

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra Měření

Inteligentní měřicí systém se vzdáleným přístupem

Bc. Jan Krys

Kybernetika a robotika – Senzory a přístrojová technika

Květen 2018

Vedoucí práce: Doc. Ing. Jaroslav Roztočil, CSc.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Krys** Jméno: **Jan** Osobní číslo: **383271**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**
Studijní obor: **Senzory a přístrojová technika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Intelligentní měřicí systém se vzdáleným přístupem

Název diplomové práce anglicky:

Intelligent Measurement System with Remote Access

Pokyny pro vypracování:

Navrhnete a realizujete měřicí systém s možností vzdáleného přístupu na bázi jednodeskového počítače BeagleBone. Proveďte rozbor možností využití této HW platformy pro dlouhodobé měření, sběr, zpracování a archivaci dat ze senzorů nebo jednoduchých měřicích modulů s rozhraním RS-232, USB a Ethernet. Realizovaný měřicí systém by měl být dostupný nejen prostřednictvím Internetu, ale také s využitím GSM služeb. Systém bude mít kromě síťového napájení zálohované napájení z akumulátorů umožňující provoz po dobu nejméně 5 hodin. Systém by měl být schopen komunikace s dalšími počítači v rámci lokální počítačové sítě a bude umožňovat jejich monitoring případně restartování pro obnovení jejich činnosti.

Seznam doporučené literatury:

[1] Dokumentace k platformě BeagleBone, viz <http://beagleboard.org/>
[2] Hrusha, V. - Kochan, R. - Kurylyak, Y.: Osolinsky, O.: Development of Measurement System with Remote Access Based on Internet. IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. September 2007, Dortmund, Germany

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Jaroslav Roztočil, CSc., katedra měření FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **25.09.2017** Termín odevzdání diplomové práce: **25.05.2018**

Platnost zadání diplomové práce: **28.02.2019**

doc. Ing. Jaroslav Roztočil, CSc.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Poděkování / Prohlášení

Chtěl bych poděkovat své rodině, přítelkyni a vedoucímu práce, za čas a trpělivost, kterou mi poskytli.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 25. 5. 2018

.....

Abstrakt / Abstract

Tato práce se zabývá návrhem a realizací měřicího systému, který je založen na hardwarové platformě BeagleBone Black a operačním systému Linux. K systému je možné připojit senzory přes univerzální binární vstupy, analogové vstupy, USB, I2C, UART a další. Je zálohováno pomocí olověného akumulátoru. Zařízení je také schopné komunikovat přes Ethernet s dalšími zařízeními jak na lokální síti, tak přes internet. Může tak kontrolovat jejich funkčnost a případně nahlásit chybu zasláním SMS zprávy, nebo e-mailu. Dále je také možné tato zařízení restartovat, pro obnovení správné činnosti.

Klíčová slova: sběr dat, monitoring, vzdálený přístup, vysoká dostupnost, BeagleBone Black, měřicí přístroj

This thesis discusses the concept and realisation of measurement system based on BeagleBone Black hardware platform and Linux operating system. The system supports universal binary inputs, analog inputs, USB, I2C, UART and other interfaces. It is also backed up by lead-acid battery. Device can also communicate through Ethernet with other ones on local network and internet. It can check their proper functionality, report errors by SMS and e-mail and restart them to restore functionality.

Keywords: data acquisition, monitoring, remote access, high availability, BeagleBone Black, measuring device

Title translation: Intelligent measurement system with remote access

Obsah /

Zkratky	1	6.2 Nastavení a ovládání.....	30
1 Úvod	3	6.2.1 Konfigurační soubor	30
2 Platforma BeagleBone Black a jeho konkurenti	4	6.2.2 Čtení statusu.....	32
2.1 Základní rozložení konektorů na desce	4	6.2.3 Restart serveru.....	32
2.2 Napájení.....	5	7 Závěr	34
2.3 Rozšiřující headery.....	6	Literatura	36
2.3.1 GPIO.....	6		
2.3.2 PWM.....	7		
2.3.3 Analogové vstupy	7		
2.3.4 UART	8		
2.3.5 I2C	8		
2.3.6 SPI a PRU	9		
2.4 Raspberry Pi 3.....	9		
2.4.1 Rozšiřující headery	10		
2.5 Shrnutí	10		
3 Sběr dat	12		
3.1 Připojení senzorů a měřících modulů	12		
3.1.1 Čtení analogových sig- nálů.....	13		
3.1.2 Komunikace přes RS- 232.....	13		
3.1.3 Komunikace přes USB a Ethernet.....	14		
3.2 Zpracování a archivace dat	14		
4 Příklad realizace systému pro sběr dat	16		
4.1 Měření teploty	16		
4.1.1 Senzor teploty.....	16		
4.1.2 Bezdrátové stanice	17		
4.2 Vzdálený přístup	20		
4.2.1 Vzdálená plocha.....	21		
4.2.2 SSH/SCP.....	22		
4.2.3 Prohlížeč	22		
4.3 Neveřejná IP adresa	24		
5 Vysoká dostupnost	25		
5.1 Zdroj napětí.....	25		
5.2 Komunikace	26		
6 Monitoring	28		
6.1 Základní struktura.....	28		
6.1.1 SSHFS.....	28		
6.1.2 rsync.....	28		
6.1.3 Připojení přes SSH	29		
6.1.4 Kontrola souboru	30		



Zkratky

- ADC ■ Analog-Digital Converter (analogově-digitální převodník)
- AIN ■ Analog input (analogový vstup)
- ARM ■ Advanced RISC Machine - architektura procesorů využívaných v mobilních zařízeních.
- BBB ■ BeagleBone Black
- CPU ■ Central Processing Unit (Centrální Procesorová Jednotka)
- CSI ■ Camera Serial Interface
- DAQ ■ Data acquisition
- DHCP ■ Dynamic Host Configuration Protocol
- DSI ■ Display Serial Interface
- GPIO ■ General-purpose input/output - (obecný vstupní/výstupní pin)
- GSM ■ Global System for Mobile Communications
- GUI ■ Graphical User Interface - Grafické uživatelské rozhraní
- HDMI ■ High-Definition Multi-media Interface - rozhraní pro přenos nekomprimovaného obrazového a zvukového signálu v digitálním formátu.
- HTTP ■ Hypertext Transfer Protocol
- IC ■ Integrated Circuit (integrovaný obvod).
- ID ■ Identifier
- IP ■ Internet Protocol
- I2C ■ Inter-Integrated Circuit – počítačová sběrnice.
- LAN ■ Local Area Network (lokální síť).
- LCD ■ Liquid Crystal Display
- LED ■ Light-Emitting Diode (světlo vyzařující dioda).
- MCU ■ microcontroller unit - mikropočítač skládající se pouze z jednoho integrovaného obvodu
- PMIC ■ Power Management IC (integrovaný obvod řídicí spotřebu).
- PWM ■ Pulse Width Modulation (Pulzně šířková modulace)
- SAR ■ Successive-approximation-register
- SCP ■ Secure Copy Protocol - protokol pro přenos souborů pomocí SSH
- SMS ■ Short Message Service
- SMTP ■ Simple Mail Transfer Protocol
- SQL ■ Structured Query Language

- SSH ■ Secure Shell - zabezpečení komunikační protokol, který byl navržen jako náhrada za telnet
- SSHFS ■ SSH File System
- UART ■ universal asynchronous receiver-transmitter (univerzální asynchronní vysílač/přijímač)
- UPS ■ Uninterruptible Power Supply - nepřerušitelný zdroj napájení
- USB ■ Universal Serial Bus (Univerzální Sériová Sběrnice) - komunikační protokol pro připojení periférií k počítači.
- VNC ■ Virtual Network Computing

Kapitola 1

Úvod

Sběr a zpracování dat se dnes používá v mnoha odvětvích od průmyslu, přes meteorologii, sport až po použití v domácnostech. Měřicí systémy se obvykle skládají z více senzorů napojených na jednu nebo více řídicích jednotek, které data vyčítají a případně zpracovávají. Řídící jednotky mohou být od jednoduchých MCU, přes složitější mikropočítače, až po velké průmyslové systémy složené z více počítačů.

Díky velkému rozvoji komunikačních technologií, dnes není problém se připojit k internetu z téměř jakéhokoliv místa. To nám poskytuje další možnosti ve sběru dat - není již potřeba pravidelně navštěvovat měřicí stanice pro extrakci dat. Další výhodou použití komunikačních technologií je možnost sledovat stav stanice vzdáleně. Můžeme se tedy dozvědět, že něco není v pořádku hned, jak chyba nastane a ne až když přijdeme vyčítat data.

Většina měřicích systémů má v sobě zabudovaný nějaký druh samodiagnostiky, aby mohli hlásit případné problémy, nebo úspěšné vykonání akcí. Může ovšem dojít k závažnějšímu problému, kdy pak samotný měřicí systém není schopen nic reportovat, a je potřeba externí sledovač.

Tato práce se zabývá možností vytvoření takového zařízení. Je kladen důraz na to, aby systém mohl být provozován dlouhodobě a byl dostupný vzdáleně. Dalším požadavkem je také vysoká dostupnost, což znamená, že je systém k dispozici téměř vždy, když je to potřeba. Měl by také mít možnost kontrolovat ostatní zařízení připojené v lokální síti nebo i přes internet a nahlásit přerušování měření, případně i zařízení restartovat, kvůli obnovení správné činnosti.

V následující kapitole je popsána platforma BeagleBone Black a její konkurenti. Dále 4 pak bude popsán modul pro vyčítání dat z různých senzorů, jejich zpracování a archivaci. V další kapitole 5 bude rozebráno téma vysoké dostupnosti. A nakonec 6 se podíváme na téma monitoringu jiných měřicích systémů jak v rámci lokální počítačové sítě, tak přes internet.

Kapitola 2

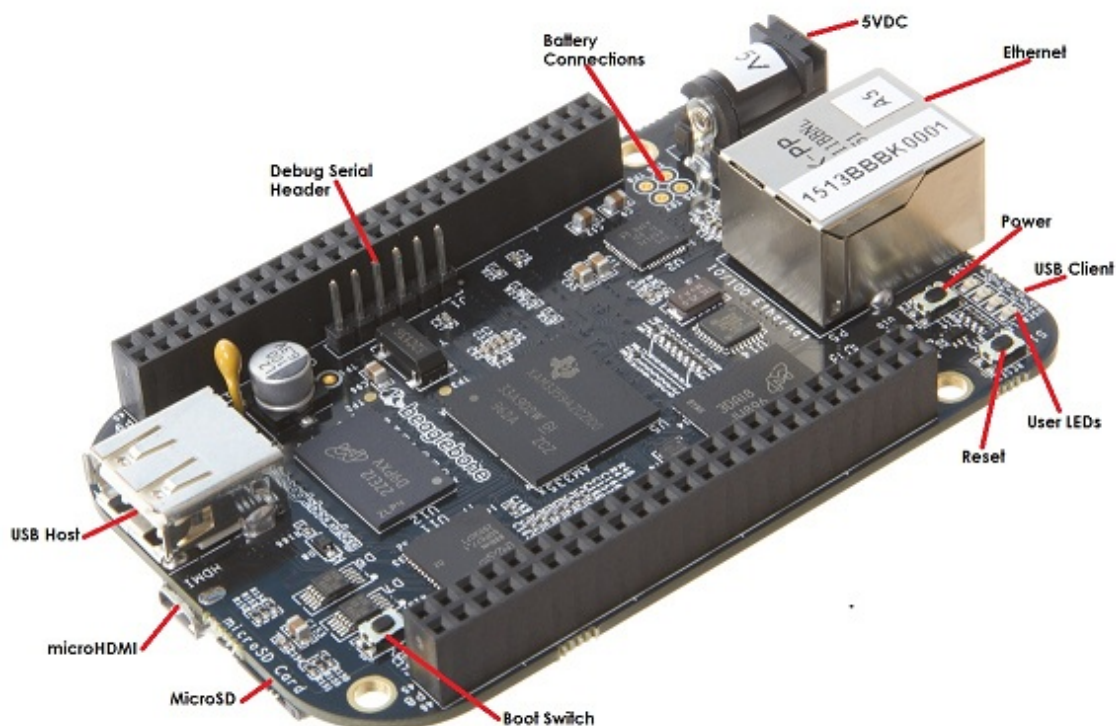
Platforma BeagleBone Black a jeho konkurenti

BeagleBone Black je jednodeskový počítač od BeagleBoard.org Foundation. Srdcem tohoto počítače je Cortex A8 ARM procesor od Texas Instruments. Je zaměřen hlavně na vysokou rozšiřitelnost skrz dva 46 pinové headery, USB host, Ethernet a HDMI rozhraní. Na desce je předinstalován systém Linux, který bootuje do 10 sekund.

