



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Firmware over the air (FOTA) pro p ípravku Arduino
Student:	Janusz Wijas
Vedoucí:	Ing. Ivo Hále ek
Studijní program:	Informatika
Studijní obor:	Po íta ové inženýrství
Katedra:	Katedra íslicového návrhu
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Cílem práce je analýza možností, jak aktualizovat firmware v p ípravku Arduino pomocí bezdrátové technologie (p ístup typu FOTA - Firmware over the air) a implementace nejvhodn jšího ešení.

Konkrétní úkoly:

1. Analyzujte použití FOTA pro Arduino pomocí Bluetooth, Wifi, p ípadn jiných bezdrátových technologií.
2. Na základ analýzy a po dohod s vedoucím práce vyberte vhodnou bezdrátovou technologii a implementujte v ní FOTA pro Arduino.
3. Experimentáln ov te na jednoduché demonstra ní aplikaci.
4. P ípadnou neúsp šnou implementaci (nap . z dvodu chyb v nízkonákladovém hardwaru) zd vodn te a zdokumentujte.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Hana Kubátová, CSc.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 11. ledna 2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA ČÍSLICOVÉHO NÁVRHU



Bakalářská práce

Firmware over the air (FOTA) pro přípravek Arduino

Janusz Piotr Wijas

Vedoucí práce: Ing. Ivo Háleček

15. května 2017

Poděkování

Děkuji všem a za vše. A opravdu si to myslím. Ale hlavní dík patří mým rodičům za jejich péči, lásku a nekonečnou psychickou výdrž. Dále bratrovi, je to jedná z mála osob, která mi rozumí a i přes všechny hádky, vždy mi pomůže. Také bych chtěl velmi poděkovat celé kanceláři A-1051 na Fakultě Informačních technologií v první řadě s mým vedoucím Ing. Ivo Hálečkem. V neposlední řadě Gabče "Unicorn" Hánové, jelikož je to osoba, díky které jsem v době psaní pořád ještě studentem FITu a která je jednou z osob mně nejbližších. Nemohu opomenout ani kladenskou Sluneční čajovnu, které patří dík především za její skvělou atmosféru, která mi při psaní této práce velmi pomohla. Poděkování patří také mé milé medicce, Pavlíně Muchové.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 15. května 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Janusz Piotr Wijas. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Wijas, Janusz Piotr. *Firmware over the air (FOTA) pro přípravek Arduino*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017. Dostupný také z WWW: (<https://gitlab.fit.cvut.cz/wijasjan/BI-BAK>).

Abstrakt

Tato bakalářská práce se zabývá problematikou vzdáleného nahrávání firm-ware do přípravku Arduino. Rozebírá vlastnosti a využití Wi-Fi a Bluetooth a zkoumá jejich zapojení na deskách Arduino Uno a Arduino Mega ADK. Zjiš-ťuje problémy při různých konfiguracích a poukazuje na možné důvody jejich vzniku.

Klíčová slova Arduino, Wi-Fi, Bluetooth, Firmware, Zavaděč, ESP8266, HC-06, Sériová linka

Abstract

This bachelor thesis focuses on distant bootloading of firmware into Arduino board. Describes properties and utilization of Wi-Fi and Bluetooth, looks into their plug in on Arduino Uno and Arduino Mega ADK boards. In this thesis we point problems with some configurations and we show reasons of their appearances.

Keywords Arduino, Wi-Fi, Bluetooth, Firmware, Bootloader, ESP8266, HC-06, Serial

Obsah

Úvod	1
1 Rešerše	3
1.1 Arduino	3
1.2 Bluetooth	7
1.3 Wi-Fi	9
1.4 Komunikační příkazy Hayes	10
1.5 Program Avrdude, používaný pro nahrávání	11
2 Implementace	13
2.1 Komunikace s HC-06 Bluetooth	13
2.2 Nahrávání programu pomocí Bluetooth	16
2.3 Komunikace s Arduinem skrze Wi-Fi	17
2.4 Nahrávání programu pomocí Wi-Fi	19
2.5 Shrnutí	24
3 Testování	25
3.1 Testování komunikace s Bluetooth modulem HC-06	25
3.2 Testování komunikace skrze Wi-Fi s modulem NodeMCU	26
3.3 Testování nahrávání programu do Arduino Uno skrze Wi-Fi modul NodeMCU	26
Závěr	29
Literatura	31
A Seznam použitých zkratk	35
B Uživatelská příručka pro nahrávání skrze Wi-fi	37
B.1 Technické požadavky	37

B.2	Postup při nahrávání programu do Arduino Uno	37
C	Obsah přiloženého CD	39

Seznam obrázků

1.1	Arduino Uno	4
1.2	Arduino Mega	5
1.3	Arduino Esplora	6
1.4	HC-06 Bluetooth modul	8
1.5	NodeMCU pinout	10
1.6	AT komunikace	11
2.1	Propojení Arduino Uno a HC-06	14
2.2	FT232R převodník	15
2.3	Arduino Mega s pripojenym NodeMCU	19
2.4	Hlavní stránka ESP-Link	21
2.5	Výpis Avrdude při nepovedeném nahrávání	22
2.6	Výpis programu Avrdude po úspěšném nahrání programu do pří- pravku	23
3.1	Zapojení pro testování	27
B.1	Arduino Uno s pripojenym NodeMCU	38

Úvod

V posledních letech je velmi rozšířený trend ve vytváření takzvaných inteligentních domácností. Jedná se o propojení rozličných zařízení do sítě a automatizaci jejich fungování. Příkladem může být připojení termostatu ve vinném sklípku, v kuchyni nebo garáži a následné automatické kontrolování teplot a vlhkostí vzduchu.

Problém nastává ve chvíli, kdy chceme provádět úpravy firmware, ať už aktualizace nebo úplné změny. Můžeme zajít pro zařízení, odpojit, zanést do pracovny, připojit kabelem k počítači, aktualizovat firmware, nasadit zpět na místo a následně zjistit jestli funguje jak jsme chtěli.

Druhou možností je připojit se z skrze Wi-Fi k vývojové desce a přeposlat aktualizaci během několika vteřin.

Tato bakalářská práce se zabývá programováním vývojové desky Arduino Uno bezdrátově. K tomu to účelu využívá komunikace skrze Bluetooth a Wi-Fi. Arduino je otevřená platforma nacházející využití v mnoha oblastech. Mezi tyto oblasti patří 3D tisk, Internet of Things (IoT) nebo seznámení studentů s programováním vestavěných systému, jako na předmětu BI-ARD na FITu.

V kapitole 1 se věnuju popisu jednotlivých prostředků potřebných k vyřešení této bakalářské práce. V sekcích postupně popisují platformu Arduino, Bluetooth, Wi-Fi, komunikaci pomocí Hayes příkazů a program pro nahrávání programu do procesorů – Avrdude.

V kapitole 2 popisují vlastní postup práce. Nejprve v sekci 2.1 zkoumám komunikaci pomocí Bluetooth a poté v sekci 2.2 popisují způsoby nahrávání programu do přípravku pomocí Bluetooth. V sekci 2.3 zkoumám komunikaci skrze Wi-Fi a následně v 2.4 popisují a implementují nahrávání programu do vývojové desky pomocí Wi-Fi komunikace.

Nakonec v kapitole 3 popisují testování funkčnosti implementovaného řešení.

Všechny překlady v textu jsou vytvořeny autorem, není-li řečeno jinak.

Rešerše

V této kapitole nejprve nastíním vznik jednotlivých prostředků použitých v této bakalářské práci. Následně uvedu jejich vlastnosti a možná využití. Jedná se o přípravky platformy Arduino, Wi-Fi a Bluetooth komunikaci spolu s odpovídajícími komponentami – HC-06 a ESP8266. Nazávěr zmíním komunikaci pomocí Hayes (AT) příkazů.

1.1 Arduino

Platforma Arduino začala vznikat v roce 2005 na italské univerzitě Interaction Design Institute Ivrea jako jednoduchá a levná platforma pro studenty, kteří s vývojem hardware nebo softwaru neměli zkušenosti [1]. Platforma se velmi rychle uchytila a začala se rapidně vyvíjet a adaptovat. Z původně 8-bitové platformy vznikly i 16-bitové verze desek pro Internet věcí, 3D tisk a mnohé jiné.

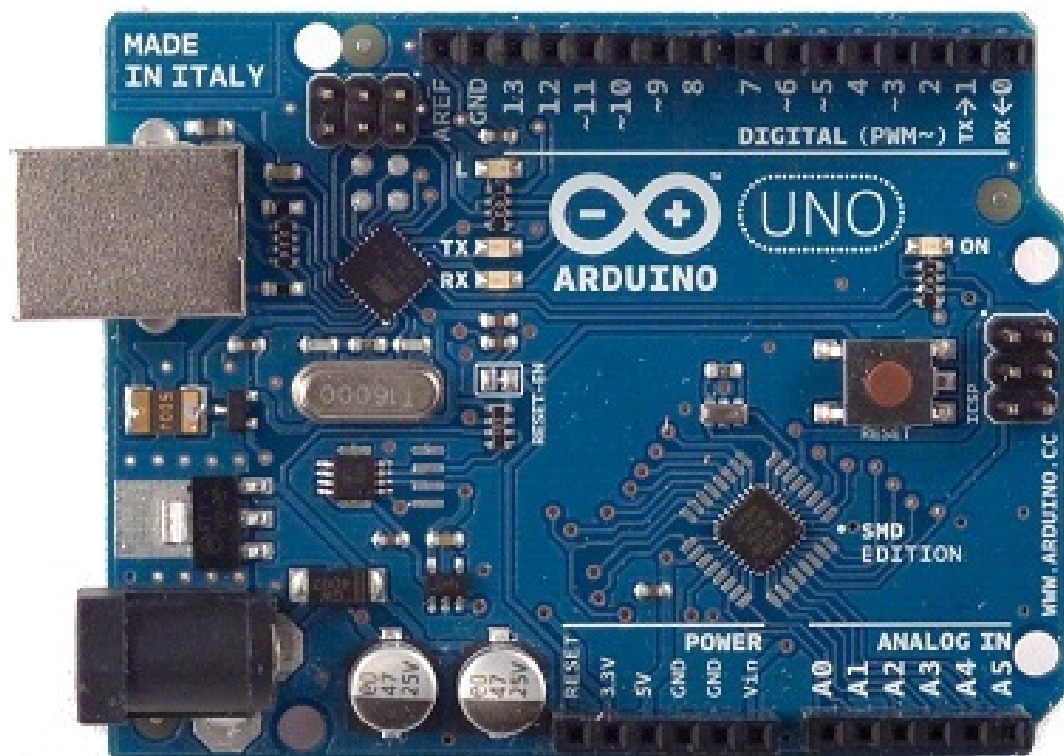
Platforma Arduino je vyvíjena pod Creative Commons Attribution Share-Alike 3.0. licencí. Znamená to, že je možno používat platformu Arduino, ať už celou nebo její část, pro svojí práci pod podmínkou zachování šíření své práce pod stejnou licencí.

