Středoškolská technika 2014

Setkání a prezentace prací středoškolských studentů na ČVUT

TEPLOMĚR A HODINY S ATMEGA168

Lukáš Tatarin

Střední průmyslová škola elektrotechniky a informatiky, Ostrava,
příspěvková organizace
Kratochvílova, 7/1490, Ostrava - Moravská Ostrava, 702 00

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor SOČ: 10. Elektrotechnika, elektronika a telekomunikace

Teploměr a hodiny s ATmega168

Thermometer and clock with ATmega168

Autor: Lukáš Tatarin

Škola: Střední průmyslová škola elektrotechniky a informatiky, Ostrava

Ostrava 2014

PROHLÁŠENÍ

Prohlašuji, že jsem práci vypracoval samostatně, použil jsem pouze podklady (literaturu, SW atd.) uvedené v přiloženém seznamu a postup při zpracování a dalším nakládání s prací je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů v platném znění.

V Ostravě dne 7.3.2014

X

Lukáš Tatarin
autor

PODĚKOVÁNÍ

Chtěl bych poděkovat Lukáši Herudkovi za zapůjčení programátoru, bez kterého by nebylo možné projekt dokončit.

ANOTACE

Cílem této konstrukce je jednoduchý a levný teploměr, který dokáže měřit teplotu a navíc počítat ještě datum a čas. Má široké spektrum využití, především v domácnosti a domácí automatizaci. Pro měření, přepočítávání a vyhodnocování výsledků je použit mikroprocesor ATmega168 naprogramovaný v jazyce C. Při návrhu a konstrukci jsme si rozšířili znalosti v oblasti elektrotechniky, zvláště pak programování mikroprocesorů.

Klíčová slova: ATmega168; teploměr; hodiny; jazyk C; DS18B20.

ABSTRACT:

The aim of this project was to construct a simple and cheap thermometer which can measure temperature, date and time, too. It has a wide range of usage in households in general and in so called home automation. ATmega8 microchip is used to measure, recalculate and interpret the results. While making this project I extended my knowledge of electronics, especially programming of microprocessors.
Key words: ATmega168; thermometer; clock; programing language C; DS18B20.

OBSAH“

ÚVOD

Výsledkem projektu je digitální teploměr s hodinami a datem. Veškeré informace jsou zobrazovány na LCD displeji se standardním rozlišením šestnácti znaků na dvou řádcích, běžně označovaném jako 16x2 či 2x16. K dispozici je rovněž údaj o aktuální hodnotě napětí na vývodech mikrokontroléru. Jako teplotní čidlo bylo zvoleno digitální, konkrétně DS18B20, které je velmi rozšířené a populární. Od výroby je kalibrováno na přesnost +/- 0,5°C, což pro většinu účelů postačuje. Komunikace s tímto čidlem probíhá po sběrnici 1-Wire, tedy prostřednictvím pouze jednoho kabelu. Čidlo má celkem 3 vývody, kladné napájecí napětí, připojení nulového potenciálu a již zmíněný datový pin. Srdcem celého zařízení je mikrokontrolér ATMEGA168, který vyrábí firma Atmel. Tento mikrokontrolér je velmi často používán, neboť má mnoho zabudovaných periferií a mnoho místa v paměti. Základními charakteristikami jsou: 16 kB FLASH paměti, 1 kB paměti SRAM a 512 B paměti EEPROM. Dále disponuje několika přerušeními, zabudovanou jednotkou UART, SPI sběrnicí nebo např. několika analogově-digitálními převodníky (ADC). Výstup jednoho kanálu ADC je zde připojen na odporový dělič, pomocí kterého snímá velikost napětí. Veškerý firmware je napsán v jazyce C, který je jednoduchý a velmi srozumitelný. Na projektu jsem si vyzkoušel práci s čítači/časovači v rodině 8-mi bitových mikrokontrolérů AVR.