Když desku připojíme k počítači přes USB, nahlásí se jako velkokapacitní paměťové zařízení, na kterém najdeme ovladače pro systémy Windows, Mac OS i Linux. Po nainstalování ovladačů se při připojení BBB vytvoří virtuální síťové připojení a můžeme komunikovat přes SSH.

2.1 Základní rozložení konektorů na desce

Na následujícím obrázku 2.1 je vidět rozložení konektorů a tlačítek na desce BBB.



Obrázek 2.1. Rozložení konektorů a tlačítek na BBB (zdroj: [1]).

Jsou zde 3 tlačítka:

- **Power** – funguje jako tlačítko napájení u běžného PC – zmáčknutím se pošle přerušení do procesoru aby uložil soubory, odpojil externí zařízení a uspal se. Dalším zmáčknutím se zase probudí. Pokud tlačítko podržíme 8 vteřin, tak se přeruší napájení, pokud ho budeme držet dál tak se znovu obnoví.
- **Reset** – po stisknutí a uvolnění tohoto tlačítka se restartuje procesor.
- **Boot Switch** – pokud držíme toto tlačítko zmáčknuté při spuštění, donutíme BBB naběhnout z microSD karty místo interní paměti.

Pro propojení s PC je zde **USB Client**, který je ve formě miniUSB a kterým se dá deska i napájet. Dále je zde **USB Host**, kam můžeme připojit jakékoliv zařízení s touto sběrnici, jako například WiFi adaptér, klávesnici, Flash disk, ... Pokud napájíme BBB pouze přes USB Client a chceme připojit více zařízení přes USB Host, doporučuje se použít hub s externím napájením. Pro připojení do LAN a potažmo internetu je zde připraven **10/100 Ethernet**. Když potřebujeme dodatečné úložiště, nebo přeinstalovat systém je zde připraven **MicroSD** slot. BBB také disponuje dostatečným výkonem, aby mohl používat grafické rozhraní, pro jehož zobrazení je **microHDMI** port, do kterého můžeme připojit monitor, a zvládá rozlišení až 1280x1024. **Debug Serial Header** můžeme použít, pokud chceme ladit přímo CPU. Další možnost napájení je přes **5VDC** nebo z baterie přes **Battery Connections**.

Pro jednoduché optické zjištění stavu desky jsou instalovány 4 modré LED. Ty může uživatel ovládat podle svých potřeb, ale v základu jsou nastaveny takto:

- **USER0** – heartbeat – blikání podobné tlukotu srdce, které je rychlejší při větším zatížení CPU.
- **USER1** – svítí pokud se přistupuje na microSD kartu.
- **USER2** – indikátor aktivity – rozsvítí se pokud není jádro v nečinné smyčce.
- **USER3** – svítí pokud se přistupuje do vnitřní paměti.

2.2 Napájení

O napájení se stará jednočipový PMIC *TPS65217C*. Ten může být napájen buď z USB portu, nebo přes DC adaptér. Při napájení z USB portu je proud typicky limitován na 500mA, což se dá změnit přes I2C. Druhá limitace při tomto napájení je, že není poskytováno napětí do VDD_5V, takže pokud chceme něco napájet, musíme použít SYS_5V. Když je připojen jak 5VDC adaptér, tak i USB port, *TPS65217C* automaticky používá 5VDC, toto se dá také nastavit pomocí I2C. Pro naše použití je důležité to, že se přes tuto sběrnici dá vyčíst, který ze zdrojů je připojen.

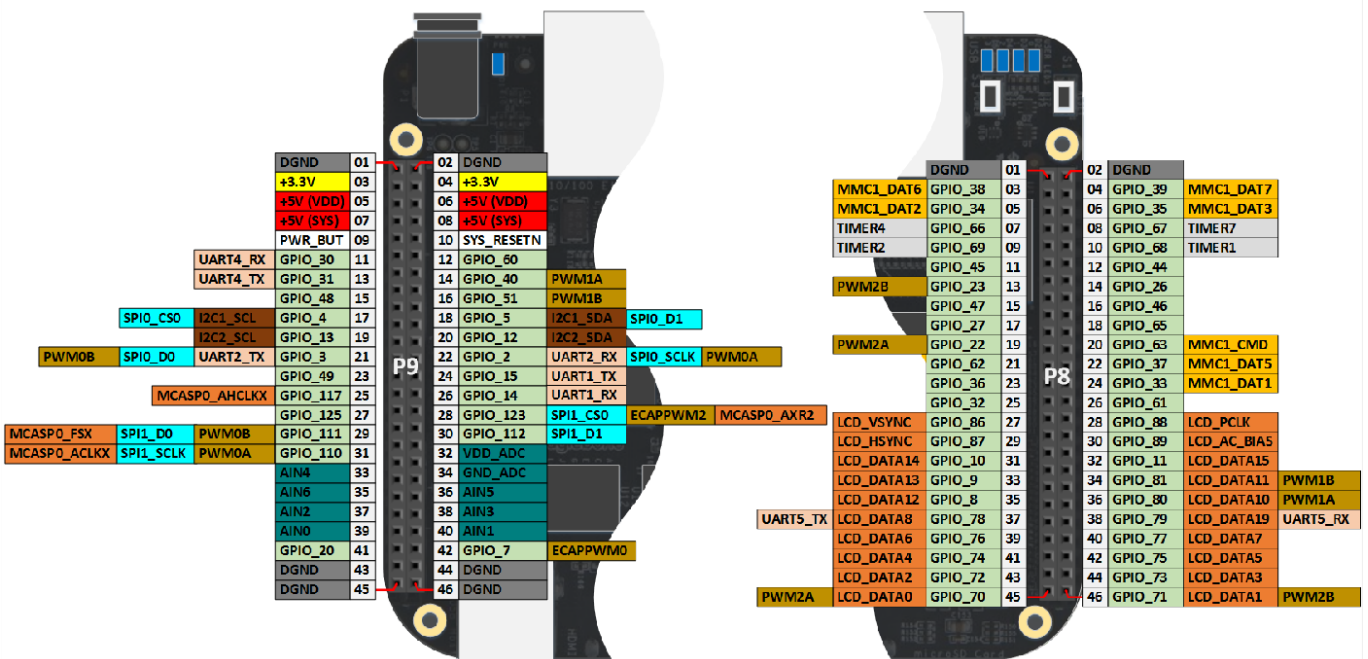
Další možností napájení je z baterie, pro kterou jsou tu připraveny následující 4 piny:

- BAT – kladný pól baterie
- Sense – sledování napětí baterie
- TS – sledování teploty baterie
- GND – záporný pól baterie

Bohužel nejde z *TPS65217C* vyčíst jak moc je baterie nabitá

2.3 Rozšiřující headery

Jednou z velkých předností BBB je počet digitálních a analogových vstupů a výstupů, z nichž většina může vykonávat více různých funkcí. Máme k dispozici 2 headery po 46 pinech, což je celkem 92 pinů. Na následujícím obrázku 2.2 je jejich mapa.



Obrázek 2.2. Mapa pinů na rozšiřujících headerech BBB (zdroj: [2]).

Z mapy je vidět, že většina pinů plní více než jednu roli (celkem můžou být nastaveny až na 8 různých), bohužel však nemohou plnit více rolí najednou, takže při návrhu musíme dávat pozor na to, co všechno budeme chtít využívat a podle toho zvolit správnou kombinaci.

2.3.1 GPIO

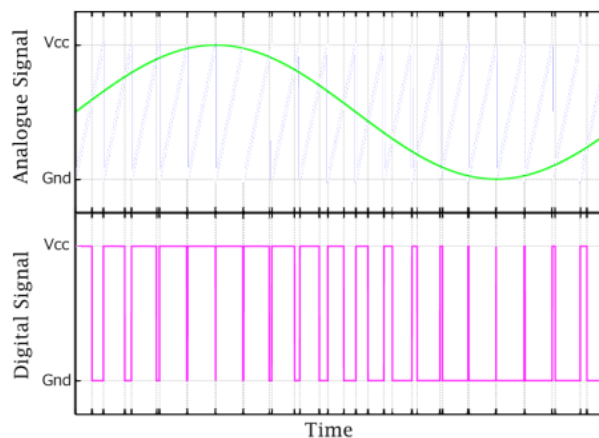
Těchto pinů je celkem 69 (všechny piny kromě analogových vstupů a napájení), ale ne všechny se dají vždy používat. 10 z nich je používáno pro přístup do eMMC paměti, takže je můžeme používat pouze, pokud na ni nechceme přistupovat – musíme mít

system načtený z microSD karty. 20 je použito pro HDMI výstup, takže pokud chceme mít možnost připojit monitor, musíme se vyhnout jejich používání. Všechny tyto použité piny jsou na headeru P8, proto se snažíme využívat header P9, vždy když je to možné.

Piny lze použít jako digitální vstup nebo výstup, všechny mohou vyvolávat přerušení.

■ 2.3.2 PWM

Až 8 z GPIO pinů může být nastaveno jako PWM, čímž můžeme kontrolovat motory nebo vytvářet pseudo analogové signály. Aproximace analogového signálu je ukázána na následujícím obrázku 2.3:



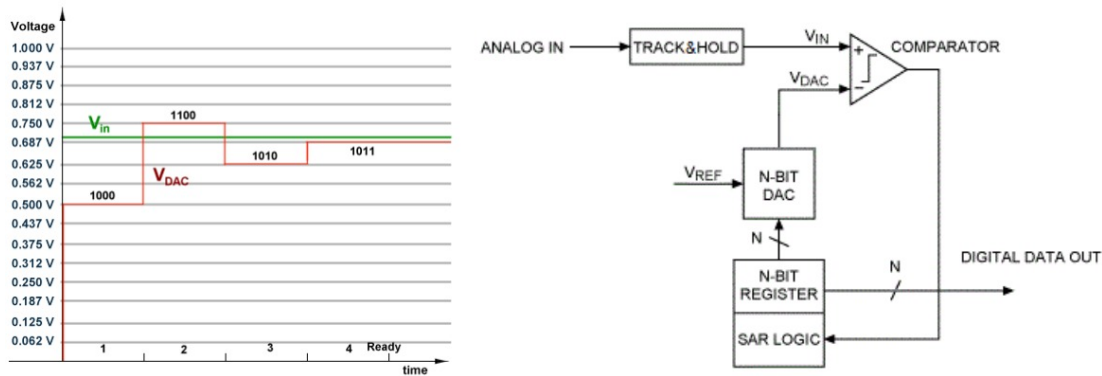
Obrázek 2.3. Ukázka signálu PWM (zdroj: [3]).

Minimální frekvence, kterou lze nastavit je 0Hz a maximální 400MHz. PWM by bylo možné emulovat softwarově, ale je to velmi náročné na procesor, obzvláště u vyšších frekvencí, a musí se zajistit přesné načasování. Do PWM modulů obsažených přímo v CPU stačí poslat frekvenci a pracovní cyklus, ty již vše obstarají.

■ 2.3.3 Analogové vstupy

BBB dále disponuje 7 analogovými vstupy. Napětí na nich je měřeno 12-bitovým analogově digitálním konvertorem s postupnou aproximací, který zvládá až 200k samplů za vteřinu. Je to kompenzační převodník a jeho blokové schéma vidíme na obrázku 2.4 vpravo. Funguje následujícím způsobem: převod je prováděn v N krocích, kde N je počet bitů převodníku. Začíná se od nejvýznamnějšího bitu, kam se nastaví logická 1 a na zbytku logická 0. Vždy se porovnává napětí z digitálně analogového převodníku V_{DAC} se vstupním napětím V_{IN} . Pokud je V_{IN} nižší než V_{DAC} – výstup z komparátoru je záporný, nastaví se na aktuálním bitu logická 0, jinak se nastaví logická 1. Tento postup se opakuje pro všechny další bity.