V současné době jsou přípravky platformy Arduino, pokud započítáme i neoficiální, ale legální kopie, vyráběno v desítkách modelů. Mnohé z nich jsou „pouhé“ procesory s vyvedenými piny a lze k nim přidávat periférie pomocí shieldu [2].

Dále představím přípravky platformy Arduino, a to Arduino Uno, Arduino Mega a Arduino Esplora, poté vysvětlím funkci shieldu a na závěr popíšu důležitost zavaděčů.

1.1.1 Arduino Uno

Arduino Uno (viz obrázek 1.1) je vývojovou deskou platformy Arduino, která dává mnoho prostoru pro uživatelské úpravy, a zároveň je možné na ní velmi



Obrázek 1.1: Na obrázku je vidět přípravek Arduino Uno rev3. Lze zpozorovat v pravé dolní části procesor ATmega328p, na spodním i horním okraji přístupné patice pinů a v levé části USB konektor spolu s převodníkem UART – serial. Obrázek převzat z [3]

jednoduše připojit standardizované shieldy.

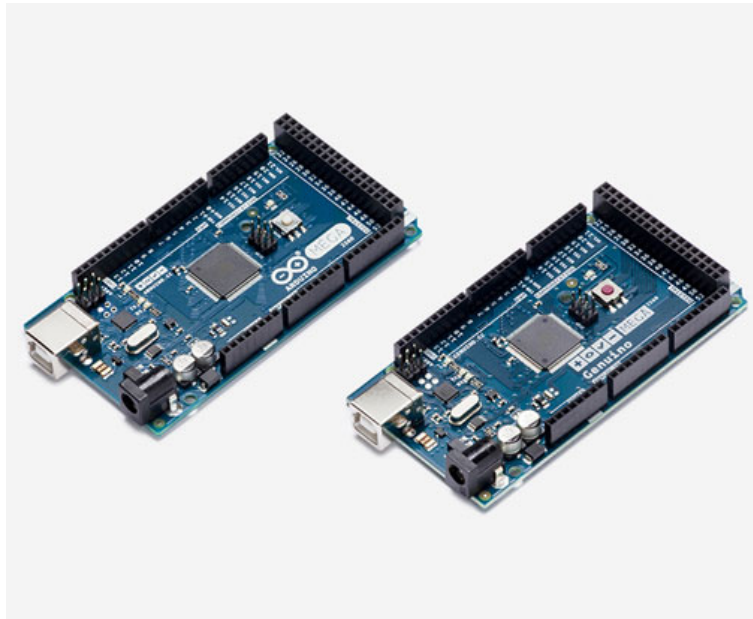
V době psaní této bakalářské práce je oficiální cena této vývojové desky 600 Kč bez DPH.

1.1.1.1 Technické údaje

Arduino Uno je mikropočítač založený na Atmel procesoru ATmega328 a v největších verzích 328p. Jako u většiny opravdových přípravku Arduino jedná se o 8-bitový procesor. Tato vývojová deska pracuje na kmitočtu 16 MHz, poskytuje 3 kB Flash paměti, 1 kB EEPROM a 2 kB RAM [4]. Na těchto deskách se nachází pouze dvacet vstupně-výstupních pinů, LED a převodník mezi USART a seriovou linkou.

1.1.2 Arduino Mega

Arduino Mega je vývojová deska platformy Arduino, která za cenu větší desky dává k dispozici více možností. Jedná se o desku využívanou pro rozsáhlejší



Obrázek 1.2: Přípravek Arduino Mega 2560 vlevo a Genuino Mega 2560 vpravo. Na obou vývojových deskách je vidět v centrální části procesor ATmega 2560. V porovnání s přípravkem Arduino Uno, obrázek 1.1, je na těchto deskách možno zpozorovat skoro čtyřikrát více dostupných pinů. Tento procesor je opět 8-bitový. Obrázek je k nalezení zde [5]

projekty, ve kterých je využíváných více sériových linek, jsou v nich rozsáhlejší programy nebo je nutné připojit mnoho periférií.

V době psaní této bakalářské práce je oficiální cena této vývojové desky 900 Kč bez DPH.

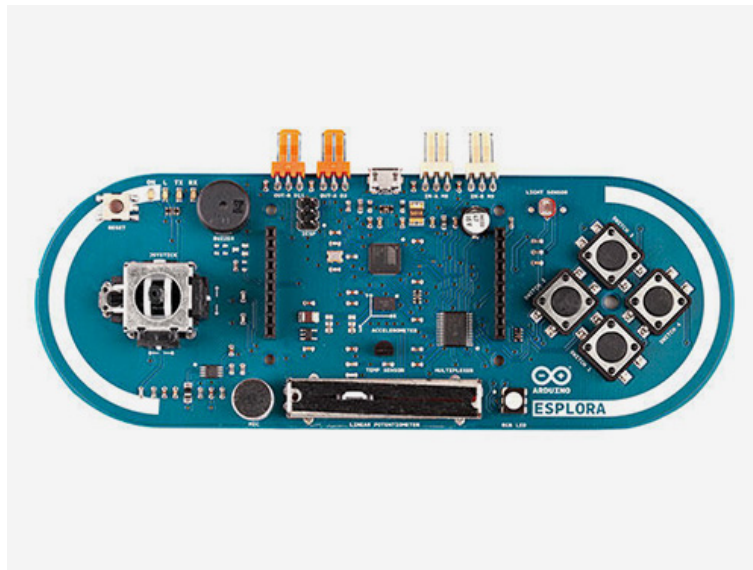
1.1.2.1 Technické údaje

Tento přípravek je založený na procesoru ATmega2560[6]. Pracuje na hodinovém knitočtu 16 MHz a poskytuje 4 kB EEPROM, 8 kB RAM a až 256 kB Flash paměti. Je na něm k dispozici 70 pinů, LED a až čtyři připravené hardware sériové linky[7].

1.1.3 Arduino Esplora

Arduino Esplora je přípravek, na rozdíl od výše uvedených, se zabudovanými perifériemi a kromě pinů pro sériovou komunikaci a patice pro zasunutí obrazovky, žádné volné piny nemá. Je to tedy deska vytvořena hlavně pro začátečníky nebo umělce.

V době psaní této bakalářské práce je oficiální cena této vývojové desky 1300 Kč bez DPH.



Obrázek 1.3: Na obrázku se nachází přípravek Arduino Esplora. Je možné zpozorovat patici pinů připravenou pro jednoduché nasazení displaye. V pravé části jsou vidět čtyři tlačítka v horní části světloměr a v dolní RGB LED, v levé – joystick, piezzo měnič (nahore) a mikrofon(dole). V prostřední části se kromě posuvného potenciometru nacházejí i teploměr, akcelerometr a nad ním procesor ATmega32u4. Obrázek pochází z [8]

1.1.3.1 Technické údaje

Arduino Esplora je deska založena na procesoru ATmega32u4, tento procesor má vestavěný převodník mezi rozhraním USART a sériovou linkou, 2,5 kB RAM, 1 kB EEPROM, a 32 kB Flash paměti. Pracuje na hodinovém kmitočtu 16 MHz [9].

Na desce jsou k dispozici konkrétně tyto periférie:

- 4 tlačítka
- analogový joystick
- posuvný potenciometr
- mikrofon
- teploměr
- RGB LED
- piezzo měnič

1.1.4 Shield

Shield je plošný spoj s připájenými součástkami a patící pinů standardizovaně umístěnou tak, aby shield šel jednoduše napojit na přípravek Arduino Uno, pro jenž je primárně určen.

Shieldy umožňují přidat velmi jednoduchým způsobem do vytvářeného přípravku další rozšíření bez nutnosti pájení nebo propojování jednotlivých součástek kabely. Piny shieldu přesně sedí na piny na přípravku Arduino Uno a zjednodušují připojování periférií. Mimo přímého připojení modulu, je tak přidání shieldů možností, jak vývojovou desku platformy Arduino rozšířit o pomocné periférie, nebo způsobem, jak rychle měnit celé sady těchto periférií. Tímto způsobem lze připojit například Bluetooth, Wi-Fi, ale i další procesor s VGA výstupem nebo čtečku SD karet či LED [10].

1.1.5 Zavaděč

Pro tuto práci je asi nejzásadnější částí přípravku Arduino jejich bootloader (neboli zavaděč). Jedná se o malý program, který se spouští před startem uživatelské aplikace a zajišťuje aktualizace firmware v paměti přípravku. Tento zavaděč v případě potřeby může celou instrukční paměť nahradit novým programem.

Během svého běhu naslouchá, zda na seriové lince nebyla poslána žádost o synchronizaci. Pokud ano, sesynchronizuje se s programem na druhé straně, například s programem Avrdude, spustí svoji sekvenci zapisování a instrukce, které obdrží, uloží do paměti. Po dokončení zapisování všech instrukcí restartuje přípravek.

Pokud na vstupu nečeká žádost o synchronizaci, zapisovací sekvence se nespustí a přípravek pokračuje ve vykonávání instrukcí, které už v paměti obsahoval[11].

Zavaděče platformy Arduino se během vývoje desek také měnily. V porovnání s počátečními verzemi, zabírají několikrát méně místa v paměti. Důsledkem je zvětšení maximální velikosti uživatelsky nahraného programu. Například oproti svému předchůdci je verze Optiboot, která je aktuálně používaná, až čtyřikrát menší, tedy uvolňuje uživateli přibližně 1,5 kB místa v paměti.

