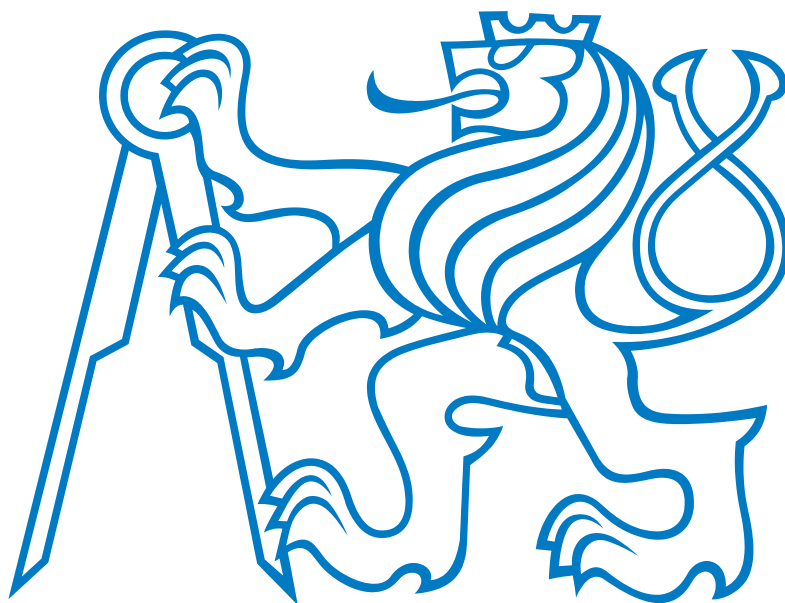


Univerzální čítač s STM32G031 pro laboratorní experimenty

Uživatelský manuál

Ondřej Hloušek

České vysoké učení technické v Praze



Fakulta elektrotechnická

Katedra měření

26. května 2023

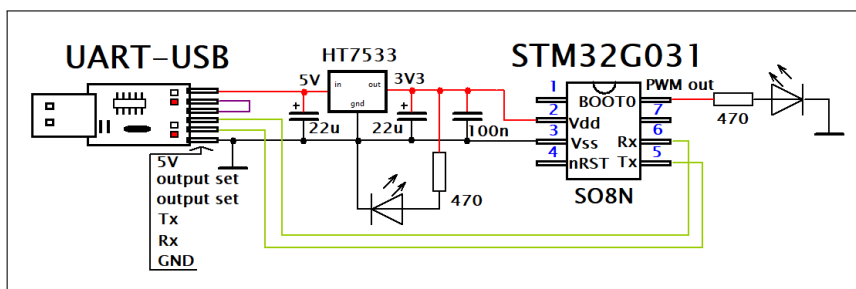
Obsah

1	Základní nastavení a zprovoznění mikrokontroleru	3
1.1	Zapojení na nepájivém poli	3
1.2	Nahrání firmware-u do MCU a nastavení reset pinu	4
2	Nastavení aplikace Data Plotter	6
3	Menu uživatelského terminálu	6
4	PWM Generátor	7
5	Osciloskop	8
6	Logický analyzátor	9
7	Čítač	10
7.1	Měření periody a střídy, reciproční měření frekvence	11
7.2	Měření časového odstupu dvou po sobě jdoucích hran	11
7.3	Záznam časového odstupu N po sobě jdoucích hran	12
7.4	Měření frekvence	13

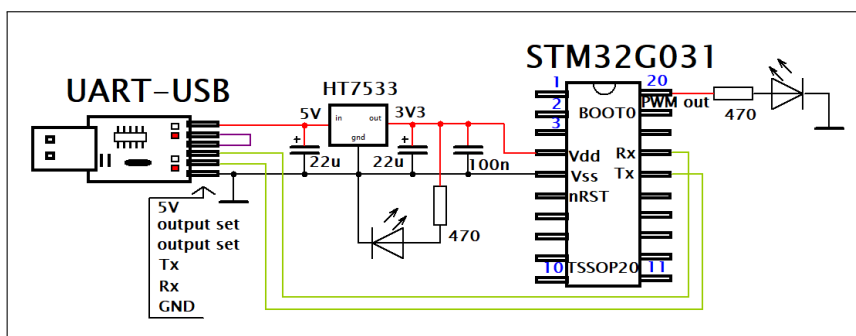
1 Základní nastavení a zprovoznění mikrokontroleru

