

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

MĚŘENÍ VZDÁLENOSTI A RYCHLOSTI POMOCÍ VYSÍLAČŮ A PŘIJÍMAČŮ ULTRAZVUKU

BAKALÁŘSKÁ PRÁCE

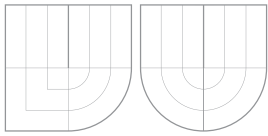
BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

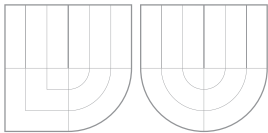
MARTIN VLÁČIL

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ



FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

MĚŘENÍ VZDÁLENOSTI A RYCHLOSTI POMOCÍ VYSÍLAČŮ A PŘIJÍMAČŮ ULTRAZVUKU

DISTANCE AND SPEED MEASUREMENT BY ULTRASONIC TRANSMITTERS AND RECEIVERS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN VLÁČIL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JOSEF STRNADEL, Ph.D.

BRNO 2015

Abstrakt

Práce pojednává o principu měření vzdálenosti a rychlosti těles pomocí ultrazvuku, jeho realizaci za pomoci jednoduchého obvodu s mikrokontrolérem PIC12F629, jakožto modulu pro platformu Arduino UNO. Popisuje jednotlivé funkční bloky a hardwarovou i softwarovou realizaci. Dále zahrnuje rozbor možných úskalí spojených s tímto způsobem měření.

Abstract

This thesis deals with distance and speed measurement of objects based on ultrasound principle, its realization by simple circuit based on microcontroller PIC12F629 as module for platform Arduino UNO. Describes its functional blocks and its hardware and software realization. Further on it covers some possible problems connected with this kind of measurement.

Klíčová slova

Ultrazvuk, Vzdálenost, Rychlost, Měření, PIC, 12F629, Arduino, UNO, Atmel, Modul, Mikrokontrolér

Keywords

Ultrasound, Distance, Speed, Measurement, PIC, 12F629, Arduino, UNO, Atmel, Module, Microcontroller

Citace

Martin Vláčil: Měření vzdálenosti a rychlosti pomocí vysílačů a přijímačů ultrazvuku, bakalářská práce, Brno, FIT VUT v Brně, 2015

Měření vzdálenosti a rychlosti pomocí vysílačů a přijímačů ultrazvuku

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing., Ph.D. Josefa Strnadela

.....

Martin Vláčil

20. května 2015

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce, Ing., Ph.D Josefu Strnadelovi, který mi hlavně v začátcích mého snažení značně pomohl a poskytl mi mnoho materiálů, ze kterých jsem mohl čerpat. Pak bych také velice rád poděkoval svému otci, Milanu Vláčilovi, za odborné rady z oblasti elektrotechniky, bez kterých bych tuto práci nikdy včas nedokončil.

© Martin Vláčil, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

| | |
|--|-----------|
| 1 Úvod | 3 |
| 1.1 Ultrazvuk | 5 |
| 1.2 Ultrazvuk v praxi | 5 |
| 1.3 Teorie měření vzdálenosti | 7 |
| 1.4 Teorie měření rychlosti | 8 |
| 1.5 Výpočet rychlosti zvuku | 9 |
| 1.5.1 Faktory ovlivňující rychlost zvuku | 11 |
| 1.5.2 Odraz zvuku | 14 |
| 2 Návrh výrobku | 17 |
| 2.1 Měřicí modul | 17 |
| 2.1.1 Výběr komponent pro modul | 20 |
| 2.1.2 Obsluha modulu | 26 |
| 3 Realizace | 29 |
| 3.1 Deska plošného spoje | 30 |
| 3.1.1 Popis výroby DPS | 30 |
| 3.1.2 Osazení | 31 |
| 3.1.3 Rozpis součástek | 32 |
| 3.2 Program modulu | 32 |
| 3.2.1 Inicializace | 33 |
| 3.2.2 Konfigurace | 33 |
| 3.2.3 Funkce main | 33 |
| 3.2.4 Funkce přerušní | 34 |
| 3.2.5 Funkce detekce signálu | 34 |

| | | |
|----------|--|-----------|
| 3.2.6 | Generování kmitočtu | 34 |
| 3.2.7 | Funkce posláání dat | 35 |
| 3.2.8 | Funkce zprostředkování sériového přenosu | 36 |
| 3.2.9 | Metriky kódu | 37 |
| 3.3 | Obslužný program | 37 |
| 3.3.1 | Funkce programu | 37 |
| 3.3.2 | Nastavení | 37 |
| 3.3.3 | Příjem sériové komunikace | 38 |
| 3.3.4 | Měření teploty | 38 |
| 3.3.5 | Infračervený přijímač | 38 |
| 3.3.6 | Výpočet a měření vzdálenosti | 39 |
| 3.3.7 | Uživatelské nastavení | 39 |
| 3.3.8 | Metriky kódu | 39 |
| 3.4 | Zobrazení dat | 40 |
| 4 | Testování | 41 |
| 4.1 | Hardware | 41 |
| 4.2 | Prostředí | 42 |
| 4.2.1 | Omezení způsobená odrazem | 42 |
| 4.2.2 | Omezení způsobená tlumením | 44 |
| 4.3 | Zvýšení efektivity | 45 |
| 4.3.1 | Zvýšení dosahu | 45 |
| 4.3.2 | Rušení | 46 |
| 5 | Závěr | 47 |
| A | Seznam příloh | 50 |

Kapitola 1

Úvod

Tématem bakalářské práce je využití ultrazvuku jako prostředku pro měření **vzdálenosti** a **rychlosti** objektu v prostoru. K tomu je potřeba si uvědomit, co to ultrazvuk je a jaká omezení tento způsob měření obnáší již z jeho podstaty.

Výhody takového měření oproti ručnímu jsou značné. Měření jsou uskutečněna velice rychle, řádově do pár vteřin, a to bez pomoci jiné osoby a nutnosti složité manipulace s měřidlem¹.

V dnešní době se k elektronickému měření vzdálenosti využívají převážně dva různé způsoby, a to pomocí laseru² nebo ultrazvuku. Oba způsoby pracují na podobném principu.

K úspěšnému měření je potřeba znát vlastnosti ultrazvuku, jako je jeho rychlost, frekvence či odrazivost. To platí i o objektu, jehož vzdálenost a rychlost se měří, tedy hlavně faktor útlumu, velikost objektu nebo úhel plochy objektu vůči ultrazvukovému vysílači. Dále je potřeba znát vlivy prostředí na ultrazvukové vlny a rozhodnout, které z nich je nezbytně nutné zahrnout do výpočtu a které naopak nehrají až tak významnou roli pro samotný výpočet měřených veličin a je možné je ve výpočtu vynechat.

Cílem práce je tedy navrhnout, realizovat a odzkoušet zařízení, které je schopno takováto měření uskutečnit.

¹ Například metr či pásmo

² kvantový generátor světla - z angl. „Light Amplification by Stimulated Emission of Radiation“, neboli „Zesílení světla pomocí stimulované emise záření“



Obrázek 1.1: Zvukové spektrum³

Netopýr využívá ultrazvuku k orientaci v prostoru a lovu. Ústy vytváří ultrazvukové kmity o frekvenci kolem **80 kHz**. Jeho citlivýma ušima poté zachycuje jejich ozvy a podle zpoždění mezi vysláním ultrazvukových vln a zachycením jejich ozev dokáže přesně určit vzdálenost i pozici překážek. Tento princip se nazývá echolokace⁴.

Stejného principu využiji ve svém zařízení, jehož konstrukce je náplní mé bakalářské práce.

³ Zařízení využívající ultrazvuk pracují s frekvencemi od 20kHz po několik gigahertz (GHz)

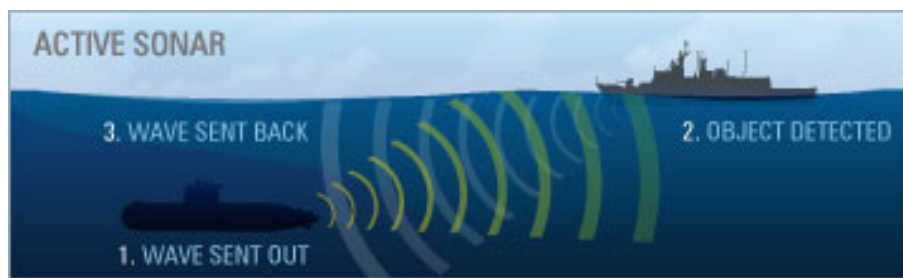
⁴ z ang. *echo* - ozvěna a *location* - pozice

1.1 Ultrazvuk

Ultrazvuk je **akustická** (zvuková) tlaková vlna oscilující při frekvenci vyšší, než je horní limit slyšitelnosti lidského ucha[10]. Ultrazvuk je tedy oddělen od „normálního“ (slyšitelného) zvuku pouze faktem, že jej lidské ucho již nedokáže zachytit. Avšak takovýto limit je jiný pro každého jedince, a tak je ultrazvuk definován na přibližně 20 kilohertz ($20kHz$).

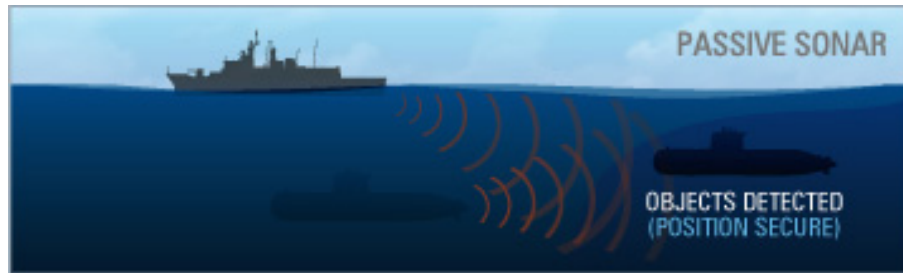
1.2 Ultrazvuk v praxi

Jak již bylo zmíněno (Kapitola 1), některá zvířata aktivně využívají ultrazvuku k orientaci či komunikaci, a to již od nepaměti. Lidstvo však na využití ultrazvuku přišlo teprve roku 1915, kdy byl vynalezen první sonar⁵ fyzikem Paulem Langevinem. Princim byl převzat z tehdy již existujícího radaru, akorát rádiové vlny byly nahrazeny ultrazvukovými. Jeho první využití bylo ve válečných ponorkách. U sonaru se ultrazvukové vlny šíří jinak, jelikož přenosovým médiem není vzduch, nýbrž voda, kde se ultrazvukové vlny pohybují přibližně pětinasobnou rychlostí [7].



Obrázek 1.2: Aktivní sonar [2]

⁵ z angl. „Sound Navigation And Ranging“



Obrázek 1.3: Pasivní sonar [2]

Existují dva druhy sonaru, **aktivní** (Obrázek 1.2) a **pasivní** (Obrázek 1.3). Liší se od sebe způsobem generování ultrazvukového vlnění. U aktivního sonaru je z ultrazvukových měničů⁶ vyslán ultrazvukový signál, který se odrazí od překážky. Jeho nevýhodou v boji je, že zároveň prozradí pozici objektu, který signál vyslal. Pasivní sonar naopak pouze sleduje pomocí hydrofonů⁷ zvukové vlny šířící se vodou a následně je vyhodnotí.

Asi nejnámější je využití ultrazvuku v lékařství, kdy se pomocí ultrazvuku sleduje vývoj člověka v prenatálním stádiu vývoje (Obrázek 1.4) či při diagnostice stavu orgánů v těle. Tato metoda využití ultrazvuku se jmenuje **ultrasonografie**⁸ (zkr. Sono). Je to dnes velmi rozšířená a bezpečná, neinvazivní metoda, která ani po dlouhodobém či opakovaném podstupování nepřináší žádná zdravotní rizika (dříve, když neexistovala ultrasonografie se používaly pouze rentgeny, které vyzařují tělu nebezpečné ionizující záření). Touto metodou lze vyšetřovat většinu orgánů v těle pacienta (nelze vyšetřit střeva či žaludek).

