

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ
KATEDRA MĚŘENÍ

Programování mikroprocesorového modulu pro výuku fyziky

bakalant: Petr Kůrka
vedoucí práce: doc. Ing. Jan Fischer, CSc.

květen 2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kůrka** Jméno: **Petr** Osobní číslo: **457191**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**
Studijní obor: **Senzory a přístrojová technika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Programování mikroprocesorového modulu pro výuku fyziky

Název bakalářské práce anglicky:

Programming of Microprocessor Module for Teaching of Physics

Pokyny pro vypracování:

Navrhněte a vytvořte programy pro moduly s mikroprocesory s jádrem ARM Cortex ? M0 pro použití v laboratořích při výuce fyziky. Orientujte se na modul BBC micro:bit a modul s mikrořadičem STM32F042. Programy zajistí měření statických i proměnných napětí a sběr dat ze senzorů fyzikálních veličin a jejich zobrazení na nadřazeném PC. Programy budou také podporovat měření intervalů a čítání impulsů i generování impulsních signálů definovaných parametrů. Vytvořte potřebné programy pro mikroprocesorové moduly i aplikaci pro nadřazené PC. Sestavte příslušné moduly, doplňte pomocnými a ochrannými obvody a vytvořte vzorové laboratorní úlohy pro ověření správnosti řešení.

Seznam doporučené literatury:

- [1] The Definitive Guide to ARM-Cortex-M0 and Cortex-M0+ Processors
- [2] STMicroelectronics: RM0091: STM32F0x2 Reference manual
- [3] <http://microbit.org/>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Jan Fischer, CSc., katedra měření FEL

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

Datum zadání bakalářské práce: **22.01.2018**

Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce:

do konce letního semestru 2018/2019

doc. Ing. Jan Fischer, CSc.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

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

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce doc. Ing. Jana Fischera, CSc. a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

V Praze dne 25. 5. 2018

.....
Podpis bakalanta

Poděkování

Rád bych zde poděkoval za veškerou podporu, které mi během psaní bakalářské práce bylo poskytnuto. Hlavní poděkování patří vedoucímu práce doc. Ing. Janu Fischerovi, CSc. za dobré vedení a nesčetné rady při řešení záležitostí nejen ohledně bakalářské práce. Také bych chtěl poděkovat svojí rodině a přátelům, kteří mi během celého studia byli neocenitelnou oporou.

Anotace

Tato bakalářská práce se zabývá použitím mikrokontroléru jako laboratorního měřícího systému pro výuku fyziky na středních školách. Využívá mikroprocesorových kitů BBC micro:bit a STM32F042 s jádrem ARM-Cortex-M0. Programování kitů probíhá v rámci prostředí ARM mbed. Součástí je také samostatná QT aplikace, ze které lze celý systém ovládat a zobrazovat měřená data.

Klíčová slova

BBC micro:bit, STM32F042, ARM mbed, QT, Software Defined Instrumentation, programování

Summary

This bachelor thesis takes interest in using microcontroller as laboratory measuring system for teaching of physics and electronics at secondary school. Microprocessor modules BBC micro:bit and STM32F042 with ARM-Cortex-M0 core are used. A separate part of the thesis is a QT application, generally used to control the system and present measured data.

Index terms

BBC micro:bit, STM32F042, ARM mbed, QT, Software Defined Instrumentation, programming

Obsah

1	Úvod	11
2	Rozbor dané problematiky	12
2.1	SDI zařízení	12
2.1.1	Hardwarové vlastnosti modulu BBC micro:bit	12
2.1.2	Hardwarové vlastnosti kitů s STM32F042	13
2.2	Komunikace zařízení s nadřazeným PC	15
2.2.1	Komunikace kitu BBC micro:bit s PC	16
2.2.2	Komunikace kitů s STM32F042	17
2.3	Uživatelské rozhraní pro ovládání zařízení	17
2.3.1	Textové rozhraní - emulátor terminálu	17
2.3.2	Grafické rozhraní pro zobrazení průběhů	18
2.3.3	Zobrazení průběhů - Serial plotter	19
2.3.4	Zobrazení průběhů - Serial chart	20
2.4	Konkrétní funkce měřicího systému	21
2.4.1	Funkce PWM generátoru	21
2.4.2	Funkce Voltmetr	22
2.4.3	Funkce Softwarový čítač	22
2.4.4	Funkce Primitivní osciloskop	23
2.4.5	Grafický zobrazovač pro hodnoty okamžitého zrychlení	23
3	Vývojové prostředí ARM mbed	24
3.1	Alternativní označení pinů BBC micro:bit v ARM mbed	24
4	Generování impulzních signálů pomocí PWM	25
4.1	Softwarová realizace obdélníkového signálu	25
4.2	Použití PWMout v ARM mbed	25
4.3	Vícekanálové generování PWM signálů o různých frekvencích v BBC micro:bit	26
4.4	Měření parametrů impulzů čítačem LEO pro kit STM32F303 Nucleo	26
4.5	Parametry PWM pro BBC micro:bit	27
4.6	Odlišnosti pro STM32F042	27
5	Měření napětí tříkanálovým voltmetrem	28
5.1	Použití analogového vstupu v ARM mbed	28
5.2	Měření napětí na volném pinu BBC micro:bit	28
5.3	Charakteristika A/D převodu v BBC micro:bit	29
5.4	Chyby měření napětí v BBC micro:bit	30
5.5	Odlišnosti pro STM32F042	31
6	Měření hodnot okamžitého zrychlení	31
6.1	Použití akcelerometru MMA8652	31
7	Měření nízkých frekvencí softwarovým čítačem	32
7.1	Použití funkce InterruptIn v ARM mbed	32
7.2	Použití funkce Ticker v ARM mbed	32
7.3	Kompenzace chyby čítače v BBC micro:bit	33
7.4	Odlišnosti pro STM32F042	33
8	Měření proměnlivých napětí	34

9 Aplikace měřicího systému	35
9.1 Hlavního okno	35
9.2 Okno Voltmetru	36
9.3 Okno pro PWM generátor	37
9.4 Okno pro softwarový čítač	37
9.5 Nástroj pro grafické vykreslení okamžitého zrychlení	38
9.6 Okno pro grafické vykreslení proměnného napětí	39
10 Závěr	40

Seznam obrázků

2.1 BBC micro:bit	12
2.2 STM32F042K6 Nucleo	13
2.3 STM32F042F6P6 - pouzdro TSSOP20	14
2.4 STM32F042F6P6 - na adaptoru	14
2.5 STM32F042F6P6 - schéma zapojení	14
2.6 STM32F042F6P6 na nepájivém poli	15
2.7 UART komunikace po sériové lince	15
2.8 externí převodník UART - USB	16
2.9 Model externího převodníku UART - USB	16
2.10 Převodník UART - USB pro BBC micro:bit	16
2.11 programová simulace převodníku UART - USB na STM32F042F6P6	17
2.12 Emulátor terminálu Putty	18
2.13 Serial plotter	19
2.14 Serial chart	20
2.15 Detail PWM signálu, parametry	21
2.16 BBC micro:bit zapojení odporového děliče	22
4.1 Ukázka výpisu programu PWM pro BBC micro:bit	25
4.2 Leo Counter v režimu čítače nízkých frekvencí	26
4.3 Tabulka - pwm analýza - BBC Micro:bit	27
5.1 ukázka výstupu Voltmeter na terminál	28
5.2 Aproximace AnalogIn do napětí BBC micro:bit	29
5.3 Schéma zapojení zdroje napájení	30
5.4 VA charakteristika BAT60A	30
6.1 Výpis z akcelerometru na terminál	31
6.2 Výpis z akcelerometru v aplikaci pod QT	32
7.1 Ukázka čítače pro terminál - BBC micro:bit	33
8.1 Ukázka vykreslení proměnného napětí - BBC micro:bit	34
9.1 Hlavní okno v QT aplikaci	36
9.2 okno voltmetru v QT aplikaci	36
9.3 okno PWM v QT aplikaci	37
9.4 okno čítače v QT aplikaci	37
9.5 okno Zobrazovače hodnot zrychlení v QT aplikaci	38
9.6 okno zobrazovače průběhů napětí v QT aplikaci	39
10.1 Kompletní modul BBC micro:bit - přední strana	42
10.2 Kompletní modul s STM32F042F6P6	42

1 Úvod

Při výuce fyziky popřípadě elektroniky je nejlepším způsob, jak upoutat pozornost studentů, demonstrovat probíranou látku na nějakém vhodném pokusu. K realizaci takových pokusů je často potřeba laboratorních přístrojů nebo specificky navržených mechanických modelů. Profesionální

laboratorní přístroje jako například osciloskop, čítač nebo generátor však stále stojí desítky tisíc korun. Jejich obsluha bývá náročná a pro některé učitele natolik neintuitivní, že se radši pokusů vzdají úplně.

Existuje však způsob, jakým nepohodlné a drahé přístroje do jisté míry nahradit. Dnešní mikroprocesory nabízejí dostatečný výkon na to, aby mohly žádané funkce s nižšími nároky vykonávat. SDI neboli software defined instrumentation se zabývá právě realizací těchto funkcí na levnějších procesorech s využitím jejich periférií. Pro rozhraní se pak využívá počítač. Právě komunikace s počítačem je v tomto konkrétním případě klíčová. Výstup z daného experimentu je možné promítnout na jakémkoli běžně dostupném dataprojektoru s použitím počítače, který najdeme v každé učebně. Nároky na pořízení, přípravu i prezentaci jsou tedy mnohem menší než v profesionální laboratoři.

Jedním z hitů výuky programování posledních let je mikroprocesorový kit BBC micro:bit. Jedná se o modul s mikroprocesorem Nordic nRF51822. Pro tento modul existuje vlastní výukové programovací rozhraní, pomocí kterého lze přiblížit problematiku programování a mikroprocesorů studentům, kteří s programováním doposud nepřišli do styku. Využívá se hojně na základních i středních školách po celém světě. Zde se nabízí příležitost ukázat možnosti tohoto kitu právě na fyzikálních a elektronických pokusech. Cílem mé práce proto tedy bude vytvořit uživatelsky přívětivý výukový systém, který do jisté míry nahradí drahé laboratorní přístroje. Při realizaci také určím limity tohoto zařízení.

Pro programování hardwaru použiji integrované vývojové prostředí ARM mbed. To je vhodné jak pro názornost při výuce programování, tak v přenosnosti na další podporovaná zařízení. Jedná se o prostředí dostupné výhradně online.

Realizace pomocí BBC micro:bit má však svoje limity. Proto jako alternativní hardware použiji procesor STM32F042. Ten je ještě o poznání levnější než micro:bit a v některých ohledech ještě výkonnější. S ním se také nabízí možnost, zabudovat daný experiment do laboratoře nastálo.

2 Rozbor dané problematiky

Jak už bylo řečeno v úvodu, cílem této práce bude realizovat jednotlivé fyzikální a elektronické experimenty s využitím kitů BBC micro:bit a STM32F042. Tyto moduly budou vykonávat měřící funkce. řízení a zobrazování dat bude realizováno na nadřazeném PC. Pro komunikaci těchto platforem se využije rozhraní USB.

2.1 SDI zařízení

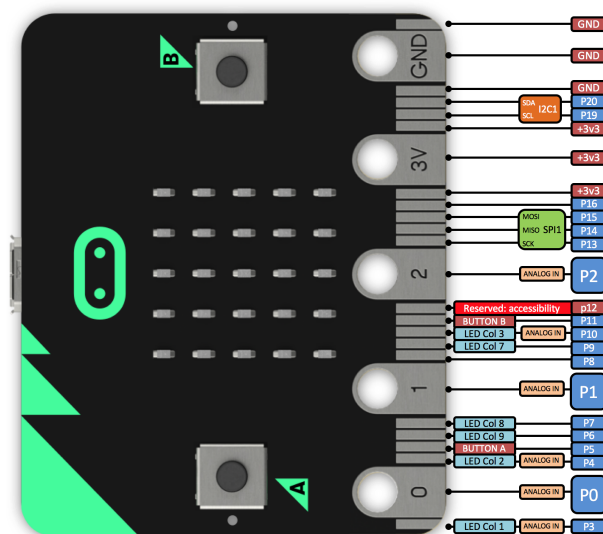
Software defined instrumentation je trendem posledních let. Na katedře měření ČVUT v nedávné době vzniklo více projektů zabývajících se touto problematikou. S rostoucím výkonem procesoru s jádrem ARM Cortex už se naskytá možnost spoustu laboratorních přístrojů nahradit. Procesorové kity disponují velkou řadou periferií pro A/D, D/A převod. Pomocí USB komunikace s PC se pak celý systém ovládá. K tomuto účelu slouží speciálně navrhované aplikace pro konkrétní potřeby měření. V mém případě použiji mikroprocesorové kity BBC micro:bit a STM32F042.