1.1 Zapojení na nepájivém poli

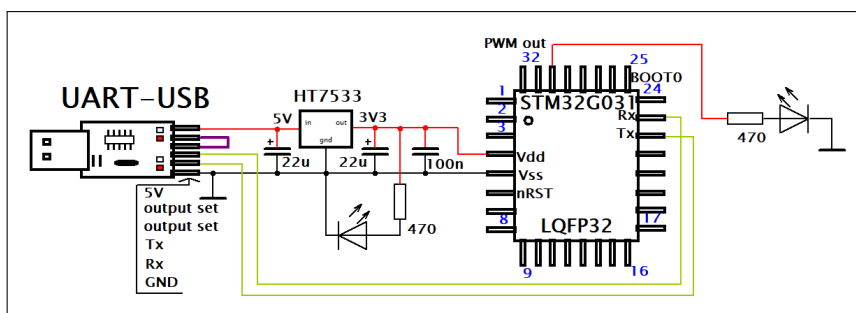
Ještě před započítím samotného programování MCU musíme zajistit připojení napájecího napětí a vodičů Rx a Tx k bráně UART, který bude sloužit nejen jako komunikační rozhraní mezi mikrokontrolerem a aplikací v PC, ale i jako prostředek pro nahrání programu (firmware-u) do MCU. Pro samotné napájení pak použijeme napětí z USB zprostředkované USB-UART převodníkem, který ovšem poskytuje jen napětí 5 V, což však našemu mikrokontroleru nevyhovuje. Budeme tedy používat lineární stabilizátor z 5 V na 3,3 V, které jsou již pro naše MCU přípustné. Zapojení pro všechna tři pouzdra vidíme na obrázcích 1, 2, 3. Na všech schématech pak vidíme kromě již zmíněného stabilizátoru napětí i kondenzátor na jeho vstupu zabraňující jeho rozkmitání, dále pak kondenzátory keramický a elektrolytický na vstupu napájení mikrokontroleru, které zde jsou pro pokrytí odběrových špiček MCU, keramický kondenzátor kompenzuje krátké, málo výkonové špičkové odběry a ten elektrolytický pak delší, více výkonové špičkové odběry. Nakonec je zde LED připojená přes rezistor na napájecí napětí sloužící k indikaci napájení mikrokontroleru a LED přes rezistor připojená k pinu MCU s výstupem PWM Generátoru, která slouží jednak k testování samotného PWM Generátoru, ale i k indikaci začátku běhu programu po úspěšném nahrání vytvořeného firmware-u do mikrokontroleru.