1.2 Bluetooth

Technologie Bluetooth vznikla koncem 90. let 20. století na podnět pěti firem, které založily Bluetooth skupinu SIG (Special Interest Group). Účelem Bluetooth SIG byla komunikace mezi mobilními telefony a jinými zařízeními bez nutnosti použití kabelu. Skupina Bluetooth SIG zveřejnila první verzi Bluetooth specifikace v roce 1999 [12].

Základní jednotkou Bluetooth je takzvaná pikosít, která se skládá z jednoho řídicího (master) uzlu, a maximálně sedmi aktivních klient (slave) uzlů.



Obrázek 1.4: Na obrázku je znázorněna spodní strana Bluetooth modulu HC-06. Jsou zde vidět dva vyvedené piny pro sériovou komunikaci, napájení a zem. Modul je napájen na stejné desce jako HC-05 a proto zde jsou dva nepoužité místa pro piny. Obrázek je k nalezení zde [14]

Navíc v každé síti může být až 255 neaktivních uzlů, které ale mohou být v synchronizaci s master uzlem a pouze čekají na pokyn k aktivaci. V jedné místnosti se tak může nacházet mnoho pikosítí, které mohou být propojeny pomocí slavy uzlu[13].

1.2.1 Architektura Bluetooth

Protokoly systémů Bluetooth dodnes neodpovídají žádnému běžně používanému modelu (OSI, TCP/IP nebo 802.11), nicméně IEEE vyvíjí úsilí nad modifikací Bluetooth tak, aby pasoval na standard 802.

Jedná se o systém s velmi nízkým příkonem. Podle výkonu se rozděluje Bluetooth na tyto kategorie dosahu:

třída 1. (100 mW) nejdelší dosah, teoreticky až 100 m

třída 2. (2,5 mW) nejčastěji využívaná třída s teoretickým dosahem až 10 m

třída 3. (1 mW) nejméně využívaná, dosah teoreticky 1 m

1.2.2 HC-06

HC-06 je levný Bluetooth chip, který je vztvořen pouze pro fungování v módu slavy. Je založen na modulu BC04-B, který obsluhuje Bluetooth verze 2.1 [15]. Vzhledem k jeho ceně, která v době psaní této bakalářské práce činila přibližně 90 Kč bez DPH, je velmi rozšířený v IoT (Internet věcí). Na obrázku 1.4 je zobrazen modul HC-06. Na této komponentě se krom samotného chipu nachází LED a několik odporů a diod, sloužících především k přizpůsobení napětových úrovní tak, aby bylo možné tento modul použít i s 5 V jako například v platformě Arduino. HC-06 je vlastně převodník mezi Bluetooth a sériovou linkou a vše kromě AT příkazů přeposílá na druhou stranu.

1.3 Wi-Fi

V této sekci poskytnu základní informace o Wi-Fi komunikaci, dále přípravky ESP8266, ze kterých vyberu dva kandidáty, které použiji pro svojí implementaci.

1.3.1 Standard 802.11

Standard 802.11 obsahuje skupinu protokolů IEEE popisujících fyzickou vrstvu a MAC. Obecně existují čtyři nezávislé protokoly kódování, a to a , b , g , n . Ty se dále kombinují s protokoly šifrování (standard i) a jinými rozšiřujícími standardy. Spektrum frekvencí používaných těmito protokoly je od 2,401 GHz do 2,495 GHz. Toto spektrum je ve své nejrozšířenější verzi (*802.11b*) rozděleno na 14 kanálů, nicméně pouze tři z nich se navzájem nepřekrývají. Pro rozlišení požadovaných dat od šumu vytvářeného například jinými vysílači, pracujícími na částečně se překrývajících kanálech, se používá kvadratická modulace. Vzhledem k tomu, že pro využití frekvencí, které využívá tento standard, nejsou potřeba licence, je možné instalovat Wi-Fi sítě bez obtíží. Nicméně na stejném pásmu můžou fungovat i mikrovlnné trouby a jiné přístroje, je tedy třeba počítat s rušením.

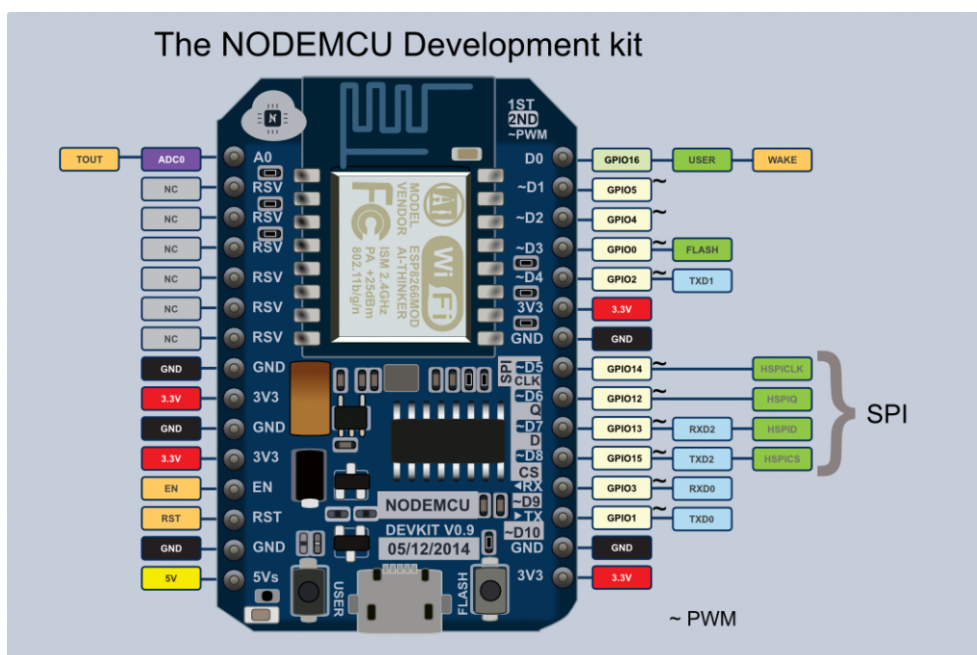
1.3.2 ESP8266

ESP8266 je pravděpodobně nejrozšířeněji používaný Wi-Fi chip používaný v Internetu věcí. Jeho nespornou výhodou je jeho cena, v době psaní této bakalářské práce se pohybuje kolem 60 Kč za verzi modulu 01 nebo 90 Kč za NodeMCU devkit. Výpočetní síla chipu, která je větší než u procesoru nacházejícího se v Arduino Uno, je dalším důvodem uživatelské oblíbenosti. Reálná rychlost chipu je snížena z důvodu vyhrazení části chipu pouze pro bezdrátovou komunikaci. V následujícím textu popíšu kandidáty na použití Wi-Fi modulu do mé bakalářské práce a to ESP-01 a NodeMCU, založený na ESP-12.

1.3.2.1 ESP-01

V době psaní této bakalářské práce existovalo 16 verzí těchto modulů. Pro účely této práce byla k dispozici verze 01 a verze 12 zabudovaná na NodeMCU devkitu. Pro tuto bakalářskou práci byl zaveden předpoklad, že verze 01 bude vzhledem ke svým parametrům naprosto dostačující. Tento předpoklad byl založen na následujících faktech.

- Obsahuje 6 aktivních pinů, plánované použití bylo 2 pro sériovou komunikaci a jeden pro resetování přípravku Arduino Uno.
- Informativní LED



Obrázek 1.5: Na tomto obrázku je znázorněn NodeMCU Development kit V0.9. Je zde vidět GPIO piny připojené k jednotlivým výtupům, a označeny jejich základní funkce. Obrázek je možné najít zde [16]

- Je přítomná anténa PCB trace, nebylo tedy nutné připojovat externí anténu jako v jiných verzích ESP8266.
- Obsahuje 64 kB instrukční RAM.

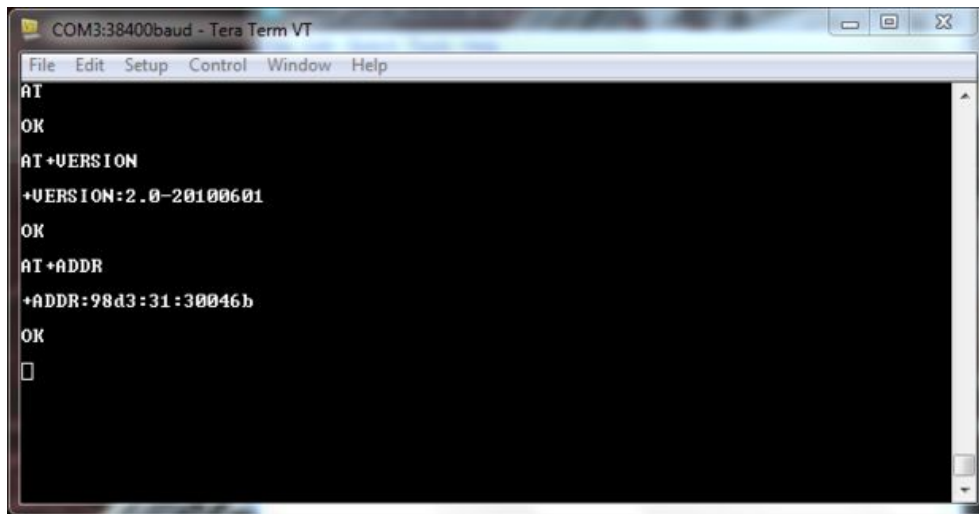
1.3.2.2 NodeMCU

Modul NodeMCU, obrázek 1.5, je open source platforma Internetu věcí. Jedná se o desku založenou na ESP-12. Jako hlavní rozdíl vnímám zvětšení počtu volných pinů z 6 na 14 a obdržení certifikátů FCC a CE. Jedná se o americký respektive evropský certifikát zaručující dodržení norem v daných oblastech.

Za pozornost stojí fakt, že tento modul je schopen používání MQTT protokolu Internetu věcí. Jedná se o ISO standart umožňující spojení pomocí TCP/IP protokolu za použití minimálního počtu dat, tedy lze jej použít i ve velmi omezených sítích.

1.4 Komunikační příkazy Hayes

Sada Hayes příkazů, současně známějších pod názvem AT příkazů, je standard pro komunikaci s modemy vytvořený v roce 1981 Dennisem Hayesem.



