

Připojení vývojové desky Arduino k bezdrátové síti WiFi pomocí modulu ESP8266

Martin Hipík

Bakalářská práce
2018



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martin Hipík**
Osobní číslo: **A17735**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **kombinovaná**

Téma práce: **Připojení vývojové desky Arduino k bezdrátové síti WiFi pomocí modulu ESP8266**

Téma anglicky: **Connecting Arduino-boards to a Wi-Fi Wireless Network Using the ESP8266 Module**

Zásady pro vypracování:

1. Prostudujte a popište modul ESP8266 a jeho využití na platformě Arduino.
 2. Vyberte vhodný hardware pro realizaci ukázkové aplikace pro uvedený modul.
 3. Realizujte hardwarové zapojení podle svého návrhu.
 4. Implementujte programové vybavení pro mikropočítač zajišťující sběr dat a jejich přenos přes WiFi síť.
 5. Vytvořte aplikaci pro zobrazení získaných dat na vhodném zařízení.
-

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. BELL, Charles A. Beginning sensor networks with Arduino and Raspberry Pi. New York, New York: Apress, 2013. ISBN 978-1-4302-5825-4.
2. LADMAN, Josef. Elektronické konstrukce pro začátečníky. Praha: BEN – technická literatura, 2001. ISBN 80-730-0015-6.
3. MARGOLIS, Michael. Arduino cookbook. 2nd ed. Sebastopol, Calif.: O'Reilly, 2012, xx, 699 p. ISBN 1449313876.
4. MAZIDI, Muhammad Ali, Sarmad NAIMI and Sepehr NAIMI. The AVR microcontroller and embedded systems: using Assembly and C. Upper Saddle River, N.J.: Prentice Hall, 2011, xiv, 776 p. ISBN 01-380-0331-9.
5. PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN – technická literatura, 2004. ISBN 80-7300-110-1.

Vedoucí bakalářské práce:

Ing. Jan Dolinay, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

15. prosince 2017

Termín odevzdání bakalářské práce:

25. května 2018

Ve Zlíně dne 15. prosince 2017

doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

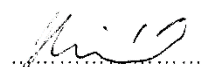
Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s přípoštěním tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 21.5.2018


.....
podpis diplomanta

ABSTRAKT

Cieľom práce je návrh a vytvorenie programového a hardwarového vybavenia pre bezdrôtovú komunikáciu s vývojovou doskou Arduino UNO pomocou modulu ESP8266-01.

Teoretická časť obsahuje pohľad na platformu Arduino, technické špecifikácie, priblíženie sériovej komunikácie a príkazov AT. Bližší pohľad na modul ESP8266-01, princíp komunikácie v bezdrôtovej Wi-Fi sieti.

Praktickú časť tvorí návrh zapojenia prototypu na vývojovú dosku. Programové vybavenie pre platformu Arduino a popis jednotlivých funkcií programu. Ďalej ukážkovú aplikáciu pre komunikáciu s vývojovou doskou, v programovacom jazyku JAVA so zdrojovým kódom.

Kľúčová slova: Wi-Fi, Arduino UNO, ESP8266, ovládanie, monitoring

ABSTRACT

The aim of the thesis is to design and create software and hardware equipment for wireless communication with the Arduino UNO development board using the ESP8266-01 module.

The theoretical part includes an overview of the Arduino platform, technical specifications, serial communication and AT commands. A closer look at ESP8266-01, the communication principle in a wireless Wi-Fi network.

The practical part describes the proposed circuit for the development board. It also deals with the software for Arduino platform and description of program functions. It also shows sample application for communication with the development board, in the JAVA programming language with the source code.

Keywords: Wi-Fi, Arduino, ESP8266, remote, monitoring

Týmto by som chcel poďakovať vedúcemu mojej bakalárskej práce Ing. Janovi Dolinayovi, Ph.D za trpezlivosť, odborné vedenie, cenné rady a ochotu ktoré mi poskytol pri písaní tejto práce. Ďalej by som rád poďakoval mojej rodine a blízkym za podporu počas celú dobu štúdia.

Prehlasujem, že odovzdaná verzia bakalárskej/diplomovej práce a verzia elektronická nahraná do IS/STAG sú totožné.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 OPEN-SOURCE PLATFORMA ARDUINO	10
1.1 VÝVOJOVÉ PROSTREDIE	11
1.2 ARDUINO UNO	12
1.3 TECHNICKÉ INFORMÁCIE ARDUINO UNO.....	13
1.4 PINOUT – SCHÉMA	14
1.4.1 Napájanie, informácie o I/O	14
1.5 SÉRIOVÁ KOMUNIKÁCIA	15
1.5.1 UART	15
1.5.1.1 Priebeh komunikácie a chyby	16
1.5.2 Monitor sériovej komunikácie	17
1.5.3 I ² C.....	17
1.5.4 AT príkazy.....	18
2 WI-FI KOMUNIKÁCIA	19
2.1.1 Princíp Wi-Fi.....	19
2.1.2 Protokoly TCP/IP	20
2.1.2.1 Protokol TCP	21
2.1.2.2 Protokol UDP.....	21
2.1.2.3 Sieťový port	21
3 WI-FI MODUL ESP8266	22
3.1 TECHNICKÉ INFORMÁCIE ESP8266-01	23
3.1.1 Napájanie, Pinout schéma	23
3.1.2 AT príkazy pre ESP 8266.....	25
3.2 POUŽITIE NA PLATFORME ARDUINO	26
3.2.1 Remote - režim	27
3.2.2 Server - režim	28
II PRAKTICKÁ ČÁST	29
4 REALIZÁCIA	30
4.1 NÁVRH ZAPOJENIA	30
4.1.1 Použité komponenty	31
4.1.2 Test spojenia.....	34
4.1.3 Schéma zapojenia.....	35
4.1.4 Zostavenie a oživenie	36
4.2 PROGRAMOVÉ VYBAVENIE	37
4.2.1 Arduino.....	37
4.2.1.1 Zoznam vytvorených funkcií a ich popis.....	40
4.2.1.2 Použité funkcie a knižnice	42
4.2.2 Java aplikácia	44
4.3 OVERENIE.....	46
ZÁVĚR	47

SEZNAM POUŽITÉ LITERATURY.....	48
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	50
SEZNAM OBRÁZKŮ	51
SEZNAM TABULEK.....	52
SEZNAM PŘÍLOH.....	53

ÚVOD

V dnešnej dobe je trendom pripojenie rozličných zariadení ako napríklad žiarovky, zabezpečovacie systémy, multimedialne systémy, prehrávače, televízory, chladničky, práčky a veľa ďalších do siete Internet. Spájať ich do celkového systému, ktorý si dokáže medzi sebou vymieňať informácie a prispôsobovať sa užívateľovi. Vzájomnou komunikáciou doceliť ich jednotné ovládanie, správu, ale aj pracovať s informáciami jednoduchšie ako kedkoľvek predtým. Takýto systém nazývame IoT (Internet of Things – Internet vecí).

Súčasná technika umožňuje výrobu čipov s veľmi nízkou spotrebou energie, rozširujú sa zdroje obnoviteľnej energie a ich použitie. To nám dáva možnosť použitia efektívnych čipov tam, kde to doteraz nebolo možné.

Taktiež celý proces programovania a ovládania zariadení sa posúva vpred. Pomocou Open-source platforiem je možné jednoducho realizovať projekty podľa želaní užívateľa. Vytvoriť zariadenia vhodné pre konkrétne použitie. Pracovať na projektoch s celým svetom, jednoducho si vymieňať informácie a získané poznatky s inými ľuďmi. Rozširuje sa použitie rôznych vývojových platforiem, s multifunkčným využitím. Jednou takouto platformou je aj Arduino.

Cieľom práce je vytvorenie jednoduchého a lacného systému, ktorý dokáže pracovať nezávisle a bez použitia serveru, možnosť spracovávať rôzne dáta (pripojenie senzorov fyzikálnych veličín a práca s nameranými hodnotami), poskytovať ich distribúciu v sieti alebo jednoduchý systém pre ovládanie. Návrh bude možné prispôbiť pre konkrétnu aplikáciu ako je ovládanie zariadení (audio sústava – pre prepínanie vstupov audio signálu, ovládanie hlasitosti,...), ovládanie zariadení v domácnosti – vytvorenie Smart zásuvky a tak ovládať spotrebiče alebo použitie na regulovanie teploty, osvetlenia, ventilácie v dome. Vytvorenie napríklad domácej meteo stanice, ktorá je spojená s automatickým ovládaním žalúzií alebo TUV. Celý modul bude možné rozšíriť v prípade, ako bude počet I/O na Arduine UNO nedostačujúci, je možné dosku nahradiť napríklad radou MEGA. Na trhu však už podobné zariadenia existujú, avšak ich cena je v čase písania tejto práce okolo tisícok korún. Taktiež modularita alebo výmena jednotlivých komponentov a tak rozšírenie nie je možné. Navyše je vo väčšine takýchto zariadení potrebné pripojenie k internetu, pretože komunikácia s takými zariadeniami je smerovaná cez externé servery použitím MQTT protokolu. Príkladom môžu byť rôzne smart lampy, inteligentné zásuvky a podobne. Tu nastáva otázka bezpečnosti a možnosti nasadenia.

I. TEORETICKÁ ČÁST

1 OPEN-SOURCE PLATFORMA ARDUINO

Arduino je Open-source platforma postavená na mikrokontroléri od firmy Atmel z rady ATmega. Softwarová časť čerpá z prostredia Wiring. Vývoj začal v roku 2005, kde táto platforma bola založená ako myšlienka vytvorenia lacného vývojového kitu. Arduino je vhodné pre vytváranie nových projektov, ich testovanie a ladenie. Všetky schémy a návody pre platformu Arduino sú zverejnené a voľne dostupné. Platforma podporuje veľké množstvo knižníc, ktoré je možné editovať alebo vytvárať nové.

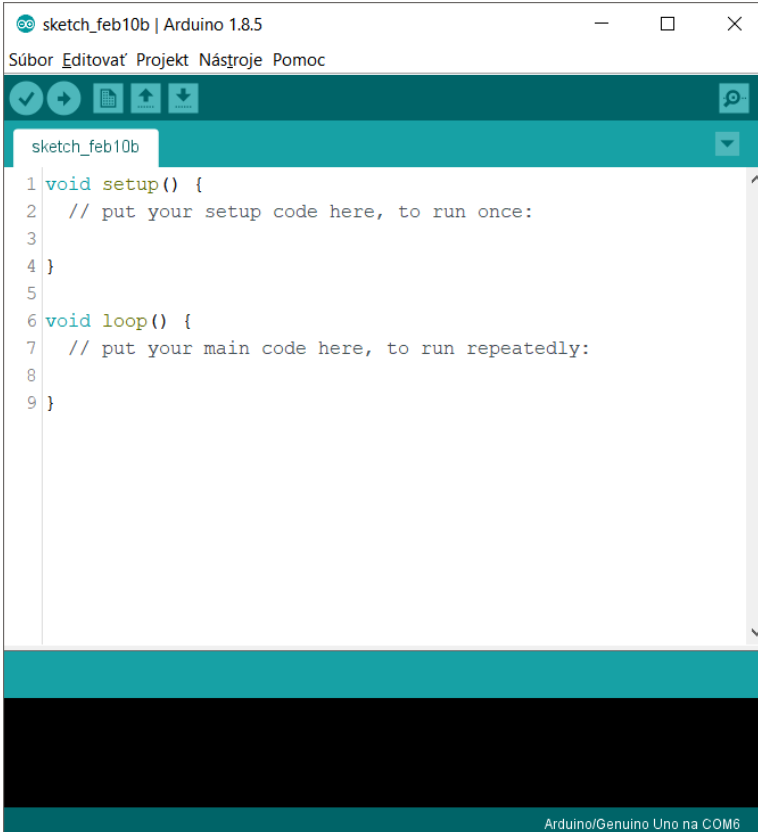
Momentálne je na trhu k dispozícii 22 verzii oficiálnych dosiek Arduino. K dispozícii sú ale aj rôzne upravené alebo prispôsobené pre konkrétne použite. Príkladom je modul pre stavbu robotov alebo ovládanie krokových motorov a iné. Výber vývojovej dosky Arduino je podľa aplikácie. Pre jednoduché aplikácie sú k dispozícii vývojové dosky menšieho výkonu a s menším počtom I/O, pre aplikácie vyžadujúce väčší výkon je možné použiť výkonnejšie vývojové dosky napríklad radu MEGA.

Pre platformu Arduino je typická ich farba, tá je tmavo modrá. Každá vývojová doska je označená názvom, za ktorým sa môže nachádzať označenie verzie. Toto označenie verzie sa používa, ak boli vykonané na doske zmeny alebo vylepšenia. Podľa typu vývojové dosky sa líši aj ich výbava, niektoré obsahujú prevodník ktorý sprostredkujú komunikáciu medzi PC a čipom, iné typy dosiek tento prevodník neobsahujú z dôvodu úspory miesta a je nutné použiť prevodník externý. [1]

Vývojové dosky je možné rozšíriť o tzv. Shield – ide o dosku, ktorá obsahuje hardware pre konkrétnejšiu aplikáciu. K danému Shieldu môžu byť vytvorené knižnice pre zjednodušenie práce a ovládania. Napríklad ak potrebujeme s vývojovou doskou komunikovať pomocou Bluetooth tak použijeme Bluetooth Shield vhodný pre daný typ vývojovej dosky. Týmto získame rozšírenie konektivity a schopnosti dosky. Existujú rôzne Shildy a rôzni výrobcovia, ktorý ponúkajú svoje produkty pre túto platformu. Výber je naozaj veľký od rôznych LED panelov, snímačov teploty, tlaku, ovládanie servomotorov, tlačidiel až po pripojenie cez Ethernet. Komunita vývojárov pre Arduino je naozaj veľká a stále narastajúca je to výborný spôsob pre realizáciu nových alebo vylepšenie projektov.