⁶ Vysílač/přijímač ultrazvukového vlnění

⁷ Podvodní mikrofón, umožňující pasivně naslouchat zvukům z okolní vody [8]

⁸ Metoda zobrazující měkké tkáně a orgány za pomoci ultrazvuku



Obrázek 1.4: Pohled na dítě v děloze pomocí ultrazvuku

1.3 Teorie měření vzdálenosti

Proces je založen na odrazivosti akustických vln. Akustické vlny jsou definovány jako podélné tlakové vlny v médiu, ve kterém se pohybují. Předměty, jejichž dimenze jsou větší, než je vlnová délka dorážejících akustických vln, takovou vlnu mohou odrazit. Takovéto vlny jsou nazývány ozvy či anglicky echa.

Pokud tedy známe rychlost těchto vln v daném médiu a jsme schopni změřit čas, který uběhne mezi vysláním vlnění a návratem echa, jsme schopni přesně změřit vzdálenost objektu, od kterého se vlny odrazily.

Oproti laserovým přístrojům jsou ultrazvukové přístroje na měření vzdálenosti méně výkonné co se dosahu⁹ a přesnosti týče. Výhodou a zároveň nevýhodou přístrojů na bázi ultrazvuku je, že zvukové vlny neprojdou ani čirými pevnými materiály, jako je třeba sklo, což se někdy může hodit a někdy je to považováno jako nedostatek, avšak na druhou stranu oproti světelnému záření mohou projít neprůsvitnými materiály (měkká tkáň, kov). Cenově jsou navíc ultrazvukové měřiče oproti laserovým velice levné.

⁹ Maximálně kolem dvaceti metrů, obvykle však pouze šest

1.4 Teorie měření rychlosti

Existují dva přístupy k měření rychlosti objektu pomocí ultrazvuku. A to pomocí dopplěrova jevu, kdy zvuková vlna odražená od přibližujícího se objektu má menší frekvenci než vlna vyslaná, vice versa vlna odražená od vzdalujícího se objektu má frekvenci vyšší. Tohoto jevu si můžeme všimnout třeba u projíždějící sanitky, kdy je zvuk sirén zkreslen v závislosti na tom, jestli se k nám přibližuje nebo se vzdaluje. Při tomto principu se musí spočítat rozdíl mezi těmito frekvencemi. Poté stačí dosadit do vzorce:

$$f = f_0 \frac{v}{v-v_z} \quad v_z = v - \frac{f_0 \cdot v}{f}$$

Kde:

- f_0 = frekvence vyslaného signálu
- f = frekvence odraženého signálu
- v = rychlost vlnění
- v_z = rychlost měřeného objektu

Tento princip lze uplatnit i bez nutnosti použití pro měření ultrazvuk a existují aplikace na mobilní telefony¹⁰, které tohoto jevu využívají pro amatérské měření rychlosti (např. projíždějících aut nebo RC modelů).

Druhý, přímočařejší princip využívá jednoduchého vzorce $v = s/t$, kde je rychlost rovna dráze vydělené časem. Tento princip ve své základní podobě nedokáže určit, zdalipak se objekt přibližuje či vzdaluje, jelikož dráha ze své podstaty nemůže být záporná, ale vypočteme-li tuto dráhu ze dvou krajních naměřených bodů, tento problém se vyřeší.

Pro své řešení jsem zvolil druhý princip z důvodu jeho snadnější aplikovatelnosti, jelikož je principiálně totožný s měřením vzdálenosti, které již bude uděláno.

¹⁰ Např. RC Speedo nebo Sound Speed Gun

1.5 Výpočet rychlosti zvuku

Nejdříve je nutné si uvědomit, že vlnová délka nemá žádný vliv na rychlost zvukových vln, stejnou rychlost tedy má slyšitelný zvuk, infrazvuk i ultrazvuk, proto se jejich rychlosti počítají totožně.

Pro výpočet rychlosti zvuku v různém homogenním (stejnorodém) prostředí slouží několik vzorců v závislosti na skupenství materiálu, kterým vlnění prochází (tzv. médiu):

Obecně je rychlost zvuku v plynu dána *Newton-Laplacovou rovnicí*:

$$c = \sqrt{\frac{K}{\rho}}$$

Kde:

- c = rychlost zvuku
- k = koeficient tuhosti, nebo-li objemový modul pružnosti
- ρ = hustota

Z toho můžeme vypočítat, že se akustické vlny šíří rychleji v tuhých nepružných materiálech (materiály odolné vůči deformaci působící silou) a pomaleji v hustých materiálech.

V ideálním plynu, který má na rozdíl od skutečného plynu ideální vlastnosti¹¹ je K jednoduše tlak plynu vynásobený *adiabatickým*¹² indexem daného plynu.

Adiabatický index vzduchu za normálních podmínek (teploty 0°C a tlaku 1010hPa) je přibližně 1,4.

Tím pádem lze rovnice zapsat ve tvaru:

¹¹ dokonale stlačitelný, bez vnitřního tření

¹² odehrávající se beze ztráty či příjmu tepla

$$c = \sqrt{\frac{K}{\rho}} = \sqrt{\gamma \cdot \frac{p}{\rho}}$$

Kde:

- γ = adiabatický index
- p = tlak vzduchu

V podmínkách ideálního plynu pak víme, že platí vztahy:

$$p = \frac{n \cdot R \cdot T}{V}$$

$$\rho = \frac{n \cdot M}{V}$$

Kde:

- n = počet molů
- R = molární plynová konstanta
 $R = 8,3145 \text{ J} \cdot \text{Mol}^{-1} \cdot \text{K}^{-1}$
- T = absolutní teplota v kelvinech
- V = objem
- M = molární hmotnost vzduchu
 $M = 0,0289645 \text{ Kg} \cdot \text{Mol}^{-1}$

Z čehož po dosazení do původní rovnice dostaneme vztah pro výpočet rychlosti zvuku ve vzduchu, který budeme používat při výpočtech.

$$c = \sqrt{\gamma \cdot \frac{p}{\rho}} = \sqrt{\frac{\gamma \cdot n \cdot R \cdot T}{\frac{n \cdot M}{V}}} = \sqrt{\frac{\gamma \cdot n \cdot R \cdot T \cdot V}{n \cdot M \cdot V}} = \sqrt{\frac{\gamma \cdot R \cdot T}{M}}$$

S touto znalostí již není těžké vypočítat rychlost zvuku ve vzduchu při ideálních podmínkách (teplotě 0°C , tlaku 1010hPa a vlhkosti 0%).

$$c = \sqrt{\frac{\gamma \cdot R \cdot T}{M}} = \sqrt{\frac{1,4 \cdot 8,3145 \cdot 273,15}{0,0289645}} = 331,45\text{m} \cdot \text{s}^{-1}$$

1.5.1 Faktory ovlivňující rychlost zvuku

1.5.1.1 Teplota vzduchu

U vzduchu a plynů obecně platí, že jejich teplota je hlavním faktorem ovlivňujícím rychlost šíření vlnění v nich. Tyto změny jsou lineární a výslednou rychlost lze jednoduše vypočítat po dosazení do vzorce pro výpočet rychlosti zvuku.

$$c = \sqrt{\frac{1,4 \cdot 8,3145 \cdot (273,15 + T)}{0,0289645}}$$

Kde:

- T = teplota ve stupních Celsia

1.5.1.2 Tlak vzduchu

Jak již bylo zmíněno (Kapitola 1.5), u ideálního plynu rychlost nezávisí na tlaku vzduchu. Skutečný plyn však není ideální. Vliv na rychlost v závislosti na tlaku vzduchu je však tak malý, že se může zanedbat. Tlak vzduchu samotný je závislý na mnoha různých faktorech, jako je nadmořská výška, velikost tíhového zrychlení, teplota či hustota atmosféry v daném místě. Z tohoto důvodu byl zaveden tzv. normální atmosférický tlak, který je definován jako průměrná hodnota tlaku vzduchu v nulové nadmořské výšce na $45.$ stupni severní šířky (s.š.) při teplotě 15 stupňů Celsia a tíhovém zrychlení $g_n = 9,80665\text{m}^{-2}$. Hodnota normálního atmosférického tlaku je **1013,25 hPa**. Atmosférický tlak s výškou rychle klesá¹³. Ve výšce přibližně $5,5\text{km}$ je tlak již poloviční, tedy přibližně **500 hPa**.

¹³ do jednoho kilometru zhruba o 1hPa co 10metrů

| Teplota vzduchu | rychlost zvuku |
|-----------------|----------------|
| -20 | 319,30m/s |
| -10 | 325,20m/s |
| 0 | 331,45m/s |
| 5 | 334,50m/s |
| 10 | 337,50m/s |
| 15 | 340,00m/s |
| 20 | 343,80m/s |
| 25 | 346,35m/s |
| 30 | 349,05m/s |
| 50 | 360,37m/s |
| 100 | 387,25m/s |

Tabulka 1.1: Tabulka rychlosti šíření vzduchu podle teploty

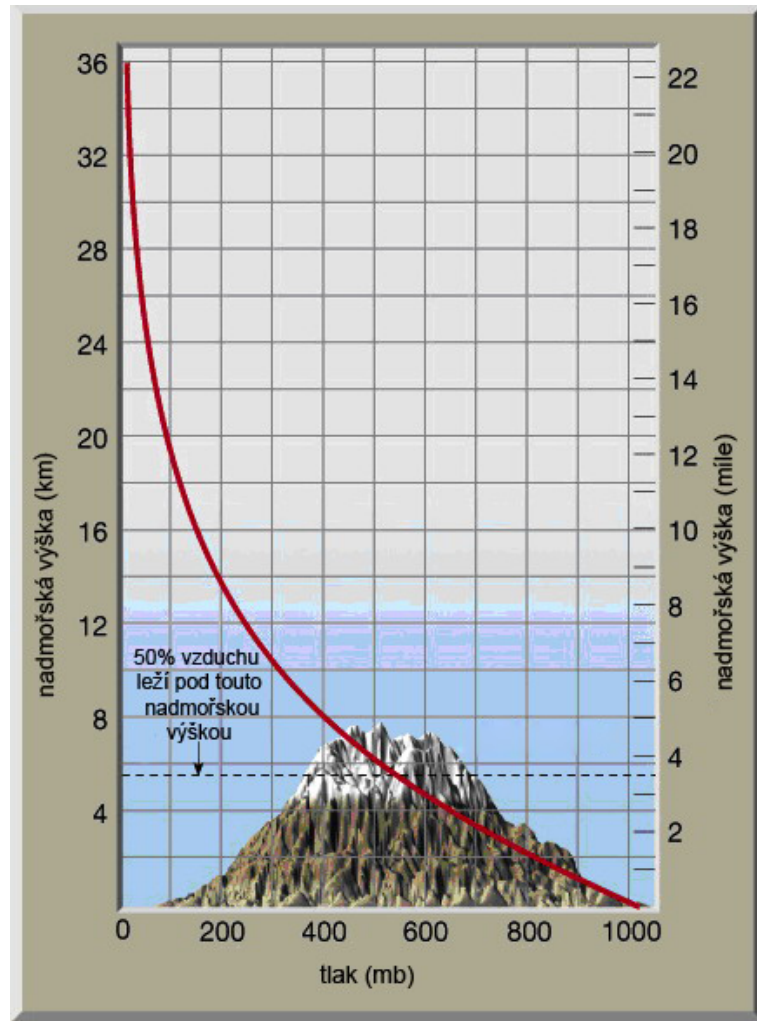
1.5.1.3 Vlhkost vzduchu

Posledním, alespoň trochu významným faktorem ovlivňujícím rychlost zvuku ve vzduchu, je jeho vlhkost. Ovlivnění je ovšem opět tak nepatrné, že ji také můžeme zanedbat. Rozdíl mezi suchým vzduchem (0% vlhkosti) a plně nasyceným vzduchem (100% relativní vlhkosti) je asi $1,32m \cdot s^{-1}$.