Obrázek 1.6: Příklad komunikace mezi Bluetooth chipem a počítačem pomocí Hayes příkazů. Jsou zde znázorněny dotazy na verzi firmware a adresu modulu. Obrázek je možno najít zde [17]

Funguje na principu posílání krátkých řetězců znaků začínajících prefixem AT (ATtention), odtud vzniknul název AT příkazy. Časem byly tyto příkazy rozšířeny a používají se dnes například pro komunikaci s Bluetooth nebo Wi-Fi moduly. Na obrázku 1.6 je zobrazen příklad komunikace mezi Bluetooth modulem a počítačem pomocí těchto příkazů. Je zřejmé, že každý příkaz začíná prefixem AT, poté následuje kombinace znaků vyžadující nějakou funkci.

Níže popis dotazů a odpovědí z obrázku 1.6:

AT Dotaz na spojení.

OK Odpověď označující, že příkaz proběhl v pořádku

AT+VERSION Dotaz na verzi firmware v chipu.

+VERSION:2.0-20100601 Verze firmware je *2.0-20100601*

OK Odpověď označující, že příkaz proběhl v pořádku

AT+ADDR Dotaz na výchozí adresu chipu.

+ADDR:98d3:31:30046b Adresa chipu je *98d3:31:30046b*

OK Odpověď označující, že příkaz proběhl v pořádku

1.5 Program Avrdude, používaný pro nahrávání

Avrdude je program umožňující nahrávání, modifikaci nebo stahování obsahu paměti ROM a EEPROM mikrokontrolerů rodiny AVR. Jedná se o aplikaci za-

loženou na projektu avrprog vytvořeném pro Atmel Brian S. Deanem. Kvůli rostoucího zájmu o kompatibilitu s jinými systémy byl zpřístupněn veřejně a název byl změněn na Avrdude. Obecně je používán například v Arduino-IDE jako fakticky programující část. Níže jsou vypsány důležité přepínače programu:

- p** je jediný povinný přepínač, určuje chip, který bude nahráván. V této bakalářské práci používám hodnoty m328p pro Arduino Uno a m2560 pro Arduino Mega.
- b** přepínač nastavující rychlost, s jakou se Avrdude komunikuje se zavaděčem v přípravku. V této práci je využita hodnota 115200 boudů.
- c** tento přepínač určuje jaký zavaděč je používán. V této práci jsou to *arduino* pro Arduino Uno a *wiring* pro Arduino Mega.
- P** je přepínač, který dovoluje programovat vzdáleně. Je možné ho nastavit na *net:IP:port*. což způsobí, že Avrdude bude vyhledávat zařízení a jeho zavaděč na cílové síťové adrese.
- U** specifikuje program, který do cílového přípravku Avrdude nahrává.

Pro více dostupných přepínačů a dokumentaci k Avrdude odkazují na[18].

Implementace

V této kapitole se věnuji komunikaci skrze Bluetooth s modulem HC-06 a následným použitím tohoto modulu při nahrávání programu do vývojové desky rodiny Arduino. Poté popisuji komunikaci skrze Wi-Fi s chipem ESP8266 verze ESP-12 umístěným na vývojovém kitu NodeMCU. Na závěr implementuji programování skrze Wi-Fi pomocí modulu NodeMCU a firmware ESP-Link.

2.1 Komunikace s HC-06 Bluetooth

Prvním krokem v praktické části bylo zapojení Bluetooth modulu k vývojové desce Arduino Uno a následné vyzkoušení komunikace. Pro tuto část byl k dispozici Bluetooth chip HC-06 spolu s Arduino Uno. Dokumentaci k HC-06 je k nalezení v [19]. Pro zapojení a komunikaci byl použit návod nalezitelný v [20]. Níže jsou uvedeny vybrané důležité příkazy:

AT základní příkaz, který pouze zkoumá zda modul komunikuje

AT+NAME příkaz vypisující název pod kterým se modul zobrazuje v síti

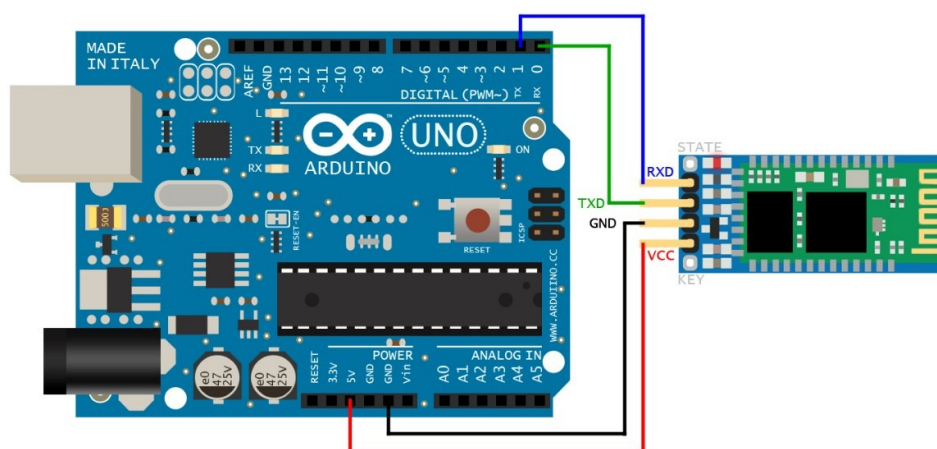
AT+RESET příkaz resetující modul

AT+PSWD příkaz zobrazující heslo modulu

V případě *AT+NAME* a *AT+PSWD* lze na konec přidat *=name* respektive *=password* a tímto změnit jméno/heslo modulu.

Návod [20] obsahuje informaci, že HC-06 je v AT modu vždy, pokud není ve spojení s jiným zařízením. Tento modul funguje po sériové lince na základní komunikační rychlosti 38400 baudů s ukončením řádku NL & CR.

2. IMPLEMENTACE



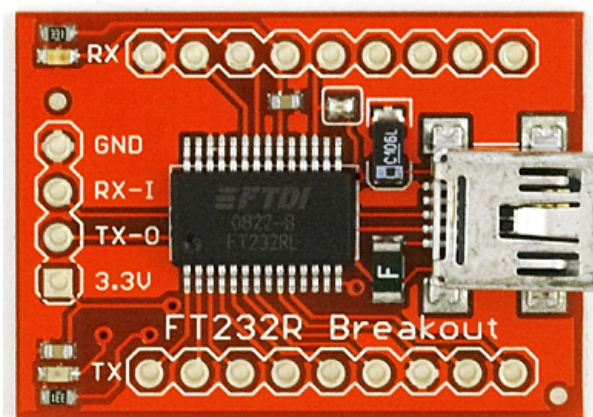
Obrázek 2.1: Ukázka připojení Bluetooth modulu HC-06 na Arduino Uno. Ke správnému fungování je nutné připojit Rx a Tx piny modulu na Tx, respektive Rx, piny vývojové desky Arduino Uno a propojit napájení a zem v případě napájení modulu z přípravku Arduino Uno. Obrázek je možné najít zde [21]

2.1.1 Zapojení

Na obrázku 2.1 je zobrazeno zapojení modulu do přípravku. Ke správnému fungování je nutné připojit Tx Bluetooth modulu k Rx Arduino Uno, Rx k Tx a propojit jejich napájení a země. Po zapojení, při pokusu o nahrání do Arduino Uno základního programu se ukázalo, že pokud je na seriové lince při pokusu o nahrávání připojen Bluetooth modul, pak se s přípravkem nemůže flashovací program Avrdude spojit. Jedinou možností je modul odpojit.

2.1.2 SoftwareSerial

Pro komunikaci s Bluetooth module jsem použil SoftwareSerial knihovnu dodávanou spolu s ArduinoIDE. Nicméně data, která byla přijímána z přípravku Arduino Uno, byla viditelně poškozená (došlo k modifikaci dat po kvalitativní i kvantitativní stránce). Vzhledem k tomu, že opětovné nahrání programu na mikrokontroler nemělo na tok dat pozitivní efekt, rozhodl jsem se pro výměnu Arduino Uno za jiný kus, pro vyloučení případného poškození používaného Arduino Uno. Ani tento manévř nevedl k vyřešení problému. Při konzultaci s odbornými zdroji jsem narazil na přednášku Ing. Petra Stehlíka[22] z Install-Fest 2016, kde popisuje, že tento problém způsobuje knihovna SoftwareSerial, a i když se pokoušel, nedokázal jí opravit. Ukázalo se, že tato knihovna má problémy, pokud je používána zároveň softwarová a hardwarová seriová linka. Dochází k poškození dat a tudíž je tato kombinace naprosto nepoužitelná. K obejití komplikace jsem použil převodník mezi sériovou linkou a uartem.



Obrázek 2.2: Převodník FT232R Breakout[23]

Rovněž jsme použil Arduino Mega, a to z důvodu, že má vestavěné čtyři sériové linky, je velmi vhodná právě v případech, kdy je nutností komunikace několika zařízení.

2.1.3 UART – Serial převodník

Převodník je elektronická součástka, která mění v tomto případě signál z UART na signál sériové linky. Využil jsem převodník FT232R, který je zobrazen na obrázku 2.2.

2.1.3.1 Zapojení

Zapojit modul do převodníku je v principu stejné jako zapojit jej k Arduino Uno. Je tedy nutné propojit Tx modulu k Rx pinu na převodníku a opačně, propojit země a napájení.

2.1.3.2 Komunikace

Po propojení převodníku s modulem a následním připojením převodníku s počítačem v skrze PuTTY bylo možné s Bluetooth modulem komunikovat. Výše popsanými příkazy jsem dokázal změnit základní nastavení, tedy název a připojovací heslo. V mobilním telefonu jsem modul vyhledal a spároval se s ním, nicméně vzhledem k nekompatibilitě mobilního telefonu s aplikacema pro ovládání Bluetooth jsem více udělat nedokázal. Důvodem byla zastaralá verze systému Androidu na mém mobilním telefonu (Android 2.3.6).

2.1.4 Arduino Mega

Zapojení s Arduino Mega bylo stejné jako s Arduino Uno, s tím rozdílem, že místo zapojování Tx a Rx pinu modulu do Rx a Tx desky, zapojil jsem je do Rx3 a Tx3 pinů. I v tomto případě se komunikace zdařila a já tedy dokázal mobilní telefon s Bluetooth modulem spárovat.

2.2 Nahrávání programu pomocí Bluetooth

V tuto chvíli bylo jisté, že přeposílat data je možné a tedy, že by byla možnost pokusit se přeposlat celý program. Z počítače pomocí programu avrdude, který se stará o nahrávání programů do procesorů, je možné program nahrát. Nicméně teprve v tuto chvíli bylo zjištěno, že HC-06 nemá volný GPIO (General Purpose Input/Output)¹ pin. Tento pin byl potřeba k automatizovanému resetu přípravku nutnému pro start zavaděče.

2.2.1 Ruční reset

První možností by bylo resetovat přípravek ručně, což znamená, že při každé aktualizaci firmware v přípravku by bylo nutné k přípravku dojít, ztlachtit reset a až poté nahrát program. To je ovšem v rozporu s myšlenkou této bakalářské práce. Není sice v tomto případě nutné přípravek přenášet k počítači nebo notebook k přípravku, ale jedná se o polovičaté řešení.

2.2.2 Inbuild function

Druhým možným řešením je vytvořit v programu, který je už nahrán do přípravku, příkaz, který po zaslání umožní, aby se mikrokontroler Arduino sám restartoval. Tato možnost je rozhodně vhodnější pro účely této práce, nicméně obsahuje několik problémů. Například pokud v jakékoli verzi firmware opomeneme danou funkci, nebo se daná část kódu nevykoná, ať už kvůli zacyklení či špatně vykonávané podmínce, pak bude opět nutné přípravek nahrát pomocí kabelu, eventuálně resetovat ručně a poté nahrát. Zároveň tento kus kódu zabírá místo pro cílový program. Je také nutné jeden z výstupních pinů vyhradit čistě pro spojení s pinem reset a pomocí tohoto pinu přípravek resetovat.

2.2.3 Arduino jako programátor

Do třetice je možné použít dvě vývojové desky Arduino. Pro minimalizaci nákladů a místa lze kromě programovaného přípravku použít Arduino Nano nebo Arduino Mini. Na tento přípravek je možné napojit Bluetooth modul a po zadání správného příkazu skrze Bluetooth k přídatnému zařízení resetovat a nahrát do cílového mikrokontroleru požadovaný program. Komplikací se

¹Všeobecně použitelný vstupně-výstupní

objevuje opět několik, tato kombinace je téměř dvakrát dražší, jedna deska Arduino je vlastně nepoužívaná, důležitým faktorem je také zvýšená spotřeba energie a větší požadavky na prostor. Z výše uvedených možností je to sice nejelegantnější varianta, nicméně stále se nejedná o možnost, kterou bychom si přáli a chtěli používat.

2.3 Komunikace s Arduinem skrze Wi-Fi

Pro tuto část jsem měl k dispozici dva ESP8266 chipy. Jednalo se o dvě různé verze. První modul obsahoval verzi ESP-01. Bohužel s tímto konkrétním chipem jsem se nedokázal spojit vzhledem ke konstrukční vadě.

Druhým modulem byla, zabudovaná na NodeMCU devkit-u, ESP8266 verze ESP-12[24]. S tímto chipem komunikace od začátku probíhala bez problémů. Poučen z problémů z Bluetooth chipem jsem použil od začátku Arduino Mega a převodník.

2.3.1 Zapojení

NodeMCU jsem připojil do nepájivého pole. Piny Arduino Mega (Tx, Rx, 5 V, zem) jsem propojil skrze nepájivé pole kabely s piny Wi-Fi modulu. Použil jsem opět Arduino Mega jako převodník s nastavenou přenosovou rychlostí 115200 baudů pro komunikaci s Wi-Fi a 9600 baudů pro komunikaci s počítačem.

2.3.2 AT příkazy

Použil jsem seznam AT příkazů [25]. Níže jsou vypsány nejdůležitější příkazy, které byly použity. Přidáním otazníku na konec řádku umožňuje vypsání aktuálního stavu.

AT Příkaz pro vyzkoumání konektivity

AT+RST Způsobuje restart Wi-Fi chipu

AT+CWMODE=<mode> Příkaz sloužící k nastavení módu Wi-Fi modulu. Možné mody:

- 1 - Station
- 2 - Přístupový bod
- 3 - Kombinace obojího

AT+CWLAP Příkaz žádající o sken okolních přístupových bodů Wi-Fi a vypsání po seriové lince

AT+CWJAP=<accesspointname>,<password> Příkaz připojující chip k zadanému přístupovému bodu

AT+CIFSR Příkaz vypisující aktuální IP adresu modulu

AT+CWSAP=„Sinewa“,„Sinewa“,3,0 Příkaz nastavující parametry přístupového bodu. V tomto případě s názvem Sinewa, heslem Sinewa, na 3 kanálu, bez šifrování

AT+CIPSTA=„192.168.0.100“,„192.168.0.1“,„255.255.255.0” Příkaz nastavující statickou IP adresu, bránu a masku pro Wi-Fi chip. Tento příkaz se nenachází ve výše uvedeném pdf souboru.

AT+CIPSEND=„<id>“,„<length>“ Nasledujících *length* bajtů bude poslano na daný channel

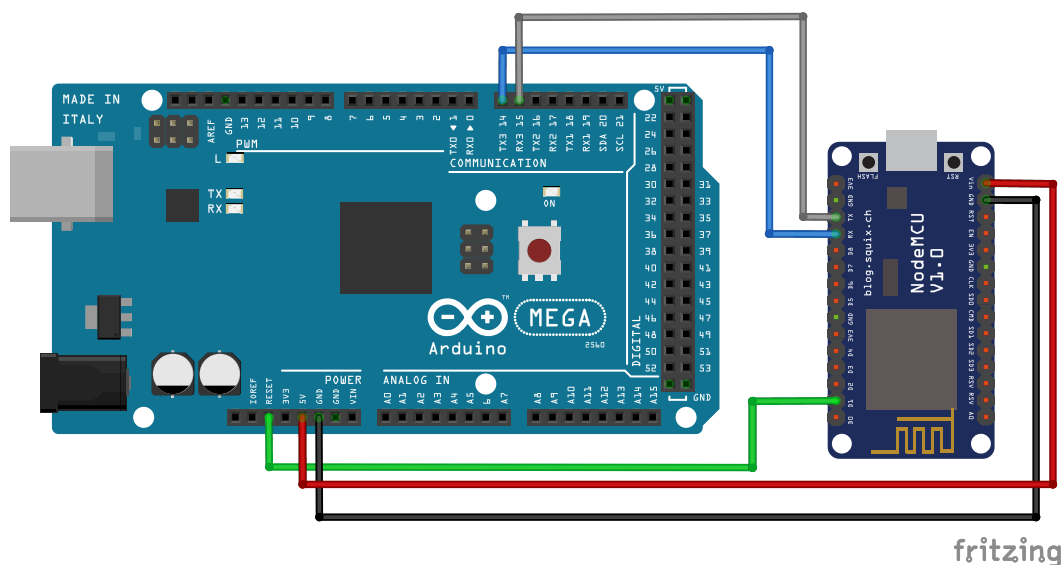
+IPD, <id>, <len> Nasledujících *len* bajtů bude očekáváno na channelu *id* a přeposláno po sériové lince

2.3.3 Komunikace v módu AP

Tento modul od začátku velmi dobře odpovídal na AT příkazy, vytvoření AP, nebo-li Access Point, česky přístupového bodu, bylo tedy záležitostí několika minut. Použil jsem příkazy *AT+CWMODE=2*, pro nastavení AP módu, *AT+RST* pro restart, a pro nastavení parametrů tohoto bodu *AT+CWSAP*. Následně jsem se na utvořený přístupový bod připojil počítačem. V tuto chvíli jsem byl skrze seriovou linku informován, že bylo připojeno zařízení. Spustil jsem na počítači spojení telnetem s adresou přístupového bodu a skrze sériovou linku přeposlal příkazem *AT+CIPSEND* několik znaků. Znaky se úspěšně objevily v PuTTY telnetu. Komunikace tedy funguje. Poté jsem vyzkoušel komunikaci ve druhém směru, tedy z počítače skrze Wi-Fi. Použil jsem tedy příkaz *+IPD*. I v tomto směru komunikace fungovala. Tento pokus dopadl úspěšně.

2.3.4 Komunikace v módu STA

Následně jsem vyzkoušel komunikaci s modulem v módu STA, nebo-li Station případně Klient. Použitím příkazu *AT+CWMODE=1* jsem dostal modul do tohoto módu a po resetování jsem příkazem *AT+CWLAP* vypsal všechny dostupné AP. Vybral jsem domovní síť, kterou jsem měl plně pod kontrolou a pokusil se připojit příkazem *AT+CWJAP*. Nicméně tento příkaz vracel neúspěch. Nepomohla změna sítě ani odheslování. Ukázalo se, že problém byl v dynamickém získávání IP adresy na některých rozmezích adres. Nejjednodušším řešením bylo nastavení IP adresy staticky. Je k tomu nutné znát vlastnosti sítě. Přesněji je potřeba vědět, která adresa je volná, bránu sítě a její masku. Pro nastavení výše uvedených parametrů jsem použil příkaz *AT+CIPSTA*. Po této operaci se modul dokázal k síti připojit. Abych se ale připojil z počítače k tomuto modulu, bylo nutné nastavit modul jako server



Obrázek 2.3: Na obrázku se nachází Arduino Mega s připojeným NodeMCU. Pro správné fungování je nutné propojit Rx a Tx piny Arduino Mega s Tx, respektive Rx, piny modulu NodeMCU, RST vývojové desky na pin NodeMCU a připojit napájení a zem.

příkazem `AT+CIPSERVER=1,1336`, kde první číslo z rozmezí 0-3 uvádí kanál, na kterém má být vytvořen na ESP, a druhé číslo označuje port, na kterém má server naslouchat. Po této operaci opětovně příkazem `AT+CIPSEND` lze komunikovat z počítačem.

