

**Bakalářská práce**



**České  
vysoké  
učení technické  
v Praze**

**F3**

**Fakulta elektrotechnická  
Katedra měření**

## **Virtuální přístroj s mikrořadičem pro analýzu signálu v modulační doméně**

**Martin Cejp**

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

**Obor: Kybernetika a robotika**

**Studijní program: Senzory a přístrojová technika**

**Květen 2017**





## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Martin Cejp**

Studijní program: **Kybernetika a robotika**  
Obor: **Senzory a přístrojová technika**

Název tématu česky: **Virtuální přístroj s mikrořadičem pro analýzu signálu v modulační doméně**

Název tématu anglicky: **Microcontroller-based Virtual Instrument for Signal Analysis in the Modulation Domain**

### Pokyny pro vypracování:

Navrhnete a realizujete virtuální přístroj pro analýzu signálu v modulační doméně. Pro realizaci funkcí se orientujte na využití čítačových periférií v mikrořadičích řady STM32. Přístroj bude vyhodnocovat parametry signálu, jako jsou perioda, délka impulsů, časový posun impulsů, a zaznamenávat změny těchto parametrů v čase. Dále bude umožňovat funkce standardního čítače, jako je přímé měření frekvence a poměru frekvencí, čítání impulsů a měření zpoždění impulsů ve více kanálech.

Vytvořte potřebné programové vybavení pro mikrořadič i aplikaci pro nadřazené PC. Experimentálně ověřte vlastnosti realizovaného zařízení při měření testovacích signálů s definovanými parametry.

### Seznam odborné literatury:

- [1] Yiu, J.: The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors,
- [2] STMicroelectronics: DS10362- STM32F303RE Data sheet
- [3] STMicroelectronics: RM0316- STM32F3 Reference manual

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

Datum zadání bakalářské práce: 24. ledna 2017

Platnost zadání do<sup>1</sup>: 30. září 2018

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



Prof. Ing. Pavel Rípka, CSc.  
děkan

V Praze dne 24. 1. 2017

<sup>1</sup> Platnost zadání je omezena na dobu tří následujících semestrů.



## Poděkování

Tímto bych chtěl poděkovat doc. Ing. Janu Fischerovi, CSc. za všechnen čas, který mi věnoval během konzultací. Dále pak své rodině, za veškerou podporu během studia.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 26. května 2017

.....

## Abstrakt

Práce se zabývá realizací virtuálního přístroje pro vyhodnocování parametrů signálu, jako jsou perioda, délka impulsů, časový posun impulsů, a zaznamenávání těchto parametrů v čase. Přístroj umožňuje i funkce klasického čítače, jako je přímé měření frekvence a poměru frekvencí, měření periody a měření zpoždění impulsů ve dvou kanálech.

Přístroj je založen na mikroprocesorové platformě STM32 a je ovládán řídicí aplikací na PC, se kterým komunikuje prostřednictvím rozhraní USB. Aplikace využívá framework Qt.

**Klíčová slova:** virtuální přístroje, univerzální čítač, STM32, modulační doména

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

## Abstract

This thesis describes the implementation of a virtual instrument for signal parameter measurement and logging of variations over time. The measured parameters include period, impulse duration and impulse time shift. The instrument also supports the features of a conventional counter, including direct counting measurement of frequency and frequency ratio, period measurement, and measurement of signal delay between two channels.

The instrument is based on the STM32 microprocessor platform and it is controlled through a PC application, making use of the Universal Serial Bus for communication. The application is based on the Qt framework.

**Keywords:** Virtual Instruments, universal counter, STM32, modulation domain

### **Title translation:**

Microcontroller-based Virtual Instrument for Signal Analysis in the Modulation Domain

## Obsah

<b>1 Úvod</b>	<b>1</b>		
1.1 Motivace	1		
<b>2 Problematika a cíle práce</b>	<b>3</b>		
<b>3 Rozbor funkcí čítače</b>	<b>5</b>		
3.1 Přímé měření frekvence	5		
3.1.1 Přístrojová realizace	8		
3.2 Měření periody, reciproční měření frekvence	9		
3.2.1 Přístrojová realizace	11		
3.3 Měření intervalu dvou událostí, měření fázového posunu	12		
3.3.1 Přístrojová realizace	13		
3.4 Měření poměru dvou frekvencí	14		
3.4.1 Přístrojová realizace	15		
<b>4 Realizace virtuálního přístroje</b>	<b>17</b>		
4.1 Představení platformy STM32	17		
4.2 Synchronní čítač	19		
4.2.1 Vznik aliasingu	20		
4.2.2 Detekce aliasingu	21		
4.3 Časová základna přístroje	22		
4.3.1 Generování časové základny	23		
4.3.2 Předzpracování časové základny děličkou a fázovým závěsem	25		
4.3.3 Možnosti zpřesnění časové základny	25		
4.3.4 Použité řešení	26		
4.4 Architektura přístroje	27		
4.5 Realizace přístrojových funkcí	28		
4.5.1 Přímé měření frekvence	28		
4.5.2 Měření periody	31		
4.5.3 Měření intervalu dvou událostí, měření fázového posunu	36		
4.5.4 Měření poměru dvou frekvencí	37		
4.6 Komunikace se zařízením	40		
4.6.1 Pakety pro přenos požadavků a odpovědí	41		
4.7 Realizace ovládací aplikace	45		
4.7.1 Softwarová platforma	45		
4.7.2 Architektura aplikace	46		
4.7.3 Grafické uživatelské rozhraní	46		
<b>5 Ověření vlastností přístroje měření</b>	<b>49</b>		
5.1 Měření referenčního signálu 1 MHz, 10 MHz	49		
5.2 Měření signálu PPS z GPS přijímače	49		
5.3 Porovnání zdrojů časové základny	52		
5.4 Ověření projevu aliasingu	52		
<b>6 Zhodnocení dosažených výsledků</b>	<b>55</b>		
<b>Bibliografie</b>	<b>57</b>		
<b>A Seznam souborů na CD</b>	<b>59</b>		

## Obrázky

1.1 Tři pohledy na analýzu signálu . .	1	4.4 Aliasing při synchronním čítání	21
3.1 Univerzální čítač HP 53131A . . . .	5	4.5 Graf závislosti zdánlivé frekvence na skutečné frekvenci signálu při synchronním čítání . . . . .	21
3.2 Časový průběh čítání událostí . . .	6	4.6 Krystal a jeho náhradní schéma při fundamentální frekvenci . . . . .	24
3.3 Časový průběh hradlovaného čítání	6	4.7 Teplotní charakteristika krystalu a obvodová kompenzace . . . . .	25
3.4 Nejzákladnější podoba čítače . . . .	6	4.8 Organizace modulů firmware přístroje. Šipky znázorňují směr volání funkcí. . . . .	27
3.5 Čítač s hradlem . . . . .	6	4.9 Stavový automat přístroje . . . . .	27
3.6 Kvantizační chyba při hradlovaném čítání . . . . .	7	4.10 Zapojení periférií mikrokontroléru pro přímé měření frekvence čítáním	29
3.7 Závislost chyby kvantizace přímého měření frekvence . . . . .	8	4.11 Časový průběh měření frekvence čítáním . . . . .	30
3.8 Schéma čítače v režimu přímého měření frekvence (podle [3]) . . . . .	8	4.12 Řetězení čítačů . . . . .	30
3.9 Příklady vstupního signálu čítače	9	4.13 Konfigurace pro měření N period	31
3.10 Měření periody . . . . .	9	4.14 Měření $n_p = 5$ period . . . . .	32
3.11 Časový průběh měření 1 periody	9	4.15 Konfigurace pro měření periody (32-bit, IC) . . . . .	33
3.12 Chyba kvantizace měření periody při $f_{timebase} = 100$ MHz a vybraných hodnotách $n_p$ . . . . .	10	4.16 Využití Input Capture pro měření periody a šířky pulzu . . . . .	33
3.13 Schéma přístroje realizujícího funkci měření periody . . . . .	11	4.17 Konfigurace pro měření periody (32-bit, IC, DMA) . . . . .	34
3.14 Časový průběh měření $n_p$ period	11	4.18 Ztracený DMA požadavek při vysoké frekvenci měřeného signálu	35
3.15 Kladný a záporný pulz . . . . .	12	4.19 Využití Input Capture pro měření periody s DMA a předdělením /4 .	36
3.16 Měření intervalu dvou událostí	12	4.20 Konfigurace pro měření intervalu dvou událostí (32-bit, IC) . . . . .	37
3.17 Časový průběh měření intervalu dvou událostí . . . . .	12	4.21 Zapojení periférií mikrokontroléru pro měření poměru dvou frekvencí	38
3.18 Schéma přístroje realizujícího měření intervalu dvou událostí . . . .	13	4.22 Měření poměru frekvencí . . . . .	39
3.19 Časový průběh měření intervalu náběžných hran . . . . .	13	4.23 Blokující model komunikace . . .	40
3.20 Měření poměru dvou frekvencí .	14	4.24 Asynchronní model komunikace s periodickým dotazováním . . . . .	41
3.21 Schéma přístroje realizujícího měření poměru dvou frekvencí . . . .	15	4.25 Asynchronní model komunikace se značkováním požadavků a odpovědí	42
3.22 Časový průběh měření poměru dvou frekvencí, $n_B = 5$ . . . . .	15	4.26 Formát paketu a konkrétní příklady . . . . .	43
4.1 Blokové schéma nejdůležitějších částí mikrokontroléru STM32F042	17		
4.2 Schéma periférie TIM2/TIM3 v mikrokontroléru STM32F042 . . . . .	19		
4.3 Činnost synchronního čítače . . . .	20		



4.27 Komunikační logika při použití paketového protokolu . . . . .	44
4.28 Signály a sloty: připojení přístroje a provedení jednoho měření . . . . .	46
4.29 Podoba grafického uživatelského rozhraní . . . . .	47
5.1 Kolísání měřené frekvence při použití interního oscilátoru HSI48 a zpětnovazebním doladováním (CRS) . . . . .	52

## Tabulky

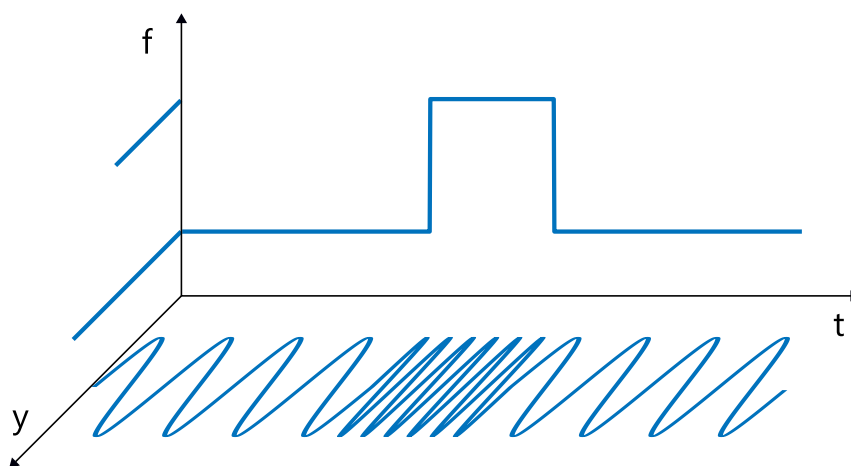
4.1 Vybrané elektrické parametry mikrokontroléru STM32F042 při napájecím napětí 3.3 V . . . . .	18
4.2 Tovární kalibrace interních oscilátorů vybraných mikrokontrolérů . . . . .	23
4.3 Přečtené a zapsané hodnoty během přímého přístupu do paměti . . . . .	35
5.1 Měření referenčního signálu 1 MHz metodou přímého měření frekvence . . . . .	50
5.2 Měření referenčního signálu 10 MHz metodou přímého měření frekvence . . . . .	50
5.3 Měření periody referenčního signálu 1 Hz (GPS - PPS) . . . . .	51
5.4 Měření 1MHz referenčního signálu za použití různých zdrojů časové základny . . . . .	52
5.5 Měření 66MHz signálu při frekvenci časové základny 48 MHz. . . . .	53



# Kapitola 1

## Úvod

### 1.1 Motivace



**Obrázek 1.1:** Tři pohledy na analýzu signálu. Časová doména  $y(t)$ , frekvenční doména  $y(f)$  a modulační doména  $f(t)$ . Podle [1].

Modulační doménu signálu lze chápat jako funkci frekvence v čase. Obr. 1.1 znázorňuje tři pohledy na vlastnosti signálu – časovou doménu, frekvenční doménu a modulační doménu.

Důvodů k měření časových vlastností signálu lze nalézt mnoho. Zkoumanou veličinou může být přímo frekvence, například při měření oscilátorů pro hodiny reálného času, nebo veličina, která je frekvencí přímo určená – otáčky motoru.

Na výstupu senzorů se zase často setkáváme s pulzně šířkovou modulací, pro jejíž interpretaci potřebujeme znát střídu signálu, tedy podíl šířky pulzu a periody.

V neposlední řadě, ultrazvukový senzor vzdálenosti je založen na vyslání signálu a měření doby do příchodu jeho odrazu. Pro určení vzdálenosti je tedy potřeba vyhodnotit interval mezi dvěma událostmi.

Zatímco levné přístroje pro měření veličin jako napětí a proud jsou běžně k dispozici, cenově dostupné a zároveň kvalitní zařízení pro analýzu v modulační doméně na trhu chybí. Realizace takového přístroje tradičním způsobem by mohla spočívat v návrhu schematického zapojení, výrobě plošného spoje a osazení vhodnými součástkami. Nelze si však nepovšimnout dramatického vývoje, který v posledních letech zažívá vestavitelná výpočetní technika, a to především mikrokontroléry a čipy typu System-on-a-Chip (SoC). Moderní mikrokontrolér nabízí širokou paletu funkcí; kromě 32-bitového výpočetního jádra je k dispozici několik komunikačních rozhraní, pokročilé čítače a časovače, převodníky A/D, D/A nebo DMA řadič. Přitom náklady na použití mikrokontroléru jsou řádově nižší, než použití tradiční technologie. Důsledkem tohoto pokroku je trend virtuálních přístrojů (Virtual Instruments), což jsou měřicí a laboratorní přístroje založené na univerzálních hardwarových platformách.

Díky moderní a rychlé polovodičové technologii mohou mikrokontroléry pracovat na frekvencích v řádech desítek a někdy i stovek MHz. To jim dává výborné předpoklady pro měření časových veličin. Použití mikrokontroléru také přináší nové možnosti zpracování zachycených dat – např. průměrování, hledání minima a maxima. Díky flexibilitě softwarového řešení je možné virtuální přístroj upravovat pro potřeby konkrétních měřicích úloh.

## Kapitola 2

### Problematika a cíle práce

Cílem této práce je návrh měřícího přístroje založeného na mikrokontroléru z rodiny STM32. Navrhovaný přístroj by měl sloužit jako univerzální čítač s možností sledovat časové změny měřených parametrů signálu. Dostupné čítačové funkce budou:

1. měření frekvence čítáním impulsů po stanovenou dobu  $T$  (0,1 s; 1 s; 10 s)
2. reciproční měření frekvence z doby trvání jedné periody
3. měření kladného, záporného pulzu, měření střídy
4. měření intervalu  $t_{AB}$  signálů na dvou vstupech (náběžná/spádová hrana)
5. měření poměru dvou frekvencí

Součástí přístroje nebude ovládací panel, místo toho bude zahájení měření a zobrazení výsledků zajišťovat ovládací aplikace pro PC, která bude s přístrojem komunikovat přes rozhraní USB<sup>1</sup>. Pro použití aplikace studenty v laboratoři je žádoucí přenositelnost minimálně na operační systémy MS Windows (XP a novější), macOS a Linux.

Z kvantitativního hlediska nás budou zkoumat následující parametry:

1. přípustný rozsah měřených časových veličin (frekvence, perioda)
2. rozlišení měřené veličiny
3. nejistota měření

V rámci této práce se realizuje implementace pouze pro vybrané mikrořadiče z rodiny STM32. Nicméně jedním z hlavních cílů je, aby vzniklý zdrojový kód byl co nejlépe přenositelný i na jiné mikrořadiče z této rodiny.

Jedním z vedlejších cílů práce je, aby mohla sloužit jako výukový materiál k problematice čítačů, a to i pro studenty, kteří s měřícími přístroji nemají předchozí zkušenosti. Může se zdát, že některé myšlenky nebo schémata se v textu opakují jen s malými změnami. Považuji však za důležité vysvětlovat problematiku postupně, už proto, aby byl zřejmý rozdíl mezi omezeními vyplývajícími z použitých metod měření a těmi, které jsou dány realizací přístroje.

---

<sup>1</sup>Universal Serial Bus - Univerzální sériová sběrnice



## Kapitola 3

### Rozbor funkcí čítače



**Obrázek 3.1:** Univerzální čítač HP 53131A. Tento model disponuje dvěma kanály o šířce pásma 225 MHz s možností třetího kanálu s pásmem 1,2 GHz. Základními funkcemi čítače jsou: měření frekvence, měření periody, měření střídy, měření poměru frekvencí. Přístroj umožňuje nastavit rozhodovací úroveň, terminaci ( $50\ \Omega$ ,  $1\ M\Omega$ ), vazbu (AC, DC) a další parametry zpracování signálu. Zdroj: [2]

Čítač je měřicí přístroj určený k měření časových veličin, především frekvence a periody nějakého signálu. Na obr. 3.1 je vyobrazen čítač HP 53131A. Jedná se o starší model, pro ilustraci základních funkcí však postačí.

Funkce čítače nejprve rozebereme poněkud abstraktně, se zaměřením na použité metody měření a jejich omezení.

### 3.1 Přímé měření frekvence

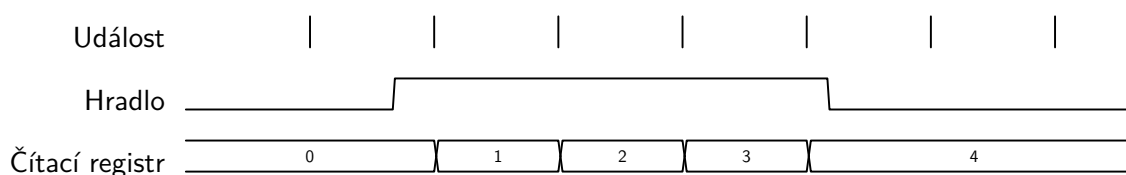
Základní funkcí klasického čítače je počítání nějakých *událostí*. Jeho jádrem je *čítací registr*, jehož hodnota se na začátku měření nastaví na 0 a následně se inkrementuje

s každým příchozím impulsem. Hodnota v registru tedy udává počet zaznamenaných událostí. Časový průběh čítání je znázorněn na obr. 3.2.



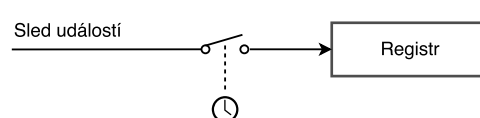
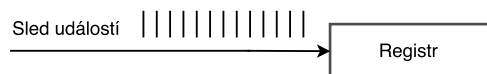
**Obrázek 3.2:** Časový průběh čítání událostí

Čítač také obsahuje *hradlo*, které umožňuje omezit dobu, po kterou čítání probíhá. Pokud se hradlo otevře jen na přesně stanovený interval, je možné určit počet událostí za jednotku času. Obr. 3.3 znázorňuje průběh měření s použitím hradlování.



**Obrázek 3.3:** Časový průběh hradlovaného čítání

Obr. 3.4 a 3.5 schematicky znázorňují základní čítač a čítač s hradlem.



**Obrázek 3.4:** Nejzákladnější podoba čítače

**Obrázek 3.5:** Čítač s hradlem

Hradlo je tedy otevřené po nějakou dobu  $T_{gate}$ , za kterou přístroj napočítá  $N$  událostí. Frekvence událostí  $f_{event}$  je pak dána vzorcem

$$f_{event} = \frac{N}{T_{gate}} \quad (3.1)$$

Je nezbytné si uvědomit, že tato rovnice platí přesně jen za určitých předpokladů:

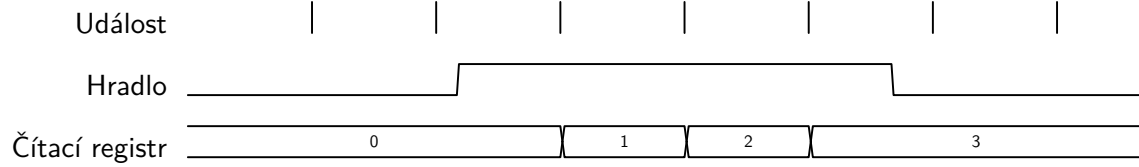
1. frekvence událostí je během hradlování neměnná
2. doba hradlování  $T_{gate}$  je celočíselným násobkem periody událostí

Pokud není dodržena první podmínka, vypočítaná hodnota nemá význam frekvence, nýbrž její střední hodnoty po dobu hradlování. Porušení první podmínky implikuje, že nemůže být splněna ani druhá podmínka (nelze mluvit o periodě událostí), z čehož nutně vyplývá, že měření bude zatíženo určitou chybou.

