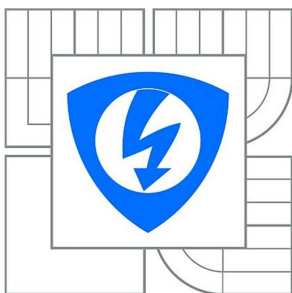


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ULTRAZVUKOVÉ MĚŘENÍ VZDÁLENOSTI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL KLICH

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. IVO LATTENBERG, Ph.D.

BRNO 2017

ANOTACE

Cílem bakalářské práce je návrh, výroba a vývoj řídicího programu zařízení pro měření vzdálenosti pomocí ultrazvukového čidla. Zaměřuje se na detailní popis tvorby celé práce od základních vlastností použitých komponentů přes tvorbu schémat a návrh DPS. Dále detailně popisuje vývoj softwaru, řídicího celé zařízení. Stručně popisu a předkládá nejdůležitější části kódu. Shrnuje všechny teoretické i praktické poznatky získané během celé tvorby, vysvětluje principy funkcí a práce jednotlivých částí a navrhuje nejlepší možné varianty celého zapojení. Popisuje mikroprocesor řady AVR ATmega8 od společnosti ATMEL, který řídí celé zařízení, dále paměť typu EEPROM čip 24LCXXX, ultrazvukový sensor SRF05 a komunikaci s PC pomocí rozhraní USB a mikročipu FT230XS-R.

KLÍČOVÁ SLOVA

AVR, ATmega8, 24LCXXX, SRF05, FT230XS-R, EEPROM, LMP36GT9Z, MAX232, ultrazvukový senzor, teplotní čidlo, PC, USB, USART, RS232, DPS

ABSTRACT

The aim of this bachelor thesis is the design and manufacture of a controlling program of equipment for distance measurement using ultrasonic sensors. It focuses on a detailed description of the whole design from the basic properties of the components to the creation of schemes and PCB design. It subsequently describes in detail the development of controlling software of the equipment. The work briefly describes the most important parts of the code. It summarizes all theoretical and practical knowledge gained during the creation of the whole design, it explains the principles of the functions and works of individual parts and it suggests the best possible options of connection of the whole equipment. It also suggests the best possible variants of the entire circuit. The bachelor thesis also describes the microprocesor AVR ATmega8 from the Atmel Corporaton, which controls the entire device, as well as EEPROM memory chip 24LCXXX, SRF05 ultrasonic sensor and communication with the PC via USB using a microchip FT230XS-R.

KEYWORDS

AVR, ATmega8, 24LCXXX, SRF05, FT230XS-R, EEPROM, LMP36GT9Z, MAX232, ultrasonic sensor, temperature sensor, PC, USB, USART, RS232, DPS

KLICH, M. *Ultrazvukové měření vzdálenosti*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 36s. p7s. Vedoucí semestrální práce doc. Ing. Ivo Lattenberg, Ph.D..

PROHLÁŠENÍ

Prohlašuji, že svou semestrální práci na téma Ultrazvukové měření vzdálenosti jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené semestrální práce dále prohlašuji, že v souvislosti s vytvořením této semestrální práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení S 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 8.6.2017

.....
(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu semestrální práce doc. Ing. Ivu Lattenbergovi, Ph.D. za odbornou pomoc v teoretických a praktických částech této semestrální práce. Panu Lukáši Pazderovi za praktickou pomoc při výrobě DPS. Dále své rodině za skvělou psychickou podporu.

V Brně dne 8.6.2017

.....
(podpis autora)

OBSAH

OBSAH.....	1
SEZNAM OBRÁZKŮ.....	3
SEZNAM TABULEK	4
1 ÚVOD.....	5
2 MĚŘENÍ VZDÁLENOSTI	6
2.1 ULTRAZVUKOVÝ SENZOR SRF05.....	6
2.2 MĚŘENÍ TEPLoty.....	9
3 POPIS ČÁSTÍ ZAŘÍZENÍ.....	10
3.1 MIKROPROCESOR ATMEGA8.....	10
3.1.1 ČÍTAČ/ČASOVAČ.....	11
3.1.2 UKLÁDÁNÍ DAT.....	11
3.1.3 OVLÁDÁNÍ.....	12
3.1.4 ZVUKOVÁ SIGNALIZACE.....	12
3.2 KOMUNIKACE S PC PŘES ROZHRANÍ USB.....	12
3.2.1 PRINCIP SÉRIOVÉ LINKY (USART).....	14
3.2.2 PŘEVODNÍK FT230XS-R.....	15
3.2.3 PŘEVODNÍK MAX232	15
4 VÝVOJ ULTRAZVUKOVÉHO ZAŘÍZENÍ V PROSTŘEDÍ EAGLE	17
4.1 TVORBA SCHÉMAT	17
4.2 TVORBA VLASTNÍ KNIHOVNY S KOMPONENTY	19
4.2.1 TVORBA SYMBOLU	20
4.2.2 TVORBA POUZDRA.....	21
4.2.3 FINÁLNÍ TVORBA SOUČÁSTKY.....	21
4.3 NÁVRH DESKY PLOŠNÝCH SPOJŮ (DPS).....	22
4.4 VÝROBA DPS A OSAZENÍ SOUČÁSTEK.....	23
5 VÝVOJ ŘÍDÍCÍHO PROGRAMU	25
5.1 PROGRAM AVR STUDIO 6	25
5.2 PROGRAM ULTRAZVUKOVÉHO ZAŘÍZENÍ.....	26
5.2.1 KOMUNIKACE S LCD	27
5.2.2 KOMUNIKACE S KLÁVESNICÍ.....	29
5.2.3 ZPRACOVÁNÍ DAT ULTRAZVUKOVÉHO A TEPLOTNÍHO SENZORU	31
5.2.4 UKLÁDÁNÍ DAT.....	33
6 OŽIVENÍ A TESTOVÁNÍ.....	35
6.1 BOOTLOADER.....	35
6.2 ULOŽENÍ ŘÍDÍCÍHO PROGRAMU	36
6.3 TESTOVÁNÍ ULTRAZVUKOVÉHO ZAŘÍZENÍ	37

7	ZÁVĚR.....	40
8	SEZNAM POUŽITÉ LITERATURY.....	41
9	SEZNAM ZKRATEK	42
10	SEZNAM PŘÍLOH	43
10.1	SCHÉMA ZAPOJENÍ	44
10.2	DPS (DESKY PLOŠNÝCH SPOJŮ)	46
10.3	SEZNAM SOUČÁSTEK	48
10.3.1	ŘÍDÍCÍ DESKA	48
10.3.2	ULTRAZVUKOVÝ SENZOR S ČIDLEM TEPLoty	49
10.3.3	KLÁVESNICE	49
10.3.4	LCD	49
10.4	FOTKY ZAŘÍZENÍ	50

SEZNAM OBRÁZKŮ

Obr. 2.1 - Ultrazvukový senzor SRF05.....	7
Obr. 2.2 - Náhradní schéma zapojení SRF05 pro přijímání a vysílání.....	7
Obr. 2.3 - Seznam a ukázka výstupů ultrazvukového senzoru SRF05.....	8
Obr. 2.4 - Ukázka signálů buzení, vysílání a přijímání.....	9
Obr. 2.5 - Směrová charakteristika vyzařování senzoru SRF05.....	9
Obr. 2.6 - Teplotní čidlo TMP36GT9Z.....	9
Obr. 3.1 - Mikroprocesor ATmega8.....	10
Obr. 3.2 - Výpis všech portů mikroprocesoru ATmega8.....	10
Obr. 3.3 - EEPROM 24LCXXX.....	11
Obr. 3.4 - LCD displej WH1602A.....	12
Obr. 3.5 - Elektromagnetický bzučák.....	12
Obr. 3.6 - Standartní TTL hradlo NAND.....	13
Obr. 3.7 - Koncovky rozhraní USB.....	13
Obr. 3.8 - Diagram pro odesílání a přijímání 8-bitů.....	14
Obr. 3.9 - Převodník FT230XS-R.....	15
Obr. 3.10 - Schéma zapojení převodníku USB do RS232 dle výrobce.....	15
Obr. 3.11 - MAX232 - převodník TTL do RS232.....	16
Obr. 3.12 - Náhradní schéma MAX232 s ATmega8.....	16
Obr. 4.1 - Control Panel programu Eagle.....	17
Obr. 4.2 - Editor schémat.....	18
Obr. 4.3 - Použití BUS v editoru schémat.....	19
Obr. 4.4 - Použití odkazu pro propojení součástek v editoru schémat.....	19
Obr. 4.5 - Editor knihovny součástek.....	20
Obr. 4.6 - Editor schématické značky (FT230XS-R).....	20
Obr. 4.7 - Editor pouzder (FT230XS-R).....	21
Obr. 4.8 - Editor finálního vytvoření součástky (FT230XS-R).....	21
Obr. 4.9 - Výběrové okno pro propojení pinů (FT230XS-R).....	22
Obr. 4.10 - Editor návrhu DPS.....	23
Obr. 6.1 - Programátor USBasp.....	35
Obr. 6.2 - Zapojení ATmega8 s USBasp.....	36
Obr. 10.1 - Řídící deska.....	44
Obr. 10.2 - Ultrazvukový senzor s teplotním čidlem.....	45
Obr. 10.3 - Klávesnice.....	45
Obr. 10.4 - LCD s ovládáním kontrastu a podsvětlením.....	45
Obr. 10.5 - Řídící deska – pohled TOP (spoje bez součástek).....	46
Obr. 10.6 - Řídící deska - pohled TOP (osazení vč. propojek J1-J4).....	46
Obr. 10.7 - Ultrazvukový senzor s teplotním čidlem – pohled TOP (spoje bez součástek).....	46
Obr. 10.8 - Ultrazvukový senzor s teplotním čidlem – pohled TOP (osazení).....	46
Obr. 10.9 - Klávesnice - pohled TOP (spoje bez součástek).....	47
Obr. 10.10 - Klávesnice - pohled TOP (osazení).....	47
Obr. 10.11 - LCD s ovládáním kontrastu a podsvětlením – pohled TOP (spoje bez součástek).....	47
Obr. 10.12 - LCD s ovládáním kontrastu a podsvětlením - pohled TOP (osazení).....	48
Obr. 10.13 - Zařízení.....	50
Obr. 10.14 - Hardware zařízení.....	50

SEZNAM TABULEK

Tab. 2.1 - Rychlost zvuku závislá na teplotě.....	6
Tab. 5.1 - Zvolení porovnávacího napětí pro ADC převodník.....	32
Tab. 5.2 - Dělicí poměr pro výpočet pracovní frekvence ADC převodníku	32
Tab. 6.1 - Testování zařízení v uzavřeném prostoru	38
Tab. 6.2 - Testování zařízení v otevřeném prostoru	38
Tab. 6.3 - Testování teplotního senzoru TMP9GTZ	38

1 ÚVOD

Tato bakalářská práce se zabývá problematikou měření vzdálenosti pomocí ultrazvukového senzoru. Cílem bylo vytvořit zařízení ovládané mikroprocesorem společnosti ATMEL. Výstup zařízení je směřován na LCD přes ovládání klávesnicí. Pro ukládání naměřených hodnot je vytvořena komunikace s externí pamětí EEPROM. Dále je zařízení obohaceno o teplotní senzor pro přesnější výpočet vzdálenosti a zvukový výstup pro upozornění dle nastavených požadavků.

Toto téma jsem si vybral pro zajímavé řešení práce a z důvodu využití v praxi jako např.: parkovací asistent ve vozidlech, měření vzdálenosti překážky při jízdě autonomního robota apod.

Tento dokument bakalářské práce popisuje celý postup návrhů, výroby zařízení a vývoje řídicího programu. Popisuje nejpodstatnější informace k úspěšné tvorbě funkčního zařízení, vysvětluje principy a teorii pro pochopení probírané problematiky a nastiňuje nejjednodušší a nejefektivnější postupy.

Zařízení bude dále testováno v uzavřených a venkovních prostorách a z vypočtených výsledků a vytvořených grafů bude určena průměrná odchylka vyrobeného zařízení. Na přiloženém CD je umístěna dokumentace, všechny programy potřebné k výrobě zařízení, důležité datasheety součástek, ovladače pro programátor USBasp a řídicí program zařízení v jazyce C.

2 MĚŘENÍ VZDÁLENOSTI

Pro měření vzdálenosti se v praxi využívá velkého množství senzorů a snímačů jakou jsou např.: optické (infračervený, laserový) a zvukové (ultrazvukový). Každá z těchto skupin má své výhody i nevýhody.

U optických senzorů platí pravidlo, že okolní světlo a teplota povrchu (platí pro infračervené senzory) značně ovlivňují přesnost daného měření. Signál (paprsek) se může buďto zkreslit nebo zcela ztratit. Jejich výhodou je však velmi dobrá přesnost a díky rychlosti světelného paprsku také rychlost a velká dosažená vzdálenost měření.

U zvukových senzorů se nemusíme starat o teplotu objektu nebo okolní osvětlení. Problém je však velká náchylnost na teplotu okolního vzduchu. Čím je teplota vyšší, tím je vzduch řidší a zvuk se tak může pohybovat mnohem rychleji. Opačný efekt mají zase nízké teploty. Rozdíl rychlosti zvuku v závislosti na teplotě je uveden v tabulce 2.1. Pro přesnější měření je proto vhodné využít teplotní senzor umístěný co nejbližší ke zdroji signálu. Výhodou je však nízká pořizovací cena a, vyjma okolní teploty, minimální citlivost na okolní prostředí. Právě na tuto skupinu senzorů je zaměřena tato práce.

Ultrazvukový signál se skládá z akustického vlnění, jehož frekvence je vyšší než schopnost poslechu lidského ucha, která je udávána v rozsahu 20Hz-20kHz. Jelikož je vlnová délka ultrazvuku menší než vlnová délka slyšitelného zvuku, je ultrazvuk méně náchylný na ohyb, je méně absorbován kapalinami a pevnými látkami. Díky svým vlastnostem má velké uplatnění v medicíně či v průmyslu (SONO, ultrazvukové čištění, zvlhčování vzduchu, měření průtoku kapalin a plynů).

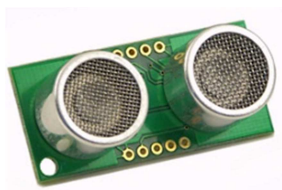
t [°C]	v [ms^{-1}]
-20	319
-10	325
0	331
10	337
20	343
30	349
40	355
50	360
100	387
200	436
300	480
400	520

Tab. 2.1 - Rychlost zvuku závislá na teplotě

2.1 ULTRAZVUKOVÝ SENZOR SRF05

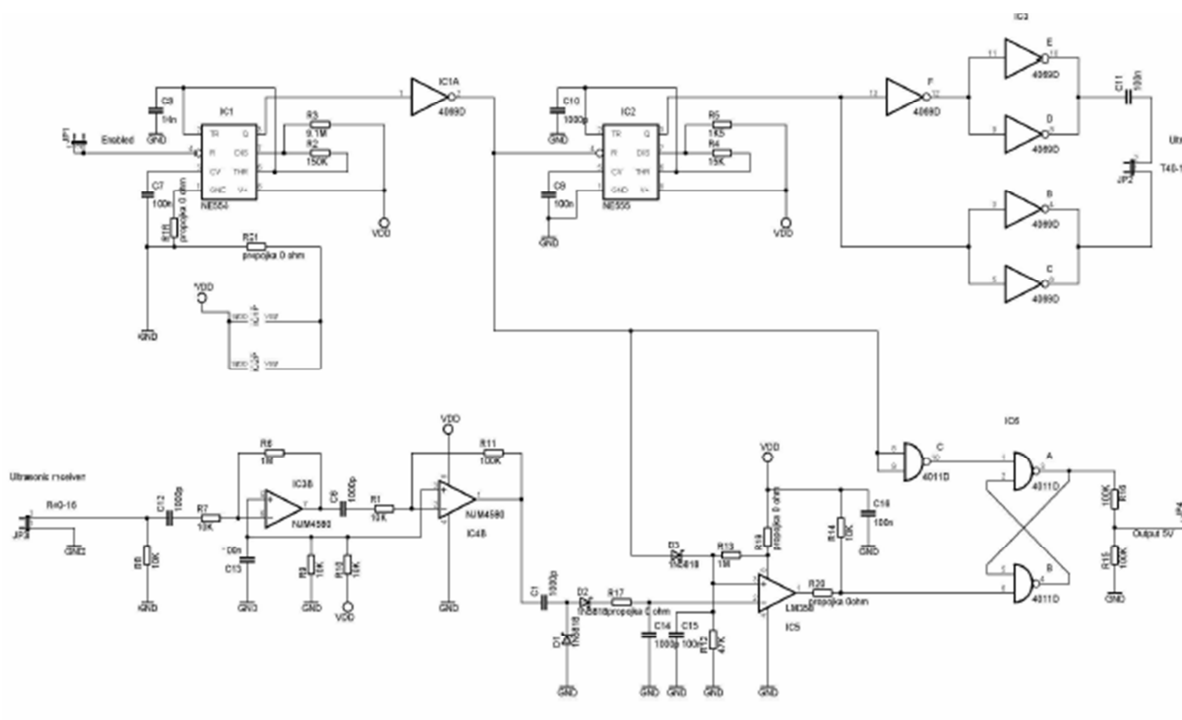
Pro svou práci jsem si vybral ultrazvukový senzor SRF05. Má zabudovaný obvod pro tvorbu horní a dolní půlvlny signálu a pro jeho zpětné zpracování. Práce s ním je velmi jednoduchá a podmínky pro jeho správnou funkci jsou minimální. V praxi se často využívá

ve školních projektech a u složitějších elektronických stavebnic, nejčastěji u malých autonomních robotů.



Obr. 2.1 - Ultrazvukový senzor SRF05

Je zkompletován ze dvou samostatných ultrazvukových senzorů. Ty jsou připojeny na dva samostatné obvody pro rozložení a vysílání, a pro přijímání a zpracování daného signálu. Senzory jsou vysílač UST-40T (T - transmitter) a přijímač UST-40R (R - receiver). Oba pracují o kmitočtu 40kHz s citlivostí 5.0mV/Pa/1kHz a SPL 120dB. Toto rozlišení nepatří mezi nejvýkonnější ultrazvukové senzory, pro tuto práci však bohatě postačí. Výrobce udává přesnost měření 3-5mm v závislosti na podmínkách měření. U novějších modelů, např. SRF08, je udávána přesnost výrobcem 3mm a nižší. Maximální měřená vzdálenost Je pak u tohoto modelu 6m. Jejich cena je však podstatně vyšší.



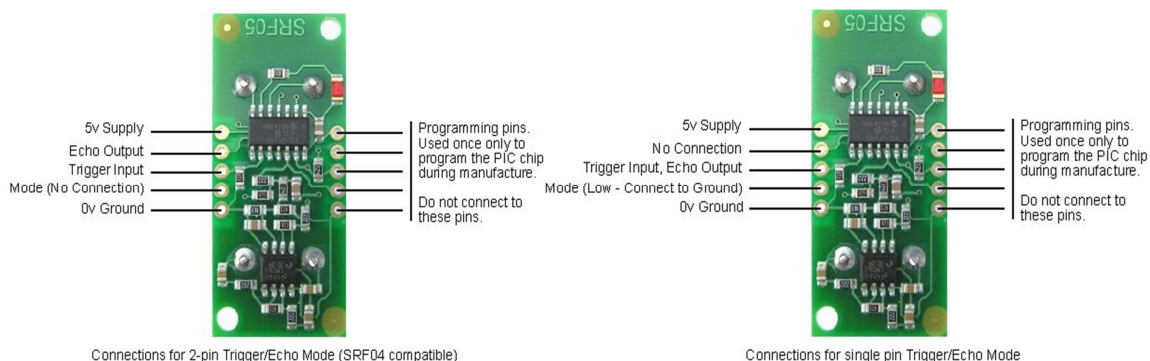
Obr. 2.2 - Náhradní schéma zapojení SRF05 pro přijímání a vysílání

Ultrazvukový senzor SRF05 pracuje ve dvou typech nastavení komunikace s mikroprocesorem:

- vysílá a přijímá impulzy pomocí dvou na sobě nezávislých pinů
- vysílá a přijímá impulzy pouze jedním pinem

Druhá možnost šetří místo a snižuje počet propojení. V programu mikroprocesoru se pak přepíná mezi výstupem na senzor a vstupem, který přijímá odražený signál. Z důvodů

lepší přehlednosti ukázky práce se senzorem jsem se rozhodl pro tuto práci využít první z těchto možností.



Obr. 2.3 - Seznam a ukázka výstupů ultrazvukového senzoru SRF05¹

Pro práci se senzorem SRF05 je nutné znát doby udávající, jak dlouho máme odesílat impuls pro buzení vysílače, kdy můžeme začít přijímat odražený signál a jak dlouho. Aby senzor začal pracovat, je nutné na něj nejprve vyslat z mikroprocesoru budící signál o hodnotě log.1, který je následně zpracován interním obvodem, viz obrázek 2.2. Odeslaný signál má mít dle výrobce minimální délku 10 μ s, což je doba potřebná pro vybuzení membrány ultrazvukového čidla. Zpracovaný signál je posléze odeslán ve formě ultrazvukových vln v pásmu 40kHz do prostoru.

Po určité době se vlny odrazí od objektu a putují zpět k přijímači. Zde jsou opět pomocí druhého interního obvodu, viz obrázek 2.2 zpracovány a signál je odeslán v hodnotě log.1 na vstupní pin mikroprocesoru, kde je měřena prodleva této odezvy. Po změření její délky můžeme vypočítat vzdálenost objektu. Časový interval mezi odesláním a příjmem signálu však nesmí být delší než 30ms. Pokud během této doby přijímač nezachytí odražený signál, je signál považován za ztracený a objekt není detekován. Tímto časovým úsekem je dána i maximální měřitelná vzdálenost. Doby délek trvání jsou znázorněny na obrázek 2.4.

$$s = v_0 * t_{max} = 330ms^{-1} \cdot 0,03s = 9,9 \quad [m] \quad (2.1)$$

s – dráha uražená signálem od zdroje k překážce a zpět
 v_0 – rychlost zvuku za sec při teplotě 0°C
 t_{max} – maximální zadaná návratová doba

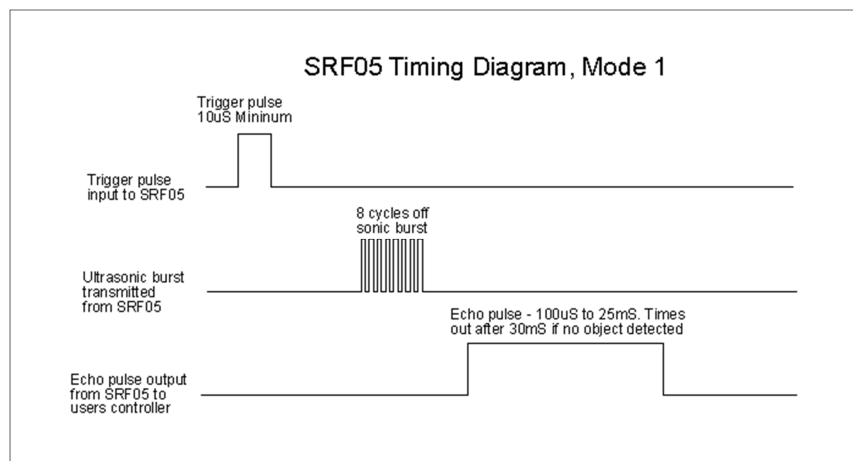
Vztah pro výpočet dráhy se však udává vzdálenost uraženou od zdroje k překážce a zpět k přijímači. Pro zjištění vzdálenosti pouze k překážce je proto nutné tuto dráhu vydělit dvěma.

$$s_{max} = \frac{s}{2} = \frac{9,9m}{2} = 4,95 \quad [m] \quad (2.2)$$

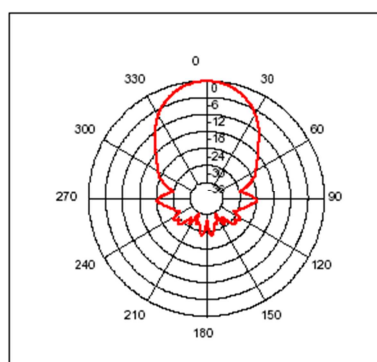
s_{max} – maximální měřitelná vzdálenost zařízení při teplotě 0°C

Výsledná dráha 495cm je tak maximální měřitelná vzdálenost zařízení při teplotě 0°C.

¹ Zdroj: datasheet součástky <http://www.robot-electronics.co.uk/html/srf05tech.htm>



Obr. 2.4 - Ukázka signálů buzení, vysílání a přijímání²



Obr. 2.5 - Směrová charakteristika vyzařování senzoru SRF05³

2.2 MĚŘENÍ TEPLOTY

Jelikož je ultrazvukový signál citlivý na okolní teplotu, bylo nutné do zařízení přidat teplotní čidlo. To je umístěno hned vedle ultrazvukového senzoru, aby byl odečet teploty co nejpřesnější. V této práci je použito čidlo LMP36GT9Z s analogovým výstupem. Jeho teplotní rozsah je -40°C do 125°C a citlivost je výrobcem udávána $10\text{mV}/^{\circ}\text{C}$. Ze vztahu napětí $750\text{mV}/^{\circ}\text{C}$ v datasheetu je možné vyčíst, že je hodnota posunuta o 500mV . Tato hodnota určuje rozdíl mezi kladnými a zápornými teplotami. Hodnota 500mV je rovna 0°C . Jeho výstup je přiveden na port mikroprocesoru obsahující A/D převodník. Signál tak bude převeden do digitální podoby a následně zpracován a vložen do vzorce pro přesný výpočet vzdálenosti.



Obr. 2.6 - Teplotní čidlo TMP36GT9Z

² Zdroj: datasheet součástky <http://www.robot-electronics.co.uk/html/srf05tech.htm>

³ Zdroj: datasheet součástky <http://www.robot-electronics.co.uk/html/srf05tech.htm>

3 POPIS ČÁSTÍ ZAŘÍZENÍ

Zařízení je ovládáno mikročipem ATmega8 řady AVR od společnosti ATMEL. Hlavní rozhraní je ovládáno pomocí klávesnice osazené mikrospínači a zobrazení výstupu programu je realizováno za pomoci alfanumerického LCD displeje. Pro upozornění např. mimo rozsah je použit elektromagnetický bzučák. Napájení zařízení je prováděno dvěma způsoby. Uživatel si může zvolit mezi 9V baterií nebo 4-12V DC trafem. Vnitřní obvod pracuje s napětím 5V DC. Do obvodu je proto zařazen stabilizátor napětí 7805V. Zařízení je dále osazeno obvodem s možností připojení k PC za pomoci rozhraní USB.

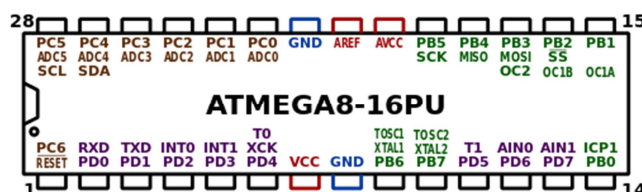
3.1 MIKROPROCESOR ATMEGA8

ATmega8 je mikroprocesor řady AVR společnosti ATMEL. Jeho 8bitový výkon není ve srovnání s novějšími mikroprocesory nijak závratný. Díky své cenové dostupnosti, jednoduchosti ovládání, vlastnostem a obsahujícími komponenty se však stal jedním z nejpoužívanějších mikroprocesorů, zejména ve školních projektech. Obsahuje tři porty: PORTB, PORTC, PORTD a dále:

- 8KB flash paměti programu
- 1KB vnitřní paměti dat SRAM
- 512B EEPROM
- 23 programovatelných I/O pinů
- 2x 8-bit čítač/časovač
- 1x 16-bit čítač/časovač
- 3x PWM (Pulse with modulation) kanály
- 6x A/D (analog to digital) převodníků
- Analogový komparátor
- Programovatelná sériová linka (USART)
- Vnitřní oscilátor 1,2,4,8 MHz
- Pracovní frekvence s externím oscilátorem do 16MHz
- Pracovní napájení 4.5-5.5V



Obr. 3.1 - Mikroprocesor ATmega8



Obr. 3.2 - Výpis všech portů mikroprocesoru ATmega8

3.1.1 ČÍTAČ/ČASOVAČ

Čítač/časovač umožňuje čítání událostí, měření a generování časových intervalů nezávisle na MCU. To znamená, že může běžet program a čítač nezávisle na sobě, aniž by jedna funkce ovlivňovala druhou. Čítač/časovač může pracovat ve dvou nastaveních:

Čítač – čítá impulzy (digitální signál) z I/O portu, např.: snímač otáček servomotoru, jízdního kola... Frekvence čítaného signálu však musí být menší než f_{osc} (frekvence oscilátoru), na kterou je mikroprocesor nastaven. V opačném případě by čítač nestíhal všechny impulzy správně zaznamenat.

Časovač – čítá impulzy vnitřního nebo externího RC oscilátoru. Z frekvence f_{osc} vyjádříme čas, který dále využíváme pro měření nebo generování signálu.

$$T = \frac{1}{f_{osc}} \quad [s] \quad (3.1)$$

T – perioda signálu

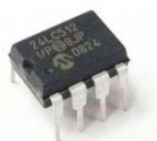
f_{osc} – frekvence použitého oscilátoru

ATmega8 má dva 8-bitové a jeden 16-bitový čítač/časovač. My využijeme 16 bitový čítač/časovač TCNT1, protože jím můžeme měřit mnohem delší časové úseky.

3.1.2 UKLÁDÁNÍ DAT

Mikroprocesor ATmega8 obsahuje 512B EEPROM paměť a 8kB flash paměť. Pro zvýšení této kapacity paměti je do obvodu přidána paměť EEPROM 24LCXXX vyráběná o velikosti od 2Kbit do 512Kbit. Zařízení je nastaveno na využívání nejvyšší paměti 512Kbit. Do ni ukládá nejen měřené hodnoty, ale i proměnné pro nastavení zařízení. Komunikace mezi touto pamětí a mikroprocesorem ATmega8 je zajištěna pomocí pinů SCL a SDA.

Celá komunikace probíhá za přispění počítačové sériové sběrnice I²C, která rozděljuje připojená zařízení na řídící, zahajuje a ukončuje komunikaci a generuje hodinový signál SCL, a na řízené, zařízení adresované masterem. To umožňuje propojení až 128 různých zařízení s pomocí dvou obousměrných vodičů. Jeden tvoří hodinový signál a druhý datový kanál. Využití této sběrnice je velice rozsáhlé. Např. pro řízení LCD a OLED displejů, změnu kontrastu, teploty barev a jejich vyváženost v monitorech, k přístupu k nízko rychlostním D/A a A/D převodníkům.



Obr. 3.3 - EEPROM 24LCXXX

3.1.3 OVLÁDÁNÍ

Aby bylo možné zařízení ovládat, obsahuje klávesnici sestavenou z mikropsínačů. Obsahuje tlačítka **NAHORU**, **DOLŮ**, **+**, **-**, **OK**, **ZPĚT**, **MĚŘENÍ** a **RESET** pro restartování zařízení. Toto ovládání umožňuje listování v menu nabídky a změnu nastavení programu zařízení, např. max. měřitelnou vzdálenost, zvukové oznámení, apod. Pro oznámení o běhu měření je tlačítko **MĚŘENÍ** vybaveno červenou LED diodou. Schéma navrhnuté klávesnice je přiloženo v seznamu příloh.

Pro zobrazení menu nabídky a probíhajícího měření, je k řídící desce připojen alfanumerický LCD displej WH1602A s dvěma řádky, každý o šestnácti znacích. DPS s displejem obsahuje odporový trimer pro nastavení kontrastu a spínač pro nastavení podsvícení. Jelikož je při zapnutí podsvícení LCD odběr proudu výrazně vyšší, stabilizátor 7805V se velmi zahřívá. Proto je k jeho chladiči přimontován 5V DC ventilátor pro snížení teploty, který se aktivuje pouze při zapnutí podsvícení LCD.



Obr. 3.4 - LCD displej WH1602A

3.1.4 ZVUKOVÁ SIGNALIZACE

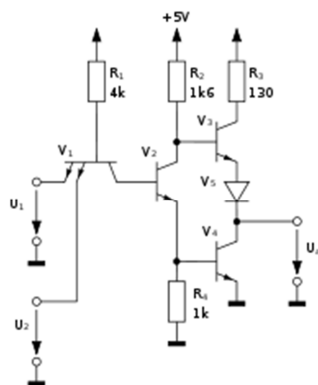
V menu nabídky programu je možné několik nastavení hodnot a podmínek pro běh programu, včetně zvukového upozornění. Pokud například bude aktuální měřená vzdálenost vyšší, než je povolená nastavením, spustí se zvukové upozornění. Jako zdroj zvuku je zvolen elektromagnetický bzučák s rezonanční frekvencí 2,4kHz. Bzučák je ovládán mikročipem, který spíná vysokofrekvenční tranzistor BC549B.



Obr. 3.5 - Elektromagnetický bzučák

3.2 KOMUNIKACE S PC PŘES ROZHRAŇÍ USB

Mikroprocesor ATmega8 umožňuje komunikaci s jiným zařízením pomocí rozhraní TTL neboli tranzistorově tranzistorová logika. Zařízení s touto technologií jsou napájena 4,5-5,5V. Komunikace v tomto rozhraní je na principu odesílání log.1 v rozmezí napětí 2-5V a log.0 v rozmezí 0-0,8V pomocí NAND hradel. Zařízení s touto komunikací může těchto hradel obsahovat hned několik.



Obr. 3.6 - Standardní TTL hradlo NAND

Je však velmi málo zařízení, které TTL podporuje. Pro PC používající komunikaci USART s rozhraním RS232 to je problém. To totiž může fungovat s jinou velikostí napětí a při přímém spojení s TTL by mohly nastat dva nežádoucí případy, kdy by mohly vypadnout vnitřní pojistky mikroprocesoru, nebo by mikroprocesor či připojené zařízení mohly vyhořet. Proto je nutné do obvodu vložit převodník.

Ne všechny počítače však v dnešní době obsahují rozhraní USART. Je jich naprosté minimum. Dnes se používá univerzální port s názvem USB. Ten byl vytvořen za účelem minimalizovat množství rozhraní, jako je například USART, LPT, PS/2, GamePort apod. Každé zařízení mělo totiž své vlastní rozhraní, což vedlo k nepřehledným zapojením a velkému množství komponentů v PC.

Rozhraní USB pracuje s napětím 5V a proudem do 500mA. Jeho hlavní výhody jsou univerzální možnost použití pro jakékoli zařízení a systém připojení Plug&Play, díky kterému je možné jej k PC připojit kdykoliv bez nutného restartování. Existuje několik typů koncovek a tři hlavní typy rozhraní USB:

- USB 1.1 – navrženo v roce 1995 s maximální přenosovou rychlostí 12Mbit/s
- USB 2.0 – navrženo v roce 2000 s maximální přenosovou rychlostí 480Mbit/s
- USB 3.0 – navrženo před rokem 2010 s maximální přenosovou rychlostí 5Gbit/s

Pro ultrazvukový měřič vzdálenosti byl vybrán typ rozhraní USB 2.0, které je v dnešní době stále nejpoužívanější a koncovka typu B, která má téměř nulovou vůli v pohybu konektorů samec samice díky své hloubce a tvaru.



Obr. 3.7 - Koncovky rozhraní USB

Celé rozhraní obsahuje pouze čtyři až pět vodičů dle typu koncovky a rozhraní. Například USART obsahuje devět vodičů a LPT (paralelní port) dokonce dvacet pět vodičů. Vodiče USB 2:0 mají tuto funkci:

1. VCC - napájení z PC 5V
2. D+ - příjem dat ze zařízení
3. D- - vysílání dat do zařízení
4. GND - uzemnění

Aby bylo možné USB rozhraní použít, je nutné opatřit jej mikročipem, který bude data z tohoto rozhraní zpracovávat. K tomu je využit převodník FT230XS-R, který je přímo navržen na převod rozhraní TTL – USB. V počítači je pak nutné pro tento mikročip nainstalovat ovladač.