2.1.1 Hardwarové vlastnosti modulu BBC micro:bit

Tento kit byl vyvinut ve Velké Británii za účelem výuky programování zábavnou formou pro žáky již od základní školy. Jeho základem je procesor Nordic nRF51822. Tento procesor má jádro ARM Cortex M0. Disponuje pamětí RAM o velikosti 16 kB. Pro naše účely je zajímavější údaj o velikosti paměti flash. Ta činí 256 kB. Jedním z důvodů volby právě tohoto kontroléru byla i nativní podpora bluetooth komunikace.

Na desce se nachází dvě ovládací tlačítka (A, B) a jedno tlačítko pro reset. Z desky jsou vyvedeny konektory pro připojení banánkových pinů nebo případné osazení do speciálního konektoru od firmy Kitronik. Jako výstup může sloužit matice 5×5 LED diod. Je zde přítomen i tříosý akcelerometr MMA8652. Další část tvoří magnetometr MAG3110. Oba dva tyto moduly komunikují s procesorem prostřednictvím I2C.

Deska je napájena prostřednictvím micro USB konektoru. Je zde i konektor pro připojení externích baterií (2 krát AAA).



Obrázek 2.1: BBC micro:bit

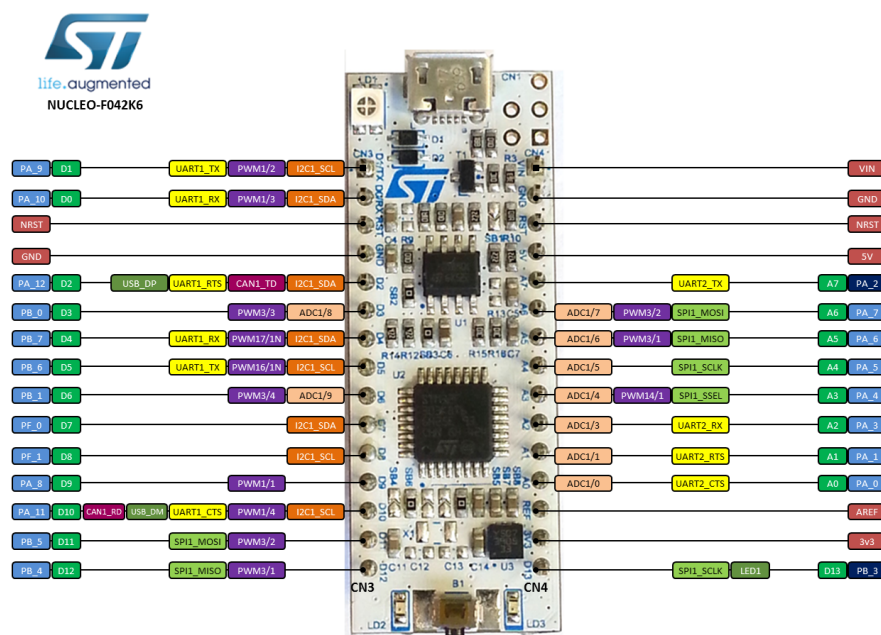
Jak již bylo řečeno dříve, kit je primárně určen do škol. To je právě důvod, proč jej použijí pro účely měřicího systému. Bude se jednat o rozšíření možností použití tohoto zařízení. Po výuce programování pak tedy poslouží i pro demonstraci fyzikálních a elektronických pokusů.

BBC micro:bit však není primárně určen pro aplikace měření. V průběhu realizace jednotlivých funkcí očekávám jisté limity pro přesnost nebo rychlost. Nevýhodou tohoto kitu je pak i jeho cena a špatná dostupnost. Jeden kit se pohybuje kolem 500,- korun a objednat se dá pouze v balení po více kusech. Tyto důvody vedly k volbě druhého zařízení s mikroprocesorem STM32F042.

2.1.2 Hardwarové vlastnosti kitů s STM32F042

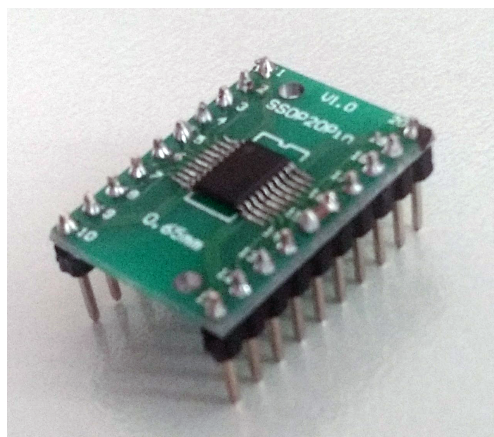
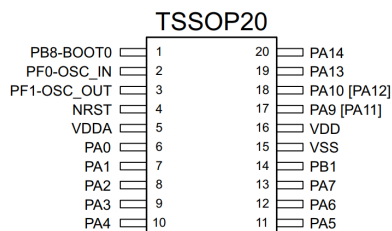
Hlavní výhodou realizace měřicího přístroje pomocí sestavy s tímto procesorem je cena. Samotný procesor se pohybuje v řádu desítek korun. Cena celé sestavy pak závisí na potřebě pro konkrétní úlohu. Pro zabudování do experimentu nastalo je to vhodné řešení.

Tento procesor má rovněž jádro ARM Cortex M0. Nabízí se však ve více pouzdrech a sestavách. Ty se mohou lišit počtem vyvedených pinů nebo velikostí flash paměti. Nejvíce se nabízí kit STM32F042K6 Nucleo. Procesor je v 32pinovém pouzdře na desce, která disponuje dalšími perifériemi. Jednou z největších výhod tohoto kitu je přítomnost ST-Linku pro jednodušší nahrávání programů do procesoru. ST-Link také disponuje můstkem UART - USB, který usnadní komunikaci programu s nadřazeným PC.



Obrázek 2.2: STM32F042K6 Nucleo

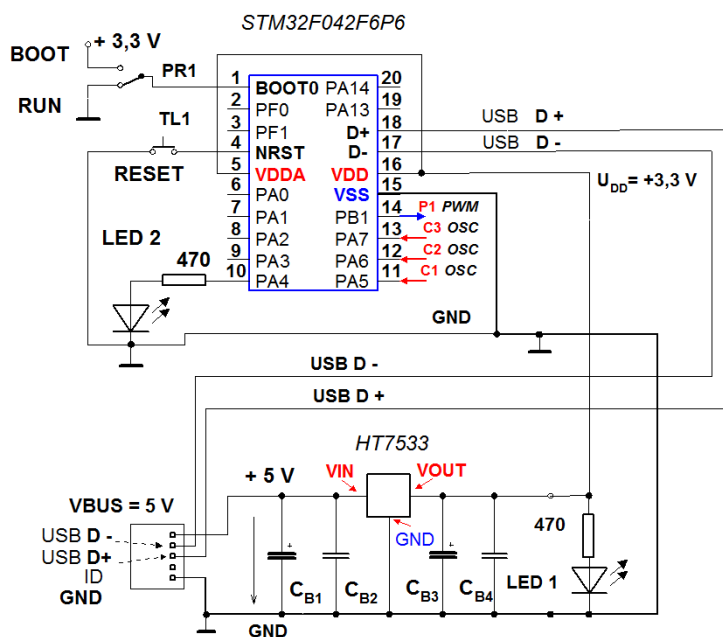
Já použiji holý procesor STM32F042F6P6 v pouzdře TSSOP20. Ten disponuje 20 piny a pro jeho použití je potřeba zapájet na adaptor. Neobejdeme se bez externích součástek na nepájivém poli. Komunikací s PC pro tento konkrétní model se budu zabývat ve vlastní kapitole.



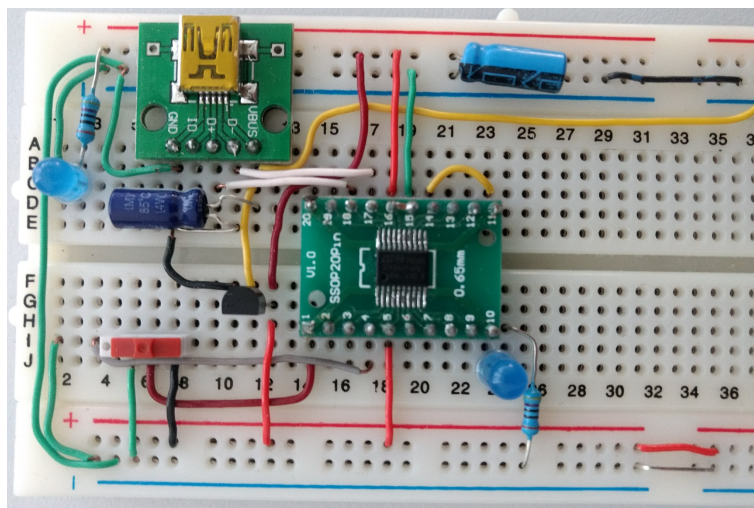
Obrázek 2.3: STM32F042F6P6 - pouzdro TSSOP20

Obrázek 2.4: STM32F042F6P6 - na adaptoru

Zapojení procesoru je realizováno na nepájivém poli s použitím externího USB konektoru, zdroje referenčního napětí a přepínače BOOT - program. Přikládám i schéma zapojení.



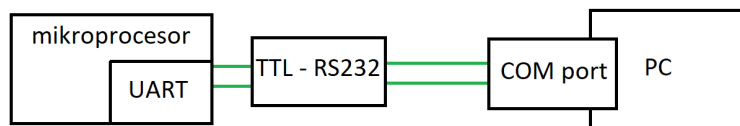
Obrázek 2.5: STM32F042F6P6 - schéma zapojení



Obrázek 2.6: STM32F042F6P6 na nepájivém poli

2.2 Komunikace zařízení s nadřazeným PC

V celé mé práci se využívá asynchronní sériová komunikace. Jedná se o komunikační rozhraní ve formě vysílač (TX), přijímač (RX). Pro tyto účely se využívá blok procesoru UART. Většina desktopových počítačů dříve disponovala rozhraním COM port pro asynchronní sériovou komunikaci. Zařízení podporující UART poté přímo komunikovala s počítačem prostřednictvím sériového rozhraní RS-232(kabel), využívající zvýšené napěťové úrovně. Tato konfigurace je znázorněna v následujícím obrázku.

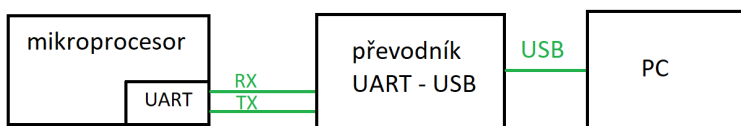


Obrázek 2.7: UART komunikace po sériové lince

Většina dnešních notebooků i desktopových počítačů COM port nemá. Z toho důvodu hledáme alternativní způsob, jak komunikaci uskutečnit. Jednou z možností je využití externího převodníku UART - USB. Výstupní piny procesoru v režimu UART se připojí k daným pinům takového převodníku. Ten se pomocí USB kabelu připojí k PC, které jej registruje jako virtual COM port. Virtuální COM port se od klasického COM portu neliší a obsluha je stejná.



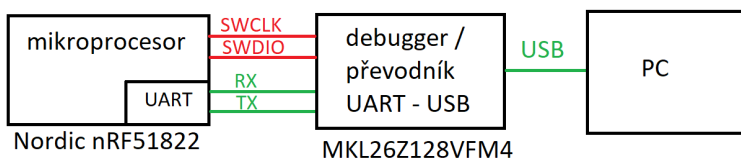
Obrázek 2.8: externí převodník UART - USB



Obrázek 2.9: Model externího převodníku UART - USB

2.2.1 Komunikace kitu BBC micro:bit s PC

Jak již bylo řečeno výše, procesor Nordic nRF51822 v BBC micro:bit přímo podporuje asynchronní sériovou komunikaci, protože disponuje blokem pro UART. Pro naše účely je podstatné, že je na kitu modul MKL26Z128VFM4 pro debugging a nahrávání programů z PC do flash paměti procesoru, který funguje i jako převodník UART - USB. Při správném nastavení sériové komunikace v programu procesoru (USB_RX , USB_TX) tedy probíhá komunikace prostřednictvím tohoto převodníku. Na PC se pak zařízení jeví jako Virtual COM port.

