

Bakalářská práce

Návrh a realizace základních přístrojových funkcí využitím mikrořadičů

Jakub Klíma



Květen 2017

Vedoucí práce: doc. Ing. Jan Fischer, CSc.

České vysoké učení technické v Praze
Fakulta elektrotechnická, Katedra měření



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jakub Klíma**

Studijní program: **Otevřená informatika**
Obor: **Počítačové systémy**

Název tématu česky: **Návrh a realizace základních přístrojových funkcí využitím mikrořadičů**

Název tématu anglicky: **Design and Realisation of Basic Instrumentation Functions Using STM32 Microcontrollers**

Pokyny pro vypracování:

Analyzujte možnosti realizace základních přístrojových funkcí v laboratorních přípravcích pro výukové účely s využitím mikrořadičů STM32F042, STM32L072 a STM32L432 a jejich periferií. Navrhněte a prakticky ověřte vzorové řešení přípravku (např. na nepájivém kontaktním poli nebo univerzální desce), který bude realizovat přístrojové funkce typu měření napětí, odporu, frekvence, periody a funkce zpracování signálů z jednoduchých senzorů, jako jsou např. fototranzistor, fotodioda, ultrazvukový snímač vzdálenosti a dalších, Vámi vybraných senzorů. Navrhněte a realizujte též způsoby generace signálů s využitím číslicových i analogových výstupů mikrořadičů. Ovládání přípravku bude realizováno lokálními ovládacími prvky (např. typu tlačítko nebo potenciometr) i pomocí nadřazeného PC s terminálovým programem s komunikací prostřednictvím UART a USB. Pro mikrořadiče vytvořte potřebné programové vybavení.

Seznam odborné literatury:

- [1] The Definitive Guide to ARM® Cortex®-M0 and Cortex-M0+ Processors
- [2] STMicroelectronics: RM0091: STM32F0x2 Reference manual
- [3] STMicroelectronics: DS10147 STM32F042 Data

Vedoucí bakalářské práce: doc. Ing. Jan Fischer, CSc.
Datum zadání bakalářské práce: 1. února 2017
Platnost zadání do¹: 30. září 2018


Prof. Ing. Jan Holub, Ph.D.
vedoucí katedry



V Praze dne 1. 2. 2017


Prof. Ing. Pavel Růžka, CSc.
děkan

¹ Platnost zadání je omezena na dobu tří následujících semestrů.

Poděkování

Rád bych poděkoval doc. Ing. Janu Fischerovi, CSc. za odborné vedení a konzultace k mé bakalářské práci. Dále bych rád poděkoval rodině a přátelům, kteří mě psychicky podporovali po celou dobu studia.

Prohlášení

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.....

Podpis autora práce.....

Abstrakt

Tato práce ukazuje možnosti realizace přístrojových funkcí s využitím mikrořadičů. Zaměřuje se zejména na měření napětí, odporu, frekvence a zpracování signálů z jednoduchých senzorů, jako jsou například fotodiody, fototranzistory nebo ultrazvukový snímač vzdálenosti.

Klíčová slova

mikrořadič, voltmeter, ohmmeter, generátor, napětí, frekvence, pwm

Abstrakt

This work explores possible realizations of instrumental function using microcontrollers. Main focus is measurement of voltage, resistance, frequencies and signal processing from simple sensors, such as photodiode, phototransistor or ultrasound range detector.

Keywords

microcontroler, voltmeter, ohmeter, generator, voltage, frequency, pwm

Obsah

1	Úvod	1
2	Rozbor zadání	2
3	Nástroje potřebné k vývoji	3
3.1	Mikrokontroléry	3
3.1.1	STM32F042K6	3
3.1.2	STM32L432KC	3
3.1.3	STM32L072KZT	3
3.2	Periferie	4
3.2.1	USB	4
3.2.2	UART	4
3.2.3	ADC	4
3.2.4	DAC	4
3.2.5	Timer	4
3.3	Senzory	5
3.3.1	Ultrazvukový snímač vzdálenosti	5
3.3.2	Fotodioda	5
3.3.3	Fototranzistor	5
3.4	Softwarové nástroje	6
3.4.1	CubeMX	6
3.4.2	Keil	6
3.4.3	DFuse	6
3.4.4	Putty	6
3.4.5	Mbed	6
4	Funkce využívající měření napětí	8
4.1	Princip	8
4.2	Implementace v Mbedu	8
4.2.1	Voltmeter	8
4.2.2	Ohmeter	9
4.3	Implementace pomocí Cube a Keil	9
4.3.1	Voltmeter	10
4.3.2	Ohmeter	10
4.3.3	Generace napětí	11
5	Funkce využívající čítače	12
5.1	Princip	12
5.2	Implementace v Mbedu	13
5.2.1	Měření frekvence	13
5.2.2	Generování PWM	14
5.3	Implementace pomocí Cube a Keil	14
5.3.1	Měření frekvence	15
5.3.2	Generace pwm	16
6	Senzory	17
6.1	Implementace v Mbedu	17
6.1.1	Měření vzdálenosti	17

6.2	Implementace pomocí Cube a Keil	17
6.2.1	Měření vzdálenosti	17
6.2.2	Fotodioda	19
6.2.3	Fototranzistor	19
6.2.4	Optická závora	20
7	BBC Micro:Bit	21
8	Závěr	23
	Literatura	24

1 Úvod

Úkolem této práce je analyzovat možnosti realizace základních přístrojových funkcí v laboratorních přípravcích za použití mikrokontrolérů s pozdějším využitím při výuce. V dnešní době jsou mikrokontroléry na takové úrovni, že je možné uvažovat o nahrazení klasických laboratorních přístrojů. Tato práce si klade za cíl některé takové přístroje realizovat a ukázat postup při tvorbě obdobných programů. Především půjde o funkce měřící, jako je měření napětí, odporu, frekvence a generující, jako je generování pwm a určité napěťové úrovně. Dalším cílem této práce je ukázat zpracování signálů z jednoduchých senzorů, jako je ultrazvukový snímač vzdálenosti, fotodioda a fototranzistor.

2 Rozbor zadání

Funkce zmíněné v úvodu budou řešeny na třech mikrokontrolérech STM32F042, STM32L072 a STM32L432, za využití různých periférií, především AD a DA převodníku, timerů a komunikační periférie USART a USB. V části věnující se zpracování signálů budou využity jednoduché senzory konkrétně ultrazvukový snímač vzdálenosti HC-SR04, fotodioda BPW-34 a BPW-41N, fototranzistor PT204-6c. Při měření frekvence je důležité dát si pozor na nastavení hodin timeru, které musí mít 2x větší frekvenci, než je měřená frekvence. V opačném případě bude docházet k aliasingu, tedy k efektu, kdy je nemožné správně určit měřenou frekvenci. Z čehož plyne, že k dosažení co největší frekvence a přesnosti bude nutné použít hardwarové řešení při měření. V případě měření napětí je důležité nepřipojit na měřící pin napětí větší než je jeho napájecí napětí, které je obvykle 3.3V, protože pokud pin není 5V tolerantní tak by mohlo dojít k jeho zničení.

