

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra měření

Jednoduché laboratorní přístroje pro výuku realizované mikrořadiči ATmega 328 a STM32

Stanislav Novák

Vedoucí práce: doc. Ing. Jan Fischer, CSc.
Studijní program: Kybernetika a Robotika
Srpen 2021

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Novák** Jméno: **Stanislav** Osobní číslo: **483683**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Jednoduché laboratorní přístroje pro výuku realizované mikrořadiči ATmega 328 a STM32

Název bakalářské práce anglicky:

Basic laboratory instruments for teaching purposes based on ATmega 328 and STM32 microcontrollers

Pokyny pro vypracování:

Vyhleďte dostupné realizace jednoduchých laboratorních přístrojů (osciloskop, logický analyzátor, voltmetr, spektrální analyzátor, impulsní generátor, generátor audiosignálu) založené na mikrořadičích ATmega328, případně STM32, které spolupracují s PC. Experimentálně prověřte jejich funkci, analyzujte parametry, vytvořte potřebné návody a posuďte využitelnost v experimentální distanční výuce.

V případě, že řešení přístroje s vhodnými parametry není k dispozici, vytvořte potřebné programy, které budou příslušný přístroj s využitím mikrořadiče řady STM32 a PC realizovat. Zde se zaměřte především na realizaci spektrálního analyzátoru s FFT.

Navrhněte a popište metodiku rychlého návrhu jednoduchých přístrojů s mikrořadičem STM32 s využitím zjednodušujících nástrojů typu STM32Duino, Ardublok a dalších volně dostupných nástrojů tak, aby byla použitelná i pro výuku na středních školách, případně v zájmových kroužcích.

Seznam doporučené literatury:

- [1] Yiu, J.: The Definitive Guide to ARM Cortex -M0 and Cortex-M0+ processors
- [2] STMicroelectronics: RM0091 STM32F0x2 Reference manual
- [3] STMicroelectronics: DS10147 STM32F042 Data

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: **22.01.2021**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce:

do konce letního semestru 2021/2022

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

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
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

Poděkování

Rád bych poděkoval svému vedoucímu bakalářské práce doc. Ing. Janu Fischerovi, CSc. za trpělivé vedení, vstřícný přístup a přínosné připomínky a podněty.

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, 13. srpna 2021

.....

Abstrakt

Tato práce se zabývá realizacemi měřících přístrojů realizované mikrořadiči STM32 a ATmega328. Práce obsahuje popis parametrů, nedokonalostí, a návody k použití k jednotlivým realizacím. Realizace byly byly otestovány na různých platformách a jejich reálné parametry byly srovnány s deklarovanými popisy autorů.

Práce má soužit jako zdroj informací pro zájemce o elektroniku, kteří hledají levnou alternativu k přístrojům pro profesionální využití.

Klíčová slova: STM32, Osciloskop, Voltmetr, Logický analyzátor, Arduino, STM32duino

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

Abstract

The thesis handles measuring tools implemented on STM32 and ATmega328 microcontrollers. Text is consisted of distribution of parameters, problems and instructions for use. Measuring tools were tested and acquired parameters of measuring tools were compared to author's distribution.

Purpose of thesis is to give information for electronics enthusiasts searching for alternative to devices for professional use.

Keywords: STM32, Oscilloscope, Voltmeter, Logic analyzer, Arduino, STM32duino

Title translation: Basic laboratory instruments for teaching purposes based on ATmega 328 and STM32 microcontrollers

Obsah

1 Úvod	1
2 Volně dostupné realizace měřících přístrojů	3
2.0.1 Požadavky na hledané realizace přístrojů	3
2.1 Existující měřící přístroje pro moduly Arduino s mikrořadiči ATmega328	5
2.1.1 Osciloskop ZaidaScope	5
2.1.2 Osciloskop BegOscopio	7
2.1.3 Logický analyzátor SUMP compatible logic analyzer for Arduino	8
2.2 Existující měřící přístroje pro moduly STM32	9
2.2.1 Osciloskop Miniscope	10
2.2.2 Spektrální analyzátor LogicAlNucleo	11
3 Přístroje vyvinuté v rámci práce	15
3.1 Port programu zero-elabviewer pro Arduino	15
3.1.1 Parametry a vlastnosti přístroje:	16
3.1.2 Návod k obsluze přístroje zero_elabviewer s modulem Arduino:	16
3.2 Spektrální Analyzátor FFTScope	17
3.2.1 Návod k obsluze přístroje	18
4 Balíky STM32duino a ardublock	21
4.0.1 Prostředí Arduino IDE	21
4.1 STM32duino	23
4.1.1 Instalace do prostředí Arduino IDE:	23
4.1.2 Instalace a nastavení STM32CubeProgrammer	25
4.1.3 Programové možnosti při použití balíku STM32duino	26
4.1.4 Konfigurace STM32G031	28
4.1.5 Offline instalace balíku STM32duino	29

4.2 Prostředí Ardublock	30
4.2.1 Kombinace balíků STM32duino a Ardublock	30
5 Zhodnocení	33
Literatura	35
A Obsah přiloženého CD	37

Obrázky

2.1 Schéma požadované funkční struktura měřících přístrojů	3
2.2 Moduly s mikrořadiči, zleva Arduino Uno, Blue Pill, Nucleo F303, F0-Lab	4
2.3 Okno aplikace ZaidaScope	6
2.4 Okno aplikace Begoscopio	8
2.5 Okno programu LogicSniffer při inicializaci	10
2.6 Okno programu LogicSniffer po měření	11
2.7 Okno aplikace miniscope zobrazující spektrální obraz 10kHz obdélníkového signálu.	12
2.8 Okno aplikace miniscope zobrazující spektrální obraz 10kHz obdélníkového signálu.	13
3.1 Okno programu zero_elabviewer	15
3.2 Okno programu zero_elabviewer při spojení s modulem	17
3.3 Okna jednotlivých částí přístroje zero_elabviewer	18
3.4 Okno votmetru se záznamem realizace zero_elabviewer	19
3.5 Okno aplikace FFTscope zobrazující spektrální obraz na indukovaného 50Hz signálu ze sítě	19
4.1 Prostředí Arduino IDE	22
4.2 Okno Vlastnosti aplikace Arduino IDE	23
4.3 Okno Manažér desek	24
4.4 Výběr modulu rodiny STM32	24
4.5 Výběr metody nahrávání firmware do modulu	25
4.6 Okno Vlastnosti systému	25
4.7 Přidání aplikace STM32CubeProgrammer do proměnného prostředí	26
4.8 Nastavení bootování a resetu pomocí STM32CubeProgrammer	29
4.9 Nastavení Arduino IDE pro nahrávání firmware pomocí UART	29

4.10 Ukázka prostředí Ardublock..... 31

Tabulky

4.1 Tabulka serových portů definovaných v balíku STM32duino	27
4.2 Tabulka přiřazení jednotlivých pinů pinu modulu Arduino pro testované moduly	32



Kapitola 1

Úvod

Měřicí přístroje na profesionální úrovni nabízí velmi dobré parametry, jsou však nákladné a při výuce elektroniky jich není vždy dostatek. To může být překážkou k používání na středních školách a zvláště pak v zájmových kroužcích, kde bývají rozpočty velmi omezené.

Tato práce si klade za cíl představit některé možnosti použití mikroprocesorových modulů jako měřících přístrojů za minimální pořizovací náklady.

V první řadě je v práci obsažen přehled realizací měřících přístrojů, které mohou vyučující nebo vedoucí kroužků využívat. V druhé řadě obsahuje dosti podrobné návody, jak je možné realizace instalovat, používat a jaké funkce obsahují. V neposlední řadě ukazuje některé slabiny i silné stránky realizací měřících přístrojů a jejich parametry. To může být opravdu přínosné, protože vycházejí ze zkušeností z reálného testování na různých platformách. Popisy funkcí k jednotlivým realizacím lze dohledat u jejich autorů, ale popis problémů se dohledává podstatně hůře.

U některých realizací vybraných jsem stávající kód podle potřeby doplnil. Jedná se o realizaci `zero_elabviewer`, kterou je díky tomu možné používat na modulu `arduino`.

Celkově by mohla být, jak doufám, užitečným pomocníkem, který může znatelně ušetřit čas a být inspirací pro náplň výuky nebo zájmové činnosti v daném oboru.

Kapitola 2

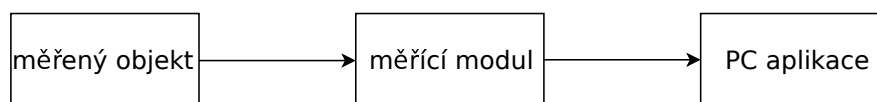
Volně dostupné realizace měřících přístrojů

V této kapitole je jsou popsány volně dostupné realizace laboratorních přístrojů založené na mikrořadičích ATmega328 a STM32. Je popsána jejich funkčnost a uvedeny jejich parametry. Je zhodnocena také případná využitelnost v experimentech při výuce. Realizace přístrojů byly prakticky vyzkoušeny a jejich funkčnost ověřena v typických úlohách. Realizace přístrojů byly zhodnoceny i z hlediska použití na různých operačních systémech (Windows, Linux, macOS).

2.0.1 Požadavky na hledané realizace přístrojů

Požadovaná funkční struktura

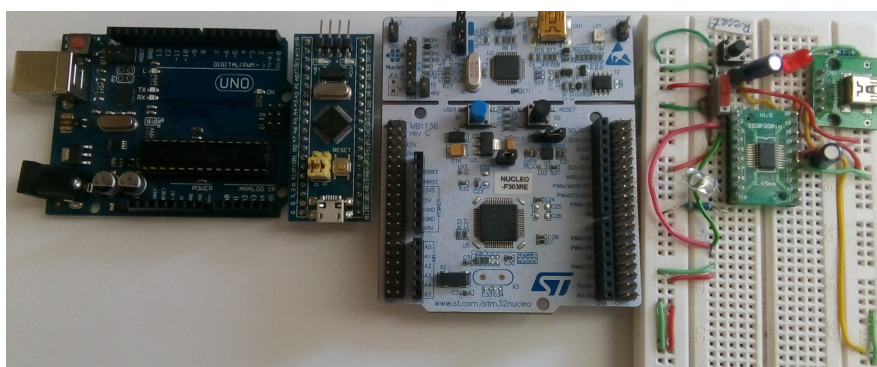
Požadovaná funkční struktura pro realizaci přístroje je uvedena na obrázku 2.1. Měření probíhají na modulu, odtud jsou naměřená data odesílána do PC, kde jsou v aplikaci konkrétní realizace přístroje zobrazena. Tato struktura nabízí řadu výhod, předně velká část dostupných přístrojů umožňuje naměřená data ihned po naměření uložit do souboru a zpracovat v externí aplikaci např. Microsoft Excel. Možnost zpracování dat v tabulkovém procesoru dovoluje data rychle zpracovat, vyhodnotit a zobrazit. Vhodnost této struktury pro využití ve výuce je také ve skutečnosti, že modul stačí pouze připojit k PC a není nutné rozšiřovat modul o další komponenty např. LCD, které mohou komplikovat uvedení do chodu přístroje. Výstupy ze standalone přístroje nejsou obvykle stejně rychle zpracovatelné, jako u realizací přístrojů s námi požadovanou funkční strukturou.



Obr 2.1: Schéma požadované funkční struktura měřících přístrojů

Použité moduly, jejich parametry

Moduly na kterých jsou přístroje realizovány, jsou moduly Arduino využívající mikrořadič ATmega328 a moduly Nucleo, Discovery, Blue Pill využívající mikrořadiče rodiny STM32 (obrázek 2.2). ATmega328 je osmibitový mikrořadič architektury AVR. Jedná se o řadič obsažený v modulech Arduino Uno a Arduino Nano. ATmega 328 disponuje jedním desetibitovým AD převodníkem s osmi kanály a třemi čítači. Mikrořadič nedisponuje DA převodníkem ani jednotkou DMA. STM32 je rodina mikrořadičů architektury ARM. Pro tuto práci jsou důležité moduly s mikrořadiči STM32F042 (obsažené v modulu Nucleo a v kitu F0-Lab), STM32F303RE (obsažené v modulu Nucleo), STM32F103C8 (obsažené v modulu Blue Pill). Všechny tyto mikrořadiče disponují jednotkou DMA a alespoň jedním dvanáctibitovým AD převodníkem. DA převodníky (konkrétně dvěma) disponuje pouze mikrořadič STM32F303RE.



Obr 2.2: Moduly s mikrořadiči, zleva Arduino Uno, Blue Pill, Nucleo F303, F0-Lab

Testování realizací přístrojů

Testování realizací přístrojů probíhalo na třech typických úlohách. Typickými úlohami je myšleno měření časové konstanty RC článku, odečet periody a střídy obdélníkového signálu, měření napětí v daném čase. Měření časové konstanty RC článku probíhalo připojením integračního RC článku na referenční zdroj obdélníkového signálu. Odečtením časové konstanty a porovnáním odečtené hodnoty se spočtenou časovou konstantou $\tau = R.C$ získáme představu, do jaké míry je možné u daného osciloskopu považovat naměřené hodnoty za relevantní.

Odečet periody a střídy probíhal připojením obdélníkového signálu z referenčního přístroje. Měření probíhalo na frekvencích 1, 10, 100 Hz, 1, 10, 100 kHz a pro hodnoty střídy 1, 2, 5, 10, 20, 50, 75, 100 %. Měření stejnosměrného napětí probíhalo připojením referenčního zdroje, konkrétně čtyř hodnot 0, 1, 2, 3.3 V.

Při testování bylo použito jako referenčního zdroje přístroje Little Embedded Oscilloscope provozovaného na modulu Nucleo STM32F303RE. U každého přístroje byla vždy provedena měření odpovídající jejich určení (např. u logického analyzátoru nedává smysl odečítat časovou konstantu).

2.1 Existující měřící přístroje pro moduly Arduino s mikrořadiči ATmega328

Realizace přístrojů, kterými jsem se zabýval, byly: logický analyzátor, voltmetr, spektrální analyzátor, impulsní generátor, generátor audiosignálu. Z těchto přístrojů jsou podrobně zpracovány dva osciloskopy a jeden logický analyzátor. Voltmetr s generátorem obdélníkového signálu byl v rámci práce vyvinut, je uveden v kapitole 3.1. Spektrální analyzátor ve volně dostupných realizacích neodpovídá struktuře definované v kapitole 2.0.1, obvykle je k modulu připojen display nebo matice LED. Zajímavou realizací využívající spektrální analýzy a modulu Arduino je **Open Source FFT Spectrum Analyzer**¹, má však dva zásadní nedostatky, neměří data z AD převodníku a desktopová aplikace je napsaná ve staré verzi jazyka Python.

Generátor audiosignálu není díky absenci DA převodníku možné realizovat přímo, jednou z možností může být použití PWM s dolnoproputným filtrem, tato realizace je uvedena v kapitole 4.1.3 této práce jako příklad použití balíku STM32duino (zdrojový kód je bez úprav použitelný i pro moduly Arduino). Existující realizace jsou obvykle založeny na realizaci DA převodníku ze součástek, příkladem může být projekt **Arduino Waveform Generator**².

2.1.1 Osciloskop ZaidaScope

Stránka projektu se nachází zde³. Autor předřazuje před mikrořadič napěťový sledovač pro získání relevantnějších výsledků v určitých aplikacích, pro základní měření však není jeho použití nezbytně nutné. Aplikace je šířena pod licencí CC-ND, proto není možné ji nikterak upravovat.

Parametry a vlastnosti ZaidaScope

ZaidaScope je realizace osciloskopu měřící až na osmi kanálech. Přístroj může být nastaven na vzorkování frekvencí až 100 kSa/s při měření na jednom kanálu, při měření na více kanálech se maximální vzorkovací frekvence snižuje. Přístroj vypisuje reálnou vzorkovací frekvenci. Přístroj obsahuje funkci trigger na náběžnou i sestupnou hranu. PC aplikace obsahuje možnost měření pomocí kurzorů. Realizace neobsahuje generátor obdélníkového signálu. Aplikace automaticky vypočítává frekvenci signálu z naměřeného průběhu. PC aplikace je dostupná pouze pro operační systém Windows.

Maximální vzorkovací frekvence je u přístroje omezená několika faktory, prvním je použitý převodník modulu. Pokud je použit obyčejný převodník UART-USB, je možné nastavit vzorkovací frekvenci na 100 kSa/s při měření na jednom kanálu. Při použití modulu s převodníkem ATmega16U2 je omezena vzorkovací frekvence na 10 kSa/s při měření na jednom kanálu. Při měření na více kanálech se vzorkovací frekvence snižuje, protože mikrořadič ATmega328 disponuje pouze jedním

¹<https://hackaday.io/project/12109-open-source-fft-spectrum-analyzer>

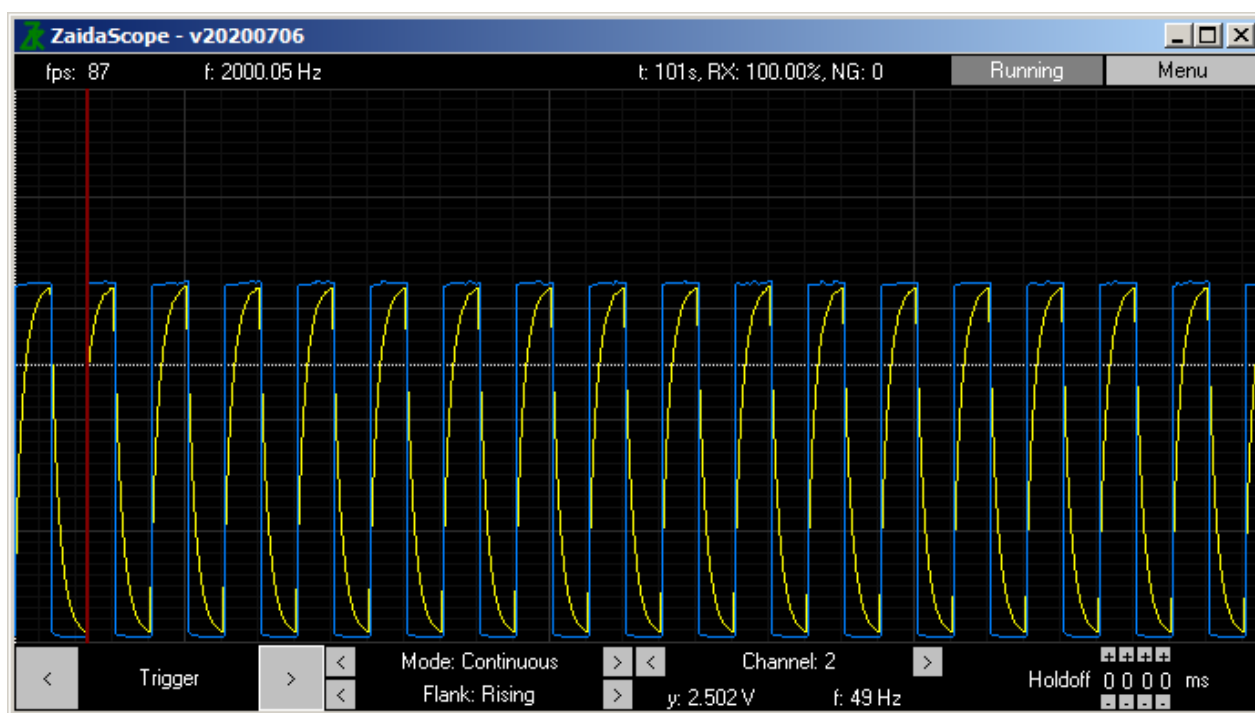
²<https://www.instructables.com/Arduino-Waveform-Generator/>

³<https://www.hackster.io/zaidatek/zaidascope-arduino-oscilloscope-8-ch-max-100-khz-c774f5>

AD převodníkem. Při použití modulu s UART-USB převodníkem je sice možné zvýšit vzorkovací frekvenci nad 10 kSa/s, ale za cenu možných nepřesností měření, protože AD převodník na takto vysoké vzorkovací frekvence není navržen.

■ Návod k obsluze ZaidaScope

Uvedení do provozu: Z webu⁴ stáhneme archiv obsahující firmware a aplikaci. Archiv rozbalíme, zdrojový kód firmware nalezneme v souboru `at328p\atm328p.ino`. Firmware nahrajeme pomocí aplikace Arduino IDE do modulu. Aplikaci nalezneme jako soubor `ZS.exe` ve staženém archivu. Při spuštění aplikace `ZS.exe` se zobrazí okno (obrázek 2.3), ve kterém je potřeba nastavit aktivní kanály, vzorkovací frekvenci, seriový port modulu. Tlačítkem **Connect** zahájíme měření. Měření může probíhat na pinech A0 – A8 modulu.



Obr 2.3: Okno aplikace ZaidaScope

Ovládací prvky osciloskopu sloužící pro ovládání přístroje se nachází v několika nabídkách v dolní části okna. Mezi těmito nabídkami je možné přepínat pomocí tlačítek se šipkami vlevo dole. V těchto nabídkách nalezneme nabídky nastavující trigger, jednotky měřených veličin, kurzory, režimy zobrazení dat. V horní části okna se nachází výpis reálné vzorkovací frekvence, tlačítko **Running** zastavující měření, tlačítko **Menu** ukončující měření a návrat do nabídky nastavení přístroje.

⁴<https://github.com/ZaidaTek/ZaidaScope/raw/master/ZS-latest.zip>

2.1.2 Osciloskop BegOscopio

Odkaz na stránku projekt je zde⁵. Aplikace je v originální verzi v portugalštině, proto jsem ji přeložil do angličtiny.

Parametry a vlastnosti BegOscopio

BegOscopio je realizace osciloskopu měřící až na čtyřech kanálech. Maximální vzorkovací frekvence pro jeden kanál je 35 kSa/s, při měření na více kanálech se maximální vzorkovací frekvence snižuje. Přístroj vypisuje reálnou vzorkovací frekvenci. Přístroj disponuje funkcí trigger na náběžnou hranu. Aplikace umožňuje měření v zaznamenaném průběhu pomocí kurzorů. Přístroj disponuje vlastním generátorem obdélníkového signálu s nastavitelnou frekvencí a střídou. Přístroj umožňuje zobrazení kanálu v režimu x,y. Aplikace umožňuje uložit naměřená data do souboru. Aplikace je dostupná pro operační systémy Windows, Linux a macOS.

Návod k obsluze přístroje BegOscopio

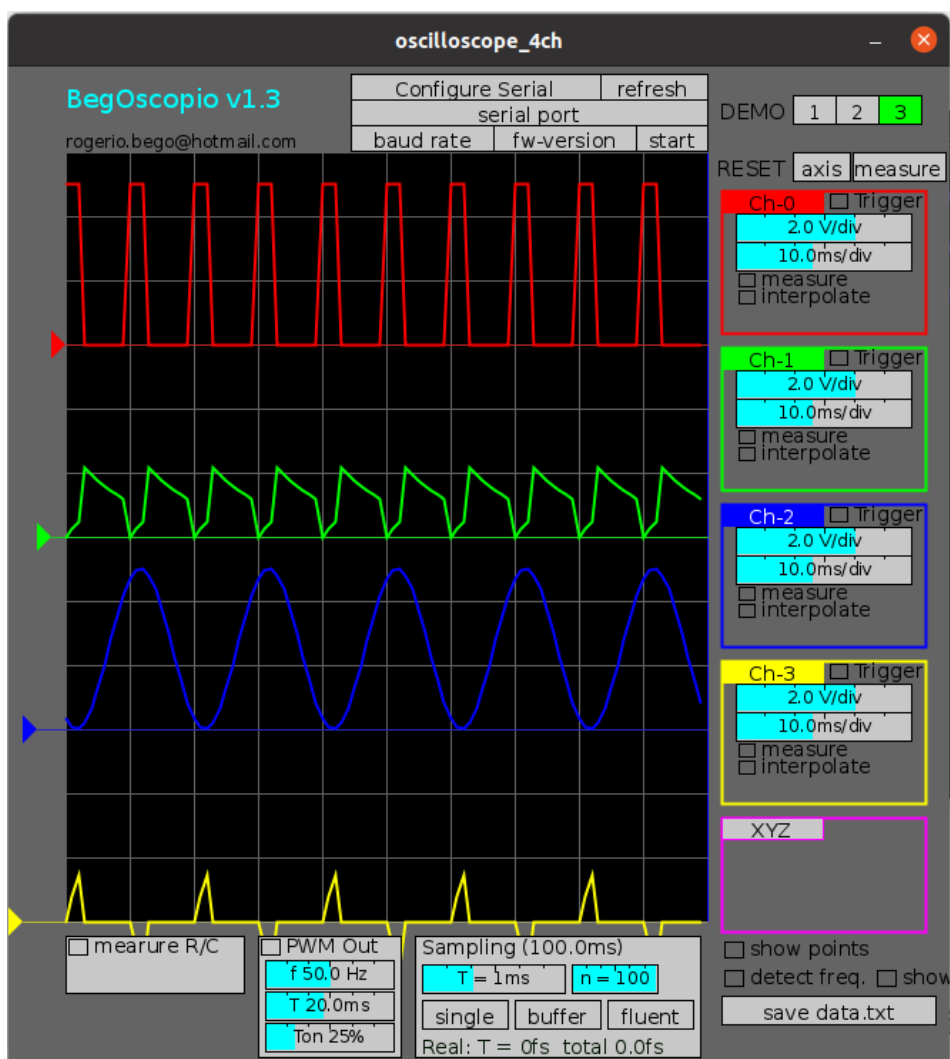
Uvedení do provozu: Z webu⁶ stáhneme v archivu veškerý obsah. Ze staženého archivu nahrajeme pomocí aplikace Arduino IDE soubor s firmware BegOscopio/BegOscopio.ino. Ve staženém archivu se také nachází aplikace, konkrétně v cestě `application.windows64/oscilloscope_4ch.exe`.

Spustíme aplikaci `oscilloscope_4ch.exe`. V horní části desktopové aplikace (obrázek 2.4) nastavíme sériový port modulu a přenosovou rychlost 115200 baud. Přístroj spustíme tlačítkem **start**. Periodu vzorkování nastavíme ve spodní části okna (slider **T**). Počet vzorků nastavíme sliderem **n**. Režim single spustíme tlačítkem **single**, kontinuální režim tlačítkem **buffer** (jediný režim, který respektuje periodu vzorkování, funguje tak, že navzorkuje data, která následně pošle). Tlačítko **fluent** nastaví odeslání dat ihned po naměření (pomalejší, než předchozí režimy). Pod těmito prvky se zobrazuje výpis reálné vzorkovací periody **dt**.

Generovaný signál lze upravit buď sliderem **T**, nebo **f**; závisí na tom, zda chceme nastavit periodu, nebo frekvenci; střída je nastavitelná sliderem **Ton**. Jednotlivé kanály můžeme přidat nebo odebrat kliknutím na popisek s jejich jmény (např. **CH-0**). U každého kanálu je možné zvlášť nastavit napěťovou (v/div) a časovou základu (ms/div). Při zatrhnutí položky **measure** je zapnut režim měření pro daný kanál; klikem a tažením po obrazovce osciloskopu je možné odečítat hodnoty napětí (pozor, pokud nestačíme vzorkovat (reálná hodnota červená), tak odečtené hodnoty nebudou souhlasit). Zatrhnutím položky **interpolate** nastavíme interpolaci naměřených dat. Data je možné uložit tlačítkem **save datax.txt**

⁵<https://www.instructables.com/Oscilloscope-Arduino-Processing/>

⁶https://drive.google.com/drive/folders/1YqMa1qUi1SLNL6FzCZbFwx_z4hl0ooxN?usp=sharing



Obr 2.4: Okno aplikace Begoscopio

2.1.3 Logický analyzátor SUMP compatible logic analyzer for Arduino

Tento logický analyzátor byl vytvořen uživatlem gillham⁷. Jedná se o přístroj na internetu hojně rozšířený, existuje velké množství méně či více vydařených PC aplikací. Realizace využívá komunikační protokol obvykle nazývaný SUMP⁸. Tento protokol vychází z projektu logického analyzátoru realizovaného pomocí FPGA, díky své jednoduchosti je často implementován do nenáročných realizací spektrálních analyzátorů.

⁷https://github.com/gillham/logic_analyzer/

⁸<https://www.sump.org/projects/analyzer/>

■ Parametry a vlastnosti

Realizace disponuje šesti kanály na pinech D2 – D7. Realizace dosahuje vzorkovací frekvence až 4 MSa/s, není však možné měřit kontinuálně. Minimální nastavitelná vzorkovací frekvence je 10 Sa/s. Délka měření je omezená pamětí 1024 vzorků. Přístroj měří vždy zadaný počet vzorků určitou vzorkovací frekvencí. Naměřená data následně odešle do aplikace. Součástí přístroje je i generátor obdélníkového signálu s frekvencí 100 kHz na pinu D11. Přístroj disponuje také funkcí trigger, ta však není pro vzorkovací frekvence vyšší 500 kSa/s dostupná.

K firmware autor nepřikládá aplikaci, protože komunikační protokol převzal z existujícího projektu. Aplikace zde uváděná je kompatibilní se systémy Windows, Linux i macOS. Přístroj by měl být kompatibilní se softwarovým balíkem sigrok, přesto se při testování vyskytly komplikace. Pod Windows se nepodařilo modul s aplikací spojit, pod Linux se spojení navázat podařilo, ale přístroj po několika měřeních vypisoval nesprávné údaje.

Přístroj je kompatibilní i s moduly Arduino využívající jiné mikrokontroléry než ATmega328. Pro moduly používající mikrokontroléry ATmega168 je snížena maximální vzorkovací frekvence na 2 MSa/s a paměť na 512 vzorků. Pro moduly používající mikrokontroléry ATmega2560 je zvýšena paměť přístroje na 7168 vzorků a přístroj disponuje osmi kanály.

■ Návod k obsluze

Uvedení do provozu: Z webu⁹ stáhneme veškerý obsah jako archiv. Zdrojový kód firmware nalezneme v archivu v cestě `logicSUMP/logicSUMP.ino`, nahrajeme pomocí aplikace Arduino IDE do modulu. Jako aplikaci použijeme program `LogicSniffer`¹⁰. Ten spustíme souborem v cestě `ols-0.9.7.2/run.bat`. Po spuštění programu `LogicSniffer` (obrázek 2.5) nastavíme parametry měření pomocí nabídky **Capture** → **Begin capture**. Zde nastavíme položku **Device type** na **Arduino Generic Logic Analyzer**, v kartě **Acquisition** nastavíme vzorkovací frekvenci a počet zaznamenaných vzorků. Tlačítkem **Capture** zahájíme měření. Jako vstupy jsou použity piny D2 – D7, generátor obdélníkového signálu nalezneme na pinu D11. Na obrázku 2.6 je zachycen naměřený průběh obdélníkového signálu generovaného z pinu D11 při vzorkovací frekvenci 500 kSa/s.

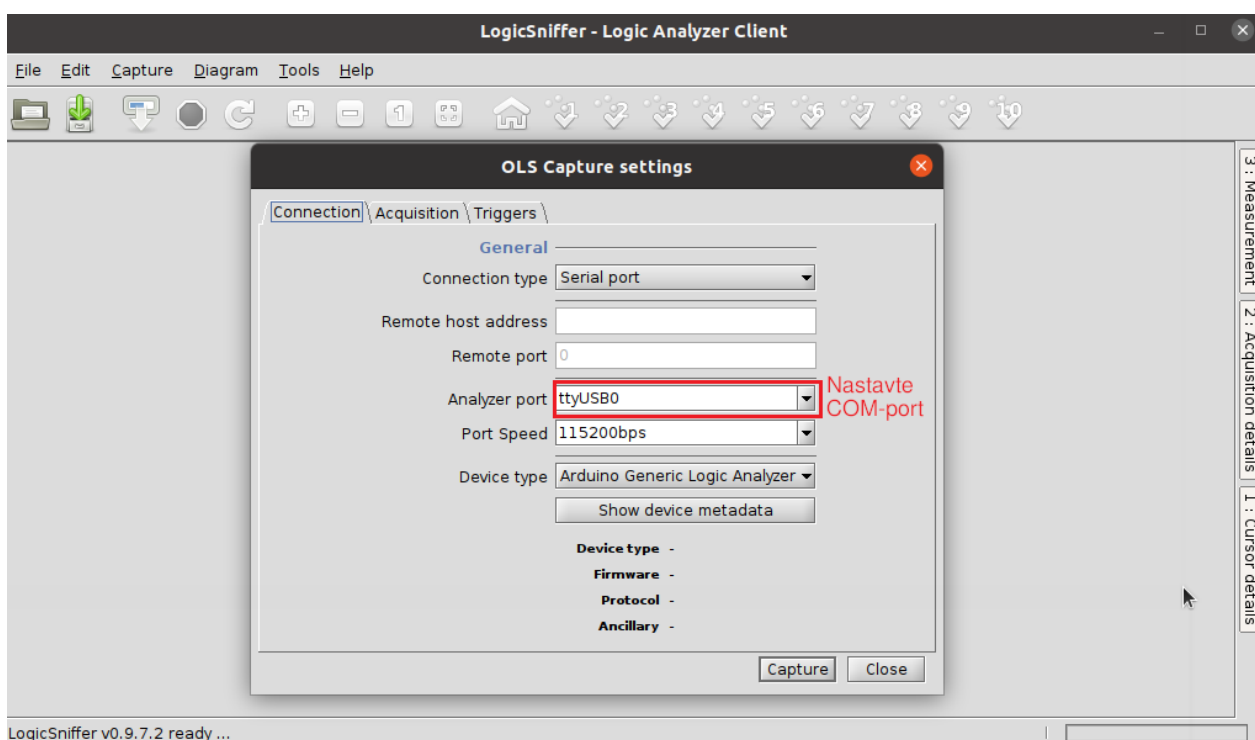
■ 2.2 Existující měřící přístroje pro moduly STM32

Pro moduly STM32 existuje relativně velké množství realizací měřících přístrojů. Velmi dobře zpracované přístroje lze nalézt na webu katedry měření FEL¹¹, na které jsou často v rámci závěrečných prací tyto přístroje vyvíjeny. Příkladem může být realizace `Little Emebeded Oscilloscope` obsahující voltmetr, osciloskop, generátor funkcí a zdroj napětí. Dalším příkladem je realizace `zero_elabviewer` obsahující voltmetr, osciloskop a generátor obdélníkového signálu. V této podkapitole jsou popsány dvě volně dostupné realizace osciloskopu a logického analyzátoru.

⁹<https://drive.google.com/drive/folders/13cwYv6wEWSeYn9DD2mUMdfQh1fPMx7-J?usp=sharing>

¹⁰<https://www.lxtreme.nl/ols>

¹¹<https://embedded.fel.cvut.cz/>



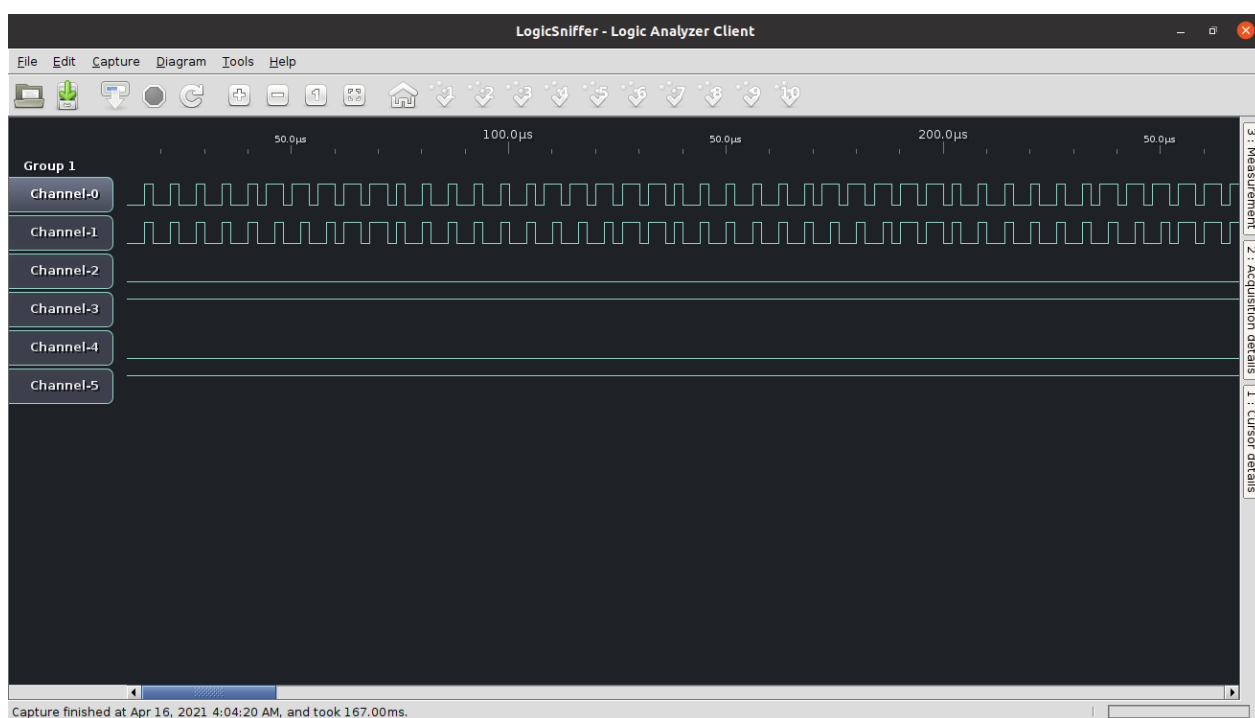
Obr 2.5: Okno programu LogicSniffer při inicializaci

2.2.1 Osciloskop Miniscope

Realizace Miniscope¹² je realizace osciloskopu dostupná pro různé mikrořadiče STM32. Aplikace je realizovaná pro kontroléry STM32F103, STM32F042, STM32F303, STM32F401. Na hlavní stránce je vždy uvedeno schema zapojení obvodu vstupu, realizaci je však možné používat pouze s holým mikrořadičem. Hlavním problémem této realizace je, že používá vlastní ovladač USB, který není podepsán a jeho instalace je na Windows vyšší verzi než XP problematická, návod neuvádím, protože se postup liší verzemi systému a není spolehlivý. PC Aplikace obsahuje základní nástroje jako kurzory, nastavení triggeru a vzorkovací frekvence. Zajímavou funkcionalitou PC aplikace je výpočet spektra pomocí FFT (obrázek 2.7).

Každá verze se díky rozdílnému hardwaru liší v parametrech, uvedu tedy parametry pro realizaci v2e používající mikrořadič STM32F042: 1 kanál, 480 kSa/s maximální vzorkovací frekvence, trigger. Aplikace byla otestována.

¹²http://tomeko.net/miniscope_v2e/index.php?lang=en



Obr 2.6: Okno programu LogicSniffer po měření

2.2.2 Spektrální analyzátor LogicAlNucleo

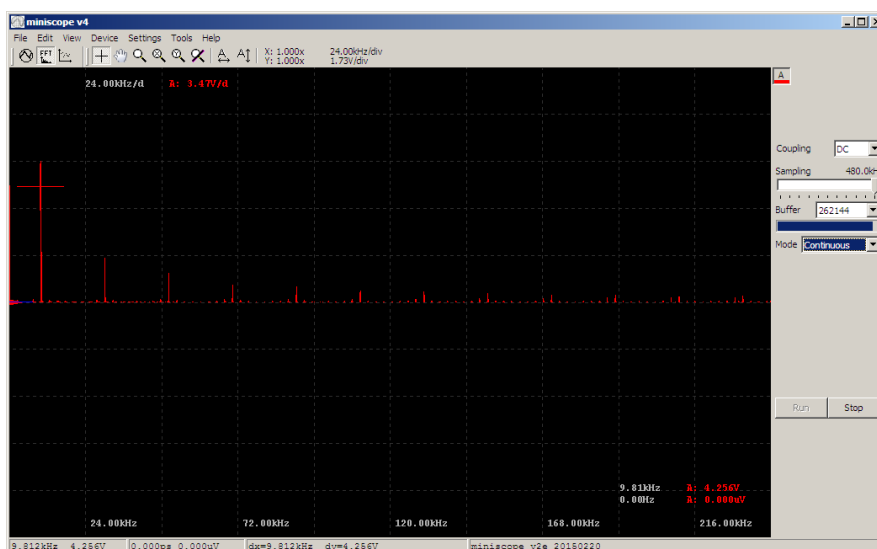
Realizace spektrálního analyzátoru LogicAlNucleo¹³ využívá modul Nucleo STM32F401RE a je napsána s využitím knihovny mbed. Jako komunikační protokol je využit protokol projektu logického analyzátoru SUMP¹⁴. Signál je nejdříve navzorkován, pak odeslán.

Parametry a vlastnosti

Realizace měří na osmi kanálech, konkrétně na pinech PB0 – PB7. Maximální vzorkovací frekvence je 10 MSa/s. Realizace zaznamenává průběh do bufferu s kapacitou až 32 768 vzorků. Realizace má funkční trigger. Realizaci lze spojit s programem PulseView obsažené v balíku sigroc. Součástí balíku sigroc jsou i dekodéry komunikace na sběrnících, díky tomu může být realizace použita nejen pro experimentální výuku, ale i jako levný diagnostický nástroj.

¹³<https://github.com/jpbarraca/LogicAlNucleo>

¹⁴<https://www.sump.org/projects/analyzer/>



Obr 2.7: Okno aplikace miniscope zobrazující spektrální obraz 10kHz obdélníkového signálu.

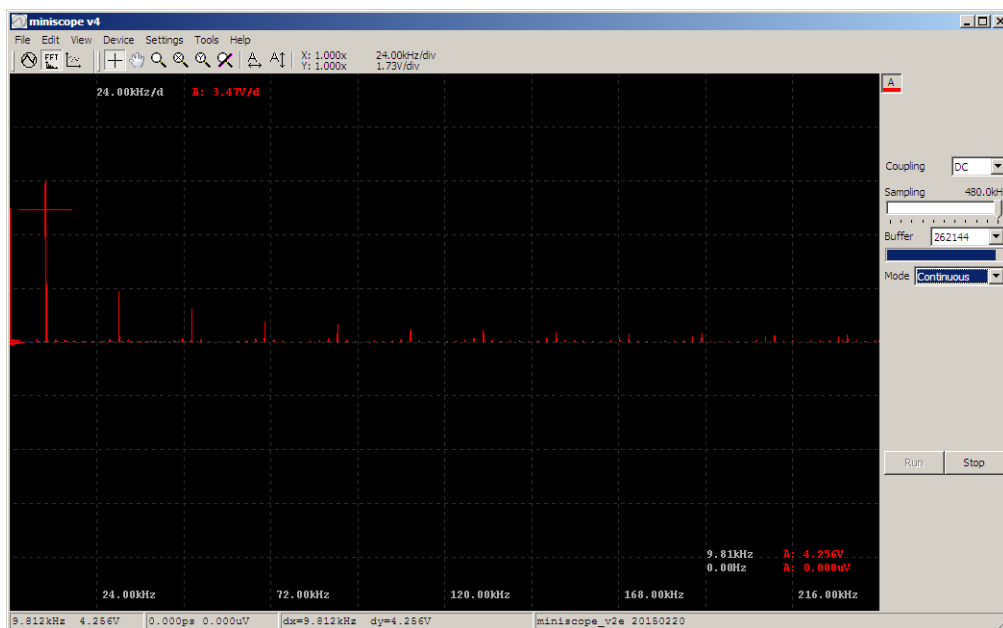
Návod k obsluze

Uvedení do provozu: Z webu¹⁵ stáhneme zkompileovaný firmware. Firmware nahrajeme do modulu, (např. pomocí aplikace STM32CubeProgrammer). Z webu¹⁶ stáhneme aplikaci PulseWiew pro námi používaný operační systém. Aplikaci spustíme, nejdříve se spojíme se modulem tlačítkem <No Device>. V nově otevřeném okně **Connect to device** nastavíme driver **Openbench Logic Sniffer & SUMP compatibles (ols)**, vybereme možnost připojení jako **Serial port**, z nabídky vybereme COM port našeho modulu a přenosovou rychlost 115200 baud. Tlačítkem **Scan ...** zjistíme, zda je zařízení v dosahu. Pokud vše proběhlo správně, mělo by okno vypadat takto ??, pokud je vše v pořádku, nastavení potvrdíme tlačítkem **OK**. (obrázek 2.8 vlevo).

Měření zahájíme tlačítkem **Run** nahoře, parametry měření se nastavují na nástrojové liště.

¹⁵<https://github.com/jpbarraca/LogicAlNucleo/releases/download/v0.1.2.1/LogicAlNucleo-0.1.2.bin>

¹⁶<https://sigrok.org/wiki/Downloads>



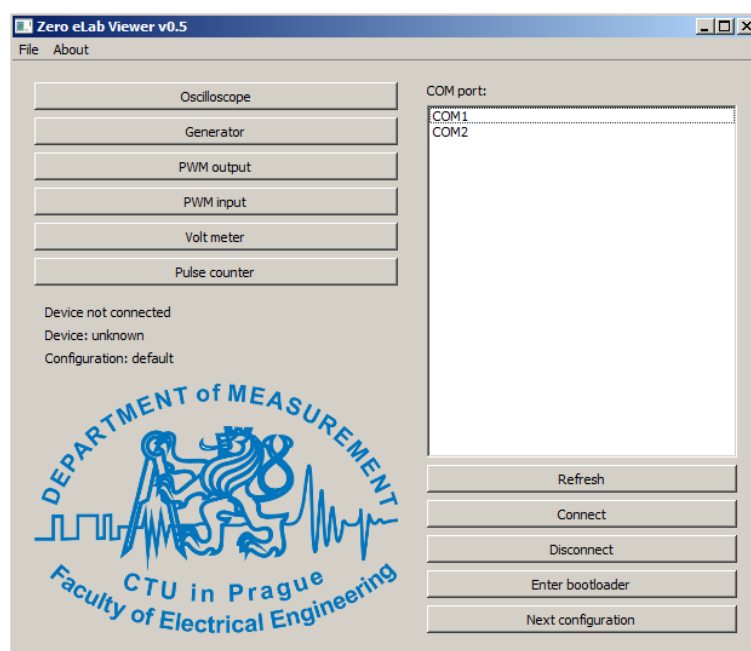
Obr 2.8: Okno aplikace miniscope zobrazující spektrální obraz 10kHz obdélníkového signálu.

Kapitola 3

Přístroje vyvinuté v rámci práce

3.1 Port programu zero-elabviewer pro Arduino

Přístroj zero-elabviewer byl vyvinut jako univerzální měřicí přístroj pro mikrokontroléry STM32F042 (obrázek 3.1). Jeho autorem je Adam Berlinger, aplikaci popsal ve své diplomové práci[1]. Tento



Obr 3.1: Okno programu zero_elabviewer

přístroj je hojně využíván při výuce na katedře měření FEL-ČVUT. Aplikace dostupná pro Windows, Linux a nově i pro macOS.

Při vývoji voltmetru pro moduly Arduino se mi jevilo jako výhodné použít již existující aplikaci

tohoto přístroje a vyvinul jsem pouze firmware tak, aby zachovával všechny možnosti původního přístroje a zároveň aby nebylo nutné aplikaci nijak modifikovat. To se mi podařilo.

■ 3.1.1 Parametry a vlastnosti přístroje:

Voltmetr:

- Tři kanály voltmetru (piny A0 – A2)
- Zobrazení napájecího napětí v aplikaci; naměřené hodnoty jím korigované (Přesnost voltmetru je závislá na napětí vnitřní reference AD převodníku, ta je v katalogovém listu uvedena jako $1.1 \pm 0.1V$.)
- Vnitřní analogová reference mikrořadiče využita ke korekci měřeného rozsahu
- Pevná frekvence měření 100 Sa/s
- Průměrování vzorků v rozsahu 1 – 255 zvyšující zdánlivě přesnost
- Zobrazování naměřených dat v grafu a uložení dat do souboru *.csv

Generátor obdélníkového signálu:

- Jeden kanál s výstupem na pinu D9
- Plynule nastavitelná frekvence v rozmezí 0.477 Hz – 8 MHz s výpisem reálně nastavené frekvence
- Plynule nastavitelná střída v desetínách procenta (platí pro frekvence do 100 kHz, pro vyšší frekvence je díky řešení generátoru pomocí čítače přesnost nastavení střídy nižší.)

Generátor obdélníkového signálu je založen na volně dostupně knihovně Timer1.¹

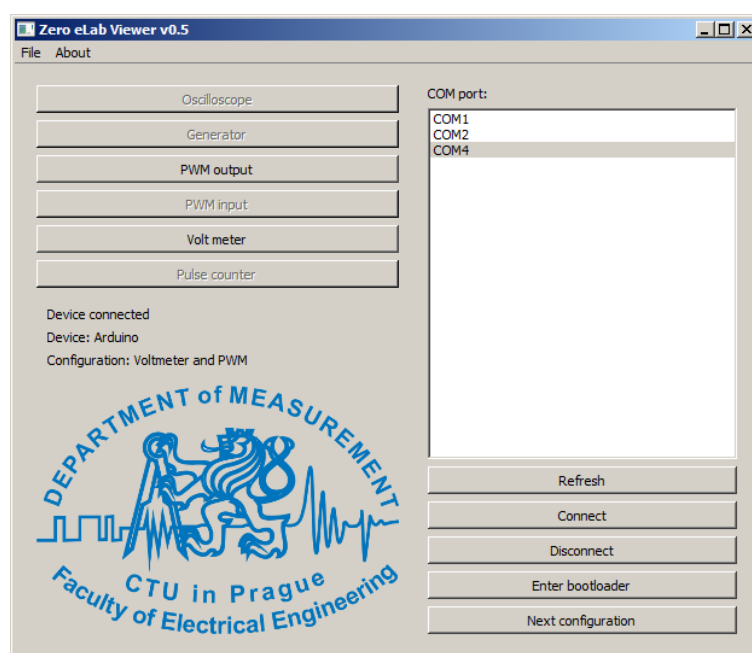
■ 3.1.2 Návod k obsluze přístroje zero_elabviewer s modulem Arduino:

Uvedení do provozu: Z webu² stáhneme archiv `arduino.zip` a rozbalíme ho. Pomocí aplikace Arduino IDE nahrajeme do modulu firmware ze souboru `elab_sb.ino` obsaženého v staženém archivu. Aplikaci stáhneme z webu `embedded.fel.cvut.cz`.

Po spuštění aplikace vybereme z nabídky vlevo sériový port modulu a stiskneme tlačítko **Connect**. Po chvíli by se měla část tlačítek na levé straně okna zablokovat a měla by se objevit data o zařízení v popisu nalevo pod tlačítky (obrázek 3.2). Pokud se tak nestalo, zkusíme stisknout tlačítko **Refresh** a **Connect**, případně si ověříme, zda se připojujeme ke správnému sérovému portu. Pro vyvolání okna konkrétní části přístroje stiskneme příslušné tlačítko nalevo (obrázek 3.3). V ovládacím okně

¹<https://github.com/PaulStoffregen/TimerOne.git>

²<https://drive.google.com/drive/folders/18uE963qJL-c53zRwD5CDa1x5yLt-Fqjm?usp=sharing>



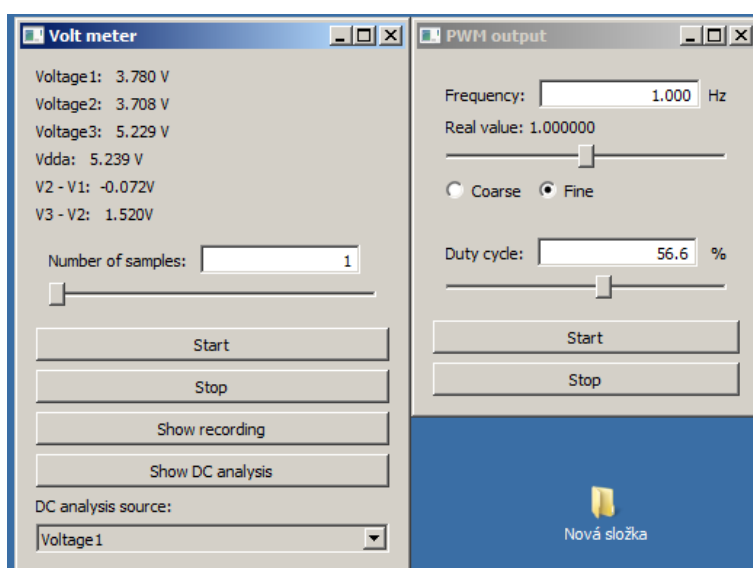
Obr 3.2: Okno programu zero_elabviewer při spojení s modulem

PWM generátoru se nachází dva slidery (tahové ovládací prvky) nastavující frekvenci a střihu, výpisy zadaných hodnot jsou nad prvky. Dalšími prvky v jsou tlačítka **Start** a **Stop** spouštějící a zastavující generátor. Přepínačem **Coarse/Fine** můžeme nastavit rychlost posouvání slideru upravujícího hodnotu frekvence. Hodnotu frekvence můžeme upravit i v textovém poli nahoře. Výpis **Real value** pod textovým polem slouží ke zjištění reálné frekvence generátoru.

V ovládacím okně voltmetru se nachází výpisy jednotlivých hodnot napětí nahoře, textové pole s možností zadání počtu průměrovaných vzorků a slider se stejnou funkcí. Pod těmito ovládacími prvky nalezneme tlačítka **Start** a **Stop**. Tlačítkem **Show recording** zobrazíme okno s vykreslováním naměřených hodnot (obrázek 3.4), jeho ovládání je intuitivní. Klikem pravého tlačítka myši na graf je možné vyvolat nabídku s nastavením zobrazovaných kanálů a případně uložením naměřených dat do souboru.

3.2 Spektrální Analyzátor FFTScope

Realizace spektrálního analyzátoru FFTScope je PC aplikace vypočítavající na jednom kanálu spektrální obraz signálu. Technicky je založen na firmwre a komunikačním protokolu aplikace zero_elabviewer z diplomové práce Adama Berlingera[1]. V současnosti je z této realizace použita i část kódu zajišťující komunikaci s modulem. Realizace zachovává vlastnosti osciloskopu, pozbývá však proti původní verzi generátoru obdélníkového signálu. Přístroj má pevnou délku vstupního bufferu dat 768 vzorků. Použitým modulem je STM32F042F6 s firmwrem realizace zero_elabviewer.



Obr 3.3: Okna jednotlivých částí přístroje zero_elabviewer

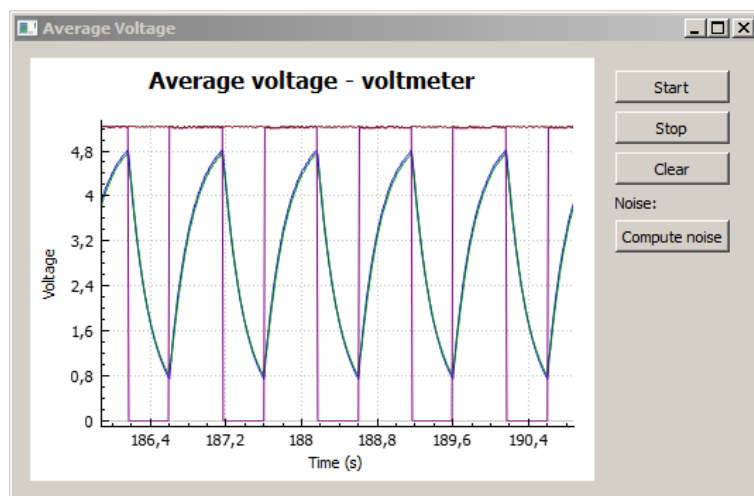
3.2.1 Návod k obsluze přístroje

Po spuštění aplikace (obrázen 3.5) začneme s připojením k modulu, v nabídce na liště **Command** → **Edit Connection** vyvoláme nabídku vybírající COM-port. Používaný COM-port vybereme a potvrdíme **OK**. Pokud byla komunikace úspěšně navázáno, potvrdí nám to vyskakovací okno. Tlačítkem **Run** zapínáme osciloskop, slidery nebo zadávacími okénky lze upravit frekvenci a napětovou úroveň triggeru. Tlačítka **Rising**, **Falling** a **Auto** ovládají trigger. Checkboxy **CH 1 – CH 3** zapínáme jednotlivé kanály v režimu osciloskopu. Checkboxem **Fourier transform** zapínáme výpočet FFT. FFT je realizovaná pomocí knihovny volně dostupně knihovny KISS FFT³. Celá aplikace používá framework Qt, k vykreslení grafu byla použita knihovna QCustomPlot⁴. Tlačítkem **Reset zoom** v nabídce **Command** nastavíme velikost grafu podle zobrazovaných dat.

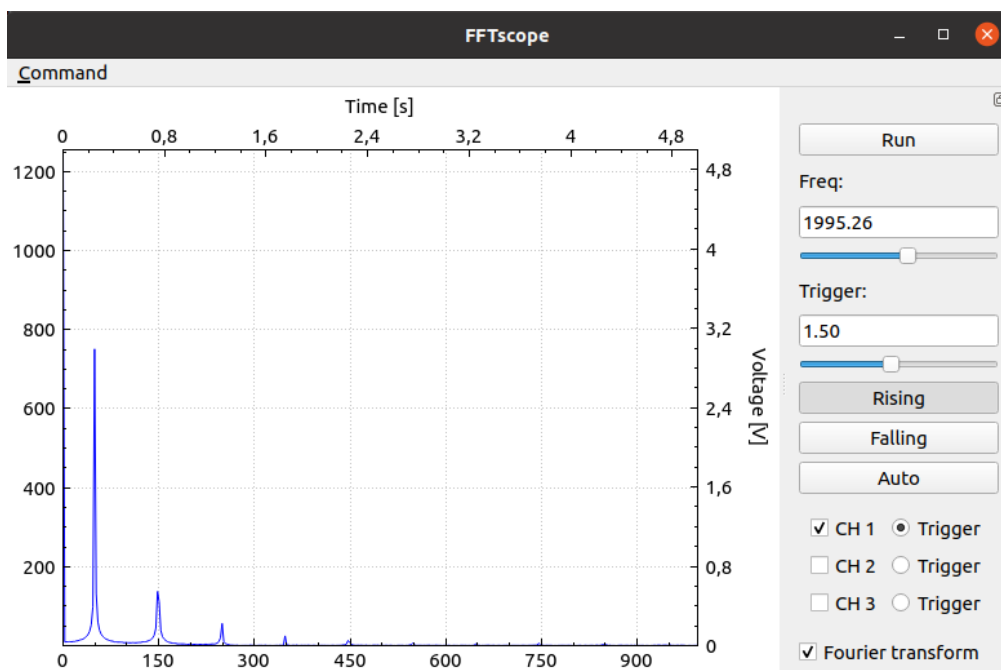
Zatím se jedná o PC aplikaci se základními funkcemi, které by bylo vhodné doplnit, vlastním firmware, vlastním protokolem pro komunikaci, vlastním výpočtem FFT. Aplikace je použitelná pro operační systémy Windows a Linux.

³<https://github.com/mborgerding/kissfft.git>

⁴<https://www.qcustomplot.com/>



Obr 3.4: Okno votmetru se záznamem realizace zero_elabviewer



Obr 3.5: Okno aplikace FFTscope zobrazující spektrální obraz naindukovaného 50Hz signálu ze sítě

Kapitola 4

Balíky STM32duino a ardublock

V této části se zabýváme balíky zjednodušujícími programování mikrokontrolérů STM32. Tyto balíky umožňují programování mikrokontrolérů bez hluboké znalosti problematiky. Návody pro používání balíků mohou být velmi užitečné pro zájemce o elektroniku a programování mikrořadičů, na střeních školách případně v zájmových kroužcích.

STM32duino rozšiřuje prostředí Arduino o možnost programování mikrokontrolérů rodiny STM32. Díky tomu je možné určitou část programů napsaných pro Arduino provozovat i pod platformou STM32 s možností využít i některé její další vlastnosti (např. více pinů, více komunikačních rozhraní, rychlejší periferie ...). V dokumentaci¹ se nachází kompletní seznam modulů, které mají svou definici v balíku STM32duino. Projekt má svou githubovou stránku².

4.0.1 Prostředí Arduino IDE

Arduino IDE je multiplatformní nástroj, který bude v této práci primárně používán pro nahrávání programů do modulů. Arduino IDE se skládá ze základního AVR toolchain (kompilátor jazyka C, knihovny dodané výrobcem, debugger a nástroje pro komunikaci mezi PC a modulem), jazyku Wiring (knihovny projektu Arduino) a desktopové aplikace.

Použití:

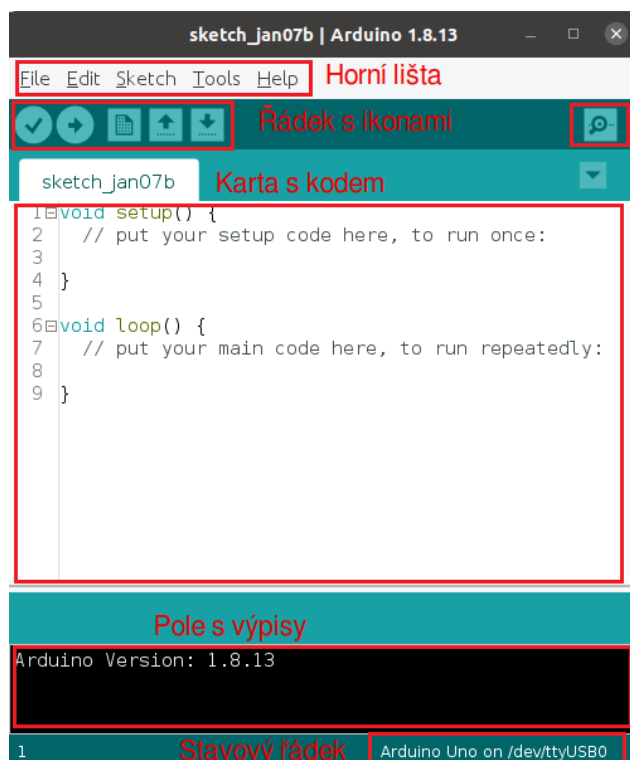
Instalace: Z webu³ stáhneme program pro námi používaný operační systém a dle instrukcí nainstalujeme.

Po instalaci spustíme program Arduino IDE. Sama aplikace má pro přehlednost jen několik málo prvků, rozdělme je na horní lištu, řádek s ikonami, bílou kartu sloužící k zápisu kódu, černé pole s výpisem informací a stavový řádek, viz obrázek 4.1. Pro nahrání programu je nutné program otevřít

¹https://github.com/stm32duino/Arduino_Core_STM32#supported-boards

²<https://github.com/stm32duino/wiki/wiki>

³<https://www.arduino.cc>



Obr 4.1: Prostředí Arduino IDE

(pomocí horní lišty **Soubor** → **Otevřít**). Po nahrání programu do modulu stačí kliknout na ikonu se symbolem "→" na řádku s ikonami. Pokud proběhlo nahrání do modulu správně, vypíše se nám tato skutečnost do černého pole, v opačném případě se vypíše chyba. K nejčastějším chybám patří:

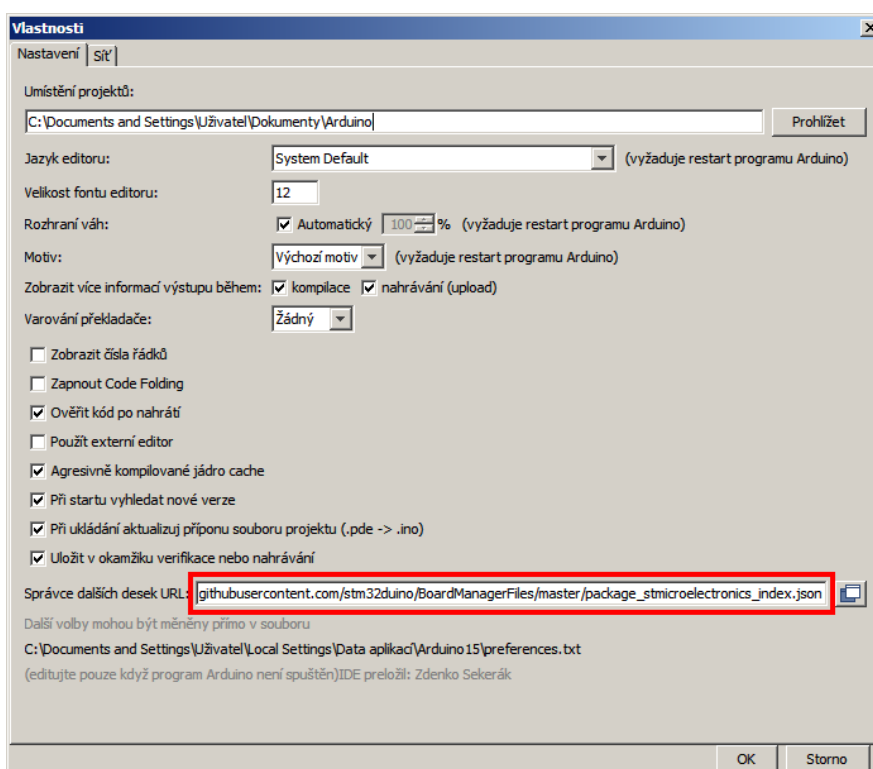
- Chybně nastavený COM-port – ten nastavíme pomocí horní lišty **Nástroje** → **Port**. Pokud nevíme, který port používáme, zkusíme nejdříve vyšší čísla portů. Pozn.: Pro čínské klony Arduino (využívající UART/USB převodník CH340) je obvykle nutné doinstalovat ovladače. Pod Linuxem je také možná nepřístupnost tty portů – je nutné přidat uživatele do skupiny dialout.
- Chybné nastavení typu modulu – na horní liště v nabídce **Nástroje** → **Vývojová deska**: vybereme typ námi používaného modulu.
- Chyba v kódu – přečteme si, co nám prostředí hlásí a chybu opravíme (týká se především námi psaných kódů)

4.1 STM32duino

4.1.1 Instalace do prostředí Arduino IDE:

(celý postup vychází z dokumentace⁴)

1. Spustíme aplikaci Arduino IDE, v nabídce **Soubor** vybereme **Vlastnosti**.
2. V okně **Vlastnosti** vložíme do kolonky **Správce desek** (obrázek 4.2) https://github.com/stm32duino/BoardManagerFiles/raw/master/package_stmicroelectronics_index.json.

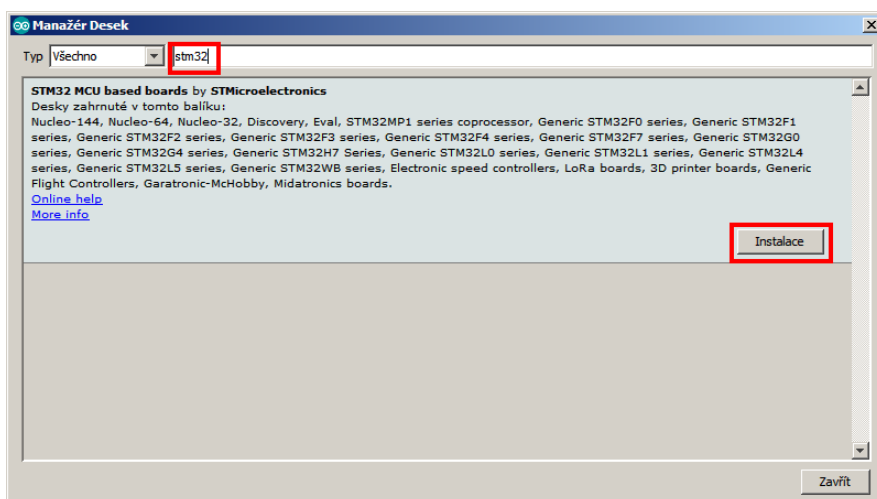


Obr 4.2: Okno **Vlastnosti** aplikace Arduino IDE

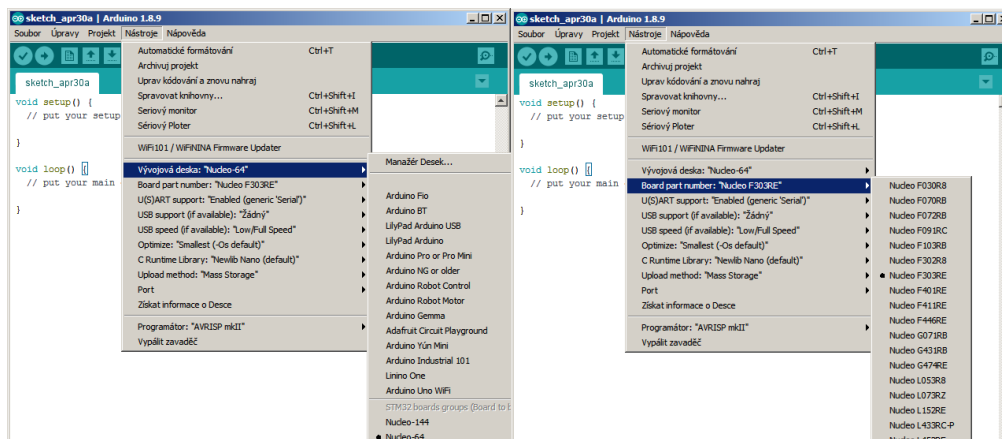
Okno **Vlastnosti** zavřeme pomocí tlačítka **OK**.

3. Otevřeme nabídku **Nástroje** → **Vývojová deska** → **Manažer Desek**. Zde zadáme do vyhledávání STM32 (obrázek 4.3) a nainstalujeme balík **STM32 MCU based boards**. Po instalaci okno **Manažér desek** zavřeme.
4. V nabídce **Nástroje** → **Vývojová deska** → **STM32 Boards** na horní liště už pouze zvolíme nejdříve sérii modulu (obrázek 4.4, vlevo), a poté v nabídce **Nástroje** → **Board part number** konkrétní modul (obrázek 4.4, vpravo).

⁴<https://github.com/stm32duino/wiki/wiki/Getting-Started>

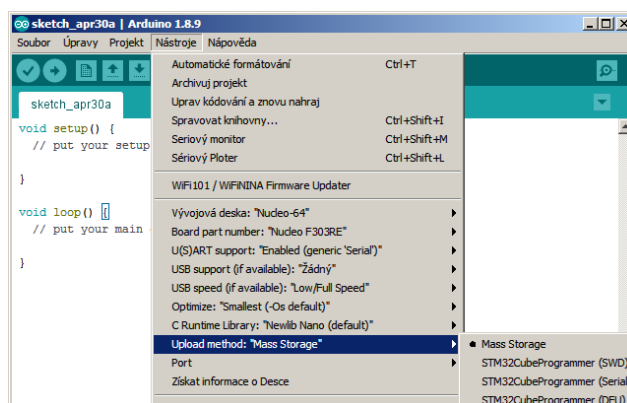


Obr 4.3: Okno Manažér desek



Obr 4.4: Výběr modulu rodiny STM32

V nabídce **Nástroje** → **Upload Method**: zvolíme metodu nahrávání firmware do desky (obrázek 4.5). Uživatelsky nejprívětivější je **Mass Storage**. Možnost **Mass Storage** je dostupná jen pro moduly Nucleo, případně novější moduly Discovery. I přesto se u některých desek stává, že STM32duino nemá správně definován název zařízení – tento problém lze dočasně vyřešit přejmenováním zařízení **Mass Storage** v **Průzkumníku Windows** (např. testovaný modul G031-DISCO byla dočasně přejmenována z DIS_G031J6 na NOD_G031J6). Pokud programujeme modul, který možnost **Mass Storage** neposkytuje, musíme použít jinou metodu a musíme instalovat a nastavit aplikaci STM32CubeProgrammer.



Obr 4.5: Výběr metody nahrávání firmware do modulu

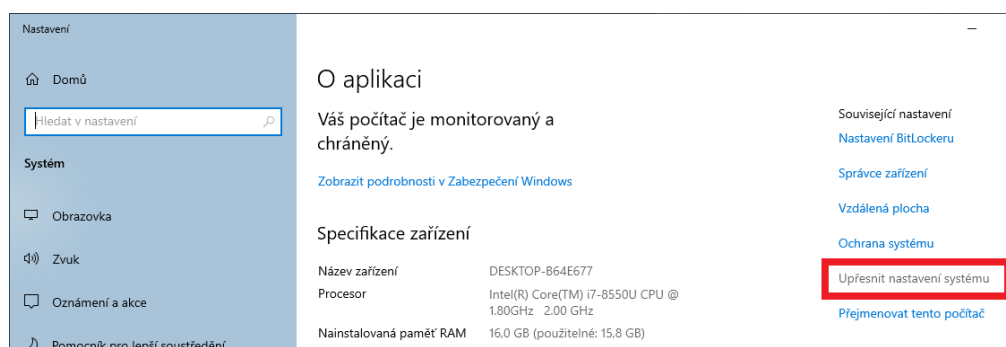
4.1.2 Instalace a nastavení STM32CubeProgrammer

Aplikaci STM32CubeProgrammer potřebujeme, pokud nemáme pro náš modul dostupné nahrávání pomocí **Mass Storage**. Samotná aplikace však pro nahrání programu STM32duina nestačí, musíme ji přidat do proměnné PATH systému.

Stahování a instalace: Z webu⁵ stáhneme aplikaci a nainstalujeme (přímočaré). Do schránky si zkopírujeme cestu, kam program instalujeme.

Přidání aplikace do proměnné PATH systému:

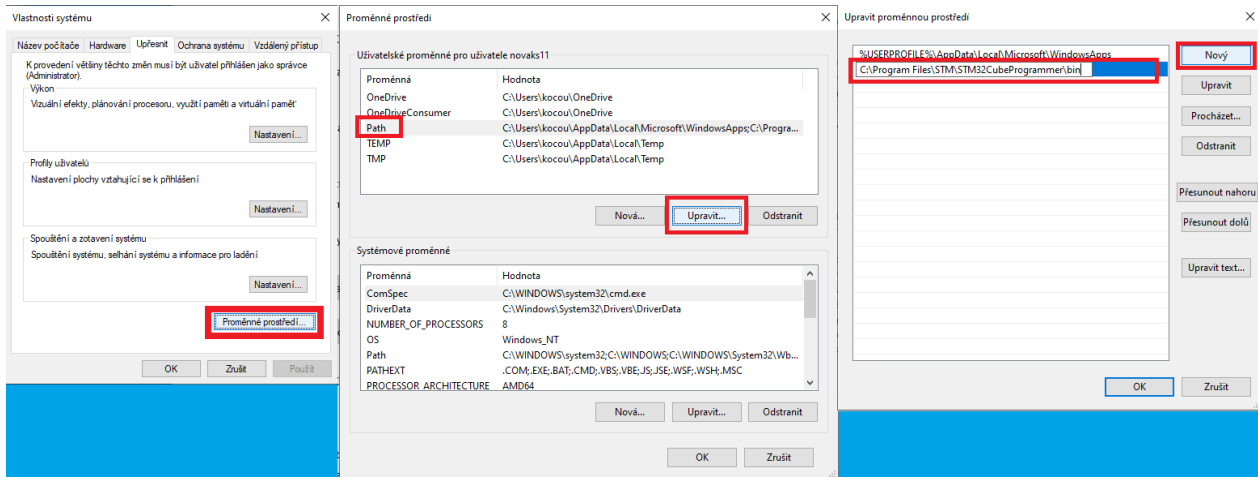
1. Otevřeme **Průzkumník Windows**, pravým tlačítkem myši klikneme na položku **Tento Počítač**, v nabídce vybereme položku **Vlastnosti**.



Obr 4.6: Okno Vlastnosti systému

⁵<https://www.st.com/en/development-tools/stm32cubeprog.html>

2. V okně **Vlastnosti Systému** klikneme na položku **Upřesnit nastavení systému** (obrázek 4.6). V okně **Vlastnosti systému** v kartě **Upřesnit** klikneme na tlačítko **Proměnné prostředí...** (obrázek 4.7, vlevo).



Obř 4.7: Přidání aplikace STM32CubeProgrammer do proměnného prostředí

3. V tabulce okna s názvem **Uživatelské prostředí** (nahore) vybereme proměnou **Path** a klikneme na tlačítko **Upravit...** (obrázek 4.7, uprostřed). V tabulce uprostřed okna vybereme volné místo a klikneme na tlačítko **Nový**. Do volného místa vložíme cestu k STM32CubeProgrammer za kterou připišeme \bin (typicky C:\Program Files\STMMicroelectronics\STM32Cube\STM32CubeProgrammer\bin).
4. Zavřeme všechna okna pomocí **OK** a restartujeme systém.

4.1.3 Programové možnosti při použití balíku STM32duino

Základem balíku STM32duino je toolchain shodný s balíkem STM32Cube. Je tedy možné používat standardní definice jmen pinů (např. PA1) a periférií (např. GPIOA->ODR) s možnostmi jazyka C. Díky tomu je možné přistupovat přímo k registrům periférií, případně volat program napsaný v assembleru.

K základnímu toolchainu je přidán rozšiřující software zajišťující kompatibilitu s Arduino IDE a jazykem Wiring. Díky tomu však ubylo některých funkcionalit, především je definován pouze jediný seriový port pro modul a to vždy na konkrétních dvou pinech. Tato definice je vždy uvedena pro každou variantu modulu; definice se nacházejí v adresářích podle rodin mikrokontrolérů:

C:\Users\<uzivatel>\AppData\Local\.arduino15\packages\STMMicroelectronics\hardware\stm32\2.0.0\variants\

V tabulce 4.1 jsou vypsány piny, na kterých je definován seriový port. Pro desky Nucleo se vždy jedná o port připojený k vestavěnému ST-Linku. Zde uvedené desky Discovery nemají k vestavěnému ST-Linku daný seriový port přiveden.

	Nucleo 64			Nucleo 32		Discovery		Generic	
	F303	F401	L073	G031	L031	F030	G031	F042	G031
RX	PA3	PA3	PA3	PA3	PA15	PA10	PB7	PA3	PA10_R
TX	PA2	PA2	PA2	PA2	PA2_ALT1	PA9	PB6	PA2	PA9_R

Tabulka 4.1: Tabulka serových portů definovaných v balíku STM32duino

Možnost přístupu k registrům periférií mikrokontrolérů: Uvedený zdrojový kód bliká LED bliká LED a po každé změně stavu LED vypíše hodnotu GPIO brány s LED na UART.

```
uint32_t value;

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, LOW);
  value =GPIOA->ODR;//GPIOA
  Serial.println(value);
  delay(1000);
  digitalWrite(LED_BUILTIN, HIGH);
  value =GPIOA->ODR;//GPIOA
  Serial.println(value);
  delay(1000);
}
```

Možnost volání kódu napsaného v assembleru: Uvedený zdrojový kód bliká LED, ale každou změnu stavu LED realizuje pomocí funkce napsané v assembleru.

```
void turn_on_pa5(void){
  __asm__ volatile (//vložit instruke ASM
    "MOV R0, #0x20\n\t"
    "LDR R1, =0x48000000\n\t"
    "LDR R2, [R1,#0x14]\n\t"
    "ORR R2, R0\t\n"
    "STR R2,[R1,#0x14]\n\t"

  );
}

void turn_off_pa5(void){
  __asm__ volatile (
    "MOV R0, #0x20\n\t"
```

```

    "EOR R0, 0xFFFFFFFF\n\t"
    "LDR R1, =0x48000000\n\t"
    "LDR R2, [R1,#0x14]\n\t"
    "AND R2, R0\n\t"
    "STR R2, [R1,#0x14]\n\t"
  );
}

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  turn_on_pa5();
  delay(200);
  turn_off_pa5();
  delay(200);
}

```

Generování sinusového signálu pomocí PWM. Uvedený zdrojový kód generuje obdélníkový signál odpovídající svou střední hodnotou sinusovému signálu. Jedná se o velmi jednoduchou a nenáročnou možnost jak např. rozsvěcet LED zdánlivě sinusovým signálem.

```

unsigned char c=0;
unsigned char sine[256];

void setup() {
  for(int i=0; i<256; ++i) sine[i]=(sin(i*2*PI/255.))+1)*255/2;//přepočítání hodnot sinu
}

void loop() { //smýčka blikající LED s periodou 1.024 s
  analogWrite(D6, sine[c]); // výstup PWM de zadanou střídou
  delay(4); //čekání 4ms
  c++; //
}

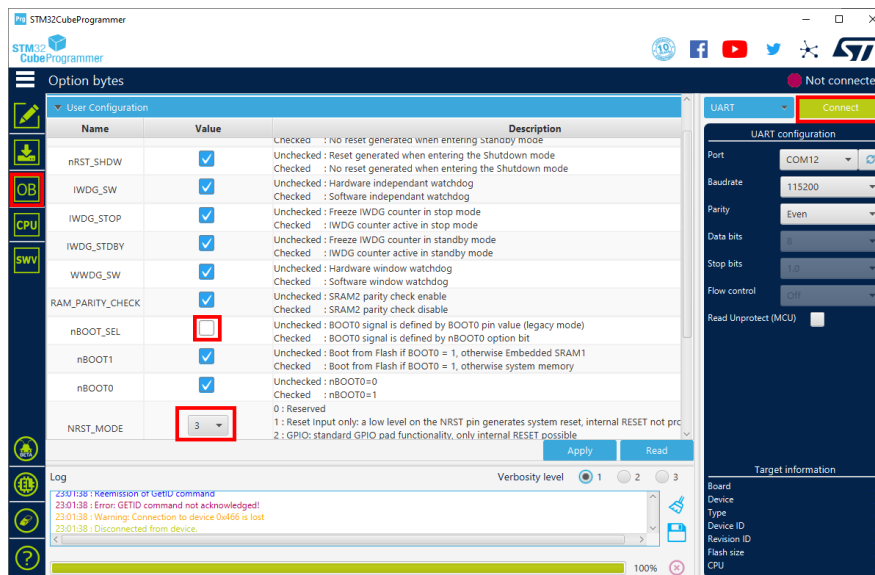
```

4.1.4 Konfigurace STM32G031

Konfigurace modulů STM32G031 má při dodání deaktivovanou funkci NRST. To mimo jiné znamená, že po nahrání firmware do modulu není vyvolán reset. Tuto funkci je možné aktivovat pomocí aplikace **STM32CubeProgrammer** (obrázek 4.8). Pomocí ST-Linku se připojíme k modulu tlačítkem **Connect**, přejdeme do nabídky **Option bytes** tlačítkem **OB** vlevo. Zde změním hodnotu položky **NRST_MODE** na **3**. Tento zásah mimo jiné aktivuje funkci NRST pinu (při přivedení země je vyvolán reset).

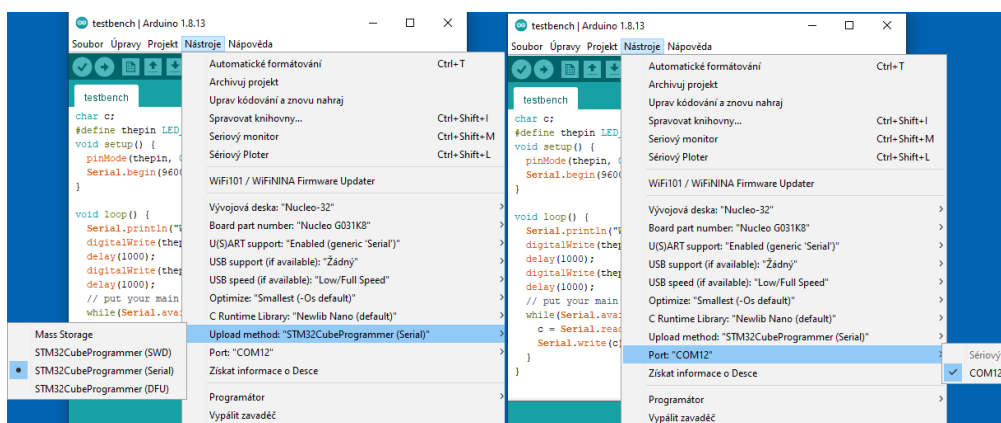
Pro bootování z rozhraní UART je nutné před nahráním dostat modul do stavu BOOT. K tomuto

účelu je vhodné aktivovat funkci pinu **BOOT0**. Při připojení nastavení pinu BOOT0 k napájecímu napětí přejde modul po restartu do režimu BOOT. Funkci pinu BOOT0 aktivujeme tak, že v nabídce **Option bytes** odznačíme položku **nBOOT_SEL**.



Obr 4.8: Nastavení bootování a resetu pomocí STM32CubeProgrammer

Pro nahrání firmware pomocí UART nastavíme aplikaci Arduino IDE podle obrázku 4.9. Zde je zachyceno vlevo nastavení nahrávací metody, vpravo je zachycen výběr seriového portu.



Obr 4.9: Nastavení Arduino IDE pro nahrávání firmware pomocí UART

4.1.5 Offline instalace balíku STM32duino

Balík STM32duino lze v nouzi instalovat tímto postupem (například při nedostupnosti internetového připojení):

1. Z webu stáhneme archiv a15.zip⁶, nakopírujeme ho do cílovho počítače.
2. Na cílovém počítači otevřeme **Průzkumník Windows**, do panelu adresy zadáme `%appdata%`. Přejdeme o adresář výše (měly by se zde nacházet adresáře **Local**, **LocalLow**, **Roaming**).
3. Rozbalíme archiv a15.zip a jeho obsah nakopírujeme do adresáře **Local**, existující soubory přepisujeme.

4.2 Prostředí Ardublock

Ardublock je rozšíření Arduina o grafické programování, prováděné pomocí spojování bloků v okně. Technicky je tato vlastnost realizována při kompilaci převodem bloků okna na kód.

Instalace: Z webu⁷ stáhneme archiv ardublock.zip, rozbalíme ho. Jeho obsah (složku tools) přesuneme do složky, ve které máme uložené projekty (zjistíme v okně Vlastnosti (horní lišta **Soubor** → **Vlastnosti**)). Spustíme prostředí Arduino IDE a Ardublock vyvoláme z horní lišty **Nástroje** → **Ardublock**. Pokud používáme českou verzi Arduino IDE, pak bude i Ardublock alespoň částečně přeložen do češtiny.

Možnosti: Použití prostředí Ardublock se svým uživatelským rozhraním vykazuje podobnosti s podobnými prostředím (např. Espruino). Samotné programování je realizováno přetahováním bloků v prostředí Ardublock – z levé nabídky kategorií objektů do šedého pracovního prostoru. Bloky je možné odstranit přetažením zpět kamkoli do prostoru nabídky. Další funkcionalitou je možnost přidávat komentáře k blokům a bloky duplikovat (pravým tlačítkem myši u konkrétního bloku). Program je nahrán do modulu tlačítkem **Nahrát do Arduina**. Na obrázku 4.10 je zobrazeno prostředí Ardublock obsahující jednoduchý program blikající LED na pinu D13.

Díky svému technickému řešení (převodu bloků na kód) je kompatibilní s balíkem STM32duino (odzkoušeno např. na modulech Nucleo-64 s STM32F303RE a F0_lab).

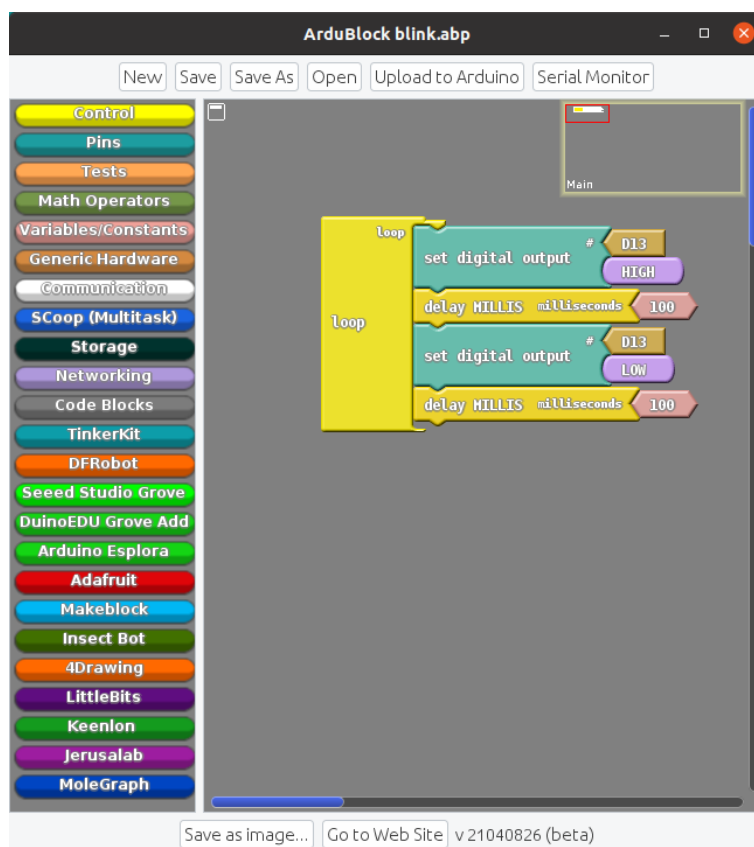
4.2.1 Kombinace balíků STM32duino a Ardublock

Současné použití balíků STM32duino a Ardublock je možností, jak seznámit s programováním moderních procesorových modulů velké množství zájemců s minimálními znalostmi.

Problémem balíku Ardublock je, že při nastavování pinu není možné zadat hodnotu přímo (např. PA5). Balík STM32duino s touto možností počítá a předefinovává piny jednotlivých modulů STM32 na ekvivalentní na modulech Arduino. Toto přiřazení je vždy definováno v datech balíku STM32duino. V tabulce 4.2 je uvedeno přiřazení pinů u desek používaných na katedře měření FEL.

⁶https://drive.google.com/file/d/1Iy1aCXgrLPaK_XV2LVZtj24sijHWAwbh/view?usp=sharing

⁷??



Obr 4.10: Ukázka prostředí Ardublock

Arduino	Nucleo 64			Nucleo 32		Discovery		Generic	
	F303	F401	L073	G031	L031	F030	G031	F042	G031
D0	PA3	–	–	PB7	PA10	PC13	–	–	–
D1	PA2	–	–	PB6	PA9	PC14	–	–	–
D2	PA10	PA10	PA10	PA15	PA12	PC15	–	–	–
D3	PB3	PB3	PB3	–	–	PF0	PA8	–	PA8
D4	PB5	PB5	PB5	PA10	PB7	PF1	PA9	–	–
D5	PB4	PB4	PB4	PA9	PB6	–	PA10	–	–
D6	PB10	PB10	PB10	–	–	–	–	–	–
D7	PA8	PA8	PA8	–	PC14	–	–	–	–
D8	PA9	PA9	PA9	PB8	PC15	–	–	PA8	PA15
D9	PC7	PC7	PC7	PA8	PA8	–	–	PA9	–
D10	PB6	PB6	PB6	PB9	PA11	–	PA15	PA10	–
D11	PA7	–	–	PB5	PB5	–	–	PA11	PB5
D12	PA6	–	–	PB4	PB4	–	–	PA12	PB6
D13	PA5	–	–	PB3	PB3	PF4	PB5	PA13	–
D14	PB9	PB9	PB9	–	–	PF5	PB6	PA14	PB8
D15	PB8	PB8	PB8	–	–	–	–	PA15	PB9
A0	PA0	PA0	PA0	PA0	PA0	PC0	PA0	PA0	PA0
A1	PA1	PA1	PA1	PA1	PA1	PC1	PA1	PA1	PA1
A2	PA4	PA4	PA4	PA4	PA3	PC2	PA2	PA2	PA2
A3	PB0	PB0	PB0	PA5	PA4	PC3	PA11	PA3	PA11
A4	PC1	PC1	PC1	PA12	PA5	PA0	PA12	PA4	PA12
A5	PC0	PC0	PC0	PA11	PA6	PA1	PA13	PA5	PA13
A6	PA7	PA7	PA7	PA6	PA7	PA2	PA14	PA6	PA14
A7	PA6	PA6	PA6	PA7	PA2	PA3	PB0	PA7	PB0
A8	PC2	PC2	PC2	–	PB0	PA4	PB1	PB0	PB1
A9	PC3	PC3	PC3	PB1	PB1	PA5	PB7	PB1	PB7
A10	PC5	PC5	PC5	PB0	–	PA6	–	–	–
A11	PB11	PB1	PC4	PB2	–	PA7	–	–	–

Tabulka 4.2: Tabulka přiřazení jednotlivých pinů pinu modulu Arduino pro testované moduly



Kapitola 5

Zhodnocení

Koncepce práce byla naznačena už v zadání do tří částí. První část se zabývala volně dostupnými realizacemi měřících přístrojů, jejich popisem, testováním, použitím. Druhá část se zabývala vývojem vlastních realizací. Třetí část se zabývala softwarovými balíky zjednodušující programování mikrořadičů.

Každá část má potenciál pro další zpracování, domnívám se však, že práce může pomoci v základní orientaci v problematice a může sloužit jako zdroj pro další využití. Druhá část mohla být zpracována důkladněji, zvláště realizace spektrálního analyzátoru. První a třetí část byla podle mého názoru zpracována dostatečně.

Doufám, že práce bude užitečným zdrojem, který může být využit ve standardní výuce stejně jako při mimořádných situacích, kdy je nutné volit alternativní možnosti výuky.



Literatura

- [1] Berlinger, Adam *Implementace přístrojových funkcí s využitím mikrořadičů STM32*. Praha, 2016. Diplomová práce. České vysoké učení technické v Praze, Fakulta Elektrotechnická, katedra řídicí techniky. Vedoucí práce doc. Jan Fischer Ing., CSc.



Příloha A

Obsah přiloženého CD

- Bakalářská práce ve formátu PDF
- Zdrojový kód firmware zero_elabviewer pro Arduino
- Zdrojový kód PC aplikace FFTScope se spustitelnými soubory pro Windows a Linux