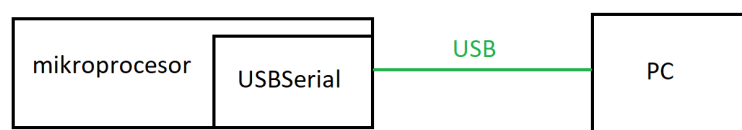


Obrázek 2.10: Převodník UART - USB pro BBC micro:bit

2.2.2 Komunikace kitů s STM32F042

Při použití procesorového kitu STM32F042K6 Nucleo je situace obdobná jako u BBC micro:bitu. Funkci převodníku UART-USB zde zastává ST-Link, který mimo to také slouží pro nahrávání programu do flash paměti mikroprocesoru a případný debugging. Pro všechny účely (debugging, sériová komunikace, napájení) postačí připojení prostřednictvím USB kabelu. V případě potřeby lze však signály RX, TX bloku UART vyvést na jiné piny a použít pro sériovou komunikaci například po sériové lince.

My se však soustředíme levnější variantu s procesorem STM32F042F6P6. V zapojení tohoto procesoru žádný převodník UART - USB není k dispozici. Nabízí se tedy možnost využít externí převodník, což je možné, ale je třeba toto zařízení zakoupit. My však pro minimalizaci počtu elektronických součástí a nákladů využijeme softwarové řešení, které potřebný převodník nahradí. V mbedu existuje knihovna *USBSerial*, která právě převodník UART - USB simuluje a na PC se pak zařízení jeví jako virtual COM port. Jedinou nevýhodou této realizace je velikost použité knihovny, která po překladu zabírá 20 kB z celkové paměti flash 32 kB. Pokud bychom měli nějaký rozsáhlý program a navíc bychom využili tuto knihovnu, mohlo by se stát, že výsledný sestavený program bude příliš veliký a do mikroprocesoru nepůjde nahrát. Pokud by k této situaci došlo, je možné využít externí převodník. Model této konkrétní situace se simulovaným převodem je na následujícím obrázku.



Obrázek 2.11: programová simulace převodníku UART - USB na STM32F042F6P6

2.3 Uživatelské rozhraní pro ovládání zařízení

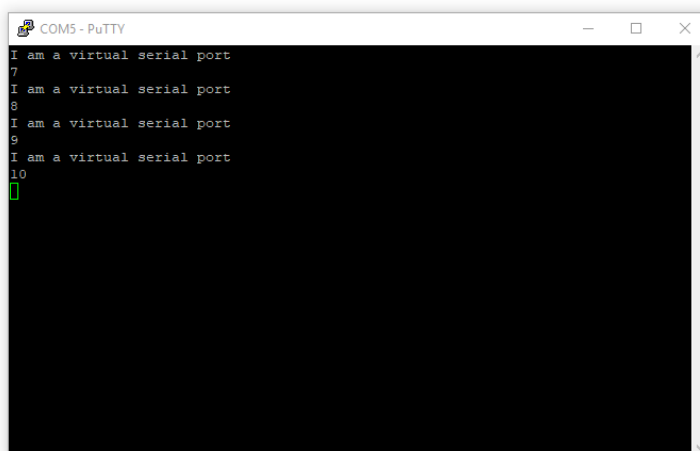
Každé měřicí zařízení potřebuje nějaké uživatelské rozhraní, pomocí kterého bude uživatel schopen číst data, případně modifikovat parametry měření. V mém případě bude uživatel vše ovládat přímo z nadřazeného PC. Pro veškeré příkazy i přijímaná data použijeme ASCII znaky. Možností jak data číst a vysílat je hned několik.

2.3.1 Textové rozhraní - emulátor terminálu

Textové uživatelské rozhraní je nejjednodušší způsob komunikace mezi zařízením a uživatelem. Zařízení vypisuje znaky - zprávy do textového okna a uživatel zadává klávesnicí předem definované příkazy. Volba těchto příkazů by měla být intuitivní a závisí čistě na způsobu programování zařízení.

Terminálem se rozumí zařízení, které takový způsob komunikace zprostředkuje. Historie terminálů sahá již do dob sálových počítačů, kde byl terminál tvořen monitorem a klávesnicí. Monitor zobrazoval textový výstup počítače a klávesnicí se řídil vstup. Komunikace probíhala výhradně po sériové lince nebo modemem a telefonní linkou. Dnes již počítače disponují graficky pokročilými zobrazovači. Pokud chceme použít uživatelské rozhraní typu terminál, musíme využít emulátor terminálu. Výhodou je, že pro každý operační systém nějaký emulátor terminálu existuje. Program mikroprocesoru určený pro terminálové prostředí

je tedy bez omezení použitelný na libovolném PC. Ve Windows však standardně takový program emulátoru není zahrnut, proto je třeba využít program třetí strany. Příkladem takového programu je Putty.



Obrázek 2.12: Emulátor terminálu Putty

Po instalaci takového programu si uživatel nastaví parametry komunikace a může zahájit spojení s připojeným zařízením. V Putty lze také nastavit barvy i velikost použitých fontů, což je důležité, chceme-li experiment předvádět ve třídě plné studentů, kde je potřeba aby na výpisy všichni viděli.

Dalším zajímavým programem pro emulaci terminálu je Termite. Tento program sice nenabízí tak rozsáhlé možnosti modifikace jako Putty, ale pro běžného uživatele je mnohem jednodušší, co se konfigurace připojení a parametrů komunikace týče. Tento program budu také v průběhu realizace práce používat.

Terminálové zobrazení je ze své podstaty vhodné pro řízení výstupních veličin nebo pro statická měření. Pokud budeme chtít zobrazovat data kontinuálně nebo graficky, bude třeba lepších nástrojů. Terminálové zobrazení není vhodné pro vykreslování grafů průběhů.

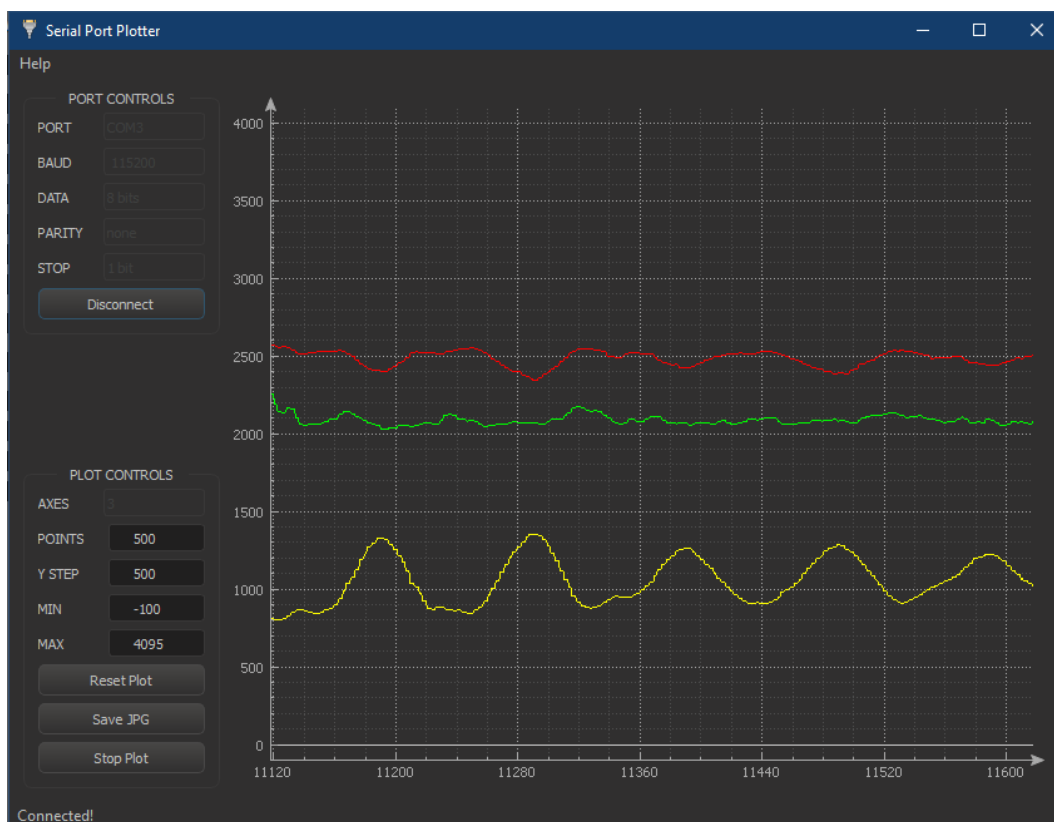
2.3.2 Grafické rozhraní pro zobrazení průběhů

Grafická rozhraní používáme převážně, pokud chceme vykreslovat časově závislé průběhy. Samozřejmě se dají použít i pro statická měření nebo řízení experimentu. Výhodou je, že uživatel nemusí znát konkrétní textové příkazy a stačí mu klikat na zrovna potřebná tlačítka. Problémem však může být kompatibilita. Tato uživatelská rozhraní jsou často vyvíjena pro jeden konkrétní operační systém a jejich přenositelnost na ostatní platformy není zaručena. Tato situace nastává převážně kvůli tomu, že systémy s jádrem Windows a Unix (Linux, Mac OS...) používají jiné protokoly pro instalaci aplikací a jejich spouštění.

Pro rychlá měření se dají použít již existující, běžně dostupné aplikace. Jejich výhodou je univerzálnost a snadná obsluha. Využití pak závisí na konkrétní potřebě a je limitováno pouze vlastnostmi kytu popřípadě komunikace. Funkce grafického zobrazení bude zahrnuta i v mé vlastní QT aplikaci.

2.3.3 Zobrazení průběhů - Serial plotter

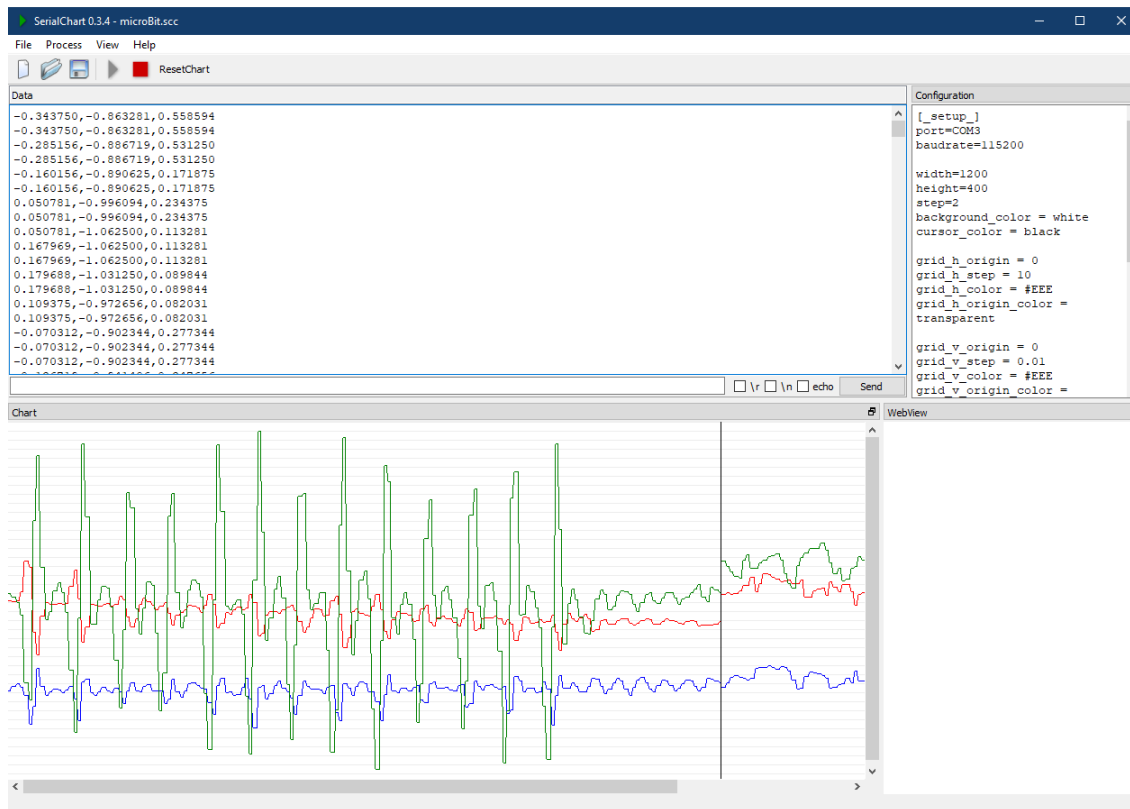
Jednou z možností je zobrazení pomocí nástroje, zvaného Serial plotter. Jedná se o aplikaci, která vykresluje grafy na základě dat, které přijímá sériovou linkou ve formátu ASCII. Data pro jednotlivé osy jsou oddělena mezerou a za každou sekvencí je znak `$`. Nevýhodou tohoto programu je, že se musí manuálně nastavovat parametry grafu při každém spuštění a z neznámého důvodu nerozlišuje kladné a záporné hodnoty. Jediným přípustným typem dat je *integer*. V příloženém obrázku je vidět kyvadlový pohyb kitu BBC micro:bit zavěšeného na přívodním USB kabelu.