1.5.1.4 Kmitočet

Mezi lidmi panuje neopodstatněná domněnka, že ultrazvuk je rychlejší než například „normální“ slyšitelný zvuk, ale je tomu jinak, kmitočet ve skutečnosti nijak neovlivňuje rychlost šíření vlnění. Důvodem, proč se tedy nepoužívá k měření vzdáleností slyšitelný zvuk, ale ultrazvuk je, že ultrazvukem jsem schopni zaměřit menší objekty, na tak vysoké frekvenci již neexistuje mnoho okolního rušení a navíc měření není slyšet, což je také výhoda.

Můžeme vidět, že na rozdíl od vlhkosti vzduchu či tlaku vzduchu, které by v extrémních



Obrázek 1.5: Tlak vzduchu v závislosti na nadmořské výšce

případech dělaly rozdíl do $\pm 5 \text{ m} \cdot \text{s}^{-1}$, hraje ve výpočtu rychlosti signifikantní roli teplota okolního vzduchu, což by, pokud by s tím výpočet nepočítal, mohlo způsobovat nemalé odchylky v měření. Konkrétní hodnoty pro některé teploty lze dohledat v tabulce (Tabulka 1.1), ale pro přesné měření je i tak nutné měřit aktuální teplotu vzduchu přímo během měření.

1.5.1.5 Vliv média

Na rychlost zvuku má zajisté největší vliv prostředí, ve kterém se vlny pohybují. Zatím jsem uvedl pouze aspekty ovlivňující pohyb zvukových vln ve vzduchu, což je samozřejmě

jediné médium, které se bere v úvahu v tomto projektu, avšak pro úplnost bych rád doplnil i jiná přenosová média a chování vln v nich (Tabulka 1.2).

| Prostředí | rychlost zvuku [ms^{-1}] |
|---|------------------------------|
| Oxid uhličitý ($25^{\circ}C$) | 259,00 |
| Kyslík ($25^{\circ}C$) | 316,00 |
| Suchý vzduch ($0^{\circ}C$) | 331,40 |
| Suchý vzduch ($25^{\circ}C$) | 346,30 |
| Helium ($0^{\circ}C$) | 970,00 |
| Vodík ($25^{\circ}C$) | 1270,00 |
| Destilovaná voda ($25^{\circ}C$) | 1497,00 |
| Mořská voda ¹⁴ ($13^{\circ}C$) | 1500,00 |
| Led ($-4^{\circ}C$) | 3250,00 |
| Sklo ($20^{\circ}C$) | 5200,00 |
| Železo ($20^{\circ}C$) podélně | 5900,00 |
| Železo ($20^{\circ}C$) příčně | 3200,00 |

Tabulka 1.2: Závislost rychlosti vzduchu na prostředí

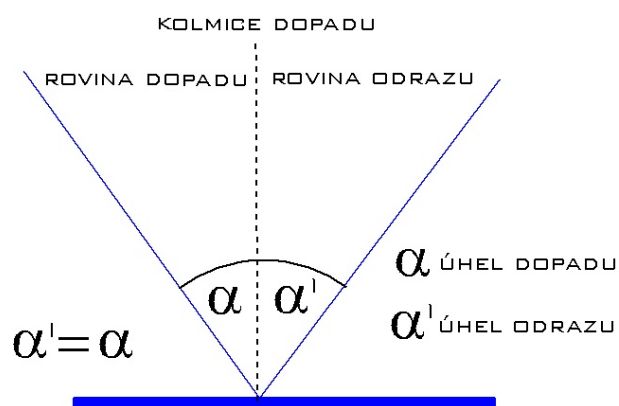
1.5.2 Odraz zvuku

Pokud ultrazvukový signál dopadne na plochu větší, než je velikost vlnové délky signálu, může být tato vlna odražena. Zdalipak se odrazí, záleží na několika aspektech, jako je **úhel** dopadu či **faktor útlumu** daného materiálu. Pokud je tento faktor příliš vysoký, je vlna pohlcena a nic se neodrazí. Je-li faktor útlumu objektu dostatečně malý, je nepohlcená část energie signálu odražena podle zákona odrazu¹⁵ (Obrázek 1.6). V praxi to znamená, že úhel dopadu se rovná úhlu odrazu ke kolmici k povrchu dopadu.

¹⁴ 35 g soli na litr

¹⁵ „Úhel odrazu je roven úhlu dopadu, přičemž odražené paprsky zůstávají v rovině dopadu“

$$\alpha = \alpha'$$



Obrázek 1.6: Zákon odrazu

Vyslaná vlna se tedy rozdělí na dvě, odraženou vlnu a vlnu prostupující materiálem. Velikost těchto vln lze zjistit pomocí tzv. *akustické impedance* obou prostředí. Ta se vypočítá z hustoty přenosového média a rychlosti šíření vzduchu v něm.

$$Z = \rho \cdot c$$

Kde:

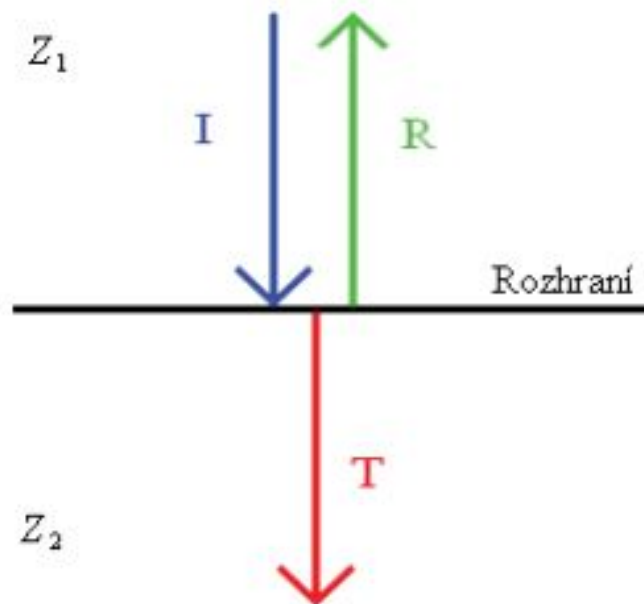
- Z = Akustická impedance
- ρ = Hustota přenosového média
- c = Rychlost zvuku v daném médiu

Pak tedy můžeme spočítat akustickou impedanci pro obě prostředí a vypočítat koeficient odrazu:

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1}$$

a koeficient prostupu

$$T = \frac{2Z_1}{Z_2 + Z_1}$$



Obrázek 1.7: Pohlčení a odraz vlny

Čím blíže se koeficient odrazu blíží jedné, tím přesnějšího měření můžeme dosáhnout, jelikož se neztrácí tolik energie a více je odraženo zpět.

Kapitola 2

Návrh výrobku

Kapitola pojednává o způsobu a principu měření vzdálenosti a jeho technickém provedení. Dále popisuje jednotlivé hardwarové komponenty použité v mém řešení.

Pro sestavení výrobku je nejprve nutné vhodně zvolit jednotlivé komponenty a vypracovat kvalitní návrh. Při navrhování a výběru komponent jsem se zaměřil na jednoduchost, nízkou pořizovací cenu a miniaturizaci. Nakonec jsem se ale rozhodl nepoužít SMD¹ prvků, jelikož velikost přijímače a vysílače ultrazvuku, stejně jako ICSP konektoru² by zůstala stejná a velké úspory místa by se již nedocílilo.

2.1 Měřicí modul

První část technického řešení se týká samotného vysílání a přijímání ultrazvukového signálu, měření času, který uběhl mezi vysláním a příjmem ultrazvukového signálu a poslání této hodnoty pomocí sériového přenosu do řídicí jednotky, která jej zpracuje, vyhodnotí a zobrazí.

Tato část je tedy řešena jako samostatný modul, který bude obsahovat mikrokontrolér, vysílač a přijímač ultrazvukového signálu o frekvenci 40 kHz, zesilovač přijímaného signálu a konektoru pro připojení ICSP programátoru.

Inspirací pro realizaci měřiče, jakožto přípojného modulu, pro mne byl podobný modul (Obrázek 2.1) pro open-source platformu Arduino, kterou jsem se rozhodl použít jako prvek, na kterém bude spuštěn obslužný program modulu a ke které bude modul následně připojen.

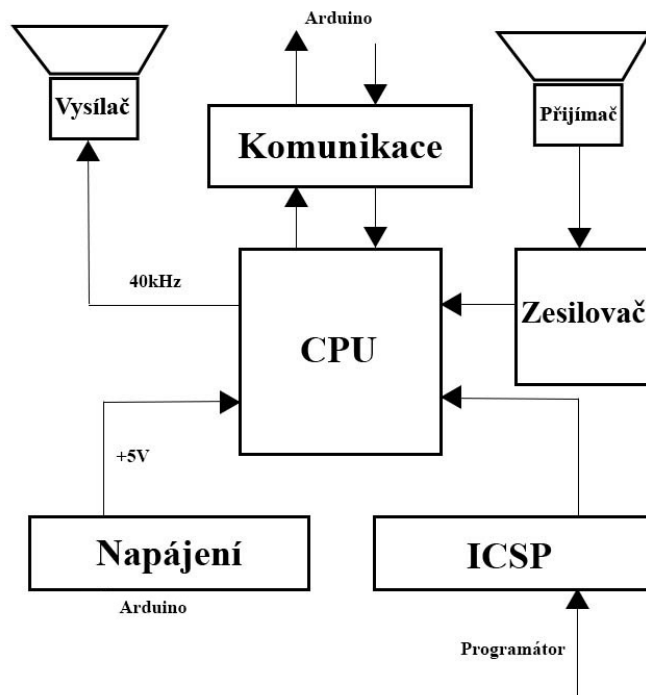
¹ surface mount device - součástky pájené přímo na povrch plošného spoje

² In-Circuit Serial Programming



Obrázek 2.1: Ultrazvukový modul pro Arduino

Toto řešení jsem zvolil z důvodu jeho kompaktnosti, použitelnosti jako zásuvný modul pro jiné mikrokontroléry či složitější systémy a pro jeho malou velikost a nízkou hmotnost i jako prvek pro RC modely nebo roboty. Nevýhodou je, že je samostatně nepoužitelný.



Obrázek 2.2: Blokové schéma modulu

Z blokového schématu (Obrázek 2.2) je patrné, že jsem nepoužil zesilovač pro zesílení vyslaného signálu, ale zvolil jsem elegantnější, nepatrně méně účinnou metodu simulace střídavého napětí, kdy je střídavě na + a - vysílače přiváděno napětí a vysílač se tak dostává i do zpětných zákmitů a celkový výkon je tak zesílen.

Modul funguje na principu otázka/odpověď, kdy je potřeba vyslat dotaz v podobě krátkého impulsu na příslušný pin, než modul vyšle ultrazvukový signál a pošle odpověď. Tím se zamezí nežádoucím rušením způsobeným odrazy zvukových vln. Kontroluje se však jen stav pinu, a tak je možné, že by se mohl modul sám uvést do běhu vlivem okolního rušení.

2.1.1 Výběr komponent pro modul

Pro výsledné zařízení se musí nejprve vybrat vhodné komponenty:

- Mikrokontrolér
- Zesilovač
- Vysílač ultrazvukového signálu
- Přijímač ultrazvukového signálu

Modul musí splňovat tyto požadavky:

- Měření doby mezi odesláním a příjmem signálu
- Sériová komunikace
- Přeprogramovatelnost / Ladění
- Napájení pomocí 5V+
- Spuštění na vyžádání

Známá omezení:

- Použití pouze v místnosti
- Měřitelná vzdálenost max 4m
- Neminiaturizováno (SMD prvky)
- Široký úhel záběru

2.1.1.1 Ultrazvukové vysílače

Ultrazvukové vysílače se používají v mnoha průmyslových odvětvích k různým aplikacím. Vedle jiných jde především o měření nebo detekci výšky hladiny, detekci pozice materiálu ve výrobním procesu, lékařství, klenotnictví či k nedestruktivnímu (NDT) testování materiálů.

Jedna z metod získávání ultrazvukových kmitů je založena na tom, že destičky vyříznuté určitým způsobem z krystalu křemene se při stlačení na povrchu elektrizují³. Jestliže naopak povrch takové destičky periodicky nabíjíme, destička se pod vlivem elektrických nábojů střídavě smršťuje a rozpíná, tj. kmitá. Vznikají ultrazvukové kmity.