1 HARDWARE

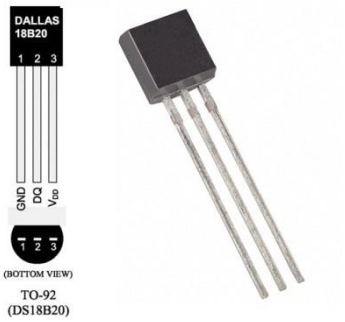
1.1 Mikropočítač ATmega 168

Pro můj projekt jsem potřeboval zvolit vhodný mikroprocesor, který hlavně disponoval velkou pamětí FLASH, tedy pamětí pro uložení programu a to proto, že knihovna, pro ovládání digitálního teplotního čidla je značně velká. Jiné požadavky, které by se týkaly rychlosti mikropočítače, nebo velikosti SRAM paměti jsem již nekladl nějak zvlášť vysoké. Jako každý jiný procesor v této výkonové i cenové kategorii disponuje také několika časovači, PWM kanály AD převodníky (ADC)…atd. Zde jsem navíc využil časovač, který počítá čas a datum a také funkci ADC, kterým kontroluji napětí celého zařízení. Pro realizaci komunikace s LCD displejem a teplotním čidlem jsem žádné zvláštní dispozice mikrokontroléru nepotřeboval. Mikroprocesor je v pouzdře TQFP 32 určené k montáži SMT technologii.

Obrázek 1: Mikroprocesor

1.2 Teplotní čidlo DALLAS DS18B20

Měření teploty lze realizovat několika způsoby. V jednodušších konstrukcích je využito parazitní teplotní závislosti PN přechodu, kde je takový výsledek převáděn již hotovými obvody. Chtěl jsem ale dosáhnout lepších a přesnějších výsledků, a proto naše volba padla právě na digitální teplotní čidlo od firmy Dallas, které komunikuje s okolím zcela digitálně, a to s přesností na 0,1°C. Digitální komunikace spočívá v zcela unikátní „1-WIRE“ sběrnici, vyvinutou právě firmou Dallas. Je založena na velmi přesném časování, kterým se vymezuje prostor, kdy komunikuje MASTER, což je v tomto případě mikropočítač, a kdy komunikuje SLAVE tedy čidlo.



Obrázek 2: Teplotní čidlo Dallas

1.3 LCD displej s řadičem HD44780

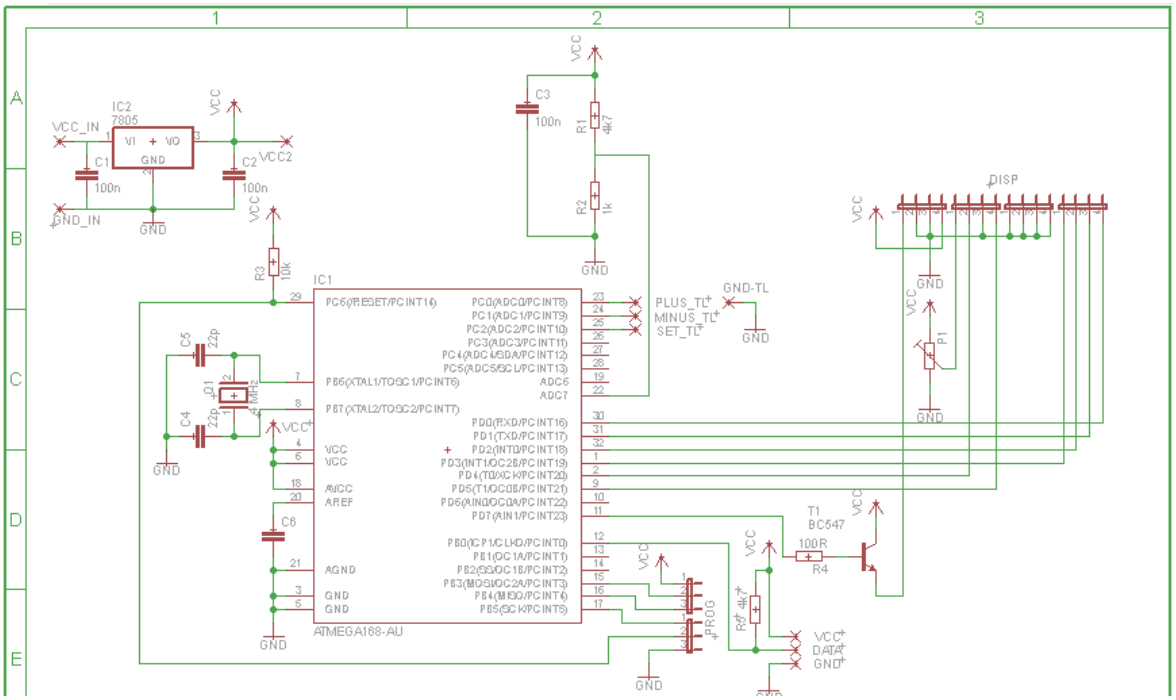
Bylo zcela jednoznačné použití oblíbeného a známého LCD displeje s 2x16 znaky a řadičem HD44780 s vlastním znakovým fontem a CGRAM pamětí pro uložení 16 vlastních znaků. V mém zapojení, byla využita pro českou diakritiku., ale lze do ní zapsat například i indikační symboly nebo značky o rozměru 5x8 pixelů.