1.1 Vývojové prostredie

Vývojové prostredie je multiplatformová aplikácia napísaná v programe JAVA. Je navrhnutá tak, aby umožňovala programovanie aj ľuďom, ktorí nemajú s programovaním veľké skúsenosti. Editor Arduino IDE obsahuje základné nástroje pre úpravu kódu a jeho formátovanie. Výhodou tohto softwaru je možnosť vytvorený program jednoducho skompilovať a nahrať do Arduina a to kliknutím na jedno tlačidlo. Taktiež užívateľa oslobodzuje od inštalovania a konfigurovania ovládačov. Programy pre platformu Arduino sa píšu v jazyku C alebo C++. Arduino platforma je rozšírená knižnicami s rôznymi funkciami, ktoré uľahčujú prácu. Samozrejmosťou je jednoduché rozšírenie o knižnice vytvorené používateľmi. Knižnice je možné jednoducho importovať.



```
sketch_feb10b | Arduino 1.8.5
Súbor Editovať Projekt Nástroje Pomoc
sketch_feb10b
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
Arduino/Genuino Uno na COM6
```

Obr. 1. Vývojové prostredie Arduino IDE

Program pre Arduino je založený na dvoch funkciách `setup()` a `loop()`. Funkcia `setup()` je spustená na začiatku programu a používa sa na definovanie parametrov, premenných, funkciu pinov a pod. Funkcia `loop()` je spúšťaná neustále pokiaľ je mikropočítač zapnutý.

Každý projekt je uložený do samostatného pracovného adresára, ktorý ma rovnaké meno ako projekt. V tomto adresári môžeme nájsť súbory s príponou *.ino* (novšia) alebo *.pde*. Pri každej kompilácii a nahrávaní programu je nutné vybrať správny typ dosky do ktorého bude nahraný kód a priradený COM port.

1.2 Arduino UNO



Obr. 2. Vývojová doska Arduino UNO [1]

V súčasnosti ide o najpoužívanejšiu verziu vývojovej dosky Arduino. Na doske sa nachádza procesor ATmega328, ktorý obsahuje 32KB pamäte pre program, 2KB SRAM a 1KB EEPROM, ďalej je tu prevodník pre komunikáciu cez USB port, ktorý sa nachádza na doske. Podľa vyhotovenia dosky môže byť typ USB konektora odlišný. Doska Arduino UNO je ale vyrábaná v dvoch ďalších verziách. Prvá verzia je Arduino UNO Ethernet, ktorá má namiesto USB pripojenia port Ethernet. Druhá verzia je s použitím Bluetooth čipu pre pripojenie, ktorý je priamo integrovaný na doske. Ďalej na doske nájde stabilizátor napätia pre externé napájanie, vstupné a výstupné piny pre pripojenie zariadení a komunikáciu. [2]

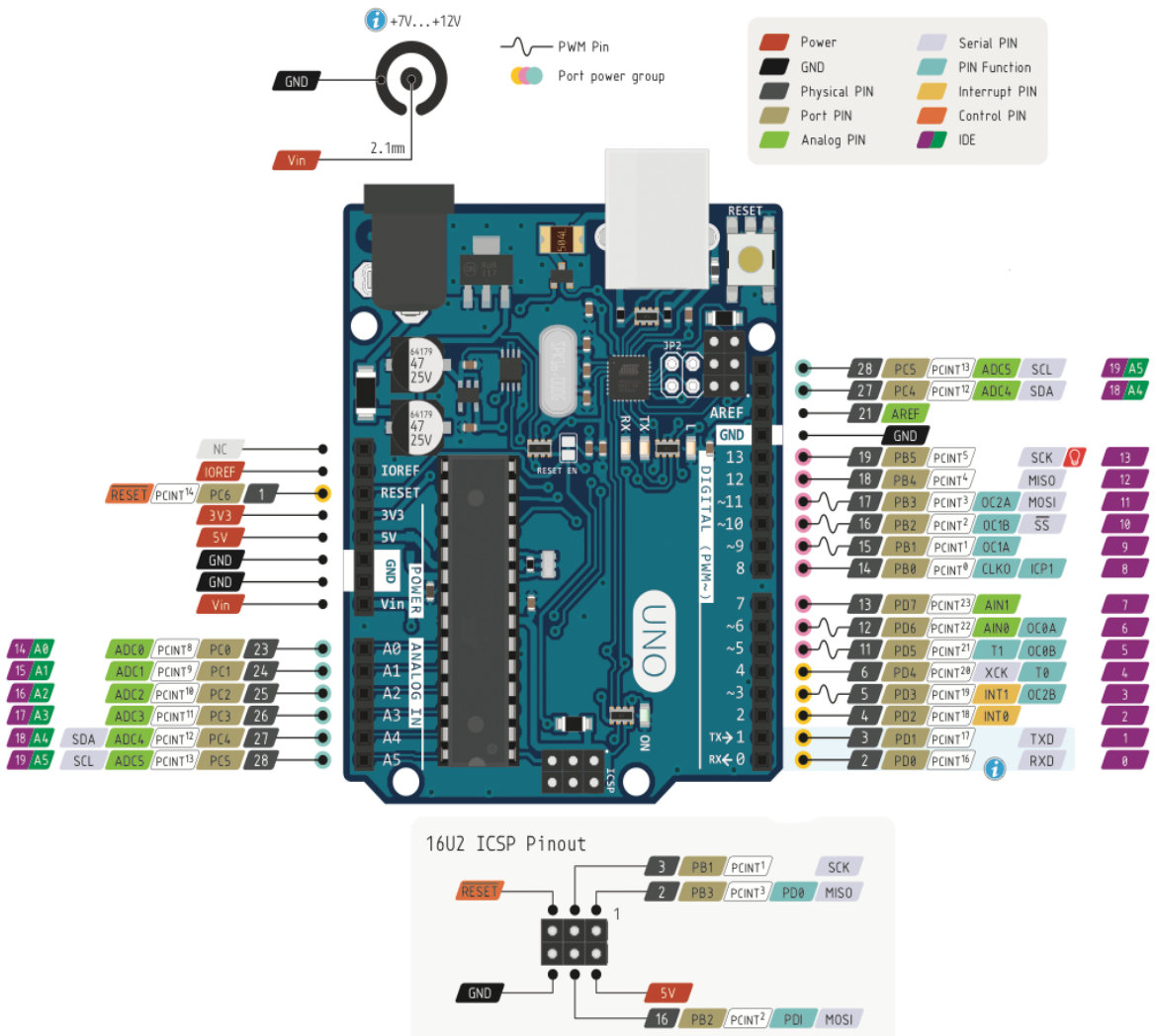
1.3 Technické informácie Arduino UNO

Tab. 1. Technické informácie Arduino UNO [3]

Technické informácie Arduino UNO	
Mikrokontrolér	ATmega328P
Pracovne napätie	5 V
Vstupné napätie (odporúčané)	7 - 12 V
Vstupné napätie (limitné)	6 - 20 V
Digitálne I/O Piny	14
PWM Digital I/O Piny	6
Vstup Analog I/O Pin	6
DC Odber pre I/O Pin	40 mA
DC Odber pre 3.3V Pin	50 mA
Flash Pamäť	32KB (ATmega328P) 0.5KB použité pre bootloader
SRAM	2KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Taktovacia frekvencia	16 MHz
LED_PIN	13
Dĺžka	68.6 mm
Šírka	53.4 mm
Váha	25 g

- Použitý procesor ATMEL ATmega328P
- Na doske sa nachádza stabilizátor napätia AMS1117, ktorý stabilizuje vstupné napájacie napätie. Konektor pre pripojenie externého napájacieho zdroja.
- Prevodník starajúci sa o prevod z USB zbernice pre TTL logiku.
- Informačné diódy signalizujúce zapnutie, RX TX prebiehajúci prenos a LED diódu, ktorá je pripojená na PIN 13, testovacia LED.
- Micro-USB alebo USB typ B záleží od verzie. Slúži na spojenie s PC, nahrávanie programu, komunikáciu pomocou sériovej linky.
- Tlačidlo RESET, slúži na resetovanie mikrokontroléru. V prípade kedy chceme program spustiť od znova. Doska obsahuje aj reset PIN pre tvorbu automatického resetu alebo resetovanie externým zariadením.
- ICSP hlavica pre externé programovanie pomocou USB-Serial prevodníka. Monitorovanie a rozšírené funkcie.
- napájacie piny

1.4 Pinout – schéma



Obr. 3. Schéma výstupných pinov vývojovej dosky Arduino UNO[3]

1.4.1 Napájanie, informácie o I/O

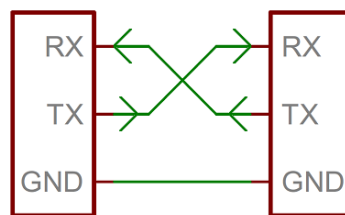
Napájanie vývojovej dosky Arduino UNO môže byť realizované dvomi spôsobmi. Pripojením stabilizovaného napätia 5 V alebo použitím externého napájania, ktoré môže byť v rozsahu 7 - 12 V (maximálne 6 - 20 V). O stabilizáciu na 5 V sa postará integrovaný stabilizátor napätia AMS1117. Táto možnosť nám uľahčuje výber možnosti napájania a jeho stabilizáciu. Napájacie napätie musí byť jednosmerné. Zdroj napätia by mal dosahovať minimálne odporúčaný prúd 300mA. Použiť je možné batérie, impulzné zdroje, solárne články s regulátorom a iné.

Analógové vstupné piny – Arduino UNO obsahuje 6 kanálov pre analógový signál, signál je vzorkovaný na hodnotu 0 - 1023 (UNO obsahuje 10-bitový A/D prevodník). Analógové vstupy je možné použiť aj ako digitálne I/O. Maximálna hodnota signálu je 5 V.

Digitálne I/O – počet pinov je 14, možnosť použiť 6 analógových v režime digital. Digitálne piny podporujú PWM moduláciu, tieto piny sú označené vlnovkou (~). Každý digitálny pin je možné zaťažiť prúdom max. 40 mA. [3;4]

1.5 Sériová komunikácia

Pre komunikáciu čipu s iným zariadením napríklad PC je potrebné použiť sériovú komunikáciu UART. Využíva sa na posielanie a prímanie dát medzi pripojenými zariadeniami. Sériová komunikácia používa dva vodiče TX (Transmitted - odoslané) / RX (Received - prijaté) a používa logické úrovne TTL (5 V alebo 3,3 V v závislosti od dosky).

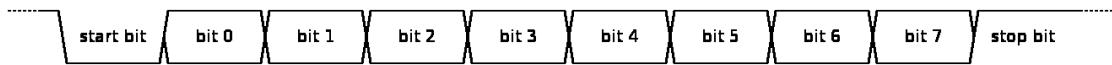


Obr. 4. Schéma pripojenia dvoch zariadení

1.5.1 UART

Komunikácia pomocou dvoch vodičov RX a TX pre posielanie a prijímanie dát. Ide o asynchrónnu sériovú komunikáciu, to znamená, že nie je prenášaný žiadny hodinový signál. Zariadenia obsahujú vysielateľ, prijímač, generátor hodinového signálu. Rýchlosť hodín je možné nastavovať spravidla na 8-násobok prenosovej rýchlosti. Tiež je možné nastavovať dĺžku slova, paritu, Start a Stop bit. Toto nastavenie musí byť na oboch zariadeniach rovnaké. Pre správnu komunikáciu je potrebné nastavenie prenosovej rýchlosti – Baud rate. Typické rýchlosti sú 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 a 115200 Bd. Prijímač musí zaisťovať správnu detekciu prenášaného dátového bitu. Kedy jeden končí a začína druhý, toto nastavenie je docielené pomocou Start bitu.

Príklad dátového bloku prenášaného po UART:



Obr. 5. Dátový blok prenášaný po sériovej zbernici [5]

Prijímač testuje stav prichádzajúceho signálu na každom časovom impulze a hľadá Start bit. Ak je Start bit nájdený, zapíše prijaté dáta do posuvného registra, po uplynutí definovanej dĺžky slov je obsah registra dostupný pre ďalšie spracovanie. Prijímač nastaví príznak prijatia nových dát alebo môže vygenerovať prerušenie.

1.5.1.1 *Priebeh komunikácie a chyby*

- Synchronizácia interných hodín pomocou START BIT
- Odosielanie jednotlivých bitov v definovanom presnom čase daným hodinami
- Odoslanie Paritného bitu – ak je použitý, slúži na kontrolu chyby
- Odoslanie Stop bit – oznamuje o ukončení prenosu
- Spracovanie dát – odstránenie Start, paritného a stop bitu.
- Generovanie príznaku pre upozornenie na nové dáta

Typy chýb

Framing error – chyba, ktorá nastane, ak nebol nájdený START alebo STOP bit.

Overrun error – chyba pretečenia nastane, ak prijímač nedokázal spracovať znak, ktorý prišiel predtým ako prišiel ďalší. Ide o hardwarové limity dané veľkosťou vyrovnávacie pamäte pre UART komunikáciu. [6;7]

1.5.2 Monitor sériovej komunikácie

Pre monitorovanie, komunikáciu, ladenie sériovej zbernice je možné využiť množstvo programov, ktoré dokážu pracovať so sériovou linkou. K dispozícii sú freeware programy alebo platené profesionálne verzie. Výhodou Arduino IDE prostredia je integrovaný jednoduchý, ale funkčný Sériový monitor. Alternatívou môže byť klientsky program PuTTY, ktorý je dostupný pre Windows a Linuxovú platformu. Program PuTTY je šírený pod licenciou Open-source. Podporuje Telnet, SSH a Rlogin komunikáciu. Tieto protokoly sú používané na vzdialenú komunikáciu cez sieť. Program PuTTY predstavuje klienta, ktorý sa pripojí k vzdialenej relácii. Obsahuje aj sériový monitor pre sériovú komunikáciu.