Příklad průběhu aproximace je na obrázku 2.4 vlevo. Jedná se o 4-bitový převodník s rozsahem 0–1V.



Obrázek 2.4. Blokové schéma a příklad průběhu převodu SAR (zdroj: [4]).

Výstupem převodníku v BBB je číslo od 0 do 4096, které představuje napětí od 0 do 1,8V. 1,8V je taky maximální napětí, které můžeme na tento vstup přivést – pokud bychom přivedli větší, můžeme zničit procesor, na který je pin přímo napojen. Pokud chceme měřit větší napětí, musíme použít dělič napětí.

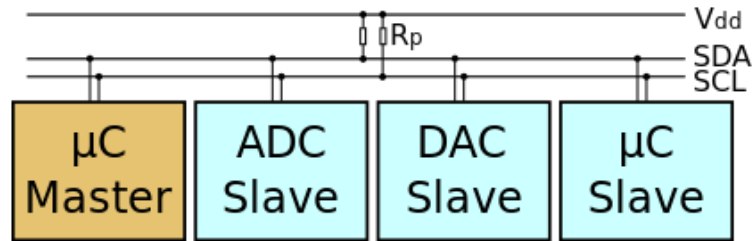
2.3.4 UART

UART je komunikační rozhraní, kterým je vybaven téměř každý IC, proto je výhodné ho mít připravené pro komunikaci s jednoduššími měřicími přístroji. Jelikož to je rozhraní, přes které se mohou propojit pouze 2 zařízení, je dobré, že BBB má integrované hned 4 tato rozhraní. UART je plně duplexní, takže můžeme zároveň vysílat a přijímat data.

2.3.5 I2C

I2C je sériová sběrnice typu master–slave, což znamená, že je zde řídicí zařízení (master), který může zahajovat komunikaci, a několik řízených zařízení (slave). Používá se hlavně na připojení periférií, u kterých nepotřebujeme vysokou rychlost přenosu dat, hlavně pro vyčítání nebo zapisování jednoduchých dat o velikosti jednotek bytů. Výhodou je, že na tuto sběrnici můžeme připojit až 128 zařízení. Nevýhodou je citlivost na kapacitní zátěž, což znamená, že nemůžeme používat moc dlouhé vodiče. Maximální kapacita vodičů je 400pF, ale pokud chceme rychlost 3,4Mbit/s, je maximální přípustná kapacita pouze 100pF[5]

Komunikace probíhá přes 2 vodiče: jeden pro hodinový synchronizační signál a druhý pro datový signál. Každý z těchto vodičů musí být v klidovém stavu na vysoké úrovni, čehož je docíleno pomocí pull–up rezistorů. Příklad zapojení můžeme vidět na obrázku 2.5



Obrázek 2.5. jednoduché schéma zapojení I2C (zdroj: [6]).

BBB disponuje dvěma těmito sběrnicemi a vystupuje na nich jako řídicí uzel.

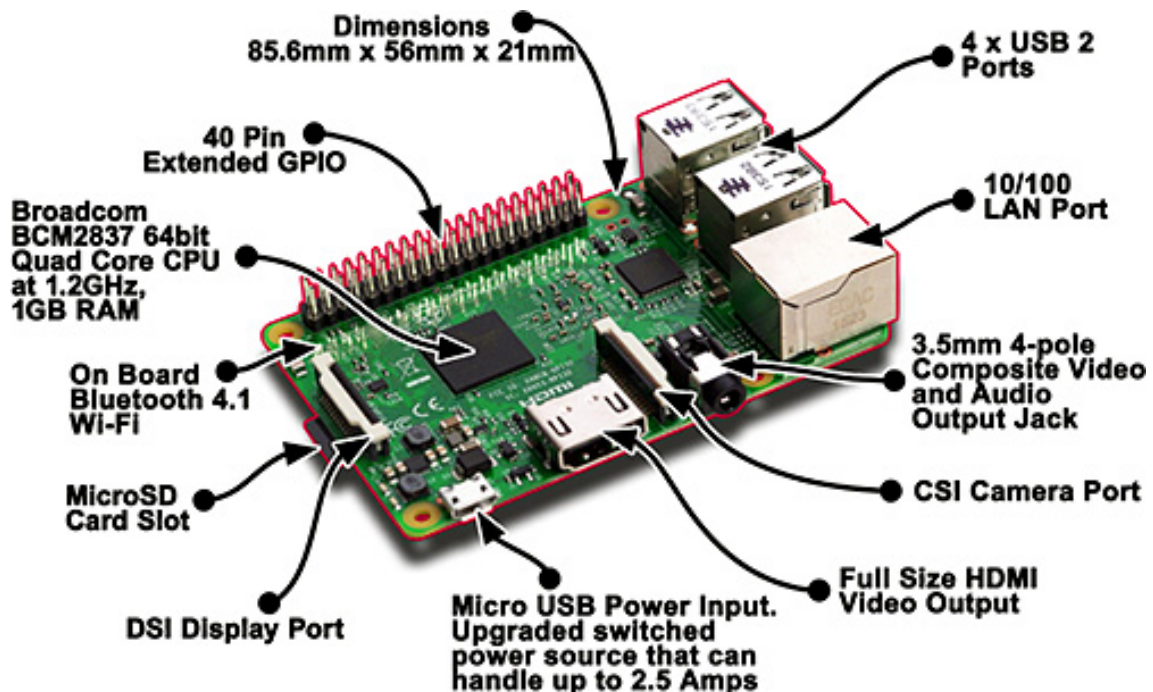
■ 2.3.6 SPI a PRU

Dále jsou zde 2 SPI sběrnice, která může mít každá 2 podřízená zařízení a používá se pro rychlý přenos větších dat.

Nakonec tu je 25 PRU pinů díky kterým můžeme programovat systémy reálného času, nebo se dají použít pro softwarovou implementaci UART nebo dokonce PWM.

■ 2.4 Raspberry Pi 3

Dalším kandidátem pro tento projekt byl jednodeskový minipočítač Raspberry Pi 3 od Raspberry Pi Foundation. Poskytuje větší výpočetní výkon než BBB, dokonce má i samostatný grafický procesor pro jednodušší zobrazování.



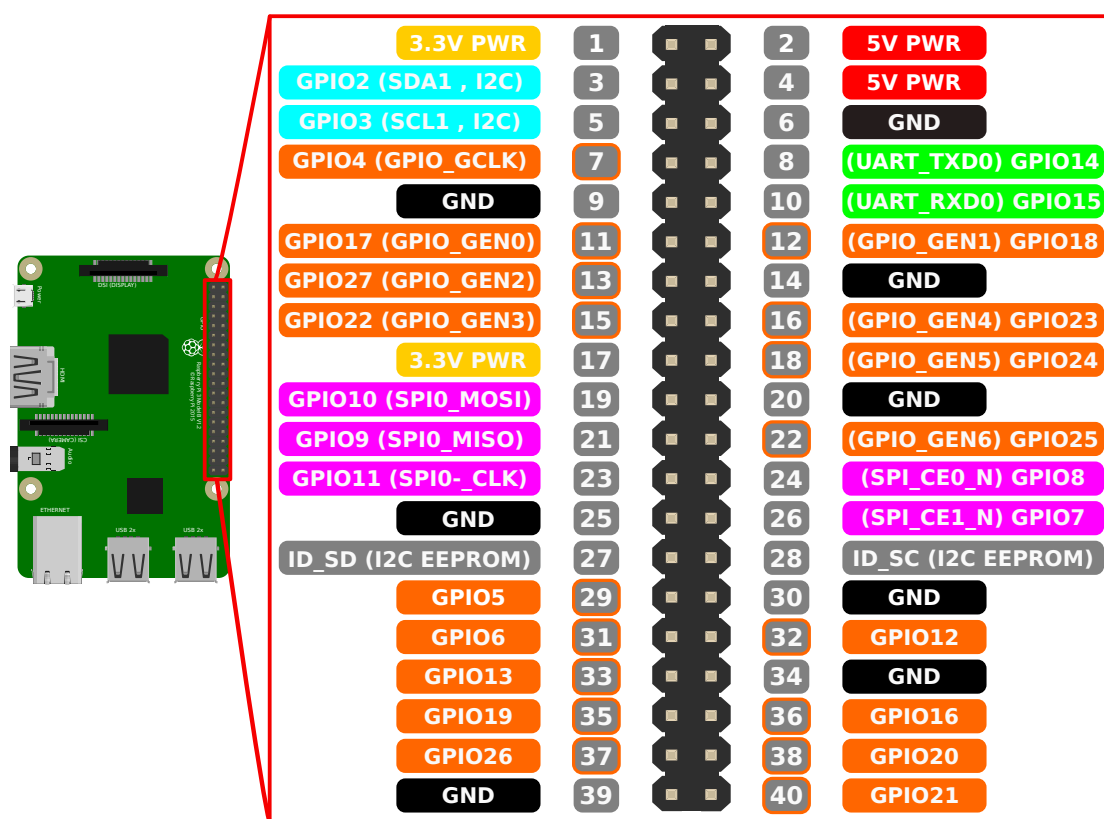
Obrázek 2.6. Rozložení konektorů na Raspberry Pi 3 (zdroj: [7]).

Na obrázku 2.6 vidíme základní rozložení konektorů na desce. Stejně jako u BBB má 10/100 Ethernet a MicroSD slot, ale má už rovnou 4 USB Host porty, takže

nepotřebujeme hub, pokud budeme potřebovat připojit více než jedno zařízení. **Micro USB** zde funguje pouze jako napájení, takže pokud chceme Pi připojit k počítači, musíme to udělat přes Ethernet. Narozdíl od BBB je zde připraven plnohodnotný HDMI port, takže nepotřebujeme žádný převodník micro HDMI, když chceme připojit monitor. Dále je tu „navíc“ **DSI port** pro přímé připojení LCD a **CSI port** pro přímé připojení digitální kamery. Pokud nechceme pro zvukový výstup používat HDMI, je zde i 3,5mm Audio jack. Pro bezdrátovou konektivitu je tu i WiFi a bluetooth modul.

2.4.1 Rozšiřující headery

Stejně jako BBB má Raspberry Pi 3 rozšiřující GPIO piny, jejich mapu můžeme vidět na obrázku 2.7.



Obrázek 2.7. Mapa pinů na Raspberry Pi 3 (zdroj: [8]).

Je zde pouze 40 pinů z čehož 24 je programovatelných GPIO. Je tu hardwarově připraven 1 sériový UART, 1 sběrnice I2C a 2 sběrnice typu SPI, kde k jedné můžeme připojit 2 zařízení a ke druhému 1.

2.5 Shrnutí

Oproti BBB je Raspberry Pi lépe vybaveno pro použití s displayem – má samostatný grafický procesor, plnohodnotné HDMI a celkově větší výpočetní výkon. Díky hned 4 USB portům, je také jednodušší připojit periferie stejně jako ke klasickému PC.

Pro účely této práce nejsou však tyto věci tak podstatné – vyvíjím systém, který bude provozován vzdáleně, takže nepotřebuji mít připojenou klávesnici ani myš a display je také zbytečný. Na druhou stranu je zde méně GPIO a žádné AIN, takže pokud chceme vyčítat analogové signály, musíme použít nějaké externí ADC.

Kapitola 3

Sběr dat

Sběr dat (DAQ) je proces, při kterém se měří podmínky v reálném světě, například teplota, napětí nebo tlak, a převádějí se na číselné hodnoty, které pak mohou být zpracovány v počítači. Profesionální systém pro sběr dat se obvykle skládá ze senzoru, počítače a nějakého hardwaru, který čte data ze senzoru a přes standardní sběrnici je předává počítači. Ta poslední část může být jak samostatná, tak může být už integrovaná přímo v senzoru, takové senzory se pak nazývají inteligentní, ale může být integrovaná i v počítači.