Obrázek 3: LCD displej

1.4 Ostatní součástky

Zbytek konstrukce již není nijak zvláštní. Napájení zajišťuje stabilizovaný zdroj napětí 5V tvořený stabilizátorem 7805 v pouzdře TO-220. Původně byl použit stabilizátor v SMD pouzdře, ale jeho výkon nebyl dostačující. Doplňujícím zapojením této částí jsou kondenzátory C1 a C2. Kondenzátor C3 je blokovací kondenzátor, Rezistory R1 a R2 tvoří

napěťový dělič na jehož výstupu měříme AD převodníkem napájecí napětí a to zobrazujeme na LCD displeji. Rezistor R3 je tzv. pull-up rezistor, který drží log.1 na resetovacím vstupu mikroprocesoru. Rezistor R4 zmenšuje proud procházející do báze tranzistoru T1, který zapíná a vypíná podsvětlení závisle na velikosti napájecího napětí. Když například napájíme zařízení z baterie, tak v momentě, kdy začne napětí klesat mikropočítač vypne podsvětlení displeje a sníží tím výrazně proudový odběr celého zařízení. Rezistor



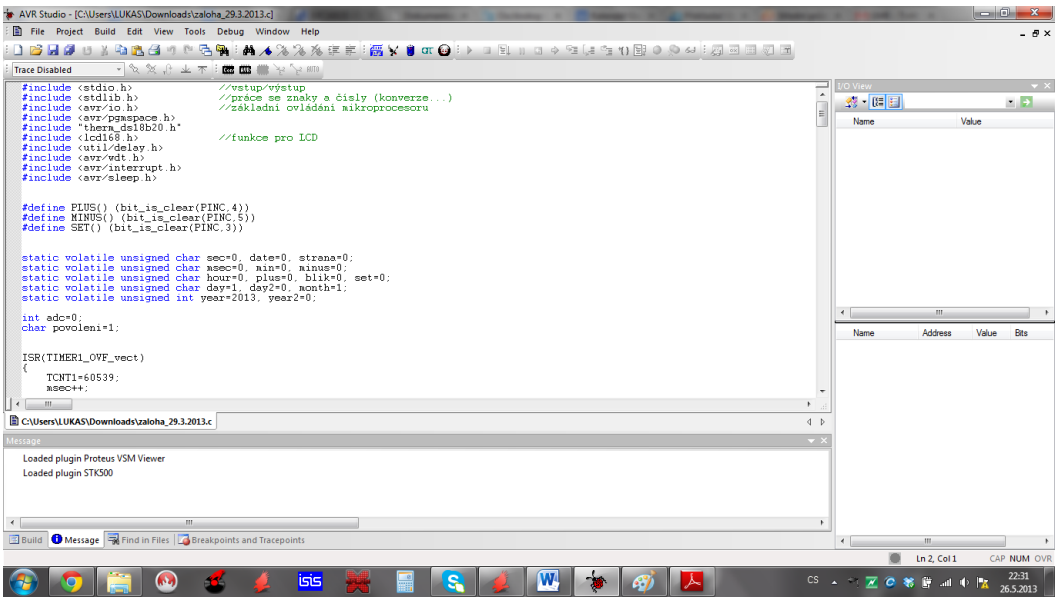
Obrázek 4: Schéma

R5 je opět pull-up rezistor, jenž drží log. 1 na datovém vývodu teplotního čidla. Jedná se o doporučení výrobce. Trimr P1 reguluje kontrast displeje. Svým způsobem se opět jedná o napěťový dělič. Kondenzátory C4, C5 a krystal Q1 tvoří oscilátor, který udává taktovací kmitočet mikroprocesoru. Ten jsme zvolili na 4 MHz. Kondenzátor C6 je zapojen na externí referenci ADC- Opět se jedná o doporučené zapojení výrobce.

2 FIRMWARE

2.1 Vývojové studio

S volbou programovacího jazyka C a výběrem mikrokontroléru ATMEL padla zároveň i volba na vývojové prostředí AVR Studio 4, které je zcela volně dostupné na oficiálních stránkách www.atmel.com. Samotné prostředí má v sobě zabudovaný překladač ASM. Pro práci se studiem v jazyce C je třeba doinstalovat překladač WinAvr, nebo AVR Toolchain. Posléze již můžeme po založení a nastavení nového projektu začít psát zdrojový kód. Prostředí rovněž umožňuje simulaci a odladění celého programu.



