



## **Středoškolská technika 2011**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

### **Digitální teploměr**

**Lukáš Löwinger, Tomáš Veselý**

Střední průmyslová škola sdělovací techniky  
Panská 856/3, Praha 1

## **Anotace**

Naším úkolem je vytvořit digitální teploměr, který bude ovládán mikrokontrolérem PIC. Bude jej možné používat místo digitálního teploměru, který je v domácnostech běžný. Teplotu bude získávat ze dvou čidel, a to z vnitřního, které bude analogové, a z vnějšího - digitálního. Dva rozdílné druhy čidel jsme zvolili proto, abychom výrobek vylepšili a v praxi ukázali různé možnosti měření. Teploměr bude mimo jiné vybaven zvukovými signály, bude poskytovat možnost vyčtení naměřených hodnot z paměťové karty a jejich vyhodnocení (minimální, maximální a průměrná teplota). Díky použití paměťové karty je možné přenést data do počítače a dále je zpracovávat. Úlohu jsme museli vyřešit jak po hardwarové, tak po softwarové stránce.

## **Annotation**

Our task is to create a digital thermometer controlled by the PIC microcontroller. It can be used instead of a home digital thermometer common in households. Temperature is obtained from two sensors, inner analog and the outer digital. We have chosen two different types of sensors to improve the product and to show the different possibilities of measurement in a practice. The thermometer is equipped with a sound signal and also provide an opportunity to read the measured values from the memory card and to interpret them (minimal, maximal and average temperature). Usage of the memory card allows to transfer data to the computer and process them additionally. We had to solve both hardware and software aspects of this task.

## **Poděkování**

Děkujeme vedoucímu dlouhodobé maturitní práce Ing. Janu Tučkovi, CSc., za obětavý přístup a cenné rady při zpracování práce. Taktéž bychom chtěli poděkovat Mgr. Jaroslavu Reichlovi za zprostředkování měřícího přístroje LabQuest a za poskytnutí materiálů k pochopení lineární regrese.

## Obsah

1	Úvod, slova autorů .....	6
2	Historie teploměru.....	7
3	Součástky .....	10
3.1	Analogové čidlo TMP36GT9Z .....	10
3.1.1	Technické specifikace čidel TMP .....	11
3.1.2	Proč toto čidlo ?.....	14
3.2	Digitální senzor DS18B20 .....	14
3.2.1	Technické specifikace DS18B20.....	15
3.2.2	Rozhraní 1-Wire™ .....	15
3.3	Mikroprocesor PIC18LF2520 .....	18
3.3.1	Technické specifikace PIC18LF2520 .....	19
3.4	Znakový LCD MC1604B-SYL.....	21
4	Popis zapojení .....	26
5	Výroba DPS.....	27
6	Postup měření.....	28
7	Matematické zpracování naměřených hodnot.....	29
7.1	Lineární regrese.....	29
7.1.1	Zpracování dat pomocí programu Mathematica .....	33
8	Komunikace s paměťovou kartou .....	35
8.1	SD/MMC karta.....	35
8.1.1	Inicializace MMC/SD karty .....	35
8.1.2	Formát odesílání příkazu .....	36
8.1.3	Odeslání dat.....	37
8.1.4	Příjem dat .....	37
8.1.5	Formát dat .....	38
8.2	Souborový systém FAT.....	38
8.2.1	Boot sektor .....	39
8.2.2	FAT .....	40
8.2.3	Root adresář.....	41
9	Popis programu .....	42
10	Ovládání teploměru .....	45
11	Výběr krabičky.....	48
11.1	Úpravy krabičky.....	49

12	Závěr.....	54
13	Seznam použité literatury a zdrojů informací .....	55
13.1	Seznam použitých literárních publikací .....	55
13.2	Jiné zdroje informací.....	55
14	Použitý software.....	57
15	Přílohy .....	58

# 1 Úvod, slova autorů

Za cíl jsme si kladli navrhnout a zkonstruovat digitální teploměr, který bude měřit teplotu v rozsahu  $-40\text{ }^{\circ}\text{C}$  až  $+100\text{ }^{\circ}\text{C}$ . Jeho teplotní čidlo bude jak analogové, tak i digitální. Mozkem celého výrobku bude mikroprocesor PIC, který obstará veškeré funkce. Naměřené hodnoty čidlem (digitálním nebo analogovým) bude možné ukládat na SD kartu, vhodnou pro přenos k dalšímu vyhodnocování na počítači. Mikrokontrolér se stane řídicím prvkem i LCD displeje, kde uvidíme právě měřené hodnoty zároveň s informačními prvky (datum, čas, teplotní omezení,...).

Konstrukce výrobku začíná osazením plošného spoje, který jsme museli nejprve navrhnout a vyrobit. Plošný spoj s uvedenými, ale i dalšími součástkami, které budou připájené, umístíme do hliníkové krabičky. Na ní bude viditelný jen LCD displej, dále 5 tlačítek pro ovládání, otvor na SD kartu, vypínač a konektory pro napájení a pro vnější čidlo. Tlačítka budou sloužit k obsluze PIC, který řídí celý teploměr. Napájení bude možné jak z jedné baterie (9 V), tak ze sítě (230 V).

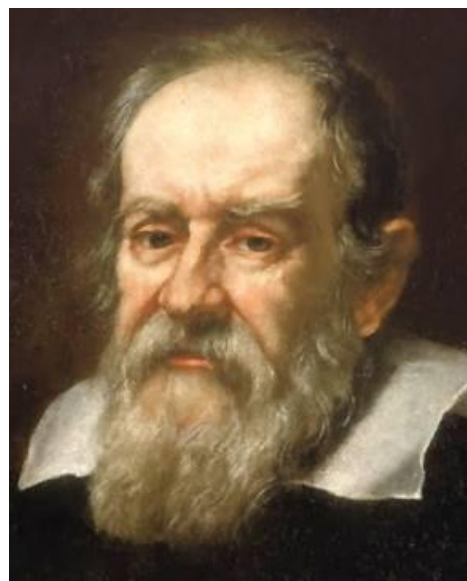
## 2 Historie teploměru

Před několika staletími se nikdo o teploměru ani nebavil. Teplotu lidé určovali podle tělesných pocitů nebo například podle barvy, což se projevovalo nejvíce u výroby kovů a keramiky. Podle toho, jak moc byly předměty rozžhaveny, se měnila jejich barva. Ale lidé už ve starověku věděli, že teplem se vzduch roztahuje a ochlazováním smršťuje. Tohoto faktu se držel i



Obr. 2

Nejprve baňku zahřál rukou a *vzduchový teploměr* vložil otevřeným koncem trubičky do nádoby s obarvenou vodou. Chladnoucí vzduch se smršťoval a vlivem tlaku okolního vzduchu na hladinu voda vnikala do trubičky. Po vychladnutí přejímala baňka teplotu okolního vzduchu a výška vodního sloupce v trubičce se měnila podle změn objemu vzduchu v baňce, který se zase měnil podle teploty vzduchu. Na rozdíl od dnešních teploměrů při oteplení hladina klesala a při ochlazení stoupala. Galileův teploměr se rozšířil díky Sanctorisovi, který byl lékařem. Po Galileovi experimentovali s podobnými teploměry Otto von Guericke a Gaspar Schott. Zdokonalili *termoskop* tím, že použili dvou trubiček se dvěma baňkami, které spojili třetí trubičkou ve tvaru U, v níž byla tekutina. Ještě v témže století se objevují teploměry, v nichž je teploměrnou látkou kapalina. Zřejmě první sestrojil roku 1631 francouzský lékař, který použil jako teploměrnou látku právě vodu. Nevýhodou tohoto teploměru byla malá roztažnost vody a vlastnost, že velmi brzo začíná mrznout. Proto se hledaly jiné vhodné tekutiny. Jako nejvhodnější se ukázaly líh a rtuť [27]. První lihový



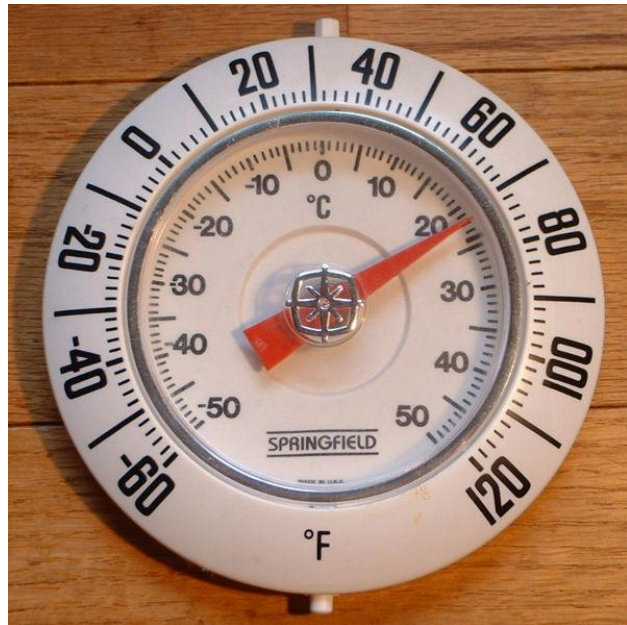
Obr. 1

Galileo Galilei (Obr.1), slavný profesor univerzity v Padově v Itálii, který na samém začátku 17.století využil tepelné

roztažnosti vzduchu k měření teploty.

Díky této fyzikální vlastnosti Galilei sestrojil první primitivní teploměr, nazývaný „vzduchový teploměr“ nebo „vzduchový termoskop“ [26]. Neměl stupnici, byl velmi nedokonalý, ale pomohl k rozvoji dalších teploměrů. Byl tvořený tenkou skleněnou trubičkou, dlouhou asi 30 cm, která byla zakončena baňkou. Jak zhruba výrobek vypadal, je znázorněno na obrázku (Obr.2).

teploměr sestrojil roku 1641 toskánský velkovévoda Ferdinand II. Zjistil, že teploměr s lihovou náplní má lepší vlastnosti a dokáže měřit v širším teplotním rozsahu než voda. V té době sice teploměry již měly stupnice, ty však nebyly jednotné, takže údaje změřené jednotlivými teploměry se nemohly porovnat. První teploměry s „normalizovanou“ stupnicí byly sestrojeny až kolem roku 1650. Vznikly díky vědecké společnosti Accademia del Cimento. Kromě jednotné



**Obr. 3 - Teploměr s Celsiovou a Fahrenheitovou stupnicí**

stupnice bylo také nutno stanovit její počátek. Anglický fyzik Robert Boyle stanovil v r. 1664 u svého teploměru jako základní bod stupnice teploty tání ledu. V roce 1665 určil další stálý bod stupnice holandský vědec Christian Huygens. Byla to teplota varu vody při normálním tlaku ovzduší (tj. 1013 hPa, neboť jak víme, s klesajícím tlakem klesá i bod varu vody). A tak Huygens navrhl, aby se za základ stupnice teploměru vzala buď teplota tání ledu nebo teplota varu vody, čímž vlastně navrhl způsob používaný dodnes. To však nebránilo Danielu Gabrielu Fahrenheitovi, který začal o nějakých 50 let později vyrábět lihové a posléze i rtuťové teploměry v Holandsku, aby si svérázně vybral za počátek stupnice svých teploměrů teplotu směsi ledu, vody a chloridu amonného. Právě díky této směsi naměřil nejnižší teplotu a definoval ji jako 0 °F. Druhá důležitá hodnota je teplota lidského těla, při které naměřil 96 °F. Později byly tyto referenční body upraveny na 32 °F pro bod mrazu vody a 212 °F bod varu vody. Tato stupnice se využívá v USA dodnes. René de Réaumur, pařížský zoolog, navrhl stupnici s nulou při teplotě tání ledu a s hodnotou 80 při teplotě varu lihu (později tato hodnota odpovídala teplotě varu vody). Švédský matematik a geodet Anders Celsius zavádí do měření desítkovou soustavu, kde teplota varu vody má číslo 0 a teplota tání ledu číslo 100. Hodnoty jsou oproti dnes používaným navzájem obráceny. Stupnici, kterou známe dnes, zavedl Carl Linné a dodnes se jmenuje Celsiova, jak ji každý z nás zná. Jednotkou této stupnice, jak víme, je teplotní stupeň Celsiův (°C). Je důležité vědět, že z fyzikálního hlediska je stupeň celzia vedlejší jednotkou soustavy SI. Přepočítání Fahrenheitů (F) na stupně Celsia (°C) je  $C = ([^{\circ}F] - 32) \times 5/9$ . Tento vztah lze odvodit, pokud si uvědomíme pár důležitých faktů. Jak již bylo řečeno, pro  $F = 32$  je  $C = 0$ . Z čehož vyplývá, že výsledný tvar musí mít tuto podobu  $C = (F-32)k$ , kde  $k$  je konstanta, kterou hledáme. Nalézt její



hodnotu není těžké, protože víme, že pro  $F = 212$  je  $C = 100$ . Dosazením těchto hodnot do již napsaného vzorce nám vyjde, že  $k = 5/9$ . Tudíž  $C = (F-32)*5/9$ .

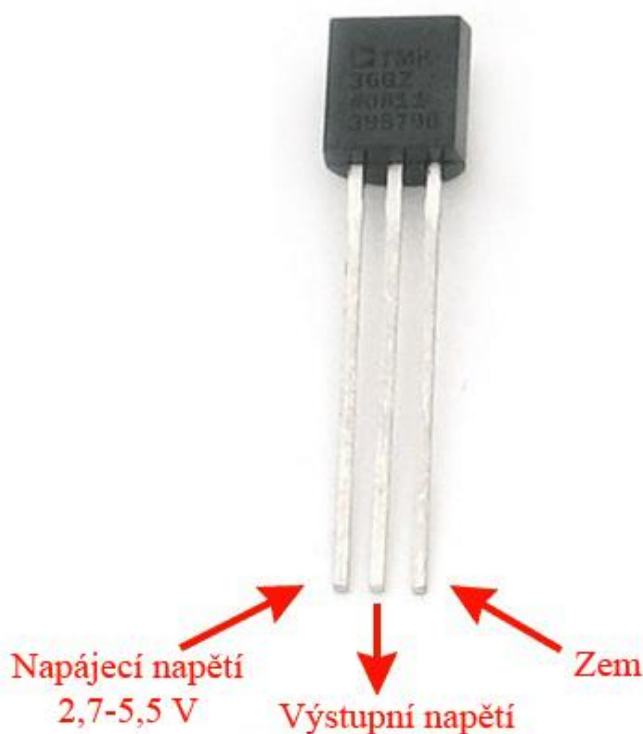
Všechny tyto teploměry jsou založeny na stejném principu, na teplotní roztažnosti látek (kapalin), a proto se jim říká *dilatační*. Celsiem byl jejich vývoj prakticky ukončen. V dalších letech byly už jen modifikovány jednotlivé druhy. Věda a technika nevystačila s klasickým rtuťovým teploměrem. Jeho nevýhodou bylo, že dokázal měřit teplotu v dost úzkém rozsahu. Vědci zjistili, že pokud se z kapiláry vysaje vzduch, a volný prostor se naplní dusíkem, vlastnosti rtuťového teploměru se zlepší. Takto upravený přístroj dokáže měřit teplotu až do 1100 stupňů. U lékařských rtuťových teploměrů (35 až 42 °C) je kapilára nad nádobkou se rtutí zúžena, takže se v tomto místě při poklesu teploty rtuťový sloupec přetrhne a teploměr tak stále ukazuje maximální naměřenou teplotu (před dalším použitím se musí „sklepnout“).

Později se objevily teploměry také na jiných fyzikálních principech. Například *odporové* teploměry měří teplotu změnou elektrického odporu vodiče nebo polovodiče (termistoru), *termoelektrické* teploměry jsou založeny na termoelektrickém jevu, kdy změnou teploty spoje dvou různých kovů se mění vzniklé termoelektrické napětí. Teplotu lze měřit i bezdotykově, např. pomocí tzv. *infrateploměrů*, které měří záření vysílané tělesy do okolí (na stejném principu pracují i známá světelná infračidla a bohužel i naváděné střely). Všechny tyto teploměry mohou mít kromě klasické analogové stupnice také stupnici digitální a hlavně mohou mít elektrický výstup, což umožňuje elektrické zpracování naměřených hodnot nebo přímo automatizované řízení procesu.

## 3 Součástky

### 3.1 Analogové čidlo TMP36GT9Z

Co je teplotní analogové čidlo? Je to součástka, která vám jednoduše řekne, jaká je teplota prostředí, ve kterém se čidlo nachází. Tyto senzory obecně užívají polovodičových prvků ke zjištění teploty. Nutno tedy podotknout, že nepoužívají rtuť (jako starší teploměry), bimetalické pásky (v některých domácích spotřebičích, např. v žehličce), ani termistor (elektrotechnická součástka, jejíž odpor je závislý na teplotě). Místo toho fungují tak, že při stoupaní teploty se zvyšuje napětí na diodě v určitém poměru (je to vlastně úbytek napětí mezi bází a emitorem tranzistoru) [31]. Přesným zesilováním této změny napětí lze jednoduše vygenerovat analogový signál, který je přímo úměrný teplotě. Protože tato analogová čidla nepotřebují žádné pohyblivé části, nikdy se neopotřebují, jsou přesná, nepotřebují kalibraci a dokáží pracovat v nejrůznějších prostředích. Navíc jsou velmi levná a mají docela snadné použití. Důvod, proč jsme si vybrali právě toto čidlo, je ten, že má veliký rozsah a nepotřebuje záporné napětí pro teploty pod bodem mrazu.



Obr. 4 - Analogové čidlo TMP36GT9Z

### 3.1.1 Technické specifikace čidel TMP

TMP35, TMP36 a TMP37 [31] patří do jedné rodiny teplotních čidel, která jsou nízkonapěťová, přesná a pracují s Celsiovou stupnicí. Poskytují výstupní napětí, které je lineárně úměrné stupňům Celsia. Nepotřebují žádné externí kalibrace a poskytují přesnost  $\pm 1$  °C při +25 °C a  $\pm 2$  °C od -40 °C do +125 °C teplotního rozsahu. Nízká výstupní impedance, lineární výstup a jejich přesná kalibrace zjednodušuje propojení s obvody, jejichž hlavním úkolem je rozpoznání teploty (např. teploměr) a s A/D převodníky. Všechna tři čidla jsou určena pro napájení stejnosměrným napětím od 2,7 V do 5,5 V. Napájecí proud se pohybuje pod 50  $\mu\text{A}$ , díky čemuž se čidlo neohřívá více jak o 0,1 °C za bezvětří. Kromě toho se při vypnutí čidla (shutdown) omezí dodávka proudu pod 0,5  $\mu\text{A}$ .

