

Obsah

1	Program	2
1.1	Stavový automat vysielateľ/príjmač	2
1.2	Stavové automaty kanálov	3
1.3	Hlavičkové súbory	4
1.4	.C Súbory	4
1.5	Postup merania	5
1.6	Postup v prípade automatického aktualizácie parametrov transformácie	5
2	Nastavenia	7
2.1	Transformácia merania	7
2.2	Nastavenie prahového napätia a časovačov merania	7
2.3	Nastavenie Hold-off časovačov	8
3	Delič napätia	9
4	Používateľská príručka	12
4.1	Používateľské rozhranie	12
4.2	Spustenie merania	14
5	Vyhotovenie	15
6	Literárne zdroje, použitý software a použité knižnice	17

1 Program

Program je tvorený 3 stavovými automatmi, ktorých činnosť je previazaná systémom udalostí. Každý stavový automat má štruktúru udalostí s príznakmi, ktoré sa nastavujú v prípade, že udalosť nastane. V hlavnej slučke programu, sa vykonáva aktualizácia jednotlivých stavových automatov. Túto aktualizáciu vykonáva funkcia `*_update`, kde `*` môže byť buď `OSCI_transceiver` (príjmač a vysielač) alebo `OSCI_channel_update` (stavový automat riadiaci činnosť kanála). V krátkosti opíšeme činnosť týchto stavových automatov, pre detaily je vhodné konzultovať zdrojový kód.

1.1 Stavový automat vysielač/príjmač

Úlohou tohto stavového automatu, je reakcia na príchod nových dát, distribúcia nových nastavení a posielanie nových meraní do užívateľského prostredia. Jednoduché zobrazenie môžeme vidieť na obr. 1.

Počiatkový stav stavového automatu je *IDL – IDLE*, v tomto stave čaká na požiadavku z GUI, alebo na požiadavku od stavových automatov kanálov, na zaslanie nových dát.

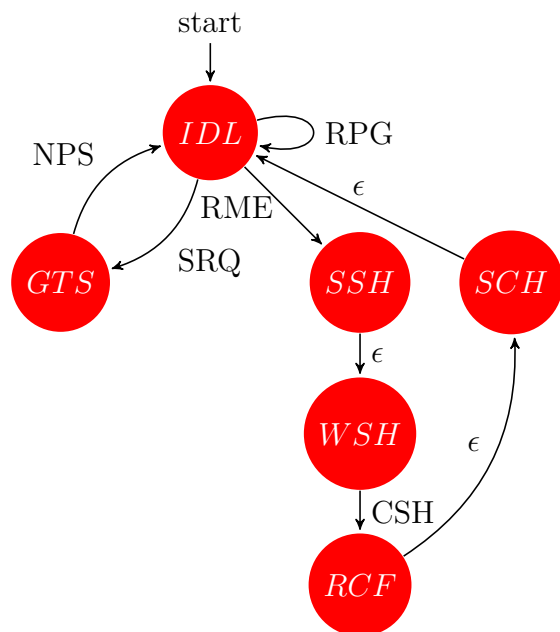
Cez perifériu UART sa pri požiadavke z GUI prijmu nové nastavenia. Prerušenie DMA TC pri prijatí, celej štruktúry nastavení, nastaví príznak stavovému automatu vysielač/príjmač. Tento príznak zodpovedá požiadavke na vykonanie merania (RME), načo sta-

vový automat prejde do stavu SSH - Shutting down channels. V stave SSH nastaví príznaky kanálom, ktoré reagujú ukončením meraní a prejdú do stavu SHD - shutdown. Vysielač/prijmač čaká, až kým sa kanály nedostanú do stavu SHD v stave WSH - waiting for channels to shutdown. Potom prejde do stavu RCF - reconfiguring channels. V tomto stave sa zmenia parametre transformácie dát, kanály dostanú nové nastavenia a stavový automat prepne do stavu SCH - starting channels. V tomto novom stave sa nastaví príznak stavovým automatom kanálov, aby začali kanály monitorovať, čo znamená, že sa AD prevodníky spustia v Continuous stave a čaká sa na prerušenie Watchdogu. Nakoniec sa vysielač/príjmač vráti do stavu IDLE, v ktorom zostane pokiaľ nedostane ďalšiu požiadavku na meranie alebo iný typ požiadavky.

V prípade, že sa vykoná meranie, stavový automat daného kanála, nastaví príznak stavovému automatu vysielač/príjmač, aby vykonal transformáciu nameraných dát a poslal tieto nové dáta do GUI. Stavový automat vysielač/príjmač, reaguje na tento príznak v stave IDLE a prejde do stavu GTS - gathering transforming and sending, v tomto stave sa vykoná už spomínaná transformácia dát a ich zaslanie do GUI. Vysielač/príjmač sa nakoniec vráti do stavu IDLE potom čo vyriešil všetky požiadavky na odoslanie dát, keďže tie sú rozdielne pre X a Y kanál.