1.5.3 I²C

Sériová zbernica vyvinutá firmou Philips, primárne určená k nízko rýchlostnému pripojeniu periférnych zariadení. Existujú tri verzie zbernice, rozlišujeme ich podľa komunikačnej rýchlosti : štandardná – 100 kHz, rýchla – 400 kHz a vysokorýchlostná – 3,4 MHz. Obvody, ktoré používajú I²C protokol sú pamäte EEPROM a RAM, senzory teploty, expandéry portov, hodiny reálneho času atď. Zbernica umožňuje pripojiť pri 7-bitovom adresovaní max. 119 I/O zariadení pomocou dvoch vodičov. Jeden vodič slúži na prenos hodinového signálu (označenie SCL - Synchronous Clock), druhý vodič je dátový (označenie SDA - Synchronous Data). Každé zariadenie je označené jedinečnou 7 bitovou adresou, ktorá sa používa pri komunikácií. Zbernica obsahuje zariadenie riadiace (master) a riadené (slave). Výhodou zbernice je jednoduché pripojenie zariadení a komunikácia pomocou adres.

Master – začína a končí dátový prenos generovaním Start a Stop bitu, generuje hodinový signál.

Slave – pripojené zariadenie, komunikácie prebieha, ak je oslovené svojou adresou

Prenos bitu – počas dátového prenosu sa stav na dátovom vodiči bitu mení iba keď je hodinový signál na nízkej úrovni. [6;7;8]

1.5.4 AT příkazy

Boli vytvorené pre konfigurovanie modemov v roku 1981. Občas nazývané ako Haysove príkazy. Ich pôvodný účel bol pre konfiguráciu, vzdialené ovládanie, riadenie modemov, ale aj iných zariadení ako napríklad GSM moduly. V súčasnosti sa používajú na konfiguráciu jednoduchých zariadení. Presné používanie pojmu AT sa mierne líši od výrobcu často podliehajúceho danému zariadeniu. Funkcie parametru sú určené výrobcom alebo podporou Firmware. Formát príkazov ako aj samotne podporujúce parametre sú presne definované v dokumentácii k čipu, je možné použitie aj špeciálnych príkazov.

Ako komunikačný kanál sa využíva sériová linka. Najčastejším komunikačným rozhraním je program typu Terminal. Pri komunikácii je potrebné zvoliť správnu komunikačnú rýchlosť a ukončovací znak. Každý príkazový riadok musí začínať s príkazom *AT* + *príkaz* a musí byť ukončený znakom *<CR> End-of-Line*. Nie je možné na jeden príkazový riadok napísať viacero príkazov. Zariadenie má odoslať odpoveď pre všetky príkazy, ktoré prijalo ak nie je možnosť odpovede vypnutá. Tieto odpovede môžu byť buď vo forme reťazcov ASCII alebo číselných hodnôt. AT príkazy sú prijaté zariadením iba v režime príkazov.

Príkazy sa rozdeľujú na 4 druhy podľa toho čo je ich význam alebo návratová hodnota:

Test – vracia hodnotu, slúži na testovanie napríklad prítomnosti.

Query – zistenie stavu alebo hodnoty.

Set – nastavenie stavu alebo hodnoty zariadenia.

Execute – spúšťa príkazy bez definovaných parametrov.

[9;10]

2 WI-FI KOMUNIKÁCIA

Wi-Fi je súbor štandardov pre bezdrôtový prenos dát, založený na špecifikácií IEEE 802.11. Pre pripojenie zariadenia na Wi-Fi sieť je potrebný adaptér podporujúci štandard, ktorým komunikuje s prístupovým bodom (Access point). Oblasť pokrytá Wi-Fi signálom sa nazýva hotspot. Wi-Fi prenos bol vyvinutý v roku 1991 vo firme NCR Corporation/AT&T v Holandsku. Cieľom Wi-Fi siete bolo prepojenie zariadení a ich vzájomná komunikácie napríklad v lokálnej (firemnej sieti). Neskôr sa toto pripojenie začalo využívať aj na pripojenie do siete Internet. Wi-Fi sieť pracuje najčastejšie v frekvenčnom pásme 2.4 Ghz a 5 GHz ide o „voľné“ frekvenčné pásma.

Ku každej bezdrôtovej sieti musí mať prevádzkovateľ od štátu licenciu pre vysilenie v určitej frekvencii, tzv. licenčné pásmo. Frekvenčné pásma sú definované štandardmi a ich použitie je obmedzené jednotlivými krajinami. [12]

2.1.1 Princíp Wi-Fi

Typická Wi-Fi sieť obsahuje minimálne jeden prístupový bod a aspoň jedného klienta, ktorý je na daný prístupový bod pripojený. Prístupový bod je charakteristický svojim SSID, ide o sieťový identifikátor vysielaný pomocou paketov beacons (maják, signál), SSID je vysielaný zvyčajne každých 100ms. Klienta predstavuje zariadenie pripájané na prístupový bod, klientov pripojených na jeden prístupový bod môže byť viac. Pre komunikáciu rôznych zariadení od rozličných výrobcov existujú štandardy, ktorými sa zaoberá inštitút IEEE, tento inštitút vydáva špecifikáciu štandardov bezdrôtových sietí pod číslom 802.11. Tento dokument je rozšírený o užšie špecifikácie pod revíznymi písmenami 802.11b, 802.11g, 802.11ac líšia sa maximálnou rýchlosťou alebo použitým komunikačným pásmom. Na ktorú Wi-Fi bude klient pripojený je celkovo na jeho režii, výber môže byť na základe komunikačnej rýchlosti, sile signálu a pod. Pripojenie na Wi-Fi sieť môže byť chránené pomocou niektorého zo zabezpečení. Siete Wi-Fi umožňujú byť chránené napríklad heslom. [12;13]

2.1.2 Protokoly TCP/IP

Vzhľadom k zložitosti sieťovej komunikácie je sieťový prenos rozdelený do vrstiev. Výmena dát a komunikácia je medzi vrstvami presne daná. Každá z vrstiev využíva služby vrstvy podriadenej a poskytuje služby vrstve nadriadenej. Komunikácia medzi dvoma rovnakými vrstvami rôznych zariadení je riadená pomocou komunikačného protokolu. Architektúra TCP/IP je tvorená pomocou 4 vrstiev.

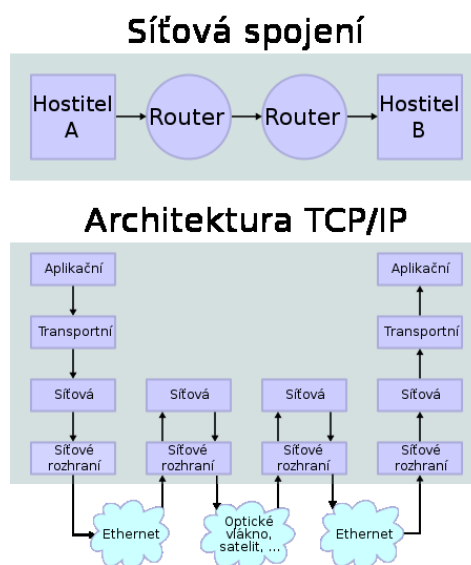
Popis jednotlivých vrstiev:

Sieťové rozhranie – najnižšia vrstva umožňujúca prístup k fyzickému prenosovému médiu. Je prispôsobená podľa danej siete napríklad Ethernet, FDDI.

Sieťová vrstva – zaisťuje predovšetkým adresovanie v sieti, smerovanie, predávanie datagramov, je implementovaná vo všetkých prvkoch siete.

Transportná vrstva – vytvára transportné služby zamerané na kontrolu spojenia, prenosu celistvých dát použitím TCP protokolu. Použitie UDP protokolu sa vyznačuje nespoľahlivým spojením pre služby, ktoré nevyžadujú kontrolu.

Aplikačná vrstva – poskytuje rozhranie pre užívateľské služby napríklad WWW, Telnet, FTP, e-mail,... Každéj službe je priradené číslo – port podľa ktorého sú pakety rozdeľované jednotlivým aplikáciám. [12]



Obr. 6. Diagram TCP/IP vrstiev [17]

2.1.2.1 Protokol TCP

Protokol TCP (Transmission Control Protocol) je najpoužívanější protokol pracující na transportnej vrstve. Pomocou protokolu TCP je možné vytvoriť obojsmerné (duplexné) spojenie medzi zariadeniami. Protokol sa vyznačuje spoľahlivým doručením dát a v správnom poradí. Protokol podporuje rozdelenie dát pre viac aplikácií napríklad webový server a e-mailový server bežiaci na jednom zariadení. TCP protokol umožňuje opakované odoslanie poškodených alebo stratených paketov, ďalej umožňuje zoradenie paketov do správneho poradia. Konce spojenia sú určené portami.

Zjednodušený princíp prenosu použitím TCP :

Aplikácia posiela dáta , TCP rozdeľuje dáta do segmentov (veľkosť segmentu je daná parametrom MTU), TCP predá vzniknuté segmenty IP protokolu k prenosu do druhého zariadenia v sieti. Po prijatí TCP overuje poradie tým, že bolo pridané každému segmentu poradové číslo. TCP protokol odošle odosielateľovi potvrdenie, ktoré pakety boli úspešne prijaté, dáta sú kontrolované pomocou kontrolného súčtu pre detekciu poškodenia. [12][14]

2.1.2.2 Protokol UDP

Protokol UDP (User Datagram Protocol) pracuje na transportnej vrstve, ide o tzv. „nespoľahlivý“ protokol. UDP prenáša dáta medzi počítačmi v sieti, ale v porovnaní s TCP protokolom nezaručuje, že odoslané dáta budú doručené, alebo budú doručené v správnom poradí, alebo nedoručí niektorý paket viac krát. UDP nachádza použitie v sieťach, kde je potrebný rýchly a efektívnejší prenos dát, napríklad video stream, prenos hlasu, online hry a iné. UDP protokol poskytuje jednoduché rozhranie pre prenos dát, neposkytuje žiadne záruky doručenia paketu. UDP pridáva kontrolné súčty a podporu pre rozdelenie paketov medzi viaceré aplikácie bežiacie na jednom zariadení. [12;14]

2.1.2.3 Sieťový port

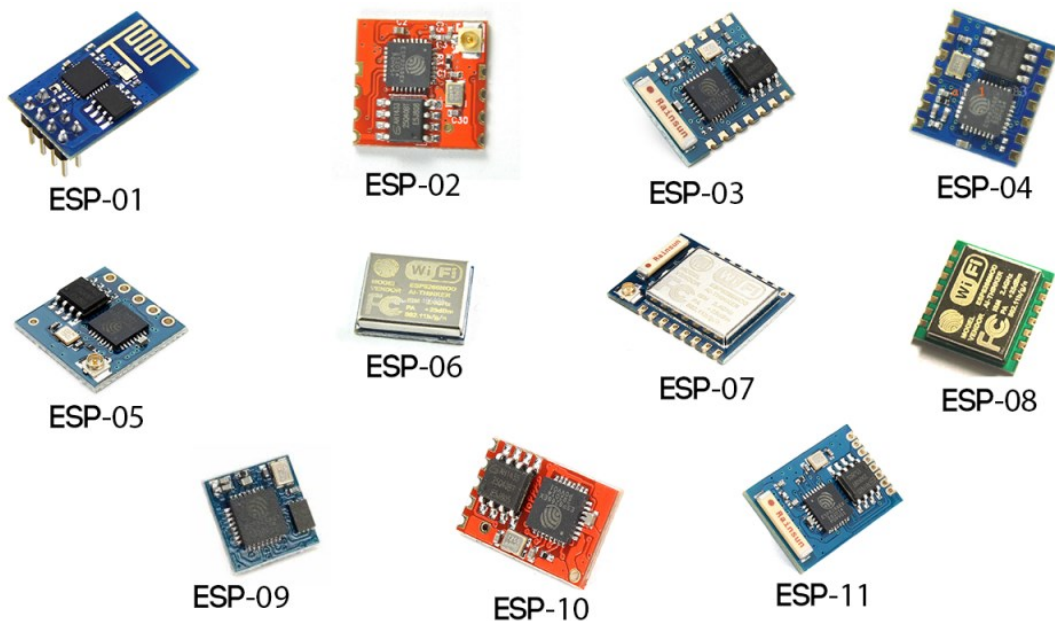
Je číslo, ktoré určuje spojenie pre konkrétnu službu, s ktorou bude komunikácia prebiehať. Identifikátor je obsiahnutý v hlavičke paketu. Ide o 16 bitové celé číslo bez znamienka v rozsahu 0 až 65 535. Niektoré služby používajú pevne určený port, zoznam týchto služieb udržiava IANA. Delia sa na známe porty (rozsah 0 – 1 023), registrované porty (rozsah 1 024 – 49 151) a dynamické alebo prívätne porty (rozsah 49 152 – 65 535). [12;14]

3 WI-FI MODUL ESP8266

ESP8266 je lacný Wi-Fi modul pracujúci v pásme 2,4 Ghz. Vyznačuje sa prijateľnou spotrebou a rozmermi, výrobcom je Espressif Systems. Ide o čínsku spoločnosť zaoberajúcu sa návrhom a výrobou mikročipov.

Výhodou modulu je jeho výkon ale aj možnosť uloženia programu priamo do pamäte, napríklad program pre ovládanie osvetlenia. Modul sa teší rozsiahlej podpore užívateľov, ktorá ho predurčuje pre projekty IoT. Materiály pre vývoj alebo úpravu Firmwaru sú voľne dostupné od výrobcu. Samotný čip ESP8266 je malých rozmerov. Preto jeho samotné použitie by bolo pre domáce podmienky náročné. Existuje ale 14 verzii modulu, ktoré sa líšia svojím prevedením (rozmerom), počtom I/O, typom antény, napájaním a pod. Výber vhodnej verzie závisí od použitia.