3 Nástroje potřebné k vývoji

Využívané nástroje lze rozdělit do dvou skupin, hardwarové komponenty a softwarové pomůcky. Do hardwarové skupiny jsou zařazeny samotné mikrokontroléry a jejich periferie a také použité senzory, do druhé skupiny patří vývojové prostředí a různé ostatní počítačové programy nezbytné pro vývoj a využívání vytvořených funkcí.

3.1 Mikrokontroléry

Cílem této práce je otestovat možnosti primárně tří konkrétních mikrokontrolérů a to STM32F042K6, STM32L432KC6 a STM32L072KZT. První dva zmíněné mikrokontroléry jsou na vývojovém kitu STM32 nucleo-32, třetí mikrokontrolér je nutno připájet na pouzdro. K pájenému mikrokontroléru bylo nutné přidat blokovací kondenzátory a také propojit společné napájecí a zemní piny.

3.1.1 STM32F042K6

STM32F042 s jádrem Cortex-M0 s maximální frekvencí 48MHz je jeden z nejlevnějších mikrokontrolerů od firmy STMicroelectronics. Disponuje jedním 12-ti bitovým AD převodníkem s až 10 kanály, jedním 32ti bitovým timerem a dalšími 16ti bitovými timery s jednotkami input capture a output compare, dvěma USART jednotkami a jedním USB, ale bohužel není vybaven DA převodníkem [1].

3.1.2 STM32L432KC

STM32L432KC jehož jádro Cortex-M4F s rychlostí až 80MHz činí z těchto tří mikrokontrolerů jednoznačně nejlepší a také nejdražší kus. Disponuje také jedním 32ti bitovým timerem a dále s 16ti bitovými timery, jedním 12ti bitovým AD převodníkem a dvěma 12ti bitovými DA převodníky. Pro komunikaci jsou k dispozici tři USART jednotky a jeden USB modul[2].

3.1.3 STM32L072KZT

STM32L072KZT, s jádrem Cortex-M0+ o maximální rychlosti 32MHz, je také poměrně levný mikrokontroler který na rozdíl od STM32F042 disponuje i DA převodníkem. Má přítomné dokonce dva 12ti bitové DA převodníky, na druhou stranu ale nemá 32ti bitový timer a má pouze 16ti bitové. Dalé disponuje 12ti bitovým AD převodníkem s až 16ti kanály, jedním USB a 4x USART jednotkou[3].

3.2 Periferie

Z periferií jsou využity jednotky USART a USB na komunikaci s nadřazeným PC a v případě STM32L072 je využito USB k bootování pomocí bootloaderu a PC programu DFuse. Jako další z periferií, která je využita, je převodník z analogového na číslicové zobrazení, neboli ADC, k měření velikosti napětí. Na opačném principu, tedy k generování určitého napětí, je určen převodník z digitálního na analogové zobrazení, neboli DAC. K časování a přesnému vysílání či měření signálů využívám čítačové jednotky takzvaných Timerů.

3.2.1 USB

V případě, že je během inicializace mikrokontroleru připojeno USB, bude se komunikovat přes USB a nikoliv přes UART. Při komunikaci přes USB je nutné nainstalovat na ovládacím PC ovladač virtual com portu. V případě, že procesor při inicializaci nenalezne USB připojení, bude preferovat UART.

3.2.2 UART

Ke komunikaci s PC je použit blok usart. Periferie usart je použita v asynchroním módu s baud rychlostí 9600, 8 datovými bity a jedním stop bitem bez parity. Na NUCLEO kitech je využit vnitřní převodník UART-USB, který je připojen k jednotce USART2 a je tedy možné komunikovat přes připojení k ST-linku, z toho plyne, že se nemusí připojovat další kabel. V případě použití čistého mikrokontroleru je zapotřebí převodník napěťových úrovní a využít RS-232 nebo komunikovat přes USB.

3.2.3 ADC

AD převodník je určen k měření napětí na pinu a převodu této hodnoty na hodnotu reprezentovatelnou v binární soustavě. Každý ze třech testovaných mikrokontrolerů měl jiné možnosti nastavení AD převodníku, hlavně různé velikosti obslužných hodin a různé velikosti doby vzorkování.

3.2.4 DAC

DA převodník používám ke generování stálé napěťové úrovně v rozsahu 0 - 3,3V. DA převodník je 12-ti bitový, může se tedy do DA převodníku zadat celé číslo v rozsahu 0-4095, což koresponduje s napětím na výstupu v rozsahu 0-3,3.

3.2.5 Timer

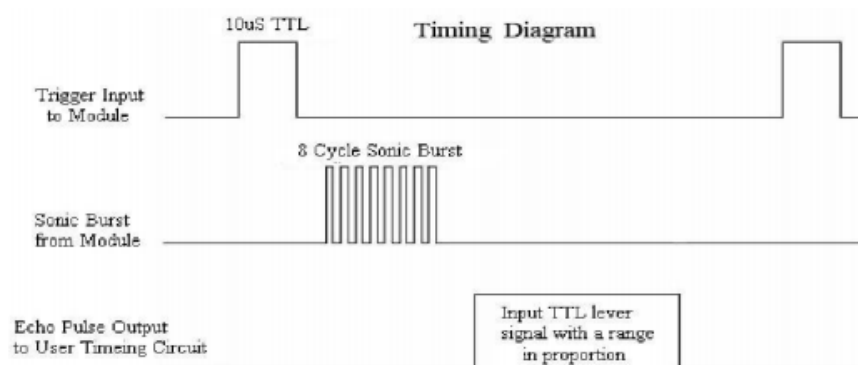
Čítače používám v různých módech. Pro generování PWM output mód a one pulse mod. PWM output mód generuje signál s definovanou periodou a délkou pulsu, jinak řečeno střídou. One pulse mód vyšle jen jeden puls s nastavitelnou délkou a možností posílat pulsy opakovaně, čímž se stane velmi podobný PWM output módu, zde ale budou z pravidla vysílány pulsy o znatelně menší frekvenci. V input módu používám input capture jednotku pro počítání periody a délky pulsu vstupujícího signálu. Output compare jednotku používám jako časovací jednotku k jiné periférii případně jinému timeru

3.3 Senzory

K sběru dat je využito pár základních senzorů. Prvním je ultrazvukový snímač vzdálenosti HC-SR04, který měří vzdálenost na principu odrazu ultrazvukového signálu. Dalšími jsou senzory reagující na světlo, konkrétně fotodioda BPW-34 detekující infračervené záření a i viditelné spektrum, fototranzistor PT204-6c, který je citlivý na infračervené záření i na záření z viditelného spektra a k optické závoře je využita fotodioda BPW-41N detekující jen infračervené záření a infračervená led dioda.

