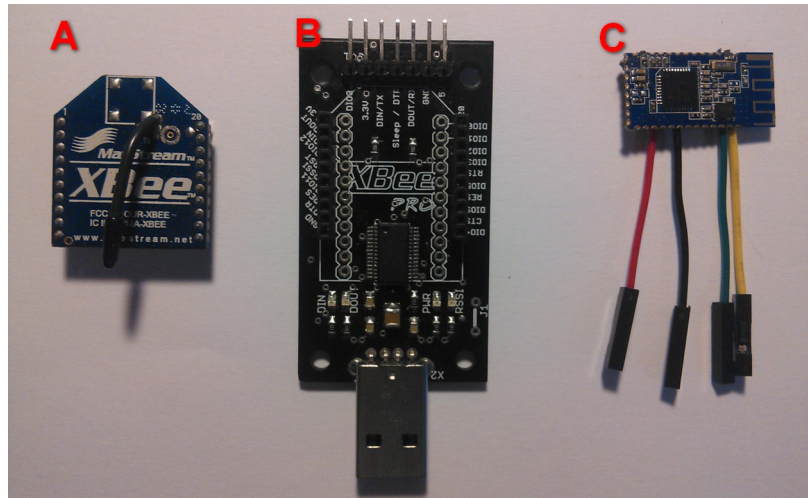
Hall sensor	KY-024	Hallův senzor (analogový a digitální výstup).
Hall sensor_s	KY-003	Hallův senzor (detektor magnetického pole).
Line sensor	KY-033	IR senzor sledování čáry.
Sound sensor	KY-038	Mikrofon.
RG led	KY-011	3mm červená a zelená led dioda (společná katoda).
RG led_5	KY-029	5mm červená a zelená led dioda (společná katoda).
Button	KY-004	Tlačítko.
Touch sensor	-	Kapacitní dotykové tlačítko.
HC-SR501	-	Senzor pohybu (pasivní infračervený detektor).
YL-69	-	Senzor půdní vlhkosti.
RS-360SH	-	Vodní čerpadlo.

Tabulka B.1: Seznam senzorů a efektorů.