O komunikáciu po sieti sa stará samotný čip, užívateľ programuje dáta, ktoré budú posielané prípadne sa stará o spracovanie prijatých dát. Je ho možné používať ako samostatný obvod alebo ako rozširujúci modul napríklad pre platformu Arduino alebo Raspberry pre pripojenie alebo vytvorenie Wi-Fi siete. Čip umožňuje režim AP alebo STA. Režim AP je použiteľný pre vytvorenie Wi-Fi siete a následne pripojenie klienta, režim STA je režim klienta, teda klient sa pripojí na AP. Štandardne je modul vybavený základným AT firmware, ktorý slúži na komunikáciu pomocou AT príkazov. [11]



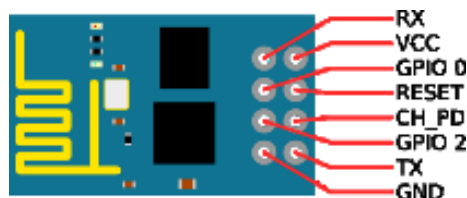
Obr. 7. Verzie modulu ESP8266

3.1 Technické informácie ESP8266-01

Tab. 2. Technické informácie pre ESP8266-01 [11]

Technické informácie ESP8266-0	
Mikrokontrolér	Tensilica L106 32-bit
Taktovacia frekvencia	80 MHz
Flash Pamäť	1MB
Komunikácia	SPI, I ² C, UART
Režim Wi-Fi:	802.11 b/g/n
Serial/UART komunikácia, rýchlosť	115200 Bd
Integrovaný TCP/IP protokol	Ano
Počet I/O	2
Vstupné napájanie	3.3 V (Napájanie príslušenstva 3.3 V)
I/O tolerancia	3.6 V Max
Regulovaný odber prúdu	~70 mA
Maximálny odber prúdu	~300 mA
Výstupný výkon 802.11b mód	+19.5 dBm
Flash pamäť veľkosť	1MB (8Mbit)
WiFi zabezpečenie	WPA/WPA2 PSK
WiFi režimy	Station/SoftAP/SoftAP+Station
Typ antény	2x1 MIMO
Rozmery (d x v)	24.75 mm x 14.5 mm

3.1.1 Napájanie, Pinout schéma



Obr. 8. Schéma výstupných pinov ESP8266-01

Na doske je osadený čip ESP8266EX (Tensilica L106 32-bit) s predvolenou frekvenciou 80 MHz (možnosť pretaktovať na 160 MHz), ktorý disponuje s externou pamäťou 1MB pripojenou po SPI zbernici. Modul obsahuje 2 GPIO piny pre pripojenie senzorov alebo periférneho zariadenia. Ďalej piny sériovej zbernice RX a TX slúžiacie na komunikáciu s čipom. RESET pin. Napájacie piny VCC a GND. Pin CH_PD určujúci režim čipu. Obvod neobsahuje stabilizátor ani regulátor napätia, preto ho je nutné napájať maximálnym napätím 3.6 V aby nedošlo k poškodeniu čipu. Operačná teplota čipu sa pohybuje od -40 °C do 125 °C. Čip

obsahuje vstavanú PCB anténu pre Wi-Fi čo uľahčuje použitie. Bez možnosti pripojenia externej antény. Výhodou však zostáva jej veľkosť a dostačujúci výkon.

VCC a GND – napájacie piny. Napájanie dosky ESP8266-01 je napätím maximálne 3.6 V. Táto hodnota nesmie byť prekročená aby nedošlo k poškodeniu čipu. Spotreba sa pohybuje pri komunikácii okolo 210 mA.

CH_PD - režim čipu určujeme pripojením na napájanie 3,3 V ako aktívny, alebo deep-sleep režim pripojením na GND.

RX/TX - komunikačné sériové rozhranie, používa sa pre komunikáciu a ovládania modulu ESP8266. Logika pre komunikáciu je 3.3 V

GPIO_0 GPIO_2 sú piny I/O pre pripojenie zariadení. Na modul je teda možné napojiť napríklad relé alebo LED diódu a ovládať ju jednoduchými príkazmi. Modul umožňuje použitie týchto pinov pre pripojenie na I2C zbernicu. Taktiež pripojením *GPIO_0* na GND vstupujeme do Flash módu.

RESET - doska obsahuje reset PIN pre možnosť externého resetu.

3.1.2 AT príkazy pre ESP 8266

Zoznam nižšie udáva základné príkazy modulu pre ovládanie a konfiguráciu. Pomocou týchto príkazov prebieha komunikácia modulu a Arduina po sériovej linke. Príkazy sú odosielane, prípadne prijímané modulom a následne spracované. Podrobný výpis príkazov je daný podľa verzie FW, vyššie verzie obsahujú špeciálne alebo rozšírené príkazy pre ovládanie modulu. Tabuľka popisuje príkaz a taktiež udáva vzor použitia.

Tab. 3. ESP8266 Zoznam základných AT príkazov [10]

ESP8266 AT príkazy		
Funkcia	AT príkaz	Odpoveď / poznámka
Test prítomnosti	AT	OK
Reset zariadenia	AT+RST	OK [System Ready]
Firmware verzia	AT+GMR	Vráti verziu FW
Dostupné AP	AT+CWLAP	(kanál, meno, sila signálu, MAC adresa)
Pripojenie na AP	AT+CWJAP?	Vypíše stav pripojenia
	AT+CWJAP="SSID","Password"	Pripojí na dané AP
Odpojenie AP	AT+CWQAP	OK
Zistenie IP adresy modulu	AT+CIFSR	Vypíše priradenú IP adresu
Vytvorenie AP	AT+ CWSAP?	Vypíše údaje o AP
	AT+ CWSAP= <ssid>,<pwd>,<chl>,<ecn>	Vytvorí AP s danými parametrami
WiFi Mode	AT+CWMODE?	Vypíše mód zariadenia
	AT+CWMODE=1	STA mód
	AT+CWMODE=2	AP mód
	AT+CWMODE=3	STA + AP mód
Nastavenie TCP/UDP protokol	AT+ CIPMUX?	Výpis aktuálneho nastavenia
	AT+ CIPMUX=0	Jeden klient
	AT+ CIPMUX=1	Multi pripojenie
Nastavenie TCP/UDP protokol	AT+CIPSTART=?	Výpis aktuálneho nastavenia
	(CIPMUX =0) AT+CIPSTART = <type>,<addr>,<port>	Vytvorenie spojenia
	(CIPMUX =1) AT+CIPSTART= <id><type>,<addr>,<port>	Vytvorenie spojenia na danú adresu s daným ID
Poslanie dát	(CIPMUX=0) AT+CIPSEND=<length>	Pošle dáta zariadeniu
	(CIPMUX=1) AT+CIPSEND= <id>,<length>	Pošle dáta na dané zariadenie podľa ID
Ukončenie spojenia TCP/UDP	AT+CIPCLOSE	Zatvorí všetky spojenia
	AT+CIPCLOSE=<id>	Zatvorí spojenie na danom ID
Nastavenie Servera	AT+ CIPSERVER= <mode>[,<port>]	Spusti server na danom porte

3.2 Použitie na platforme Arduino

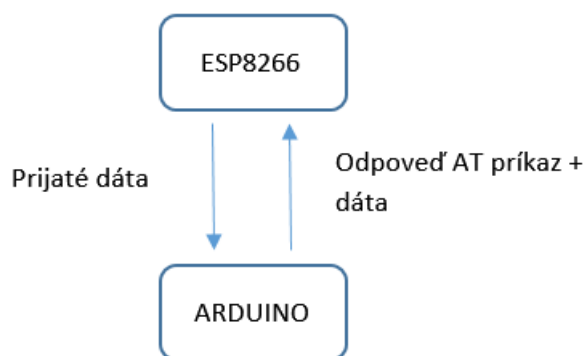
Použitie modulu ESP8266 na platforme Arduino umožňuje pripojenie, ovládanie, konfiguráciu a diagnostiku pomocou bezdrôtového pripojenia v sieti Wi-Fi. Toto rozšírenie otvára nové možnosti v oblasti diaľkovej správy a prenosu dát. Modul ESP8266-01 je lacným a funkčným pripojením do siete Wi-Fi, podporuje moderné štandardy predpisu IEEE 802.11n. Vyznačuje sa nízkou spotrebou, jednoduchou konfiguráciou a nízkou cenou. Modul ESP je možné priamo programovať pomocou vývojového prostredia Arduino IDE, je nutné doinštalovať potrebný plugin pre podporu ESP modulu. Taktiež je potrebný externý UART prevodník pre komunikáciu a nahratie kódu.

3.2.1 Remote - režim

Modul ESP môže v spojení s Arduino platformou pracovať v režime ovládača. Celá réžia komunikácie prebieha na strane Arduina, modul ESP je ovládaný pomocou Firmware podporujúceho AT príkazy. To nám zabezpečí ovládateľnosť modulu pomocou príkazov. Príkazy sú posielané cez sériovú linku. V prípade Aruino UNO je potrebné použiť softwarovú sériovú linku. AT príkazy sú jednoduché príkazy ovládajúce funkcie modulu, súbor použiteľných príkazov a taktiež ich všeobecný predpis je možné nájsť na stránkach výrobcu. [10]

Princíp je nasledovný, pre tento režim je potrebný externý riadiaci systém, ktorý sa stará o obsluhu modulu ESP8266 a o spracovanie dát. Externý systém spustí inicializáciu tá obsahuje príkazy pre nastavenie ESP modulu, odovzdá mu teda základné nastavenia, ktoré definujú režim čipu, názov Wi-Fi siete, heslo, port a iné. Následne sú vykonané vnútorné nastavenia obsluhy. Ako napríklad vnútorná inicializácia I/O pinov. Potom riadiaci systém sleduje prijaté dáta na sériovej zbernici. ESP8266 posiela stavové dáta a tiež dáta po pripojení klienta. Riadiaci systém vykoná požadované operácie a vygeneruje odpoveď. Následne posiela odpoveď vo forme AT príkazov, ktorými upúta pozornosť modulu a dáta mu predá. Modul ESP8266 spracuje prijaté dáta na sieťovej vrstve a pošle ich prostredníctvom Wi-Fi siete klientovi.

Výhody tohto režimu je písanie kódu iba pre platformu Arduino a teda menší čas a prostriedky potrebné pre vývoj ďalej priame ovládanie a teda konkrétnejšia kontrola čipu.

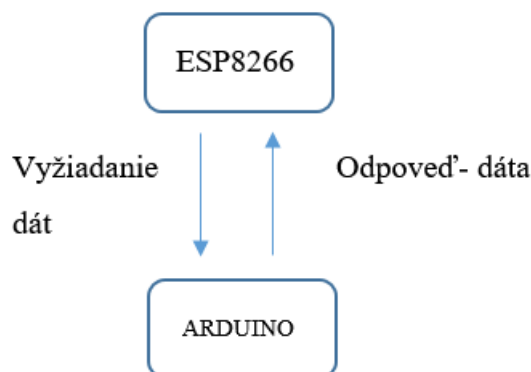


Obr. 9. Princíp komunikácie ESP8266
a ARDUINO režim Remote

3.2.2 Server - režim

Druhým režimom je samostatná práca modulu ESP ako nezávislého prvku starajúceho sa o komunikáciu, Arduino je v tomto prípade len ako rozhranie pre zariadenia, ktoré sú pripojené ako externé alebo ich počet je vyšší ako by bolo možné pripojiť na modul ESP8266. Arduino sa stará sa o meranie signálov a ich spracovanie, následne dáta posiela po sériovej linke na modul ESP. Dáta, ktoré sú posielané predstavujú hodnoty ktoré si vyžiadal modul ESP. Modul ma na starosti prenos po sieti a obsluhu pripojených klientov. Arduino v tomto prípade neobsluhuje klientov a ich požiadavky, nie sú využité AT príkazy, iba poskytuje namerané a spracované dáta. Dáta sú teda v dohodnutom formáte posielané sériovou linkou.

Pre prácu s modulom ESP8266 je potrebný výber vhodného Firmware podporujúci napríklad skriptovací jazyk LUA, príkladom je NodeMCU. Ten je potrebné do modulu nahráť a následne je možné modul programovať. Modul ESP je možné programovať v prostredí Arduino. Jazyk LUA je jednoduchý programovací jazyk, ktorý sa používa pri práci s grafickými operáciami v hrách, animáciách alebo napríklad pre ovládanie robotov. Kód je kompilovaný do bytecode, tento kód je následne spustený na virtuálnom stroji LUA Machine v ESP module. Kód sa nahráva do Flash pamäte modulu cez prevodník, nutné použiť externý keďže niektoré verzie ESP8266 prevodník nemajú. Následne je spustená inicializácia a potom časť nekonečného cyklu. Výhoda je hlavne v rýchlosti, použitie tohto riešenia je vhodné z hľadiska výkonu. Výkon zariadenia je tak rozdelený na dve časti. Arduino môže vykonávať zložitejšie operácie napríklad spracovanie nameraných dát, ich zaokrúhlenie, výpočet. Obsahovať väčší počet zariadení, alebo v neposlednom rade modularita systému. Nevýhodou je potreba písania samostatného kódu aj pre modul ESP a taktiež pre platformu Arduino.



Obr. 10. Princíp komunikácie ESP8266
a ARDUINO

II. PRAKTICKÁ ČÁST

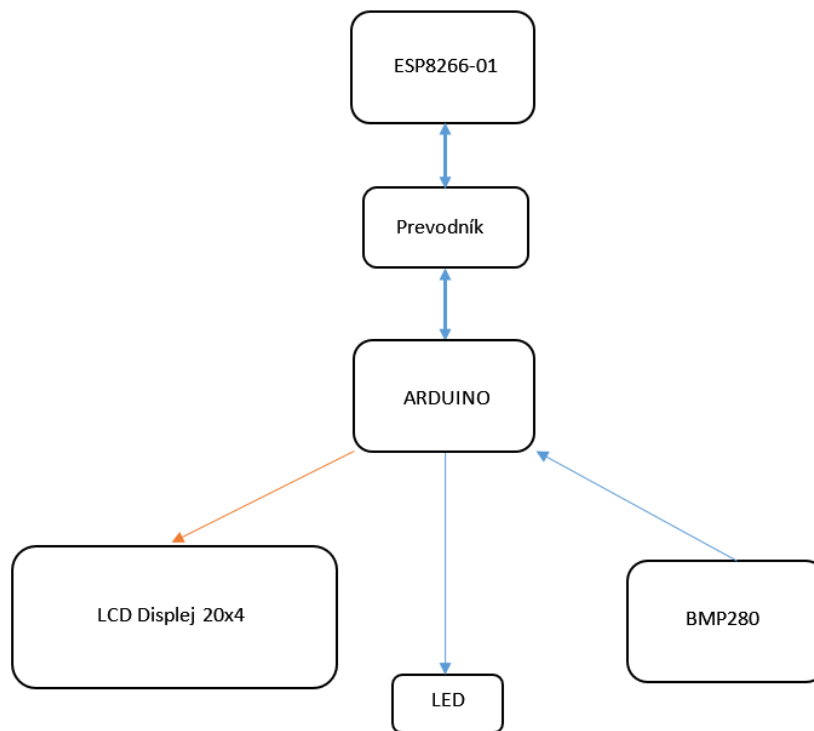
4 REALIZÁCIA

Úlohou je návrh zapojenia prototypu obvodu s pripojeným senzora fyzikálnych veličín. Vypracovanie návrhu zapojenia, testovanie a realizácia zapojenia. Overenie funkčnosti kódu.