3.2.1 PRINCIP SÉRIOVÉ LINKY (USART)

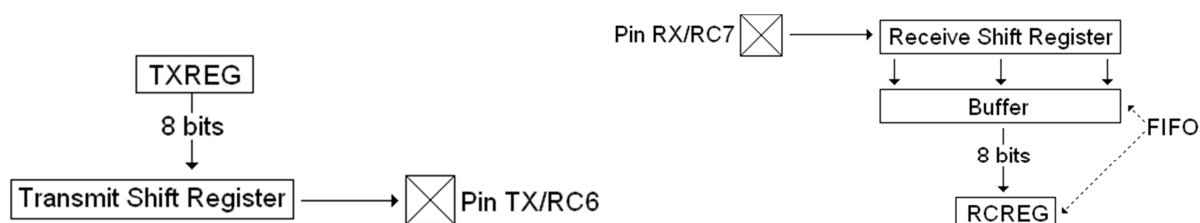
USART je rozhraní pro sériovou komunikaci. Pracuje ve dvou modech:

Asynchronní - v tomto módu je nakonfigurován jako plně duplexní asynchronní systém, který může komunikovat s periferiemi, jako jsou např. CRT terminály, osobní počítače (PC) atd..

Synchronní - Druhá možnost je nakonfigurovat USART jako “poloduplexní” synchronní systém, který může komunikovat např. s perifériemi, jako jsou A/D a D/A převodníky, sériová EEPROM atd. Tento synchronní mód můžeme nastavit jako - Master nebo Slave.

Příjem signálu - Po detekci start bitu na pinu RX, se další data bity po bitu přesunou do registru pro příjem. Po přesunutí posledního bitu se zkontroluje stop bit a data se pošlou do bufferu, který je předá registru RCREG, pokud je prázdný. Odtud se data zpracovávají v komunikačním programu.

Vysílání signálu - Jakmile jsou data zapsána do registru TXREG, všechny bity jsou přepsány do registru pro vysílání. Odtud jsou dále přeposlány na TX pin, přičemž je jim přiřazen na začátek start bit a na konec stop bit. Použití speciálního registru pro vysílání umožňuje načítání nových dat do registru TXREG už během vysílání dat předchozích. To urychluje vysílání dat.



Obr. 3.8 - Diagram pro odesílání a přijímání 8-bitů⁴

⁴ Zdroj: <https://cs.wikipedia.org/wiki/USART>

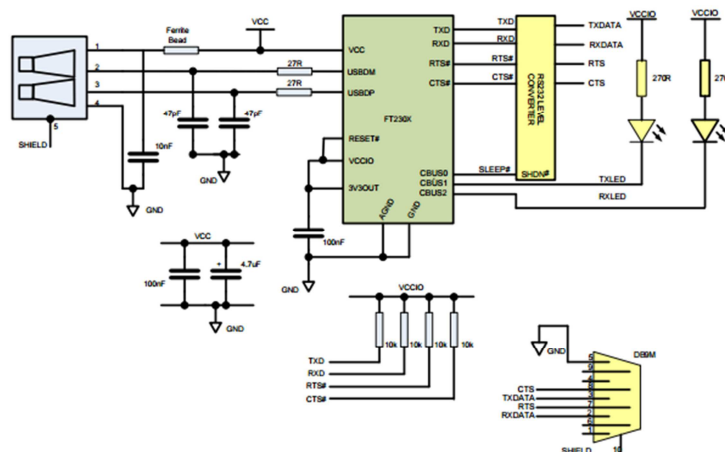
3.2.2 PŘEVODNÍK FT230XS-R

Převodník FT230XS-R má pouzdro typu SMD. Je cenově dobře dostupný, jen trochu obtížně sehnatelný. Výhodou je, že k němu výrobce dává možnost zdarma stáhnout ovladače pro všechny možné platformy.

Je napájen napětím v rozmezí 3V-5V. Jeho zapojení je velmi jednoduché. Dokonce všechny typy zapojení převodníku s mikroprocesory dodává výrobce přímo v datasheetu. Pomocí ovladače se po připojení zařízení vytvoří virtuální sériový rozhraní USART a port COM. Nastavení komunikace s převodníkem je popsáno níže v popisu programu zařízení.



Obr. 3.9 - Převodník FT230XS-R

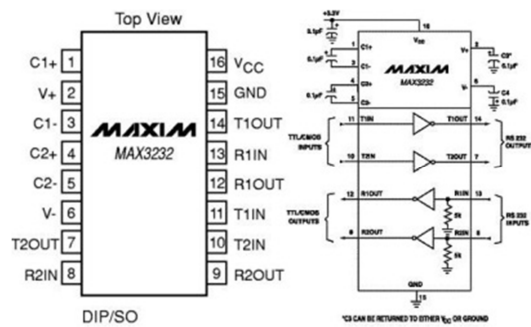


Obr. 3.10 - Schéma zapojení převodníku USB do RS232 dle výrobce⁵

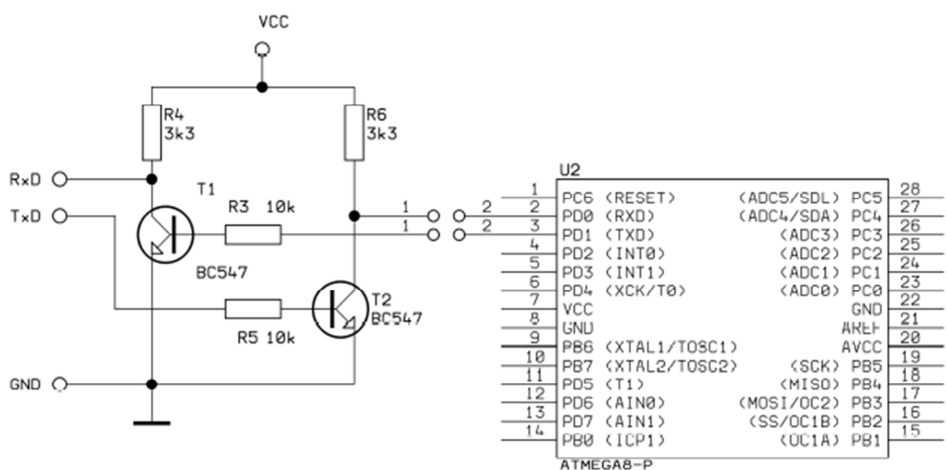
3.2.3 PŘEVODNÍK MAX232

Zařízení je navrženo tak, aby bylo možné komunikovat jak přes rozhraní USB tak i přes RS232. Převodník FT230XS-R, však převod TTL na RS232 nepodporuje a proto je možné do obvodu dodatečně vložit převodník MAX232. V zařízení je patice, do které je vložena miniaturní DPS s přemostěním na FT230XS-R. Pokud však bude potřeba, je možné otevřít kryt zařízení a toto miniaturní přemostění nahradit přiloženou miniaturní DPS obsahující MAX232 a vývody pro připojení konektoru RS232, který je do krytu zařízení instalován.

⁵ Zdroj: datasheet součástky http://www.soselectronic.cz/a_info/resource/c/FTDI/_DS_FT230X.pdf



Obr. 3.11 - MAX232 - převodník TTL do RS232



Obr. 3.12 - Náhradní schéma MAX232 s ATmega8

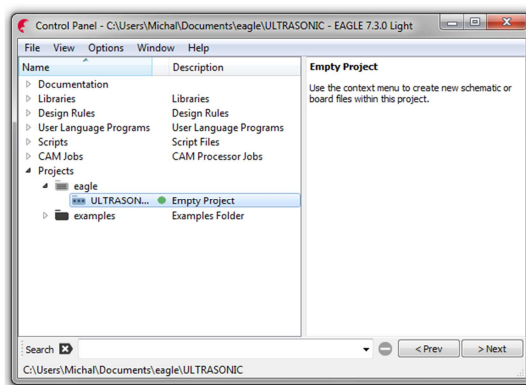
4 VÝVOJ ULTRAZVUKOVÉHO ZAŘÍZENÍ V PROSTŘEDÍ EAGLE

K vývoji elektronických schémat a návrhu DPS se používá velké množství programů. Z mých osobních zkušeností jsou nejvhodnější programy Eagle, Formica, SimuLink a Labcenter ISIS Proteus. Poslední dva jsou určeny hlavně pro simulaci elektronických obvodů, včetně simulace programování mikroprocesorů a připojení virtuálních mechanik. Dokáží také ze simulovaného elektronického obvodu vytvořit i schéma a návrh DPS. Jejich ovládání je však složitější.

Pro tuto práci jsem se rozhodl použít program Eagle. Obsahuje největší množství knihoven součástek, velký počet dalších je možné stáhnout z internetu a jeho ovládání je jednoduché a velmi přehledné.

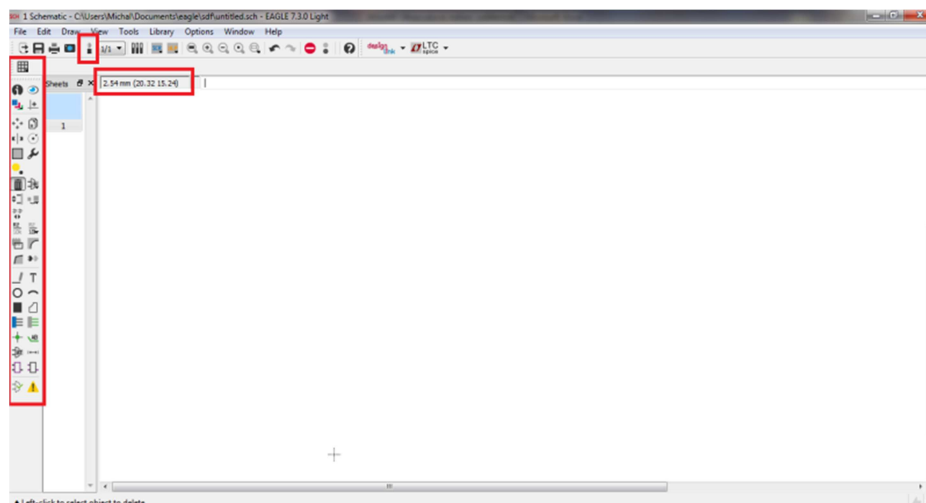
4.1 TVORBA SCHÉMAT

Spuštěním programu Eagle se nám otevře tzv. Control Panel. Okno nefunguje jako samotný editor, ale je určeno pro vytváření projektů či samostatných schémat, knihoven a návrhů DPS. Abychom při každém spuštění programu nemuseli naši práci znovu vyhledávat, je možné založit projekt, jehož odkaz je uložen přímo v editačním okně vlevo v záložce Projekts. Celá složka se nám pak uloží do adresáře >>**Dokumenty**→**eagle**. Po kliknutí pravým tlačítkem na odkaz našeho projektu je možné vkládat popisky a nové složky. Vytvoříme si tedy projekt výběrem možnosti **File**→**New**→**Projekt** a odkaz si označíme, tzn., že napravo od názvu se objeví zelený pointer. Dále přistoupíme k založení prvního schématu výběrem **File**→**New**→**Schematic**. Vytvoří se okno editoru schémat, které si uložíme.



Obr. 4.1 - Control Panel programu Eagle

Pro začátek je potřeba správně nastavit měřítko skryté mřížky, na kterou se pokládají součástky. Základní nastavení je v jednotkách inch (palce) s měřítkem 0,1 inch. Jednotky a pozice kurzoru se nám zobrazuje v levém horním rohu editorového okna. Kliknutím na ikonu **GRID** se nám otevře editorové okno měřítka. Zde ve výběru vlastnosti **SIZE** zvolíme “mm” a výběr potvrdíme. Nyní můžeme přistoupit k tvorbě návrhu našeho elektronického obvodu.



Obr. 4.2 - Editor schémat

(označeno ovládací pole, měřítko a pozice kurzoru, funkce SWITCH to BOARD)

Pod ikonou GRID se nalézá panel pro práci s editorem. Tabulku všech součástek, které obsahují knihovny programu eagle otevřeme tlačítkem **ADD**. Zde je jak kompletní záložkový seznam, tak i možnost vyhledávání součástek. **POZOR!** Pokud chceme vyhledávat součástku pomocí vyhledávače, je nutné zadat celý její název. Součástka se nám jinak buďto nevyhledá, nebo vyhledá všechny, které obsahují zadanou frázi. Pokud se zde požadovaná součástka nenalézá, nemá ji program uloženou ve vložených knihovnách. Ty je možno stáhnout z internetu, např. z oficiálních stránek výrobce. Dále je možné vytvořit vlastní knihovnu, ve které si požadovanou součástku můžeme sami vytvořit pomocí datasheetů výrobce. O druhé možnosti se budeme bavit níže.

Veškerá manipulace se součástkami se provádí buď to pomocí ovládacího panelu, nebo kliknutím pravého tlačítka myši na danou součástku což otevře interní ovládací panel samotné součástky. V jejich vlastnostech se pak mohou nastavit všechny potřebné hodnoty včetně pozice. Rotace součástky se provádí jejím označením kliknutím levého tlačítka myši a následně pravým tlačítkem je uvedena do pohybu. Změna názvu se provádí funkcí **NAME** a úprava hodnoty pomocí funkce **VALUE**. Pokud chceme upravit pozici názvu a hodnoty dané součástky musíme použít funkci **SMASH**. Součástka je se svými vlastnostmi pevně spojena a touto funkcí se navzájem oddělí. Při následujícím pohybu některé z vlastností se objeví čára spojená se středem součástky, ke které patří. To zabrání tomu, aby se nám vlastnosti pomíchaly.

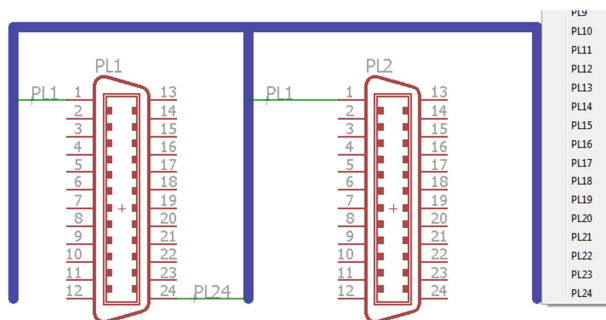
Po umístění všech součástek je na řadě jejich propojení. K tomu se využívá funkce **NET** v ovládacím panelu. Existují tři možnosti propojení tímto prvkem:

- a) pomocí přímého spojení
- b) pomocí sběrnice
- c) pomocí odkazu

Podle první možnosti se vývody součástek propojují napřímo tak, jak je potřeba. Takovéto zapojení je vhodné pro jednoduché obvody s nízkým počtem součástek. Pro ty složitější jsou určeny další dvě možnosti.

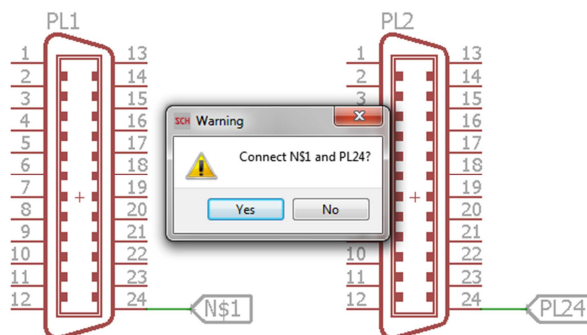
Sběrnice, funkce **BUS**, nám vytvoří propojení podobné funkci NET. Je vhodné ji nakreslit tak, aby procházeli poblíž budoucích připojených pinů. Při volbě vlastností sběrnice,

lze do pole NAME zadat jakékoli množství propojení. Na názvech nezáleží, musí být však odděleny čárkou. Pokud vytváříme sběrnici obsahující piny některého z mikroprocesorů (např. PIND0 - PIND8) můžeme si ulehčit práci zadáním hodnoty ve formátu "PIND[0..8]". Následné připojení pinů se provádí funkcí NET tak, že nejdříve připojíme čáru ke sběrnici, tím se otevře možnost výběru ze sběrnice a dále dokončíme připojení k požadovanému pinu. Funkcí **LABEL** pak můžeme k propojením ze sběrnice připojit pro přehlednost popisek.



Obr. 4.3 - Použití BUS v editoru schémat

Třetí možnost je vhodná pro extrémně složité obvody s velkým množstvím propojení. Z požadovaného pinu si vytáhneme krátké propojení funkcí NET a dále k němu vložíme funkcí LABEL jeho název. Ve vlastnostech tohoto názvu zaškrtneme políčko **XREF** a potvrdíme. Tím se nám vytvoří odkaz na toto propojení. Pro jeho dokončení provedeme stejný postup u pinů, které mají být s tímto odkazem spojeny. Zobrazí se nám dotaz, zda chceme oba odkazy spojit, který potvrdíme. Odkaz je hotov.



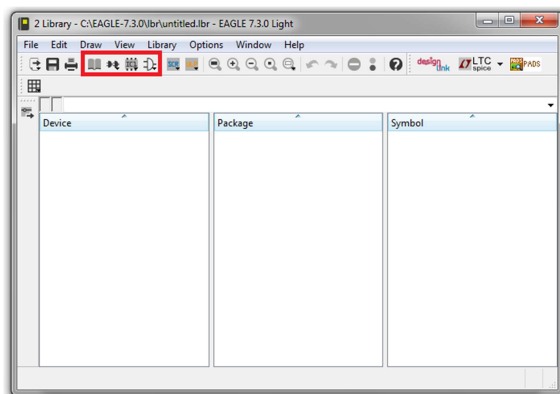
Obr. 4.4 - Použití odkazu pro propojení součástek v editoru schémat

Takto pokračujeme pro celý navrhovaný obvod. Doporučuji průběžně ukládat postup práce. Starší verze programu Eagle bývají občas při složitých projektech nestabilní a program se samovolně vypne bez uložení.

4.2 TVORBA VLASTNÍ KNIHOVNY S KOMPONENTY

Při tvorbě semestrální práce jsem narazil na problém, kdy program Eagle neobsahoval některé ze součástek, v tomto případě převodník FT230XS-R, EEPROM 24LCXXX, LCD WH1602A, jednopinový jumper a další. Proto bylo nutné přikročit k možnosti tvorby vlastní

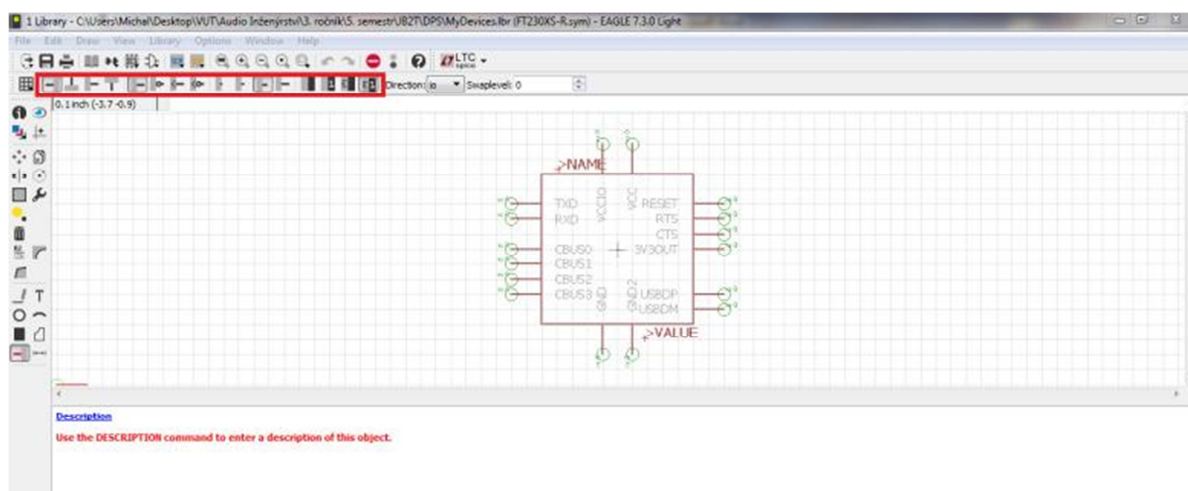
knihovny součástek. K tomu je zapotřebí mít k dispozici detailní datasheet součástky od výrobce. V Control Panelu vybereme možnost **File→New→Library**.



Obr. 4.5 - Editor knihovny součástek

4.2.1 TVORBA SYMBOLU

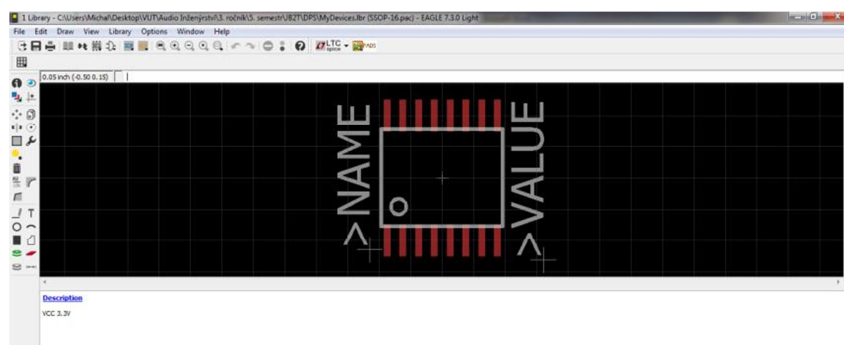
V horní části je hlavní ovládací panel editoru. Zde vybereme funkci **SYMBOL**. Otevře se dialogové okno, kde zadáme nový název symbolu součástky a potvrdíme. Spustí se rozhraní podobné editoru schémat. Opět je potřeba upravit měřítko funkcí GRID. Pomocí Funkcí **WIRE** se kreslí tvar schématické značky vytvářené součástky. Funkcí **PIN** se pak vkládají potřebné vývody, které nalezneme v datasheetu součástky vydané výrobcem. Rotace pinů se ovládá v horním ovládacím panelu nebo stejným způsobem jako v editoru schémat, stejně tak i název pinů. Dále je pak možné v tomto panelu nastavit, co se má u vloženého pinu zobrazit za text (bez názvu, název bodu na DPS, název pinu, obě zobrazení názvu). Jako poslední použijeme funkci **TEXT** pro vypsání názvu a hodnoty. Text musí být zapsat ve formátu >NAME nebo >VALUE, aby byl po vložení součástky do editoru schémat automaticky doplněn. Výsledek uložíme.



Obr. 4.6 - Editor schématické značky (FT230XS-R)

4.2.2 TVORBA POUZDRA

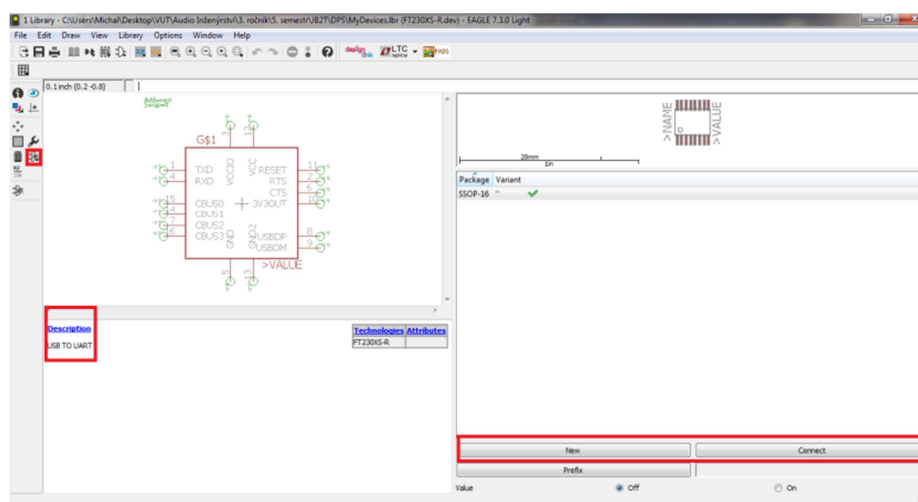
Nyní je nutné navrhnout pouzdro součástky. To je detailně popsáno se všemi rozměry v jejím datasheetu. Použijeme funkci **PACKAGE** z hlavního ovládacího panelu. Opět zadáme název a potvrdíme. Otevře se rozhraní podobné editoru návrhu DPS. Zde je postup stejný jako v editoru schématických značek. Vložení pinů je zde prováděno funkcí **PAD**. V jejich vlastnostech je možné nastavit velikost díry, velikost pole pro pájení, jeho tvar a název. Dále opět funkcí **TEXT** vložit název a hodnotu součástky, a pokud je potřeba, popisky všech pinů. V poli **DESCRIPTION** je možné vkládat poznámky, které se později zobrazí při výběru součástky v editoru schémat. Postup si znovu uložíme.



Obr. 4.7 - Editor pouzder (FT230XS-R)

4.2.3 FINÁLNÍ TVORBA SOUČÁSTKY

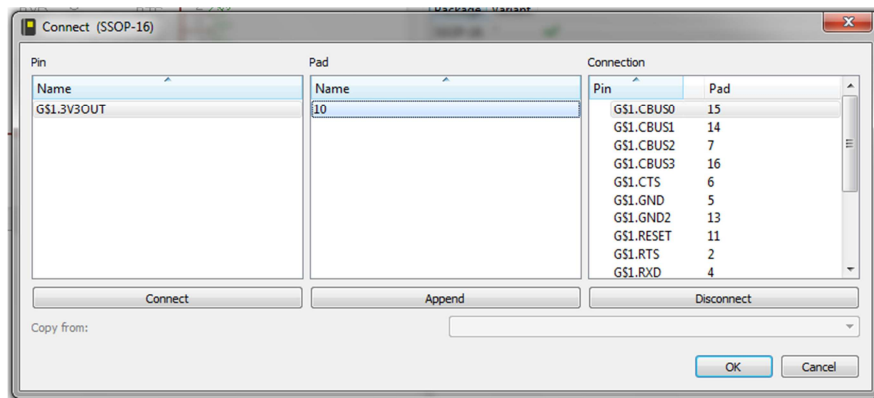
Z hlavního menu vyvoláme poslední funkci **DEVICE**. Zde zadáme konečný název součástky. Otevře se editorové okno, které je rozděleno na čtyři části: schématická značka, pouzdro, propojení pinů schématické značky a pouzdra a popisky.



Obr. 4.8 - Editor finálního vytvoření součástky (FT230XS-R)

V ovládacím panelu funkcí **ADD** vybereme požadovanou schématickou značku. Dále pomocí možnosti **NEW** v okně propojení pinů přidáme požadované pouzdro součástky.

Následně funkcí **CONNECT** otevřeme výběrové okno. Zde jsou tři tabulky. První obsahuje piny schématické značky, druhá piny pouzdra a třetí je tabulka jejich propojení. Zvolením příslušných pinů a tlačítkem **CONNECT** se k sobě přivolí a jejich propojení zobrazí ve třetí tabulce. Takto propojíme všechny piny pouzdra a schématické značky.



Obr. 4.9 - Výběrové okno pro propojení pinů (FT230XS-R)

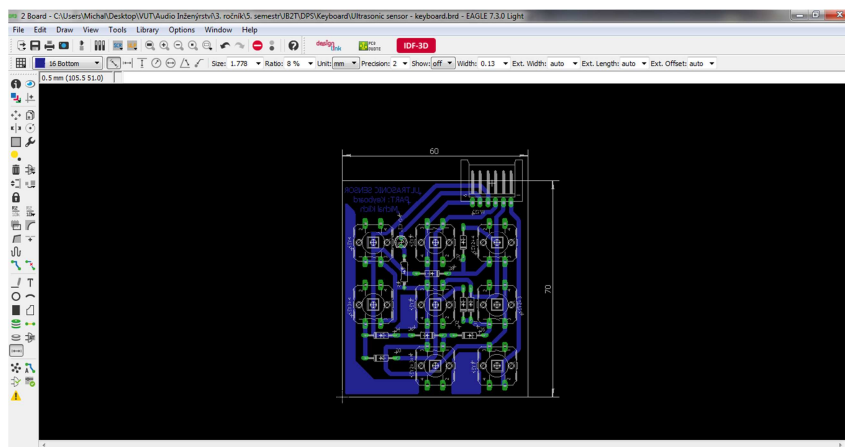
Jako poslední se do možnosti **DESCRIPTION** vloží popis součástky, který se bude zobrazovat při výběru součástky v editoru schémat.

Takto je možné vytvořit nekonečné množství součástek dle potřeby. Knihovnu je nakonec potřeba vložit do programu eagle. To se provádí v editoru schémat v možnosti **Library→Use**.

4.3 NÁVRH DESKY PLOŠNÝCH SPOJŮ (DPS)

Po vytvoření potřebné knihovny a konečné podoby obvodu je na řadě návrh DPS (Desky plošných spojů). V editoru schémat klikneme na funkci **SWITCH to BOARD**. Otevře se editor pro návrh DPS, do kterého se automaticky vloží všechny součástky obsažené v obvodu. Každá změna, která je provedena v editoru schémat se okamžitě projeví i v editoru DPS. **POZOR!** Je důležité, aby se všechny úpravy vlastností součástek prováděly v editoru schémat a navzájem se oba uložené soubory .sch a .brd neoddělovali do jiných složek. Oba soubory by pak nebyly kompatibilní, což znamená, že nebude nefungovat automatická aktualizace souboru .brd. Stejně jako v editoru schémat je potřeba nastavit měřítko neviditelné mřížky. V hlavním okně editoru se objevilo navíc pole pro umístění součástek. Velikost tohoto pole se dá měnit pomocí funkce **MOVE** stejně jako přemístění součástek v editoru schémat.

Nelze přehlédnout, že mezi součástkami jsou nataženy žluté tenké čáry. Jedná se o virtuální propojení, které se aktualizuje z editoru schémat. Tyto čáry budou viditelné, dokud se nenahradí reálným spojením. Rozmístíme součástky dle potřeby a pomocí funkce **ROUTE** začneme vytvářet propojení mezi vývody. Pokud je nakreslené propojení nutno odstranit používá se funkce **RIPUP**. Ta vymaže nakreslené propojení a nahradí jej zpět virtuálním.



Obr. 4.10 - Editor návrhu DPS
(klávesnice)

Je možné využít automatické propojování tzv. autorouting funkcí **AUTOROUTER**. Program Eagle sám navrhne kompletní propojení DPS. Pokud se jedná o složité elektronické obvody o několikavrstvé DPS, je tato funkce velmi výhodná, jelikož velmi usnadňuje a urychluje práci. Neplatí to ale vždy. Funkce **AUTOROUTER** navrhuje a vytváří DPS metodou nejbližšího spojení, tudíž se často stává, že výsledný návrh DPS i s jednoduchým elektronickým obvodem obsahuje několikavrstvou DPS s velkým množstvím propojek. Proto, pokud je to možné, doporučuji vytvářet návrh DPS bez využití této funkce. Často se ale stává, že se návrh DPS bez propojek neobejde. Ten nejjednodušší postup je v místě, kde bude umístěn pad pro propojku udělat krátkou cestu jiným směrem. Tuto cestu označit pravým tlačítkem, vybrat vlastnosti a v nastavení vrstev vlastností vybrat požadovanou vrstvu propojení DPS. Základní je **TOP** a **BOTTOM**. Krátká cesta se nám zabarví do vybrané vrstvy. Stačí ji pak natáhnout do místa, kde končí a stejným způsobem se přepnout na původní vrstvu. Tento postup má tu výhodu, že se automaticky vytvoří otvory pro vložení a pájení propojky, např. nulového odporu.

Dále je nutné nastavit správné velikosti otvorů a tvary bodů pro pájení. Každý bod má tuto možnost ve svých vlastnostech. Další možností je nastavení v možnostech **Edit→Design Rules→Restring**.

Pokud je návrh DPS hotový a zbylo místo, kudy nevede žádné propojení, je možné do těchto míst vložit text nebo je zacelit funkcí **POLYGON**. Výsledný návrh DPS je pak připraven k výrobě.

4.4 VÝROBA DPS A OSAZENÍ SOUČÁSTEK

Existují dva způsoby výroby DPS:

- osvětlování a následné leptání v chloridu železitém
- výroba pomocí miniaturizovaného CNC přístroje na DPS

První možnost se využívala dříve a používá se stále v domácích dílnách či školních laboratořích. Práce touto metodou je jednoduchá, člověk však musí být opatrný, jelikož

chlorid železitý je jedovatý a proto je doporučeno pracovat s patřičnými ochrannými pomůckami a v dobře větrané místnosti.

Na průhlednou fólii se vytiskne návrh DPS bez zobrazených součástek. V editoru návrhu DPS se toho docílí vypnutím vrstev vlastností pomocí funkce **DISPLAY** a odznačením všech čísel kromě **16, 17, 18, 39-43, 47-49, 52, 101, 102**. Tím zůstane čistý návrh DPS. Vytisknutý obvod se přiloží k předem vyřezané DPS s fotocitlivou vrstvou a vloží do osvětlovací jednotky. Doba osvětlování je podle vrstev průhledné fólie a kvality tisku od 2:30min do 5min. Standardně se osvětluje 3min. Následně se z osvětlené DPS smyje fotocitlivá vrstva a deska se vloží do směsi vody a chloridu železitého. U této výroby je však dost častá chyba leptání, kdy je chlorid železitý zředěný příliš a tak deska obsahuje části mědi, které zde nemají být. Druhou možností je příliš vysoká koncentrace a z mědi na DPS nemusí zůstat prakticky nic. Pokud se však leptání podaří, výrobek se omyje vlažnou vodou. Nakonec se ručně provrtají díry do pájecích bodů.

Druhá možnost výroby je použití miniaturizovaného CNC přístroje. Soubor .brd je nahrán do ovládacího programu, který ovládá vrtačku s frézkou. Do přístroje je vložena DPS, následně vrtačka vyfrézuje celý obvod a vyvrtá všechny potřebné díly. Nakonec vyřeže požadovaný tvar DPS, který je připraven na další postup zpracování.

Další postup je shodný s první technologií. Vyrobená DPS se ponoří do cínovací lázně, kde se na měď přichytí molekuly cínu. Pájení je pak mnohem jednodušší, hlavně pokud jde o součástky SMD, které se pájí horkovzdušnou pájkou. Nakonec se výrobek potře ochranným lakem. Takto upravená DPS je připravena k osazení součástek.

Osazování probíhá od součástek, které mají nejvyšší tepelnou odolnost, což jsou propojky, rezistory patice pro mikročipy apod. Mezi poslední pájené součástky patří kondenzátory, tranzistory, LED diody a SMD mikročipy. Jako poslední se do patic osazují požadované mikročipy.

5 VÝVOJ ŘÍDÍCIHO PROGRAMU

Pro vývoj programů mikroprocesorů je k dispozici nepřeberné množství překladačů a prostředí. Pro mikroprocesory ATMEL je podmínkou mít nainstalovaný kompilátor AVR-GCC. Ten je možné nainstalovat spolu s programem WinAVR. Jako prostředí pak může být využit jakýkoli program, který je schopný tento překladač použít. Program WinAVR obsahuje své vlastní prostředí Programmers Notepad. Je však nutné si některé důležité soubory, např. Makefile, vytvářet dodatečně, jelikož to samotné prostředí neumožňuje. Tvorba takového souboru není zvlášť jednoduchá, avšak program obsahuje manuál, ve kterém je detailně popsáno jako postupovat.

Společnost ATMEL vydává i své vlastní programy pod názvem AVR studio. Ta jsou dostupná freeware přímo na stránkách výrobce. Využívají prostředí Microsoft Visual Studia. Tyto programy vytvářejí všechny potřebné soubory automaticky a na rozdíl od Programmer Notepad nabízejí nápovědy během programování, což usnadňuje a urychluje práci.

Pro tvorbu programu vyrobeného zařízení jsem se rozhodl použít AVR studio 6. Nemá vyšší hardwarové nároky na PC jako novější verze AVR studio 7 a obsahuje vyšší kompatibilitu s mnohem více mikroprocesory společnosti ATMEL než předchůdce AVR studio 5. Dále obsahuje vlastní kompilátor AVR-GCC.

K vývoji programu byl použit jazyk C. Ten je poměrně jednoduchý, dnes velmi rozšířený a vyučuje se prakticky na všech školách včetně VUT

5.1 PROGRAM AVR STUDIO 6

Na přiloženém DVD je jak instalační soubor programu WinAVR, tak i AVR studio 6. AVR studio bude během instalace požadovat instalaci pluginu pro využití Microsoft Visual Studia 2010 Essential neboli prostředí, ve kterém bude program fungovat a dále ovladače USB JUNGLO LTD. Po instalaci programů můžeme přistoupit k založení projektu a vytváření řídicího programu.

Po spuštění AVR studia možností **FILE**→**NEW**→**PROJECT** založíme nový projekt. V levé části **Installed Templates** zvolíme možnost **C/C++**, v prostřední **GCC C Executable Project**, ve spodní části v možnosti **Name** zadáme název projektu a možností **Location** zvolíme jeho umístění. Po potvrzení se zobrazí okno s výběrem všech mikroprocesorů, které AVR studio 6 podporuje. V poli **Device Family** vybereme řadu **megaAVR, 8-bit** a zde najdeme, v zařízení použitý, mikroprocesor **ATmega8**. V pravé části okna se zobrazí informace o dané součástce. Dalším potvrzením dokončíme tvorbu projektu.

V AVR studiu je výhodou, že si během zakládání projektu program sám nastaví vše potřebné. Proto pak stačí pro kompilaci souboru pouze stisk klávesy F7.

Pokud by však přece jen nastala chyba, kdy není možné nalézt kompilátor AVR-GCC nebo překladač jazyka C make.exe, musíme nastavit jejich cestu. Možností **Tools**→**Options** spustíme okno s nastavením AVR studia a dále výběrem Toolchain otevřeme nastavení kompilátoru. Zde je nutné vybrat, pro jaký typ mikroprocesoru potřebujeme nahrát kompilátor. Tento výběr se nalézá v kolonce Toolchains jako **Atmel AVR 8-bit**. Tlačítek

AddFlavour přidáme umístění kompilátoru a překladače buď z instalační složky Atmel nebo WinAVR.

5.2 PROGRAM ULTRAZVUKOVÉHO ZAŘÍZENÍ

Před začátkem vývoje řídicího programu pro zařízení, je nejprve nutné vytvořit si scénář, případně diagram, podle kterého má celý program fungovat.

V našem případě jde o určení:

- a) typy měření dat – dle zadání má být zařízení schopno měřit absolutní a relativní vzdálenost. Dále bude možné měřit průměr až z 10 naměřených hodnot.
- b) snímání teploty – pro co nejpřesnější měření je nutné snímat okolní teplotu pomocí teplotního senzoru umístěného vedle ultrazvukového snímače
- c) ovládání – pro ovládání zařízení je potřeba jej osadit vstupním rozhraním – klávesnicí. Ta obsahuje celkem 8 tlačítek, která jsou popsána v kapitole 3.1.3.
- d) zobrazovací výstup dat – pro zobrazení toho, co právě zařízení provádí, je potřeba výstupní rozhraní. V tomto případě alfanumerický LCD displej s matrixem 2x16znaků.
- e) ukládání a zpracování dat – dle zadání, naměřená data by mělo být možné uložit do několika paměťových míst. Pro tuto funkci je zvolena externí EEPROM paměť popsaná v kapitole 3.1.2.
- f) základní nastavení zařízení – do této části se řadí nastavení zvukového upozornění, max. a min. vzdálenosti měření či styl ukládání nebo použití naměřených dat
- g) komunikace s vnějším zařízením – ještě před zahájením vývoje by mělo být rozhodnuto, zda bude výsledné zařízení schopné komunikovat s vnějším zařízením. Program ve finální fázi nemusí být pro tuto možnost jednoduché upravit a odladit.

V případě ultrazvukového zařízení bylo vytvořeno jednoduché menu, které může uživatel volně procházet a zároveň má minimální nároky na paměť programu. Položky menu jsou:

- 1) 1.Absolutní měření – zařízení navrací měřenou vzdálenost v absolutní hodnotě
- 2) 2.Relativní měření – zařízení navrací měřenou vzdálenost v relativní hodnotě, dříve vybranou uživatelem, kterou je možné vybrat z minulých naměřených hodnot. Tuto hodnotu je poté možné dále upravit dle potřeby.
- 3) 3.Průměr – zařízení navrací určité množství měření zvolené uživatelem v nastavení. Z těchto hodnot dále vypočítá průměr měření.
- 4) 4.Teplota – zařízení navrací hodnotu okolní teploty. Tato funkce je pouze informativní. Hodnoty nelze ukládat.
- 5) 5.Nastavení – podnabídka, která umožňuje měnit nastavení zařízení. Její položky jsou:
 - a) 1.Ref.Od. – výběr hodnoty pro referenční měření
 - b) 2.Max – nastavení maximální měřitelné vzdálenosti
 - c) 3.Min – nastavení minimální měřitelné vzdálenosti
 - d) 4.Průměr – nastavení počtu měření pro výpočet průměru
 - e) 5.Bzučák – nastavení zvukového upozornění ON/OFF

- f) Vymazat paměť – celá paměť externí EEPROM bude vymazána
- 6) 6.FREE MOD – zařízení navrací měřenou hodnotu neustále dokola. Tato funkce je pouze informativní. Hodnoty nelze ukládat.

Všechny hodnoty naměřené v 1-4 jsou po stisku tlačítka BACK, navrácení zpět do hlavního menu, automaticky uloženy do externí paměti viz kapitola 5.2.4. Do externí paměti jsou dále ukládány všechny proměnné z nastavení zařízení. Ty jsou při spuštění zařízení načteny a tak je možné s nimi dále pracovat.

V úvodu programování je nejprve nutné načíst potřebné knihovny. Pro mikroprocesory AVR je to základní knihovna io.h. Dále je nutné nastavit frekvenci pracovního oscilátoru, na které má mikroprocesor pracovat. Poté se definují hodnoty, které budou použity jako matrix, například potvrzení TRUE či zamítnutí FALSE. V hlavní funkci se poté inicializují všechny části zařízení a spustí se hlavní smyčka programu.

```
//Ukázka úvodu programu
#include <avr/io.h>           //Vložení základní knihovny
#define F_CPU 8000000UL      //Nastavení frekvence oscilátoru na 8MHz
#define TRUE 1               //Definice potvrzení
#define FALSE 0              //Definice zamítnutí

void main(void){              //Hlavní funkce programu
    LCD_INIT();                //Inicializace LCD displeje
    KEY_INIT();                //Inicializace klávesnice
    EEOpen();                  //Inicializace externí EEPROM
    //pokračování kódu
    while(1){                  //Hlavní nekonečná smyčka programu
        //program
    }
}
```

5.2.1 KOMUNIKACE S LCD

Pro komunikaci s LCD displeji je možné využít několika desítek knihoven, které jsou volně ke stažení na internetu. Jedná se o knihovny oficiálně vydané výrobcí až po knihovny vytvořené nadšenci. Složitost knihoven je neomezená, záleží pouze na tom, co vše je po displeji požadováno. Po vložení takovéto knihovny do projektu však musíme počítat s tím, že nám budoucí velikost souboru a tudíž i celková velikost využitě paměti mikroprocesoru prudce vzroste. Je to dáno tím, že zkomprimovaný soubor obsahuje všechna data knihovny, ať jsou pro běh programu relevantní či ne. V takovém případě je lepší variantou navrhnout vlastní knihovnu, která obsahuje pouze a jen ty funkce, které jsou potřebné pro běh programu. Odlazený program podle potřeby a zbavení se všech nepotřebných funkcí může ve výsledku znamenat snížení podílu velikosti knihovny, v celkové velikosti programu, z desítek na jednotky procent.

V první řadě je potřeba seznámit se s vlastnostmi komunikace mezi mikroprocesorem a LCD displejem. Alfamerické displeje většinou obsahují 16 vstupních pinů. Tak je tomu i

u použité displeje WH1602A. Popis těchto pinů, název a pořadí jejich umístění, je detailně popsán v každém datasheetu displeje. Následuje stručný popis použitých pinů v zařízení:

- a) K, A – piny katoda a anoda sloužící pro podsvícení displeje
- b) VSS, VCC(VDD) – hlavní napájení LCD
- c) V0 – Tento pin nastavuje kontrast displeje pomocí trimru. Tato možnost je vhodná při zapnutí podsvícení displeje. Může však být přímo spojen se zemí.
- d) RS – 1. řídicí pin
V případě odeslání log.0 na pin, je displeji přikázáno přijímat příkazy, jako je např. vymazání displeje, určení řádku či sloupce apod.
V případě log.1 je displej nucen přijatá data zpracovávat jako řetězec – text
- e) RW – 2. řídicí pin
Tento pin nastavuje, zda jsou data na LCD odesílána (log.0) nebo z něj čtena (log.1).
- f) E – 3. řídicí pin
Tento pin udává, kdy má displej začít zpracovávat data. Ta se totiž nejprve odešlou na datové piny v hodnotách log.1 a 0. Poté odesláním log.1 na pin E je povoleno jejich zpracování. Doba k tomu potřebná, je uvedena v datasheetu součástky. Pro většinu příkazů či dat se pohybuje kolem 40μs. Nakonec je zpracování ukončeno log.0 a na datové piny je tak možné odeslat další sadu dat.
- g) DB0-DB3 – datové piny sloužící pro 8 bitovou komunikaci s piny DB4-DB7
- h) DB4-DB7 – datové piny sloužící pro 4 bitovou komunikaci, v případě nepoužití pinů DB0-DB3. Tato možnost zjednodušuje zapojení obvodu a jeho přehlednost čehož je využito v této práci.

Nejprve je nutné nastavit všechny datové a řídicí piny mikroprocesoru podle jejich využití. V tomto případě se u všech jedná o výstup. Dále je potřeba vytvořit funkce, které budou data odesílat na LCD a přerozdělovat je na určité piny. U 4-bitové komunikace je nevýhodou, že všechna data je nutno odeslat dvakrát a to jako náhradu za nepoužité spodní piny DB1-DB4. Data jsou odesílána nejdříve ve formátu horní hodnoty a poté spodní hodnoty příkazu. Formát odeslání je popsán v datasheetu součástku. Pomocí ovládacích pinů E a RS je pak určeno, kdy a jakým způsobem se mají odeslaná data zpracovat, viz stručný popis pinů.

Díky těmto funkcím je možné přistoupit k inicializaci LCD displeje. Během inicializace je displeji určeno o jaký typ komunikace se jedná, jakou znakovou řadu má použít, dále nastavení kurzoru, vymazání displeje a nastavení výchozí pozice znaků.

Další funkce jsou pak odeslání řetězce a čísel. Jelikož displej neumí zpracovávat samostatná čísla, je nutné je převést nejprve na řetězec. Ten se poté stejně jako text posílá znak po znaku na displej a postupně vykresluje.

Ukázka komunikace s LCD:

```
//Nastavení portu použitého pro odesílání data
#define PORT_DATA    PORTB
#define PORT_CMD_R   PORTD //Nastavení portu pro řídicí piny RS a RW
#define RS           PD6    //Nastavení pinu pro řízení pinu RS
#define DB4          PB1    //Nastavení pinu pro datovou komunikaci
```

```

void LCD_send(char cmd){          //Funkce pro odeslání dat na LCD
//Vymazání a dále přerozdělení dat na datové piny
    PORTD_DATA &=~((1<<DB4)|(1<<DB5)|(1<<DB6)|(1<<DB7));
    PORT_DATA |= (((cmd>>0x00)&0x01) << DB5);
    PORT_DATA |= (((cmd>>0x01)&0x01) << DB6);
    PORT_DATA |= (((cmd>>0x02)&0x01) << DB7);
    PORT_DATA |= (((cmd>>0x03)&0x01) << DB7);

//Log.0 na řídících pinech pro nastavení zpracování dat na LCD jako
příkaz a zápis na LCD
//Log.1 na RS pro zpracování řetězce
    PORT_CMD_R &=~(1<<RS)|(1<<RW);
    PORT_CMD_E |= 1<<E;          //Povolení zpracování
    _delay_ms(2);                //Nutno počkat určitou dobu dle datasheetu
    PORT_CMD_E & =~(1<<E);       //Ukončení zpracování
}

void LCD_cmd(char cmd){          //Funkce přerozdělující bitové části dat
//Odeslání vyšší bitové části dat
    LCD_send((cmd >> 0x04) & 0x0F);
    LCD_send(cmd & 0x0F);       //Odeslání nižší bitové části dat
}

```

Ukázka konverze čísla na řetězec a jeho odeslání na LCD:

```

char buffer[10];                //Buffer pro uložení řetězce
int amount=0;                   //Proměnná pro uložení počtu cifer v řetězci
while(cislo){                    //Cyklus procházející všechny cifry

//Uložení zbytku čísla do bufferu po dělení 10
    buffer[amount] = cislo%10;
    cislo = cislo/10;           //Dělení čísla
    amount++;                    //Zvýšení počtu cifer
}

//Projde buffer pozpátku od poslední uložené cifry
for(int i=amount-1; i>=0; i--)    LCD_cmd(buffer[i]);
    //Odešle znak na zpracování

```

5.2.2 KOMUNIKACE S KLÁVESNICÍ

Pro ovládání zařízení je potřeba použití vstupního rozhraní. To je řešeno mikrospínači na samostatné DPS, které jsou připojeny konektorem CON4 k pinů PC0-PC2 mikroprocesoru.

Ty jsou nastaveny při spuštění zařízení na log.1. Odesíláním kombinací log.0 (GND) z klávesnice na tyto piny, je v programu detekováno, která klávesa je právě stisknuta a poté je provedena požadovaná funkce.

Ukázka definice klávesnice:

```
//Pin mikroprocesoru připojený na pin 4 konektoru klávesnice
#define K_SW4 PC0
//Hodnoty spínače součtem kombinací výstupních pinů 2-4 klávesnice
#define K_PLUS 7
#define K_CLEAR 0
```

Následně se funkcí KEY(int key_value) zjistí, zda je stisknut daný spínač součtem všech přijatých log.0 na pinech PC0 (hodnota 4), PC1 (hod. 3), PC2(hod. 2).

Ukázka zjištění stisku klávesy:

```
int key_buffer = 0; //Proměnná pro uložení hodnoty kombinace
//Pokud je na PC0 log.0, přičte hodnotu ke kombinaci
if(bit_is_clear(PINC, K_SW4)) key_buffer+=4;
if(bit_is_clear(PINC, K_SW3)) key_buffer+=3;
if(bit_is_clear(PINC, K_SW2)) key_buffer+=2;
//Pokud je hodnota požadované shodná s výslednou kombinací, navrátí
//potvrzení
if(key_value == key_buffer) return TRUE;
return FALSE; //Pokud není shoda, zamítne ji
```

Pro správnou funkci klávesnice je potřeba ještě jedna velmi krátká, ale podstatná část kódu. Tou je samotná kontrola použití klávesnice. Ta zabraňuje například opakovanému proběhnutí funkce či nemožnost stisku jiného tlačítka, během uživatelem stisknuté klávesy. Stačí k tomu jedna proměnná, která se vřadí do všech podmínek volajících klávesnici. Nakonec je už jen potřeba kontrolovat, kdy a zda je klávesnice opět nepoužita.

Ukázka kontroly použití klávesnice:

```
int key_press = FALSE; //klávesnice je nepoužívaná
//Pokud není klávesnice používána a je stisknuto UP
if(!key_press && KEY(K_UP)){
    //část funkce
    key_press = TRUE; //Potvrdí použití klávesnice
}
//Pokud bylo potvrzeno použití klávesnice a nyní je nepoužívána
if(key_press && KEY(K_CLEAR)){
    //Potvrzení použití klávesnice je negováno
    key_press = FALSE;
}
```

5.2.3 ZPRACOVÁNÍ DAT ULTRAZVUKOVÉHO A TEPLOTNÍHO SENZORU

Prioritní prací programu je měření vzdálenosti. K tomu je využit ultrazvukový senzor SRF05. Práce s odesláním a příjmem dat není nikterak složitá. Jako první se resetují a spustí vnitřní hodiny mikroprocesoru. Těmi se měří délka přijatého signálu. Dále je pak potřeba nastavit, kterými piny se bude se senzorem komunikovat. Senzor SRF05 umožňuje dva typy komunikace, jak je výše popsáno: po jednom či dvou vodičích. V tomto případě je komunikace nastavena na druhou možnost. Příjem signálu ze senzoru je nastaven na pin PD4 a vysílání signálu na senzor na pin PD3 mikroprocesoru.

Pro správné odeslání signálu z ultrazvukového senzoru je potřeba vybudit vysílač signálem log.1 o délce minimálně 10 μ s. Doporučeno je však alespoň 20 μ s. Vysílací pin se pak přepne zpět na log.0. V tuto chvíli mikroprocesor čeká, kdy ze senzoru začne putovat přijatý signál. Pokud senzor do 30ms neobdrží žádný odražený signál, překážka není detekována. Proto odešle zpět na mikroprocesor velmi krátký impuls. Tím oznámí, že překážka nebyla detekována. Jeho délka je pouze několik mikrosekund, což je dostatečně dlouhá doba, aby byl čekací cyklus v mikroprocesoru ukončen. Jelikož je zpětný impuls tak krátký, mikroprocesor není schopen změřit jeho délku a tak nemůže vypočítat naměřenou vzdálenost. Proto odešle na senzor nový budící signál. Před tímto úkonem je však nutné vyčkat alespoň 50ms. Senzor by totiž mohl během nového měření zachytit zbloudilé signály a vzdálenost špatně detekovat.

Ukázka ultrazvukového senzoru:

```
#define RECEIVER      PD4    //Přijímací pin
#define TRANSMITTER  PD3    //Vysílací pin

TCCR1A = 0x00;           //Reset vnitřních hodin mikroprocesoru
TCCR1B = 1<<CS11;       //Spuštění vnitřních hodin

PORTD |= 1<<TRANSMITTER; //Odeslání budícího signálu
_delay_us(20);           //O délce 20 $\mu$ s
PORTD |= 1<<TRANSMITTER; //Konec budícího signálu

//Čekací smyčka na impuls či přijatý signál ze senzoru
while(!(PIND, 1<<RECEIVER));
//Vynulování vnitřních hodin a start měření délky přijatého signálu
TCNT1 = 0;
//Cyklus probíhající, dokud bude mikroprocesor přijímat signál ze
//senzoru
while(PIND, 1<<RECEIVER);
//Uložení délky přijatého signálu do proměnné
volatile unsigned int time = TCNT1;
```

Tímto je nastavena komunikace s ultrazvukovým senzorem. Dále je pro co nejpresnější měření potřeba zjistit okolní teplotu vzduchu. Teplotní čidlo TMP36GT9Z má

analogový výstup, který je připojen na pin PC3 mikroprocesoru. Ten obsahuje jeden z pěti analogo-digitálních převodníků, ADC3. Ten porovnává zadané napětí v registru ADMUX s přijatým. Registr ADMUX je možné nastavit za pomoci dvou registrů REFS0 a REFS1 na:

REFS0	REFS1	Zvolené napětí
0	0	Referenční napětí AREF
0	1	Napájecí napětí AVCC
1	0	Rezervováno
1	1	Vnitřní napětí ATmega8 o 2,56V

Tab. 5.1 - Zvolení porovnávacího napětí pro ADC převodník

Jelikož je teplotní senzor připojen na napájecí napětí 5V, zvolíme druhou možnost AVCC. Dále je potřeba nastavit pomocí registru ADCSRA hodnotu pracovní frekvence ADC převodníku. Ta musí být pro naše účely mezi 50-200kHz. Pomocí registrů ADPS0-2 se nastavuje dělicí poměr pro výpočet této frekvence. Hodnota se dosadí do vzorce:

$$f_{\text{ADC}} = \frac{f_{\text{OSC}}}{\text{dělicí poměr}} = \frac{8\text{MHz}}{128} = \mathbf{62.5} \quad [\text{kHz}] \quad (5.1)$$

f_{ADC} – pracovní frekvence ADC převodníku

f_{OSC} – frekvence pracovního oscilátoru

Výsledná hodnota 62,5kHz je uvnitř námi požadovaného rozsahu.

ADPS2	ADPS1	ADPS0	Dělicí poměr
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Tab. 5.2 - Dělicí poměr pro výpočet pracovní frekvence ADC převodníku

Dále je potřeba nastavit, který kanál ADC převodníku chceme použít. Registr ADMUX poté nastavíme na porovnávání s tímto kanálem. Nakonec je potřeba zahájit převod z analogové na digitální hodnotu. Výsledná hodnota ovšem není ve stupních celsia. Senzor navrácí digitální hodnotu. Proto se musí celková teplota z této hodnoty vypočítat. Zjištěná hodnota na ADC převodníku se vynásobí podílem napájecího napětí AVCC a digitální hodnoty při tomto napětí, která je 1024. Tímto vztahem se z digitální hodnoty vypočte výstupní hodnota senzoru v milivoltech. Nyní je možné vypočíst teplotu podělením výstupního napětí senzoru poměrem napětí vůči teplotě udané v datasheetu, který je 10mV/°C. Jelikož je napětí posunuto o 500mV kvůli záporným hodnotám, výsledná teplota bude o 50°C vyšší. Proto je nutné tuto hodnotu odečíst. Výsledná hodnota poté pomocí funkce dosadí do vzorce pro výpočet vzdálenosti požadovanou rychlost zvuku.

$$U_{TMP36} = U_{ADC} \frac{AVCC}{ADC_{5V}} \quad [\text{mV}] \quad (5.2)$$

$$t = \frac{U_{TMP36}}{\text{poměr}} - 50 \quad [^{\circ}\text{C}] \quad (5.3)$$

U_{TMP36} – výstupní napětí na teplotním senzoru

U_{ADC} – napětí odeslané z teplotního čidla a převedeno na digitální podobu

$AVCC$ – napájecí napětí

ADC_{5V} – digitální hodnota 5V na $ADC = 1024$

T – výsledná teplota

Ukázka teplotního čidla:

```
ADMUX |= 1<<RESF1;           //Nastavení porovnávacího napětí
//Spuštění převodníku a nastavení dělicího poměru pro pracovní
//frekvenci
ADCSRA |= (1<<ADEN) | (1<<ADSP2) | (1<<ADSP1) | (1<<ADSP0);

//Nastavení, který kanál ADC převodník je použit
uint8_t channel = 3&0x07;
ADMUX |= channel;             //Nastavení porovnávání s tímto kanálem
ADCSRA |= (1<<ADSC);           //Zahájení převodu
while(!(ADCSRA & (1<<ADIF)));
ADCSRA |= (1<<ADIF);
return ADCW;                   //Navrácení naměřené hodnoty
```

Poslední součástí měření zvukové oznámení. Pokud je hodnota vyšší či nižší než je nastaveno uživatelem, mikroprocesor vybudí pin PD5 připojený na elektromagnetický bzučák.

Ukázka zvukového upozornění:

```
#define SOUND PC3              //Nastavení zvukového pinu
for(int i=0; i<1000; i++){     //Nastavení počtu period
    PORTD ^= 1<<SOUND;         //Přepínání mezi log.1 a log.0
    _delay_us(500);            //Nastavení délky periody
}
```

5.2.4 UKLÁDÁNÍ DAT

Komunikace a práce s externí EEPROM pamětí je o něco složitější. Správné nastavení všech potřebných registrů je celkem složité, zdlouhavé a velmi snadno vznikají chyby, které mohou v nejhorším vést i ke zničení externí paměti. Proto doporučuji použít knihovnu, která má všechny potřebné náležitosti již přednastaveny. V tomto případě jde o knihovnu 24c64.h, která je určena přesně pro vybraný typ EEPROM paměti. U komunikace jde v principu o

nastavení časového signálu, který určuje rychlost přenosu dat. Dále je zapotřebí vybrat správně adresu zařízení. EEPROM paměti obsahují tři piny A0-2, které tuto adresu určují. Na ně se nastavuje dle potřeby log.0 nebo 1. V tomto případě jsou všechny nastaveny na log.0. Toto adresování má tu výhodu, že je možné použít několik pamětí za sebou pouze na dvou vodičích, aniž bychom museli počet vodičů zvyšovat. Adresa uložená v odeslaných datech pak prochází všemi zařízeními, avšak pouze zařízení s požadovanou adresou data přijme a zpracuje.

Další problematikou komunikace je schopnost pracovat pouze v 8-bitovém rozhraní. To umožňuje ukládat hodnoty pouze o velikost 0-255. Důvodem je samotný mikroprocesor a paměť EEPROM. Součástky jsou na toto rozhraní nastaveny již od výroby. Proto je nutné data s vyšší hodnotou přeložit stejně jako je tomu u LCD displeje, kde se využívá 4-bitové komunikace.

Data se na paměť ukládají postupně za sebou a jejich zobrazení je možné v menu nastavení ve výběru referenční hodnoty. Dále je možné jejich smazání tak, že se resetuje počítačový cyklus a data se následně přepíše.

Ukázka práce s externí pamětí EEPROM:

```
#include "24c64.h"          //Vložení knihovny pro ovládání EEPROM

//Funkce pro čtení uložených hodnot a jejich převedení z 8-bitových
//na 16-bitové
int EERead(unsigned int address){
    unsigned int lowB = EEReadByte(address);
    _delay_ms(50);          //Čekací inicializační smyčka čtení
    unsigned int highB = EEReadByte(address);
    _delay_ms(50);
    //Převod hodnoty ze 16-bitové na 8-bitovou a její vrácení
    return ((highB<<8)+lowB);
}

//Funkce pro uložení hodnot do EEPROM a převod z 16-bitových na 8-
//bitové
void EEWrite(unsigned int address, int number){
    //Uložení spodní bitové části
    unsigned int lowB = (number & 0xFF);
    unsigned int highB = (number>>8);
    //Převod horní části na 8-bitovou
    EEWriteByte(address, lowB);
    _delay_ms(50);          //Čekací inicializační smyčka zápisu
    EEWriteByte(address+1, highB);
    _delay_ms(50);
}
```

6 OŽIVENÍ A TESTOVÁNÍ

Zkompilovaný soubor .hex je nyní potřeba uložit do použitého mikroprocesoru. Pomocí výše popsané komunikace, to však v tuto chvíli není možné, jelikož mikroprocesor tento typ komunikace nezná. Všechny mikroprocesory jsou totiž od výroby nastaveny na komunikaci pomocí sběrnice SPI. Jedná se o stejný typ externí sběrnice jako I²C. Umožňuje však mnohem vyšší přenosovou rychlost, díky vyšší frekvenci hodinového signálu. Ta může být až 70MHz, pro I2C platí v případě nejrychlejšího módu 3,4MHz.

Pro nastavení potřebné komunikace je nutné pomocí programátoru mikroprocesorů AVR USBasp nahrát do mikroprocesoru tzv. bootloader neboli zavaděč. Ten vždy po spuštění zařízení oznámí mikroprocesoru, jakou komunikace má primárně použít. Jako nahrávací program v PC se používá program ExtremeBurner.

Programátor USBasp je cenově dostupný a lehce sehnatelný. Je ale možné si jej vyrobit tzv. podomácku. Na internetu je velké množství schémat a návodů jak při jeho tvorbě postupovat. Problém je však ten, že systém velmi často tyto domácí zařízení nerozpozná a není schopen nainstalovat požadovaný ovladač kvůli chybné komunikaci se zařízením. Další velkou nevýhodou, těchto podomácku vyrobených zařízení, je často nemožnost nastavit pojistky mikroprocesoru, např. pro externí oscilátor či v našem případě pro komunikaci v rozhraní USART.



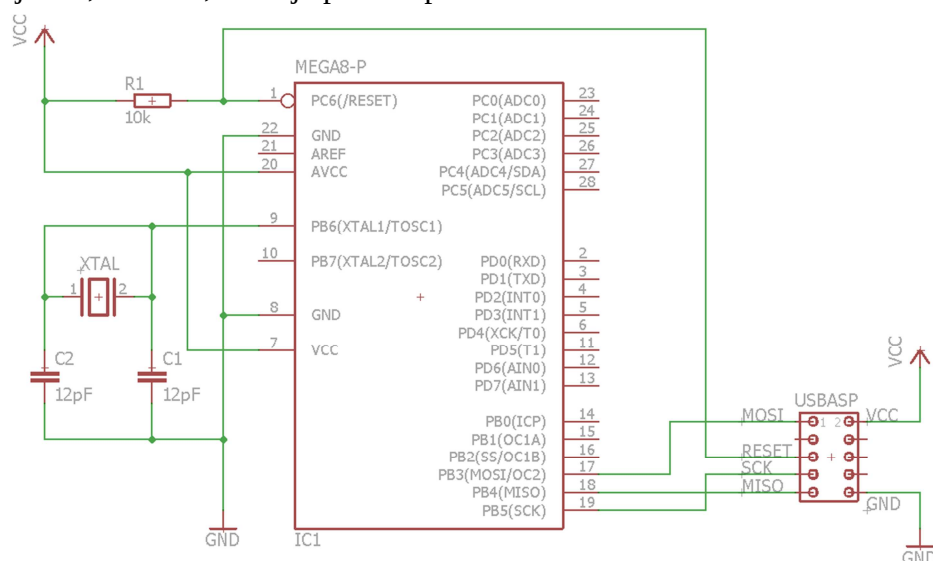
Obr. 6.1 - Programátor USBasp

6.1 BOOTLOADER

Po připojení programátoru USBasp je nutné nainstalovat ovladače zařízení. Velmi často se stává, že systém není schopen ovladač sám najít. Proto je nutné jej instalovat manuálně. Zařízení je možné najít v **Ovládací panely**→**Správce zařízení**. V možnostech **Vlastnosti**→**Ovladač** stiskneme tlačítko **Aktualizovat ovladač** a z následujících možností vybereme **Vyhledat ovladač v počítači**. PC se nás dále zeptá na umístění ovladače a sám jej poté nainstaluje.

Nyní je potřeba připojit náš mikroprocesor k programátoru. Celé zapojení je možné provést na nepájivém kontaktním poli. Schéma zapojení je velmi prosté. Je nutné nezapomenout k programovanému mikroprocesoru připojit externí oscilátor. Mikroprocesor má od výroby nastaveno používat vnitřní oscilátor o frekvenci 1MHz. Nahrávaný program však může požadovat mnohem vyšší takt a mikroprocesor se tak může zablokovat nebo vyhořet. Proto je nutné použít externí oscilátor, který nám pomůže s přetaktováním

mikroprocesoru. Napájení programované součástky je možné přímo z programátoru. Ten nabízí napájení 3,3V a 5V, které je pro nás podstatné.



Obr. 6.2 - Zapojení ATmega8 s USBasp

Po zapojení obvodu je možné přikročit k vypálení bootloaderu na mikroprocesor. V programu ExtremeBurner v možnosti **Chip** zvolíme požadovaný mikroprocesor ATmega8. Pro kontrolu, zda je vše v pořádku zapojeno, stiskneme v záložce **Flash** možnost **Read All**. Program sám naváže spojení a přečte všechna relevantní data z mikroprocesoru.

Nyní můžeme v záložce Flash vložit .hex soubor s bootloaderem. V programu se vypíše jeho kód, který se ukládá až téměř na konec paměti mikroprocesoru. Vypálení se provádí pomocí možnosti **Write→Flash**. Soubor je tak uložen ve flash paměti. Stále však není možné použít požadovanou komunikaci. Je potřeba ještě nastavit pojistky mikroprocesoru. V záložce **Fuses** označíme možnost **Write** pro kolonky **LOW FUSE** a **HIGH FUSE**. Dále nastavením jejich hodnot:

- LOW FUSE – 0xE4
- HIGH FUSE – 0xDA

a tlačítkem Write zapíšeme jejich hodnoty do mikroprocesoru. Tímto je plně připraven na komunikaci v rozhraní USART.

6.2 ULOŽENÍ ŘÍDÍČÍHO PROGRAMU

Jelikož program ExtremeBurner přepisuje celou paměť flash, není jím možné do mikročipu nahrát jak bootloader tak i řídicí program. K tomu se využívá program AvrISP II. Ten komunikuje se zařízením v rozhraní USART. Zařízení osazené připraveným mikročipem připojíme k baterii či DC trafozdroji a zapneme jej. Rozsvítí se zelená dioda označující zapnuté zařízení a dále bílá dioda určující snahu mikroprocesoru komunikovat s vnějším zařízením. Dále ultrazvukové zařízení propojíme kabelem s USB portem PC, které automaticky naistaluje ovladač převodníku FT230XS-R. Pokud se tak nestane, je nutné ovladač instalovat stejným způsobem, jako tomu bylo u programátoru USBasp. Po jeho

instalaci se v PC vytvoří virtuální port COM. Jeho hodnotu lze nalézt ve Správce zařízení v záložce **Porty (COM a LPT)**.

V programu AvrOSP je nutné v záložce **Configure** nastavit požadovaný port COM, dále přenosovou rychlost **Baud** na 9600 a **Protocol** na AVR911. Detekce zařízení se v programu ovládá pomocí funkce **AutoDetect** v záložce **Program**. Program AvrOSP bohužel, na rozdíl od programu ExtremeBurner a programátoru USBasp, postrádá jednu z primárních funkcí a tou je automatický reset připojeného zařízení. Zařízení je tak nutno před jeho detekcí resetovat tlačítkem reset na jeho klávesnici. Tlačítko je nutné stisknout těsně před spuštěním detekce. Mikroprocesor má krátkou prodlevu mezi resetem a spuštěním nahraného programu, kdy čeká, zda přijme na pin RXD start bit odeslaný z vnějšího zařízení. Pokud se tak stane, spuštění nahraného programu je zamítnuto a je zahájena komunikace. Prodleva je velmi krátká a z tohoto důvodu může chvíli trvat, než se uživateli podaří připojené zařízení detekovat. Úspěšné zahájení komunikace se vypíše v dialogovém okně ve spodní části programu. **POZOR!** Pokud je při práci s přenosem dat program určitou dobu v klidu a není na něm prováděna žádná práce, spojení je ukončeno a je nutné zařízení znovu detekovat. Tato doba je většinou několik minut.

Program AvrOSP má na rozdíl od ostatních programů jednu bonusovou funkci. V záložce **FUSE BITS** jsou po připojení zařízení zobrazeny všechny vlastnosti, kterých se dá pomocí nastavení pojistek dosáhnout. Program pomocí výběru sám nadefinuje jejich hodnoty, které je možné přímo uložit do mikroprocesoru.

Nyní zbývá uložit do zařízení jeho řídicí program. V záložce Program, v části FLASH, nahrajeme zkompileovaný .hex soubor řídicího programu pomocí tlačítka Browse. Nakonec celý program vypálíme do paměti mikroprocesoru tlačítkem Program. Ve spodním dialogovém okně se zobrazí postup nahrávání. Program AvrOSP není v ukládání dat tak rychlý jako program ExtremeBurner a proto může ukládání probíhat i několik minut včetně kontroly zapsaných dat.

Po dokončení nahrávání a restartu je zařízení plně připraveno k použití.

6.3 TESTOVÁNÍ ULTRAZVUKOVÉHO ZAŘÍZENÍ

Testování zařízení probíhalo v uzavřené místnosti a v otevřeném prostoru. Následující tabulky znázorňují skutečnou vzdálenost objektu, naměřenou vzdálenost pomocí ultrazvukového zařízení a rozdíl mezi těmito hodnotami. Další tabulka znázorňuje naměřené teploty pomocí teplotního senzoru. Měření proběhlo na několika odlišných místech za pomoci dvou kontrolních teploměrů – rtuťového a elektronického.

č.	Nastavená vzdálenost	Naměřená vzdálenost	Rozdíl vzdáleností
	[cm]	[cm]	[cm]
1.	20,0	20,4	-0,4
2.	50,0	50,3	-0,3
3.	80,0	79,8	0,2
4.	100,0	100,0	0,0
5.	120,0	119,8	0,2
6.	150,0	149,8	0,2
7.	180,0	179,1	0,9

8.	200,0	200,1	-0,1
9.	220,0	220,4	-0,4
10.	250,0	249,6	0,4
11.	280,0	280,1	-0,1
12.	300,0	300,6	-0,6
13.	320,0	320,6	-0,6
14.	350,0	351,1	-1,1
15.	380,0	375,0	5,0
16.	400,0	401,1	-1,1

Tab. 6.1 - Testování zařízení v uzavřeném prostoru

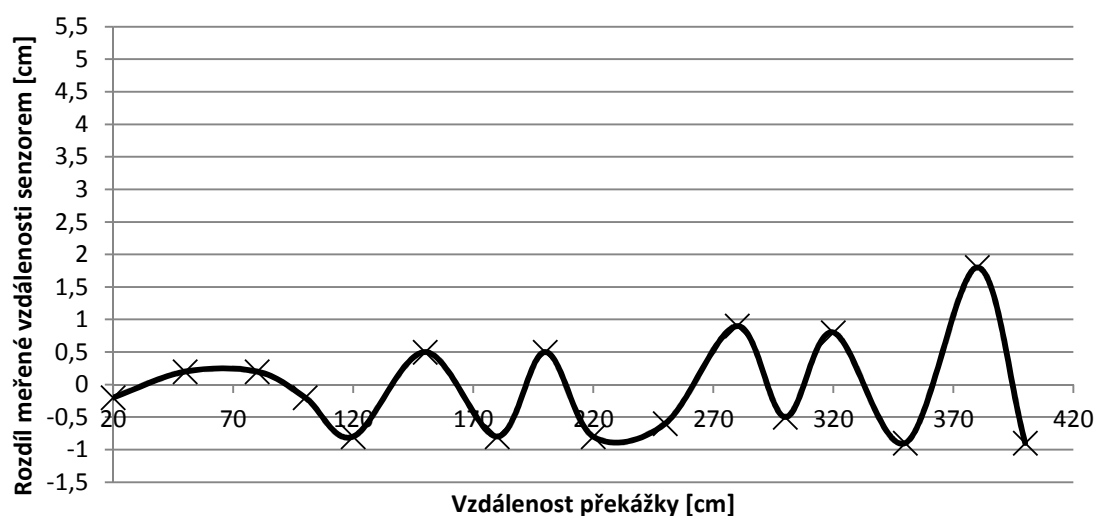
č.	Nastavená vzdálenost	Naměřená vzdálenost	Rozdíl vzdáleností
	[cm]	[cm]	[cm]
1.	20,0	20,2	-0,2
2.	50,0	49,8	0,2
3.	80,0	79,8	0,2
4.	100,0	100,2	-0,2
5.	120,0	120,8	-0,8
6.	150,0	149,5	0,5
7.	180,0	180,8	-0,8
8.	200,0	199,5	0,5
9.	220,0	220,8	-0,8
10.	250,0	250,6	-0,6
11.	280,0	279,1	0,9
12.	300,0	300,5	-0,5
13.	320,0	319,2	0,8
14.	350,0	350,9	-0,9
15.	380,0	378,2	1,8
16.	400,0	400,9	-0,9

Tab. 6.2 - Testování zařízení v otevřeném prostoru

č.	Místo měření	Elektronický	Rtuťový	TMP36GT9Z
		[°C]	[°C]	[°C]
1.	Byt	27,9	27,5	27,9
2.	Chodba	23,5	23,5	23,5
3.	Sklep	17,2	17,0	17,2
4.	Park	28,9	29,0	28,9
5.	Lednice	11,5	11,5	11,5

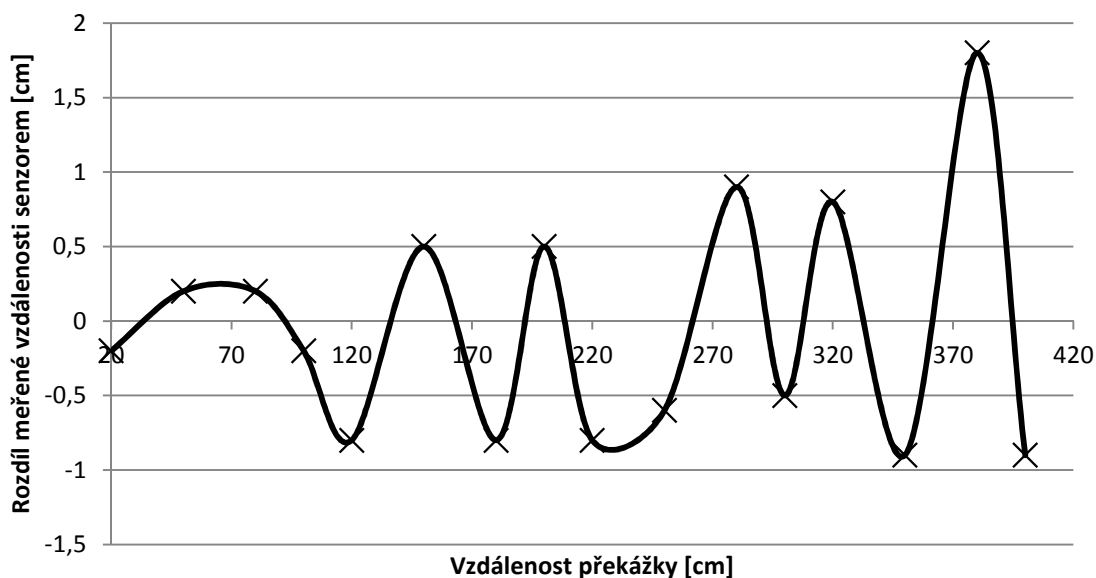
Tab. 6.3 - Testování teplotního senzoru TMP9GTZ

Rozdílová charakteristika vzdálenosti mezi překážkou a senzorem - vnitřní prostory



Graf 6.1 - Rozdílový grav pro měření ve vnitřním prostoru

Rozdílová charakteristika vzdálenosti mezi překážkou a senzorem - vnější prostory



Graf 6.2 - Rozdílový gra pro měřenív otevřeném prostoru

7 ZÁVĚR

Cílem této bakalářské práce bylo navrhnout, výroba a oživení ultrazvukového měřiče vzdálenosti. Jsou zde popsány principy a vlastnosti ultrazvukového signálu, jeho možné využití a rozdíl mezi měřením vzdálenosti pomocí ultrazvukových vln a jinými způsoby měření.

V následujících kapitolách jsou popsány části vytvořeného zařízení pro měření vzdálenosti. Popis vybraného ultrazvukového senzoru SRF05, mikroprocesoru ATmega8, který ovládá celé zařízení. Následně jsou popsány jednotlivé vlastnosti všech komponentů jako je klávesnice, výstup na LCD, ukládání do externí paměti a rozhraní pro komunikaci s PC.

Práce dále obsahuje detailní popis návrhu elektronických schémat, tvorbu knihoven součástek, návrh DPS v programu Eagle a jeho následnou výrobu. Dále je uveden detailní popis řídicího programu zařízení od principů zpracování po ukázky jednotlivých částí kódu. Ten je celý k dispozici včetně zkompilevaného souboru .hex na přiloženém DVD.

Během konstrukce a programování bylo nutné vyřešit zásadní problém a tím je životnost mikroprocesoru ATmega8. Kvůli testování a neustálému přepisování paměti nakonec docházelo k degradaci jeho funkcí. ATMEL udává maximální počet přepisů paměti na cca 1000 cyklů. Při vývoji bylo tohoto čísla téměř plně dosaženo. Proto bylo nutné obstarat záložní mikroprocesory, které postupně nahradily první pokusný mikroprocesor.

Hotové zařízení bylo testováno ve vnitřních a venkovních prostorech. Rozdíl u vnitřních prostor byl maximálně 5cm. Po vypočtení a zprůměrování všech rozdílů byla vypočtena průměrná odchylka 0,138cm. U venkovních prostor se jedná o maximální rozdíl 1,8cm. Průměrná odchylka pak byla vypočtena na 0,05cm. Podle předpokladu je měření v otevřeném prostoru o něco přesnější. Senzor totiž nezachytává zbloudilé signály od okolních objektů, které jsou v uzavřeném prostoru blíže.

Teplotní čidlo bylo testováno v 5 rozdílných prostorech s razantně rozdílnou teplotou. Teplotní senzor TMP36 udával stejné výsledky jako elektronický teploměr umístěný v budíku. Rtuťový teploměr má od elektronických senzorů drobné odchylky. To je způsobeno nemožností přesně odečíst hodnoty, kvůli malému rozlišení stupnice.

Zařízení celkově funguje dle požadovaných parametrů. Ukládání dat je prováděno na externí EEPROM, odkud se hodnoty postupně načítají dle potřeby. Všechny uložené hodnoty je možné si prohlédnout v menu nastavení v záložce retenčního měření.

Při tvorbě bakalářské práce jsem získal spoustu nových teoretických i praktických poznatků týkajících se problematiky ultrazvukového měření, ukládání dat na externí EEPROM, práci s LCD, komunikaci s PC a navrhování schémat a DPS.

Zařízení je možné použít pro ovládání autonomních robotů, jako senzor vzdálenosti do palubního počítače automobilu či řízení výšky létajících dronů. Dále ve stavebnictví místo pásma nebo jako samostatný teploměr s velkou přesností.

8 SEZNAM POUŽITÉ LITERATURY

- [1] Stránky s kompletním vysvětlením funkcí AVR a se spousty ukázek programu
<http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=70673>
- [2] Návod k obsluze ATmega pro začátečníky
<http://hw.cz/Teorie-a-praxe/Zacatecnici/ART1767-Zaciname-s-mikroprocesory-Atmel-AVR.html>
- [3] Popis všech prací s AVR
http://cs.wikibooks.org/wiki/Programujeme_jedno%C4%8Dipy
- [4] Datasheet výrobce pro převodník FT230XS-R
http://www.soselectronic.cz/a_info/resource/c/FTDI/DS_FT230X.pdf
- [5] Popis ovládání FT230XS-R
<http://www.electronics-lab.com/project/usb-to-serial-converter-using-ftdi-ft230x/>
- [6] Popis UART
<http://cs.wikipedia.org/wiki/UART>
- [7] Datasheet výrobce pro ultrazvukový senzor SRF05
<http://www.picaxe.com/docs/srf005.pdf>
<http://www.robot-electronics.co.uk/htm/srf05tech.htm>
- [8] Popis funkcí časovače/čítače
[http://cs.wikipedia.org/wiki/%C4%8Casova%C4%8D_\(po%C4%8D%C3%ADta%C4%8D\)](http://cs.wikipedia.org/wiki/%C4%8Casova%C4%8D_(po%C4%8D%C3%ADta%C4%8D))
- [9] Datasheet výrobce pro LCD WH1602A
<https://www.gme.cz/data/attachments/dsh.513-218.1.pdf>
- [10] Datasheet výrobce pro ultrazvukový vysílač UST-40T
<http://www.hestore.hu/files/ust40t.pdf>
- [11] Datasheet výrobce pro ultrazvukový přijímač UST-40R
http://files.elv.de/Assets/Produkte/4/462/46269/Downloads/46269_Ultraschall_Empfänger_DS.pdf
- [12] Datasheet výrobce pro teplotní čidlo
http://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf
- [13] Datasheet výrobce pro krystal 12MHz
<http://www.gme.cz/img/cache/doc/131/074/krystal-hc49-s-qm-12-000mhz-datasheet-1.pdf>
- [14] Datasheet výrobce pro elektromagnetický bzučák
<http://www.gme.cz/img/cache/doc/641/067/al1210mte-p5r-datasheet-1.pdf>
- [15] Tabulky pro rychlost zvuku
<http://www.converter.cz/tabulky/rychlost-zvuku-vzduch.htm>
- [16] Popis použití EEPROM
<http://www.scienceprog.com/atmega-EEPROM-memory-writing/>
- [17] Popis práce EEPROM
<http://www.sakshieducation.com/Engineering/Story.aspx?cid=12&nid=90539>
- [18] Popis ovládání LCD
<http://www.electronicshub.org/interfacing-16x2-lcd-avr-microcontroller/>
- [19] Popis externí sběrnice SPI a I²C
<https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/>

9 SEZNAM ZKRATEK

A/D	- <i>Analog/Digital</i> – převod analogového signálu na digitální
CRT	- <i>Catode ray tube</i> – zařízení s katodovou trubicí
D/A	- <i>Digital/Analog</i> – převod digitální signálu na analogový
DPS	- Deska plošných spojů
EEPROM	- <i>Electrically Erasable Programmable Only Read-Only memory</i> Elektricky mazatelná paměť
I ² C	- <i>Inter-Integrated Circuit</i> – sériová počítačová sběrnice
LCD	- <i>Liquid Crystal Display</i> – displej z tekutých krystalů
MCU	- <i>Microcontroller</i> – mikrokontroler
NAND	- Logický člen
RS232	- Sériové rozhraní
RXD	- <i>Received of port D</i> – přijímač portu D
SCL	- <i>Synchronous clock</i> – hodinový signál
SDA	- <i>Synchronous data</i> – synchronní data
TCNT	- <i>Time Counter</i> – čítač časovač
TTL	- Tranzistor-Tranzistor-Logic
TXD	- <i>Transmitted of port D</i> – vysílač portu D
USART	- <i>Universal Synchronous Asynchronous Receiver Transmitter</i> Univerzální synchronní asynchronní přijímač, vysílač
USB	- <i>Universal Serial Bus</i> – univerzální sériová sběrnice

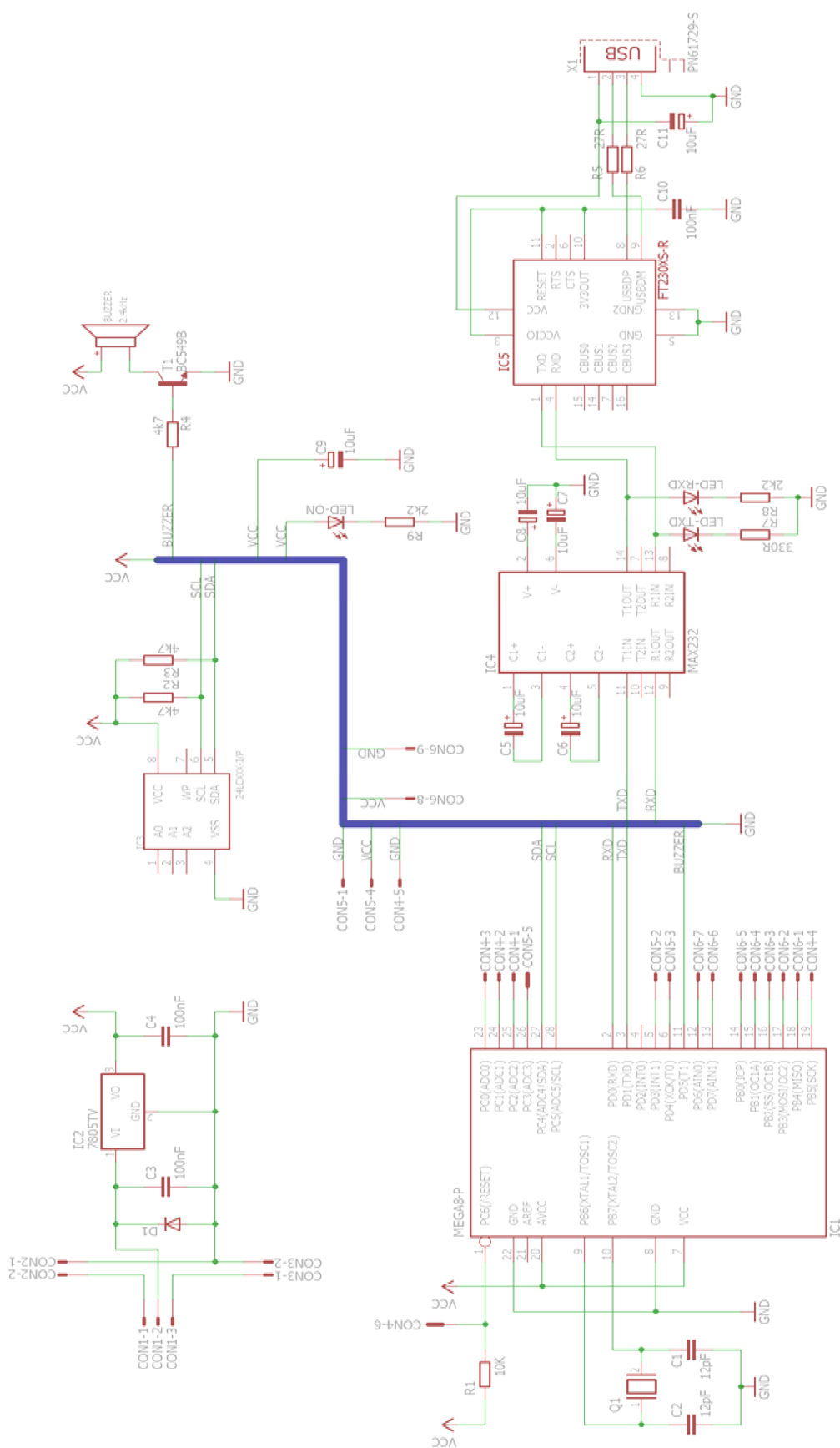
10 SEZNAM PŘÍLOH

10 SEZNAM PŘÍLOH.....	43
10.1 SCHÉMA ZAPOJENÍ	44
10.2 DPS (DESKY PLOŠNÝCH SPOJŮ)	46
10.3 SEZNAM SOUČÁSTEK	48
10.3.1 ŘÍDÍCÍ DESKA	48
10.3.2 ULTRAZVUKOVÝ SENZOR S ČIDLEM TEPLoty	49
10.3.3 KLÁVESNICE	49
10.3.4 LCD	49
10.4 FOTKY ZAŘÍZENÍ	50