Důsledek porušení druhé podmínky znázorňuje obr. 3.6. Frekvence událostí i doba hradlování jsou stejné jako na obr. 3.3, pouze se posunula doba zahájení a ukončení



hradlování. V důsledku toho přístroj změřil o jeden impuls méně a podle rovnice (3.1) určí jinou frekvenci událostí  $f_{event}$ .



**Obrázek 3.6:** Kvantizační chyba při hradlovaném čítání

V praxi často není možné zaručit ani jednu z výše uvedených podmínek. Jev znázorněný na obr. 3.6 se označuje jako *kvantizační chyba*<sup>1</sup> a omezuje rozlišení měřené frekvence.

Veličina  $N$  v rovnici (3.1) je nezáporné celé číslo, rozlišení měřené frekvence  $\Delta f_{event}$  je tedy určeno jako

$$\Delta f_{event} = \frac{N+1}{T_{gate}} - \frac{N}{T_{gate}} = \frac{1}{T_{gate}} \quad (3.2)$$

$$\frac{\Delta f_{event}}{f_{event}} = \frac{1}{T_{gate} \cdot f_{event}} \quad (3.3)$$

Důležitým poznatkem je, že rozlišení frekvence je **nezávislé na skutečné frekvenci událostí** a relativní chyba tedy klesá s rostoucí frekvencí.

Zkoumanou veličinou často není přímo frekvence, nýbrž perioda, definovaná vztahem

$$T_{event} = \frac{1}{f_{event}} \quad (3.4)$$

Rozlišení určení periody  $\Delta T_{event}$  při použití hradlovaného čítání je

$$\Delta T_{event} = \frac{T_{gate}}{N-1} - \frac{T_{gate}}{N} = \frac{NT_{gate} - NT_{gate} + T_{gate}}{N(N-1)} = \frac{T_{gate}}{N(N-1)} = \frac{T_{event}}{N-1} \quad (3.5)$$

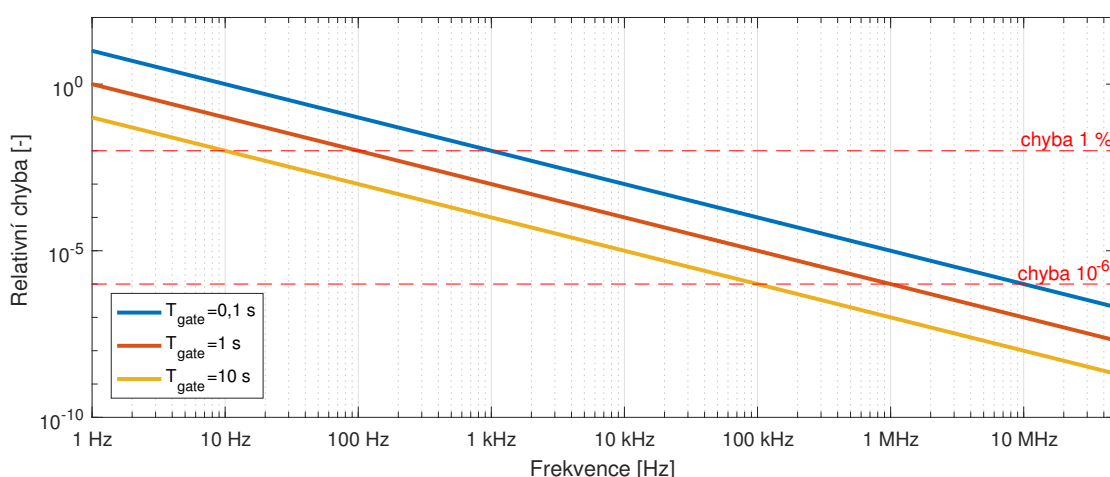
$$\Delta T_{event} = \frac{T_{event}}{\frac{T_{gate}}{T_{event}} - 1} = \frac{T_{event}^2}{T_{gate} - T_{event}} \quad (3.6)$$

$$\frac{\Delta T_{event}}{T_{event}} = \frac{T_{event}}{T_{gate} - T_{event}} \quad (3.7)$$

Rozlišení periody závisí jak na době hradlování, tak na skutečné periodě signálu. S klesající periodou signálu se zlepšuje rozlišení periody.

Pro lepší představu graf 3.7 ukazuje závislost relativní chyby měření frekvence na skutečné frekvenci signálu pro vybrané hodnoty  $T_{gate}$ .

<sup>1</sup>V zahraniční literatuře také  $\pm 1$  count error [3].

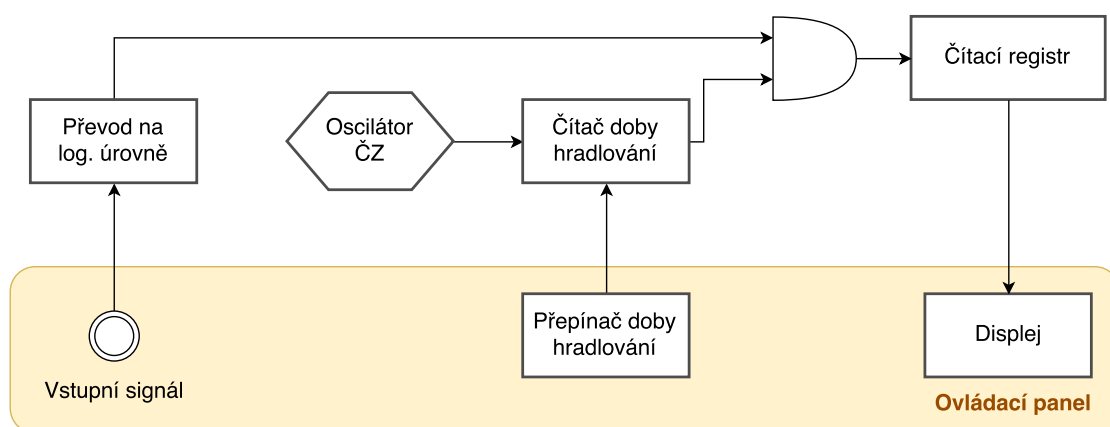


Obrázek 3.7: Závislost chyby kvantizace přímého měření frekvence

### 3.1.1 Přístrojová realizace

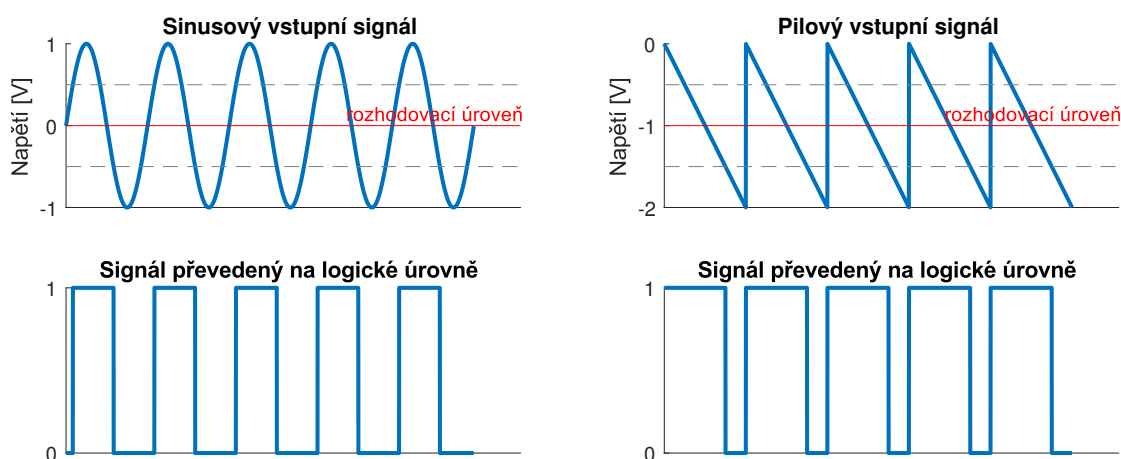
Dosud jsme popisovali čítání nějakých obecných *událostí*. Skutečný měřicí přístroj, jehož blokové schéma je na obr. 3.8, pracuje s elektrickým signálem. Na vstup přístroje může být zapojen periodický signál o různém průběhu, např. obdélník, sinus, pila. Přístroj typicky umožňuje přinejmenším nastavení rozhodovací úrovně a vazby AC nebo DC.

Na obr. 3.9 jsou ukázky signálů před a po převodu na logické úrovně, včetně nastavené rozhodovací úrovně a z ní vyplývajících skutečných rozhodovacích hladin, které jsou dvě, protože vstupy přístroje mají určitou hysterezi.



Obrázek 3.8: Schéma čítače v režimu přímého měření frekvence (podle [3])

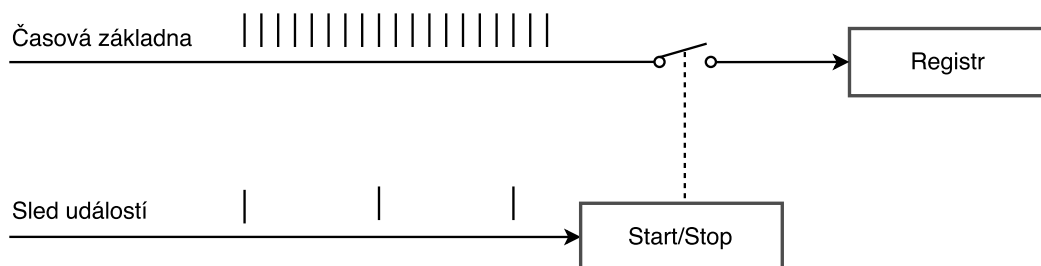
Podrobnější rozbor analogového předzpracování je nad rámec této práce, nadále tedy budeme uvažovat pouze logický signál, ve kterém je událost představována přechodem z log. 0 do log. 1.



Obrázek 3.9: Příklady vstupního signálu čítače

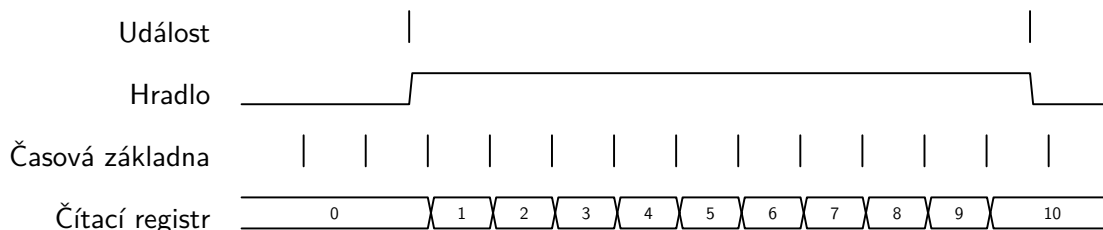
## 3.2 Měření periody, reciproční měření frekvence

Další funkcí klasického čítače je měření periody. Principiální schéma této metody měření je na obr. 3.10.



Obrázek 3.10: Měření periody

Při zahájení měření je vynulován čítací registr. Po detekci události je otevřeno hradlo a zahájeno čítání. Dokud je hradlo otevřené, registr počítá pulzy časové základny. Další příchozí událost hradlo uzavře a měření je dokončeno. Hodnota registru pak představuje periodu signálu jako násobek periody časové základny  $T_{timebase}$ . Průběh měření periody je zachycen na obr. 3.11.



Obrázek 3.11: Časový průběh měření 1 periody

Po dokončení měření je v čítacím registru uložena hodnota  $N_c$ . Perioda událostí  $T_{event}$  se následně určí jako

$$T_{event} = N_c \cdot T_{timebase} \quad (3.8)$$

Podle vztahu 3.4 můžeme z periody  $T_{event}$  určit frekvenci  $f_{event}$ . Mluvíme pak o *recipročním měření frekvence*.

Protože  $N_c$  je kladné celé číslo, rozlišení periody bude

$$\Delta T_{event} = (N_c + 1) \cdot T_{timebase} - N_c \cdot T_{timebase} = T_{timebase} \quad (3.9)$$

Rozlišení určení periody je nezávislé na skutečné periodě událostí. Pokud je zkoumanou veličinou frekvence, lze obdobným způsobem jako v předchozí kapitole ukázat, že rozlišení recipročního určení frekvence se s rostoucí frekvencí zhoršuje.

Jak vyplynulo z rovnic 3.3 a 3.7, při přímém měření frekvence lze zvýšit přesnost měření prodloužením doby hradlování. Při měření periody je možné místo 1 periody změřit interval  $n_p$  po sobě jdoucích period událostí. Rovnice 3.8 pak přejde na

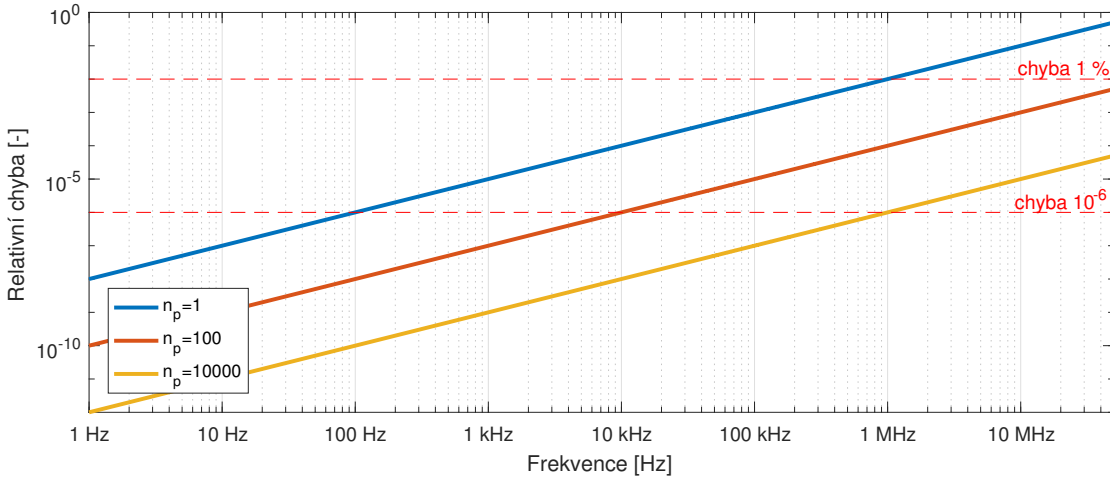
$$T_{event} = \frac{N_c \cdot T_{timebase}}{n_p} \quad (3.10)$$

Opět určíme rozlišení periody:

$$\Delta T_{event} = \frac{(N_c + 1) \cdot T_{timebase}}{n_p} - \frac{N_c \cdot T_{timebase}}{n_p} = \frac{T_{timebase}}{n_p} \quad (3.11)$$

$$\frac{\Delta T_{event}}{T_{event}} = \frac{1}{n_p} \frac{T_{timebase}}{T_{event}} \quad (3.12)$$

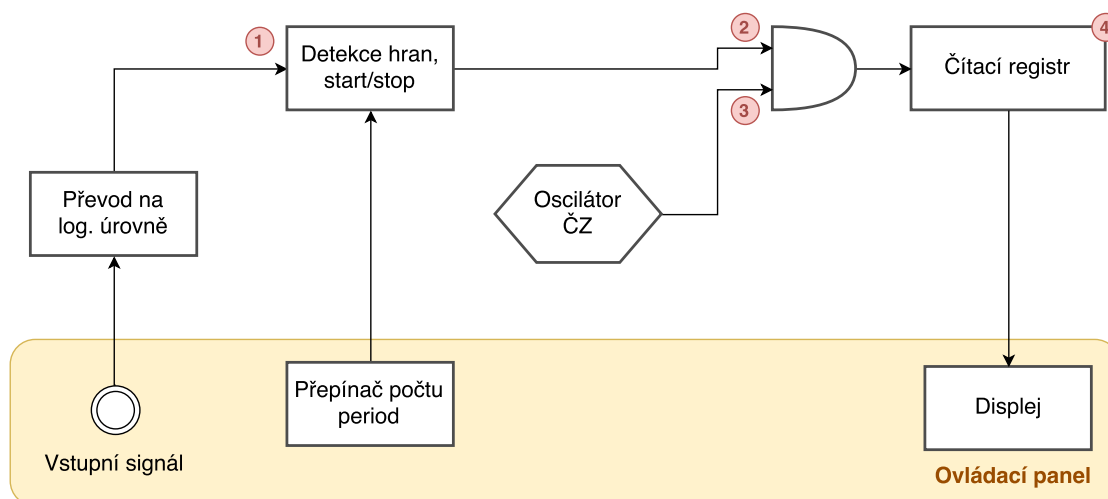
Je zřejmé, že rozlišení se zvýší  $n_p$ -krát. Relativní chyba je tentokrát nepřímo úměrná periodě a závisí na rozlišovací schopnosti přístroje. Závislost chyby na frekvenci pro různá  $n_p$  ilustruje graf na obr. 3.12.



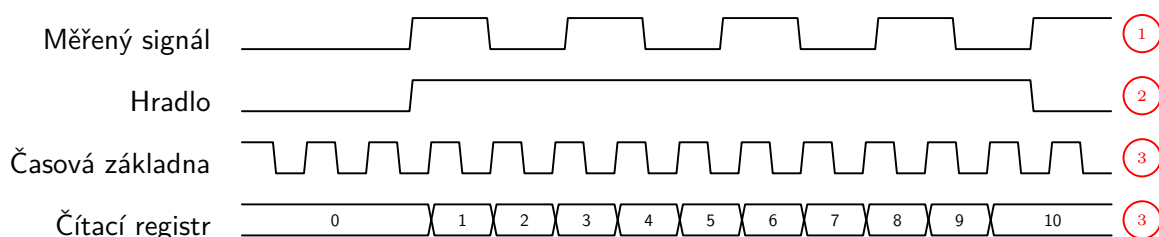
**Obrázek 3.12:** Chyba kvantizace měření periody při  $f_{timebase} = 100$  MHz a vybraných hodnotách  $n_p$

### 3.2.1 Přístrojová realizace

Přístroj realizující měření periody je schematicky znázorněn na obr. 3.13, průběh měření  $n_p$  period je pak na obr. 3.14.



Obrázek 3.13: Schéma přístroje realizujícího funkci měření periody



Obrázek 3.14: Časový průběh měření  $n_p$  period

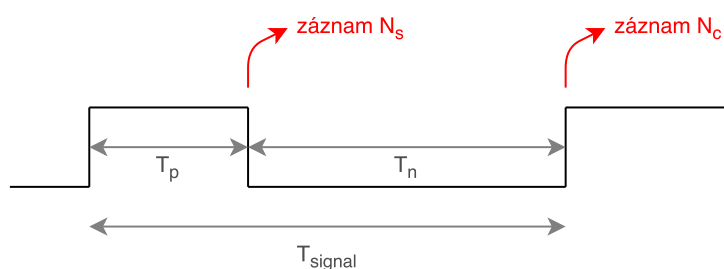
Pokud potřebujeme měřit i střídu signálu, příp. šířku pulzů, lze přístroj měřící periodu rozšířit ještě jedním způsobem – přidáním registru, do nějž se hodnota čítacího registru přenesou při *sestupné* hraně měřeného signálu. Zaznamenejme tedy navíc hodnotu  $N_s$ , ze které odvodíme šířku kladného pulzu  $T_p$ , záporného pulzu  $T_n$ , a střídu  $d$ . Pro jednoduchost předpokládáme  $n_p = 1$ .

$$T_p = N_s \cdot T_{timebase} \quad (3.13)$$

$$T_n = (N_c - N_s) \cdot T_{timebase} \quad (3.14)$$

$$d = \frac{N_s}{N_c} \quad (3.15)$$

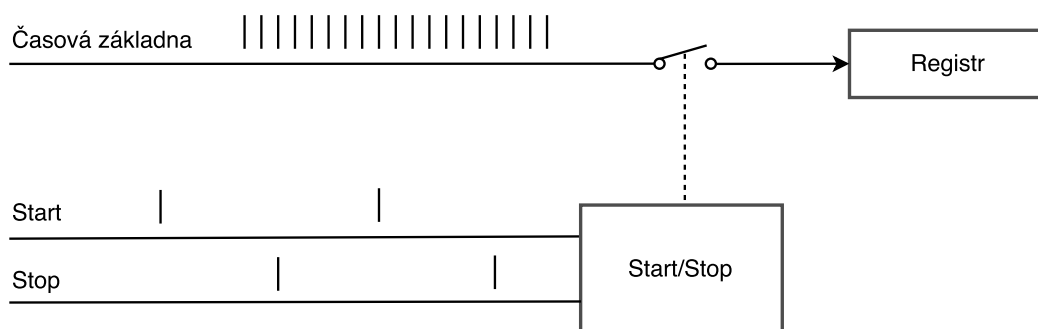
Vztah veličin ilustruje obr. 3.15.



Obrázek 3.15: Kladný a záporný pulz

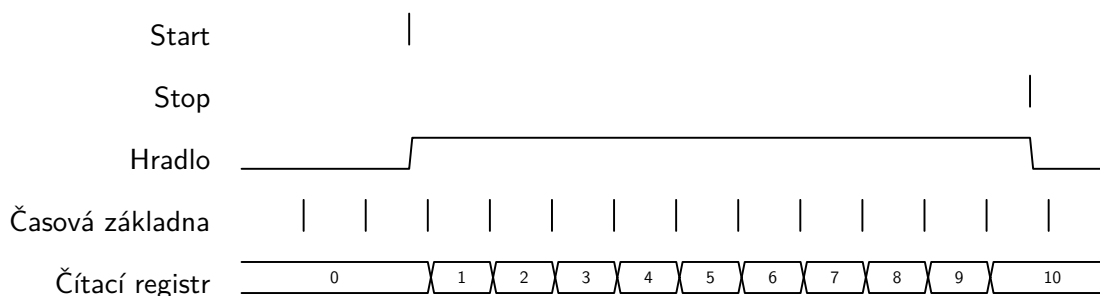
### 3.3 Měření intervalu dvou událostí, měření fázového posunu

Připomeňme, že při měření periody jsme uvažovali hradlo řízené sledem událostí, kdy jedna událost hradlo otevřela a některá následující událost hradlo uzavřela. Tuto myšlenku lze zobecnit a oddělit od sebe signály zahajující a ukončující měření. (obr. 3.16)



Obrázek 3.16: Měření intervalu dvou událostí

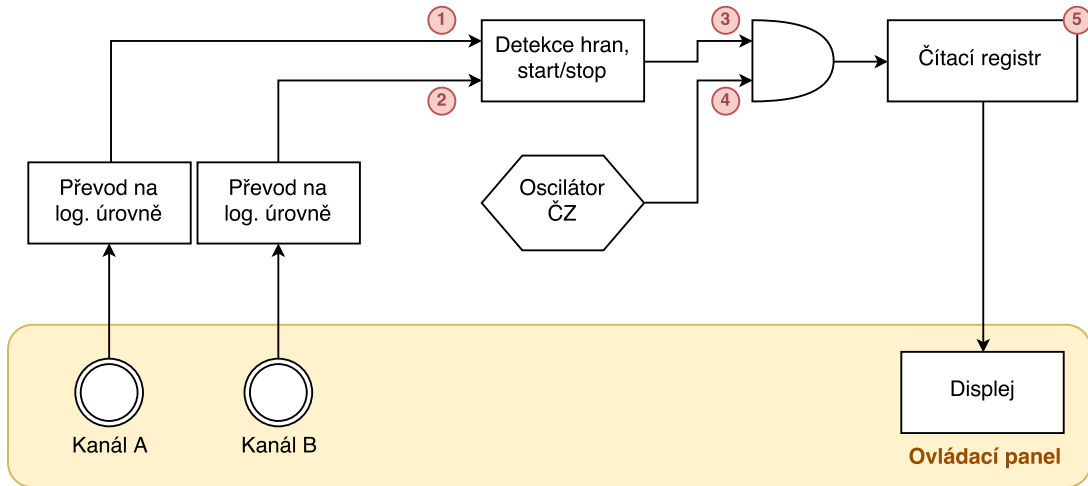
Touto metodou můžeme měřit délku intervalu mezi signály Start a Stop (obr. 3.17). Rozlišení měření je opět dané periodou časové základny.



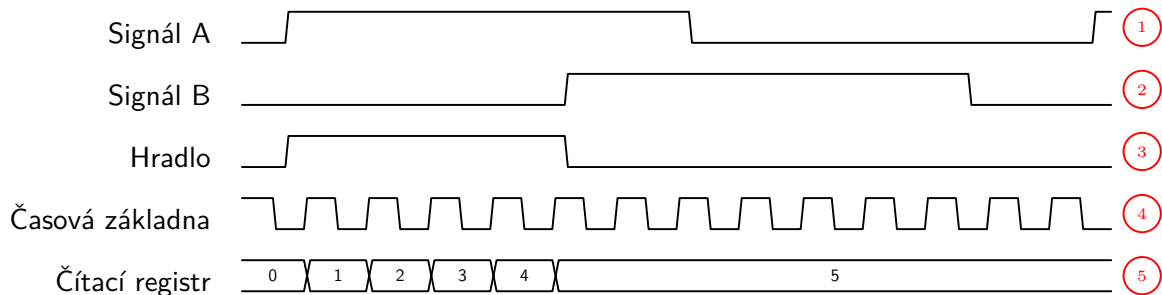
Obrázek 3.17: Časový průběh měření intervalu dvou událostí

### 3.3.1 Přístrojová realizace

Abychom mohli zavést pojem *fáze* a zkoumat možnosti jejího měření, musíme opět přejít od měření abstraktních událostí k periodickým signálům. Na obr. 3.18 je schéma přístrojové realizace, na obr. 3.19 pak průběh jednoho měření.



Obrázek 3.18: Schéma přístroje realizujícího měření intervalu dvou událostí



Obrázek 3.19: Časový průběh měření intervalu náběžných hran

Řekněme, že náběžné hraně signálu A odpovídá impuls Start a náběžné hraně signálu B odpovídá impuls Stop. Po dokončení měření je v čítacím registru uložena hodnota  $N_i$ . Interval událostí se následně určí jako

$$T_{interval} = N_i \cdot T_{timebase} \quad (3.16)$$

Pokud bude splněna ještě podmínka, že signály A i B jsou periodické a jejich frekvence se rovnají, tedy  $f_A = f_B = f_{signal}$ , pak můžeme mluvit o *fázovém posunu signálů*. Protože jsou oba signály periodické, nelze jednoznačně určit, který z nich „předbíhá“. Volba je tedy na nás. V souladu s časovým diagramem 3.19 rekneme, že signál B je zpožděn za signálem A, jeho fáze je tedy menší. Fáze signálu B vůči signálu A se pak určí jako

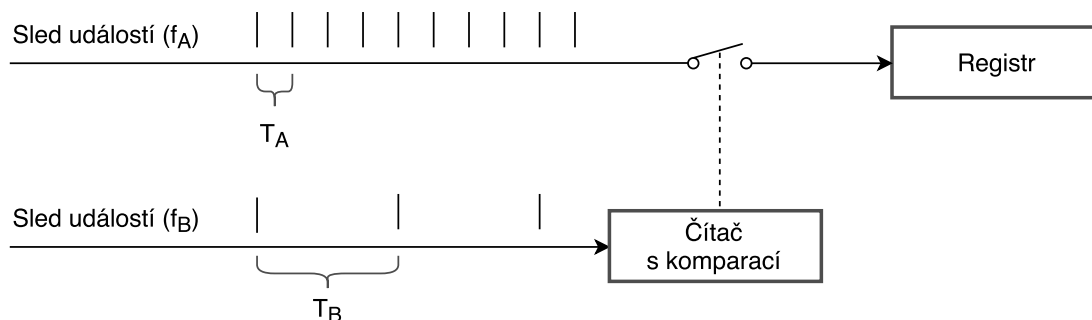
$$\varphi_{B-A} = \frac{-2\pi \cdot T_{interval}}{T_{signal}} = -2\pi \cdot T_{interval} \cdot f_{signal} \quad (3.17)$$

V časovém diagramu je perioda signálů 13 taktů časové základny. Přístroj naměří interval 5, fázi tedy určí jako

$$\varphi_{B-A} = \frac{-2\pi \cdot 5}{13} = -\frac{10}{13}\pi \approx -138^\circ$$

### 3.4 Měření poměru dvou frekvencí

Na obr. 3.20 je principiální schéma měření poměru frekvencí.



Obrázek 3.20: Měření poměru dvou frekvencí

Pro měření poměru frekvencí se použije hradlo, které je otevřené po pevně daný počet  $n_B$  period  $T_B$ . Během doby otevření hradla registr napočítá  $n_A$  událostí o periodě  $T_A$ , kde platí

$$n_A = \frac{n_B T_B}{T_A} \quad (3.18)$$

Z definice:

$$f_A = \frac{1}{T_A}, \quad f_B = \frac{1}{T_B} \quad (3.19)$$

Můžeme dosadit a vyjádřit:

$$\frac{f_A}{f_B} = \frac{n_A}{n_B} \quad (3.20)$$

Pro lepší uchopení zavedeme bezrozměrnou veličinu  $r$  představující zkoumaný poměr frekvencí:

$$r = \frac{f_A}{f_B} \quad (3.21)$$

Protože počet událostí  $n_A$  je nezáporné celé číslo, bude podobně jako při přímém měření frekvence vznikat kvantizační chyba  $\Delta r$ . Odvodíme:

$$\Delta r = \frac{n_A + 1}{n_B} - \frac{n_A}{n_B} = \frac{1}{n_B} \quad (3.22)$$

$$\frac{\Delta r}{r} = \frac{1}{n_B} \frac{f_B}{f_A} = \frac{n_B}{n_A n_B} = \frac{1}{n_A} \quad (3.23)$$

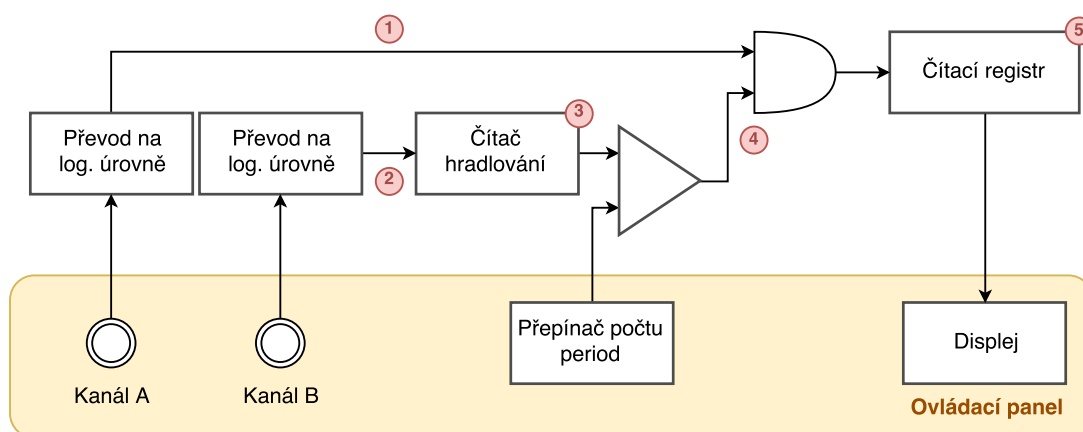
Ze vzorců (3.22) a (3.23) plynou dva poznatky; v první řadě je zřejmé, že pro přesné měření poměru dvou blízkých frekvencí bude nutné měřit velký počet period. Druhým



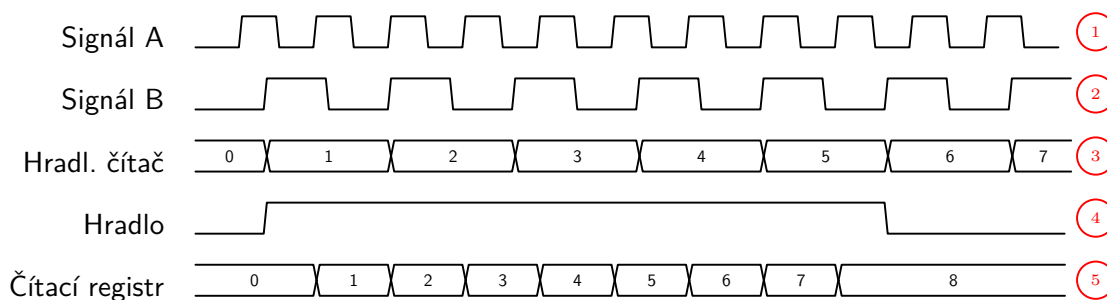
výsledkem je, že přesnost měření není invariantní vůči vzájemné záměně signálů. Jinými slovy, pro dosažení maximálního rozlišení je výhodné pro hradlování (kanál B) použít z obou signálů ten o nižší frekvenci. Lze si to představit tak, že kvantizační chyba „vzniká“ na kanále A, a pokud je zde kratší perioda, tedy více pulzů celkem, bude menší i relativní chyba způsobená jednou nesprávně změřenou periodou.

### 3.4.1 Přístrojová realizace

Na obr. (3.21) je schematicky znázorněn přístroj, který tuto funkci realizuje. Signál kanálu B se používá pro inkrementaci hradlovacího čítače. Komparátor výstupu hradlo otevírá a uzavírá podle počtu period zaznamenaných na kanálu B. Hlavní čítací registr počítá periody kanálu A po dobu otevření hradla. Průběh jednoho měření je na obr. (3.22).



Obrázek 3.21: Schéma přístroje realizujícího měření poměru dvou frekvencí



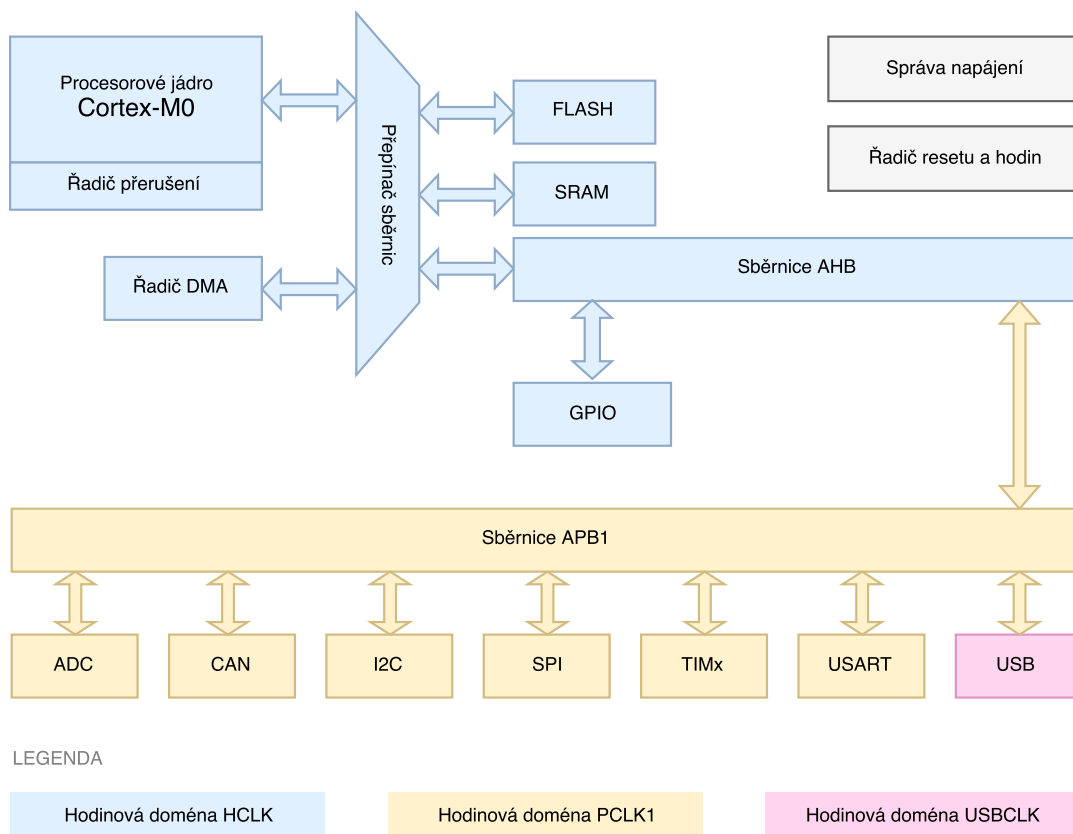
Obrázek 3.22: Časový průběh měření poměru dvou frekvencí,  $n_B = 5$



# Kapitola 4

## Realizace virtuálního přístroje

### 4.1 Představení platformy STM32



**Obrázek 4.1:** Blokové schéma nejdůležitějších částí mikrokontroléru STM32F042

STM32 od společnosti STMicroelectronics je rodina mikrokontrolérů založená na jádrech ARM Cortex-M. Rodina je podle cílového určení diferenciována do několika řad, které se liší použitou variantou jádra ARM Cortex i další výbavou. Funkce potřebné pro

realizaci požadovaného přístroje jsou dostupné i v základní řadě STM32F0, která je cenově nejdostupnější, a je proto hlavním objektem zájmu této práce. Na obr. 4.1 je značně zjednodušené schéma mikrokontroléru STM32F042, zaměřené na znázornění uspořádání sběrnic, čítačů a komunikačních periférií<sup>1</sup>.

Z výbavy mikrokontroléru jsou pro následující rozbor nejdůležitější periférie TIMx<sup>2</sup>. Periférie TIM realizuje univerzální synchronní čítač v několika variantách. V nejjednodušší variantě se jedná o jednoduchý čítač s předděličem a kladným směrem čítání. Maximalistická varianta umožňuje externí řízení čítání několika způsoby, čítání kladné, záporné i střídavé, zachycení hodnoty čítače při externím impulzu, generování výstupního signálu a různé způsoby vzájemného propojení čítačů.

Mikrokontroléry řady STM32 obsahují několik čítačů TIM v různých variantách s šířkou registru 16 bitů a typicky 0 až 2 čítače s 32-bitovým registrem. 32-bitový registr umožňuje dosáhnout výrazně vyššího rozlišení nebo rozsahu čítání bez nutnosti periférie řetězit. V procesoru F042 jsou nejpokročilejšími čítači TIM2 (32-bit) a TIM3 (16-bit), jejichž společné blokové schéma je na obr. 4.2.

Některými z dalších kritérií volby mikrokontroléru pro realizaci přístroje jsou:

- periférie pro obsluhu komunikačního rozhraní USB
- řadič přímého přístupu do paměti (DMA)
- USB bootloader (možnost nahrát program přes USB bez dalšího příslušenství)
- dostupnost v pouzdrech TSSOP20 a QFP32 (možnost použití s adaptérem v kontaktním poli)
- dostupnost vývojových kitů

Nakonec byly pro vývoj zvoleny mikrokontroléry F042, F303, a F411. Při vysvětlování některých principů se zaměříme na nejjednodušší F042, protože prostředky dostupné v tomto procesoru jsou dostupné i ve vyšších řadách, což opačně nutně neplatí.

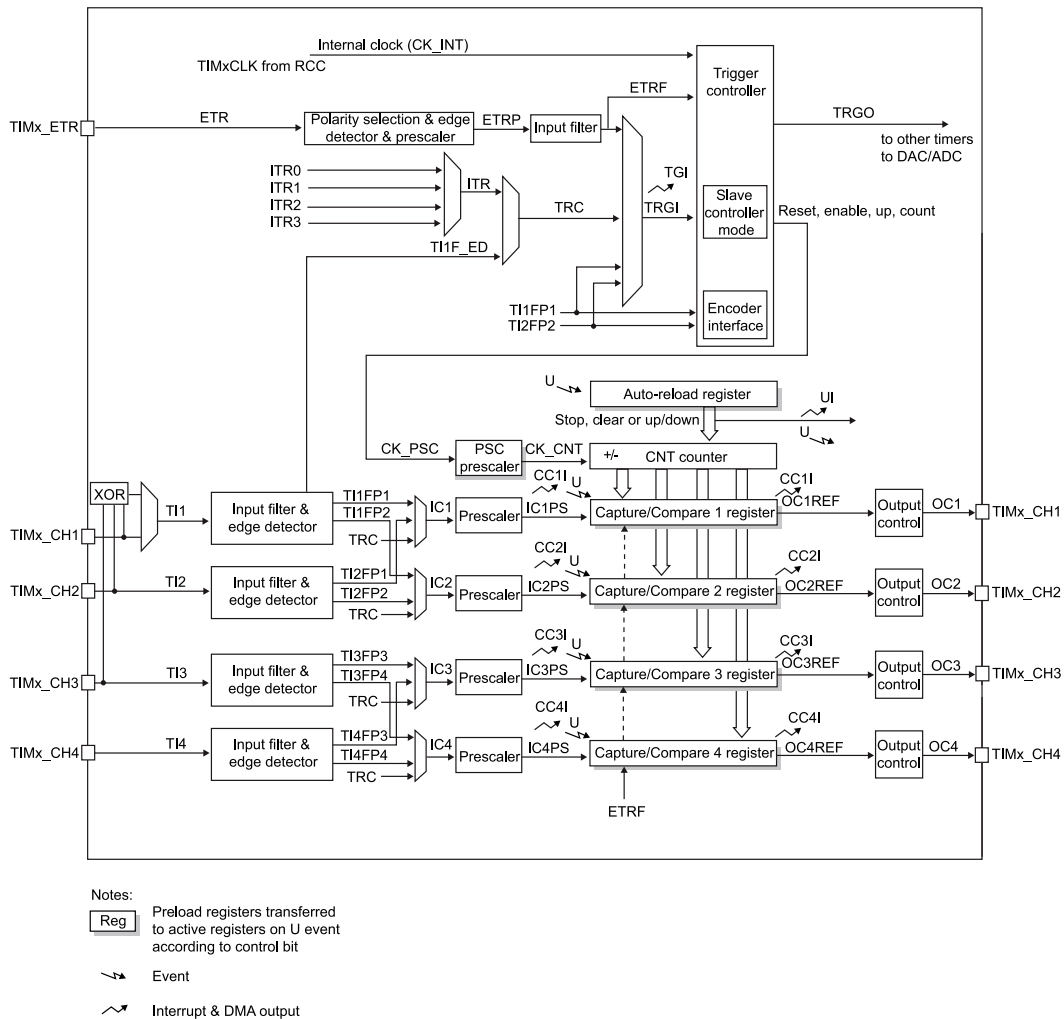
Jelikož se nezabýváme analogovým zpracováním vstupního signálu, využijeme přímo digitální vstupy mikrokontroléru. Elektrické parametry měřícího přístroje, jako např. rozhodovací napěťové úrovně, tak budou vycházet ze specifikace použitého čipu. V tab. 4.1 jsou uvedeny nejdůležitější parametry mikrokontroléru F042.

Parametr	Minimum	Maximum	Jednotka
Přípustné napětí na pinech struktury TTa	-0,3	4	V
Přípustné napětí na pinech struktury FT	-0,3	7,3	V
Napětí log. 0		1	V
Napětí log. 1	1,87		V
Hodinová frekvence		48	MHz

**Tabulka 4.1:** Vybrané elektrické parametry mikrokontroléru STM32F042 při napájecím napětí 3.3 V ([4])

<sup>1</sup>Podrobnější schéma lze nalézt v [4].

<sup>2</sup>Označením TIMx myslíme množinu periférií TIM1, TIM2, TIM3 atd.

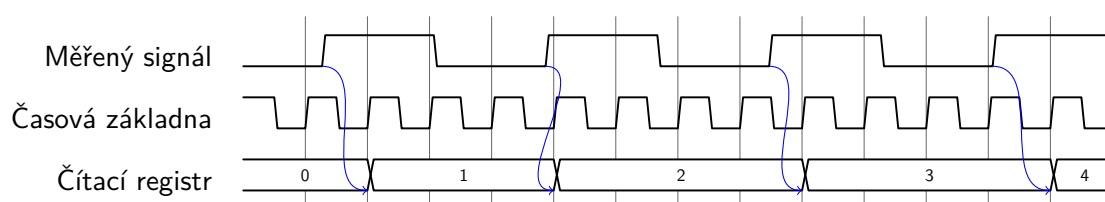


Obrázek 4.2: Schéma periferie TIM2/TIM3 v mikrokontroléru STM32F042 (z [5])

## 4.2 Synchronní čítač

Všechny čítače, které jsou v mikrokontroléru k dispozici, jsou synchronního typu. Synchronní čítač (obr. 4.3) se vyznačuje tím, že kromě čítaného signálu vyžaduje také signál hodinový. Hodinový signál o frekvenci  $f_{timebase}$  a periodě  $T_{timebase}$  tvoří časovou základnu čítače. Vstupní signál je vzorkován synchronně s hodinovým a detekuje se změna logické úrovně vstupního signálu mezi dvěma hodinovými cykly. Z toho vyplývají určitá omezení, která nyní rozebereme.

Aby byla spolehlivě zaznamenána každá změna logické úrovně, musí být doba trvání vysoké i nízké úrovně vstupního signálu delší než perioda časové základny přístroje. Tento vztah vyjadřuje dvojice nerovnic



**Obrázek 4.3:** Činnost synchronního čítače. Vertikální čáry znázorňují, v jakých okamžicích je vstupní signál vzorkován.

$$\begin{aligned} t_{high} &\geq T_{timebase} \\ t_{low} &\geq T_{timebase} \end{aligned} \quad (4.1)$$

Protože nás zajímá měření periodických signálů, budeme zkoumat, co tato podmínka říká o frekvenci takového signálu. Předpokládejme, že měřený signál má periodu  $T_{signal}$ , frekvenci  $f_{signal}$  a střihu 50 %. Pak lze odvodit:

$$\begin{aligned} T_{signal} &= t_{high} + t_{low} \\ t_{high} &= t_{low} \\ t_{high} &= \frac{T_{signal}}{2} \\ \frac{T_{signal}}{2} &\geq T_{timebase} \\ T_{signal} &= \frac{1}{f_{signal}} \\ T_{timebase} &= \frac{1}{f_{timebase}} \\ \frac{f_{timebase}}{2} &\geq f_{signal} \end{aligned} \quad (4.2)$$

Pro názornost tuto podmínku přepíšeme jako:

$$f_{signal} \leq \frac{1}{2} f_{timebase} \quad (4.3)$$

Pokud bude podmínka (4.3) porušena, hrozí, že impuls nebude detekován a dojde k jevu zvanému *aliasing*. Tento efekt je obecně nežádoucí, je však možné vypočítat jeho důsledky, a v určitých případech je i kompenzovat.

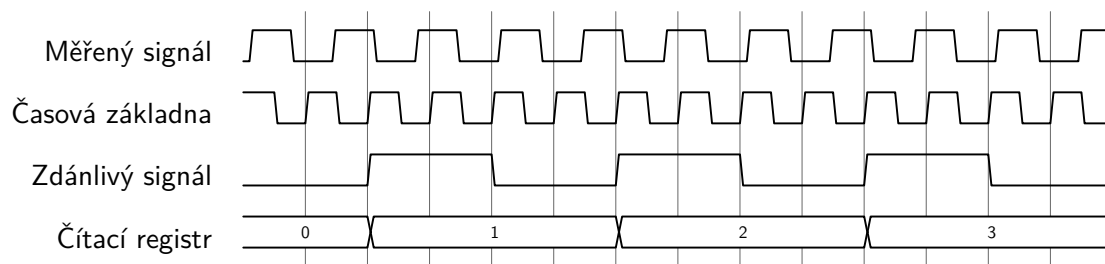
#### 4.2.1 Vznik aliasingu

Na obr. 4.4 do přístroje vstupuje signál, jehož frekvence je  $0,75 f_{timebase}$ ; jeho perioda je tedy 1,33krát delší než perioda časové základny.

Měřený signál je navzorkován při hraně časové základny (předpokládáme náběžnou), a tato hodnota se drží až do dalšího cyklu. Takto vzniká zdánlivý signál.

Přístroj změří zdánlivou frekvenci  $f_Z$ , která pro  $f_{signal} \in \left(\frac{1}{2} f_{timebase}; f_{timebase}\right)$  odpovídá vztahu

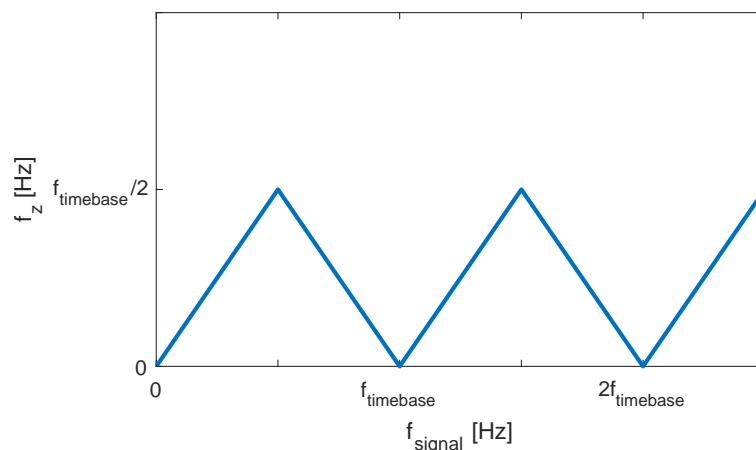
$$f_Z = f_{timebase} - f_{signal} \quad (4.4)$$



Obrázek 4.4: Aliasing při synchronním čítání

Z obrázku je skutečně vidět, že na 4 periody časové základny připadá 1 inkrementace čítacího registru. Zdánlivá frekvence je tedy  $0,25 f_{timebase}$ . Pokud bychom měřenou frekvenci  $f_{signal}$  dále navýšili až na hodnotu  $f_{timebase}$ , bude zdánlivá frekvence 0 Hz. Intuitivně si to lze představit tak, že přístroj na vstupu „uvidí“ v každém vzorkovacím cyklu stejnou hodnotu.

Pokud frekvence signálu bude stoupat dále, průběh zdánlivé frekvence bude periodický, jak znázorňuje obr. 4.5.



Obrázek 4.5: Graf závislosti zdánlivé frekvence na skutečné frekvenci signálu při synchronním čítání

## 4.2.2 Detekce aliasingu

Ze samotné naměřené frekvence nelze vznik aliasingu poznat, klíč k detekci je však ukrytý v rovnici 4.4.

Předpokládejme signál s neznámou frekvencí  $f_{signal}$ , o které víme, že  $f_{signal} < f_{timebase}$ . Budeme chtít určit, ve kterém z intervalů

$$\left\langle 0 \text{ Hz}; \frac{1}{2} f_{timebase} \right\rangle$$

a

$$\left\langle \frac{1}{2} f_{timebase}; f_{timebase} \right\rangle$$

neznámá frekvence leží.

Pokud jsme schopni za běhu přístroje měnit frekvenci časové základny  $f_{timebase}$ , například z hodnoty  $f_{t1}$  na  $f_{t2}$ ,

$$f_{t2} = 1,1f_{t1} \quad (4.5)$$

můžeme provést 2 měření s různou časovou základnou. Naměříme 2 různé zdánlivé frekvence  $f_{Z1}$  a  $f_{Z2}$ . Pokud je měřený signál v rozsahu  $\langle 0 \text{ Hz}; \frac{1}{2}f_{t1} \rangle$ , kde k aliasingu nedochází, bude platit

$$f_{Z1} = f_{Z2} = f_{signal} \quad (4.6)$$

Pokud však  $f_{signal}$  leží v  $(\frac{1}{2}f_{t2}; f_{t1})$ , bude platit

$$f_{Z1} = f_{t1} - f_{signal} \quad (4.7)$$

$$f_{Z2} = f_{t2} - f_{signal} = 1,1f_{t1} - f_{signal} \quad (4.8)$$

$$f_{Z2} = f_{Z1} + 0,1f_{t1}$$

Tento test lze rozšířit i na vyšší obrazy frekvence. Omezujícím faktorem bude šířka pásma analogového vstupu přístroje. S vyššími obrazy se také násobí nejistota měření, což ukážeme na dalším příkladu. Např. pro  $f_{signal} = 2,25f_{timebase}$  by zdánlivá frekvence byla určena jako

$$f_Z = f_{signal} - 2f_{timebase} \quad (4.9)$$

ovšem frekvence  $f_{timebase}$  je oproti nominální hodnotě  $f_{nominal}$ , se kterou přístroj počítá, zatížena nějakou chybou, tedy

$$f_{timebase} = f_{nominal} + f_{error} \quad (4.10)$$

Po dosazení:

$$f_Z = f_{signal} - 2f_{nominal} - 2f_{error} \quad (4.11)$$

Je tedy vidět, že odchylka časové základny od nominální hodnoty se projeví výrazněji.

### 4.3 Časová základna přístroje

Přesnost časové základny je klíčová. Při přímém měření frekvence se z časové základny odvozuje čas  $T_{gate}$ , který vystupuje v rovnici 3.1, při měření periody a intervalu zase pracujeme přímo s hodnotou  $T_{timebase}$ . Kvalita časové základny tedy určuje přesnost všech těchto měření.

V mikrokontroléru je časová základna odvozena z interního hodinového signálu. Ten může být použit přímo, nebo zpracován děličkou či fázovým závěsem. Mikrokontrolér může využívat několik různých zdrojů hodinového signálu, a dokonce mezi nimi přepínat za běhu. Je tedy možné po spuštění přístroje vybrat zdroj s největší přesností.

Frekvenci interního hodinového signálu volíme na základě několika kritérií. V první řadě je nutné držet se v rozsahu povoleném výrobcem. U moderních procesorů používajících statickou paměť je typicky stanoven pouze horní limit pro frekvenci, který může být navíc závislý na napájecím napětí.



Výrobce	Produkt	Oscilátor	Teplota [°C]	Napětí [V]	Frekvence
Microchip	ATmega328P	Internal RC	25	3	8 MHz ± 10 %
Microchip	PIC18F4550	INTOSC	25	neuvedeno	8 MHz ± 2 %
Microchip	SAM D21	OSC8M	25	3,3	8 MHz ± 0,75 %
Microchip	SAM D21	OSC8M	25	1,62 – 3,63	8 MHz ± 1 %
Microchip	SAM D21	OSC8M	-40 – 85	1,62 – 3,63	8 MHz ± 2 %
STM	STM32F042	HSI	25	neuvedeno	8 MHz ± 1 %
STM	STM32F042	HSI48	25	neuvedeno	48 MHz ± 3 %

**Tabulka 4.2:** Tovární kalibrace interních oscilátorů vybraných mikrokontrolérů (hodnoty uváděné výrobcí). Zdroje: [6, 7, 8, 4]

Dalším faktorem jsou požadavky periferií, a to především řadičů rychlých nebo asynchronních sběrnic. Periferie pro obsluhu sběrnice USB například vyžaduje hodinový signál o frekvenci 48 MHz. Asynchronní rozhraní UART a sběrnice CAN zase omezují povolenou odchylku frekvence od nominální hodnoty, aby byla zaručena správná funkčnost.

Konečně, zvolená systémová frekvence je kompromisem mezi dosažitelným výpočetním výkonem a energetickou spotřebou procesoru. V případě, že spotřeba procesoru není rozhodující, tedy pokud je zařízení napájeno z elektrické sítě, nebo se procesor na celkové spotřebě zařízení podílí zanedbatelně (např. v aplikacích elektromobility), a zároveň vyšší odběr proudu neklade zvláštní požadavky na napájení a chlazení, použijeme nejvyšší přípustnou frekvenci.

Pokud frekvence vstupního hodinového signálu není v požadovaném rozsahu, je možné využít další možnosti předzpracování, popsané níže.

### ■ 4.3.1 Generování časové základny

#### ■ Integrovaný RC oscilátor

Mikrokontrolér ve výchozím nastavení pro generování hodinového signálu využívá RC oscilátor integrovaný na čipu. Nejsou tedy potřeba žádné externí součástky nebo složité nastavování.

RC oscilátor je kalibrován při výrobě na pokojovou teplotu a určité napájecí napětí. Přesnost tovární kalibrace vybraných procesorů je uvedena v tabulce 4.2.

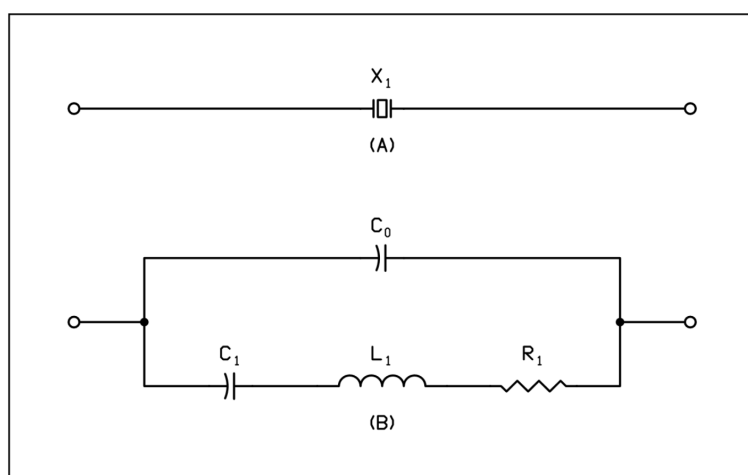
Uživatelská kalibrace umožňuje dosáhnout přesnosti okolo 1 % na celém teplotním rozsahu. Pro přesné měření časových veličin je taková tolerance stále příliš velká.

#### ■ Krystalové oscilátory

V realizaci krystalového oscilátoru se využívá mechanická rezonance krystalu z piezoelektrického materiálu. Na obr. 4.6 je znázorněn krystal a jeho ekvivalentní obvod. Aby

krystal osciloval, musí být zapojen do budicího obvodu. Mikrokontrolér má tento obvod již zabudovaný, takže kromě samotného krystalu je třeba připojit jen 2 stabilizační kondenzátory.<sup>3</sup>

Pro nízké frekvence se využívá základní oscilační frekvence krystalu. S rostoucí frekvencí se zmenšuje velikost řezu křemene a od cca 30 MHz už by byl takový krystal nepraktický (křehký křemen a tedy nízká odolnost součástky). Úpravou budicího LC obvodu je však možné krystal provozovat na vyšší harmonické frekvenci (používají se pouze liché násobky, tedy 3, 5, 7). Takto lze dosáhnout až frekvencí okolo 200 MHz. Ve většině aplikací je však výhodnější použít nižší frekvenci krystalu a frekvenci hodinového signálu zvýšit pomocí fázového závěsu.



**Obrázek 4.6:** Krystal (A) a jeho náhradní schéma při fundamentální frekvenci (B). Převzato z [10].

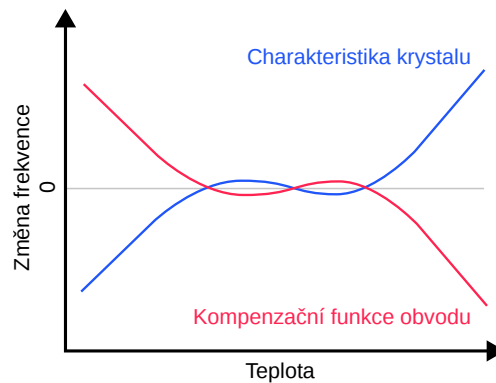
Běžně dostupné krystaly mají přesnost 20, 30 nebo 50 ppm. Výrobce často udává dvě hodnoty tolerance, jednu pro  $t = 25^{\circ}\text{C}$  a druhou pro celý teplotní rozsah.

Ačkoli je přesnost řádově lepší než u RC oscilátoru, vykazuje běžný krystal teplotní závislost, která brání dosažení menší odchylky. Vzhledem k tomu, že tato závislost je poměrně přesně definovaná a odhadnutelná, je možné ji kompenzovat sítí kapacitorů, rezistorů a termistorů, které v určitém rozsahu negují teplotní charakteristiku krystalu (obr. 4.7).

Místo samotného krystalu je také možné použít zapouzďřený krystalový oscilátor. Ten kromě krystalu obsahuje i budicí obvod, takže se chová jako zdroj obdélníkového signálu o požadované frekvenci s nízkým vnitřním odporem. Nejčastěji se můžeme setkat se 3 typy krystalových oscilátorů:

- SPXO (Simple Packaged Crystal Oscillator)
- TCXO (Temperature-Compensated Crystal Oscillator)
- VCXO (Voltage Controlled Crystal Oscillator)

<sup>3</sup>Popis vnitřního zapojení mikrokontroléru pro buzení krystalu je nad rámec této práce. Zevrubný rozbor lze nalézt například v [9].



**Obrázek 4.7:** Teplotní charakteristika krystalu a obvodová kompenzace. Převzato z [11].

Teplotně kompenzované krystalové oscilátory (TXCO) dosahují vyšší přesnosti přibližně o jeden řád, tedy například 3 ppm na rozsahu  $-40$  až  $85$  °C [11].

### ■ 4.3.2 Předpracování časové základny děličkou a fázovým závěsem

Frekvence časové základny musí být taková, aby bylo možné pomocí integrované předděličky a fázového závěsu dosáhnout požadované interní frekvence. V principu je možné frekvenci násobit a dělit kladnými celými čísly, dosažitelné frekvence tedy určuje vztah

$$f = \frac{M}{N} f_{in} \quad (4.12)$$

Přípustný rozsah parametrů  $M$  a  $N$  závisí na použitém mikrokontroléru.

### ■ 4.3.3 Možnosti zpřesnění časové základny

#### ■ Kalibrace

Pokud je odchylka časové základny neměnná, je možné odchylku určit s pomocí nějakého referenčního signálu o známé frekvenci, a následná měření korigovat.

Tento způsob kalibrace demonstrujeme na příkladu s použitím signálu Pulse Per Second (PPS), který je vyveden z některých GPS přijímačů.

**Příklad: kalibrace PPS signálem z GPS přijímače.** U PPS signálu předpokládáme dostatečnou přesnost ( $<1$  ppm)<sup>4</sup>. Čítačem změříme periodu signálu (víme, že  $T = 1$  s) jako  $N$  hodinových taktů. Oscilátor má nominální frekvenci  $f_0$  a skutečnou frekvenci  $f$ , kterou určíme

$$f = \frac{N}{T} \quad (4.13)$$

<sup>4</sup>Tato přesnost je garantována pouze dlouhodobě. Z krátkodobého hlediska bude signál vykazovat určitý fázový šum (jitter). Míra tohoto šumu závisí na kvalitě použitého přijímače. Podrobnosti lze nalézt např. v [12].

Bude-li přístroj předpokládat, že  $f = f_0$ , určí nepřesnou periodu signálu

$$T_{meas} = \frac{N}{f_0} \quad (4.14)$$

Pokud tedy známe skutečnou periodu signálu a nominální frekvenci oscilátoru, skutečnou frekvenci oscilátoru dopočítáme jako

$$f = \frac{T_{meas} f_0}{T} \quad (4.15)$$

Relativní odchylka frekvence oscilátoru od nominální hodnoty je

$$\Delta f = \frac{f - f_0}{f_0} = \frac{T_{meas} f_0 - T f_0}{T f_0} = \frac{T_{meas}}{T} - 1 \quad (4.16)$$

### ■ Průběžné doladování

Mikrokontrolér STM32F042 disponuje funkcí Clock Recovery System (CRS), která umožňuje doladovat integrovaný oscilátor HSI48 podle časování signálů na sběrnici USB. Předpokládá se, že hostitelský počítač, ke kterému je přístroj připojen, má poměrně přesnou časovou základnu, kterou využívá pro časování USB<sup>5</sup> a časování na této sběrnici lze tedy považovat za určitou referenci. Regulační smyčka v mikrokontroléru pak frekvenci RC oscilátoru kontinuálně upravuje tak, aby se co nejvíce přiblížila příslušnému násobku reference.

### ■ 4.3.4 Použité řešení

Pro realizaci časové základny tedy připadá v úvahu použití jednoho z interních RC oscilátorů nebo externí krystal. Nakonec jsme přistoupili k hybridní variantě, která kombinuje výhody obou možností.

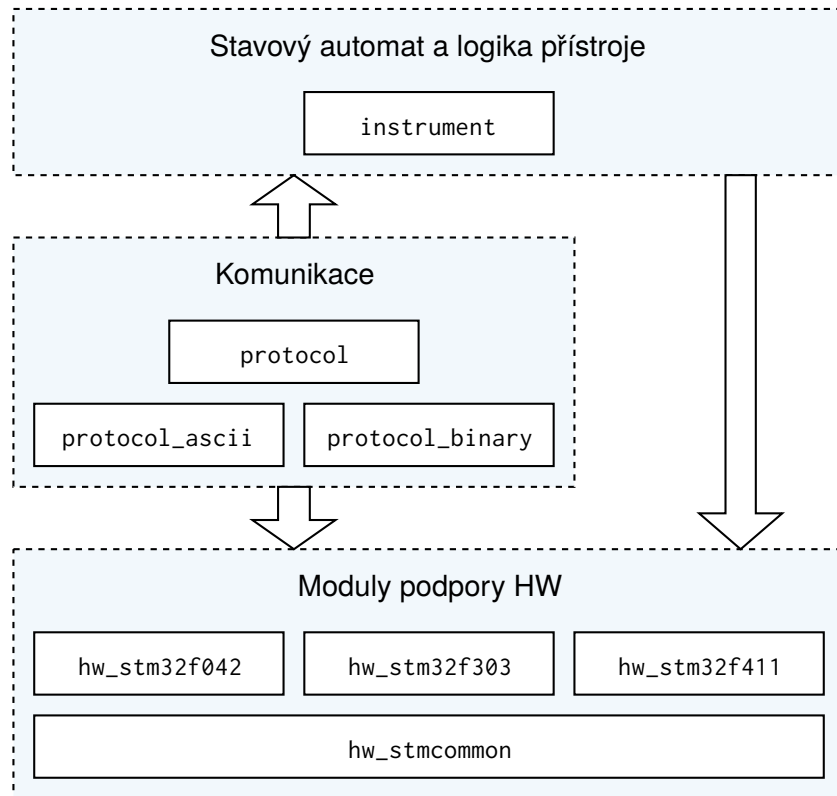
Řídící program přístroje při startu detekuje připojení externího krystalu a pokud je krystal připojen, nastaví se jako zdroj vnitřního hodinového signálu. Předpokládá se nominální frekvence 8 MHz. Pokud krystal nalezen není, použije se interní oscilátor HSI48 a aktivuje se funkce CRS.

Díky této dynamické detekci není třeba generovat více variant firmwaru pro různé konfigurace. Zároveň je v minimalistické variantě přístroje omezen počet nutných externích součástí.

<sup>5</sup>Tento předpoklad je opodstatněný, protože specifikace USB 2.0 připouští odchylku 0,5 ‰ neboli 500 ppm od nominální frekvence. V počítači tedy musí být nějaký dostatečně přesný oscilátor, který je prakticky vždy realizován krystalem.

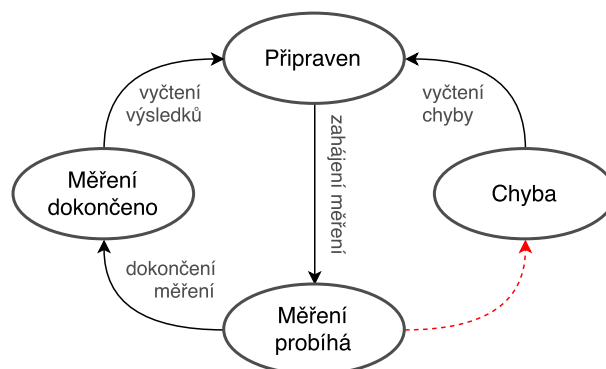
## 4.4 Architektura přístroje

Firmware přístroje je strukturován do několika modulů, které jsou znázorněné na obr. 4.8. Podobné členění používá i práce [13].



**Obrázek 4.8:** Organizace modulů firmware přístroje. Šipky znázorňují směr volání funkcí.

Logika přístroje se řídí jednoduchým stavovým automatem (obr. 4.9). Zahájení měření a vyčtení výsledku nebo chybového kódu je iniciováno komunikačním modulem po přijetí příslušného požadavku.



**Obrázek 4.9:** Stavový automat přístroje

Komunikační modul využívá dva protokoly, které je možné přepínat za běhu. Po startu přístroj očekává příkazy v režimu ASCII, což umožňuje ovládání přístroje kterýmkoli sériovým terminálem. Po přijetí speciálního řídicího znaku dojde k přepnutí na efektivnější binární protokol, který využívá grafická aplikace.

Veškerý přístup k perifériím mikrokontroléru se odehrává v modulech podpory HW. Díky tomu je zbytek programu odstíněn od rozdílů mezi různými platformami. Moduly podpory HW se starají o správnou konfiguraci periférií pro realizaci přístrojových funkcí a jejich spouštění a zastavování. Pro minimalizaci duplikace kódu existuje pomocný modul, společný pro rodinu STM32. Ten je parametrizovaný hlavičkovým souborem, který přiřazuje různým přístrojovým funkcím dostupné periferie. Následující úryvek kódu ilustruje konkrétní parametrizaci pro mikrokontrolér F042:

```
#define INPUT_CAPTURE_TIMER      TIM2
#define INPUT_CAPTURE_HTIM      htim2

#define INPUT_CAPTURE_DMA        hdma_tim2_ch2
#define INPUT_CAPTURE_DMA_TC_FLAG DMA_FLAG_TC3

#define INPUT_CAPTURE_RISING_CHAN TIM_CHANNEL_2
#define INPUT_CAPTURE_RISING_CCR (TIM2->CCR2)
#define INPUT_CAPTURE_RISING_CCIF TIM_SR_CC2IF
#define INPUT_CAPTURE_FALLING_CHAN TIM_CHANNEL_3
#define INPUT_CAPTURE_FALLING_CCR (TIM2->CCR3)
```

## 4.5 Realizace přístrojových funkcí

### 4.5.1 Přímé měření frekvence

Pro realizaci přímého měření potřebujeme čítací registr a čítač hradlování. V mikrokontroléru máme k dispozici 0 až 2 32-bitové čítače a větší počet čítačů 16-bitových.

Pokud čítací registr realizujeme 16-bitovým čítačem, maximální dosažitelné (relativní) rozlišení je pouze

$$\frac{\Delta f_{signal}}{f_{signal}} = \frac{1}{2^{16}} = \frac{1}{65536} \approx 1,5 \cdot 10^{-5} \quad (4.17)$$

Při době hradlování 1 sekunda by navíc maximální měřitelná frekvence byla pouze 65535 Hz. Uživatel přístroje by tedy musel velmi opatrně volit dobu hradlování podle frekvence signálu. Protože tato omezení nejsme ochotni připustit, máme dvě možná východiska. Je-li k dispozici 32-bitový čítač, můžeme využít ten. V opačném případě musíme použít 2 16-bitové čítače v kaskádovém zapojení.

Na čítač hradlování takové požadavky nemáme. Doba hradlování je daná vzorcem

$$T_{gate} = \frac{(PSC + 1)(ARR + 1)}{f_{CLK}} \quad (4.18)$$

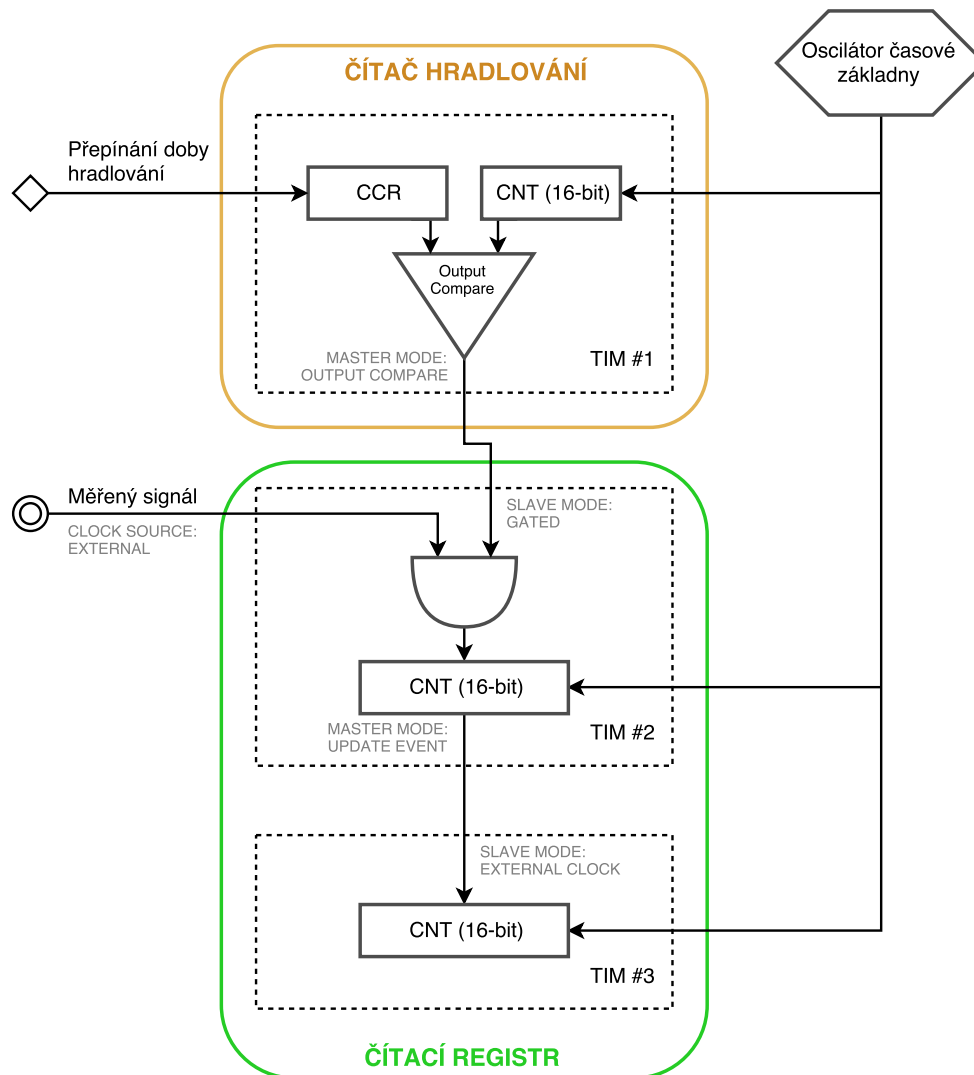
kde  $PSC$  a  $ARR$  jsou celočíselné registry o stejné bitové šířce jako čítačí registr. Frekvence hodin čítače  $f_{CLK}$  je typicky celočíselným násobkem 1 MHz. Např. pro  $f_{CLK} = 144$  MHz a  $T_{gate} = 1$  s můžeme nastavit:

$$PSC = 36000 - 1$$

$$ARR = 4000 - 1$$

Tím nastavíme dobu  $T_{gate}$  přesně.

Nyní tedy zbývá jen určit potřebnou konfiguraci čítačích periférií. Ta je znázorněna na obr. 4.10.



**Obrázek 4.10:** Zapojení periférií mikrokontroléru pro přímé měření frekvence čítáním

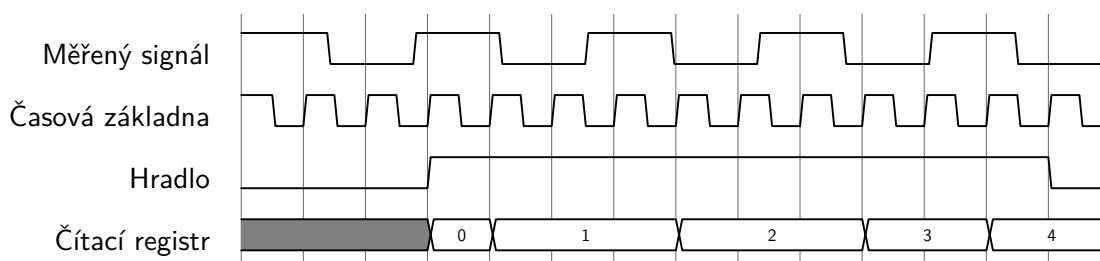
Čítač #1 v režimu *Master Mode: Output Compare* generuje hradlovací pulz.

Do čítače #2 vstupují 3 řídicí signály:

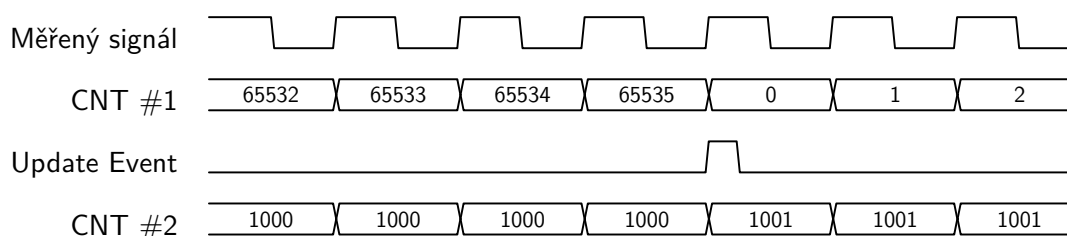
1. interní hodinový signál o frekvenci  $f_{CLK}$ , který určuje vzorkovací frekvenci synchronního čítače
2. měřený signál, který je přiveden pinem mikrokontroléru do čítače v režimu *Clock Source: External Trigger Mode 2*
3. hradlovací signál, přivedený z čítače #1 v režimu *Slave Mode: Gated*

Čítač #2 také využívá výstup Master Mode, a to v módu *Update Event*. Tím je zajištěno, že přetečení čítače #2 vyšle impuls do čítače #3, který v důsledku toho inkrementuje svůj registr *CNT*. Dvojici kaskádních čítačů #2 a #3 lze nahradit jedním 32-bitovým čítačem, je-li k dispozici.

Všechny čítače pracují synchronně, takže změny v měřeném signálu se vzorkují s náběžnou hranou časové základny. Na obr. 4.11 je znázorněn průběh jednoho měření. Obr. 4.12 pak ilustruje řetězení registrů a zaznamenání přetečení.



**Obrázek 4.11:** Časový průběh měření frekvence čítáním



**Obrázek 4.12:** Řetězení čítačů

Při použití kaskádovaných čítačů je třeba dbát zvláštní pozornosti při vyčítání registrů *CNT*. Pokud čítání stále probíhá, hrozí, že právě mezi přečtením obou registrů dojde k přetečení nižšího (master) čítače. Vyčtená hodnota by pak byla nesmyslná.

Možným řešením je dočasně vypnout zpracování přerušování a dvakrát ihned po sobě přečíst hodnotu čítacího registru v pořadí nejdříve nižší čítač a následně vyšší čítač. Budou vyčteny dvě hodnoty  $C_1$ ,  $C_2$ . Pokud časování vyjde tak, že uprostřed čtení dolní čítač přeteče, vztah mezi hodnotami bude  $C_1 < C_2$ , což je zjevně nesmysl<sup>6</sup>. Program musí tuto situaci rozpoznat a vyčtení opakovat.

<sup>6</sup>Může se také jednat o důsledek přetečení vyššího čítače. Tuto událost lze detekovat příznakem ve stavovém registru periferie.

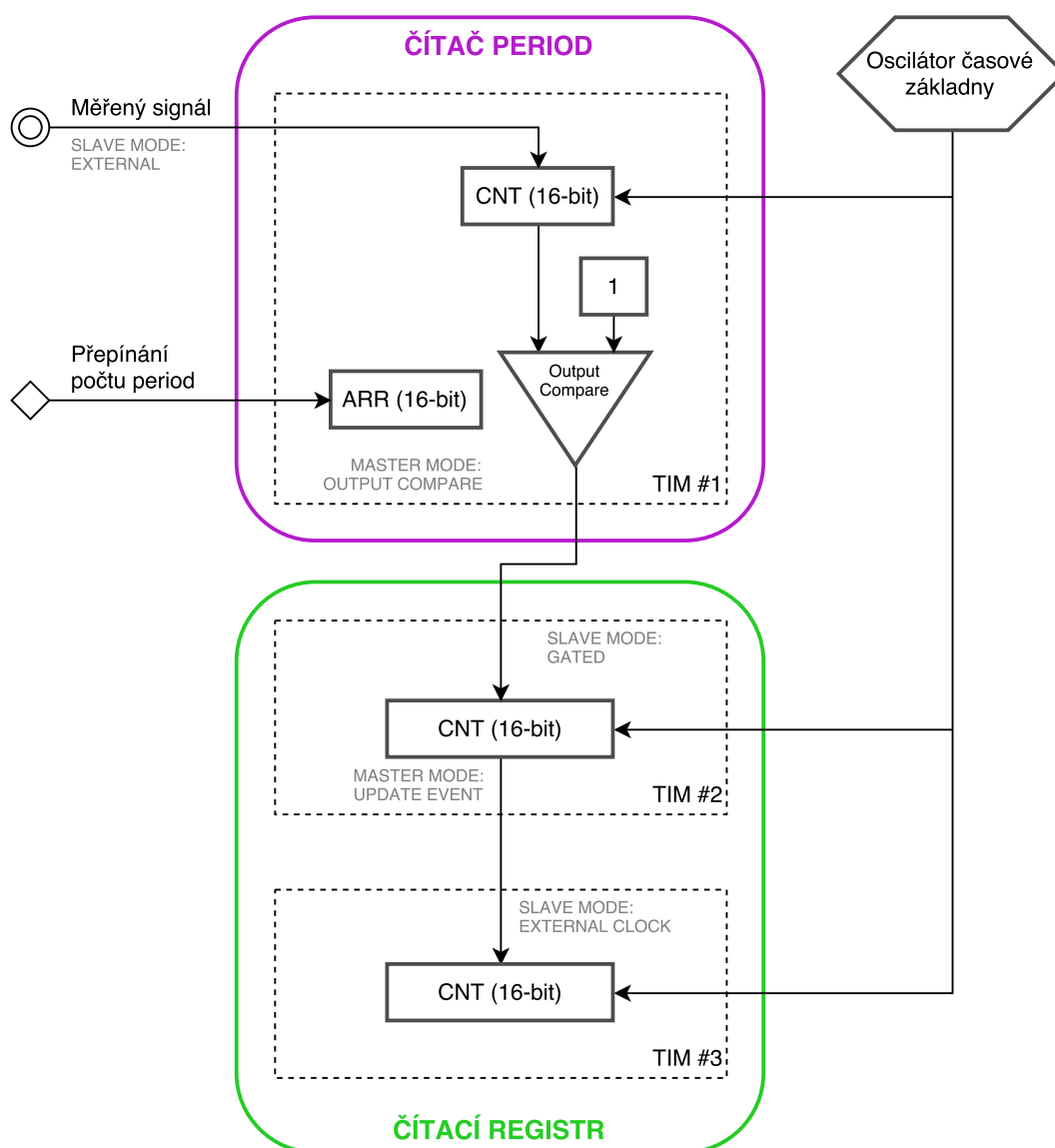


## 4.5.2 Měření periody

Pro měření periody budeme opět potřebovat nějakou realizaci čítacího registru, a dále hradlo řízené hranami, nebo jeho funkční ekvivalent. V úvahu připadá několik různých způsobů konfigurace, podle dostupných periférií a požadavcích na měření.

### Měření N period dvěma 16-bitovými čítači

Pokud je cílem co nejpřesnější změření periody a nepotřebujeme znát střihu signálu, vhodnou konfigurací bude ta z obr. 4.13.



**Obrázek 4.13:** Konfigurace pro měření N period. Output Compare jednotka v čítači #1 pracuje v režimu PWM 2.

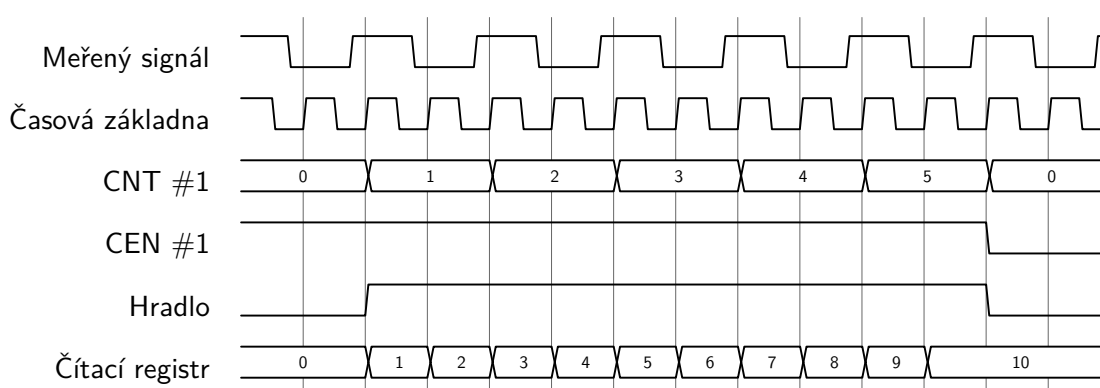
Čítač #1 hradluje čítání pulzů časové základny v čítači #2. Kaskádní řešení čítacího

registru a z toho plynoucí důsledky byly diskutovány v části 4.5.1.

Pro správné ohradlování právě  $n_p$  celých period se využije funkce *One-Pulse Mode*. V tomto módu bude hradlo otevřené, právě když je splněna nerovnost

$$CCR \leq CNT \leq ARR \quad (4.19)$$

Dolní nerovnost je nutná k tomu, aby se hradlo otevřelo až po zaznamenání první hrany signálu. V opačném případě by čítání bylo zahájeno v neurčené části „nulté“ periody a vznikla by velká chyba měření. Registr *CCR* se tedy nastaví napevno na hodnotu 1 a registr *ARR* se nastaví na hodnotu  $n_p$ . Funkce *One-Pulse Mode* zajistí, že když hodnota registru *CNT* přeteče *ARR*, bude vynulován příznak *CEN*, čítač #1 bude zastaven a hradlo se uzavře. Po dokončení měření pak čítačí registr bude obsahovat hodnotu odpovídající  $n_p \cdot T_{signal}$  (obr. 4.14).



**Obrázek 4.14:** Měření  $n_p = 5$  period.  $f_{signal} = \frac{1}{2}f_{timebase}$ ;  $CCR = 1$ ;  $ARR = 5$ . Měření bylo zahájeno uprostřed periody signálu, hradlo se však otevře až po zaznamenání náběžné hrany.

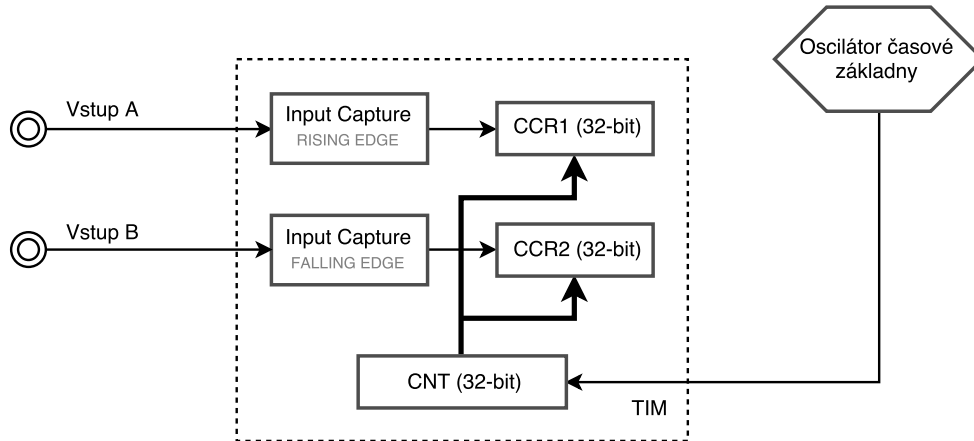
Čítač #1 může být nastaven jak v režimu *Slave Mode: External*, u kterého lze volit mezi více vstupními signály (TI1FP1, TI2FP2 na obr. 4.2), tak v režimu *Clock Source: External Trigger Mode 2*, kdy je vstupní signál přiveden z příslušného pinu TIMx\_ETR. Jelikož *Slave Mode* v čítači #1 není potřeba k jinému účelu, jediným kritériem volby je výhodnost použití odpovídajících pinů mikrokontroléru.

Zapojení lze dále obměnit například následujícími způsoby:

- nahrazení TIM #1 32-bitovým čítačem (možnost měřit větší počet navazujících period)
- nahrazení dvojice čítačů #2 a #3 jedním 32-bitovým čítačem

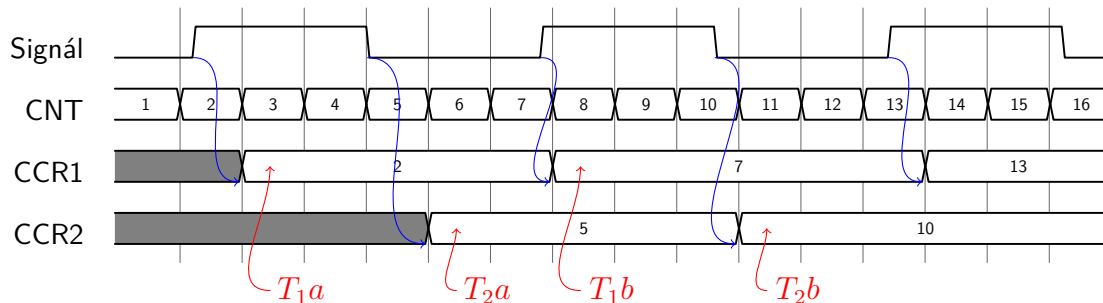
#### ■ Měření periody a šířky pulzu jedním 32-bitovým čítačem

Předchozí varianta neumožňuje určit šířku pulzu a střihu signálu. Jedním způsobem jak zaznamenat obě potřebné hrany je použití jednotek Input Capture pro zachycení délky kladného pulzu a délky celé periody (obr. 4.15).



**Obrázek 4.15:** Konfigurace pro měření periody (32-bit, IC)

Využívá se jedna periferie TIM se dvěma Capture/Compare jednotkami, jejichž vstupy musí být externě (mimo mikrokontrolér) spojeny. Čítač běží s frekvencí vnitřních hodin  $f_{CPU}$ . Náběžná hrana vstupu A způsobí uložení hodnoty čítače  $CNT$  do registru  $CCR1$ , sestupná hrana vstupu B způsobí uložení hodnoty do registru  $CCR2$ . Pro vyhodnocení periody signálu a šířky pulzu je nutné porovnat 2 po sobě následující záznamy registru  $CCR1$  ( $T_{1a}$ ,  $T_{1b}$ ), a jeden záznam registru  $CCR2$  ( $T_{2a}$ ), ležícího mezi nimi. V obr. 4.16 je pro úplnost vyznačena ještě hodnota  $T_{2b}$ , která přijde vhod později.



**Obrázek 4.16:** Využití Input Capture pro měření periody a šířky pulzu

Z těchto záznamů určíme periodu signálu  $T_{signal}$  a šířku pulzu  $T_{pulse}$ :

$$T_{signal} = T_{CLK} \cdot (T_{1b} - T_{1a}) \quad (4.20)$$

$$T_{pulse} = T_{CLK} \cdot (T_{2a} - T_{1a}) \quad (4.21)$$

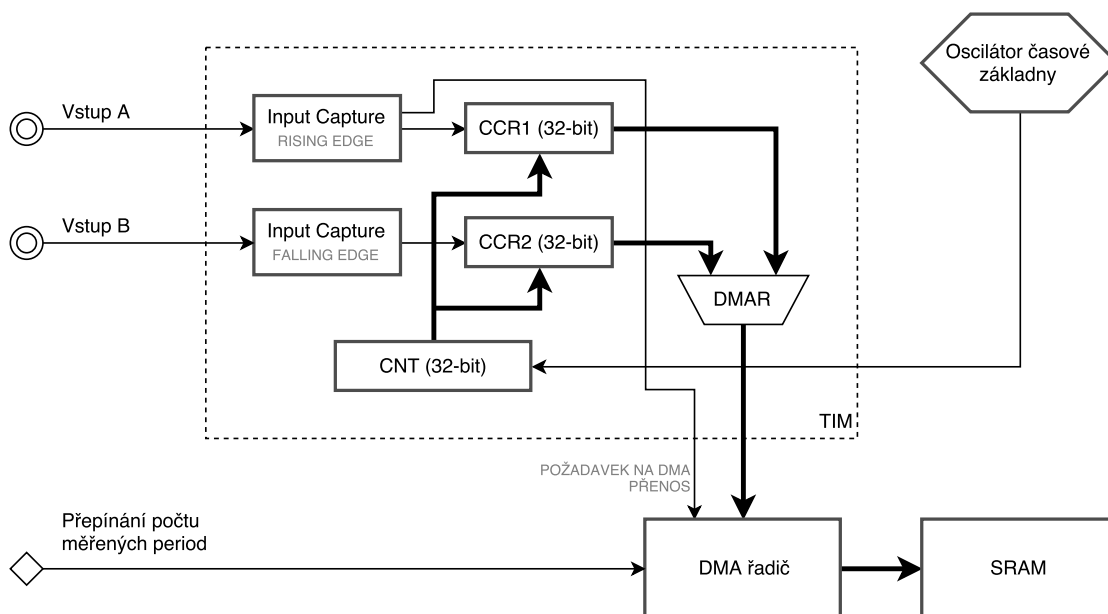
Zaznamenání hodnot  $T_{1a}$ ,  $T_{1b}$  a  $T_{2a}$  lze dosáhnout více způsoby.

Nabízí se možnost, že program bude ve smyčce nepřetžitě kontrolovat změny obsahu registrů, a ukládat je do paměti, dokud nebude zaznamenáno dostatek dat pro vyhodnocení. Tento přístup je nevhodný proto, že program v přístroji musí obsluhovat další funkce a periferie. Během měření by tedy například nebyla možná komunikace se zařízením. Při

použití sběrnice USB, kde je vyžadována reakce v přísném časovém limitu (a tato reakce vyžaduje součinnost programu), pak nepřipadá tato možnost v úvahu.

Pro efektivnější využití výpočetního času je možné použít rutinu přerušení, která bude vyvolána periferií TIM po záznamu do registru  $CCR1$ . Program tedy nemusí hlídat, kdy k záznamu došlo, protože je o zpracování automaticky požádán. Dosažitelná frekvence měřeného signálu je pak omezena latencí přerušení a trváním samotné přerušovací rutiny. Minimální latence přerušení procesoru Cortex-M0 je 16 hodinových cyklů, ke kterým se v nejlepším případě přičte několik desítek cyklů strávených jeho zpracováním. Nemáme tedy šanci přiblížit se teoretickému limitu  $f_{signal} = f_{CLK}/2$ .

Třetím způsobem je použití přímého přístupu do paměti (DMA), tato konfigurace je znázorněna na obr. 4.17. Po zaznamenání náběžné hrany je vyslán požadavek řadiči DMA, který vyčte data z dvojice registrů  $CCR1$ ,  $CCR2$  a uloží je do paměti SRAM. Periferie TIM má speciální podporu pro DMA, která umožňuje na vyhrazené paměťové adrese (registr  $DMAR$ ) střídatě zpřístupnit více registrů čítače. Řadič DMA je tedy nastaven tak, aby po každém požadavku z této adresy dvakrát načtel a uložil do paměti 32-bitovou hodnotu. Periferie TIM při prvním načtení vrátí hodnotu registru  $CCR1$  a při druhém načtení hodnotu registru  $CCR2$ . Nakonec řadič DMA inkrementuje vnitřní čítač, aby další přenosy použily pro zápis následující paměťové lokace. Počet odebraných vzorků se určí nastavením DMA a je omezen volným místem v paměti.



**Obrázek 4.17:** Konfigurace pro měření periody (32-bit, IC, DMA)

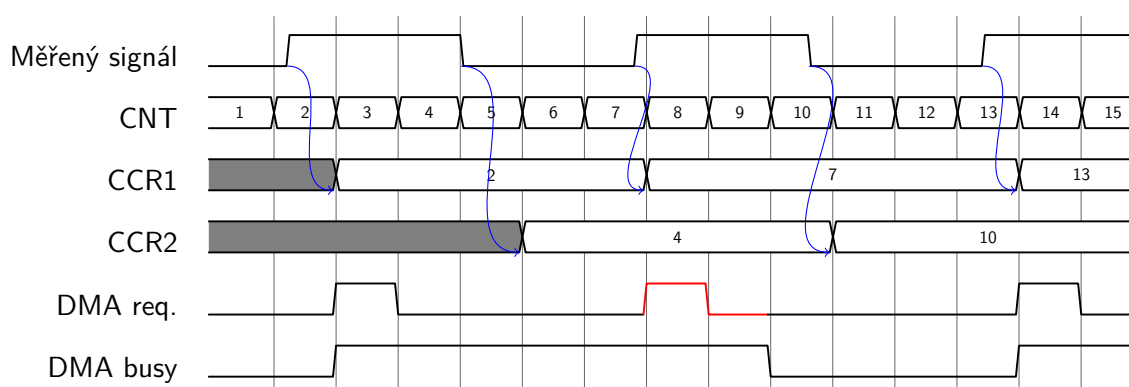
Použití DMA je ze 3 popsanych způsobů nejvýhodnější (nejvyšší maximální měřitelná frekvence signálu), a proto se nadále budeme věnovat pouze jemu.

Maximální měřitelná frekvence je omezena dobou zpracování DMA požadavku. Pokud bude zaznamenána další náběžná hrana dříve, než je dokončeno zpracování předchozí, data budou ztracena a vznikne hrubá chyba měření. To ilustruje obr 4.18.

Zabránit vzniku této chyby lze využitím další funkce periferie TIM. Tu je možné nastavit

Číslo DMA přenosu	Adresa čtení	Skutečný přečtený registr	Adresa zápisu
1	TIM:DMAR	TIM:CCR1	SRAM:buffer[0]
1	TIM:DMAR	TIM:CCR2	SRAM:buffer[1]
2	TIM:DMAR	TIM:CCR1	SRAM:buffer[2]
2	TIM:DMAR	TIM:CCR2	SRAM:buffer[3]
3	TIM:DMAR	TIM:CCR1	SRAM:buffer[4]
3	TIM:DMAR	TIM:CCR2	SRAM:buffer[5]
...	...	...	...

**Tabulka 4.3:** Přečtené a zapsané hodnoty během přímého přístupu do paměti



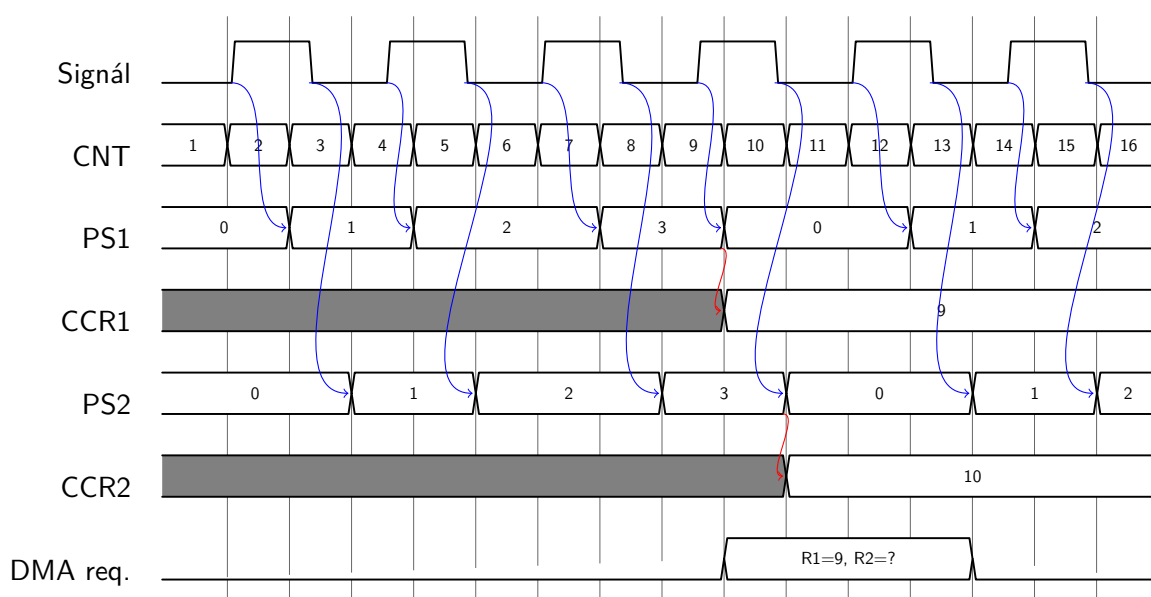
**Obrázek 4.18:** Ztracený DMA požadavek při vysoké frekvenci měřeného signálu. Červeně zvýrazněná část signálu představuje požadavek, který kvůli zaneprázdněnosti řadiče DMA nebyl zpracován.

tak, že po zkopírování registru *CNT* do *CCR1* při zaznamenání náběžné hrany se registr *CNT* automaticky vynuluje. Při zpracování vzorků pak není třeba odečítat předchozí hodnotu a výpadek jednoho nebo více vzorků kvalitu měření neovlivní.

Dosud popsané způsoby měří šířku pulzu se stejnou přesností jako periodu signálu. Pokud můžeme relaxovat požadavek na přesnost určení šířky pulzu, lze měření dále optimalizovat.

Rozlišení určení periody je dáno počtem měřených period, který je omezený místem v paměti. Pokud by bylo možné nároky na paměť redukovat, bylo by možné naměřit více period, a měření tak zpřesnit. Přesně tohoto lze docílit využitím předděličky, která je součástí Capture/Compare kanálu. Předděličkou lze zajistit, že bude registrován jen každý  $N$ -tý pulz, kde  $N$  nabývá hodnot 1, 2, 4, 8. Využití předděličky ilustruje obr. 4.19. Registry *PS1* a *PS2* představují interní čítače předděličky, které procesoru nejsou přístupné.

Z obrázku je vidět, proč je rozlišení šířky pulzu menší, než rozlišení periody. Zatímco perioda se změří jako  $4N_p$ , šířka pulsu se změří jako  $3N_p + N_w$  (kde nás zajímá právě  $N_w$ ).



**Obrázek 4.19:** Využití Input Capture pro měření periody s DMA a předdělením /4

Při prvním DMA požadavku navíc registr  $CCR2$  ještě nemusí obsahovat smysluplnou hodnotu (v případě, že přístroj dříve zaznamená první náběžnou než první sestupnou hranu).

Použití předděličky také není možné zkombinovat s automatickým nulováním čítače, protože signál pro nulování vychází z Capture/Compare kanálu ještě před děličkou (signály TI1FP1, TI2FP2 v obr. 4.2). Hodnota čítače by tedy byla zaznamenána jen po každé  $N$ -té hraně, ale po každé jedné hraně by byla vynulována.

### 4.5.3 Měření intervalu dvou událostí, měření fázového posunu

Měření intervalu znázorněné na obr. 4.20 vychází z konfigurace pro měření periody (obr. 4.17).

Vstupní kanály již nejsou spojené a u každého je možno nezávisle nastavit detekovanou hranu. Budeme předpokládat, že impuls na kanálu A (Start) přijde vždy dříve, než impuls na kanálu B (Stop).

Možnosti zpracování jsou obdobné jako při měření periody. Opět budeme pracovat s hodnotami  $T_{1a,b}$  a  $T_{2a,b}$ . Interval dvou událostí se určí jako

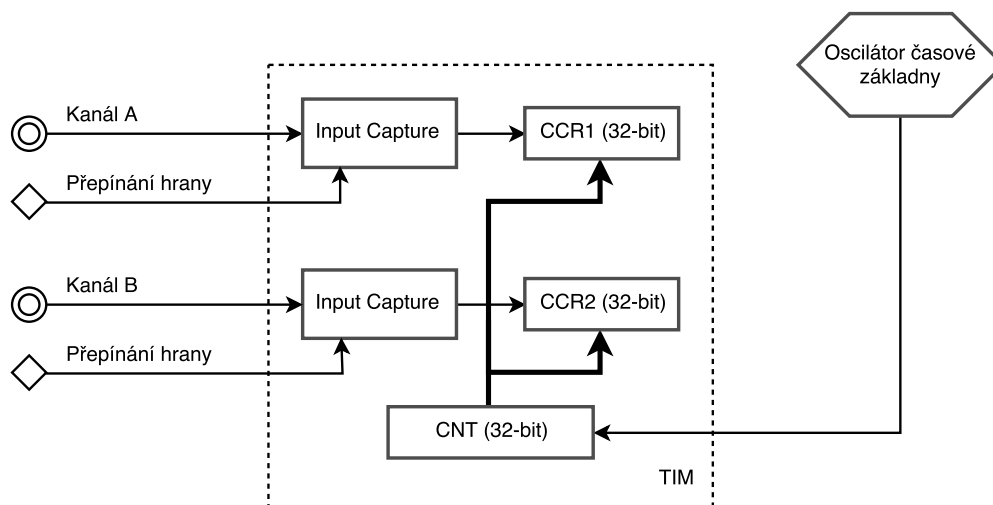
$$T_{interval} = T_{CLK} \cdot (T_{2a} - T_{1a}) \quad (4.22)$$

Pokud jsou signály v obou kanálech periodické, můžeme jejich periodu určit jako

$$T_1 = T_{CLK} \cdot (T_{1b} - T_{1a}) \quad (4.23)$$

$$T_2 = T_{CLK} \cdot (T_{2b} - T_{2a}) \quad (4.24)$$

Platí-li  $T_1 = T_2 = T_{signal}$ , má smysl mluvit o fázovém posunu signálů. Hodnoty vypočítané výše jednoduše dosadíme do vzorce (3.17) a můžeme určit vzájemnou fázi signálů  $\varphi_{B-A}$ .

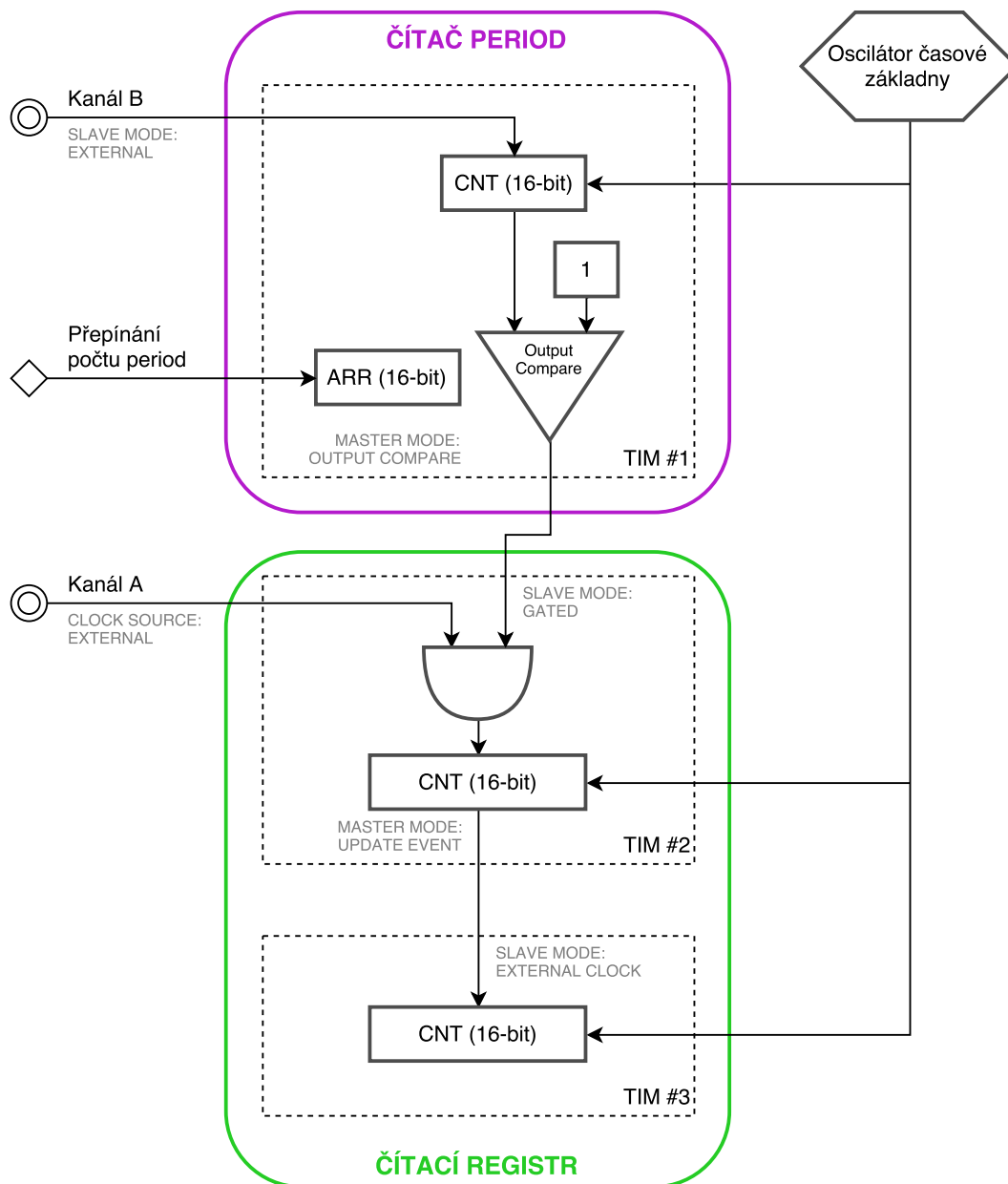


**Obrázek 4.20:** Konfigurace pro měření intervalu dvou událostí (32-bit, IC)

$$\varphi_{B-A} = \frac{-2\pi \cdot T_{interval}}{T_{signal}} \quad (4.25)$$

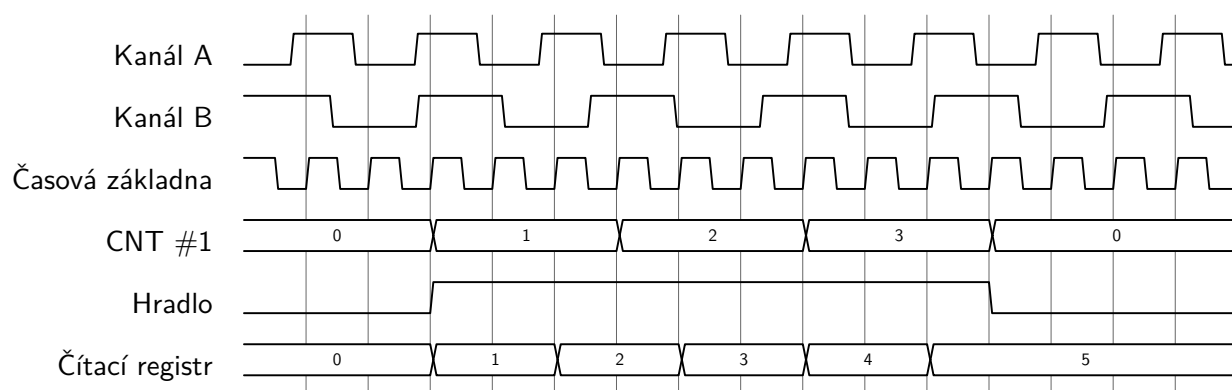
#### ■ 4.5.4 Měření poměru dvou frekvencí

Konfigurace pro měření poměru frekvencí (obr. 4.21) z velké části vychází, možná poněkud překvapivě, ze způsobu měření periody popsaného v části 4.5.2. Připomeňme, že čítač period na měřeném signálu byl použit k hradlování čítání pulzů časové základny. Zde je rozdíl v tom, že čítač period jednoho signálu používáme pro hradlování čítání period druhého signálu. Princip hradlování je jinak identický, je tedy zajištěno, že hradlování bude zahájeno i ukončeno synchronně s hranou signálu B.



Obrázek 4.21: Zapojení periférií mikrokontroléru pro měření poměru dvou frekvencí

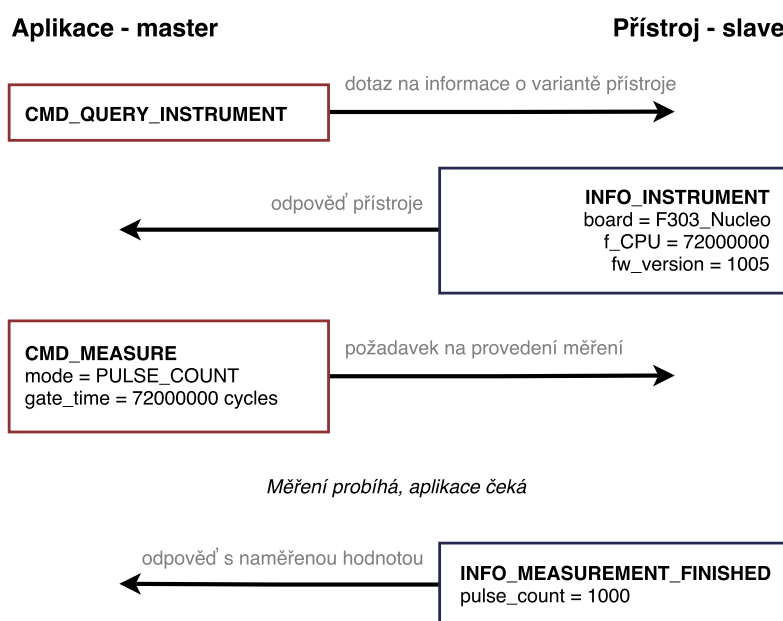




**Obrázek 4.22:** Měření poměru frekvencí. *CNT #1* čítá kanál B, Čítací registr čítá kanál A.  $CCR = 1$ ;  $ARR = 3$ .

## 4.6 Komunikace se zařízením

Model komunikace s PC lze navrhnout různými způsoby. Jednodušším způsobem je blokovácí model. Komunikace pak probíhá tak, že ovládací aplikace vyšle dotaz nebo příkaz přístroji, ten začne požadavek zpracovávat a až po jeho úplném zpracování vyšle odpověď. Toto ilustruje obr. 4.23.

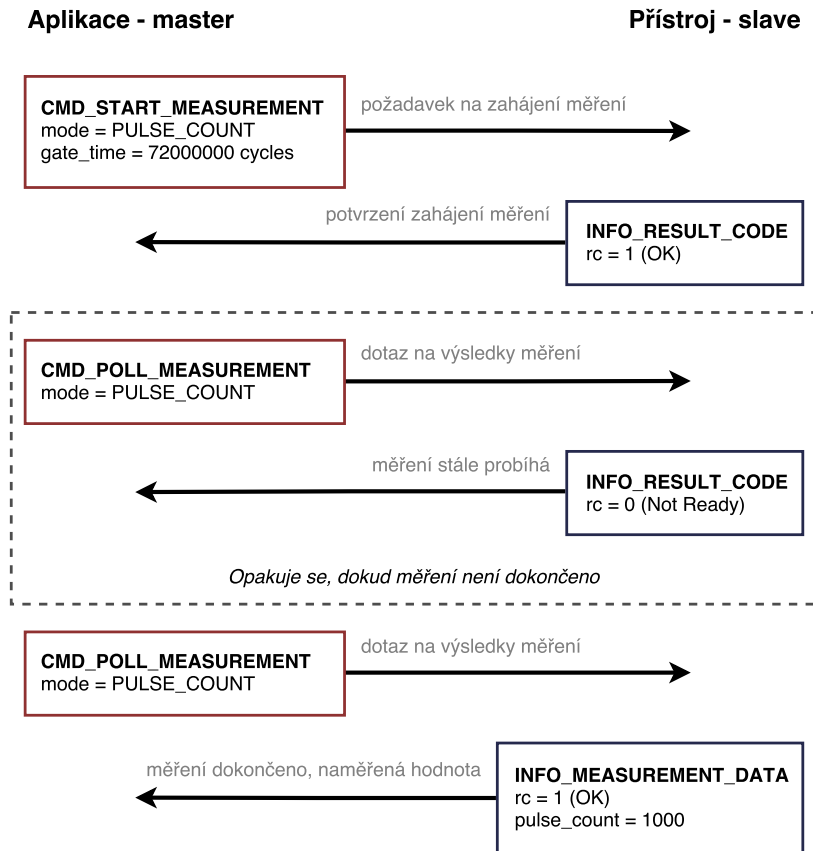


Obrázek 4.23: Blokovácí model komunikace

Tento model má 2 nevýhody: přístroj v jednu chvíli nemůže vykonávat více než jeden příkaz; a zatímco aplikace čeká na odpověď, nepozná, jestli je přístroj zaneprázdněn měřením, nebo došlo k přerušení komunikace (při použití sériové komunikace nemusí být k dispozici postranní kanály, které by poruchu indikovaly).

Oba nedostatky řeší asynchronní model s periodickým dotazováním (*polling*), který je znázorněn na obr. 4.24. Na každý požadavek aplikace přístroj neprodleně odpoví potvrzením nebo chybovým kódem (zahájení měření například může selhat z důvodu nedostatku paměti pro požadovaný počet vzorků). Komunikaci vždy zahajuje aplikace, přístroj pouze odpovídá na požadavky.

V úvahu ještě připadá model zcela asynchronní, kde jsou požadavky i odpovědi značeny číslem, aby bylo zřejmé, na který požadavek odpověď reaguje. Tento model demonstruje obr. 4.25. Protože naše aplikace neumožňuje provádět v jednu chvíli více než jedno měření, tento způsob komunikace by neměl přínos.



Obrázek 4.24: Asynchronní model komunikace s periodickým dotazováním

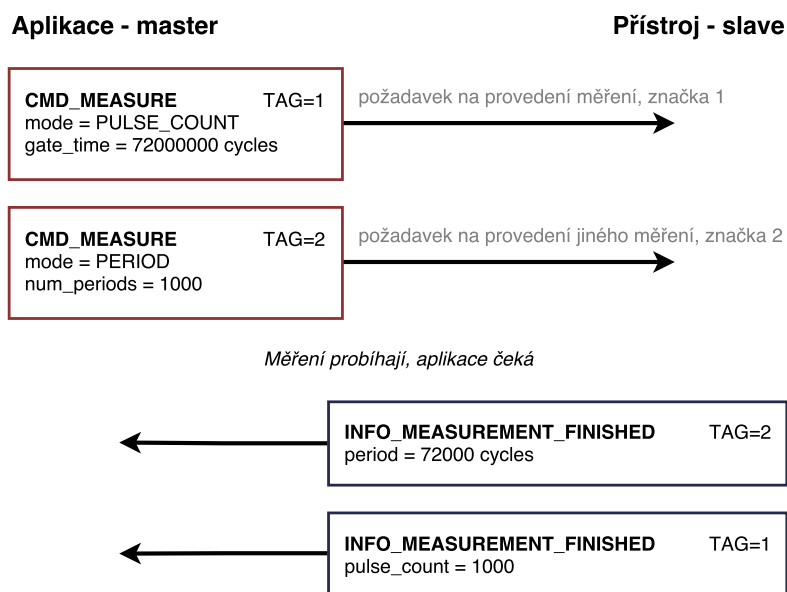
### 4.6.1 Pakety pro přenos požadavků a odpovědí

Na sběrnici USB jsou data přenášena v rámcích o různé délce. Na úrovni aplikace obecně nelze určit jak budou zprávy do rámců rozděleny, může se dokonce stát, že jeden rámec bude obsahovat konec jedné zprávy a zároveň začátek následující, příjemce zprávy tedy musí být schopen nějakým jiným způsobem poznat, kde zpráva začíná a končí. Bylo by možné například označit každou zprávu kódem určujícím její typ, a program by pak na základě tohoto typu mohl porovnat počet přijatých bajtů. Problémem je, že program by musel pro každý typ zprávy toto vyhodnocení provádět odděleně, což by vedlo k duplikaci kódu, náchylnosti k chybám, a v neposlední řadě k tomu, že vysílací program by nesměl vyslat zprávu, o které není jisté, zda ji příjemce dokáže zpracovat.

Řešením je tyto dva úkoly – tedy přijetí zprávy a interpretace jejího obsahu – oddělit.

Veškerá komunikace probíhá v paketech, které začínají hlavičkou nesoucí informaci o typu požadavku nebo zprávy a informaci o délce datové části paketu. Kromě hlavičky může paket nést až 255 bajtů dalších dat. Příkazy, které nenesou žádné další parametry, datovou část neobsahují. Ilustraci paketů poskytuje obr. 4.26.

Pokud přístroj obdrží paket, jehož typ nezná, odpoví zprávou typu **INFO\_RESULT\_CODE** s chybovým kódem **RESULT\_CODE\_PROTOCOL\_ERROR**. Přijatý paket je zahozen



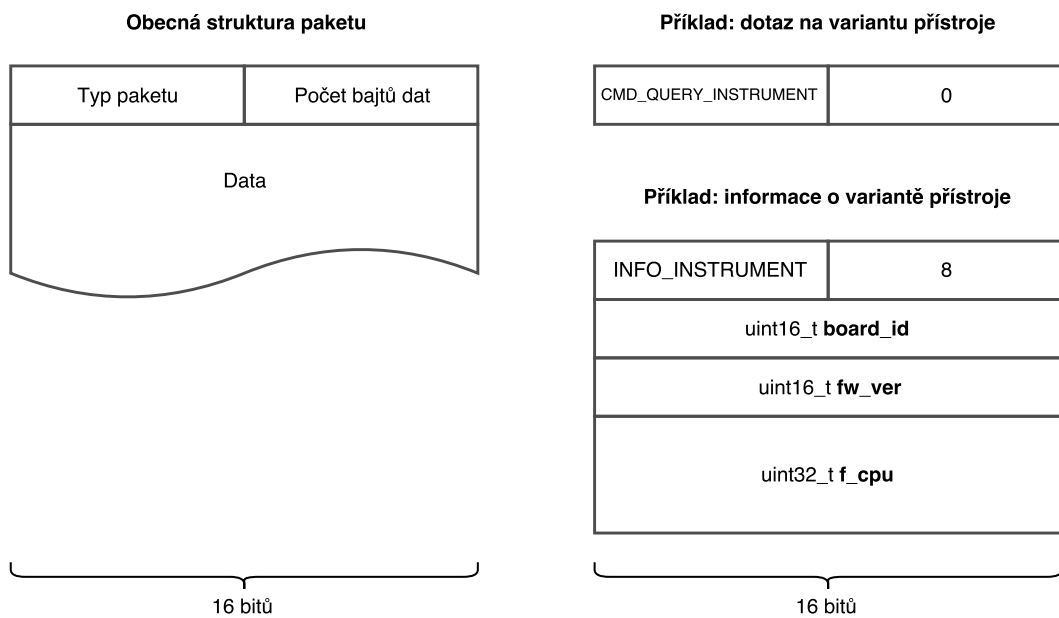
**Obrázek 4.25:** Asynchronní model komunikace se značkováním požadavků a odpovědí

a přístroj je připraven přijímat další požadavky.

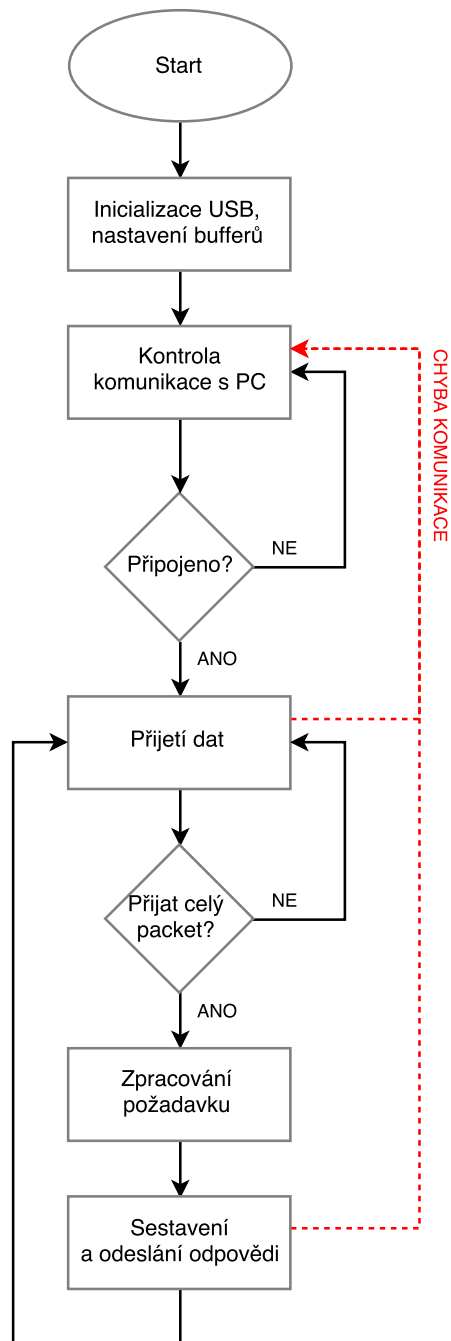
Součástí komunikačního protokolu bývá také detekce, případně oprava chyb. Protože tato problematika je řešena již na nižší úrovni (protokol sběrnice USB), zde se jí nemusíme zabývat.

Tento způsob komunikace s použitím stejné řady mikrokontroléru se mi osvědčil již v minulosti <sup>7</sup>, a z této zkušenosti jsem při implementaci vycházel. Obr. 4.27 ukazuje finální podobu komunikační logiky v přístroji.

<sup>7</sup> *Ocarina II*, analyzátor sběrnice CAN. eForce FEE Prague Formula, FEL ČVUT, 2016



**Obrázek 4.26:** Formát paketu a konkrétní příklady



**Obrázek 4.27:** Komunikační logika při použití paketového protokolu

## ■ 4.7 Realizace ovládací aplikace

Úkolem grafické ovládací aplikace je zpřístupnit všechny funkce přístroje intuitivním a smysluplným způsobem. Přidaná hodnota aplikace spočívá také v dalších funkcích, které samotný měřicí přístroj nerealizuje (výpočet nejistoty, grafy, zaznamenávání dat).

V následující části jsou rozebrány některé technické aspekty, které jsou pro fungování aplikace nezbytné.

### ■ 4.7.1 Softwarová platforma

Pro realizaci aplikace a uživatelského rozhraní připadaly v úvahu 3 softwarové platformy:

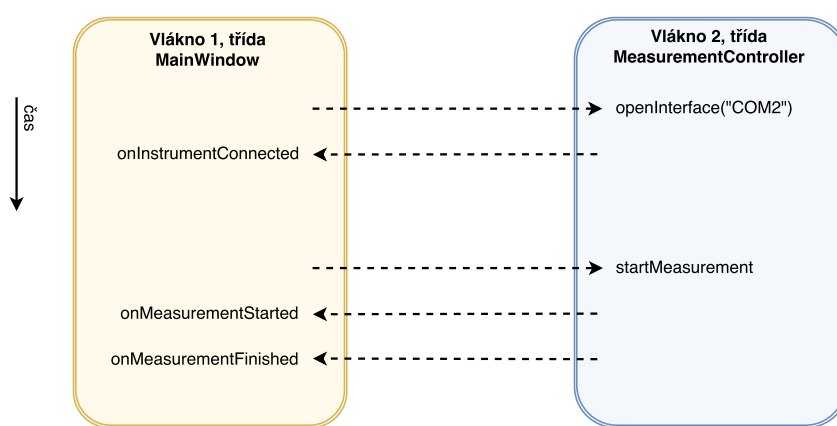
- .NET, WPF – Jazyk C#, s frameworky .NET a Windows Presentation Foundation, je jedním z nejpobulárnějších nástrojů pro vývoj moderních aplikací. Nabízí všechny potřebné funkce (především přístup k zařízením pro sériovou komunikaci) a velice kvalitní prostředky pro vývoj uživatelského rozhraní. Zásadním problémem je omezení na operační systém Microsoft Windows. Za určitou nevýhodu lze také považovat neexistenci otevřené implementace platformy WPF.
- Java, Swing – multiplatformní knihovna Swing umožňuje tvorbu uživatelského rozhraní v jazyce Java. Je starší než WPF, nicméně nezbytné funkce poskytuje. Přístup ke komunikačním rozhraním lze řešit knihovnamí třetí strany, jejichž kvalita je uspokojivá. Projekt OpenJDK poskytuje zcela otevřenou implementaci běhového prostředí i potřebných knihoven. Nevýhodou je nutnost přítomnosti běhového prostředí JRE pro spuštění výsledné aplikace.
- C++, Qt – platforma Qt je tvořena několika knihovnami, které kromě uživatelského rozhraní implementují práci se soubory, přístup k síti, sériovou komunikaci a další funkce. Knihovny jsou dostupné pod licencí GPL. Aplikace lze provozovat na operačních systémech Windows, macOS i GNU/Linux. Určitou komplikací je nutnost překladu programu pro každý systém zvlášť. Pro spuštění není vyžadována instalace běhového prostředí, je však nutné s aplikací distribuovat dynamické knihovny o celkové velikosti v řádu desítek megabajtů.

Rozhodl jsem se pro použití platformy Qt, a to z následujících důvodů:

- přenositelnost napříč operačními systémy bez instalace běhového prostředí
- kvalitní dokumentace a dostupnost zdrojů na internetu
- knihovny pro sériovou komunikaci a vykreslování grafů jsou součástí platformy
- vývojové prostředí Qt Creator řeší velkou část problémů spojených s překladem programu
- možnost využít nejmodernější možnosti jazyka C++17

### 4.7.2 Architektura aplikace

Aplikace běží ve dvou vláknech. Jedno vlákno je vyhrazeno jen pro komunikaci s přístrojem, druhé vlákno obsluhuje uživatelské rozhraní a zbylé funkce. Komunikace mezi vlákny je realizována systémem *signálů* a *slotů* z knihovny Qt. Nevýhoda je v duplikaci kódu – pro zavedení signálu s jedním příjemcem je nutné seznam argumentů opakovat přinejmenším 4x (deklarace signálu ve třídě, deklarace slotu příjemce, implementace slotu příjemce a napojení signálu na slot). Každou změnu v argumentech je nutné reflektovat na všech místech. Knihovna Qt pak ale zcela transparentně řeší přenos argumentů a vyvolání slotu napříč vlákny, což je přijatelný kompromis. Využití signálů a slotů pro volání funkcí napříč vlákny ilustruje obr. 4.28.



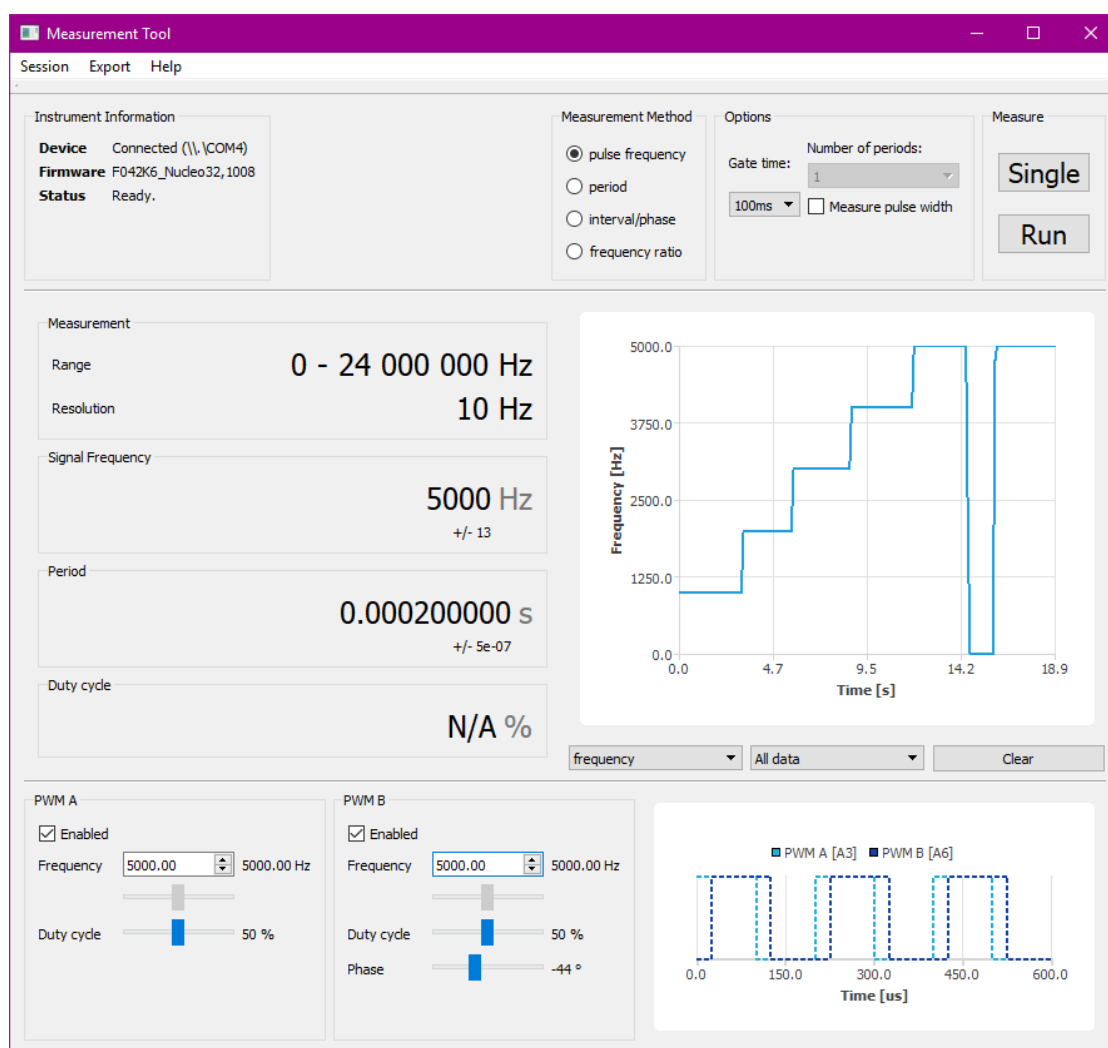
Obrázek 4.28: Signály a sloty: připojení přístroje a provedení jednoho měření

### 4.7.3 Grafické uživatelské rozhraní

Podoba uživatelského rozhraní po připojení měřicího přístroje je zachycena na obr. 4.29. Okno aplikace má 3 hlavní části:

1. Horní část zobrazuje informaci o připojené variantě přístroje a stavu (připraven, probíhá měření, chyba). V pravé části se nachází ovládací prvky pro nastavení měření. Po výběru metody měření se zpřístupní nastavení parametrů, např. u měření čítáním je možné zvolit dobu hradování. Konečně se zde nacházejí tlačítka pro provedení jednoho odměru a pro zahájení a ukončení kontinuálního měření.
2. Střední část okna, a zároveň největší část prostoru, je věnována zobrazení naměřených hodnot a jejich průběhu v čase. U veličin je uvedena nejistota měření. Graf umožňuje volbu zobrazené veličiny a rozsahu časové osy (celé měření, posledních 10 sekund, posledních 60 sekund).
3. Dolní část obrazovky je věnována nastavení vestavěného generátoru signálu. Je možné nezávisle měnit parametry obou výstupních kanálů, tedy frekvenci a střídu. Dále je umožněno nastavit fázový posun obou signálů – toto nastavení má smysl, jen





Obrázek 4.29: Podoba grafického uživatelského rozhraní

pokud jsou frekvence stejné. Uživateli je poskytnuta okamžitá zpětná vazba, a to dvěma způsoby. Vedle políček se zobrazuje skutečná hodnota nastaveného parametru vzhledem k možnostem přístroje. Pro větší názornost je vztah výstupních signálů znázorněn i graficky.



## Kapitola 5

### Ověření vlastností přístroje měřením

Pro ověření některých výše uvedených předpokladů a tvrzení o realizaci měřícího přístroje byla provedena série měření časových parametrů různých signálů. Není-li uvedeno jinak, probíhalo měření v těchto podmínkách:

- teplota prostředí 25 °C
- zdroj časové základny: externí krystal 8 MHz v pouzdře HC-49 (podrobnosti o výrobcí a typu nejsou známy), zapojený v nepájivém kontaktním poli, buzen interním obvodem mikrokontroléru, uzemňovací kapacita 22 pF na každém vývodu
- zdroj signálu: generátor funkcí HP 33120A 15MHz
- charakteristika signálu: obdélníkový signál se střídou 50 % a úrovněmi 0 V a 3,3 V.

#### 5.1 Měření referenčního signálu 1 MHz, 10 MHz

Cílem tohoto měření bylo určit orientační přesnost, které přístroj dosahuje při použití běžně dostupného krystalu. Metodou přímého měření frekvence byl 10x měřen referenční signál. Naměřené hodnoty jsou v tab. 5.1 a 5.2. Celé měření bylo pro kontrolu stability krystalu opakováno následující týden; od původního měření se hodnoty v průměru lišily o méně než 1 Hz.

#### 5.2 Měření signálu PPS z GPS přijímače

Zdrojem signálu byl GPS modul Trimble ICM osazený na evaluační desce popsané v [14]. Měřeno bylo vždy 10 po sobě následujících period. Naměřené periody jsou v tab. 5.3.

Číslo měření	Naměřená frekvence [Hz]
1	999958
2	999958
3	999958
4	999957
5	999957
6	999957
7	999957
8	999957
9	999956
10	999957

**Tabulka 5.1:** Měření referenčního signálu 1 MHz metodou přímého měření frekvence

Číslo měření	Naměřená frekvence [Hz]
1	9999569
2	9999568
3	9999569
4	9999569
5	9999569
6	9999568
7	9999569
8	9999569
9	9999569
10	9999569

**Tabulka 5.2:** Měření referenčního signálu 10 MHz metodou přímého měření frekvence

Číslo měření	Naměřená perioda [s]
1	1,000043157
2	1,000043158
3	1,000043152
4	1,000043148
5	1,000043153
6	1,000043146
7	1,000043147
8	1,000043158
9	1,000043159
10	1,000043152

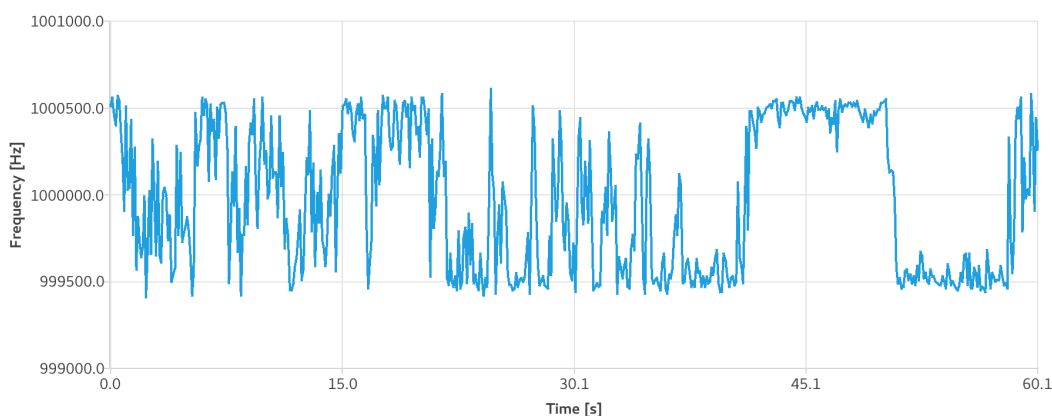
**Tabulka 5.3:** Měření periody referenčního signálu 1 Hz (GPS - PPS)

### 5.3 Porovnání zdrojů časové základny

Cílem bylo porovnat odchylky měření při použití různých zdrojů časové základny. Metodou přímého měření frekvence byl měřen referenční signál o frekvenci 1 MHz. Byla použita doba hradlování 10 s.

Zdroj časové základny	Naměřená frekvence [Hz]
8MHz krystal	999957
HSI48, doladování CRS	999600 <sup>1</sup>
HSI48	990054
HSI8	1001379

**Tabulka 5.4:** Měření 1MHz referenčního signálu za použití různých zdrojů časové základny



**Obrázek 5.1:** Kolísání měřené frekvence při použití interního oscilátoru HSI48 a zpětnovazebním doladováním (CRS)

### 5.4 Ověření projevu aliasingu

Pro ověření projevu aliasingu jsme použili plechovkový oscilátor s TTL výstupem o nominální frekvenci 66 MHz. Laboratorní čítač HP 53131A naměřil frekvenci 65,996015 MHz.

Přístroj používal časovou základnu o frekvenci 48 MHz, lze tedy očekávat naměření zdánlivé frekvence signálu  $f_Z$ :

$$f_Z = f_{signal} - f_{timebase} = 18 \text{ MHz}$$

<sup>1</sup>Jak je vidět z obr. 5.1, naměřená frekvence v čase značně kolísá

Byla použita metoda přímého měření frekvence. Naměřené hodnoty jsou uvedeny v tab. 5.5.

Číslo měření	Naměřená frekvence [MHz]
1	17,941
2	17,935
3	17,934
4	17,929

**Tabulka 5.5:** Měření 66MHz signálu při frekvenci časové základny 48 MHz.





## Kapitola 6

### Zhodnocení dosažených výsledků

Cílem práce bylo navrhnout a realizovat virtuální přístroj pro analýzu signálu v modulační doméně, tedy přístroj realizující funkce čítače s možností sledovat a zaznamenat vývoj parametrů v čase.

Dosažené parametry pro jednotlivé metody měření při použití mikrokontroléru STM32F042 jsou uvedeny v následujícím přehledu:

#### 1. Přímé měření frekvence

Rozsah:	0 – 24 000 000 Hz
Doba hradlování $T_{gate}$ :	100 ms, 1 s, 10 s
Rozlišení:	$\frac{1}{T_{gate}}$
Vhodné pro:	měření signálů o vysoké frekvenci (např. krystalový oscilátor), orientační určení frekvence signálu (doba měření je konstantní)

#### 2. Měření periody

Rozsah:	41,67 ns – 89,48 s (0,01118 - 24 000 000 Hz)
Počet měřených period:	1, 10, 100, 1 000, 10 000
Rozlišení:	20,83 ns při měření 1 periody, $\frac{20,83}{n_p}$ ns při měření $n_p$ period <sup>1</sup>
Vhodné pro:	přesné měření signálů o nízké frekvenci (např. signál Pulse Per Second)

#### 3. Měření periody a pulzní šířky

---

<sup>1</sup>Maximální měřitelná délka periody se sníží  $n_p$ -krát

Rozsah:	500 ns – 89,48 s (0,01118 - 2 000 000 Hz) pro střihu 4–96 % 2000 ns – 89,48 s (0,01118 - 500 000 Hz) pro střihu 1–99 %
Počet měřených period:	1, 10, 100
Rozlišení:	20,83 ns při měření 1 periody, $\frac{20,83}{n_p}$ ns při měření $n_p$ period
Vhodné pro:	měření signálů využívajících pulzně šířkovou modulaci (např. výstupy některých senzorů)

#### 4. Měření intervalu, časového posunu a fázového posunu

Rozsah frekvence signálu:	0 – 100 000 Hz
Rozlišení:	20,83 ns
Vhodné pro:	měření zpoždění mezi dvěma signály o stejné frekvenci

#### 5. Měření poměru dvou frekvencí

Rozsah frekvence signálů:	0 – 24 000 000 Hz
Počet měřených period:	1, 10, 100, 1 000, 10 000 (počítáno na kanálu B)
Rozlišení:	$\frac{1}{n_B}$ , kde $n_B$ představuje počet měřených period
Vhodné pro:	využití přesné (referenční) časové základny pro určení frekvence signálu

Přístroj prezentovaný v této práci se kvantitativními parametry pochopitelně nemůže rovnat špičkovým komerčním řešením. Spolu s ovládací aplikací však poskytuje kombinaci funkcí, která chybí i v poměrně drahých přístrojích. To ho činí dobrou volbou pro mnoho laboratorních měření.

Na začátku jsme stanovili tři konkrétní platformy pro realizaci: mikrokontroléry STM32F042, STM32F303 a STM32F411. V době odevzdání práce je v produkční kvalitě připravená pouze varianta pro procesor F042. Díky struktuře firmwaru by rozšíření na další procesory nemělo být složité, jde však už o další vývoj.

Ovládací software splňuje formální požadavky zadání a zpřístupňuje důležité funkce přístroje. Zasloužil by však ještě další vývoj, aby jeho používání bylo více efektivní a komfortní. Největší prostor pro zlepšení je v oblasti zaznamenávání dat po delší dobu, pro které se současná podoba rozhraní ukázala být poněkud kostrbatou.

Co se týče možností dalšího rozšíření a navázání na dosavadní práci, bylo by možné pokračovat několika různými směry. Programové vybavení by bylo možné rozšířit o podporu dalších rodin mikrokontrolérů, nebo naopak záběr zúžit na jeden konkrétní procesor, a důsledněji využít veškerý jeho potenciál. Lze uvažovat o využití vysokofrekvenčního čítače HRTIM, jímž jsou vybaveny mikrokontroléry řady STM32F334. Jeho využití by mohlo značně zvýšit rozsah i rozlišení měření. Přidáním ovládacích prvků a displeje by zase mohl být vytvořen samostatný laboratorní přístroj, který pro provoz nebude vyžadovat připojení k PC.



## Bibliografie

- [1] *ABCs of Modulation Domain Analysis. Application Note.* Spectracom. 2012. URL: [https://spectracom.com/sites/default/files/document-files/ABCs-of-Modulation-Domain-Analysis\\_AN01-201\\_rev3.pdf](https://spectracom.com/sites/default/files/document-files/ABCs-of-Modulation-Domain-Analysis_AN01-201_rev3.pdf).
- [2] *HP 53131A.* URL: [http://www.sglabs.it/public/SgLabs\\_HP\\_53131A\\_1.JPG](http://www.sglabs.it/public/SgLabs_HP_53131A_1.JPG).
- [3] *Fundamentals of the Electronic Counters.* Agilent Technologies, 1997. URL: <http://cp.literature.agilent.com/litweb/pdf/5965-7660E.pdf>.
- [4] *STM32F042x4 STM32F042x6: ARM-based 32-bit MCU, up to 32 KB Flash, crystal-less USB FS 2.0, CAN, 9 timers, ADC and comm. interfaces, 2.0 - 3.6 V.* STMicroelectronics. 2017.
- [5] *RM0091: STM32F0x1/STM32F0x2/STM32F0x8 advanced ARM-based 32-bit MCUs.* STMicroelectronics. 2014.
- [6] *ATmega48A/PA/88A/PA/168A/PA/328/P.* Atmel Corporation. 2015.
- [7] *SAM D21E / SAM D21G / SAM D21J.* Microchip Technology Inc. 2014.
- [8] *PIC18F2455/2550/4455/4550 Data Sheet.* Microchip Technology Inc. 2009.
- [9] *AN2867. Oscillator design guide for STM8S, STM8A and STM32 microcontrollers.* STMicroelectronics. 2015. URL: [http://www.st.com/resource/en/application\\_note/cd00221665.pdf](http://www.st.com/resource/en/application_note/cd00221665.pdf).
- [10] *Crystal Oscillator Basics and Crystal Selection for rfPIC and PICmicro Devices.* Microchip Technology Inc. 2002. URL: <http://ww1.microchip.com/downloads/en/AppNotes/00826a.pdf>.
- [11] Ian Poole. *TCXO, Temperature Compensated Crystal Oscillator.* URL: <http://www.radio-electronics.com/info/data/crystals/tcxo.php>.
- [12] *TI GPS PPS Timing Application Note.* Texas Instruments. 2012. URL: [http://processors.wiki.ti.com/images/f/f1/TI\\_GPS\\_PPS\\_Timing\\_AppNote.pdf](http://processors.wiki.ti.com/images/f/f1/TI_GPS_PPS_Timing_AppNote.pdf).
- [13] Adam Berlinger. “Implementace přístrojových funkcí s využitím mikrořadičů STM32”. Dipl. FEL ČVUT, 2016.
- [14] V. Vigner, J. Roztočil a B. Čemusová. “Evaluation of timing GPS receivers for industrial applications”. In: *Proceedings of the 12th IMEKO TC10 Workshop on Technical Diagnostics.* Università di Firenze. Florencie, 2013.





## Příloha A

### Seznam souborů na CD

<b>firmware/</b>	Zdrojové kódy firmware přístroje
<b>gui/</b>	Zdrojové kódy grafické ovládací aplikace
<b>manual.pdf</b>	Stručný návod k instalaci a používání aplikace (v angličtině)
<b>thesis.pdf</b>	Plný text bakalářské práce