Zvolil som režim Remote, Arduino v tomto prípade bude ovládať samotný ESP modul pomocou AT príkazov a starať sa o obsluhu klientov, ďalej bude mať na starosti meranie a spracovanie dát. Vhodnou formou ich bude poskytovať klientovi, ktorý o dáta požiada. Klient bude reprezentovaný JAVA aplikáciou. ESP modul bude komunikovať s klientom pomocou TCP protokolu. Tento protokol je vhodný na túto aplikáciu, prináša spoľahlivé spojenie a doručenie dát. Nie je potrebné použiť protokol UDP, objem prenášaných dát bude minimálny a je nutne zaručiť doručenie dát a minimalizáciu chýb.

4.1 Návrh zapojenia

Pri výbere prvkov a spôsobu zapojenia modulu ESP8266 na platformu Arduino UNO som bral v úvahu jednoduchosť a možnosť univerzálnosti aplikácie pre neskoršie použitie pre konkrétnu aplikáciu. Použitie SoftwareSerial knižnice pre komunikáciu pomocou digitálnych I/O pinov je dostačujúce pre bežnú komunikáciu a ovládanie. Táto knižnica je už obsiahnutá v základnom balíku vývojového prostredia. V súvislosti s touto myšlienkou som musel zvoliť vhodné komponenty a spôsob ich zapojenia. Pre komunikáciu s ESP8266 je potrebné použiť Logic Level Converter – ide o prevodník úrovni signálu, Arduino používa 5 V logiku a modul ESP8266 používa 3.3 V úroveň. Preto je potrebné toto zapojenie prevodníku do UART komunikácie. Postará sa o prevod napät'ovej úrovne pre jednotlivé signály a teda zabezpečí korektnú komunikáciu. Stabilita modulu ESP8266 je ovplyvnená kvalitou napájania, spotreba čipu je počas komunikácie zvýšená a vyžaduje sa kvalitný konštantný zdroj energie. Použil som preto externý zdroj 5 V/1 A so stabilizátorom na 3.3 V. Senzor BMP280, ktorý sníma teplotu a tlak bude pripojený na I²C zbernicu. Displej zobrazujúci základné údaje bude taktiež pripojený pomocou I²C zbernice.



Obr. 11. Bloková schéma zapojenia

Návrh základných funkcií:

- Pripojenie na bežné Wi-Fi siete
- Jednoduchý výpis na displej
- Spracovanie nameraných dát
- Meranie teploty, tlaku

4.1.1 Použité komponenty

Arduino UNO R3

Wi-Fi modul ESP8266-01

Displej s I2C zbernicou 20x4 znakov

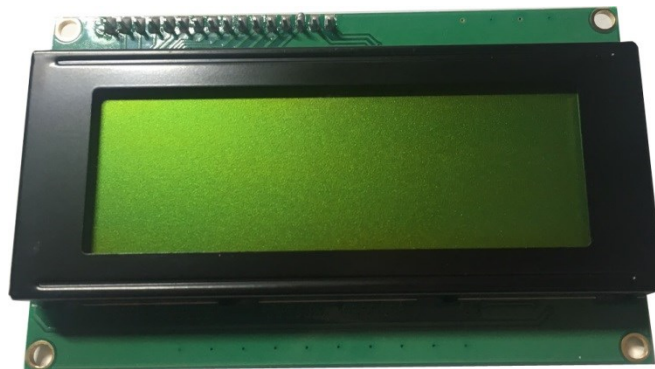
Senzor teploty, tlaku, typ BMP280

Univerzálna vývojová doska

LED dióda, červená, 3 mm

Displej s I²C zbernicou 20x4 znakov

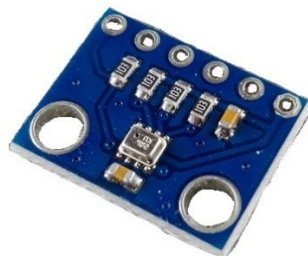
LCD displej s podsvietením, komunikujúci pomocou I²C zbernice. Je schopný zobrazit' 20 znakov na 4 riadky. Kontrast displeja je možné nastavovať trimrom na zadnej strane. Taktiež je možné zapínať a vypínať podsvietenie displeja. K displeju je možné použiť knižnice dostupne ako Open-source obsahujúce znaky abecedy ASCII. Tiež je možné vytvoriť vlastné znaky a používať ich. Výhodou displeja je jednoduchosť obsluhy a nenáročnosť na požiadavky napájania. V projekte je použitý na zobrazenie základných údajov a stavov.



Obr. 12. Displej LCD 20x4 znakov s I²C zbernicou

Senzor teploty, tlaku typ BMP280

Modul pre meranie teploty a barometrického tlaku so senzorom BMP280 od firmy BOSCH. Modul komunikuje cez I²C zbernicu do rýchlosti 3,4 MHz. Prevádzkovateľný rozsah teplôt sa pohybuje v rozmedzí $-40\text{ }^{\circ}\text{C}$ až $+85\text{ }^{\circ}\text{C}$, tlak je možné merať v rozsahu 300 hPa až 1100 hPa. Rozlišovacia schopnosť senzora je $0,01\text{ }^{\circ}\text{C}$ a $0,16\text{ Pa}$ pri tlaku. Presnosť teploty sa pohybuje v $\pm 1\text{ }^{\circ}\text{C}$ a $\pm 100\text{ Pa}$ u tlaku. Knižnicu pre modul so senzorom BMP280 je možné stiahnuť ako Open-source kód zo zdroja [18].

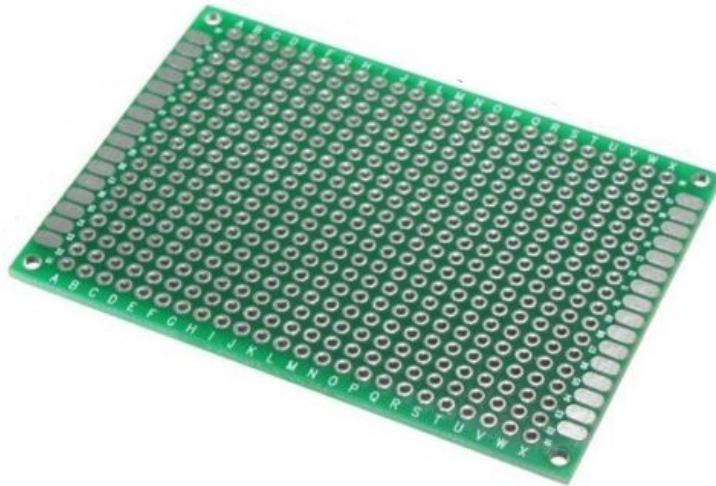


Obr. 13. Senzor

BMP280 [15]

Univerzálna vývojová doska

Univerzálna doska sa používa na vytvorenie prototypu projektu a jeho testovanie. Na účel tohto projektu som zvolil dosku s rozmerom 10 x 14 cm. S možnosťou obojstranného spájovania.

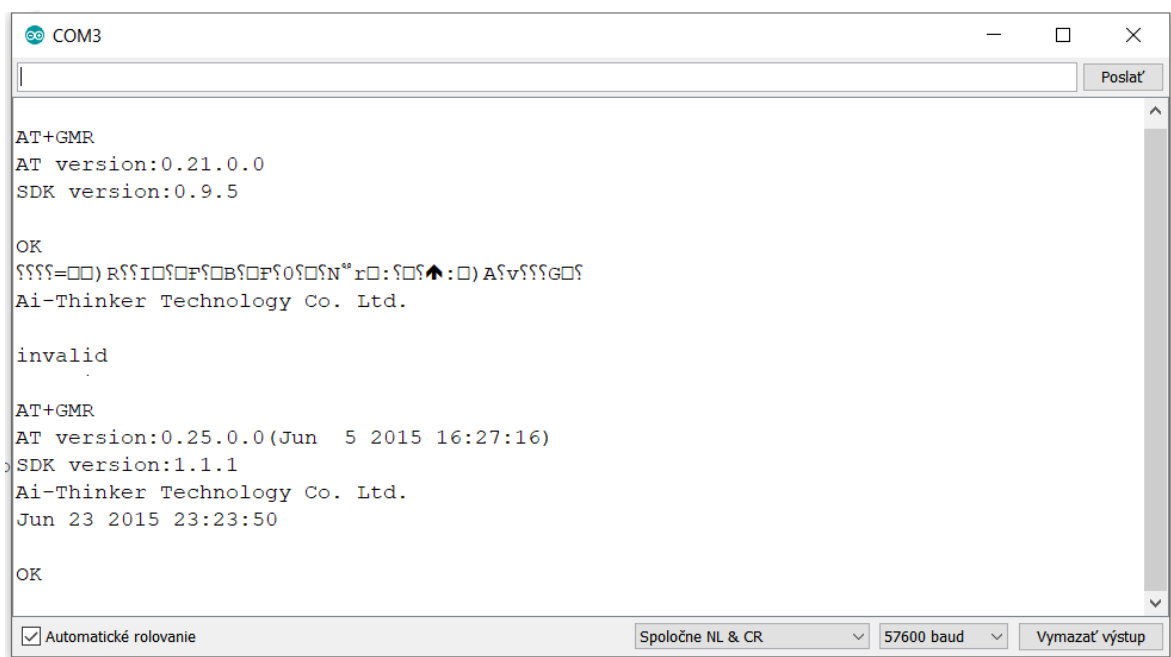


Obr. 14. Univerzálna vývojová doska

4.1.2 Test spojenia

Na účel testovania spojenia s ESP8266 modulom som vytvoril jednoduchý program pre Arduino. Slúži na testovanie spojenia, nastavenie a ovládanie. Ide o doplnkový program, ktorý vznikol len pre podporu a overenie. Myslím ale, že je dobré ho uviesť v prípade potreby je možné ho použiť na komunikáciu s iným zariadením na Software Serial zbernici. Program načíta znaky zadané na klávesnici a pošle ich modulu. Následne vypíše odpoveď zariadenia. Použitelnosť je, ak nemáte k dispozícii externý prevodník sériovej komunikácie pre USB.

Zdrojový kód pre Arduino je v Elektronickej prílohe práce.



Obr. 15. Monitor sériovej komunikácie, test spojenia s ESP8266

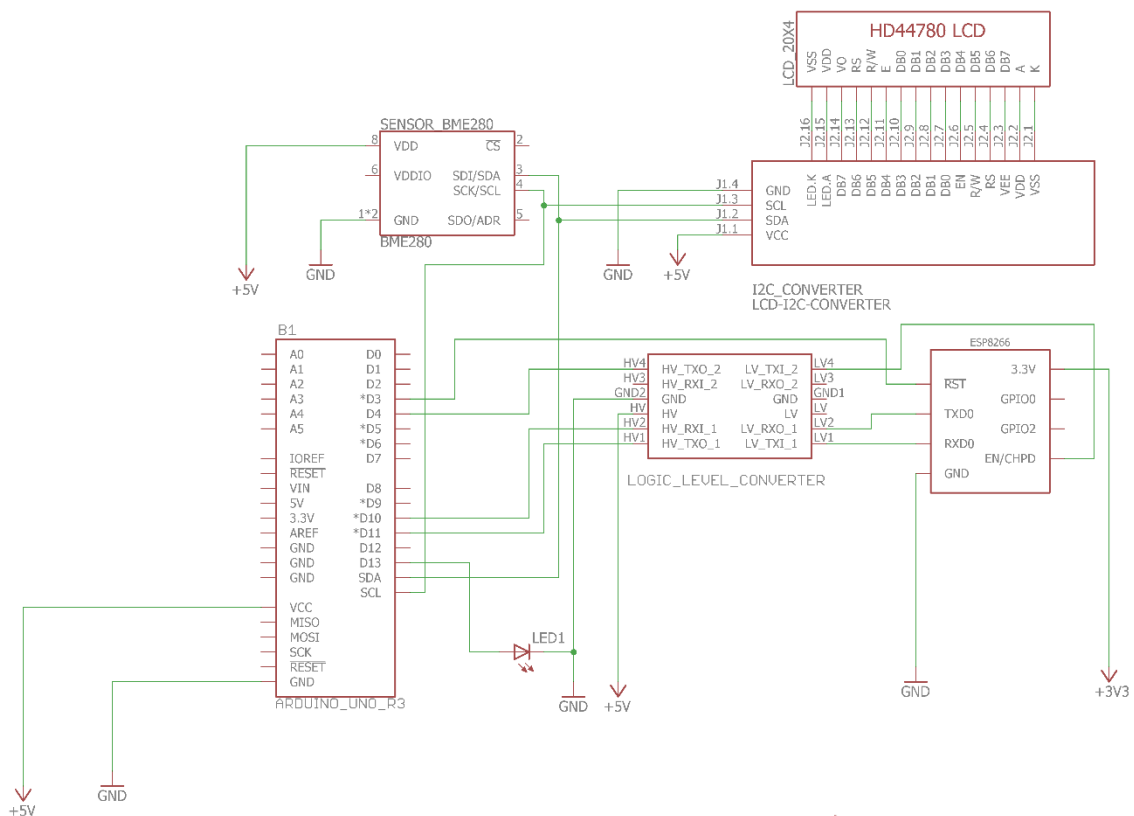
4.1.3 Schéma zapojenia

Kompletná schéma zapojenia je vytvorená pomocou programu EAGLE od spoločnosti AUTODESK [16]. Autodesk je lídrom v oblasti 3D dizajnu, inžinierstva a zábavného softvéru. Schéma obsahuje označenie pinov a spôsob zapojenia. Výhodou tohto programu je priamy export pre výrobu dosky plošného spoja.