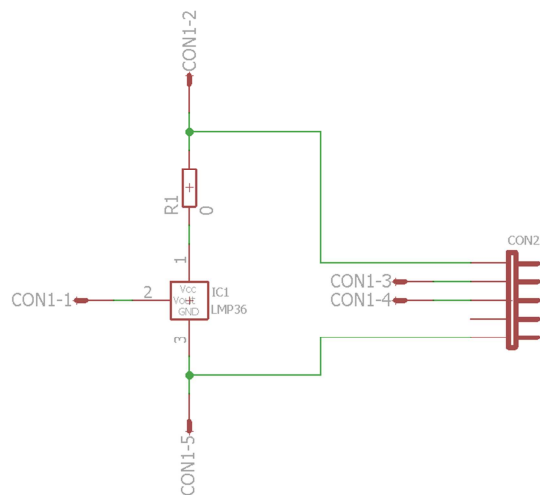
SEZNAM OBRÁZKŮ

Obr. 10.1 - Řídící deska.....	44
Obr. 10.2 - Ultrazvukový senzor s teplotním čidlem	45
Obr. 10.3 – Klávesnice	45
Obr. 10.4 - LCD s ovládáním kontrastu a podsvětlením	45
Obr. 10.5 - Řídící deska – pohled TOP (spoje bez součástek)	46
Obr. 10.6 - Řídící deska - pohled TOP (osazení vč. propojek J1-J4).....	46
Obr. 10.7 - Ultrazvukový senzor s teplotním čidlem – pohled TOP (spoje bez součástek).....	46
Obr. 10.8 - Ultrazvukový senzor s teplotním čidlem – pohled TOP (osazení)	46
Obr. 10.9 - Klávesnice - pohled TOP (spoje bez součástek).....	47
Obr. 10.10 - Klávesnice - pohled TOP (osazení)	47
Obr. 10.11 - LCD s ovládáním kontrastu a podsvětlením – pohled TOP (spoje bez součástek).....	47
Obr. 10.12 - LCD s ovládáním kontrastu a podsvětlením - pohled TOP (osazení).....	48
Obr. 10.13 - Zařízení	50
Obr. 10.14 - Hardware zařízení	50

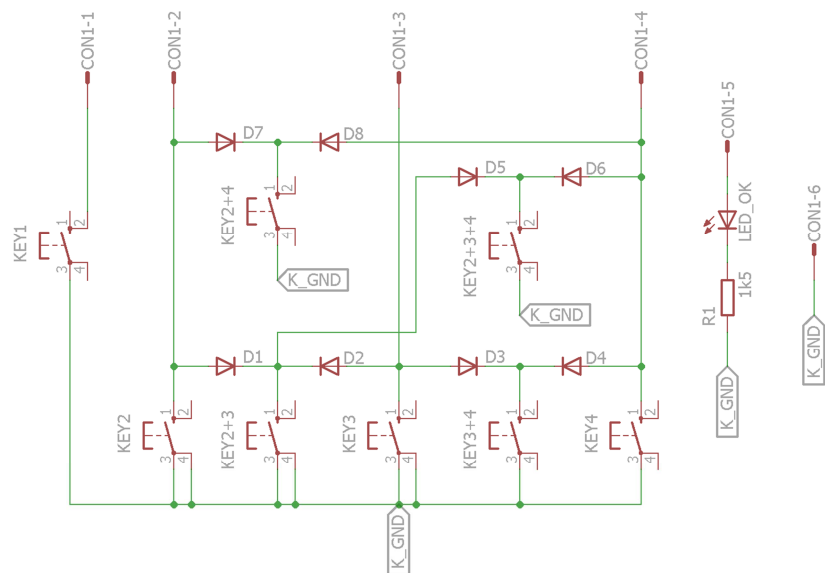
10.1 SCHÉMA ZAPOJENÍ



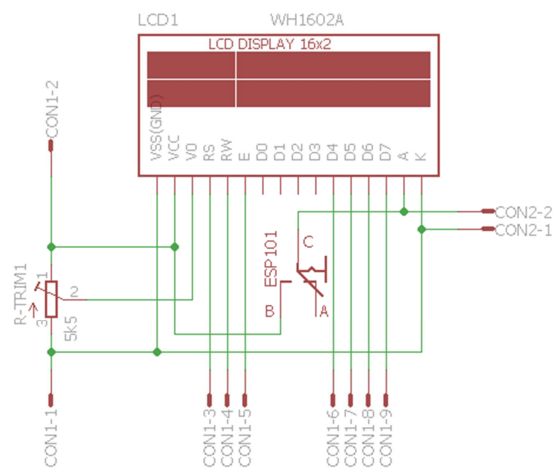
Obr. 10.1 - Řídící deska



Obr. 10.2 - Ultrazvukový senzor s teplotním čidlem

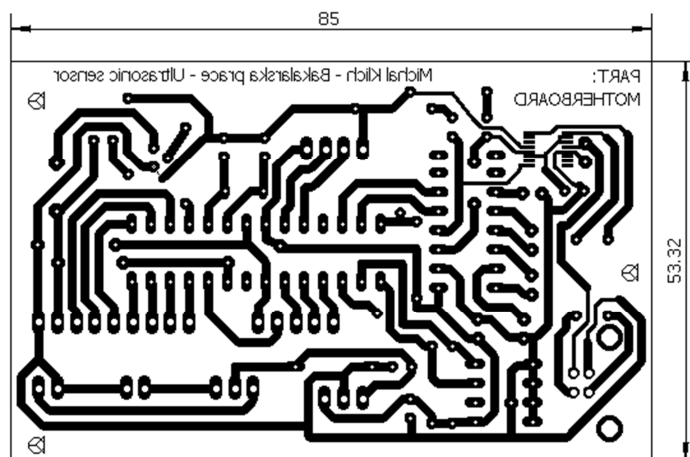


Obr. 10.3 – Klávesnice

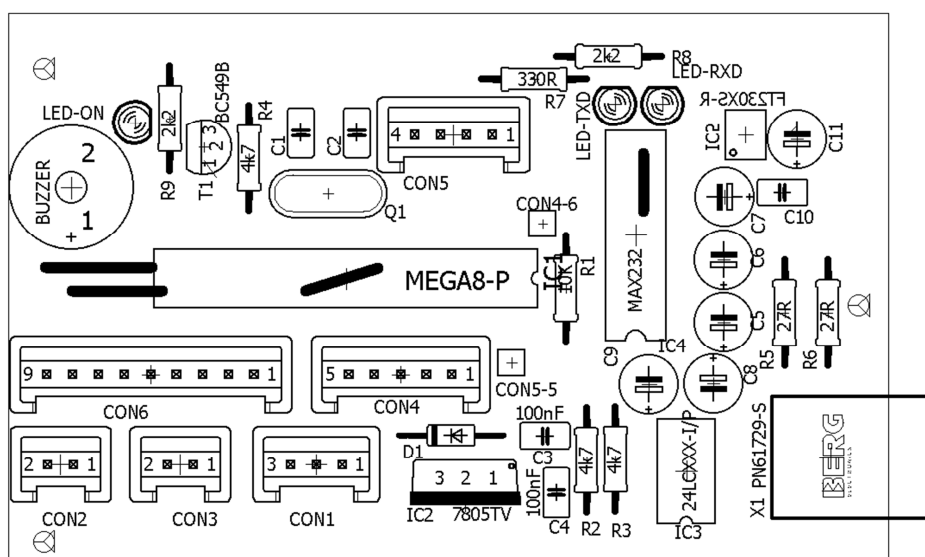


Obr. 10.4 - LCD s ovládáním kontrastu a podsvětlením

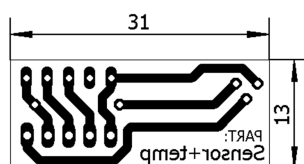
10.2 DPS (DESKY PLOŠNÝCH SPOJŮ)



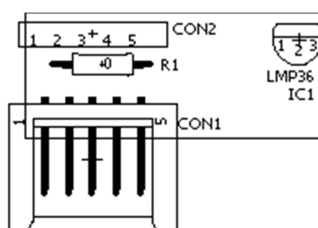
Obr. 10.5 - Řídící deska – pohled TOP (spoje bez součástek)



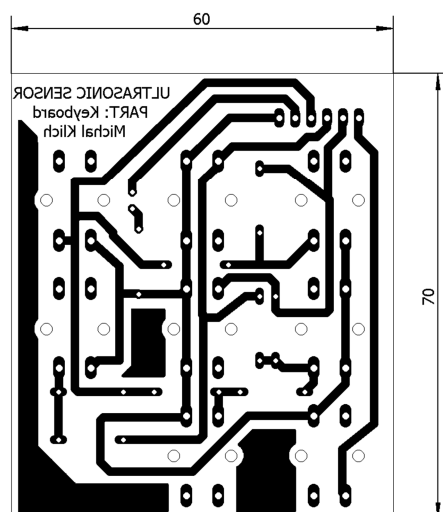
Obr. 10.6 - Řídící deska - pohled TOP (osazení vč. propojek J1-J4)



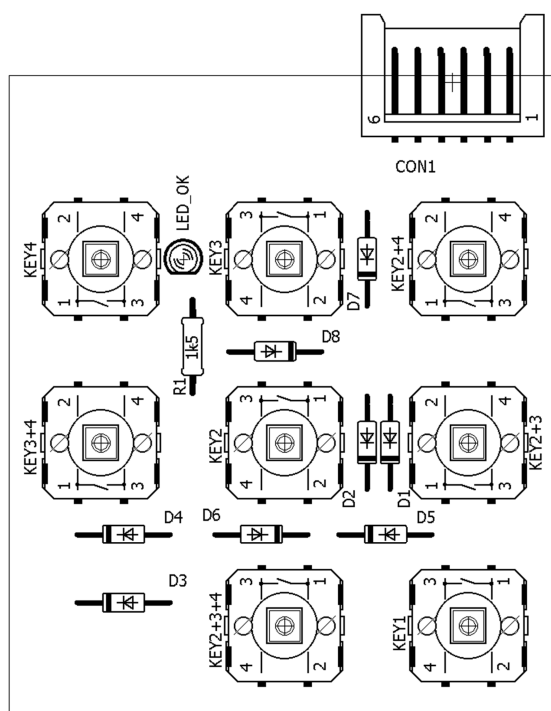
Obr. 10.7 - Ultrazvukový senzor s teplotním čidlem – pohled TOP (spoje bez součástek)



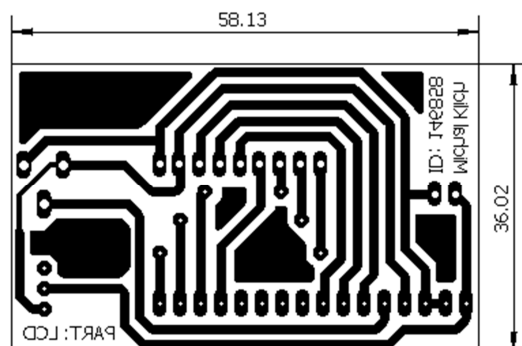
Obr. 10.8 - Ultrazvukový senzor s teplotním čidlem – pohled TOP (osazení)



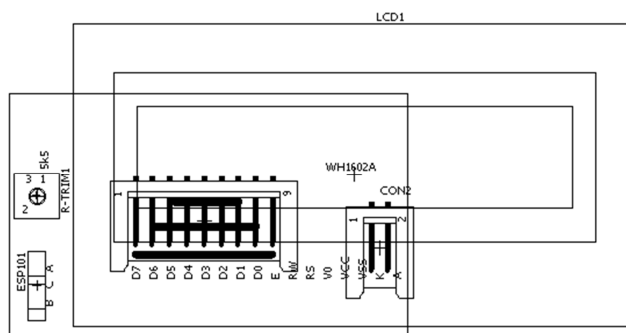
Obr. 10.9 - Klávesnice - pohled TOP (spoje bez součástek)



Obr. 10.10 - Klávesnice - pohled TOP (osazení)



Obr. 10.11 - LCD s ovládáním kontrastu a podsvětlením – pohled TOP (spoje bez součástek)



Obr. 10.12 - LCD s ovládáním kontrastu a podsvětlením - pohled TOP (osazení)

10.3 SEZNAM SOUČÁSTEK

10.3.1 ŘÍDÍCÍ DESKA

Název	Hodnota	Pouzdro	Typ součástky
Buzzer	2,4kHz	PPO/Black	Elektromagnetický bručák
C1	12pF	CO25	Keramický kondenzátor
C2	12pF	CO25	Keramický kondenzátor
C3	100nF	CO25	Keramický kondenzátor
C4	100nF	CO25	Keramický kondenzátor
C5	10μF	E2,5-6E	Elektrolytický kondenzátor
C6	10μF	E2,5-6E	Elektrolytický kondenzátor
C7	10μF	E2,5-6E	Elektrolytický kondenzátor
C8	10μF	E2,5-6E	Elektrolytický kondenzátor
C9	10μF	E2,5-6E	Elektrolytický kondenzátor
C10	100nF	CO25	Keramický kondenzátor
C11	10μF	E2,5-6E	Elektrolytický kondenzátor
CON1	-	PSH02-03P / S03P	Konektor pro přepínač napájení
CON2	-	PSH02-02P / S02P	Konektor pro trafo 4-12V
CON3	-	PSH02-02P / S02P	Konektor pro 9V baterii
CON4	-	PSH02-05P / S05P	Konektor pro připojení klávesnice
CON4-6	-	S1G01	Konektor pro připojení klávesnice
CON5	-	PSH02-03P / S03P	Konektor pro připojení senzoru
CON5-5	-	S1G01	Konektor pro připojení senzoru
CON6	-	PSH02-09P / S09P	Konektor pro připojení LCD
D1	1N4448	DO35	Dioda 75V/0,15A
IC1	-	DIL28	Mikroprocesor ATmega8
IC2	-	TO220V	Stabilizátor napětí 7805V
IC3	-	DIL8	EEPROM 21LCXXX
IC4	-	DIL16	Převodník MAX232
IC5	-	SSOP-16	Převodník FT230XS-R
J1-5	-	-	Propojky na DPS
LED_ON	Zelená	LED3MM	LED dioda
LED_RXD	Červená	LED3MM	LED dioda
LED_TXD	Žlutá	LED3MM	LED dioda
Q1	12MHz	QS	Krystal HC49/S

R1	10k Ω	0207	Rezistor
R2	4k7 Ω	0207	Rezistor
R3	4k7 Ω	0207	Rezistor
R4	4k7 Ω	0207	Rezistor
R5	27 Ω	0207	Rezistor
R6	27 Ω	0207	Rezistor
R7	330 Ω	0207	Rezistor
R8	2k2 Ω	0207	Rezistor
R9	2k2 Ω	0207	Rezistor
T1	BC549B	TO92	Vysokofrekvenční NPN tranzistor
X1	-	PN61729-S	Konektor rozhraní USB

10.3.2 ULTRAZVUKOVÝ SENZOR S ČIDLEM TEPLoty

Název	Hodnota	Pouzdro	Typ součástky
CON1	-	PSH02-05W / L05P	Konektor pro připojení řídicí desky
CON2	-	S1G05W	Výstup pro SRF05
IC1	LMP36GT9Z	TO92	Teplotní čidlo
R1	0 Ω	0207	Rezistor

10.3.3 KLÁVESNICE

Název	Hodnota	Pouzdro	Typ součástky
CON1	-	PSH02-09W / L09P	Konektor pro připojení řídicí desky
D1-8	1N4448	DO35	Dioda 75V/0,15A
KEY1 - KEY2+3+4	-	B3F-40XX	Mikrospínač
LED_OK	Červená	LED3MM	LED dioda
R1	1k5 Ω	0207	Rezistor

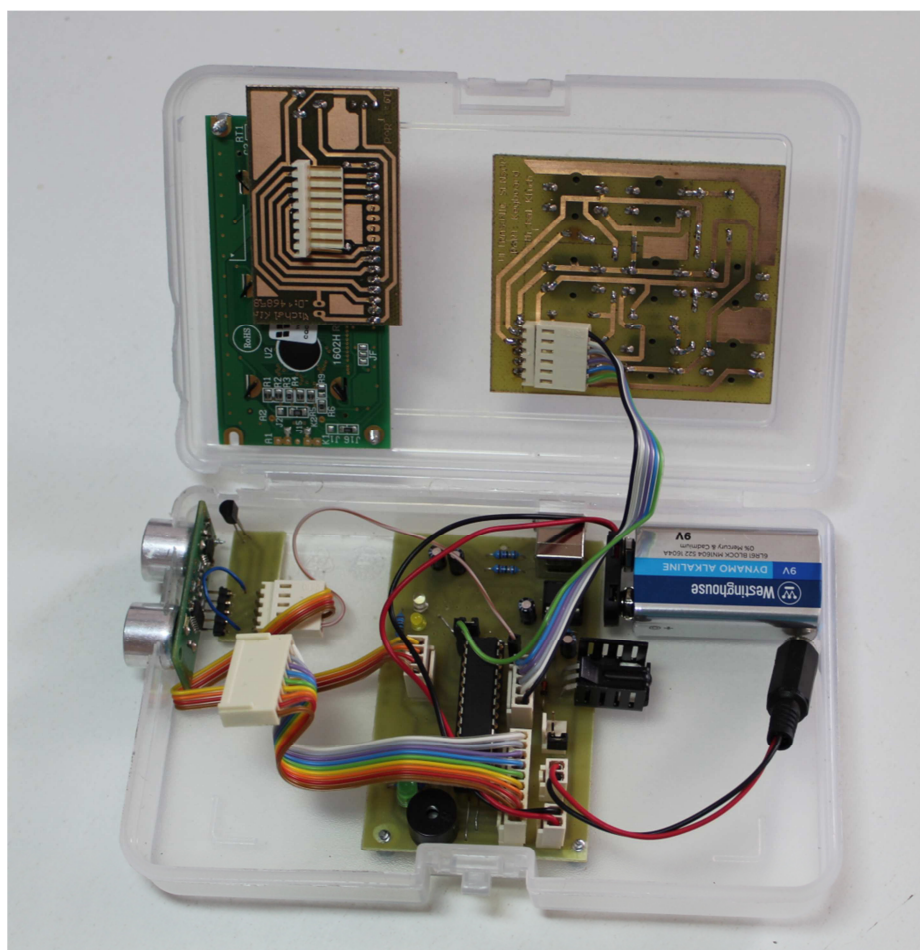
10.3.4 LCD

Název	Hodnota	Pouzdro	Typ součástky
CON1	-	PSH02-05W / L05P	Konektor pro připojení řídicí desky
CON2	-	PSH02-02P / S02P	Konektor ventilátoru chladiče
J1-J3	-	-	Propojka na DPS – nulový odpor
LCD1	-	ATM1602B	LCD display 16x2
P-ESP101	-	ESP10XX	Přepínač podsvícení LCD
R-TRIM1	5k5 Ω	CA6-V5	Manuální rezistorový trimr

10.4 FOTKY ZAŘÍZENÍ



Obr. 10.13 - Zařízení



Obr. 10.14 - Hardware zařízení