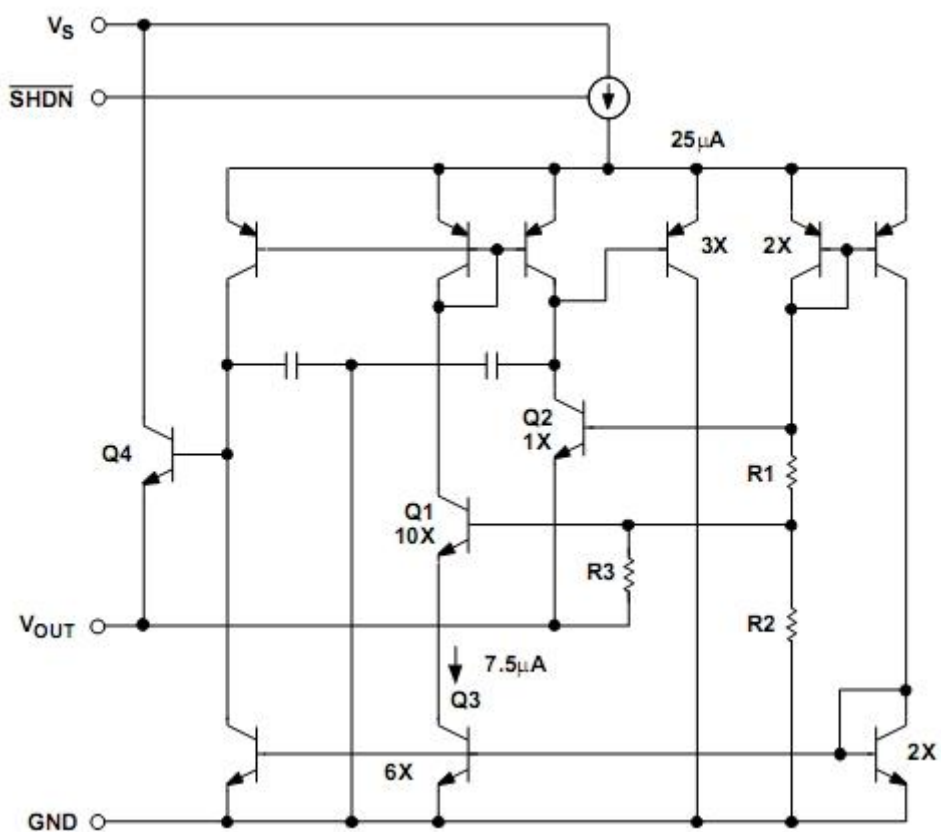
TMP35 je funkčně kompatibilní s čidly LM35/LM45 a poskytuje 250 mV na výstupu při 25 °C. Jeho teplotní rozsah je od 10 °C do 125 °C.

TMP36 zvládne teploty od -40 °C do +125 °C, má 750 mV napětí na výstupu při teplotě +25 °C a dokáže měřit do +125 °C při jediném stejnosměrném napětí 2,7 V. TMP36 je kompatibilní s LM50.

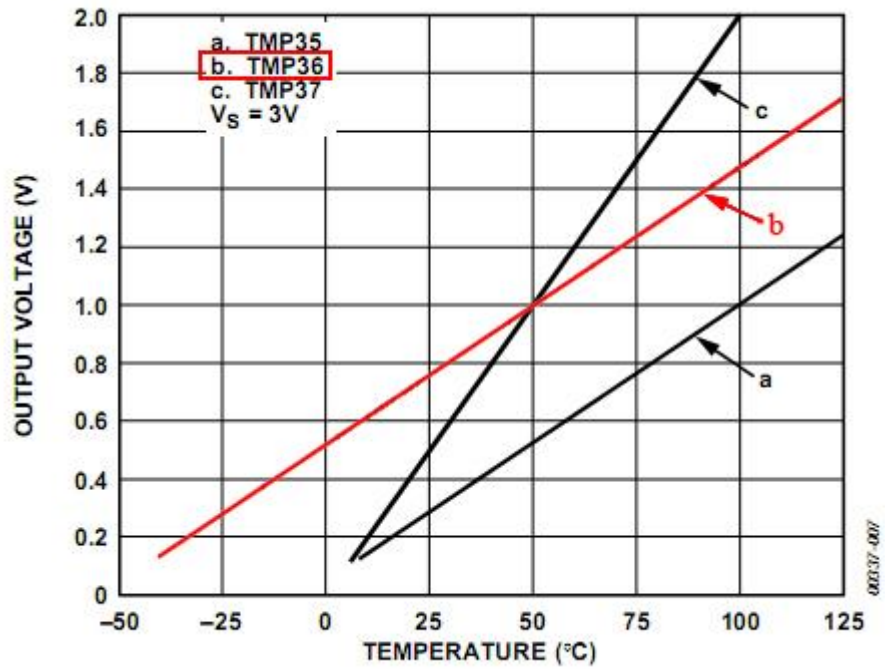
Obě čidla TMP35 a TMP36 mají výstupní škálování stejné a to 10 mV / °C.

TMP37 je určeno pro měření teplot v rozsahu od +5 °C do 100 °C a výstupní škála je 20 mV / °C. Jeho výstupní napětí při +25 °C je 500 mV.

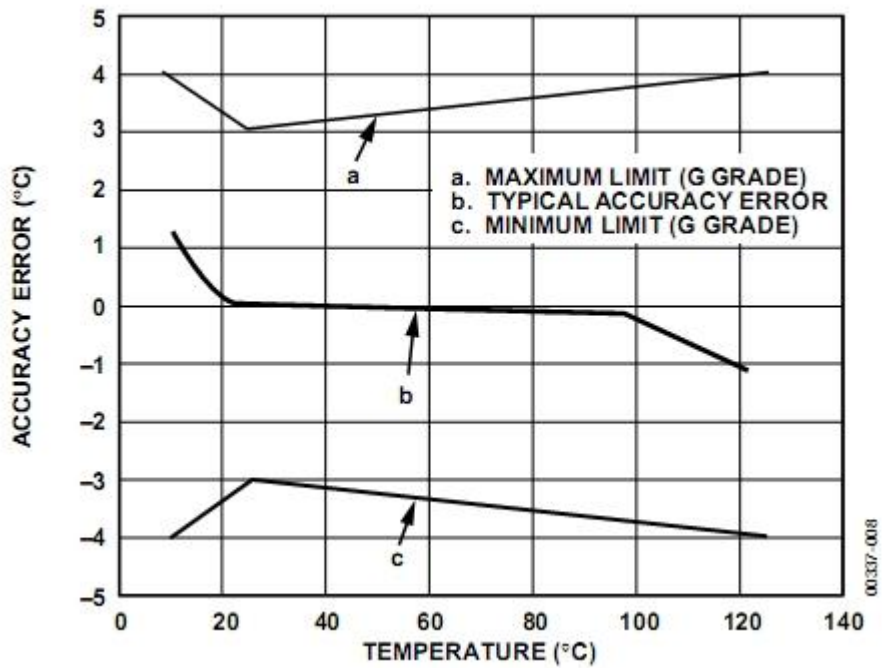
Níže je znázorněno schéma čidla TMP36 (Obr.5), dále pak graf závislosti výstupního napětí na teplotě (Obr.6), kde je červenou barvou zvýrazněna závislost, se kterou budeme pracovat. Poslední obrázek (Obr.7) pak ukazuje při jakých teplotách je nejmenší nebo největší zkreslení.



Obr. 5



Obr. 6



Obr. 7

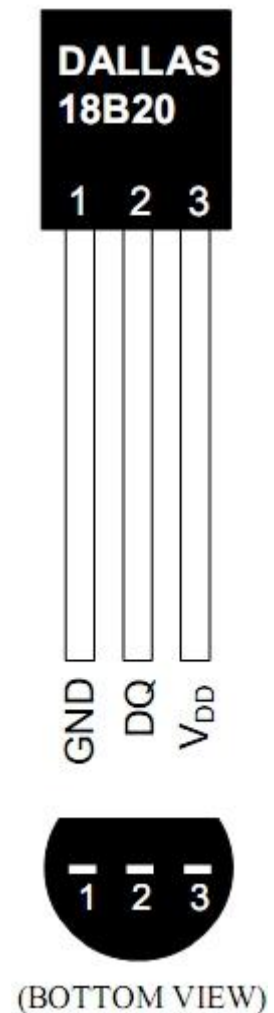
### 3.1.2 Proč toto čidlo ?

Snažili jsme se vybrat čidlo co nejlevnější, nejpřesnější, aby měřilo ve stupních Celsia a aby zvládalo i záporné hodnoty teploty. Podle těchto kritérií jsme zvolili již zmíněné čidlo TMP36GT9Z, které vyhovuje našim požadavkům. Mohli jsme samozřejmě vybrat i jiná čidla, ale o vítězi rozhodla hlavně pořizovací částka, která je přijatelná. Kdyby TMP35 nebo TMP37 dokázalo měřit i při záporných teplotách, bylo by naše rozhodování o dost těžší a museli bychom zkoumat další faktory.

### 3.2 Digitální senzor DS18B20

Jedná se znovu o teplotní čidlo, podobně jako u analogového TMP36. Díky němu tedy budeme moci měřit teplotu a dále s ní jakkoli pracovat. Jeho rozdíl oproti TMP36 je v tom, že se jedná o digitální senzor. Není tedy zapotřebí pracovat s A/D převodníky, toto čidlo totiž při určité teplotě posílá rovnou binární kód, s kterým už PIC velmi dobře pracuje.

Princip snímání teploty [33] je ve zkratce takový, že uvnitř čidla jsou dva oscilátory. Jeden má nízký teplotní koeficient frekvence a druhý naopak vysoký. Jakmile se dá po sběrnici čidla příkaz k zahájení převodu (měření) teploty, spustí se čítání obou těchto oscilátorů. Tato akce trvá (podle teploty) až 700 ms a jejím výsledkem je 12b hodnota odpovídající aktuální teplotě. Ta je poté uložena ve vnitřních registrech obvodu a čeká na její vyčtení z nadřazeného obvodu.



Obr. 8

### 3.2.1 Technické specifikace DS18B20

Obvod DS18B20 [32] je běžně dostupný digitální teplotní senzor, nejčastěji používaný v pouzdře TO-92 se třemi vývody. Je schopen měřit v rozsahu  $-55\text{ }^{\circ}\text{C}$  až  $+125\text{ }^{\circ}\text{C}$  s přesností  $\pm 0,5\text{ }^{\circ}\text{C}$  garantovanou v teplotním rozsahu  $-10\text{ }^{\circ}\text{C}$  až  $+85\text{ }^{\circ}\text{C}$ . Data jsou ze senzoru získávána pomocí jedno-vodičového sériového rozhraní (1-wire interface) s uživatelsky programovatelnou přesností 9 - 12 bitů (tzn. rozlišení  $0.5 - 0.0625\text{ }^{\circ}\text{C}$ ). Pro komunikaci se používá jeden datový a dva napájecí vodiče (GND, VDD, DQ). Napájecí napětí VDD se může pohybovat v rozmezí 3.3 V až 5.5 V. Mimo to lze senzor provozovat v tzv. parazitním režimu, využívajícím pouze dvou vodičů. Kromě měření teploty obvod disponuje funkcí digitálního termostatu, jehož dolní a horní hranici lze programově nastavit a uložit do EEPROM paměti na čipu. Při čtení naměřené hodnoty jsou pak k dispozici bity informující o překročení naprogramovaných rozsahů. Každý obvod má již z výroby přiděleno své vlastní unikátní 64bitové sériové číslo, které slouží k adresaci konkrétního senzoru. Sériové číslo společně s unikátním protokolem identifikace připojených zařízení ke sběrnici umožňuje provozovat několik senzorů na jednom datovém vodiči. Mimo to nabízí obvod možnost využít napěťový signál pro přenos dat, čímž se celkový počet vodičů nutný k připojení senzorů redukuje na dva (GND, VDD/DQ).

### 3.2.2 Rozhraní 1-Wire<sup>TM</sup>

Sběrnice 1-Wire [25], navržená firmou Dallas Semiconductor, umožňuje připojit několik zařízení k řídicí jednotce prostřednictvím pouhých dvou vodičů. Tato sběrnice je použita také u technologie iButton, která se používá v elektronických zabezpečovacích systémech (vrátníky, docházkové systémy).

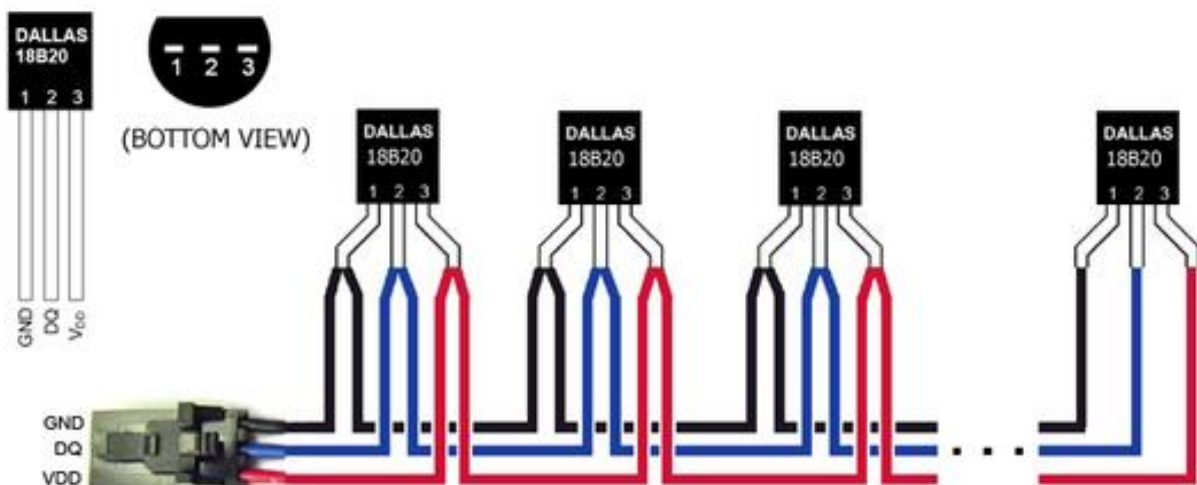
Sběrnice má jeden řídicí obvod (master) a jeden či více ovládaných zařízení (slave). Všechny obvody jsou zapojeny jednak na společné uzemnění, jednak paralelně na společný datový vodič. Tento datový vodič je připojen přes odpor cca 5k na napájecí napětí a „zdvihá“ tak sběrnici do logické 1.

Komunikaci zahajuje vždy master reset pulsem. Nejprve „stáhne“ datový vodič do log. 0 (*uzemní ho*) a drží ho na této úrovni minimálně 480 mikrosekund. Pak sběrnici uvolní a naslouchá. Odpor zatím vrátí sběrnici zpět do logické 1. Pokud je na sběrnici připojené nějaké

1-Wire zařízení, tak detekuje tuto vzestupnou hranu a po prodlevě (15 - 60  $\mu$ s) stáhne sběrnici na 60 - 240  $\mu$ s k logické 0.

Pokud se zařízení správně ohlásí, může master začít vysílat a přijímat data. Data jsou vysílána v tzv. „time slotech“, česky bychom řekli nejspíš v „časových úsecích“ nebo v „okénkách“. Slot je dlouhý 60 až 120  $\mu$ s a během jednoho slotu je vyslán nebo přijat jeden bit informace. Mezi jednotlivými sloty musí být minimálně 1  $\mu$ s mezera, kdy je sběrnice v klidu. Existují 4 druhy slotů: Zápis 1, Zápis 0, Čtení 1 a Čtení 0. Zápisové sloty slouží k tomu, aby master vyslal data do zařízení. Zápis 1 probíhá tak, že master stáhne sběrnici k nule minimálně na 1  $\mu$ s a nejpozději do 15  $\mu$ s od začátku ji opět uvolní a ponechá uvolněnou. Zdvíhací odpor ji tedy vytáhne k logické 1. Zápis 0 je o něco jednodušší: Master stáhne sběrnici k 0 a ponechá ji tak po celý slot, tedy min. 60  $\mu$ s. Zařízení vzorkuje stav na datovém vodiči zhruba 30  $\mu$ s po začátku timeslotu.

Čtecí sloty opět inicializuje master tím, že stáhne sběrnici k nule na minimum 1  $\mu$ s a opět ji uvolní. Po tomto zahájení může zařízení vyslat 1 bit buď tím, že ponechá sběrnici v klidu (logická 1) nebo že ji stáhne (logická 0). Podrobnosti snad osvětlí následující obrázek (Obr.9).

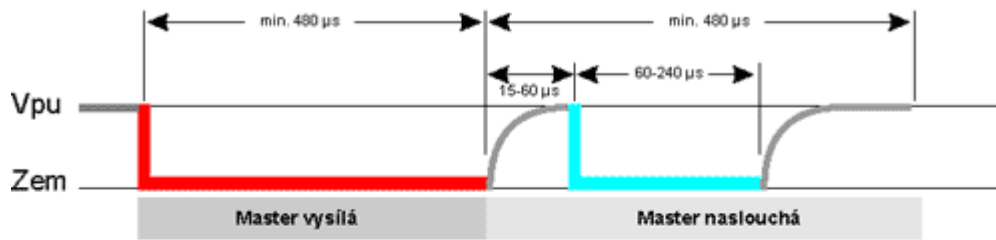


Obr. 9

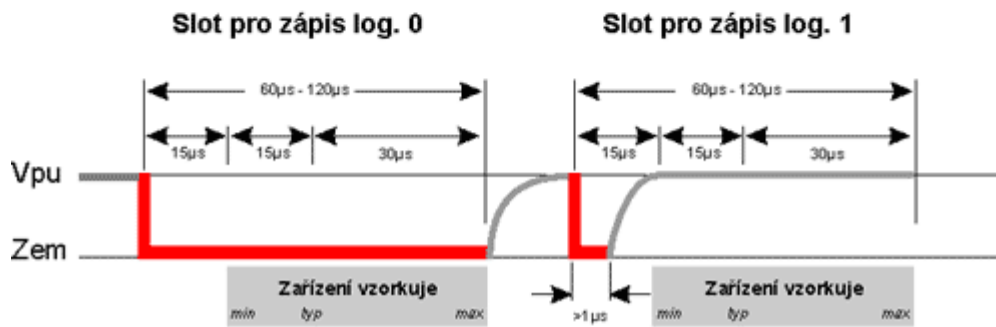
Na následujícím obrázku (Obr.10) jsou teoretické grafy pro reset, vysílání dat a příjem dat. V přílohách jsou námi naměřené grafy pro zápis logické 0 (Příloha 2) a logické 1 (Příloha 3).



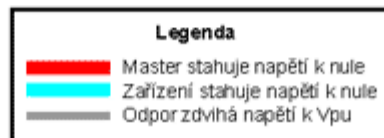
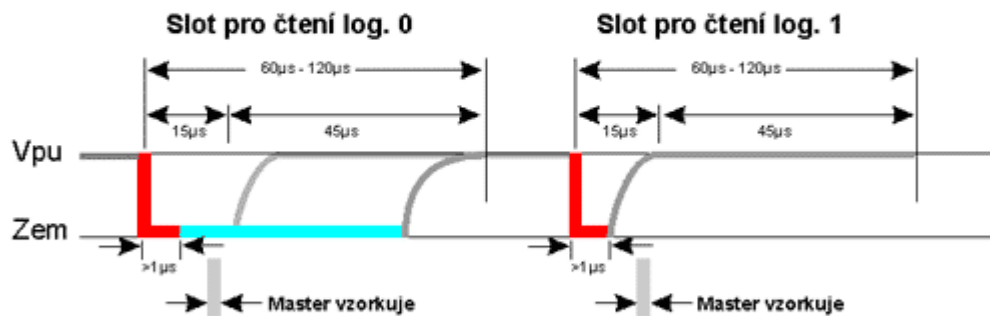
## 1-Wire™ Reset



## 1-Wire™ Vysílání dat



## 1-Wire™ Příjem dat



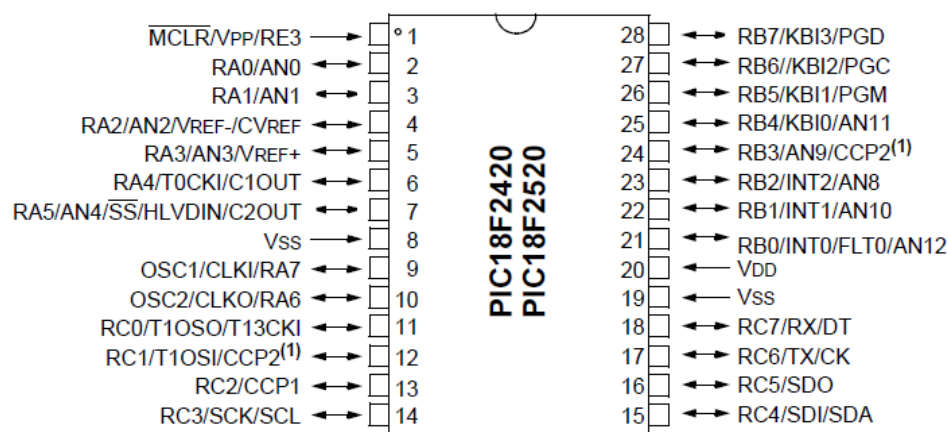
Obr. 10

### 3.3 Mikroprocesor PIC18LF2520

Na mikroprocesor jsme měli tyto požadavky:

- alespoň 1kB paměti RAM
- malá spotřeba
- dostupnost v ČR
- podpora programovátka PICKIT 2
- SPI
- A/D převodník

Nejdříve jsme byli připraveni pracovat s PIC18F252, ale po určité době jsme se rozhodli pořídit PIC18LF2520, který má vnitřní oscilátor a je ho možné napájet 3,3 V. Díky tomu jsme byli schopni zjednodušit zapojení. Popis jednotlivých pinů je znázorněn na Obr. 11.

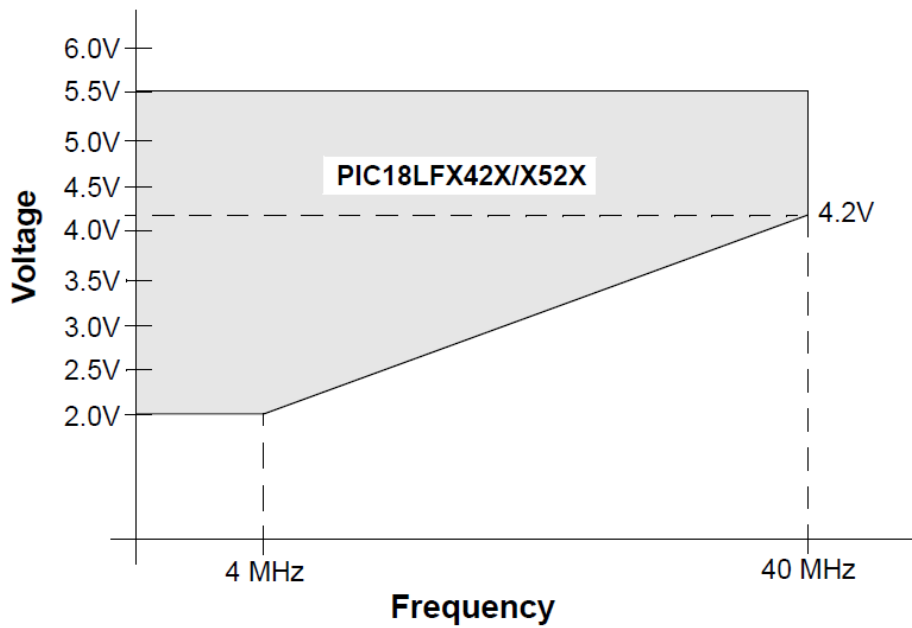


Obr. 11

### 3.3.1 Technické specifikace PIC18LF2520

- SPI (master / slave)
- I<sup>2</sup>C (master / slave)
- EUSART
  - RS-485, RS-232 a LIN 1.2
  - RS-232 – možnost použití vnitřního oscilátoru
  - Automatické probuzení na Start bit
  - Automatická detekce přenosové rychlosti
- 10bitový A/D převodník (10 kanálů)
  - Konverze během spánku
- 2x Capture/Compare/ PWM modul
- Vnitřní oscilátor 31 kHz až 8 MHz
- 4x Phase Lock Loop (PLL)
- 4 krystalové módy, až do 40 MHz
- Druhý oscilátor, který používá Timer1
- ICSP pomocí dvou pinů
- ICD pomocí dvou pinů
- Priority přerušení
- 8 x 8 hardwarová násobička
- 1x 8bitový Timer
- 3x 16bitový Timer
- Rozsah napájecího napětí 2-5,5 V
- Datová paměť SRAM 1536 B
- Datová paměť EEPROM 256 B
- Paměť programu FLASH 32 kB

Více informací lze nalézt v [10].



Obr. 12

$$F_{\text{MAX}} = 16,36 * (U_{\text{MIN}} - 2) + 4 \quad (1)$$

Vzorec (1) slouží pro výpočet maximální frekvence ( $F_{\text{MAX}}$ ), na kterou může být mikroprocesor nastaven při minimálním napětí ( $U_{\text{MIN}}$ ). Po dosazení napětí 3,3 V (napájecí napětí mikroprocesoru) dostaneme výsledek, že maximální frekvence je 25,268 MHz. My jsme zvolili frekvenci 16 MHz, která je dostačující pro rychlou komunikaci s paměťovou kartou a dá se jednoduše realizovat vnitřním oscilátorem (4 MHz a 4xPLL).

### 3.4 Znakový LCD MC1604B-SYL

Displej slouží ke komunikaci mezi uživatelem a zařízením. Každý displej obsahuje řadič, který slouží k ovládní LCD. Nejčastěji je použit řadič HD44780 od firmy HITACHI [22]. Displeje se vyrábějí ve velikostech 8x2 až 40x4 (počet znaků x řádky). Obvykle displeje obsahují 16 vývodů.

Vývod	Název	Popis
1	Vss	GND
2	Vdd	+5 V
3	Vo	Kontrast (od 0,65 V do Vdd)
4	RS	Register select (0 = data, 1 = instrukce)
5	R/W	Read / Write (0 = zápis, 1 = čtení)
6	E	impulz 0-1-0 zapíše data ze sběrnice do LCD
7	DB0	Data (pouze při 8bitové komunikaci)
8	DB1	Data (pouze při 8bitové komunikaci)
9	DB2	Data (pouze při 8bitové komunikaci)
10	DB3	Data (pouze při 8bitové komunikaci)
11	DB4	Data
12	DB5	Data
13	DB6	Data
14	DB7	Data
15	LED(+)	+5V
16	LED(-)	GND

Tabulka 1

Instrukce	Signál									Popis	Čas	
	RS	D7	D6	D5	D4	D3	D2	D1	D0			
Smazat displej	0	0	0	0	0	0	0	0	1	Smaže displej a nastaví adresu DDRAM na 0.	1,64 ms	
Nastavení módu	0	0	0	0	0	0	1	I/D	S	Nastaví směr pohybu kurzoru, a specifikuje posun displeje. Operace jsou prováděny během zápisu a čtení dat.	40us	
Kontrola displeje zap/vyp	0	0	0	0	0	1	D	C	B	Zapne/vypne displej (D), zapne/vypne kurzor (C) a blikání kurzoru (B).	40us	
Posun kurzoru, displeje	0	0	0	0	1	S/C	R/L	*	*	Pohyb kurzoru a posun displeje bez změny obsahu DDRAM.	40us	
Nastavení funkce	0	0	0	1	DL	N	F	*	*	Délka rozhraní (DL), počet řádek displeje (N) a velikost fontu (F).	40us	
Nastavení adresy CGRAM	0	0	1	CGRAM adresa						Nastaví adresu CGRAM. Data jsou přenesena po tomto nastavení.		40us
Nastavení adresy DDRAM	0	1	DDRAM adresa						Nastaví adresu DDRAM. Data jsou přenesena po tomto nastavení.		40us	
Zápis dat	1	Data						Zapíše data do CGRAM nebo DDRAM.			40us	

**Tabulka 2**

Detailní popis jednotlivých instrukcí lze nalézt na [22]

Název bitu	Popis	
I/D	0 - Snížení	1 - Zvýšení
S	0 - Není posun displeje	1 - Posun displeje při zápisu dat
D	0 - Vypnout displej	1 - Zapnout displej
C	0 - Vypnout kurzor	1 - Zapnout kurzor
B	0 - Vypnout blikání kurzoru	1 - Zapnout blikání kurzoru
S/C	0 - Posun kurzoru	1 - Posun displeje
R/L	0 - Posun doleva	1 - Posun doprava
DL	0 - 4bitová komunikace	1 - 8bitová komunikace
N	0 - 1 řádek	1 - 2 řádky
F	0 - 5x8 bodů	1 - 5x10 bodů
BF	0 - Operace je ukončena	1 - Operace probíhá

**Tabulka 3**

Pro zobrazování jsme zvolili displej MC1604B-SYL od firmy Everbouquet [13]. Displej má 4 řádky se 16ti znaky. Kvůli úspoře vývodů mikroprocesoru jsme použili 4bitovou komunikaci. U 4bitové komunikace se nejdříve odešlou 4 horní bity a poté 4 spodní bity.

Po zapnutí displeje je nutné provést inicializaci.

RS	DB7	DB6	DB5	DB4
Čekáme minimálně 15 ms				
0	0	0	1	1
Čekáme minimálně 4,1 ms				
0	0	0	1	1
Čekáme minimálně 100 $\mu$ s				
0	0	0	1	1
0	0	0	1	0
0	0	0	1	0
0	N	F	*	*
Čekáme minimálně 39 $\mu$ s				
0	0	0	0	0
0	1	0	0	0
Čekáme minimálně 39 $\mu$ s				
0	0	0	0	0
0	0	0	0	1
Čekáme minimálně 1,53 ms				
0	0	0	0	0
0	0	1	I/D	S
Čekáme minimálně 39 $\mu$ s				

**Tabulka 4**



Displej obsahuje paměť DD RAM a CG RAM. V paměti DD RAM jsou uloženy hodnoty znaků z jazykové sady displeje, které jsou právě zobrazeny.

Pozice	Adresa v DDRAM
1. řádek	00h...0Fh
2. řádek	40h...4Fh
3. řádek	10h...1Fh
4. řádek	50h...5Fh

**Tabulka 5**

Například pro zobrazení znaku „F“ na 2. řádku a 3. pozici nastavíme adresu DD RAM na hodnotu 43h a odešleme hodnotu znaku „F“ z jazykové sady displeje (46h). Všechny hodnoty znaků jsou v příloze 9.

Do paměti CG RAM je možno nadefinovat 8 uživatelských znaků. Velikost znaků, které zapisujeme do displeje, je 8x8 bodů. Například pro znak „ý“ na pozici 5 nastavíme adresu CG RAM na hodnotu 28h a zapíšeme hodnotu prvního řádku znaku (02h - viz Tabulka 6) a postup opakujeme pro všechny řádky.

7	6	5	4	3	2	1	0	Hex
0	0	0	0	0	0	1	0	02h
0	0	0	0	0	1	0	0	04h
0	0	0	1	0	0	0	1	11h
0	0	0	1	0	0	0	1	11h
0	0	0	1	1	1	1	1	1Fh
0	0	0	0	0	0	0	1	01h
0	0	0	0	1	1	1	0	0Eh
0	0	0	0	0	0	0	0	00h

**Tabulka 6**

Tabulky hodnot pro české znaky můžete nalézt na [23].

## 4 Popis zapojení

Popis zapojení je pro přehlednost vysvětlen ve třech blocích, na které je schéma zapojení rozděleno. Toto schéma se nachází v přílohách (Příloha 5), spolu se schématem tlačítek (Příloha 7).

### Blok 1

Slouží k připojení baterie nebo adaptéru. Vstup pro připojení adaptéru je chráněn proti přepólování diodou 1N4933. Stejnoseměrné napájecí napětí z adaptéru je regulátorem 7810 sníženo na 10 V. Aby nedošlo ke zničení baterie při připojení adaptéru, jsou svorky baterie a výstup regulátoru propojeny diodou 1N4933, která zabrání proudu téct do baterie a zároveň slouží i jako ochrana proti přepólování. Přes vypínač zařízení je napětí dále vedeno na regulátory napětí (LF50CV a LF33CV), piezo měnič a zároveň je napájecím napětím operačního zesilovače TL061. Regulátor napětí LF50CV snižuje napájecí napětí na 5 V a slouží k napájení LCD displeje, podsvícení (LCD displeje a tlačítek) a napájení analogového teplotního čidla TMP36GT9Z. Regulátor napětí LF33CV snižuje napájecí napětí na 3,3 V a slouží k napájení mikrokontroléru Microchip PIC 18LF2520, digitálního teplotního čidla Dallas DS18B20 a SD karty.

### Blok 3

Invertuje napětí z baterie (resp. z regulátoru 7810). Časovač 555 [14] je zapojen jako astabilní multivibrátor a převádí stejnosměrné napětí na obdélníkové o frekvenci 3,2 kHz, které střídavě nabíjí kondenzátor  $C_7$  přes diodu  $D_2$  a vybíjí ho přes diodu  $D_3$ . Vybíjecí proud vytváří na kondenzátoru  $C_8$  záporné napětí, které je použito jako záporné napájecí napětí operačního zesilovače TL061.

### Blok 2

Operační zesilovač TL061 je zapojen jako neinvertující zesilovač, který zesiluje, při vhodném nastavení potenciometru, napětí z čidla tak, aby výstupní napětí při 100 °C odpovídalo hodnotě napětí na regulátoru LF33CV. Krystal Q2 o frekvenci 32,768 kHz je použit jako zdroj reálných hodin. Tranzistor T1 slouží ke spínání podsvícení tlačítek a displeje. Tranzistor Q1 slouží ke spínání piezo měniče. Signál, který jde na tranzistor Q1, je vidět v příloze 4.

## 5 Výroba DPS

### Výroba předlohy

DPS jsme navrhli v programu EAGLE. Se zapnutými vrstvami bottom, pins a dimension a se scale factorem nastaveným na hodnotu 1 jsme předlohu vytiskli na průhlednou fólii v laserové tiskárně s úrovní kontrastu nastavenou na maximum.

### Osvícení

Desku jsme pilkou na železo ořízli podle předlohy. Na světlocitlivou vrstvu jsme potiskem dolů přiložili folii návrhu a zatížili sklem. Pod 100 UV LED ze vzdálenosti 15cm jsme osvětlovali DPS po dobu 10 minut, kterou jsme předtím experimentálně určili.

### Vyvolání

Osvícenou desku jsme vložili do roztoku NaOH (hydroxid sodný) v koncentraci 1,5% a nechali ji ležet do doby, než se přestal lak rozpouštět. Poté jsme desku vyjmuli, důkladně opláchli pod tekoucí vodou a osušili.

### Leptání

Na hladinu leptacího roztoku  $\text{FeCl}_3$  (chlorid železitý) jsme položili DPS mědí dolů. Po odleptání přebytečné mědi jsme desku vyjmuli a důkladně opláchli pod tekoucí vodou.

### Finální úprava

Zbylou emulzi jsme odstranili acetonem. Modelářskou vrtačkou připevněnou ve stojanu jsme vyvrtali otvory pro jednotlivé součástky. Po připájení součástek jsme DPS natřeli kalafunou rozpuštěnou v lihu.

Schéma plošného spoje celého teploměru je v přílohách (Příloha 6) spolu se schématem plošného spoje tlačítek (Příloha 8).

## 6 Postup měření

Napadlo nás, že nejlepším způsobem, jak kvalitně naměřit závislost výstupního napětí na teplotě pro kladné hodnoty teploty, bude měření poklesu teploty vody v rychlovarné konvici od bodu varu. Prvním krokem tedy bylo zapnout rychlovarnou konvici, dalším krokem sehnat nějaký teploměr, který by teplotu zaznamenával. Získali jsme od našeho třídního učitele pana Mgr. Jaroslava Reichla měřicí přístroj LabQuest s čidlem (Příloha 1), který je schopen proměřit téměř jakýkoliv fyzikální jev. Toto čidlo měřicího přístroje jsme spojili gumičkou s naším čidlem (Obr.4) a vše zabalili do igelitového sáčku, protože čidla vodotěsná nejsou. Výstupní napětí našeho čidla jsme zesilovali operačním zesilovačem. Měření jsme tedy prováděli tak, že jsme na měřicím přístroji nastavili režim, aby každou sekundu zaznamenával teplotu a napětí, které naše čidlo produkovalo. Maximální teplotu jsme naměřili chvilku po vypnutí rychlovarné konvice a náš teploměr naměřil 98 stupňů Celsia. Při této teplotě bylo napětí 4,92 V. Při klesání teploty, pak klesalo i napětí ze 4,92 V až cca k 0,9 V, což už byla hodnota pro záporné hodnoty teploty.

Pro nízké teploty (záporné hodnoty teploty) jsme zvolili měření se zmrzlou pet láhví. Pet láhev jsme naplnili vodou a nechali ji zmrznout. Aby se nám lépe měřilo, vyvrtali jsme do této pet lahve s ledem otvor. Do něho jsme zasunuli naše čidlo a již zmíněný měřicí přístroj LabQuest (Příloha 1) a postupně jsme zaznamenávali hodnoty stejně jako při vyšších teplotách s rychlovarnou konvicí. Nejnižší teplotu jsme naměřili kolem  $-25\text{ }^{\circ}\text{C}$ , což je limit použitého domácího mrazicího zařízení.

Z naměřených hodnot jsme lineární regresí (kapitola 7 Matematické zpracování naměřených hodnot) vytvořili rovnici pro výpočet teploty. Dosazením naměřeného napětí vypočítáme výslednou teplotu. Za toto napětí jsme dosadili 4,92 (napětí při  $98\text{ }^{\circ}\text{C}$  - maximální), vydělili jej  $2^{10}$  (počet kvantizačních úrovní) a tento podíl násobíme hodnotou z A/D převodníku. Celou rovnici jsme vynásobili  $10^4$ , abychom nemuseli počítat s desetinnými čárkami.

## 7 Matematické zpracování naměřených hodnot

Při zpracování dat naměřených při určitém fyzikálním experimentu ve většině případů chceme naměřenými daty proložit určitou křivku. Tato křivka přitom má být taková, aby co nejlépe odpovídala příslušné fyzikální teorii a přitom chyba mezi naměřenými daty a touto křivkou byla co možná nejmenší.

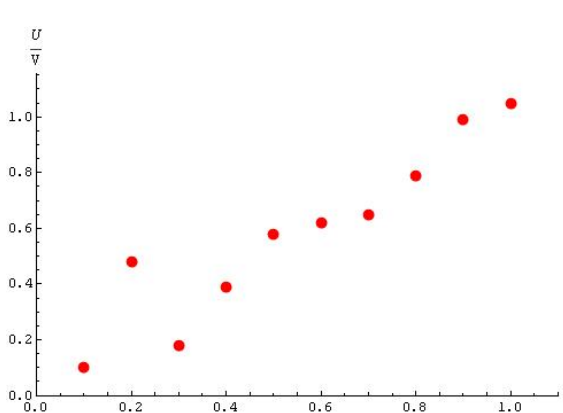
Existuje několik metod, jak tuto křivku nalézt. Jednou z nich je i lineární regrese. A právě touto metodou jsme se zabývali.

### 7.1 Lineární regrese

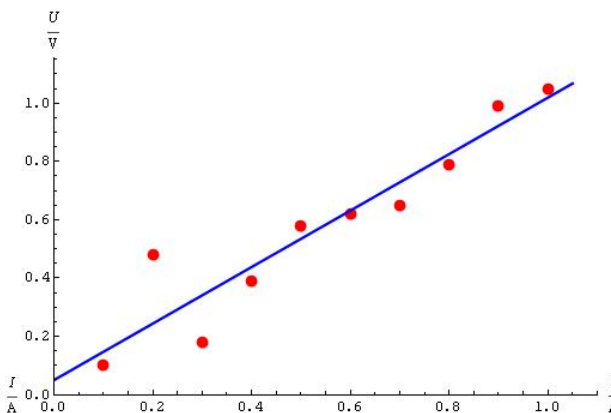
Pojem lineární regrese se používá v matematice a aplikovaných vědách většinou pro dvě mírně odlišné metody:

- Lineární regrese je aproximace daných bodů v grafu polynomem prvního řádu (tj. lineární funkcí). Koeficienty této lineární funkce lze určit metodou nejmenších čtverců (viz dále).
- V obecnějším případě může lineární regrese znamenat aproximaci zadaných bodů v grafu pomocí funkce, kterou lze vyjádřit jako lineární kombinaci jiných funkcí. Tyto dílčí funkce přitom nemusí být už lineární.

V dalším textu se budeme zabývat pouze první metodou lineární regrese. Druhá metoda je principiálně velmi podobná, jen pracuje s nelineárními funkcemi. A proto se komplikuje samotný výpočet; jeho princip ovšem zůstává stejný.



Obr. 13



Obr. 14

Předpokládejme tedy, že máme naměřená určitá data - např. závislost napětí na rezistoru na proudu, který tímto rezistorem prochází. Tato data můžeme zobrazit do grafu – viz Obr.13. Z příslušné teorie (Ohmův zákon pro část elektrického obvodu) vyplývá, že naměřená závislost by měla být lineární (pokud nebudeme uvažovat teplotní změnu odporu rezistoru resp. tento rezistor budeme během měření chladit). Proto se budeme snažit aproximovat naměřená data lineární funkcí, jejímž grafem je přímka nahrazující optimálně naměřená data (viz Obr.14).

S lineárním průběhem naměřené závislosti se dále bude pracovat lépe, než s jednotlivými body. Navíc touto metodou lze potvrdit, případně vyvrátit soulad teorie s reálným experimentem. Zobrazené body jsou zvoleny záměrně tak, aby se od ideální lineární závislosti odchylovaly.

Nyní odvodíme na základě naměřených dat hodnoty koeficientů lineární funkce, kterou naměřená data budeme aproximovat. Použitá metoda se nazývá metoda nejmenších čtverců.

Předpokládejme, že máme naměřená data ve tvaru

$$[x_i; y_i], \quad (2)$$

kde  $i$  je z množiny čísel  $1, 2, \dots, n$ , přičemž  $n$  je počet měření, která jsme provedli.

Máme prostě  $n$  uspořádaných dvojic získaných na základě měření - např. výše uvedený příklad měření proudu a napětí v obvodu s rezistorem.

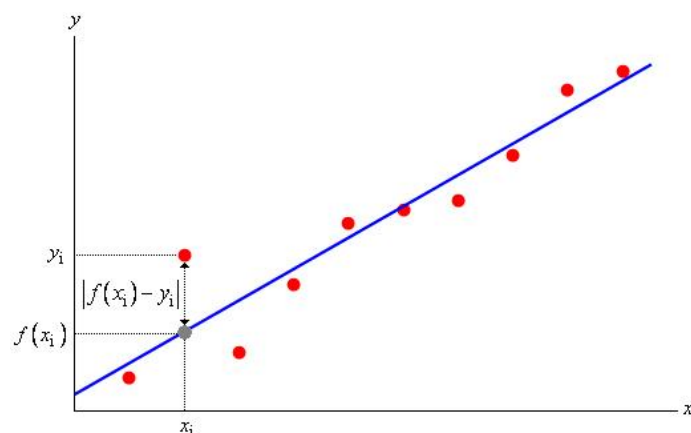
Tato data chceme aproximovat lineární funkcí  $f$  ve tvaru

$$f(x) = ax + b, \quad (3)$$

kde  $a$  a  $b$  jsou reálná čísla a navíc  $a \neq 0$ . Přitom chceme, aby chyba, které se při aproximaci dopustíme, byla co možná nejmenší. Jinými slovy to znamená, aby odchylka naměřené hodnoty  $y_i$  od teoreticky dopočítané funkční hodnoty  $f(x_i)$  lineární závislosti popsané vztahem (3) byla pro všechna přípustná  $i$  minimální. Budeme uvažovat absolutní odchylku, tj. bez ohledu na to, zda je pro danou hodnotu  $x_i$  větší naměřená hodnota  $y_i$  nebo funkční hodnota  $f(x_i)$  hledané aproximace. Budeme chtít tedy minimalizovat výraz  $|f(x_i) - y_i|$ . Pro jeden vybraný bod je tento rozdíl vyznačen v grafu na obrázku (Obr.15). Analogickou úvahu můžeme provést pro všechny naměřené a v grafu zobrazené body, a proto bychom měli uvažovat součet výrazů  $|f(x_i) - y_i|$  pro všechny přípustné hodnoty  $i$ , tj. součet

$$S = |f(x_1) - y_1| + |f(x_2) - y_2| + \dots + |f(x_n) - y_n|, \quad (4)$$

Přitom chceme určit koeficienty  $a$  a  $b$  lineární funkce (3) tak, aby součet (4) byl minimální.



Obr. 15

Pro zvýšení přesnosti nalezené aproximační lineární funkce ve tvaru (3) se místo součtu ve tvaru (4) uvažuje součet druhých mocnin popsanych odchylek, tj. součet ve tvaru

$$S = (f(x_1) - y_1)^2 + (f(x_2) - y_2)^2 + \dots + (f(x_n) - y_n)^2, \quad (5)$$

Jedná se součet druhých mocnin odchylek, tj. „čtverců“ odchylek. Proto se tato metoda nazývá metoda nejmenších čtverců.

Dosazením předpisu lineární funkce (3) do součtu (5) získáme funkci  $S$  závislou na dvou proměnných  $a$  a  $b$  (koeficienty lineární funkce (3)), ve tvaru

$$S(a, b) = (ax_1 + b - y_1)^2 + (ax_2 + b - y_2)^2 + \dots + (ax_n + b - y_n)^2, \quad (6)$$

Funkce (6) má být minimální, čehož docílíme vhodnou volbou koeficientů  $a$  a  $b$ . Naměřená data (2) jsou pevně dána, jediné, co se může ve vztahu (6) měnit, jsou právě koeficienty  $a$  a  $b$ , které určují lineární funkci (3), která nejlépe vystihuje (aproximuje) naměřená data.

Hledáme-li extrémní hodnotu funkce (6) (v našem případě hledáme minimum), je nutné určit první derivace dané funkce. Funkce (6) je funkcí dvou proměnných, proto budeme určovat parciální derivace této funkce podle každé proměnné zvlášť.

Parciální derivace se řídí stejnými pravidly jako „normální“ derivace. Rozdíl je pouze v tom, že proměnné, podle nichž právě nederivujeme, považujeme za konstantní.

Derivací funkce (6) podle proměnné  $a$  získáme

$$\frac{\partial S(a, b)}{\partial a} = 2(ax_1 + b - y_1)x_1 + 2(ax_2 + b - y_2)x_2 + \dots + 2(ax_n + b - y_n)x_n, \quad (7)$$

a derivací téže funkce podle proměnné  $b$  dostaneme

$$\frac{\partial S(a, b)}{\partial b} = 2(ax_1 + b - y_1) + 2(ax_2 + b - y_2) + \dots + 2(ax_n + b - y_n), \quad (8)$$

Hledáme-li extrém funkce (6), musí být obě parciální derivace (7) a (8) nulové. To znamená, že musí platit

$$2(ax_1 + b - y_1)x_1 + 2(ax_2 + b - y_2)x_2 + \dots + 2(ax_n + b - y_n)x_n = 0, \quad (9)$$

a

$$2(ax_1 + b - y_1) + 2(ax_2 + b - y_2) + \dots + 2(ax_n + b - y_n) = 0, \quad (10)$$

Rovnici (9) nyní postupně upravíme. Vydělením číslem dva a roznásobením závorek dostaneme rovnici  $ax_1^2 + bx_1 - y_1x_1 + ax_2^2 + bx_2 - y_2x_2 + \dots + ax_n^2 + bx_n - y_nx_n = 0$ , v níž můžeme z příslušných členů vytknout koeficient  $a$  i  $b$ , čímž dostaneme rovnici ve tvaru

$$a(x_1^2 + x_2^2 + \dots + x_n^2) + b(x_1 + x_2 + \dots + x_n) - y_1x_1 - y_2x_2 - \dots - y_nx_n = 0, \quad (11)$$

Podobným způsobem upravíme i rovnici (10). Vydělíme jí číslem dva a dostaneme rovnici ve tvaru  $ax_1 + b - y_1 + ax_2 + b - y_2 + \dots + ax_n + b - y_n = 0$ . Nyní vytkneme z příslušných členů koeficient  $a$  a získáme rovnici ve tvaru

$$a(x_1 + x_2 + \dots + x_n) + nb - y_1 - y_2 - \dots - y_n = 0 \quad (12)$$

Rovnice (11) a (12) jsou dvě rovnice pro dvě neznámé  $a$  a  $b$ . S využitím maticového počtu lze dokázat, že tato soustava má pro všechna  $n \geq 2$  právě jedno řešení, což znamená, že aproximační lineární funkce (3) je určena jednoznačně. Soustavu rovnic (11) a (12) vyřešíme dosazovací metodou: z rovnice (12) vyjádříme koeficient  $b$  ve tvaru

$$b = \frac{1}{n}(y_1 + y_2 + \dots + y_n - a(x_1 + x_2 + \dots + x_n)), \quad (13)$$

a dosadíme jej do rovnice (10). Dostaneme rovnici ve tvaru

$$a(x_1^2 + x_2^2 + \dots + x_n^2) + \frac{1}{n}(y_1 + y_2 + \dots + y_n - a(x_1 + x_2 + \dots + x_n))(x_1 + x_2 + \dots + x_n) - y_1x_1 - y_2x_2 - \dots - y_nx_n = 0,$$

kterou lze upravit na tvar

$$na(x_1^2 + x_2^2 + \dots + x_n^2) + (y_1 + y_2 + \dots + y_n)(x_1 + x_2 + \dots + x_n) - a(x_1 + x_2 + \dots + x_n)^2 = n(y_1x_1 + y_2x_2 + \dots + y_nx_n)$$

Vytknutím koeficientu  $a$  dostaneme rovnici ve tvaru

$$a(n(x_1^2 + x_2^2 + \dots + x_n^2) - (x_1 + x_2 + \dots + x_n)^2) + (y_1 + y_2 + \dots + y_n)(x_1 + x_2 + \dots + x_n) = n(y_1x_1 + y_2x_2 + \dots + y_nx_n)$$

odkud můžeme vyjádřit koeficient  $a$  ve tvaru

$$a = \frac{n(y_1x_1 + y_2x_2 + \dots + y_nx_n) - (y_1 + y_2 + \dots + y_n)(x_1 + x_2 + \dots + x_n)}{n(x_1^2 + x_2^2 + \dots + x_n^2) - (x_1 + x_2 + \dots + x_n)^2}. \quad (14)$$

Dosazením výrazu (14) do výrazu (13) určíme koeficient  $b$  a vyjádříme ho pouze na základě naměřených dat. Tímto dosazením dostaneme výraz ve tvaru



$$b = \frac{1}{n} \left( y_1 + y_2 + \dots + y_n - \frac{n(y_1x_1 + y_2x_2 + \dots + y_nx_n) - (y_1 + y_2 + \dots + y_n)(x_1 + x_2 + \dots + x_n)}{n(x_1^2 + x_2^2 + \dots + x_n^2) - (x_1 + x_2 + \dots + x_n)^2} (x_1 + x_2 + \dots + x_n) \right),$$

který dále upravíme do tvaru

$$b = \frac{1}{n} \left( y_1 + y_2 + \dots + y_n - \frac{n(y_1x_1 + y_2x_2 + \dots + y_nx_n)(x_1 + x_2 + \dots + x_n) - (y_1 + y_2 + \dots + y_n)(x_1 + x_2 + \dots + x_n)^2}{n(x_1^2 + x_2^2 + \dots + x_n^2) - (x_1 + x_2 + \dots + x_n)^2} \right).$$

Nyní převedeme zlomky uvnitř závorky na společného jmenovatele. Touto úpravou, která je na první pohled nepřehledná, se výraz nakonec zjednoduší

$$b = \frac{1}{n} \left( \frac{n(y_1 + y_2 + \dots + y_n)(x_1^2 + x_2^2 + \dots + x_n^2) - (y_1 + y_2 + \dots + y_n)(x_1 + x_2 + \dots + x_n)^2}{n(x_1^2 + x_2^2 + \dots + x_n^2) - (x_1 + x_2 + \dots + x_n)^2} - \frac{n(y_1x_1 + y_2x_2 + \dots + y_nx_n)(x_1 + x_2 + \dots + x_n) - (y_1 + y_2 + \dots + y_n)(x_1 + x_2 + \dots + x_n)^2}{n(x_1^2 + x_2^2 + \dots + x_n^2) - (x_1 + x_2 + \dots + x_n)^2} \right). \text{ Další úpravou tedy}$$

dostaneme výraz ve tvaru

$$b = \frac{1}{n} \frac{n(y_1 + y_2 + \dots + y_n)(x_1^2 + x_2^2 + \dots + x_n^2) - n(y_1x_1 + y_2x_2 + \dots + y_nx_n)(x_1 + x_2 + \dots + x_n)}{n(x_1^2 + x_2^2 + \dots + x_n^2) - (x_1 + x_2 + \dots + x_n)^2}.$$

Po zkrácení dostaneme výslednou podobu vztahu

$$b = \frac{(y_1 + y_2 + \dots + y_n)(x_1^2 + x_2^2 + \dots + x_n^2) - (y_1x_1 + y_2x_2 + \dots + y_nx_n)(x_1 + x_2 + \dots + x_n)}{n(x_1^2 + x_2^2 + \dots + x_n^2) - (x_1 + x_2 + \dots + x_n)^2}. \quad (15)$$

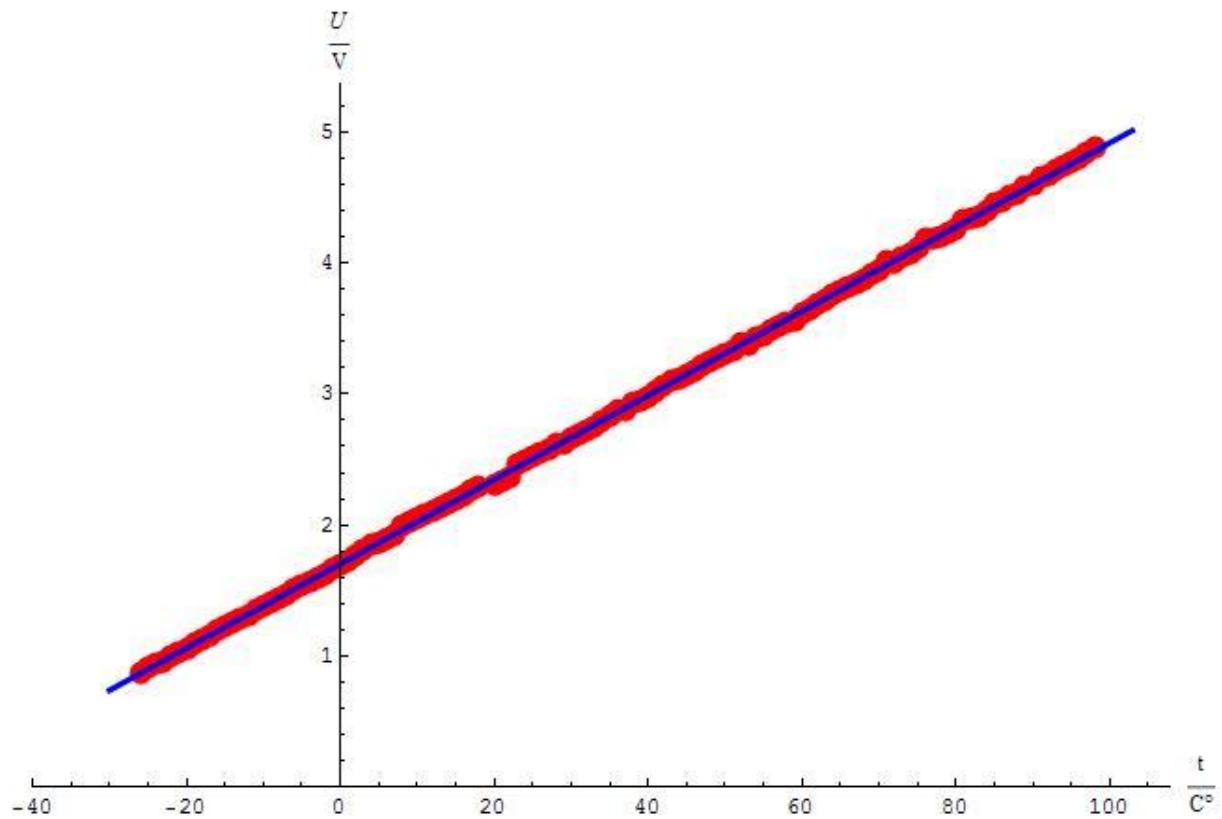
Dosazením vztahů (14) a (15) do předpisu lineární funkce (3) získáme hledaný předpis aproximační funkce.

### 7.1.1 Zpracování dat pomocí programu Mathematica

V kapitole 6 Postup měření bylo popsáno, jak jsme naměřené hodnoty získali. Dalším krokem bylo přepsat naměřené hodnoty do počítače. Po přepsání do tabulky Microsoft Excel jsme už využívali pouze software Mathematica. To je program široce používaný ve vědeckých, technických a matematických kruzích. Program byl původně vytvořen Stephenem Wolframem a následně vyvíjen týmem matematiků a programátorů, který jej i nadále vede. Mathematica je rozdělena do dvou částí – jádra a front endu. Jádro interpretuje výrazy a vrací výsledky. Front end poskytuje GUI, ve kterém výsledky vhodně zobrazuje.

Vše co jsme potřebovali k programování, je popsáno v předchozí kapitole. Kód vytažený z programu, je v příloze 12. Program byl psán tak, aby se dobře „četl“ a bylo z něj pochopitelné, co které instrukce provádějí. Obsahuje i vysvětlivky jednotlivých instrukcí,

kté jsou psány ve tvaru (\*vysvětlení instrukce\*). Instrukce, které nejsou popsány, jsme považovali za nepodstatné. Výsledná lineární regrese s výpisem aproximační funkce (16) je znázorněna na obrázku (Obr.16).



Obr. 16

$$U = 0,0321739 t + 1,7022 \quad (16)$$

## 8 Komunikace s paměťovou kartou

### 8.1 SD/MMC karta

Jako paměťové médium jsme se rozhodli použít SD/MMC kartu, která bude naformátována na souborový systém FAT12 nebo FAT16. Data budou ukládána do .txt souboru a to nám umožní náhled dat na PC.

#### Parametry

	SD	MMC
Délka	32 mm	32 mm
Šířka	24 mm	24 mm
Tloušťka	2,1 mm	1,4 mm
Max. frekvence na CLK	25 MHz	20 MHz
Napájecí napětí	2,7 V - 3,6 V	2,7 V - 3,6 V

Tabulka 7

Oba typy karet podporují komunikaci po SPI sběrnici.

Komunikace mikroprocesoru s SD/MMC kartou probíhá pomocí sběrnice SPI. Tato sběrnice je tvořena třemi vodiči: serial data in (SDI), serial data out (SDO) a serial clock (CLK). Dále je využit čtvrtý vodič chip select (CS).

#### 8.1.1 Inicializace MMC/SD karty

- Deaktivovat kartu (CS=1)
- Provést 80 hodinových cyklů
- Aktivovat kartu (CS=0)
- Odesílat CMD0 dokud odpověď typu R1 není 0x01 (nutno použít platné CRC (0x95))
- Odesílat CMD1 dokud 1. Byte odpovědi typu R3 (resp. R1) není 0x00

Před skončením inicializace se musí frekvence na vodiči CLK pohybovat v rozmezí 100 kHz - 400 kHz [21]. Před ukončením komunikace na sběrnici je nutné provést 8 hodinových cyklů, aby karta dokončila prováděnou operaci [9].

### 8.1.2 Formát odesílání příkazu

Každý odeslaný příkaz je dlouhý 48 bitů.

Bit	0	1	2–7 (6 bitů)	8–39 (32 bitů)	40–46 (7 bitů)	47
Význam	Start bit	Host	Příkaz	Argument	CRC	End bit

**Tabulka 8**

Platná hodnota CRC není nutná při komunikaci po sběrnici SPI s výjimkou příkazu CMD0 při inicializaci. Pro příkaz CMD0 je CRC uvedeno v dokumentaci (0x95) [8].

Příkazy, které jsme použili při komunikaci:

Příkaz	Parametr	Odpověď	Popis
CMD0	Žádný	R1	Resetuje kartu
CMD1	Žádný	R3	Zahájení inicializace
CMD16	Délka bloku	R1	Nastaví délku komunikace
CMD17	Adresa	R1	Načte data z adresy
CMD24	Adresa	R1	Uloží data na adresu

**Tabulka 9**

Odpověď typu R1 je dlouhá 1B. Hodnota 7. bitu je nula, ostatní hodnoty odpovídají chybám, které jsou vyznačeny v tabulce

Bit	Chyba
7	-
6	Chyba parametru
5	Chyba adresy
4	Erase sequence chyba
3	Chyba CRC
2	Chybný příkaz
1	Reset mazání
0	Idle stav

**Tabulka 10**

Odpověď typu R3 je dlouhá 5 B. První Byte obsahuje odpověď typu R1 a další 4 B obsahují OCR registr.

### 8.1.3 Odeslání dat

Poté, co karta přijme příkaz CMD24 a odpoví odpovědí typu R1, může master odesílat data ve formátu, který je popsán v kapitole 8.1.5 Formát dat. Po skončení odesílání dat karta odpoví 1 B dlouhým data tokenem.

7.	6.	5.	4.	3.	2.	1.	0.	
X	X	X	0	0	1	0	1	Data byla přijata
X	X	X	0	1	0	1	1	Chyba CRC
X	X	X	0	1	1	0	1	Chyba při zápisu

Tabulka 11

### 8.1.4 Příjem dat

Po přijetí příkazu CMD17 odpoví karta odpovědí typu R1 a začne vysílat data ve formátu, který je popsán v kapitole 8.1.5 Formát dat. Pokud selže čtení dat, karta odpoví error tokenem. Error token je dlouhý 1 B, druhy chyb ukazuje tabulka

Bit	Chyba
7	-
6	-
5	-
4	Karta je zamčena
3	Chybná adresa
2	Chyba opravy dat ECC
1	Chyba vnitřního kontroléru
0	Neznámá chyba

Tabulka 12

### 8.1.5 Formát dat

Každé odeslání nebo přijetí dat se skládá z:

- data tokenu (1 B)
- bloku dat (1-2048 B)
- CRC (2 B)

Data token pro příkaz CMD17 a CMD24 je 0xFE [19].

### 8.2 Souborový systém FAT

Souborový systém FAT se skládá ze čtyř hlavních oblastí [18]

- Rezervované sektory – Jsou umístěny na začátku paměti. Sektor 0 je boot sektor, na kterém jsou informace o souborovém systému FAT (kapitola 8.2.1 Boot sektor)
- Alokační tabulka souboru (FAT) – obvykle jsou dvě a jsou hned za sebou (primární a záložní). Ukazují, jaké clustery jsou využity pro soubory a adresáře
- Root adresář – informace o uložených souborech a adresářích v root adresáři.
- Datová oblast – Jsou tu uložena data jednotlivých souborů. Velikost souborů může narůstat jednoduchým přiřazením clusteru v tabulce FAT.

Souborový systém FAT používá formát *little endian* pro ukládání záznamu v alokační tabulce souboru a boot sektoru. *Little endian* je formát, ve kterém se nejdříve ukládají méně významné Byty. Např. číslo 0x1234 je uloženo v paměti jako 0x3412.

## 8.2.1 Boot sektor

Důležité informace v boot sektoru

Byte offset	Délka v Bytech	Popis
0x0b	2	Počet Bytů na sektor (obvykle 512)
0x0d	1	Počet sektorů na cluster
0x0e	2	Počet rezervovaných sektorů
0x10	1	Počet alokačních tabulek souboru (FAT)
0x11	2	Maximální počet záznamů root adresáře
0x13	2	Celkový počet sektorů (pokud 0, použit 4 Byty na offsetu 0x20)
0x16	2	Počet sektorů na alokační tabulku souboru (FAT)
0x20	4	Celkový počet sektorů

**Tabulka 13**

Z těchto dat (Tabulka 13) můžeme vypočítat adresy všech oblastí

- Adresa FAT = počet rezervovaných sektorů (0x0e) \* počet Bytů na sektor (0x0b)
- Adresa root adresáře = adresa FAT + počet FAT (0x10) \* počet sektorů na FAT (0x16) \* počet Bytů na sektor (0x0b)
- Adresa datové oblasti = adresa root adresáře + (maximální počet záznamů v root adresáři (0x11) \* 32)/počet Bytů na sektor (0x0b)

## 8.2.2 FAT

Datová oblast paměti je rozdělena do clusterů. Obsazenost clusterů jednotlivými soubory a adresáři je popsána ve FAT. Každý soubor zabírá minimálně jeden cluster. První dva záznamy (2 B, vyznačené na Obr.17 červeně) obsahují speciální hodnotu, proto první cluster v datové oblasti je cluster 2 [18].

Velikost každého záznamu záleží na použitém souborovém systému. FAT16 používá jako jeden záznam 16 bitů a FAT12 používá 12 bitů. Význam jednotlivých hodnot ukazuje následující tabulka

FAT12	FAT16	Popis
0x000	0x0000	Volný cluster
0x001	0x0001	Rezervovaná hodnota
0x002- 0xFEFF	0x0002-0xFFEF	Použitý cluster (hodnota ukazuje na další cluster souboru, na Obr.17 vyznačeno modře)
0xFF0- 0xFF6	0xFFF0-0xFFF6	Rezervovaná hodnota
0xFF7	0xFFF7	Rezervovaná hodnota nebo špatný sector v clusteru
0xFF8- 0xFFFF	0xFFF8-0xFFFF	Poslední cluster souboru (na Obr.17 vyznačeno zeleně)

Tabulka 14

```

Offset (h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000C00  F8 FF FF FF 03 00 04 00 FF FF 00 00 00 00 00 00
00000C10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000C20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  
```

Obr. 17



## 8.2.3 Root adresář

Každý soubor nebo adresář je uložen jako 32 B dlouhý záznam v root adresáři. Každý záznam obsahuje název, příponu, atributy (pouze pro čtení, systém, ...), datum vytvoření, číslo prvního clusteru a velikost souboru nebo adresáře

Následující tabulka ukazuje formát záznamu

Offset	Délka v Bytech	Popis
0x00	8	Název souboru (na Obr.18 vyznačeno červeně)
0x08	3	Přípona (na Obr.18 vyznačeno modře)
0x0B	1	Atributy
0x0e	2	Čas vytvoření
0x10	2	Datum vytvoření
0x12	2	Datum posledního přístupu
0x16	2	Čas poslední úpravy
0x18	2	Datum poslední úpravy
0x1A	2	Číslo prvního clusteru (na Obr.18 vyznačeno zeleně)
0x1C	4	Velikost souboru v bytech (na Obr.18 vyznačeno černě)

Tabulka 15

```

Offset (h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0003E020 31 31 30 32 30 35 20 20 54 58 54 20 10 97 5C BB 110205 TXT .-\»
0003E030 63 3E 63 3E 00 00 98 BB 63 3E 02 00 BC 4F 00 00 c>c>...»c>..LO..
0003E040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0003E050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Obr. 18

## 9 Popis programu

Program je naprogramován v jazyce C, jako překladač jsme použili HI-TECH C pro mikrokontroléry PIC18. Kvůli velikosti programu jsme museli použít verzi PRO, dostupnou zdarma po dobu 45 dnů. Verze PRO obsahuje optimalizaci kódu (OCG), která sníží velikost programu přibližně o 50%.

Po zapnutí přístroje proběhne inicializace mikrokontroléru. Nastaví se vstupní a výstupní piny, SPI sběrnice, vnitřní oscilátor, zdroje přerušení, priority přerušení a A/D převodník.

Dále se inicializuje displej a nadefinují české znaky. Uživatel je vyzván k zadání data a času. Po úspěšném nastavení data a času program pracuje ve smyčce. Smyčka obsahuje tyto akce:

- převod teploty (sudá sekunda)
- vyčtení teploty (lichá sekunda)
- kontrola teplotního alarmu (lichá sekunda)
- změna textu na displeji
- kontrola časového alarmu (každá celá minuta)
- uložení teploty (každých 15 minut)
- sleep mód

Ze sleep módu se přístroj probudí stisknutím tlačítka nebo změnou času (přetečení TMR1). Program reaguje na stisk tlačítek podle jejich funkce - viz kapitola 10 Ovládání teploměru.

### Vytvoření souboru

Nejdříve ve FAT nalezneme volný cluster a zapíšeme kód pro poslední cluster souboru. Po obsazení volného clusteru najdeme 32 B volného místa v root adresáři a zapíšeme 32 B dlouhý záznam, který obsahuje název souboru ve tvaru RRRMMDD.txt, datum a čas vytvoření, datum a čas posledního přístupu, atributy a číslo prvního clusteru. Vše uložíme ve tvaru dle normy (tabulka 15).

ČAS	UVNITŘ [°C]	VENKU [°C]
00:00	19,4	10,1
00:15	19,2	10,0
00:30	19,4	10,1
00:45	19,2	10,0
01:00	19,4	9,9
01:15	19,2	9,9
01:30	19,4	10,1
01:45	19,4	10,2
02:00	19,4	10,3
02:15	19,4	10,3
02:30	19,2	10,3
02:45	19,5	10,4
03:00	19,2	10,3
03:15	19,1	10,4
03:30	19,2	10,4

Obr. 19

### Otevření souboru

V root adresáři vyhledáme název souboru, který chceme otevřít, a načteme všechny důležité informace o souboru (velikost souboru, první cluster souboru, adresa v root adresáři). Ve FAT načteme všechna čísla clusterů, která patří k souboru.

### Zavření souboru

Zapišeme aktuální velikost souboru, datum a čas poslední úpravy.

### Úprava souboru

Vypočítáme adresu posledního clusteru souboru a podle počtu bitů zjistíme konec souboru. Za konec souboru zapišeme text, který chceme uložit. Pokud je potřeba obsadit další cluster, tak ve FAT nalezneme volný cluster, do FAT zapišeme kód pro poslední cluster souboru a na předchozím záznamu clusteru zapišeme číslo nového clusteru. Vypočítáme adresu nového clusteru a dokončíme zápis dat.

### Nalezení dat v souboru

Vypočítáme adresu prvního clusteru a začneme hledat data odpovídající zadání. Po nalezení dat proběhne jejich zpracování (minimální, maximální, průměrná teplota atd). Postup opakujeme pro všechny clusteru souboru.

## Výpočet dne v týdnu z data

Pro výpočet dne v týdnu z data jsme použili Zellerův algoritmus [17], díky kterému jsme schopni přesně spočítat den ze zadaného data. Slouží k tomu dvě rovnice (17) a (18), jejichž význam je popsán níže.

$$h = \left( q + \left[ \frac{(m+1)26}{10} \right] + K + \left[ \frac{K}{4} \right] + \left[ \frac{J}{4} \right] + 5J \right) \pmod{7} \quad (17)$$

$$d = ((h+5) \pmod{7}) + 1 \quad (18)$$

- $q$  číslo dne v měsíci
- $m$  je číslo měsíce (3 = březen, 4 = duben, ..., 14 = únor)
- $K$  je rok mod 100 (2010 mod 100 = 10)
- $J$  je rok/100 (2010/100 = 20)
- $d$  je číslo dne v týdnu (1 = pondělí, 2 = úterý, ..., 7 = neděle)

všechna dělení jsou celočíselná

Např. pro výpočet použijeme datum 20.2.2011

$$q = 20$$

$$m = 14$$

$$K = 2010 \pmod{100} = 10 \text{ (počítá se jako rok 2010, protože únor je 14. měsíc roku 2010)}$$

$$J = 2010/100 = 20$$

$$h = (20 + (14 + 1) * 26/10 + 10 + 10/4 + 20/4 + 5 * 20) \pmod{7} = 1$$

$$d = ((1+5) \pmod{7}) + 1 = 7 \Rightarrow \text{neděle}$$

## 10 Ovládání teploměru

### Uvedení přístroje do provozu

Odšroubujeme šroubky (7) na levé straně teploměru a sundáme boční kryt. Zapojíme 9V baterii do konektoru a přišroubujeme boční kryt zpět. Do slotu pro paměťovou kartu (10) vložíme paměťovou kartu naformátovanou na souborový systém FAT12 nebo FAT16. Připojíme venkovní čidlo do konektoru (8). Připojíme síťový 12V zdroj do konektoru (9).

### Zapnutí přístroje

Teploměr zapneme přepnutím vypínače (1) do polohy „ON“. Po zapnutí se aktivuje režim zadání času a data. Uživatel nastaví hodnoty (viz Zadávání hodnot). Po vložení hodnot se přístroj přepne na obrazovku „Aktuální teplota“, na které je zobrazena aktuální teplota vnitřního a vnějšího čidla, datum a čas.

### Změna obrazovek

Stiskem tlačítka „←“ (2) nebo „→“ (3), pokud není aktivovaný nastavovací režim, dojde ke změně hlavních obrazovek. Přepínání detailních obrazovek se provádí stiskem tlačítka „↑“ (4) nebo „↓“ (5).

### Zadávání hodnot

Při zadávání hodnot se rozblíká kurzor na příslušné pozici. Změnu hodnoty provedeme pomocí tlačítka „↑“ (4) nebo „↓“ (5). Změna pozice kurzoru se provádí stiskem tlačítka „←“ (2) nebo „→“ (3). Stiskem tlačítka „OK“ (6) se zadávání čísla ukončí.

### Nastavení času

Na obrazovce „Aktuální teplota“ podržením tlačítka „OK“ (6) po dobu 4 sekund se přístroj přepne do režimu pro nastavení času. Uživatel nastaví hodnoty (viz zadávání hodnot). Po vložení hodnot se přístroj přepne na obrazovku „Aktuální teplota“.

## **Nastavení časových alarmů**

Na obrazovce „Alarm“ je možné nastavit 9 alarmů. Stisknutím tlačítka „OK“ (6) aktivujeme nebo deaktivujeme alarm. Podržením tlačítka „OK“ (6) po dobu 4 sekund se přístroj přepne do režimu pro nastavení alarmu. Nejdříve nastavíme, zda jde o opakovaný alarm nebo ne. Pro opakovaný alarm zadáme čas (viz zadávání hodnot) a poté nastavíme jednotlivé dny opakování. Mezi jednotlivými dny se posuneme pomocí tlačítek „←“ (2) a „→“ (3). Stiskem tlačítka „↑“ (4) nebo „↓“ (5) zapneme/vypneme opakování pro příslušný den. Tlačítkem „OK“ (6) nastavení ukončíme. Pro jednorázový alarm nastavíme datum a čas (viz Zadávání hodnot).

## **Nastavení teplotních alarmů**

Na obrazovce „Teplotní alarm“ se dají nastavit 4 alarmy. Dva pro teplotu nižší než je zadaná a dva pro teplotu vyšší (pro každé čidlo jeden alarm). Mezi jednotlivými typy alarmů se pohybujeme pomocí tlačítek „↑“ (4) a „↓“ (5). Stisknutím tlačítka „OK“ (6) aktivujeme nebo deaktivujeme alarm. Podržením tlačítka „OK“ (6) po dobu 4 sekund se přístroj přepne do režimu pro nastavení teploty. Uživatel nastaví teplotu (viz Zadávání hodnot).

## **Prohlížení uložených teplot**

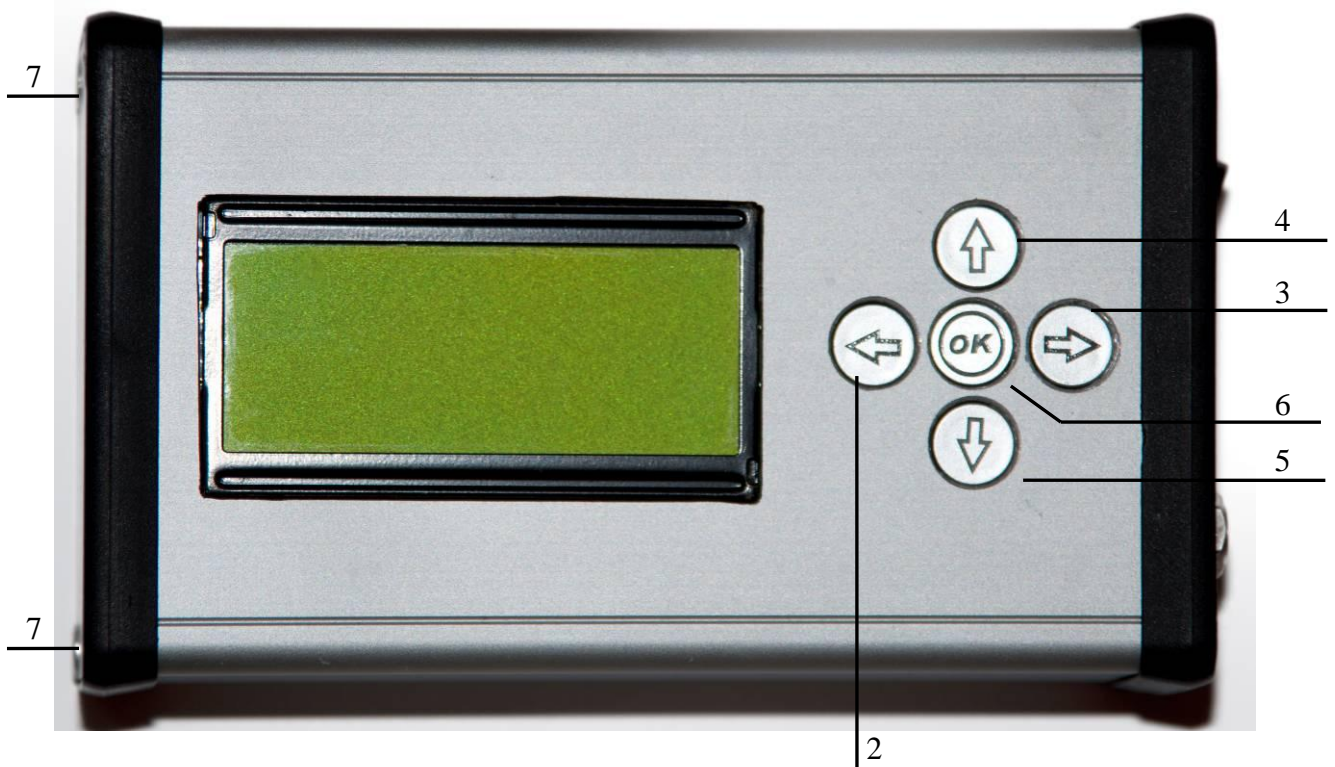
Na obrazovce „Uložené teploty“ podržením tlačítka „OK“ (6) po dobu 4 sekund se přístroj přepne do režimu pro nastavení data a času, pro který chceme znát teplotu. Po zadání hodnot (viz Zadávání hodnot, zadané minuty musí být násobky 15) dojde automaticky k vyhledání teplot v paměti. Tlačítkem „↑“ (4) nebo „↓“ (5) měníme čas v rámci jednoho souboru s okamžitým účinkem (není nutné potvrzovat tlačítkem „OK“ (6)).

## **Zpracování hodnot**

Na obrazovce „Zpracování“ podržením tlačítka „OK“ (6) po dobu 4 sekund se přístroj přepne do režimu pro nastavení rozsahu data a času. Na druhém a třetím řádku zadáváme počáteční a konečné datum a čas období, které chceme zpracovat. Na čtvrtém řádku zadáváme časový interval zpracování pro jednotlivé dny (např.: období od 1.1.2011 6:00 do 31.1.2011 18:00 pro teplotu naměřenou v časovém intervalu od 11:00 do 13:00).



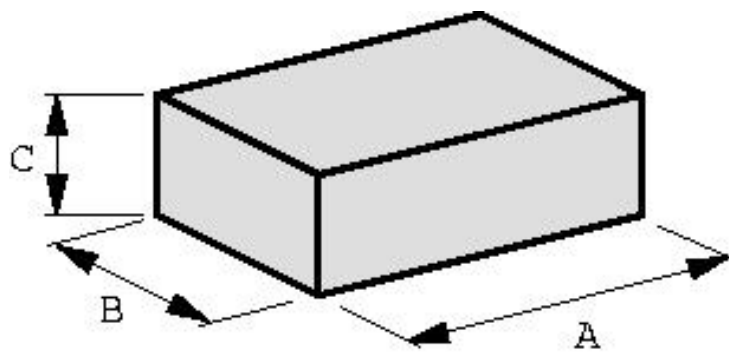
Obr. 20



Obr. 21

## 11 Výběr krabičky

Nejprve jsme museli nějakou krabičku koupit, nejlépe takovou, do které by se vešel náš plošný obvod. Sehnali jsme designově pěkně udělanou, která byla vyrobena z hliníku. Mohli jsme si vybrat z více rozměrů, což se nám hodilo (Obr.23). Podle velikosti plošného spoje, s určitou rezervou pro případné úpravy a vodiče, jsme vybrali krabičku s rozměry  $A=120$  mm,  $B=78$  mm a  $C=43$  mm (viz Obr.22).



Obr. 22



Obr. 23



## 11.1 Úpravy krabičky

Nejprve jsme museli na krabičku načrtnout, kde budou tlačítka a rozložit je tak, aby byla stejně daleko od sebe. Krabičku jsme polepili papírovou lepenkou, aby se na ni dalo dobře rýsovat. Narýsovali jsme středy tlačítek. Vzdálenost mezi tlačítky jsme zvolili 2 mm.