Piny D10 a D11 sú vyžívané na komunikáciu s ESP modulom, D4 je pin, ktorý spustí čip do činnosti. Spomínané výstupy sú pomocou LOGIC LEVEL prevodníka prevedené na nižšiu napäťovú úroveň, ktorú používa ESP a to 3,3 V.

Pin D13 je použitý ako ovládací pin, je pripojený na LED diódu slúžiacu len ako testovací výstup (môže byť nahradená napríklad s relé).

SDA a SCL sú výstupy pre sériovú zbernicu, na ktorú je pripojený LCD Displej s adresou 0x27 a čidlo BMP280 s adresou 0x76.

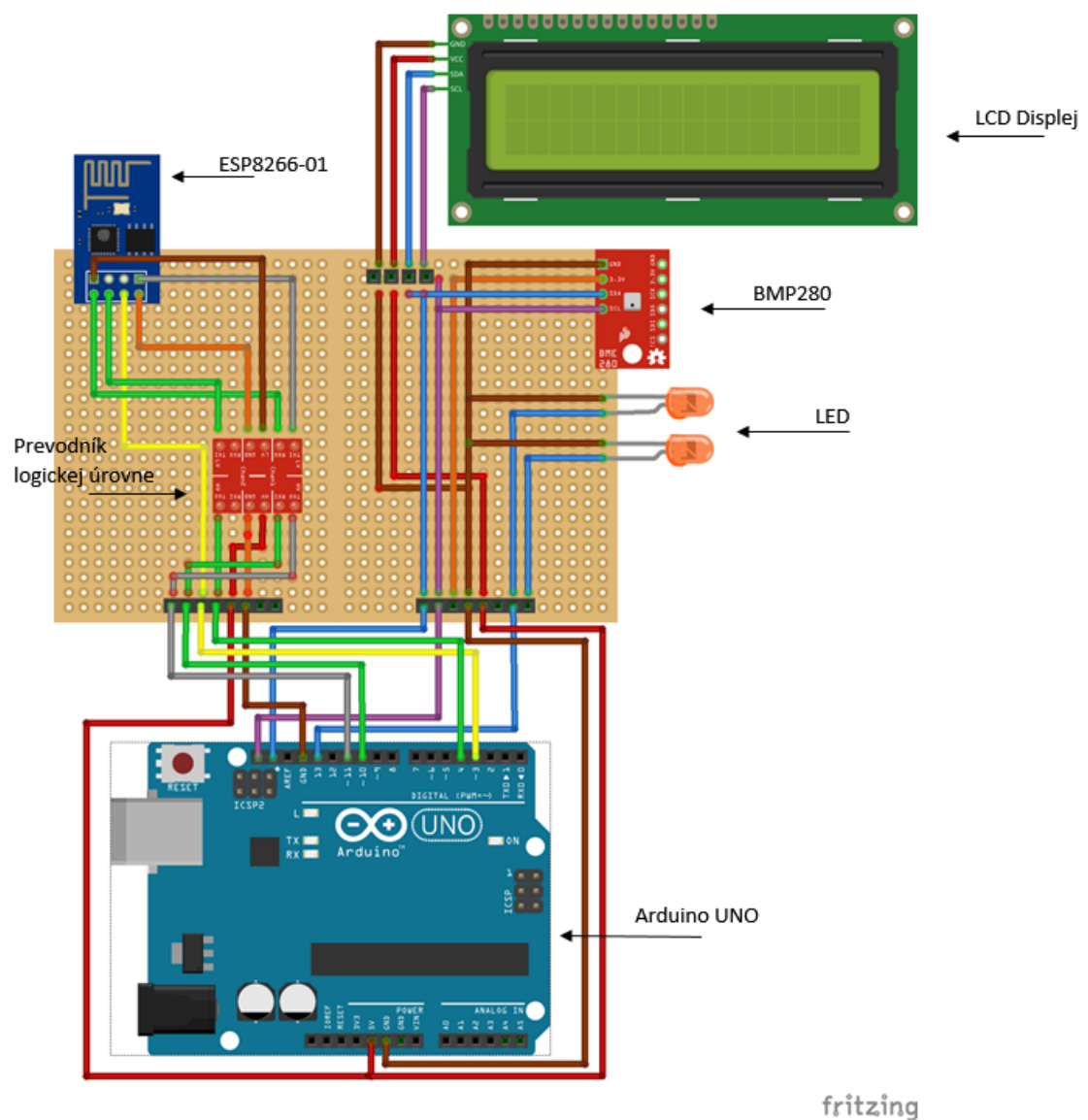


Obr. 16. Schéma zapojenia jednotlivých komponentov

4.1.4 Zostavenie a oživenie

Pre vytvorenie zostavy ako jedného funkčného celku som použil univerzálnu dosku plošného spoja. Je vhodná na riešenie prototypu, odladenie a testovanie. Na grafickú vizualizáciu bol použitý Open-source program Fritzing.

Bloková schéma zapojenia :



Obr. 17. Grafická vizualizácia zapojenia

4.2 Programové vybavenie

Programová časť tvorí platforma Arduino a obslužná aplikácia v programovacom jazyku JAVA. Aplikáciou je predstavená schopnosť spolu komunikovať za dohodnutých pravidiel. Samozrejmosťou je myšlienka neskoršej podpory mobilnej integrácie a teda možnosť napríklad ovládania na smart zariadeniach ako sú tablety, smartfóny a podobne. Návrh som však koncipoval na použitie v osobných počítačoch a tak aplikácia je vytvorená práve pre osobné počítače.

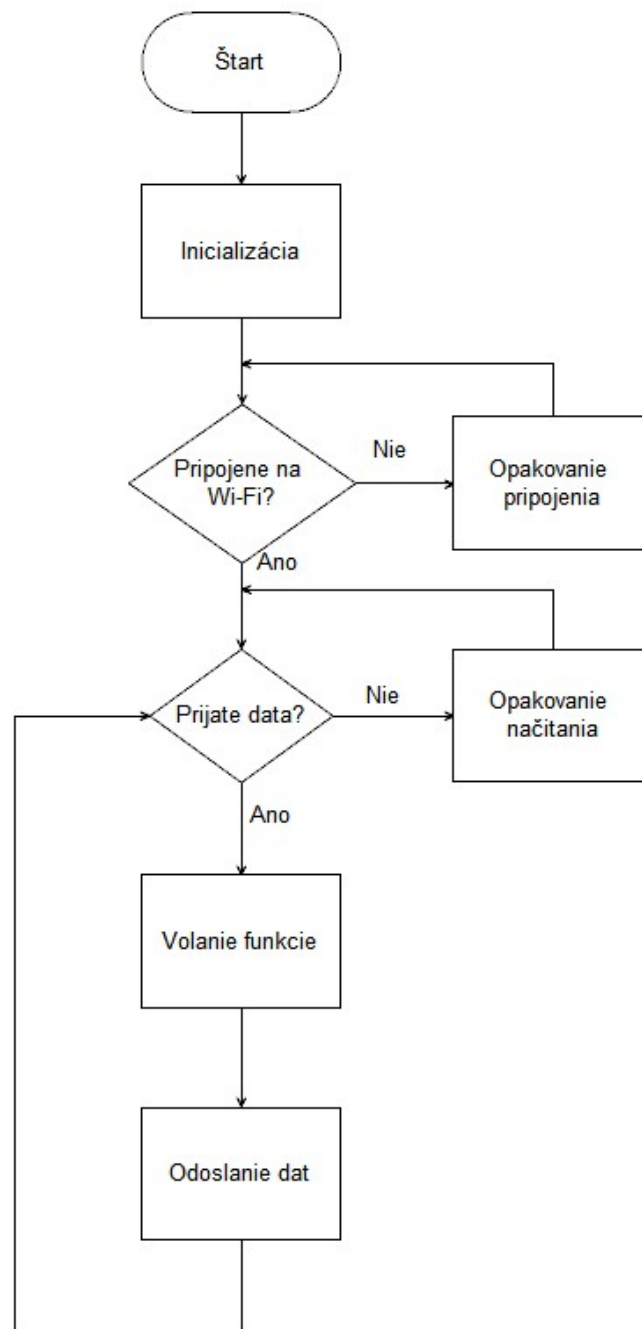
4.2.1 Arduino

Pre celkovú funkčnosť celého systému bolo nutné vytvoriť program pre mikropočítač, ktorý bude obsluhovať komunikáciu cez Wi-Fi pomocou TCP protokolu – jeho úlohou je spracovanie a odpoveď na požiadavky prijaté od ovládacieho programu, ďalej bude pracovať s nameranými dátami a obsluhovať snímače pomocou funkcií.

Popis procesu je nasledovný: Arduino testuje prítomnosť dát na softvérovej sériovej linke od ESP8266, sériová linka je požitá emulovaná pomocou knižnice Software Serial, ide o voľne dostupnú knižnicu, ktorá slúži na emulovanie sériovej zbernice tam kde nie je možné použiť HW komunikáciu. Táto knižnica je postačujúca na komunikáciu medzi ESP a Arduino UNO. Po prijatí dát po sériovej linke Arduino príkaz spracuje a hľadá výraz „IPD“, ak je úspešný, formát prijatých dát *+IPD, id, length:data*, vyhodnotí ID klienta, ktorý požiadal o danú funkciu. ID klienta je uložené kvôli neskoršej odpovedi do globálnej premennej. Ďalej sú vyhodnotené prijaté dáta, ktoré sa porovnávajú s danými funkciami programu. Ak je nájdená zhoda požiadavku, je zavolaná funkcia, ktorá prevedie požadovanú operáciu, výsledok vráti klientovi, ktorý o ňu požiadal, pomocou funkcie na odoslanie dát, ktorá využíva AT príkaz *AT+CIPSEND= <id>, <length>*. Proces komunikácie prebieha pomocou AT príkazov, ktoré sú odoslane po sériovej linke do ESP. Celý program beží v nekonečnej slučke a teda je opakovaný.

Pre jednoduchšiu obsluhu, ale aj prehľadnosť programu, som vytvoril funkcie, ktoré sa starajú o inicializáciu, pripojenie a spustenie. Taktiež je vytvorená funkcia na odoslanie dáta. Tieto funkcie uľahčujú prácu pri neskoršom možnom editovaní a prispôbovaní pre konkrétnu aplikáciu.

Vývojový diagram pre komunikáciu s ESP modulom a generovaním odpovedí:



Obr. 18. Vývojový diagram pre program Arduino

Základne definovanie premenných a použitých knižníc:

```
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
//vytvorenie objektu pre BMP280 cidloa
Adafruit_BMP280 bmp;
/*
    definovanie pouziteho LCD Displeja a jeho adresa
    0x27 adresa na I2C zbernici
    20 počet znakov na displeji
    4 počet riadkov displeja
*/
LiquidCrystal_I2C lcd(0x27, 20, 4);
/*
    SSID definovanie názvu Wi-Fi siete na ktoru sa modul pripoji
    PASS heslo Wi-Fi siete
*/
#define SSID "SSID"
#define PASS "HESLO"
/*
    The circuit:
    RX je digitalny pin 10 (pripojit na TX pin modulu)
    TX je digitalnypin 11 (pripojit na RX pin modulu)
*/
SoftwareSerial dbgSerial(10, 11); // RX, TX

// pocitadlo stavu LED diody
int counter = 0;
// Stav led diody
bool statusLED = false;
// Id pripojenia
int connectionId;
// IP a MAC adresa modulu
String IPadresa = "";
```

Obr. 19. Základne definovanie globálnych
premenných a použitých knižníc

Definovaný parameter *SSID* označuje názov Wi-Fi siete, ku ktorej sa má pripojiť ESP modul. Parameter *PASS* je heslo Wi-Fi siete, ktoré patrí k danej sieti. Ďalej je definované typ pripojeného LCD displeja a jeho adresa. Vytvorený objekt *bmp* patrí pre BMP280 čidlo. Ďalej je definovaná komunikácia pre SoftwareSerial na pinoch *D10* – predstavujúce TX a *D11* predstavujúce RX. Ďalšími sú globálne premenné pre uloženie stavu alebo informácií všeobecne potrebných alebo prospešných.

Inicializačné funkcie programu

```
void espServer(int port) {  
    Serial.println("ESP8266 multi pripojenie");  
    sendCommand("AT+CIPMUX=1\r\n", 1000);  
    delay(500);  
    Serial.print("ESP8266 konfiguracia serveru na porte ");  
    Serial.println(port);  
    String cmd = "AT+CIPSERVER=1,";  
    cmd += port;  
    cmd += "\r\n";  
    sendCommand(cmd, 1000);  
    delay(500);  
    sendCommand("AT+CIPSTO=7100\r\n", 1000); // cas po ktorom server ukonci spojenie  
    delay(500);  
}
```

Obr. 20. ukážka funkcie pre konfigurovanie ESP8266 modulu v režime server

4.2.1.1 Zoznam vytvorených funkcií a ich popis

`connectWiFi()` funkcia slúži na pripojenie na danú Wi-Fi sieť. Pre pripojenie na Wi-Fi sieť je vyžadované definovanie meno siete *SSID* a jej heslo *PASS*. Funkcia vracia návratovú hodnotu *true*, ak bol pokus o pripojenie úspešný, ak nie, návratová hodnota je *false*. Definovaný je čas odpovede na 15 sekúnd.

`sendCommand()` slúži na komunikáciu po softwarovo sériovej linke. Má na starosti odoslanie príkazu. Vstupný parameter je príkaz – *command*, ktorý je typ `String` a bude poslaný modulu, ďalej čas, ktorý má modul na odpoveď definovaný ako *timeout*. Odpovede modulu sú vypísané na sériovú linku Arduina.