3.3.1 Ultrazvukový snímač vzdálenosti

Ultrazvukový snímač vzdálenosti HC-SR04 je poměrně levný senzor pro měření vzdálenosti. Funguje na principu odrazu, přičemž má jeden ultrazvukový vysílač a jeden přijímač. Vysílač vyšle signál a po odrazu ho zachytí přijímač, z doby mezi těmito dvěma jevy se vypočte vzdálenost objektu, od kterého se signál odrazil. Aby se předešlo rušení, neposílá se jeden puls, ale různé sekvence. V případě HC-SR04 se vyšle 8 pulzů o frekvenci 40kHz. Zapojení je jednoduché, senzor má 4 piny, Gnd, Vcc, Echo a Trig. Na pin Trig se přivede 10us puls, který odstartuje vyslání ultrazvukového signálu a na pinu echo se v zápětí vygeneruje puls o délce přímo úměrné vzdálenosti objektu. Na následujícím obrázku (obrázek 1) je vidět znázornění tohoto popisu.



Obrázek 1 Časový diagram signálů HC-SR04 [4].

Vzdálenost se tedy vypočítá jako $\frac{D \cdot c}{2}$, kde D je délka pulsu a c rychlost zvuku. Po jednoduchých úpravách se dostaneme ke tvaru $\frac{D[\mu s]}{58} = x[cm]$, kde x je výsledná vzdálenost.

3.3.2 Fotodioda

Fotodioda je dioda, které na PN přechod dopadá světlo, jestliže je PN přechod zastíněn chová se fotodiody jako normální dioda. Princip fotodiody je založen na dopadu světla na PN přechod, kde se po dopadu fotonu uvolní elektron a vznikne tak pár elektron - díra a tím se zvyšuje vodivost součástky. Fotodiody má dva vývody katodu a anodu.

3.3.3 Fototranzistor

Fototranzistor je tranzistor, kterému na PN přechod dopadá světlo stejně jako u fotodiody. Čím větší osvětlení fototranzistoru, tím více se otevře přechod mezi bází a emitorem a tím pádem protéká větší proud. Fototranzistor má na rozdíl od obvyčejného tranzistoru jen dva vývody emitor a kolektor.

3.4 Softwarové nástroje

K vývoji je využit CubeMX k nastavení periférií a vygenerování potřebného zdrojového kódu. Dále pak vývojové prostředí Keil k psaní kódu, kompilaci a debugu. Ke komunikaci je použit terminálový program Putty, ovladač virtual com port ke komunikaci přes USB a program DFuse k nahrávání firmwaru na mikrokontroler pomocí bootloaderu. V neposlední řadě je využita platforma Mbed.org k vývoji jednoduchých ukázkových funkcí.

3.4.1 CubeMX

CubeMx je program vyvinutý a distribuovaný od firmy STM, který slouží k jednoduššímu nastavení periférií a následně vygenerování šablony zdrojových kódů, na které uživatel staví své programy. Nejprve je nutné si vybrat, který mikrokontroler nebo vývojovou desku chcete použít a následně je možné nastavit jednotlivé periferie a hodiny dle daných možností.

3.4.2 Keil

Keil MDK je vývojový kit pro mikrokontrolery z rodiny ARM. Jeho součástí je vývojové prostředí μ Vision, debugger a ARM C/C++ compiler. Díky velmi rozsáhlému debuggeru a možnosti sledovat hodnotu registrů a paměti za běhu, se z Keilu stává velmi efektivní pomůcka při vývoji. Bohužel free verze tohoto programu omezuje velikost kódu na menší než 32KB, na tuto limitaci jsem během vývoje narazil a byl jsem nucen rozdělit některé funkce a udělat více samostatných aplikací.

3.4.3 DFuse

S DFuse a celým procesem bootování byl ze začátku velký problém, protože s STM32L072 nefunguje DFuse nejnovější verze a musí se používat starší.

3.4.4 Putty

Putty je terminálový program s možností komunikovat přes vybraný COM port. Možností nastavení Putty je celá řada, v této práci se používá základní nastavení s komunikačními parametry sériová linka COMx, kde číslo com portu se musí dohledat ve správci zařízení, s baudovou rychlostí 9600, ostatní nastavení nechávám výchozí. Pokud by uživatel chtěl změnit velikost písma a fontu, může tak učinit v záložce Window - Appearance. Změna barvy písma a pozadí se provádí v záložce Window - Colours. Uživatel musí vybrat Default foreground a změnit na požadovanou barvu k volbě barvy písma a analogicky Default background pro změnu pozadí.

3.4.5 Mbed

Jak již bylo zmíněno, Mbed[5] je internetové vývojové prostředí, které je určeno pro rychlé a snadné programování na mikrokontrolerech z rodiny ARM. Embed je tak vhodný pro začátečníky a také pro ty, kteří si chtějí naprogramovat nějaké jednoduché přístroje. Jeho velkou nevýhodou je nemožnost debugu, protože je vše uloženo v cloudu a uživatel má k dispozici pouze binární soubor k nahrání na mikrokontrolér. V případě, že program nefunguje tak jak uživatel zamýšlel, je poměrně těžké najít chybu.

K používání EMbedu je nutná registrace a přihlášení. Všechny programy, které uživatel

3 Nástroje potřebné k vývoji

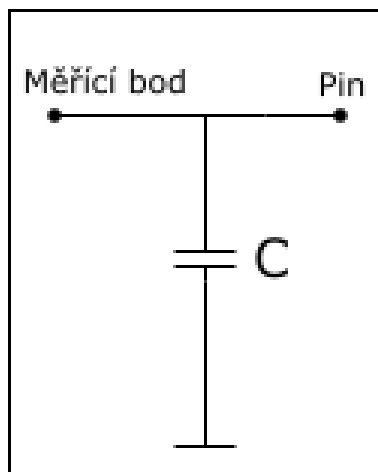
vytvoří, jsou přístupné po přihlášení. Při tvorbě nového programu musí uživatel zvolit na jaký mikrokontroler je program určen. Po vybrání cíle je možnost využití z několika vytvořených demo programů nebo si vytvořit svůj vlastní. Výhoda vytvořených demo programů je hlavně pro začátečníky, jako seznámení se s embedem. Demo programy zpravidla bývají blikání LED, generování pwm a komunikace přes UART případně USB (virtual com port).

4 Funkce využívající měření napětí

V této kapitole jsou podrobně popsány funkce, které se zakládají na měření nebo generování napětí. Měření napětí se využívá u více funkcí, přesnou hodnotu napětí je důležité znát například pro funkci voltmetru a nebo ohmetru. Měření napětí se využívá i u senzorů, ale ty jsou v kapitole samostatně.

4.1 Princip

U měření napětí, hlavně ve funkci ohmeter, může docházet k problému, při měření velmi malých napětí, což může nastat, právě pokud jsou na vstupu zapojeny velké dopory. Řešení této situace jsou možné dvě, první je softwarově zvětšit dobu odebrání vzorku a druhý je na vstup zapojit paralelně kondenzátor, který bude fungovat, jako pomocný zdroj napětí během odměru. V AD převodníku bývá kondenzátor okolo 8pF, je tedy potřeba na vstup připojit několikrát větší kondenzátor. Možné zapojení je znázorněno na obrázku níže.



Obrázek 2 Schéma měření napětí s blokovacím kondenzátorem.