Na tomto principu fungují nejrozšířenější vysílače ultrazvuku, tzv. piezoelektrické, ve kterých však z důvodu vysoké ceny je nahrazen krystal křemene syntetickými látkami, např. keramickým *titaničitanem barnatým*.

Díky vlastnostem takovýchto vysílačů je lze použít zároveň jako přijímače ultrazvuku, dohromady se jim potom říká ultrazvukové *měníče*.

Pro své řešení jsem zvolil vysílač **UST-40T**⁴ (Obrázek 2.3) pro jeho nízkou pořizovací cenu a přitom dobrou kvalitu, která dostatečně poslouží k účelům konstrukce konečného výrobku. Výhodou je také jeho malá velikost.

Analogicky k němu existuje přijímač **UST40R**⁵ (Obrázek 2.3), který jsem pochopitelně vybral jako přijímač do svého modulu.

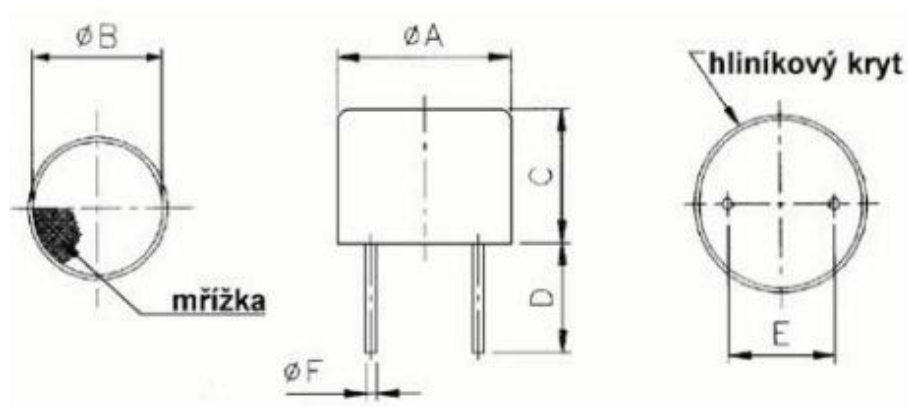


Obrázek 2.3: UST-40T/R

³ Takto vzniklá elektřina se nazývá **piezoelektřina**

⁴ též označován jako 400ST160

⁵ též označován jako 400SR160

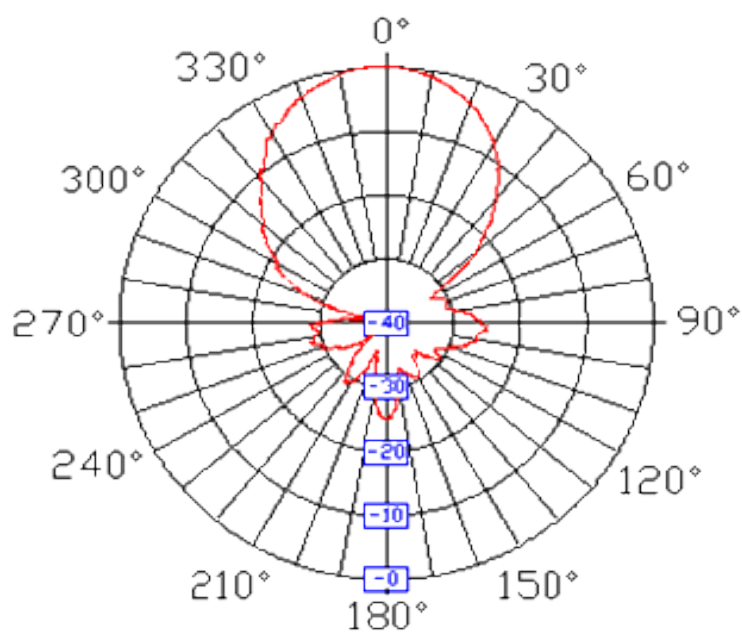


Obrázek 2.4: Schéma ultrazvukového měniče

| UST40T/R | A | B | C | D | E | F |
|--------------|------|------|------|------|------|-----|
| Rozměry [mm] | 16,2 | 13,0 | 12,0 | 10,0 | 10,0 | 1,0 |

Tabulka 2.1: tabulka rozměrů ultrazvukového vysílače a přijímače (Obrázek 2.4)

Mezi nejdůležitější parametry patří nominální frekvence $40kHz$, pracovní teplota -30 až $+80^{\circ}C$, vyzařovací úhel 55° a napájecí napětí $20V$. Důležitý je také dosah šesti metrů (při plném napájení, já však budu uvažovat pouze poloviční výkon), který je možné ještě zvýšit použitím akustických zrcadel za cenu ztráty vyzařovacího úhlu, což by ovšem v tomto případě nebylo na škodu, spíše naopak. Veškeré parametry poskytnuté výrobcem jsou popsány v tabulce 2.2.



Obrázek 2.5: Vyzařovací úhel vysílače UST40T

| | UST40T | UST40R |
|--|---------------|---------------|
| Jmenovitá frekvence [kHz] | 40,0 ± 1,0 | 40,0 ± 1,0 |
| Přijímací citlivost [dB] | ... | -65,0 $min.$ |
| Úroveň tlaku vzduchu [dB] | 119 $min.$ | ... |
| Šířka pásma [kHz] | 2,0 | 2,5 |
| Kapacita [pF] | 2400 | 2400 |
| Pracovní teplota [$^{\circ}C$] | -30 – – + 80 | -30 – – + 80 |
| Skladovací teplota [$^{\circ}C$] | -40 – – + 85 | -40 – – + 85 |
| Maximální napájecí napětí U_{PP} [V] | 20 | ... |

Tabulka 2.2: Technický popis

2.1.1.2 Zesilovač

Zesilovač je elektronické zařízení, které je schopno transformací elektrické energie z vnějšího napájecího zdroje měnit parametry vstupního signálu. Z hlediska elektroniky bývá považován za aktivní dvojbran (nelineární), který je tvořen zesilovacím prvkem a pomocnými obvody zajišťující nastavení a stabilizaci pracovního bodu [3].

Obvykle zesilovač slouží především k zesílení amplitudy signálu nebo úrovně (u stejnosměrných zesilovačů) na požadovanou hodnotu. Používá se ale i v zapojeních, kde je potřeba změnit tvar signálu a jiných.

Je mnoho druhů zesilovačů a dělí se podle jejich funkce či způsobu, jakým fungují.

Pro mé řešení jsem zvolil operační zesilovač **NE5532** od firmy Texas Instruments, se kterým mám dobré osobní zkušenosti a jakožto operační zesilovač se pro výrobek výborně hodí, jelikož má nízký vlastní šum.

2.1.1.3 Mikrokontrolér

Mikrokontrolér je programovatelná elektronická součástka, která má nejčastěji podobu integrovaného obvodu. Mikrokontrolér, někdy rovněž označovaný jako mikropočítač nebo jednočipový mikropočítač, je miniaturní počítač, který je integrován na jediném čipu a který typicky obsahuje procesor (rovněž označovaný jako CPU), paměť, programovatelné vstupně-výstupní rozhraní a další periferní obvody. Mikrokontrolér je vhodný pro použití v řízení a je navržen a určen pro tzv. vestavné aplikace, tj. mikrokontrolér je buď řídicí jednotkou nějakého přístroje nebo je součástí nějakého dalšího zařízení, kde plní určitou specifickou funkci (na rozdíl od běžných počítačů, které jsou určeny k univerzálnímu použití). Mikrokontrolér je proto navržen jako samostatná jednotka schopná komunikace a interakce s okolím.

2.1.1.3.1 CPU

Jádrem a nejdůležitější částí mikrokontroléru je mikroprocesor, obvykle označovaný anglickou zkratkou CPU⁶. Mikroprocesor je zodpovědný za správné načítání, dekódování a vykonávání jednotlivých instrukcí uložených v programové paměti a obecně řídí činnost celého mikrokontroléru.

⁶ Central Processing Unit

Ve spojení s aritmeticko-logickou jednotkou ALU ⁷ vykonává mikroprocesor aritmetické a logické operace, dále řídí čtení a zápis dat do paměti a komunikaci mezi vstupně-výstupními porty a dalšími periferními obvody.

2.1.1.3.2 Paměť

Paměť je dalším nepostradatelným prvkem mikrokontroléru. Paměť je nezbytná jak k uložení vlastního programu v podobě instrukcí, které řídí činnost mikrokontroléru, tak dat, se kterými mikrokontrolér pracuje. Ačkoliv může být paměť programu a dat sdílená a namapována do jediného adresního prostoru (Von Neumannova architektura), u mikrokontrolérů je paměť pro program a data často oddělená (Harvardská architektura).

Programová paměť mikrokontroléru bývá dnes obvykle realizována jako paměť FLASH, kterou lze v případě potřeby snadno přeprogramovat. Jako datovou paměť pro uložení proměnných, výsledků aritmeticko-logických operací a dalších dat využívá mikrokontrolér statickou paměť RAM⁸, u níž informace zaniká s vypnutím napájecího napětí. Mikrokontroléry bývají navíc často vybaveny ještě menší datovou pamětí typu EEPROM, která slouží k zálohování důležitých dat, jelikož je persistentní a přežívá dobu běhu programu.

Kapacita programové paměti bývá přibližně $1kB$ až $256kB$, kapacita datové paměti bývá nižší a pohybuje se mezi $32B$ u nejjednodušších mikrokontrolérů a $16kB$ (a více) u složitých mikrokontrolérů.

2.1.1.3.3 Vstupně-výstupní rozhraní

Vstupně-výstupní rozhraní mikrokontroléru, často označovaná jako I/O⁹ porty, zprostředkovávají komunikaci mezi mikrokontrolérem a externími zařízeními. Přenos dat probíhá skrze vývody integrovaného obvodu, které lze, jak už název napovídá, obvykle programově nastavit jako vstupy nebo výstupy. Na vstupně-výstupní porty mikrokontroléru můžeme přímo připojit např. tlačítka, klávesnici, LED, relé, displej nebo další logické obvody. Protože je počet vývodů mikrokontroléru často velice omezen, některé I/O vývody mohou sdílet funkci s dalšími periferními obvody, např. A/D převodníkem, budičem LCD displeje, sériovým rozhraním, komparátorem apod.

⁷ Arithmetic Logic Unit

⁸ Random-access Memory

⁹ Input/Output

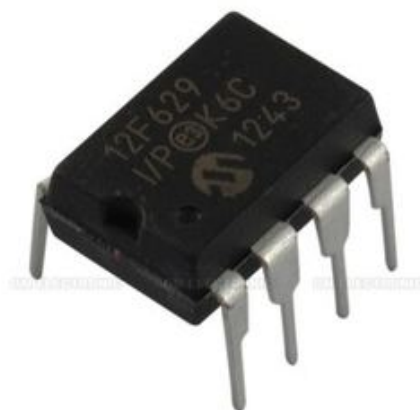
2.1.1.3.4 Obvody čítačů a časovačů

Mikrokontrolér obvykle obsahuje jeden nebo více čítačů / časovačů, které lze programově konfigurovat. Čítače a časovače umožňují programu mikrokontroléru např. měřit časové intervaly, detekovat počet impulsů za určitý časový interval apod.

2.1.1.4 Mikrokontrolér PIC

Mikrokontroléry PIC představují rodinu mikrokontrolérů založených na Harvardské architektuře, které vyrábí firma Microchip Technology. Mikrokontroléry PIC jsou populární mezi amatéry i profesionály zejména díky široké nabídce typů, přijatelné ceně, množství dostupné literatury, snadnému programování, přeprogramování díky paměti FLASH, sériovému programovacímu rozhraní a zdarma dostupnému softwarovému vývojovému prostředí [6].

Pro realizaci modulu pro měření doby odrazu jsem si vybral mikrokontrolér PIC12F629. Jedná se o mikrokontrolér s šesti I/O porty, dvěma časovači, analogovými komparátory a přesným interním oscilátorem ($4MHz + -1\%$) a délkou instrukčního cyklu 1μ [5].