Obrázek 2.13: Serial plotter

2.3.4 Zobrazení průběhů - Serial chart

Další možnost pro zobrazení je aplikace Serial chart. Zde se parametry grafu určují textovým souborem, který se dá opětovně nahrát a uložit. Tento program již podporuje vykreslování floatu a zároveň s vykreslováním zobrazuje i náhled komunikace, jak probíhá v ASCII.



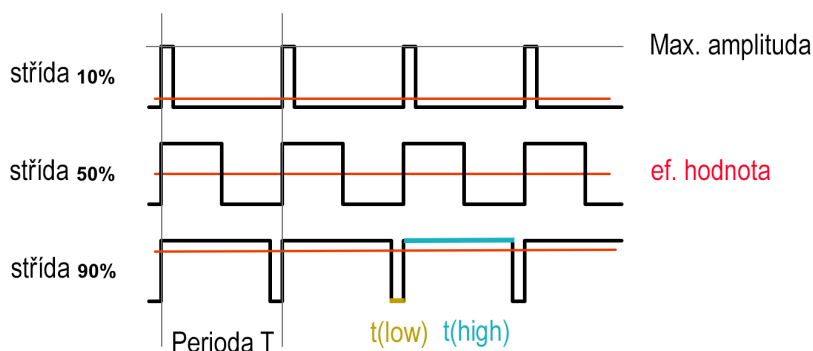
Obrázek 2.14: Serial chart

2.4 Konkrétní funkce měřicího systému

Cílem práce je vytvoření měřicího systému. Ten bude disponovat více funkcemi pro dané fyzikální experimenty. Funkce lze řídit prostřednictvím vlastní QT aplikace. V aplikaci bude každé funkci věnováno vlastní okno a budou na sobě nezávislé. Vzhledem k omezenému počtu výstupních pinů u BBC micro:bit nebude možné provozovat více těchto funkcí najednou. Pro změnu funkce bude třeba ukončit funkci předchozí. Alternativně lze funkce systému řídit z terminálového prostředí. Pro tuto možnost bude separátní verze firmwaru pro daná zařízení.

2.4.1 Funkce PWM generátoru

PWM neboli pulsně šířková modulace je užitečný nástroj. Jedná se o diskretní modulaci. Výstupem generátoru je dvouhodnotový (obdélníkový) signál. Amplituda je dána napájecím napětím procesoru. Volitelnými parametry jsou střída a perioda (frekvence).



Obrázek 2.15: Detail PWM signálu, parametry

Střídu D definujeme jako poměr trvání amplitudy t_{high} k době celé periody T . Matematicky vyjádřeno jako

$$D = \frac{t_{high}}{t_{high} + t_{low}} = \frac{t_{high}}{T}. \quad (1)$$

Takto definovaný signál lze využít dvěma způsoby. Při vhodné volbě střídy 50% do jisté míry může nahradit sinusový signál. Takovému signálu lze poté nastavit požadovanou periodu (frekvenci) a dále s ním pracovat. Možné použití si ukážeme dále.

Druhým způsobem využití tohoto signálu je náhrada funkce analogového výstupu (DA převodníku) procesoru. Tato skutečnost se dá demonstrovat při připojení běžné LED diody. S nastavením střídy pozorujeme změnu jasu LED diody. Pokud bychom chtěli napětí co nejvíce podobné konstantnímu, museli bychom výstup doplnit o potřebný filtr.

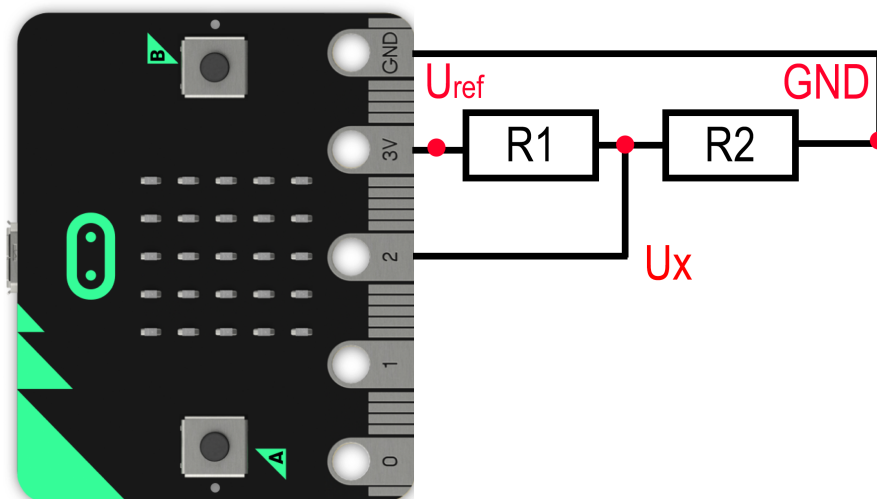
PWM generátor lze využít ve spoustě úloh. Zajímavá je například úloha na test citlivosti lidského ucha. Pro tuto úlohu budeme potřebovat náš měřicí systém a reproduktor. Vstup reproduktoru připojíme na výstupní pin PWM. Poté nastavujeme parametry PWM. Střída v této úloze se doporučuje 50%. Postupně zvyšujeme frekvenci generovaného signálu. Sledujeme, že se charakter reprodukováného zvuku mění. Na této úloze si můžeme ověřit jak vysoké frekvence je lidské ucho schopné rozpoznat.

Pokud budeme mít PWM signál generovaný z více kanálů zároveň, můžeme demonstrovat interferenci signálů. Jedná se o skládání dvou signálů o blízkých frekvencích. Při připojení dvou signálů o různých frekvencích na reproduktor uslyšíme zajímavý zvukový projev.

2.4.2 Funkce Voltmetr

Funkce voltmetru je jednoduchá. Má za cíl určit velikost neznámého napětí a tu poté prezentovat uživateli. Velikost napětí určujeme v daném bodě vzhledem k určenému bodu. Typicky vzhledem k zemi (GND).

Jednou z úloh měření napětí může být ověření vztahu pro výpočet napětí na odporovém děliči. Obvod zapojíme dle obrázku.



Obrázek 2.16: BBC micro:bit zapojení odporového děliče

Pokud použijeme známé hodnoty rezistorů R_1 a R_2 , U_{ref} jako referenční napětí z procesoru, typicky 3,3V nebo 5V, můžeme dopočítat napětí U_x dle vzorce

$$U_x = U_{ref} \times \frac{R_2}{R_1 + R_2}. \quad (2)$$

Pokud použijeme místo jednoho z rezistorů potenciometr, můžeme tímto způsobem generovat stálé napětí, jehož hodnotu můžeme sledovat právě funkcí voltmetru.

Dale lze využít odporový dělič pro zjištění neznámé hodnoty odporu, například rezistoru R_2 . Výpočet tohoto odporu provedeme pomocí vzorce

$$R_2 = \frac{U_x \times R_1}{U_{ref} - U_x}. \quad (3)$$

2.4.3 Funkce Softwarový čítač

Zařízení označované jako čítač slouží k analýze nestatických signálů. V případě neperiodických signálů může počítat náběžné nebo sestupné hrany. U periodického signálu navíc analyzuje frekvenci, periodu. Některá zařízení disponují i funkcí pro určení střídy obdélníkového signálu, ale primárně to není účel čítače.

Čítače pro periodické signály dělíme do dvou skupin podle rozsahů frekvencí, které jsou schopné identifikovat. Měření nízkých frekvencí, řádově do 10 kHz se realizuje pomocí měření doby periody, ze které se poté vypočte frekvence. Eventuálně lze změřit čas trvání více period a poté výsledný čas vydělit počtem period. Takto se realizuje průměrování.

Pro vysoké frekvence, řádově větší než 10 kHz pak měření probíhá konstantní dobu, po kterou čítač počítá náběžné hrany. Poté se provede přepočítání náběžných hran na jednu sekundu, čímž se určí frekvence signálu. Průměrování je tedy v tomto způsobu obsaženo.

2.4.4 Funkce Primitivní osciloskop

Osciloskop je sofistikované komplexní zařízení. Využívá rychlé paralelní AD převodníky a používá se k vykreslování grafických průběhů signálů. Pokud se zaměříme na digitální osciloscipy, je pro nás klíčové znát vzorkovací rychlost. Ta se udává ve vzorcích (samplech) za sekundu. Průměrný digitální osciloskop vzorkuje rychlostí 200 MSa/sec a více. V případě jednoduchého osciloskopu realizovaného mikroprocesorovým kitem s jedním převodníkem jsme limitováni na desítky kSa/s . V mém měřícím systému bude funkce osciloskopu pouze ilustrativní. Neočekávám realizaci pokročilých funkcí jako například trigger. Spíše půjde o "rychlejší voltmetr s krátkou pamětí".

2.4.5 Grafický zobrazovač pro hodnoty okamžitého zrychlení

BBC micro:bit disponuje akcelerometrem MMA8652. Toho plánuji využít a zhotovit nástroj pro vykreslování hodnot z tohoto senzoru do grafu. Tato funkce by měla být nejvíce hravá. Na průbězích bude možné demonstrovat základní principy mechanického kmitání.

3 Vývojové prostředí ARM mbed

ARM mbed je online vývojové prostředí, které využijí pro programování obou modulů. Mbed využívá programovací jazyk C++. Oproti klasickým prostředím pro programování mikroprocesorů nabízí velice zjednodušené způsoby pro inicializaci funkcí, vstupů a výstupů pinů a jejich následné řízení. Není třeba aktivovat brány ani GPIO parametry. Mbed sám o sobě podporuje softwarové funkce jako ticker, timer atd. Je zde také dostupné velké množství knihoven třetích stran pro práci se specifickým hardwarem. Například pro řízení externích senzorů, jako je v případě BBC micro:bitu akcelerometr MMA8652. Pro STM32F042F6P6 využijí například knihovnu USBSerial, pro komunikaci s PC.

Samotná relace v prostředí začíná registrací uživatele. Po registraci si uživatel zvolí, jaký hardware chce programovat. V současné době je podporováno asi 140 zařízení. Psaní programů pro odlišný hardware je pak totožné, liší se v pojmenování vstupů a výstupů daného hardware. Tato pojmenování jsou přímo v mbed dokumentaci pro daná zařízení uvedena. Další odlišnost může nastat v případě, že využíváme nějaké periferie specifické pouze pro daný kit. Pokud bychom měli nějakou jednoduchou úlohu, například blikání LED na kitu, bude tento program shodný pro většinu zařízení. Jedinou odlišností pak bude pojmenování použitých pinů. Pro názornost přikládám ukázkou takového programu pro STM32F042.

```
#include "mbed.h"

DigitalIn myButton(PB_1);
DigitalOut myLed(PA_5);

int main(){
    while(1){
        if(myButton == 0){
            myLed = !myLed;
            wait_ms(200);
        }
    }
}
```

Algoritmus 1: Kód blikacího programu na STM32F042

Prostředí zvládne program sestavit i zkompilovat. Výstupem takového procesu je stažení souboru, zpravidla .hex nebo .bin. Ten pak po připojení kitu k PC přepokopujeme do složky zařízení, obdobně jako na flash disk. Po nahrání souboru do zařízení se program automaticky spustí.

3.1 Alternativní označení pinů BBC micro:bit v ARM mbed

Kit BBC micro:bit disponuje pěti hlavními piny hardwarově identifikovatelnými podle zdířky na banánkový konektor. Ty jsou popsány jako 0, 1, 2, 3V, GND. GND je pin uzemnění. Pin označený jako 3V je fixní napětí VDD, jmenovitě 3.3V. Napětí na tomto pinu však kolísá s různým proudovým odběrem viz. *Nepřesnost měření napětí*. Piny 0, 1, 2 jsou pak vstupně výstupní piny pro použití v praktických úlohách. Mbed tyto piny označuje jako P0, P1, P2. Toto označení odpovídá číslování na kitu. Je zde ale i alternativní značení a piny lze inicializovat i jako PAD1, PAD2, PAD3. Toto označení je matoucí, protože číslování nekoresponduje s čísly pinů na kitu, jeho funkčnost je ale stejná jako u předchozí varianty.

4 Generování impulzních signálů pomocí PWM

První funkcí, kterou jsem řešil je Generátor impulzních signálů. Vzhledem k tomu, že pro vývoj programu využívám ARM mbed, kde se stejný program pro jiný hardware liší pouze v označení vstupních a výstupních pinů, budu ve všech realizacích primárně uvažovat BBC micro:bit. Případné modifikace nebo záležitosti ohledně kitu s procesorem STM32F042 pak doplním.

