

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FAKULTA ELEKTROTECHNICKÁ
KATEDRA MĚŘENÍ

FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF MEASUREMENT

UNIVERZÁLNÍ ČÍTAČ PRO SYSTÉM LEO
UNIVERSAL COUNTER FOR SYSTEM LEO

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

JAN MUCALA

LEDEN 2018



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ
V PRAZE

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FAKULTA ELEKTROTECHNICKÁ
KATEDRA MĚŘENÍ

FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF MEASUREMENT

UNIVERZÁLNÍ ČÍTAČ PRO SYSTÉM LEO
UNIVERSAL COUNTER FOR SYSTEM LEO

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAN MUCALA

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. JAN FISCHER, CSc.

LEDEN 2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mucala** Jméno: **Jan** Osobní číslo: **393055**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**
Studijní obor: **Senzory a přístrojová technika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Univerzální čítač pro systém LEO

Název diplomové práce anglicky:

Universal Counter for System LEO

Pokyny pro vypracování:

Navrhněte a realizujte blok softwarově definovaného univerzálního čítače pro výukovou platformu LEO (Little embedded oscilloscope) s mikrořadičem STM32F303RE, který tak bude rozšiřovat funkce systému LEO. Blok bude umožňovat měřit frekvenci, poměr frekvencí, periodu, délku impulsů a zpoždění impulsů ve dvou kanálech. Dalšími funkcemi bude generace vícekanálového signálu PWM a generace signálů PWM s časově proměnnou střídou danou definovaným průběhem. Vytvořte potřebné programy pro mikrořadič i programovou nadstavbu do systému LEO pro nadřazené PC.

Seznam doporučené literatury:

- [1] RM0316 Reference Manual, STMicroelectronics, 2016 www.st.com
- [2] DS10362 STM32F303 Data, STMicroelectronics, 2016
- [3] Yiu, J.: Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors

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: **26.09.2017**

Termín odevzdání diplomové práce: _____

Platnost zadání diplomové práce: **28.02.2019**

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

podpis vedoucí(ho) ústavu/katedry

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

III. PŘEVZETÍ ZADÁNÍ

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

ABSTRAKT

Tato práce se zabývá návrhem a realizací univerzálního čítače, který rozšiřuje výukovou platformu LEO (Little embedded oscilloscope) vyvinutou pro mikrokontrolér STM32F303RE. Čítač využívá dvou základních metod měření frekvence, s jejichž pomocí dosahuje velkého rozsahu měřených hodnot. Reciproční metoda je určena pro měření v rozsahu od desítek mHz až po desítky kHz. Přímá metoda ve výchozím nastavení pak pokryje s nižší chybou rozsah frekvencí od desítek kHz až po stovky MHz. Univerzální čítač dále poskytuje funkce pro měření poměru dvou frekvencí, měření střídy, délky pulsu a nakonec měření časového intervalu mezi dvěma zvolenými událostmi na dvou nezávislých kanálech.

V práci lze rovněž nalézt návrh a realizaci funkce čtyřkanálového generátoru synchronizovaných PWM signálů za použití jednoho čítače a v neposlední řadě též implementaci dvoukanálového generátoru signálů PWM s časově proměnnou střídou danou definovaným průběhem.

KLÍČOVÁ SLOVA

ARM; Cortex-M4; STM32F3xx; univerzální čítač; měření frekvence; přímá metoda; reciproční metoda; poměr frekvencí; měření střídy a délky pulsu; chyba měření; synchronizované PWM; regulace střídy PWM; logický analyzátor

ABSTRACT

This thesis deals with the design and realization of a universal counter which extends the learning platform LEO (Little Embedded Oscilloscope) developed for the STM32F303RE microcontroller. The counter uses two basic frequency measurement methods to reach a large range of measured values. The reciprocal method is designed for measurements ranging from tens of mHz to tens of kHz. By default, the conventional method covers a range of frequencies ranging from tens of kHz to hundreds of MHz with a lower error. The universal counter further provides the function of two-frequency ratio measurement, duty-cycle, pulse length measurement and finally the measurement of time interval between two selected events on two independent channels.

In the thesis, there is also described the design and realization of 4-channel generator of synchronized PWM signals using a single timer and, last but not least, the implementation of 2-channel PWM signal generator with time-varying duty-cycle given by the defined waveform.

KEYWORDS

ARM; Cortex-M4; STM32F3xx; universal counter; frequency measurement; conventional method; reciprocal method; frequency ratio; duty-cycle and pulse width measurement; measurement error; synchronized PWM; PWM duty-cycle regulation; logic analyzer

MUCALA, Jan *Univerzální čítač pro systém LEO*: diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra měření, 2018. 82 s. Vedoucí práce byl doc. Ing. Jan Fischer, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Univerzální čítač pro systém LEO“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Praha

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu práce panu doc. Ing. Janu Fischerovi, CSc. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Praha

.....

(podpis autora)

OBSAH

1	Úvod	12
2	Přehled rozšíření aplikace LEO	14
2.1	Předchozí stav aplikace	14
2.2	Cíl práce	18
3	Charakteristiky čítačů	20
3.1	Obecné charakteristiky	20
3.2	Zdroje chyb	22
3.2.1	Chyba časové základny	22
3.2.2	Kvantizační chyba	23
3.2.3	Chyba triggeru	23
3.2.4	Systematická chyba	24
4	Návrh a realizace univerzálního čítače	25
4.1	Přímá metoda měření frekvence	25
4.1.1	Obvodové řešení přímého čítání	26
4.1.2	Programová obsluha přímého čítání	29
4.1.3	Chyba měření přímého čítání	30
4.1.4	Grafické rozhraní čítače přímého čítání	32
4.2	Reciproční metoda měření frekvence	33
4.2.1	Obvodové řešení recipročního čítače	33
4.2.2	Programová obsluha recipročního čítače	36
4.2.3	Chyba měření recipročního čítače	37
4.2.4	Grafické rozhraní recipročního čítače	38
4.3	Srovnání chyby měření přímého a recipročního čítače	39
4.4	Měření střídy a délky pulsu	41
4.4.1	Obvodové řešení měření střídy a délky pulsu	41
4.4.2	Programová obsluha měření střídy a délky pulsu	44
4.4.3	Grafické rozhraní pro měření střídy a délky pulsu	45
4.5	Měření poměru frekvencí	45
4.5.1	Obvodové řešení měření poměru frekvencí	45
4.5.2	Programová obsluha měření poměru frekvencí	47
4.5.3	Chyba měření poměru frekvencí	48
4.5.4	Grafické rozhraní pro měření poměru frekvencí	49
4.6	Měření časového intervalu mezi dvěma událostmi	50
4.6.1	Obvodové řešení měření časového intervalu	51

4.6.2	Programová obsluha měření časového intervalu	53
4.6.3	Chyba měření časového intervalu	53
4.6.4	Grafické rozhraní pro měření časového intervalu	54
5	Generátor PWM signálu s časově proměnnou střídou	55
5.1	Princip funkce generátoru PWM s časově proměnnou střídou	55
5.2	Zkreslení předpisu střední hodnoty PWM	57
5.3	Grafické rozhraní generátoru PWM s časově proměnnou střídou	60
6	Čtyřkanálový generátor synchronizovaných signálů PWM	62
6.1	Princip funkce generátoru synchronizovaných signálů PWM	62
6.2	Grafické rozhraní generátoru synchronizovaných signálů PWM	64
7	Logický analyzátor	66
7.1	Princip funkce logického analyzátoru	66
7.2	Grafické rozhraní logického analyzátoru	68
8	Konečná podoba platformy LEO	70
8.1	Struktura .C projektu Instrulab	70
8.2	Výsledné GUI aplikace LEO	72
9	Závěr	74
	Literatura	76
	Seznam symbolů, veličin a zkratk	78
	Seznam příloh	79
A	Pinout mikroprocesoru	80
B	Vyvedené funkce na konektory vývojového kitu	81
C	Obsah přiloženého CD	82

SEZNAM OBRÁZKŮ

2.1	Blokové schéma mikrořadiče STM32F303RE [3]	14
2.2	Vývojový kit NUCLEO-F303RE [4]	15
2.3	Původní verze aplikace LEO	16
2.4	Čtyřkanálový osciloskop	16
2.5	Dvoukanálový funkční generátor	17
2.6	Dvoukanálový zdroj stejnosměrného napětí	17
2.7	Čtyřkanálový voltmetr.	17
3.1	Vstupní charakteristika elektronického obvodu <i>Schmitt trigger</i>	20
3.2	Hysterezní smyčka napěťového komparátoru <i>Schmitt trigger</i>	21
3.3	AC-DC coupling, vyloučení stejnosměrné složky ze střídavého signálu	22
3.4	Kvantizační chyba čítače	23
3.5	Chyba triggeru; žlutá - očekávaný trigger, černá - skutečný trigger	24
4.1	Blokové schéma zobrazující princip přímého měření frekvence	25
4.2	Relativní rozlišení čítače v závislosti na době otevření hradla	26
4.3	Blokové schéma obvodového řešení čítače přímého měření	27
4.4	Schéma vnitřního propojení čítačů TIM2 (*) a TIM4 (*) [2]	28
4.5	Vývojový diagram programové obsluhy přímého čítání	29
4.6	Uživatelské rozhraní čítače založeného na přímé metodě měření	32
4.7	Blokové schéma principu reciproční metody měření frekvence	33
4.8	Blokové schéma obvodového řešení recipročního čítače	34
4.9	Schéma vnitřního zapojení periferie TIM2 recipročního čítače [2]	35
4.10	Vývojový diagram programové obsluhy recipročního čítání	36
4.11	Uživatelské rozhraní čítače založeného na reciproční metodě měření	39
4.12	Srovnání přesnosti měření recipročního a přímého čítače ($T_g = 100$ ms)	40
4.13	Znázornění aktivity vstupu <i>Input Capture</i> v <i>PWM input mode</i>	42
4.14	Blokové schéma obvodového řešení měření střídy a délky pulsu	42
4.15	Schéma vnitřního propojení TIM2 pro měření střídy a délky pulsu [2]	43
4.16	Vývojový diagram programové obsluhy měření střídy a délky pulsu	44
4.17	Uživatelské rozhraní čítače v módu měření střídy a délky pulsu	45
4.18	Blokové schéma obvodového řešení čítače měřícího poměr frekvencí	46
4.19	Vývojový diagram programové obsluhy měření poměru frekvencí	47
4.20	Uživatelské rozhraní univ. čítače v módu měření poměru dvou frekvencí	50
4.21	Blokové schéma principu měření časového intervalu	50
4.22	Blokové schéma obvodového řešení čítače měřícího časový interval v <i>Dependent</i> módu	51
4.23	Důsledek obrácené sekvence příchozích událostí v <i>Dependent</i> módu	52

4.24	Blokové schéma obvodového řešení čítače měřícího časový interval v <i>Independent</i> módu	52
4.25	Vývojový diagram programové obsluhy měření časového intervalu . . .	53
4.26	Uživatelské rozhraní metody měření časového intervalu	54
5.1	Závislost střední hodnoty signálu na střídě PWM	55
5.2	Zkreslení předpisu konfigurací nízké frekvence PWM signálu	58
5.3	Snížení počtu vzorků konfigurací vysoké frekvence předpisu	59
5.4	Uživatelské rozhraní generátoru PWM s časově proměnnou střídou . . .	60
6.1	Prinip funkce generátoru synchronizovaných signálů PWM. Dvojitá změna <i>C/C</i> registru pomocí DMA během jedné periody čítače	63
6.2	Uživatelské rozhraní generátoru synch. sig. PWM v <i>Equidistant</i> módu	64
6.3	Uživatelské rozhraní generátoru synch. sig. PWM v <i>Independent</i> módu	65
7.1	Blokové schéma obvodového řešení logického analyzátoru	67
7.2	Rotace pole do požadovaného uspořádání	68
7.3	Grafické uživatelské rozhraní logického analyzátoru	69
8.1	Struktura projektu	70
8.2	Architektura vytvořeného softwaru	71
8.3	Volání funkce <code>HAL_TIM_Base_MspInit()</code> z funkce <code>HAL_TIM_Base_Init()</code> ovladače periferie čítače/časovače	71
8.4	Finální grafické uživatelské rozhraní výukové platformy LEO	72
A.1	Pinout mikrokontroléru STM32F303RE a zobrazení mapování funkcí	80
B.1	Vývedené funkce na konektory vývojového kitu NUCLEO-STM32F303RE	81

SEZNAM TABULEK

4.1	Naměřené hodnoty čítačem založeným na přímé metodě měření . . .	31
4.2	Hodnoty naměřené recipročním čítačem	38
4.3	Vypočtené mezní frekvence, při kterých přímý a reciproční čítač do- sahují stejné chyby měření	41
4.4	Naměřené hodnoty poměru dvou vstupních frekvencí	49
4.5	Naměřené hodnoty časového intervalu mezi dvěma událostmi	54

1 ÚVOD

V současné době lze na trhu nalézt velkou řadu laboratorních měřících přístrojů různých parametrů. Prakticky se lze setkat s voltmetry, ohmmetry, multimetry přes generátory stejnosměrného a střídavého napětí, generátory signálů PWM až po osciloskopy, logické a spektrální analyzátoři ad. Všechny tyto přístroje většinou spojuje poměrně vysoká cena, zapříčiněná dlouhým a nákladným vývojem, drahými součástkami a přidanou hodnotou "know how".

Z hlediska nutnosti využití vývojových a testovacích prostředků může být tedy samotný vývoj velice nákladný, ne vždy je však zapotřebí moderních technologií poskytujících vysokou přesnost měření či frekvenci generovaných signálů. Z těchto důvodů byla již dříve vyvinuta aplikace Little embedded oscilloscope, která nabízí vynikající alternativu zastupující širokou škálu měřících zařízení a generátorů libovolných průběhů.

Takovéto zařízení může být například nedílnou součástí výuky ve školách, kde není snadné a z finančních důvodů ani možné zajistit všem studentům potřebné měřící přístroje. Může tak urychlit práci studentů i zefektivnit studium mimo laboratorní prostory. Jediným materiálním požadavkem zařízení LEO je PC s operačním systémem Windows, vývojový kit NUCLEO-F303RE a propojující USB kabel. Díky tomu se systém stává dostupným široké veřejnosti a konfigurace velice jednoduchou.

Výpočetní jednotka vývojového kitu obsahuje periferii výkonné čítací jednotky, s jejíž využitím lze rozšířit aplikaci LEO o funkci univerzálního čítače. Školní laboratoře povětšinou nedisponují zařízeními specializujícími se na měření frekvence, a tak lze hledat využití právě tam. Organické začlenění univerzálního čítače do platformy LEO přináší výhodu ve zjednodušení a urychlení přístupu k němu v případě, jsou-li zároveň využívány i jiné funkcionality této platformy. Uživatel má větší přehled a mimo jiné se nemusí učit ovládat zvláštní aplikaci. Na základě zmíněné čítací jednotky lze vyvinout i různé PWM generátory, jejichž začleněním se z platformy LEO stává velmi užitečný testovací nástroj.

Výhodou aplikace LEO jsou především nízké náklady na její plnohodnotné zprovoznění, které jsou závislé pouze na ceně vývojového kitu NUCLEO-F303RE od firmy STMicroelectronics. Systém poskytuje software s otevřeným zdrojovým kódem¹, který lze jednoduše stáhnout z oficiálních stránek elektrotechnické fakulty

¹V angličtině označován jako *Open source*.

ČVUT na <http://leo.fel.cvut.cz/> a v případě potřeby jej lze upravovat.

Struktura vlastního firmware určeného mikrořadiči je navržena takovým způsobem, aby byl jednoduše přenositelný na jiný mikrokontrolér na jiném vývojovém kitu a aplikace tak mohla být rozšířena o další funkcionality požadující nové prostředky dostupné na nových mikroprocesorech. Tato struktura byla navržena a vytvořena již v první verzi aplikace LEO, kterou vyvinul pan Ing. Jiří Hladík.

2 PŘEHLED ROZŠÍŘENÍ APLIKACE LEO

V první části kapitoly je popsáno stádium vývoje, ve kterém se aplikace nacházela před započítím této práce. V druhé části jsou pak zmíněny cíle rozšíření platformy LEO.

2.1 Předchozí stav aplikace

Přibližně před dvěma lety vznikla laboratorní experimentální výuková platforma LEO s podporou mikrokontroléru STM32F303RE. Jedná se o mikrokontrolér spadající do výkonnostní řady STM32F303xD/E, která je založena na 32 bitovém RISC jádře ARM®Cortex®-M4, jež integruje FPU a operuje na frekvenci vnitřních hodin 72 MHz. Více informací k mikrořadiči lze nalézt na webových stránkách společnosti STMicroelectronics [3].



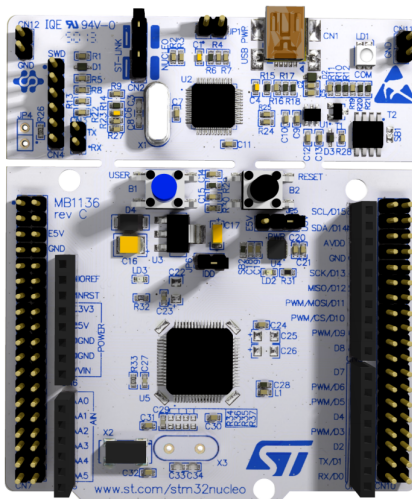
Obr. 2.1: Blokové schéma mikrořadiče STM32F303RE [3]

Původním základním požadavkem aplikace LEO na parametry mikrokontroléru byl počet ADC a DAC převodníků a jejich rychlost převodu signálu¹. Pro komunikaci s nadřazeným PC bylo zvoleno USB s využitím VCP. Samotná komunikace mezi ST-Link a mikrořadičem STM32F303RE je uskutečněna pomocí sériové linky

¹Prvními vytvořenými funkcemi byl osciloskop, voltmetr a generátor stejnosměrného napětí.

UART.

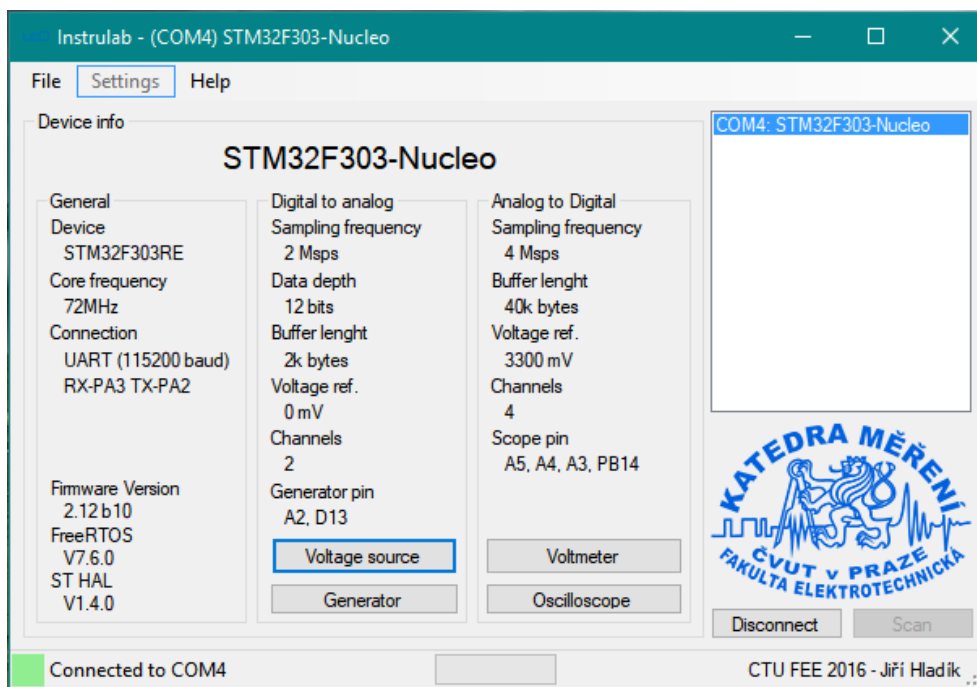
Aplikace byla vypracována za pomoci vývojového kitu NUCLEO-F303RE (viz. 2.2), který propojuje všechny GPIO piny mikroprocesoru s konektory a stává se tak vhodným nástrojem pro vývoj a testování. Z obrázku NUCLEO kitu je patrné, že je deska osazena pouze jedním 8 MHz HSE krystalem, který tvoří oscilátor ST-Linku. Nicméně díky výchozímu nastavení propojky SB50 lze přivádět hodinový signál z MCO výstupu ST-Linku na vstup OSC IN mikrokontroléru a není tak třeba osazovat krystal X3 určený pro mikrokontrolér. Komunikace prostřednictvím sériové linky UART je taktéž povolena výchozím nastavením propojek, a tak je uživatel ušetřen jakýchkoli úprav vývojové desky a již hned po její koupi lze aplikaci v plném rozsahu využívat.



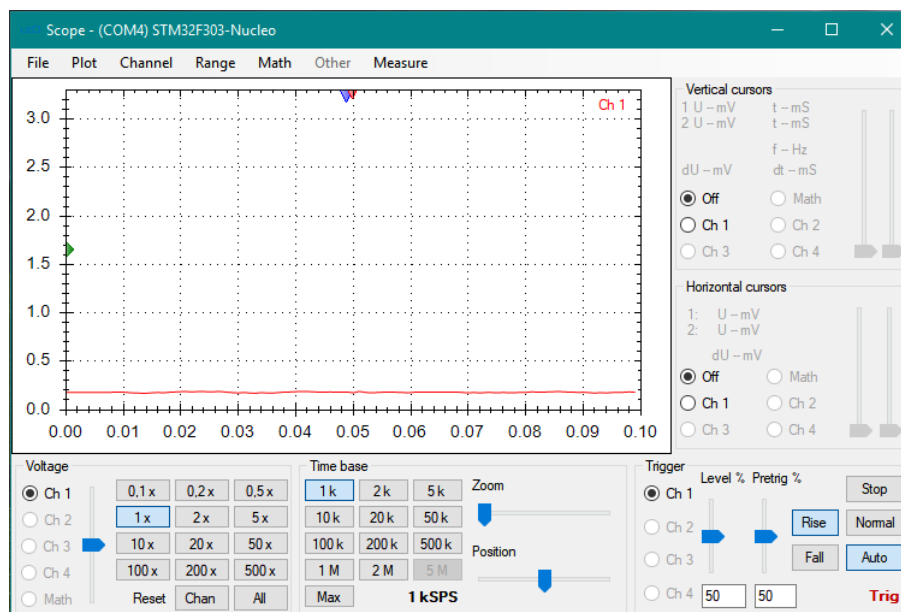
Obr. 2.2: Vývojový kit NUCLEO-F303RE [4]

Aplikace před vznikem této práce obsahovala dvoukanálový funkční generátor (sinus, trojúhelník, obdélník, volitelná funkce - arbitrary generátor dle nahraného souboru .CSV z MS Excel), čtyřkanálový číslicový osciloskop s možností uložení průběhu do souboru .CSV nebo do obrázku, dvoukanálový zdroj stejnosměrného napětí a závěrem čtyřkanálový voltmetr. Zároveň byla k NUCLEO kitu vytvořena analogová deska, se kterou lze kit rozšířit o standardní konektory reálných měřících přístrojů. Více informací je poskytnuto na webových stránkách komunity společnosti STMicroelectronics [5].

PC aplikace byla programována v jazyce C# s využitím balíku .NET Framework verze 4.0, díky čemuž je funkční i na operačním systému Windows XP a novějších



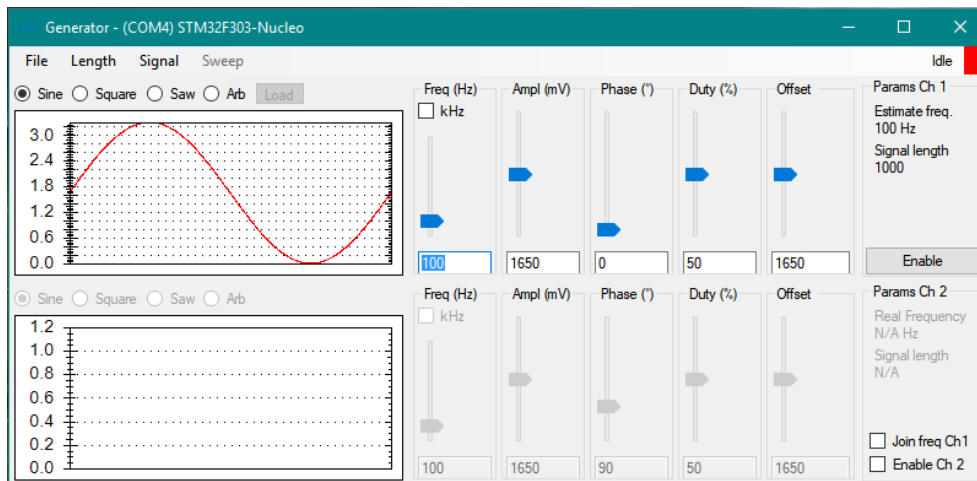
Obr. 2.3: Původní verze aplikace LEO



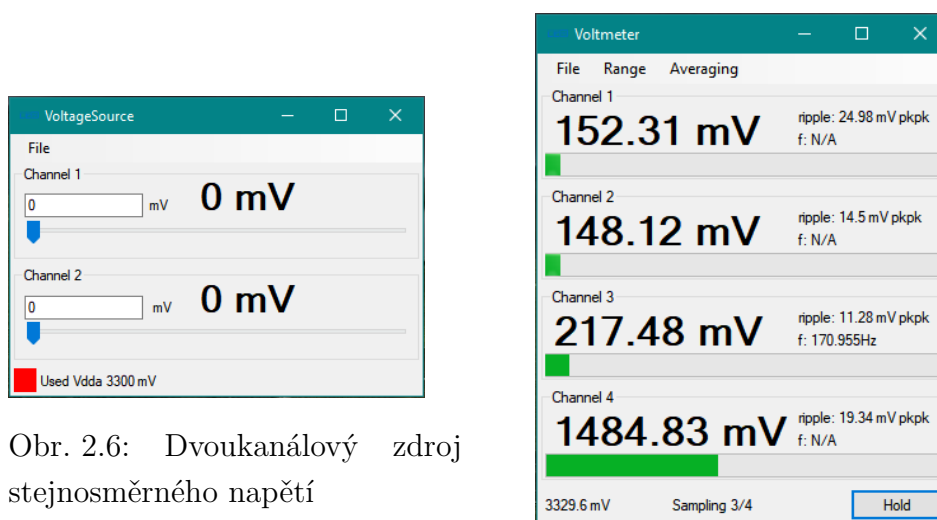
Obr. 2.4: Čtyřkanálový osciloskop

verzích [10]. Firmware pro mikrokontrolér byl psán v jazyce C.

Chod firmwaru je zajištěn operačním systémem FreeRTOS [7], který poskytuje prioritní plánování běhu vláken. FreeRTOS přináší řadu výhod, jako je snadná rozšiřitelnost a přehlednost programu. Konkrétně lze mluvit o abstrakci časování, kde



Obr. 2.5: Dvoukanálový funkční generátor



Obr. 2.6: Dvoukanálový zdroj stejnosměrného napětí

Obr. 2.7: Čtyřkanálový voltmetr.

je namísto hardwarových časovačů a zpožďovacích smyček využito API operačního systému poskytujícího prostředky pro obsluhu vláken. Dále pak lze mluvit o zjednodušení kolektivního vývoje a testování.

Aplikace využívá standardních funkcí FreeRTOS jako je správa vláken, správa sdílených prostředků, k nimž je přistupováno za pomoci mutex, komunikace mezi vlákny pomocí front, správa přerušení a jejich priorit atd.

Operační systém FreeRTOS nabízí pro správu paměti [8] několik způsobů definovaných samostatnými soubory. Tato aplikace využívá soubor `heap_2.c`, který poskytuje možnost dynamické alokace paměti pomocí *thread safe* funkcí `pvPortMalloc()`

a `pvPortFree()`. Pokud jde o `heap_1.c`, ten možnost uvolnění paměti nenabízí, a tudíž je vhodnější spíše pro malé aplikace. Naopak značnou výhodou souboru `heap_2.c` je jeho velikost a možná fragmentace paměti způsobená funkcemi zmíněnými výše, v porovnání s funkcemi `heap_4.c`.

Stojí za zmínku, že i přes to, že je v aplikaci intergován soubor `heap_2.c`, paměť je alokována ve všech případech staticky. V tomto trendu se během dalšího vývoje pokračovalo, nicméně je v zájmu šetření pamětí budoucími globálními úpravami tuto tendenci změnit.

Firmware dále využívá knihoven HAL [9], které jsou volně ke stažení z webových stránek STMicroelectronics. Částečná inicializace periférií byla provedena za pomoci grafického konfiguračního generátoru kódu STM32CubeMX [6].

2.2 Cíl práce

Úkolem diplomové práce bylo rozšířit výukovou platformu LEO o funkcionality univerzálního čítače, o vícekanálový generátor synchronizovaných PWM signálů a generátor signálů PWM s časově proměnnou střídou danou definovaným průběhem.

V laboratorních prostorách se občas můžeme setkat se specializovanými přístroji měřícími frekvenci. Tuto funkci však ve většině případů nabízejí i osciloskopy, kterými jsou dnes laboratoře standardně vybaveny. Osciloskop v aplikaci LEO sice frekvenci vzorkovaného signálu zobrazuje, výpočet je však prováděn softwarově, a to s sebou nese značné nepřesnosti. Přesnějších výsledků měření frekvence signálu má být proto dosaženo hardwarově implementovaným univerzálním čítačem. Ten by měl být dále schopen měřit poměr dvou vstupních frekvencí, střídu signálu, délku pulsu a časový interval mezi dvěma událostmi (vzestupná, sestupná hrana signálu) na dvou nezávislých vstupních kanálech.

Postupem času se čítače vyvíjely a zdokonalovaly. Vznikly nové způsoby měření frekvence jako *Timestamping* nebo Interpolační metoda. Přesto však jejich základní princip zůstává stejný a je vždy založený na funkci reciproční či přímé metody měření frekvence. Tato práce se proto dále zabývá realizací přímé a reciproční metody².

Dalším úkolem bylo vytvořit vícekanálový generátor signálů PWM. Přesněji řečeno se má jednat o generátor synchronizovaných PWM signálů s možností individu-

²V angličtině jsou tyto metody označovány jako *Conventional counting* a *Reciprocal counting*.

ální změny jejich fáze a střídy, resp. délky pulsu. Využití lze hledat v aplikacích, kde je vyžadován přesně daný časový interval mezi dvěma nebo více příchozími pulsy dané délky.

Zadání diplomové práce závěrem stanoví generátor PWM signálu s časově proměnnou střídou danou definovaným průběhem. V původní verzi aplikace byl vytvořen funkční generátor s možností nahrání souboru .CSV. Pro zachování stejnorodosti platformy by grafické rozhraní PWM generátoru mělo být funkčnímu generátoru alespoň podobné s možností nahrání stejného typu souboru s předpisem.

Prakticky lze takovýto PWM generátor použít pro řízení stejnosměrného motoru či například regulaci jasu LED diody.

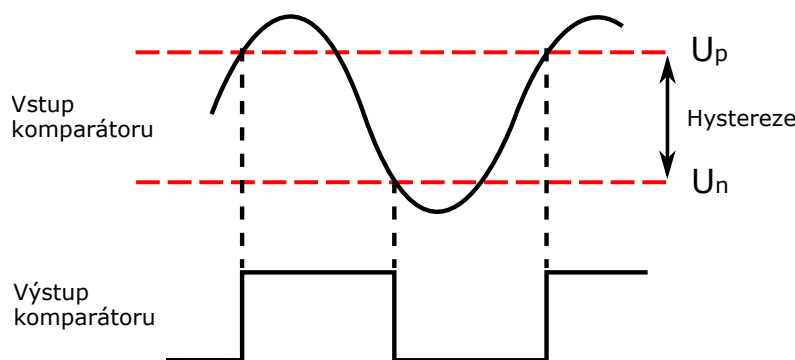
3 CHARAKTERISTIKY ČÍTAČŮ

Tato kapitola stručně popisuje obecné charakteristiky univerzálních čítačů a zdroje jejich chyb.

3.1 Obecné charakteristiky

Nejprve definujme univerzální čítač jako číslicový elektronický přístroj schopný měřit frekvenci vstupního signálu. Může poskytovat další funkce, jako je měření periody, poměru frekvencí dvou vstupních signálů, časový interval mezi dvěma událostmi, střídu obdélníkového signálu a délku pulsu a další kombinace s průměrováním.

Čítače lze obecně charakterizovat několika vlastnostmi, které se liší v závislosti na použitých integrovaných obvodech a použité metodě měření frekvence. Jednou z důležitých vlastností čítačů je citlivost jejich integrovaných obvodů na vstupní signál. Citlivost může být definována jako minimální amplituda signálu, kterou je čítač při konkrétní frekvenci schopen ještě čítat. Citlivost je závislá na velikosti hystereze vstupního napětového komparátoru, nebo obvodu v angličtině nazývaného jako *Schmitt trigger*. Tato skutečnost je znázorněna obrázky 3.1 a 3.2. Citlivý čítač mající malé hysterezní okno nebo nulovou hysterezi může reagovat na zákmity způsobené rušením nebo šumem, a tak čítat nežádoucí hodnoty (obrázek 3.5).

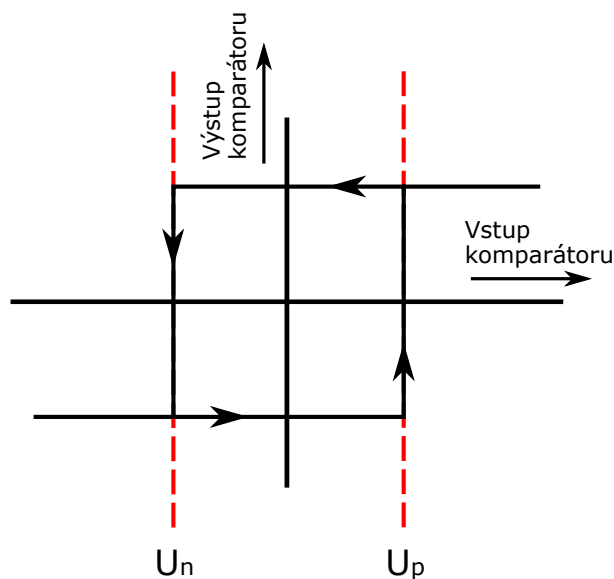


Obr. 3.1: Vstupní charakteristika elektronického obvodu *Schmitt trigger*

Citlivost na sinusový vstupní signál je charakterizována jeho střední kvadratickou hodnotou, kterou lze spočítat jako

$$Citlivost = \frac{U_p - U_n}{2\sqrt{2}},$$

kde U_p je horní a U_n dolní napětová úroveň hystereze.



Obr. 3.2: Hysterezní smyčka napětového komparátoru *Schmitt trigger*

Jak referenční manuál mikrokontroléru STM32F303RE [2] stanoví, pokud je I/O port nakonfigurován jako vstup, *Schmitt trigger* je automaticky aktivován. Proto lze na vstup mikrořadiče přivádět i jiný signál než jen obdélníkový. Manuál [1] uvádí hodnotu hystereze *Schmitt triggeru* 200 mV pro čítačem využitě 3,3 V tolerantní I/O vstupy PA0 a PA1. Jediný pin PA8, který je využíván jako reference v módu měření poměru frekvencí, je 5 V tolerantní s hysterezí *Schmitt triggeru* 100 mV.

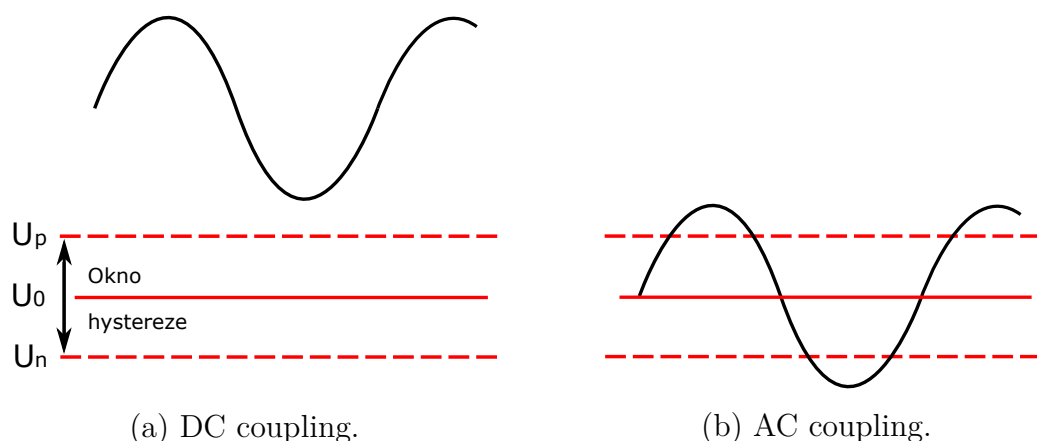
Univerzální čítače často poskytují mnoho užitečných funkcí. Standardem je možnost volby triggeru¹ v závislosti na vzestupné či sestupné hraně signálu. Další funkcí bývá například schopnost odstranit stejnosměrnou složku ze vstupního střídavého signálu a posunout ho tak do požadovaného měřicího napětového rozsahu², jak je znázorněno obrázkem 3.3.

U některých čítačů pak lze nalézt adaptabilní funkci posouvající úroveň triggeru, resp. okna hystereze do rozsahu vstupního signálu. Tento přístup se používá pro obdélníkové signály, u kterých první zmíněná metoda ac-dc coupling nemusí fungovat např. kvůli nízké hodnotě střídavy.

Dalšími funkcemi pak mohou být útlumové články pro případ, že signál přesáhne maximální vstupní rozsah. Automatické zesilovače naopak dokáží zvyšovat

¹Do češtiny lze přeložit jako zdroj spuštění nebo doslovně spouštěč. Obvykle se však překlad nepoužívá, a proto je v celé práci tento výraz označován jako *trigger*.

²V anglické technické literatuře označováno jako *AC-DC coupling*.



Obr. 3.3: AC-DC coupling, vyloučení stejnosměrné složky ze střídavého signálu

citlivost vstupu čítače. Podrobnější charakteristiky a více informací je poskytnuto např. dokumentem [11].

3.2 Zdroje chyb

Nepřesnosti měření univerzálních čítačů jsou způsobeny několika zdroji chyb, které lze rozčlenit do následujících čtyř kategorií:

- Chyba časové základny
- Kvantizační chyba
- Chyba triggeru
- Systematická chyba

V této části jsou probrány všechny zdroje chyb, nicméně v diplomové práci šlo smysluplně zahrnout pouze první dva - zdroj chyby časové základny a chyby kvantizační.

3.2.1 Chyba časové základny

Časovou základnou systému se rozumí frekvence systémových hodin mikrokontroléru. Přesnost této frekvence je udána použitým krystalem. Chybou časové základny se pak míní odchylka skutečné frekvence od její nominální hodnoty.

Protože jsou od systémových hodin odvozeny hodiny všech periférií mikrořadiče a většiny prostředků potřebných pro čítač, jakákoli chyba krystalu se pak odrazí jako chyba měření. Velikost chyby se udává v ppm (parts per million). Tento fakt je

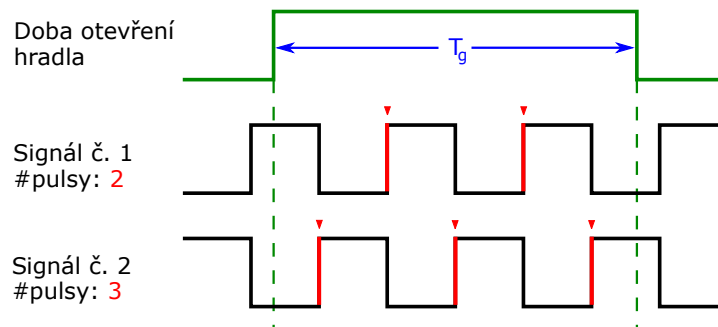
podnětem k použití přesného externího oscilačního elementu.

Vývojový kit NUCLEO-F303RE je osazen 8 MHz krystalem, jehož specifikace³ [12] udává chybu ± 50 ppm v celém funkčním teplotním rozsahu. S touto hodnotou je v celé práci počítáno.

Přesnost krystalu je dále ovlivněna jeho rychlostí stárnutí. Stručně řečeno se jedná o časovou kumulativní změnu generované frekvence. Stárnutí krystalu osazeného na vývojovém kitu je stanoveno jako ± 5 ppm/rok. Posledním zde zmíněným faktorem je citlivost na změnu zdrojového napětí, se kterou se mění výstupní frekvence krystalu.

3.2.2 Kvantizační chyba

Jednou z chyb čítačů je chyba vzorkování o ± 1 vzorek. Chyba se nazývá kvantizační a důvodem její existence je posuv fáze vnitřních hodin čítače oproti vzorkovanému signálu v daném okamžiku. Tuto skutečnost znázorňuje obrázek 3.4.



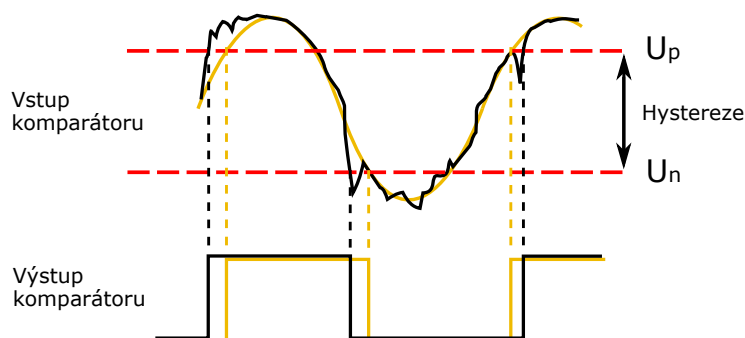
Obr. 3.4: Kvantizační chyba čítače

3.2.3 Chyba triggeru

Chyba triggeru je náhodná chyba způsobená šumem na vstupních kanálech čítače nebo šumem ve vzorkovaném signálu. Vliv má např. na reciproční metodu měření frekvence (viz. podkapitola 4.2) a na měření časového intervalu mezi dvěma událostmi (viz. podkapitola 4.6). Obě metody totiž vyžadují spuštění hradla čítače událostí od vstupního signálu na vstupním pinu. Ze zmíněného principu funkce *Schmitt*

³Daný typ krystalu 9SL8000012AFXHF0-SJ28 stanovený v BOM NUCLEO kitu nebylo možno dohledat, proto je odkázáno na dostupnou dokumentaci krystalu 9SL8000012AFXHF0-RE02.

triggeru, obvodu na vstupu čítače, může vlivem šumu dojít k dřívějšímu či pozdějšímu spuštění hradlování signálu, jak je zobrazeno obrázkem 3.5.



Obr. 3.5: Chyba triggeru; žlutá - očekávaný trigger, černá - skutečný trigger

3.2.4 Systematická chyba

Systematická neboli soustavná chyba je způsobena nedokonalostmi měřícího přístroje nebo použitých součástí a kabelů. V některých případech ji lze korigovat nebo alespoň určit z příslušné dokumentace výrobku.

Značný vliv má tato chyba na metodu měření časového intervalu mezi dvěma událostmi na dvou nezávislých kanálech. Je způsobena nedokonalostmi integrovaných obvodů a rozdílnými propagačními zpožděními na prvním a druhém kanálu (od startující a ukončující hrany). Dalším příspěvkem k chybě může být rozdíl délek přívodních vodičů nebo použití rozdílných materiálů těchto vodičů.

4 NÁVRH A REALIZACE UNIVERZÁLNÍHO ČÍTAČE

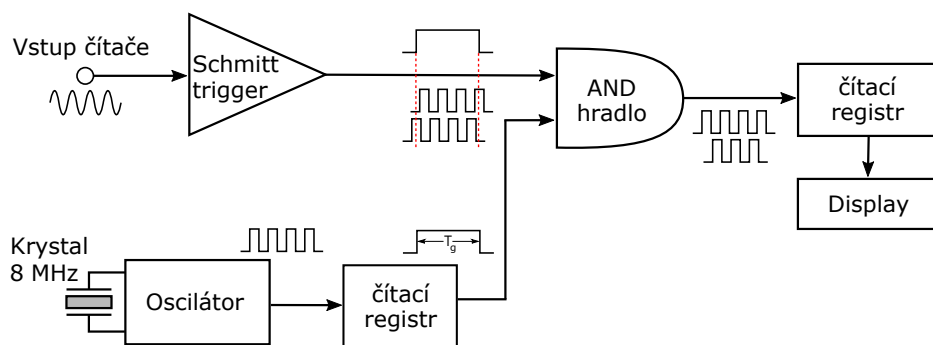
Tato kapitola pojednává o jednotlivých funkcích navrženého univerzálního čítače, jeho grafickém rozhraní, obvodovém řešení, programové obsluze, výhodách, nevýhodách a chybách měření.

4.1 Přímá metoda měření frekvence

První zde zmíněnou metodou měření frekvence je metoda přímá. Principem funkce je čítání period měřeného střídavého signálu v daném časovém intervalu, který je určen otevřením a uzavřením hradla (viz. obrázek 3.4). Výslednou frekvenci lze jednoduše počítat jako

$$f = \frac{n}{T_g}, \quad (4.1)$$

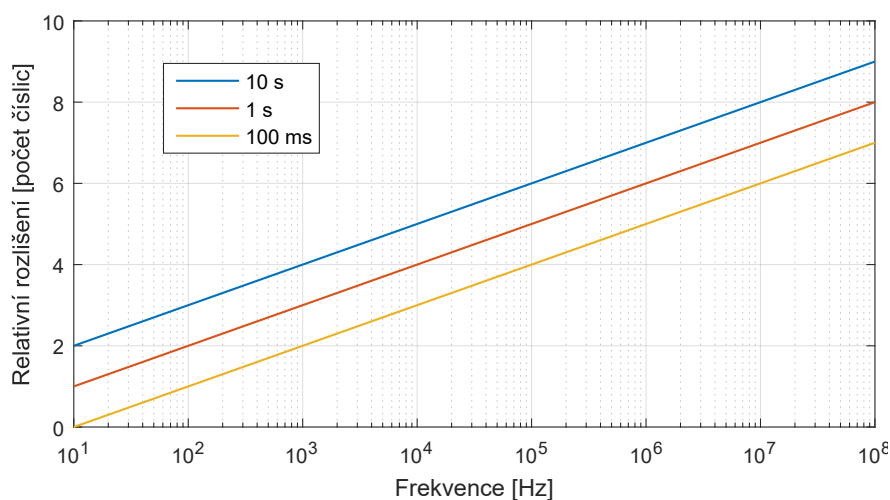
kde n je počet hran signálu navzorkovaných za časový interval T_g (čas otevření hradla). Pro časový interval 1 sekundu platí $f(T_g) = f(1) = n [Hz]$. Následující obrázek 4.1 znázorňuje blokové schéma čítače založeného na této metodě.



Obr. 4.1: Blokové schéma zobrazující princip přímého měření frekvence

Vstupní signál prochází přes *Schmitt trigger* a je čítán v časovém intervalu odvozeném od vnitřních hodin mikrokontroléru. Přesnost požadované doby měření je tedy závislá na chybě použitého krystalu, jak bylo zmíněno v podkapitole *Zdroje chyb* 3.2.

Relativní rozlišení čítače při měření frekvence je určeno dobou otevření hradla a je znázorněno následujícím grafem 4.2. Zvýšením časového intervalu lze dosáhnout vyššího rozlišení čítače. Důvodem je relativní snížení kvantizační chyby.



Obr. 4.2: Relativní rozlišení čítače v závislosti na době otevření hradla

4.1.1 Obvodové řešení přímého čítání

Mikrokontrolér STM32F303RE obsahuje několik 16-bitových časovačů/čítačů¹ a jeden 32-bitový TIM2, který je využit jako čítací element pro čítání period vstupního signálu. V případě potřeby lze další 32-bitový čítač vytvořit kaskádním propojením a synchronizací dvou 16-bitových čítačů. Doba čítání je určena jedním 16-bitovým časovačem TIM4.

Výhodou použitého 32-bitového čítače je možnost jeho taktování z PLL až na 144 MHz, dvojnásobek maxima systémových hodin, díky čemuž je dosaženo rozlišení přibližně 7 ns.

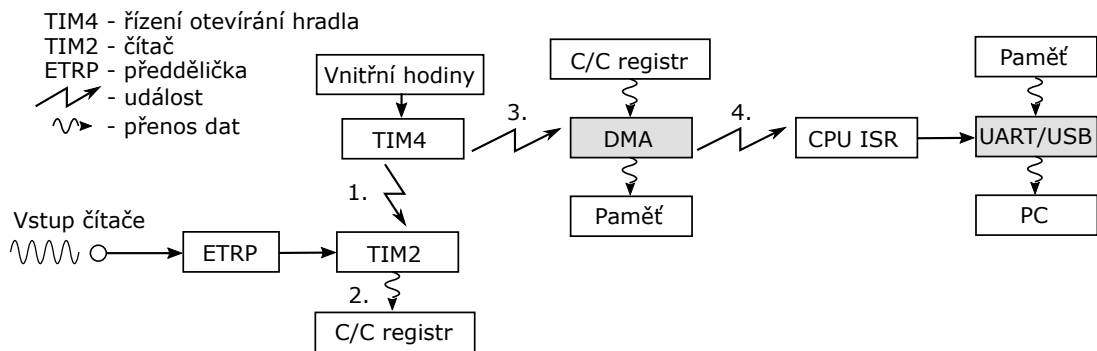
Blokové schéma 4.3 zjednodušeně dokumentuje obvodové řešení čítače založeného na principu přímého čítání. Čítací periferie TIM2 je nakonfigurována ve slave módu *Combined reset + trigger* a časovač TIM4 jako jeho master. Díky tomuto nastavení je TIM2 resetován a znovu spuštěn po každé události od naplnění časovače TIM4, který zde představuje funkci řízení otevírání hradla. Zároveň s touto událostí je automaticky zachycena hodnota čítače TIM2 do *Capture/Compare* registru, aby mohla být následně přenesena pomocí DMA do paměti. DMA je zmíněnou událostí taktéž spouštěno. Hradlo je otevíráno kontinuálně a DMA nastaveno na *circular* pro plnění kruhového bufferu.

Čtvrtý bod v blokovém schématu znázorňuje vyvolání přerušení od dokončené datové transakce pomocí DMA. V tomto přerušení je vyvolána příslušná softwarová

¹Jedná se o čítací jednotku v angličtině nazývanou jako *timer* a přeložit lze jako *časovač* nebo *čítač*. V této práci je použito obou označení a určují, jakou funkci tato periferie v daném případě zastává.

obslužná rutina (ISR), ve které jsou data zpracována a poslána po sériovém komunikačním rozhraní UART a dále pak přes USB do PC. Více informací k programové obsluze lze nalézt v následující sekci 4.1.2.

- 1.) Reset, aktualizace registrů a trigger čítače TIM2 od události naplnění časovače TIM4.
- 2.) Zachycení hodnoty čítače TIM2 do Capture/Compare registru.
- 3.) Spuštění DMA od události naplnění časovače TIM4.
- 4.) Vvolání přerušení od dokončeného přenosu dat z Capture/Compare registru do paměti.



Obr. 4.3: Blokové schéma obvodového řešení čítače přímého měření

Jako vstup čítače byl vybrán ETR (External Trigger), který zde zastupuje vstup vnitřních hodin. Místo vnitřního hodinového signálu CK_INT (Internal clock) je do čítacího *Counter* registru (*CNT*) čítače TIM2 přiveden externí měřený signál. Nicméně hodinový signál o zmíněné frekvenci 144 MHz přesto musí být přiveden na CK_INT.

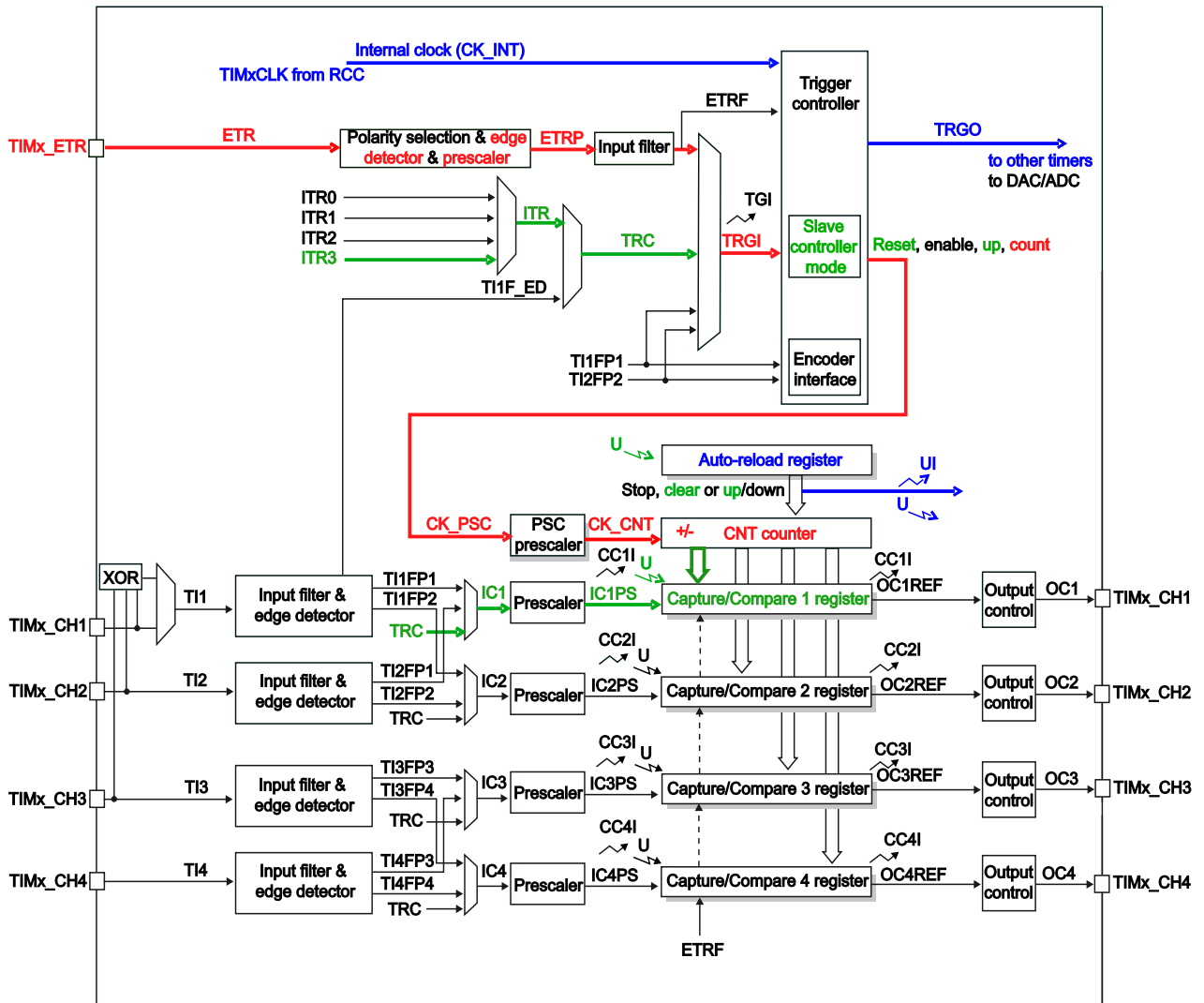
Před čítacím registrem *CNT* čítače TIM2 je umístěna předdělička ETRP (External Trigger Prescaler), kterou je možno konfigurovat na hodnoty 1, 2, 4 a 8. Čítací registr pak čítá každou, každou druhou, čtvrtou či osmou hranu vstupního signálu. Jelikož referenční manuál stanoví, že maximální velikost frekvence signálu za touto předděličkou může být čtvrtina frekvence hodinového signálu přivedeného z PLL na vnitřní hodiny CK_INT (144 MHz) čítače TIM2, jednoduchou matematikou spočteme potenciální maximální frekvenci, kterou lze přivést na tento vstup. Počítejme tedy s nejvyšším stupněm předděličky 8

$$f_{max} = \frac{1}{4} \cdot 144 \cdot 10^6 \cdot 8 = 288 \text{ MHz}. \quad (4.2)$$

Z teoretických výpočtů vyplývá, že maximální frekvence, kterou by mělo být možno měřit, je 288 MHz.

Následující schéma 4.4 podrobněji zobrazuje vnitřní obvodové propojení 32-bitového čítače TIM2. Červenou barvou je zvýrazněno zapojení čítače TIM2, zelenou

pak propojení s časovačem TIM4, který určuje počátek a konec měření. Modrou barvou byly zvýrazněny pouze důležité prvky, které využívá TIM4. Jednoduše řečeno, červená a zelená barva zobrazují zapojení TIM2, samotná modrá pak stručně TIM4. Vnitřní propojení bylo zvýrazněno ve schématu, které bylo přejato z referenčního manuálu [2].



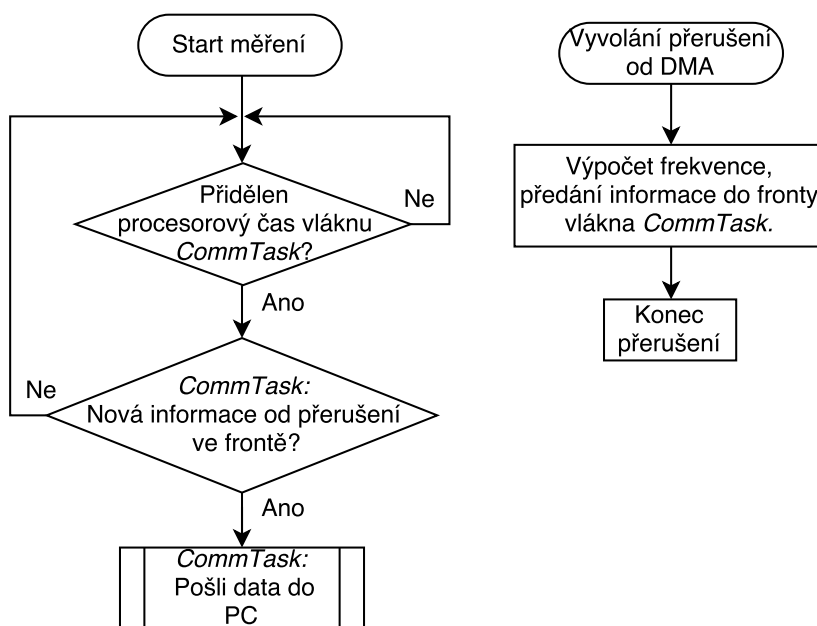
Obr. 4.4: Schéma vnitřního propojení čítačů TIM2 (***) a TIM4 (*) [2]

Naplnění časovače TIM4 je dáno dosažením požadované hodnoty *Auto-Reload* registru. Tato událost (modré U) vyvolá hranu na signálu TRGO, který je propojen s ITR3 a předá tak informaci čítači TIM2. Vygenerovaná hrana je propagována přes TRC až do *Capture/Compare* registru, kde dochází k zachycení hodnoty z *Counter* registru, který je následně díky nastavení slave módu *Combined reset + trigger* zresetován a znovu spuštěn.

Od zmíněné zachycené hodnoty z *Counter* do *Capture/Compare* registru je vyvoláno DMA pro přenos dat z *C/C* registru do paměti. Zde je dobré zmínit, že hodnotu nelze přenášet přímo z *CNT* registru, jelikož je resetován dříve, než dojde k přístupu DMA.

4.1.2 Programová obsluha přímého čítání

Čítač založený na přímé metodě měření frekvence je konfigurován a spuštěn ve výchozím stavu univerzálního čítače. Díky již navržené struktuře programu založené na FreeRTOS stačilo vytvořit vlákno univerzálního čítače (*CounterTask*), které je spolu s ostatními vlákny voláno plánovačem operačního systému. Komunikace mezi vlákny je zajištěna pomocí front. V tomto případě je vyžadována komunikace s parsovacím vlákem (*CmdParserTask*), které zajišťuje interpretaci přijatého příkazu od hosta, a vlákem naopak poskytujícím odesílání dat do PC, tedy směrem k hostovi (*CommTask*).



Obr. 4.5: Vývojový diagram programové obsluhy přímého čítání

Po přijetí příkazu ke startu od hosta je informace dále předána do vlastní fronty univerzálního čítače. Vlákno *CounterTask* pak čeká na přidělený čas procesoru, aby mohlo zapnout měření frekvence. Během měření jsou dle priorit vykonávána ostatní vlákna programu. Přerušení je nastaveno s nižší prioritou než mají vytvořená vlákna, a proto je vyvoláno až po dokončení jejich procesů.

Ve vyvolané obslužné rutině je zjištěna velikost hardwarové předděličky ETRP a následně spočtena frekvence vstupního signálu jako

$$f = \frac{n}{T_g} \cdot ETRP [Hz] \quad (4.3)$$

$$f = \frac{n}{\frac{(ARR+1)(PSC+1)}{TIM4CLK}} \cdot ETRP [Hz], \quad (4.4)$$

kde n je načítaný počet hran vstupního signálu čítačem TIM2. ARR je *Auto-Reload* registr a předdělicí PSC registr vnitřních hodin časovače TIM4. Konstanta $TIM4CLK$ představuje velikost frekvence taktování vnitřních hodin periferie TIM4. Obě výše zmíněné rovnice jsou srovnatelné s rovnicí 4.1. Po provedeném výpočtu hodnoty frekvence je překonfigurována předdělička ETRP tak, aby frekvence signálu za touto předděličkou splnila podmínku $f_{ETRP} < \frac{1}{4} \cdot TIM4CLK$. Nakonec je informace o aktualizaci dat předána do fronty vlákna *CommTask*, aby jím mohla být poslána do PC.

4.1.3 Chyba měření přímého čítání

Přesnost měření čítače je kalkulována PC aplikací a je přímo zobrazována pod měřenou veličinou. Vypočtená chyba čítače reprezentuje absolutní chybu Δf , ve které je zahrnuta chyba kvantizace Δf_q a chyba časové základny Δf_{tb}

$$\Delta f_q = \pm \frac{1}{T_g} \cdot ETRP \quad (4.5)$$

$$\Delta f_{tb} = \pm f \cdot \frac{xtal\ accuracy}{10^6} \quad (4.6)$$

$$\Delta f = \pm \Delta f_q \pm \Delta f_{tb} [Hz], \quad (4.7)$$

kde T_g je doba otevření hradla, f je měřená frekvence a *xtal accuracy* je velikost chyby krystalu (± 50 ppm)

Aplikace též zahrnuje průměrování frekvence s klouzavým bufferem $f_{avg} = \frac{1}{n} \sum_{n=2}^n s_n$. Průměrováním lze snížit chybu kvantizace, nevztahuje se však na chybu časové základny

$$\Delta f_{avg} = \pm \frac{\Delta f_q}{n} \pm \Delta f_{tb} [Hz], \quad (4.8)$$

kde n je počet vzorků na průměr. Navíc se zvyšující se frekvencí začíná převažovat chyba časové základny nad chybou kvantizace a průměrování tak ve smyslu relativní chyby ztrácí vliv. S využitím rovnice 4.6 můžeme jednoduše vypočítat, při jaké vstupní frekvenci bude mít chyba časové základny stejnou hodnotu jako chyba

kvantizační. Počítejme s dobou otevření hradla rovnou 1 s ($\Delta f_q = \pm 1 \text{ Hz}$) a s chybou krystalu 50 ppm. Na levou stranu rovnice si vyjádříme měřenou frekvenci a vypočteme ji:

$$\Delta f = \Delta f_{tb} \cdot \frac{10^6}{xtal \text{ accuracy}} = 1 \cdot \frac{10^6}{50} = 20 \text{ kHz} \quad (4.9)$$

V aplikaci je zobrazena chyba periody, která je vypočtena z průměru frekvence následujícím způsobem:

$$\Delta T_{err} = \pm \left(\frac{1}{f_{avg} - \Delta f_{avg}} - \frac{1}{f_{avg}} \right) \pm \left(\frac{1}{f - \Delta f_{tb}} - \frac{1}{f} \right) [s] \quad (4.10)$$

Čítač byl závěrem otestován s funkčním generátorem Agilent 33250A o rozlišení $1 \mu\text{Hz}$ a maximální frekvenci 80 MHz. Jako reference přesnosti generování byl použit univerzální čítač Agilent 53131A. Naměřené hodnoty nebyly průměrovány.

První sloupec tabulky 4.1 představuje vstupní generovanou frekvenci f_{gen} , druhý měřenou frekvenci f_{meas} , třetí a čtvrtý naměřené absolutní a relativní chyby. Předposledním sloupcem je zobrazena vypočtená absolutní chyba měření čítače, která je rozepsána do absolutní chyby kvantizace Δf_q a absolutní chyby časové základny Δf_{tb} . V posledním sloupci jsou spočítány relativní chyby čítače dle vzorce

$$\delta f = \sqrt{\delta f_q^2 + \delta f_{tb}^2}, \quad (4.11)$$

kde δf_q je relativní chyba kvantizace a δf_{tb} je relativní chyba časové základny.

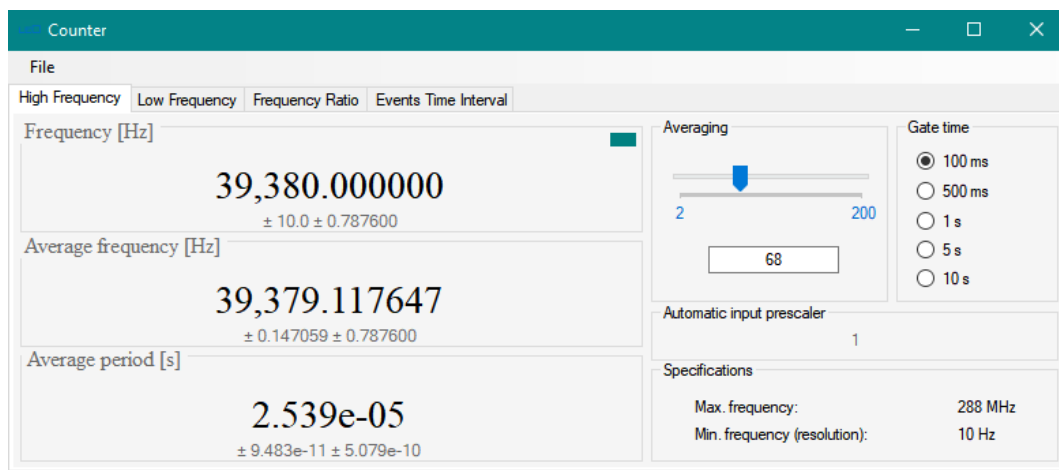
		Chyba měření		Přesnost měření	
$f_{gen} [Hz]$	$f_{meas} [Hz]$	$\Delta f_m [Hz]$	$\delta f_m [\%]$	$\pm \Delta f_q \pm \Delta f_{tb} [Hz]$	$\delta f [\%]$
10,000016	10	1,60e-05	1,60e-04	$\pm 1 \pm 0,0005$	10,000
100,00017	100	1,70e-04	1,70e-04	$\pm 1 \pm 0,0050$	1,000
1,0000017k	1k	1,70e-03	1,70e-04	$\pm 1 \pm 0,0500$	0,100
10,000017k	10k	1,70e-02	1,70e-04	$\pm 1 \pm 0,5000$	0,010
100,00017k	100,002k	1,83e00	1,83e-03	$\pm 1 \pm 5,0001$	0,005
1,0000017M	1,000023M	2,13e01	2,13e-03	$\pm 1 \pm 50,0012$	0,005
10,000017M	10,000234M	2,17e02	2,17e-03	$\pm 1 \pm 500,0117$	0,005
50,000089M	50,001160M	1,07e03	2,14e-03	$\pm 1 \pm 2500,0580$	0,005
80,000136M	80,001844M	1,71e03	2,14e-03	$\pm 1 \pm 4000,0922$	0,005

Tab. 4.1: Naměřené hodnoty čítačem založeným na přímé metodě měření

Měření bylo provedeno při pokojové teplotě 25°C , a tak by hodnota chyby krystalu měla odpovídat přibližně 20 ppm. Aplikace počítá s 50 ppm, proto je vypočtená chyba přibližně dvojnásobek naměřené². Relativní chyba kvantizace začíná být maskována chybou časové základny od hodnoty mezi 10 a 100 kHz (výpočet 4.9).

4.1.4 Grafické rozhraní čítače přímého čítání

Přímá metoda měření je nastavena jako výchozí funkce univerzálního čítače po jeho spuštění a tvoří jednu ze čtyř záložek. Název příslušné záložky je *High Frequency* a jak sám napovídá, hodí se pro měření vyšších frekvencí v porovnání se sousední záložkou *Low frequency*, pod kterou se skrývá čítač založený na reciproční metodě.



Obr. 4.6: Uživatelské rozhraní čítače založeného na přímé metodě měření

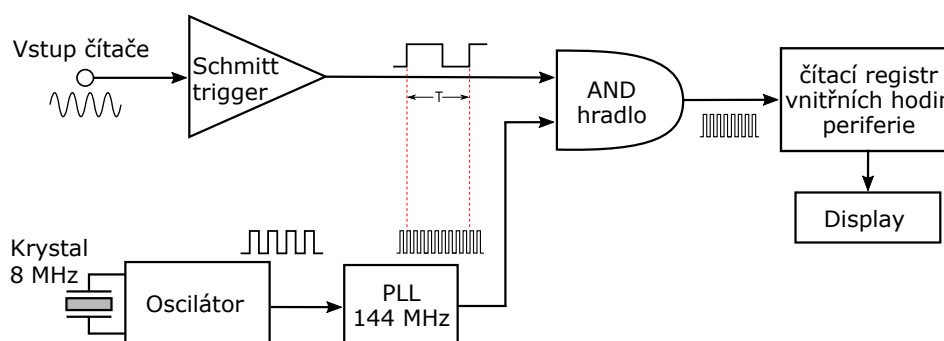
Uživatelské rozhraní zobrazuje frekvenci, průměrnou frekvenci a průměrnou periodu. Pod těmito veličinami jsou zobrazované chyby měření, přičemž první hodnota představuje kvantizační chybu a druhá pak chybu časové základny. V pravé horní části je umístěn jezdec pro průměrování frekvence ze 2 až 200 vzorků a vedle něj se nachází okno s možností výběru doby měření. V pravé dolní části je zobrazována aktuální hodnota automatické předděličky ETRP a rozlišení čítače v závislosti na vybrané době měření. Uživatel je taktéž seznámen s maximální frekvencí, již je čítač schopen měřit.

²Platí pro hodnoty od cca 100 kHz, kde začíná převažovat chyba časové základny nad chybou kvantizace.

4.2 Reciproční metoda měření frekvence

Novějším způsobem měření frekvence oproti přímé metodě je metoda reciproční, která je založena na měření periody signálu. Délka periody signálu, neboli doba otevření hradla periodou signálu, je odvozena od vnitřních hodin mikrokontroléru. Výpočet frekvence je jednoduše proveden obrácením naměřeného časového intervalu.

Tento přístup přináší několik výhod. Jednou z nich je nezávislost kvantizační chyby na vstupní frekvenci signálu, a pokud zanedbáme chybu triggeru a časové základny, tak i nezávislost rozlišení na vstupní frekvenci.



Obr. 4.7: Blokové schéma principu reciproční metody měření frekvence

Z blokového schématu 4.7 je zřejmé, že otevření hradla je v reálném čase přímo řízeno vstupním signálem. Během této doby jsou čítány cykly vnitřních hodin systému. Schéma zobrazuje zjednodušenou verzi čítače, nicméně je možné provádět odběr více period a průměrováním tak zvýšit přesnost měření. Perioda vstupního signálu se pak jednoduše spočte jako

$$T = \frac{t}{m}, \quad (4.12)$$

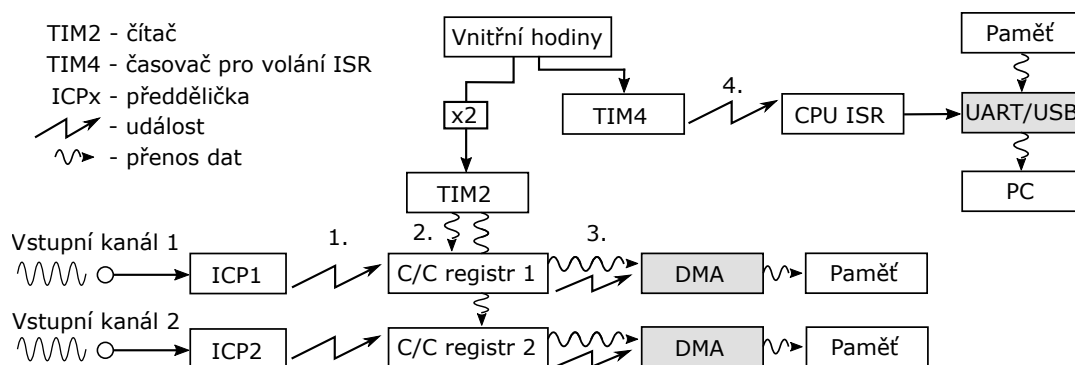
kde t je naměřený celkový čas a m počet načítaných period signálu během této doby.

4.2.1 Obvodové řešení recipročního čítače

Na rozdíl od přímého měření frekvence, implementace návrhu recipročního čítače vyžaduje pouze jeden čítač/časovač pro měření periody, protože sama perioda vstupního signálu určuje dobu měření. Přesto je však využito druhého časovače, jehož funkcí je periodické spouštění přerušení.

Navrženým recipročním čítačem lze měřit frekvenci na dvou kanálech, které jsou konfigurovány jako *Input Capture*³. Oba kanály využívají hardwarové předděličky, která je dle požadavků uživatele konfigurována na 1, 2, 4 a 8. Příchozí hrana signálu vyvolá událost, od které je zachycena hodnota čítacího registru čítače TIM2 do jeho *Capture/Compare* registru. Od události zachycení hodnoty je vyvoláno DMA, které přenesou vzorek z *C/C* registru do paměti. Přijdou-li dvě vzestupné hrany, je znám relativní čas jejich příchodu, a tak i perioda signálu. Posledním elementem je již zmíněný časovač TIM4, jenž periodicky každých 100 ms vyvolává přerušování a s ním i jeho obslužnou rutinu, ve které je zjišťováno, zda-li na daném kanále došlo ke konci měření a případně předává informaci o aktualizaci dat vláknu *CommTask*, které přeposílá naměřené hodnoty frekvence do PC.

1. a 2.) Příchozí hrana signálu vyvolá zachycení hodnoty čítače TIM2 do *C/C* registru.
- 3.) Zachycení hodnoty čítače do *C/C* registru vyvolá DMA přenos dat z *C/C* reg. do paměti.
- 4.) Časovač TIM4 periodicky vyvolává ISR, ve které je zajištěn přenos dat do PC.



Obr. 4.8: Blokové schéma obvodového řešení recipročního čítače

Existuje více způsobů, jakými testovat dokončení odběru vzorků. Mohla by se naskytnout otázka, proč přerušování není vyvoláno přímo událostí od dokončeného přenosu vzorků do paměti. Problém by v tomto případě nastal při měření signálů vyšších frekvencí, které by způsobily plné vytížení mikrořadiče a v případě posílání dat do PC zahlcení komunikace.

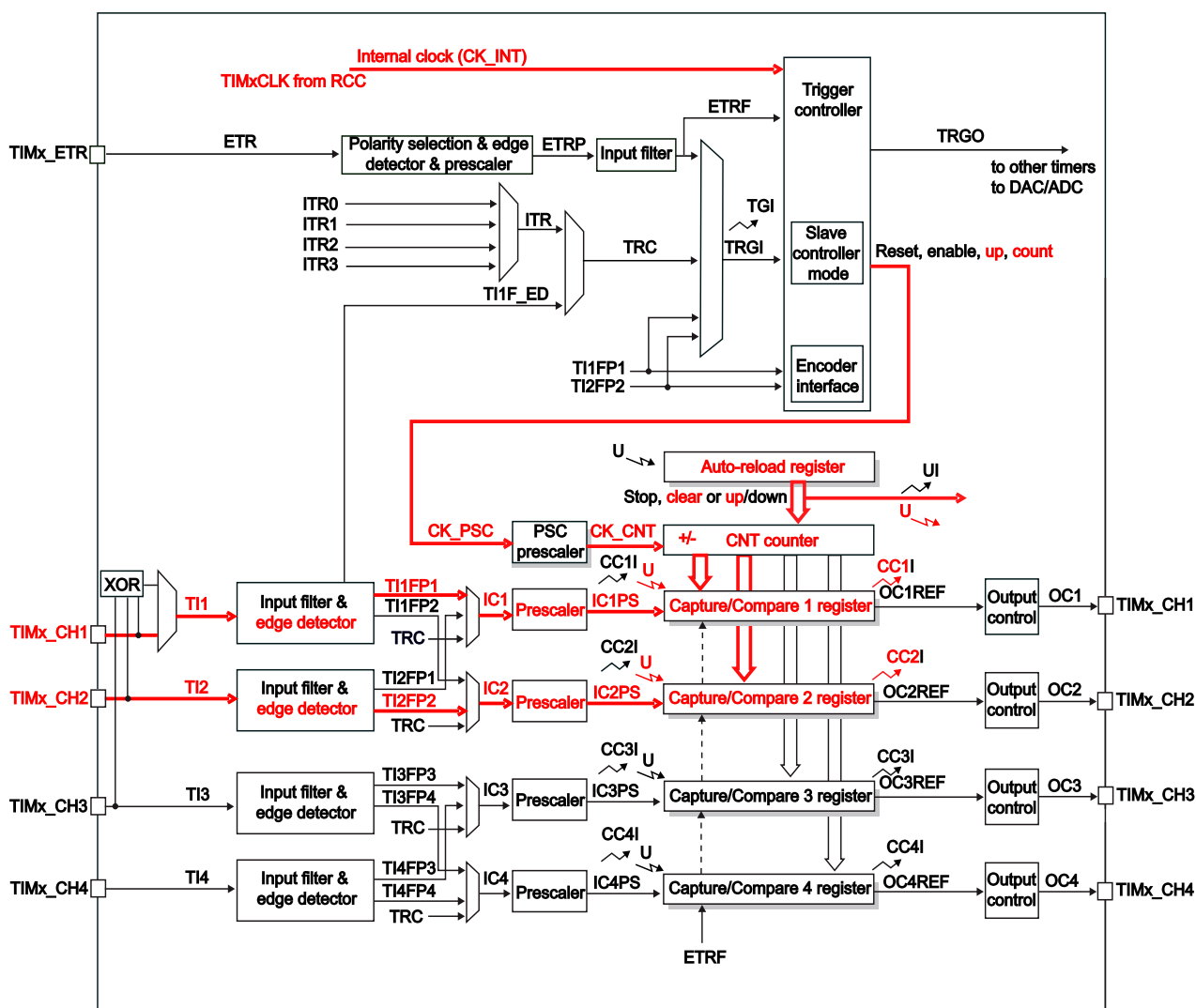
Reciproční čítače obvykle využívají dva čítací registry. Jeden pro čítání vzorků pro průměrování, druhý pro zachycení času vzorku. V tomto čítači je dán počet vzorků velikostí bufferu, do kterého jsou vzorky přenášeny pomocí DMA. Periferie DMA si drží ukazatel na počátek bufferu, její registr *CNDTR* pak obsahuje počet

³V překladu znamená zachycení vstupu. Jedná se o metodu uložení relativního času, ve kterém byla na vstupu zachycena daná událost.

jeho prvků. S každým přenosem je registr dekrementován.

V recipročním měření na rozdíl od přímého měření, je buffer DMA nastaven jako *normal* místo *circular*, čímž je zabráněno zbytečnému vytížení DMA pro frekvence vyšší než 10 Hz. DMA je po dokončení transakce znovu spuštěno v příslušné ISR (vyvolávána každých 100 ms). Velikost bufferu DMA, žádá-li uživatel, může být změněna kdykoli během přenosu vzorků. V tom případě je aktuální transakce přerušena a je započata nová.

Schéma 4.9 zobrazuje vlastní zapojení a vedení signálů v čítači TIM2. Čítač



Obr. 4.9: Schéma vnitřního zapojení periferie TIM2 recipročního čítače [2]

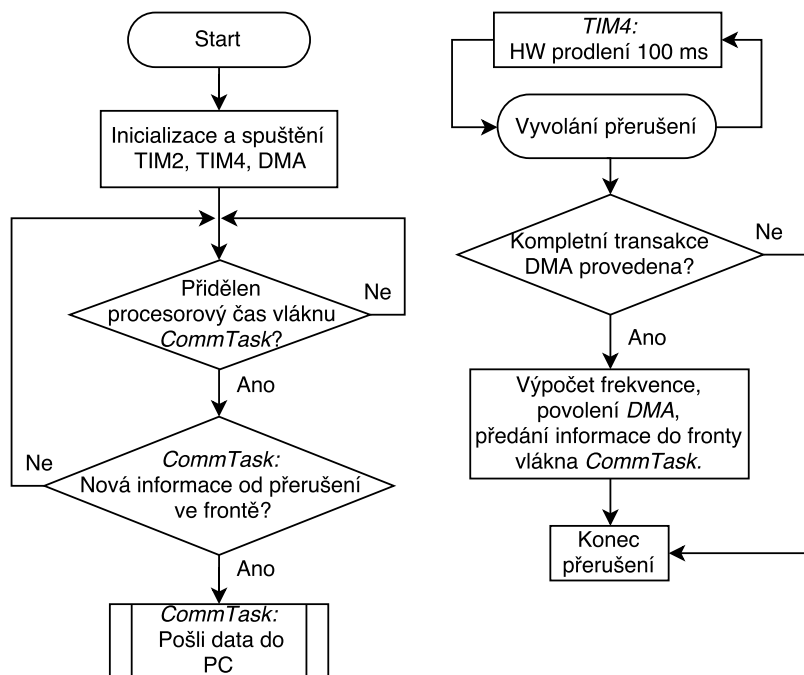
kontinuálně plní svůj CNT registr. Mezitím přichází hrana na jednom z kanálů a

prochází až k ICxPS, kde dochází k události (červené U), od které je zachycena hodnota čítače TIM2 z CNT do C/C registru. Dále pak událost CCxI spouští DMA pro přenos dat do paměti. Po naplnění *Auto-Reload* registru je čítač automaticky resetován a spuštěn znovu.

Pro výpočet jedné periody je třeba přenést alespoň dvě hrany. Časy příchodu hran jsou pak od sebe odečteny. Více informací k softwarovému řešení je poskytnuto následující sekci.

4.2.2 Programová obsluha recipročního čítače

Stejně jako v případě programové obsluhy přímé metody je frekvence vypočtena v příslušné obslužné rutině přerušeni. Zde je však výpočet proveden v přerušeni od časovače TIM4, místo od periferie DMA. V následujícím vývojovém diagramu je zachycena funkce programu od počátku měření.



Obr. 4.10: Vývojový diagram programové obsluhy recipročního čítání

Výpočet frekvence je proveden v obslužné rutině přerušeni níže daným způsobem

$$f = \frac{TIM2CLK \cdot n}{s_n - s_0} \cdot ICxPSC [Hz], \quad (4.13)$$

kde $TIM2CLK$ (144 MHz) je frekvence vnitřních hodin periferie TIM2, s_0 je první a s_n poslední přenesený vzorek vnitřních hodin a n je celkový počet přenesených vzorků. $ICxPSC$ představuje velikost hardwarové předděličky vstupního signálu na

kanále x .

Přestože výpočet vyžaduje jen první a poslední vzorek, periferie DMA musí přenést všechny vzorky až do počtu n . Dvaatřicetibitové pole DMA bufferu, který je určen pro průměrování, tak může zabírat hodně paměti. Proto byla využita hardwarová předdělička signálu, s jejíž pomocí je snížen počet vyvolaných událostí od přijatých hran na polovinu, čtvrtinu nebo osminu. Pro pole bylo staticky alokováno 400 bajtů neboli 100 32-bitových prvků. Díky předděličce lze tedy v aplikaci dosáhnout průměrování z až 800 vzorků.

Maximální měřitelná frekvence je značně omezena chybou kvantizace a rychlostí přenosu vzorků pomocí DMA, jemuž přenos jednoho vzorku trvá několik cyklů systémových hodin. Tento problém se nevyskytuje u přímého měření frekvence, kde je pro výpočet vyžadován jen jeden přenesený vzorek. Recipročním čítačem bylo možno měřit frekvenci přibližně 18 MHz. V aplikaci je však maximální hodnota měřitelné frekvence omezena na 10 MHz, jelikož existuje určitá nižší mez, při které se vyplatí přepnout univerzální čítač do módu přímého měření a dosáhnout tak nižší chyby. Touto problematikou se zabývá podkapitola *Srovnání chyby měření přímého a recipročního čítače* 4.3.

4.2.3 Chyba měření recipročního čítače

Aplikace zobrazuje spočtenou absolutní chybu měření frekvence Δf , která zahrnuje kvantizační chybu Δf_q a chybu časové základny Δf_{tb} . Tato chyba měření je spočtena jako

$$\Delta f = \pm \Delta f_q \pm \Delta f_{tb} = \pm \frac{f}{f_{clk} - f} \cdot f \pm \frac{50}{10^6} \cdot f \text{ [Hz]}, \quad (4.14)$$

kde f je vstupní frekvence a f_{clk} je frekvence vnitřních hodin čítače TIM2. První zlomek zde představuje relativní chybu kvantizace, druhý číselný zlomek pak relativní chybu časové základny.

Zároveň s měřenou hodnotou frekvence je zobrazována vypočtená perioda a s ní i její absolutní chyba ΔT , která je dána jako

$$\Delta T = \pm \left(\frac{1}{f - \Delta f_q} - \frac{1}{f} \right) \pm \left(\frac{1}{f - \Delta f_{tb}} - \frac{1}{f} \right) \text{ [s]}, \quad (4.15)$$

přičemž druhou závorku lze přepsat jako $\Delta T_{tb} = \frac{50}{10^6} \cdot T$.

Čítač byl závěrem otestován se stejnými přístroji jako čítač založený na přímé metodě měření. Tabulka 4.2 obsahuje generované f_{gen} a naměřené hodnoty f_{meas}

frekvencí a absolutní Δf_m a relativní δf_m chyby měření. Taktéž jsou zahrnuty aplikační spočtené chyby, resp. přesnosti měření. První hodnota vypočtené absolutní chyby měření Δf představuje kvantizační chybu Δf_q , druhá pak chybu časové základny Δf_{tb} . V pravém sloupci jsou uvedeny výsledné relativní chyby čítače δf , které byly spočítány jako

$$\delta f = \sqrt{\delta f_q^2 + \delta f_{tb}^2}, \quad (4.16)$$

kde δf_q je relativní chyba kvantizace a δf_{tb} je relativní chyba časové základny.

		Chyba měření		Přesnost měření	
f_{gen} [Hz]	f_{meas} [Hz]	Δf_m [Hz]	δf_m [%]	$\pm \Delta f_q \pm \Delta f_{tb}$ [Hz]	δf [%]
100,001m	100,002m	1,00e-06	9,99e-04	$\pm 6,945e-11 \pm 5,0e-06$	0,005
500,002m	500,012m	1,00e-05	1,99e-03	$\pm 1,736e-09 \pm 2,5e-05$	0,005
1	1,000023	2,30e-05	2,30e-03	$\pm 6,945e-09 \pm 5,0e-05$	0,005
10	10,00023	2,30e-04	2,30e-03	$\pm 6,945e-07 \pm 5,0e-04$	0,005
100,000	100,0023	2,30e-03	2,30e-03	$\pm 6,945e-05 \pm 5,0e-03$	0,005
500,001	500,0122	1,12e-02	2,24e-03	$\pm 1,736e-03 \pm 2,5e-02$	0,005
1,000002k	1,000021k	1,90e-02	1,90e-03	$\pm 6,945e-03 \pm 5,0e-02$	0,005
10,00002k	10,00000k	2,00e-02	1,99e-04	$\pm 6,945e-03 \pm 5,0e-01$	0,005
100,0002k	100k	2,00e-01	1,99e-04	$\pm 6,949e01 \pm 5,0e00$	0,070
1,000001M	1,000000M	1,00e00	1,00e-04	$\pm 6,993e03 \pm 5,0e01$	0,700
2,000004M	2,000000M	4,00e00	2,00e-04	$\pm 2,817e04 \pm 1,0e02$	1,410
4,000007M	4,000000M	7,00e00	1,75e-04	$\pm 1,143e05 \pm 2,0e02$	2,860
8,500015M	9,000000M	4,99e04	5,88e00	$\pm 6,000e05 \pm 4,5e02$	6,670

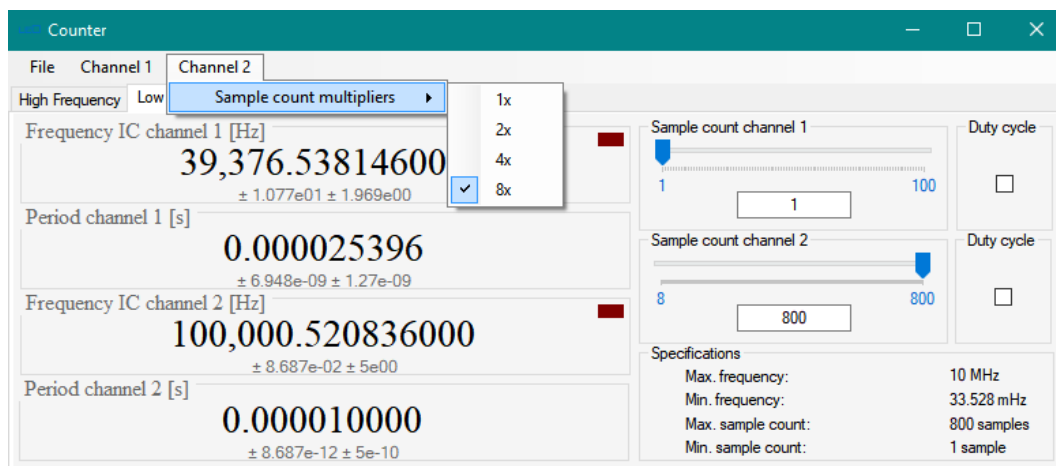
Tab. 4.2: Hodnoty naměřené recipročním čítačem

Přestože hodnoty ve spodní části tabulky jsou správné, jsou měřeny s velkou nepřesností, která je zapříčiněna převážně kvantizační chybou. Univerzální čítač je tak dobré přepnout od dané frekvence na čítač přímého měření. Tuto skutečnost lze ověřit srovnáním chyb obou tabulek 4.1 a 4.2.

4.2.4 Grafické rozhraní recipročního čítače

Čítač založený na principu recipročního měření frekvence lze nalézt pod druhou záložkou grafického uživatelského rozhraní univerzálního čítače. Protože je čítač z principu uzpůsoben k měření spíše nízkých frekvencí, byla tato záložka nazvána *Low Frequency*.

V levé části grafického rozhraní čítače jsou zobrazeny měřené frekvence vstupních signálů, spočtené periody i chyby měření na obou kanálech. V pravé části jsou umístěna táhla pro volbu počtu vzorků, ze kterých je měření průměrováno. Rozsah těchto hodnot je určen předdělicím hardwarovým prvkem, který je zvolen z horního menu čítače. Za táhly průměrování se nacházejí zaškrťovací políčka pro konfiguraci čítače do módu měření střídý a délky pulsu (více informací v podkapitole 4.4). V pravém dolním rohu jsou nakonec zobrazeny specifikace čítače.



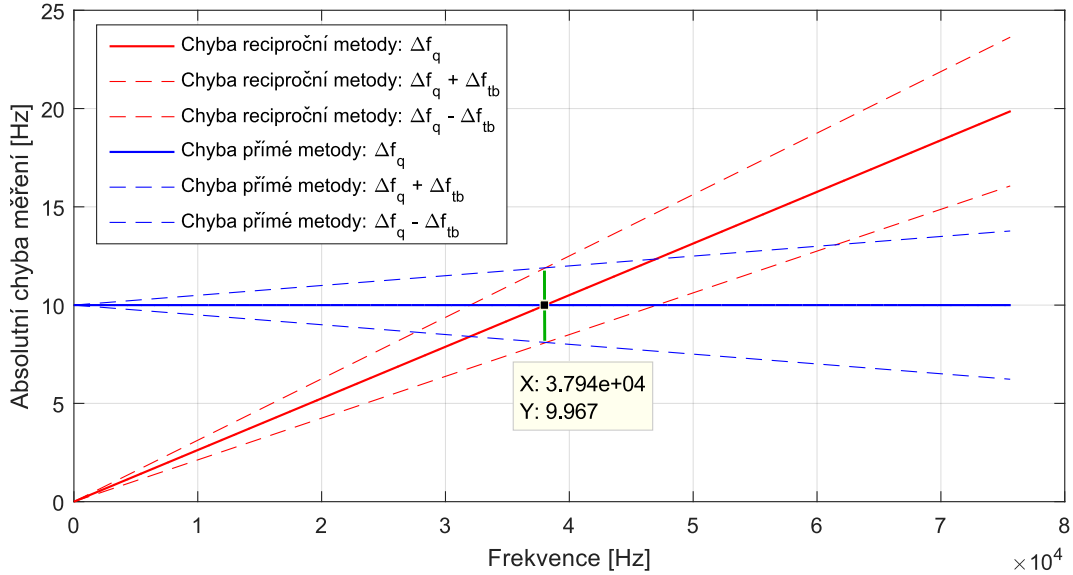
Obr. 4.11: Uživatelské rozhraní čítače založeného na reciproční metodě měření

Čítač zvládá měřit frekvence v rozsahu 33,528 mHz - 10 MHz, přičemž minimum je dáno frekvencí vnitřních hodin čítače TIM2 (144 MHz) a velikostí jeho čítacího registru (32 bitů). Hodnota je spočtena jako $f_{min} = \frac{144 \cdot 10^6}{2^{32}} = 33,5276 \cdot 10^{-3} \text{ Hz}$ a odpovídá periodě signálu 29,8 sekund. Předdělička vnitřních hodin PSC není využita, aby nebyla zvýšena chyba kvantizace. Horní limit měření recipročního čítače byl již zdůvodněn v sekci *Programová obsluha recipročního čítače* 4.2.2.

4.3 Srovnání chyby měření přímého a recipročního čítače

Vestavěné univerzální čítače občas nabízejí možnost automatické rekonfigurace čítače do vhodnějšího módu dle aktuální měřené frekvence, čímž je dosaženo minimalizace chyby. V této práci navržený univerzální čítač nenabízí zmíněnou funkcionalitu, a je tak na samotném uživateli, který je prostřednictvím grafického rozhraní informován o velikosti absolutní chyby, aby si zkontroloval, že využívá v daném rozsahu měřených frekvencí ten správný mód čítače.

Pro představu jsou zde uvedeny hodnoty frekvence udávající mez, při které je z hlediska chyby měření za daných okolností výhodnější čítač přepnout do druhého módu. Pod touto mezí je výhodnější používat reciproční čítač, nad ní pak spíše čítač založený na přímé metodě měření.



Obr. 4.12: Srovnání přesnosti měření recipročního a přímého čítače ($T_g = 100$ ms)

Následujícími výpočty je dosaženo souřadnic průsečíku dvou lineárních charakteristik (z obrázku 4.12), které představují příslušné chyby obou metod. V potaz je brána pouze kvantizační chyba recipročního čítače, stejně jako čítače založeného na přímé metodě. Chyba se však může pohybovat po celé délce zeleně vyznačené přímky, je-li brána v ohled i odchylka frekvence časové základny od nominální hodnoty. Samotná chyba kvantizace přímé metody je konstantní a je dána pouze dobou otevření hradla. Výpočet je proveden pro dobu otevření hradla přímého čítače $T_g = 0,1$ s a není zahrnuto průměrování ani využití předděliček.

Vezměme nejprve kvantizační chyby obou stanovených rovnic přímé 4.7 a reciproční 4.14 metody a dejme je do rovnosti

$$\frac{f}{f_{clk} - f} \cdot f = \frac{1}{T_g} \quad (4.17)$$

Jednoduchou algebraickou úpravou a dosazením dostaneme následující kvadratickou rovnici

$$f^2 + 10f - 1,44 \cdot 10^9 = 0, \quad (4.18)$$

jejímž kořenem je $f = 37942,3$ Hz. Zpětným dosazením kořene do jedné z rovnic dostáváme souřadnici osy y , tzn. chybu měření $\Delta f = 10$ Hz. Stejným způsobem

dojdeme k dalším výsledkům. Souhrn je zobrazen tabulkou 4.3.

T_g [s]	f [Hz]	Δf [Hz]
0,1	37942,3	10,0
1,0	11999,5	1,0
10,0	3794,7	0,1

Tab. 4.3: Vypočtené mezní frekvence, při kterých přímý a reciproční čítač dosahují stejné chyby měření

4.4 Měření střídy a délky pulsu

Nejprve definujme střídu jako část periody, ve které je signál aktivní. Standardně se střída D^4 udává v procentech jako poměr aktivní části periody PW^5 ku periodě signálu T :

$$D = \frac{PW}{T} \cdot 100 [\%] \quad (4.19)$$

Chyba měření v této části není uvedena a není ani počítána a zobrazována v příslušném modulu univerzálního čítače. Otevření hradla je zde řízeno hranami vstupního signálu (*Input Capture*), a tak se projevují chyba kvantizace i chyba časové základny. Výpočet by byl srovnatelný s výpočtem chyby recipročního čítače.

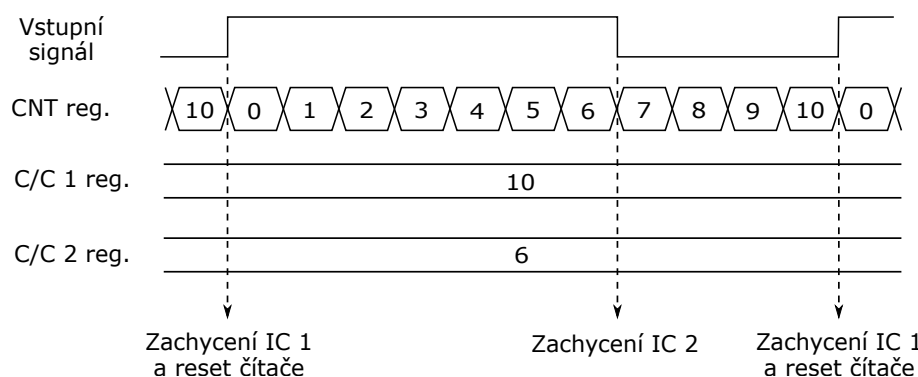
4.4.1 Obvodové řešení měření střídy a délky pulsu

Obvodové řešení čítače pro měření střídy a délky pulsu je stejně jako reciproční metoda založeno na realizaci vstupu *Input Capture*, díky němuž lze použít tzv. *PWM input mode*. V tomto módu je pouze jeden kanál/vstup namapován na dva *Capture/Compare* registry. První *C/C* registr reaguje na náběžnou hranu, od které je nejprve přenesena hodnota čítače do paměti za pomoci DMA a následně je čítač zresetován a spuštěn znovu. Druhý *C/C* registr reaguje na sestupnou hranu, od které je pouze přenesena hodnota čítače do paměti. V paměti jsou tedy uloženy dvě hodnoty; čas příchodu sestupné hrany a pozdější čas příchodu hrany vzestupné. Obrázek 4.13 dokumentuje časování v tomto módu.

Zmíněný přístup značně zjednodušuje pozdější výpočet požadovaných veličin. První zachycený vzorek od vzestupné hrany (po inicializaci a následném spuštění čítače) sice nemusí být z principu funkce správný a musí být zahozen, nicméně další vzorky jsou již správné a díky resetu čítače příchozí vzestupnou hranou je délka

⁴Z anglického *Duty cycle*.

⁵Z anglického *Pulse Width*.

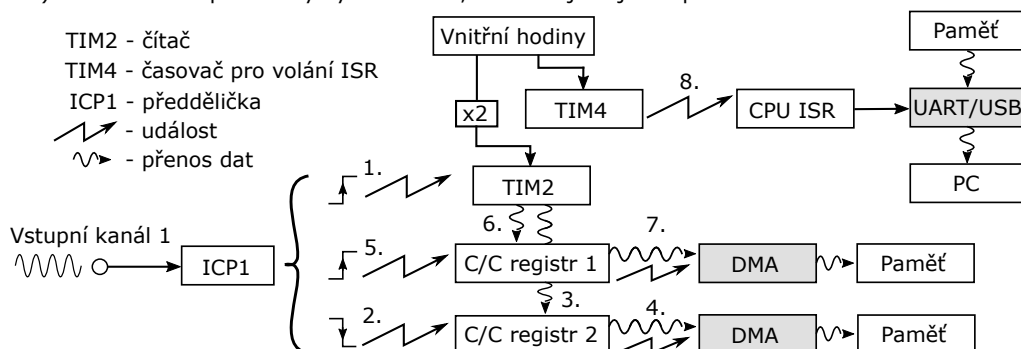


Obr. 4.13: Znázornění aktivity vstupu *Input Capture v PWM input mode*

pulsu určena pouze jedinou přenesenou hodnotou. Další zachycená a přenesená hodnota pak představuje periodu signálu. Střída je dána poměrem těchto dvou hodnot. Nevýhodou implementace je fakt, že nelze měřit střídu a délku pulsu na dvou použitých kanálech zároveň, jelikož vyžaduje dvou *C/C* registrů.

Blokové schéma 4.14 přesněji zobrazuje princip funkce měření střídy a délky pulsu. Kromě mapování vstupu na dva *C/C* registry je schéma srovnatelné s blokovým schématem recipročního čítače 4.7. Buffer DMA je zde znovu nastaven jako

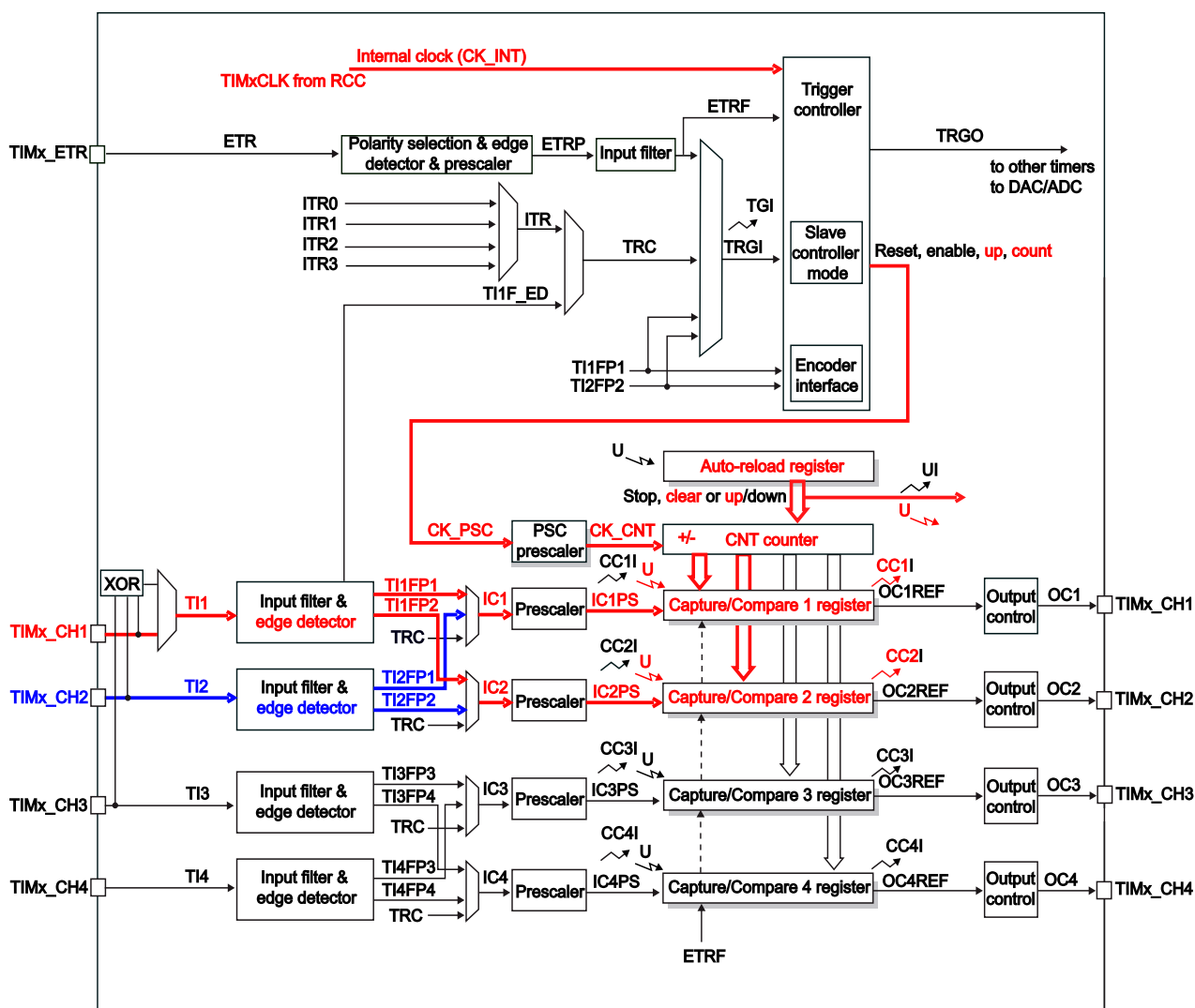
- 1.) Příchozí vzestupná hrana signálu vyvolá reset a spuštění čítače TIM2.
- 2.) Příchozí sestupná hrana signálu vyvolá (3.) zachycení hodnoty čítače TIM2 do *C/C* registru 2.
- 3.) Zachycení hodnoty čítače do *C/C* reg. 2 vyvolá (4.) DMA přenos dat z *C/C* reg. 2 do paměti.
- 5.) Příchozí vzestupná hrana signálu vyvolá (6.) zachycení hodnoty čítače TIM2 do *C/C* registru 1.
- 6.) Zachycení hodnoty čítače do *C/C* reg. 1 vyvolá (7.) DMA přenos dat z *C/C* reg. 1 do paměti.
- 8.) Časovač TIM4 periodicky vyvolává ISR, ve které je zajištěn přenos dat do PC.



Obr. 4.14: Blokové schéma obvodového řešení měření střídy a délky pulsu

normal, aby nedocházelo k přepisování hodnot a potenciálně tak nemohlo dojít k nesprávným výpočtům. Periferie DMA je vždy v příslušné obslužné rutině po zdařilém přenosu vzorků znovu spuštěna. Stejně tak časovač TIM4 je zde použit pro periodické spouštění přerušování každých 100 ms. V obslužné rutině je pak testováno, zda-li

již došlo k přenosu požadovaných vzorků.

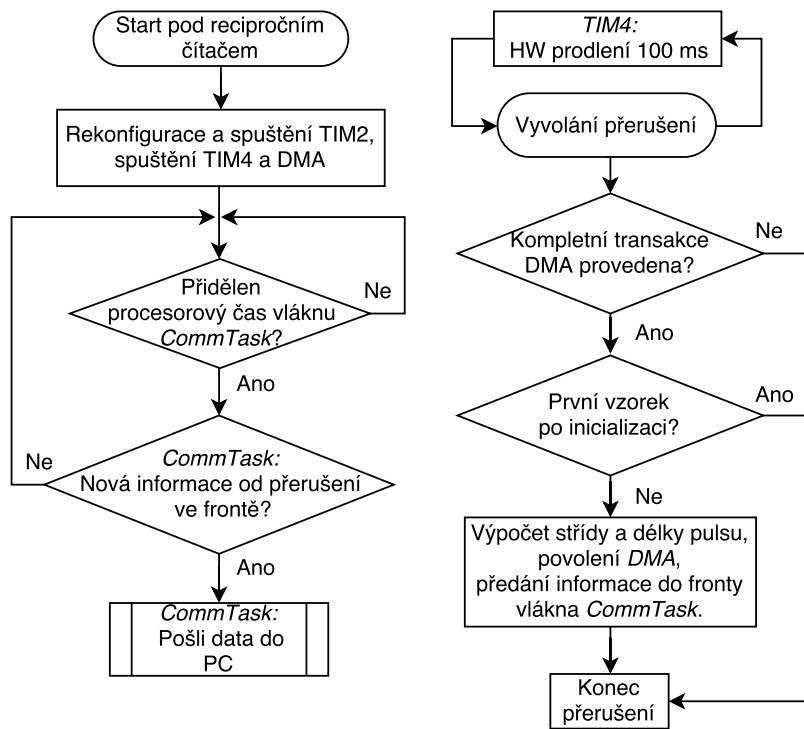


Obr. 4.15: Schéma vnitřního propojení TIM2 pro měření střídy a délky pulsu [2]

Schéma na obrázku 4.15 podrobněji vykresluje mapování vstupu TI1 nebo TI2 na dva *C/C* registry čítače TIM2. Mapování vstupu TI1 na *C/C* 1 registr je umožněno prostřednictvím signálů TI1FP1 a TI1FP2. Konfigurace pro měření obou veličin na vstupu TI2 lze dosáhnout prostřednictvím signálů TI2FP1 a TI2FP2. Tato možnost je ve schématu vyznačena modrou barvou. Jelikož je měření založeno na principu *Input Capture*, vnitřní propojení z větší části kopíruje schéma recipročního čítače 4.9.

4.4.2 Programová obsluha měření střídý a délky pulsu

Řešení programové obsluhy čítače měřícího střídý a délku pulsu je v nejvyšší vrstvě firmwaru podobné jako řešení programové obsluhy recipročního čítače a je zobrazeno podobným vývojovým diagramem 4.16. Díky stejnému způsobu zachycení vstupu pomocí *Input Capture* je přepnutí do módu měření střídý a délky pulsu jen otázkou jednoduché sekvence příkazů konfigurace registrů periferie TIM2. Tato sekvence je dobře popsána v referenčním manuálu [2] sekcí 21.3.6 *PWM input mode*. Této skutečnosti bylo využito, a tak je mód měření střídý a délky pulsu konfigurován pod recipročním čítačem v grafickém rozhraní pod záložkou *Low Frequency*.



Obr. 4.16: Vývojový diagram programové obsluhy měření střídý a délky pulsu

Výpočet střídý D a délky pulsu P je řešen v obslužné rutině přerušení vyvolané od naplnění časovače TIM4 jako

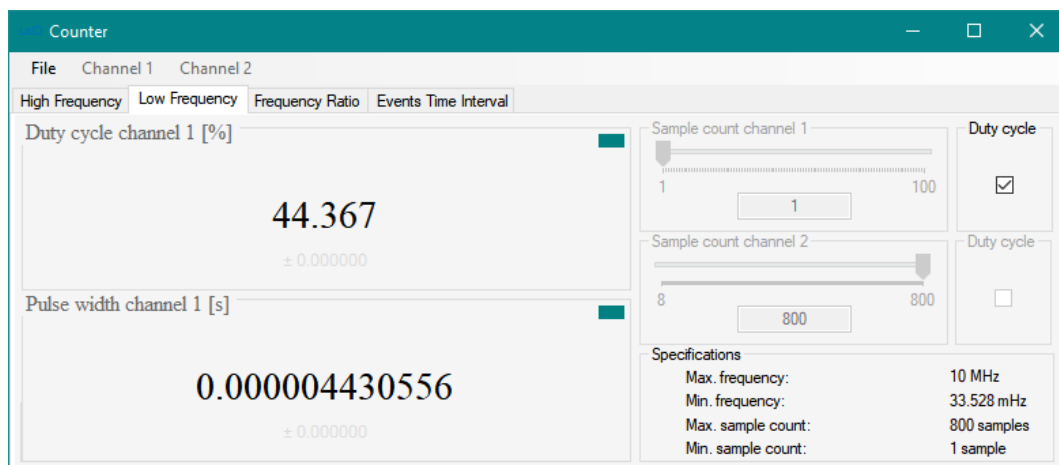
$$D = \frac{PW}{T} \cdot 100 = \frac{s_1}{s_2} \cdot 100 [\%] \quad (4.20)$$

$$P = \frac{PW}{f_{clk}} = \frac{s_1}{TIM2CLK} [s], \quad (4.21)$$

kde s_1 je přenesená hodnota čítacího registru v čase příchodu sestupné hrany a s_2 v čase příchodu vzestupné hrany. Konstanta $TIM2CLK$ představuje frekvenci hodin periferie TIM2.

4.4.3 Grafické rozhraní pro měření střídy a délky pulsu

V grafickém uživatelském rozhraní lze funkci měření střídy a délky pulsu najít pod čítačem založeném na reciproční metodě měření frekvence, pod záložkou *Low Frequency*. V pravé části GUI se u obou vstupních kanálů za táhly průměrování nachází zaškrtačací políčko *Duty cycle*, pomocí něhož je mikrokontrolér spolu s grafickým rozhraním překonfigurován do požadovaného módu. V levé části tohoto módu jsou zobrazeny měřené veličiny. Zpět na měření frekvence se lze dostat změnou zaškrtačacího tlačítka do původního stavu.



Obr. 4.17: Uživatelské rozhraní čítače v módu měření střídy a délky pulsu

4.5 Měření poměru frekvencí

Aplikace nabízí mód měření poměru dvou vstupních frekvencí; frekvence f_A vstupního signálu a frekvence f_B druhého vstupního signálu:

$$ratio = \frac{f_A}{f_B} \quad (4.22)$$

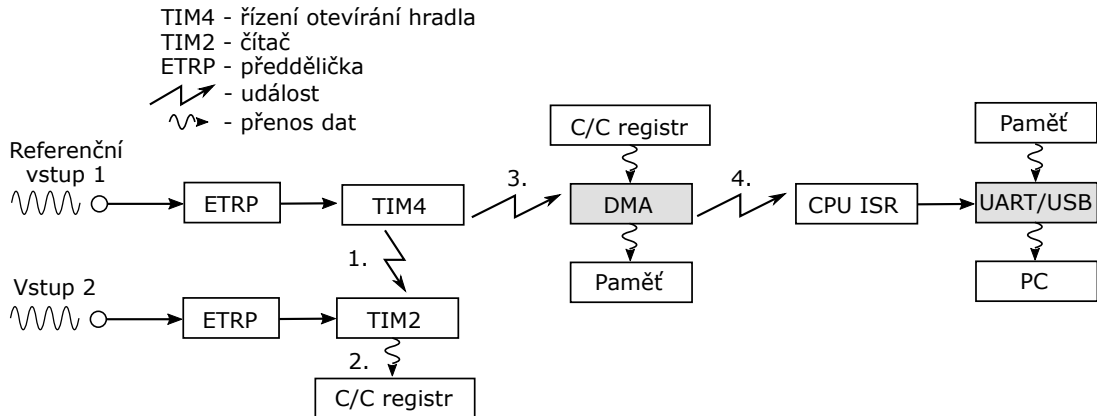
Díky matematické operaci dělení se neuplatní chyba časové základny. S vyžitím přesného externího zdroje hodin tak lze dosahovat dobrých výsledků měření, ve kterých je zahrnuta pouze chyba kvantizace.

4.5.1 Obvodové řešení měření poměru frekvencí

Obvodové řešení měření poměru frekvencí lze považovat za téměř stejné jako v čítači založeném na přímé metodě. Jediným rozdílem je zdroj hodin periferie TIM4, která zde plní jak funkci časovače, tak i funkci čítače. Jejím zdrojem nejsou vnitřní hodiny mikrořadiče, nýbrž externí signál, který je přiveden přes vstup ETR do příslušného

CNT registru.

- 1.) Reset, aktualizace registrů a trigger čítače TIM2 od události naplnění časovače/čítače TIM4.
- 2.) Zachycení hodnoty čítače TIM2 do Capture/Compare registru.
- 3.) Spuštění DMA od události naplnění časovače TIM4.
- 4.) Vyvolání přerušování od dokončeného přenosu dat z Capture/Compare registru do paměti.



Obr. 4.18: Blokové schéma obvodového řešení čítače měřícího poměr frekvencí

Máme zde dva čítače TIM2 a TIM4, které jsou plněny současně dvěma různými vnějšími signály. Jakmile dojde k naplnění čítače/časovače TIM4 od referenčního signálu, dochází k zachycení hodnoty *CNT* registru čítače TIM2 do jeho *Capture/Compare* registru. Hodnota je pak standardně přenesena pomocí DMA do paměti. Díky nastavení čítače TIM2 ve slave módu *Combined reset + trigger* (TIM4 jako jeho master) je *CNT* registr resetován a spuštěn znovu ve stejnou chvíli jako TIM4. Měření tak dál pokračuje bez vnějšího zásahu softwaru.

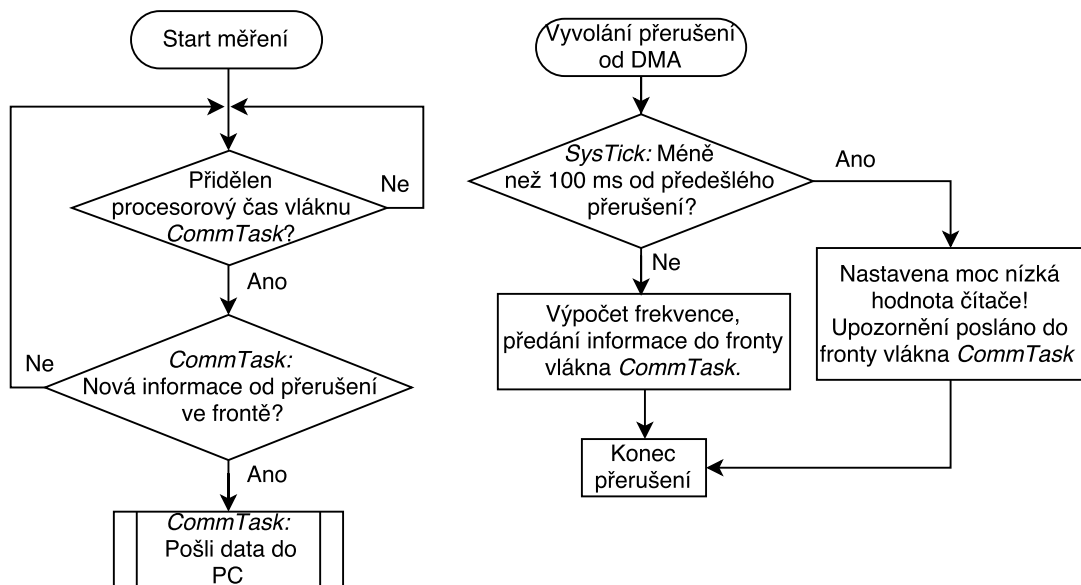
Jako reference, která čítá vždy až do požadovaného počtu vzorků, byl zvolen čítač/časovač TIM4, jehož vnitřní hodiny jsou oproti TIM2 nastaveny na nižší frekvenci 72 MHz. Neznámá frekvence je tak přivedena na čítač TIM2, jehož vnitřní hodiny činí 144 MHz a je tak dosaženo lepšího rozlišení vstupního signálu. Vstup periferie TIM2 ani TIM4 v tomto případě nevyužívá hardwarové předděličky ETRP, tudíž je na vstupy možné přivést signál o maximální frekvenci jedné čtvrtiny vnitřních hodin TIM2 (36 MHz) a TIM4 (18 MHz).

Schéma vnitřního zapojení čítače TIM2 zde není uvedeno. Nicméně obrázek 4.4 z podkapitoly přímého čítače může být použit jako reference. Červená a zelená barva zobrazují zapojení TIM2, samotná modrá pak částečné zapojení TIM4. Modře vyznačený vstup systémových hodin CK_INT čítače/časovače TIM4 by v tomto případě nebyl vyznačen, jelikož je zde taktován vnějším signálem přivedeným na

vstup ETR a dále veden přes své signály ETRP, TRGI, CK_PSC, CK_CNT až do svého vlastního *CNT* registru.

4.5.2 Programová obsluha měření poměru frekvencí

Metoda měření poměru frekvencí využívá v softwarovém řešení stejné obslužné rutiny vyvolané od DMA jako metoda přímého měření. V metodě přímého měření je však velikost čítače/časovače TIM4 (doba otevření hradla) nastavena uživatelem předdefinovanými velikostmi a její minimální hodnota je 100 ms. V metodě měření poměru frekvencí je však velikost naplnění čítače/časovače TIM4 zvolena svévolně uživatelem. Může se tak stát, že uživatel zvolí nízkou hodnotu plnění pro vysokou vstupní frekvenci a DMA bude vyvoláváno s velmi krátkou periodou. Výpočetní jádro mikrořadiče by bylo vytížené a komunikace zahlcena. Proto je zde za pomoci *SysTicku* testována perioda vyvolání obslužné rutiny a jakmile je menší než 100 ms, čítač/časovač TIM4 je zastaven a uživatel informován, že je třeba zvýšit počet požadovaných vzorků čítaných periferií TIM4. Tím je dosaženo delší doby měření.



Obr. 4.19: Vývojový diagram programové obsluhy měření poměru frekvencí

Obslužná rutina přerušení předává informaci komunikačnímu vláknům *CommTask*, že je velikost čítače/časovače TIM4 nastavena na moc nízkou hodnotu, anebo že uložený vzorek může být odeslán do PC. Výpočet poměru frekvencí je proveden až v PC aplikaci jako

$$r = \frac{n_{in}}{n_{ref}} = \frac{n_{TIM2}}{n_{TIM4}} [-], \quad (4.23)$$

kde n_{ref} je uživatelem požadovaný počet vzorků načítaný referenčním čítačem/časovačem TIM4 a n_{in} je načítaný počet vzorků čítačem TIM2.

4.5.3 Chyba měření poměru frekvencí

Chybu poměru dvou frekvencí si lze jednoduše představit jako chybu kvantizace způsobenou vzorkováním signálu na druhém kanále než referenčním. Počet vzorků načítaný na referenčním kanále je předem dán a určuje tak dle vstupní frekvence dobu otevření hradla druhého kanálu. Absolutní chybu měření poměru frekvencí lze jednoduše počítat jako

$$\Delta r_q = \pm \left(\frac{n_{in}}{n_{ref}} - \frac{n_{in} \pm 1}{n_{ref}} \right) = \pm \frac{1}{n_{ref}} [-], \quad (4.24)$$

kde n_{ref} je předem známý počet vzorků načítaný referenčním vstupem a n_{in} počet vzorků načítaný druhým vstupem.

Výpočet daný rovnicí 4.24 je plně platný pro $n_{ref} > n_{in}$. Pro $n_{ref} < n_{in}$ platí také, ale čím větší je v tomto případě rozdíl mezi n_{ref} a n_{in} , tím větší rozsah vzorků od n_{in} může být načten⁶. Kvantizační chyba tak může být násobena.

Samotný trigger může taktéž způsobit chybu. Bohužel uživatelská příručka [1] nestanoví propagační zpoždění integrovaného obvodu *Schmitt trigger*, a tak není v programu počítána.

Následující tabulka 4.4 dokumentuje výsledky měření poměru dvou frekvencí generovaných dvěma stejnými funkčními generátory Agilent 33250A, jejichž specifikace stanoví chybu časové základny na 2 ppm⁷. Tabulka obsahuje dvě generované vstupní frekvence; referenční f_{ref} a frekvenci f_{in} . Za nimi následuje naměřená hodnota poměru frekvencí $r = \frac{f_{in}}{f_{ref}}$ a počet vzorků n_{ref} načítaný od referenčního vstupu. Pravá část tabulky zobrazuje absolutní a relativní chyby měření a přesnosti měření vypočtené PC aplikací dle vzorce 4.24. Pro dosažení co nejlepších výsledků je dobré nastavit co největší referenční počet vzorků n_{ref} , nebo taktéž přivést nižší ze dvou frekvencí na referenční vstup.

⁶Frekvence signálu nemusí být vždy zcela stabilní.

⁷Chyba však může být větší, vezmeme-li v potaz stáří obou generátorů, s nímž souvisí i stárnutí vnitřního krystalu.

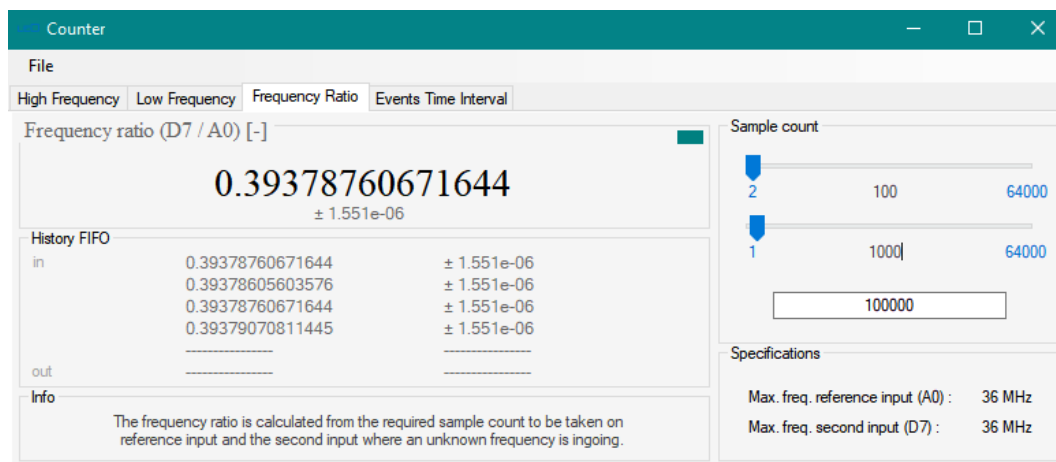
				Chyba měření		Přesnost měření	
f_{ref} [Hz]	f_{in} [Hz]	r [-]	n_{ref} [-]	Δr_m [-]	δr_m [%]	Δr_q [-]	δr_q [%]
10	1	0,080000	25	4,00e-02	40,00	$\pm 4,00e-02$	40,00
10	1	0,120000	25	4,00e-02	40,00	$\pm 4,00e-02$	40,00
10	10	1,000000	25	0	0	$\pm 4,00e-02$	4,00
10	100	10,00000	25	0	0	$\pm 4,00e-02$	0,40
10	1,01k	101,0000	25	0	0	$\pm 4,00e-02$	0,04
10	11	1,100000	10	0	0	$\pm 1,00e-01$	9,10
10	11	1,090909	11	9,09e-03	0,83	$\pm 9,09e-02$	8,30
11	10	0,900000	10	9,09e-03	1,01	$\pm 1,00e-01$	11,10
11	10	0,909090	11	0	0	$\pm 9,09e-02$	8,30
100k	10	0,000100	100k	0	0	$\pm 1,00e-05$	10,00
100k	1k	0,010000	1M	0	0	$\pm 1,00e-06$	0,01
100k	100k	1,000000	10k	0	0	$\pm 1,00e-04$	0,01
100k	1M	10,000030	100k	3,00e-05	3,00e-06	$\pm 1,00e-05$	0,01
100k	10M	10,000320	100k	3,20e-04	3,00e-06	$\pm 1,00e-05$	0,01
1,050k	20	0,019000	2k	4,76e-05	0,25	$\pm 5,00e-04$	2,60
1,070k	20	0,018500	2k	1,92e-04	1,04	$\pm 5,00e-04$	2,70
32,077M	32,077M	1,00000310	20M	3,10e-06	3,10e-06	$\pm 5,00e-08$	5,00e-06
32,077M	32,077M	1,00000315	20M	3,15e-06	3,15e-06	$\pm 5,00e-08$	5,00e-06
30M	10	0,00000330	20M	3,33e-08	1,01	$\pm 5,00e-08$	1,5
10	30M	3000,00946k	100	9,46e-03	3,15e-06	$\pm 1,00e-02$	3,33e-04

Tab. 4.4: Naměřené hodnoty poměru dvou vstupních frekvencí

4.5.4 Grafické rozhraní pro měření poměru frekvencí

Mód měření poměru frekvencí je schován pod záložkou *Frequency Ratio* navrženého GUI univerzálního čítače. V levé části se nachází výpočet poměru frekvencí, historie měření a krátký popis funkce. V pravé části lze volit požadovaný počet vzorků nabraných referenčním čítačem/časovačem. Hodnoty dvou táhel jsou mezi sebou násobeny, proto lze dosáhnout velikosti až $64000 \cdot 64000 = 4096 M$ vzorků. Hodnota prvního táhla je zapsána do registru *ARR*, hodnota druhého táhla do registru *PSC*. V pravém dolním rohu jsou zmíněny maximální vstupní frekvence, se kterými čítač dokáže pracovat.

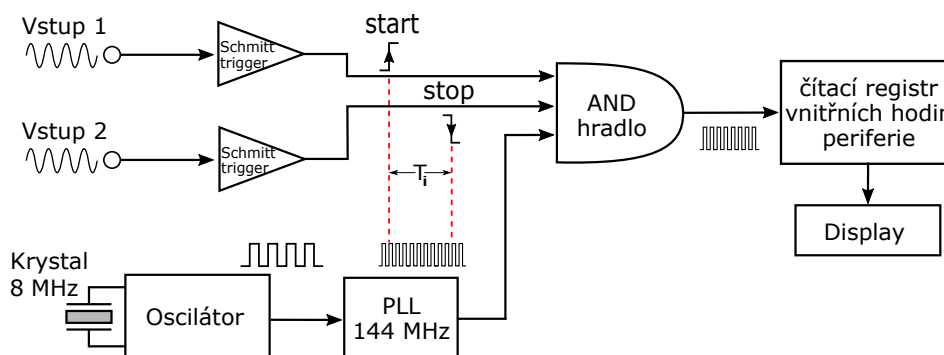
Výběrem záložky *Frequency Ratio* je započato měření poměru frekvencí. Ve výchozím stavu měření poměru frekvencí je nastavena vysoká hodnota požadovaných



Obr. 4.20: Uživatelské rozhraní univ. čítače v módu měření poměru dvou frekvencí vzorků, a tak je uživatel nejprve požádán o změnu její velikosti.

4.6 Měření časového intervalu mezi dvěma událostmi

Poslední realizovanou funkcionalitou navrženého univerzálního čítače je mód pro měření časového intervalu mezi dvěma událostmi, které nastanou na dvou nezávislých kanálech. Událostí je zde myšlena vzestupná či sestupná hrana signálu. Zahajující příchozí událostí je započato čítání vnitřních hodin mikrořadiče, ukončující událost pak zavírá hlavní bránu a ukončuje čítání. Načítaný počet cyklů vnitřních hodin mezi těmito dvěma událostmi představuje měřený časový interval T_i .

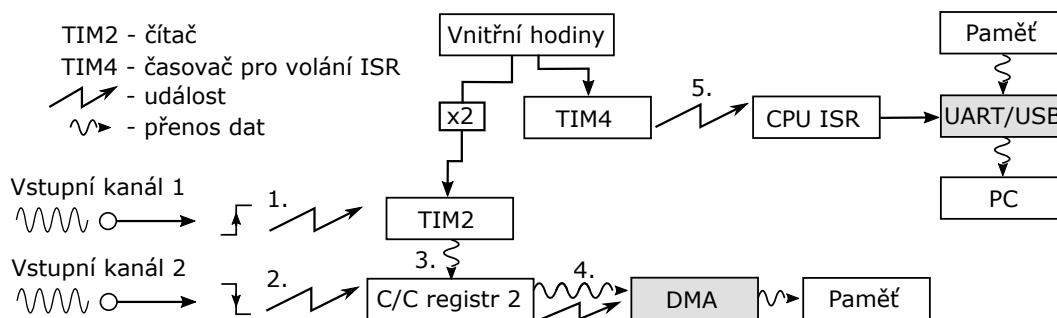


Obr. 4.21: Blokové schéma principu měření časového intervalu

4.6.1 Obvodové řešení měření časového intervalu

Obvodové řešení čítače pro měření časového intervalu lze chápat jako kombinaci vnitřního zapojení recipročního čítače a způsobu implementace měření střídy. Vnitřní zapojení čítače TIM2 je zcela totožné se zapojením TIM2 recipročního čítače (viz. schéma 4.9), tzn. oba dva vstupy TI1 a TI2 jsou mapovány přes vnitřní signály TI1FP1 a TI2FP2 na své vlastní *Capture/Compare* registry 1 a 2. Způsob resetování a zachytávání času příchodu hrany je stejný jako v případě měření střídy a délky pulsu.

- 1.) Příchozí spouštěcí událost na kanálu 1 vyvolá reset a spuštění čítače TIM2.
- 2.) Příchozí událost na kanálu 2 vyvolá (3.) zachycení hodnoty čítače TIM2 do C/C registru 2.
- 3.) Zachycení hodnoty čítače do C/C reg. 2 vyvolá (4.) DMA přenos dat z C/C reg. 2 do paměti.
- 5.) Časovač TIM4 periodicky vyvolává ISR, ve které je zajištěn přenos dat do PC.

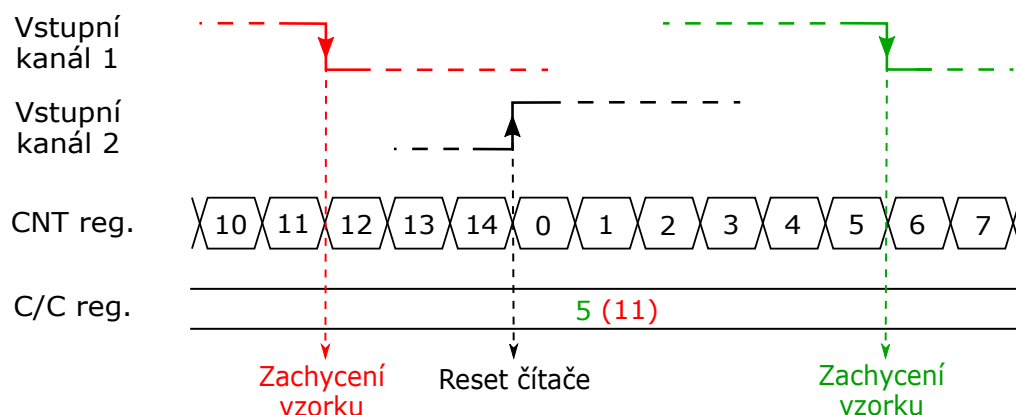


Obr. 4.22: Blokové schéma obvodového řešení čítače měřícího časový interval v *Dependent* módu

Čítač TIM2 je zde nastaven ve slave módu *Combined reset + trigger* a jeden ze dvou vstupních kanálů jako jeho master. Uživatel si vybere, od které události a od kterého ze dvou kanálů bude čítač resetován a následně spuštěn. Čítač tak po první spouštěcí události čítá od nuly a druhá událost na druhém kanále může kdykoli spustit proces zachycení hodnoty registru *CNT* do *C/C* reg. s následným přenesením časového intervalu do paměti za použití DMA. Buffer DMA je zde nakonfigurován na *normal* a přenáší pouze jediný vzorek. Výhodou celého návrhu je, že lze měřit časový interval mezi dvěma příchozími hranami bez nutnosti příchodu dalších hran.

Tento návrh však přináší jeden problém. Přejde-li jako první událost, od které má být přenesen časový interval do paměti, je přenesena falešná hodnota. Následný reset čítače původně zamýšlenou první událostí je již bezvýznamný (viz. obrázek 4.23). Je-li ale předem uživatel obeznámen, v jakém pořadí události nastanou, lze tímto přístupem dosáhnout měření nejkratšího časového intervalu, který periferie TIM2 a hardwarové provedení vstupů umožňuje. Tento způsob byl nazván jako *User*

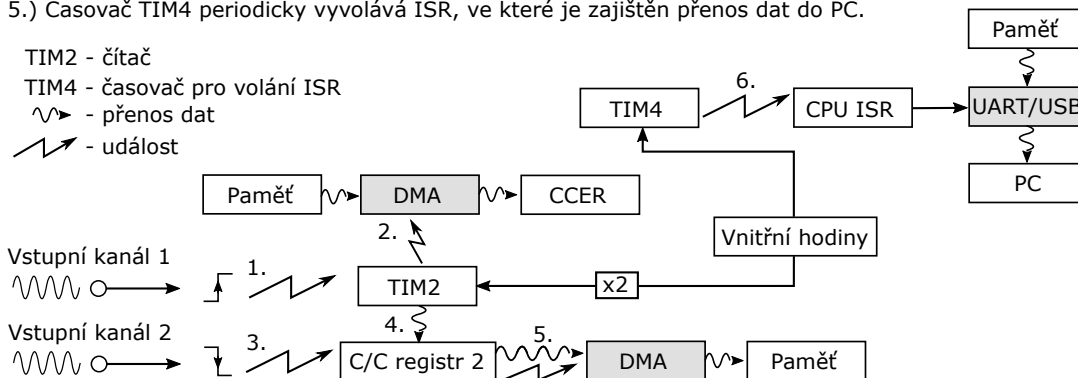
dependent, v aplikaci označován jako *Dependent* a byl podnětem k vývoji *Independent* módu.



Obr. 4.23: Důsledek obrácené sekvence příchozích událostí v *Dependent* módu

Independent mód se liší tím, že povoluje vzorkování vstupního kanálu až po resetu čítače. K tomuto účelu je využito druhé DMA, které je spuštěno událostí resetu čítače a přenáší do registru *CCER* periferie TIM2 takovou hodnotu, kterou je vzorkování na daném kanále povoleno. Nevýhodou tohoto přístupu je určité prodloužení mezi resetem čítače a povolením vzorkování, které je dáno dobou přenosu dat pomocí DMA. Výhodou je fakt, že nezáleží na sekvenci příchozích událostí a uživatel je tak zbaven zodpovědnosti. Obrázek 4.24 zobrazuje blokové schéma řešení čítače měřícího časový interval mezi dvěma událostmi v *Independent* módu.

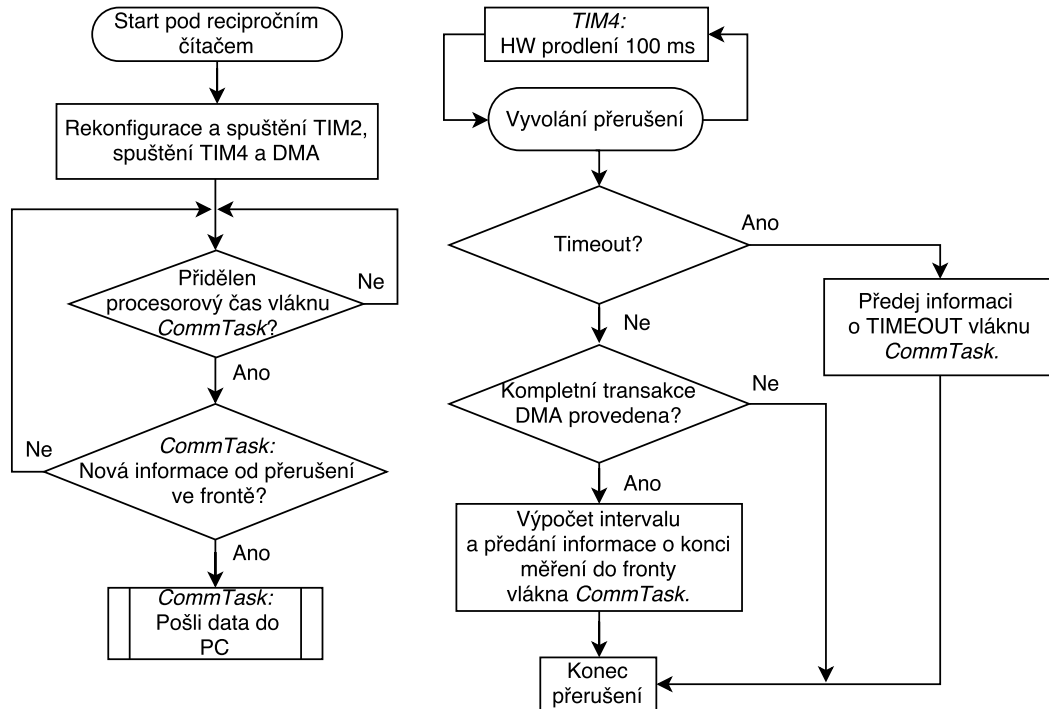
- 1.) Příchozí spouštěcí událost na kanálu 1 vyvolá reset a spuštění čítače TIM2. Zároveň je vyvoláno (2.)
- 2.) DMA přenáší předdefinovaná data do *CCER* registru, čímž je povoleno vzorkování na kanálu 2.
- 3.) Příchozí událost na kanálu 2 vyvolá (4.) zachycení hodnoty čítače TIM2 do *C/C* registru 2.
- 4.) Zachycení hodnoty čítače do *C/C* reg. 2 vyvolá (5.) DMA přenos dat z *C/C* reg. 2 do paměti.
- 5.) Časovač TIM4 periodicky vyvolává ISR, ve které je zajištěn přenos dat do PC.



Obr. 4.24: Blokové schéma obvodového řešení čítače měřícího časový interval v *Independent* módu

4.6.2 Programová obsluha měření časového intervalu

Návrh a řešení softwarové obsluhy, nepočítáme-li správnou konfiguraci registrů, je velmi jednoduchý a je popsán následujícím vývojovým diagramem.



Obr. 4.25: Vývojový diagram programové obsluhy měření časového intervalu

Výpočet časového intervalu je proveden ve vyvolané obslužné rutině jako

$$T_i = \frac{n_{clk}}{TIM2CLK} [s], \quad (4.25)$$

kde n_{clk} je počet načítaných cyklů vnitřních hodin periferie o frekvenci $TIM2CLK$.

4.6.3 Chyba měření časového intervalu

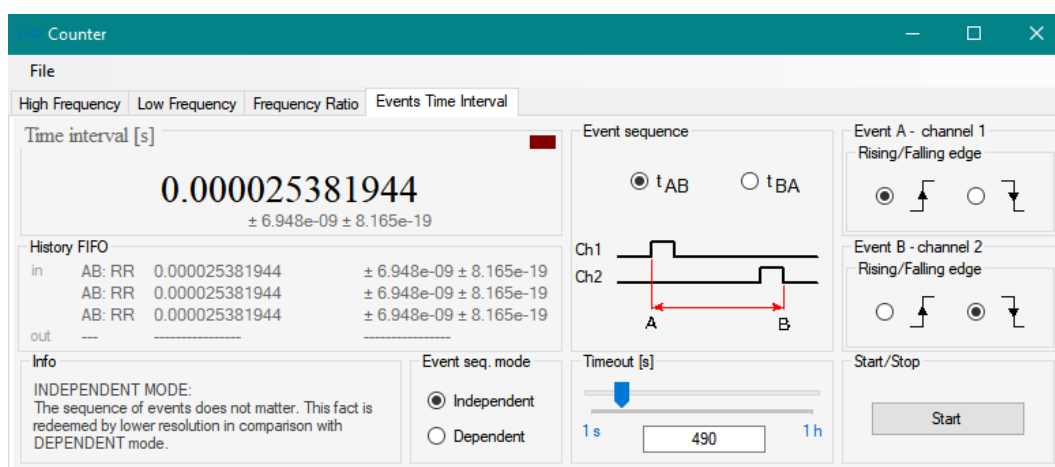
V metodě měření časového intervalu mezi dvěma událostmi se projevuje chyba kvantizace a chyba časové základny. Výpočet je proveden stejným způsobem jako v případě recipročního čítače a stejně tak byl otestován se stejnými přístroji jako přímá a reciproční metoda měření. Následující tabulka 4.5 zobrazuje výsledky měření intervalu mezi vzestupnou a sestupnou hranou pulsu generovaného přístrojem Agilent 33250A.

		Chyba měření		Přesnost měření	
PW_{gen} [s]	PW_{meas} [s]	ΔT_{im} [s]	δT_{im} [%]	$\pm \Delta T_{iq} \pm \Delta T_{itb}$ [s]	δT_i [%]
50n	41,67n	8,33e-09	16,60	$\pm 10,4e-09 \pm 2,08e-12$	20,80
100n	90,28n	9,72e-09	9,70	$\pm 8,21e-09 \pm 4,51e-12$	8,20
500n	486,11n	1,39e-08	2,80	$\pm 7,15e-09 \pm 2,43e-11$	1,40
1u	0,99u	1,00e-08	1,00	$\pm 7,04e-09 \pm 4,95e-11$	0,70
10u	9,98u	2,00e-08	0,20	$\pm 6,95e-09 \pm 5,00e-10$	0,07
100u	99,990u	1,00e-08	1,00e-02	$\pm 6,95e-09 \pm 5,00e-09$	8,56e-03
1m	999,9653u	3,47e-08	3,47e-03	$\pm 6,96e-09 \pm 5,00e-08$	5,05e-03
10m	9,99975m	2,50e-07	2,50e-03	$\pm 6,94e-09 \pm 5,00e-07$	5,00e-03
100m	99,997625m	2,38e-06	2,36e-03	$\pm 6,94e-09 \pm 5,00e-06$	5,00e-03
1	0,9999765	2,35e-05	2,35e-03	$\pm 6,94e-09 \pm 5,00e-05$	5,00e-03
10	9,99976808	2,32e-04	2,32e-03	$\pm 6,94e-09 \pm 5,03e-04$	5,03e-03

Tab. 4.5: Naměřené hodnoty časového intervalu mezi dvěma událostmi

4.6.4 Grafické rozhraní pro měření časového intervalu

Poslední záložkou univerzálního čítače je záložka *Events Time Interval*, pod kterou lze najít čítač pro měření časového intervalu mezi dvěma vybranými událostmi. Za pomoci grafického rozhraní je možno volit hrany, na které budou vstupy reagovat. Událost na kanálu 1 je označena jako A, na kanálu 2 pak jako písmeno B. Jejich pořadí lze prohazovat v bloku nazvaném *Event sequence*. Maximální doba čekání na příchozí hrany je stanovena blokem *Timeout*, ve kterém lze volit čas od 1 sekundy až po 1 hodinu. V levé dolní části se nachází přepínač pro volbu módu měření časového intervalu, který byl popsán v sekci *Obvodové řešení měření časového intervalu* 4.6.1.



Obr. 4.26: Uživatelské rozhraní metody měření časového intervalu

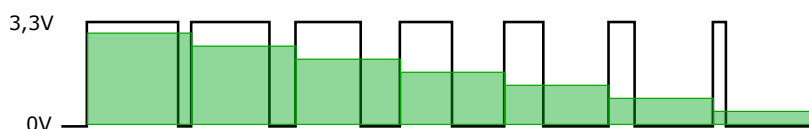
5 GENERÁTOR PWM SIGNÁLU S ČASOVĚ PROMĚNNOU STŘÍDOU

Součástí diplomové práce je generátor PWM signálu s časově proměnnou střídou, který byl vyvinut na základně již hotového funkčního generátoru. Generátory se v některých aspektech funkce shodují, proto byl PWM generátor do funkčního generátoru integrován. Nejedná se tedy o samostatný modul, který by zbytečně zvyšoval velikost PC aplikace a firmwaru. Mimoto oba generátory využívají obou dvou DMA periferií¹, které jsou oběma generátory velmi vytíženy a nešlo by je proto ani zároveň spouštět.

Úkolem navrženého a realizovaného PWM generátoru je měnit střihu výstupního PWM signálu předem definovaným průběhem (předpis), který je nahrán pomocí .CSV souboru skrze grafické rozhraní. Soubor obsahuje pouze data reprezentující velikost napětí. V aplikaci lze následně měnit parametry signálu jako je frekvence, amplituda, fáze nebo offset a v případě obdélníkového signálu střída.

5.1 Princip funkce generátoru PWM s časově proměnnou střídou

Funkční generátor na rozdíl od PWM generátoru využívá DAC, pomocí něhož lze v podstatě generovat přímo nahraná data, která představují velikost napětí. Pro PWM generátor však nahraná data představují střední hodnotu periodického signálu PWM, a proto je třeba je přepočítat na velikost střidy, která tuto střední hodnotu udává. Příklad střední hodnoty PWM signálu v závislosti na jeho střídě je znázorněn obrázkem 5.1.



Obr. 5.1: Závislost střední hodnoty signálu na střídě PWM

¹Mikrokontrolér STM32F303RE poskytuje pouze dvě DMA periferie.

Obecně lze přepočet střední hodnoty signálu na střidu psát jako

$$PWM_{dc} = \frac{PWM_{mean}}{U_{log1}} \cdot 100 [\%], \quad (5.1)$$

kde PWM_{mean} je střední hodnota signálu a U_{log1} je hodnota napětí v úrovni logické 1, tedy v úrovni 3,3 V.

Generátor PWM, stejně jako univerzální čítač, využívá periferie integrované čítací jednotky, jejíž obvody jsou navíc uzpůsobeny ke generování *Output Compare* pulsů i PWM signálů s možností jejich vyvedení až na 4 kanály. Generátor využívá pro 2 kanály zvlášť dvou čítacích jednotek.

Princip funkce generování PWM signálu čítací jednotkou lze popsat jednoduše za pomoci tří registrů; *Counter (CNT)*, *Auto-reload* a *Capture/Compare* registru. *Counter* registr představuje samotný čítací element, který je plněn vnitřními hodinami přivedenými z APB (Advanced Peripheral Bus) nebo PLL (Phase Locked Loop). Velikost *Auto-reload* registru stanoví, po jakou hodnotu má být *Counter* registr naplněn, a tím i udává hodnotu frekvence PWM signálu. Po události naplnění je čítací registr automaticky vynulován a spuštěn znovu. V *Capture/Compare* registru je uložena hodnota menší než v registru *Auto-reload*, a dosáhne-li čítač této hodnoty, výstupní logická úroveň je změněna. S resetem čítače je pak logická úroveň přepnuta do původního stavu. Je třeba dávat pozor, aby *Capture/Compare* registr obsahoval menší hodnotu než registr *Auto-reload*, jinak nedojde k přepnutí logické úrovně.

Teď již známe význam registrů čítače, a tak zde může být zmíněn konkrétní výpočet pro realizaci střídání signálu. Z předchozího odstavce je patrné, že nakonfigurovaná hodnota do *Auto-reload* registru představuje maximální velikost střídání $PWM_{dc} = 100 \%$ a proto lze požadované střídání dosáhnout nastavením registrů dle poměru $\frac{Capture/Compare\ reg.}{Auto-reload\ reg.} \cdot U_{log1}$. Změnou *Capture/Compare* registru lze modulovat střídání PWM, proto jsou nahraná data U_i z .CSV souboru přepočítána jako

$$C/C\ data_i \doteq \frac{U_i}{3,3} \cdot \text{hodnota } Auto-reload\ reg. \quad (5.2)$$

a následně poslána do mikrokontroléru a uložena do paměti RAM. Vypočtená hodnota je zaokrouhlena na nejbližší celé číslo.

Jsou-li vzorky $C/C\ data_i$ uloženy v paměti, je třeba nastavit periferii DMA pro jejich přenos do registru *Capture/Compare*. Frekvence spouštění DMA, s jakou jsou vzorky přenášeny z paměti do registru, je určena konfigurací časovače TIM6 nebo

TIM7 pro výstupní kanál 1, resp. 2. Tato frekvence je dána počtem vzorků v paměti a nastavenou frekvencí požadovaného předpisu v grafickém rozhraní. Například, je-li v paměti uloženo 1000 vzorků na periodu předpisu a jeho frekvence činí 200 Hz, pak frekvence spouštění DMA musí být 1000x větší a činit tak 200 kHz.

5.2 Zkreslení předpisu střední hodnoty PWM

Generátor PWM využívá některých implementací a výpočtů funkčního generátoru, jako je způsob určení počtu vzorků a časování přenosu vzorků pomocí čítacích periférií. Proto jsou některé parametry omezeny možnostmi periferie DAC, na které je funkční generátor založen.

Maximální rychlost aktualizace výstupního napětí DAC je v datasheetu [1] stanovena na 2 MS/s a funkční generátor zároveň alokuje paměť maximálně pro 1000 vzorků. Proto je nastavením frekvence předpisu na 1 MHz omezen počet vzorků na 2. Počtu 20 vzorků odpovídá 100 kHz. Optimálním řešením pro dosažení maximálního počtu vzorků s co nejvyšší frekvencí předpisu jsou 2 kHz. Je třeba poznamenat, že v případě zapnutí obou dvou kanálů generátoru přísluší každému z nich polovina vzorků maxima.

Nahráný předpis nemusí být v každém případě výstupním signálem reálně splněn, jelikož je ovlivněn konfigurací parametrů, jako je jeho frekvence a frekvence PWM. Problematiku zkreslení předpisu střední hodnoty PWM signálu lze rozdělit do tří bodů, které jsou zde blíže rozebrány.

- 1) Konfigurací vysoké frekvence PWM, tedy zapsanou nízkou hodnotou do 16-bitového *Auto-reload* registru² je omezeno rozlišení předpisu.
- 2) Konfigurací nízké frekvence PWM, tedy zapsanou vysokou hodnotou do 16-bitového *Auto-reload* registru je sice dosaženo vysokého rozlišení předpisu, avšak přesto dochází k jeho zkreslení.
- 3) Konfigurací vysoké frekvence předpisu je snížen počet vzorků na jeho periodu, čímž dochází taktéž k jeho zkreslení.

Zabývejme se nejprve bodem (1), který stanoví, že nízká hodnota *Auto-reload* registru ovlivňuje rozlišení signálu. Tento fakt lze podložit zmíněným vzorcem 5.2 pro přepočítání velikostí napětí na hodnoty zapisované do *Capture/Compare* registru.

²Zapsaná hodnota do *Auto-reload* registru představuje rozlišení PWM signálu.

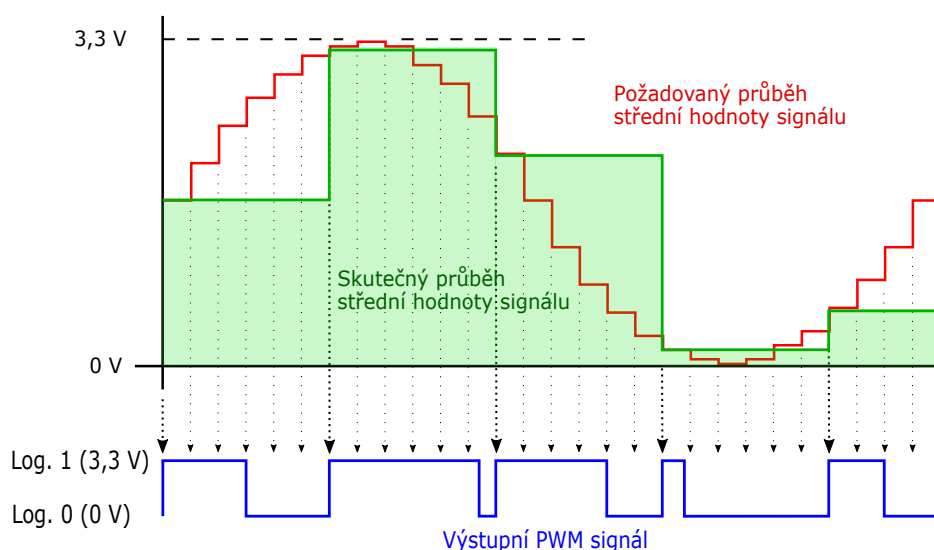
Oba registry *ARR* i *CCR* musí obsahovat celé číslo, a proto je výsledná hodnota zaokrouhlena na nejbližší celé číslo. Je-li například PWM nastavena na frekvenci 9 MHz, je rozlišení generovaného PWM signálu 4 bity, a tak je celý napěťový rozsah poměrově rozdělen pouze do 16 hodnot (*ARR* obsahuje hodnotu 16). Výpočet rozlišení PWM signálu je proveden jako

$$PWM_{res} = \frac{144 \cdot 10^6}{9 \cdot 10^6} = 16 \text{ bits}, \quad (5.3)$$

kde 144 MHz je frekvence vnitřních hodin čítače TIM1 použitého pro generování PWM na prvním kanále. Čítací periferie TIM3 pak generuje na kanále druhém a její vnitřní hodiny činí 72 MHz.

Je třeba upozornit, že pro nižší frekvence se do výpočtů musí zahrnout předdělička *PSC* (*Prescaler* registr) integrovaná před čítacím elementem - *Counter* registrem, aby bylo možno požadované nízké frekvence dosáhnout. Pak je ale nutné velikost *Auto-reload* registru podělit hodnotou této předděličky a tím je i sníženo rozlišení PWM signálu. Algoritmus pro výpočet *ARR* a *PSC* je navržen takovým způsobem, aby *Auto-reload* registr vždy obsahoval co nejvyšší možnou hodnotu.

Posuňme se k bodu (2), který stanoví, že stejně jako nízká hodnota *Auto-reload* registru ovlivňuje zkreslení předpisu i jeho hodnota vysoká, tedy požadovaná nízká frekvence PWM signálu. Takovouto konfigurací lze sice dosáhnout maximálního rozlišení PWM signálu, nicméně tím může být sníženo rozlišení předpisu. Čítač je konfi-

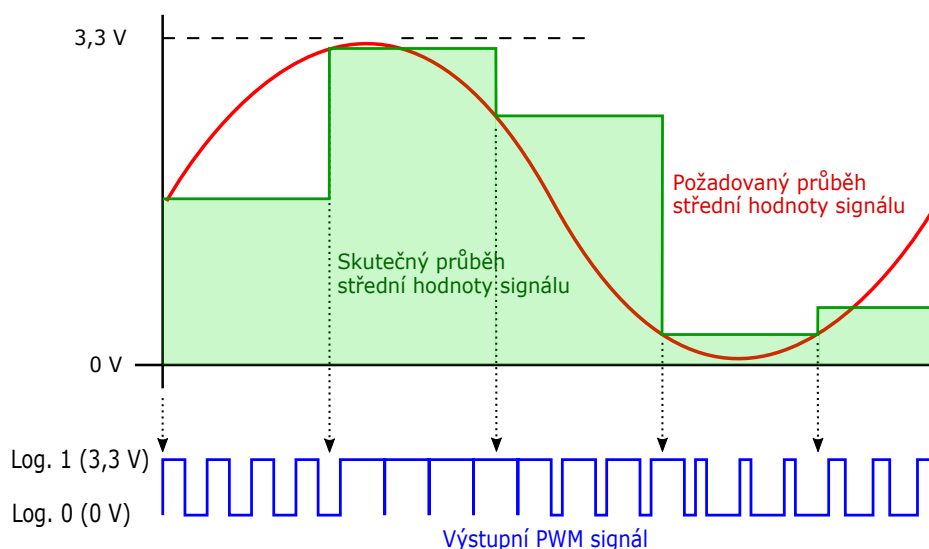


Obr. 5.2: Zkreslení předpisu konfigurací nízké frekvence PWM signálu

gurován s pomocí *Preload* funkce registru *Capture/Compare*, díky které je zapisovaná

hodnota nejdříve zapsána do stínového registru a až při zahájení další periody je touto hodnotou *Capture/Compare* registr aktualizován. Z toho vyplývá, že střída je během jedné periody neměnná, a nemůže se tedy stát, že by byla její hodnota stále přepisována. Přenášení dat pomocí DMA z paměti do *Capture/Compare* registru tak během právě probíhající periody nevyvolá žádnou změnu. Obrázkem 5.2 je tato skutečnost ilustrována.

Posledním bodem (3) je zmíněn důsledek zvýšení frekvence předpisu střední hodnoty výstupního PWM signálu. Zvýšením frekvence je omezen počet vzorků předpisu³. Může se tedy stát, že bude sice nastavena optimální frekvence PWM, ale vůči redukovanému počtu vzorků nebude mít žádný vliv. Tato skutečnost je znázorněna obrázkem 5.3, na kterém by počet vzorků 5 odpovídal frekvenci předpisu 400 kHz.



Obr. 5.3: Snížení počtu vzorků konfigurací vysoké frekvence předpisu

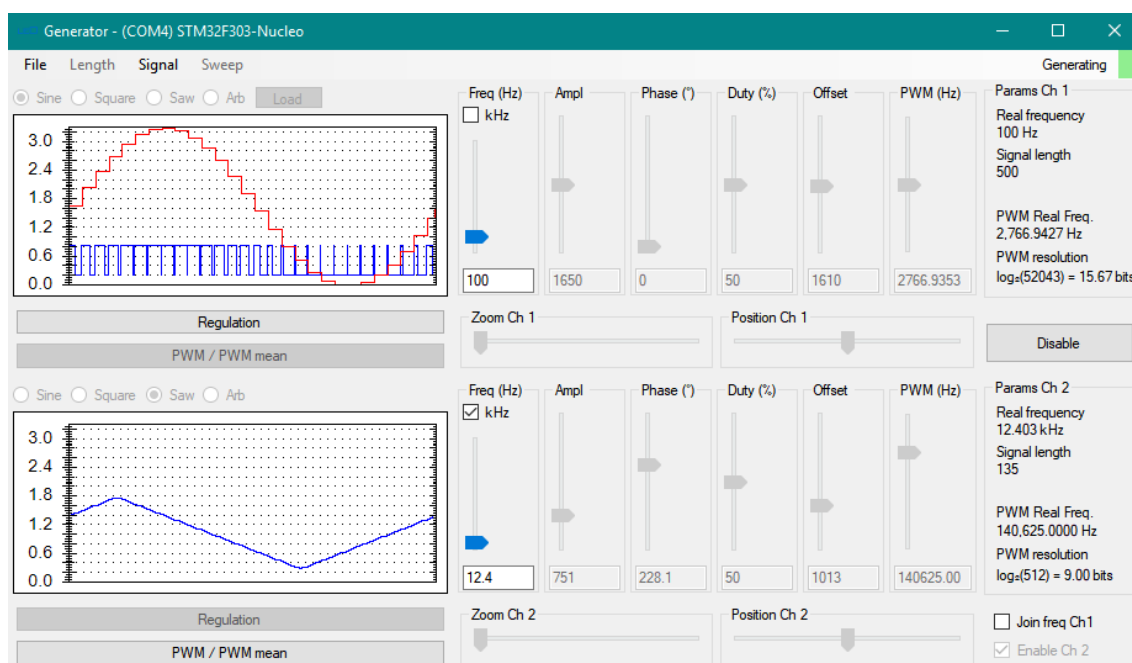
Aby měl uživatel představu o reálné podobě výstupního signálu PWM a jeho střední hodnotě, jsou oba průběhy propočítávány PC aplikací dle zadaných parametrů a vykreslovány v grafickém uživatelském rozhraní.

³Určeno dle specifikace maximální rychlosti aktualizace výstupu DAC periferie.

5.3 Grafické rozhraní generátoru PWM s časově proměnnou střídou

Grafické rozhraní bylo vyvinuto na základně funkčního generátoru. Obsahuje dva kanály, na kterých lze v základním nastavení generovat sinusový, trojúhelníkový a obdélníkový průběh či volitelnou funkci, kterou je možno nahrát pomocí .CSV souboru z MS Excel. Každému ze dvou kanálů přísluší jeden graf, jehož charakteristiky jsou přepočítávány a vykreslovány se změnou dvou vstupních parametrů; frekvence předpisu a frekvence PWM signálu. Oba parametry lze měnit za pomoci dvou jezdců, které se nacházejí na dvou koncích příslušného bloku s táhly. Charakteristiky lze v grafech libovolně přibližovat a oddalovat.

Pod grafem se nacházejí dvě tlačítka pro přepnutí zobrazení charakteristik. Prvním tlačítkem *Regulation* je zobrazen požadovaný průběh střední hodnoty dané generovaným PWM signálem, druhým tlačítkem *PWM / PWM mean* je graf přepnut do zobrazovacího režimu dvou charakteristik, přičemž první představuje reálný průběh PWM signálu v jedné periodě předpisu a druhá pak průběh střední hodnoty tohoto PWM signálu v jedné periodě předpisu. Grafy tak pro uživatele představují jistý způsob zpětné vazby pro správné nastavení parametrů požadovaného signálu.



Obr. 5.4: Uživatelské rozhraní generátoru PWM s časově proměnnou střídou

Změnu frekvence výstupního PWM signálu v rozsahu 1 Hz - 9 MHz lze provést

logaritmickým jezdcem v pravé části grafického rozhraní. Na levé straně se nachází jezdec pro změnu frekvence předpisu, kterou lze na rozdíl od frekvence PWM měnit v průběhu generování.

V některých případech konfigurace frekvence nemůže být uživateli zcela vyhověno, např. pokud je zadáno prvočíslo. Vytvořené GUI proto zobrazuje skutečné hodnoty obou frekvencí v pravé části pod nápisy *Real frequency* a *PWM Real Freq.* Pod nápisem *Signal length* je zobrazen počet vzorků nahraný do paměti MCU a pod *PWM resolution* aktuální rozlišení v bitech a čísle vypočteném jako 2^n , kde n je počet bitů.

6 ČTYŘKANÁLOVÝ GENERÁTOR SYNCHRONIZOVANÝCH SIGNÁLŮ PWM

Posledním úkolem bylo vytvořit vícekanálový generátor signálů PWM, který je stejně jako generátor PWM s předpisem založen na integrované čítací jednotce. Generátor nabízí synchronizaci signálů na čtyřech výstupních kanálech s možností konfigurace tří parametrů, kterými jsou délky pulsů (střídy), společná frekvence a jednotlivé fáze signálů.

Takovýto generátor synchronizovaných signálů PWM o čtyřech kanálech lze nejlépe realizovat za pomoci dvou nebo čtyř synchronizovaných čítacích jednotek v tzv. *Asymmetric* módu. Tím lze dosáhnout maximální možné frekvence i libovolné konfigurace frekvence jednotlivých kanálů. V tomto módu ale musí být pro realizaci jednoho výstupního kanálu využity dva *Capture/Compare* registry, jeden udávající čas vzestupné hrany a druhý hrany sestupné. Čítač musí být zároveň nastaven jako *Center-aligned*¹, přičemž během inkrementace je *CNT* registr srovnáván s prvním *C/C* registrem a při dekrementaci s druhým *C/C* registrem. Z toho vyplývá, že se pulsy v *Asymmetric* módu musí z principu funkce vždy překrývat, nebo na sebe alespoň navazovat a to je v tomto návrhu nežádoucí vlastností. Navíc většina čítacích jednotek, které mikrokontrolér poskytuje, je již zabrána ostatními funkcemi platformy LEO a moc jich nezůstává. Jedním z požadavků na tento generátor je možnost jeho spuštění spolu s dalšími funkcemi jako je např. osciloskop, logický analyzátor nebo univerzální čítač. Proto místo zmíněné implementace bylo navrženo jiné řešení, které využívá pouze jediné čítací jednotky.

6.1 Princip funkce generátoru synchronizovaných signálů PWM

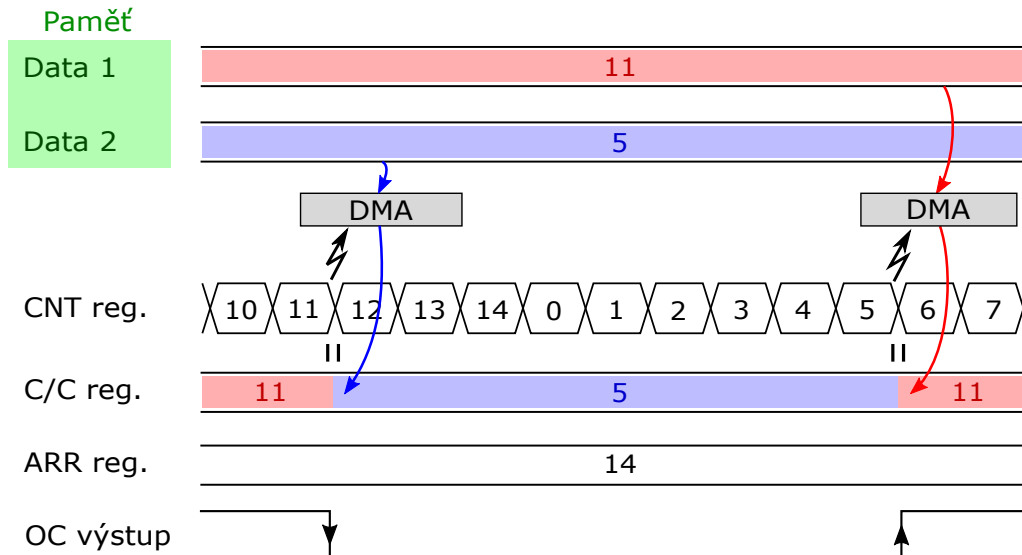
Pro generátor synchronizovaných signálů PWM byl zvolen čítač TIM8, jehož vnitřní hodiny činí 72 MHz. Čítač obsahuje 4 *Capture/Compare* registry, které jsou samostatně vyvedeny na 4 *Output Compare* výstupy. Generátor tak poskytuje 4 synchronizované kanály. Princip změny logické úrovně na jednom kanálu je založen na dosažení hodnoty *Capture/Compare* registru čítacím registrem *CNT*, jak je popsáno

¹Čítač nejprve inkrementuje velikost registru *CNT* a po dosažení nastavené hodnoty v *Auto-reload* registru dekrementuje zpět, dokud nedosáhne nuly. Dosažením nuly je vygenerována událost, od které jsou aktualizovány registry čítače a sám je znovu spuštěn.

v podkapitole *Princip funkce generátoru PWM s časově proměnnou střídou* 5.1.

Z principu funkce je patrné, že během jedné periody signálu (naplnění *CNT* registru po hodnotu nastavenou v *ARR*) lze na jednom kanále za pomoci jednoho *Capture/Compare* registru změnit logickou úroveň výstupního signálu pouze jednou. Aby bylo dosaženo dvou změn v jedné periodě, je třeba změnit hodnotu *Capture/Compare* registru v průběhu této periody. K tomuto účelu bylo využito periferie DMA, která od události změny logické úrovně přenáší předem uložená data z paměti do *Capture/Compare* registru. Za pomoci registru *Output Compare x mode (OCxM)* je výstup nakonfigurován do *Toggle* módu, čímž je logická úroveň výstupu vždy změněna do opačného stavu.

V paměti jsou uloženy dvě hodnoty zastupující náběžnou a sestupnou hranu (pro jeden kanál) a jsou seřazeny obráceně, tzn. nejprve je do *Capture/Compare* registru přenášena hodnota pro sestupnou hranu a až druhá pak pro hranu vzestupnou. Toto pořadí je určeno tím, že již před startem čítače je do *Capture/Compare* registru programově zapsána hodnota vzestupné hrany. Princip funkce je ilustrován obrázkem 6.1.



Obr. 6.1: Princip funkce generátoru synchronizovaných signálů PWM. Dvojitá změna *C/C* registru pomocí DMA během jedné periody čítače

S tímto přístupem je spojen problém omezené rychlosti přenosu vzorků pomocí DMA. Obsluhu transakcí z paměti do periferie čítače TIM8 může provádět pouze DMA2, které musí obsloužit všechny 4 kanály. Jsou-li uživatelem nastaveny extrémní

hodnoty (jako např. vysoká frekvence a pulsy na všech kanálech začínají nebo končí ve stejný okamžik a jsou velmi krátké), pak k přenosům dat nemusí vůbec dojít.

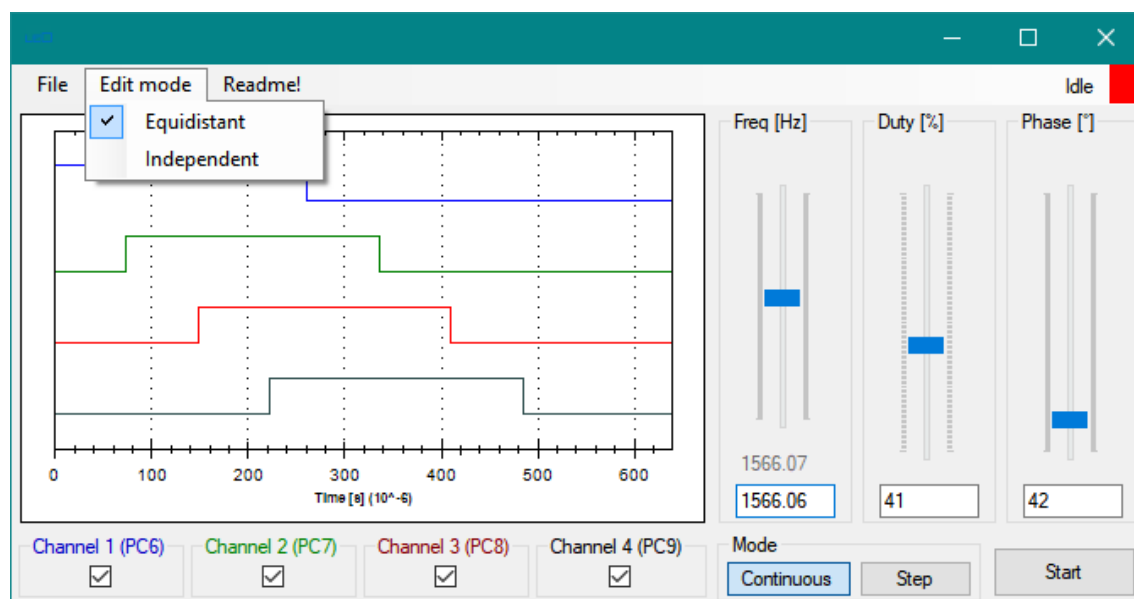
Testováním byla stanovena maximální/limitní hodnota frekvence výstupních signálů PWM, při které nedochází k výpadkům přenosů a lze se tak spolehnout, že jsou na výstupech generovány požadované signály. Limitní frekvence byla stanovena na 20 kHz, nicméně je uživateli povoleno generovat frekvence až do 1 MHz, musí si však ověřit jinými testovacími prostředky, že jsou hodnoty na výstupu správné.

6.2 Grafické rozhraní generátoru synchronizovaných signálů PWM

Funkci generátoru synchronizovaných signálů PWM lze v aplikaci LEO najít pod tlačítkem *Sync. PWM*. Funkce je rozdělena do dvou módů nazvaných jako *Equidistant* a *Independent*. Ve výchozím stavu je nastaven mód *Equidistant*, ve kterém jsou pulsy vůči sobě automaticky konfigurovány se stejnou fází a mají zároveň stejnou střídu. Fáze n -tého signálu je dána jako

$$phase_n = phase_2 \cdot (n - 1), \quad n = 1, 2, 3, 4 \quad (6.1)$$

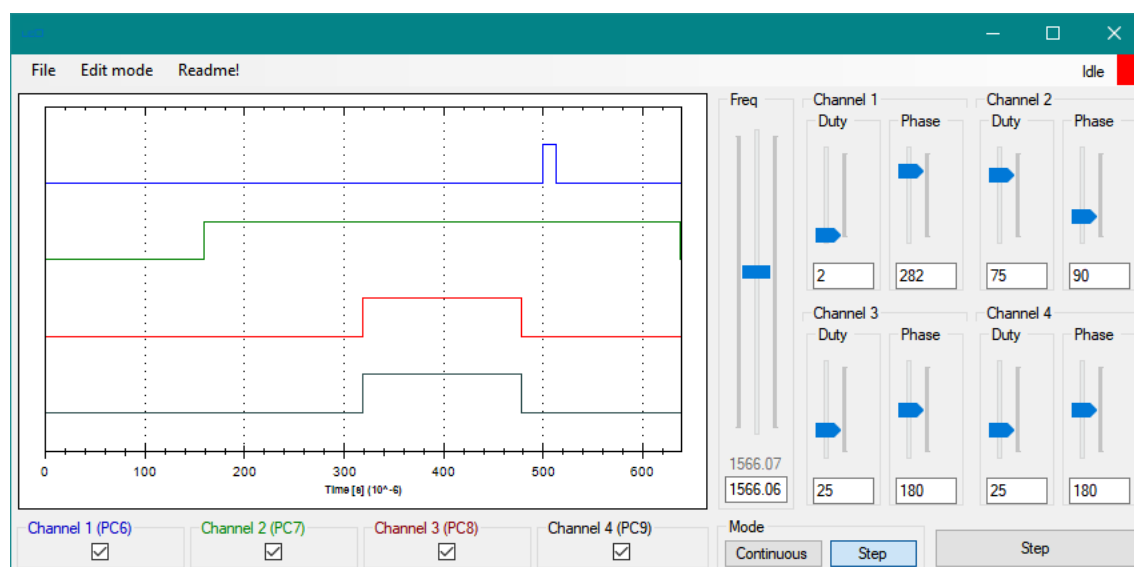
kde $phase_n$ představuje fázi signálu, resp. vzdálenost od počátku na n -tém kanálu. Dále lze v aplikaci přepínat mezi periodickým (*Continuous*) a krokovým (*Step*) mó-



Obr. 6.2: Uživatelské rozhraní generátoru synch. sig. PWM v *Equidistant* módu

dem generování. Jednotlivé kanály je možno zapínat a vypínat zaškrtnutými tlačítky ve spodní části grafického rozhraní.

Independent mód lze zvolit pomocí horního menu pod záložkou *Edit mode*. S jeho pomocí mohou být s výjimkou frekvence generovány zcela nezávislé obdélníkové signály na jednotlivých kanálech. Nezávisle na sobě lze konfigurovat vypnutí/zapnutí kanálu, střihu a fázi. Módy *Step* a *Continuous* pro možnost krokování či neustálého generování jsou taktéž k dispozici.



Obr. 6.3: Uživatelské rozhraní generátoru synchronizovaných signálů PWM v *Independent* módu

Uživatel je informován skrze *Readme!* hlavního menu o omezeních daných přenosem vzorků za pomoci DMA.

7 LOGICKÝ ANALYZÁTOR

Poslední dosud navrženou a realizovanou funkcí platformy LEO, přestože není v zadání, je osmikanálový logický analyzátor. Jeho realizace je omezena velikostí zbývající paměti RAM mikrokontroléru, které je již málo, a tak byl pro naměřené vzorky využit buffer již alokovaný pro osciloskop.

7.1 Princip funkce logického analyzátoru

Z počátku návrhu logického analyzátoru bylo třeba stanovit způsob vzorkování vstupních signálů. Některé mikrokontroléry integrují rychlé obvody rozhraní DCMI určené pro digitální kamery. Použitým mikrokontrolérem STM32F303RE však DCMI není podporováno, a tak jsou změny signálů vzorkovány přímo z GPIO brány.

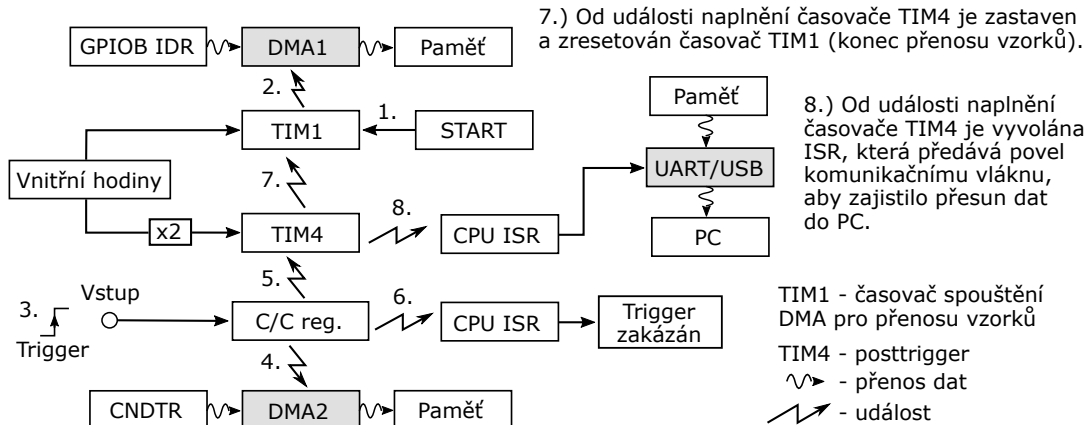
Posledními nevyužitými I/O vstupy mikrokontroléru bylo 8 pinů PB6 až PB13 na bráně GPIOB. Na tyto vstupy může být přivedeno 8 nezávislých signálů. Rychlost vzorkování, resp. přenášení dat ze vstupního registru IDR brány GPIOB do paměti, je určena frekvencí spouštění DMA, kterou udává časovač TIM1. Periferie DMA zde musí přenášet data z celé 16-bitové brány, jelikož jsou signály přiváděny od pinu PB6, a nikoliv od počátku registru od pinu PB1¹. Protože je pro vzorky využita již alokovaná paměť pro osciloskop o velikosti 40 kB, maximální počet přenesených vzorků logickým analyzátozem činí 20 k.

Trigger lze nastavit zvlášť na dvou kanálech - pinech PB6 a PB7, aby reagovaly na vzestupnou či sestupnou hranu. Vybraný kanál, který plní funkci triggeru, je nakonfigurován jako *Input Capture* časovače TIM4 (TI1 - PB6 nebo TI2 - PB7). Časovač TIM4 je zde nastaven v *One Pulse* a slave *Combined reset + trigger* módu vstupu TI1 nebo TI2, tzn. přijde-li na daný vstup požadovaná hrana, časovač je spuštěn a čítá cykly vnitřních hodin, dokud nedosáhne hodnoty nastavené v *Auto-reload* registru. Od události naplnění *CNT* registru časovače TIM4 je zresetován časovač TIM1, který periodicky spouští přenos vzorků do paměti za pomoci DMA, a tím je ukončen jejich odběr. Jinými slovy, TIM4 zde představuje funkci posttriggeru. Zároveň s resetem časovače TIM1 je vyvolána obslužná rutina přerušení, ze které je předán příkaz komunikačnímu vláknu, aby byla data poslána do PC.

Pretrigger je řešen aktivním čekáním za pomoci funkce *vTaskDelay()*. V závislosti na nastaveném módu triggeru *Auto* nebo *Normal* je časovač TIM4 (posttrigger)

¹Periferie DMA si drží adresu registru.

- 1.) Softwarový start časovače TIM1.
- 2.) Časovač TIM1 zajišťuje periodické spouštění DMA pro přenos vzorků z IDR registru do paměti.
- 3.) Příchozí hrana na triggerovacím kanále způsobí (4.) přenos hodnoty CNDTR registru DMA1 do paměti. Zároveň je spuštěn časovač TIM4 představující posttrigger (5.). Nakonec je vyvoláno přerušení (6.), ve kterém je triggerování zakázáno.



Obr. 7.1: Blokové schéma obvodového řešení logického analyzátoru

po uplynulé době aktivního čekání softwarově povolen, nebo není. V *Auto* triggeru je TIM4 softwarově spuštěn po uplynutí doby pretriggeru, proto nepřijde-li hrana na spouštěcí kanál, navzorkována data jsou i tak poslána do PC po uplynutí doby posttriggeru (TIM4). Přijde-li hrana kdykoli během měření a TIM4 je již softwarově spuštěn, je TIM4 díky jeho nastavení v *Combined reset + trigger* resetován a znovu spuštěn. Jsou tak vždy navzorkovaná všechna požadovaná data. Během měření se může ale stát, že přijde více hran na triggerovacím kanále. Z tohoto důvodu je od triggeru vyvolána ISR, ve které je trigger příslušného kanálu zakázán. V módu *Normal* triggeru není posttrigger po startu softwarově spuštěn, a proto nepřijde-li hrana na daném kanále, žádná data do PC nejsou poslána a vzorkování tak probíhá dále.

Periferie DMA1 je nastavena pro plnění kruhového bufferu. Aby bylo možno určit, v jakém čase došlo k triggeru na daném vstupním kanále, je zde využito DMA2, které od události triggeru přenáší do paměti velikost *CNDTR* registru periferie DMA1. Protože je *CNDTR* registr dekrementován s každým přeneseným vzorkem a po dosažení nuly je znovu naplněn, je tak znám *n*-tý vzorek představující čas příchodu triggeru. Tento vzorek je poslán spolu s daty do PC, kde je provedena rotace bufferu dle hodnoty tohoto vzorku.

PC aplikací přijatá data jsou v podobě 16-bitového GPIOB registru, a je proto zapotřebí je rozparsovat na jednotlivé bity. Nejdříve je ale proveden proces rotace pole dle uživatelem nastaveného triggeru a dle ukazatele na *n*-tý vzorek, který udává

čas příchodu triggeru. Protože je *CNDTR* registr dekrementován, je ukazatel na *n*-tý vzorek dán rovnicí

$$p_n = n - p_x, \quad (7.1)$$

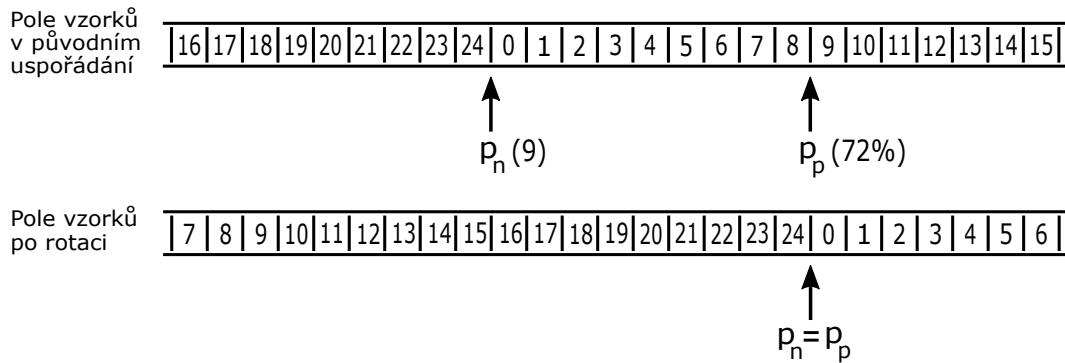
kde p_n je skutečná hodnota ukazatele na *n*-tý vzorek dat a p_x je přijatý ukazatel. Ze znalosti skutečné hodnoty ukazatele, uživatelem zadané hodnoty triggeru a počtu navzorkovaných dat je vypočítáno, o kolik je třeba pole rotovat. Je-li $\frac{p_n}{n}$ větší než uživatelem požadovaná hodnota pretriggeru p_p [%], pak platí

$$n_{left} = p_n - \frac{p_p}{100} \cdot n, \quad (7.2)$$

a tak je třeba provést rotaci pole směrem doleva n_{left} krát. Je-li naopak poměr $\frac{p_n}{n}$ menší než uživatelem požadovaná hodnota triggeru p_p [%], pak platí

$$n_{right} = \frac{p_p}{100} \cdot n - p_n. \quad (7.3)$$

To znamená, že je třeba pole rotovat doprava o n_{right} pozic. Pro ilustraci je uveden obrázek 7.2, kde nula značí počátek posttriggeru, resp. čas příchodu spouštěcí hrany.



Obr. 7.2: Rotace pole do požadovaného uspořádání

Z obrázku je zřejmé, že $\frac{p_n}{n}$ je menší než p_p , a tak platí rovnice 7.3. Její aplikací dostáváme

$$n_{right} = \frac{p_p}{100} \cdot n - p_n = \frac{72}{100} \cdot 25 - 9 = 9. \quad (7.4)$$

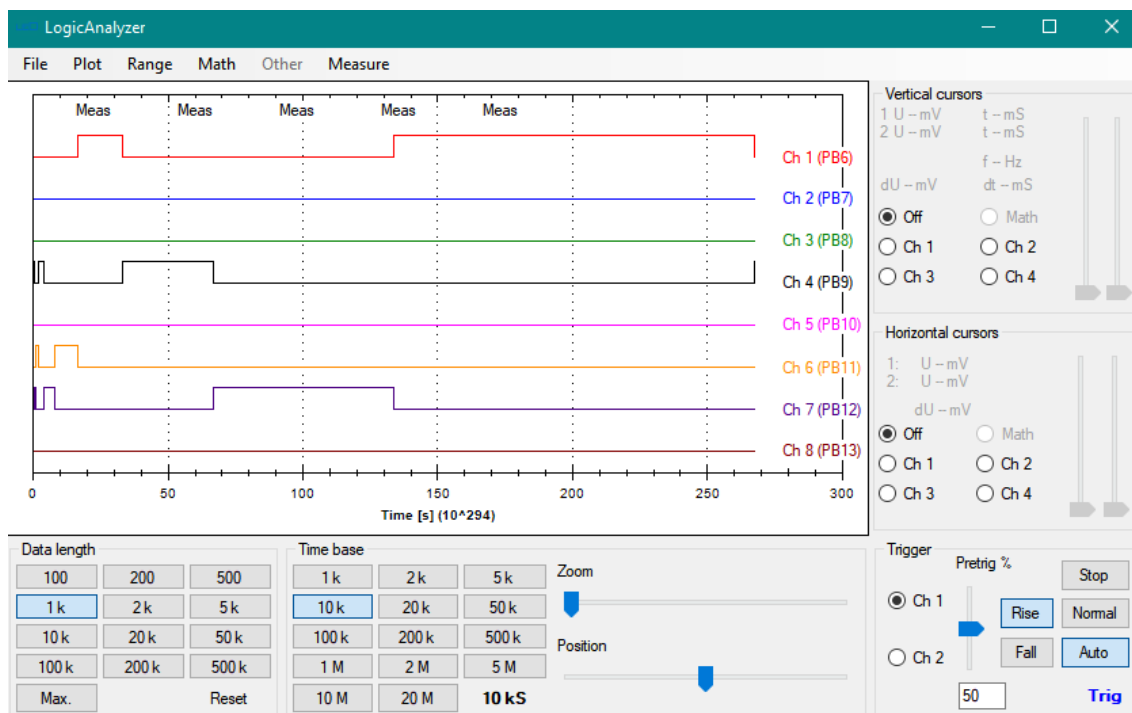
Takové pole je třeba rotovat o 9 míst doprava. Hodnoty pole jsou následně napsány na jednotlivé bity brány GPIOB a zobrazeny v grafu.

7.2 Grafické rozhraní logického analyzátoru

Grafické uživatelské rozhraní logického analyzátoru napodobuje grafické rozhraní osciloskopu a nabízí v podstatě i stejné funkce. Uživateli se tak stačí naučit s jednou

ze dvou funkcionalit a dokáže manipulovat i s druhou.

V tuto chvíli lze konfigurovat rychlost vzorkování, počet vzorků (omezen na 20000), trigger na kanálu 1 (PB6) nebo 2 (PB7), pretrigger, citlivost triggeru na vzestupnou či sestupnou hranu a závěrem mód triggeru. Implementován je *Auto* a *Normal* trigger, a po zastavení vzorkování tlačítkem *Stop* je možno využít *Single* trigger.



Obr. 7.3: Grafické uživatelské rozhraní logického analyzátoru

Je třeba poznamenat, že logický analyzátor je v rozpracovaném stavu a není zatím zcela funkční. Například si lze všimnout neodpovídající časové osy. V budoucnosti bude nutno dodělat přibližování a posuv grafu, stejně jako doprogramovat vertikální a horizontální kurzory.

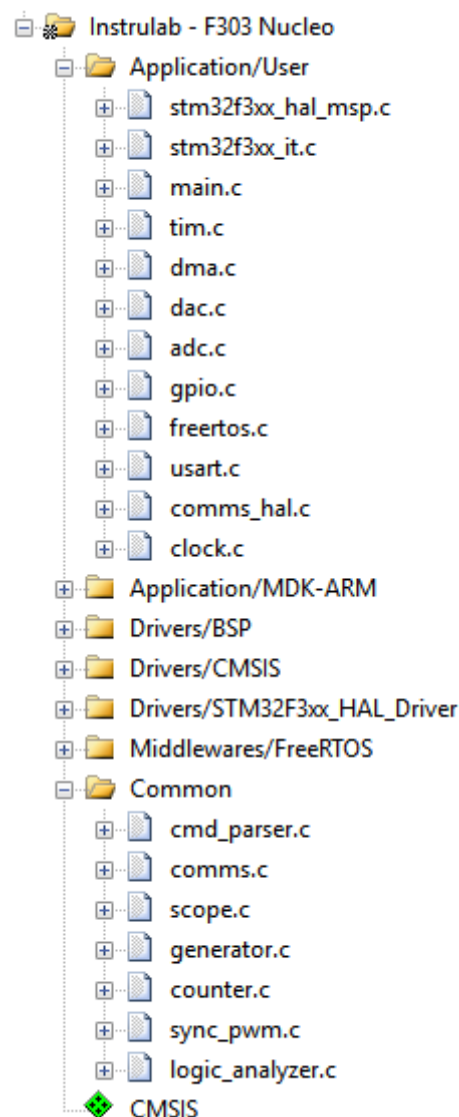
8 KONEČNÁ PODOBA PLATFORMY LEO

Tato kapitola se stručně zabývá strukturou *Instrulab* projektu (.C) pro mikroprocesor a popisuje grafické uživatelské rozhraní aplikace LEO. Znázornění uspořádání funkcí a mapování jimi využitých periférií na piny mikrokontroléru je uvedeno v příloze A. Vyvedení funkcí na konektory vývojového kitu je zobrazeno v příloze B.

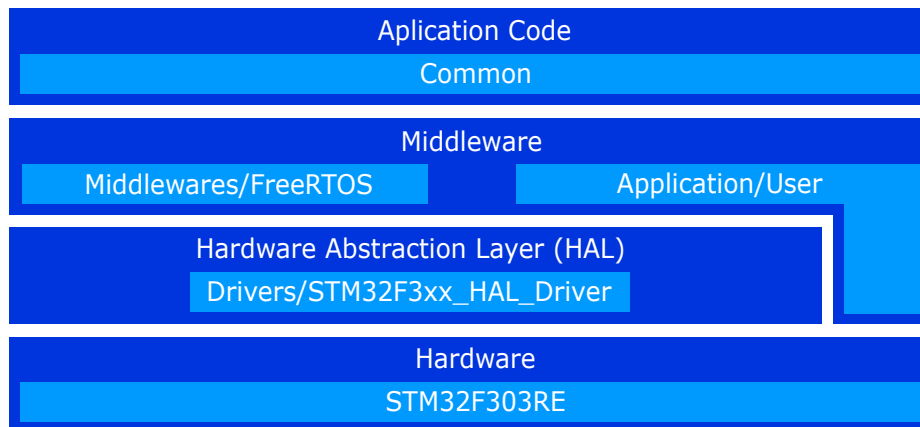
8.1 Struktura .C projektu Instrulab

Projekt obsahuje několik složek, ze kterých jsou nejpodstatnější *Application/User* a *Common*, ve kterých jsou vytvořené .C soubory. Ostatní složky obsahují importované knihovny a soubory důležité pro správnou funkci programu. Knihovna ovladačů HAL je umístěna ve složce *STM32F3xx_HAL_Driver* a soubory operačního systému FreeRTOS jsou obsaženy složkou *FreeRTOS*. Ve zbývajících složkách lze nalézt například soubor spuštění systému *startup_stm32f303xe.s* nebo soubor nastavení hodin systému *system_stm32f3xx.c*.

Aplikační vrstva systému je tvořena soubory obsaženými ve složce *Common*, přičemž všechny obsahují své vlastní vlákno, které je při startu systému vytvořeno v hlavním programu *main.c*. Tato vlákna jsou volána plánovačem operačního systému a obstarávají správnou funkci všech přístrojových funkcionalit. Soubor *cmd_parser.c* zajišťuje parsování a interpretaci významu přijatých zpráv od hosta, přičemž definice a registrace příkazů je provedena v hlavičkovém souboru *commands.h*. Program v souboru *comms.c* zajišťuje odesílání zpráv a naměřených dat do PC. Dále jsou zastoupeny samotné přístrojové funkcionality; tzn. osciloskop a voltmetr - *scope.c*, funkční generátor, generátor signálů PWM s předpisem a generátor stejnosměrného napětí - *generator.c*, univerzální čítač - *counter.c*, generátor synchronizovaných signálů PWM - *sync_pwm.c* a závěrem logický analyzátor - *logic_analyzer.c*.

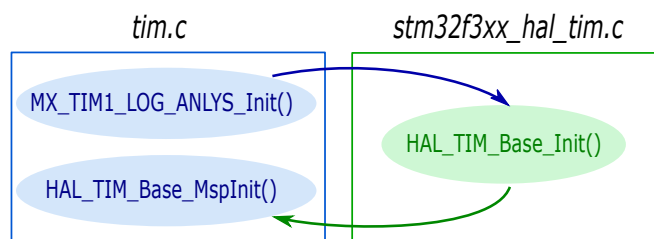


Obr. 8.1: Struktura projektu



Obr. 8.2: Architektura vytvořeného softwaru

Middleware je tvořen složkou *Application/User*. V této práci vytvořené funkce se nacházejí převážně v souboru `tim.c`, jelikož využívají vlastností periférií čítačů/časovačů TIMx. Je možno zde najít definice funkcí, které využívají knihoven HAL, ale i přímého přístupu do registrů, který zjednodušuje a zrychluje průběh programu. Naprostá většina funkcí souboru `tim.c` je volána ze souborů obsažených složkou *Common*, tedy z aplikační vrstvy. Jsou zde ale dvě funkce `HAL_TIM_Base_MspInit()` a `HAL_TIM_Base_MspDeInit()` volané univerzálními inicializačními funkcemi ovladače `stm32f3xx_hal_tim.c` periférie TIMx knihovny HAL. Je-li tedy provedena inicializace čítače/časovače v souboru `tim.c` některou univerzální funkcí ovladače, je následně volána funkce `HAL_TIM_Base_MspInit()` obsažená tímto souborem (viz. obrázek 8.3).



Obr. 8.3: Volání funkce `HAL_TIM_Base_MspInit()` z funkce `HAL_TIM_Base_Init()` ovladače periférie čítače/časovače

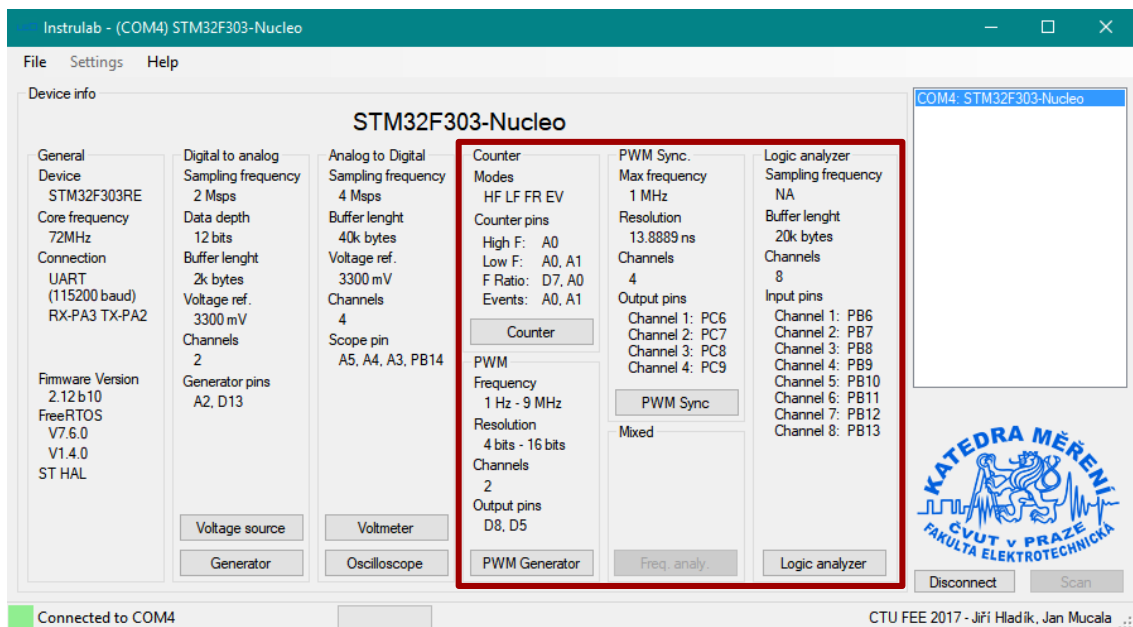
Dalšími dvěma důležitými soubory jsou `stm32f3xx_it.c`, kde jsou definovány obslužné rutiny přerušování, a `clock.c`, který inicializuje vnitřní hodiny jednotlivých

periferií.

Konfigurace operačního systému FreeRTOS je definována hlavičkovým souborem `FreeRTOSConfig.h`. Dále pak každý mikrokontrolér, na který je systém portován, musí mít vytvořený svůj vlastní hlavičkový soubor `mcu_config.c` s informacemi o konfiguraci systému. Soubor definuje využití pinů pro jednotlivé měřicí funkce a generátory, maximální počet vzorků či rychlost vzorkování, vnitřní hodiny periferií, nabízené funkcionality apod. PC aplikace tyto informace zobrazuje uživateli, nebo dle nich přepočítává přijatá data na požadovanou veličinu.

8.2 Výsledné GUI aplikace LEO

Printscreen aktuální podoby grafického rozhraní je zobrazen obrázkem 8.4. V pravé dolní části se nacházejí dvě tlačítka; jedno pro skenování VCP portů, druhé pro připojení k vývojovému kitu. Část zvýrazněná obdélníkem představuje nově vzniklé funkce výukové platformy LEO. Každému spouštěcímu tlačítku přísluší blok, ve kterém jsou zobrazeny parametry dané implementované přístrojové funkce jako např. rychlost vzorkování, počet vzorků, rozlišení, maximální rozsah, počet kanálů a jednotlivé využití pinů nebo případně konektory vývojového kitu.



Obr. 8.4: Finální grafické uživatelské rozhraní výukové platformy LEO

V levé části GUI je jako v původní verzi aplikace zobrazeno připojené zařízení, rychlost systémových hodin a datového přenosu USART. Taktéž jsou zobrazeny

verze použitých knihoven HAL a FreeRTOS. Výslednou podobu aplikace je možno srovnat s obrázkem 2.3 její původní verze.

9 ZÁVĚR

Hlavním cílem práce bylo navrhnout a realizovat univerzální čítač nabízející standardní funkce přístrojových univerzálních čítačů a začlenit ho do výukové platformy LEO vyvinuté pro mikrokontrolér STM32F303RE. Vytvořeny byly dva typy čítače pro měření frekvence, založené na přímé a reciproční metodě měření.

Čítač založený na přímé metodě je schopen měřit s rozlišením až 0,1 Hz při nastavení maximální možné doby otevření hradla 10 sekund. Díky možnosti průměrování ze 2 až 200 vzorků může být dále tato hodnota snižována. Z principu funkce je ale vhodnější metodu přímého měření aplikovat na vyšší frekvence v rozsahu od desítek kHz až po stovky MHz. Doposud byl tento čítač testován pouze do 90 MHz a jeho funkčnost při měření vyšších hodnot zatím není známa. Pro měření nižších frekvencí je v této práci vytvořen čítač založený na reciproční metodě čítání.

Reciproční čítač lze efektivně využít na měření nízkých frekvencí. Minimální měřitelná frekvence 33,528 mHz je určena velikostí čítacího registru čítací periferie použitého mikrokontroléru a maximální měřitelná frekvence 18 MHz je dána rychlostí přenosu vzorků pomocí periferie DMA. Nicméně horní limit byl omezen na 10 MHz, jelikož od spočtené meze 37942,3 Hz je již stejně výhodnější přepnout na čítač přímého měření s dobou otevření vstupní brány 100 ms. Chybu kvantizace je zde možno snížit průměrováním až z 800 vzorků.

V návaznosti na reciproční čítač byla vytvořena funkce měření střídy a délky pulsu, která je založena na stejném principu měření. Výhodou návrhu měření střídy a délky pulsu je, že po uživateli není vyžadována manipulace s již připojenými vodiči, resp. nemusí připojovat jeden vodič na dva kanály, jelikož je automaticky provedena rekonfigurace vnitřních obvodů do požadovaného stavu.

Dále byl navržen a realizován mód čítače pro měření poměru frekvencí, který má za úkol z měření zcela odstranit chybu časové základny systému. Za pomoci externího přesného zdroje hodin lze měřit vstupní frekvenci s vysokou přesností. Počet vzorků od tohoto přesného referenčního zdroje lze konfigurovat až na 4,096 miliardy, čímž lze chybu dále snižovat.

Poslední vytvořenou funkcí univerzálního čítače je měření časového intervalu mezi dvěma událostmi. Čas lze měřit mezi dvěma libovolnými událostmi od vzestupné či sestupné hrany na jakémkoli ze dvou kanálů. Čítačem je od časového intervalu 1 μ s k delším intervalům dosaženo nižší relativní chyby měření než 1 %.

Dále byly vytvořeny dva nezávislé generátory signálů PWM. První, dvoukanálový, generuje PWM signál o frekvenci 1 Hz až 90 MHz se změnou střídy dle nameraného předpisu o frekvenci 0.1 Hz až 1 MHz. Druhý generátor je čtyřkanálový, poskytující synchronizované PWM signály, přičemž lze libovolně konfigurovat střídu a délku pulsu jednotlivých signálů v *Independent* módu. Za pomoci *Equidistant* módu pak lze poměrově udržovat fázi mezi signály a nastavovat všem stejnou délku pulsu.

Závěrem byl navržen a z části do platformy LEO naprogramován logický analyzátor poskytující 8 vstupních kanálů. Dva kanály lze využít jako zdroj triggeru s možností výběru spouštěcí události v podobě vzestupné či sestupné hrany. Logický analyzátor nabízí možnost konfigurace pretriggeru a tři způsoby spouštění; *Auto*, *Normal* a *Single*. Maximální počet vzorků je zatím omezen využitou pamětí, a proto bude třeba alokaci paměti v budoucnu předělat na dynamickou.

LITERATURA

- [1] STMicroelectronics. *STM32F303xD STM32F303xE: ARM®Cortex®-M4 32b MCU+FPU*. 2016, DocID026415 Rev 5. Dostupné z: <<http://www.st.com/content/ccc/resource/technical/document/datasheet/2c/6f/d7/64/1f/a3/4f/c9/DM00118585.pdf/files/DM00118585.pdf/jcr:content/translations/en.DM00118585.pdf>>
- [2] STMicroelectronics. *RM0316 Reference manual: STM32F303xB/C/D/E, STM32F303x6/8, STM32F328x8, STM32F358xC, STM32F398xE advanced ARM-based MCUs*. 2017, DocID022558 Rev 8. Dostupné z: <http://www.st.com/content/ccc/resource/technical/document/reference_manual/4a/19/6e/18/9d/92/43/32/DM00043574.pdf/files/DM00043574.pdf/jcr:content/translations/en.DM00043574.pdf>
- [3] *STMicroelectronics, Microcontrollers* [online]. Copyright © 2017 STMicroelectronics [cit. 03.12.2017]. Dostupné z: <<http://www.st.com/en/microcontrollers/stm32f303re.html>>
- [4] STMicroelectronics. *UM1724 User manual: STM32 Nucleo-64 board* 2016, DocID025833 Rev 11. Dostupné z: <http://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf>
- [5] *LEO - Little Embedded Oscilloscope based on STM32 | STMicroelectronics Community. Welcome | STMicroelectronics Community* [online]. Copyright © 2017 Jive Software [cit. 03.12.2017]. Dostupné z: <<https://community.st.com/videos/1336-leo-little-embedded-oscilloscope-based-on-stm32>>
- [6] *STMicroelectronics, STM32CubeMX* [online]. Copyright © 2017 STMicroelectronics [cit. 03.12.2017]. Dostupné z: <<http://www.st.com/en/development-tools/stm32cubemx.html>>
- [7] *FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions*. [online]. Copyright © 2017 FreeRTOS [cit. 03.12.2017]. Dostupné z: <<https://www.freertos.org/>>
- [8] *FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions*. [online]. Copyright © 2017 FreeRTOS [cit. 03.12.2017]. Dostupné z: <<https://www.freertos.org/a00111.html>>

- [9] *STMicroelectronics, Embedded software* [online]. Copyright © 2017 STMicroelectronics [cit. 03.12.2017]. Dostupné z: <<http://www.st.com/en/embedded-software/stm32cubef3.html>>
- [10] *Microsoft, .NET Framework system requirements | Microsoft Docs*. [online]. Copyright © 2017 Microsoft [cit. 05.12.2017]. Dostupné z: <<https://docs.microsoft.com/en-us/dotnet/framework/get-started/system-requirements>>
- [11] *HP Memory Project* [online]. Copyright © [cit. 07.12.2017]. Dostupné z: <http://hpmemoryproject.org/an/pdf/an_200.pdf>
- [12] Hong Kong X'tals Limited. *HC49US Thru-hole Crystals*. 2017, Ver 1.1. Dostupné z: <http://www.hongkongcrystal.com/wp-content/uploads/images/cataloguePDF/92-93-9S-94-95_hkc.pdf>
- [13] Keysight. *53131A 225 MHz Universal Frequency Counter/Timer [Discontinued] | Keysight (formerly Agilent's Electronic Measurement)*. [online]. Copyright © Keysight Technologies 2000 [cit. 30.12.2017]. Dostupné z: <<https://literature.cdn.keysight.com/litweb/pdf/5967-6039EN.pdf?id=1000071171:epsg:dow>>
- [14] Keysight. *33250A Function / Arbitrary Waveform Generator, 80 MHz [Discontinued] | Keysight (formerly Agilent's Electronic Measurement)*. [online]. Copyright © Keysight Technologies 2000 [cit. 30.12.2017]. Dostupné z: <<https://literature.cdn.keysight.com/litweb/pdf/5968-8807EN.pdf?id=1000034211:epsg:dow>>

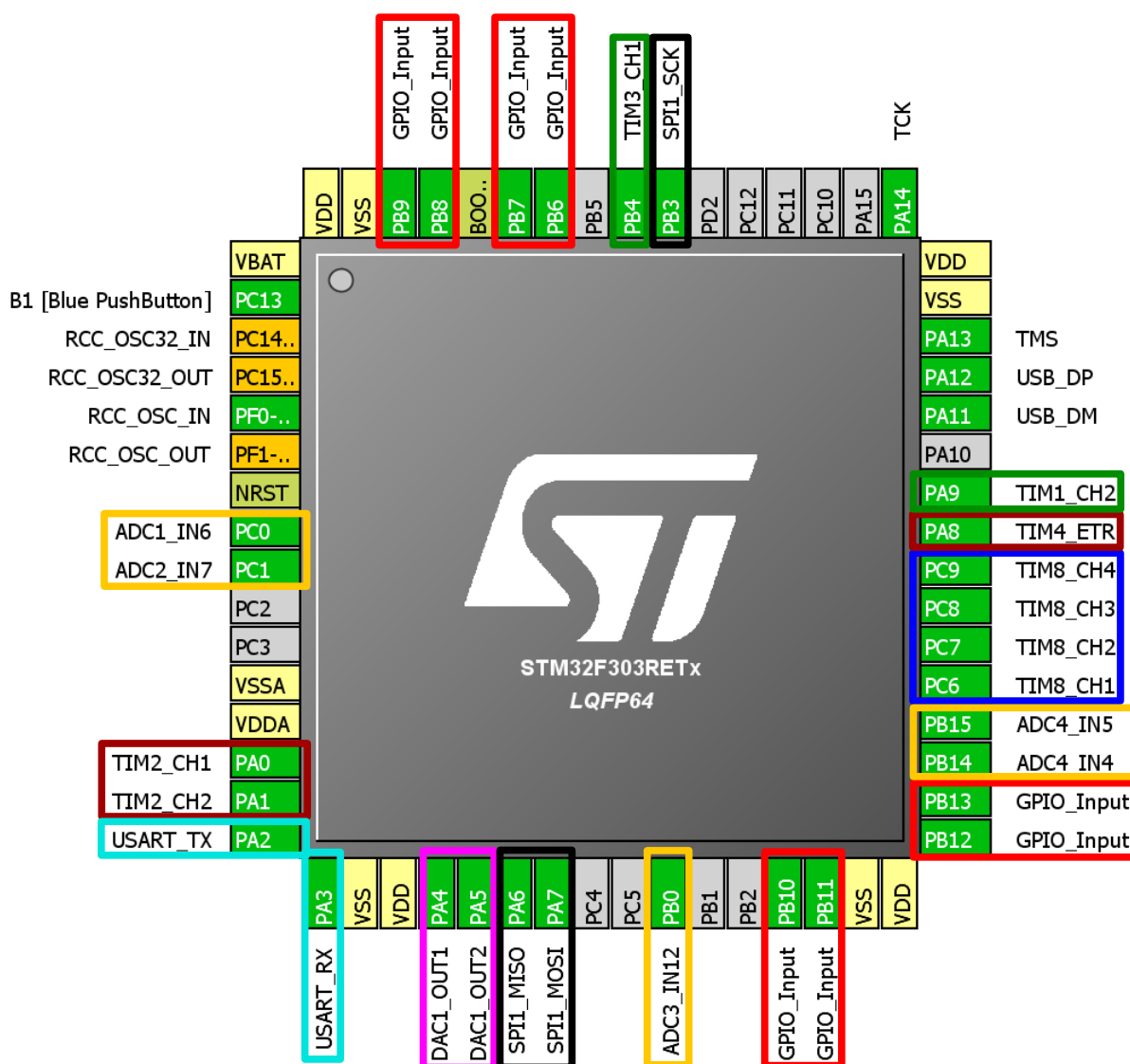
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ADC	Analog to Digital Converter
APB	Advanced Peripheral Bus
BOM	Bill Of Materials
DAC	Digital to Analog Converter
DCMI	Digital Camera Interface
DMA	Direct Memory Access
ETR	External Trigger
ETRP	External Trigger Prescaler
FreeRTOS	Free Real Time Operating System
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
HSE	High Speed External
ISR	Interrupt Service Routine
LED	Light Emitting Diode
LEO	Little Embedded Oscilloscope
MCU	Micro-Controller Unit
MCO	Master Clock Output
ppm	parts per million
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SB	Solder Bridge
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VCP	Virtual COM Port

SEZNAM PŘÍLOH

A Pinout mikroprocesoru	80
B Vyvedené funkce na konektory vývojového kitu	81
C Obsah přiloženého CD	82

A PINOUT MIKROPROCESORU



Obr. A.1: Pinout mikrokontroléru STM32F303RE a zobrazení mapování funkcí

Barevně jsou vyznačeny funkce; **dvoukanálový generátor PWM signálu s časově proměnnou střídou**, **čtyřkanálový generátor synchronizovaných PWM signálů**, **dvoukanálový funkční generátor + generátor stejnosměrného napětí**, **osmikanálový logický analyzátor**, **univerzální čítač**, **čtyřkanálový osciloskop + voltmetr**, **sériové komunikační rozhraní USART**, **sériové komunikační rozhraní SPI**. Periferie komunikačního rozhraní SPI je vyhrazena pro případnou možnost komunikace s rozšiřujícími deskami, tzv. *expansion boardy*.

C OBSAH PŘILOŽENÉHO CD

MucalaJan_Universal_Counter_for_System_LEO.pdf

Elektronický text diplomové práce

Instrulab_V2

Vytvořené programy

- Cesta k projektu s programy pro mikrokontrolér STM32F303RE (Keil μ Vision):
Instrulab_V2/V2/Projects/STM32F303 Nucleo/Instrulab_F303_Nucleo.uvprojx
- Cesta k projektu PC aplikace psané v jazyce C# (Visual Studio):
Instrulab_V2/PC_APP/InstruLab/InstruLab.sln

Images

Obrázky použité v práci