

1 Program

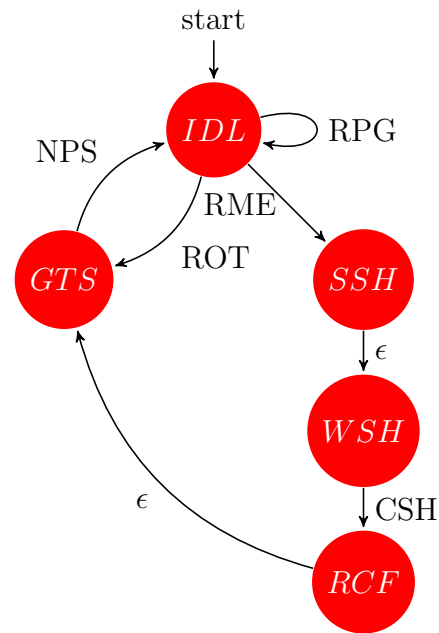
Program je tvorený 3 stavovými strojmi, ktorých činnosť je previazaná systémom udalostí. Každý stavový stroj má vlajky udalostí, ktoré sa nastavujú v prípade, že udalosť nastane. V hlavnej slučke programu, sa vykonáva aktualizácia jednotlivých stavových strojov. Túto aktualizáciu vykonáva funkcia **_update*, kde *** môže byť buď *Transceiver* (príjmač a vysielateľ) alebo *ChannelStateMachine* (stavový stroj riadiaci činnosť meracieho kanála). V krátkosti opíšeme činnosť týchto stavových strojov, pre detaily je vhodné konzultovať zdrojový kód. Schematické znázornenie programu je na konci tejto kapitoly na ??.

1.1 Stavový stroj vysielateľ/príjmač

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

Po príchode dát cez *UART* linku, sa na základe obsahu prijatej správy, vysielateľ rozhodne, či má poslať *pong* správu, zmeniť transformáciu dát, alebo prekonfigurovať stavový stroj kanálu na iný typ merania. Prekonfigurovanie prebieha vypnutím meraní, prepísaním parametrov a prevodom stavových strojov kanálov do stavu *MONITORING*, v ktorom *watchdog* obvody sledujú hodnoty prevodné AD prevodníkom. Ak tieto hodnoty prekročia definované limity spustí

sa meranie. Po vykonaní merania nastane udalosť, na ktorú stavový stroj vysielateľ/príjmač reaguje zaslaním nových dát do užívateľského prostredia.



Obr. 1: Stavový stroj vysielateľ/príjmač

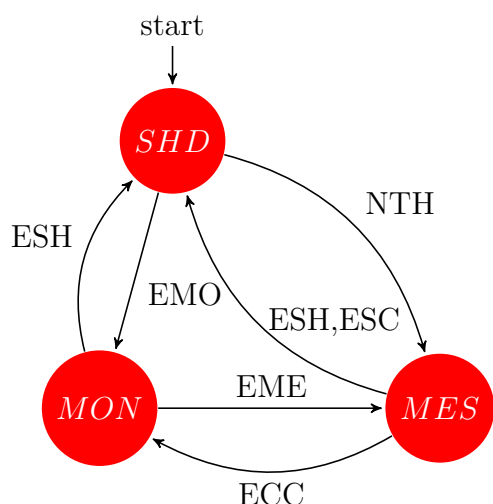
IDL	idle
GTS	gathering, transforming and sending data
SSH	starting channel shutdown
WSH	waiting for channels to shutdown
RCF	reconfiguring channels
RPG	requested pong
RME	requested measurement
ROT	requested only transform
NPS	no pending send
CSG	channel shutdown
ε	unconditionaly

Tabuľka 1: Vysvetlenie skratiek stavového stroja príjmač/vysielateľ

1.2 Stavový stroj meracieho kanála

Úlohou tohto stavového stroja je zastavovanie a spúšťanie merania. Štruktúra je zobrazená na obr. 2.

Bufre nastavení nastavuje vysielateľ/príjmač. Po príchode požiadavky na meranie nastane udalosť, ktorá prevedie stavový stroj meracieho kanála do stavu *SHUTDOWN*, zastavením časovačov a prevodov AD prevodníkov. Vysielač/príjmač reaguje na *SHUTDOWN* stav kanálov vo svojej stavovej slučke, a prevedie ich do stavu *MONITORING*. V stave *MONITORING* kanále vzorkujú svoj vstup a *watchody* porovnávajú prevedené hodnoty s nastavenými limitmi. V prípade, že sú limity určené pre začatie merania prekročené, začne sa meranie a kanál prejde do stavu *MEASURING*. V prerušení od *DMA* modulu, sa kanály prevedú znova *SHUTDOWN* módu a nastane udalosť, na ktorú vysielač bude reagovať zaslaním nových dát do užívateľského prostredia.