Jak již bylo řečeno ve druhé kapitole, využití generátoru pulzních signálů je široké. Vize programu je taková, že bude uživatel moci nastavovat střihu a frekvenci PWM generátorů, pokud možno na více kanálech zároveň. U této funkce je přijatelné využití terminálového uživatelského rozhraní. Obsluha bude tedy dostupná jak v terminálové formě, tak pod QT.

4.1 Softwarová realizace obdélníkového signálu

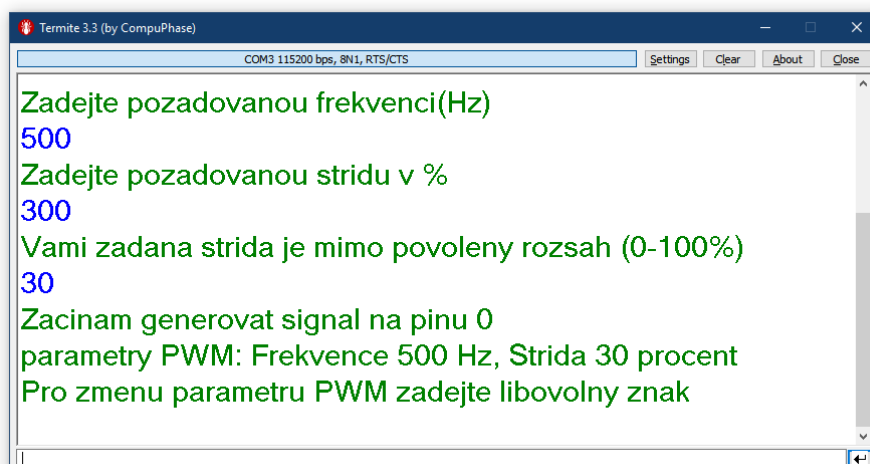
U této úlohy jsem se zprvu potýkal s nedostatečnou znalostí mbed prostředí a dokumentací kitu BBC micro:bit. Zde se totiž funkce PWM nikde neuváděla. Pokusil jsem se tedy realizovat PWM pomocí přepínání *DigitalOut(PAD0)* a systémového čekání *wait_ms()*. Tato realizace naráží na fakt, že při použití nekonečného cyklu procesor není schopen udržet stejnou frekvenci, výstup tedy nebyl pravidelný a docházelo k chybě ve střídě i periodě. Taková realizace je pro jakékoli účely nevhodná.

4.2 Použití PWMout v ARM mbed

V kitu je použit procesor Nordic nRF51822, ten však už ve svém 'skromném reference manuálu' funkci PWM uvádí. Realizace v mbed už tedy nebyla problémem. PWM lze použít na všech třech PAD pinech jednoduchou inicializací (viz. Algoritmus 2). Výpis z terminálové verze programu přikládám níže.

```
PwmOut pwm(PAD3);  
pwm.period_us(1000000/freq);  
pwm.write((float)strida/100);
```

Algoritmus 2: Inicializace PWM pro BBC micro:bit



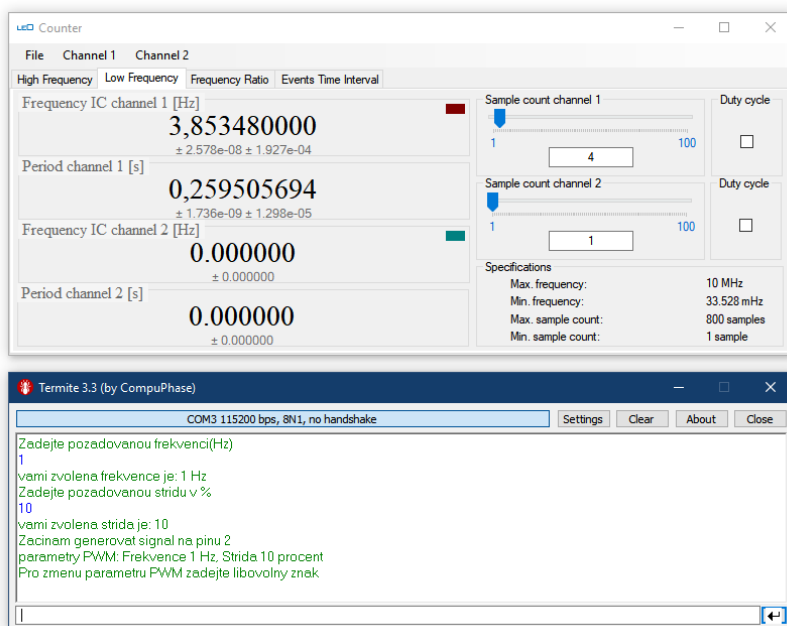
Obrázek 4.1: Ukázka výpisu programu PWM pro BBC micro:bit

4.3 Vícekanálové generování PWM signálů o různých frekvencích v BBC micro:bit

BBC micro:bit se ukázal jako nevhodný pro PWM výstup na dvou a více pinech zároveň. V okamžiku nastavení parametru druhého PWM kanálu dojde k nechtěné změně vlastností prvního. Zpravidla nastaví stejnou frekvenci, jako nová PWM a střída poklesne. Tento jev je dán tím, že kit podporuje pouze jeden výstup PWM, který však lze směřovat na více konkrétních pinů. V Terminálové realizaci proto použijí pouze jeden výstup a v aplikaci pod QT na to upozorním.

4.4 Měření parametrů impulzů čítačem LEO pro kit STM32F303 Nucleo

Pro tuto část úlohy jsem využil kit STM32F303 Nucleo a program Leo Counter ve verzi 2.23 vyvinutého na katedře měření. Program disponuje měřením frekvence a střídou pro nízkofrekvenční signály a měřením vysokých frekvencí s průměrováním. V režimu nízké frekvence je možno nastavit počet odebraných vzorků - period, pro stanovení frekvence. Při vysokofrekvenčním měření se pak nastavuje čas, po který se budou odebírat vzorky a teprve poté probíhá výpočet. Tímto nástrojem jsem analyzoval vlastnosti PWM výstupů.



Obrázek 4.2: Leo Counter v režimu čítače nízkých frekvencí

4.5 Parametry PWM pro BBC micro:bit

Analyzoval jsem signály o frekvencích v rozsahu 1 Hz až 50 kHz. Pro každou frekvenci jsem analyzoval chování při změně střídy viz. tabulka. Analýzou jsem zjistil, že nejmenší frekvence, kterou je micro:bit schopen vysílat je 3.85 Hz a to i při volbě frekvence nižší. Limit shora pak je 39 kHz. Čím nižší používáme frekvenci, tím přesněji se nastaví střída. Pro vysoké frekvence, řádově od 1 kHz už je chyba v desítkách Hz. Pro použití v experimentu citlivosti lidského ucha nebude mít vliv na výsledek, ale při nějaké specifické aplikaci už by mohla způsobit problémy. Zda je PWM definována softwarově nebo hardwarově se nepodařilo zjistit. Vzhledem k velikosti chyb při vyšších frekvencích se dá soudit, že se jedná o SW řešení.

f [Hz]	změřená f[Hz] pro střídu: 10		30		50		70		90	
1	3,853	10,001	3,854	30,001	3,854	50,001	3,853	70	3,853	90
5	5,051	10,003	5,051	30,002	5,051	50,002	5,05	70	5,05	90
10	10,101	10,005	10,101	30,004	10,101	50,002	10,101	70,002	10,101	90
20	20,201	10,011	20,202	30,008	20,202	50,006	20,202	70,003	20,203	90,001
50	50,492	10,026	50,494	30,02	50,496	50,015	50,498	70,009	50,499	90,003
100	100,95	10,052	100,96	30,041	100,97	50,029	100,97	70,017	100,96	90,006
500	503,65	10,26	503,6	30,2	503,6	50,14	503,6	70,09	503,6	90,03
1000	1004,3	10,52	1004,3	30,4	1004,3	50,29	1004,4	70,17	1004,3	90,06
3000	2990	11,2	2990	30,1	2990	50,3	2990	70,4	2990	89,34
5000	4907	12,5	4907	31,9	4907	51,4	4907	70,8	4907	90,3
8000	7779	13,7	7779	32,2	7779	50,7	7778	69,2	7789	87,6
10000	9543	13,1	9543	32	9542	50,9	9542	69,7	9542	88,6
20000	18750	18,3	18750	33	18751	55	18752	70	18751	85
30000	x	x	26673	36	26673	58	26671	68	26673	89
50000	x	x	39042	38	39042	54	39042	69	39042	84
100000					39042	53				

Obrázek 4.3: Tabulka - pwm analýza - BBC Micro:bit

4.6 Odlišnosti pro STM32F042

Realizace programu generátoru je shodná s tou pro BBC micro:bit. Rozdíl je v tom, že STM32F042 disponuje více kanály pro výstup PWM. Konkrétně v pouzdře TSSOP20 je možné použít až 3 PWM současně. Je tedy možné na více kanálech generovat impulzní signály o rozdílných střídách i frekvencích aniž by docházelo k paralelním ovlivněním.

Pro PWM zde využívám piny PA_4 a PA_6 . Stejným měřením jako v předchozí části jsem určoval chyby impulzního generátoru. Při generování signálů o frekvencích v jednotkách kHz jsem zjistil chyby v řádu jednotek Hz . S rostoucí požadovanou frekvencí roste i chyba. Nejvyšší frekvenci, kterou jsem byl schopen pomocí Leo Counteru zaznamenat bylo až 500 kHz . Zde se chyba pohybovala ve stovkách Hz . Do budoucna bych rád přeměřil parametry tohoto generátoru laboratorním čítačem a provedl podrobnější analýzu.

5 Měření napětí tříkanálovým voltmetrem

Pro měření statického napětí je vhodné použít vnitřní A/D převodník procesoru. V obou použitých zařízeních se jedná o převodníky s postupnou aproximací. Pro tyto převodníky je důležité referenční napětí, podle kterého pak určují napětí neznámé. S tím je spojen problém při stabilizaci napětí v BBC micro:bit. Tímto omezením se zabývám v části (5.4). V této úloze opět uvažuji terminálovou variantu i pod QT, neboť se jedná o statické měření, se kterým terminál nebude mít problém.

5.1 Použití analogového vstupu v ARM mbed

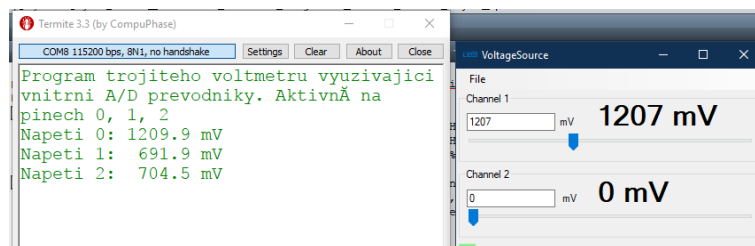
Funkce *analogIn* umožňuje převést vstupní napětí pinu na float $\langle 0;1 \rangle$. Pro BBC micro:bit je měřitelný rozsah $0\text{ mV} - 3300\text{ mV}$. Chceme-li vypsát hodnotu napětí v milivoltech, vynásobíme vstupní hodnotu 3300 krát. Napětí chceme měřit tříkanálově. V kitu je však pouze jeden AD převodník. Pro měření napětí na více vstupech zároveň tak dochází k přepínání mezi jednotlivými piny. O to se však v mbedu starat nemusíme. Inicializujeme tři analogové vstupy a dále s nimi zacházíme dle potřeby.

```

AnalogIn voltage0 (PAD1);
AnalogIn voltage1 (PAD2);
AnalogIn voltage2 (PAD3);

```

Algoritmus 3: Inicializace AnalogIn pro BBC micro:bit



Obrázek 5.1: ukázka výstupu Voltmeter na terminál

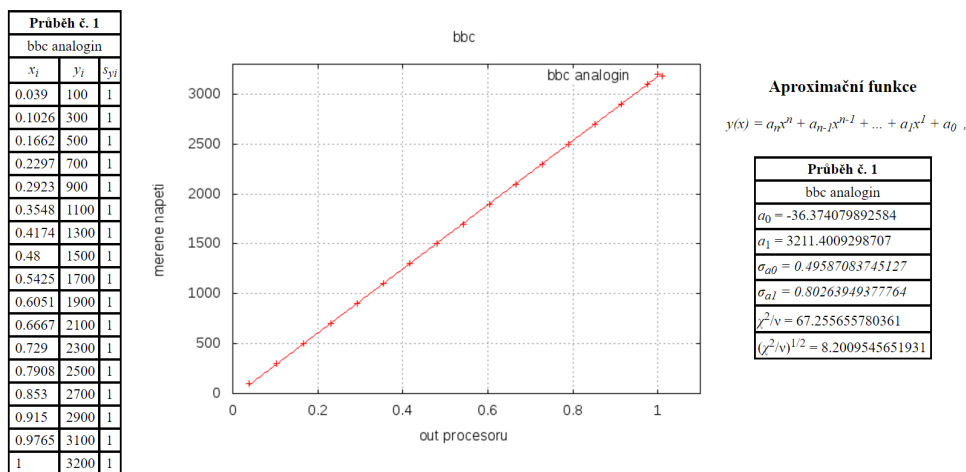
5.2 Měření napětí na volném pinu BBC micro:bit

Jak je vidět na přiloženém obrázku z provozu, na pinech PAD1, PAD2 čte program napětí cca 700mV i když k nim reálně není žádný zdroj napětí připojen. Ten je pouze na PAD0. Jedná se o napětí "ve vzduchu" a není třeba se jej obávat.