Obrázek 5: Vývojové studio

2.2 Hardwarové prostředky

Po návrhu a zapsání programu v jazyce C je třeba jej nahrát do procesoru. To se provádí pomocí speciálního zařízení. Programátoru. Ten si můžeme buďto zakoupit nebo postavit. Z časových důvodů jsem pořádal Lukáše Herudka, který mi programátor zapůjčil. Zařízení, které jsem použil je klonem tzv. USB-ASP programátoru od německého autora Thomase Fischle. Všechny podklady pro jeho konstrukci jsou volně stáhnutelné na webu: www.fischl.de.

Programátor zapisuje program pomocí ISP rozhraní. To znamená, že využívá linek MOSI, MISO, SCK a RST. RST je resetovací linka- provádí reset mikrokontroléru. SCK je hodinový signál, kterým se určuje rychlost zápisu programu. MOSI a MISO (Master Out

Salve In a naopak) vysílají data, které se zapisují do FLASH paměti procesoru. Program, který je zkompilovaný a ve formátu Intel HEX je možno do procesoru nahrát. To se děje jednak pomocí programátoru a jednak pomocí zapisovacího prostředí. V našem případě se jedná o freewarový program PROGISP (Ver.1.72).

V nahrávacím prostředí otevřeme daný soubor ve formátu *.hex a vybereme typ mikroprocesoru. Můžeme zde také nastavit další parametry jako například vymazání čipu, verifikace zápisu…atd. Musíme zde také zkontrolovat tzv. „fuses“ neboli pojistky. Ty určují a ovládají interní periférie mikroprocesoru. Jedná se například o povolení interního oscilátoru, vypnutí JTAG, nebo vypnutí resetovacího pinu. Tady tahle část je obzvlášť důležitá, protože tímto krokem můžeme vše pokazit! Špatný zápis pojistek může i znamenat nefunkčnost procesoru, například vypnutí resetovacího pinu, změnou interního oscilátoru nebo dokonce zcela špatný zápis.