`sendData()` funkcia slúži na odosielanie dát po sériovej linke, bola vytvorená pre samotné poslanie dát. Po odoslaní príkazu vracia *true*. Vstupné parametre sú vo formáte `String` s názvom *command* a *timeout*, ktorý definuje čas na odpoveď modulu.

```
bool sendData(String command, const int timeout)
{
    dbgSerial.print(command);
    unsigned long time = millis();
    while ( (time + timeout) > millis())
    {
        ;
    }
    Serial.println("Data odoslane");
    return true;
}

void sendCommand(String command, const int timeout)
{
    unsigned long time = millis();
    dbgSerial.print(command);
    while ( (time + timeout) > millis()) {
        if (dbgSerial.available()) {
            Serial.write(dbgSerial.read());
        }
    }
    Serial.println("Prikaz odoslany");
}
```

Obr. 21. Ukážka kódu funkcií `sendData()` a `sendCommand()`

- `sendIPData()` funkcia zostavuje dáta do predpísaného vzoru pre poslanie modulu ESP. Vstupné parametre musia obsahovať ID - *connectionId* pripojenia a samotné dáta. Dĺžka dát je spočítaná pomocou funkcie *length()*. Pridáva ukončovacie znaky pre sériovú komunikáciu a upravuje správny formát príkazu. Následne využíva funkciu *sendData()* pre poslanie dát do modulu ESP. Funkcia pošle hlavičku a následne dáta.
- `getIP()` funkcia slúži na zistenie IP adresy, ktorá bola pridelená modulu ESP po pripojení na Wi-Fi prístupový bod. IP adresa je uložená do globálnej premennej *IPadresa*.
- `vypisIP()` načíta hodnotu z globálnej premennej *IPadresa* a vykona úpravu formátu. Následne vypíše IP adresu na displej a sériovú linku.
- `manageLED()` funkcia ktorá ovláda digitálny pin definovaný pri inicializácii. Podľa predošlého stavu nastaví stav nový a stav pinu zapíše do globálnej premennej *statusLED*

getTemperature() funkcia pre zistenie teploty z čidla a následné spracovanie. Výpis na sériovú linku. Dáta z čidla spracuje a prevedie na formát vhodný pre poslanie. Následne dáta pošle klientovi.

getPressure() funkcia pre obsluhu čidla BMP280 a zistenie barometrického tlaku. Výpis na sériovú linku. Dáta z čidla spracuje a prevedie na formát vhodný pre poslanie. Následne dáta pošle klientovi.

```
void getPressure(bool info) {
    Serial.println("getPressure");
    String pressure = String(bmp.readPressure());
    sendIPData(connectionId, pressure);

    if (info) {
        Serial.print("Pressure = ");
        Serial.println(pressure);
        //lcd.clear();
        lcd.setCursor(0, 2);
        lcd.print("getPressure");
    }
}
```

Obr. 22. Ukážka kódu funkcie getPressure()

4.2.1.2 Použité funkcie a knižnice

lcd.print() funkcia z knižnice LiquidCrystal_I2C.h ,ktorá je voľne dostupná pre ovládanie LCD displeja. Slúži na zobrazovanie znakov na displej.

lcd.setCursor() nastaví polohu kurzora na dané miesto. Od tohto miesta bude vypisovať text. Slúži na formátovanie textu.

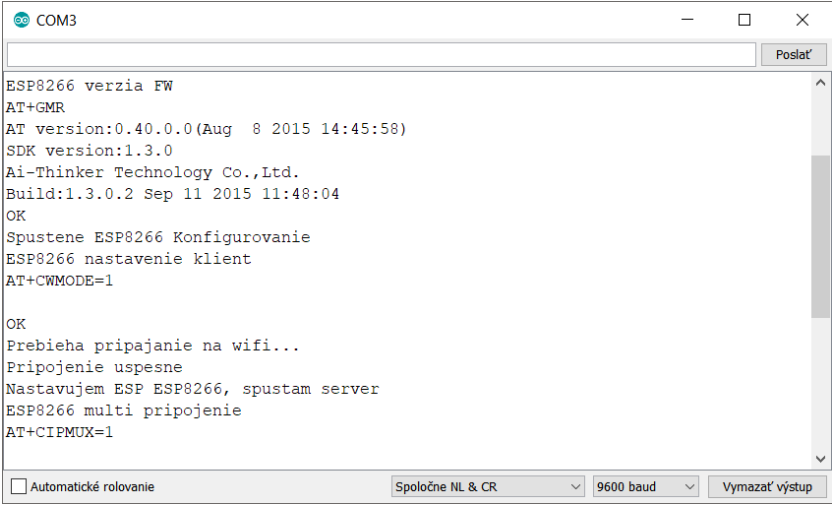
lcd.clear() vymaže obsah zobrazený na LCD displeji.

dbgSerial.available() funkcia pre detekciu prijatých dát na SoftwareSerial linke, ak sú dáta prijate vráti *true*.

dbgSerial.find() hľadá dané znaky parametrom v dátach prijatých v sériovej linke

dbgSerial.print() výpis na SoftwareSerial linku v ASCII ako čitateľný text

- `Serial.println()` výpis na SoftwareSerial linku so znakom nový riadok
- `Serial.write()` zapisuje binárne údaje do sériového portu. Tieto údaje sa odosielajú ako bajt alebo séria bajtov
- `delay()` pozastaví program na čas (v milisekundách) určený ako parameter funkcie.
- `millis()` vráti počet milisekúnd, odkedy doska Arduino spustila aktuálny program. Toto číslo sa po približne 50 dňoch prepadne (vráti na nulu).
- `digitalWrite(pin, value)` ak je pin nakonfigurovaný ako OUTPUT(výstupný), jeho napätie bude nastavené na hodnotu: 5 V pre parameter - *HIGH* , 0V (ground) pre parameter - *LOW* .
- `bmp.readPressure()` funkcia z knižnice Adafruit_BMP280 [16] slúži na získanie teploty z pripojeného senzoru tlaku a teploty. Návrátová hodnota je teplota v °C.
- `bmp.readTemperature()` podobne ako funkcia `readPressure()` je taktiež z knižnice Adafruit_BMP280 [16], táto funkcia však slúži na získanie barometrického tlaku v Pa.



```
COM3
ESP8266 verzia FW
AT+GMR
AT version:0.40.0.0(Aug 8 2015 14:45:58)
SDK version:1.3.0
Ai-Thinker Technology Co.,Ltd.
Build:1.3.0.2 Sep 11 2015 11:48:04
OK
Spustene ESP8266 Konfigurovanie
ESP8266 nastavenie klient
AT+CWMODE=1
OK
Prebieha pripajanie na wifi...
Pripojenie uspesne
Nastavujem ESP ESP8266, spustam server
ESP8266 multi pripojenie
AT+CIPMUX=1
```

Automatické rolovanie Spoločne NL & CR 9600 baud Vymazať výstup

Obr. 23. Výstup na sériovú linku

4.2.2 Java aplikácia

Aplikácia slúži ako rozhranie pre užívateľa a hardware. Má na starosti spojenie a spracovanie dát, taktiež sa stará o komunikáciu. Aplikácia obsahuje základné funkcie ktoré je možné rozšíriť. Funkcie ktoré obsahuje je možné použiť na overenie komunikácie a funkčnosť pripojenia. Je napísaná v programovacom jazyku JAVA. Grafické rozhranie je navrhnuté s dôrazom na jednoduchosť a efektivitu so základnými nastaveniami pripojenia. Keď bolo prioritou pre mňa rýchla komunikácia a teda možnosť komunikovať s čo najmenším oneskorením.

Základný princíp ServerSocket je použitý z nasledujúcej literatúry [19]. Program je rozšírený ale o funkcie potrebné pre komunikáciu so samotným ESP modulom. Obsahuje funkcie pre posielanie dát a taktiež funkciu ktorá sleduje prijaté dáta z ESP modulu.

Server Socket vytvorí spojenie na danú IP adresu a daný port. IP adresa je adresa, ktorá je pridelená modulu ESP8266 po pripojení na Wi-Fi sieť (IP adresa je zobrazená na displeji po pripojení). Číslo portu, je číslo na ktorom beží server (predvolená hodnota 80). Aplikácia pošle ako prvý príkaz *ok ready*, čo značí že je pripravená na komunikáciu. Súbežne je vytvorená funkcia, ktorá beží vo vlákne a má na starosti spracovanie prijatých dát od ESP modulu. Ak dáta prijme, zapíše ich do výstupu aplikácie. Pomocou tlačidiel Ovládanie 1 LED, Ovládanie 2 Teplota a Ovládanie 3 Tlak sa posielajú príkazy.

System komunikácie je postavený na štandarde TCP/IP, takže je možné použiť rôzne aplikácie , ktoré sú schopne pripojiť sa na IP adresu a port. A posielat' obslužné príkazy pre ESP.

Aplikácia dosahovala rýchlu odozvu. Priemer od odoslania požiadavku po prijatí odpovede je 70 ms.

Popis rozhrania a funkcii:

- IP adresa – pole pre zadanie IP adresy pre pripojenie k ESP modulu
- Port – port, na ktorom je spustený server ESP
- Tlačidlo Pripojiť a Odpojiť – slúži na vytvorenie spojenia alebo ukončenie spojenia
- Ovládanie 1 LED – slúži na ovládanie LED diódy, modul informuje o stave LED diódy.
- Ovládanie 2 Teplota – tlačidlo pre poslanie požiadavku o hodnotu teploty, ESP vráti hodnotu teploty, ktorá je zobrazená v časti získané dáta
- Ovládanie 3 tlak – tlačidlo pre poslanie požiadavku o hodnotu tlaku, ESP navráti hodnotu tlaku, ktorá je zobrazená v časti získané dáta.

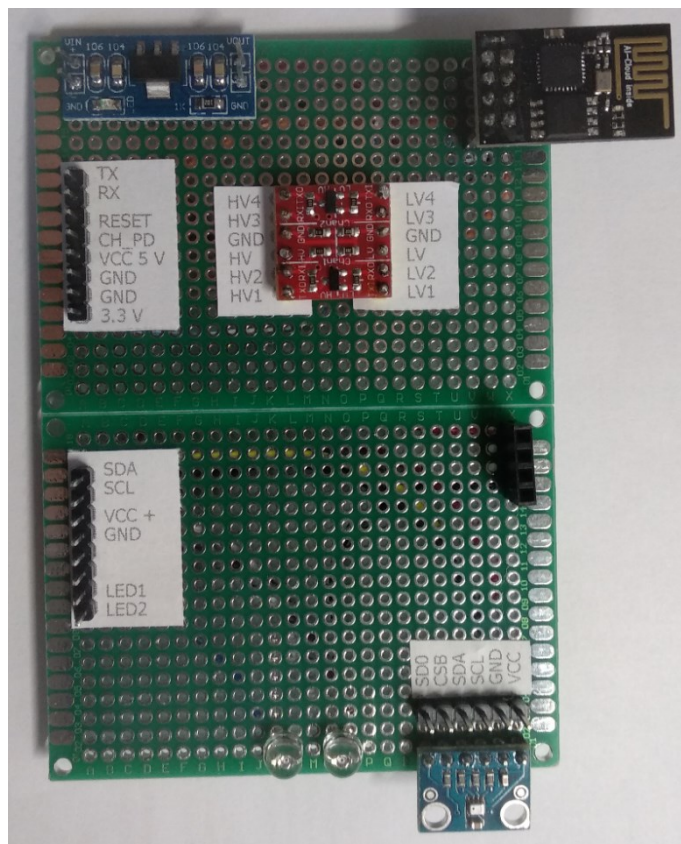
The screenshot shows a window titled "Client_gui" with standard Windows window controls (minimize, maximize, close). The interface is organized into several sections:

- IP adresa zariadenia**: A text input field containing "192.168.1.190".
- Port**: A text input field containing "80".
- Buttons**: Two buttons labeled "Pripojiť" and "Odpojiť".
- Control Buttons**: Three buttons labeled "Ovládanie 1 LED", "Ovládanie 2 Teplota", and "Ovládanie 3 Tlak".
- Získané dáta**: A text area for displaying received data.
- Ver. 1.0**: The application version number at the bottom left.

Obr. 24. Rozhranie obslužnej aplikácie

4.3 Overenie

Finálna podoba vývojovej dosky na platforme Arduino UNO pre komunikáciu pomocou ESP8266-01 modulu. Napájanie Arduino UNO a modulu ESP8266 je realizované externým napájaním pomocou stabilizátora, bola overená stabilita napájania. Pre fungovanie je potrebné kvalitný napájací zdroj. Boli vykonané základné testy funkčnosti a stability pripojenia. Všetky testy boli vykonané s kladným výsledkom a teda pripojenie bolo aj po dňoch stabilné a funkčné. Modul bol počas testovania aj po dlhšej dobe schopný odpovedať na požiadavky klienta a spracovávať dáta. Komunikačný kanál pre výmenu informácií je Wi-Fi sieť vytvorená pomocou AP. ESP modul je pripojený na daný AP pomocou mena a hesla siete a vystupuje ako Server pre komunikáciu. Rozhraním je aplikácia v JAVA alebo iná aplikácia podporujúca dané komunikačné protokoly TCP/IP s možnosťou posielania príkazov a zobrazovania odpovedí. Napríklad projekt pracuje aj s aplikáciami pre Android platformu. A pomocou vybraných aplikácií je možné ich využiť na ovládanie. Ako príklad uvediem aplikáciu Simple TCP Socket Tester dostupný pre Android v obchode Google Play.



Obr. 25. Vývojová doska pre platformu Arduino UNO s modulom ESP8266-01

ZÁVĚR

Za cieľ v tejto práci bolo priblíženie a návrh univerzálnej komunikácie na platforme Arduino použitím modulu ESP8266-01 s využitím sériovej komunikácie medzi modulmi. Riešením bolo použitie hardwarovej sériovej komunikácie modulu Arduino UNO pre komunikáciu s počítačom a použitie softwarovej – virtuálnej komunikácie medzi Arduino UNO a modulom ESP8266-01.

V prvej časti je priblížená platforma Arduino a ESP8266. Obsahuje popis jednotlivých modulov, ich základné vlastnosti a parametre. Ďalej je priblížený spôsob ovládania modulu ESP8266-01 a navrhnutý model pre ich spoločnú komunikáciu. Prezentovaná práca s dátami v sieti, aby bolo možné ich dopraviť z jedného systému do druhého.

V druhej časti som prakticky realizoval zapojenie modulov. Zapojenie som prezentoval na možnosti ovládania prvkov – v mojom prípade som použil LED diódu. A na možnosti získania dát – čidlo BME280 pre meranie teploty a tlaku. Ďalej sú vytvorené funkcie pre zber a spracovanie dát zo senzoru. Následne bol vytvorený prototyp na univerzálnej doske plošného spoja, ktorý bol použitý na vývoj a test. Softwarovú časť tvorí kód pre platformu Arduino, ktorý obsahuje použité funkcie s vysvetlením a popisom ich funkcií. Taktiež práca zahŕňa ukážkovú aplikáciu v jazyku JAVA. Aplikácia je navrhnutá pre spojenie HW a užívateľa pomocou grafického rozhrania.

Priblíženie celkovej komunikácie považujem za prínosnú vec využiteľnú v rôznych smart aplikáciách kde je potrebné diaľkové ovládanie alebo monitoring. Taktiež tu svoje postavenie hrá cena jednotlivých komponentov, ktorá je rádovo do stoviek českých korún, dostupnosť jednotlivých dielov na trhu je veľmi dobrá a tak celý projekt považujem za prístupný s vysokým potenciálom využiteľnosti.

Projekt bol zostavený na vývojovej doske, aby do budúca bolo možné jednoducho rozšíriť alebo upraviť. Finálnu verziu by bolo možné postaviť ako Shield pre Arduino platformu a táto možnosť by zjednodušila ďalší vývoj a úpravu. Ďalší potenciál vidím v možnosti vyučovania a tak zdokonaľovanie znalostí z oblasti mikropočítačov a ich pripojenie do siete. Nahradením modulu Arduino UNO je možné získať väčší počet I/O vstupov a taktiež pracovať s výkonom a náročnejšími operáciami. Pridať viac senzorov alebo prvkov pre aktívne ovládanie a monitorovanie veličín, prípadne použiť menší modul pre integráciu do zariadení kde je kladený dôraz na veľkosť. Projekt je teda plne modulárny a v rámci platformy Arduino je kód použiteľný pre iné modely.

SEZNAM POUŽITÉ LITERATURY

- [1] Arduino UNO R3. Arduino [online]. [cit. 2018-01-18]. Dostupné z: <https://www.arduino.cc>
- [2] BELL, Charles A. Beginning sensor networks with Arduino and Raspberry Pi. New York, New York: Apress, 2013. ISBN 978-1-4302-5825-4.
- [3] Arduino UNO R3. Arduino [online]. [cit. 2018-01-18]. Dostupné z: <https://store.arduino.cc/arduino-uno-rev3>
- [4] ATmega328. Microchip [online]. c2018 [cit. 2018-04-16]. Dostupné z: <http://www.microchip.com/wwwproducts/en/atmega328>
- [5] UART timing diagram. Wikimedia [online]. 2016 [cit. 2018-03-15]. Dostupné z: https://commons.wikimedia.org/wiki/File:UART_timing_diagram.svg
- [6] PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004, 159 s. ISBN 80-7300-110-1
- [7] MARGOLIS, Michael. Arduino cookbook. 2nd ed. Sebastopol, Calif.: O'symbol{39}Reilly, 2012, xx, 699 p. ISBN 1449313876.
- [8] MAZIDI, Muhammad Ali, Sarmad NAIMI and Sepehr NAIMI. The AVR microcontroller and embedded systems: using Assembly and C. Upper Saddle River, N.J.: Prentice Hall, 2011, xiv, 776 p. ISBN 01-380
- [9] AT Commands Reference Guide. Telit [online]. 2016 [cit. 2018-04-08]. Dostupné z: https://www.telit.com/wp-content/uploads/2017/09/Telit_AT_Commands_Reference_Guide_r24_B.pdf
- [10] ESP8266 AT Instruction Set. Espressif [online]. 2018 [cit. 2018-04-08]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf
- [11] ESP8266EX Datasheet. *Espressif Systems* [online]. [cit. 2018-01-24]. Dostupné z: https://dl.dropboxusercontent.com/content_link/CHz5ijT1wBIfxuUE-pYZA62Nq8T9reb49oFYxiVcI9AeQzydBoyumgzPEF3NqzqXe/file
- [12] KABELOVÁ, Alena a Libor DOSTÁLEK. Velký průvodce protokoly TCP/IP a systémem DNS. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5

- [13] Wi-Fi. Wikipedia [online]. [cit. 2018-02-20]. Dostupné z: <https://en.wikipedia.org/wiki/Wi-Fi>
- [14] SOSINSKY, Barrie A. Mistrovství - počítačové sítě: [vše, co potřebujete vědět o správě sítí]. Brno: Computer Press, 2010, 840 s. ISBN 978-80-251-3363-7.
- [15] BMP280 Sensor. Electrothing [online]. 2016 [cit. 2018-04-07]. Dostupné z: <http://electrothing.co.za/products/112-bmp280-sensor#>
- [16] PCB Desing & Schematic Software Eagle: Autodesk. Autodesk.com [online]. Autodesk, ©2018 [cit. 2018-04-07]. Dostupné z: <https://www.autodesk.com/products/eagle/overview>
- [17] Tcpiip vrstvy. Wikimedia [online]. 2008 [cit. 2018-03-15]. Dostupné z: https://commons.wikimedia.org/wiki/File:Tcpiip_vrstvy.svg
- [18] Adafruit BMP280 Library. Github [online]. 2017 [cit. 2018-04-07]. Dostupné z: https://github.com/adafruit/Adafruit_BMP280_Library
- [19] KISZKA, Bogdan. 1001 tipů a triků pro programování v jazyce Java: nejbohatší sbírka řešených problémů a užitečných nápadů v Javě. Brno: Computer Press, 2003, 519 s. ISBN 8072269895.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
GPIO	General-purpose Input/Output – Všeobecný vstup/výstup
FW	Firmware –program slúžiaci na riadenie systému
Hz	Hertz
True	Logický výrok znamenajúci pravda
False	Logický výrok znamenajúci nepravda
UART	Universal Asynchronous Receiver-Transmitter – univerzálny asynchrónny prijímač/vysielač
PWM	Pulse Width Modulation – pulzná šírková modulácia
LED	Light Emiting Diode – svetlo emitujúca dióda
USB	Universal Serial Bus – univerzálna sériová zbernica
Bd	Baud
AP	Acess Point – prístupový bod Wi-Fi
STA	Station Mode – režim stanice
Open-source	Je prístupný pod licenciou, ktorá umožňuje študovanie, používanie, vkladanie a vylepšenie zdrojového kódu
SSH	Secure Shell – zabezpečený komunikačný protokol v počítačovej sieti
SSID	Service Set Identifier –Identifikátor bezdrôtovej siete Wi-Fi
Access Point Mode	Režim prístupového bodu sa používa pre pripojenie bezdrôtových klientov
Client Mode	Režim zariadenia umožňujúci pripojenie na AP a komunikáciu v sieti

SEZNAM OBRÁZKŮ

<i>Obr. 1. Vývojové prostredie Arduino IDE</i>	11
<i>Obr. 2. Vývojová doska Arduino UNO [1]</i>	12
<i>Obr. 3. Schéma výstupných pinov vývojovej dosky Arduino UNO[3]</i>	14
<i>Obr. 4. Schéma pripojenia</i>	15
<i>Obr. 5. Dátový blok prenášaný po sériovej zbernici [5]</i>	16
<i>Obr. 6. Diagram TCP/IP vrstiev [17]</i>	20
<i>Obr. 7. Verzie modulu ESP8266</i>	22
<i>Obr. 8. Schéma výstupných</i>	23
<i>Obr. 9. Princíp komunikácie ESP8266</i>	27
<i>Obr. 10. Princíp komunikácie ESP8266</i>	28
<i>Obr. 11. Bloková schéma zapojenia</i>	31
<i>Obr. 12. Displej LCD 20x4 znakov s I2C zbernicou</i>	32
<i>Obr. 13. Senzor</i>	32
<i>Obr. 14. Univerzálna vývojová doska</i>	33
<i>Obr. 15. Monitor sériovej komunikácie, test spojenia s ESP8266</i>	34
<i>Obr. 16. Schéma zapojenia jednotlivých komponentov</i>	35
<i>Obr. 17. Grafická vizualizácia zapojenia</i>	36
<i>Obr. 18. Vývojový diagram pre program Arduino</i>	38
<i>Obr. 19. Základne definovanie globálnych</i>	39
<i>Obr. 20. ukážka funkcie pre konfigurovanie ESP8266 modulu v režime server</i>	40
<i>Obr. 21. Ukážka kódu funkcie sendData() a sendCommand()</i>	41
<i>Obr. 22. Ukážka kódu funkcie getPressure()</i>	42
<i>Obr. 23. Výstup na sériovú linku</i>	43
<i>Obr. 24. Rozhranie obslužnej aplikácie</i>	45
<i>Obr. 25. Vývojová doska pre platformu Arduino UNO</i>	46

SEZNAM TABULEK

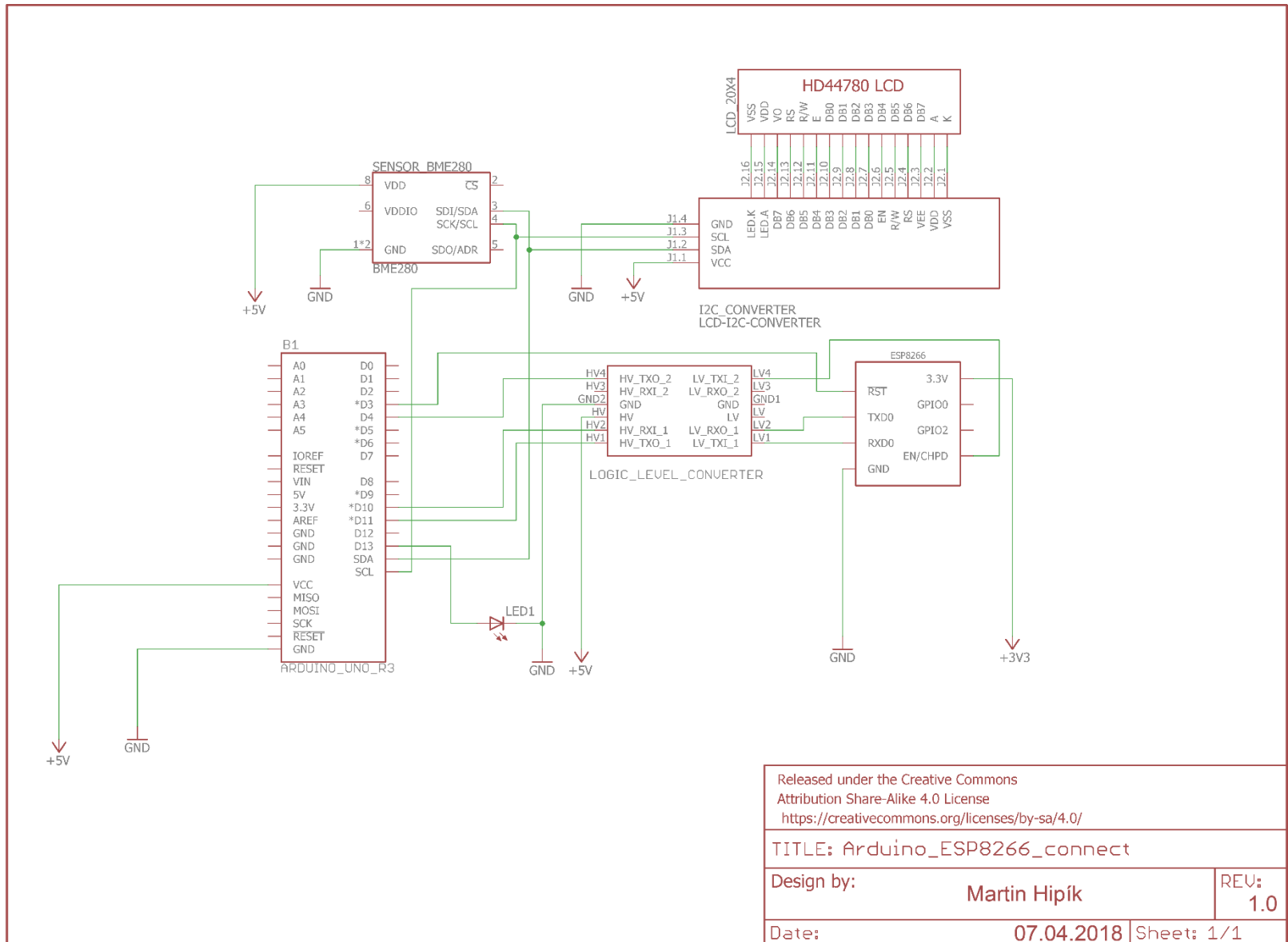
<i>Tab. 1. Technické informácie Arduino UNO [3]</i>	13
<i>Tab. 2. Technické informácie pre ESP8266-01 [11]</i>	23
<i>Tab. 3. ESP8266 Zoznam základných AT príkazov [10]</i>	25

SEZNAM PŘÍLOH

PŘÍLOHA P I: SCHÉMA ZAPOJENIA

PŘÍLOHA P II: CD OBSAHUJÚCE ZDROJOVÉ KÓDY

PŘÍLOHA P I: SCHÉMA ZAPOJENIA



Released under the Creative Commons Attribution Share-Alike 4.0 License https://creativecommons.org/licenses/by-sa/4.0/	
TITLE: Arduino_ESP8266_connect	
Design by:	Martin Hipik
	REV: 1.0
Date:	07.04.2018 Sheet: 1/1