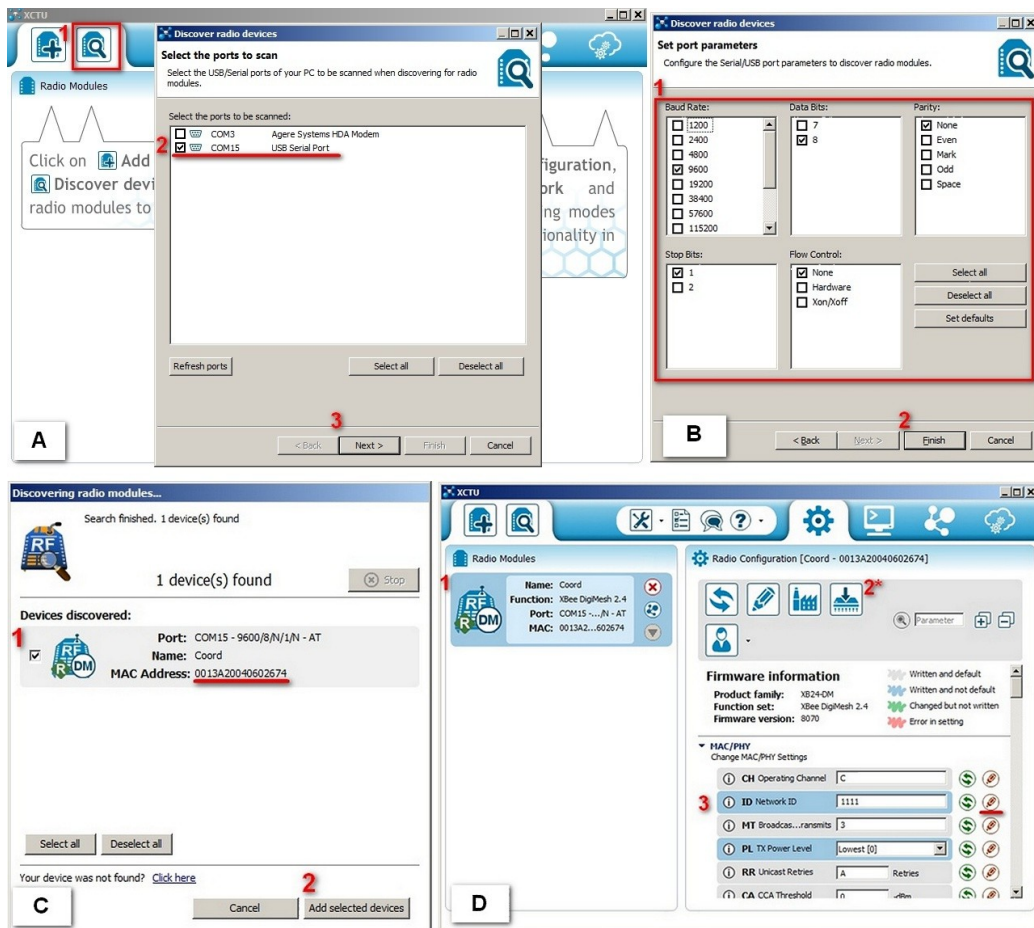


Obrázek B.2: Foto senzorů a efektorů.

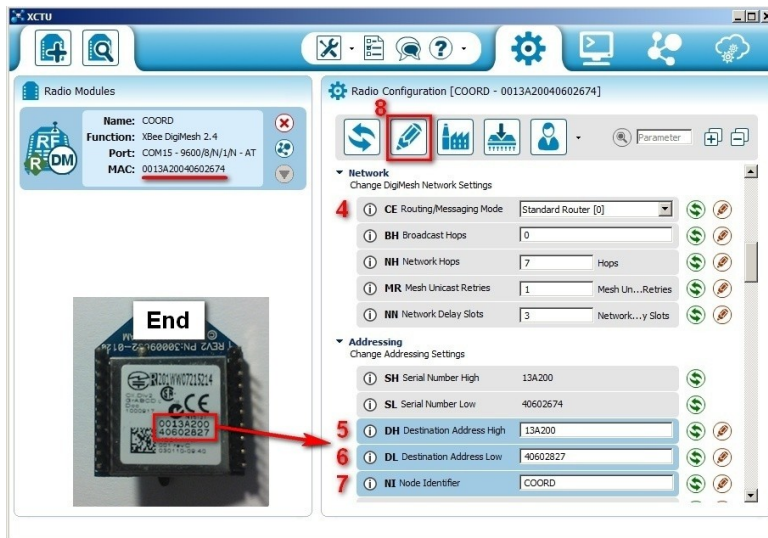
Příloha C



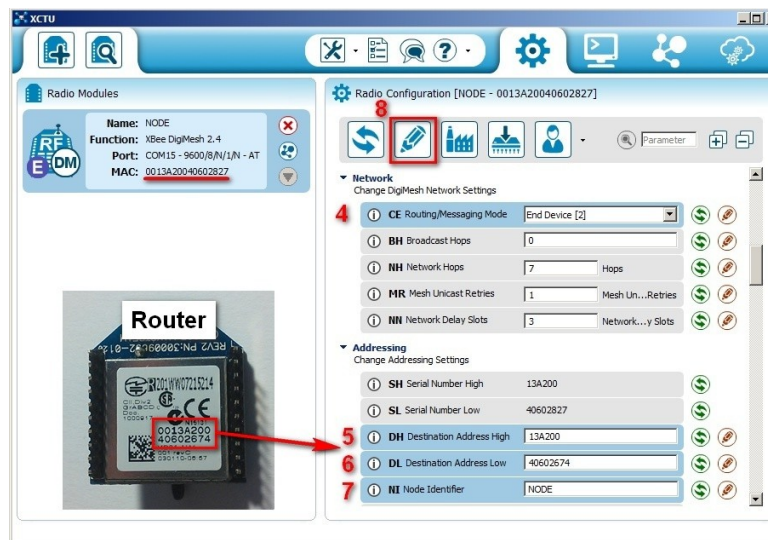
Obrázek C.1: HW prostředky pro konfiguraci modulů.