5.3 Charakteristika A/D převodu v BBC micro:bit

Pro ověření převodní charakteristiky jsem použil generátor napětí na kitu STM32F303 (Little Embedded Oscilloscope), podobně jako v předchozí kapitole režim counter. Hodnotu *AnalogIn* ze vstupu BBC micro:bitu jsem poté přímo vypisoval na terminál. Naměřené a referenční hodnoty jsem aproximoval přímkou a tak určil přibližnou převodní charakteristiku převodníku. Konstanta a_0 značí offset vůči referenčnímu napětí BBC micro:bitu a a_1 je pak směrnice aproximační přímky. V ideálním případě by bylo $a_0 = 0$ a $a_1 = 3300$.

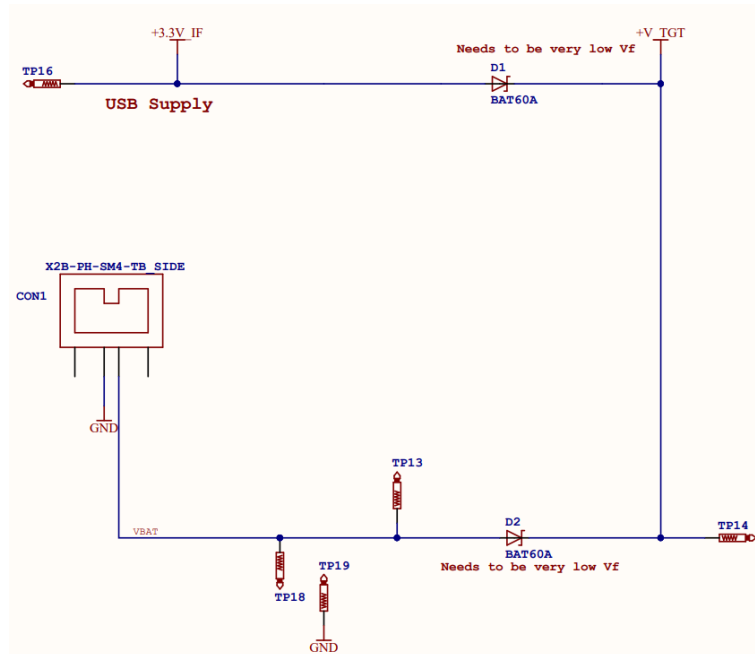
$$a_0 = -36.374, \quad a_1 = 3211.1 \tag{4}$$



Obrázek 5.2: Aproximace AnalogIn do napětí BBC micro:bit

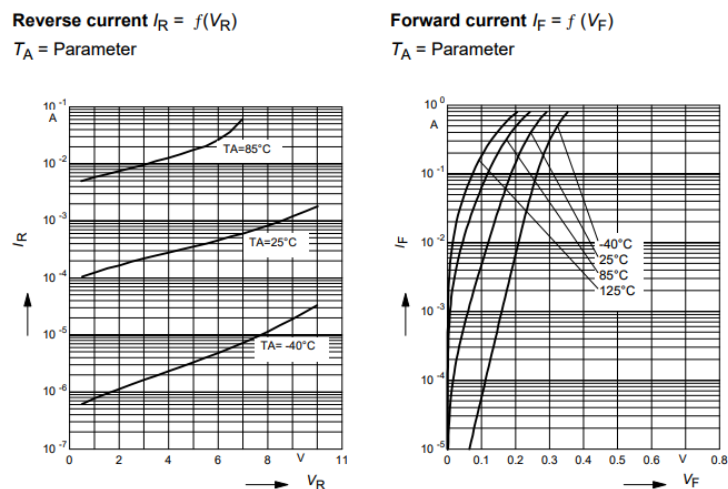
5.4 Chyby měření napětí v BBC micro:bit

V laboratoři jsem s použitím přesného zdroje a multimetru zjistil, že chyba při měření napětí se v některých případech pohybuje v řádu desítek milivoltů. Referenční napětí pro AD převodník je odebíráno přímo z napájení procesoru. Není zde napěťový stabilizátor.



Obrázek 5.3: Schéma zapojení zdroje napájení

V napájecím obvodu je v sérii i ochranná Schottkyho dioda BAT60A, která má primárně zabránit přelévání energie mezi zdrojem z USB a případné externí baterie. Schottky dioda využívá spojení polovodič - kov. V tomto případě je použita kvůli nízkému úbytku napětí v propustném směru. Ten však činí až 0.3 V. Dioda má však také svoji voltampérovou charakteristiku a v důsledku nekonstantního proudového odběru procesoru nebo proměnlivé okolní teplotě tak napájecí napětí, tedy i reference pro AD kolísá. Výpočet měřeného napětí ovšem počítá s konstantním referenčním $U_{ref} = 3.3V$ a dochází tedy k chybě. Přikládám VA charakteristiku z datasheetu BAT60A.



Obrázek 5.4: VA charakteristika BAT60A

5.5 Odlišnosti pro STM32F042

Ve druhém kitu využívám pinů PA_4 , PA_5 , PA_6 . O chybě měření se zde dá říci, že roste s velikostí napětí. Bohužel závislost chyby není lineární a nedá se kompenzovat. Příčinu vzniku této chyby se mi do termínu odevzdání práce nepodařilo odhalit.

6 Měření hodnot okamžitého zrychlení

Tato úloha se opírá o využití magnetometru MMA8652, který je k dispozici pouze v BBC micro:bitu. Odpadá tedy část s STM32F042, kde by realizace byla možná pouze v případě rozšíření modulu právě o tento senzor.

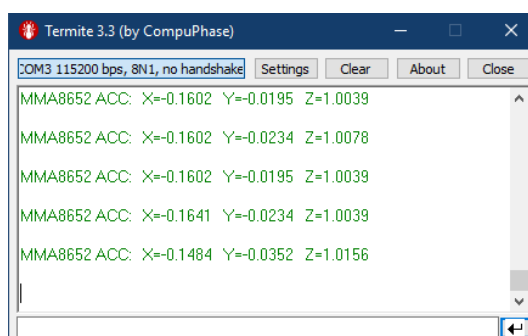
6.1 Použití akcelerometru MMA8652

Kit BBC Microbit v sobě obsahuje mimo jiné i AD převodník - akcelerometr MMA8652. V mbedu existuje open source knihovna z MIT právě pro tento převodník. K inicializaci akcelerometru je třeba přiřadit piny pro I2C komunikaci. V případě Microbitu jsou to P19 a P20.

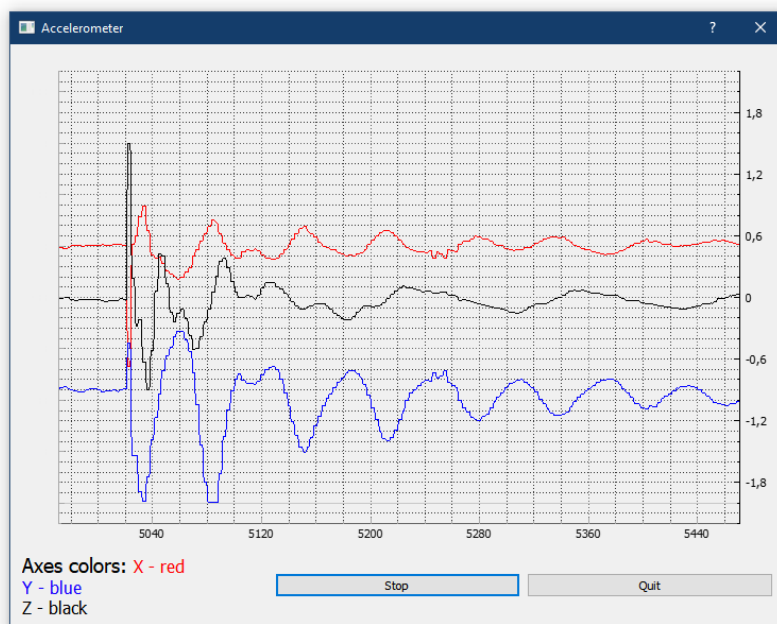
```
MMA8652 acc(P20, P19);  
float acc_data [3];  
acc.ReadXYZ(acc_data);
```

Algoritmus 4: Inicializace akcelerometru pro BBC micro:bit

Výstupem akcelerometru (*acc_datta*) jsou hodnoty okamžitého zrychlení v osách: x, y, x. Rozsah každé z os je $\pm 2g$. Data se dají vyčítat rovnou jako float přepočtené vzhledem k tíhovému zrychlení v ms^{-2} nebo jako int_{16t}. Data se pak dají vypisovat kvazistaticky na terminál. To však vzhledem k tomu, že chceme sledovat průběhy není vhodné. Proto využiji k vykreslování dat vlastní zobrazovač pod QT, kterému se pak budu věnovat v kapitole o QT aplikaci. Při drobné modifikaci programu lze samozřejmě využít jeden z již existujících zobrazovačů, o kterých se zmiňuji v rozboru.



Obrázek 6.1: Výpis z akcelerometru na terminál



Obrázek 6.2: Výpis z akcelerometru v aplikaci pod QT

7 Měření nízkých frekvencí softwarovým čítačem

Pro začátek je nutné zvolit způsob, jakým budeme frekvenci měřit. Já jsem si zvolil metodu, při které se po určité době počítají náběžné hrany na vstupním pinu v digitálním režimu. Protože jsem časový úsek pro měření zvolil jednu sekundu, odpovídá počet náběžných hran frekvenci daného signálu. Rozlišení takové metody měření je potom 1 Hz . Předpokládá se, že analyzujeme periodický signál rozumných vlastností, tím se myslí například sinusovka, pila nebo obdélníkový signál. Funkce čítače bude podporována pro BBC micro:bit i kit s STM32F042. Samozřejmostí je verze pro terminálové uživatelské rozhraní i pro aplikaci pod QT.

7.1 Použití funkce InterruptIn v ARM mbed

V úloze s čítačem je zapotřebí detekovat náběžné hrany zkoumaného signálu. Číst opakovaně ze vstupu *DigitalIn* není vhodné, protože by mezi jednotlivými čteními docházelo k prodlevě. Během té doby by tak náběžná hrana nemusela být detekována. V Mbedu pro detekci náběžných hran existuje funkce *InterruptIn*. Ta se používá pro sledování náběžných hran na daném pinu. S detekcí náběžné hrany pak volá funkci zvolenou při inicializaci.

```
InterruptIn event(PAD1);
event.rise(&countEdge);
```

Algoritmus 5: Použití InterruptIn BBC micro:bit

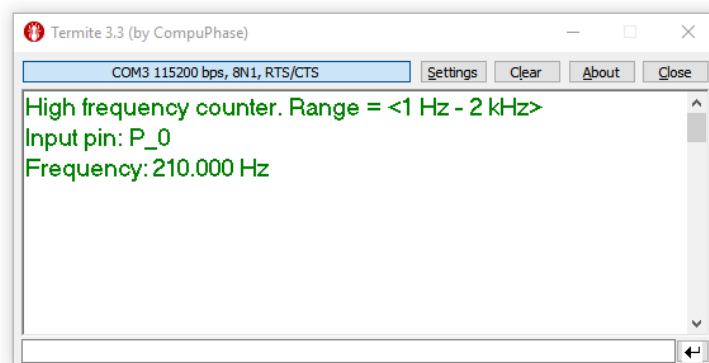
7.2 Použití funkce Ticker v ARM mbed

Objekt *Ticker* slouží k volání funkcí programu nezávisle na hlavní smyčce. Jak je vidět v příložené části kódu, jako parametry mu přiřadíme funkci, která se má provést a čas, ve kterém k tomu má dojít v sekundách. Pro konkrétní případ čítače volím časový interval

jednu sekundu a funkci pro vyhodnocení počtu náběžných hran a odeslání údaje o frekvenci uartem do PC.

```
Ticker interval;
interval.attach(&getFrequencyHigh, 1);
```