Na vyvrtání jsme použili vrták zhruba o 3 mm menší, než je velikost tlačítek. Vyvrtali jsme první díрку, poté jsme použili větší vrták, kterým jsme už jen zvětšovali původní díрку. Nakonec jsme vzali pilník a dopilovali ji, aby vypadala dobře a nebyla ostrá. Tímto způsobem jsme vrtali všechny dírky na krabičce s tím, že pro otvory na konektor pro napájení a pro vývod čidla jsme použili vrtáky menší. Pro čidlo 5,5 mm a pro konektor na napájení 7,5 mm.

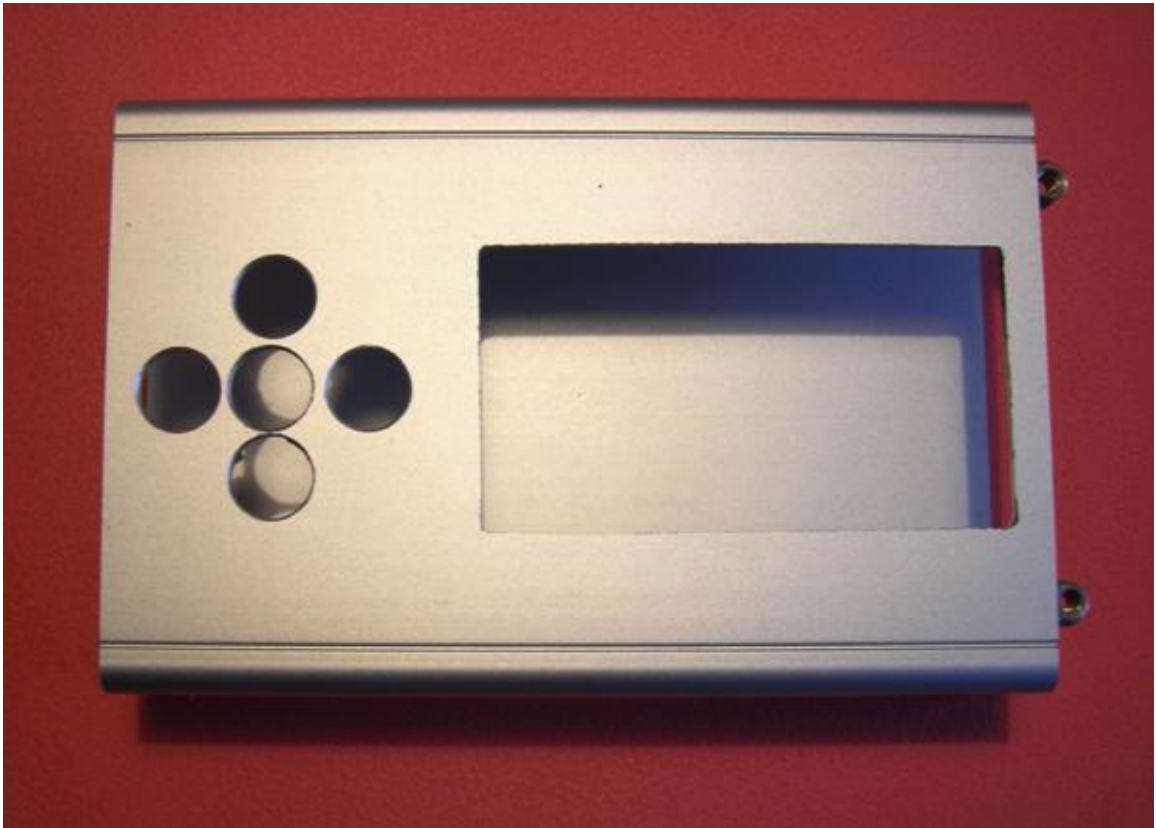
Dále jsme nakreslili obdélníkový obrys displeje a vyvrtali díрку, kterou jsme použili na prostrčení listu lupínkové pilky. Po vyříznutí obdélníku jsme pilníkem dopilovali hrany, aby nebyly ostré. Stejný postup jsme použili i u otvoru na SD/MMC kartu. Protože jsme neměli vrták o velikosti 20, tak jsme u vypínače postupovali stejně jako u displeje.

Krabičku bylo potřeba zvětšit o několik mm, proto jsme mezi bočnice a horní část vložili distanční sloupky o velikosti 5mm.

Umístění tlačítek a displeje jsme zvolili tak, že display je co nejvíc vlevo a tlačítka co nejbližší k displeji.

Finální podobu vidíte na následujícím obrázku (Obr.24). Další obrázky znázorňují boční stěnu krabičky (Obr.26) a použité nářadí plus vrtáky (Obr.25).

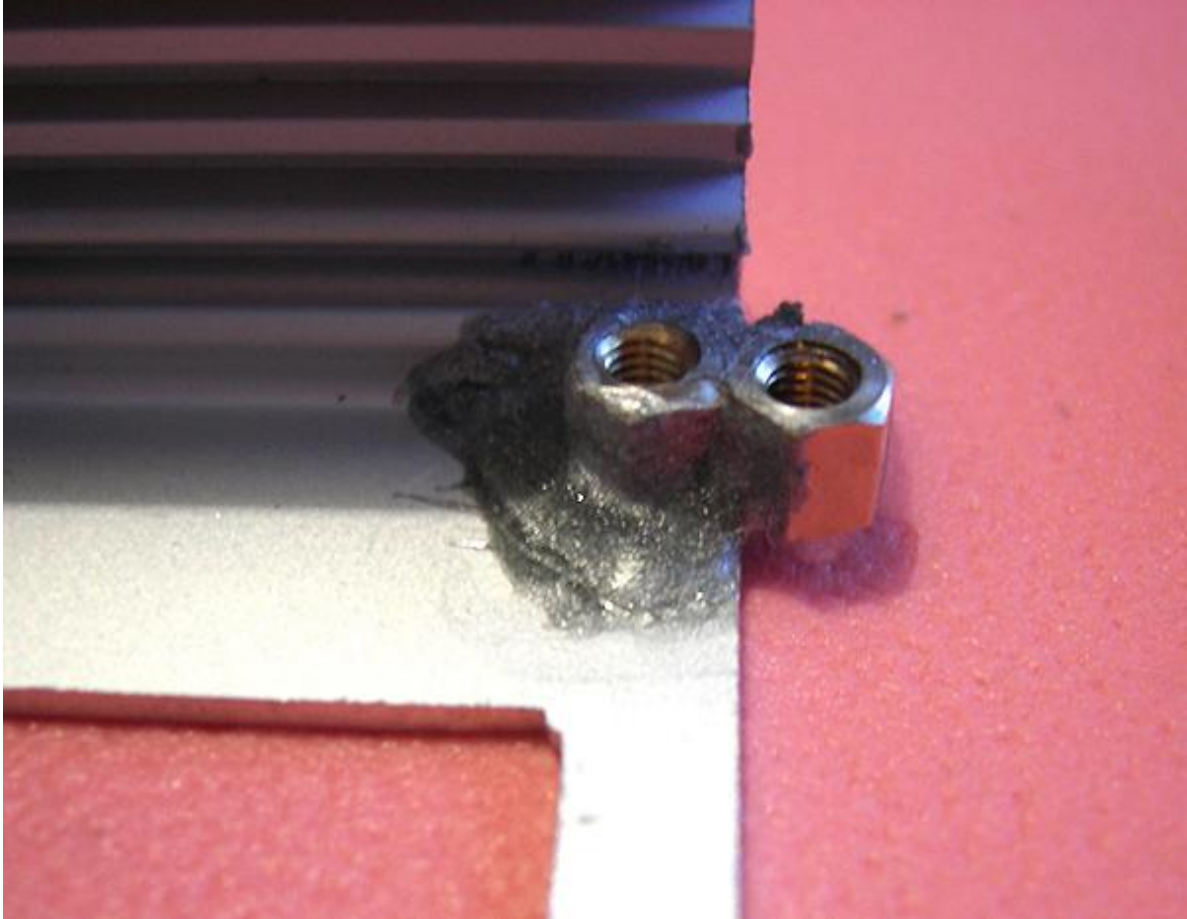
Krabička se skládá z více dílů. Z horní části, bočních stěn a podstavy. Po odšroubování šroubků z boční strany, můžeme vysunout spodní podstavu krabičky. A pak lze s krabičkou i obsahem lépe manipulovat.



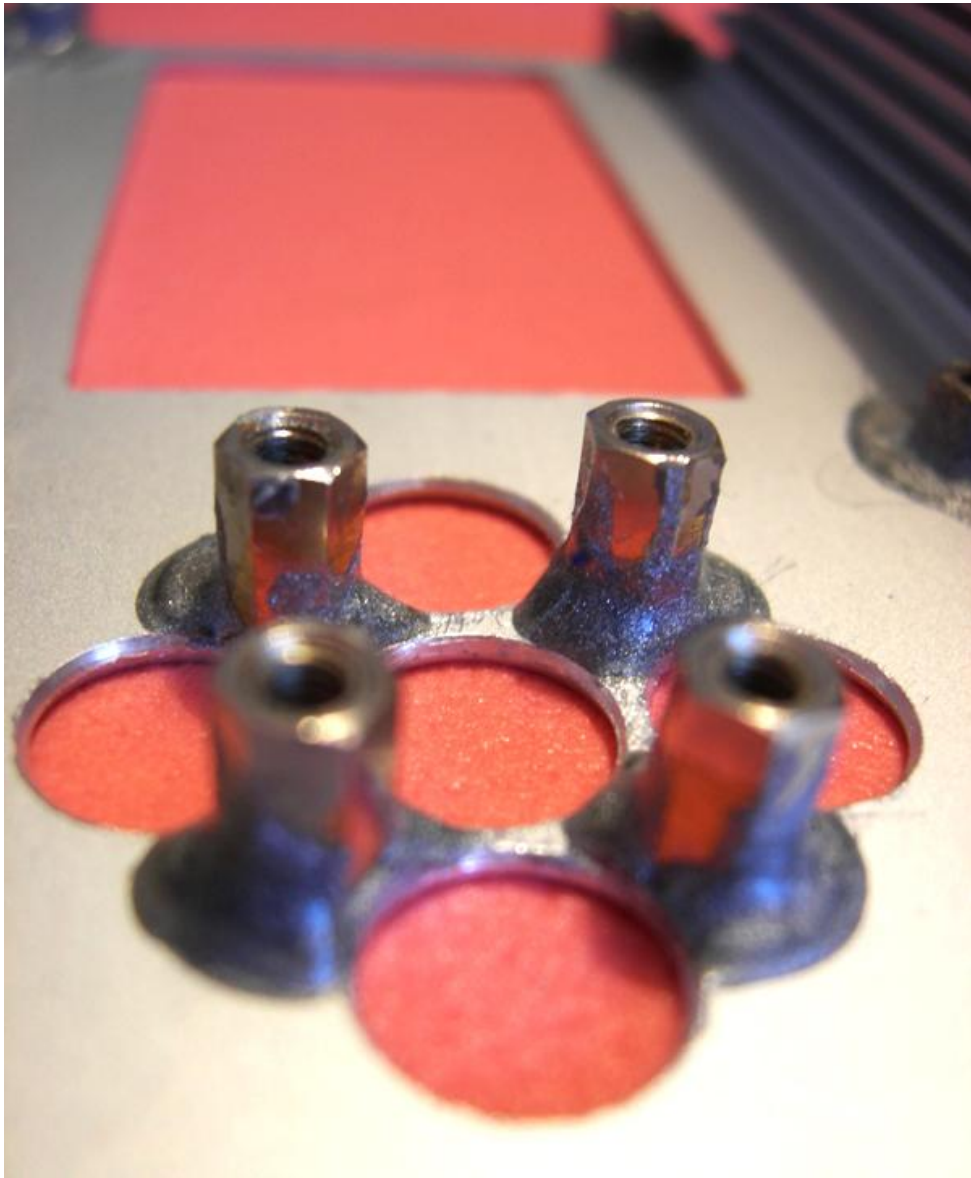
Obr. 24



Obr. 25



Obr. 26



Obr. 27

Závity, které jsou potřeba k uchycení displeje (Obr.27) a tlačítek ke krabičce, jsme přilepili dvousložkovým lepidlem Metalceys (Obr.28). Jednou složkou je syntetická pryskyřice a druhou jsou železné piliny – naprosto jemně nadrcené. Obě složky se pak v daném poměru smíchají a nanasou na lepenou plochu.



Obr. 28

## 12 Závěr

Teploměr, který jsme navrhli a realizovali byl nepřetržitě v provozu od 6.2.2011 do 17.3.2011. Po tuto dobu sbíral data z obou čidel a ukládal je na SD kartu. Vzhledem ke spotřebě zařízení je vhodné, aby bylo trvale připojeno k síťovému zdroji. Baterie slouží pouze jako záložní zdroj při výpadku dodávky elektrické energie nebo manipulaci se zařízením. V režimu chodu na baterii je podsvícení z důvodu vysoké energetické náročnosti softwarově deaktivováno.

Problém se objevil při výrobě krabičky, protože jsme na první pokus přesně neodhadli správnou velikost a plošný spoj s dalšími součástkami se do ní nevešel. Když jsme upravovali druhou krabičku (větší), nepovedlo se nám vyříznout dírky naprosto přesně, a proto výřezy nevypadají příliš „elegantně“. Na koupi další krabičky, kvůli těmto nezdarům, bohužel nevystačily peníze.

Naměřené teploty lze dále zpracovat v programu na PC (Příloha 11), který jsme také vytvořili. Tento program je naprogramován v jazyce C# ve vývojovém prostředí Visual Studio 2010. Výstup z tohoto programu je vidět v Příloze 10.

Zařízení by se v budoucnu dalo rozšířit o další teplotní čidla, bezdrátovou komunikaci mezi čidly a zařízením nebo o přidání funkce termostat s ovládáním externích zařízení.

## 13 Seznam použité literatury a zdrojů informací

### 13.1 Seznam použitých literárních publikací

- [1] Hájek, J. *Časovač 555 - praktická zapojení*. BEN, Technická literatura 2002
- [2] Herout, P. *Učebnice jazyka C - 1. díl* - 6. vydání České Budějovice, Kopp 2009

### 13.2 Jiné zdroje informací

- [3] <http://www.jreichl.com>
- [4] <http://www.vernier.cz>
- [5] <http://www.microchip.com/>
- [6] <http://cz.farnell.com/>
- [7] <http://support.microsoft.com>
- [8] <http://www.sandisk.com/Assets/File/OEM/Manuals/ProdManRS-MMCv1.3.pdf>
- [9] <http://alumni.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf>
- [10] <http://ww1.microchip.com/downloads/en/DeviceDoc/39631E.pdf>
- [11] [http://www.gme.cz/\\_dokumentace/dokumenty/311/311-038/dsh.311-038.1.pdf](http://www.gme.cz/_dokumentace/dokumenty/311/311-038/dsh.311-038.1.pdf)
- [12] [http://www.gme.cz/\\_dokumentace/dokumenty/513/513-073/dsh.513-073.1.pdf](http://www.gme.cz/_dokumentace/dokumenty/513/513-073/dsh.513-073.1.pdf)
- [13] [http://www.gme.cz/\\_dokumentace/dokumenty/513/513-073/dsh.513-073.2.pdf](http://www.gme.cz/_dokumentace/dokumenty/513/513-073/dsh.513-073.2.pdf)
- [14] [http://www.gme.cz/\\_dokumentace/dokumenty/312/312-003/dsh.312-003.1.pdf](http://www.gme.cz/_dokumentace/dokumenty/312/312-003/dsh.312-003.1.pdf)
- [15] [http://www.gme.cz/\\_dokumentace/dokumenty/330/330-058/dsh.330-058.1.pdf](http://www.gme.cz/_dokumentace/dokumenty/330/330-058/dsh.330-058.1.pdf)
- [16] [http://www.gme.cz/\\_dokumentace/dokumenty/330/330-165/dsh.330-165.1.pdf](http://www.gme.cz/_dokumentace/dokumenty/330/330-165/dsh.330-165.1.pdf)
- [17] [http://en.wikipedia.org/wiki/Zeller's\\_congruence](http://en.wikipedia.org/wiki/Zeller's_congruence)
- [18] [http://en.wikipedia.org/wiki/File\\_Allocation\\_Table#Initial\\_FAT16](http://en.wikipedia.org/wiki/File_Allocation_Table#Initial_FAT16)
- [19] [http://elm-chan.org/docs/mmc/mmc\\_e.html](http://elm-chan.org/docs/mmc/mmc_e.html)
- [20] <http://www.rw-designer.com/icon-detail/1939>
- [21] [https://dip.felk.cvut.cz/browse/pdfcache/richtp1\\_2007bach.pdf](https://dip.felk.cvut.cz/browse/pdfcache/richtp1_2007bach.pdf)
- [22] <http://doveda.byl.cz/lcd/index.htm>
- [23] [http://doveda.byl.cz/lcd/lcd\\_cz.htm](http://doveda.byl.cz/lcd/lcd_cz.htm)
- [24] <http://csharp.net-informations.com/excel/csharp-excel-tutorial.htm>
- [25] <http://hw.cz/Rozhrani/ART1215-Sbernice-1-Wire%E2%84%A2.html>

- [26] <http://korzar.sme.sk/c/4714555/vzduchovy-termoskop-zostrojl-galileo-galilei.html>
- [27] <http://www.quido.cz/objevy/teplomer.htm>
- [28] [http://nd02.jxs.cz/763/687/2ff11933a5\\_53353717\\_o2.jpg](http://nd02.jxs.cz/763/687/2ff11933a5_53353717_o2.jpg)
- [29] [http://www.solarnavigator.net/inventors/galileo\\_galilei.htm](http://www.solarnavigator.net/inventors/galileo_galilei.htm)
- [30] <https://www.distrelec.cz>
- [31] [http://www.gme.cz/\\_dokumentace/dokumenty/530/530-093/dsh.530-093.1.pdf](http://www.gme.cz/_dokumentace/dokumenty/530/530-093/dsh.530-093.1.pdf)
- [32] [http://www.gme.cz/\\_dokumentace/dokumenty/530/530-067/dsh.530-067.1.pdf](http://www.gme.cz/_dokumentace/dokumenty/530/530-067/dsh.530-067.1.pdf)
- [33] <http://www.adafruit.com/blog/2009/06/09/temperature-sensor-tutorial-using-the-tmp36/>
- [34] [http://www.mlab.cz/Articles/HowTo/How\\_to\\_make\\_PCB/DOC/HTML/How\\_to\\_make\\_PCB.cs.html](http://www.mlab.cz/Articles/HowTo/How_to_make_PCB/DOC/HTML/How_to_make_PCB.cs.html)



## **14 Použitý software**

Microchip MPLAB IDE v8.50

HI-TECH C COMPILER for PIC18 MCUs (PRO) v9.65

HI-TECH Universal Toolsuite v1.37

HxD – Hexeditor v1.7.7.0

Microsoft VISUAL STUDIO 2010

Microsoft WORD

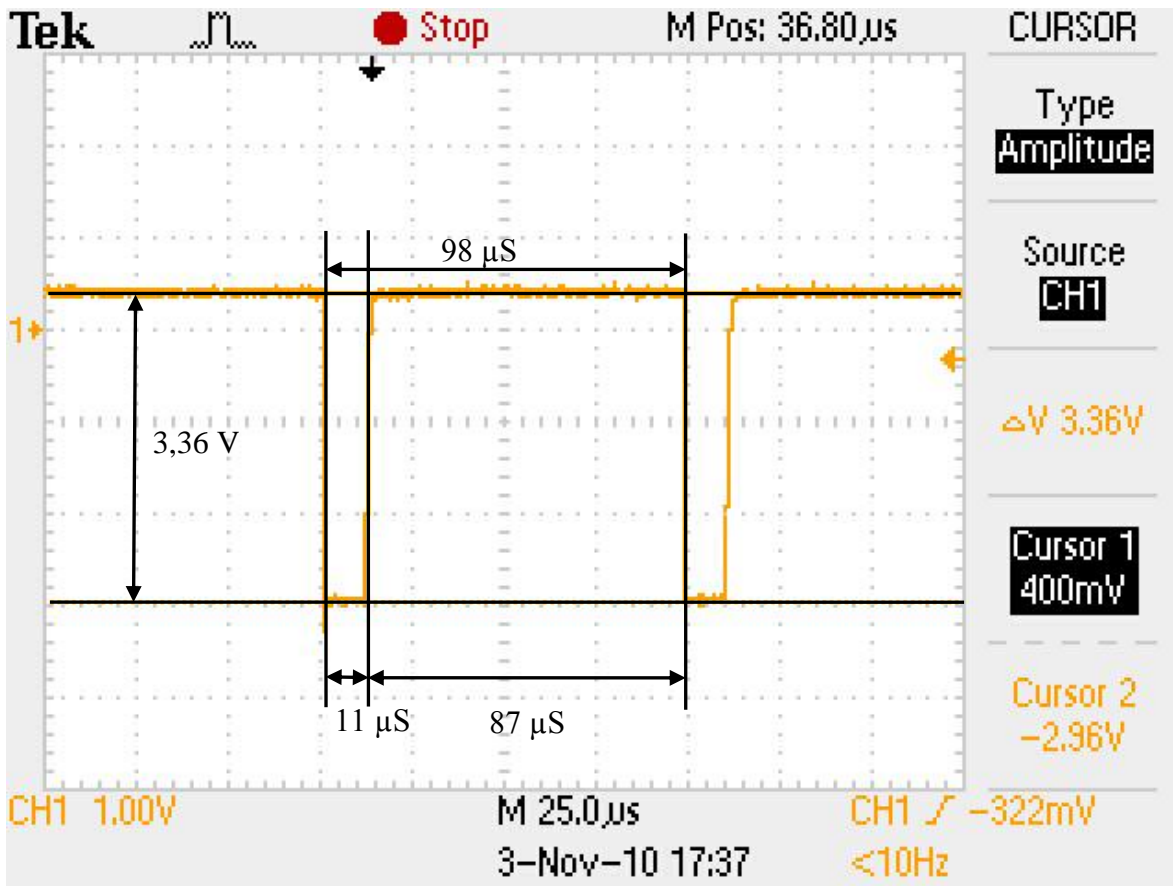
Microsoft EXCEL

EAGLE 5.7.0

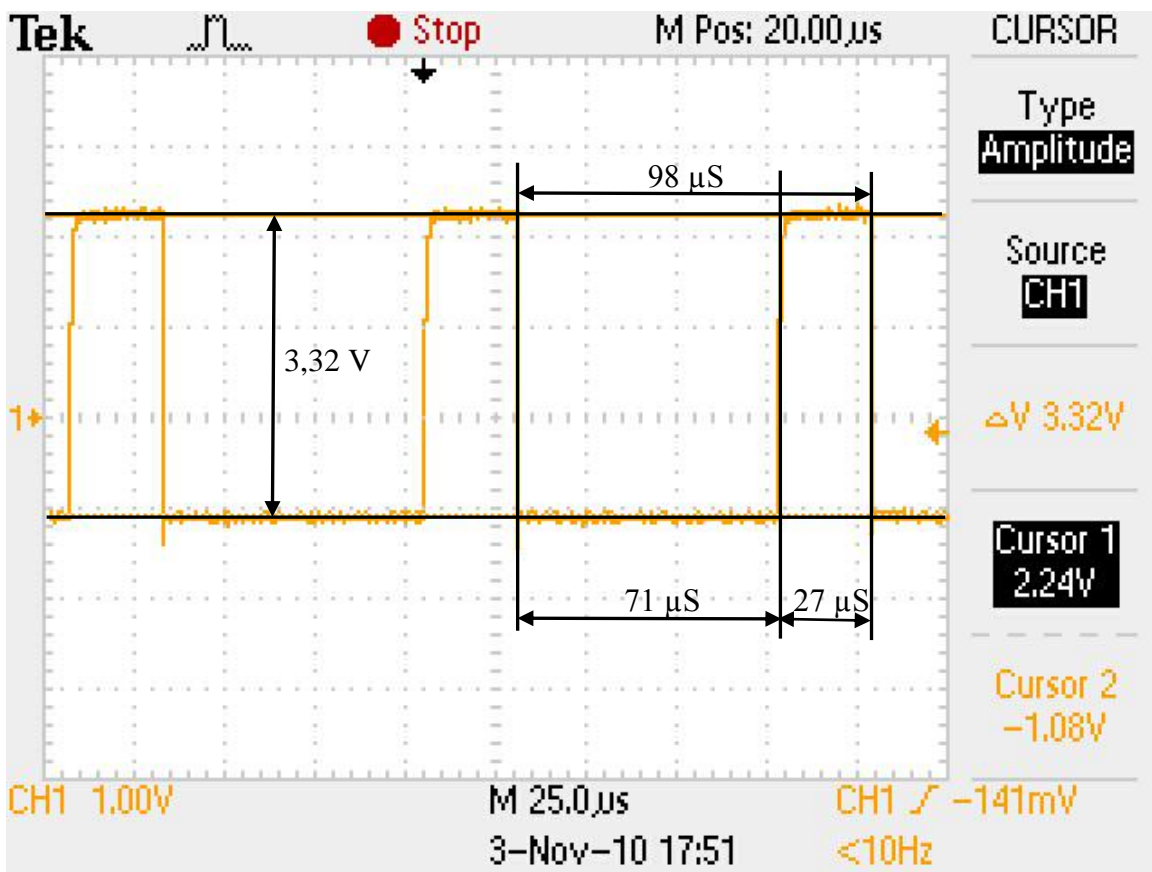
## 15 Přílohy



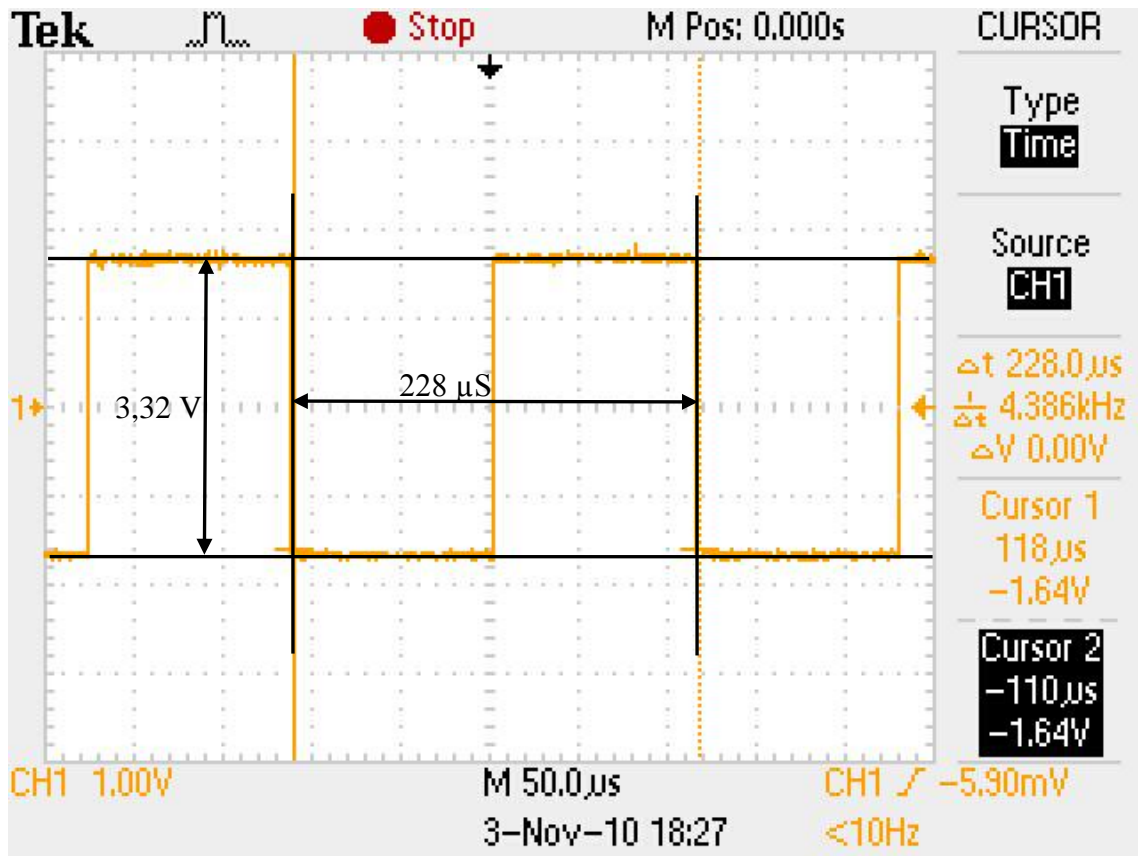
**Příloha 1**



Příloha 2

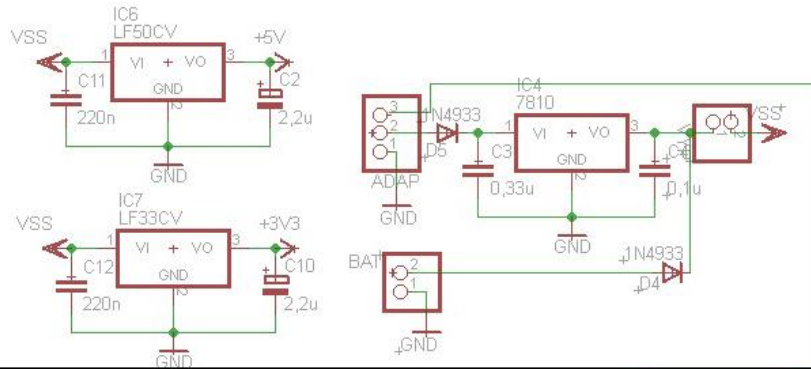


Příloha 3

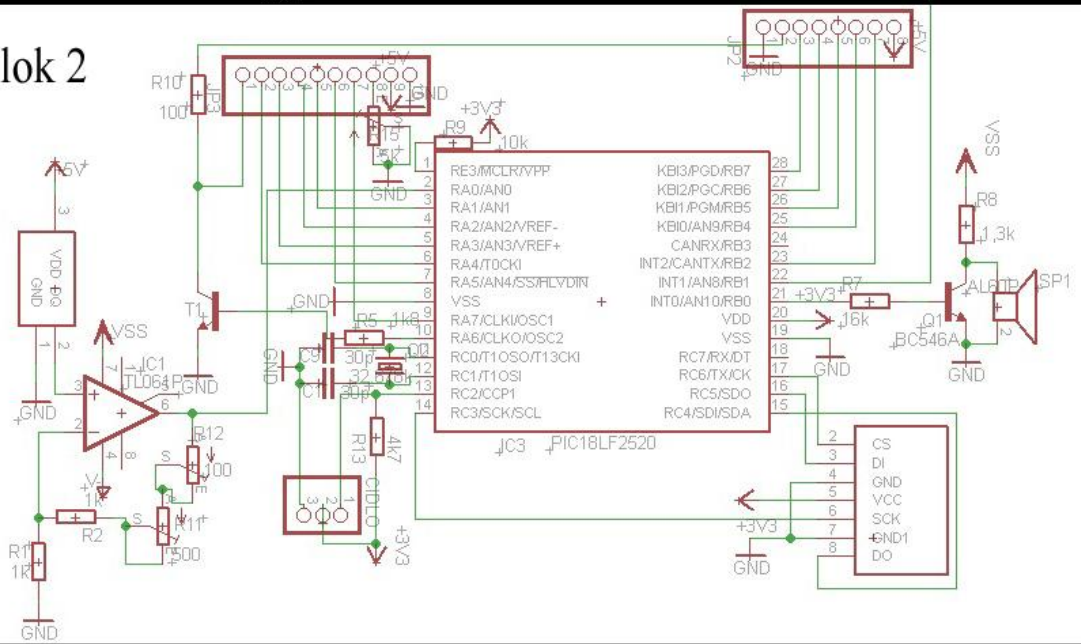


Příloha 4

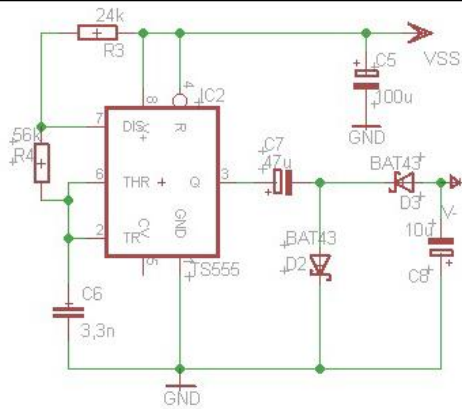
### Blok 1



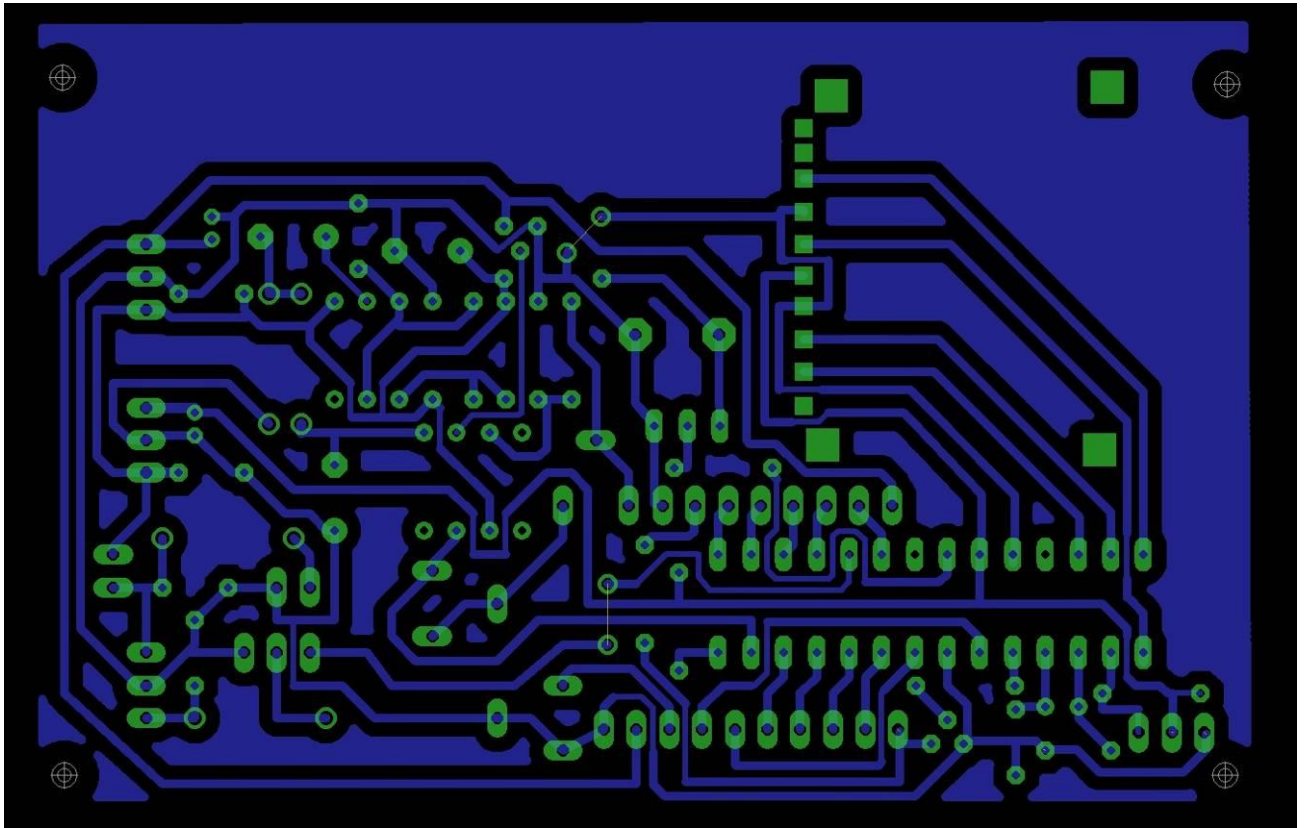
### Blok 2



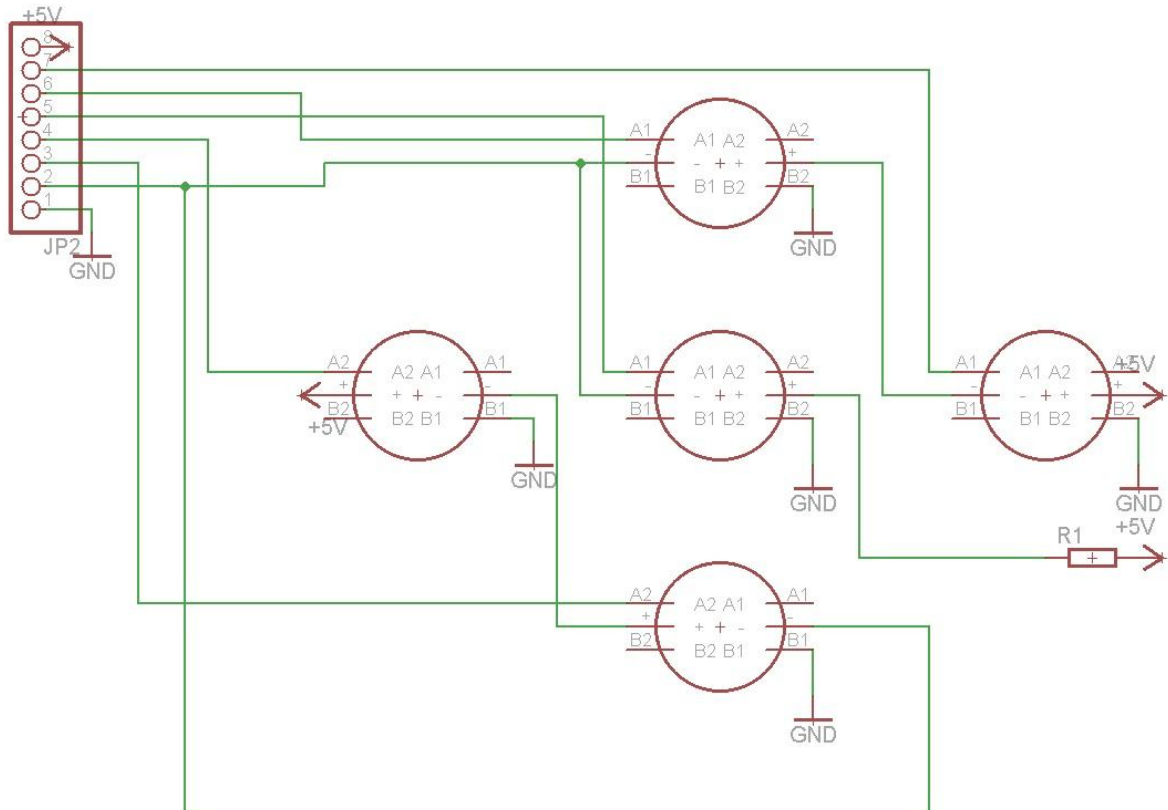
### Blok 3



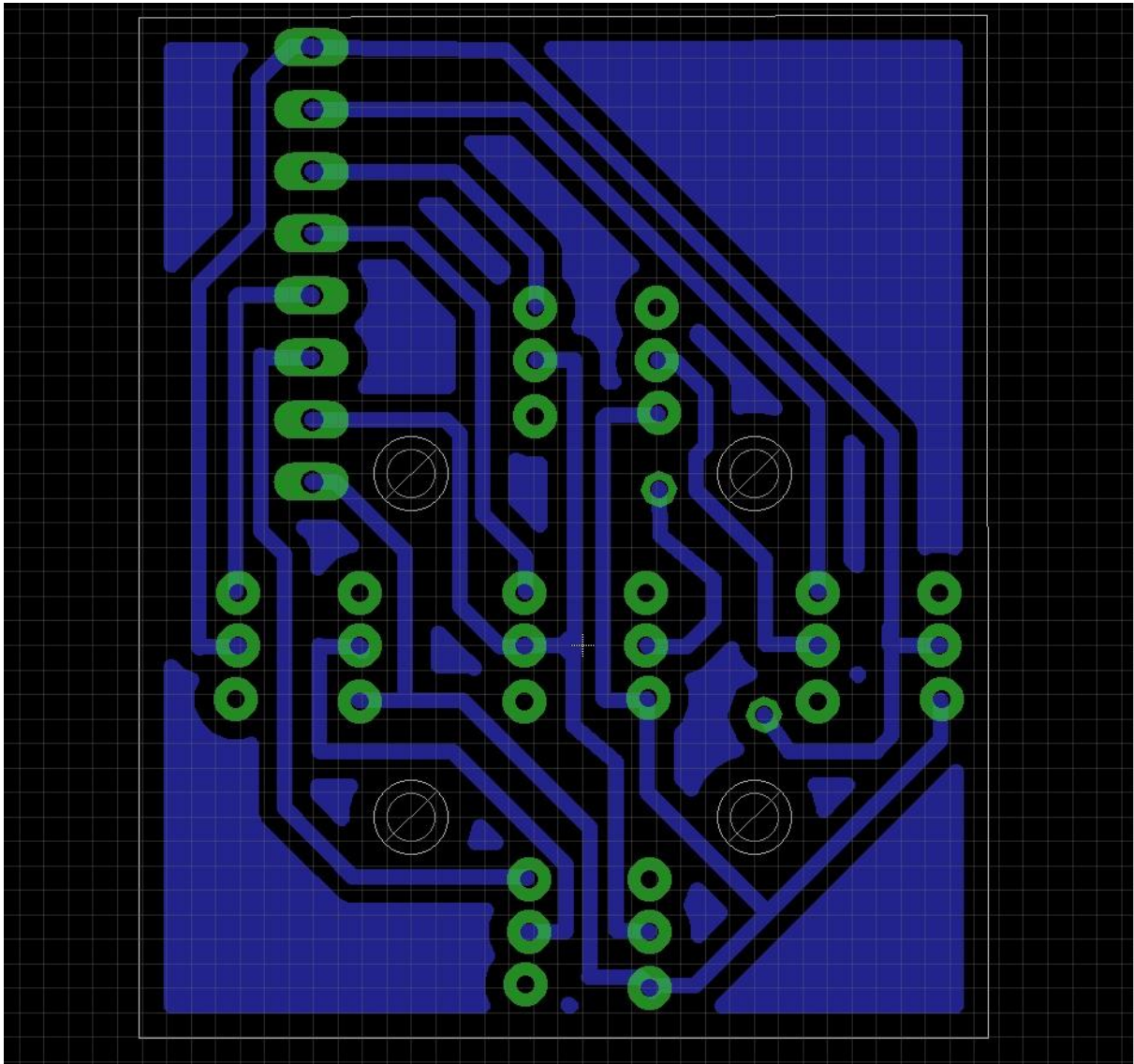
### Příloha 5



Příloha 6



Příloha 7



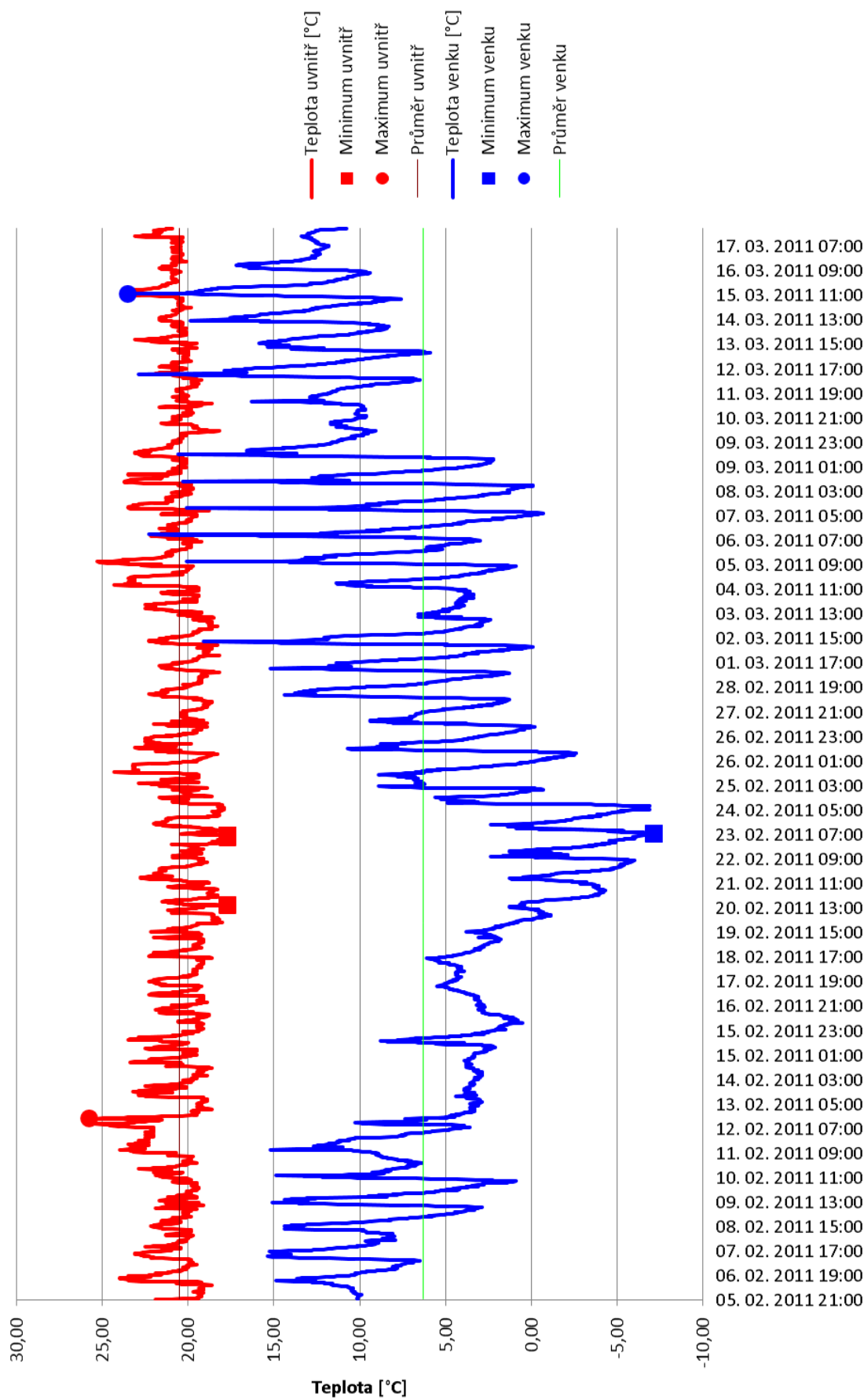
**Příloha 8**

Lower 4 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	a	P	`	P				-	タ	ミ	α	ρ
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	△	Δ	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	×	β	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	⊗
xxxx0100	(5)		\$	4	D	T	d	t			、	エ	ト	ト	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	工	σ	Ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ア	キ	ヌ	ラ	g	π
xxxx1000	(1)		(	8	H	X	h	x			ィ	ク	ネ	リ	γ	×
xxxx1001	(2)		)	9	I	Y	i	y			ウ	ケ	ル	ル	γ	γ
xxxx1010	(3)		*	:	J	Z	j	z			エ	コ	ン	レ	j	キ
xxxx1011	(4)		+	;	K	C	k	c			オ	サ	ヒ	ロ	*	π
xxxx1100	(5)		,	<	L	¥	l	l			カ	シ	フ	ワ	φ	円
xxxx1101	(6)		-	=	M	]	m	)			ユ	ズ	ン	ン	も	÷
xxxx1110	(7)		.	>	N	^	n	→			ヨ	セ	ホ	°	ñ	
xxxx1111	(8)		/	?	O	_	o	+			ッ	ソ	マ	°	ö	■

Příloha 9



Teploty za období od 05. 02. 2011 00:00 do 17. 03. 2011 23:45 v časovém intervalu od 00:00 do 23:45



Příloha 10

**Zpracování hodnot**

**Výběr období**

Začátek rozsahu: 05. 02. 2011 00:00

Konec rozsahu: 17. 03. 2011 23:45

Časový interval od: 00:00 do: 23:45

**Zpracování**

Exportovat data a vytvořit graf

Zobrazit extrémy v grafu

Otevřít soubor vyexportovaných dat

Zpracovat

**Teplota uvnitř**

Minimální 17,7 °C

20. 02. 2011 14:45

23. 02. 2011 04:00

23. 02. 2011 04:30

23. 02. 2011 05:00

Maximální 25,8 °C

12. 02. 2011 15:00

Průměrná 20,5 °C

**Teplota venku**

Minimální -7,1 °C

23. 02. 2011 07:00

Maximální 23,5 °C

15. 03. 2011 11:45

Průměrná 6,3 °C

Příloha 11

```

ClearAll["Global`*"];
jmenosouborudata = "hodnoty2.xls"; (*Jmeno souboru,
ve kterem jsou ulozena namerena data.*)
cesta = NotebookDirectory[]; (*V promenne cesta je nastavena aktualni cesta,
tj. cesta k souboru, který je ve stejne slozce jako tento notebook.*)
dataa = Import[StringJoin[cesta, jmenosouborudata]; (*Nacteni hodnot z excelu.*)
napeti = Table[ToExpression[dataa[[1, i, 2]], {i, 1, Length[dataa[[1]]}]];
(*Napeti odpovida druhemu sloupci v excelu.*)
teplota = Table[ToExpression[dataa[[1, i, 1]], {i, 1, Length[dataa[[1]]}]];
(*Teplota odpovida prvniku sloupci v excelu.*)
data = Table[{teplota[[i]], napeti[[i]], {i, 1, Length[teplota]}}];
(*Vytvoreni usporadanych dvojic pro graf.*)

pocetdat = Length[data];

velikostbodu = 0.02; (*Zadefinovani velikosti a barev s kterymi budeme pracovat.*)
velikostpisma = 12;
tloustka = 0.005;
barva = Black;
barvaprimky = Blue;
barvadat = Red;

data = Table[{data[[i]][1], data[[i]][2]}, {i, pocetdat}]; (*Vyber dat z excelu.*)
body = ListPlot[data, PlotStyle -> Directive[PointSize[velikostbodu], barvadat]];

soucetodchylek[a_, b_] := Sum[(a data[[i]][1] + b - data[[i]][2])^2, {i, 1, pocetdat}]; (*Soucet odchylek.*)
derivacea = D[soucetodchylek[a, b], a]; (*Derivace podle a.*)
derivaceb = D[soucetodchylek[a, b], b]; (*Derivace podle b.*)

reseni = Solve[{derivacea == 0, derivaceb == 0}, {a, b}]; (*Vyreseni rovnice zderivovanych.*)
akoef = a /. reseni[[1, 1]]; (*Vytazeni vysledku(koeficientu) z prvni rovnice.*)
bkoef = b /. reseni[[1, 2]]; (*Vytazeni vysledku(koeficientu) z druhe rovnice.*)

funkce[i_] := akoef i + bkoef; (*Vysledny tvar predpisu funkce.*)

graf = Plot[funkce[i], {i, -30, 1.05 data[[Length[data]]][1]},
PlotStyle -> {barvaprimky, Thickness[tloustka]}]; (*Funkce pro vykresleni dat.*)
Show[body, graf,
PlotRange -> {{-40, 1.1 data[[Length[data]]][1]}, {0, 1.1 data[[Length[data]]][2]}},
TicksStyle -> Directive[barva, velikostpisma],
AxesLabel -> {Style[" $\frac{t}{C^\circ}$ ", velikostpisma, barva, FontFamily -> Times],
Style[" $\frac{U}{V}$ ", velikostpisma, barva, FontFamily -> Times]},
ImageSize -> 500, FormatType -> StandardForm]
Print["Aproximacni funkce ma tvar: U = ", funkce'[x], " t + ", funkce[0], "."]
(*Vypis vysledneho predpisu funkce.*)

```

## Příloha 12