Obrázek 1: Zapojení mikrokontroleru v pouzdře SO8N



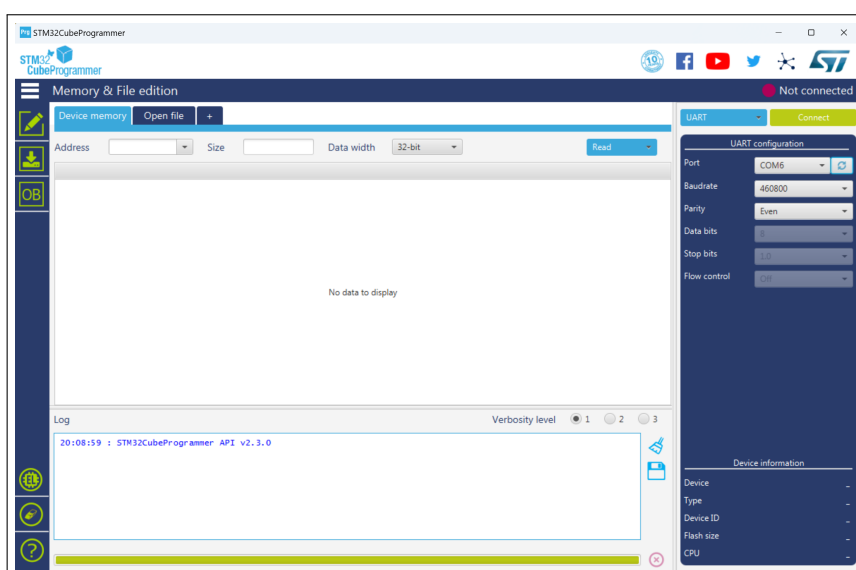
Obrázek 2: Zapojení mikrokontroleru v pouzdře TSSOP20



Obrázek 3: Zapojení mikrokontroleru v pouzdře LQFP32

1.2 Nahrání firmware-u do MCU a nastavení reset pinu

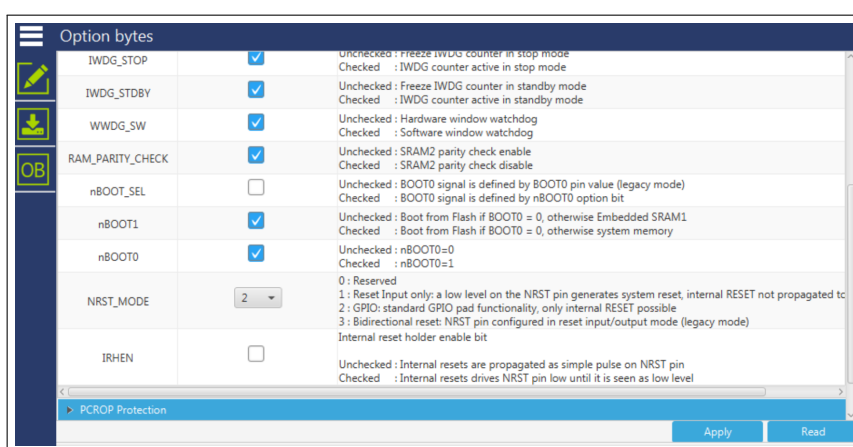
Pro nahrání firmware-u do mikrokontroleru budeme používat aplikaci STM32CubeProgrammer. Konkrétně použijeme jeho verzi 2.3.0, jak vidíme v náhledu aplikace na obrázku 4. Novější verze tohoto programu již některá nastavení neumožňují, proto s nimi nebude možné níže uvedený postup zopakovat. Po řádném zapojení mikrokontroleru, viz. 1.1, a připojení převodníku USB-UART do vhodného portu počítače musíme MCU přepnout do BOOT režimu, kdy do něj lze nahrát program. To provedeme připojením BOOT0 pinu na log.1 (3,3 V) přes rezistor cca 470 Ω a zároveň resetu mikrokontroleru přivedením na pin nRST log.1 (3,3 V) přes rezistor 470Ω. Dále pak nastavíme v pravé části okna aplikace komunikaci po UART, konkrétní port, rychlost komunikace „Baudrate“ na 460800Bd/s, paritu sudou „even“ a připojíme se se k MCU tlačítkem „Connect“.