2.3 Popis programu

Jako první jsou do programu vloženy knihovny, které obsahují instrukce pro ovládání jednotlivých periférii mikroprocesoru a ovládání LCD a teplotního čidla. Zapisují se ve formátu <xxx.h> pro headery vytvořené výrobcem i “xxx.h“ pro soubory vytvořené uživatelem.

```
#include <stdio.h>              //vstup/výstup
#include <stdlib.h>             //práce se znaky a čísly (konverze...)
#include <avr/io.h>             //základní ovládání mikroprocesoru
#include <avr/pgmspace.h>
#include "therm_ds18b20.h"
#include <lcd168.h>             //funkce pro LCD
#include <util/delay.h>
#include <avr/wdt.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>
```

Následují jednotlivé definice proměnných a maker pro práci se spínači

```
#define PLUS() (bit_is_clear(PINC,4))
#define MINUS() (bit_is_clear(PINC,5))
#define SET() (bit_is_clear(PINC,3))


static volatile unsigned char sec=0, date=0, strana=0;
static volatile unsigned char msec=0, min=0, minus=0;
static volatile unsigned char hour=0, plus=0, blik=0, set=0;
static volatile unsigned char day=1, day2=0, month=1;
static volatile unsigned int year=2013, year2=0;

int adc=0;
char povoleni=1;
```

V následující funkci je řešena obsluha přerušení, konkrétně obsluha časovače, kterým počítáme hodiny a datum. Ty jsou následně zobrazeny na displej. Stejným časovačem také řešíme blikání údajů při nastavování a častost měření hodnot ADC.

```
ISR(TIMER1_OVF_vect)
{
    TCNT1=60539;
    msec++;

    if(msec==50)
    {
        adc=1;
        blik=1;
    }
    else if(msec==100)
    {
        if(set==0)
            sec++;

        msec=0;
        adc=1;
        blik=0;
    }
}
```

Zde můžeme vidět pouze inicializační data portů a periferií mikroprocesoru:

```
void ioinit (void)
{
   DDRC  = 0b00000000; //1 = output, 0 = input
   PORTC = 0b11111111; //Enable pin 5,4,3 internal pullup

   DDRD = 0xFF;
   PORTD |= 0b10000000;
}

int ReadADC(uint8_t __channel)
{
    ADMUX = __channel;                // Channel selection
    ADMUX |= 0b11000000;              //1.1 V reference

    ADCSRA |= _BV(ADSC);               // Start conversion
    while(!bit_is_set(ADCSRA,ADIF));   // Loop until conversion is complete
    ADCSRA |= _BV(ADIF);               // Clear ADIF by writing a 1 (this sets the value to 0)

    return(ADC);
}
```

Následujícím blokem programu je hlavní funkce. Zde se už jen řeší běh celého programu a výpis na displej. Za zmínku by snad ještě stála ukázka ošetření chronologických zvláštností. Zejména jevu 29. února…atd.

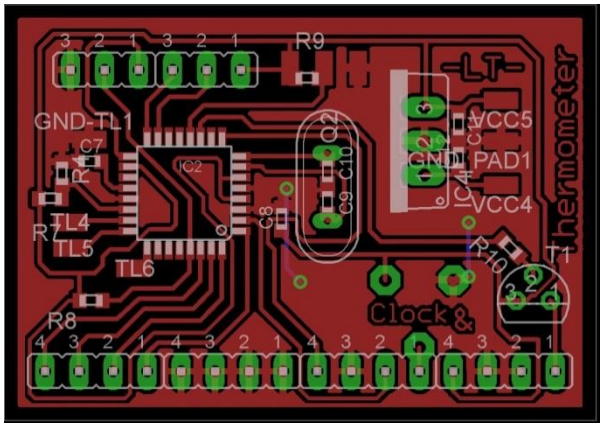
```
if((month==1 || month==3 || month==5 || month==7 || month==8 || month==10) && day>31)
{
    day=1;
    month++;
}

if((month==4 || month==6 || month==9 || month==11) && day>30)
{
    day=1;
    month++;
}

if((month==2) && ((day>=30 && year%4==0) || (day>=29 && year%4!=0)))
{
    day=1;
    month++;
}

if(month>=12 && day>31)
{
    day=1;
    month=1;
    year++;
}
```

Z popisu programu je to již vše.