Obrázek 2.6: Pic12f629

2.1.2 Obsluha modulu

Jak již bylo zmíněno (Kapitola 2.1), k ovládání a obsluze modulu bude zapotřebí dalšího kontroléru, který tyto funkce obstará. Jelikož jsem kladl důraz na to, aby byl modul co možná nejmenší (bez využití SMD součástek), tak se výborně hodí třeba pro pojízdného

robota, který by ultrazvuku využíval k vyhýbání se překážkám.

Z tohoto důvodu jsem se rozhodl využít platformu **Arduino** od firmy *Arduino* a to konkrétně typ **UNO**. Jedná se o *open-source* platformu založenou na mikrokontrolerech **ATMega** od firmy *Atmel* společně s grafický vývojovým prostředím, které vychází z prostředí *Wiring*¹⁰.

2.1.2.1 Historie Arduina

Projekt vznikl v roce 2005 v *Itálii*. Cílem bylo vytvořit jednoduchou prototypovací platformu pro studenty, která umožní rychlý vývoj a jednoduché používání. Projekt ihned zaznamenal úspěch, a tak na přání zákazníků začaly vznikat další verze původní platformy doplněné o různé komponenty nebo naopak odlehčené.

Projekt byl již od svého počátku volně dostupný (*open-source*) všem uživatelům, kteří jej chtějí používat či vylepšovat. Dokumentace a referenční příručka jazyka a externích knihoven je pak většinou vydávána pod licencí *Creative Commons*. Díky tomu jsou k dispozici všechny zdrojové soubory a je dodržována určitá kompatibilita (ať už pomocí jazyka Java, ve kterém je napsáno vývojové prostředí (IDE) nebo díky použití USB-to-RS232 převodníkového čipu, který zajišťuje virtuální sériovou linku a pro který jsou běžně dostupné ovladače na většinu platforem). [9]

Arduino Uno, které bylo použito v tomto projektu vyšlo roku 2010, kdy nahradila svého předchůdce typu **Duemilanove**.

2.1.2.2 Arduino Uno

Arduino Uno je mikrokontrolérová deska založená na čipu **ATMega328**. Tato verze poskytuje čtrnáct vstupně/výstupních digitálních pinů (I/O), ze kterých může být šest využito pro výstup PWM. Šest analogových vstupů, 16MHz keramický resonátor, USB konektor pro připojení k PC, napájecí konektor na 5-12V, ICSP hlavičce, programovatelnou LED diodu a resetovací tlačítko.

¹⁰ obdobný projekt jako Arduino

Uno se liší od všech svých předchůdců tím, že nepoužívá FTDI USB-to-serial driver čip. Místo toho představuje čip ATmega16U2, naprogramovaný jako převaděč USB na sériovou komunikaci.

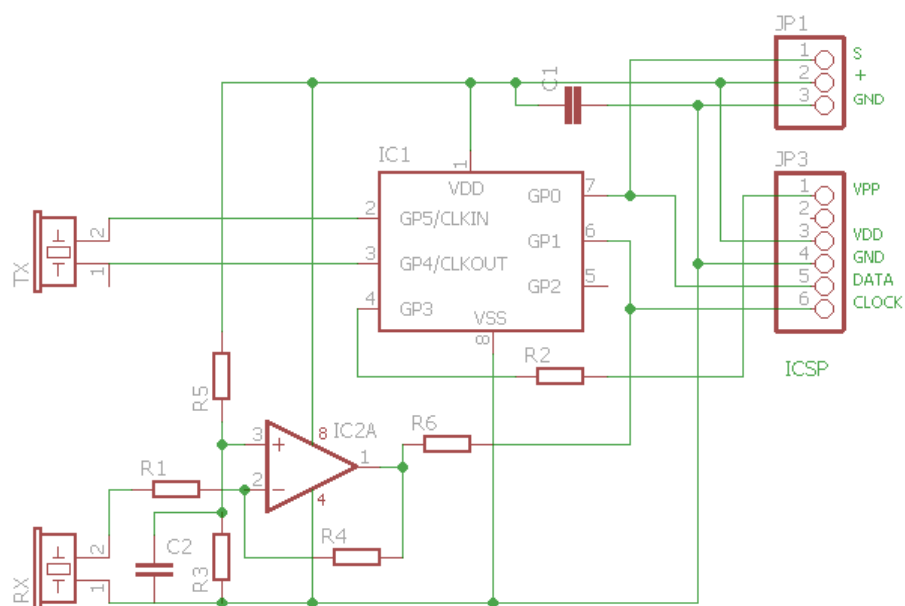
| | |
|---------------------------------|--|
| Mikrokontrolér | ATMega328 |
| Operační napětí | 5V |
| Vstupní napětí | 7 – 12V |
| Vstupní napětí - limitní | 6 – 20V |
| Digitální I/O piny | 14 (6 poskytuje PWM výstup) |
| Analogové I/O piny | 6 |
| Flash paměť | 32KB (z toho 0,5 KB je využito bootloaderem) |
| SRAM | 2KB |
| EEPROM | 1KB |
| Takt | 16MHz |
| Délka/šířka/hmotnost | 68,6mm/53,4mm/25g |

Tabulka 2.3: Přehled specifikací Arduina UNO [1]

Kapitola 3

Realizace

Další fází je samotná realizace, ke které bylo nutné nejprve vytvořit elektrické schéma zapojení (Obrázek 3.1) a to následně převést na desku plošného spoje. Tuto desku poté vyrobit, osadit a přivést k životu.



Obrázek 3.1: Elektrické schéma modulu

3.1 Deska plošného spoje

Deska plošného spoje (*DPS*) byla navržena v programu *Eagle*, který je ve své základní verzi zcela zdarma pro menší projekty jako je tento. Z důvodu domácí výroby *DPS* a nízkého počtu použitých komponent jsem uvažoval pouze jednovrstvou desku bez prokovů. Program *Eagle* umožňuje po rozvržení pozic součástek automaticky navrhnout cesty spojů, této funkce jsem však nevyužil a to opět z důvodu nízké náročnosti obvodu. Tento způsob jsem sice odzkoušel, ale výsledek byl méně kompaktní, než mnou navržený způsob spojů.

3.1.1 Popis výroby DPS

Výroba plošných spojů je technologický proces náročný na přesnost, čistotu, vybavení a zkušenost. Avšak existují způsoby, jak si vyrobit plošný spoj, slušné kvality, i doma. Já jsem si zvolil metodu výroby fotocestou. [4]

Nejprve bylo potřeba vystříhnout z jednostranného **kuprexitu**¹ obdelník, o něco větší než je velikost konečné desky (asi 5mm na každé straně). Ke stříhání **kuprexitu** jsem použil pákové nůžky. Vyčnívající kousíčky mědi na okrajích jsem lehce srazil smirkovým papírem, což je důležité, aby později nenadzdvihávaly předlohu. Jakmile byl **kuprexit** vystřížený, bylo nutné měď velice důkladně očistit, tento proces je klíčový, jelikož každá nečistota (oxidovaná měď, mastnota) může ohrozit celý proces výroby. Měď jsem proto čistil jak mechanicky (Solvina), tak chemicky (aceton). Po očištění jsem desku důkladně opláchl a usušil, čímž jsem jí připravil k dalšímu kroku výroby.

Na očištěnou desku jsem nanesl fotocitlivý² lak (Positiv 20). Při tomto procesu je nutné, aby deska byla o něco větší, než je předloha, jelikož nanesený lak vytvoří dokonale rovnou vrstvu, ale na krajích desky se mírně nakupí a jeho vrstva je zde tlustší. Poté bylo nutné lak nechat takzvaně vytvrdit, což vlastně znamená nechat ho zaschnout při pokojové teplotě alespoň 24 hodin. *Positiv 20* funguje tak, že se rozpustí ve vývojce, pouze pokud byl předtím osvětlen UV zářením. I tak je však dobré nechat nalakovanou desku na tmavém místě, jelikož by mohl reagovat i s obyčejným viditelným světlem.

Během vytvrzování jsem vyrobil předlohu. Tu jsem navrhl v programu *Eagle* a vytiskl ji na čirou fólii. Důležité je, aby černá barva byla co nejsytější, k čemuž se lépe hodí inkous-

¹ sklolaminát po jedné nebo po obou stranách plátovaný měděnou fólií

² mění své vlastnosti po vystavení světelnému záření

tová tiskárna než laserová, jelikož pro tisk používá více barvy. Návrh je nutné vytisknout horizontálně převrácený, jelikož se na desku bude přikládat stranou s potiskem.

Poté bylo již vše připraveno na další fázi, kterou bylo osvětlení desky. To se provádí tak, že se přiloží předloha na desku (potiskem dolů, aby nedocházelo k podsvitu) a zatíží se tabulkou čirého skla, aby se během osvětlování nepohnula a dobře doléhala. Poté se nad takto připravenou desku rozsvítí výbojka (UV světlo). Mně se nejvíce osvědčilo nechat výbojku (rtuťovou) zapnutou 5 minut ve výšce asi 20 centimetrů nad povrchem desky.

Po osvětlení se musela deska vyvolat. Vyvolávání je proces, kdy se do 1 – 2% roztoku NaOH (hydroxid/louh sodný) ponoří osvětlená deska (nalakovanou stranou vzhůru). Vyvolávání způsobí, že se osvětlený lak rozpustí a na desce zůstane lak pouze na místech, kde předloha nebyla prosvícena, čili v místech spojů. Doba vyvolání je asi 2 minuty. Po tomto procesu se opět musela deska důkladně omýt a usušit. Poté již bylo možné přejít k poslední části výroby, kterou je leptání.

Leptání je proces, při kterém se povrch mědi, zbavený laku, odleptá a zůstane pouze sklolaminát []. Jako leptadlo jsem použil chlorid železity. Nejprve jsem jeho roztokem potřel povrch desky (předejítí nechtěným bublinkám, které by se mohly vytvořit) a poté jsem desku položil na jeho hladinu tak, aby plula na hladině (roztok má mnohem větší hustotu než voda) měděnou stranou dolů. Tím se docílí samovolného odpadávání rozleptané mědi a proces se tak dá lehce kontrolovat. Jakmile byla všechna nenalakovaná místa zbavena mědi, bylo možné desku z lázně vyjmout, důkladně omýt a usušit. Poté jsem povrch se spoji natřel pájivým lakem (pryskyřice rozpuštěná v acetonu) a deska plošného spoje byla hotová a připravená k osazení součástkami.

3.1.2 Osazení

Desku jsem osazoval klasickými (ne SMD součástkami). Nejprve jsem osadil rezistory a kondenzátory, jelikož jsou nejmenší a později by se špatně vměstnávaly mezi větší a vyšší součástky. Následovalo osazení patice pro PIC společně se zesilovačem. Dále jsem osadil ultrazvukové měniče a nakonec hlavici ICSP a I/O piny. Než jsem osadil procesor, překontroloval jsem veškeré spoje vedoucí do patice, zda nedošlo k nějakým nechtěným propojením. Vše se zdálo být v pořádku, mohl jsem tedy do patice vložit mikrokontrolér.

3.1.3 Rozpis součástek

| | |
|------------------------|---------------------|
| Rezistory - R1, R2, R6 | 1k |
| Rezistory - R3, R5 | 100k |
| Rezistor - R4 | 750k |
| Kondenzátory - C1, C2 | 100n |
| TX | UST-40T |
| RX | UST-40R |
| IC1 | PIC12F629 |
| IC2 | NE5532 |
| Konektor JP1 | Lišta 3x pin |
| Konektor JP3 | ICSP (lišta 6x pin) |

Tabulka 3.1: Rozpis součástek

3.2 Program modulu

Program pro mikrokontrolér **PIC12F629** jsem vyvíjel nejprve v prostředí *MPLAB IDE* od firmy *Microchip Technology Inc.*, avšak následně jsem toto prostředí opustil, a to z důvodu absence simulátoru. Přešel jsem do vývojového prostředí programu *Proteus 8*, u kterého jsem později zjistil, že obsahuje prostředí pro návrh plošných spojů, což již bylo pro mou práci irelevantní. V obou prostředích lze programovat buďto v jazyce **C** či **Assembleru**. Zvolil jsem jazyk **C**, který je pro mě známější a přehlednější. Převedení kódu na **assembler** jsem pak využíval k optimalizaci časování při generování kmitočtu pro ultrazvukové měniče.