Takto nasbíraná data, se většinou dále nějak zpracovávají nějakými programy. Můžeme je zpracovávat buď hned v měřicím systému, nebo můžeme mít těchto systémů spojených více dohromady a zpracovávat až výsledný soubor dat. Nejčastěji se data předzpracují ve sběrných systémech, tak aby se dala jednoduše analyzovat, a následně se zpracovávají společně s daty z jiných systémů a případně i s historickými. Na to je pak už potřeba velká výpočetní kapacita, a proto se to děje ve velkých počítačových clusterech. Výstupy těchto analýz se dají používat v mnoha odvětvích.

Podle toho co se zrovna snažíme analyzovat, používáme i různé typy zařízení. Můžeme potřebovat velkou přesnost, nebo rychlost měření a na to jsou také potřeba specializované senzory a převodníky. Toto je potřeba například u experimentů v CERNu, kde je po srážce dvou částic potřeba zpracovat velké množství dat přijatých ve velmi krátkém čase. Takovéto systémy jsou velmi drahé. Pokud ale měříme klimatické změny, tak stačí odebírat hodnoty pouze několikrát za den. Hodnoty, například teploty, také nemusíme měřit s takovou přesností a proto stačí jednodušší a hlavně levnější systémy.

V této práci jsem se zaměřil především na řešení systému, který by měl být schopný fungovat dlouhodobě a měl by být odolný vůči krátkodobému výpadku energie a spojení. Měl by být schopný komunikovat s velkým množstvím různých senzorů a jednoduchých měřicích modulů, připojených přes různá rozhraní. Musí být schopen data zpracovat a uložit pro další rozbor.

3.1 Připojení senzorů a měřicích modulů

V minulé kapitole bylo popsáno několik možností, kterými se dají připojit různé senzory. Pokud nám toto nestačí je možno také připojit k BBB takzvané *cape*.

To jsou rozšiřující desky, které mají obdobnou velikost jako BBB a mají stejné konektorové lišty. Díky tomu jich můžeme skládat více na sebe.

■ 3.1.1 Čtení analogových signálů

Jak již bylo popsáno v minulé kapitole, je možné analogové signály číst přímo přes analogové vstupy BeagleBone. Pokud však chceme větší přesnost nebo rychlost čtení, musíme použít externí ADC.

Jednou z možností je použití PRUDAQ cape.¹ Tato deska využívá programovatelné jednotky reálného času k zajištění co nejrychlejšího měření. Pro připojení sond jsou tu 2 SMA konektory, další tento konektor je také pro připojení externích hodin. Obsahuje dvoukanálový analogově digitální převodník s 10-bitovým rozlišením a možností souběžného odebírání vzorků. Na každém kanálu můžeme dosáhnout až 19.9MS/s, což dává dohromady téměř 40MS/s. Je ale nutné použít ještě BeagleLogic cape² a externí hodiny. Při takto vysoké vzorkovací frekvenci musíme umět zpracovat velké množství dat (400Mbps), což BBB nezvládne uložit na žádná úložiště. Takto rychle se dají data uložit pouze do RAM, která má 512MB, takže takto můžeme měřit přibližně jednu sekundu (část paměti zabírá operační systém).

Reálně je tedy možné dlouhodobě měřit 5MS/s na dvou kanálech souběžně. A to s použitím vnitřního oscilátoru (10MHz) a bez použití dalších zařízení. Další omezení je maximální vstupní napětí 2V a žádná ochrana, takže přivedením většího napětí můžeme poškodit převodník.

I přes tato omezení, stojí tato deska za zmínku hlavně kvůli tomu, že si schéma můžeme zdarma stáhnout a desku sestavit. Systémy, které poskytují takto vysoké vzorkovací frekvence stojí i desetitisíce korun.

Pokud je potřeba vyšší přesnost je možné využít například převodník ADS1271 jako to bylo použito v [9]. Při použití programovatelných jednotek reálného času je možné spolehlivě zaznamenávat dva 24-bitové kanály vzorkovací frekvencí 130kS/s. Ve článku také prokázali, že rychlost, přesnost i šumový práh byli srovnatelné s dražšími komerčními systémy.

■ 3.1.2 Komunikace přes RS-232

RS-232 je standart pro sériový přenos dat mezi dvěma zařízeními. Přenos dat je asynchronní a duplexní. Při komunikaci se používá +3V až +15V pro signalizaci logické nuly a -15V až -3V pro logickou jedničku. Zařízení musí být schopná přijímat tyto hladiny a také musí být schopná přežít zkrat $\pm 25V$ na zem.

Z těchto důvodů nemůžeme připojit takovéto zařízení přímo k BBB. Ale pro připojení je několik možností. Můžeme použít převodník RS-232 na USB. Ten po připojení a zavedení ovladačů emuluje sériovou linku – dá se pak používat stejně jako UART, akorát je na jiné adrese. Druhou možností je použít cape.

¹ <https://github.com/google/prudaq/wiki>

² <https://github.com/abhishek-kakkar/BeagleLogic>

Existuje buď *micro cape*¹, která zabírá pouze prvních 10 pinů na P9 headeru, většina z nich je pak propojena na výstupní header, ale zakrývá další piny a ztěžuje tak připojení dalších zařízení. Je možno si vybrat jestli bude signál vyveden na UART0 nebo UART4, ale je k tomu potřeba pájení.

Další možností je *4xRS232 cape*². Jak již název napovídá jedná se rozšíření, které poskytuje možnost připojit 4 zařízení přes RS-232, se kterými pak je možno komunikovat přes UART1,2,4 a 5. Jsou zde také hodiny a baterie pro uchování času i po odpojení napájení. Pokud ji chceme používat, musíme vypnout integrovanou grafickou kartu, potažmo HDMI rozhraní. Není tu také možnost připojit již nic jiného.

Poslední *cape* je *RS-232 Cape*³, která poskytuje jeden konektor pro RS233. Pomocí propojek si můžeme vybrat, na který z UART0,1,2 a 4 bude signál přiveden. Při použití této desky jsou pro další použití k dispozici všechny piny, kromě právě používaného UART.

■ 3.1.3 Komunikace přes USB a Ethernet

Díky tomu, že je na BeagleBone Black nainstalován operační systém Linux, můžeme použít všechny měřicí moduly, ke kterým existují ovladače. Jediný rozdíl oproti jiným PC je ten, že BBB využívá architekturu procesoru ARM, kdežto většina PC používá architekturu X86, či X64. Ne všechny ovladače, které jsou dostupné pro PC jsou i pro BBB. Ale vzhledem k velkému rozmachu ARM architektury, hlavně pro zařízení s nízkou spotřebou, většina výrobců dodává ovladače i pro ni.

BBB má pouze jeden port USB, pokud je potřeba připojit více zařízení, musíme použít hub. Je vhodné mít pro něj externí napájení, protože BBB poskytuje pouze 500mA, což může být pro více zařízení málo. Hub také snižuje maximální přenosovou rychlost pro každé připojené zařízení, ale pokud jich použijeme nějaký smysluplný počet, neměl by to být problém.

Stejně tak je na desce pouze jeden ethernet port. Pokud chceme připojit pouze jedno zařízení, můžeme ho připojit napřímo, ale musíme zajistit správné nastavení IP adres. Jinak můžeme použít nějaký síťový prvek (hub, switch, ...). Většina těchto zařízení má DHCP server, který zařizuje správné nastavení sítě.

■ 3.2 Zpracování a archivace dat

Jak jsem již psal na začátku kapitoly, pro zpracování dat se hodí spíše výkonnější počítače, případně i víc jich zapojených do clusteru. Pokud však chceme pouze základní analýzy nebo filtrování, můžeme použít například knihovnu ScyPie⁴. V ní jsou imple-

¹ <https://www.logicsupply.com/cbb-ttl-232/>

² <https://www.hobrasoft.cz/en/hardware/4x232/>

³ https://elinux.org/RS232_Cape_RevB

⁴ <https://www.scipy.org/index.html>

mentovaný různé nástroje jako je fourierova transformace, spektrální analýza, korelace a také funkce na vykreslení různých grafů. Pokud však chceme zpracovávat data z více různých senzorů najednou a pak je analyzovat, výkon BBB na to stačit nebude.

Vzhledem k nízké kapacitě vnitřní paměti (4GB), ze které velkou část zabírá samotný operační systém, je potřeba k BBB připojit nějaké externí úložiště, pokud chceme uchovávat větší množství dat. Lze vložit microSD kartu, až do velikosti 32GB, což už může být dostatečně velký prostor. Pokud však chceme uchovávat data z více senzorů, po dlouhé doby, doporučil bych je odesílat v pravidelných intervalech na nějaký server, kde pak mohou být i analyzována ve větší míře.

Kapitola 4

Příklad realizace systému pro sběr dat

Jak bylo napsáno v minulé kapitole, BBB obsahuje mnoho vstupů a výstupů a díky tomu lze připojit více různých senzorů najednou. Na rozdíl od vývojových desek s mikrokontrolérem, kde může běžet pouze jeden program ve smyčce, funguje BBB jako klasický počítač. Můžeme zde pustit více různých programů, které běží nezávisle na sobě. Každý z těchto programů může obsluhovat jeden nebo více senzorů.

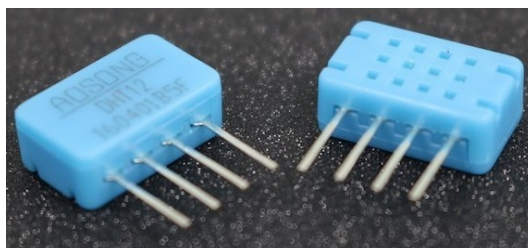
Pro tuto práci jsem zvolil následující architekturu: každý senzor je obsluhován vlastním programem nebo skriptem a pravidelně vyčítá data ze senzoru. Data se poté odešlou přes HTTP POST na lokální server, který data uloží do databáze. Server může být nastaven tak, že v určitých intervalech posílá data na vzdálený server. Dále u každého senzoru mohou být nastaveny limity a když je naměřená hodnota mimo ně, tak se provede předdefinovaná akce.

4.1 Měření teploty

Jako jednu z ukázkových aplikací jsem zvolil měření teploty na více místech v domě. Tato ukázka zahrnuje jak přímé vyčítání dat ze senzoru, tak kooperaci více stanic.

4.1.1 Senzor teploty

Pro měření jsem zvolil senzor DHT12 od firmy AOSONG. Dokáže měřit teplotu v rozmezí -20°C – 60°C s odchylkou $\pm 0,5^{\circ}\text{C}$ při teplotě 25°C , při krajních teplotách se pak odchylka může dostat až na $\pm 2^{\circ}\text{C}$. Rozlišení senzoru je $0,1^{\circ}\text{C}$ a opakovatelnost $0,2^{\circ}\text{C}$. Senzor také měří relativní vlhkost a to v rozmezí 20-95% s rozlišením 0,1%, ale odchylkou $\pm 5\%$ [10].

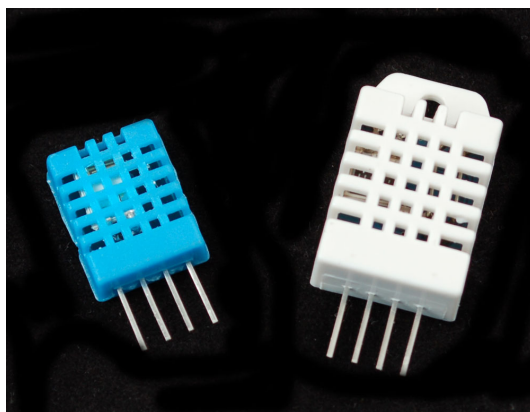


Obrázek 4.1. DHT12.

Tento senzor nabízí 2 různé možnosti připojení a to buď přes I2C, nebo přes 1-wire. Senzor bohužel není nijak konfigurovatelný, takže při použití I2C sběrnice můžeme na

každou připojit pouze jeden – všechny mají stejnou adresu. Sběrnice jsou pouze 2, takže takto můžeme připojit jen 2 teploměry. Pokud ovšem použijeme sběrnici 1-wire můžeme připojit senzorů více, protože se k připojení může využít jakýkoliv GPIO pin a těch je na desce 69. Protože jsem připojoval vždy jen 1 senzor, použil jsem v této práci připojení přes I2C.