3 KONSTRUKCE

3.1 Návrh desky plošných spojů

DPS je zcela náš vlastní návrh. Zvolil jsem si program EAGEL od společnosti CadSoft. Tento program není zcela volně šiřitelný, ale v základní demoverzi je pouze omezení velikosti DPS a to na 10x15 palců. To ale vůbec nevadí. Jeho instalace je zcela obdobná jako u jiných aplikací. Nejprve je vhodné vytvořit si schéma, a podle něj teprve DPS. Jedině tehdy je zcela vyloučeno splést se v návrhu DPS. Musíme však dávat pozor na to, aby se nám i ve schématu patřičné linky spojily. Program se vždy zeptá a teprve potom je návrh přesně podle schématu stoprocentní. Myslím, že zmiňovat zakreslení schématu je zbytečné, neboť se zde nevyskytují žádná zvláštnosti. Na návrh DPS byl snad kladen jen jediný požadavek a to takový, aby nebyla překročena velikost LCD displeje. Zvolili jsme tedy technologii SMT a diskrétní součástky o velikosti 0603. Tedy ty nejmenší, se kterýma se v amatérských podmínkách dá pracovat.



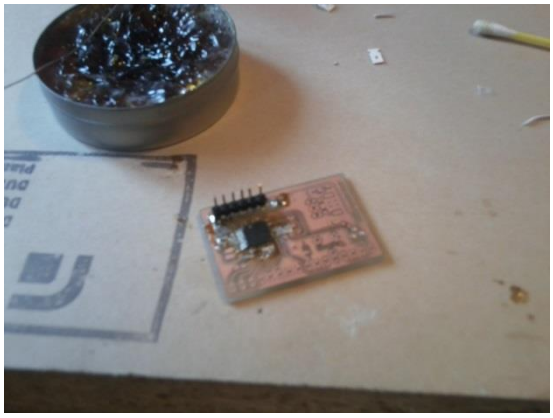
Obrázek 6: Deska plošných spojů

3.2 Výroba DPS

DPS byla vyrobena domácí technologií fotocesty. Tedy zakoupením fotocitlivého polotovaru plošného spoje a následným vytištěním a prosvětlením motivu na fotocitlivou desku cuprextitu. Taková to deska se na krátký čas vloží do pozitivní foto-emulze a neosvětlený motiv tímto zůstane na desce. Následuje již známé leptání roztokem chloridu železitého a závěrečných úprav, jako je například lakování vrtání broušení hran…atd.

