

Žilinská univerzita
Katedra mechatroniky a elektroniky

Návrh elektronických zariadení

Meranie času s možnosťou merania medzičasov a výpis na displej

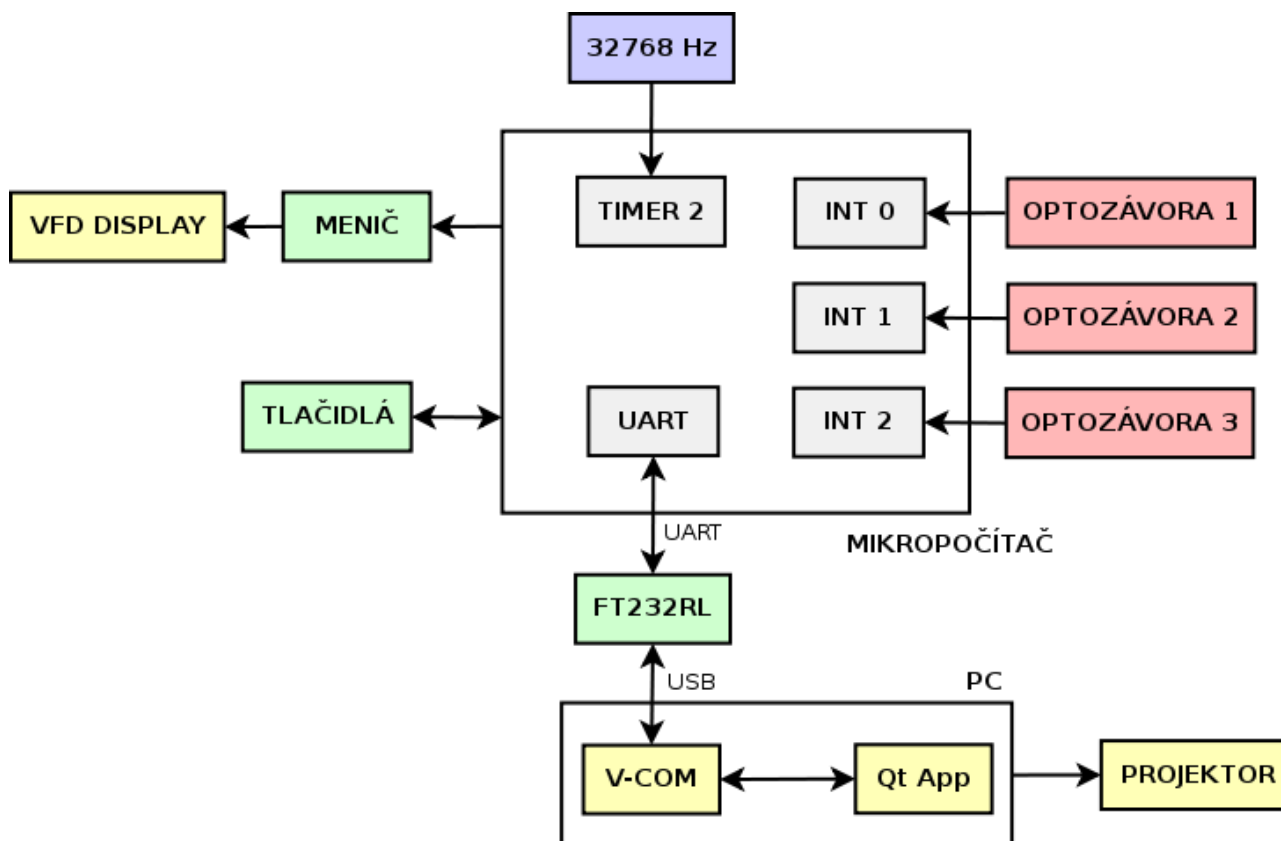
1 Úvod

Úlohou zadania je navrhnuť a skonštruovať zariadenie pre meranie času s pridanou možnosťou merania medzičasov. Zobrazenie meraného času sa bude vykonávať na displej a takisto bude zabezpečené posielanie údajov o čase do PC cez sériovú linku so zobrazením v programe s grafickým užívateľským prostredím GUI.

Pre presné meranie času sa využije časovanie kryštálom, ktorý využíva väčšina súčasných hodín či už digitálnych alebo analógových, pripojený k mikropočítaču na jednotku čítača. Dôležitou problematikou tejto práce a jej riešenie bude taktiež pripojenie 32 miestneho vákuového displeja kde každé zobrazované miesto obsahuje 16 zobrazovacích segmentov. Keďže zariadenie má byť využité ako stopky k autodráhe tak spúšťanie merania času a stopnutie sa bude vykonávať pomocou optických brán resp. optozávor. Optozávory budú využité aj pri stopnutí medzičasov.

2 Koncepcia zariadenia pre meranie času

Bloková schéma navrhovaného zariadenia je na obrázku Obr. 2.1. Ako vidno základnou časťou je mikropočítač, ktorý bude riadiť činnosť zariadenia.



Obr. 2.1 Bloková schéma zariadenia pre meranie času

Vstupné časti zariadenia budú tvoriť:

- kryštál 32,768 kHz pripojený na vstup čítača/časovača2;
- optozávory pripojené na vývody s možnosťou extérneho prerušenia na logickú úroveň.

Výstupné časťami zariadenia sú:

- VFD (vacuum fluorescent display) displej.

Vstupno – výstupné časti:

- tlačidlá pre nastavenie zariadenia,
- USB rozhranie pre pripojenie osobného počítača s programom s GUI; pripojenie sa vykoná s jednotkou UART.

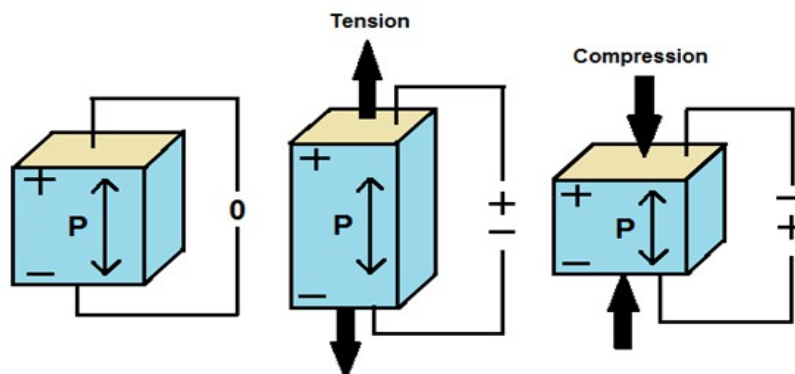
3 Návrh jednotlivých častí zariadenia

V tejto kapitole bude opísaný vývoj jednotlivých vstupných a výstupných častí zariadenia pre meranie času.

3.1 Meranie času

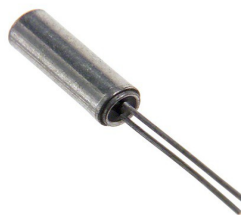
V súčasnosti sa pre meranie času v bežných hodinách využíva časovanie kremenným kryštálom typicky s frekvenciou generovania kmitov 32768 Hz. Ide o kryštál kremeňa, ktorý využíva vlastnosť piezoelektrického javu.

Piezoelektrický jav je jav, ktorý vzniká pri deformácii kryštálu, pričom jeho výsledkom je generovanie elektrického náboja na jeho povrchu. Princíp jeho vzniku je ukázaný aj na obrázku Obr. 3.1.



Obr. 3.1 Princíp piezoelektrického javu

Pri generovaní impulzov sa však vyžíva obrátený piezoelektrický jav, kedy prechodom prúdu cez kryštál sa tento kryštál deformuje a rozkmitá na požadovanú frekvenciu. Takto sa generujú kmity potrebné pre presné časovanie.

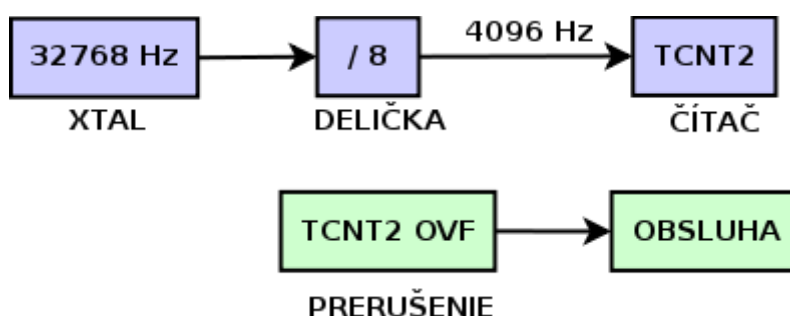


Obr. 3.2 Púzdro s kremenným kryštálom 32,768 kHz

Takýto kryštál je pripojený k mikropočítaču Atmega16 na vstup čítača 2, ktorý sa nachádza na vývodoch portu C, označených PC7 (TOSC2) a PC6 (TOSC1). Frekvencia z oscilátora je najprv delená deličkou 8 z čoho vyplýva, že výsledná čítaná frekvencia je 4096 Hz. Takouto frekvenciou je inkrementovaný 8-bitový čítač TCNT2. Po pretečení tohto čítača (teda po načítaní 256 impulzov) sa generuje prerušenie od pretečenia. Takého prerušenie nastáva každých 62,5 ms, z čoho vyplýva presnosť merania času na 6,25 stotiny sekundy.

V obsluhu prerušenia sa vykoná inkrementácia premennej, ktorá nesie informáciu o sekundách, teda k tejto premennej sa pripočíta hodnota 0,0625. V obsluhu sa ešte vykoná konverzia premenných nesúcich informáciu o sekundách a minútach na tvar v ASCII znakoch do tzv. buffer-a, z ktorého sa budú dáta zobrazovať na displej a vysielat' na UART do osobného počítača pre projekciu. Formát tohto reťazca ASCII znakov je nasledovný:

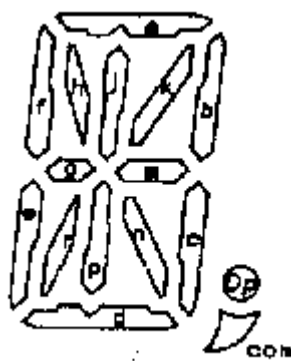
MM:SS:ss , kde MM sú minúty (s úvodnou nulou na dve miesta), SS sekundy (s úvodnou nulou na dve miesta), ss stotiny sekúnd. Napríklad: 02:45:78 značí, že aktuálny čas je teda 2 minúty, 45 sekúnd a 78 stotin sekundy.



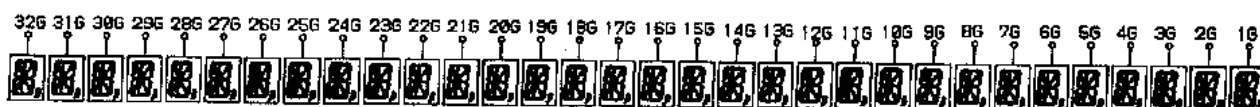
Obr. 3.3 Znázornenie merania času

3.2 Pripojenie VFD displeja

Jednou z foriem pre zobrazenie aktuálneho času a medzičasov bol zvolený spôsob výpisu na displej. Ako displej sme využili VFD displej (vacuum fluorescent display) s označením 32-SY-03ZL, ktorý obsahuje 32 zobrazovacích miest pričom každé zobrazované miesto obsahuje 16 segmentov, podľa obrázka Obr. 3.4.



Obr. 3.4 Jedno zobrazovacie miesto VFD displeja



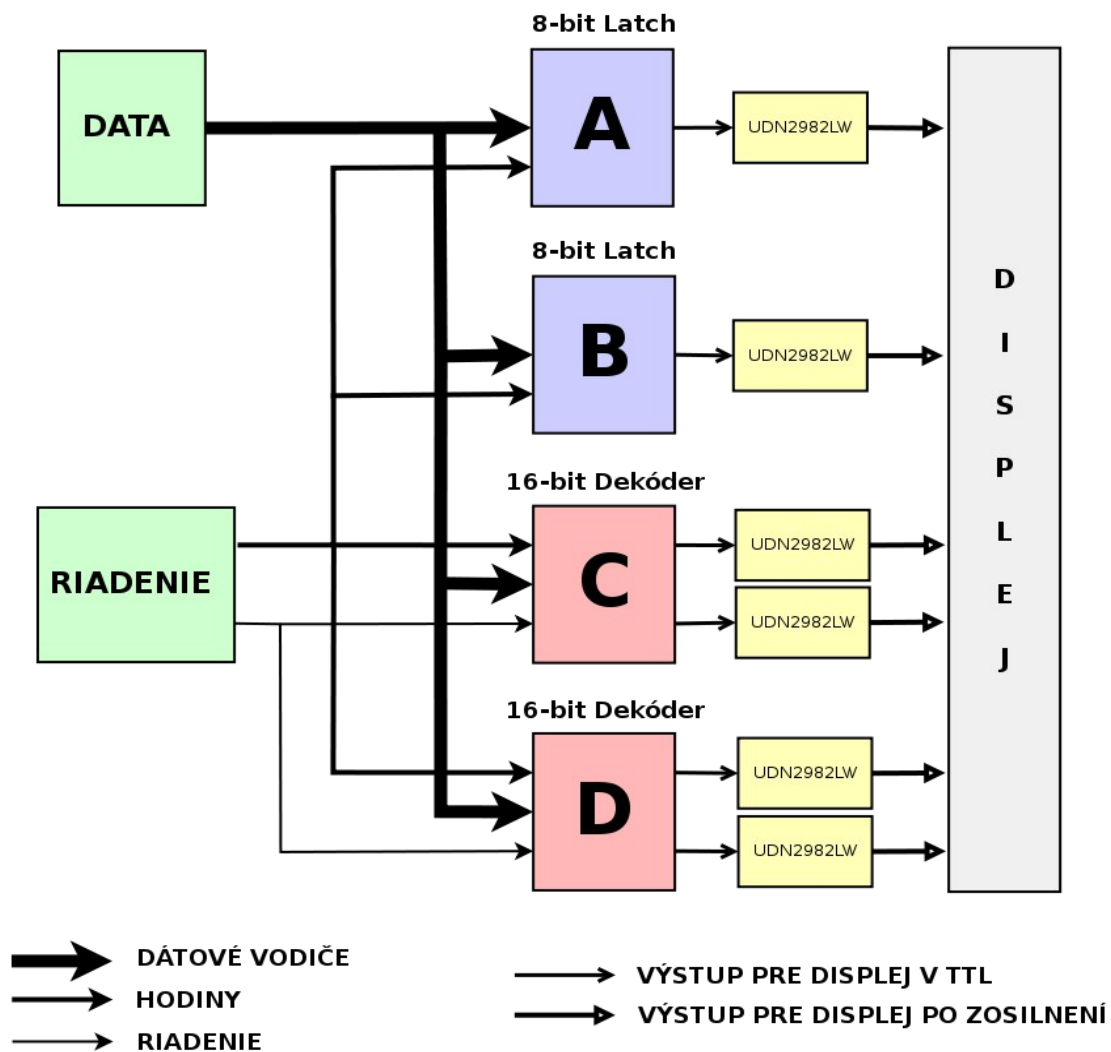
Obr. 3.5 Rozmiestnenie zobrazovacích miest VFD displeja

Segmenty s tým istým označením na jednotlivých miestach sú spolu prepojené to znamená že displej obsahuje 16 vývodov, ktorými sa vyberá, ktoré segmenty majú práve svietiť. Aby sme zabezpečili, že dané segmenty budú svietiť na požadovanom zobrazovacom mieste displeja, tak každé miesto obsahujú tzv. mriežku (grid), ktorá po privedení napätia s určitou úrovňou, povolí rozsvietiť vybrané segmenty. Riadiaci program bude musieť teda zabezpečiť spínanie jednotlivých mriežok s potrebnou frekvenciou tak, aby bol zabezpečný statický obraz. Frekvencia však nesmie byť príliš vysoká pretože platí: čím vyššia frekvencia prepínania, tým menší jas displeja.

Pre zobrazovanie jednotlivých segmentov sa využívajú vyššie úrovne napätia na jeho vývodoch (40 V) ako je úroveň napätia ktoré využíva mikropočítač (5 V), preto bolo potrebné navrhnuť jeho riadiacu časť s posilnením výstupu pre displej.

Pre funkčnosť displeja sa pripája aj napätie pre rozžhavenie vlákna, ktoré býva väčšinou striedavé s úrovňou 7 V_{šš}. Striedavé napätie sa využíva preto, aby bol zabezpečný rovnaký jas displeja, po celej jeho zobrazovacej dĺžke. V našom prípade sme nepoužili striedavé napájanie pre žhavenie ale symetrické napätie $\pm 3,5V$, ktoré je tvorené DC/DC meničom pre $\pm 5V$ a znížené na $\pm 3,5$ úbytkom na diódach zapojených do série.

Pri návrhu riadenia displeja sme vychádzali z blokovej schémy zobrazenej na obrázku Obr. 3.6. Vstupom do riadenia sú dve 8-bitové porty, kde jeden z nich je označený ako dátový (privádzajú sa cez neho zobrazované dáta) a druhý je označený ako riadiaci pretože umožňuje riadenie jednotlivých blokov ako sú hodiny a povolenie funkcie daného bloku.



Obr. 3.6 Bloková schéma riadenie displeja

D	N	P	COM	R	E	G	F
7	6	5	4	3	2	1	0

Tab. 3.1 Register A

A	B	K	J	H	M	C	DP
7	6	5	4	3	2	1	0

Tab. 3.2 Register B

V riadení sme využili dva 8-bitové záchytné registre tzv. latch-e (A a B). Umožňujú zapamätať si priviesť na ich výstup hodnotu vstupu. Hodnota vstupu sa prevedie na výstup po privedení hodinového signálu. Tieto zachytné registre sa využívajú na zopnutie jednotlivých segmentov displeja a ich výstup je posilnený obvodom UDN2928, na ktorého výstupe sú logické stavy s úrovňou 40 V. Výstupy týchto obvodov sme ošetrili rezistorovou sieťou tak aby v stave vysokej impedancie boli stiahnuté na logickú úroveň 0,

čo značí, že všetky segmenty sú zhasnuté.

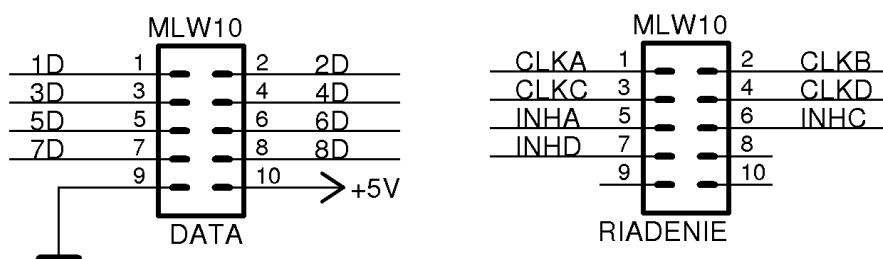
V zapojení sa ďalej vyžívajú dva 16-bitové dekódery (C a D) z dvojkového stavu na desiatkový. Pomocou nich sa vyberá príslušné miesto na displeji, ktorého segmenty majú byť práve rozsvietené. Ich výstupy sú takisto posilnené obvodom UDN2928. Pre zobrazenie len jedného miesta, musí byť jeden z týchto obvodov zakázaný, príslušnou logickou úrovňou na vstupnom pine označom INHIBIT. Potvrdenie zápisu hodnoty na výstup sa takisto vykonáva privedením hodinového impulzu.

Schéma zapojenia riadenia displeja je na obrázku Obr. 3.7 a plošný spoj na Obr. 3.8.

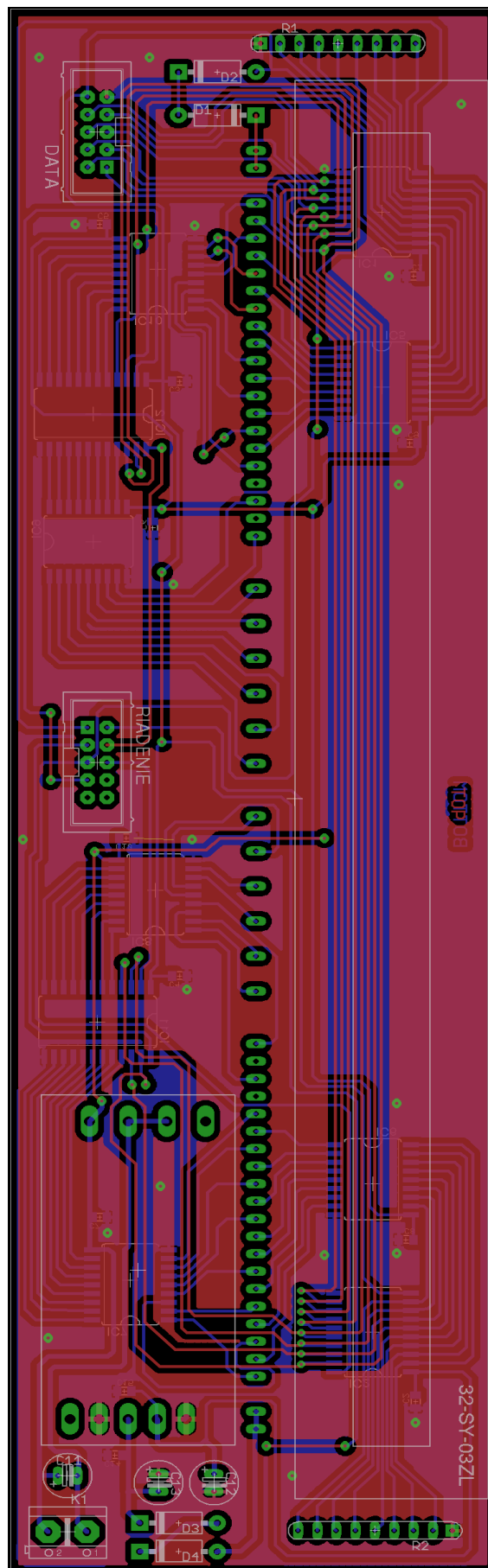
Zoznam použitých súčiastok:

Part	Hodnota	Púzdro
C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C14, C15	100 nF	0805
C11, C12, C13	10 uF	C-EL 2,5
R1, R2	8 x 10K	RR_8X
D1, D2, D3, D4	1N4007	D041
IC1, IC2	HC573M	SO20W
IC5, IC6, IC7, IC8, IC9, IC10	UDN2928LW	SO20W
IC11, IC12	4514	SO24W
U1	32-SY-03ZL	
U2	70KMX7-05-05-901	
K1	ARK500	ARK500/2
DATA, RIADENIE	MLW10	MLW10G

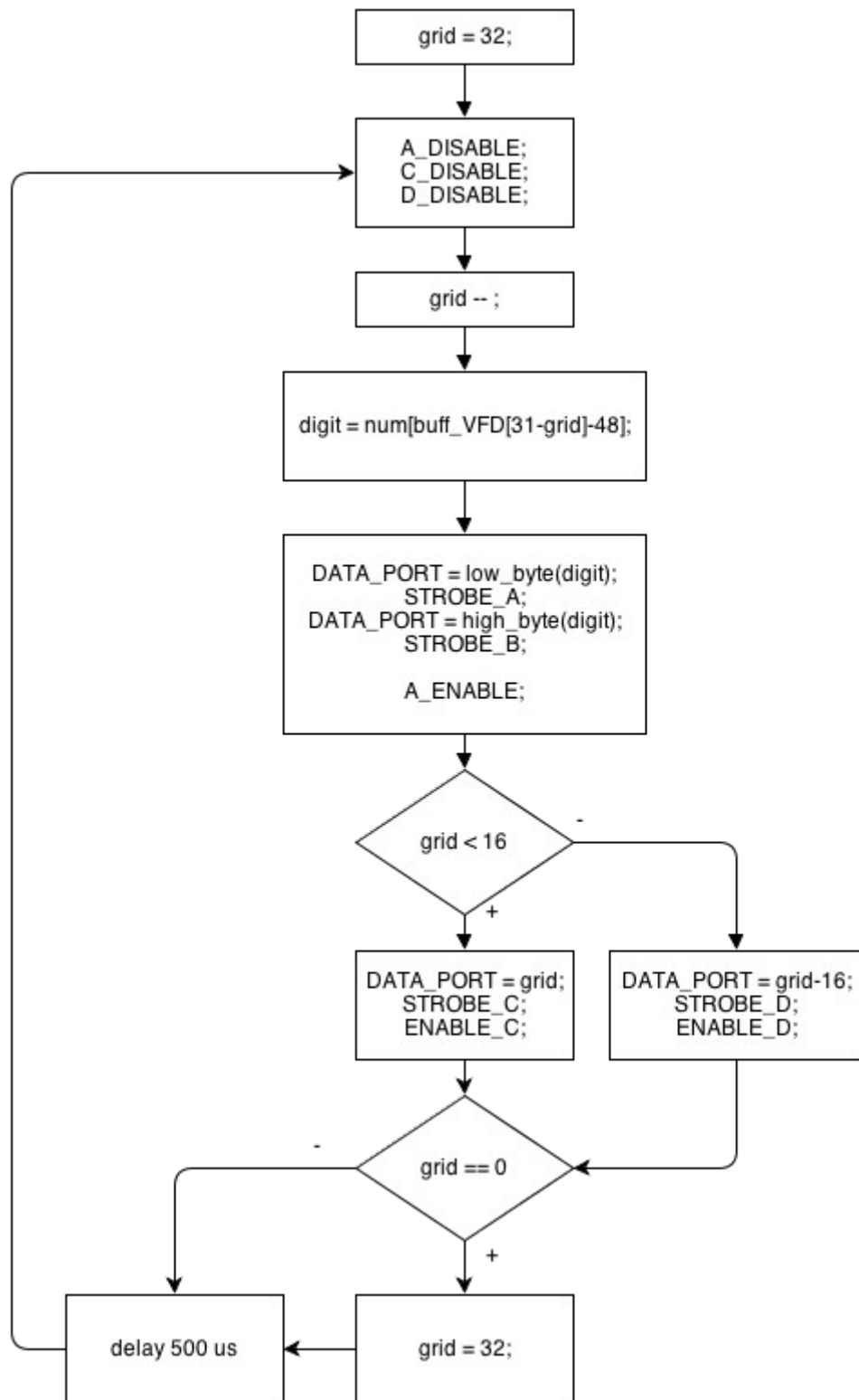
Tab. 3.3 Požité súčiastky



Obr. 3.7 Konektory pre pripojenie dosky displeja k mikropočítaču



Obr. 3.9 Plošný spoj riadenia displeja



Obr. 3.10 Vývojový diagram riadenia displeja

Algoritmus zobrazovania znakov na displeji je nasledovný: premenná *grid*, ktorá nesie informáciu o prave zobrazovanom mieste na displeji, sa najprv inicializuje na hodnotu 32, čo je prve miesto na displeji zľava. Obe registre (A,B) a dekódery sú na začiatku deaktivované, t.j. na displeji sa nič nezobrazuje. Dekrementuje sa premenná *grid* a do premennej *digit* sa načíta hodnota, ktorá predstavuje miesto v tabuľke ASCII znakov pre zobrazovaný znak. Zobrazovať sa bude znak z buffer-a na mieste *grid*. Na port, na ktorom je pripojený dátový vstup riadenia displeja sa najprv vystaví spodných 8 bitov premennej *digit*, a dáta sa potvrdia hodinami pre register A. Dáta sa takto v registri uchovávajú. Potom sa na dátový port vystavia horné bity premennej *digit* a takisto sa potvrdia hodinami pre register B. Funkcia, resp. výstup oboch registrov sa povolí príkazom A_ENABLE (ním sa povoľuje register A a zároveň aj register B). Ďalej sa v algoritme zisťuje stav premennej *grid*. Ak je jej hodnota menšia ako 16, výber zobrazovaného miesta sa uskutoční dekóderom C a ak je hodnota väčšia ako 16 (a zároveň rovná) tak sa zobrazované miesto vyberá dekóderom D. Vybrané miesto sa potvrdí hodinovým signálom pre vybraný dekóder a takisto sa dekóder povolí *inhibit* signálom. Po tomto procese sa skontroluje stav premennej *grid*, ak je nulový tak sa nastaví na hodnotu 32. Po vysvietení znaku na danom mieste sa počká 500 us a algoritmus sa opakuje. Časovanie 500 us sa v programe realizuje pomocou prerušenia od časovača 0, aby čas 500 us bol využitý na inú činnosť mikroprocesora.

Priebeh riadenia displeja na jeho dátových a riadiacich linkách sme zachytili logickým analyzárom a priebeh je možné vidieť na obrázkoch Obr. 3.11.

a) zobrazuje priebeh komunikácie na jednotlivých linkách, je tu možné všimnúť si striedavé prepínanie medzi dekóderom C a D na *inhibit* linkách (D5 a D6) resp. hodinových impulzoch (D2 a D3); takisto hodinové impulzy riadenia registrov A a B (D0 a D1);

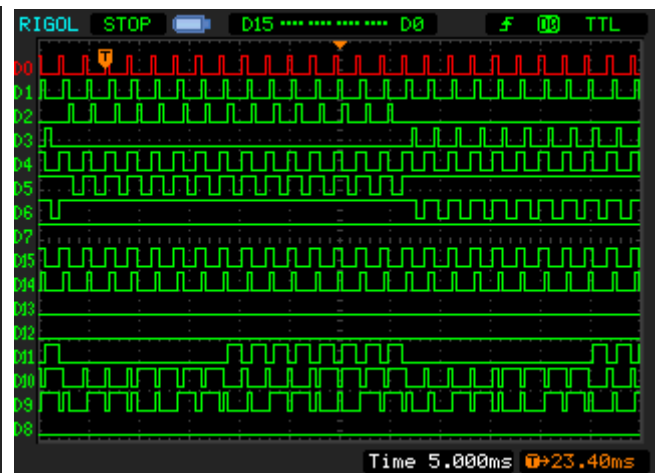
b) na tomto obrázku je detailnejší pohľad na zápis jedného zobrazovacieho miesta displeja;

c) screen vytvorený pri zápise znaku „0“ (dátové linky D8 až D15) na dekóder D (jedno jeho miesto); vidno že pri hodinovom impulze na register A je na dátových linkách spodných 8 bitov premennej *digit* a pri hodinovom impulze na register B je na dátových linkách horných 8 bitov premennej *digit*;

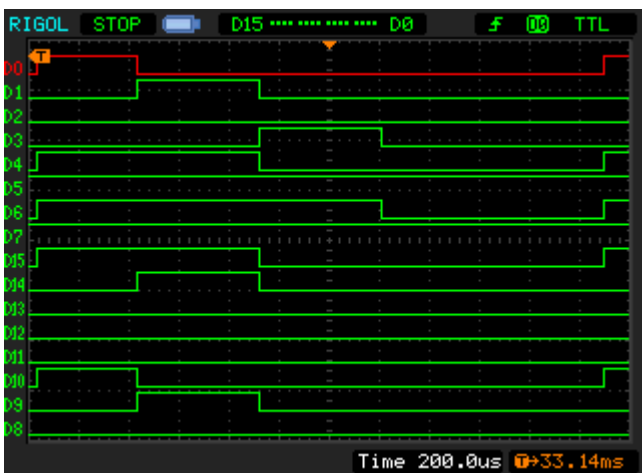
d) podobne ako screen c) len s tým rozdielom že s znak zapisuje na jedno miesto pomocou dekódera C.



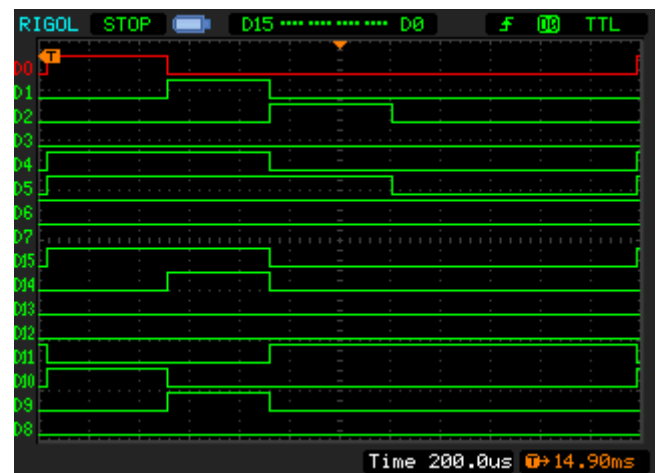
a)



b)



c)



d)

Obr. 3.11 Skeny riadenia displeja logickým analyzárom

Linka	Opis
D0	Hodiny pre register A
D1	Hodiny pre register B
D2	Hodiny pre dekóder C
D3	Hodiny pre dekóder D
D4	Povolenie A/B registra
D5	Povolenie C dekódera
D6	Povolenie D dekódera
D8 – D15	Dáta na zápis

Tab. 3.4 Popis liniek logického analyzára

Pre dobré naskenovanie priebehov sme museli pred meraním nastaviť dĺžku hodinové signálu na 500 us, aby bola hrana signálu dobre viditeľná. Takisto bolo obnovovanie znaku nastavené na 1 ms. Po odmeraní analyzárom sme dĺžku hodinového signálu skrátili naspäť na 1 us a obnovenie na displeji nastavili na 500 us.

3.3 Riadenie stopiek pomocou tlačidiel a optozávor

Riadenie činnosti stopiek sa vykonáva pomocou tlačidiel. Tlačidiel je celkom 5. Tri tlačidlá slúžia na spustení stopiek, stopnutie a vynulovanie a zvyšné dva pre zachytenie medzičasu, ktoré sa môžu pripojiť na optozávory. Tlačidlá pre medzičas sú pripojené na externé prerušenia INT0 a INT1. Pre nedostatok prerušení boli tlačidlá pre štart, stop a reset nastavené tak, že sa v behu programu kontroluje ich stav. Tlačidlá sa dajú v programe nastaviť na ľubovoľné piny portov. Takéto portovanie bude popísané v tejto dokumentácii ďalej.

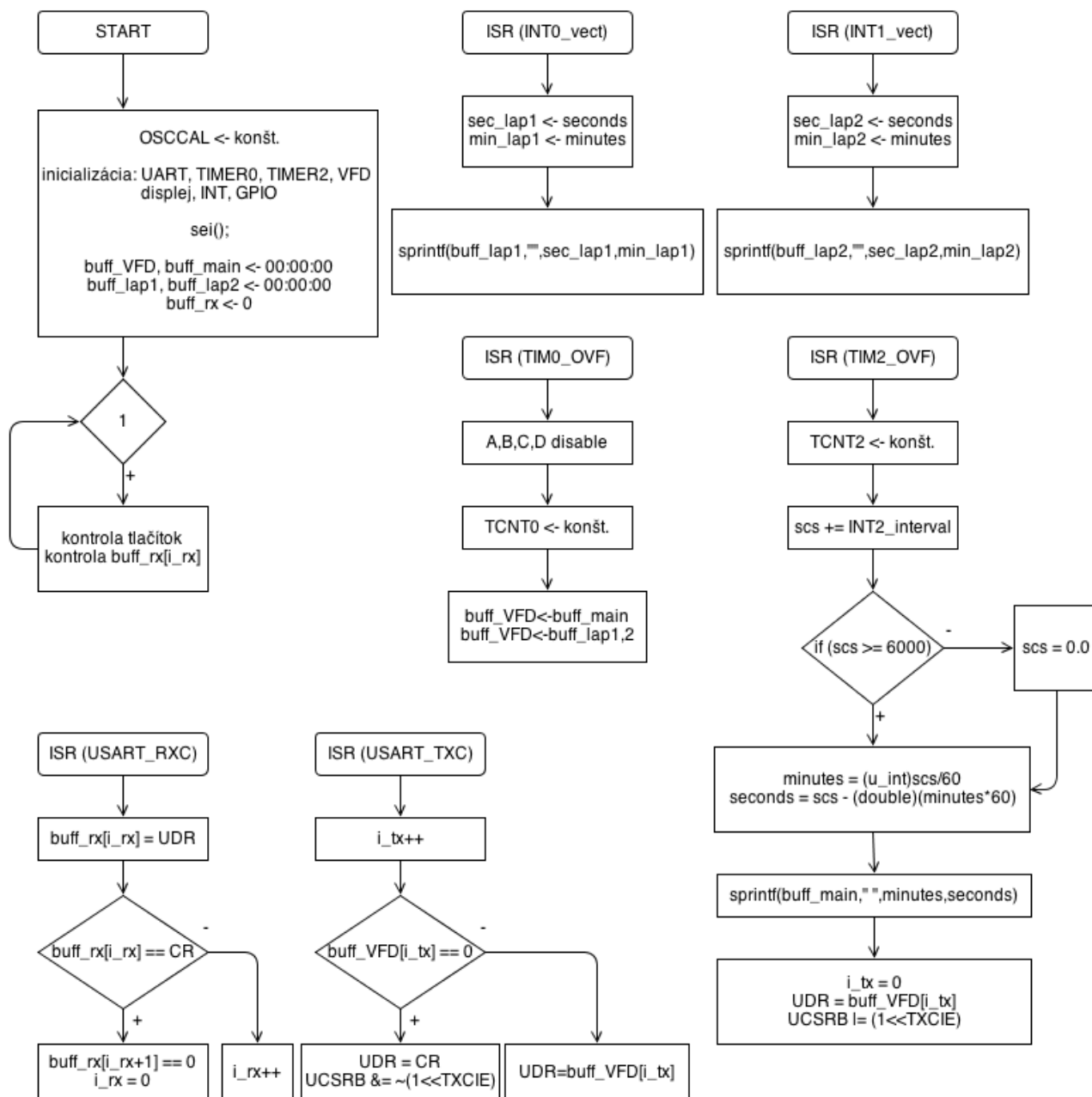
3.4 Prenos údajov o čase cez sériovú linku do PC

Na prenos dát do osobného počítača sa využíva radič UART, ktorý poskytuje mikropočítač ATmega16. Pomocou neho je vytvorená sériová linka, ktorou sa komunikuje s počítačom. Na strane počítača je vytvorený COM port, ktorý je vstupom/výstupom do počítača. Fyzicky je rozhranie medzi mikropočítačom a PC realizované rozhraním RS232 a na strane mikropočítača prevodníkom z úrovne TTL na úroveň, ktoré využíva protokol RS232 prenosu. Prevodník je realizovaný pomocou obvodu od firmy MAXIM, konkrétne MAX232.

Vysielanie údajov o čase je realizovaný vždy po zmeraní času. Vysiela sa celý 32 bajtový buffer z ktorého sa zobrazuje aj na VFD displej. Pre čo najefektívnejšie využitie mikropočítača a radiča UART-u je vysielanie realizované pomocou prerušenia od vyslania jedného bajtu. Takto môžeme využiť čas medzi vysielaním jednotlivých bajtov na inú činnosť (napr. zobrazovanie na displeji).

V programe je riešený aj príjem reťazca znakov, pomocou ktorého sa stopky dajú riadiť. Napríklad spustenie, štart, stop stopiek. Tento reťazec sa prijme do osobného buffera a po prijatí ukončovacieho znaku, ktorý pre nás predstavuje znak CR („\r“), sa v hlavnej slučke programu *while(1)* vyhodnotí aká činnosť sa má vykonať.

3.5 Riadiaci program pre mikropočítač ATmega16



Obr. 3.12 Screen programu s GUI pre zobrazenie času

Vývojový diagram celého programu vidno na obrázku Obr. 3.12. Po štarte programu sa inicializujú premenné na počiatočné hodnoty, inicializuje sa UART, TIMERO pre obnovovanie zobrazenia na displeji, TIMER2 pre časovanie, GPIO pre VFD displej a tlačidla a prerušenia od optozávor. Pred vstupom do *while(1)* sa ešte povolia prerušenia. Počas behu programu, teda v cykle *while(1)* sa kontrolujú tlačidla (či niektoré bolo stlačené – zmena logickej úrovne) a takisto sa kontroluje buffer prijatého reťazca z UART-u. Koniec

príjmu je označený znakom CR. Vtedy sa vykoná príslušná akcia.

V obsluhu externých prerušení sa odpamätá aktuálny čas do premenných a pomocou funkcie `sprintf()` sa uloží do buffer-a reťazec s informáciou o medzičase v tvare akom sa zobrazuje na displej. V obsluhu prerušenía časovača0 sa buffer-e v ktorých sú uložené reťazce ASCII znakov pre zobrazenie na displej a vyslanie na sériovú linku presunú do spoločného 32 bajtového buffer-a, odkiaľ sa robí výpis na disple. Na začiatku obsluhy sa ešte inicializuje počiatočná hodnota čítača. V obsluhu od prerušenía časovačom 2 sa na začiatku inicializuje čítač, aktualizujú sa hodnoty sekúnd a minút, vykoná sa ošetrovanie pretečenia merania času (čiže meranie času do maximálne 99 minút, 59 sekúnd a 99 stotín sekundy), a tieto hodnoty sa prekonvertujú na ASCII znaky a zabalia do buffera. Nakoniec sa vykoná odoslanie prvého znaku 32 bajtového buffera cez sériovú linku a nastaví sa prerušenie od konca vysielania bajtu, v ktorého obsluhu sa vyšle ďalší bajt z buffer-a.

Vlastnosti programu ako je obnovovacia frekvencia displeja, presnosť stopiek, čiže interval prerušenía od časovača 2, rýchlosť UART-u, nastavenie portov pre displej a tlačidlá je možné nastaviť podľa potreby direktívami preprocesora v súbore `define.h`. Príklad nastavenia je na obrázku Obr. 3.13.

```
/* VFD displej */

#define DATA_PORT    PORTB
#define DATA_DDR     DDRB
#define DATA_PIN     PINB

#define DRIVE_PORT    PORTA
#define DRIVE_DDR     DDRA
#define DRIVE_PIN     PINA

#define CLKA          PA0
#define CLKB          PA1
#define CLKC          PA2
#define CLKD          PA3
#define INHA          PA4
#define INHC          PA5
#define INHD          PA6

/* casove konstanty casovacov */

#define INT2_INTERVAL 0.015625//0.0625
#define INTO_INTERVAL 194

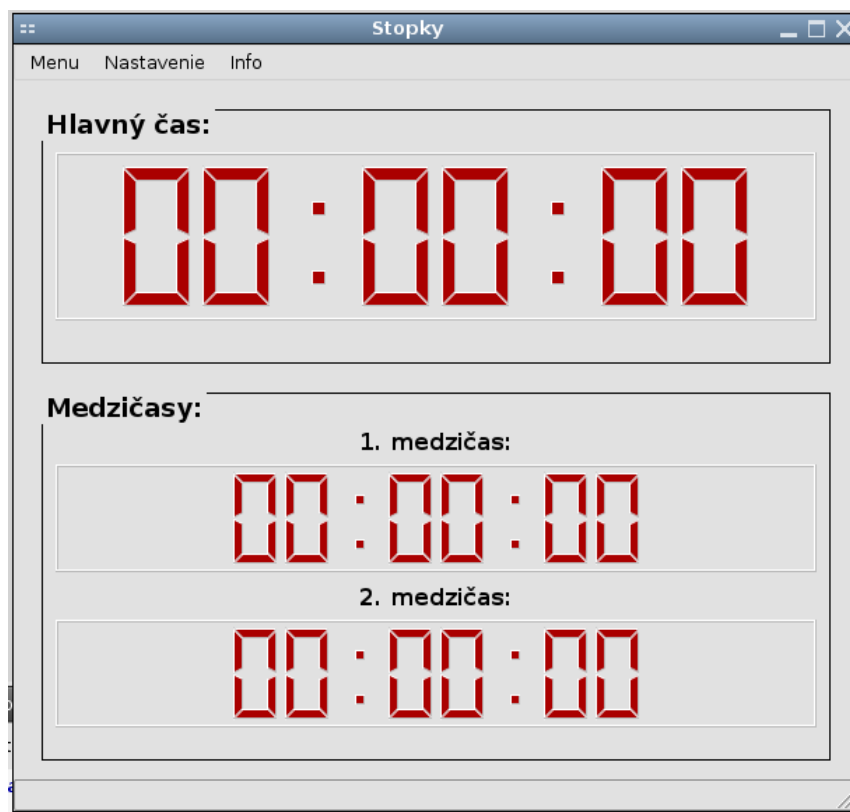
/* tlacidla */

#define BUTT_PORT     PORTD
#define BUTT_DDR      DDRD
#define BUTT_PIN      PIND
#define BUTT_STOP     PD4
#define BUTT_START    PD5
#define BUTT_RESET    PD6
```

Obr. 3.13 Nastavenie portovania displeja, tlačidiel a behu programu v subore „define.h“

3.6 Zobrazovanie času v PC pomocou programu s GUI

Pre zber dát zo sériovej linky a zobrazovanie času v PC sme vyvinuli program s grafickým užívateľským rozhraním – GUI. Tento program ako to vidno na obrázku dole zobrazuje tri segmentové displeje, na ktorých sa zobrazuje celkový čas a dva medzičasy.



Obr. 3.14 Screen programu s GUI pre zobrazenie času

Program umožňuje:

- a) pripojiť sa na sériovú linku (COM port) osobného počítača z defaultným nastavením parametrov prenosu 38400 8N1;
- b) spustiť/zastaviť/nulovať stopky;

Tieto nastavenia sa dajú vykonať pomocou *menu bar*-u na hornej lište, alebo pomocou klávesových skratiek vypísaných v tabuľke Tab. 3.5.

Klávesová skratka	Popis
Ctrl + C	Pripojí / odpojí program na sériovú linku
Ctrl + T	Štart stopiek
Ctrl + E	Stop
Ctrl + R	Nulovanie
Ctrl + P	Pauza displeja (pozastavenie zobrazovania)
Ctrl + Q	Vypne program

Tab. 3.5 Popis klávesových skratiek pre ovládanie programu

Program je vyvinutý pomocou knižníc Qt, ktoré poskytuje a vyvíja firma Digia. Program. jazyk, ktorý sa využíva pri programovaní je objektovo-orientovaný jazyk C++.

4 Záver

Výsledkom tohto projektu sú stopky resp. zariadenie pre meranie času s možnosťou zmerania medzičasu, ktoré môže byť realizované buď klasickým tlačidlom alebo pomocou optozávor napr. pri zmontovaní zariadenia na autodráhe. Presnosť stopiek sa dosahuje na 1,56 stotín sekundy. Jednou z možností pri vylepšovaní tohto projektu je zvýšiť ich presnosť. Výstup zo zariadenia je možné sledovať nie len na VFD displeji, ktorého pripojenie bolo opísané v tejto práci, ale aj pomocou osobného počítača pričom pripojenie je realizované sériovou linkou. Pre grafické zobrazenie meraného času je možné využiť navrhnutý program s grafickým užívateľským rozhraním. Program zabezpečuje pohodlné spúšťanie/stopnutie resp. resetnutie stopiek pomocou klávesových skratiek.