4.2 Implementace v Mbedu

Při měření napětí v Mbedu se neřeší nastavení ADC a používá se pouze interface AnalogIn, který pomocí funkce read() vrátí hodnotu v rozmezí 0 - 1 jako poměr z rozsahu napětí na pinu. V případě potřeby rychlého odměru je možné použít FastAnalogIn, který využívá nízkou dobu vzorkování a kontinuální mód.

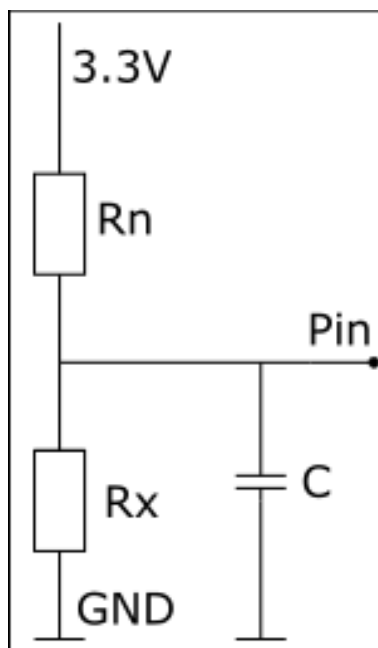
4.2.1 Voltmeter

K měření napětí na vstupním pinu se používá interface AnalogIn, kde se zadefinuje jaký pin se bude používat a pomocí funkce read, která vrací hodnotu od 0 do 1, se zjistí hod-

nota napětí na pinu. Výpočet se provádí dle vzorce $U=3.3*x$, kde x je změřená hodnota a 3.3 je maximální hodnota napětí. Poté odeslání výsledné hodnoty přes sériovou linku.

4.2.2 Ohmeter

Napětí se měří stejným způsobem jako u Voltmetru. AnalogIn používá ADC s krátkým samplovacím časem. To může způsobit špatné měření v případě velkých odporů na vstupu, proto je tento čas zvětšený zápisem přímo do registrů. V případě převedení na jiný mikrokontroler, budou ale registry jiné, což znamená, že by se tento zápis musel upravit, případně smazat. Napětí se měří na děličce napětí s horním odporem referenčním.



Obrázek 3 Schéma měření napětí s blokovacím kondenzátorem.

Ze vzorce

$$U_x = U \frac{R_x}{R_n + R_x} \quad (1)$$

se jednoduchými úpravami dostaneme k vzorci pro výpočet spodního odporu

$$R_x = \frac{R_n * U_x}{U - U_x} \quad (2)$$

4.3 Implementace pomocí Cube a Keil

Při implementaci v Cube a keilu se pro měření napětí využívá AD převodník mnohem podrobněji, než je tomu u implementace v Mbedu. Napětí se měří na jednotlivých kanálech AD převodníku, jehož nastavení se liší na každém mikrokontroléru. Ale v rámci jednoho mikrokontroléru je použito pro všechny funkce stejné nastavení ADC.

Aby bylo možné měřit i velmi malé napětí, je použit co nejdelší samplovací čas. AD převodník je v režimu continuous. Při inicializaci se provede nejprve odměr na kanálu Vrefint, pomocí kterého se vypočítá skutečná hodnota Vdd a až poté se měří napětí

k jednotlivým funkcím. Výpočet V_{dd} pomocí změřené reference probíhá podle následujícího vzorce:

$$V_x = \frac{VREFIN_CAL * ADC_DATA}{VREFINT_DATA * 4095} \quad (3)$$

kde V_x je měřené napětí, VREFIN_CAL je kalibrační konstanta uložená v paměti mikrokontroleru, ADC_DATA jsou data změřená AD převodníkem, VREFINT je hodnota VREFINT změřená AD převodníkem na kanále VREFINT a hodnota 4095 reprezentuje maximální velikost 12ti bitového převodníku. Konkrétní adresy kalibračních konstant jsou uvedeny v tabulce.

Mikročítač	Adresa
STM32F042	0x1FFF F7BA - 0x1FFF F7BB
STM32L072	0x1FF8 0078 - 0x1FF8 0079
STM32L432	0x1FFF 75AA - 0x1FFF 75AB

Tabulka 1 Adresy kalibračních konstant[1, 3, 2]

Dále po inicializaci provede automatická kalibrace. Výpočet napětí se počítá průměrováním 1000 odměrů. Nejdříve se jednotlivé odměry ukládaly do paměti pomocí DMA, ale pro toto využití je to zbytečné a méně vhodné řešení, proto bylo rozhodnuto počítat napětí real-time. Po každém odměru se přičte hodnota do proměnné a po přičtení 1000 hodnoty se vypočítá průměr všech odměrů a odešle uživateli. Doba odběru se liší na každém mikrokontroleru.

Mikročítač	Hodiny	Vzorkovací čas[počet cyklů hodin]	Čas[μs]
STM32F042	6MHz	239.5	42
STM32L072	4MHz	160.5	28
STM32L432	6MHz	640.5	108

Tabulka 2 Nastavení hodin ADC a vzorkovacího času

V tabulce jsou uvedeny vzorkovací časy na každém mikrokontroleru, byly voleny co nejnižší a zároveň, aby se moc nelišili.

4.3.1 Voltmeter

Měření napětí probíhá vždy na dvou prvních kanálech AD převodníku. Výpočet napětí probíhá dle vzorce (3)

4.3.2 Ohmeter

Měření odporu probíhá měřením napětí na odporovém děliči s horním odporem referenčním. Na obrázku níže je vidět zapojení, kondenzátor C je zde pro to, aby sloužil jako pomocný zdroj napětí v případě, že měřené napětí nestihne nabít vnitřní kondenzátor u ADC. Schéma zapojení je znázorněno na obrázku číslo 3. Výpočet odporu probíhá dle vzorce $R_x = \frac{R_n * U_x}{U - U_x}$.

Velikosti odporů nejsou naprosto přesné, přesnost je mezi 0.5% až 20%, avšak nejčastější jsou rezistory s přesností 5%. Například můj referenční odpor 10k při změřením multimetrem odpovídal 9,87kohm, což se na výpočtu neznámého odporu projeví. Proto je u výpočtu nutné brát v potaz poměrně velkou odchylku.

4.3.3 Generace napětí

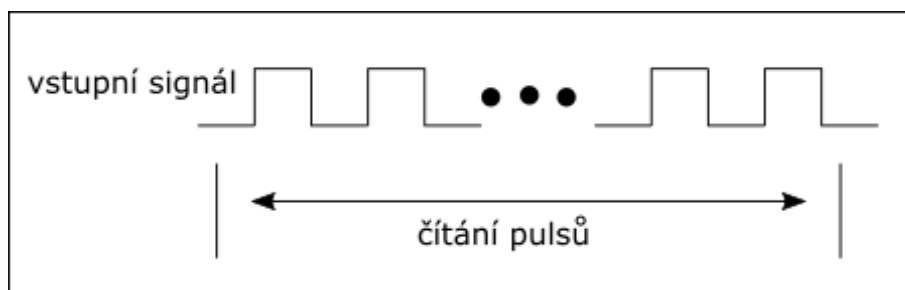
Výstupní napětí na pinu se nastaví pomocí DA převodníku. DA převodník není na mikrokontroleru STM32F042 přítomný, na ostatních dvou mikrokontrolerech je 12-ti bitový. Generované napětí se určí ze vzorce $x = V_{dd}/4095 * \text{hodnota v DA převodníku}$. Je možné generovat napětí v rozsahu 0-3.3 v závislosti na V_{dd} .