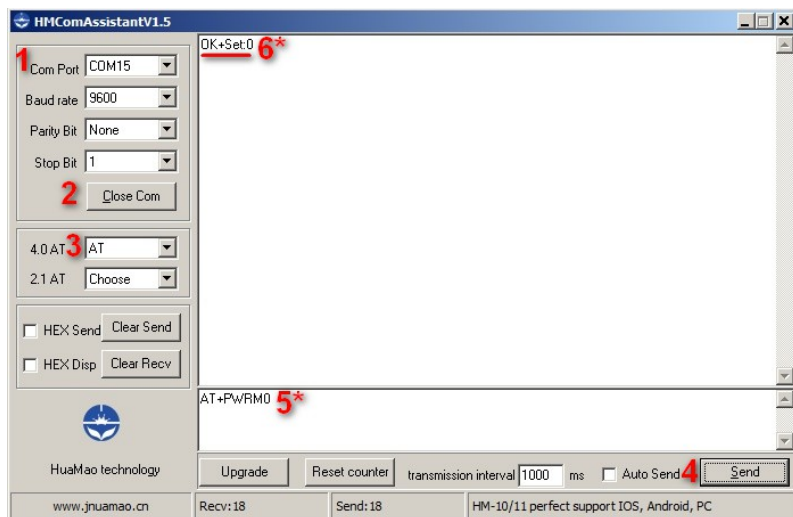
Obrázek C.2: Nastavení DIGI S1 modulu v XCTU.



Obrázek C.3: Nastavení Routeru DigiMesh.



Obrázek C.4: Nastavení End nodu DigiMesh.



Obrázek C.5: Nastavení HM-10 modulu.

Příloha D

AT-příkaz	Popis činnosti
AT	Testovací příkaz
AT+RESET	Restartovat modul
AT+VERSION?	Verze FW
AT+ORGL	Vynulovat nastavení
AT+ADDR?	MAC adresa modulu
AT+NAME=JMENO	Nastavit jméno modulu
AT+RNAME?<MAC_adresa>	Získat jméno BT zařízení (na základě MAC adresy)
AT+PSWD=1235	Změna hesla
AT+UART=Baudrate,StopBit,Parity	Změnit nastavení UART
AT+STATE?	Zjistit stav modulu
AT+ROLE=<param>	Param: 0 - Slave role; 1 - Master role; 2 - Slave-Loop role
AT+CMODE=<param>	Param: 0 - připojení k jakémukoli zařízení, 1 - připojení pouze na zařízení, které jsou zaznamenány v paměti (již byly alespoň jednou připojeny)
AT+LINK=123456abcdef	Připojit se k zařízení 12:34:56:ab:cd:ef
AT+DISC=<MAC_adresa>	Odhlásit

Tabulka D.1: Základní příkazy modulu HC-05 [11].

AT-příkaz	Popis činnosti
AT	Testovací příkaz
AT+HELP?	Pomoc
AT+VER?	Verze FW
AT+RENEW	Vynulovat nastavení
AT+RESET	Restartovat modul
AT+NAME?	Přečíst jméno modulu
AT+NAME<jmeno>	Nastavit jméno modulu (max. 12 znaků)
AT+ADDR?	Přečíst MAC adresu modulu
AT+ROLE?	Přečíst roli přístroje
AT+ROLE<param>	Nastavit roli přístroje