Algoritmus 6: Použití Tickeru BBC micro:bit



Obrázek 7.1: Ukázka čítače pro terminál - BBC micro:bit

7.3 Kompenzace chyby čítače v BBC micro:bit

Při analýze signálů o známých frekvencích jsem zjistil chybu v měření frekvence. Při referenčním signálu o frekvenci 1 kHz čítač měřil 915 Hz . Pro 2 kHz to pak bylo 1830 Hz atd. Jedná se o lineární chybu. Ta je zapříčiněna nepřesností Tickeru. Místo toho, aby intervaly odpovídaly jedné sekundě, dochází ke ztrátě 85 ms . Měření tedy probíhá 915 ms . Tento problém se dá kompenzovat při následném výpočtu frekvence. Místo jedné sekundy budeme brát za interval 915 ms a výpočet korigujeme. V ideálním případě by se tedy frekvence např. 1 kHz určila podle tohoto vztahu.

$$f = \frac{\text{Rising edges}}{\text{Interval}} = \frac{1000}{1} [\text{Hz}] = 1000\text{ Hz}. \quad (5)$$

V našem případě, kdy chceme správně určit frekvenci a známe chybu, upravíme hodnotu *Interval* a výpočet proběhne korektně.

$$f = \frac{\text{Rising edges}}{\text{Interval}} = \frac{915}{0.915} = 1000\text{ Hz}. \quad (6)$$

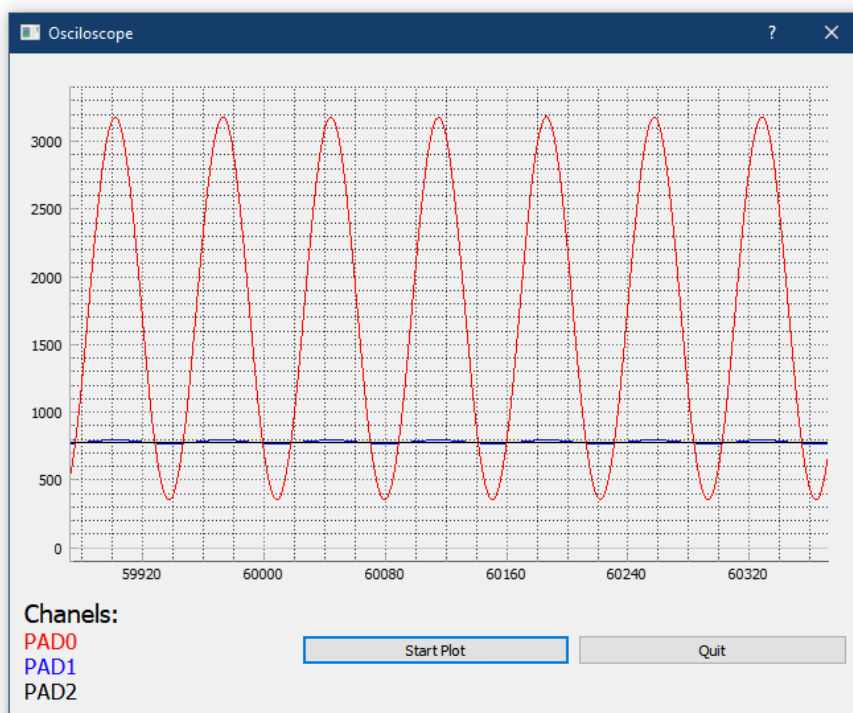
7.4 Odlišnosti pro STM32F042

Modul s mikropocesorem STM32F042F6P6 v této úloze příjemně překvapil. Pro vstup jsem využil pin *PA_6*. Problém s chybou intervalu se zde vyskytl taktéž. Zde však prodleva činila $1,6\text{ ms}$. Po kompenzaci je kit schopen měřit frekvence až do 70 kHz . Chyba se po celou dobu testování pohybovala v řádu jednotek *Hz*. Při nízkých frekvencích, typicky do 2 kHz jsem byl schopen určit frekvenci s chybou maximálně 1 Hz .

8 Měření proměnlivých napětí

Pro měření proměnlivých hodnot napětí nemá smysl užívat terminálového uživatelského rozhraní. Tato část bude tedy realizována pouze pro aplikaci pod QT. V případě obou modulů se bude jednat o cyklické měření napětí na třech pinech (kanálech). Naměřené hodnoty se pak instantně přes UART odešlou do vykreslovače průběhů pod QT. Nutno podotknout, že vykreslování v QT se mi podařilo realizovat pouze 'realtime', tudíž nástroj slouží spíše jako demonstrační, pro ukázkou změn napětí nebo zobrazení signálů o frekvencích v řádu desítek Hz. Nedokáže tak plně nahradit funkci osciloskopu. Rozsah velikostí hodnot napětí je shodný s voltmetrem, tudíž 0 mV až 3300 mV .

Vlastní realizace programu je velmi podobná voltmetru, jen s tím rozdílem, že data jsou odebírána bez prodlevy a okamžitě posílána do PC, kde jsou následně zpracována. Pro tento účel si vystačíme s nekonečným cyklem a funkcí *AnalogIn*. Následuje ukáзка zobrazení signálu sinusovky o frekvenci 15 Hz prostřednictvím příslušné funkce v mé aplikaci pod QT.



Obrázek 8.1: Ukáзка vykreslení proměnného napětí - BBC micro:bit

9 Aplikace měřicího systému

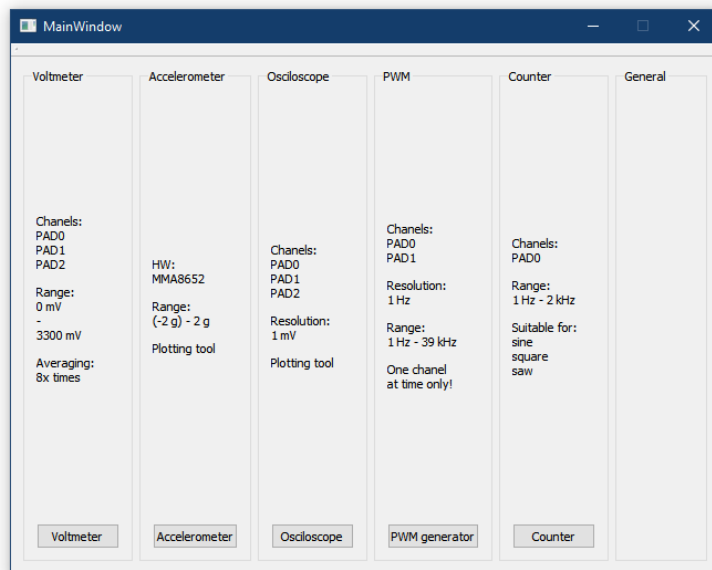
Součástí SDI laboratorního zařízení je i uživatelský interface. V mém konkrétním případě se jedná o aplikaci, vytřenou pod QT. QT je přívětivý nástroj na tvorbu aplikací pro PC. Kombinuje grafické uživatelské rozhraní pro tvorbu konkrétních oken a programování jednotlivých funkcí aplikace v jazyce C++. Já konkrétně vytvářím aplikaci pro Windows, ale QT umožňuje sestavit aplikaci i pro operační systémy na bázi Linuxu.

Aplikaci tvoří hlavní okno a podokna pro jednotlivé funkce. Primární účel této aplikace je ovládání naprogramovaných funkcí modulů (BBC micro:bit i STM32F042) uživatelsky přívětivou formou. QT s kitem spolupracuje prostřednictvím sériové komunikace na virtual COM portu počítače. Výhodou je, že v QT lze spojení s kitem navázat, aniž by uživatel věděl, na kterém COM portu zařízení hledat. Při navazování komunikace aplikace prohledá dostupné porty a připojí pouze ten správný. Ten identifikuje pomocí vendor ID, což je unikátní číslo pro dané zařízení. Například BBC micro:bit má VID 3368. Při procházení dostupných zařízení se tedy vybere to správné. Problém by nastal v okamžiku, když by uživatel měl připojené dva kity BBC micro:bit. V takovém případě by se aplikace spojila vždy s tím kitem, který by byl na nižším COM portu.

9.1 Hlavního okno

Hlavní okno má převážně informativní funkci. Jsou zde uvedeny jednotlivé funkce měřicího systému spolu s jejich popisy. Odtud si také uživatel volí, kterou funkci chce právě využít. Funkci aktivuje kliknutím na příslušné tlačítko. Poté už se zobrazí okno pro konkrétní funkci.

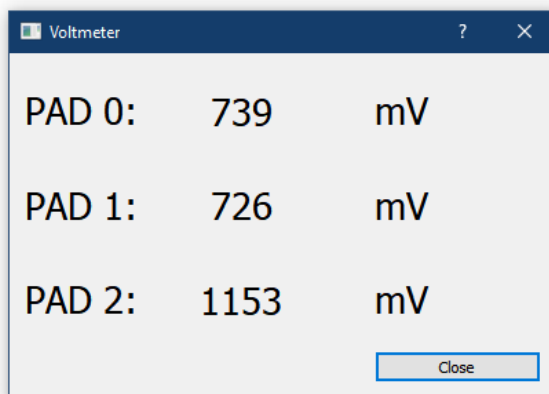
Nabízí se zde možnost navázat spojení s kitem. Ta je však realizována v každém podokně zvlášť. Tuto variantu jsem zvolil z jediného důvodu. Jako jedno ze zařízení používám BBC micro:bit, ve kterém lze bez externího konektoru využít pouze 3 piny pro vstup a výstup. Mbed má takovou vlastnost, že pokud se libovolný pin inicializuje, nelze jeho funkci dále měnit. Vzhledem k tomu, že potřebuji pro jednotlivé laboratorní funkce piny v různých režimech (*DigitalIn*, *DigitalOut*, *AnalogIn*, *Pwmout*, *InterruptIn*...), nemám dostatek pinů pro všechny funkce. Je tedy nutné při změně funkce aplikace nějakým způsobem přenastavit pin. To řeším tak, že spojení QT a kitu navážu až při spuštění konkrétní funkce. Sdělím kitu, jakou funkci chci použít a podle toho se vstupy, popřípadě výstupy inicializují. Při ukončení dané laboratorní funkce vyšlu do kitu povel k restartování. Tím dojde k uvolnění použitých pinů a ukončení komunikace. Kit po restartu čeká, než se z hlavního okna aplikace spustí jiná funkce, která s ním opět naváže spojení.



Obrázek 9.1: Hlavní okno v QT aplikaci

9.2 Okno Voltmetru

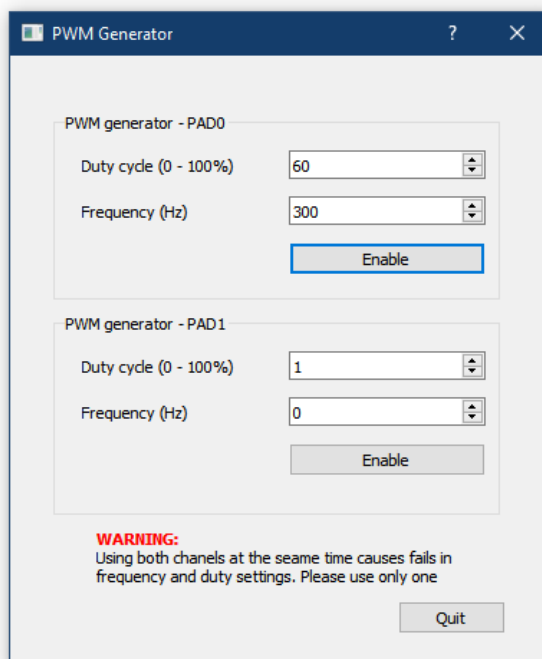
Při spuštění podokna voltmetru se nastaví připojení s kitem. Poté aplikace vyšle do kitu povel pro spuštění funkce voltmetru. Kit se nastaví do požadované funkce a začne vysílat naměřené hodnoty napětí ze tří kanálů do aplikace. Zde se provede separace dat na tři hodnoty, pro každý kanál jednu. Hodnoty se poté zobrazí v okně. Funkce voltmetru se pak ukončí stiskem tlačítka Close.