Vybral jsem ho kvůli nízké ceně (pod 1USD za kus, pokud nakupujeme z čínských velkoobchodů¹). Dále díky velkému měřicímu rozmezí, které by v našich podmínkách mělo stačit na měření po celý rok i venku. Nepřesnost měření je pro domácí použití přijatelná a senzor je kalibrován tak, aby měl v domácím prostředí (kolem 25°C) nejmenší chybu. Další výhodou je jeho nízká spotřeba, která je, při napájení 5V, průměrně 150μA [10]. A v neposlední řadě také jednoduché připojení a vyčítání dat.



Obrázek 4.2. DHT11 vlevo a DHT22 vpravo.

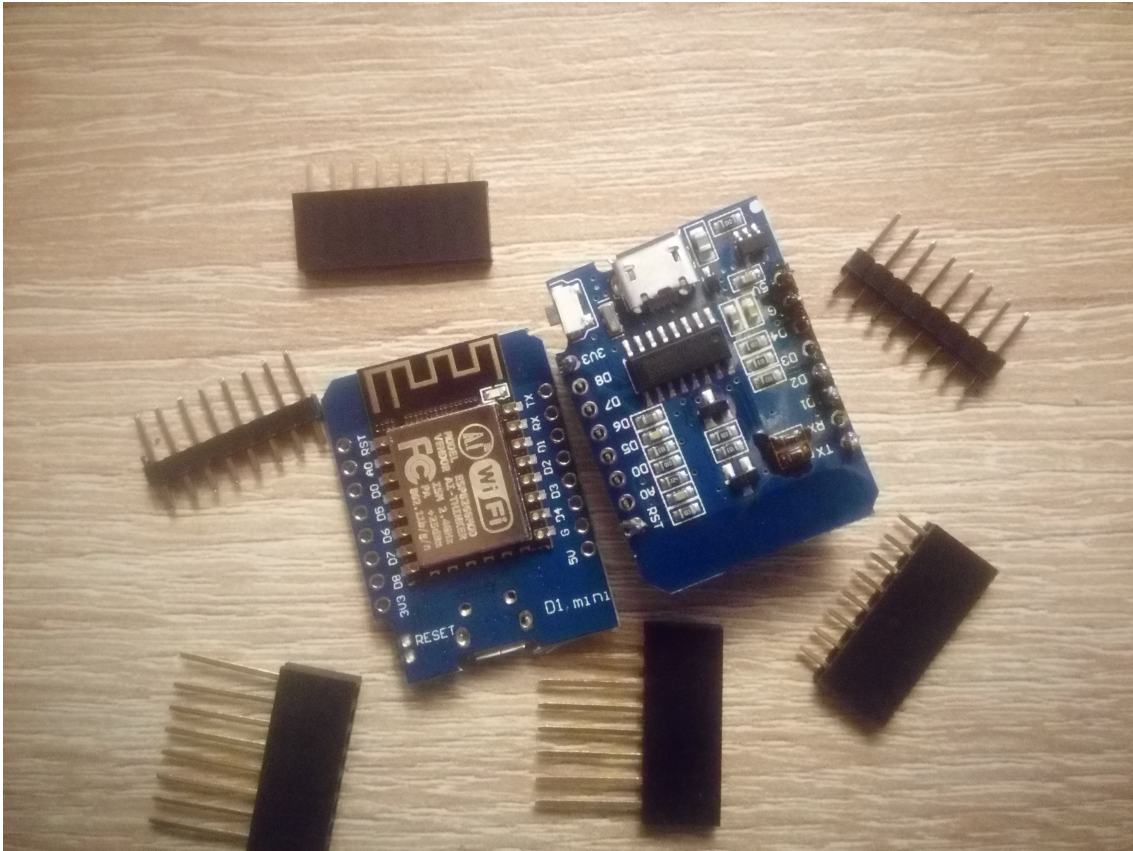
Dalšími možnostmi, které jsem zvažoval byli DHT11 a DHT22. DHT11 jsem vyloučil, protože má pracovní teploty pouze 0°C – 50°C, což by mohl být problém pro měření venkovní teploty. Dále také nepřesnost měření je $\pm 2^\circ\text{C}$. DHT22 má sice větší měřicí rozsah (−40°C – 125°C) i menší nepřesnost ($\pm 0,2^\circ\text{C}$) než DHT12, ale zase více než 2x dražší.

■ 4.1.2 Bezdrátové stanice

Abych mohl měřit teplotu na více místech než jen u BBB, tak jsem vytvořil další bezdrátové stanice. Jako jejich základ jsem použil desku Wemos D1 mini, která je založená na wifi modulu ESP8266.

Tato deska disponuje 18 piny. Kromě napájení a reset pinu mezi nimi najdeme i 11 GPIO a 1 AIN. Všechny digitální piny mohou vyvolávat přerušení a najdeme zde i I2C či UART. Také jdou použít pro připojení zařízení přes 1-wire. Je zde i microUSB port, přes který můžeme desku napájet a/nebo programovat.

¹ <https://www.aliexpress.com/>



Obrázek 4.3. Obsah balení Wemos D1 mini.

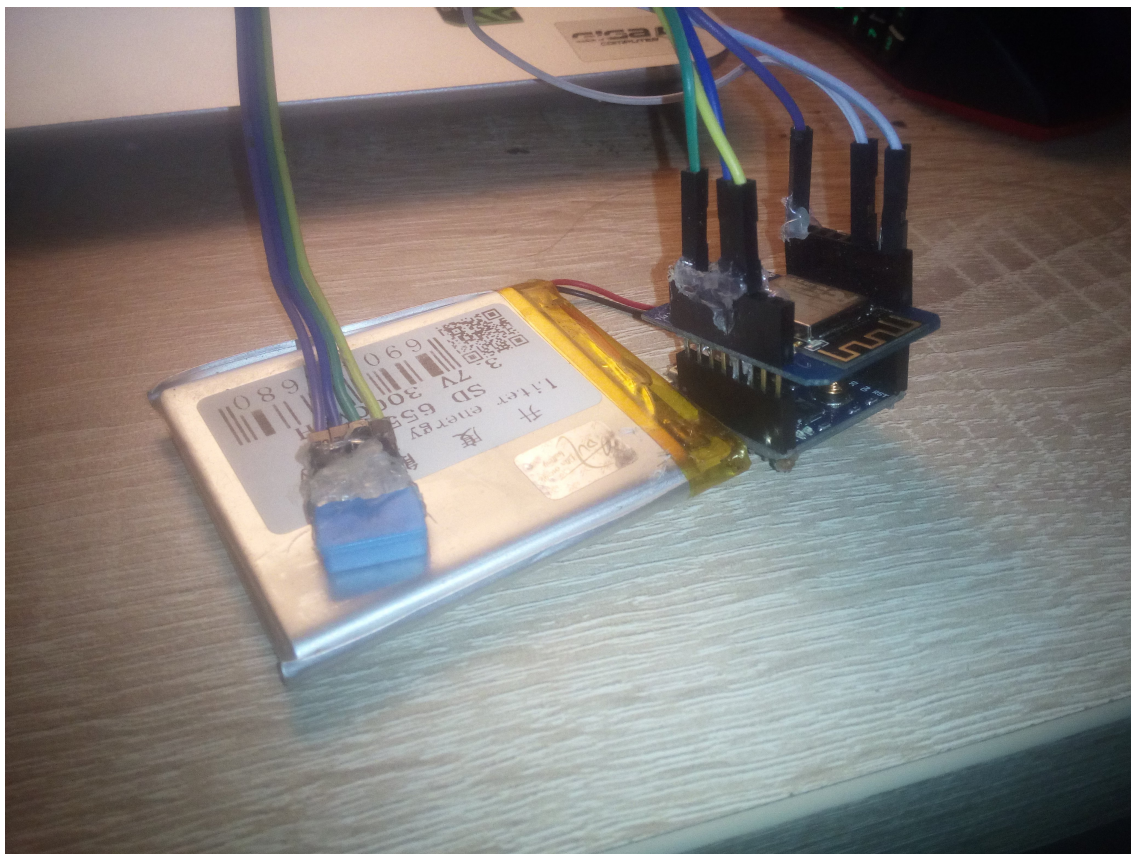
Kromě již popsaných vlastností je výhodou jeho kompatibilita s Arduinem nebo NodeMCU. Ty mají velkou uživatelskou základnu, takže není problém s připojením téměř jakéhokoliv senzoru.

Další výhodou je, že díky jeho rozšířenosti již existuje mnoho tzv. shieldů. To jsou desky se stejným rozložením pinů jako základní deska, které na sobě mají nějaké senzory, nebo jiné prvky jako například relé. Díky stejnému rozložení pinů, stejné velikosti a osazení headery s dlouhým pinem (samcem) směrem dolů a samicí směrem nahoru, můžeme několik těchto shieldů dát nad sebe a tím jednoduše připojit více senzorů. A také se jimi neblokují piny a jsou tedy využitelné pro další zařízení, které můžeme připojit pomocí vodičů.

Pro připojení senzoru teploty jsem ale využil externí připojení pomocí vodičů. Připojením senzoru v těsné blízkosti wifi modulu by mohlo dojít k přenášení tepla a tím ke zkreslení výsledků.

Pro měření teploty uvnitř jsem vytvořil verzi, kde je připojen pouze senzor teploty. Napájení je pak zajištěno přes microUSB port pomocí klasické nabíječky na telefon.

Venkovní senzor jsem nechtěl umísťovat do blízkosti domu, protože i když je zateplený, je stejně v jeho těsné blízkosti o něco vyšší teplota. Senzor jsem tedy umístil na zahradu



Obrázek 4.4. Venkovní měřící stanice. Dolní deska je battery shield, horní je samotný Wemos D1 mini, vlevo pak je baterie a na ní položený senzor.

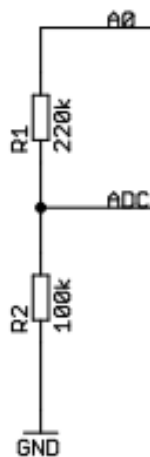
do kotce. V něm ovšem není zavedena elektrika, takže jsem musel vyřešit napájení jinak než adaptérem do zásuvky.

Vyřešil jsem to tedy použitím battery shieldu, ke kterému lze připojit Li-Pol baterie. Ten zajišťuje převod napětí z baterie na 5V, které jsou potřeba pro provoz stanice. Dále má na sobě microUSB port, ze kterého lze baterii nabíjet. Je zde IC, který řídí průběh nabíjení a také kontroluje, aby se baterie nevybila pod napětí, kdy pak už nebude fungovat.

Díky analogovému vstupu Wemosu můžeme měřit napětí na baterii a případně poslat upozornění, že je potřeba baterii dobít. ADC v ESP8266 je 8-bitový a umí měřit pouze 0–1V, všechny hodnoty mezi 1–3,3V budou prezentovány jako 1024 a pokud bychom přivedli napětí vyšší, můžeme MCU poškodit [12]. Maximální napětí použité baterie je 4,2V, musím proto použít dělič napětí. Ze schématu Wemos D1 mini 4.5 jsem zjistil, že je již použit dělič $R1 = 220k\Omega$, $R2 = 100k\Omega$.

Vzorec 4.1. $V_{out} = \frac{V_{in} \times R_2}{R_1 + R_2}$

V_{out} je napětí jdoucí do ADC, V_{in} je vstupní napětí z baterky. Když dosadíme 4,2V, dostaneme napětí jdoucí do převodníku 1,313V, což je nad jeho rozlišovací schopnost. Jelikož $R2$ je přímo mezi vstupem do ADC a zemí, nemůžeme ho změnit bez zásahu do



Obrázek 4.5. Schéma zapojení analogového vstupu Wemos D1 mini (zdroj: [11]).

desky. Jediný způsob jak změnit převodník, je zapojit další rezistor do série s R_1 a tím zvětšit jeho odpor. Ze vzorce 4.1 jsem tedy vyjádřil R_1 4.2.