3.3 Pájení

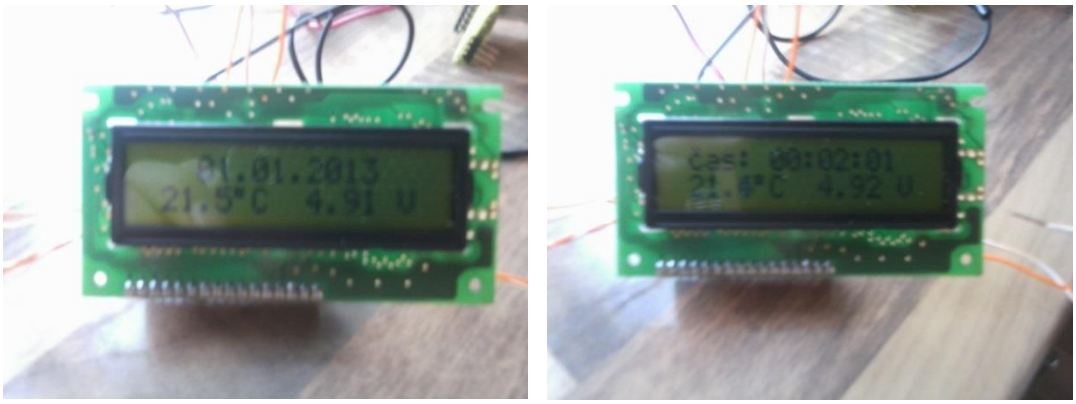
Pájení konstrukce s SMT technologií vyžaduje značnou trpělivost. Nejprve je dobré osadit mikrokontrolér. K zapájení všech jeho 32 vývodů potřebujeme prostor, který by nám již dříve osazený spoj ostatními součástkami neumožnil. Po jeho osazení doporučujeme zkontrolovat veškeré vývody, zda-li nejsou někde zkratovány, a tyto případné chyby odstranit pomocí tzv. lícny. Je vhodné také lehce připájet programovací konektor a odzkoušet, jestli po zapájení probíhá komunikace s okolím. Jestli je vše v pořádku můžeme pokračovat dále. Osadíme tedy nejprve rezistory, dále kondenzátory a až nakonec zapájíme již zcela programovací konektor, lištu na připojení displeje, trimr a tranzistor, které se pájí zespod, drátky pro připojení čidla, spínačů a napájení a až na samotný konec teprve krystal. Ten musíme pájet s velikou opatrností a ne déle než 2 vteřiny!



Obrázek 7: Pájení obrázku

3.4 Oživení a ovládání

Zařízení oživíme přivedením napájení a to buď stabilizovaného přes programovací konektor, nebo nestabilizovaného v rozsahu 6,5 až 18V. Přivedení vyššího napětí způsobuje vysoké tepelné ztráty, nebo dokonce destrukci stabilizátoru a posléze i celého zařízení. Zařízení ovládáme pomocí třech nastavovacích tlačítek. Tlačítkem „SET“ uvedeme teploměr do nastavovacího módu. Pomocí tlačítek „PLUS“ a „MINUS“ nastavíme patřičnou hodnotu a opět tlačítkem „SET“ ji potvrdíme a dostaneme se do nastavení další hodnoty. Kalibrace zařízení není nutná.



Obrázek 8: Oživení a ovládání

3.5 Umístění do krabičky

Pro umístění konstrukce do krabičky byl zvolen polotovar-plastový box, do kterého bylo potřeba vyvrtat všechny otvory včetně toho na displej. Pro vyvrtání byl vytvořen technický výkres, podle kterého se zrealizovala výroba. Po celém obvodu displeje byly naznačeny a posléze vrtákem o průměru 1,5 a 3 mm vyvrtány otvory tak, aby byly od reálné hrany vzdáleny 2mm. Vrtání se provádí nejlépe elektrickou vrtačkou o otáčkách asi 1200-1800 otáček za minutu a ostrým vrtákem. Tupý vrták a nízké otáčky způsobují zakousnutí nástroje praskání krabičky. Zase naopak příliš vysoké otáčky již plast skoro tepelně rozkládají. Po vyvrtání je třeba sekáčem proseknout prostor mezi jednotlivými otvory a tím vyjmem asi 95% celkového prostoru pro usazení displeje. V závěrečné fázi pilníkem opracujeme obdélníkový otvor na patřičný tvar, umístíme LCD displej a nalepíme potisk.

Obrázek 9: umístění konstrukce do krabičky

ZÁVĚR

Podařilo se mi vytvořit vlastní teploměr, který dokáže změřit základní veličiny. Vývoj zařízení nebyl zcela jednoduchý. Musel jsem vyřešit způsob zobrazování vlastních znaků. Zde prošel software několika rozsáhlými úpravami, které zabrali spoustu času. Cena takového teploměru se pohybuje kolem 400,- Kč. Na tomto projektu jsem si ověřil a také zdokonalil své znalosti z oblasti techniky, elektroniky, programování, ale také manuálních zručností.