2.4 Nahrávání programu pomocí Wi-Fi

Dalším krokem bylo nahrávání programu skrze Wi-Fi. Toto by bylo možné docílit i například pomocí dvou přípravku Arduino Uno. Tuto možnost jsem zamítl již při práci s Bluetooth a tedy jsem použil Arduino Mega a NodeMCU.

Existuje program pro ESP - ESP-Link [26], který po nahrání vytváří interface pro komunikaci. Mimo jiné dovoluje nastavit, na kterých pinech má být používána sériová komunikace. Rovněž nastavuje použití konkrétního pinu pro reset MCU. Po stažení a rozbalení programu je nutné daný firmware do ESP nahrát, a to například pomocí pythonu příkazem:

```
python ../esptool-master/esptool.py --port COM4
--baud 115200 write_flash --flash_freq 80m --flash_mode qio
--flash_size 32m 0x0000 boot_v1.6.bin 0x1000 user1.bin
0x3FC000 esp_init_data_default.bin 0x3FE000 blank.bin
```

2. IMPLEMENTACE

Význam jednotlivých přepínačů:

python ../esptool-master/esptool.py jedná se o python skript a tedy cestu k němu

-port COM4 port na kterém je připojen modul NodeMCU k počítači

-baud 115200 rychlost kterou zapisujeme na ESP

write_flash účel scriptu

-flash_freq 80m rychlost SPI flashování

-flash_mode qio SPI mod flashování

-flash_size 32m velikost flash paměti

0x0000 boot_v1.6.bin na adresu 0x0000 zapiš soubor boot_v1.6.bin

0x1000 user1.bin na adresu 0x1000 zapiš soubor user1.bin

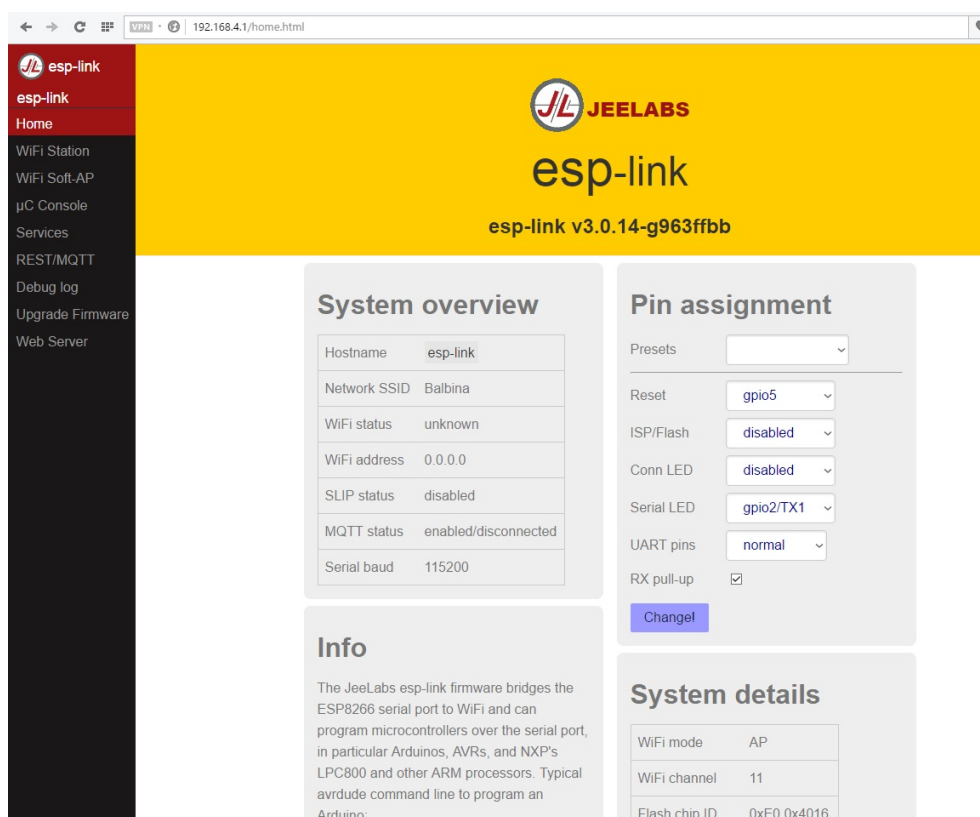
0x3FC000 esp_init_data_default.bin na adresu 0x3FC000 zapiš soubor esp_init_data_default.bin

0x3FE000 blank.bin na adresu 0x3FE000 zapiš soubor blank.bin

V případě úspěchu výstup bude přibližně takto:

```
WARNING: Flash size arguments in megabits like '32m' are deprecated.
Please use the equivalent size '4MB'.
Megabit arguments may be removed in a future release.
esptool.py v2.0-beta2
Connecting....
Detecting chip type... ESP8266
Uploading stub...
Running stub...
Stub running...
Attaching SPI flash...
Configuring flash size...
Flash params set to 0x004f
Compressed 3856 bytes to 2763...
Wrote 3856 bytes (2763 compressed) at 0x00000000 in 0.2 seconds...
Hash of data verified.
Compressed 330756 bytes to 244619...
Wrote 330756 bytes (244619 compressed) at 0x00001000 in 21.5 seconds...
Hash of data verified.
Compressed 128 bytes to 75...
Wrote 128 bytes (75 compressed) at 0x003fc000 in 0.0 seconds...
```

2.4. Nahrávání programu pomocí Wi-Fi



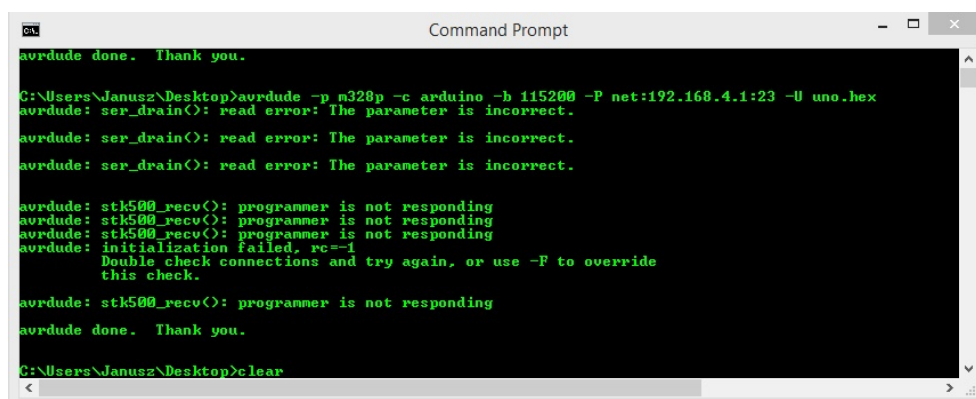
Obrázek 2.4: Na obrázku je vidět hlavní stránku ESP-Link, kterou lze zobrazit po připojení k ESP8266 skrze Wi-Fi, napravo v sekci *Pin assignment* je možné vybrat, na kterých pinech se nachází reset MCU, a na kterých je zapojená sériová komunikace.

```
Hash of data verified.  
Compressed 4096 bytes to 26...  
Wrote 4096 bytes (26 compressed) at 0x003fe000 in 0.0 seconds...  
Hash of data verified.
```

```
Leaving...  
Hard resetting...
```

Dalším krokem je do ESP-Linku nahrát soubor `user2.bin`. Ten je možno nahrát na webové stránce, standardně na adrese `192.168.4.1` po připojení k ESP skrze Wi-Fi. Není nutné zadávat přesné rozložení paměti, jediné v sekci *Upgrade Firmware* je možné soubor nahrát a potvrdit. Již v tuto chvíli je možné s deskou Arduino Mega nebo Arduino Uno komunikovat, například skrze konzoli vestavěnou ve webové aplikaci. K nahrávání souboru je nutné

2. IMPLEMENTACE



```
Command Prompt
avrdude done. Thank you.
C:\Users\Janusz\Desktop>avrdude -p m320p -c arduino -b 115200 -P net:192.168.4.1:23 -U uno.hex
avrdude: ser_drain(): read error: The parameter is incorrect.
avrdude: ser_drain(): read error: The parameter is incorrect.
avrdude: ser_drain(): read error: The parameter is incorrect.
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_recv(): programmer is not responding
avrdude: initialization failed, rc=-1
        Double check connections and try again, or use -F to override
        this check.
avrdude: stk500_recv(): programmer is not responding
avrdude done. Thank you.
C:\Users\Janusz\Desktop>clear
```

Obrázek 2.5: Na obrázku je vidět výpis programu Avrdude po nepovedeném připojení a nahrání na vývojovou desku Arduino Uno.

nastavit kromě Tx a Rx pinů, také pin, na kterém je připojen reset. Je zde možné rovněž nastavit, na kterých pinech se nacházejí informační LED (viz 2.4). Pro nahrávání programů na Arduino Mega je také nutné přepojit Tx a Rx NodeMCU modulu z Rx3 a Tx3 na Rx a Tx.

V tuto chvíli je vše nastaveno pro nahrávání, posledním článkem, který je nutný k úspěchu, je program Avrdude.