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

1.3 Hlavičkové súbory

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

osci_defines.h - nachádzajú sa tu kalibračné faktory, počiatočné nastavenia, enumerácie opkódov správ do GUI, enumerácia kanálov, makrá na MIN a MAX, definícia TRUE a FALSE, niektoré definované maximálne hodnoty unsigned registrov, enumerácia typov stavového stroja.

osci_data_structures.h - tento súbor obsahuje definície dátových štruktúr stavových strojov a rôznych abstrakcií ako napríklad meranie kanálu, čo predstavuje postupnosť vzoriek, dátové štruktúry nastavení poslaných z GUI, nastavení kanálov, udalostí jednotlivých stavových strojov, dátovú štruktúru ap-

likácie a dátovú štruktúru opisujúce parametre prechodu stavového stroja.

1.4 .C Súbor

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

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

osci_channel_state_machine.c - implementácia stavového stroja meracieho kanálu.

osci_configurator.c - v tomto zdrojovom súbore sú 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 stroja 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 stlačení tlačidla *MEASURE* sa prečítajú aktuálne nastavenia GUI,

ktoré sa pošlú cez *UART* linku. *UART* prijímač generuje DMA požiadavky až kým sa neprijme celá štruktúra nastavení, v prípade čoho DMA vygeneruje TC prerušenie daného kanálu (6). Obslužná funkcia prerušenia nastaví event stavovému stroju 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 strojom kanálov sa pošle event na vypnutie a prejde do stavu *WAITING_FOR_SHUTDOWN*, stavový stroj vysieláč/prijmač počká na prechod stavových strojov kanálov do režimu *SHUTDOWN* a prejde do stavu *RECONFIGURING_CHANNELS*, stavový stroj vysieláč/prijmač prestaví parametre kanálov a prejde do stavu *STARTING_CHANNELS* v tomto stave stavový stroj vysieláč/prijmač nastaví event stavovým strojom kanálov, že sa majú previesť do stavu monitoring, stavové stroje kanálov sa prevedú do stavu monitoring, čo predstavuje nastavenie continuous módu ADC prevodníkov, ktoré začnú vzorkovať kanály a nastavenie watchdogov na úrovne poslané z GUI, ak nastane prerušenie daného watchdogu, nastaví sa časovač, DMA, AD prevodník a spustí sa meranie, stavový stroj kanálu prejde do stavu *MEASURING*, po ukončení merania DMA generuje prerušenie TC, v ktorom sa pošle event stavovému stroju vysieláč/prijmač, že má poslať nové dáta, ten reaguje a prejde do stavu *GATHERING*.

RING_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 stroj prejde do stavu *GATHERING_TRANSFORMING_SENDING*, v ktorom sa vykoná transformácia posledného merania a aktivuje sa DMA kanál pripojený na vysielateľ 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.

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

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

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

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

$$v_\alpha = v_{adc} \alpha$$

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

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

2.2 Nastavenie 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).

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

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

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

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

kde t_{pd} je časová základňa v jednotkách $\frac{sec}{\text{dielik}}$, 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 ($ARR = (\text{hodnota z GUID}) - 1$).