Obrázek 4: Náhled aplikace STM32CubeProgrammer

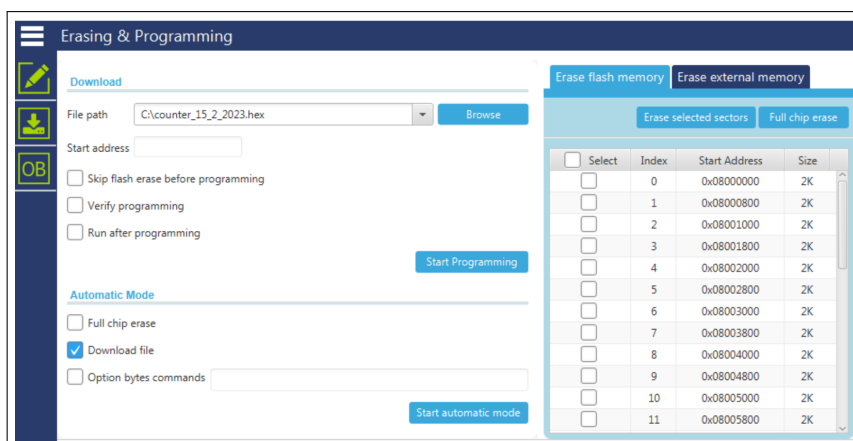
Po úspěšném propojení aplikace a mikrokontroleru v levé horní části okna aplikace otevřeme položku „Option bytes“ a v ní rozbalíme lištu „User Configuration“, kde úplně dole odstraníme označení nejprve z položky nBOOT_SEL, dále pak nastavíme NRST_MODE na číslo 2 čímž si uvolníme pin s vývodem resetu, tedy v případě připojení napájení se MCU podívá zda je na reset pinu log.1 a dokud zde log.1 je mikrokontroler bude držen

v resetu, po odpojení zdroje log.1 pak MCU reset stav opustí a začne vykonávat program, ovšem v tomto módu již nezáleží na logickém stavu reset pinu a k resetu dochází opět až při odpojení napájení. Tento mód nám tedy umožní využívat periferie vyvedené na stejný fyzický pin, jako reset, což využijeme především u nejmenšího osmi pinového pouzdra, ovšem u větších pouzder již tuto úpravu díky dostatečnému množství dělat nemusíme. Po nastavení resetu ještě odstraníme označení u položky IHREN a měli bychom tak dosáhnout nastavení, jako na obrázku 5, kde jsou všechny ostatní položky zaškrtnuté. Teď tedy zbývá pouze do mikrokontroleru uložit naše nastavení tlačítkem „Apply“.



Obrázek 5: Nastavení reset pinu v aplikaci STM32CubeProgrammer

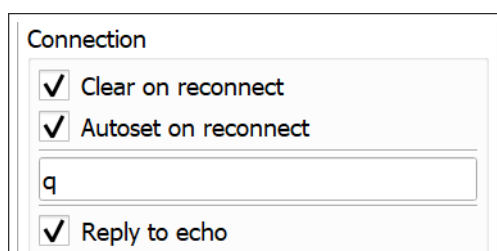
Pro samotné nahrání programu do mikrokontroleru se přesuneme do části „Erasing & programming“ nacházející se opět v levé horní části, jeho náhled pak vidíme na obrázku 6. Zde pouze vybereme požadovaný firmware, typicky soubory s příponou .bin, .elf nebo .hex, a nahrajeme jej do MCU tlačítkem „Start Programming“. Po úspěšném nahrání programu do mikrokontroleru ho musíme opět dostat z BOOT režimu, což docílíme odstraněním všeho připojeného na tlačítko BOOT0 a reset pin, tedy přivedením log.0 na BOOT0 pin a log.1 na reset pin a následným resetem MCU, což lze uskutečnit buď odpojením a opětovným připojením napájecího napětí, nebo pokud jsme u větších pouzder nezablokovali funkci reset pinu, pak i přivedením log.1 na reset a následným odejmutím log.1 z reset pinu. Při každém dalším spouštění mikrokontroleru musí zůstat piny BOOT0 a reset volné, případně připojené na log.0, aby nedocházelo k nechtěnému přechodu do BOOT režimu. Správné rozběhnutí programu si pak lze ověřit krátkým zablikáním LED na pinu s výstupem PWM Generátoru při každém startu programu po resetu.



Obrázek 6: Nahrání firmware-u do MCU pomocí aplikace STM32CubeProgrammer

2 Nastavení aplikace Data Plotter

Pro zobrazení naměřených dat a ovládání našeho měřicího přístroje budeme používat PC aplikaci Data Plotter, která je volně ke stažení na <https://github.com/jirimaier/DataPlotter/releases>. Po stažení a spuštění aplikace v ní provedeme základní nastavení, a to rychlosti přenosu v pravém horním rohu na 460800Bd/s (defaultně je zde nastavena žádná parita, osm datových bitů a jeden stop bit) a následně přejdeme do sekce nastavení, kterou rozklikneme v pravém horním rohu nad terminálem. Zde povolíme vymazání po připojení, „autoset“ po připojení, odpověď na echo a jako znak odeslaný do mikrokontroleru po připojení nastavíme na ‘q’, který pokaždé resetuje nastavení MCU a vrátí jej do režimu MENU. Abychom nemuseli vždy po spuštění aplikace tento parametr zadávat, můžeme využít uložení veškerého nastavení aplikace do souboru. To provedeme v sekci nastavení, kterou rozklikneme v pravém horním rohu nad terminálem a soubor s nastavením pak uložíme tlačítkem vpravo dole „Save settings“, kde za jméno souboru pro jednoduchost zvolíme „degault.cfg“, čímž ale přepíšeme defaultní nastavení, proto je vhodné si ho nejprve uložit do jiné složky případně pod jiným jménem.



Obrázek 7: Nastavení potvrzení připojení v Data Plotter

3 Menu uživatelského terminálu

Jak vidíme na obrázku 8, v MENU je možné zvolit jednu z kombinací měřicích režimů nebo HELP, kde nalezneme zapojení MCU a vlastnosti jednotlivých měřicích režimů. Dále je zde zobrazen aktuálně mikrokontrolerem používaný zdroj hodinového signálu.



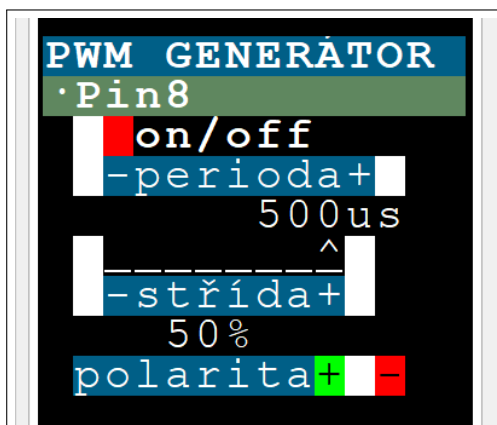
Obrázek 8: Terminál MENU

4 PWM Generátor

- Zapojení: PB6 (Pin8 pro SO8N)
- Rozsah periody: $1 \mu\text{s}$ až $32,767 \text{ s}$
- Rozsah střídání: 0% až 100% po 5%
- Možnost změny polarit (invertování) výstupního signálu

Jak vidíme na obrázku terminálu PWM generátoru 9, pod názvem se nachází označení pinu zapojení pro pouzdro SO8N, tlačítko s indikací pro spuštění/vypnutí generování PWM výstupu, dále je zde volba periody, kdy tlačítka přičítáme/odčítáme jedničku do cifry zvolené tlačítka níže. Posledním ovládacím prvkem je změna polarit, kdy je možné

volbou záporné polarity invertovat výstup na pinu generátoru. V případě vypnutého generátoru je na výstupní pin přiváděna defaultní logická úroveň, tedy pro kladnou polaritu log.0 a pro zápornou polaritu log.1, nesmí být tedy tento pin propojen s další částí obvodu bez ochranného rezistoru, který volíme minimálně 100 Ω .



Obrázek 9: Terminál PWM Generátoru

5 Osciloskop

- Zapojení: PA13 (Pin7 pro SO8N)
- „Trigger-ovací“ úroveň napětí: 0,2 V až 3,2 V po 0,2 V
- „Trigger“ na vzestupnou/sestupnou hranu
- Vzorkovací rychlost: 5 ksample/s až 2 Msample/s
- Délka záznamu: 4096 vzorků
- „Pre-trigger“: 400 vzorků
- Rozlišení vzorků: 8 bitů

Na obrázku terminálu osciloskopu 10 vidíme pod názvem číslo vstupního pinu pro pouzdro SO8N, dále tlačítko pro spouštění/vypínání měření a pod ním volbu jednoho z možných pracovních režimů. Režim AUTO po spuštění vzorkuje a vypisuje bez ohledu na „trigger“ podmínky. Režim NORMAL pak cyklicky čeká na nastavenou „trigger“ podmínku, tedy na náběžný/sestupný průchod „trigger-ovací“ napětíovou úrovní a následně ovzorkuje signál v délce vzorkovacího okna a zobrazí jej na terminál. U tohoto režimu pak osciloskop znovu čeká na splnění „trigger“ podmínky, ovšem u režimu SINGLE, který pracuje na stejném principu, dochází po jednom odměru k vypnutí osciloskopu. U obou těchto režimů se zobrazí i „pre-trigger“, tedy ovzorkovaný signál před splněním „trigger“ podmínky. Na terminálu je pak možné nastavit velikost „trigger-ovací“ napětí a charakter průchodu touto úrovní. Posledním nastavením u osciloskopu je jeho vzorkovací perioda, tedy jaký bude časový rozestup mezi jednotlivými vzorky.



Obrázek 10: Terminál Osciloskopu

6 Logický analyzátor

- Zapojení vstup: PB5 (Pin8 pro SO8N)
- Zapojení výstup (nepoužívá se): PA1 (Pin4 pro SO8N)
- Délka záznamu: 1 k až 32 k vzorků
- Vzorkovací rychlost: 125 ksample/s až 16 Msample/s
- „Trigger“ na vzestupnou/sestupnou hranu

Jak vidíme na obrázku terminálu Logického analyzátoru 11, pod názvem se nachází označení pinu zapojení pro pouzdro SO8N, tlačítko s indikací pro spuštění/vypnutí měření, kdy měření proběhne po splnění „trigger“ podmínky, která lze nastavit na náběžnou/sestupnou hranu logického signálu níže v terminálu. Po ovzorkování celé délky záznamu se záznam zobrazí a měření se vypne, princip je tedy stejný, jako u SINGLE režimu osciloskopu. Vzorkovací frekvence a délka záznamu lze opět nastavit v terminálu. Jelikož je na pouzdrě SO8N vyveden vstup logického analyzátoru na stejném pinu, jako výstup PWM generátoru je defaultně generátor vypnut, ale je možné jeho funkci příslušným tlačítkem povolit. Protože z principu vnitřního zapojení mikrokontroleru je u pouzdra SO8N vyveden na pin 4 hodinový signál SPI, která nám slouží pro vzorkování vstupního signálu při měření analyzátozem, musí pin číslo 4 zůstat nezapojený.

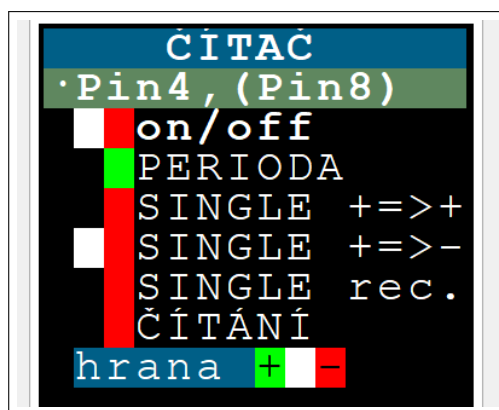


Obrázek 11: Terminál Logického analyzátoru

7 Čítač

- Zapojení (jednokanálové měření): PA0 (Pin4 pro SO8N)
- Zapojení (dvoukanálové měření), 1.kanáál: PA15 (Pin8 pro SO8N)
- Zapojení (dvoukanálové měření), 2.kanáál: PA1 (Pin4 pro SO8N)

Jak vidíme na obrázku terminálu Čítače 12, společného pro všechny měřicí režimy, pod názvem se nachází označení pinu zapojení pro pouzdro SO8N, tlačítko s indikací pro spuštění/vypnutí jednotlivých měření, výběr konkrétního měřicího režimu a volba hrany na kterou budou jednotlivá měření reagovat/začínat.

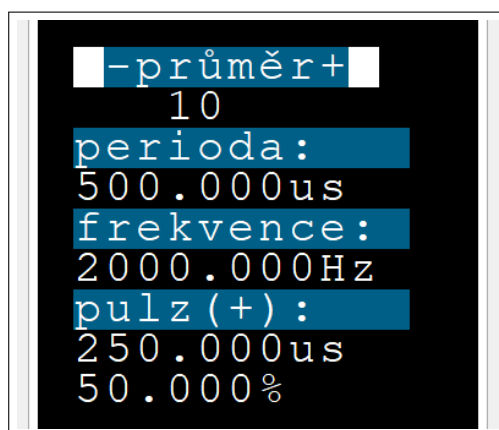


Obrázek 12: Společný terminál Čítače

7.1 Měření periody a střídy, reciproční měření frekvence

- Rozsah periody: 125 ns až 67,1 s
- Rozlišení: $\frac{1}{N}15,625$ ns
- Průměrování z N vzorků, kde N je: 1 až 250

V tomto režimu čítač měří periodu a střídu příchozího signálu v μs . Pamatuje si pak N hodnot těchto měření, ze kterých vypočítá průměrnou periodu, kde N se v terminálu nastavuje pod označením průměru. Z těchto hodnot v μs je dopočtena náležitá frekvence a střída v %, které jsou zobrazeny na terminálu, jak vidíme na obrázku 13.



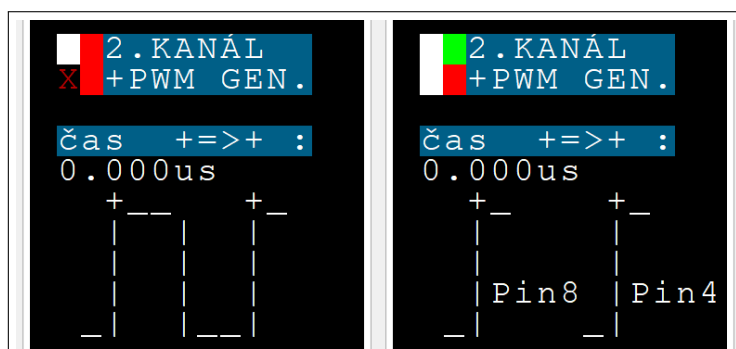
Obrázek 13: Terminál měření periody

7.2 Měření časového odstupu dvou po sobě jdoucích hran

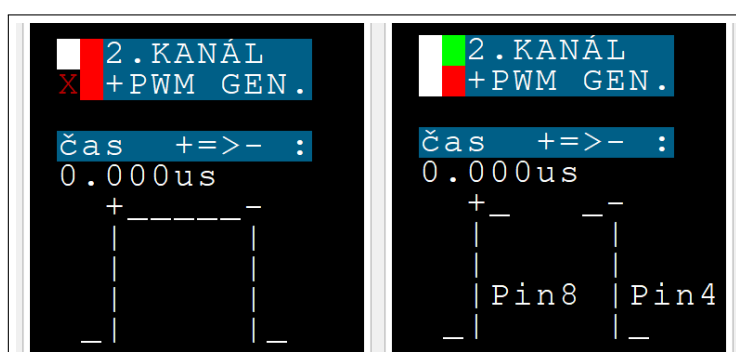
- Rozlišení: 15,625 ns
- Obě hany ze stejného zdroje/každá hrana z jiného zdroje
- 1.hrana náběžná/sestupná

- 2.hrana náběžná/sestupná
- Pro dvoukanálové měření možnost začít měřit zapnutím PWM generátoru

Toto měření je rozděleno do dvou režimů, kde ten první měří čas mezi stejnými hranami viz. 14 a druhý režim pak mezi opačnými hranami viz. 15. Na těchto obrázcích vidíme, že lze měřit jednocanálový signál na pinu č.4 (pro SO8N) nebo dvoukanálový signál, kdy první hrana je očekávána na pinu č.8 (pro SO8N) a druhá pak na pinu č.4 (pro SO8N). V principu u dvou kanálového měření každá hrana konkrétního, nastaveného charakteru na prvním kanálu započíná měření. Tedy po započnutí měření konkrétní hranou na pinu č.8 (pro SO8N) sem NESMÍ tato hrana přijít znovu dříve, než bude měření ukončeno příchodem konkrétní hrany na pin č.4(pro SO8N). Pro potřebu měření s PWM generátorem je zde možnost jeho připnutí, kdy se v principu odpojí kanál jedna a měření bude zahájeno po kliknutí na tlačítko zapínající měření, čímž se spustí i PWM generátor a měření pak bude ukončeno příchodem patričné hrany na druhý kanál, tedy na pin č.4 (pro SO8N).



Obrázek 14: Terminál měření časového odstupu dvou po sobě jdoucích hran stejného charakteru



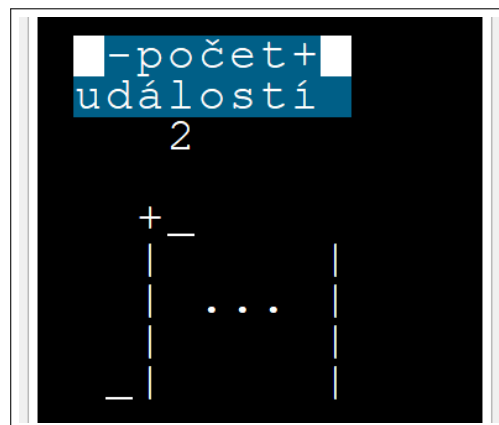
Obrázek 15: Terminál měření časového odstupu dvou po sobě jdoucích hran opačného charakteru

7.3 Záznam časového odstupu N po sobě jdoucích hran

- Rozlišení: 15,625 ns

- Maximální délka záznamu: 67,1 s
- Počet zaznamenaných hran: 2 až 500
- Začátek na náběžnou/sestupnou hranu

V tomto režimu čítač začne měření na první hranu náběžnou/sestupnou a následně zaznamená čas mezi příchodem každé z N hran a první příchozí hranou. Počet zaznamenaných hran lze nastavit v terminálu, jak vidíme na obrázku 16. Naměřená data jsou nakonec vykreslena na kanál dva do Data Plotter-u.



Obrázek 16: Terminál záznamu časového odstavu N po sobě jdoucích hran

7.4 Měření frekvence

- Rozsah: 0 MHz až 64 MHz
- Rozlišení: $\frac{1}{T_G}$ [Hz]
- Doba „hradlování“ T_G : 100 ms až 20 s
- Možnost zvolit prosté čítání hran
- Možnost čítání náběžných/sestupných hran

V tomto režimu se čítají po určitou dobu T_G příchozí hrany náběžné/sestupné a z doby T_G a počtu načítaných hran se vypočte frekvence, jako $f = \frac{N}{T_G}$. Odtud pak mikrokontroler dopočte periodu a zobrazí je spolu s počtem načítaných hran na terminálu. Z terminálu lze potom nastavovat dobu „hradlování“ T_G a to až na INF., které přepne mikrokontroler do režimu prostého čítání pulzů (hran).

```
-vzorkovací+  
okno  
100ms  
  
počet pulzů:  
10000  
frekvence:  
100.000KHz  
perioda:  
10.000000us
```

Obrázek 17: Terminál měření frekvence