Požiadavka od GUI, môže byť aj požiadavkou o vykonanie transformácie,

v prípade ktorej, sa zmenia parametre priamo v stave IDLE a vygeneruje sa požiadavka na zaslanie starého merania s novou transformáciou.



Obr. 1: Stavový stroj vysielateľ/príjmač

IDL	idle
GTS	gathering, transforming and sending data
SSH	starting channel shutdown
WSH	waiting for channels to shutdown
RCF	reconfiguring channels
SCH	starting channels
RPG	requested pong
RME	requested measurement
SRQ	send request
NPS	no pending send
CSH	channel shutdown
ϵ	unconditionally

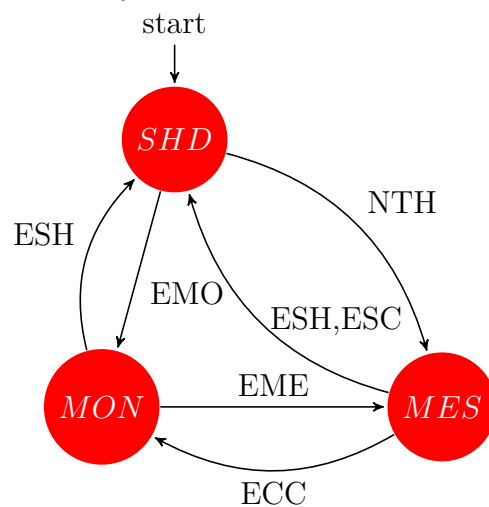
Tabuľka 1: Vysvetlenie skratiek stavového stroja príjmač/vysielač

Ešte pred meraním, je potrebné vy-

tvoriť spojenie, ktoré slúži na overenie nahraného programu. V takomto prípade, je vysielateľ/príjmač v stave IDLE, z ktorého rovno pošle PONG bez prechodu do iného stavu.

1.2 Stavové automaty kanálov

Úlohou týchto stavových automatov je zastavovanie a spúšťanie merania. Štruktúra je zobrazená na obr. 2.



Obr. 2: Stavový stroj meracieho kanálu

SHD	shutdown
MES	measuring
MON	monitoring
ESH	event shutdown
EMO	event monitoring
EME	event measuring
ESC	event single type measurement complete
ECC	event continuous type measurement complete
NTH	no voltage threshold set

Tabuľka 2: Vysvetlenie skratiek stavového stroja meracieho kanálu

Počiatkový stav týchto stavových automatov je stav SHD - shutdown. Pri nastavení príznaku začatia monitorovania kanála, čo nastavuje stavový automat vysielateľ/prijímač, sa nastaví a spustí AD prevodník v móde Continuous, čo pravidelne vzorkuje kanál a automat prejde do stavu MON - monitoring. V tomto stave sa čaká na prerušenie Watchdogu. Ktoré sa vyvolá, pokiaľ je napätie na kanály nad stanovenou hranicou, nastavenou v GUI. V tomto prerušení, sa hneď spúšťa meranie a automat prechádza do stavu MES - measuring. V stave MES ostáva, pokiaľ sa meranie nedokončí, čo sa určí na základe TC prerušenia DMA. V tomto prerušení sa nastaví príznak measurement complete, načo stavový automat reaguje buď prechodom do stavu SHD a nastavením príznaku SRQ - send request stavového automatu vysielateľ/prijímač. V prípade jednorázového merania v tomto stave ostáva, ale v prípade Continuous merania sa v prerušení Hold-off časovača kanál znova prevedie do stavu MON - monitoring.

1.3 Hlavičkové súbory

Program sa skladá z niekoľkých zdrojových, hlavičkových a .c súborov. Hlavičkové súbory slúžia ako miesto deklarácie funkcií, pričom všetky nebudeme opisovať, opíšeme len tie, v ktorých sa nachádzajú dôležité konštanty, premenné a dátové štruktúry.

osci_defines.h - nachádzajú sa tu kalibračné faktory, počiatkové nastavenia, enumerácie opkódov správ do GUI, enumerácia kanálov, makrá na MIN a MAX, definícia TRUE a FALSE, nie-

ktoré definované maximálne hodnoty unsigned registrov, enumerácia typov stavového stroja.

osci_data_structures.h - tento súbor obsahuje definície dátových štruktúr stavových automatov a rôznych abstrakcií ako napríklad meranie kanálu, čo predstavuje postupnosť vzoriek, dátové štruktúry nastavení poslaných z GUI, nastavení kanálov, udalostí jednotlivých stavových automatov, dátovú štruktúru aplikácie a dátovú štruktúru opisujúce parametre prechodu stavového automatu.

osci_channel_state_machine.h - definícia stavov automatu, deklarácia callbackov, init a update funkcií stavového automatu kanálu.

osci_transceiver.h - definícia stavov automatu, deklarácia callbackov, init a update funkcií stavového automatu vysielateľ/prijímač.