SEZNAM POUŽITÝCH ZDROJŮ

[1] [online]. 2012-09-03 [cit. 2014-03-01]. Dostupné z: http://www.nongnu.org/avr-libc/user-manual/group__avr__interrupts.html

[2] ZAVODSKY, Ondrej. Programujeme AVR v jazyku C - 1. časť: Úvod k mikropočítačom Atmel AVR, inštalácia a vytvorenie prvého projektu v prostredí AVR Studio 4. ZAVODSKY, Ondrej. *SVETELEKTRO.COM* [online]. 2012 [cit. 2014-03-01]. Dostupné z: http://svetelektro.com/clanky/programujeme-avr-v-jazyku-c-1-cast-443.html

[3] ZAVODSKY, Ondrej. Programujeme AVR v jazyku C - 2. časť: Začiatky písania kódu, práca s registrami, maskovanie, zapojenie mikrokontroléra, jeho naprogramovanie a test prvého programu. ZAVODSKY, Ondrej. *SVETELEKTRO.COM* [online]. 2012 [cit. 2014-03-01]. Dostupné z: http://svetelektro.com/clanky/programujeme-avr-v-jazyku-c-2-cast-446.html

[4] ZAVODSKY, Ondrej. Programujeme AVR v jazyku C - 3. časť: Popis funkcie prerušenia, práca s externým prerušením. ZAVODSKY, Ondrej. *SVETELEKTRO.COM* [online]. 2012 [cit. 2014-03-01]. Dostupné z: http://svetelektro.com/clanky/programujeme-avr-v-jazyku-c-3-cast-449.html

[5] ZAVODSKY, Ondrej. Programujeme AVR v jazyku C - 4. časť: Čítač/časovač a jeho použitie. Časovanie udalostí, čítač externých impulzov, meranie periódy atď... ZAVODSKY, Ondrej. *SVETELEKTRO.COM* [online]. 2012 [cit. 2014-03-01]. Dostupné z: http://svetelektro.com/clanky/programujeme-avr-v-jazyku-c-4-cast-453.html

[6] ZAVODSKY, Ondrej. Programujeme AVR v jazyku C - 6. časť: V dalších častiach seriálu budeme potrebovať zobrazovať rôzne údaje, preto som sa rozhodol v tejto časti bližšie venovať pripojeniu LCD displeja ku mikrokontroléru a jeho obsluhe v programe.

ZAVODSKY, Ondrej. *SVETELEKTRO.COM* [online]. 2012 [cit. 2014-03-01]. Dostupné z: http://svetelektro.com/clanky/programujeme-avr-v-jazyku-c-6-cast-459.html

[7] ZAVODSKY, Ondrej. Programujeme AVR v jazyku C - 9. časť: Rozhranie SPI, popis registrov a ukážkový program. ZAVODSKY, Ondrej. *SVETELEKTRO.COM* [online]. 2012 [cit. 2014-03-01]. Dostupné z: http://svetelektro.com/clanky/programujeme-avr-v-jazyku-c-9-cast-528.html

[8] GM ELECTRONIC. *GM Electronic* [online]. [cit. 9.3.2014]. Dostupný na WWW: http://www.gme.cz/atmega168-20pu-p432-192

[9] PROTOSTACK PTY LTD. *PROTOSTAC* [online]. [cit. 9.3.2014]. Dostupný na WWW: www.protostack.com

[10] APEX. *RC162021YFHLYB DATA SHEET ISSUE VERSION APPROVER CHECKER ENG* [online]. 03-04-07. Chungho City, 2003, 17 s. [cit. 2014-03-01]. Dostupné z: http://mujweb.cz/hezky.den/datasheet/RC162021YFHLYB.pdf

[11] HÄMMERLING, Mark. Engbedded Atmel AVR® Fuse Calculator. HÄMMERLING, Mark. *Engbedded: enslouling circuits* [online]. 2010 [cit. 2014-03-01]. Dostupné z: http://www.engbedded.com/