Avrdude má dostupný přepínač *-P*, kterým lze nastavit cílové zařízení. Tomuto parametru lze nastavit i hodnotu *net: „IP adresa“ : „port“*, kde v místě „IP adresa“ zadáme síťovou adresu ESP modulů a port, na kterém naslouchá. Standardně 23. Avrdude pak místo použití COM portu se pokouší nahrát program na cílovou síťovou adresu. Vytvořil jsem v programu ArduinoIDE jednoduchý testový program mající za úkol jediné rozsvícení LED. Po kompilaci jsem našel soubor *program.hex* a nakopíroval do složky s Avrdude. Připojil jsem počítač k přístupovému bodu sítě vytvořené na ESP a pokusil se o nahrání programu příkazem:

```
avrdude.exe -p m2560 -c wiring -b 115200 \
-P net:esp-link:23 -U program.hex
```

Nahrávání proběhlo neúspěšně, Avrdude oznámil, že takové zařízení nemůže najít. Změnil jsem tedy v příkazu název *esp-link* na síťovou adresu, ale i když Avrdude dokázal ESP najít, nahrávání proběhlo neúspěšně. Jednu z chybových hlášek je vidět zde 2.5. Proběhlo ještě několik pokusů, ale i přes obdržení známých chybových hlášek, jako například:

```
programmer does not respond
```

jsem program nahrát nedokázal.

```

z@janusz-Aspire-E1-572G: ~/Dokumenty/FIT/ard
janusz@janusz-Aspire-E1-572G:~/Dokumenty/FIT/ard$ avrdude -p m328p -c arduino -b
115200 -P net:192.168.4.1:23 -U BlinkRed.cpp.hex
^C
janusz@janusz-Aspire-E1-572G:~/Dokumenty/FIT/ard$ avrdude -p m328p -c arduino -b
115200 -P net:192.168.4.1:23 -U BlinkRed.cpp.hex
ioctl("TIOCMGET"): Inappropriate ioctl for device
ioctl("TIOCMGET"): Inappropriate ioctl for device

avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s

avrdude: Device signature = 0x1e950f
avrdude: NOTE: "flash" memory has been specified, an erase cycle will be perform
ed
      To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "BlinkRed.cpp.hex"
avrdude: input file BlinkRed.cpp.hex auto detected as Intel Hex
avrdude: writing flash (1034 bytes):

Writing | ##### | 100% 0.53s

avrdude: 1034 bytes of flash written
avrdude: verifying flash memory against BlinkRed.cpp.hex:
avrdude: load data flash data from input file BlinkRed.cpp.hex:
avrdude: input file BlinkRed.cpp.hex auto detected as Intel Hex
avrdude: input file BlinkRed.cpp.hex contains 1034 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 0.50s

avrdude: verifying ...
avrdude: 1034 bytes of flash verified

avrdude: safemode: Fuses OK (H:00, E:00, L:00)
ioctl("TIOCMGET"): Inappropriate ioctl for device

avrdude done. Thank you.

janusz@janusz-Aspire-E1-572G:~/Dokumenty/FIT/ard$ █

```

Obrázek 2.6: Na tomto obrázku e vidět výpis programu Avrdude po úspěšném nahrání skrze Wi-Fi na přípravek Arduino Uno. Je vidět, že po nahrání program paměť kontroluje.

Ukázalo se , že Avrdude není stoprocentně kompatibilní s operačním systémem Windows a přepínač `-P` v verzi pro Windows není plně naimplementován. Proto bylo nutné použít Unixový systém.

Použil jsem Ubuntu 14.04. Připojil jsem notebook k Wi-Fi síti a v terminálu spustil příkaz:

```

avrdude -p m2560 -c wiring -b 115200 \
-P net:192.168.4.1:23 -U program.hex

```

Program byl úspěšně nahrán na Arduino Mega. Výstup skriptu je vidět zde 2.6.

2. IMPLEMENTACE

Pro nahrávání na Arduino Uno je nutné příkaz upravit. Parametr *-p* označující cílový procesor je nutné nastavit na *m328p* a parametr *-c*, označující používaný programátor, na *arduino*. Tedy příkaz vypadá takto

```
avrdude -p m328p -c arduino -b 115200 \  
-P net:192.168.4.1:23 -U program.hex
```

Při změně vývojové desky je nutné pamatovat na překompilování programu pro vybraný přípravek.

2.5 Shrnutí

V této kapitole jsem popsal komunikaci a nahrávání programů do přípravku skrze Bluetooth a Wi-Fi. Zjistil jsem, že ani komunikace ani nahrávání skrze Bluetooth s méně dostupnými prostředky nefungují, ale pomocí Wi-Fi modulu NodeMCU jsem schopen komunikovat, i nahrávat programy, bezdrátově.

Testování

V této kapitole popisuji všechny testování, která jsem za dobu psaní bakalářské práce provedl. Nejdříve popisuji komunikaci s Bluetooth modulem HC-06 a následně komunikaci a nahrávání skrze Wi-Fi modul NodeMCU založený na ESP8266 verze ESP-12.

3.1 Testování komunikace s Bluetooth modulem HC-06

Tato sekce je věnována testování komunikace s modulem HC-06. V prvním kroku jsem nahrál do přípravku Arduino Uno program, kterého úkolem bylo na sériovou linku poslat sekvenci znaků „AT“ a po obdržení odpovědi „OK“ rozsvítit vestavěnou LED. Poté jsem k Arduino Uno připojil chip dle obrázku 2.1. Tento test proběhl úspěšně.

Dalším krokem bylo přepojení sériové komunikace s modulem HC-06 na software sériovou linku a použití Arduino Uno jako převodníku. V PC jsem otevřel sériový monitor a skrze Arduino Uno přeposlal do Bluetooth modulu znaky „AT“. Obdržel jsem správnou odpověď „OK“ a tedy jsem se rozhodl na této konfiguraci pokračovat. Nicméně po poslání složitějších příkazů se odpovědi modulu HC-06 začaly diametrálně lišit od očekávaných. Vycítaná z Arduino Uno data byla viditelně kvalitativně i kvantitativně poškozená. Nahrál jsem tedy do Arduino Uno program opětovně, ale jelikož tato operace nepomohla, rozhodl jsem se vyměnit používaný kus přípravku. Bohužel na nové desce Arduino problém stále přetrvával. Chybu se podařilo odstranit až použitím Arduino Mega a nepoužíváním SoftwareSerial knihovny. Druhým funkčním řešením bylo použití převodníku UART–seriová linka. Tato chyba byla způsobena chybou v SoftwareSerial knihovně, která se vyskytuje v případě použití zároveň software a hardware sériové linky.

Komunikace s Bluetooth modulem HC-06 se pomocí AT příkazů zdařila.

3.2 Testování komunikace skrze Wi-Fi s modulem NodeMCU

V této sekci popíšu testování komunikace skrze Wi-Fi s modulem NodeMCU pomocí AT příkazů. Prvním krokem bylo připojení modulu NodeMCU k Arduino Mega, jako na obrázku 2.3. Opětovně sem použil Arduino Mega jako převodník a započal testování komunikace pomocí AT příkazů. Na příkaz „AT“ jsem obdržel správnou odpověď „OK“. Proto jsem nastavil příkazem „AT+CWMODE=2“ modul na fungování v režimu AP a připojil se na něj skrze Wi-Fi počítačem. Následně jsem otevřel telnetem spojení na počítači a přeposlal úspěšně data. Test tedy proběhl v pořádku.

V dalším kroku jsem příkazem „AT+CWMODE=1“ nastavil modul do režimu klient, připojil na otevřenou Wi-Fi síť, nastavil NodeMCU jako server a opětovně se telnetem úspěšně spojil a komunikoval.

Komunikace s Wi-Fi modulem NodeMCU se zdařila, a to jak pomocí AT příkazů tak skrze Wi-fi.

3.3 Testování nahrávání programu do Arduino Uno skrze Wi-Fi modul NodeMCU

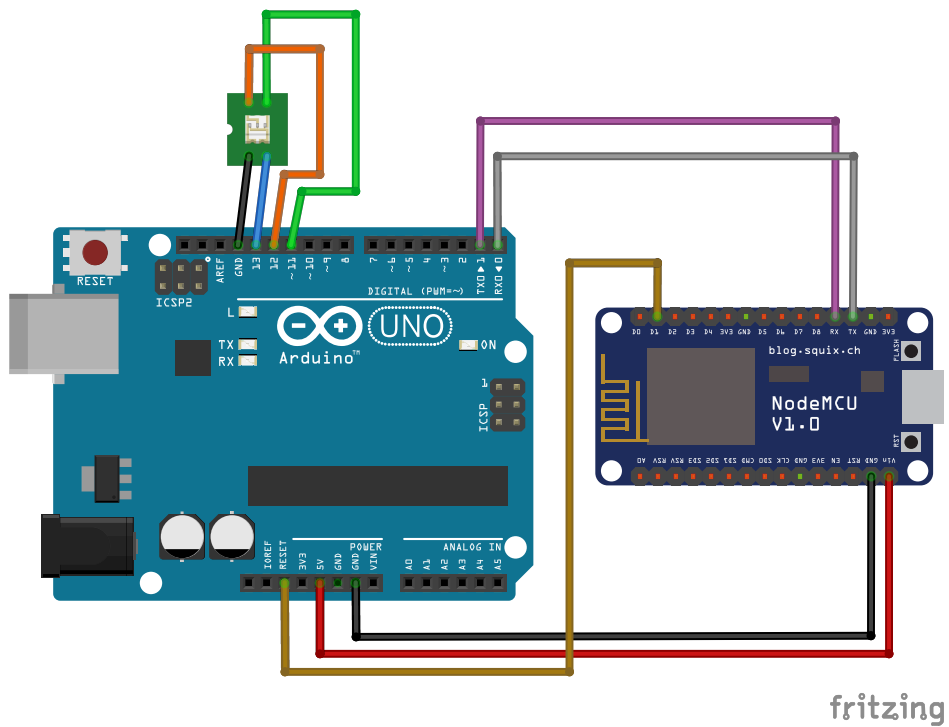
V další části byla otestována funkčnost vytvořeného řešení. Pro tento účel jsem vytvořil dva programy. Na piny 11, 12, 13 jsem připojil RGB LED dle obrázku 3.1. Zapojení modré LED na pin 13 dovoluje pozorovat fázi nahrávání. Během této akce se na pinu 13 několikrát zapne a vypne LED. Po zapojení desek do zdroje jsem se připojil na přístupový bod vytvořený ESP. Použitím příkazu

```
avrdude -p m328p -c arduino -b 115200 \  
-P net:192.168.4.1:23 -U BlinkRED.hex
```

jsem do vývojové desky Arduino Uno úspěšně nahrál program, který RGB LED rozsvítil a zhasnul každou vteřinu.

V dalším kroku jsem do přípravku nahrál obdobný program, který ale rozsvítil a zhasnul RGB LED zeleně s frekvencí dvakrát za vteřinu. Vzhledem k zpozorované úspěšné změně barvy, považuji test za splněný.