	<param>= M - Master; S - Slave
AT+PASS?	Přečíst PIN zařízení
AT+PIN<00000>	Nastavit PIN zařízení na 000000
AT+IMME?	Přečíst režim provozu zařízení (AT příkazy)
AT+IMME<param>	Nastavit režim provozu zařízení (AT příkazy) <param>= 0 - po zapnutí pracuje pouze s AT příkazy, po příkazu AT+WORK pracuje v normálním režimu; 1 - pracuje v normálním režimu po zapnutí. (pouze pro „master“ roli)
AT+ MODE<param>	Nastavit režim provozu zařízení <param>= 0 - AT-příkazy přimá jenom na UART portu; 1 - AT-příkazy přimá na UART portu a od zařízení, které je připojeno.
AT+BAUD?	Přečíst přenosovou rychlost
AT+BAUD<param>	Nastavit přenosovou rychlost <param>= 0 - 9600; 1 - 19200; 2 - 38400; 3 - 57600; 4 - 115200.
AT+POWE?	Přečíst výkon vysílání
AT+ POWE <param>	Nastavit výkon vysílání <param>= 0 - -23dBm; 1 - -6dBm; 2 - 0dBm; 3 - 6dBm

Tabulka D.2: Základní příkazy modulu HM-10 [15].

Příkaz	Popis	Platné hodnoty	Výchozí hodnota
ATRE	Obnovení výchozích nastavení z výroby (to není trvalé, pokud nenásleduje příkaz ATWR).	-	-
ATWR	Zapsat nově nakonfigurované hodnoty parametrů do permanentní (dlouhodobé) paměti.	-	-
ATCN	Ukončí příkazový režim nyní. (po uplynutí několika sekund bude příkazový režim ukončen i bez příkazu ATCN).	-	-
ATID	ID sítě.	0 - 0xFFFF	0xFFFF
ATCH	Kanál Bee modulu.	0x0B - 0x1A	0x0C

ATSH a ATSL	Sériové číslo Bee modulu (SH dává hoch 32 bitů, SL los 32 bitů). Pouze pro čtení.	0 – 0xFFFFFFFF (Pro oba SH a SL)	Jiné pro každý modul.
ATMY	16-bitová adresa modulu.	0 - 0xFFFF	Jiné pro každý modul.
ATDH a ATDL	Cílová adresa pro bezdrátovou komunikaci (DH je hoch 32 bitů, DL los 32).	0 – 0xFFFFFFFF (Pro oba DH a DL)	0x0 (Pro oba DH a DL)
ATBD	Přenosová rychlost používaná pro sériovou komunikaci s Arduino nebo počítačem.	0 (1200 bps); 1 (2400 bps); 2 (4800 bps); 3 (9600 bps); 4 (19200 bps); 5 (38400 bps); 6 (57600 bps); 7 (115200 bps).	3 (9600 baud)

Tabulka D.3: Základní AT příkazy pro moduly XBee S1, S2 [19].

Přijímání hovoru	
AT	Jednoduchý příkaz, na který GPRS Shield reaguje jen indikací správného připojení a vrátí odpověď OK.
ATA	Příkaz přijmout přichozí hovor. Při přichozím volání přichází z desky zpráva RING.
AT+CHUP	Odmítnout hovor.
AT+CLCC	Přečíst číslo volajícího.
Vytvoření hovoru	
ATD + x	Vytočit číslo x. Pokud se hovor nekonal, například není žádný signál základnové stanice, přichází z desky zpráva NO DIALTONE.
Odeslání SMS zpráv	
AT+CMGF=1	Nastaví textový režim SMS zpráv.
AT + CMGS = "x"	Nastavit x jako číslo příjemce SMS.
(char)26 nebo ctrl+z	Završuje text zprávy, po tomu symbolu se začíná zpráva odesílat.
Příjem SMS zpráv	
AT+CNMI=1,2,2,1,0	Obsahuje oznámení o nových zprávách, nové zprávy dorazí v následujícím formátu: + CMT: "<telefonní number>", "", "<date time>" a na dalším řádku s prvním znakem je obsah zprávy

Tabulka D.4: Základní AT příkazy modulu GPRS Shield V2.0 [30].

Příloha E

Obsah CD/DVD:

- Technická zpráva.
- Zdrojové kódy.
- Aplikace pro OS Android.
- Dokumentace a programy pro bezdrátové moduly.