1.4 .C Súbory

Okrem inicializačných súborov periférií a iných štandardných súborov vygenerovaných pomocou CubeMX sa tu nachádzajú nasledovné súbory.

osci.c - obsahuje funkciu, ktorá alokuje pamäť a inicializuje dátové štruktúry aplikácie

main.c - volanie inicializačných funkcií, nastavenie globálnej referencie pre prerušenia a spustenie nekonečného cyklu update funkcií stavových automatov

osci_adc.c - pomocné funkcie, na ovládanie AD prevodníka.

osci_state_machine.c - všeobecné funkcie pre stavové automaty

osci_channel_state_machine.c - implementácia stavového automatu meracieho kanálu.

osci_configurator.c - funkcie na výpočet a prepočet parametrov kanálov, teda časovačov a transformácie merania.

osci_dma.c - funkcie na jednoduchšiu obsluhu dma.

osci_error.c - vo všeobecnosti debugovacie funkcie.

osci_timer.c - pomocné funkcie na obsluhu časovačov.

osci_transceiver.c - implementácia stavového automatu vysieláč/prijmač.

osci_transform.c - aplikácia transformácie podľa vypočítaných parametrov.

1.5 Postup merania

Pre lepšiu názornosť funkcie programu, opíšeme ešte celkový reťazec udalostí, od stlačenia tlačidla merania až po vykreslenie nameraných dát. Pričom uvažujeme, že zariadenie je už pripojené, teda, že už prebehol ping-pong handshake.

Po stlačený tlačidla *MEASURE* sa prečítajú aktuálne nastavenia GUI, ktoré sa pošlú cez *UART* linku. *UART* prijímač generuje DMA požiadavky až kým sa neprijíme celá štruktúra nastavení, v prípade čoho DMA vygeneruje TC prerušenie daného kanálu (6). Obslužná funkcia prerušenia nastaví event stavovému automatu vysieláč/prijmač. Ten pri ďalšom volaní svojej update funkcie, deteguje event a rozhodne sa čo ďalej, na základe typu správy, ktorú dostal. Keďže uvažujeme, že užívateľ stlačil tlačidlo *MEASURE*, tak sa vykoná

nasledujúca sekvencia udalostí. Stavovým automatom kanálov sa pošle event na vypnutie a prejde do stavu *WAITING_FOR_SHUTDOWN*, stavový automat vysieláč/prijmač počká na prechod stavových automatov kanálov do režimu *SHUTDOWN* a prejde do stavu *RECONFIGURING_CHANNELS*, stavový automat vysieláč/prijmač prestaví parametre kanálov a prejde do stavu *STARTING_CHANNELS* v tomto stave stavový automat vysieláč/prijmač nastaví event stavovým automatom kanálov, že sa majú previesť do stavu monitoring, stavové automaty kanálov sa prevedú do stavu monitoring, čo predstavuje nastavenie continuous módu AD prevodníkov, ktoré začnú vzorkovať kanály a nastavenie watchdogov na úrovne poslané z GUI, ak nastane prerušenie daného watchdogu, nastaví sa časovače, DMA, AD prevodník a spustí sa meranie, stavový automat kanálu prejde do stavu *MEASURING*, po ukončení merania DMA generuje prerušenie TC, v ktorom sa pošle event stavovému automatu vysieláč/prijmač, že má poslať nové dáta, ten reaguje a prejde do stavu *GATHERING_TRANSFORMING_SENDING* zo stavu *IDLE* a pošle označené dáta, keďže dáta kanálov sa posielajú zvlášť.

1.6 Postup v prípade automatického aktualizácie parametrov transformácie

V prípade, že ide o požiadavku na zaslanie dát s inou transformáciou,

teda o tzv. `ONLY_TRANSFORM` typ správy, tak sa len prepočítajú parametre transformácií jednotlivých kanálov a registruje sa požiadavka na odoslanie oboch kanálov, stavový automat prejde do stavu *GATHERING_TRANSFORMING_SENDING*, v ktorom sa vykoná transformácia posledného merania a aktivuje sa DMA kanál pripojený na vysieláč UART-u, ktorý pošle jednotlivé transformované merania do GUI.