5 Funkce využívající čítače

V této kapitole jsou podrobně popsány funkce využívající primárně čítačové jednotky, jako jsou například generování pwm a nebo měření frekvence vstupního signálu. Mezi takové funkce patří měření frekvence a generování pwm.

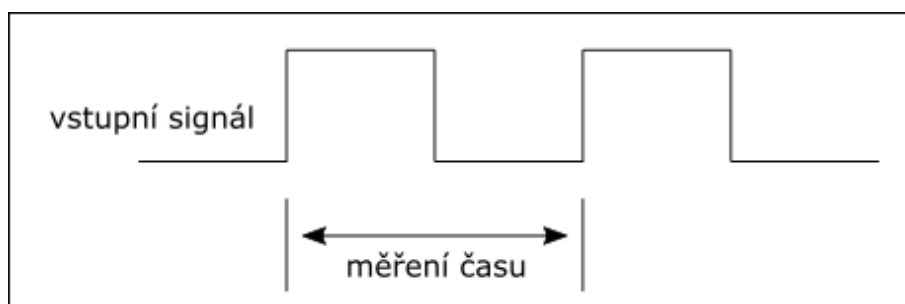
5.1 Princip

K měření frekvence využívám dvě metody, přímé měření a reciproční měření. Přímé měření znamená, že za určitý čas, typická je jedna vteřina, počítám pulsy signálu, což se rovná výsledné frekvenci. Reciproční měření je měření délky jednoho pulsu neboli



Obrázek 4 Princip měření přímou metodou.

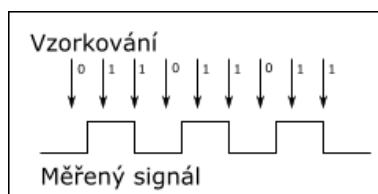
periody. K převedení na frekvenci stačí hodnotu periody převrátit, protože $f = \frac{1}{t}$. Přímé měření je vhodné pro větší frekvence, zatímco reciproční měření je lepší používat



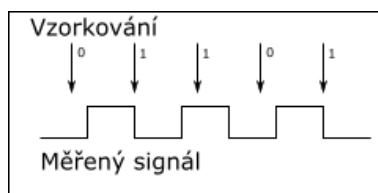
Obrázek 5 Princip měření reciproční metodou.

pro menší frekvence, tedy pro ty s delší periodou.

U měření frekvence je důležitá frekvence vzorkování, podle vzorkovacího teorému musí být měřená frekvence 2x menší než frekvence referenční. Pokud by měřená frekvence byla větší než je polovina referenční dojde k takzvanému aliasingu a měření signálu se bude zhoršovat.

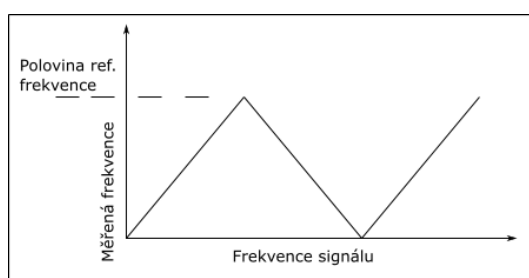


Obrázek 6 Vzorkování s dostatečně velkou frekvencí.



Obrázek 7 Vzorkování s nedostatečně velkou frekvencí.

Závislost měřené frekvence a frekvence referenční je vidět na obrázku 8. Pokud se měřená frekvence bude rovnat frekvenci vzorkovací bude v ideálním stavu měření ukazovat 0Hz, protože se bude jevit že je signál stále v jedné logické úrovni.



Obrázek 8 Grafické znázornění vzorkovacího teorému.

5.2 Implementace v Mbedu

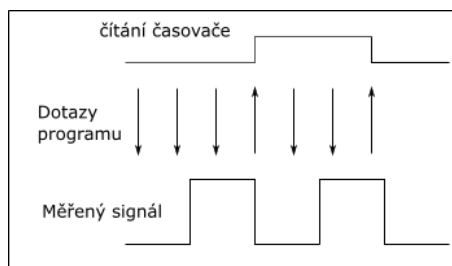
Při práci s timery v Mbedu se nepřistupuje přímo ke konkrétnímu timeru, ale využívá se interface PwmOut, Timer nebo Ticker. Pochopitelně tento způsob není tak přesný jako při práci přímo s timery.

5.2.1 Měření frekvence

Obě metody měření frekvence, tedy reciproční i přímou jsem implementoval v Mbedu softwarovým řešením, protože v embedu je tato metoda velmi názorná a lze jí snado pochopit i implementovat.

Reciproční metoda Reciproční metoda využívá interface DigitalIn, pomocí kterého zjišťuje logickou hodnotu vstupního pinu, a Timer pomocí kterého měří časový interval. Implementované řešení funguje tak, že se detekuje náběžná hrana, spustí timer a

při další náběžné hraně se timer vypne a odečte změřený čas. Detekce náběžné hrany probíhá ve while smyčce, kde se neustále dotazuje na hodnotu pinu dokud je 0. Jakmile je hodnota pinu 1, z while smyčky se vyskočí, což značí náběžnou hranu. Jakmile se zná délka periody, může se jednoduše vypočítat frekvence, podle vzorce $F = 1/t$. V případě měření více period za sebou se díky průměrování zpřesní měření, a proto program nejdříve změří jednu délku periody(t) a poté bude měřit $1/t$ period, což by mělo trvat přibližně 1 vteřinu, záleží jaká byla odchylka u prvního měření. Touto metodou mohu měřit frekvence s velmi velkou přesností. Nepřesnosti u měření jen jedné periody, vzhledem k využívání softwarového řešení, jsou s vyšší měřenou frekvencí čím dál větší, takovou situaci ilustruje obrázek níže(obrazeek 9). Kde je znázorněna situace, kde se časovač zapne až ve chvíli kdy končí logická úroveň 1 a měření je posunutě.



Obrázek 9 Dotazování programu při recipročním měření v Mbedu.

Přímá metoda Přímá metoda využívá také interface `DigitalIn` k čtení vstupu a interface `Ticker` k časování. `Ticker` opakovaně za daný časový interval vyvolá přerušeni a zavolá přiřazenou funkci. Nejprve byla implementována měření pomocí timeru, na principu stálého dotazování na uplynulý čas a čítání pulsů, ale při vyšších frekvencích již docházelo k aliasingu, protože dotazování na čas zbytečně zdržovalo program.