3.3. Testování nahrávání programu do Arduino Uno skrze Wi-Fi modul NodeMCU



Obrázek 3.1: Zapojení NodeMCU a RGB LED na přípravek Arduino Uno

Závěr

Hlavním cílem této bakalářské práce bylo vyzkoušení možností bezdrátového nahrávání programů do přípravku platformy Arduino. Pro komunikaci skrze sériovou linku jsem použil přípravek Arduino Mega a pro nahrávání jsem vybral Arduino Uno. Zkoumal jsem proto možnosti komunikace s Wi-Fi a Bluetooth moduly. Komunikace s Bluetooth modulem skrze Bluetooth se nezdařila z důvodu nedostatečných prostředků na mé straně, a to zastaralé verze software na mobilním telefonu. Dokázal jsem však komunikovat skrze sériovou linku a nastavovat tak parametry tohoto modulu AT příkazy.

Komunikace s Wi-Fi modulem NodeMCU proběhla úspěšně a to jak po sériové lince AT příkazy, tak skrze Wi-Fi.

V další části jsem vyzkoušel přeposílání programu do Arduino Uno skrze Wi-Fi. Tato část proběhla bez problému a do přípravku se podařilo program nahrát. To jsem otestoval na jednoduché aplikaci. Tento test proběhl úspěšně.

Výstupem práce je jednoduše popsáný postup pro zprovoznění nahrávání skrze Wi-Fi B. Tuto práci lze rozšířit například o popis nahrávání pomocí technologie Bluetooth. Na tomto tématu nemíním pokračovat v diplomové práci, je to ale vhodné téma pro semestrální práci z předmětu BI-ARD nebo po vhodné úpravě jako bakalářská práce.

Literatura

- [1] ArduinoCC: What is Arduino? [online], [cit. 2017-05-01]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>
- [2] Monk, S.: *30 Arduino projects for the evil genius*. New York: McGraw-Hill, druhé vydání, 2013, ISBN 0071817727;9780071817721;.
- [3] Arduino Uno. [online], [cit. 2017-05-15]. Dostupné z: https://www.arduino.cc/en/uploads/Products/Uno_SMD_0.jpg
- [4] ArduinoCC: Arduino UNO & Genuino UNO. [online], [cit. 2017-05-01]. Dostupné z: <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- [5] Arduino Mega. [online], [cit. 2017-05-15]. Dostupné z: https://www.arduino.cc/en/uploads/Main/AG_Mega.jpg
- [6] AtmelCorporation: Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V. [online], [cit. 2017-05-15]. Dostupné z: http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
- [7] ArduinoCC: Arduino MEGA 2560 & Genuino MEGA 2560. [online], [cit. 2017-05-01]. Dostupné z: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- [8] Arduino Esplora. [online], [cit. 2017-05-15]. Dostupné z: <https://www.arduino.cc/en/uploads/Main/Esplora.jpg>
- [9] ArduinoCC: Arduino Esplora. [online], [cit. 2017-05-01]. Dostupné z: <https://www.arduino.cc/en/Main/ArduinoBoardEsplora>
- [10] ArduinoCC: Arduino Ethernet Shield 2. [online], [cit. 2017-05-01]. Dostupné z: <https://www.arduino.cc/en/Main/ArduinoEthernetShield>

- [11] Osborn, K.: Everything You Always Wanted to know about Arduino Bootloading but Were Afraid to Ask. [online], Leden 2012, [cit. 2017-05-10]. Dostupné z: <http://baldwisdom.com/bootloading/>
- [12] Bluetooth: Our History. [online], [cit. 2017-05-14]. Dostupné z: <https://www.bluetooth.com/about-us/our-history>
- [13] Bluetooth: How It Works. [online], [cit. 2017-05-14]. Dostupné z: <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works>
- [14] HC-06. [online], [cit. 2017-05-15]. Dostupné z: <https://cdn.instructables.com/FKH/L2GS/HXA9PIW3/FKHL2GSHXA9PIW3.MEDIUM.jpg>
- [15] BC04-B Bluetooth Module. [online], [cit. 2017-05-10]. Dostupné z: <http://www.suptronics.com/downloads/BC04-B%20Technical%20specification.pdf>
- [16] NodeMCU. [online], [cit. 2017-05-15]. Dostupné z: http://www.cnx-software.com/wp-content/uploads/2015/10/NodeMCU_v0.9_Pinout.png
- [17] Komunikace AT. [online], [cit. 2017-05-15]. Dostupné z: <https://cdn.instructables.com/FE7/NVI8/HX6ARQFT/FE7NVI8HX6ARQFT.MEDIUM.jpg>
- [18] Wunsch, J.: AVRDUDE - AVR Downloader/UploaDEr. [online], Zář 2013, [cit. 2017-05-10]. Dostupné z: <http://www.nongnu.org/avrdude/>
- [19] Guangzhou HC Information Technology Co., L.: Product Data Sheet. [online], [cit. 2017-05-08]. Dostupné z: <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>
- [20] sayem2603: AT Command Mode of HC-05 and HC-06 Bluetooth Module. [online], [cit. 2017-05-14]. Dostupné z: <http://www.instructables.com/id/AT-command-mode-of-HC-05-Bluetooth-module/>
- [21] Zapojeni HC-06. [online], [cit. 2017-05-15]. Dostupné z: http://www.bitstoc.com/image/data/Products/arduino_bluetooth_serial%20-%20Copy.jpg
- [22] Stehlík, P.: Arduino a ESP8266. [online], [cit. 2017-05-14]. Dostupné z: https://www.pstehlik.cz/prezent/Arduino_a_ESP8266.pdf
- [23] FT232R. [online], [cit. 2017-05-15]. Dostupné z: <http://jeromeabel.net/files/ressources/xbee-arduino/illustrations/ft232R%20breakout.jpg>

- [24] NodeMCU devkit. [online], [cit. 2017-05-15]. Dostupné z: <https://github.com/nodemcu/nodemcu-devkit>
- [25] Pridopia: ESP8266 AT Command Set. [online], [cit. 2017-05-08]. Dostupné z: <http://www.pridopia.co.uk/pi-doc/ESP8266ATCommandsSet.pdf>
- [26] ESP-Link github. [online], [cit. 2017-05-15]. Dostupné z: <https://github.com/jeelabs/esp-link>

Seznam použitých zkratk

BI-ARD Bakalářský předmět Interaktivní aplikace s Arduinem

AP Access Point

PC Personal Computer

STA Station/Client

Uživatelská příručka pro nahrávání skrze Wi-fi

Tato příloha je zkrácenou uživatelskou příručkou. V několika bodech umožňuje čtenáři znalému platformy Arduino nahrát svůj program pomocí Wi-fi skrze modul NodeMCU.

B.1 Technické požadavky

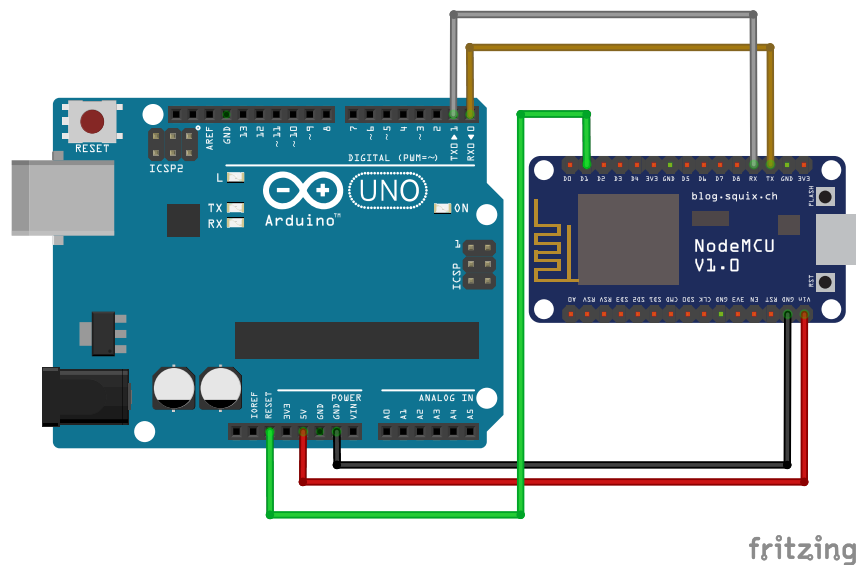
- Unixový systém (na Windows bylo vyzkoušeno, že nahrávání nefunguje)
- program pro nahrávání programu do procesoru AVRDUDE
- Arduino Uno
- NodeMCU devkit v0.9
- propojovací kabely

B.2 Postup při nahrávání programu do Arduino Uno

1. Připojte NodeMCU devkit k vývojové desce Arduino Uno podle obrázku B.1.
2. Stáhněte a nahrajte ESP-Link na NodeMCU, podle [26]. Rozbalte, a v terminálu přejděte do složky *esp-link-*"vaše_verze"**. Použijte příkaz:

```
python ../esptool-master/esptool.py --port COM4 \  
--baud 115200 write_flash --flash_freq 80m --flash_mode qio \  
--flash_size 32m 0x0000 boot_v1.6.bin 0x1000 user1.bin \  
0x3FC000 esp_init_data_default.bin 0x3FE000 blank.bin
```

B. UŽIVATELSKÁ PŘÍRUČKA PRO NAHRÁVÁNÍ SKRZE WI-FI



Obrázek B.1: Arduino Uno s připojeným NodeMCU

3. Po připojení PC k Wi-Fi nahrajte v sekci *Upgrade Firmware* soubor `user2.bin`.
4. Vytvořte pořadovaný program pro přípravu Arduino Uno a zkompilujte.
5. Připojte počítač k ESP síti nebo připojte počítač ke stejné síti jako ESP.
6. Nahrajte program použitím příkazu:

```
avrdude -p m328p -c arduino -b 115200 \  
-P net:esp-link:23 -U program.hex
```

Případně nahraďte v příkazu *esp-link* jeho IP adresou.

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
hex.....	adresář s hex formou implementace pro Arduino Uno
├─ program.hex.....	hex soubor pro nahrání na Arduino Uno, blikání zelenou LED
├─ BlinkRED.hex.....	hex soubor pro nahrání na Arduino Uno, blikání červenou LED
src.....	složka obsahující zdrojové kódy
├─ program.....	složka požadovaná ArduinoIDE
├─┬─ program.ino.....	zdrojový kód implementace program.hex
├─ BlinkRED.....	složka požadovaná ArduinoIDE
├─┬─ BlinkRED.ino.....	zdrojový kód implementace program.hex
├─ thesis.....	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
text.....	text práce
├─ thesis.pdf.....	text práce ve formátu PDF