Vzorec 4.2. $R_1 = \frac{V_{in} \times R_2}{V_{out}} - R_2$

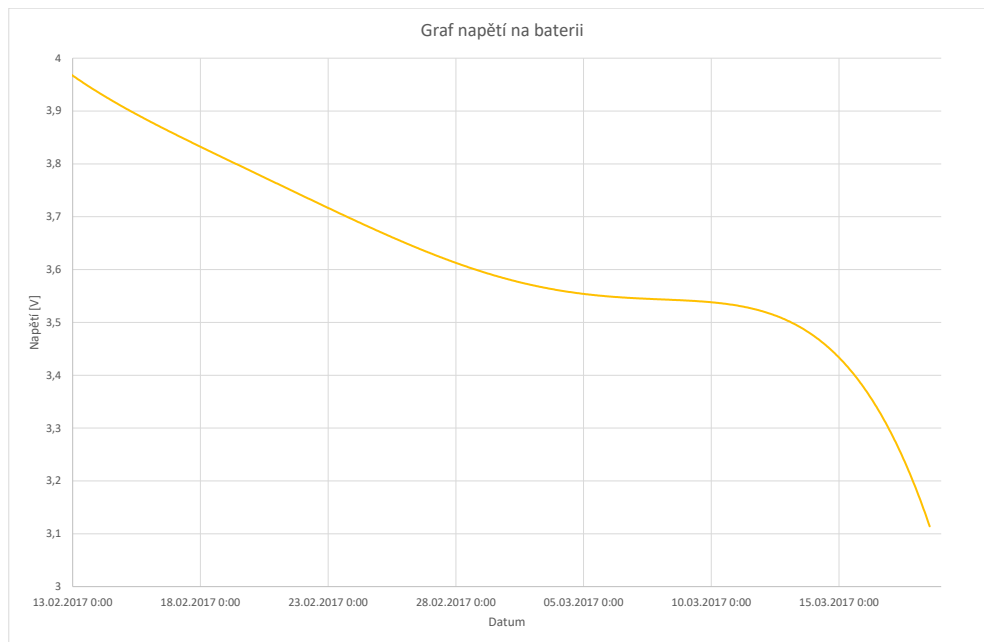
Pak jsem dosadil hodnoty a vyšlo mi že R_1 musí být $320k\Omega$, aby při vstupním napětí $4,2V$ bylo výstupní napětí $1V$. Mezi kladný pól baterie a vstup A0 musím tedy zařadit $100k\Omega$ rezistor. Celý odporový dělič tedy bude mít odpor $420k\Omega$ což znamená, že při napětí $4,2V$ na baterii, bude přes dělič procházet proud $10\mu A$, což je vzhledem ke kapacitě baterie, kterou jsem použil ($3000mAh$), zanedbatelné. Z grafu 4.6 je vidět, že v této konfiguraci baterie vydržela více než měsíc.

Později jsem jako vylepšení ještě připojil solární panel a pak stanice vydržela bez nabíjení od půlky dubna až do konce října. Později už byl moc krátký den na to, aby se stihlo dobít to, co se vybil přes noc.

Baterie by šla použít i jako záložní zdroj pro stanice uvnitř. Protože však jsou stanice k serveru připojeny přes WiFi, při výpadku proudu nebudou schopny se spojit se serverem a Wemos D1 mini nemá žádnou uživatelskou paměť, kam by se mohli dočasně ukládat naměřené hodnoty pro pozdější odeslání.

4.2 Vzdálený přístup

Jeden z hlavních cílů mé práce bylo, aby se k datům ze sensorů dalo dostat i vzdáleně. Toho jde docílit několika různými způsoby. Já jsem v této práci vyzkoušel 3 z nich a to vzdálenou plochu, pak SSH a SCP a jako poslední je, že se na stanici spustí webový server a pak se k němu přistupuje přes webový prohlížeč. Další možností by bylo vytvořit nějaký vlastní protokol se serverovou a klientskou aplikací, ale v dnešní době se již většinou používají aplikace založené na HTTP.



Obrázek 4.6. Graf napětí na baterii v čase.

■ 4.2.1 Vzdálená plocha

Když se připojíme ke vzdálené ploše, vypadá to v podstatě jako kdybychom byli přímo na tom počítači. Vyzkoušel jsem dva různé způsoby připojení a to přes VNC, a přes TeamViewer.

První z nich má výhodu, že je zcela zdarma. Je na BBB už předinstalován na původním obraze. Další výhodou je, že se připojujeme přímo na ten počítač a nejde to přes žádný jiný server. To s sebou nese i nevýhodu a to, že se o zabezpečení musíme postarat sami. Nevýhodou je, že počítač, ke kterému se chci připojit, musí být buď na stejné síti jako klient, nebo musí mít veřejnou IP adresu.

Druhý z nich je zdarma pouze pro domácí použití. Na druhou stranu můžeme použít jejich servery, abychom se spojili s jakýmkoliv počítačem na světě, který má tento program spuštěný. Stačí znát jen ID, které je přiděleno při instalaci, a heslo, které je přiděleno při každém startu. Můžeme si také vytvořit účet, pak se můžeme připojit ke všem počítačům, na kterých je přihlášen stejný účet, bez zadávání hesla. Pokud je klient a server na stejné síti, mohou se spojit i bez přístupu k internetu pouze zadáním IP adresy.

Vzhledem k tomu, že žádný z programů z této práce nepoužívá GUI, využívalo by se zde pouze spuštění terminálu. K tomu se můžeme dostat i přes následující službu.

■ 4.2.2 SSH/SCP

Přes SSH se můžeme vzdáleně připojit k linuxovému terminálu. Oproti předchozímu způsobu se při tomhle používá mnohem menší objem dat. SSH server běží takřka na každém zařízení s OS Linux.

Abychom se mohli připojit, musíme stejně jako u VNC být na stejné síti. Pokud jsme pouze na lokální síti, býváme většinou za nějakým routerem a nemůže se k nám tedy připojit nikdo přes internet. Ovšem pokud má server veřejnou IP adresu, musíme se o zabezpečení postarat sami.

V základu jsou dvě možnosti jak toho docílit:

- Heslo – přihlašujeme se klasicky pomocí uživatelského jména a hesla.
- Pár klíčů – pro přihlášení musíme mít vygenerovaný pár klíčů (veřejný a soukromý), veřejný pak nahrajeme na server a soukromý použijeme k zašifrování náhodných dat, která server pošle při pokusu o přihlášení, pokud data po rozšifrování veřejným klíčem odpovídají odeslaným, jsme vpuštěni dovnitř.

Můžeme používat i kombinaci obou možností. Pokud však použijeme přihlašování pomocí hesla, musíme zabezpečit, aby se nedalo do nekonečna zkoušet hesla, ale nastavit nějaké timeouty.

Nástavbou k SSH je SCP, přes který můžeme jednoduše přistupovat k souborovému systému. Lze pak využít editory nainstalované na vlastním počítači a ne pouze textové editory skrz terminál.

Tento způsob připojení je však jen pro pokročilé uživatele, protože se musíme spokojit s textovým rozhraním. Většina uživatelů je však zvyklá na grafické rozhraní, proto jsem použil třetí způsob vzdáleného přístupu.

■ 4.2.3 Prohlížeč

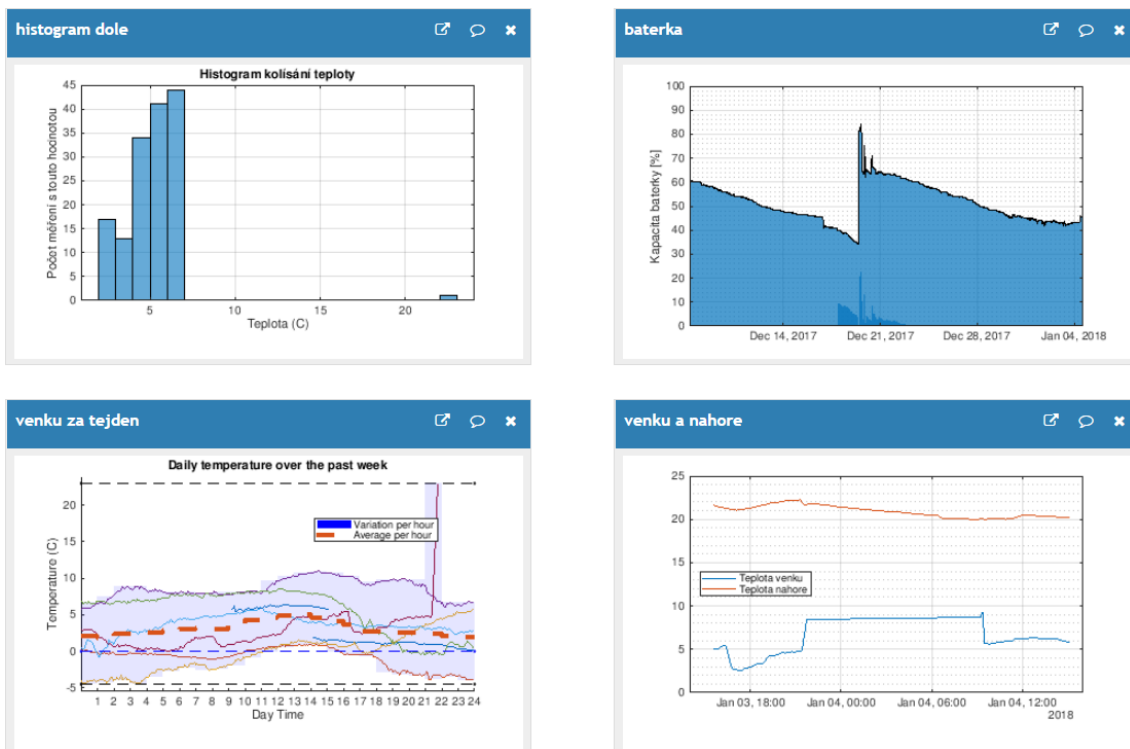
V dnešní době nejvyužívanější způsob vzdáleného přístupu k systémům je, že se přes webový prohlížeč připojíme na server. Výhodou tohoto způsobu je, že na klientském přístroji není potřeba instalovat žádný dodatečný program, jelikož nějaký prohlížeč je na téměř všech operačních systémech již předinstalován. S dnešními technologiemi je možné vytvářet webovou aplikaci, která zvládá v podstatě vše, co by zvládla klasická aplikace.

Nejpoužívanějšími technologiemi pro vytvoření webové aplikace jsou dnes Java a Python. Zkoušel jsem použít Javu, ale narazil jsem na problém, že na BBB je pouze 4GB místa a jednoduchá webová aplikace, která pouze zobrazila jednu stránku, zabírala přes 500MB. Zatímco stejná stránka v Pythonu zabírala méně než 1MB.

Rozhodl jsem se tedy pro webové služby využít jazyk Python. Vytvořil jsem jednoduchý server, který přijímá zprávy typu HTTP POST, které mají definovanou strukturu,

a data z nich pak ukládá do lokální SQL databáze. Z té si je pak může vyčíst jakýkoliv jiný program a buď je nějakou zpracovat, nebo odeslat na externí server.

Protože čtení jednotlivých hodnot většinou není moc užitečné, rozhodl jsem se, že data budu vykreslovat do grafů. Po prozkoumání již fungujících služeb jsem usoudil, že nejlepší bude použít jednu z nich. Jako nejlepší mi přišla služba Thingspeak od Mathworks¹. Pro nekomerční použití je zdarma, ale má omezení, že můžeme poslat pouze 3 miliony měření za rok, což ovšem odpovídá asi 30 stanicím, které by posílali data jednou za 5 minut. Jednou z velkých výhod je, že nabízí možnost zpracovávat data pomocí Matlabových serverů. Data z více kanálů můžeme kombinovat a vykreslovat do různých grafů jako například 4.7.



Obrázek 4.7. Grafy vytvořené přes Thingspeak.

Platforma je Open Source, takže si její zdrojové kódy můžeme stáhnout a nainstalovat si přímo na svém zařízení. Pak nejsme omezeni počtem přijatých měření, ale abychom mohli využívat zpracovávání dat pomocí Matlabu, musíme mít licenci. Nevýhodou je pak ještě to, že je to rozsáhlý projekt a má mnoho závislostí, které zaberou dost místa na disku. Vzhledem k tomu, že jsme na BBB omezeni 4GB paměti, bylo by lepší tento server nainstalovat na nějakém jiném zařízení.

¹ <https://thingspeak.com/>

4.3 Neveřejná IP adresa

Pokud nemá naše zařízení veřejnou IP adresu a není tím pádem přístupné přímo z internetu, můžeme použít VPN. Když se my, i naše měřicí zařízení, připojíme na stejný server, vytvoří se virtuální síť a pak můžeme k zařízení přistupovat stejně, jako bychom byli na stejné lokální síti. Aby toto fungovalo, je potřeba alespoň jeden počítač (server), ke kterému je možné přistoupit odkudkoliv z internetu. Pokud takovýto server nemáme, můžeme pak využít externích služeb.