Obrázek 9.2: okno voltmetru v QT aplikaci

9.3 Okno pro PWM generátor

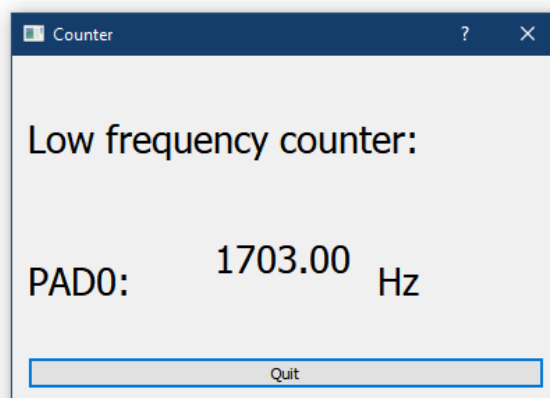
Inicializace funkce PWM je stejná jako u voltmetru, naváže se komunikace, určí se funkce. Poté co uživatel nastaví parametry frekvence a střídy stiskne tlačítko *Enable*. Aplikace vyšle do kitu instrukce pro nastavení výstupu. Pro změnu parametrů stačí přepsat hodnoty a opět odeslat.



Obrázek 9.3: okno PWM v QT aplikaci

9.4 Okno pro softwarový čítač

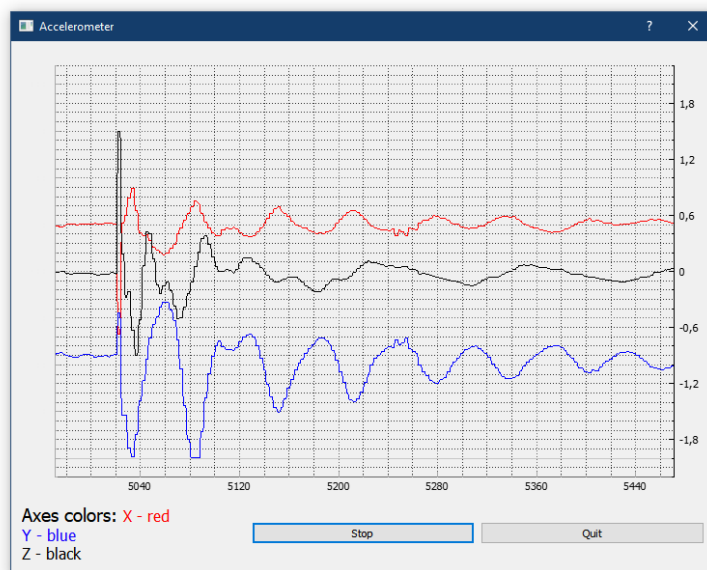
Okno pro čítač frekvencí funguje totožně jako okno voltmetru. Jediný rozdíl je, že z kitu přichází pouze jedna hodnota, není tedy nutné data separovat a může se rovnou vypsát údaj frekvence.



Obrázek 9.4: okno čítače v QT aplikaci

9.5 Nástroj pro grafické vykreslení okamžitého zrychlení

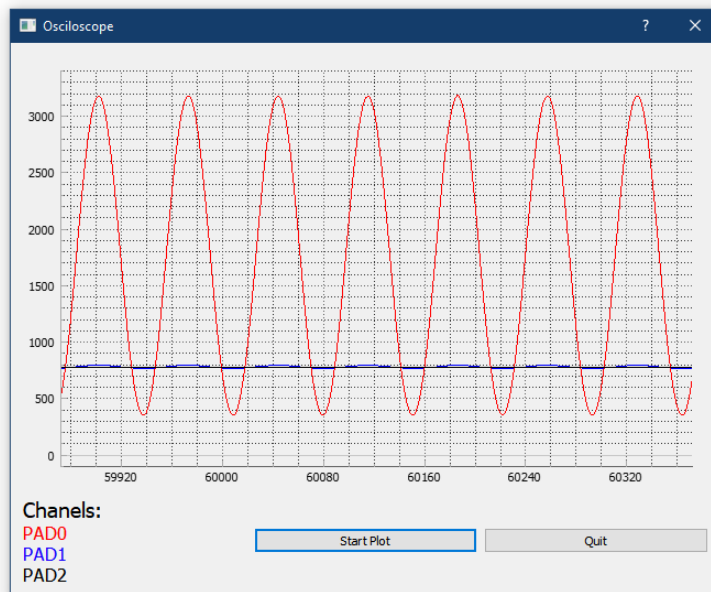
Tato část je z celé QT aplikace nejzajímavější. Jedná se o okno, v němž lze v reálném čase s minimálním spožděním vykreslovat průběhy hodnot okamžitého zrychlení z akcelerometru na BBC micro:bitu. Pro tyto účely využívám knihovnu *QCustomPlot*, která je vhodná pro vykreslování grafů. Tento konkrétní graf zobrazuje 3 hodnoty okamžitého zrychlení. Pro přehlednost se hodnoty z každé osy akcelerometru odlišují barvou. Graf se s přibývajícimi daty posouvá a zachovává záznam z předchozích 5 sekund. Horizontální osa udává čas v *ms* a na vertikální ose je pak velikost zrychlení, přepočtení na jednotky *g*. Je zde také tlačítko *Stop* pro okamžité zastavení průběhu. Po jeho opětovném stisknutí pokračuje měření.



Obrázek 9.5: okno Zobrazovače hodnot zrychlení v QT aplikaci

9.6 Okno pro grafické vykreslení proměnného napětí

Toto okno funguje na stejném principu jako předchozí. Rozdíl je ve volbě kratší časové základny. Vertikální osa udává hodnotu napětí v mV . Na přiloženém obrázku je záznam sinusového signálu o frekvenci 15 Hz.



Obrázek 9.6: okno zobrazovače průběhů napětí v QT aplikaci

10 Závěr

V rámci mé bakalářské práce se podařilo splnit všechny body zadání. Hlavním cílem bylo vytvořit funkční měřicí systém pro realizaci fyzikálních a elektronických pokusů. K tomu bylo potřeba sepsat programy pro realizaci funkcí pro moduly BBC micro:bit a STM32F042. Tyto programy dělím do dvou kategorií. Samostatně soběstačné programy pro komunikaci s terminálovým emulátorem a programy konfigurovatelné skrze aplikaci pod QT. Tato řídicí aplikace paradoxně zabrala nejvíce času, je však vhodná pro jednoduchou a intuitivní obsluhu systému.

Při programování hardwarových kitů se mi podařilo ukázat hlavní přednosti prostředí ARM mbed. Tou nejzásadnější je konverze programu mezi zařízeními. Na jednotlivých funkcích jsem však odhalil i neduhy tohoto typu programování. Jedním z problémů bylo, že většina funkcí v ARM mbed je definována softwarově a na různých zařízeních dochází k drobným odchylkám, například při využití softwarového tickeru ve funkci frekvenčního čítače. Další překážkou byla nemožnost změny funkce pinu za běhu programu. Toto omezení jsem řešil softwarovým restartem modulu s následným navedením do jiné části programu z řídicí aplikace. Za zajímavé považuji i simulování převodníku UART - USB pro modul s STM32F042, což značně snížilo nároky na propojení s nadřazeným PC. Toto usnadnění bylo však vykoupeno zabráním velké části paměti flash modulu. Zde se také ukázalo, do jaké míry jsou jednotlivé kity schopné provádět potřebné funkce. Ukázalo se, že modul s STM32F042 je oproti BBC micro:bitu mnohem více vhodný pro realizaci měřících funkcí. U modulu BBC micro:bit bych ocenil detailnější dokumentaci celého kitu. Většinu potřebných informací a parametrů jsem musel zjišťovat experimentálně.

Dalším stěžejním bodem bylo vytvoření intuitivní aplikace pod QT. Zde jsem se potýkal s opětovným navazováním komunikace po restartu kitu. Hodně času jsem věnoval nástroji pro grafické vykreslování průběhů hodnot okamžitého zrychlení akcelerometru. Tato funkce je na celém systému asi ta nejzajímavější. Akcelerometr MMA8652 je velice citlivé zařízení s rychlou odezvou. Pro fyzikální pokusy v oblasti kinematiky je tedy naprosto ideální. Do budoucna by stálo za to, uvažovat o sestavení mechanických laboratorních modelů typu kyvadlo, pružina, atd.

Ve výsledku se podařilo sestavit komplexní aplikaci pro všechny potřebné funkce. Nutno podotknout, že se v aplikaci stále vyskytují jisté bugy, které je ještě potřeba eliminovat. Při rozumném zacházení je však aplikace použitelná. Podotýkám, že i po ukončení této bakalářské práce se plánuji věnovat rozšiřování o nové funkce systému, zdokonalení stávajících funkcí a eliminaci bugů. Také plánuji vytvořit příručku pro programování takových funkcí pro začínající programátory v oblasti mikrokontrolérů.

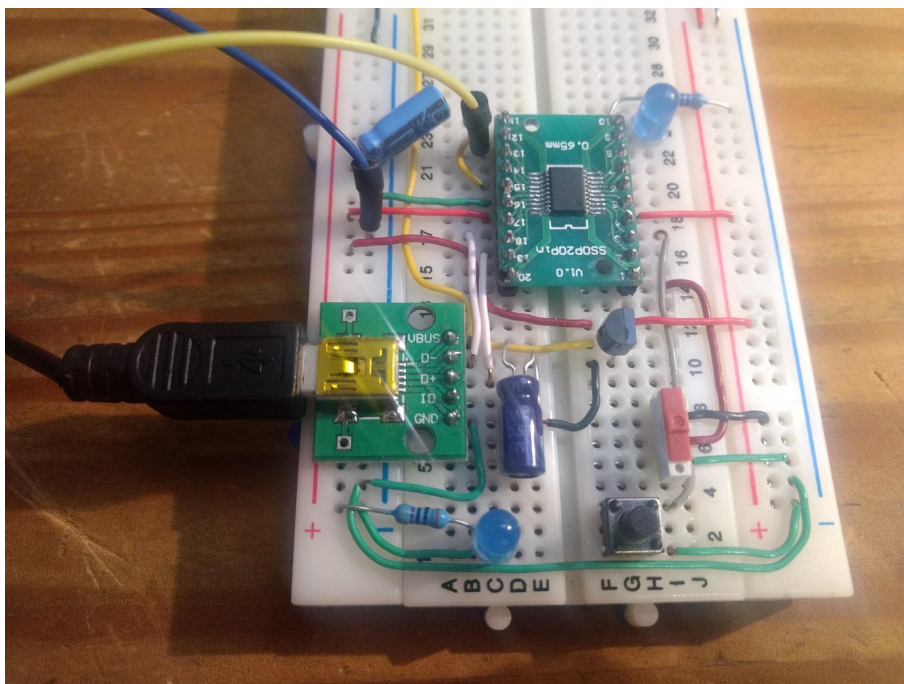
Reference

- [1] BBC. micro:bit, dokumentace o kitu (online)
<http://microbit.org/>
- [2] STMICROELECTRONICS. *STM32F042x Datasheet*
<http://www.st.com/>
- [3] FREESCALE. *NXP MMA8652FC Datasheet, Rev. 3.3, 10/2015*
<https://cache.freescale.com/>
- [4] NORDIC. nRF51822 Product Specification v3.1 2014
<http://http://infocenter.nordicsemi.com/>
- [5] STMICROELECTRONICS. RM0091, Reference manual STM32F0x1/ STM32F0x2/ STM32F0x8 advanced ARM®-based 32-bit MCUs)
<http://st.com/>
- [6] ARM, mbed. IDE with boards description and documentation (online)
<http://os.mbed.com/>
- [7] BALUTA, Sergiu. Serial chart - real time plotting tool, Oct 2016
<https://github.com/starlino/serialchart>
- [8] QCUSTOMPLOT. *Plotting library for QT Creator*
<http://www.qcustomplot.com/>

Příloha A - fotodokumentace použitého HW



Obrázek 10.1: Kompletní modul BBC micro:bit - přední strana



Obrázek 10.2: Kompletní modul s STM32F042F6P6

Příloha B - obsah přiloženého CD

```
CD
├── BBC microbit
│   ├── NRF51 MICROBIT terminalova app.cpp
│   ├── NRF51 MICROBIT terminalova app.hex
│   └── QT
│       ├── qt project.zip
│       ├── Firmware for QT NRF51 MICROBIT.cpp
│       ├── Firmware for QT NRF51 MICROBIT.hex
│       └── QT exe
│           ├── libgcc s dw2-1.dll
│           ├── libstdc++-6.dll
│           ├── libwinpthread-1.dll
│           ├── MicrobitApplication.exe
│           ├── Qt5Core.dll
│           ├── Qt5Gui.dll
│           ├── Qt5PrintSupport.dll
│           ├── Qt5PrintSupportd.dll
│           ├── Qt5SerialPort.dll
│           └── Qt5Widgets.dll
├── STM32F042
│   ├── STM32F042 Terminalova app.cpp
│   ├── STM32F042 Terminalova app.dfu
│   └── STM32F042 Terminalova app.bin
└── BP Kurka Petr 2018.pdf
```