Při implementaci pomocí tickeru je možné neustále počítat pulzy a nemusí se dotazovat na čas, protože jakmile uplyne daný interval vyvolá se přerušeni. Díky tomuto způsobu je možnost měřit frekvence až do 175kHz a u vyšších frekvencích dochází opět k aliasingu.

5.2.2 Generování PWM

Ke generování PWM používám interface `PwmOut`, u kterého se při inicializaci zvolí na jakém pinu se bude PWM generovat a potom funkcemi `period`, `pulsewidth` a `write`, se můžou nastavovat jednotlivé atributy `pwm`. Funkce `period` nastaví délku periody, což je převrácená hodnota frekvence. `Pulsewidth` nastaví délku pulsu stejně jako funkce `write`. Rozdíl mezi těmito dvěma funkcemi je ten, že s funkcí `pulsewidth` je nutné zadat délku pulzu, ale u funkce `write` je nutné zadat hodnotu střídání v intervalu $\langle 0-1 \rangle$, střída je poměr mezi délkou pulsu a periodou.

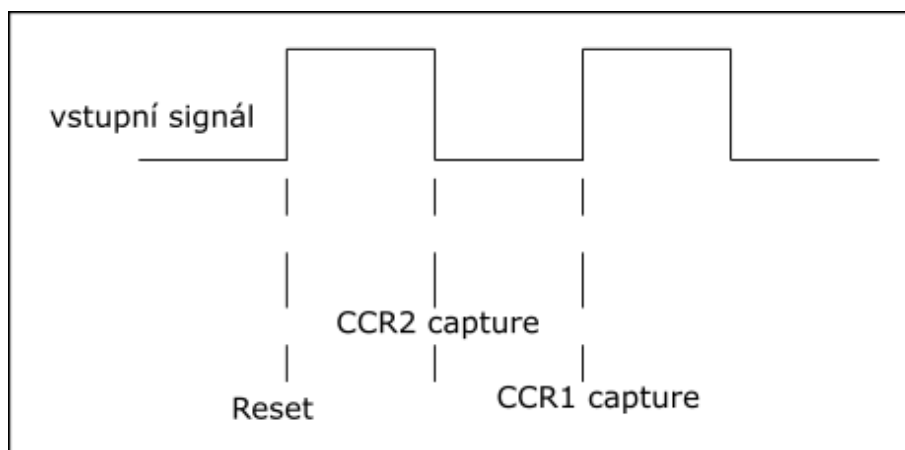
5.3 Implementace pomocí Cube a Keil

Při implementování těchto funkcí pomocí Cube a Keilu se pracuje přímo s timery a dosahuje se mnohem větších přesností. Je tedy možné měřit frekvence poloviční frekvenci hodit daného timeru, dle vzorkovacího teorému [6]. Při generování základního kódu pomocí Cube není samotné nastavení timerů nikterak složité a uživatel se velmi rychle zorientuje.

5.3.1 Měření frekvence

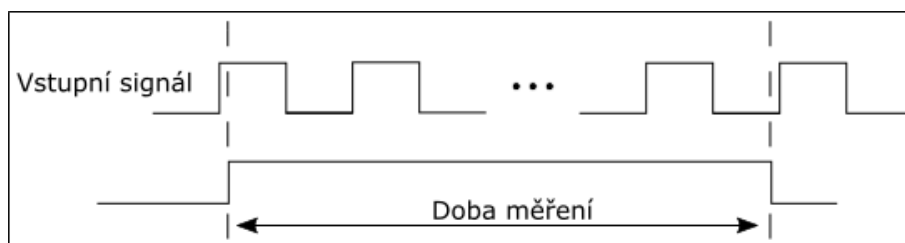
V implementaci pomocí Cube a Keil je snaha dosáhnout co nejpřesnějšího měření a je tedy řešení realizováno hardwarovou metodou, kdy se timery o měření starají bez zasahování programu. Program jen inicializuje timery, spustí a poté může už jen číst změřené údaje.

Reciproční měření K recipročnímu měření používám 32bitový timer v módu pwm input, při kterém se při náběžné hraně vyresetuje counter a spustí se. Při sestupné hraně se zaznamená hodnota do capture compare registru (*CCR*) kanálu 2 a při další náběžné hraně se zaznamená hodnota do *CCR* registru kanálu 1, vyresetuje se counter a začne od znovu. Z hodnoty v *CCR1* se vypočte délka periody a z toho frekvence. Z poměru *CCR1* a *CCR2* se zjistí střída.



Obrázek 10 Měření reciproční metodou a zachycení do registrů.

Přímé měření Přímé měření využívá dva čítače, jeden k určení času snímání a druhý k čítání pulsů. Já používám 16bitový timer k časování a 32bitový timer k čítání pulsů. 32 bitový Timer má nastaven jako trigger druhý timer a jako zdroj hodin měřený signál, takže po dobu jedné vteřiny čítá s frekvencí vstupního signálu a poté zaznamená hodnotu do *CCR* registru a proběhne reset counteru. Na obrázku níže je vidět chyba měření, kdy se během čítání zaznamená pouze náběžná hrana a pokud měření začne hned po konci náběžné hrany, tak se první puls počítat nebude a výsledná frekvence tím pádem bude o 1Hz menší.



Obrázek 11 Princip měření reciproční metodou.

Přímé měření frekvence má chybu závislou na době měření a to dle vzorce $chyba = \frac{t}{1}$, kde t je doba měření. Znamená to tedy, že čím větší časový úsek budeme čítat pulsy

tím menší bude chyba.

5.3.2 Generace pwm

PWM signál generuji na 16bitovém timeru, u kterého používám dynamicky měnící se prescaler, abych mohl generovat frekvenci až 1Hz. $F = prescaler * arr/hodiny$, střída se určuje poměrem hodnoty v registru *ARR* a *CCR* v HAL knihovnách poměr mezi periodou a délkou pulsu

6 Senzory

Tato kapitola se zaměřuje na práci se senzory. Zatímco ultrazvukový snímač vzdálenosti využívá pouze čítačové jednotky ostatní senzory jsou připojeny jen k AD převodníku a probíhá na nich pouze měření napětí.

6.1 Implementace v Mbedu

V embedu jsou implementované pouze měření vzdálenosti s ultrazvukovým snímačem, protože u ostatních senzorů lze využít voltmeter.

6.1.1 Měření vzdálenosti

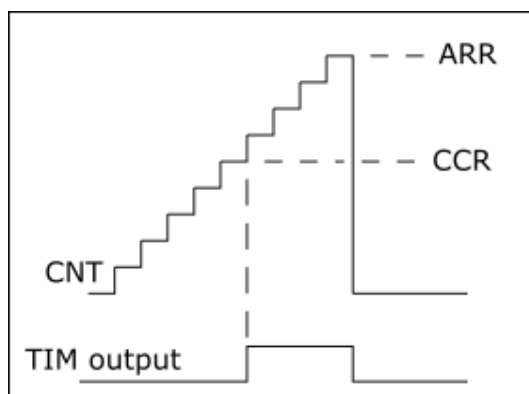
Měření vzdálenosti probíhá s využitím ultrazvukového snímače vzdálenosti, jehož popis naleznete v kapitole 3.3.1. Řešení v Mbedu je implementováno pomocí periférií DigitalOut, DigitalIn a Timer. Pomocí periférie DigitalIn se nastaví na pinu, s trigrem, logická úroveň jedna a po delayi 10us se opět vrátí do nuly. Poté program čeká ve smyčce while dokud je na pinu echo nula, jakmile detekuje jedničku vyskočí ze smyčky zapne timer a po opětovné detekci nuly timer opět vypíná a čte hodnotu timeru, která je v mikrosekundách. Poté se už jen podle vzorce $t[us]/58 = x[cm]$ vypočítá vzdálenost a odešle na sériovou linku.

6.2 Implementace pomocí Cube a Keil

Pomocí keilu je implementováno, jak měření vzdálenosti, tak i měření osvětlení a optická závora. U těchto senzorů se využívají různé metody měření a generování signálů, které se využívali již dříve, u předchozích funkcí, v případě ultrazvukového snímače vzdálenosti jsou tyto metody trochu upravené.

6.2.1 Měření vzdálenosti

Dle popisu Ultrazvukového snímače vzdálenosti HC-SR04 v kapitole 3.3.1 je zapotřebí k zjištění vzdálenosti vyslat 10us puls na jednom pinu a na druhém měřit délku příchozího pulsu. K vyslání pulsu se používá timer v režimu one pulse mode, jedná se o druh PWM generace. Timer čítá, a jakmile se hodnota timeru rovná hodnotě uložené v capture compare registru (CCR) tak přepne výstupní pin do logické jedničky a v tomto stavu zůstane až do resetu (dosažení hodnoty v auto reload register(ARR)). Lépe je tento popis čitelný z obrázku níže(obrázek 12).



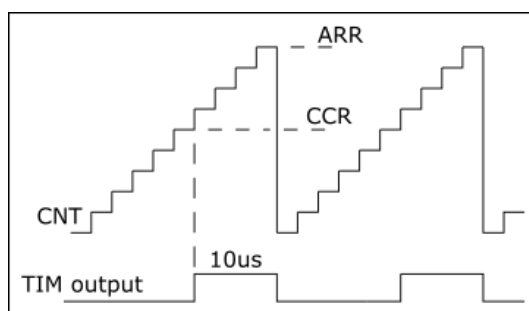
Obrázek 12 Princip funkce one pulse mode single.

Aby se nemusel stále zapínat one pulse mode, je nastavena hodnota ARR a CCR tak, aby se 10us pulz generoval každých 100ms. Nastavení na jednotlivých mikrokontrolérech je znázorněné v tabulce níže (tabulka 3).

Mikročítač	Prescaler	ARR	CCR
STM32F042	95	49999	49995
STM32L432	95	49999	49995
STM32L072	63	49999	49995

Tabulka 3 Nastavení jednotlivých jednotek timer

Při generování kódu v Cube je defaultně one pulse mode nastaven v režimu single, což znamená že se provede jen jeden cyklus, pokud se ale chce, aby se pulz generoval pravidelně, musí se přepnout do režimu repetitive “HAL_TIM_OnePulse_Init(&htim3, TIM_OPMode_REPETITIVE)”. S tímto nastavením se bude pulz generovat pravidelně bez zasahování programu, jak je znázorněno na obrázku níže.

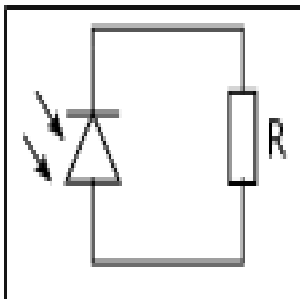


Obrázek 13 Princip funkce one pulse mode repetitive.

K měření je použito měření periody stejné jako je popsáno v kapitole reciproční měření frekvence. V tomto případě je potřeba jen délky pulsu, tedy doby od náběžné hrany k sestupné, a tuto hodnotu je možné nalést uloženou v CCR2 registru. Výpočet vzdálenosti probíhá dle vzorce $t[us]/58 = x[cm]$ kde t je délka pulsu a x je výsledná vzdálenost.

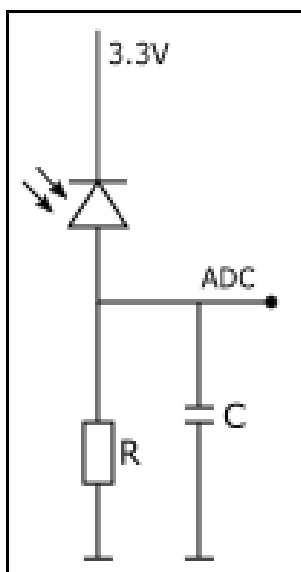
6.2.2 Fotodioda

Fotodioda se používá jako detektor intenzity osvětlení. Jsou dva druhy zapojení fotodiody fotovoltaický režim a odporový režim. Fotovoltaické zapojení se využívá například u solárních panelů, odporové zapojení se díky jeho rychlé odezvě na změnu osvětlení využívá v obvodech pro měření intenzity osvětlení.



Obrázek 14 Fotovoltaické zapojení fotodiody.

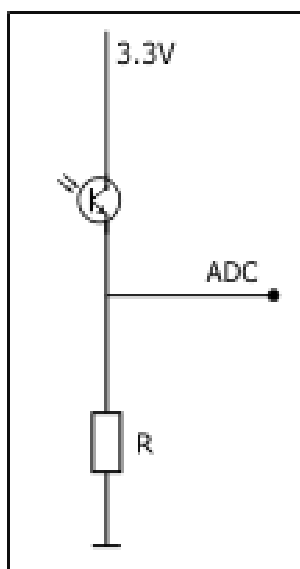
Podle změřeného napětí se zjistí jak moc je fotodioda osvětlena, čím větší napětí tím větší osvětlení na fotodiodě. Napětí se měří na AD převodníku s nastavením které je popsáno v kapitole 4.3.



Obrázek 15 Odporové zapojení fotodiody.

6.2.3 Fototranzistor

Fototranzistor se používá podobně jako fotodioda k detekci intenzity osvětlení. Použito je následující zapojení (obrázek 16). Kde se při větším osvětlení bude na měřícím pinu měřit větší napětí. Napětí se měří na AD převodníku s nastavením, které je popsáno v kapitole 4.3.

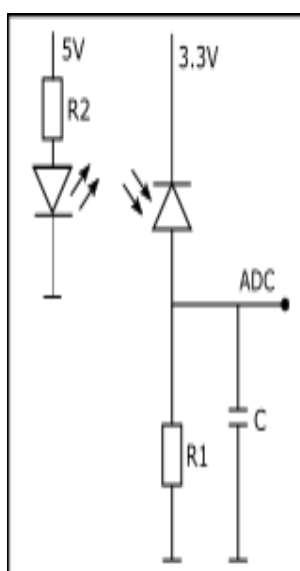


Obrázek 16 Schéma zapojení fototranzistoru.

6.2.4 Optická závora

Optická závora funguje na principu generování světla a jeho zachycení podobně jako ultrazvukový snímač vzdálenosti. V tomto případě je použito infračervené záření. Led dioda generuje neustále stejné záření a fotodiody záření detekuje, pokud dojde k poklesu pod danou mez detekovaného záření vyhodnotí se situace jako překážka mezi led a fotodiodou. Velmi snadno by se pomocí tohoto řešení dal implementovat i reflexní snímač, který by fungoval na principu odrazu, v podstatě by stačilo jen změnit kritickou hodnotu při které se vyhodnocuje překážka.

Jak již bylo zmíněno u fotodiody velikost osvětlení se odvodí od velikosti napětí změřeného na vstupním pinu. Nastavení AD převodníku měřícího napětí je podrobněji popsáno v kapitole 4.3.



Obrázek 17 Schéma zapojení optické závory.

7 BBC Micro:Bit

Během tvorby bakalářské práce se na školu dostal kit od BBC Micro:Bit[7]. Tento kit je určen pro děti ve věku okolo 12ti let a má za cíl u nich probudit zájem o elektrotechniku. Ve Velké Británii byl tento kit v roce 2016 rozdán mezi všechny děti v sedmé třídě základní školy tj. ve věku 11-12 let. Kit disponuje procesorem Nordic nRF51822, magnetometrem, akcelometrem, 17ti vyvedenými piny, dvěma tlačítky a maticí 25 led diod. K programování tohoto kitu je určen na stránkách výrobce jednoduchý blokový javascript editor(obrázek 18), v kterém se děti lehce zorientují, pro pokročilejší je možnost programovat v Pythnu(obrázek 18), také na stránkách výrobce v internetovém editoru. Poslední a pro nás nejvíce zajímavou možností je programovat kit v Mbedu.



Obrázek 18 Ukázka vývojového prostředí pro Micro:Bit.

Vzhledem k tomu, že jsem úlohy dělal v Mbedu na Nucleo, tak jsem zkusil některé programy převést i na Micro:Bit. Při tomto procesu jsem narazil na jediný problém a to ten, že dokumentace ke kitu je velmi těžko dohledatelná a nebo velmi stručná. Při převodu na Micro:Bit se musí změnit použité piny, bohužel značení pinů P0,P1 v Mbedu nefunguje a musí se dohledat schéma a použít značení z něj. Například pro pin P0, který je ve schématu značen P0.03 se musí použít značení PAD1 nebo P0_03.

Pro Micro:Bit jsem zkusil úlohy měření napětí a generování a měření pwm. Gene-

rování pwm nebylo moc přesné. V následující tabulce jsou uvedeny údaje o generované pwm a změřené, všechny pwm byli generované se střídou 50

Nastavená frekvence [Hz]	Generovaná frekvence [Hz]	skutečná střída [%]
1k	1k	50.3
5k	4.89k	50.4
10k	9.5k	50.9
50k	39k	53

Tabulka 4 Ukázka nepřesného generování pwm.

Měření pwm a napětí dosahovalo stejných výsledků podobně jako na nucleo kitech, bylo možné měřit frekvenci až 170kHz.

8 Závěr

Cílem této práce bylo navrhnout řešení přístrojových funkcí v laboratorních přípravcích a prakticky ověřit jejich realizaci. Programové řešení bylo implementováno na třech mikročítačích STM32F042, STM32L432 a STM32L072. Během vývoje se vyskytovaly různé druhy komplikací, největší problém nastal při pájení procesoru a následném bootování přes bootloader a program DFuse, který v nejaktuálnější verzi(3.0.5) s mikrokontrolerem STM32L072 nepracoval a musela se použít jeho starší verze(3.0.3). Další nepříjemnost byla při programování STM32L072, kdy se nedal využít debugger, aniž by se musel připojovat externí ST-Link. Díky těmto komplikacím a špatnému odhadu časové náročnosti při implementování programů na tento mikrokontrolér se mi nepodařilo zprovoznit přímé měření frekvence na tomto mikrokontroléru.

Na ostatních mikrokontrolerech se mi podařilo realizovat všechny navržené funkce. Řešení je rozděleno na několik samostatných částí. Na mikrokontrolér STM32F042 jsou funkce Voltmeter, Ohmeter, měření frekvence a generace pwm, jako jeden samostatný program a práce se senzory jako druhý samostatný program. Na mikrokontroleru STM32L432 bylo nutné rozdělit programy na tři části, kvůli Lite verzi Keilu, která omezuje velikost zdrojového kódu na 32KB. První program obsahuje funkce voltmeter, ohmeter, generování napětí, druhý program obsahuje funkce měření frekvence a generování pwm a třetí program obsahuje práci se senzory. Stejně rozdělení, jako je na STM32L432 bylo použito i na STM32L072.

Literatura

- [1] ST Microelectronics. *STM32F042x4 STM32F042x6 ARM®-based 32-bit MCU, up to 32 KB Flash, crystal-less USB FS 2.0, CAN, 9 timers, ADC and comm. interfaces, 2.0 - 3.6 V*. 2017. URL: <http://www.st.com/content/ccc/resource/technical/document/datasheet/52/ad/d0/80/e6/be/40/ad/DM00105814.pdf/files/DM00105814.pdf/jcr:content/translations/en.DM00105814.pdf> (cit. 10.02.2017).
- [2] ST Microelectronics. *STM32L432KB STM32L432KC Ultra-low-power ARM® Cortex®-M4 32-bit MCU+FPU, 100DMIPS, up to 256KB Flash, 64KB SRAM, USB FS, analog, audio*. 2017. URL: <http://www.st.com/content/ccc/resource/technical/document/datasheet/24/01/9f/59/f0/83/47/fc/DM00257205.pdf/files/DM00257205.pdf/jcr:content/translations/en.DM00257205.pdf> (cit. 05.02.2017).
- [3] ST Microelectronics. *STM32L072x8 STM32L072xB STM32L072xZ Ultra-low-power 32-bit MCU ARM®-based Cortex®-M0+, up to 192KB Flash, 20KB SRAM, 6KB EEPROM, USB, ADC, DACs*. 2017. URL: <http://www.st.com/content/ccc/resource/technical/document/datasheet/98/e1/71/c0/ae/f5/4b/d4/DM00141133.pdf/files/DM00141133.pdf/jcr:content/translations/en.DM00141133.pdf> (cit. 01.04.2017).
- [4] ElecFreaks. *Ultrasonic Ranging Module HC - SR04*. 2017. URL: <http://www.micropik.com/PDF/HCSR04.pdf> (cit. 10.04.2017).
- [5] Tim Wilmshurst Rob Toulson. *Fast and Effective Embedded Systems Design*. Edition 1. Elseveiser, 2012. ISBN: 9780080977690.
- [6] Maurice Dodson. *Shannons sampling theorem*. 1992. URL: http://www.currentscience.ac.in/cs/Downloads/article_id_063_05_0253_0260_0.pdf (cit. 05.05.2017).
- [7] Multiple authors. *BBC micro:bit MicroPython documentation*. 2016. URL: <https://microbit-micropython.readthedocs.io/en/latest/> (cit. 20.05.2017).