2 Nastavenia

2.1 Transformácia merania

Užívateľské prostredie očakáva dáta v normalizovanom formáte. Tento formát je prednastavený na 4095 úrovni. Kde 0 znamená zakreslenie na najnižšiu úroveň v grafickom prostredí a 4095 na najvyššiu. Pri posielaní dát to užívateľského rozhrania musí mikropočítač brať do úvahy nastavenia, ktoré si užívateľ zvolil a podľa toho transformovať body merania. Táto transformácia prebieha na základe rovn. (1).

$$v_n = \lceil \lfloor \frac{1}{s} v_\alpha + o \rfloor^{4095} \rceil_0 \quad (1)$$

kde v_n je hodnota posielaná do užívateľského prostredia, s je prepočítaná citlivosť daná rovnicou rovn. (2), o je prepočítaný posun daný rovnicou rovn. (3) a $\lfloor \cdot \rfloor^{4095}, \lceil \cdot \rceil_0$ sú funkcie definované podľa rovn. (4). Hodnota v_α je daná rovn. (5), kde α slúži na kalibráciu hodnoty v_{adc} preveden AD prevodníkom.

$$s = \frac{s_g d_g}{r_m} \quad (2)$$

$$o = o_g \frac{1}{s_g d_g} 4095 \quad (3)$$

$$\lfloor x \rfloor^{4095} = \begin{cases} x & \text{ak } x < 4095 \\ 4095 & \text{inak} \end{cases} \quad (4)$$

$$\lceil x \rceil_0 = \begin{cases} x & \text{ak } x > 0 \\ 0 & \text{inak} \end{cases} \quad (5)$$

kde hodnoty s_g, d_g sú hodnoty citlivosti a posunu poslané z užívateľského pro-

stredia v jednotkách $\lfloor \frac{V}{dielik} \rfloor$ a $[V]$. Parameter r_m je rozsah na ktorom boli dáta merané teda $r_m \in \{5, 10, 20\}$.

2.2 Nastavenie prahového napätia a časovačov merania

Okrem tejto transformácie je potrebné ešte prepočítať prahové napätie z voltov na úrovne príslušných watchdogov. Tento prepočet realizujeme vzorcom rovn. (6).

$$t = \lfloor \frac{t_g}{r_m} \frac{t_{max}}{\alpha} \rfloor \quad (6)$$

kde t je hodnota, ktorá sa zapisuje do threshold registrov watchdogov, t_g je prahová hodnota napätia zaslaná z užívateľského prostredia, daná vo $[V]$, t_{max} je maximálna hodnota threshold registra pre daný watchdog ($2^{12} - 1$ pre AWD1 a $2^8 - 1$ pre AWD2).

Tiež musíme vypočítať parametre časovačov pomocou rovn. (7).

$$psc = \lfloor \frac{c}{c_{max}} \rfloor \quad (7)$$

$$arr = \lfloor \frac{c}{psc + 1} \rfloor - 1$$

kde c je celkový počet taktov časovača obsiahnutých v nastavovanom časovom intervale v $[sec]$. Hodnotu c vypočítame pre časovače s taktom fHZ na základe rovn. (8).

$$c = \frac{f t_{pd} d}{n} \quad (8)$$

kde t_{pd} je časová základňa v jednotkách $\frac{sec}{dielik}$, d je počet dielikov, n je počet pretečení, ktoré majú za čas $t_{pd}d$ nastať.

2.3 Nastavenie Hold-off časovačov

V používateľskom rozhraní je možné nastaviť jednorazové (Single mode) alebo priebežné (Continuous mode) meranie každého kanála zvlášť. Pri jednorazovom meraní sa uskutoční práve jedno meranie po doručení požiadavky. Pošlú sa údaje z merania a zariadenie sa prepne do stavu čakania. Pri priebežnom

meraní sa spustí časovač (Hold Off timer), ktorý periodicky žiada o vykonanie merania. Períodu tohto časovača je možné nastaviť pomocou GUI. Rozlíšenie tohto časovača je 1ms, preto použijeme pevnú hodnotu preddeličky ($PSC = 31999$, pri 32MHZ impulz každú milisekundu) a počítadlo nastavíme podľa údajov z GUI ($ARR = (\text{hodnota z GUI}) - 1$).

3 Delič napätia

Na vstupe A/D prevodníka je delič napätia, ktorého parametre sú prepínateľné dvoma relátkami. Schéma zapojenia je zobrazená na obr. 3. Uvažujeme, že maximálne napätia na vstupe A/D prevodníka, by nemalo presiahnuť hodnotu $3,3V$. Pre vstupný rozsah $20V$ máme v obvode zapojené všetky odpory. Pre rozsah $10V$ je, pomocou relé, skratovaný odpor R_1 a pre rozsah $5V$ sú skratované odpory R_1 a R_2 . Odpory relé zanedbáme. Pomocou 2. Kirchhoffovho zákona a uvažovaním slučiek ako na obr. 4 dostaneme rovnice rovn. (9).

$$\begin{aligned} u_{adc1} &= u_{in1} \frac{R_4}{R_3 + R_4} \\ u_{adc2} &= u_{in2} \frac{R_4}{R_2 + R_3 + R_4} \\ u_{adc3} &= u_{in3} \frac{R_4}{R_1 + R_2 + R_3 + R_4} \end{aligned} \quad (9)$$

Navyše chceme obmedziť prúd odpormi. Z dejto podmienky potom vznikne rovnica rovn. (10). Tu považujeme napätia u_{adci} , $i = 1, 2, 3$ za rovnké, čo neskôr pridáme aj do predchádzajúcich rovníc.

$$i_{max} = \frac{u_{adc1,2,3}}{R_4} \quad (10)$$

V maticovom zápise rovn. (11).

$$\underline{V}\bar{R} = \bar{b} \quad (11)$$

potom máme rovn. (12).

$$\begin{pmatrix} 0 & 0 & u_{adc1} & (u_{adc1} - u_{in1}) \\ 0 & u_{adc2} & u_{adc2} & (u_{adc2} - u_{in2}) \\ u_{adc3} & u_{adc3} & u_{adc3} & (u_{adc3} - u_{in3}) \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{u_{adc1}}{i_{max}} \end{pmatrix} \quad (12)$$

Pre maximálnu presnosť prevodu uvažujeme rovn. (13), pri maximálnom napätí v danom rozsahu, teda ak rovn. (14). A taktiež nech platí rovn. (15).

$$u_{adc1} = u_{adc2} = u_{adc3} = 3,3V \quad (13)$$

$$\begin{aligned} u_{adc1} &= 5V \\ u_{adc2} &= 10V \\ u_{adc3} &= 20V \end{aligned} \quad (14)$$

$$i_{max} = 1 \times 10^{-3} \quad (15)$$

Potom riešením rovníc je rovn. (16).

$$\bar{R} = \underline{V}^{-1}\bar{b} \quad (16)$$

Hodnoty odporov sú v tab. 3.

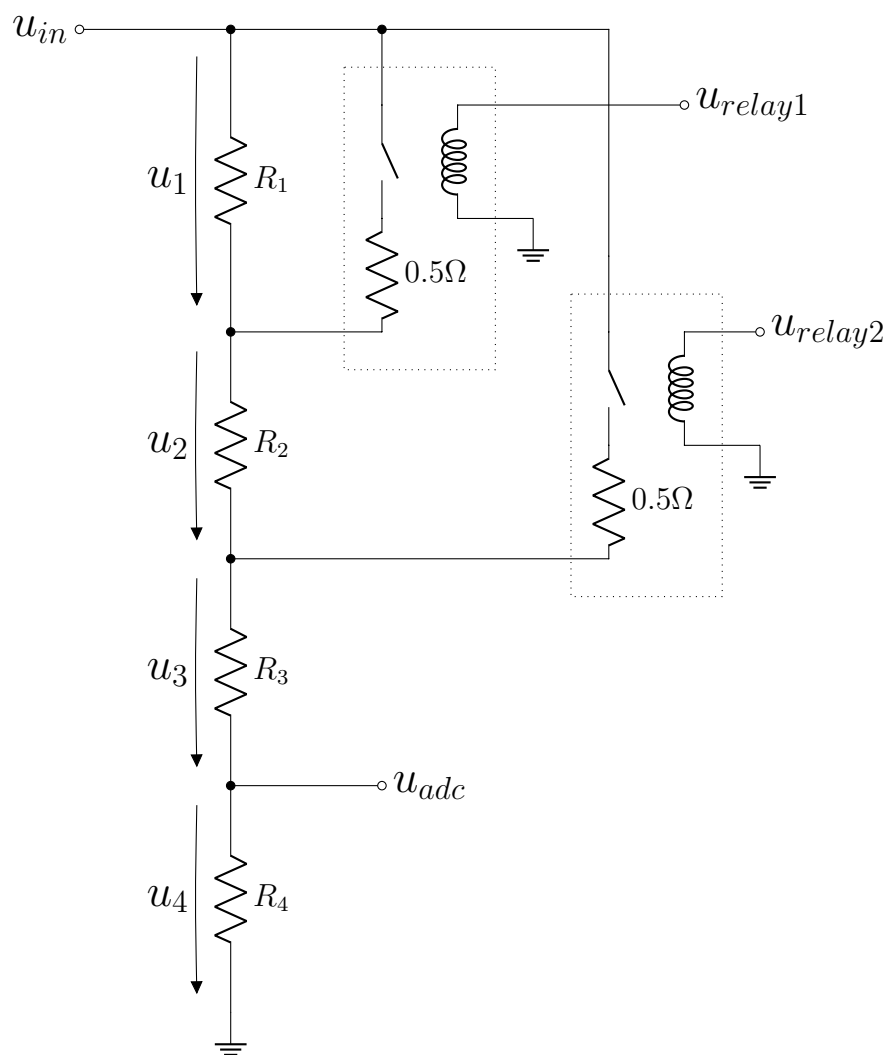
R_4	R_3	R_2	R_1
$3.3k\Omega$	$1.7k\Omega$	$5k\Omega$	$10k\Omega$

Tabuľka 3: Vypočítané hodnoty R

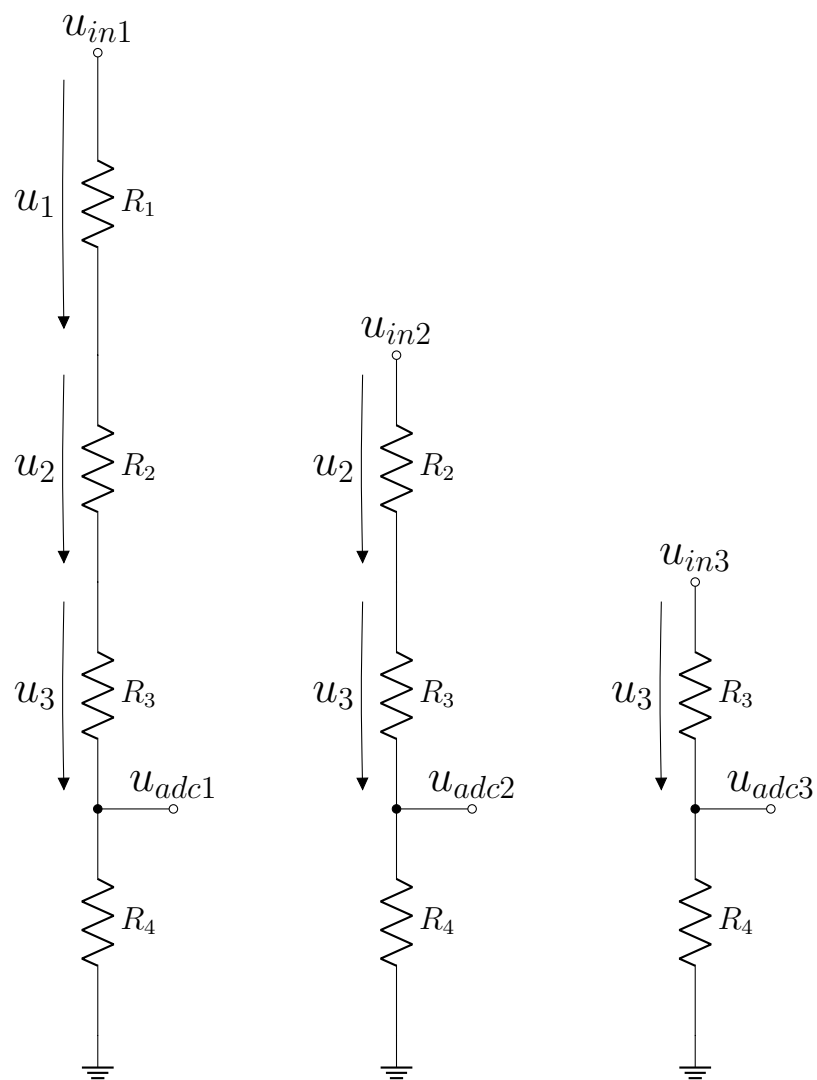
Po zaokrúhlení na štandardné hodnoty odporov, dostaneme hodnoty v tab. 4, pričom hodnota R_3 je tvorená dvoma odpormi $4.7k\Omega$ a $0.33k\Omega$.

R_4	R_3	R_2	R_1
$3.3k\Omega$	$1.8k\Omega$	$5.03k\Omega$	$10k\Omega$

Tabuľka 4: Zaokrúhlené hodnoty R



Obr. 3: Schéma zapojenia prepínateľných deličov napätia s vypočítanými a zarovnanými hodnotami napätí



(a) Slučka 2.KZ pre rozsah 20V (b) Slučka 2.KZ pre rozsah 10V (c) Slučka 2.KZ pre rozsah 5V

Obr. 4: Aplikácia 2. Kirchhoffovoého zákona na jednotlivé konfigurácie deliča

4 Používateľská príručka

4.1 Používateľské rozhranie

V krátkosti opíšeme časti používateľského rozhrania. Po spustení programu sa zobrazí okno na obr. 5. Vysvetlenie jednotlivých nastavení je v tab. 5. Začatím merania, je potrebné vytvoriť spojenie so zariadením. Najprv je treba prepnúť sa do okna *Protocol* vyznačené na obr. 5. Po prepnutí sa zobrazia nastavenia komunikácie na obr. 6, kde je potrebné vybrať správny sériový port, na ktorom je zariadenie pripojené a nastaviť vhodnú prenosovú rýchlosť. Po nastavení týchto parametrov je treba kliknúť na tlačidlo *Connect* a počkať (1sec) kým sa nezmení indikátor stavu spojenia

na zelenú alebo červenú. Spojenie prebehlo úspešne v prípade zelenej farby indikátora stavu spojenia.

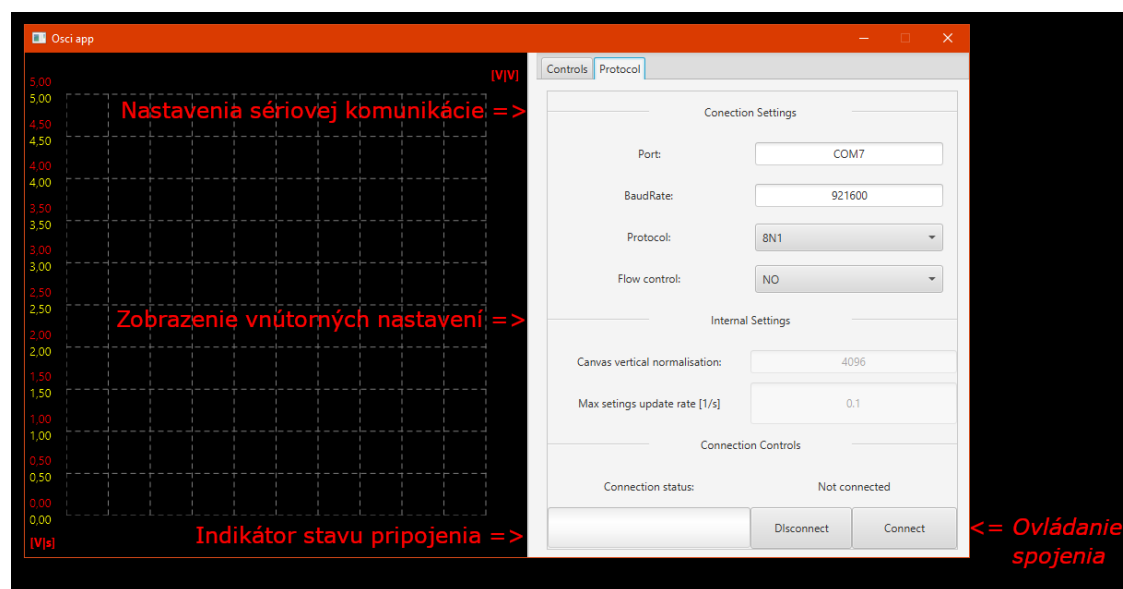
Meranie sa potom spúšťa tlačidlom *Measure*, ktorého stlačením sa zašlú aktuálne nastavenia do mikropočítača. V prípade, že žiadame aby sa citlivosť a posunutie kanálu aplikovala automaticky, je možné zaškrtnúť checkbox *AutoUpdate*. Zaškrtnutím *AutoUpdate* sa s frekvenciou 10Hz budú do mikropočítača posielat požiadavky o úpravu transformačných parametrov. Okrem *AutoUpdate* je možné takisto zaškrtnúť *AutoMeasure*, v prípade čoho sa budú automaticky aktualizovať aj časové základne a napäťové prahy.



Obr. 5: Úvodné okno používateľského rozhrania

Nastavenie	Jednotky	Vysvetlenie
Citlivosť	V/dielik mriežky	Určuje s akou citlivosťou sa budú dáta zobrazovať
Posun	V	Udáva posun, ktorý je pripočítaný k nameraným dátam pri zobrazovaní
Napäťový rozsah kanála	V	Určuje nastavenie vstupného deliča napätia
Metódy spúšťania merania	-	Buď <i>Single</i> - vykonanie jedného merania, alebo <i>Continuous</i> - pravidelné opakovanie merania
Hranica spustenia merania	-	Určuje hranicu napätia, prekročenie ktorej spustí meranie
Časový interval na dielik mriežky	sec	Určuje periódu vzorkovania pri meraní
Minimálny časový interval medzi meraniami	msec	Určuje minimálny časový interval medzi meraniami v <i>Continuous</i> móde

Tabuľka 5: Vysvetlenie nastavení



Obr. 6: Okno nastavení komunikácie

4.2 Spustenie merania

1. Zapnutie GUI
2. Prepnutie do záložky Protocol
3. Nastavenie parametrov protokolu
4. Kliknutie na tlačidlo Connect
5. Ak je ProgressBar naľavo zelený, spojenie prebehlo úspešne, inak nie a v takom prípade, je treba znova overiť, či boli zadané parametre správne.
6. Prepnutie naspäť na záložku Control
7. Nastavenie parametrov merania
8. V prípade, že vyžadujeme aj automatickú aktualizáciu parametrov transformácie, tak zaškrtneme CheckBox autoupdate.
9. Klikneme na tlačidlo Measure
10. Počkáme, až kým sa naľavo nezobrazia namerané priebehy.

5 Vyhotovenie

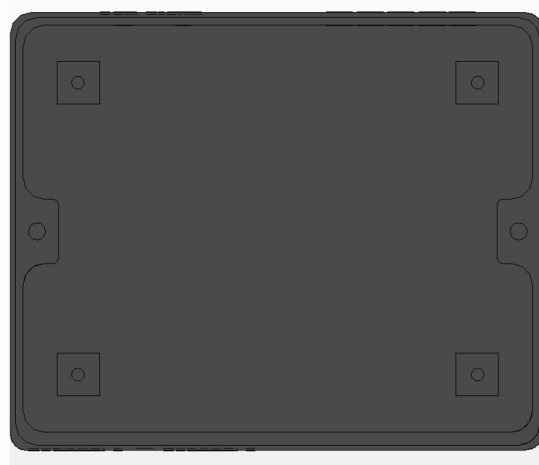
Zariadenie pozostáva z plošného spoja, na ktorom sú prispájkované odpory a sloty na relé a samotné STM. Tieto sloty umožňujú jednoduchú výmenu STM a relé. Celú elektroniku chráni krabička o rozmeroch 100x81.1x55mm. Na prednej strane krabičky sa nachádzajú štyri otvory na zapojenie dvoch kanálov. Na zadnej strane krabičky sa nachádza otvor na zapojenie Micro USB a dierky na vetranie.



Obr. 7: Model krabice - predná strana

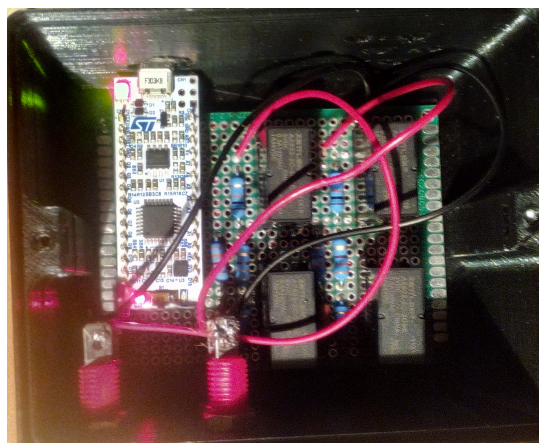


Obr. 8: Model krabice - zadná strana

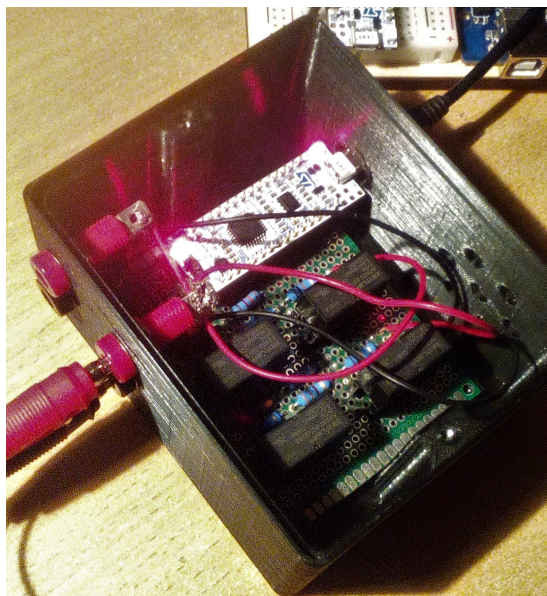


Obr. 9: Model krabice - zvrchu

Na obr. 7, obr. 8 a obr. 9 sú rôzne pohľady na model krabice. Model bol vytvorený v programe Solid Edge od Siemens. Po vyhotovení, bol do krabice vložený plošný spoj a do päťíc vložený mikropočítač STM32F303k8, takisto boli do päťíc vložené aj relé. Výsledný hardvér je na obr. 10 a obr. 11.



Obr. 10: Krabica - záber 1



Obr. 11: Krabica - záber 2

6 Literárne zdroje, použitý software a použité knižnice

V práci sme využívali nasledujúci software:

- Intelij IDEA Community Edition
- Solid Edge od Siemens (Študentská licencia)
- STM32CubeIDE

Literárne zdroje z ktorých sme čerpali:

- Oscilloscope fundamentals [Link](#)

V práci sme využili aj nasledujúce knižnice:

- jSerialComm (Platform-independent serial port access for Java) [Link](#)