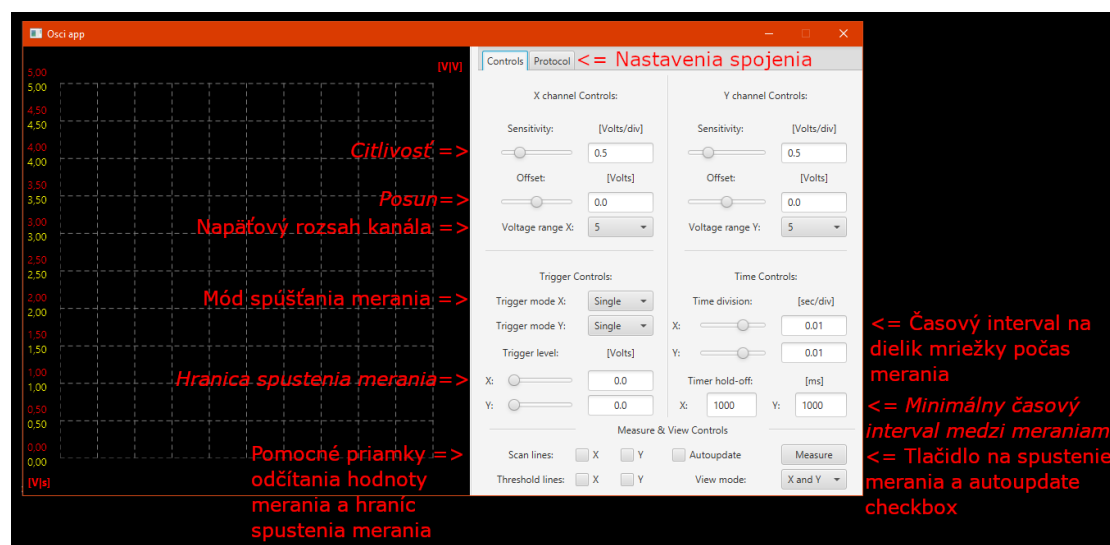
3 Používateľská príručka

3.1 Používateľské rozhranie

V krátkosti opíšeme časti používateľského rozhrania. Po spustení programu sa zobrazí okno na obr. 3. Vysvetlenie jednotlivých nastavení je v tab. 3. Začatím merania, je potrebné vytvoriť spojenie so zariadením. Najprv je treba prepnúť sa do okna *Protocol* vyznačené na obr. 3. Po prepnutí sa zobrazia nastavenia komunikácie na obr. 4, kde je potrebné vybrať správny sériový port, na ktorom je zariadenie pripojené a nastaviť vhodnú prenosovú rýchlosť. Po nastavení týchto parametrov je treba klik-

núť 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.

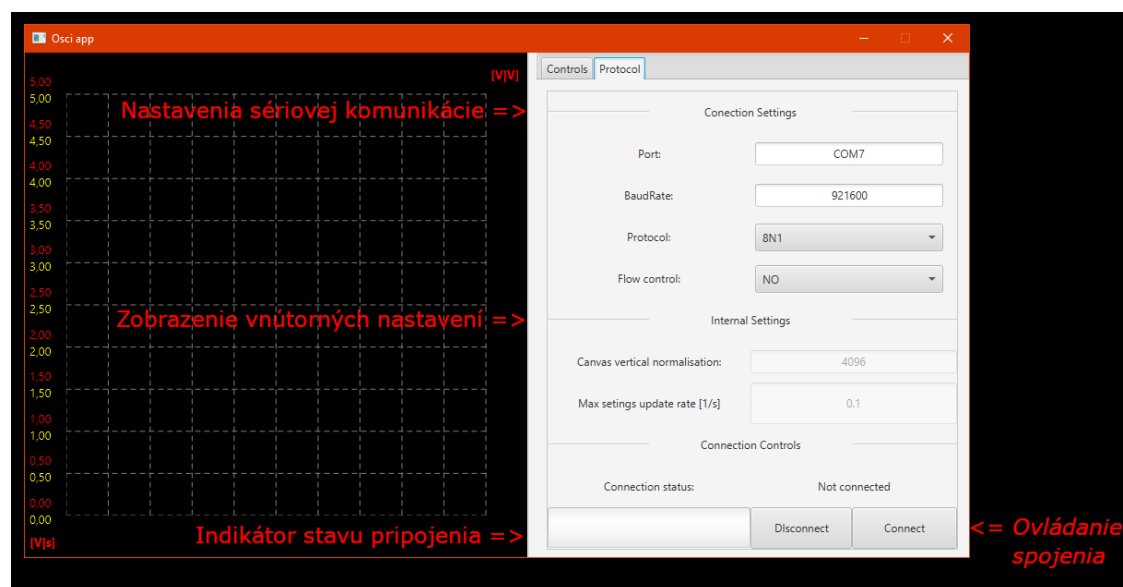
Meranie sa potom spúšťa tlačidlom *Measure*, ktorého stlačením sa zašlú aktuálne nastavenia do mikropočítača. V prípade, že žiadame aby sa citlivosť a posunutie kanálu aplikovala automaticky, je možné zaškrtnúť checkbox *autoupdate*. Zaškrtnutím *autoupdate* sa s frekvenciou 10HZ budú do mikropočítača posielat požiadavky o úpravu transformačných parametrov.



Obr. 3: 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
Napätový 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 3: Vysvetlenie nastavení



Obr. 4: Okno nastavení komunikácie

3.2 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.

4 Delič napätia

Na vstupe A/D prevodníka je delič napätia, ktorého parametre sú prepínateľné dvoma relátkami. Schéma zapojenia je zobrazená na obr. 5. 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. Kirchhoffového zákona a uvažovaním slučiek ako na obr. 6 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).

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

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).

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

$$\begin{aligned}
u_{adc1} &= 5V \\
u_{adc2} &= 10V \\
u_{adc2} &= 20V
\end{aligned} \tag{14}$$

$$i_{max} = 1 \times 10^{-3} \tag{15}$$

Potom riešením rovníc je rovn. (16).

$$\overline{R} = \underline{V}^{-1} \overline{b} \tag{16}$$

Hodnoty odporov sú v tab. 4.

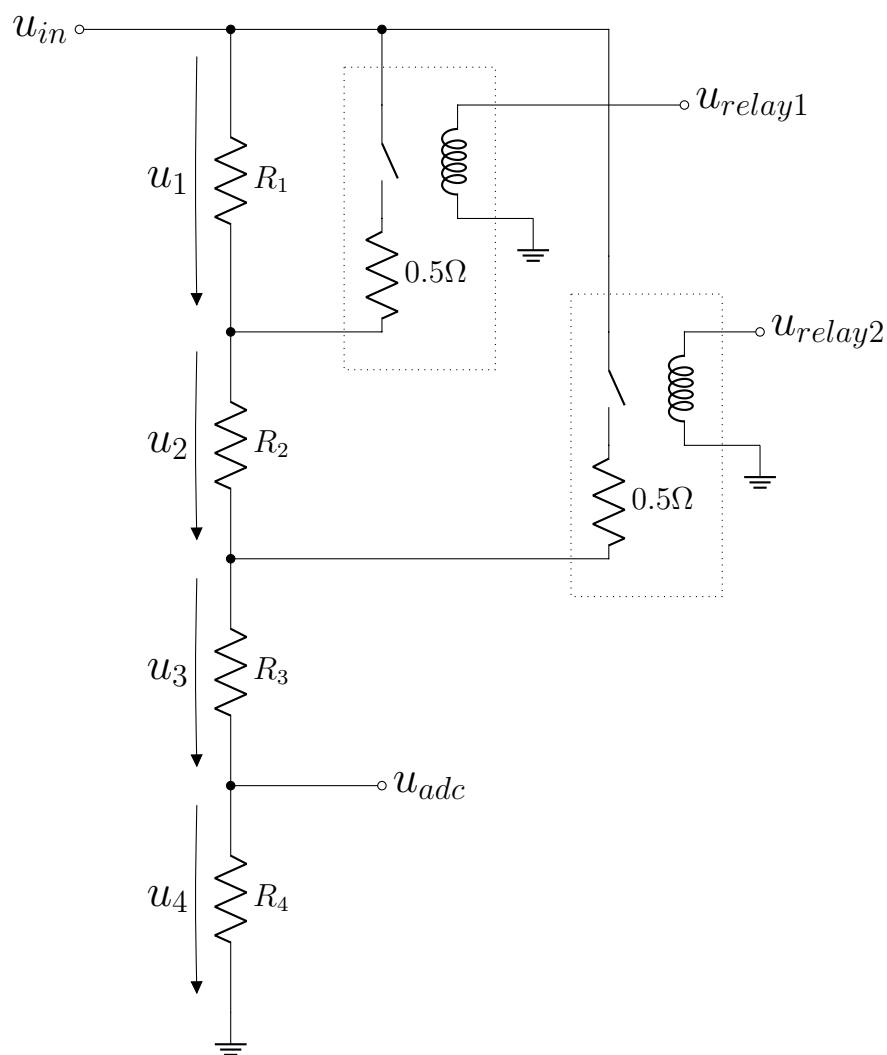
R_4	R_3	R_2	R_1
$3.3k\Omega$	$1.7k\Omega$	$5k\Omega$	$10k\Omega$

Tabuľka 4: Vypočítané hodnoty R

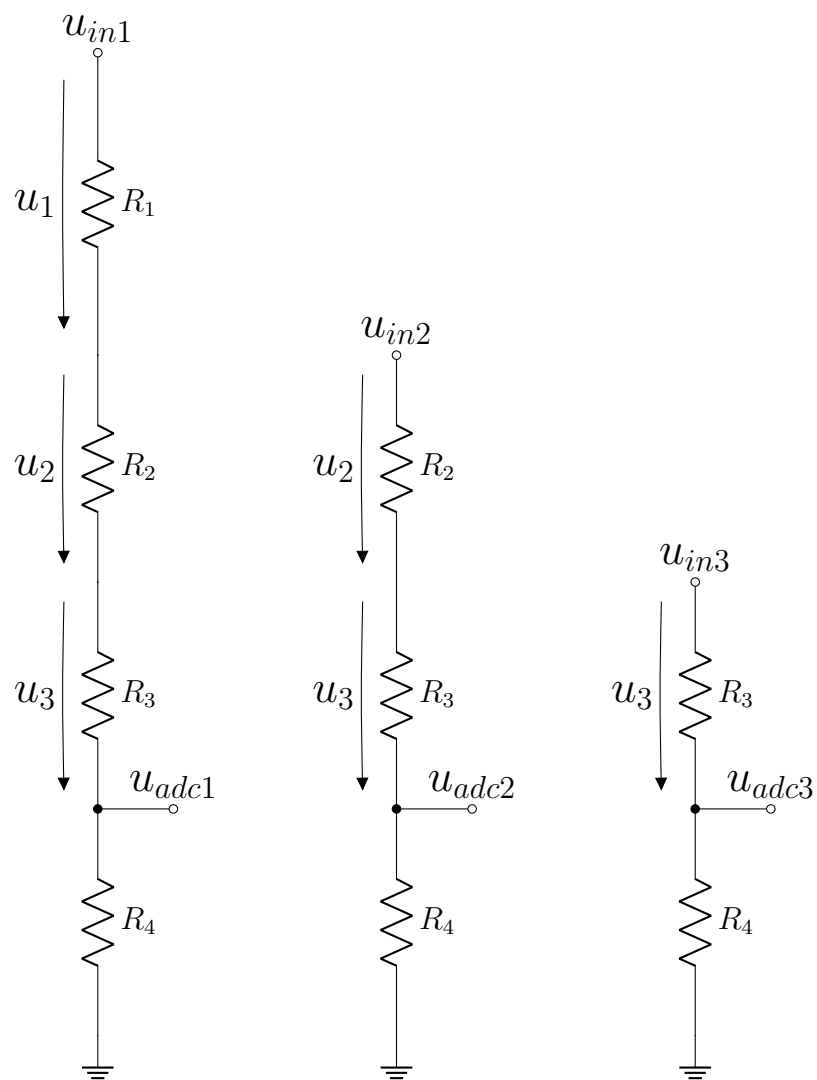
Po zaokrúhlení na štandardné hodnoty odporov, dostaneme hodnoty v tab. 5, pričom hodnota R_3 je tvorená dvoma odpormi $4.7k\Omega$ a $0.33k\Omega$.

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

Tabuľka 5: Zaokrúhlené hodnoty R



Obr. 5: 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. 6: Aplikácia 2. Kirchhofovoého zákona na jednotlivé konfigurácie deliča

5 Vyhotovenie

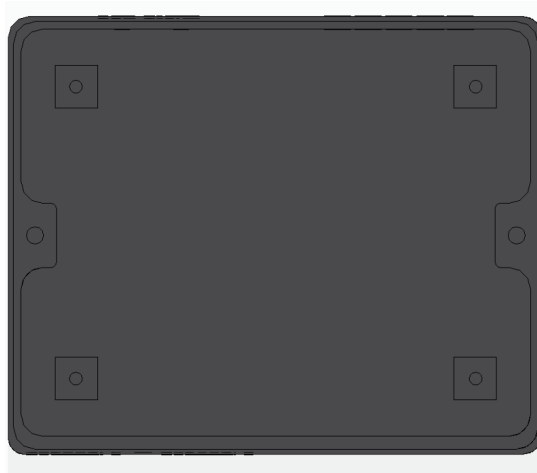
Zariadenie pozostáva z plošného spoja, na ktorom sú prispájkované odpory a sloty na relé a samotné STM. Tieto sloty umožňujú jednoduchú výmenu STM a relé. Celú elektroniku chráni krabička o rozmeroch 100x81.1x55mm. Na prednej strane krabičky sa nachádzajú štyri otvory na zapojenie dvoch kanálov. Na zadnej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

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](#)