Jednou z nich je LogMeIn Hamachi¹. Stačí si nainstalovat na obou koncích jejich klienta, zaregistrovat se a pak už můžeme vytvářet virtuální síť. Pokud chceme tuto službu využívat zdarma, můžeme propojit maximálně 5 zařízení.

Další možností, jak se připojit na server, který nemá veřejnou IP adresu, je přesměrování určitých portů. Pokud se domluvíme s poskytovatelem internetového připojení, může nám otevřít pouze některé z portů, na kterých pak můžeme provozovat svou službu. Něčím podobným je pak Pagekite². Je to program, který si nainstalujeme do zařízení a pak si nastavíme, které porty se mají přesměrovávat a na jaký „kite“ neboli adresu. Ta je pak přístupná odkudkoliv z internetu. Můžeme přesměrovávat jak webové stránky, tak i SSH.

Pokud nemáme vlastní server s veřejnou IP, můžeme použít přímo servery vývojáře, ale je to placená služba a má i omezený datový tok. V opačném případě můžeme tuto službu nainstalovat na vlastním serveru a pak se připojovat na ten. Je pak vhodné mít i vlastní doménu, abychom nemuseli zadávat IP adresu.

¹ <http://vpn.net/>

² <https://pagekite.net/>

Kapitola 5

Vysoká dostupnost

Zaručení vysoké dostupnosti se skládá ze dvou základních částí. Je potřeba zajistit napájení a je potřeba zajistit, aby byl systém schopný komunikovat s okolím.

5.1 Zdroj napětí

První část lze udělat pomocí komerčních UPS. Ty používají akumulátor, který se za normálního provozu nabíjí. Pokud se přeruší napájení, přepne se výstup, který je normálně připojen na vstup, na měnič, který je napájen tím akumulátorem. Nejčastěji se používají 6V nebo 12V olověné akumulátory. Výstupem zde obvykle bývá střídavé napětí stejně jako je v rozvodné síti, takže 220V.

To pro naše použití není úplně vhodné, protože BBB potřebuje pouze 5V stejnosměrného napětí. Byli by tu proto zbytečné ztráty energie při převodu na 220V střídavého a pak zpět na 5V stejnosměrného napětí. Rozhodl jsem se tedy, že vytvořím vlastní UPS.

Jak již jsem popisoval v předchozí kapitole je možné napájet BBB pomocí Li-Pol baterie, ovšem ty jsou poměrně drahé, pokud je požadována větší kapacita. Druhou možností bylo napájet pomocí klasické power banky s USB výstupem, ale tam narážíme na stejný problém. Rozhodl jsem se tedy použít olověný akumulátor a k tomu obvod na stabilizaci napětí.

Nejdříve jsem chtěl použít 3-článekový tedy 6V akumulátor, ale narazil jsem na problém, když se akumulátor vybíjí, tak klesá jeho napětí a může se dostat až velmi blízko k požadovanému výstupnímu napětí. S tak nízkým rozdílem se mi nepodařilo nalézt žádný regulátor, který by zvládal dodávat alespoň 1A. Rozhodl jsem se tedy použít 6-článekový (12V) akumulátor, který poskytuje dostatečné napětí, abychom mohli použít i lineární regulátor s výstupním proudem 1,5A.

Dále jsem musel vyřešit jak napájet (nabíjet) baterii. Rozhodl jsem se, že nemá cenu vyrábět něco co se již velmi hojně používá, a koupil jsem nabíječku. Z pozorování jsem ovšem zjistil, že tato nabíječka funguje tak, že po připojení začne nabíjet akumulátor. Když se pak dosáhne určité úrovně napětí (7V pro 3-článekový a 14V pro 6-článekový akumulátor), nabíjení se ukončí. Aby znovu začalo, musí se buď přerušit a obnovit napájení nabíječky, nebo se musí odpojit a znovu připojit k akumulátoru. Rozhodl jsem se tedy dát mezi nabíječku a akumulátor relé, kterým mohu propojení přerušit a tím znovu nastartovat nabíjecí proces.

Další věc, kterou jsem musel zajistit, byla, aby se akumulátor nevybil příliš. Když by se napětí na jednom článku dostalo pod 1,75V (celkově pod 10,5V), probíhá proces tzv. sulface – vznikání krystalků síranu olovnatého a tím ke snižování kapacity. BBB poskytuje analogové vstupy, takže napětí na akumulátoru můžeme měřit. Aby se procesor nezničil, musí se použít odporový dělič napětí, který převede měřené napětí na max 1,8V. Výrobce doporučuje aby rezistor R2 4.5 byl maximálně 1kΩ. Protože je dobré aby touto větví procházel co nejmenší proud, rozhodl jsem se použít právě tuto hodnotu. Podle vzorce 4.2 jsem vypočítal hodnotu pro rezistor R1. Jako vstupní napětí jsem počítal 15V – víc by na akumulátoru být nemělo ani při nabíjení a vyšlo mi 7,3kΩ, použil jsem nejbližší vyšší (abych měl rezervu) z řady, tedy 8,2kΩ. Jelikož je zde převodník 12-bitový musí se vyčtenou hodnotou vydělit maximální hodnota (4096). Tu pak vynásobit maximálním napětím (1,8V). Nakonec použít vzorec 4.1, abychom zjistili napětí na akumulátoru.

Tento systém je zapojen do BBB přes USB. To nám zaručí, že se napájení bude brát primárně ze sítě a ne z akumulátoru.

Když se zjistí, že je napětí nižší než zadaný práh, odešle se chybová zpráva a systém se vypne. Když se pak obnoví napájení které jde do kolíku 5VDC, systém se znovu zapne.

5.2 Komunikace

Když je přerušena dodávka energie, často se tím přeruší i připojení k internetu. Je proto nutné zajistit komunikaci jinou cestou. Zajistil jsem jí tedy pomocí posílání SMS.

Tuto cestu jsem zvolil z následujících důvodů: GSM signálem je pokrytý téměř celý svět, a tam kde není, tak většinou není ani internetové připojení, takže je potřeba tam zvolit zcela jiný systém. Pokud je nějaký větší výpadek energie, jsou vysílače vybavené záložními zdroji. A telefon, který umí přijmout SMS má dnes v podstatě každý a nosí ho většinou u sebe.

Toho posledního se dá využít i jiným způsobem, než při nedostupnosti připojení k internetu. SMS se odešle také při závažnějších chybách, které je potřeba řešit co nejdřív. Díky tomu, že máme mobilní telefon stále u sebe, je pravděpodobné, že si této zprávy všimneme velmi brzo.

Pro odesílání SMS jsem vyzkoušel 2 různé moduly. První z nich byla pouze jedna deska plošných spojů s vyvedenými piny, připojil se přes UART a napájen byl 5V přímo z BBB. Je připojen přes UART

Druhý z nich se prodává už jako hotový výrobek, je uzavřen v krabičce a má vlastní napájení – dodává se se síťovým adaptérem nebo ho můžeme napájet stejnosměrným napětím 8–21V. Pro komunikaci se pak používá RS232, ale já jsem použil externí převodník na USB, který emuluje UART. S oběma jsem pak mohl komunikovat stejně, stačí

jen změnit port ke kterému je modul připojen. Oba také podporují tzv. AT příkazy, které se používají pro komunikaci s většinou GSM modulů.

Kapitola 6

Monitoring

Měřicí systémy zvládají čím dále více věcí. To vyžaduje větší složitost, což s sebou nese taky větší možnost chyby. Samozřejmě, většina z nich má v sobě i nějaké kontrolní subsystémy, ale i ty mohou selhat.

Většina složitějších systémů se sestává z řídicího počítače, který vyčítá data ze senzorů a ukládá je na disk, pak je případně ještě odesílá na vzdálený server. Tyto počítače mají možnost připojení do LAN a bývá možné se k nim připojit například přes SSH. Tím se nám otevírá možnost pro vzdálenou kontrolu.

Vytvořil jsem tedy program pro OS Linux, který se připojí k měřicímu systému přes internet, nebo lokální síť a kontroluje jestli se soubory, do kterých se ukládají data, pravidelně mění – jsou do nich zapisována data.

6.1 Základní struktura

Základní funkcionální program je následující 6.1. Připojí se přes SSH ke vzdálenému systému a pomocí shellového příkazu zjistí, jak je soubor se zadanou maskou starý. Může samozřejmě kontrolovat i soubory na vlastním souborovém systému. Pokud je starší než definované maximální stáří, pošle e-mail s chybovou zprávou na zadanou adresu. Když je naše zařízení ve stejném místě jako měřicí systém, můžeme ještě kontrolovat, jestli jde elektrina. Jestliže má měřicí zařízení i nějaký výstup, který určuje, zda je zařízení v pořádku, můžeme ho zapojit do nějaké z pinů na našem BBB. Všechny tyto informace se zadávají do konfiguračního souboru.

6.1.1 SSHFS

Jednou možností jak přistupovat k souborům na vzdáleném souborovém systému bylo přes SSHFS. Pomocí něho můžeme přes SSH připojit vzdálenou složku do vlastního souborového systému a potom k souborům v ní přistupovat jako k lokálním. Problém ovšem nastává, jakmile ztratíme spojení. Pak vykonání jakékoliv akce s těmito soubory neskončí, dokud není spojení opět navázáno.

6.1.2 rsync

Další možností, kterou jsem zkoumal, bylo, že pokaždé stáhneme celou složku z externího souborového systému na lokální. Aby se nemuseli pokaždé stahovat všechny soubory, existuje nástroj zvaný rsync. Ten musí být nainstalován na obou stranách a

rušeno spojení mezi mým a vzdáleným zařízením, trvala odpověď nekonečně dlouho a já jsem to nemohl zjistit.

Musel jsem si tedy knihovnu mírně upravit. V knihovně bylo zadané, že je timeout nastaven na nekonečno, stačilo ho tedy změnit na globální timeout, který je shodný s tím pro připojení. Proto upravenou knihovnu přikládám přímo k programu.

Jakmile se přeruší spojení, program se pokouší opakovaně připojit. Když se mu to nepovede potřetí, tak odešle chybovou zprávu a zkouší to dál.

■ 6.1.4 Kontrola souboru

Pro zjištění stáří souboru používám následující příkaz:

```
echo $(($(date +%s) - $(date +%s -r "ls {maska} -tc | head -1")))
```

Je složený z několika podpříkazů a `{maska}` je souborová maska, např.: `/root/* .log`.

```
date +%s
```

Vypíše aktuální datum a čas jako počet sekund od začátku roku 1970. Přidáním `-r` a cesty k souboru za tento příkaz, se zobrazí stáří souboru (poslední modifikace) v sekundách od začátku roku 1970.

```
ls {maska} -tc | head -1
```

Tento příkaz vypíše nejnovější soubor se zadanou maskou – `ls` vypisuje všechny soubory, `-tc` je seřadí podle stáří vzestupně a `|head - 1` z výsledků vybere jen první.

Když se stane, že je soubor moc starý, tak se ihned odesílá chybová zpráva.

■ 6.2 Nastavení a ovládání

Aby se program dal použít univerzálně, musí být možnost ho nastavit. To je zde zařízeno pomocí konfiguračního souboru, který se načte vždy při startu programu. Tento soubor má danou strukturu, která bude popsána v následující kapitole.

Ovládat systém lze dvěma způsoby. Můžeme se připojit přes SSH a použijeme příkaz `watchdogCommander [příkaz]` v terminálu, nebo přes SMS zprávu na telefonní číslo SIM karty, která je v GSM modulu, ve které je pouze příkaz. Tyto příkazy jsou pro oba styly připojení stejné. Je možné buď získat aktuální status (6.2.2) nebo restartovat server (6.2.3). Odpověď se dozvíme buď v terminálu, kde jsme příkaz zadali, nebo přijde SMS zprávou na číslo, ze kterého jsme příkaz odeslali.

■ 6.2.1 Konfigurační soubor

Konfigurační soubory nemají obecně daný formát, ale obvykle se jedná o soubory, které jsou čitelné i pro člověka, aby je mohl uživatel (administrátor) lehce upravovat. V UNIXových systémech se obvykle tyto soubory dávají do složky `/etc/` a mívají příponu

conf, *cfg*, *ini* a podobně. Jsou v nich zapsané základní nastavení/vlastnosti programů, bez kterých by nemohl fungovat.

Já jsem použil *ini* soubor, což je zkratka pro inicializaci. Soubory v tomto formátu jsou rozděleny do jednotlivých sekcí, které obsahují 0 a více vlastností. Sekce jsou označeny názvem v hranatých závorkách a vlastnosti jsou ve formátu *vlastnost = hodnota*. Pokud hodnoty obsahují mezery nebo jiné zvláštní znaky, musí se dát do uvozovek, když je potřeba více řádků, musíme dát na konec každého, po kterém má následovat další, zpětné lomítko. Řádky začínající „#” fungují jako komentáře, stejně jako vše co je za „;”. Díky tomu, že je tento formát standardizován, mohl jsem použít již existující knihovnu pro načtení hodnot¹. Knihovna je „Case Sensitive”, takže je potřeba dodržovat velikost písmen.

Soubor musí obsahovat sekci *general*, ve které jsou hodnoty společné pro všechny servery. V této sekci je pouze *gsmPort*, což je adresa, kde je připojen GSM modul. Může být například */dev/ttyO2*, pokud máme modul připojený přes UART, nebo */dev/ttyUSB0*, když je připojen přes USB převodník.

Dále pak je jedna sekce pro každý ze serverů/měřících zařízení. Název je vždycky jméno tohoto zařízení, které se pak používá při ovládání a ostatní komunikaci s uživatelem. Pro správné fungování musí být všechny následující vlastnosti vyplněny. Do *url* se dává adresa, ke které se připojujeme přes SSH, pro správné připojení musí být zadáno i uživatelské jméno. Dále se nastavuje *mask*, což je souborová maska. Parametrem *delay* nastavujeme jak často se má soubor sledovat, tato hodnota je shodná s maximálním povoleným stářím souboru. Telefonní číslo, na které se mají odesílat chybové SMS zprávy se nastavuje do *phoneNumber*, zadává se v mezinárodním formátu. Obdobně do *email* se nastaví kam se mají odeslat chybové e-maily. Pak tu jsou 3 parametry, které se týkají SMTP serveru. Jeho adresa se dává do *smtp*, uživatelské jméno, které se používá i jako adresa odesílatele, do *username* a heslo do *password*. Posledním parametrem je *restartPin*, což je pin, kterým se provádí tvrdý restart. Je ve formátu *header_pinNumber* například *P8_10* a je nepovinný.

Pokud chceme sledovat více souborů na jednom serveru, můžeme souborové masky oddělit vvislou čarou. Stejně tak je možné zadat i ostatní parametry. Jestliže chceme pro všechny masky používat stejné, necháme tam pouze jednu hodnotu, jinak je potřeba zadat stejný počet hodnot jako masek oddělených vvislou čarou.

Následuje příklad takového souboru:

¹ <http://ndevilla.free.fr/iniparser/>

```
# obecné nastavení shodné pro všechny servery
[general]
gsmPort = "/dev/ttyUSB0";

# Nastavení pro jednotlivé servery
[serverName]
url = "root@192.168.7.2";
mask = "/fs1/fpga_cnt/T1_s*.dat|/fs1/fpga_cnt/T1_r*.dat";
delay = "60|60";
phoneNumber = "+420603123456";
email = "receiver@domain.com";
smtp = "smtp://smtp.gmail.com:587";
username = "smtp@user.name";
password = "smtpPassword";
restartPin = "P8_10"
```

6.2.2 Čtení statusu

Chceme-li získat stav zadáme příkaz

```
status [serverName]
```

V parametru *serverName* je jméno serveru, které je nastavené v konfiguračním souboru, pokud ho necháme prázdný, je vrácen status ze všech serverů. Odpověď je ve tvaru

```
serverName fileMask status lastOK
```

Kde *serverName* je jméno serveru, *fileMask* je jedna z masek. *status* může být buď *OK*, pokud je vše v pořádku, nebo chybová hláška, která je stejná, jako byla odeslána zprávou. V *lastOK* je pak čas, kdy byl soubor naposledy v pořádku.

6.2.3 Restart serveru

Restart serveru se provádí pomocí příkazu:

```
restart [restartType] [serverName]
```

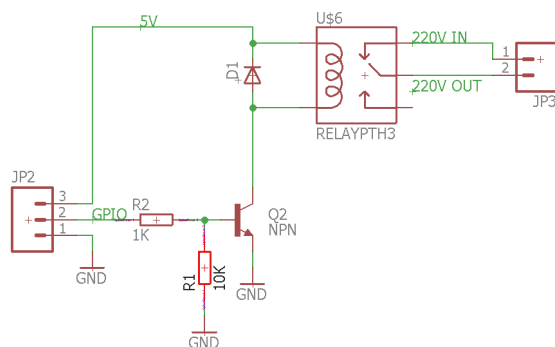
Zde je *[serverName]* povinné. A *[restartType]* může nabývat hodnot *soft* nebo *hard*. Tento příkaz je (z bezpečnostních důvodů) možné zadat jen prostřednictvím SSH, nebo z telefonního čísla, na které se z daného serveru odesílají chybové hlášky.

Soft restart, nebo-li měkký restart se provádí pomocí příkazu

```
shutdown -r now
```

zaslaného přes SSH. Může se ovšem stát, že server na příkaz nezareaguje, nebo že vůbec není možné navázat spojení. Pak je potřeba provést tzv. tvrdý restart. To znamená, že na nějaký čas (např. 5s) přerušíme napájení. K tomu můžeme použít relé v následujícím zapojení 6.2.

V tomto zapojení je proud přerušen jen pokud je připojeno napájení a zároveň je přivedeno napětí i na vstup (GPIO). Díky pull-down odporu R1 nehrozí, že by se



Obrázek 6.2. Schéma zapojení relé.

relé přepnulo pokud bude vstup „ve vzduchu“. Tím zaručíme, že se nepřeruší proud v případě, že bude BBB vypnutý.

Budeme-li napájet více/větší zařízení, je lepší použít větší relé nebo případně stykač, který je potřeba napájet větším napětím než 5V. Je pak potřeba akorát propojit země BBB a napájení a do zdířky 3 na JP2 připojíme dané napětí. Také je možné koupit už hotovou prodlužku, která se dá takto ovládat.¹

¹ <https://www.adafruit.com/product/2935>

Kapitola 7

Závěr

Cílem této práce bylo rozebrat možnosti vytvoření inteligentního měřicího systému založeného na platformě BeagleBone Black, který by byl schopný zpracovávat a archiverovat data ze senzorů a měřicích modulů připojených přes různá rozhraní. Dále bylo cílem navrhnout a realizovat takovýto měřicí systém. Systém by navíc měl být dostupný jak přes internet, tak přes GSM služby. Pak měl být vytvořen druhý systém, který bude schopný komunikovat s dalšími počítači v rámci lokální sítě nebo internetu a kontrolovat jejich správnou funkčnost a v případě detekování chyby ji nahlásit a případně počítač restartovat s cílem obnovení jeho činnosti.

Platforma BeagleBone Black se ukázala jako velmi schopná pro komunikaci se senzory přes většinu rozhraní. Bez nutnosti dalších investic je schopná komunikovat přes UART, I2C, USB, Ethernet a další protokoly, které používají 3,3V logiku. A to díky velkému počtu univerzálních vstupně/výstupních pinů. Je také možné v jisté míře měřit napětí díky analogově digitálnímu převodníku.

Díky velkému množství rozšiřujících modulů, tzv. cape, je možno připojit k BeagleBone Black téměř jakýkoliv senzor. Většina těchto cape je open-source, takže je nemusíme kupovat jako celek, ale můžeme si je vyrobit vlastními silami, případně je pozměnit k vlastním potřebám.

Vytvořil jsem systém, který sbírá data o teplotě jak ze senzorů přímo připojených k desce, tak ze senzorů na lokální síti. Tato data se uchovávají v lokální databázi a jsou zasílána na externí server, kde jsou k dispozici pro uživatele a pro další analýzy. Data na externí server se mohou odesílat buď v reálném čase, nebo v dávkách. Není-li připojení k internetu k dispozici v době, kdy se data mají odeslat, odešlou se všechna nasbíraná data, za tuto dobu, po obnovení spojení.

Dále jsem vytvořil systém, který je schopen se připojit k ostatním počítačům na síti a je schopen kontrolovat, jestli jsou data ze senzorů pravidelně zapisována. Jestliže tomu tak není, je uživateli odesláno chybové hlášení a to buď prostřednictvím e-mailu nebo SMS zprávy. Ten pak může tento počítač vzdáleně restartovat a pokusit se tak obnovit správnou činnost. Pokud je měřicí zařízení v blízkosti, může být použit i tvrdý restart, kde se na určitý čas přeruší napájení. Byl vytvořen i obraz operačního systému, kde je tento systém předinstalován.

Oba tyto systémy mohou běžet na BeagleBone Black i současně.

Dále jsem implementoval možnost záložního napájení z olověného akumulátoru. Je to samostatný modul a běží nezávisle na ostatních. Při testování s 12V/7,2Ah akumulátorem byl systém schopný běžet přibližně 8 hodin, než se napětí dostalo na úroveň, kdy se akumulátor již nevratně poškodí.

Při testování těchto systémů bylo zapisováno velké množství dat do logovacích souborů, což se ukázalo jako problém. BeagleBone Black má pouze 4GB paměti a to se během dvou měsíců provozu zaplnilo a systém zkolaboval. Z tohoto důvodu bych tuto platformu nedoporučil pro dlouhodobou archivaci naměřených dat, ale pouze jako mezičlánek mezi senzory a datovým úložištěm.

Literatura

- [1] *BeagleBone Black Connector and Switch Locations* [online]. [vid. 20. 12. 2017].
https://elinux.org/Beagleboard:BeagleBoneBlack#BeagleBone_Black_Picture.
- [2] *BeagleBone Black Pin Map* [online]. [vid. 20. 12. 2017].
<https://www.mathworks.com/help/supportpkg/beagleboneio/ug/beaglebone-black-pin-map.html>.
- [3] *[TUT] [C] Newbie's Guide to AVR PWM* [online]. [vid. 20. 12. 2017].
<http://www.avrfreaks.net/forum/tut-c-newbies-guide-avr-pwm-incomplete?page=all>.
- [4] *What is a successive approximation ADC?* [online]. [vid. 20. 12. 2017].
<https://www.quora.com/What-is-a-successive-approximation-ADC>.
- [5] *UM10204 - I2C-bus specification and user manual* [online]. [vid. 20. 12. 2017].
<https://www.nxp.com/docs/en/user-guide/UM10204.pdf>.
- [6] *I²C - Wikipedia: the free encyclopedia* [online]. [vid. 20. 12. 2017].
<https://cs.wikipedia.org/wiki/I%C2%B2C>.
- [7] *Raspberry Pi 3 layout* [online]. [vid. 20. 12. 2017].
<https://www.memoryexpress.com/Products/MX61462>.
- [8] *Raspberry Pi 3 GPIO Pin Chart with Pi* [online]. [vid. 20. 12. 2017].
<https://openclipart.org/detail/280972/raspberry-pi-3-gpio-pin-chart-with-pi>.
- [9] TRAVAGLIONE, Ben, Andrew MUNYARD a David MATTHEWS. HIGH RESOLUTION UNDERSEA ACOUSTIC DATA ACQUISITION USING SINGLE-BOARD MICROCONTROLLERS. 2015.
- [10] *DHT12 Digital temperature and humidity sensor C83989* [online]. [vid. 2. 1. 2018].
https://datasheet.lcsc.com/szlcsc/DHT12-Digital-temperature-and-humidity-sensor_C83989.pdf.
- [11] *Wemos D1 mini Datasheet* [online]. [vid. 3. 1. 2018].
https://wiki.wemos.cc/_media/products:d1:mini_new_v2_2_0.pdf.
- [12] *ESP8266EX Datasheet* [online]. [vid. 3. 1. 2018].
https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf.