

Diplomová práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra měření

## Firmware pro měřicí přístroj s mikrořadičem STM32G431

**Bc. Petr David**

Vedoucí: doc. Ing. Jan Fischer, CSc.  
Květen 2023



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **David**

Jméno: **Petr**

Osobní číslo: **420110**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávací katedra/ústav: **Katedra měření**

Studijní program: **Kybernetika a robotika**

Studijní obor: **Kybernetika a robotika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Firmware pro měřicí přístroj s mikrořadičem STM32G431**

Název diplomové práce anglicky:

**Firmware for measuring instrument based on microcontroller STM32G431**

Pokyny pro vypracování:

V návaznosti na přístroj vyvinutý v rámci DP [1] vytvořte firmware pro mikrořadič STM32G431 tak, aby jej ve spolupráci s PC aplikací Zero eLab Viewer bylo možno využít jako jednoduchý, avšak komplexní měřicí přístroj pro výukové účely. V případě potřeby proveďte nutné úpravy PC aplikace. Přístroj bude zahrnovat funkce osciloskopu i se zobrazením průběhů logických kanálů, dále funkce impulsního a signálového generátoru, čítače a voltmetru se záznamem. Při návrhu firmware můžete též využít vhodné bloky vytvořené v rámci prací [2] a [3]. Výsledný přístroj otestujte a ověřte jeho parametry.

Seznam doporučené literatury:

- [1] Berlinger, A.: „Implementace přístrojových funkcí mikrořadiči STM32“, diplomová práce ČVUT – FEL, 2016
- [2] Cejp M.: „Virtuální přístroj s mikrořadičem pro analýzu signálu v modulační doméně“, bakalářská práce, ČVUT – FEL, 2017
- [3] Dujava J.: „Softwarově definované osciloskopy s terminálovým rozhraním“, diplomová práce ČVUT – FEL, 2022
- [4] Cejp M.: „Virtuální přístroj s mikrořadičem pro analýzu signálu v modulační doméně“, bakalářská práce, ČVUT – FEL, 2017

Jméno a pracoviště vedoucí(ho) diplomové práce:

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

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

Datum zadání diplomové práce: **06.09.2022**

Termín odevzdání diplomové práce: \_\_\_\_\_

Platnost zadání diplomové práce:

**do konce letního semestru 2023/2024**

\_\_\_\_\_  
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

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

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta





## Poděkování

Děkuji ČVUT, že mi je tak dobrou *alma mater*.





## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 10. května 2023







## Abstrakt

Abstrakt v češtině

**Klíčová slova:** slovo, klíč

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



## Abstract

Abstract in English

**Keywords:** word, key

**Title translation:** Firmware for measuring instrument based on microcontroller STM32G431



## Obsah

<b>1 Úvod</b>	<b>1</b>
<b>2 Rozbor</b>	<b>3</b>
<b>3 Realizace FW</b>	<b>5</b>
3.1 Rozpoznání frekvence externího krystalu HSE .....	5
3.2 Struktura FW .....	8
3.3 Realizace funkce Voltmetru .....	10
3.4 Realizace měření frekvence .....	10
3.5 Realizace funkce generátoru .....	10
3.6 Realizace osciloskopu .....	12
3.7 Logický analyzátor .....	17
<b>4 Ověření funkčnosti</b>	<b>19</b>

<b>5 Zhodnocení</b>	<b>21</b>
<b>6 Závěr</b>	<b>23</b>
<b>A Literatura</b>	<b>25</b>
<b>B Zadání práce</b>	<b>27</b>



## Obrázky

3.1 Zkreslení průběhu měřeného signálu v důsledku nestability HSI převzato z [2] .....	5
3.2 Vstupy čítače TIM16 .....	6
3.3 Vyobrazení interního PLL bloku. Převzato z.....	7
3.4 Postup změny zdroje hodinového signálu.....	8
3.5 Kategorie zdrojového kodu a jejich posloupnost závislosti .....	8
3.6 Hlavní okno aplikace Zero eLab Viewer zobrazující aktivní moduly dané konfigurace .....	9
3.7 .....	11
3.8 .....	11
3.9 2-fázové triggerování na nástupnou hranu signálu s využitím AWDG- převzato z [2] .....	13
3.10 Vzájemné propojení dvou čítačů .....	13
3.11 Převzato z [4] .....	15
3.12 Dual Regular simultaneous mod převzato z [4].....	16

3.13 Zpoždění kanálu 2 a 4 za kanálem 1 a 3 při použití Dual simultaneous modu a stroboskopickém vzorkování s ekvivalentní vzorkovací frekvencí 78MHz .....	16
3.14 převzato z [6] .....	17
3.15 Efekt nedodržení maximálního vstupního odporu .....	17



## Tabulky

3.1 .....	6
3.2 Srovnání paměťové náročnosti jednoduchého programu s využitím různých knihoven.....	10
3.3 Přehled jednotlivých analogových kanálů a jejich na dostupnosti na ADC převodnících .....	14







# Kapitola 1

## Úvod





## Kapitola 2

### Rozbor

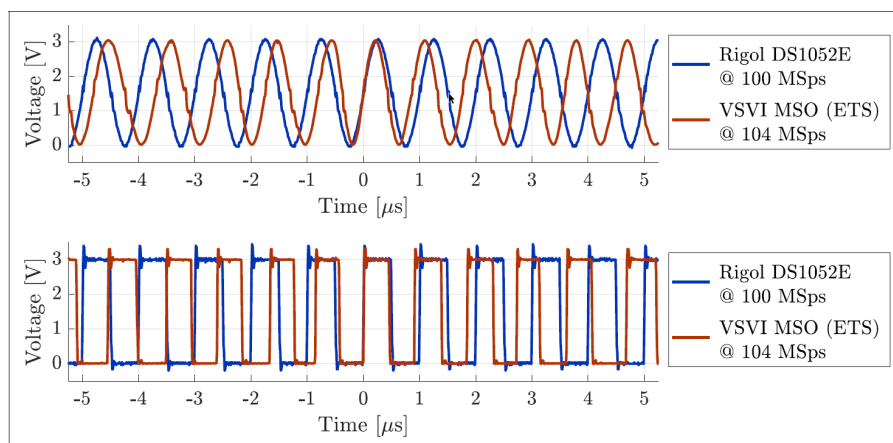


## Kapitola 3

### Realizace FW

#### 3.1 Rozpoznání frekvence externího krystalu HSE

Pro chod mikrokontroléru je zapotřebí zdroj hodinového signálu pro generování systémových hodin (System Core clock) dále jen SYSCLK. Jako základní varianta zdroje hodinového signálu se používá interní vysoko-rychlostní oscilátor (HSI), jehož výstupní frekvence ve srovnání s externími zdroji hodinového signálu vykazuje řádově vyšší nepřesnost a vyšší závislost na změnách teplot viz srovnávací tabulka 3.1. Tento rozdíl je pak obzvláště podstatný při realizaci funkce osciloskopu v režimu vzorkování v ekvivalentním čase (ETS), kde dochází k výraznému zkreslení měřeného signálu viz obrázek 3.1. Na tomto obrázku je zobrazen zkreslený záznam signálu s G431 využívajícím HSI jako zdroj hodinového signálu a druhá stopa je měřena osciloskopem Rigol DS1052E jehož přesnost vzorkovací frekvence je  $\pm 0.005\%$  [3]. Z obrázku je zřejmá vhodnost použití zdroje hodinového signálu s vyšší přesností než vykazuje HSI.



**Obr. 3.1:** Zkreslení průběhu měřeného signálu v důsledku nestability HSI převzato z [2]

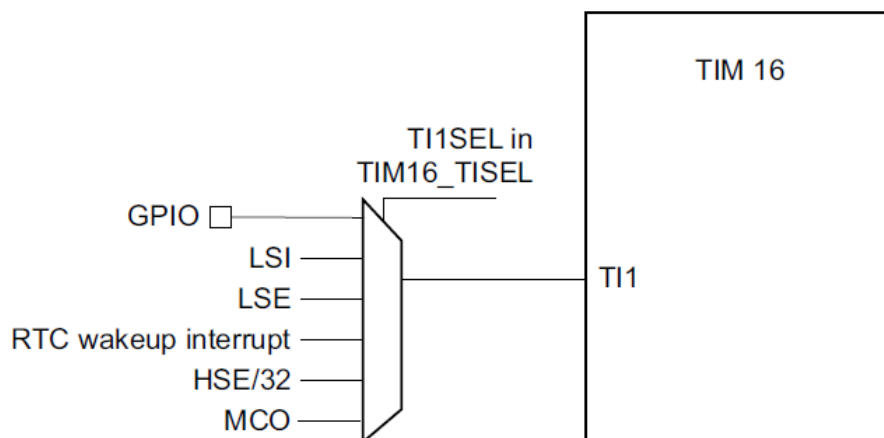
Jako zdroj vysokorychlostního externě získaného hodinového signálu(HSE) lze buď využít krystalu buzeného pomocí MCU(HSE crystal) nebo jiného externího zdroje signálu(HSE bypass). Případný externí signál musí pak splňovat nějaké podmínky a to například: pro STM32G431 musí být v rozsahu 4-48MHz a mít střihu 40-60%. Ve výuce laboratorních měření na katedře měření jsou k dispozici krystaly různých výstupních frekvencí převážně pak 8 MHz, 12MHz a 16MHz. Pro účely co nejflexibilnějšího laboratorního přístroje se zdálo účelné naprogramovat firmware pro použití s různými oscilátory tohoto typu. Tedy aby funkce FW nebyla závislá na přítomnosti krystalu ani jeho výstupní frekvenci. Toho bylo docíleno změřením výstupní frekvence oscilátoru a nastavení výsledné frekvence systémových hodin pomocí interního obvodu fázového závěsu(PLL), tak aby výsledná frekvence SYSCCLK nebyla na použitém krystalu závislá.

	$\Delta f$	$\Delta$
STM32G431 HSI 16 MHz	$\pm 1\%$	$\pm 1\%$
Adafruit krystal 16 MHz	$\pm 0.003\%$	$\pm 0.005\%$

Tab. 3.1:

### 3.1.1 Měření frekvence HSE

Existují různé způsoby měření frekvence externího hodinového signálu, ale jako nejúčelnější se v tomto případě zdálo použití čítače v režimu "Input capture"(IC) a měření délky periody externího signálu. Na rodině mikrokontrolérů STM32G4 mají čítače TIM16 a TIM17 možnost interně přivést HSE již se sníženou frekvencí. Frekvence HSE je totiž ještě před přivedením na vstup čítače zpracovaná obvodem, který frekvenci 32krát sníží. Pro tento vstup čítače se sníženou frekvencí se pak používá označení HSE32 jako je vidět na obrázku 3.2 z dokumentace.



Obr. 3.2: Vstupy dostupné na kanálu číslo 1 čítače TIM16. Převzato z[4]

Měření periody signálu HSE32 probíhá potom tak, že měříme počet cyklů čítače mezi jednotlivými náběžnými hranami nebo sestupnými hranami. Tento počet cyklů nám pak určuje poměr mezi frekvencí externího hodinového signálu na vstupu  $f_{IN}$  a hodinového signálu, který pro svůj chod využívá periferie čítače  $f_{TIM}$ . Pro správné měření je tedy podstatné, aby interní hodinový signál čítače byl výrazně

vyšší než frekvence na měřeném vstupu. Pro získání přesnějšího odhadu vstupní frekvence můžeme zaznamenat více hodnot po sobě a ty zprůměrovat. Frekvence vstupu je poté rovna:

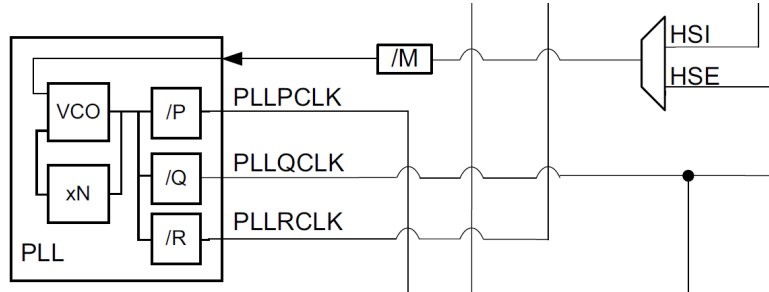
$$f_{IN} = \frac{f_{TIM}}{N_p} \quad (3.1)$$

Pro určení frekvence HSE pak ještě musíme získanou hodnotu vynásobit 32:

$$f_{HSE} = 32 \cdot f_{IN} \quad (3.2)$$

### 3.1.2 Využití PLL

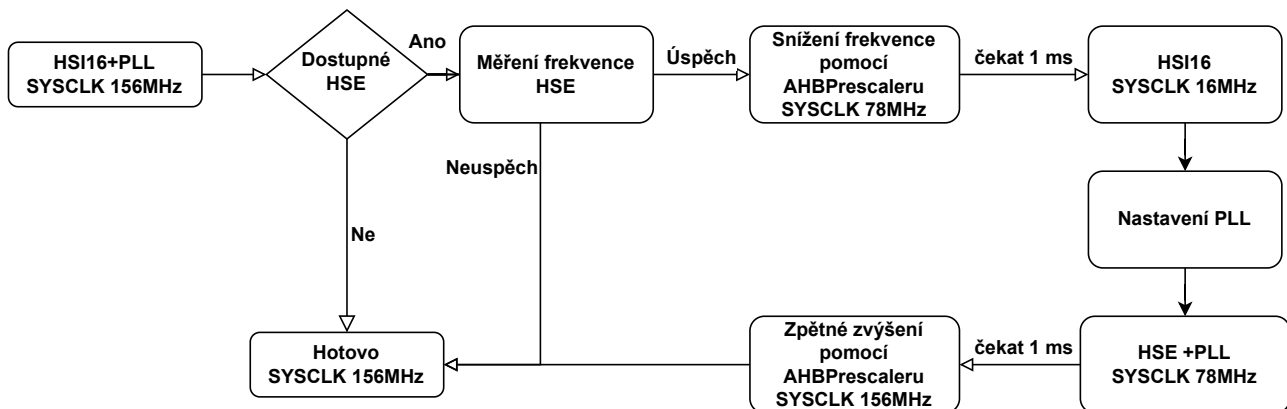
Obvod PLL je další z možných zdrojů hodinového signálu systémových hodin, který má navíc programovatelné dělení a násobení vstupní frekvence. Jako vstup pak lze použít HSI nebo HSE ve stanovené rozsahu. Například 2.66-16MHz pro stm32G431[5]. Dle obrázku 3.3 lze vidět, jak vstupní hodinový signál vstupující do PLL bloku nejdříve prochází přes děličku signálu M, dále se hodinový signál násobí N a tento signál jde pak na 3 různé výstupy s vlastními děličkami. Dále v kapitole o ADC ukazují výhodu této možnosti více výstupů z obvodu PLL, díky které vstupní hodiny ADC nemusí být závislé na frekvenci systémových hodin.



Obr. 3.3: Vyobrazení interního PLL bloku. Převzato z

### 3.1.3 Změna zdroje hodinového signálu systémových hodin

Na obrázku 3.4 je popsán postup změny zdroje SYSCLK. Při změně vstupního signálu PLL nelze PLL používat, tedy je nejdříve zapotřebí přepnout systémové hodiny na interní oscilátor 16MHz. Dle doporučení v [4] je při velkých rozdílech frekvencí mezi výstupem PLL (SYSCLK > 80MHz) zapotřebí přidat mezikrok s využitím AHB předděličky hodinového signálu systémových hodin. hodnota předděličky se nastavuje RCC\_CFGR registru. Díky tomu například ve svém programu zmenším frekvenci systémových na polovinu tedy 78 MHz a pak až nastavuji jako zdroj HSI146. Doporučená doba setrvání v tomto mezikroku je alespoň 1μs. V mém řešení program čeká 1ms s využitím připravených funkcí obsahujícím čekání v jednotkách ms.



Obr. 3.4: Postup změny zdroje hodinového signálu

## 3.2 Struktura FW

Jak vyplývá ze zadání, firmware vznikl v návaznosti na existující přístroj využívající stávající PC aplikací Zero eLab Viewer a již touto skutečností byl návrh firmware mikrokontroléru částečně vymezen. V zájmu zpětné kompatibility by totiž úpravy aplikace neměly ohrozit fungování předchozích implementací na jiných mikrokontrolérech, kterých je již celá řada. Především tedy komunikační protokol, který i když je v některých případech limitující, nemohl být upraven. Dále pak aplikace určuje, jaké softwarově definované přístroje lze implementovat a jaké budou mít možnosti ovládání či nastavení. Dále jsem se při návrhu zaměřil na tyto body

### ■ Samostatná využitelnost jednotlivých přístrojových bloků

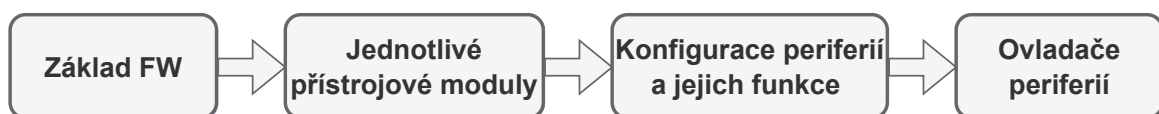
Rozdělit implementaci jednotlivých přístrojů do samostatně fungujících bloků je výhodné z 2 důvodů. Zaprvé lze tak jednodušeji výsledný FW přizpůsobit pro různé mikrokontroléry v závislosti na dostupných perifériích a velikosti FLASH paměti. Druhým důvodem je pak možnost použití kódu i v jiných aplikacích.

### ■ Jednoduchá záměna HW prostředků

Bylo žádoucí aby vznikající FW byl flexibilní co se týče použitých HW prostředků a tedy bylo jednoduché upravovat například použité piny, DMA kanály či čítače. Toto dále zjednodušuje další adaptaci firmware pro jiné MCU.

### ■ Kompaktibilita s generátorem inicializačního kódu

Nástroj STM32CubeMX velmi usnadňuje začátek vytváření FW díky využití grafického rozhraní pro definování počáteční konfigurace MCU. Pokud se zachová struktura generovaného kódu, lze pak nástroj opětovně využít v případě, že chceme jednoduše upravit HW konfiguraci.

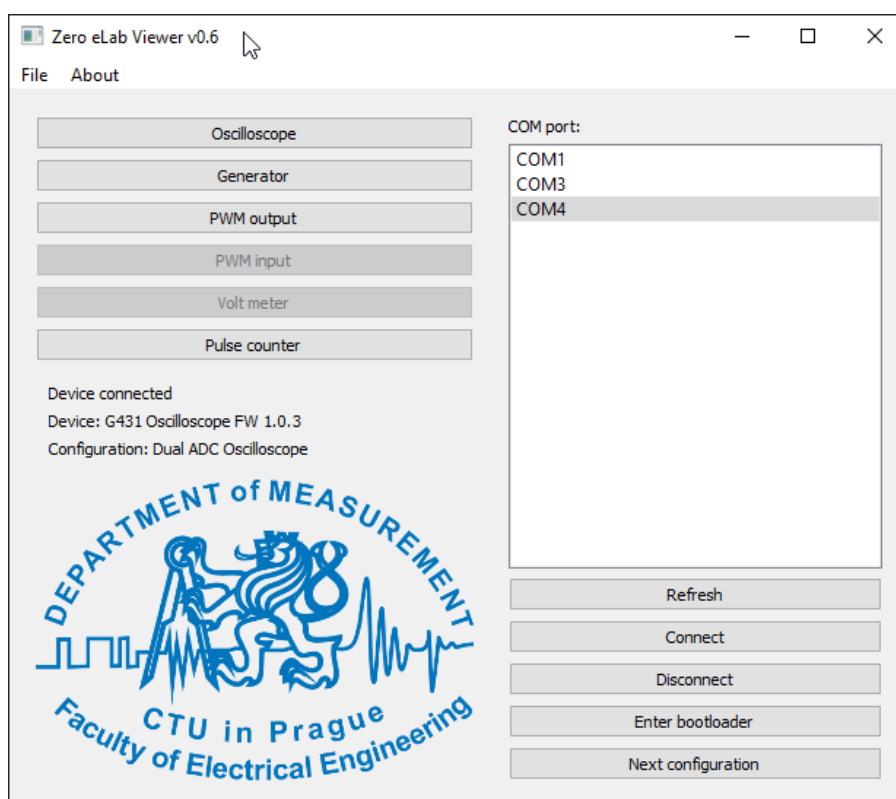


Obr. 3.5: Kategorie zdrojového kódu a jejich posloupnost závislosti



### 3.2.1 Využití možnosti přepínání mezi konfiguracemi

Především z důvodů omezených hardwarových prostředků na různých mikrokontrolérech je využito možnosti přepínání různých přístrojových konfigurací. Konfigurací v tomto případě myslíme set nástrojů, které mohou být použity současně. Přístrojové moduly pak mohou sdílet periferie za předpokladu, že v dané konfiguraci je aktivní pouze jeden z těch modulů, který má přístup k dané periférii. Příkladem může být sdílení ADC převodníku modulem Voltmetru a modulem Osciloskopu, které pak tedy nemohou být aktivní zároveň. Jako je vidět na obrázku 3.6 hlavní okna PC aplikace.



Obr. 3.6: Hlavní okno aplikace Zero eLab Viewer zobrazující aktivní moduly dané konfigurace

### 3.2.2 Využití LL ovladačů

Jedná se o hardwarově orientovanou knihovnu dodávanou pro STM32 mikrokontrolery. Její využití je velmi podobné využití CMSIS ovladačů s tím rozdílem, že nabízí určité rozšířené možnosti portovatelnosti mezi jednotlivými rodinami mikrokontrolerů a sadu rozšiřujících API pro zjednodušení implementace některých úkonů jako je například inicializace periférií. K použití této knihovny je oproti pravděpodobně známější knihovně HAL zapotřebí znalost jednotlivých periférií, jelikož velká část definovaných funkcí je pouze jednořádková modifikace registrů bez kontrol vstupních parametřů a uživatel tedy musí být více obeznámen s tím co dělá. Výhodou oproti HAL je výrazně menší paměťová náročnost.

Pro vytvoření představy o rozdílnosti paměťové náročnosti jednotlivých knihoven jsem vytvořil pomocí STM32CubeMX dva minimální projekty, jejichž úkolem bylo nastavení systémových hodin blikání jednou LED. V prvním případě byly periferie RCC(Reset and clock control) a GPIO inicializovány prostřednictvím HAL knihovny a v druhém případě pomocí LL ovladačů a byly použity odpovídající zpožďovací funkce společně s přepínáním výstupu pinu. Výsledkem bylo, že po zkompilování verze s HAL knihovnou využívala asi o 86 procent více FLASH.

	FLASH	RAM
LL ovladače	3,04 kB	1,53 kB
HAL ovladače	5,66 kB	1,55 kB

**Tab. 3.2:** Srovnání paměťové náročnosti jednoduchého programu s využitím různých knihoven

### ■ 3.2.3 Core coupled memory CCM SRAM

## ■ 3.3 Realizace funkce Voltmetru

### ■ 3.3.1 Nejistota měření

## ■ 3.4 Realizace měření frekvence

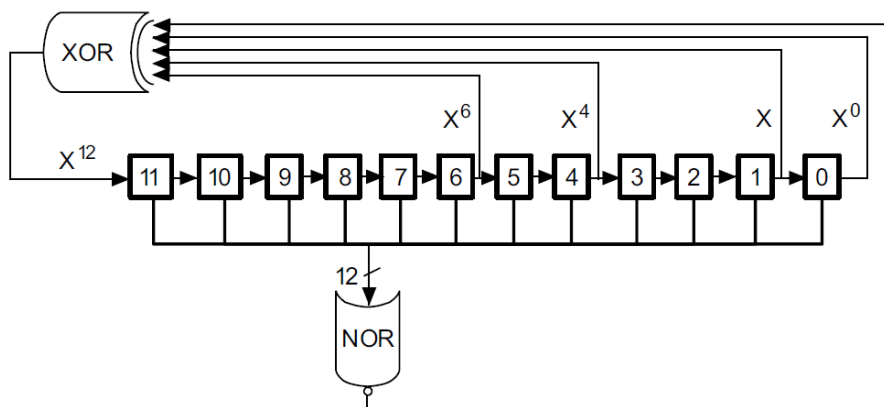
### ■ 3.4.1 Nejistota měření

## ■ 3.5 Realizace funkce generátoru

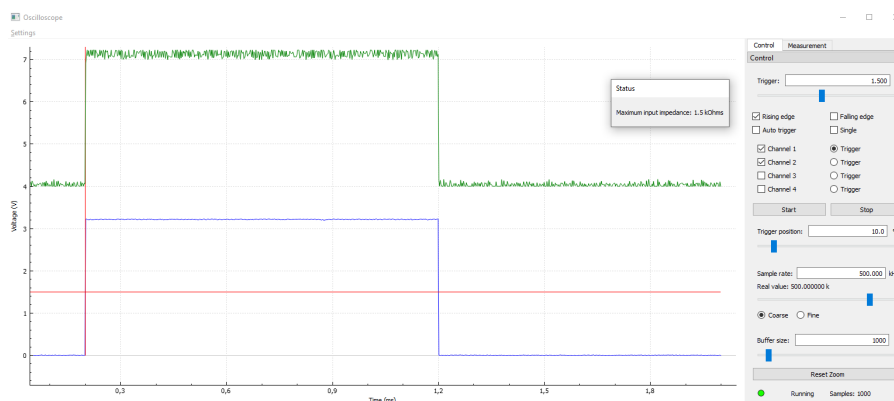
### ■ 3.5.1 Generování signálu

### ■ Využití CORDIC

CORDIC(COordinate ROTation DIgital Computer) je zkratka pro algoritmus pro vyhodnocování trigonometrických a hyperbolických funkcí



Obr. 3.7:



Obr. 3.8:

### 3.5.2 Generování šumu

DAC převodník přítomný u MCU rodiny STM32G4 je implementovaný LFSR registr umožňující generování pseudo-šumu o nastavitelném offsetu a amplitudě. Generování tohoto-šumu funguje na základě pravidelného výpočtu hodnoty registru dle algoritmu popsaného v obrázku 3.7 s tím, že na začátku a pokaždé když se v registru objeví nulová hodnota, tak je vložena do registru hodnota 1, aby nedošlo k zablokování výpočtu. Dále je offset a amplituda nastavována maskováním bitů tohoto registru. Takto produkovaný šum má plochou spektrální distribuci a může být s určitou rezervou považován za bílý šum, s tím rozdílem, že narozdíl od pravého bílého šumu nemá Gaussovské(normální) rozdělení, ale rovnoměrné.

Jedním ze způsobu využití je ve spojitosti s průměrováním vzorků pro efektivní zvýšení rozlišení ADC převodníku.

## 3.6 Realizace osciloskopu

Modul osciloskopu je nejkomplexnější z modulů implementovaných v této práci a též využívající nejvíce HW prostředků. K implementaci na STM32G431 jsem využil přítomnosti 2 ADC převodníků, triggerování pomocí analog watchdog (AWDG) funkce přítomných ADC převodníků, DMA řadiče a 2 vzájemně propojených zřetěžených čítačů.

### 3.6.1 Řešení triggerování osciloskopu

Určení okamžiku náběžné nebo sestupné hrany vstupního signálu je podstatnou funkcí osciloskopu pro zobrazování jak přechodných jevů tak periodických průběhů signálu. Existuje více přístupů, jak takovou funkci implementovat. Jednou z nejjednodušších variant je SW orientované řešení, kdy jsou naměřená data vždy cyklicky kontrolována, zda došlo k překročení zvolené napěťové úrovně a popřípadě zastavit další sběr dat. Takové řešení bylo například použito v původní variantě FW pro STM32F042 v práci [1]. Nevýhodou tohoto řešení je vyšší výpočetní náročnost ve srovnání s více HW zaměřenými řešeními, jaká jsou například využití komparátorů nebo funkce AWDG, kterou disponují ADC převodníky na mikrokontrolérech STM32G4 a kterou jsem se rozhodl uplatnit v této práci já.

AWDG umožňuje vyvolat přerušení v momentě, kdy se daném kanálu objeví napětí mimo nastavené rozmezí. Abychom mohli určit okamžik, kdy došlo k poklesu pod určitou úroveň (sestupná hrana) nebo naopak k překročení napěťové úrovně (náběžná hrana) je potřeba využití AWDG ve dvou fázích znázorněných na obrázku 3.9.

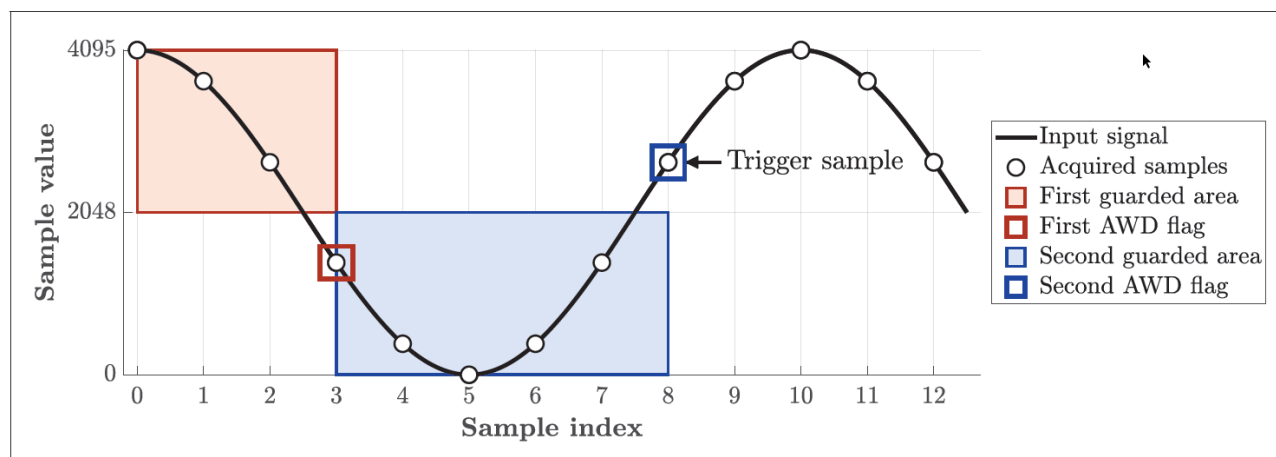
#### ■ Fáze 1

Při hledání okamžiku nástupní hrany v první fázi nejdříve nastavíme rozmezí nad střeženou napěťovou úrovní. Jakmile se napětí klesne mimo rozmezí dojde k přerušení a přesuneme se do fáze 2

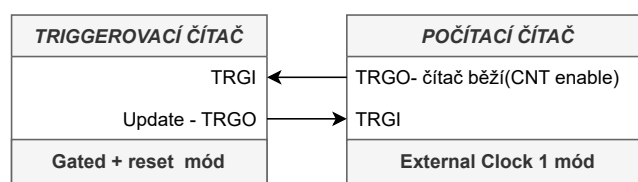
#### ■ Fáze 2

V této fázi víme že signál je pod nastavenou napěťovou úrovní. Tedy nastavíme nové střežené rozmezí napětí a víme, že jakmile dojde k přerušení, že nastal okamžik, který chceme označit jako moment nástupní hrany a uložit si číslo vzorku, kdy k tomuto došlo.

Po vyhodnocení, že došlo k události triggeru, potřebujeme zjistit číslo vzorku, kdy moment nastal a nastavit odměření zbývajících vzorků signálů, tak aby nové vzorky zapisované do kruhového bufferu nepřepsaly data vstupního signálu před touto událostí. Zde jsem využil dvou vzájemně propojených čítačů. Funkce prvního čítače je triggerování ADC převodníku a funkce druhého čítače je počítání odměřených vzorků a poté po triggeru zastavení triggerování po daném počtu vzorků.



Obr. 3.9: 2-fázové triggerování na nástupnou hranu signálu s využitím AWDG- převzato z [2]



Obr. 3.10: Vzájemné propojení dvou čítačů

### ■ Triggerovací čítač

Triggerovací čítač pomocí svého trigger výstupu (TRGO) spouští vzorkování ADC převodníku. Jako zdroj triggeru slouží UPDATE událost a kdy CNT registr dosáhne hodnoty ARR registru a čítač začíná počítat znovu od začátku. Hodnota prescaler a ARR registru tak určuje vzorkovací frekvenci. Vstupním hodinový signálem jsou interní hodiny o frekvenci systémových hodin. Navíc tento čítač operuje v slave modu "combine gated + reset", kdy tento čítač běží jen pokud na trigger vstupu (TRGI) je nastavena logická úroveň '1'. Jakmile dojde k poklesu na logickou úroveň (1) čítač se zastaví a zároveň se vyresetuje- tedy hodnota čítače CNT se vynuluje.

### ■ Čítač vzorků

Jak již název napovídá funkce tohoto čítače je určení pozice odebíraného vzorku v bufferu. Aktuální hodnota čítače odpovídá pozici v kruhovém bufferu následujícího snímaného vzorku. Tento čítač funguje ve funkci external clock 1 kdy čítač počítá náběžné hrany na vstupu TRGI. Na výstupu TRGO je potom stav čítače odpovídající enable bitu.

Vzájemné fungování by se pak dalo popsat následovně: pokud je zapnutý čítač vzorků, tak běží zároveň triggerovací čítač. Čítač vzorků čítá počet TRGO pulzů triggerovacího čítače a pokud čítač vzorků zastavíme, tak se zastaví triggerovací čítač a tím vzorkování ADC převodníkem. Pokud pak počítací čítač běží tzv 'one pulse' modu tak se oba čítače zastaví po odběru stanoveného počtu vzorků.

Analogové kanály	CH1	CH2	CH3	CH4	Vnitřní reference
Přítomnost na Pinu	PA0	PA1	PA2	PA3	-
Dostupné ADC	ADC 1, ADC2	ADC1, ADC2	ADC1	ADC1	ADC1
Vnitřní kanál ADC	Kanál 1	Kanál 2	Kanál 3	Kanál 4	Kanál 18

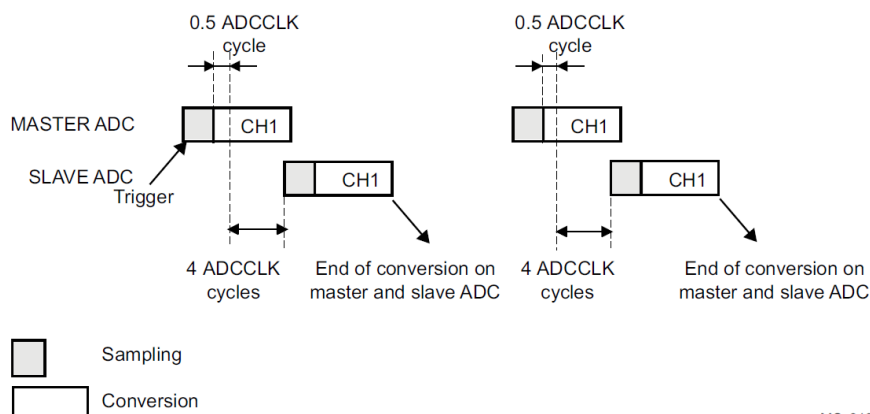
**Tab. 3.3:** Přehled jednotlivých analogových kanálů a jejich na dostupnosti na ADC převodnicích

### 3.6.2 Různé módy vzorkování

Pokud máme k dispozici pouze jeden ADC převodník, musíme v případě měření signálu na více kanálech vstup ADC převodníku přepínat mezi jednotlivými kanály. V případě využití dvou a více ADC převodníků máme daleko větší variabilitu jak vzorkovat daný set analogových kanálů. Podrobněji se jednotlivým módům věnuje práce [2]. STM32G431 má k dispozici 2 ADC, které lze využít pro zvýšení vzorkovací frekvence, respektive zvýšení doby vzorkování pro stejnou vzorkovací frekvenci oproti použití 1 ADC. Výběr použitého módu vzorkování pak záleží na zvolených kanálech, respektive na jejich počtu a na jejich přítomnosti na jednotlivých ADC převodnicích. Například kanál osciloskopu číslo 1 je na pinu PA0 a na tomto pinu také může být připojen na vstup převodníků ADC 1(kanál 1) a ADC 2(kanál1), ale například kanál osciloskopu číslo 3 je na pinu PA2, který se sice dá připojit na vstup převodníku ADC1, ale už se nedá připojit na vstup převodníku ADC2

#### Vzorkování 1 kanálu

Pokud je zvolený kanál osciloskopu přiveditelný na vstupy obou ADC převodníků, můžeme využít tzv. "Dual -interleaved modu". Tento režim spočívá ve střídavém vzorkování jedním převodníkem a v průběhu konverze odebraného vzorku vzorkováním převodníkem druhým. U mikrokontroléru rodiny STM32G4 minimální doba odběru jednoho vzorku(vzorkování + konverze) trvá 15 period hodinového signálu ADC převodníku. V případě dual - interleaved modu lze pak vzorkovat zvolený kanál druhým ADC v momentě kdy na prvním ADC převodníku ještě probíhá konverze viz obrázek 3.11. Výsledně lze pak odebrat vzorek až každých 8 cyklů. Což dovoluje téměř zdvojnásobit maximální vzorkovací frekvenci. Druhou výhodou pak je možnost zvolení delší doby vzorkování při měření s nižší vzorkovací frekvencí více v kapitole 3.6.3.



Obr. 3.11: Převzato z [4]

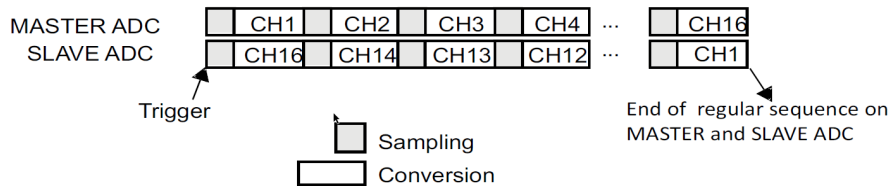
Problémem běžně popisovaného dual interleaved režimu je že vzorkování podřazeného ADC2 je spuštěno pevně stanový počet cyklů po vzorkování ADC1, což může zkreslit průběh měřeného signálu, pokud časový interval mezi odběry vzorků není stejná. V mé implementaci tedy používám něco jako Independent interleaved mod. V tomto režimu využívám output compare kanálu čítače, abych spouštěl ADC převodníky střídavě tak, aby mezi po sobě jdoucími vzorky byla pokud možno stejná vzdálenost. Na výstupu čítače tedy je nastaven obdelníkový signál se střídou 50% , ADC1 začíná vzorkovat na hranu náběžnou a ADC2 na hranu sestupnou. Každý cyklus čítače tedy znamená odebrání 2 vzorků.

Oproti klasickému dual interleaved modu tento "independent interleaved mód" využívá nezávislého triggerování obou ADC převodníků. Nicméně stále je využito možnosti vyčítání naměřených dat ze sdílených data registů jako v případě obvyklého 'dual-interleaved' modu. Díky využití sdílených data registů lze snížit vytížení DMA a AHB sběrnice a vyčítat data z obou ADC převodníků najednou a to s použitím 1 DMA kanálu přenášející 32 bitů . Přenos je pak spouštěn událostí dokončení konverze na ADC2.

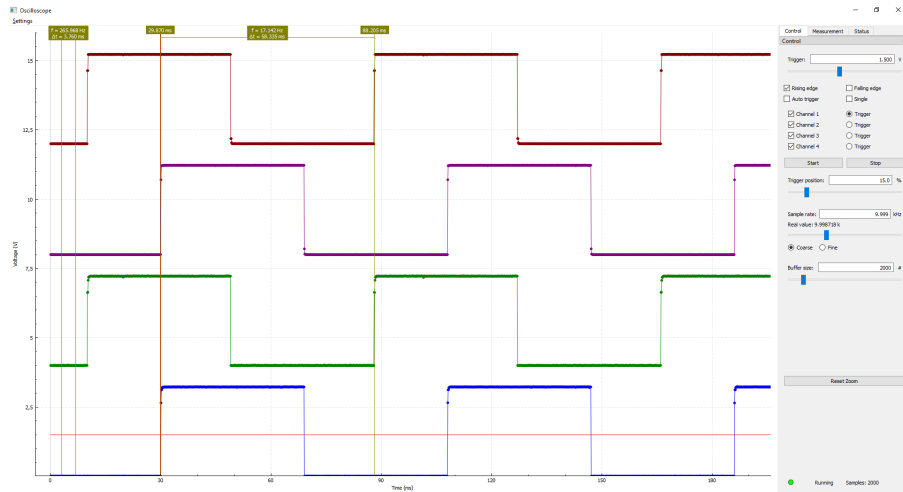
## Vzorkování 2 a více kanálů

Pokud je to možné v případě zvolení 2 a více kanálů, je výhodné použít Dual Simultaneous režim, který nám umožní vzorkování více kanálů najednou. V mém FW jsem implementoval automatické přiřazování kanálů jednotlivým převodníkům, podle jejich dostupnosti. Přiřazování probíhá tak že nejdříve se přiřadí kanály, které jsou dostupné pouze na jednom z převodníků a pak se posoupnosti kanálů doplní kanály dostupnými na obou převodnících tak, aby pokud možno oba ADC převodníky snímali stejný počet kanálů. Vznikají tak různé konfigurace na základě zvolených kanálů. Při analogových kanálech, tak jak jsou popsány v tabulce 3.3 tak je při zvolení jakýkoliv 2 kanálů CH1-CH4 zajištěno snímání ve stejný čas, kromě kombinace CH3+CH4, které jsou oba dostupné pouze na ADC1 a tedy musí být vzorkovány multiplexováním vstupu ADC převodníku. To má za následek v případě této kombinace sníženou maximální vzorkovací frekvenci v porovnání s jinými 2 kanálovými konfiguracemi. V případě zvolení 3 nebo 4 kanálů se pak ukázalo jako nejvhodnější řešení pevné nastavení skenování všech 4 kanálů, kdy jsou najednou snímány kanály CH1+CH3 a CH2+CH4. Po navzorkování bufferu je pak

případně nezvolených kanál vyfiltrován z bufferu před posláním naměřených dat do počítače.



**Obr. 3.12:** Dual Regular simultaneous mod převzato z [4]

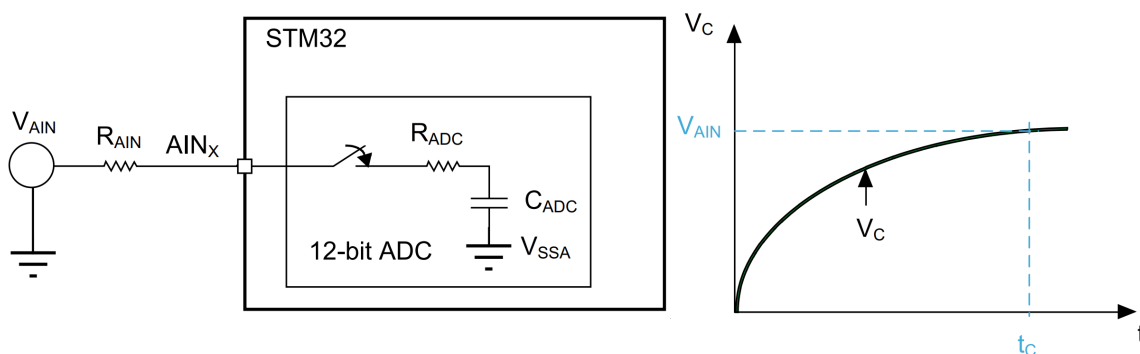


**Obr. 3.13:** Zpoždění kanálu 2 a 4 za kanálem 1 a 3 při použití Dual simultaneous modu a stroboskopickém vzorkování s ekvivalentní vzorkovací frekvencí 78MHz

### 3.6.3 Nastavení délky vzorkování a vliv na maximální vstupní odpor

Impedance analogového zdroje signálu, respektive sériový odpor ( $R_{AIN}$ ) mezi zdrojem a pinem ovlivňují proud nabíjející vzorkovací kondenzátor. Časová konstanta nabíjení  $t_c$  potom určuje minimální dobu vzorkování, tak aby se vzorkovací kondenzátor měl čas nabít na vstupní napětí (s tolerancí  $1/2$  LSB)  $V_{AIN}$  jako je znázorněno na obrázku 3.14. Čas vzorkování je limitován vzorkovací frekvencí a tedy pro různé vzorkovací frekvence existuje nějaký maximální vstupní odpor ( $R_{AIN}$ ) takový, aby se měření neodchylovalo o více než  $\pm 0.5$  LSB.

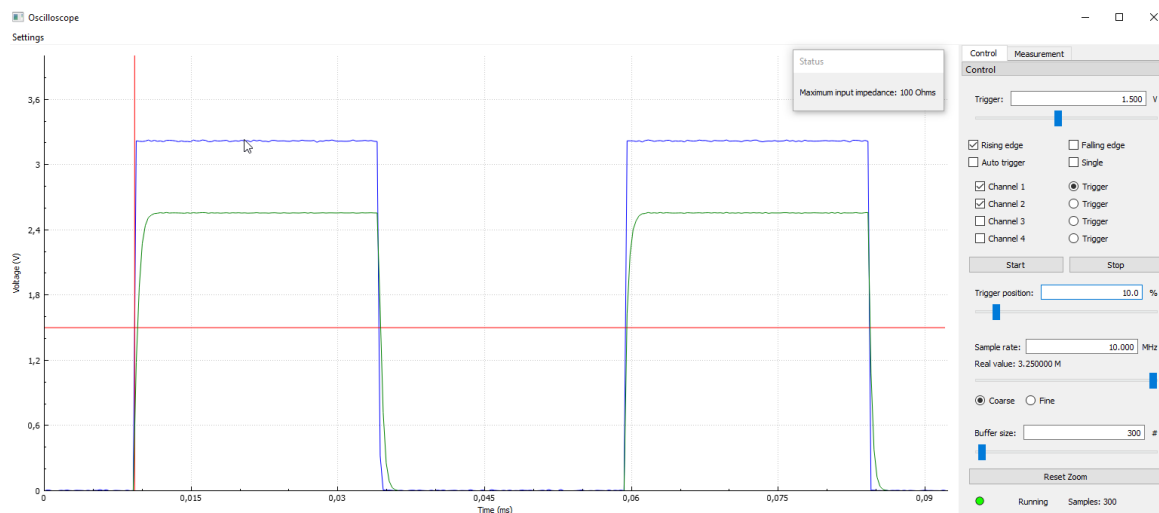




Obr. 3.14: převzato z [6]

Doba vzorkování se nastavuje v počtu period hodinového signálu ADC (ADC\_CLK) převodníku s tím že na výběr je typicky pro daný ADC převodník sada dostupných hodnot s tím, že dále lze pak ovlivnit dobu vzorkování změnou frekvence již zmíněného ADC\_CLK. Jako nejlepší řešení se jeví zvolit pro danou vzorkovací frekvenci a zvolený počet kanálů maximální možnou dobu vzorkování takovou, aby se stihly všechny zrovna zvolené kanály navzorkovat a zkonvertovat před tím, než přijde z triggerovacího čítače pokyn k dalšímu odběru vzorků. Mikrokontroler poté informuje uživatele skrz PC aplikaci o maximální impedanci zdroje signálu, aby měření nebylo zkresleno.

$$t_c = (R_{ADC} + R_{AIN}) \times C_{ADC} \quad (3.3)$$



Obr. 3.15: Efekt nedodržení maximálního vstupního odporu

## 3.7 Logický analyzátor





## Kapitola 4

### Ověření funkčnosti





## Kapitola 5

### Zhodnocení





## Kapitola 6

### Závěr





## Příloha A

### Literatura

- [1] BERLINGER, A. Implementace přístrojových funkcí s využitím mikrořadičů stm32, 2016. [cit. 2023-05-23].
- [2] DUJAVA, J. Softvérově definované osciloskopy s terminálovým rozhraním, 2023. [cit. 2023-03-15].
- [3] RIGOL TECHNOLOGIES, I. Datasheetds1000e, ds1000d series digital oscilloscopes. [online], 12 2015. [cit. 2020-04-13].
- [4] STMICROELECTRONICS. Reference manual RM0440 stm32g4 series advanced arm®-based 32-bit mcus. [online], 1 2017. [cit. 2023-04-30].
- [5] STMICROELECTRONICS. *Datasheet DS12589 STM32G431x6 STM32G431x8 STM32G431xB*. Rev 6, 10 2021. [cit. 2023-05-06].
- [6] STMICROELECTRONICS. Application note AN2834 how to get the best adc accuracy in stm32 microcontrollers. [online], 01 2022. [cit. 2023-05-02].



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **David**

Jméno: **Petr**

Osobní číslo: **420110**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávací katedra/ústav: **Katedra měření**

Studijní program: **Kybernetika a robotika**

Studijní obor: **Kybernetika a robotika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Firmware pro měřicí přístroj s mikrořadičem STM32G431**

Název diplomové práce anglicky:

**Firmware for measuring instrument based on microcontroller STM32G431**

Pokyny pro vypracování:

V návaznosti na přístroj vyvinutý v rámci DP [1] vytvořte firmware pro mikrořadič STM32G431 tak, aby jej ve spolupráci s PC aplikací Zero eLab Viewer bylo možno využít jako jednoduchý, avšak komplexní měřicí přístroj pro výukové účely. V případě potřeby proveďte nutné úpravy PC aplikace. Přístroj bude zahrnovat funkce osciloskopu i se zobrazením průběhů logických kanálů, dále funkce impulsního a signálového generátoru, čítače a voltmetru se záznamem. Při návrhu firmware můžete též využít vhodné bloky vytvořené v rámci prací [2] a [3]. Výsledný přístroj otestujte a ověřte jeho parametry.

Seznam doporučené literatury:

- [1] Berlinger, A.: „Implementace přístrojových funkcí mikrořadiči STM32“, diplomová práce ČVUT – FEL, 2016
- [2] Cejp M.: „Virtuální přístroj s mikrořadičem pro analýzu signálu v modulační doméně“, bakalářská práce, ČVUT – FEL, 2017
- [3] Dujava J.: „Softwarově definované osciloskopy s terminálovým rozhraním“, diplomová práce ČVUT – FEL, 2022
- [4] Cejp M.: „Virtuální přístroj s mikrořadičem pro analýzu signálu v modulační doméně“, bakalářská práce, ČVUT – FEL, 2017

Jméno a pracoviště vedoucí(ho) diplomové práce:

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

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

Datum zadání diplomové práce: **06.09.2022**

Termín odevzdání diplomové práce: \_\_\_\_\_

Platnost zadání diplomové práce:

**do konce letního semestru 2023/2024**

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

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

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta