



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

**Fakulta Elektrotechnická
Katedra řídicí techniky**

Návrh elektronické časomíry pro předmět A3B99RO Roboti

Design of electronic timing for subject A3B99RO Robots

Bakalářská práce

Studijní program: Elektrotechnika a informatika
Studijní obor: Kybernetika a měření
Vedoucí práce: Ing. Martin Hlinovský, Ph.D.

Erik Janoušek

Praha 2010



Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne

.....

Podpis



Poděkování

Mé poděkování patří Ing. Martinu Hlinovskému Ph.D. za kvalitní vedení této bakalářské práce. Dále kolegům, známým a kamarádům, kteří svými podněty přispívali k doladování mých idejí a myšlenek. Nejvíce z nich bych rád jmenovitě poděkoval dlouholetému kolegovi a kamarádovi Bc. Michalu Hroudovi za jeho velkou ochotu kdykoliv poradit či zkontrolovat jakýkoliv problém na mé práci. Nesmím samozřejmě opomenout na rodinu, kde všichni byli vstřícní k mým časovým nárokům a poskytovali mi výbornou psychickou a finanční podporu.

Abstrakt

V této práci najdete vypracování elektronické časomíry od prvotního návrhu elektronického schématu, přes návrh plošného spoje se snahou o maximální EMC kompatibilitu, dále naprogramování mikroprocesoru, naprogramování uživatelské aplikace pro PC a následnou mechanickou montáž. Řídící deska obsahuje mikroprocesor Freescale MC9S08AW60. Tato deska zajišťuje příjem a zpracování vstupních signálů (start/stop jak z tlačítka, tak z optických závor, nulování, nastavení časového rozsahu), posílání výstupních informací do desek zajišťujících zobrazování na displej a komunikaci s PC. Zobrazování času na segmenty je prováděno pomocí posuvných výkonových registrů. Komunikace mezi zařízením a PC je zprostředkována po USB rozhraní, nicméně přímo na desce je převodník USB-UART (sériová linka) a tedy komunikace k/od mikroprocesoru probíhá podle standartu RS232. Uživatelské rozhraní bude naprogramováno v jazyce C#.

Klíčová slova: elektronická časomíra, mikrokontrolér, segmentový displej

Abstract

In this dissertation you will find development of electronic chronometer from initial design of electronic scheme, through designing of PCB with maximal effort to EMC compatibility, as well as programming of microprocessor, programming of user interface for PC followed by mechanical assembly. Control board contains Freescale MC9S08AW60 microprocessor, this board provides receiving and processing of input signals (start/stop from button and optical gates, reset of time, set time-scale), sending output information to boards for displaying and communication with PC. Displaying time segments is performed by power shift registers. Communication between device and PC is done by USB interface, however on board itself is UART-USB converter (serial line), so communication to/from microprocessor follows the standard RS232. The user interface is programmed in C#.

Keywords: electronic chronometer, microcontroller, segment display



Obsah :

1. Úvod	1
2. Rozbor úlohy	2
3. Hardware	3
3.1. Napájecí obvody	6
3.2. Výkonový posuvný registr	8
3.2.1. Podrobnější popis funkce registru a jeho parametry	9
3.3. Komunikační převodník UART-USB	10
3.4. Mikroprocesor	12
3.5. Popis zbylé elektroniky	14
3.6. Přídavné periferie	19
4. Software	20
4.1. Programování mikroprocesoru	20
4.1.1. Popis funkce :	21
4.2. Uživatelská aplikace	22
5. Mechanické řešení	23
6. Závěr	24

Seznam obrázků:

obr. 1	Krabice na elektronickou časomíru	1
obr. 2	Řídící deska	3
obr. 3	Napájecí modul k displeji	4
obr. 4	Segmentový displej	5
obr. 5	Schéma napájení řídící desky	6
obr. 6	Schéma napájení pro posuvný registr	7
obr. 7	Zapojení posuvného registru	8
obr. 8	Princip funkce posuvného registru	9
obr. 9	Blokové schéma převodníku UART-USB	10



obr. 10	Schéma zapojení UART-USB převodníku	11
obr. 11	Mikroprocesor v pouzdře LQFP-44	12
obr. 12	Zapojení mikroprocesoru	13
obr. 13	Celé schéma řídicí desky	14
obr. 14	Hlavní deska – spoje, spodní strana	16
obr. 15	Hlavní deska – spoje, horní strana	16
obr. 16	Napájecí deska – spodní strana	17
obr. 17	Napájecí deska – spodní strana	17
obr. 18	Schéma napájecí desky	18
obr. 19	Laserová dioda	19
obr. 20	Fototranzistor	19
obr. 21	Ovládací tlačítka	19
obr. 22	Vývojový diagram programu procesoru	20
obr. 23	Váhy bitů pro jednotlivé segmenty	21
obr. 24	Uživatelská aplikace	23

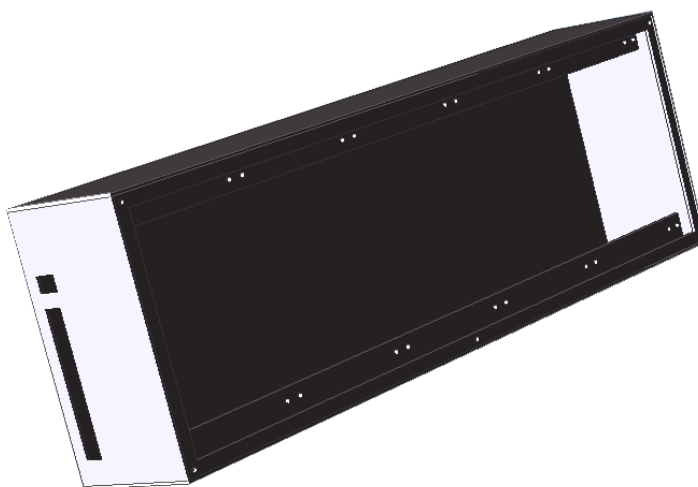
Seznam tabulek :

Tabulka 1 : soupis součástek na hlavní desce	15
Tabulka 2 : soupis součástek napájecích desek	18
Tabulka 3 : Kódování číselných hodnot pro zobrazení na displeji	22

1. Úvod

Tato úloha si pokládá za úkol kompletní návrh elektronické časomíry pro použití v předmětu A3B99RO Roboti. Bude zde podrobně vše popsáno od řešení návrhu elektroniky až po uživatelskou aplikaci.

Problém jsem se rozhodl řešit takto. Výsledný produkt obsahuje velký segmentový displej o pěti číslicích uzavřený do obdélníkové kovové krabice vyráběné na míru. Krabice má povrchovou úpravu, čelo je překryto plexisklem pro lepší vnější dojem. Vně krabice najdeme kompletní elektroniku. Na segmentech jsou ze zadu přimontovány desky pro úpravu signálu obsahující informaci o zobrazované časové hodnotě. Ty jsou mezi sebou spojeny plochým kabelem, který přenáší jak samotnou informaci, tak i napájecí napětí +15V a GND. Hned za prvním segmentovým displejem se nachází přímo na krabici přimontována řídící deska spojená s deskami pro úpravu signálu stejným plochým kabelem. Ta má na boku vyvedeny konektory, kvůli kterým je z boku krabice vyříznuta díra. Jde o napájecí konektor, dva 4-pinové konektory pro optické závory, jeden 3-pinový konektor pro ovládací tlačítek START/STOP a NULOVÁNÍ a nakonec komunikační USB konektor. Kousek nad pevně usazenou řídící deskou se nachází kolébkový spínač pro celé zařízení. Zdroj celého zařízení jsem se rozhodl řešit hotovým adaptérem a to z důvodů jednodušší montáže, lepších bezpečnostních parametrů a i rychlejší nahraditelnosti v případě jeho poruchy.



obr. 1 Krabice na elektronickou časomíru

2. Rozbor úlohy

Jelikož v zadání úlohy je mimo povinnosti měření času také komunikace s PC a ovládání více vstupů, rozhodl jsem se aplikaci řešit pomocí mikroprocesoru. V dnešní době je toto řešení finančně nenáročné, rychlé a jednoduché. Před začátkem návrhu se tedy hned nabízel otázka jaký mikroprocesor použít. Nakonec volba byla snadná. Na tuto úlohu by se hodil jakýkoliv mikroprocesor. Ale rozhodl jsem se pro produkt firmy Freescale. Důvodů je několik. Zaprvé je tato celosvětově uznávaná značka velice rozšířená hlavně ve Spojených státech amerických. Nabízejí velice jednoduché a intuitivní programovací prostředí Freescale CodeWarrior Development Studio. Dále já osobně mám k dispozici okolo sebe lidi, kteří mi mohli nabídnout ať už know-how okolo těchto procesorů, literaturu, tak i HW podporu v podobě programátoru. Použit tedy byl mikroprocesor Freescale z rodiny HCS08.

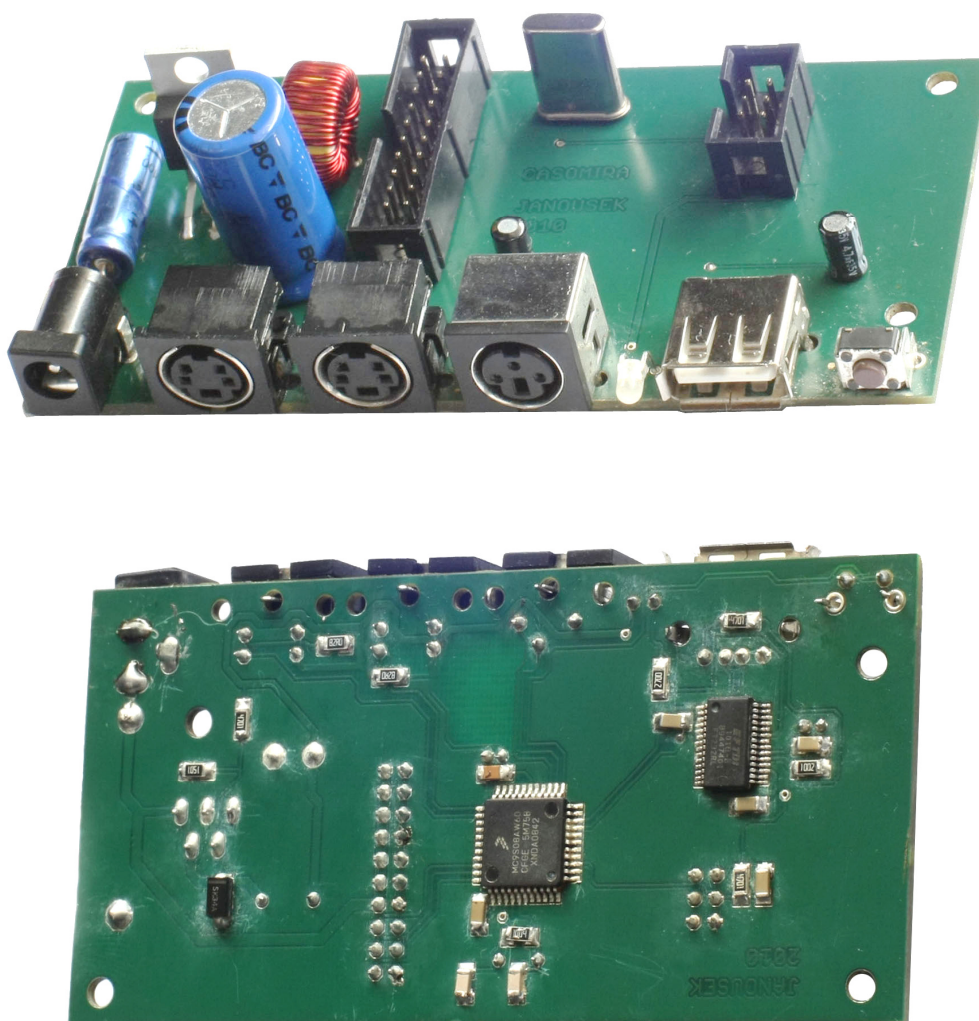
Další stěžejní otázkou bylo, jakých a kolik displejů použít. Použít již hotový zobrazovací panel nepadalo vůbec v úvahu, díky opravdu vysokým cenám a i problému najít vyhovující konstrukční řešení. Situace se tedy dala řešit vlastním návrhem rozložení LED na desku plošných spojů a nebo koupí hotových segmentových displejů. Ty se ovšem shánějí ve velikostech, co jsem potřeboval, složitě. Nakonec jsem použil hotový produkt o výšce číslice 15cm. Pro tuto úlohu jsem zvolil pěti ciferné uspořádání, s tím, že standardní nastavení bude jedna cifra pro minuty, dvě cifry pro vteřiny a dvě cifry pro desetiny a setiny vteřiny. Tuto konfiguraci považuji za nejvhodnější vzhledem k tomu, jaké soutěže probíhali pro předmět A3B99RO Roboti v minulých letech. Pro jiné využití nebude problém mikroprocesor přeprogramovat a umožnit konfiguraci jinou.

Po vyřešení otázky zobrazovací jednotky přišla na řadu otázka napájení. Jelikož k zobrazovací jednotce se dala zjistit pouze informace o odebíraném proudu na segment, musel jsem zjišťovat napájecí napětí experimentálním měřením. Na základě tedy až této informace bylo konečně možné vybrat vhodný zdroj energie.

Periferie byly řešeny takto. Tlačítka START/STOP a NULOVÁNÍ jsou průmyslová tlačítka z produkce firmy Mollier vyvedené na dlouhém kabelu. Optické závory jsou řešeny laserovou diodou a hned pod ní umístěn fototranzistor. Laserový paprsek je odražen na fototranzistor zrcátkem.

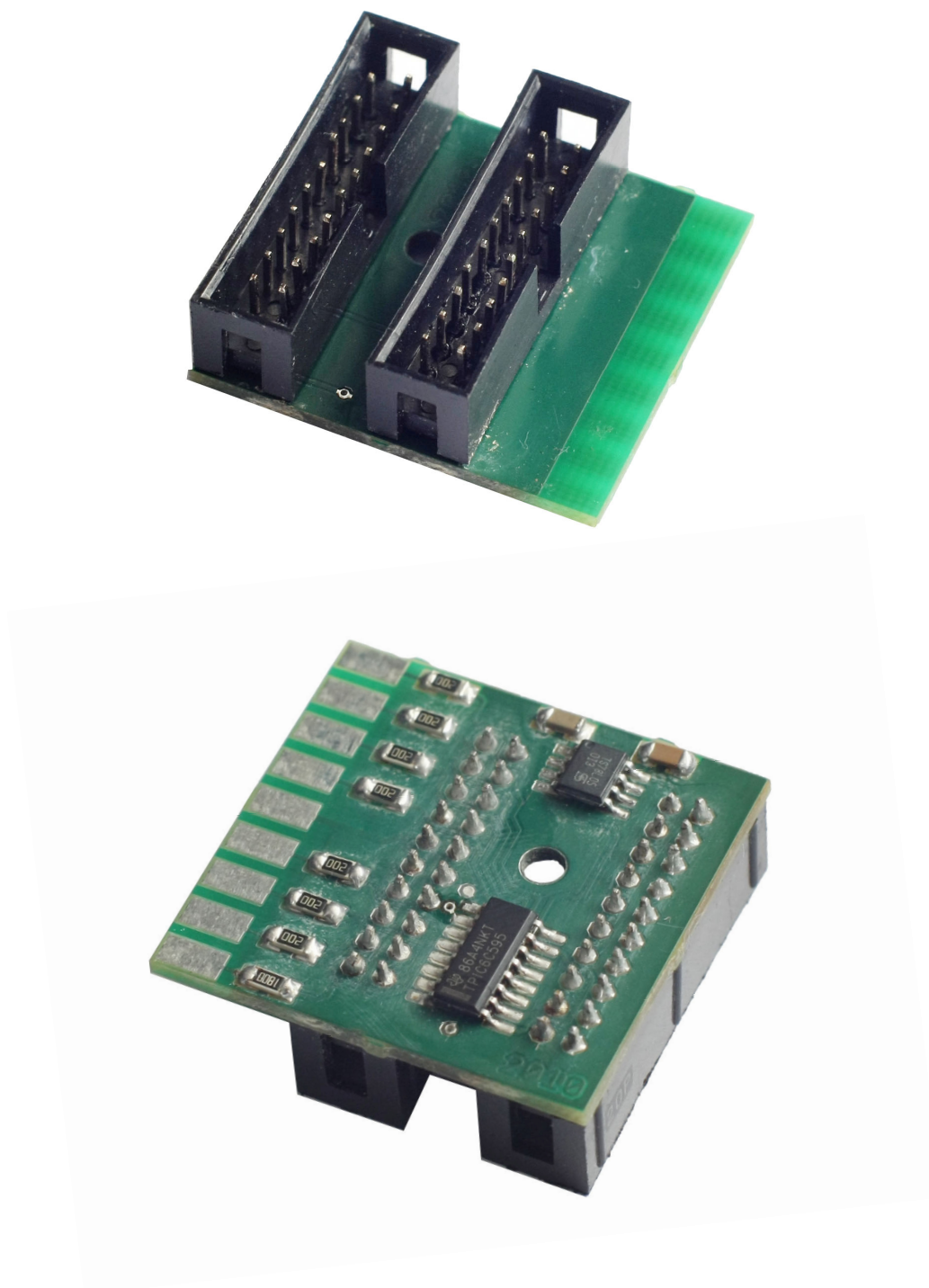
3. Hardware

Tato kapitola bude řešit otázku kompletního HW řešení. Na obr. 2 jsou fotografie řídicí desky. Z horní strany jde vidět běžný napájecí konektor se středovým kolíkem o průměru 2.1mm, dva čtyř-pinové a jeden tří-pinový konektor MiniDIN pro externí periférie, klasický USB konektor typu A, svítivou diodu pro detekci USB komunikace, tlačítko pro nastavování rozsahu měřeného času a dva konektory pro ploché kabely s roztečí mezi piny 2,54mm. Šesti-pinový konektor je pro BDM rozhraní určené k programování mikroprocesoru, dvacetipinový konektor slouží k ovládání a napájení displejů. Na spodní straně jsou k vidění odpory a kondenzátory v provedení SMD, dále schotkyho SMD dioda patřící napěťovému obvodu, samotný mikroprocesor v pouzdře LQFP-44 a převodník USB-to-UART v pouzdře SSOP-28. Důležité součástky budou blíže popsány v následujících podkapitolách.



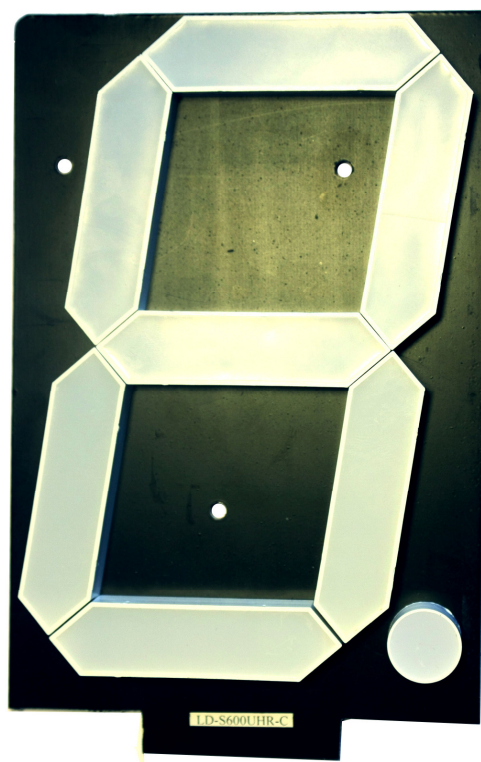
obr. 2 Řídicí deska

Další elektronickou částí, kterou jsem potřeboval navrhnout a zkompletovat jsou napájecí obvody pro jednotlivé displeje. Na obr. 3 si tuto součást můžeme také prohlédnout. Shora vidíme pouze dva konektory plochého kabelu sloužící pro příjem a další odesílání dat k zobrazení, ze spodní strany je vidět napájecí obvod a výkonový posuvný registr.



obr. 3 Napájecí modul k displeji

V neposlední řadě jako důležitý prvek je zde samotný segmentový displej. Jak bylo již řečeno, jde o už zakoupený hotový produkt. Pod obr. 4 naleznete specifikaci.

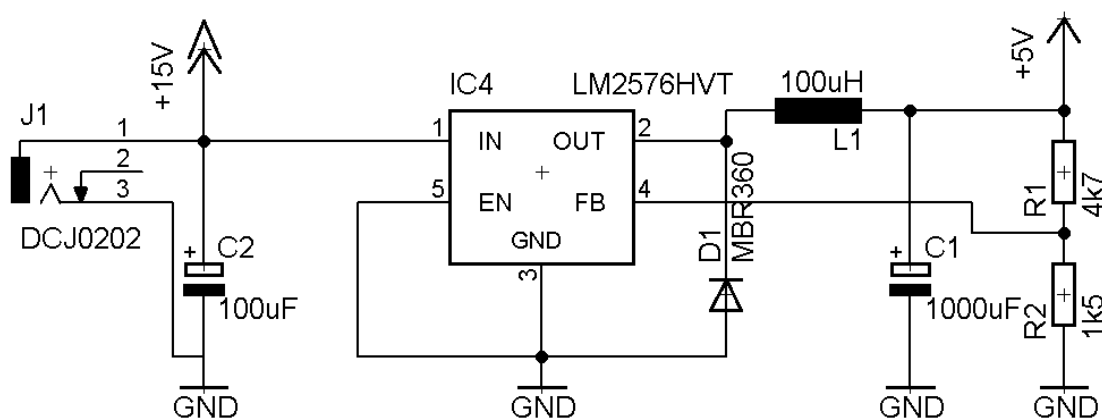


obr. 4 Segmentový displej

Typ zobrazovače :	LED jednomístný 7+ segmentový se společnou anodou
Výška znaku :	152 mm
Barva :	červená
Svítivost :	700 mcd
Vlnová délka λ_d :	660 nm
Pracovní proud :	50 mA
Prac. napětí segmentů :	14,2 V
Prac. napětí tečky :	8,5V

3.1. Napájecí obvody

V mém případě je na hlavní desku přivedeno napětí 15V, které budu dělit mezi napájení pro displeje a regulátor. Jako srážecí regulátor napětí pro řídicí desku jsem zvolil obvod LM2576-ADJ. Ten zajistí stabilních 5V pro chod jemné elektroniky na celé desce. Výhodou tohoto obvodu je jeho vysoký rozsah vstupního napětí a vysoká zatížitelnost na výstupu. Ta dosahuje hodnoty až 3A, a jelikož jsem mimo mikroprocesoru a převodníku UART-USB potřeboval napájet externí periferie jakými jsou například optické závory, nechtěl jsem nic ponechat náhodě. Tyto obvody se pyšní dalšími vlastnostmi, které jsem ani nevyužil. Například se jedná o možnost elektronického zapínání/vypínání. Jde o obvod s vysokou efektivitou a stabilitou.



obr. 5 Schéma napájení řídicí desky

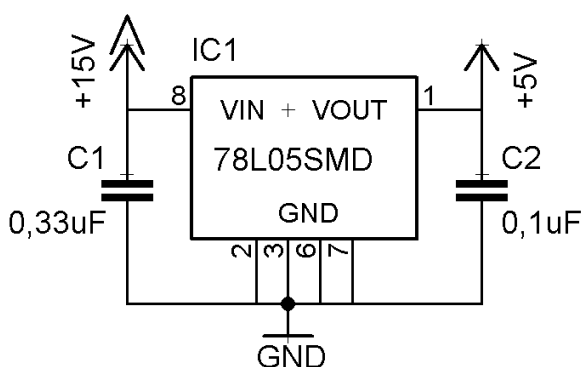
Na obr. 5 vidíme rezistory R_1 a R_2 , které přesně určují výstupní napětí. Jsou počítány podle vzorce uvedeného do výrobce v datasheetu, uvedl jsem jej níže.

$$U_{out} = U_{ref} \left(1 + \frac{R_2}{R_1} \right)$$

vzorec 1 výpočet výst. napětí

Jde vlastně o běžný vzorec pro dělič napětí, jedinou potřebnou informací od výrobce je referenční napětí, a to má hodnotu $U_{ref} = 1,23V$.

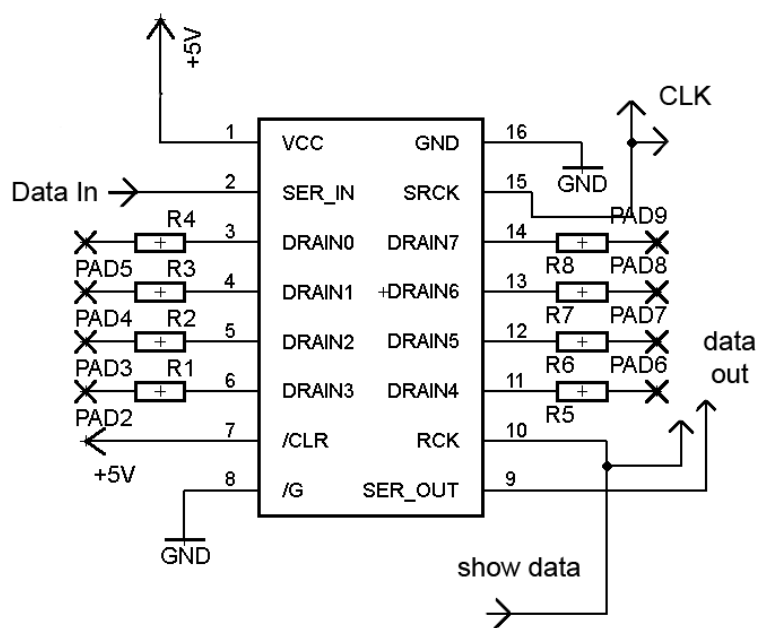
Na malé desky pro displeje jsem musel použít také napájecí obvod. I když samotný segmentový displej určitě nepotřebuje žádné stabilizované napětí o nízké hodnotě, posuvný registr zpracovávající informace pro něj již ano. Jelikož potřebné napětí 5V bylo vážně jen pro jeden výkonový posuvný registr, postačil v tomto případě jednoduchý stabilizátor 78L05 s maximální zatížitelností 100mA.



obr. 6 Schéma napájení pro posuvný registr

3.2. Výkonový posuvný registr

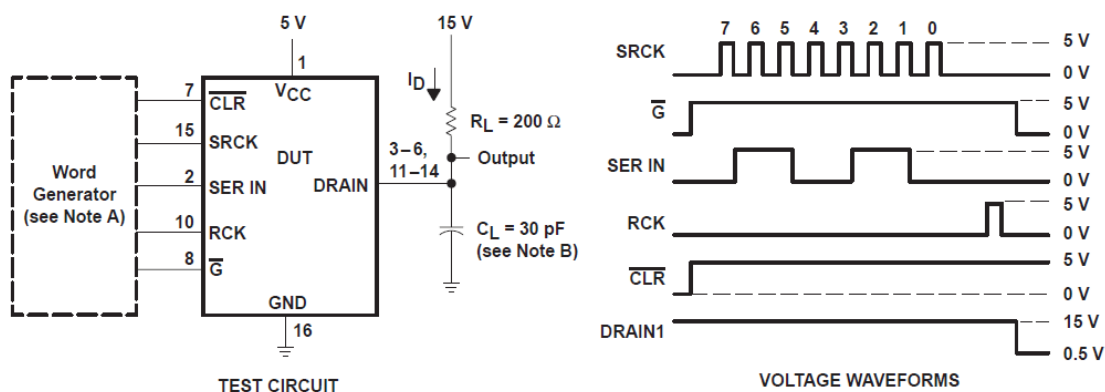
Posuvné registry jsou v této práci využity k posílání číselné informace z procesoru na segmentový displej. Použity jsou tzv. SIPO registry, kdy vstupní hodnoty jsou přijímány sériově a výstupní hodnoty jsou odesílány paralelně. Důvod tohoto použití je jasný, ušetří se tím mnoho místa a zjednoduší kompletace celého zařízení. Jde konkrétně o typ TPIC6C595, jeho principiální funkce odpovídá složením z D-registrů. Obvod je speciálně navržen pro využití v systémech vyžadujících vyšší zatížení, jako jsou třeba LED. Tato vlastnost se přesně hodila do této práce, protože odběr každého segmentu činí 50mA. To je hodnota pro běžné posuvné registry nedosažitelná. Dále se jedná o 8-bitový registr. Spousta obvodů pro číselné displeje počítá pouze se základními 7 segmenty. Jelikož je zde využívána i desetinná tečka, dvakrát, jedná se tedy o další klad.



obr. 7 Zapojení posuvného registru

3.2.1. Podrobnější popis funkce registru a jeho parametry

Rozkreslený princip funkce registru je na obr. 8. Pro ovládání registru jsem používal 3 signály vysílané procesorem. První signál SRCK je hodinovým signálem zajišťujícím každou svoji náběžnou hranou přečtení bitové informace. Ta je posílána do registru v SER_IN signálu. Pokud se registr zaplní příchozími datovými bity, posouvá je ve stejném pořadí dál výstupem s názvem SER_OUT. Takto funguje registr za předpokladu, že hodnota signálu RCK je nula. Jakmile v tomto signálu přijde náběžná hrana, pošle v sobě uložená data na paralelní výstup.



obr. 8 Princip funkce posuvného registru

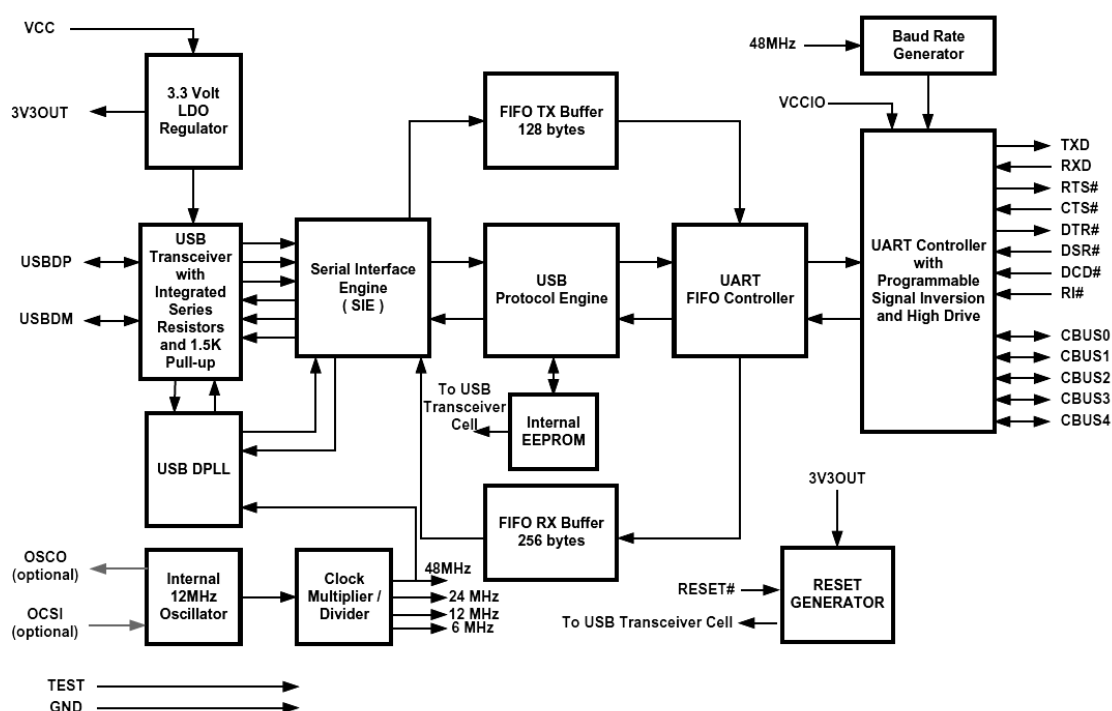
Doporučené provozní hodnoty :

- napájecí napětí:	4,5 – 5,5 V
- logická 1:	min. 0,85V
- logická 0:	max. 0,15V
- pulzní výstupní proud:	max. 250mA
- předběh SER_IN před \uparrow SRCK:	min. 20ns
- doběh SER_IN po \uparrow SRCK:	min. 20ns
- max. výstupní napětí:	33V

3.3. Komunikační převodník UART-USB

Ohledně komunikace s PC, jsem byl rozhodnut použít standardní sériovou linku. Důvodem této volby byla hlavně jednoduchost a spolehlivost komunikačního protokolu. Do cesty ovšem vstoupila překážka v podobě existence COM portů na dnešních počítačích. Toto rozhraní na stolních PC najít ještě lze, ale například na přenosných počítačích už několik let tyto porty k nalezení nejsou. Výbornou volbou jsou tedy převodníky UART-USB, které zajišťují komunikaci podle standardů RS232 přes USB rozhraní. K tomu stačí pouze nainstalovat na daný počítač ovladače pro podporu zařízení.

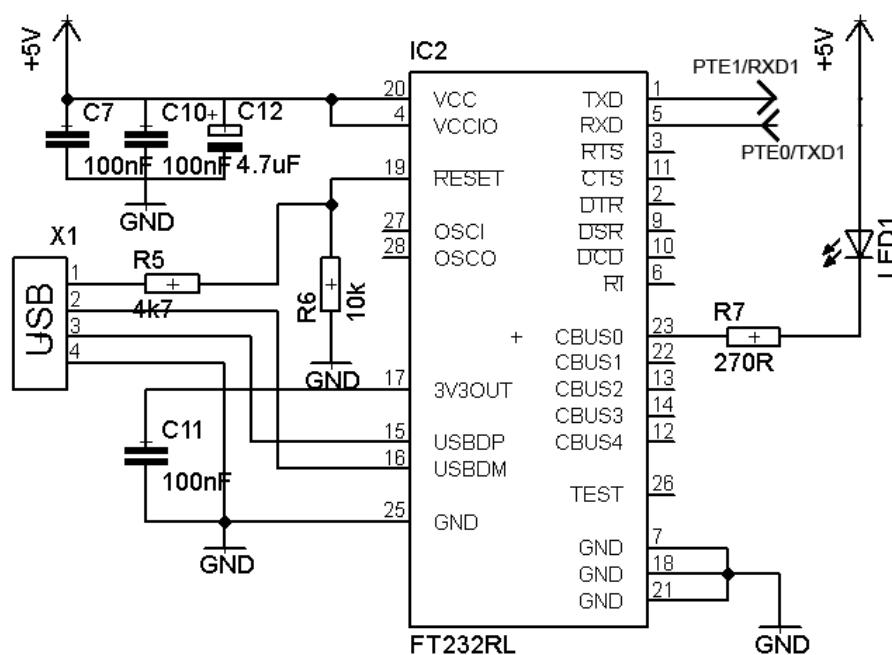
Vybral jsem převodník od firmy FTDI s názvem FT232RL. Tyto převodníky jsou léta prověřovány jejich uživateli a rozhodně patří k těm kvalitnějším. Také se to podepisuje na jejich ceně, která je vyšší než u mikroprocesoru.



obr. 9 Blokové schéma převodníku UART-USB

Převodník podporuje napájecí napětí v rozsahu 3,3V až 5,25V, veškerá HW komunikační ověřování, nastavování start, stop, paritních bitů atd. Má v sobě zabudovanou 1024 bit EEPROM s výchozím továrním nastavením. Přes software výrobce lze následně nastavit

parametry komunikace uložené v této paměti převodníku a zvolit funkci k I/O portům CBUS0-CBUS4.



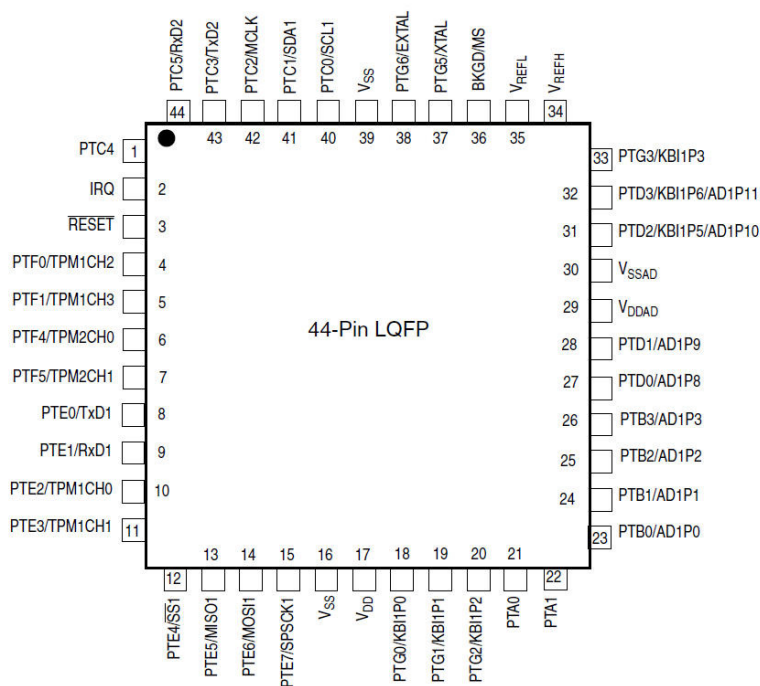
obr. 10 Schéma zapojení UART-USB převodníku

Převodník je zapojen v tzv. samo-napájecím módu. Jde o to, že zařízení musí být napájeno zvlášť a nemůže se napájet z USB sběrnice. Při zapojování jsem využil jeden z volitelných I/O portů a použil jsem CBUS0 pro zapojení signalizační LED. Signalizovat budu komunikaci v obou směrech (tedy Tx i Rx). Porty 1 a 5 (TxD a Rx) jsou propojeny s porty na mikroprocesoru dle schématu.

3.4. Mikroprocesor

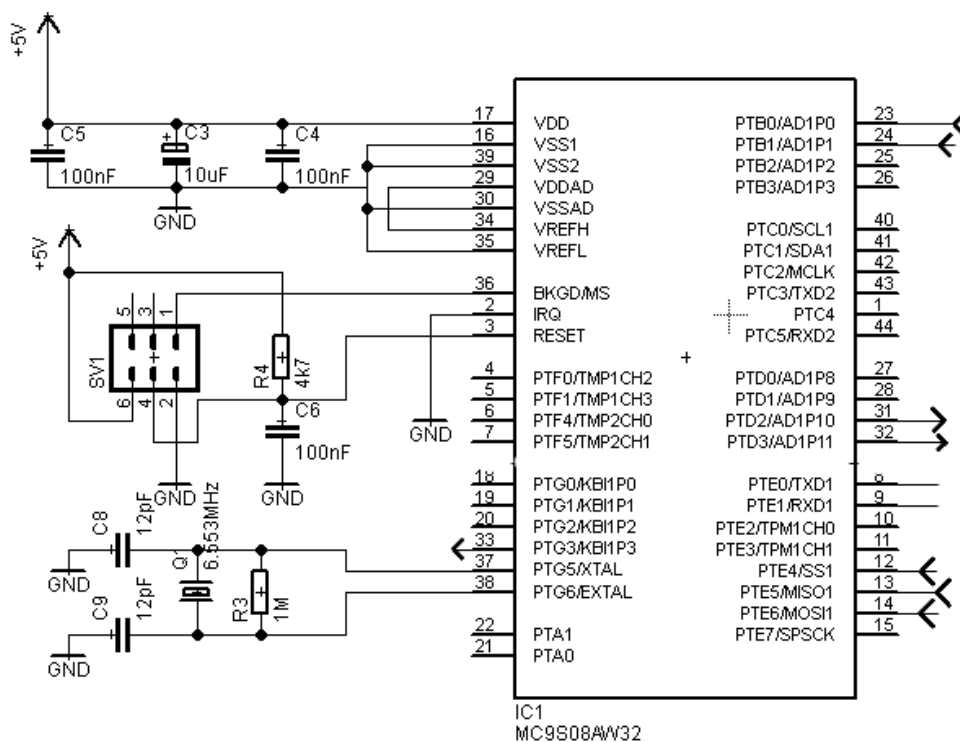
Použitým mikroprocesorem je MC9SO8AW60 z rodiny HCS08 od firmy Freescale semiconductor. Jedná se o 8-bitové mikrokontroléry se všemi potřebnými periferiemi. Tyto procesory spadají pod instrukční sadu CISC. Použitý model je bohatě vybaven, obsahuje jádro o taktu 40MHz, podporu až 32 přerušení, až 60kB programovatelné FLASH paměti, 2kB RAM, dva vlastní časovače, možnosti připojení externího krystalu, systémové ochrany, úsporný režim, 16-ti kanálový 10 bitový analogově číslicový převodník, dvě sériové komunikační rozhraní, IIC komunikační rozhraní, 8 klávesových přerušení a až 54 I/O portů.

Já jsem využil sériovou komunikaci, připojení externího krystalu, 5 portů pro vstupy, 3 porty pro výstupy. Dále mám ještě zapojeno BDM programovací rozhraní. Z toho může vypadat, že jsem použil zbytečně vysokou řadu procesorů HCS08, a je to i pravda. Důvodem je, že jsem chtěl použít variantu s napájecím napětím +5V, které nejsou u nás v prodeji. Je možné je objednávat přímo ze Spojených států amerických, avšak to jsem považoval za zdlouhavé. Využil jsem tedy zásob známého a vzal co měl „po ruce“.



obr. 11 Mikroprocesor v pouzdře LQFP-44

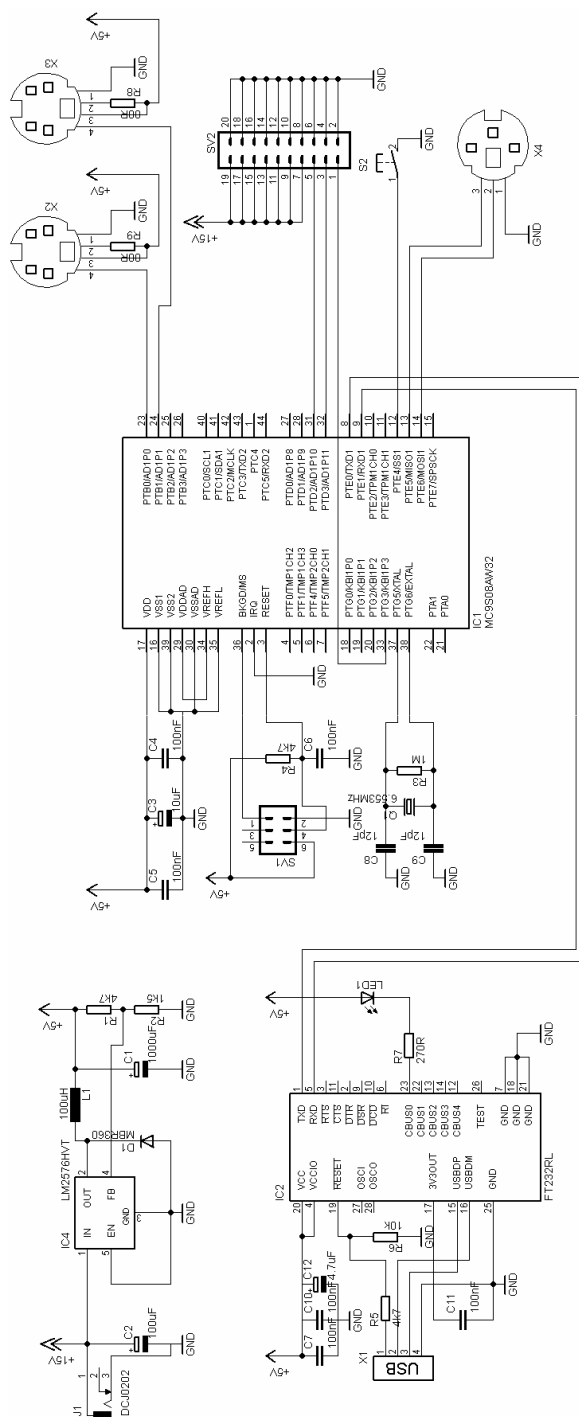
Zapojení mikroprocesoru je standardní podle datasheetu výrobce. Pro UART komunikaci jsem použil piny TXD1 a RXD1. Piny PTE4-6 jsou vyvedeny k ovládacím tlačítkům. Jedno je přímo na desce, jedná se o tlačítko nastavující časový limit čítání. Další dvě jsou pomocí kabelu vyvedeny vzdáleně a slouží k startování/zastavování a nulování měřeného času. Signály k zobrazování na displej pochází z pinů PTG3, PTD2 a PT3. Jedná se o signály SRCK, SER_IN a RCK popsané již výše u podkapitoly „výkonový posuvný registr“. A nakonec tu máme přijímanou informaci z optických bran a ty jsou do pinů PTB0 a PTB1.



obr. 12 Zapojení mikroprocesoru

3.5. Popis zbylé elektroniky

V této podkapitole budou vyobrazena kompletní schémata, návrhy desek a soupis veškerých součástek.



obr. 13 Celé schéma řídicí desky

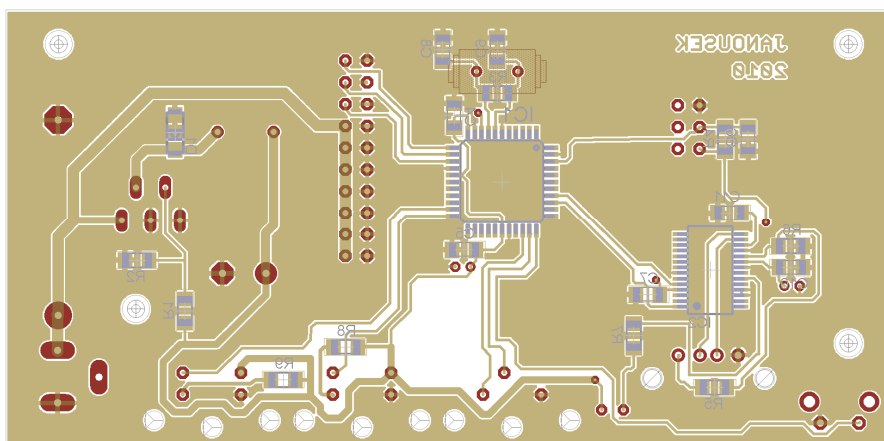


Soupis součástek:

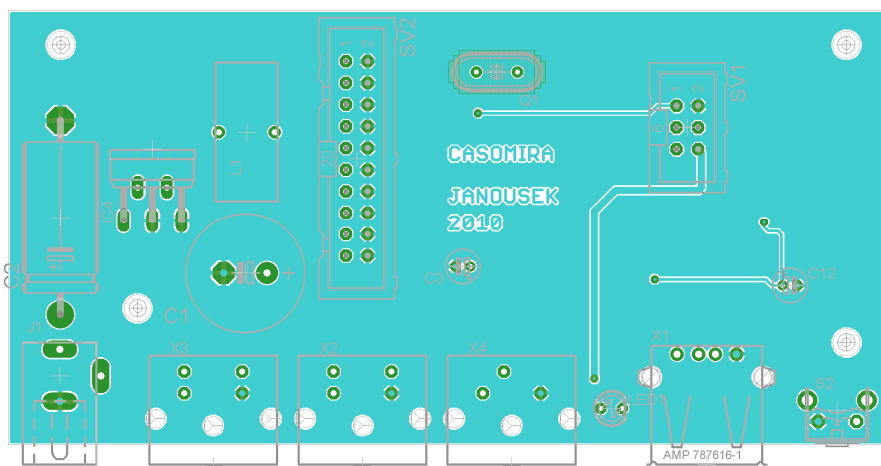
označení	typ	popis
IC1	MC9S08AW60	Mikroprocesor
IC2	FT232RL	Převodník UART-USB
IC4	LM2576-ADJ	Napěťový regulátor
L1	DPU100A1	Cívka 100uH 1A
D1	SK34SMD	Schottky dioda 40V 3A
Q1	6.553M-HC49R	Krystal 6,5536 MHz
LED1	L-937GYW	Svítilivá dioda 3mm
C1	MAL213666102E3	Elyt kond. 1000uF, 25V
C2	CE 100/40A	Elektrolytický kondenzátor
C3	E10M/16VM	Miniaturní elyt
C12	E4M7/35VM	Miniaturní elyt
C4, C5, C6, C7, C10, C11	SMD-1206 100uF	Keramický kondenzátor
C8, C9	SMD-1206 12pF	Keramický kondenzátor
R1, R4, R5	SMD-1206 4k7	Rezistor
R2	SMD-1206 1k5	Rezistor
R3	SMD-1206 1M	Rezistor
R6	SMD-1206 10k	Rezistor
R7	SMD-1206 270R	Rezistor
R8, R9	SMD-1206 82R	Rezistor
J1	K375A	Napájecí JACK kon. 2,1mm
X1	USBA-G	USB kon. do DPS
X2,X3	MDD4BB	MiniDIN konektor 4-pin
X4	MDD3BB	MiniDIN konektor 3-pin
SV1	MLW06A	Plochý 6-pin (3x2)
SV2	MLW20A	Plochý 6-pin (5x2)

Tabulka 1 : soupis součástek na hlavní desce

Hlavní deska byla navrhována tak, aby se vešli na její delší stranu všechny potřebné konektory spojující zařízení s vnějškem. To samo o sobě rozhodlo o ne zrovna malých rozměrech vzhledem k použitým SMD součástkám a dvouvrstvému plošnému spoji. Hned u napájecího konektoru jsou rozloženy součástky zajišťující napětí sražené na potřebnou úroveň. Převodník UART-USB je co nejbližší USB konektoru a mikroprocesor je usazen přibližně uprostřed pro co nejlepší tahání signálových cest. Celkově spoje byly navrhovány se snahou o největší EMC kompatibilitu. Příkladem toho jsou spoje blízko sebe a rozlité země v obou vrstvách.



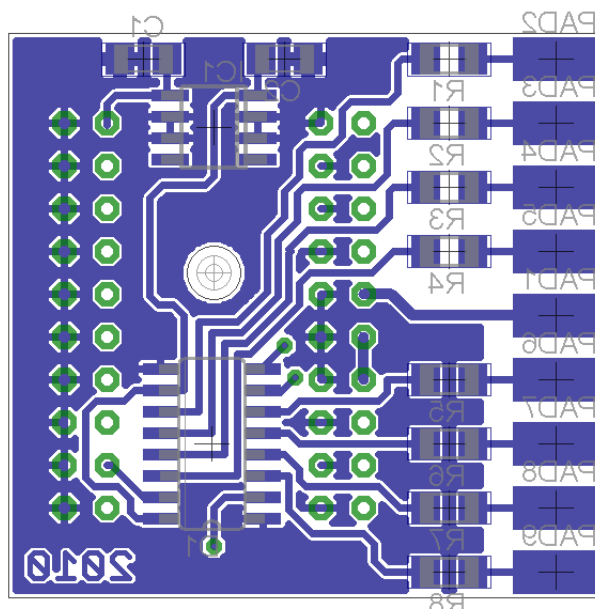
obr. 14 Hlavní deska – spoje, spodní strana



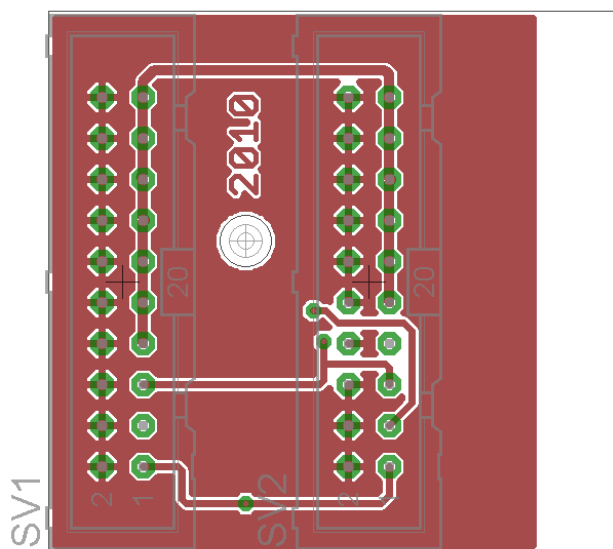
obr. 15 Hlavní deska – spoje, horní strana

Na malých napájecích deskách byla druhá vrstva využita lépe. Na horní straně jsou vidět pouze konektory 20-pinového plochého kabelu. Jeden je „vstupní“ a druhý „výstupní“. Jak byl již princip naznačen výše, přenášejí se po těchto kabelech řídicí signály pro posuvné registry. Ty

jsou tři. Zbylých 17 pinů je využito pro +15V a GND. Hlavní důvod počtu tolika napájecích žil je zajištění si bezproblémového přenosu energie po celém segmentovém poli. Přeci jen při plném rozsvícení celý panel v tu chvíli odebírá 2A.

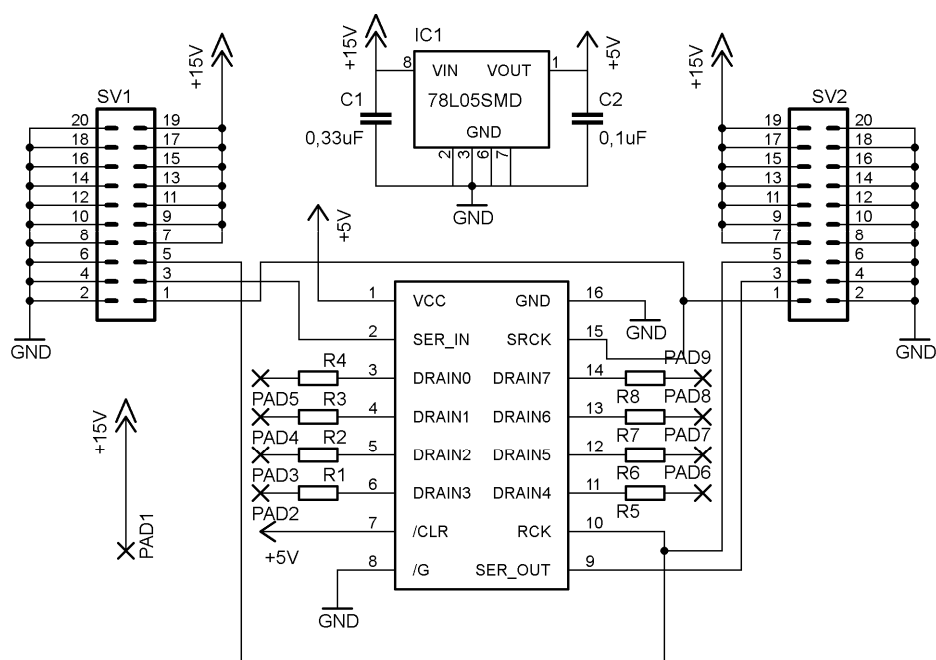


obr. 16 Napájecí deska – spodní strana



obr. 17 Napájecí deska – spodní strana

Kompletní schéma napájecí desky a soupis součástek.



obr. 18 Schéma napájecí desky

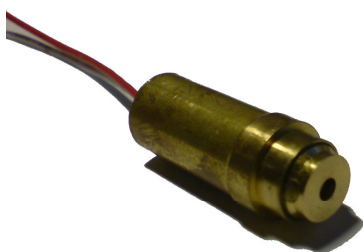
označení	typ	popis
SV1, SV2	MLW20A	Plochý 6-pin (5x2)
IC1	78L05 SMD	Napěťový stabilizátor
IC2	TPIC6-C-595	Posuvný 8-bitový registr
R1-R7	SMD1206, 20R	Rezistor
R8	SMD1206, 180R	Rezistor
C1	SMD1206, 330nF	Kondenzátor
C2	SMD1206, 100nF	Kondenzátor

Tabulka 2 : soupis součástí napájecích desek

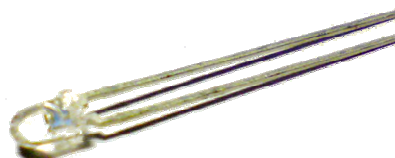
3.6. Přídavné periferie

Na obrázcích níže jsou vidět použité periferie. Jde o tlačítka, červené pro nulování a zelené pro start/stop. A o optické brány složené z laserové diody a z fototranzistoru.

Za laserové diody byl použit produkt 650 LASER MODUL, jedná se o typ s vbudovanou optikou. Jeho napájecí napětí činí 3V, běžný odebírající proud je 25mA a předpokládaný optický výkon 3mW. Jedná se o běžnou červenou barvu vlnové délky 650nm. Paprsek světleného zdroje je odražen o lesklou plochu naproti zpět do trubičky pro 3mm LED. Kde je umístěn fototranzistor v pouzdře právě velice se podobající 3mm světelným diodám. Snímací člen je citlivý na světlo vlnové délky 380-1150nm, napájen je 5V a předpokládané proudy jsou mezi 1,6-3,2mA.



obr. 19 Laserová dioda



obr. 20 Fototranzistor

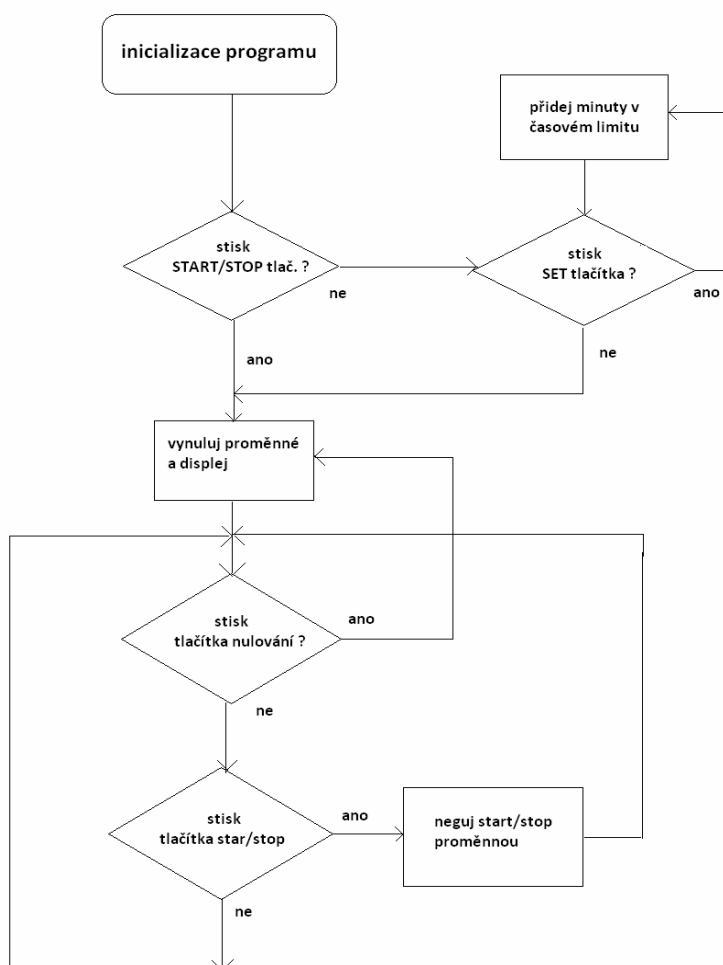


obr. 21 Ovládací tlačítka

4. Software

4.1. Programování mikroprocesoru

Mikroprocesor se programuje pomocí jazyka ANSI C, mírně upraveného pro 8-bitové kontroléry od firmy Freescale semidoncutor. Jejich vývojové prostředí má práci ulehčující grafické rozhraní, kde nastavení procesoru provádíte pomocí kliků myši na dané funkce. Jakmile máte procesor správně nastaven, stačí nechat vygenerovat kód, kde jsou všechny parametry nastaveny a nemusíte se o nic starat. Programátor tedy může programovat jen samotný kód jeho aplikace. Mnou vymyšlený kód je celý v hlavním zdroji „main.c“. Jeho část je k nalezení v příloze. Funkci kódu zobrazí následující vývojový diagram.

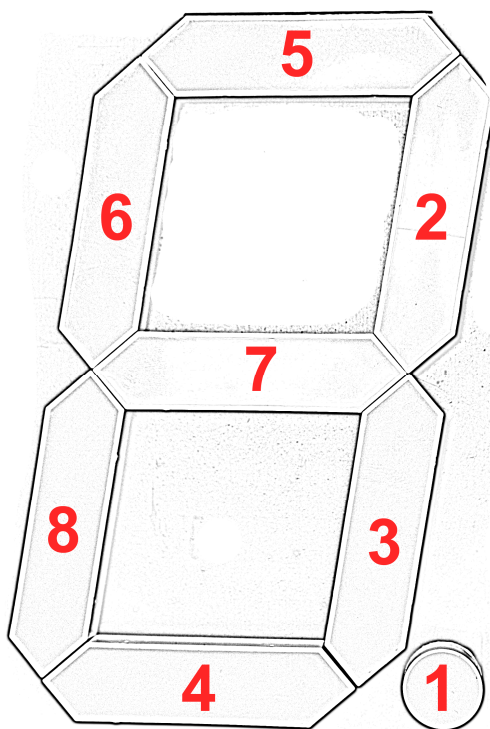


obr. 22 Vývojový diagram programu procesoru

4.1.1. Popis funkce :

Procesor si nainicializuje potřebné proměnné, funkce a kódy. Načte si tabulku znaků pro segmentový displej. Po spuštění programu kontroluje zda není zmáčkuto tl. START/STOP a dokud není, lze tlačítkem SET nastavit časový limit. Primárně je nastaven a zobrazen limit 1 minuta. Po nastaveném čase je program již v nekonečné smyčce mikroprocesoru a pouze kontroluje zmáčknutí jednoho z tlačítek START/STOP nebo NULOVÁNÍ. Zároveň s tím běží každých 10ms takt čítače, kde ihned potvrzujeme procesoru přijmutí zprávy o přerušení vyslané časovačem. Pokud je stavová proměnná `st_sp` aktivována, začnou se přičítat setiny, následně desetiny a takto až po minuty. V každém taktu se mimo přičítání času spustí funkce vykreslení času na displej. Je-li `st_sp` deaktivováno, přestane se čas přičítat. V případě zmáčknutí tlačítka NULOVÁNÍ se vynulují proměnné, zobrazí se jejich nulová hodnota na displej a deaktivuje `st_sp` stavová proměnná nezávisle na jejím předchozím stavu. Pokud časování dosáhne minutového limitu, měření času se zastaví, a po posledním vykreslení hodnoty se vynulují proměnné.

Při běhu celého procesu se real-timeově posílají informace o naměřeném čase do PC po sériové lince, případně přijímají příkazy start/stop, nulování, nastavení časového limitu.



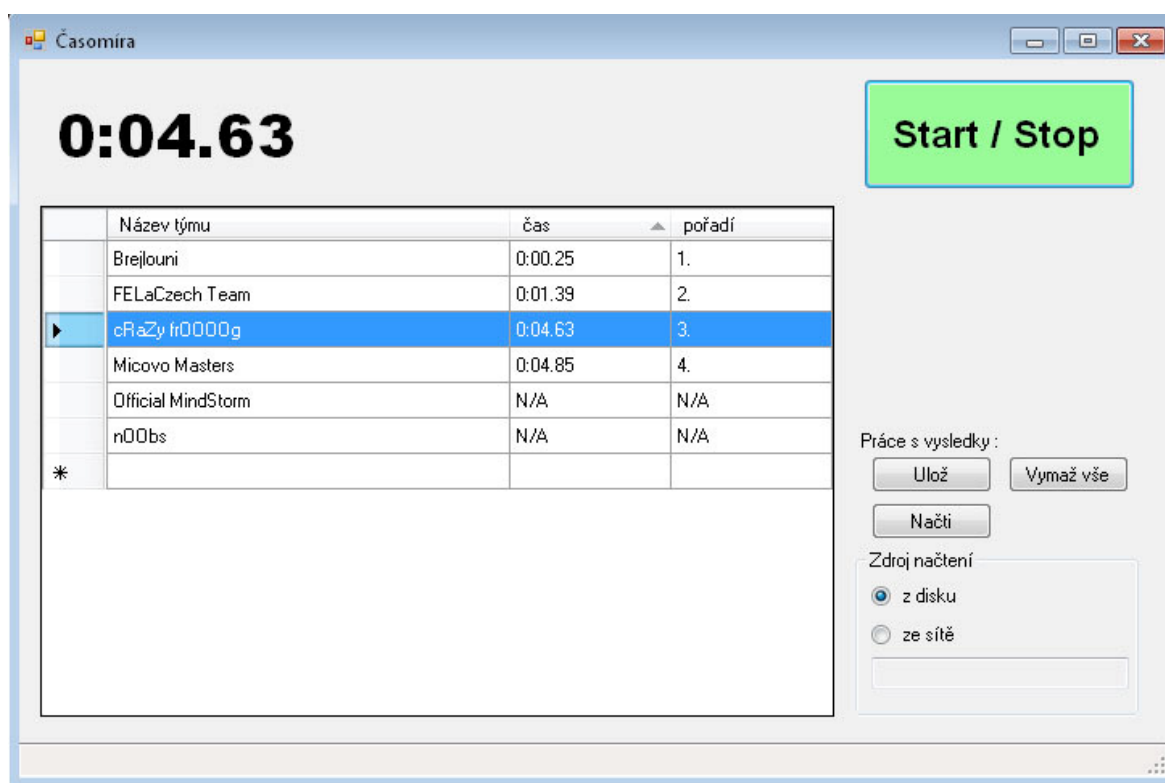
obr. 23 Váhy bitů pro jednotlivé segmenty

číslo	binární kód pro displej	hexa kód
0	10111110	BE
1	00000110	06
2	11011010	DA
3	01011110	5E
4	01100110	66
5	01111100	7C
6	11111100	FC
7	00010110	16
8	11111110	FE
9	01111110	7E

Tabulka 3 : Kódování číselných hodnot pro zobrazení na displeji

4.2. Uživatelská aplikace

Program pro ovládání časomíry z PC je psán v jazyce C#. Pro tento jazyk bylo použito vývojové prostředí Microsoft Visual Studio 2010 a později Microsoft Visual Studio 2010 Express. Jde o jednoduchou aplikaci zpracovávající informace zasílané časomírou. Do tabulky lze vkládat jména týmů, hlavním tlačítkem spouštět či zastavovat měření času a po odměření přiřadit čas správnému týmu kliknutím na jeho jméno. Automaticky okamžitě zpracovává pořadí podle časů. Výsledkové listiny lze ukládat, a naopak i načítat z uložených souborů. Uložené soubory mohou být na místním disku stejně dobře jako na síti.



obr. 24 Uživatelská aplikace

5. Mechanické řešení

Na konci si jen ve stručnosti uvedeme, jak jsem nechal vyrobit box pro tuto časomíru. V CAD systému jsem rozvrhl jednotlivé stěny krabice z plechu. Ty jsem nechal vyřezat laserem podle nákresů. Hotové plechy se svařili, nastříkali sprejem základové barvy, následně sprejem s černou barvou. Krabice má z boku výřez pro konektory, čelní strana je vlastně jen rámeček s dírami pro připevnění kouřového plexiskla. Uvnitř jsou kousek za čelní stranou dole a nahoře lišty pro upevnění segmentových displejů. Na zadní straně, v místě, kde bude řídící deska, jsou přivařeny šrouby pro její uchycení. Celá montáž do boxu je prováděna zepředu a po jejím dokončení stačí přišroubovat plexisklo. Celé zařízení má rozměry 583x169x100mm a váží něco mezi 3-5kg.

Technické výkresy jsou přiloženy v příloze.

6. Závěr

Tato práce nastiňuje jeden z několika způsobů návrhu elektronického zařízení. Pro mne bylo skoro vše nové. Mimo programování jsem nikdy nic podobného nedělal. Na většině škol se evidentně na návrh a vytváření elektroniky důraz neklade. Ať už se jedná o školy střední nebo vysoké v elektrotechnickém zaměření. Osobně to považuji za škodu, jelikož je dost pracovních pozic v oblasti návrhu elektroniky a mimo jiné mi to přišlo zábavné.

Díky práci jsem se začal i pomalu orientovat v součástkové základně na běžném trhu, získal odhad o cenách různých typů součástek a také naučil pájení s SMD součástkami.

Práce samotná ukazuje na jednoduchost a efektivnost elektronických aplikací při použití mikroprocesoru. V dnešní době jsou tyto mikrokontroléry dosti levnou součástkou a zvládnou prakticky cokoliv. Další velice užitečnou informací je práce s UART-USB převodníkem. Pro případ, že se Vám nechce programovat se surovou USB komunikací je toto zajímavá alternativa. Sériová komunikace typu RS232 a obdobné patří k jedním z nejjednodušších. Jako poslední důležitou část by jsem vybral výkonový posuvný SIPO registr. Pro jakékoliv segmentové zobrazování jde o velice užitečného pomocníka, se kterým se jednoduše spolupracuje.

Citace

1. MC9S08AW60: Technical Data Sheet for MC9S08AW60/48/32 MCUs,
[http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC9S08AW60.pdf], 12.2006
2. FT232R - USB UART IC , [<http://www.ftdichip.com/Products/ICs/FT232R.htm>], 2010
3. Texas instruments TPIC6C595 datasheet,
[<http://www.datasheetcatalog.org/datasheet/texasinstruments/tpic6c595.pdf>], 1999
4. SIMPLE SWITCHER® 3A Step-Down Voltage Regulator,
[<http://www.national.com/mpf/LM/LM2576.html#Overview>], 2004
5. L78L00 SERIES datasheet, [http://www.gme.cz/_dokumentace/dokumenty/934/934-005/dsh.934-005.1.pdf], 2005
6. 650 LASER MODULE datasheet,
[http://www.gme.cz/_dokumentace/dokumenty/759/759-040/dsh.759-040.1.pdf]
7. RGB In Wikipedia : the free encyclopedia, [<http://cs.wikipedia.org/wiki/RGB>], 2005
8. Začínáme s mikrokontrolery HCS08, [<http://www.micovo.cz/elektronika-zaciname-s-mikrokontrolery-hcs08.htm>], 2010

přílohy

main.c :

```
#include <hidef.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations */

#ifdef __cplusplus
extern "C"
#endif

#define DISP_SRCK PTGD_PTGD3
#define DISP_SER_IN PTDD_PTDD3
#define DISP_RCK PTDD_PTDD2
#define TLAC_SET PTED_PTED4
#define TLAC_STSP PTED_PTED5
#define TLAC_NULL PTED_PTED6

void MCU_init(void); /* Device initialization function declaration */
void vykresli(unsigned char x_set, unsigned char x_dset, unsigned char x_sek, unsigned char x_dsek, unsigned char x_min );
void nulovani(void);

unsigned char SciRxBuffer[16];
unsigned char SciRxBufferPointer;
unsigned char SciTxBuffer[16];
unsigned char SciTxBufferPointer;

const unsigned char digit[10] = {0xbe, 0x06, 0xda, 0x5e, 0x66, 0x7c, 0xfc, 0x16, 0xfe, 0x7e};
unsigned char st_sp;
unsigned char setiny;
unsigned char desetiny;
unsigned char sekundy;
unsigned char dessek;
unsigned char minuty;
unsigned char max;

void main(void)
{
    int i;
    MCU_init();

    DISP_SER_IN = 0;
    DISP_SRCK = 0;
    DISP_RCK = 0;

    nulovani();
    st_sp = 0;
    max=1;
    minuty=1;
    vykresli(setiny, desetiny, sekundy, dessek, minuty);

    while (TLAC_STSP==1) {
        for (i=0; i <1000; i++)
            if (TLAC_SET == 0)
            {
                minuty=minuty+1;
                if (minuty>9) minuty=0;
            }
    }
```


přílohy

```
        max=minuty;
        vykresli(setiny, desetiny, sekundy, dessek, minuty);
        for (i = 0; i < 500; i++);
        while ( TLAC_SET == 0);
    }
}
nulovani();
for(;;)
{

    for (i=0; i <1000; i++)
    if (TLAC_NULL == 0)
    {
        nulovani();
        for (i = 0; i < 500; i++);
        while ( TLAC_STSP == 0);
    }

    for (i=0; i <1000; i++)
    if (TLAC_STSP == 0)
    {
        st_sp = ! st_sp;
        for (i = 0; i < 500; i++);
        while ( TLAC_STSP == 0);
    }
}
}

void vykresli(unsigned char x_set, unsigned char x_dset, unsigned char x_sek, unsigned char x_dsek, unsigned char x_min )
{
    static unsigned char cislo;
    static unsigned char j, i, k;

    for (j=0;j<5;j++) {
        switch(j)
        {
            case 0: cislo = digit[x_set]; break;
            case 1: cislo = digit[x_dset]; break;
            case 2: cislo = digit[x_sek]+1; break;
            case 3: cislo = digit[x_dsek]; break;
            case 4: cislo = digit[x_min]+1; break;
        }
        for (k=0;k<8;k++){
            for(i=0;i<5;i++){
                DISP_SER_IN = 0x01 & cislo;
                cislo=cislo>>1;
                for(i=0;i<5;i++){
                    DISP_SRCK = 1;
                    for(i=0;i<5;i++){
                        DISP_SRCK = 0;
                    }
                }
            }
        }
        for(i=0;i<5;i++)
        DISP_RCK = 1;
    }
```

přílohy

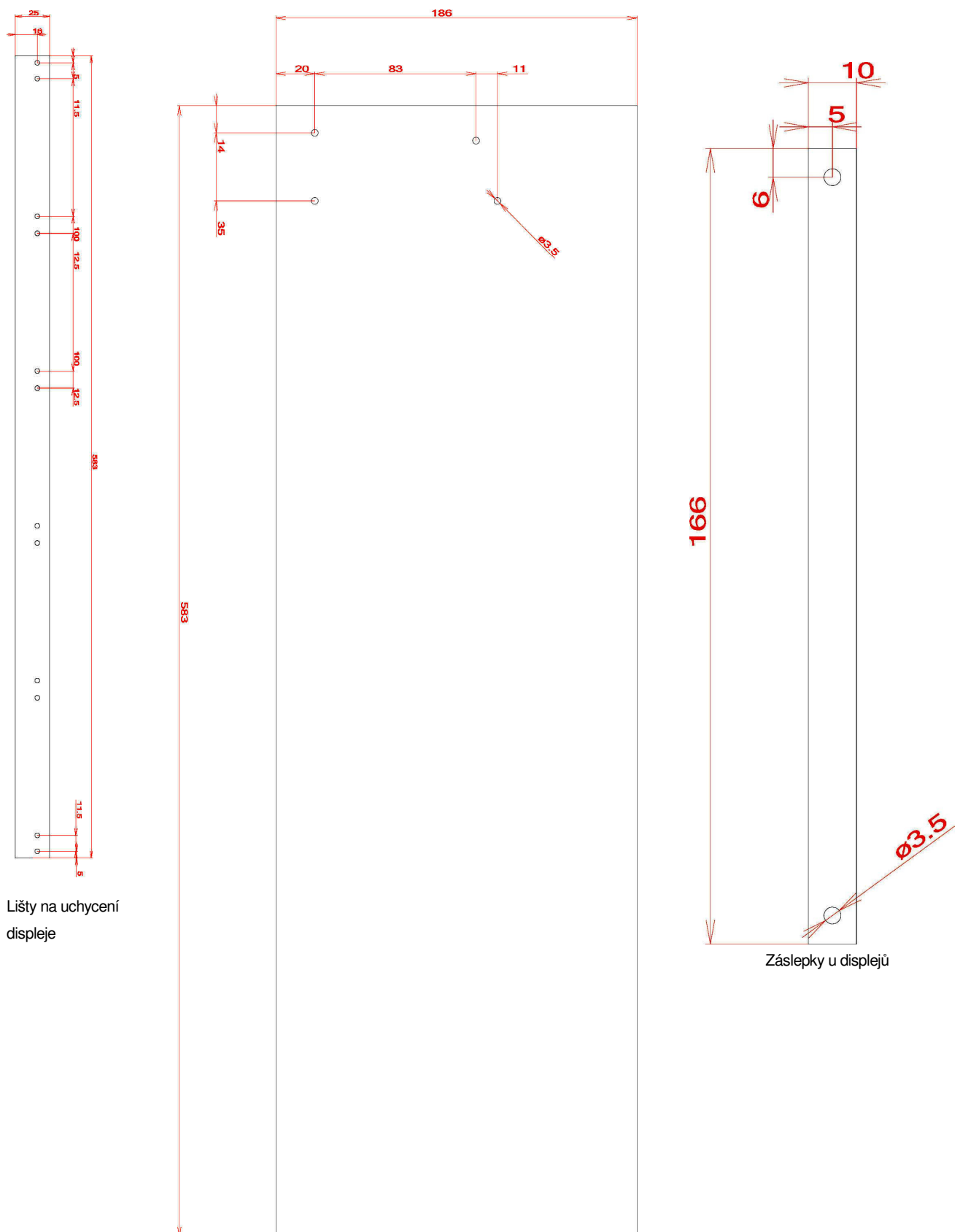
```
    for(i=0;i<5;i++)  
        DISP_RCK = 0;  
}
```

```
void nulovani(void)  
{  
    st_sp = 0;  
    setiny=0;  
    desetiny=0;  
    sekundy=0;  
    dessek=0;  
    minuty=0;  
    vykresli(0,0,0,0,0);  
}
```

```
__interrupt void isrVtpm1ovf(void)  
{  
    TPM1SC_TOF = 0;  
    if ((st_sp != 0) && (minuty != max)) {  
        setiny ++;  
        if (setiny >= 10)  
        {  
            desetiny++;  
            setiny = 0;  
        }  
        if (desetiny >= 10)  
        {  
            sekundy++;  
            desetiny = 0;  
        }  
        if (sekundy >= 10)  
        {  
            dessek++;  
            sekundy = 0;  
        }  
        if (dessek >= 6)  
        {  
            minuty++;  
            dessek = 0;  
        }  
        if (minuty >= max)  
        {  
            st_sp=0;  
        }  
        vykresli(setiny, desetiny, sekundy, dessek, minuty);  
    }  
}
```

přílohy

Výkresy boxu :

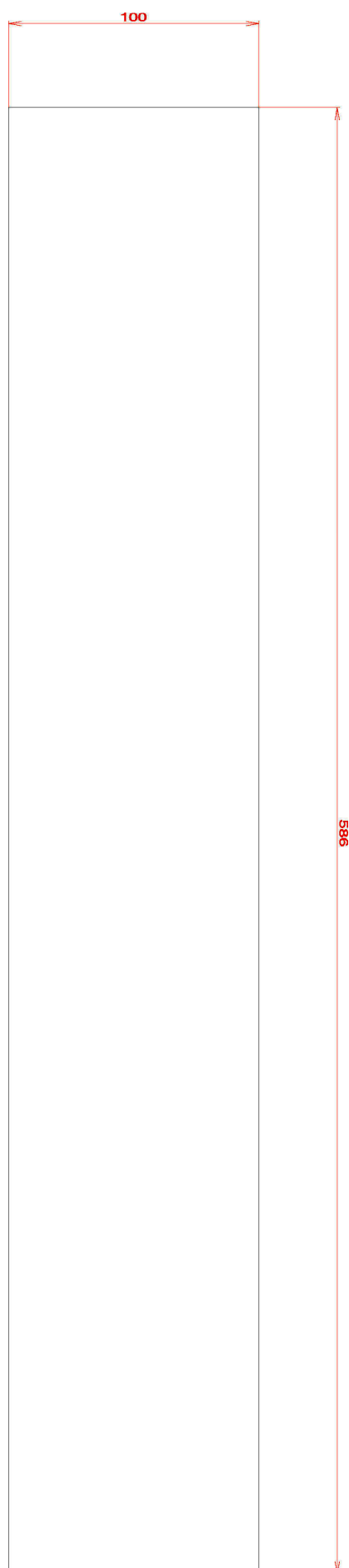


Lišty na uchycení
displeje

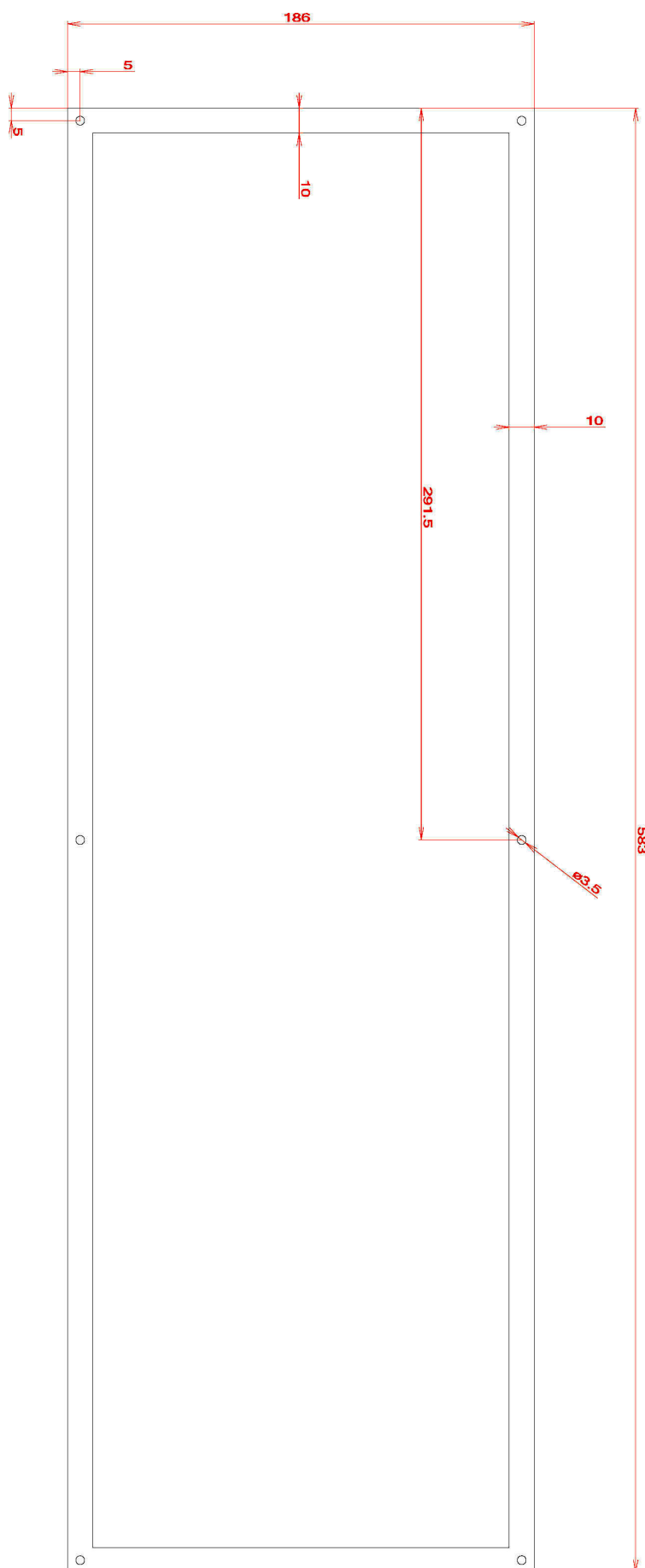
Záslepky u displejů

Zadní strana

přílohy



Horní, spodní strana



Čelní strana

přílohy



Boční strany