K překladu jsem využil volně dostupnou verzi kompilátoru *MPLab XC8*, která se od placené verze liší hlavně slabší optimalizací výsledného kódu a nemožností kompilovat programy s výslednou velikostí přesahující 1KB, což mě nijak neomezilo, jelikož velikost paměti mikrokontroléru PIC12F629 je právě 1kB.

Program je složen z několika funkcí, hlavního těla programu, definice proměnných a pinů a obsluhy přerušení, která je v tomto projektu klíčová. Program vyčkává, dokud nedostane impuls na příslušný pin, poté provede měření a po stejném pinu, na který dříve přišel impuls,

pošle odpověď obsahující naměřený čas. Jedná se o half-duplexovou komunikaci.

3.2.1 Inicializace

Co je nutné ze všeho nejdřív, je inicializace mikrokontroléru. Je nutné připojit k projektu knihovnu kompilátoru XC8 a definovat správný typ mikrokontroléru, v tomto případě **PIC12F629**. To musí být následováno definováním proměnné, která obsahuje hodnotu kmitočtu (tohle není bezvýhradně nutné pro běh programu, ale podle tohoto čísla jsou odvozené časy zpoždění pomocí funkcí...delay). Následuje nastavení konfiguračních bitů, kde se mimo jiné vybírá druh oscilátoru použitého mikrokontrolérem (v tomto projektu je použit vnitřní oscilátor³). Důležité je také vypnout watchdog⁴.

3.2.2 Konfigurace

Tato část obnáší nastavení pinů na jejich požadovanou funkci, definici proměnných pro přístup ke stavům těchto pinů.

Bylo také nutné nastavit komparátory na příslušnou hladinu a povolit jim možnost vyvolání přerušení, softwarově pro ně nastavit jiné referenční napětí a nakonfigurovat časovač na nejmenší možnou měřitelnou hodnotu ($1\mu s$). Ostatní nastavení mohla zůstat na svých defaultních hodnotách.

3.2.3 Funkce main

main()

Funkce nejprve zavolá funkci pro inicializaci proměnných a registrů, poté zkušebně provede jedno měření bez odeslání naměřených dat a poté skočí do nekonečné smyčky, kde čeká na příchozí signál z **Arduina**, který by znamenal začátek měření. Je-li takový signál přijmut, provede se měření a po jeho dokončení jsou výsledky odeslány sériovou komunikací **Arduinu** k dalšímu zpracování.

³ Hodnota INTOSC

⁴ Počítačová periferie, která resetuje systém při jeho zaseknutí

3.2.4 Funkce přerušní

interrupt_isr()

Funkce obsluhy přerušeni je krátký úsek kódu, který rozhodne, o jaké přerušeni se jedná (zda přerušeni komparátorem či časovačem), vždy vypne časovač a smaže příslušný příznak přerušeni. Pokud se jednalo o přerušeni způsobené časovačem, pak došlo k jeho přetečení a nastaví se příslušná proměnná, jednalo-li se o přerušeni v důsledku kladného vyhodnocení komparátorů, je zavolána funkce pro zpracování detekce signálu.

3.2.5 Funkce detekce signálu

sigDetect()

Jedná se pouze o uložení příslušných bytů obsahujících dobu běhu časovače do proměnných, nastavení příznaku detekce signálu, vynulování časovače a vypnutí funkce přerušeni pro komparátory.

3.2.6 Generování kmitočtu

usGen()

Nejprve se vypnou reakce na všechna přerušeni, aby nezasáhly do procesu, jelikož generování kmitočtu 40kHz musí být co možná nejpřesnější a každá μs hraje významnou roli. Poté je nastaven počet kmitů, které se mají vygenerovat (v mém případě 20 zákmitů). Proveďte se spuštění časovače následované generováním samotných vln. Zde bylo nutné počítat s časy provádění jednotlivých instrukcí, aby bylo možné přesně generovat potřebný kmitočet. Generování probíhá na principu střídavého zapínání a vypínání napětí na výstupech dvou pinů (vždy protichůdně), a tudíž simulaci střídavého napětí vedeného do ultrazvukových měničů.

```

1  int waves = 20;
2  while( waves > 0)    // 4us
3  {
4                      // 1 Period is 25us
5                      // 20 Periods is 0,5ms
6      UTXA = 0;        // 1us
7      UTXB = 1;        // 1us
8
9      __delay_us(10);  // 10us
10
11     UTXB = 0;        // 1us
12     UTXA = 1;        // 1us
13
14     __delay_us(3);   // 3us
15
16     waves--;         // 1us
17 }                    // 3us

```

Než se povolí přerušení pro komparátory, je nutné nějakou chvíli počkat, aby přijímač nezaznamenal vysílané vlny, ale až odražené. Nastaveno na přibližně *20cm*, čímž vzniká omezení dolní hranice měřitelnosti vzdálenosti. Horní hranice je omezena přetečením časovače a odpovídá asi deseti metrům, tato vzdálenost se však již nedá přístrojem reálně změřit z důvodu malého výkonu měničů.

3.2.7 Funkce posláání dat

send()

Funkce nejprve vyhodnotí, zda časové hodnoty jsou validní či zda došlo k přetečení časovače, a následně se podle toho rozhodne, co poslat.

Zdají-li se být data v pořádku, jsou jednotlivé byty obaleny primitivním protokolem skládajícím se ze čtyř kontrolních znaků a jsou posláány pomocí funkce pro zprostředkování sériového přenosu sériovou linkou k dalšímu zpracování.

Přetekl-li časovač je odeslána příslušná zpráva.

3.2.8 Funkce zprostředkování sériového přenosu

usartTX(byte)

Jelikož mikrokontrolér **PIC12F629** nedisponuje integrovaným modulem typu **USART** pro přenos dat pomocí sériové linky, byl jsem nucen toto obejít softwarově.

Základní logika je taková, že pin se stále udržuje v logické jedničce a jeho změna z 1 na 0 se považuje za startovní bit přenosu, analogicky pak z 0 na 1 jako ukončovací bit přenosu. Pro jednoduchost se nevyužívá paritního bitu ani kontroly toku. Odesílá se vždy 8 bitů dat, nebo-li jeden byte a to v pořadí od nejméně významného bitu.

Je proto nutné nastavit kmitočet přenosu u obou zařízení (přijímače i vysílače) na stejnou hodnotu. Tomuto kmitočtu se říká *baudrate* a udává, jak dlouho pin musí setrvat v příslušné poloze k úspěšnému odeslání jednoho bitu. Takto se pošle všech osm bitů a přenos se ukončí.

Po této činnosti vše přejde do počátečního stavu a modul je připraven na nové měření.

```
1  USART = 0;                               // Pin state bit
2  __delay_us(One_Bit_Delay);                // Delay computed from baudrate
3
4  for ( char i = 0; i < Data_Bit_Count; i++ )
5  {
6      //Set Data pin according to the DataValue
7      if( ((data_value>>i)&0x1) == 0x1 )    //if Bit is high
8      {
9          USART = 1;
10     }
11     else                                    //if Bit is low
12     {
13         USART = 0;
14     }
15     __delay_us(One_Bit_Delay);
16 }
17
18 //Send Stop Bit
19 USART = 1;                                // Return to idle state
20 __delay_us(One_Bit_Delay);
```


3.2.9 Metriky kódu

Zdrojový kód se skládá ze souborů `main.c` a `main.h`

Počty řádků kódu jednotlivých souborů jsou 201 a 47

Velikost hex souboru je 1,36kB

3.3 Obslužný program

Obslužný program modulu je psaný v jazyce *Arduino*, který se patrně nejvíce blíží jazyku *C++*. Modul pro měření vzdálenosti byl navržen tak, že pouze měří dobu uplynulou mezi vysláním a přijmutím ultrazvukových impulsů, které odešle. Proto je nezbytně nutné tato data dále zpracovat. A tak, chceme-li získat nějaká srozumitelná data, je tento program společně s dalšími periferiemi **Arduina** stěžejní.

3.3.1 Funkce programu

Naprogramované **Arduino** musí být schopné přijmout data vyslaná pomocí sériového přenosu z měřicího modulu, tato data na základě vytvořeného protokolu rozšifrovat a dále je zpracovat. Dalším zpracováním je myšleno spojení dvou bytů v jedno číslo (sériový přenos je omezen na přenos jednoho bytu), vypočítat aktuální rychlost zvuku díky poskytnutým údajům z periférií a pomocí této vypočítané rychlosti zvuku a času získaného z modulu, provést výpočet vzdálenosti a rychlosti⁵. Poté vše sériovým přenosem poslat k zobrazení na počítač nebo jiné zobrazovací rozhraní.

3.3.2 Nastavení

Kromě vytvoření nezbytných proměnných a konstant je nutné v programu **Arduina** nastavit patřičné piny dle potřeby na vstupní či výstupní, aktivovat periferie (infračervený přijímač, teploměr/vlhkoměr), nastavit kmitočet (baudrate) sériového přenosu totožný s kmitočtem nastaveným v modulu a spustit smyčku obslužného programu.

⁵ Na výpočet rychlosti jsou nutné alespoň dvě měření

3.3.3 Příjem sériové komunikace

Tento problém na straně Arduina nebylo nutné řešit, jelikož Arduino disponuje vlastním sériovým portem USART a to jak pro příjem, tak vysílání. Využití těchto portů však přináší i komplikace v podobě nemožnosti připojit modul na jiný port, tento problém se dá lehce vyřešit knihovnou pro arduino, která softwarově řeší takovýto přenos či tuto funkci v zjednodušené verzi doprogramovat, jak tomu bylo učiněno v mém programu pro Pic.

3.3.4 Měření teploty

Jak již bylo zmíněno v sekci o faktorech ovlivňujících rychlost zvuku (Kapitola 1.5.1), je teplota hlavním a důležitým faktorem ovlivňujícím rychlost zvuku v plynech a je tedy nutné s ní počítat. K tomu v mém projektu slouží obvod pro měření teploty a vlhkosti pomocí senzoru **DHT11**. DHT11 je samostatné digitální čidlo, které pracuje pouze po jednom vodiči (vlastní protokol 1-wire – TTL 5V). Přesnost čidla je běžná pro běžné použití, avšak má poměrně dlouhou dobu chlazení, kdy z vyšší teploty na nižší senzor reaguje velice pomalu. Tento problém by neměl vadit, jelikož v prostředí provozu by neměla teplota kolísat. Avšak pro všechny případy je program vybaven možností ignorovat data z teploměru a použít pro měření buďto data defaultní, či ručně nastavená.

Funkci měření vlhkosti senzoru DHT11 jsem nevyužil, jelikož to ve výpočtu není nutné, jak již bylo objasněno v kapitole 1.5.1.3.

3.3.5 Infračervený přijímač

Arduino jsem vybavil také infračerveným přijímačem, který mi poskytl možnost celý výrobek ovládat pomocí dálkového ovladače. Jedná o modul od firmy *Keyes*, využívající infračervený přijímač (opto-elektrický člen) *AX-1838SH*, který je uzpůsoben pro příjem signálů z dálkových ovladačů. Modul kromě přijímače obsahuje pouze rezistor a červenou signalizační LED diodu.

K obsluze infračerveného senzoru jsem využil open-source knihovnu poskytnutou výrobcem. Poté jsem musel vyčíst a zaznamenat příslušné kódy tlačítek ovladače, který jsem se rozhodl použít pro ovládání výrobku⁶.

⁶ Tyto kódy jsou zapsány v knihovně *KeyesGrub.h*

3.3.6 Výpočet a měření vzdálenosti

Výpočet vzdálenosti je již snadný, pomocí vzorce odvozeném v kapitole 1.5 je nutné nejprve spočítat aktuální rychlost zvuku. Jelikož jediná měnící se veličina ve vzorci je teplota, bylo možné uložit zbytek výpočtu jako konstantu, která se pouze příslušným způsobem upraví. Tím se vypočítá rychlost zvuku v daném prostředí a ta se aplikuje na hodnotu času dříve získanou z modulu. Čas je získán ve formě dvou bytů, které je nutné nejprve převést na jedno číslo, to je poté nutné vydělit dvěma (vlna urazila za tuto dobu cestu k překážce i zpět) a s tímto číslem dále pracovat.

Při výpočtu vzdálenosti je vše provedeno třikrát a až při třetím měření jsou provedeny výpočty.

Měření se spustí tlačítkem **OK** na dálkovém ovládní v době, kdy neprobíhá jiné měření nebo uživatel není v nabídce nastavení.

Zda-li se společně se vzdáleností bude měřit i rychlost či bude při měření využit připojený teploměr, se nastaví v uživatelské nabídce 3.3.7.

3.3.7 Uživatelské nastavení

Z důvodu větší flexibility řešení jsem přidal možnost měnit nastavení za běhu programu. Toto nastavení je uloženo do paměti EEPROM a při spuštění je znovu načteno. Do nabídky nastavení se lze dostat zmáčknutím tlačítka „#“ na dálkovém ovládní v době, kdy neprobíhá měření.

3.3.8 Metriky kódu

Zdrojový kód se skládá ze souborů zdrojovy_kod.ino (main.cpp a main.h), DHT.cpp, DHT.h, IRremote.cpp, IRremote.h, IRremoteInt.h a KeyesGrub.h

Počty řádků kódu jednotlivých souborů jsou 484, 109, 25, 1025, 118, 464 a 27

Velikost hex souboru je 35kB

3.4 Zobrazení dat

Pro zobrazení dat je nutné připojit **Arduino** k počítači a spustit program pro zobrazení sériového přenosu na sběrnici **COM**. **Arduino** je vybaveno vlastním převodníkem COM na USB, který je ke komunikaci nezbytný. Program *Arduino IDE* je naopak vybaven *sériovým monitorem* pro monitorování komunikace mezi **Arduinem** a počítačem. Avšak z důvodu značné velikosti programu a nutnosti jeho instalace jsem se rozhodl vytvořit jednoduchou aplikaci sám.

Jedná se o jednoduchý program (COMListener.exe) v jazyce C#, který umožňuje odposlouchávat komunikaci na příslušném portu **COM** a vypsát ji do *TextBoxu*. Program je použitelný nejen pro tento projekt, ale pro veškerou sériovou komunikaci na **COM** portech, díky nastavitelnosti příslušného **COM** portu, baudratu a počtu datový bitů.

Kapitola 4

Testování

4.1 Hardware

Po naprogramování a sestavení příslušných komponent se mohlo přejít k samotnému testování. Jak už tomu bývá, nic nefunguje napoprvé a tak po zapnutí **Arduina** nepřicházela žádná data. Po změření napětí na modulu jsem zjistil, že z nějakého důvodu byl na několika místech obvod přerušen. Na desce vznikly v mědi miniaturní trhlinky, přes které proud neprošel. Bylo to způsobeno vadou na tiskárně, kdy válec v jedné lince nenanášel černou barvu, tudíž se tato místa při prosvěcování prosvítily a následně odleptaly. Tento problém jsem vyřešil pokrytím všech měděných spojů cínem a přelakováním.

Po vyřešení tohoto problému jsem znovu zapojil celý obvod a konečně jsem se dočkal nějakých dat, která však nebyla úplně korektní a někdy se na vstupu objevila rušení, která zkreslovala výsledky. To mě přinutilo posílat data v jednoduchém protokolu a náhodné ruchy na sériové lince tak ignorovat.

Po nějakých korekcích v softwaru **Arduina** se zdálo být vše v pořádku a tak jsem výrobek na nějaký čas odložil, jako téměř plně funkční. V den, kdy jsem jej chtěl dokončit, se však z nějakého důvodu nechtěla rozběhnout komunikace **Arduina** s počítačem, samozřejmě jsem napřed hledal problém v mém kódu a po několika hodinách marného snažení, jsem úsilí v beznaději vzdal. Na druhý den vše opět fungovalo, ukázalo se, že **Arduino** nefunguje korektně při zapojení do **USB** s funkcí rychlého nabíjení v takzvaném „*Always on*“ módu, což by se dalo přeložit jako mód stálého zapnutí.

Poslední hardwarový problém vznikl několik dní před odevzdáním, kdy přestalo fungovat dálkové ovládání. Vše bylo způsobeno vadnými vodiči mezi **Arduinem** a infračerveným

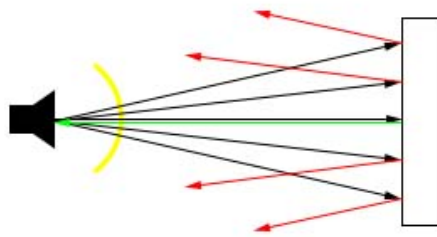
senzorem, které ve spojích nedoléhaly, a po jejich výměně, byl problém vyřešen.

4.2 Prostředí

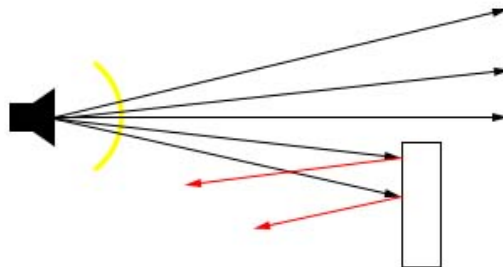
Díky charakteristice použitých ultrazvukových měničů a samotného šíření zvuku vznikají při měření vzdálenosti touto metodou některá omezení, se kterými je nutné při měření počítat. Rozdělil jsem je do dvou hlavních kategorií, a to omezení způsobená odrazem, kdy závisí na poloze a natočení měřeného předmětu, a omezení způsobené tlumící schopností měřeného předmětu.

4.2.1 Omezení způsobená odrazem

Je nutné si uvědomit, že princip měření spočívá ve vyslání a následném přijmutí zvukové vlny, tudíž se vlna musí odrazit zpět k místu svého vzniku. Z toho plynou jistá omezení.



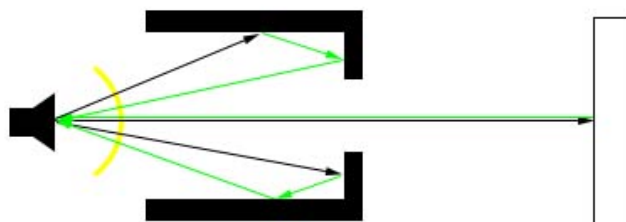
Obrázek 4.1: Ideální odraz



Obrázek 4.2: Neměřitelný stav

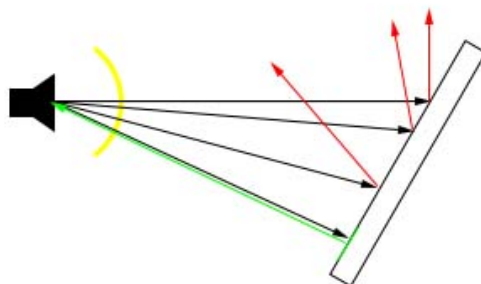
Jak je vidět na obrázku 4.1, vlnění dopadající kolmo na překážku je odraženo zpět (**zelená čára**), kde může být zaznamenáno. Ostatní odražené vlny (**červené**) se odrazí mimo oblast měniče a zaznamenány být nemohou. Toto je ideální případ odrazu, kdy v cestě není žádná překážka.

Obrázek 4.2 pak znázorňuje případ, kdy žádná část měřeného objektu není kolmá ke zdroji vlnění a všechny se tím pádem odrazí mimo něj.



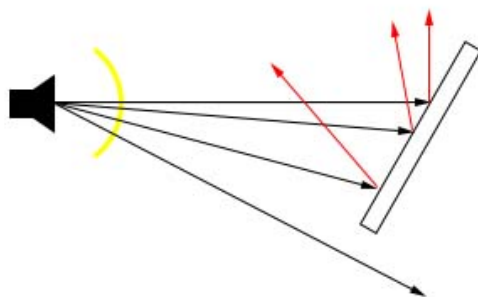
Obrázek 4.3: Dvojitý odraz

Zde (4.3) vidíme problém dvojitého odrazu¹, kdy se vlnění odrazí od dvou překážek a vrátí se ke zdroji dříve než vlnění, které chceme, a je tedy zaznamenáno dříve.



Obrázek 4.4: Odraz od šikmé plochy

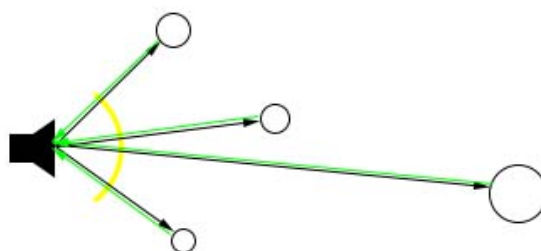
¹ počet odrazů může být větší, až po hranici, kdy by výsledná cesta byla delší než cesta validního odrazu či než by vlnění vlivem pohlcení zaniklo.



Obrázek 4.5: Odkloněný odraz

Na dalším obrázku (4.4) je ukázán případ odrazu od šikmé plochy. Vlnění se sice odrazí, ale takovéto měření je zkresleno, jelikož vzdálenost není měřena od místa, kam měřič míří, nýbrž od místa, které je kolmé k vyslaným vlnám, což hlavně u větších vzdáleností může hrát ohromnou roli.

Obrázek 4.5 popisuje podobný případ, kde však není předmět schopen odrazit zpět žádnou část vlnění a je tudíž pro měřič neviditelný.



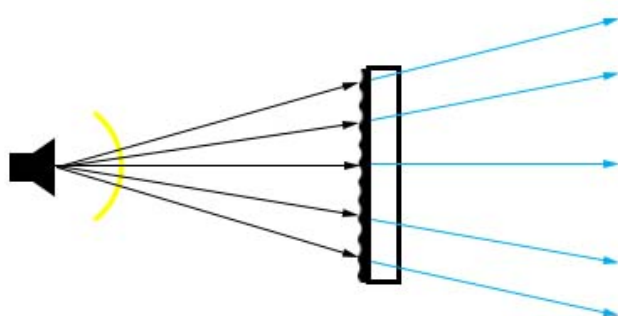
Obrázek 4.6: Odraz od kulových ploch

Asi největším problémem při měření jsou kulovité plochy (Obrázek 4.6), kdy, jsou-li dokonale kulaté, z odrazivého materiálu a ve vyzařovacím úhlu měniče, vždy odrazí vlnění zpět a je tím pádem zaznamenána nejbližší z nich.

4.2.2 Omezení způsobená tlumením

Je známo, že některé materiály pohlcují různá vlnění lépe než ostatní, to samé platí i o zvuku, každý známý materiál má jistý faktor útlumu, závisí však na jeho velikosti, zda bude

na objekt dopadající vlna zcela či jen částečně pohlcena. Nepohlcená část vlnění je poté odražena dle zákona odrazu (kapitola 1.5.2). Pohlcená část vlnění pokračuje skrze materiál, kde může projít nebo být pohlcena úplně. Mezi mnou testované nejtlumivější materiály patří vata, vlna a vesměs všechny textilie. Naopak nejlépe zvuk odráželo dřevo, sklo, zeď a kovy.



Obrázek 4.7: Pohlcení zvukových vln

4.3 Zvýšení efektivity

Efektivita měřiče by se dala v mnoha ohledech vylepšit, hlavně co se týče dosahu, omezení rušení a detekce objektů z materiálů pohlcujících zvukové vlnění.

4.3.1 Zvýšení dosahu

Aktivní dosah měřicího modulu by se dal vylepšit. Největší vliv na to má kvalita ultrazvukových měničů a jejich výkon. S kvalitnějšími měřiči by se dalo dosáhnout mnohem lepších výsledků, avšak ani profesionální ultrazvukové měřiče vzdálenosti pro domácí použití nepřesahují měřitelnou vzdálenost šestnácti metrů a to z důvody vysoké ceny výkonějších měničů a jejich větší velikosti.

Výkon mnou použitých měničů by se pak dal zvýšit i použitím zesilovače před vysílačem, kdy v mém případě využívám pouze jejich polovičního výkonu. Rozdíl by však nebyl až tak markantní, jelikož výkon v tomto případě není přímo úměrný příkonu.

Velkou roli pak hraje také výběr cíle měření, jak bylo popsáno v kapitole 4.2.

Lepší a výkonější měniče pak mohou být schopné vyslat vlnění dostatečně silné na to, aby mohla být jeho část odražena i materiály s vysokým faktorem útlumu.

4.3.2 Rušení

Rušením je myšleno zejména detekce jiného objektu, než chce uživatel měřit. Opět největší vliv na tento jev má kvalita ultrazvukových měničů, v tomto případě však hlavně jejich směrová charakteristika. Čím menší je vyzařovací a přijímací úhel měničů, tím přesněji se dá měřený objekt zaměřit.

Další způsob vylepšení je samozřejmě lepší software, který by pracoval s velikostmi útlumu a dokázal by tak rozeznat počet odrazů vlnění od předmětů a nebral by v potaz některá, již velice utlumená echa, která se stále odráží v prostoru. K tomu je však nutné využít kvalitnějších A/D převodníků a komparátorů.

Kapitola 5

Závěr

Prací na tomto projektu jsem si velice rozšířil obzory co se týče práce s hardwarem. Do této doby jsem se takovéto práci spíše vyhýbal. V praxi jsem si také vyzkoušel práci s ultrazvukem a ověřil jsem si zákony akustiky. Práce s ultrazvukem a zvukem celkově není nikterak jednoduchá, jelikož zvuk je něco, co se nedá uchopit ani vidět, ultrazvuk pak dokonce něco, co se běžně nedá ani slyšet, takže jsem se celou dobu musel spoléhat na měřicí přístroje.

Z principu šíření zvuku pak byl největší problém jeho šíření prostorem, které komplikovalo měření a testování. Přístroj ani program nejsou dokonalé, a tak stále dochází k mnoha chybám v měření způsobeným nepředvídatelnými¹ odrazy zvukového vlnění od všemožných překážek. Ideální by proto bylo provádět měření ve volném prostoru, kde v širokém okolí není nic jiného než měřicí přístroj a měřený objekt.

Aktivní dosah je přibližně 3 metry, což není zrovna moc, avšak na kratší vzdálenosti je dosaženo přesnosti v rámci několika centimetrů, na 2 metry pak v průměru 4 centimetry. Hlavně díky možnosti zkalibrovat měření a tomu, že je za běhu programu měřena teplota okolí. Celková cena výroby modulu nepřesáhla 200 korun, avšak je nutné vlastnit platformu **Arduino**.

Práce jako taková mne určitě velice obohatila a jsem rád, že jsem se tímto tématem zabýval.

¹ Obtížně předvídatelnými avšak spočítatelnými

Literatura

- [1] Arduino: Arduino UNO. <http://www.arduino.cc/en/Main/ArduinoBoardUno>, 2015 [cit. 2015-4-21].
- [2] Australia's Navy, A. N.: Sonar Technology. <http://www.defencejobs.gov.au/navy/technology/comms.aspx>, 2013-12-2 [cit. 2015-4-20].
- [3] Dostál, J.: *Operační zesilovače*. Praha, 1981, iSBN 80-7300-049-0.
- [4] Kákona, J.: Domácí výroba plošných spojů fotocestou. http://www.mlab.cz/Articles/HowTo/How_to_make_PCB/DOC/HTML/How_to_make_PCB.cs.html, 2014-2-16 [cit. 2015-4-20].
- [5] PIC12F629: datasheet. <http://akizukidenshi.com/download/PIC12F675.pdf>, [cit. 2015-4-21].
- [6] Sanchez, J.; Canton, M. P.: *Microcontroller programming: the microchip PIC*. CRC Press, 2006.
- [7] Smith, A.: Historical development of sonar. <http://science.jrank.org/pages/6289/SONAR-Historical-development-SONAR.html>, [cit. 2015-4-20].
- [8] Wikipedia: Hydrofon. <http://cs.wikipedia.org/wiki/Hydrofon>, 2012-8-22 [cit. 2015-4-20].
- [9] Wikipedia: Arduino. <http://en.wikipedia.org/wiki/Arduino>, 2013-2-2 [cit. 2015-4-21].

- [10] Wikipedia: Ultrasound. <http://en.wikipedia.org/wiki/Ultrasound>, 2015-1-11
[cit. 2015-4-20].

Dodatek A

Seznam příloh

Příloha 1: Elektrické schéma

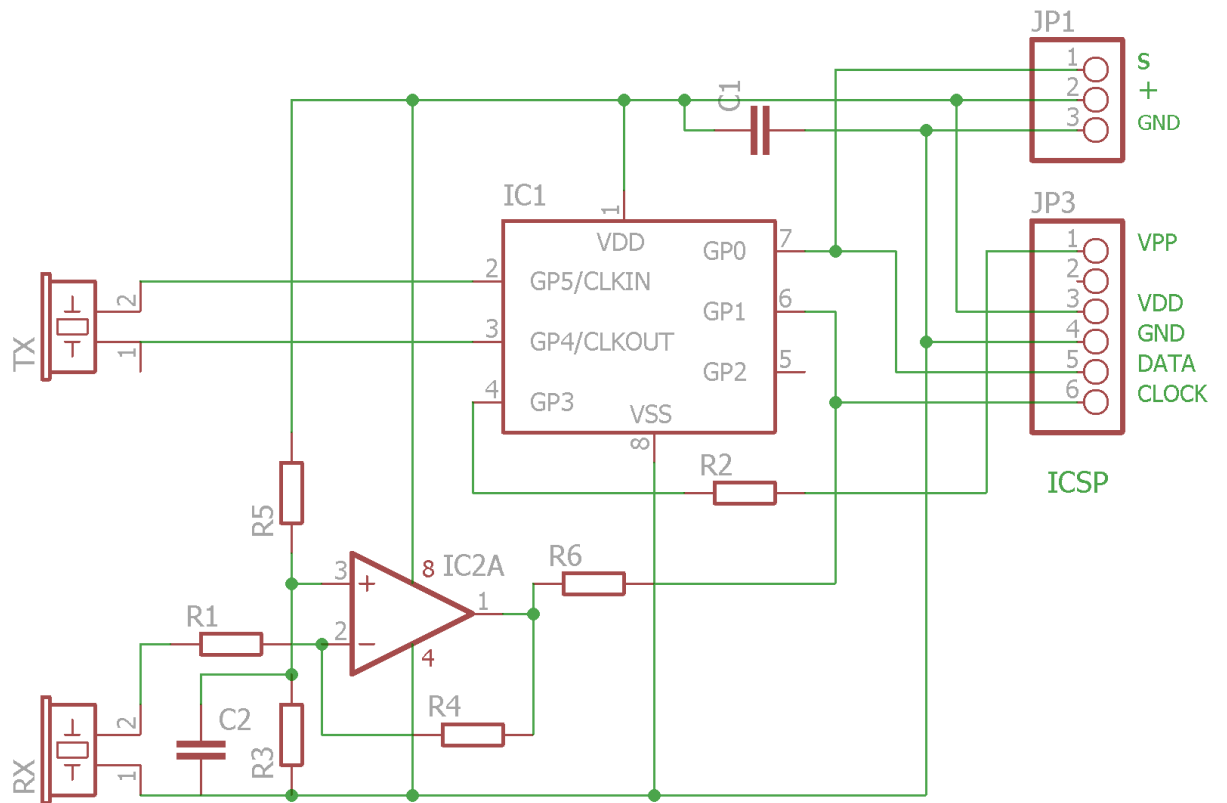
Příloha 2: Deska plošného spoje

Příloha 3: Fotodokumentace

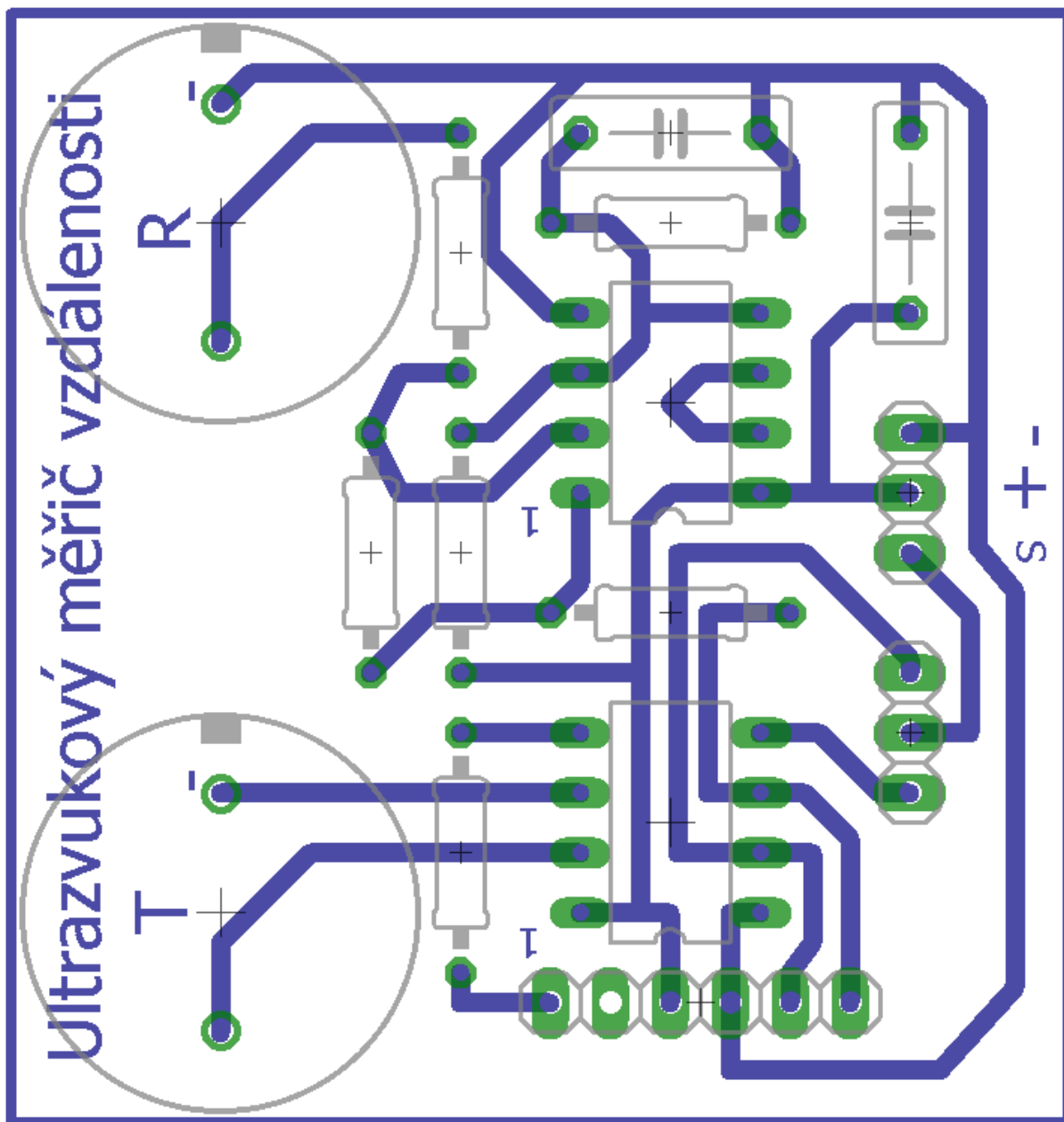
Příloha 4: CD obsahující:

- (a) Zdrojový tvar práce (latex)
- (b) Text práce (pdf)
- (c) Zdrojové kódy programu pro PIC12F629
- (d) HEX soubor programu pro PIC12F629
- (e) Datasheet mikrokontroléru PIC12F629
- (f) Zdrojové kódy programu pro Arduino
- (g) HEX soubor programu pro Arduino
- (h) Zdrojové kódy programu COMListener
- (i) Spustitelný program COMListener.exe
- (j) Schémata vytvořená v programu Eagle
- (k) Manuál

Příloha 1:



Příloha 2:



Příloha 3:

