



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

BEZDRÁTOVÁ MĚŘENÍ HLADINY VODY WIRELESS WATER LEVEL SENSORS

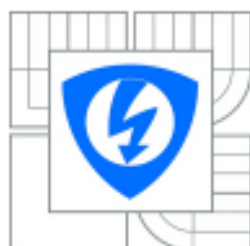
Diplomová práce
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAKUB POSPÍŠIL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ONDŘEJ MORSKÝ



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Jakub Pospíšil

ID: 78407

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Bezdrátová čidla pro měření hladiny vody

POKYNY PRO VYPRACOVÁNÍ:

Úkolem práce je navrhnout čidlo pro měření hladiny podzemní vody. Čidla musí být navržena s ohledem na cenu, životnost v nepříznivých klimatických podmínkách a spotřebu elektrické energie. Měření bude provádět řádově 10 čidel na ploše cca 500x500m. Data budou odesílána do centrální stanice, která je předá dále ke zpracování. Úkolem práce není návrh centrální stanice, ale pouze čidel a komunikačního protokolu. Praktickým výstupem bude funkční a otestované čidlo.

DOPORUČENÁ LITERATURA:

[1] MATOUŠEK, David. Práce s mikrokontroléry Atmel AVR : BEN - technická literatura, 2006. 376 s. ISBN 80-7300-209-4.

[2] BEJČEK, Ludvík, ĎAĎO, Stanislav, PLATIL, Antonín. Měření průtoku a výšky hladiny: BEN - technická literatura, 2006. 448 s. ISBN 80-7300-156-X.

Termín zadání: 29.1.2010

Termín odevzdání: 26.5.2010

Vedoucí práce: Ing. Ondřej Morský

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Abstrakt práce v češtině

Diplomová práce se zabývá návrhem a realizací zařízení na měření hladiny vody. Tato data jsou bezdrátově odesílána do 500m vzdálené stanice. Práce se postupně zabývá možnými způsoby řešení, na základě kterých je vybráno řešení pro realizaci. V navazující části je popsán podrobný postup realizace. Na měření hladiny jsou použita ultrazvuková čidla, řídicím prvkem je procesor ATmega162 a data jsou odesílána vysílačem RC1280HP. Zařízení je navrženo tak, aby mělo co nejmenší spotřebu, protože jej bude napájet pouze baterie. Součástí této práce není řešení přijímací stanice. V části realizace je popsán funkční a otestovaný vzorek.

KLÍČOVÁ SLOVA

Klíčová slova v češtině

Měření výšky hladiny vody, ultrazvuková čidla, mikroprocesor ATmega162 ATMEL, měření teploty, SMT 160, bezdrátový přenos dat, RS-232, RC1280HP, hamming kód.

ABSTRACT

The thesis deals with both scheme and its implementation of water-level metering apparatus. This data are send wireless into 500 m distant station. Potential ways of solution are gradually studied and final design suggested. Detailed implementation methods are described in the following section. Ultrasonic sensors are employed for level measurement and controlling element is processor ATmega162, data are transmitted by transceiver RC1280HP. Apparatus is suggested with a view to the lowest power consumption considering it will be supplied only with a accumulator. Solving of accepting station is not a part of the thesis. Functional tested sample is understated in the execution section.

KEYWORDS

Water-level metering, ultrasonic sensors, microprocessor ATmega162 ATMEL, temperature metering, SMT 160, wireless data transfer, RS-232, RC1280HP, hamming code.

POSPÍŠIL, J. *Bezdrátová sídla pro měření hladiny vody*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 59 s. Vedoucí diplomové práce Ing. Ondřej Morský.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma " Bezdrátová sídla pro měření hladiny vody " 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu mé diplomové práce Ing. Ondřeji Morskému za příkladnou metodickou, pedagogickou a odbornou pomoc a za další cenné rady při zpracování mé diplomové práce.

V Brně dne

.....

(podpis autora)

Obsah

Úvod.....	11
Požadavky a prostředí.....	12
Návrh realizace.....	13
1 Výběr formy vysílání.....	13
1.1 Podmínky pro vysílací zařízení.....	13
1.2 Vysílání pomocí Bluetooth.....	13
1.3 Vysílání pomocí Zigbee.....	13
1.4 Vysílání pomocí vysílače RC1280HP.....	14
2 Výběr idla pro měření hladiny.....	14
2.1 Měřicí podmínky.....	14
2.2 Idla pro měření hladiny.....	14
2.3 Vhodné typy idel.....	15
2.4 Tlakové idlo.....	15
2.5 Ultrazvuková idla.....	17
2.5.1 Atmosférické vlivy na rychlost zvuku.....	19
2.5.2 Ultrazvukový přijímač UST-40R a vysílač UST-40T.....	20
2.6 Mechanické idlo.....	20
2.7 Výběr idla pro realizaci.....	23
3 Idlo pro měření teploty.....	23
3.1 Výběr teplotního idla.....	23
4 Obvod reálného času.....	24
4.1 RTC62421.....	25
5 MAX232.....	25
6 Mikroprocesor.....	25
6.1 Požadavky na procesor.....	25
6.2 Mikroprocesor ATmega162.....	25
6.3 Programování procesoru.....	27
6.3.1 Programátor AVR Dragon.....	27
Realizace.....	30
7 Napájecí napětí.....	30
8 Konfigurace mikroprocesoru a periférií.....	30
8.1 Vývojové diagramy a popis programu.....	31
8.2 Vnější přerušení.....	31
8.3 Generování měřicího signálu.....	33
8.4 Měření času.....	37
8.4.1 Nastavení čítače.....	37
8.4.2 Vyhodnocení času.....	38
8.5 Analogový komparátor.....	39

8.6	Výpočet vzdálenosti	42
8.7	Měření teploty.....	43
8.8	Režimy spánku	44
8.9	Frekvence měření	45
8.10	Odesílání dat	46
8.10.1	Komunikační protokol.....	46
8.10.2	USART.....	48
9	Závěr	52
	Seznam literatury.....	53
	Seznam použitých zkratk.....	54
	Seznam příloh.....	55

Seznam obrázk

Obr.2. 1: Tlakové idlo (obrázek převzat z článku [4])	16
Obr.2. 2: Průtok tlakovým idlem	16
Obr.2. 3: Umístění idla do trubky i hadice upevněné ve studni	17
Obr.2. 4: Ultrazvukové vlnění.....	18
Obr.2. 5: Typická přijímací a vysílací charakteristika ultrazvukových idel.....	19
Obr.2. 6: Vysílací a přijímací charakteristika převodníků UST-40R a UST-40T (obrázek převzat z katalogového listu [6]).....	20
Obr.2. 7: Umístění ultrazvukových idel ve studni.....	20
Obr.2. 8: Náčrt realizace mechanického idla	22
Obr.2. 9: Průtok polystyrenovou koulí	22
Obr.6. 1: Pouzdro mikroprocesoru ATmega162 (převzato z katalogového listu[10]).....	27
Obr.6. 2: Zapojení programu na JTAG rozhraní.....	28
Obr.8. 1: a) hlavní program, b) vlnění při průtoku, c) analogový komparátor	31
Obr.8. 2: Vysílaný a přijímaný signál převodníkem při vzdálenosti 15cm	34
Obr.8. 3: Vysílaný a přijímaný signál převodníkem při vzdálenosti 30cm	34
Obr.8. 4: Vysílaný a přijímaný signál převodníkem, pokud není v jejich trase žádná překážka	35
Obr.8. 5: Schéma zapojení ultrazvukových převodníků	36
Obr.8. 6: Blokové schéma analogového komparátoru (Obrázek převzat z katalogového listu mikroprocesoru [10]).....	40
Obr.8. 7: Diagram napětí.....	40
Obr.8. 8: Zachycená odesílaná data	47

Seznam tabulek

Tab.2. 1: část tabulky binárních hodnot značících úroveň hladiny	21
Tab.8. 1: Nastavení registru GICR.....	32
Tab.8. 2: Nastavení registru MCUCR.....	32
Tab.8. 3: Nastavení registru TCCR1A.....	37
Tab.8. 4: Nastavení registru TCCR1B	37
Tab.8. 5: Nastavení registru ACSR.....	40
Tab.8. 6 Nastavení registru obvodu reálného času C_E	45
Tab.8. 7: Hodnoty hammingova kódu a jejich znaky ASCII.....	48
Tab.8. 8: Nastavení registru UCSR0A.....	49
Tab.8. 9: Nastavení registru UCSR0B	49
Tab.8. 10 Nastavení registru UCSR0C	50

Úvod

Cílem diplomové práce je návrh a realizace zařízení pro měření hladiny vody a odesílání naměřených dat do vzdálené přijímací stanice. Požadavkem je bezdrátový přenos těchto dat. V zadání je požadovaná vzdálenost přijímací stanice okolo 500 m. Celý návrh by měl být proveden tak, aby bylo možné jej použít pro více čidel, která budou data odesílat do jedné centrální přijímací stanice.

Úkolem je dle požadavků vybrat součásti, pomocí kterých bude možno zařízení realizovat co nejefektivněji, tzn. výběr vhodného čidla, kterým bude měřena hladina vody ve studni. Dále výběr vhodného zařízení, které bude měřit hodnoty ukládat a následně odesílat do vysílače. Úkolem je také vybrat vhodný vysílač, který bude schopen vysílat data do zadané vzdálenosti (500 m) a navrhnout komunikační protokol, pomocí kterého budou data odeslána v požadované formě.

Celá práce je rozdělena do tří částí. První část se zabývá měřeným prostředím, ve kterém bude měření probíhat.

Druhá část je zaměřena na návrh realizace. Jde zde především o výběr vhodných komponent zařízení. Tato část se tedy postupně zabývá výběrem vhodné formy vysílání dat, dále výběrem čidla pro měření hladiny a výběrem řídicího prvku. Při návrhu je diskutováno několik možných variant výběru komponent.

Třetí část je zaměřena na samotnou realizaci zařízení na základě vybraných prvků z druhé části. Vesměs se jedná o vhodnou formu řešení dané problematiky pomocí mikroprocesoru a nastavení jeho periférií.

Požadavky a prostředí

Hladina vody bude měřena ve studni, kde je hladina vody 6 m pod víkem studny. Do studny vede roura o průměru asi 25 cm a na hladinu je přímá viditelnost.

Vyhodnocení výšky hladiny vody bude provedeno měřidlem, jehož návrh a popis je uveden níže. Data z měřidla bude vyhodnocovat mikroprocesor umístěný u víka studny. K mikroprocesoru bude dále připojeno měřidlo na měření teploty ve studni, aby bylo možno vyhodnotit i teplotu.

Data bude mikroprocesor odesílat do vysílače, který bude k mikroprocesoru připojen. Vysílač bude data odesílat do přijímače a přijímacího centra, ze kterého bude možno data vyčíst. Přijímač bude od vysílače vzdálen zhruba 500 m.

Celé zařízení bude napájeno pouze baterií, což znamená, že je potřeba, aby měřidlo co nejmenší spotřebovalo elektrické energie. Proto bude mikroprocesor i celý systém, vždy po změně výšky hladiny a odeslání dat uspán. Data budou odesílána jednou za hodinu, popřípadě v kratším intervalu, podle potřeb uživatele.

Návrh realizace

1 Výběr formy vysílání

1.1 Podmínky pro vysílací zařízení

Výběr vysílače je určen několika základními kritérii. Jedním z nejdůležitějších je předpokládaná vzdálenost přijímacího centra. Tato vzdálenost se bude pohybovat okolo 500 m, proto je třeba vybírat vysílače, které budou schopny na tuto vzdálenost vysílat. V potaz je zároveň třeba brát i přirodní a umělé překážky.

Při výběru vysílače musí být také zohledněn fakt, že vysílač bude součástí bezúdržbového zařízení, které bude napájeno baterií. Proto je třeba vybrat vysílač, který má nízkou spotřebu energie a lze jej uvádět do úsporného režimu v okamžiku, kdy nezpracovává a nevysílá data.

V neposlední řadě je důležité, aby byl vysílač schopen pracovat v různých povětrnostních podmínkách nebo aby jej bylo možno umístit do schránky, která bude vysílač chránit vlivem povětrnostním vlivům.

Poslední podmínkou je samozřejmě co nejnižší cena výrobku.

1.2 Vysílání pomocí Bluetooth

Použití technologie Bluetooth by bylo relativně jednoduché, protože dnešní trh nabízí velké množství vysílačů, které lze jednoduše připojit k velkému množství procesorů, které jsou dostupné na trhu. Příkladem takového vysílače je vysílač OEMSPA-311i [1], tento vysílač lze jednoduše připojit k mikroprocesoru, který je schopen odesílat data přes sériový port. Data je pak možno odesílat právě touto sériovou linkou a automaticky je odesílat do Bluetooth přijímače. Takto zvolený princip odesílání dat je ideální, bohužel mají tyto vysílače zásadní nevýhodu. Touto nevýhodou je maximální dosah vysílání, který činí 100 m, což je pro tento případ absolutně nedostačující. Navíc je tato vzdálenost pouze teoretická a je uváděna pro vysílání na otevřeném prostranství. Bylo by možné použít několik přijímačů a vysílačů, které by byly umístěny v trase mezi vysílačem a přijímačem tak, aby data mohla být odesílána i na vzdálenost 500m. Ovšem tento způsob by byl dosti komplikovaný a hlavně drahý, což nesplňuje požadavky pro realizaci tohoto systému.

1.3 Vysílání pomocí Zigbee

Při použití technologie Zigbee by docházelo ke stejným problémům jako u technologie Bluetooth. S výhodou by tato technologie šla použít, pokud bychom vytvářeli velkou síť bodů, protože zařízení Zigbee je vytvořeno tak, aby mohlo více zařízení pracujících na stejném systému spolu komunikovat. Ovšem tohoto se využívá spíše v domácnosti. Dosah zařízení Zigbee je zhruba 75 m, což je opět nedostačující pro tuto realizaci.

1.4 Vysílání pomocí vysílače RC1280HP

Pokud má být použit vysílač, který je schopen bez jakýchkoliv opakovačů signálu a podobných zařízení vysílat na vzdálenost 500 m, je výběr velice omezen. Za nejlepší řešení této práce byly shledány systémy komunikující v bezlicenčním pásmu na frekvenci okolo 868 Hz, kdy je možné data vysílat až do vzdálenosti několika kilometrů a to i v místech, kde se v cestě signálu objevují překážky typu rozvážných zástaveb, stromů apod.

Jako ideální řešení se jeví vysílač RC1280HP [2]. S tímto vysílačem se dá velmi dobře pracovat ve spolupráci s mikroprocesorem, který je schopen komunikovat přes sériové rozhraní RS232/485, čímž práce s ním je srovnatelně jednoduchá jako u vysílače OEMSPA-311i [1], který využívá technologii Bluetooth.

Další důležitou vlastností tohoto vysílače je velmi nízká spotřeba a možnost uspaní celého zařízení, čímž se spotřeba ještě více sníží. Tyto vlastnosti by byly pro bezúdržbové měřidlo téměř ideální, bohužel cena jednoho vysílače se pohybuje příliš vysoko – zhruba 1700 Kč bez DPH.

Výrobci cenově výhodnějších vysílačů podobného typu od firem CORONIS a MIPOT nebyli schopni zaručit vysílání na vzdálenost cca 500 m a pokud ano, tak na hranici možností. Proto bude použit právě vysílač RC1280HP. Z finančních důvodů bude sestaveno měření hladiny pouze pro jeden vysílač, ale bude umožněno jednoduše přidat vysílače další, aniž by musel být složitě upravován program procesoru.

K vysílači je nutno připojit anténu. Její připojení se dělá kabelem s jedním volným koncem, který se naletuje na vysílač a na druhé straně se konektorem připojí k anténě.

2 Výběr měřidla pro měření hladiny

2.1 Měřicí podmínky

Stejně jako na všechny ostatní komponenty celého zařízení pro měření výšky hladiny je i na samotné měřidlo požadavek co nejmenší spotřeby energie.

Měřidlo bude použito na měření výšky hladiny spodní vody. Hladina vody je 6 m pod úrovní země. Průměr otvoru do studny je cca 25 cm a na vodní hladinu je přímá viditelnost.

Výška hladiny se mění maximálně o 30 cm.

2.2 Měřidla pro měření hladiny

V dnešní době patří k nejpoužívanějším měřidlům hladiny tlaková měřidla, protože se jedná ve většině případů o nejlevnější řešení. Nejedná se však vždy o správné řešení, protože kvalita některých levných snímačů není velká, proto je dobré se zamyslet nad tím, jaké následky by mohly nastat v případě poruchy měřidla. V případě tohoto projektu by sice došlo ke ztrátě aktuálních informací o stavu hladiny vody, ale následky by nebyly nijak závažné. Jednalo by se tedy pouze o koupi a výměnu nefunkčního senzoru.

Pravděpodobně neexistuje univerzální senzor, který by bylo možno použít ve všech případech. Kromě ceny musíme také uvažovat podmínky, ve kterých bude idlo použito a jakým způsobem budou zpracovávána data, která bude idlo vysílat [3].

Dalším parametrem, kterým je potřeba se při výběru idla zabývat, je přesnost senzoru. Jak je již známo, hladina vody, která bude měřena, se bude měnit v rozmezí 30 cm. Požadovaná přesnost měření bude alespoň 1 cm.

Přesnost měření udávaná výrobcem je ovšem určena zejména při laboratorních podmínkách a měření v reálných podmínkách se může lišit od hodnot udávaných výrobcem. Jak je uvedeno v podmínkách, tak prostory, ve kterých bude idlo umístěno, nejsou příliš velké. Jedná se o rouru o průměru 25 cm, proto musíme vybrat idlo, které bude možno do této roury umístit.

Protože bude idlo umístěno ve vlhkých prostorech, tak je potřeba zajistit dostatečné krytí idla.

2.3 Vhodné typy idel

Z podmínek a vlastností idel je zřejmé, že některé typy idel nejsou vhodné pro konstrukci systému.

Pro vyhodnocování dat bude použit mikroprocesor. Je potřeba, aby idlo s mikroprocesorem nějakým způsobem komunikovalo, nebo mělo výstup, kde bude elektrická veličina úměrná výšce hladiny. Výběr je také omezen tím, že není možno umístit idlo na vnější stranu nádrže, ili je potřeba idlo umístit přímo dovnitř studny. Další problém je s napájením idla. Jak už bylo několikrát zmíněno, idlo musí mít minimální odběr, proto je vyloučeno použití idla, které je potřeba permanentně napájet.

Při zohlednění výše zmíněných podmínek se jako nejlepší řešení jeví tři druhy idel a to tlakové idlo, které bude umístěno na dně studny nebo v jednostranně uzavřené trubici, dále ultrazvukové idlo, které bude umístěno nad vodní hladinou. A v poslední řadě idlo, které bude řešeno z větší části mechanicky.

2.4 Tlakové idlo

Tlakové hladinoměry jsou dosud značně populární pro svoji jednoduchost a relativně nízkou cenu. Přestože určení hladiny z hydrostatického tlaku pomocí tlakových senzorů trpí mnoha problémy, stále tvoří tlakové hladinoměry téměř polovinu všech používaných prmyslových hladinoměrů [3].

Měření vychází z jednoduchého principu hydrostatického tlaku. Sloupec kapaliny svojí tíhou vyvíjí tlak úměrný hloubce pod hladinou h , hustotě kapaliny ρ a tíhovému zrychlení g :

$$P = h \cdot \rho \cdot g \quad (4.1)$$

V tomto případě se jeví jako jediná možnost umístění idla tak, že by bylo spuštěno na kabelu na dno studny.

Pokud by se jednalo například o kapalinu, která by mohla poškodit idlo, a bylo by idlo v kapalině přímo ponořené, tak by muselo být idlo umístěno do komory s membránou. V tomto případě by byla s kapalinou v přímém kontaktu pouze membrána. V našem případě se nemusíme tímto problémem zabývat, protože měřenou kapalinou je voda a na trhu jsou k dispozici idla, která lze na toto měření použít.

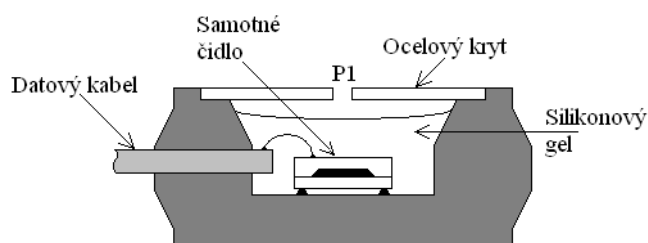
Výhodou tlakových idel je, že nejsou náchylná na neistoty v kapalině apod., ovšem problémem u těchto idel je jejich přesnost a to obzvláště při určování hustoty kapaliny z důvodů různých teplotních vlivů apod. Navíc tlakové idlo není primárně určeno pro měření tlaku kapalin, proto se přesnost pro měření hladiny udává v řádech několika centimetrů. Tato přesnost by byla pro tuto realizaci případně naprosto dostatečná.

Je známo, že na trhu existuje velké množství tlakových idel, která by bylo možno použít pro toto měření [4][5]. Tato idla bohužel nejsou konstruována přímo pro měření tlaku v kapalinách. Je potřeba je umístit buď do komory s membránou, nebo je odizolovat od kapaliny jiným způsobem. Idla jsou konstruována v tloušťce tak, aby bylo možné jejich jednoduché umístění na plošný spoj jak je vidět na obrázku 2.1.



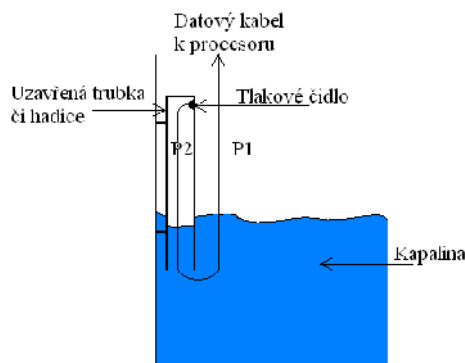
Obr.2. 1: Tlakové idlo (obrázek převzat z článku [4])

Pokud je idlo umístěno do kapaliny, musí přijít do kontaktu s kapalinou pouze měřicí část, která má v sobě již silikonovou membránu, která zabrání kontaktu kapaliny přímo s měřicím prvkem. Přez idlem je znázorněno na obrázku 2.2.



Obr.2. 2: Přez tlakovým idlem

Možností by také bylo umístit do studny z jedné strany uzavřenou trubku i hadici, kterou bychom vyvedli nad hladinu, viz obr. 2.3. V obou případech bychom nemuseli řešit izolaci idla od kapaliny, protože bychom měli změnu tlaku vzduchu, který je závislý na výšce hladiny vody. Ovšem nastává zde problém s dobrým utěsněním vzduchu. V mnoha případech, které lze nalézt na internetu, se tento problém nepodařilo zcela vyřešit. Jednalo se tedy vždy spíše o orientační řešení.



Obr.2. 3: Umístění idla do trubky i hadice upevněné ve studni

V řada tlakových idel, která je možno na našem trhu koupit je možné připojit k mikroprocesoru. Jejich výstupem je buď přímo napětí, které je úměrné tlaku, který na ně působí, nebo je možno s nimi komunikovat pomocí rozhraní I2C i SPI [6].

Tlaková idla by tedy šla relativně dobře použít pro náš případ. A už z hlediska konstrukčního, tak z hlediska komunikace s mikroprocesorem. Spotřeba těchto idel je také minimální, což je velice podstatná vlastnost pro tento případ.

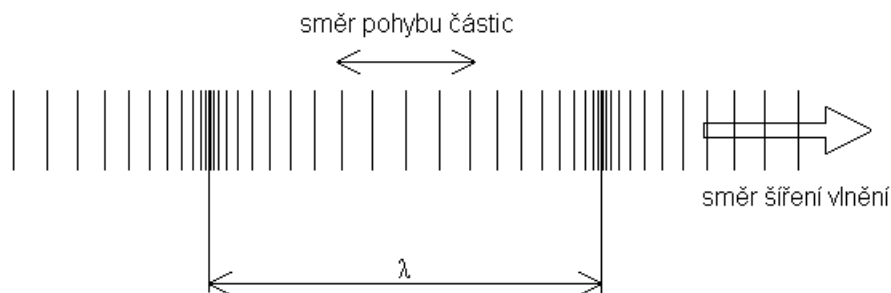
2.5 Ultrazvuková idla

Ultrazvuková idla jsou taková idla, která mají pracovní frekvenci nad hladinou slyšitelnosti člověka, to je zhruba nad 20kHz. Pokud by jejich pracovní frekvence byla nižší, tak by se jednalo o idla sonická.

Princip ultrazvukových idel spočívá ve zhuštění a zředění vzduchových částic vlivem vysílaných ultrazvukových vln. Viz obr. 2.4.

Lokální zednění i zhuštění částic vzduchu znamená změnu oproti normálnímu barometrickému tlaku.

Rychlost šíření zvuku ve vzduchu je za normálních podmínek zhruba $340\text{m}\cdot\text{s}^{-1}$. Je však důležité zmínit, že tato rychlost velice silně závisí na teplotě, vlhkosti a tlaku, což je třeba při měření zohlednit. O této problematice bude pojednáno dále.



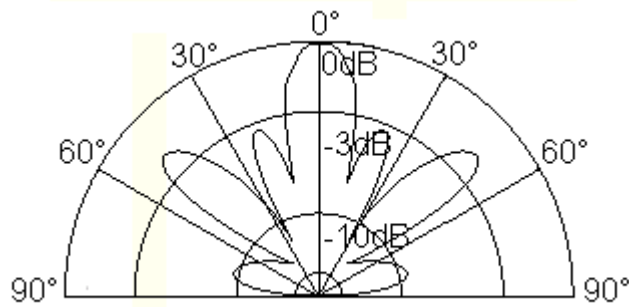
Obr.2. 4: Ultrazvukové vlnění

Intenzita zvuku, tj. plošná hustota přeneseného výkonu, klesá se zvětšující se vzdáleností a také vlivem atmosférického útlumu. Na toto je třeba dbát při umístění ultrazvukových sond. Sonda musí být umístěna ve vhodné vzdálenosti od hladiny, aby nedocházelo ke ztrátě ultrazvukového signálu. Ultrazvukový vysílač a přijímač se umístí vedle sebe nad hladinu a přijímač bude snímat, kdy se vrátí zpět odražený signál z vysílače od hladiny, tato doba bude úměrná vzdálenosti sond od hladiny.

Výhodou ultrazvukových sond je, že není potřeba jejich kontakt s měřeným materiálem, problém je však jejich závislost na vlastnostech atmosféry (teplota, tlak apod.), ale také problém s nepřesností, pokud je sonda umístěna v prašném prostředí i vznikem prachu na hladině měřené kapaliny. V tomto případě však postačí kompenzace vlivu teploty, o které bude pojednáno dále.

Pro generování ultrazvuku bude v této realizaci použit ultrazvukový převodník a to piezoelektrický a to z důvodu jeho dostupnosti a jednoduchého použití. Stejně tak bude tento typ převodníku použit i jako přijímač. Tyto převodníky využívají mechanické deformace piezoelektrického materiálu při přiložení elektrického signálu.

Další důležitou vlastností ultrazvukového převodníku je jeho vyzařovací charakteristika. Úhel hlavního vyzařovacího laloku je nepřímo úměrný průměru převodníku a vysílací frekvenci. Pokud je převodník rozměrnější, tak je úroveň odraženého signálu silnější, protože šířka hlavního laloku je menší. To samé platí o vysílací frekvenci, čím vyšší frekvence, tím silnější je odražený signál, protože se nerozptyluje do šířky. Přijímací charakteristika je shodná s vyzařovací. Typická šířka hlavního laloku je asi 15°. Viz obr. 2.5.



Obr.2. 5: Typická p ijímací a vysílací charakteristika ultrazvukových idel

Každý ultrazvukový p evodník má kone n krátkou dobu doznívání (ringing). V této době nem že p evodník fungovat jako p ijíma a vzniká tím v blízkosti senzoru tzv. „mrtvá zóna“ (asi 0,2-0,8m). Z tohoto d vodu je pot eba použít p evodníky dva, jeden jako vysílací a druhý jako p ijímací, ímž bude problém mrtvé zóny odstran n.

Úhel dopadu na m enou hladinu nesmí být p íliš velký, aby nedošlo k odrazu signálu do míst, kde jej není možné zachytit p ijíma em [3].

2.5.1 Atmosférické vlivy na rychlost zvuku

Jak již bylo n kolikrát zmín no výše, rychlost zvuku závisí na mnoha faktorech. Jedním z nejv tších initel , které ovliv ují rychlost ší ení zvuku, je teplota. Za normálních podmínek je rychlost zvuku zhruba $340\text{m}\cdot\text{s}^{-1}$. Ovšem ve vzorci 4.2 je vid t jak velký vliv na rychlost zvuku teplota má. Pro rychlost zvuku $c [\text{m}\cdot\text{s}^{-1}]$ ve vzduchu za teploty $t [^{\circ}\text{C}]$.

$$c = 331,8 + 0,61 \cdot t \quad (4.2)$$

P esn jší výpo et poskytuje vzorec 4.3:

$$c = \sqrt{\kappa \cdot R \cdot T} \quad (4.3)$$

Kde κ je adiabatická konstanta (1,402 pro vzduch), R je plynová konstanta pro vzduch ($287,05 \text{ J}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$) a T je termodynamická teplota ($t + 273,15\text{K}$).

Pokud bude hladina vody m ena ultrazvukovým idlem, je bezpodmíne n nutné, aby m ící za ízení obsahovalo zárove i m ení teploty, aby bylo možné správn vypo ítat rychlost zvuku a následn p esn zm ít výšku hladiny vody [3].

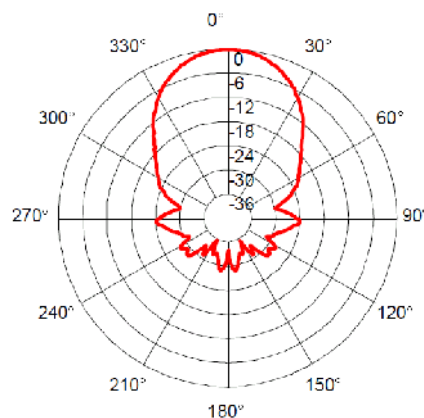
2.5.2 Ultrazvukový přijímač UST-40R a vysílač UST-40T

Jedná se o velmi dostupné součástky, které lze koupit na českém trhu, proto jsou vhodné pro realizaci systému, při použití ultrazvukových měřidel.

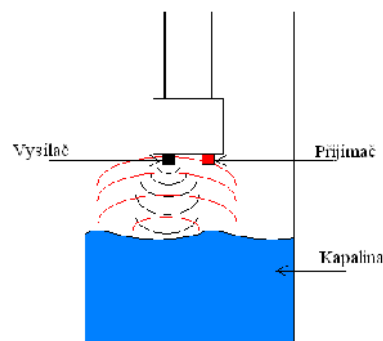
Výhodou těchto součástí je jejich jednoduché ovládání, stačí je připojit pouze na jeden pin procesoru, na který bude vysílán obdélníkový signál s frekvencí $f=40\text{kHz}$. Přijímač lze také připojit pouze na jeden vstup procesoru, kde bude mít menší odezvu.

Základní parametry:

- Pracovní frekvence **40kHz±1kHz**
- Maximální výšilací napětí **20V**
- Provozní teplota **-30° až 80°**
- Vydávací a přijímací charakteristika:



Obr.2. 6: Vydávací a přijímací charakteristika převodníků UST-40R a UST-40T (obrázek převzat z katalogového listu [6])



Obr.2. 7: Umístění ultrazvukových měřidel ve studni

2.6 Mechanické měřidlo

Existuje velké množství různých mechanických měřidel, jedná se však většinou o spínací měřidla, která je potřeba permanentně napájet. Takováto měřidla jsou pro případ této realizace

nepoužitelná. Z konstrukčního hlediska je zapotřebí, aby bylo možné pracovat s výstupem z idla pomocí procesoru, což také omezuje výběr mechanických idel.

Byl však navržen plovákový snímač, který by bylo možné realizovat pro toto měření. Jediným problémem je relativně složitě provedení z výrobního hlediska oproti například ultrazvukovému idlu.

Jedná se o jednoduchý plovákový snímač, který by byl umístěn přímo ve studni, jak je možno vidět na obrázku 2.8. Jako plovák by zde fungovala polystyrenová koule, která by byla zasunuta na stínítko, které by bylo pevně umístěno ve studni v rozmezí 32cm, ve kterých by se hladina vody pohybovala. V polystyrenové kouli je kromě otvoru na stínítko také 5 otvorů vodorovně skrz celou kouli, jak je vidět na řezu koule, který je znázorněn na obrázku 2.9, do kterých budou z jedné strany umístěny LED diody a z druhé strany opto leny, které budou snímat světlo z diod. Tvar koule je vybrán z důvodů jednoduchého umístění diod a opto len, ale hlavně z toho důvodu, že kondenzovaná nebo dešťová voda, i jiné nečistoty se nebudou držet na povrchu a nemohou tedy způsobovat chyby měření.

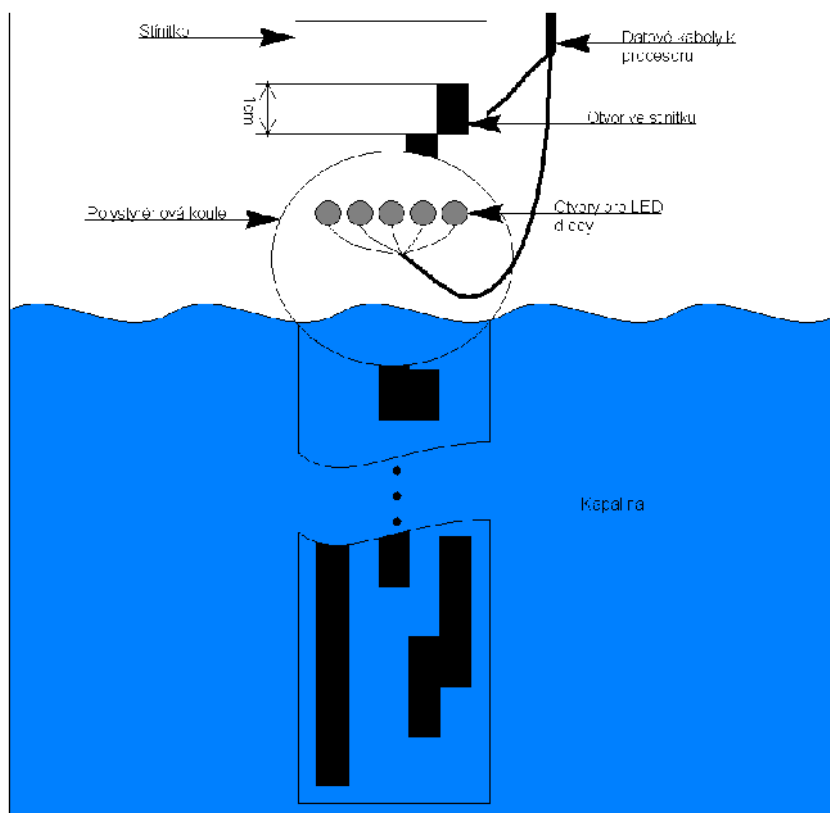
Stínítko bude zavšeno z víka studny do vody. V dolní části stínítka bude umístěno závaží, které jej bude držet ve stálé poloze. Otvory ve stínítku představují jednotlivá 5 bitová čísla, která budou odpovídat výšce hladiny, viz tab.2.1.

Jak je zřejmé z tabulky, nebylo možné použít normální 5 bitovou soustavu po sobě jdoucích číslech, protože na hranici mezi dvěma čísly by mohlo dojít k velké chybě, proto je soustava navržena tak, aby se při přechodu z jedné hladinové výšky do druhé měnila vždy jen jedna bitová hodnota, čímž bude docíleno toho, že chyba bude rovna maximálně 1cm.

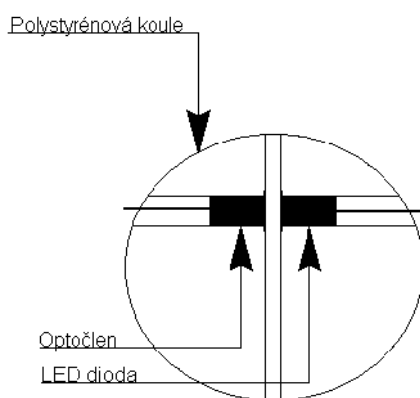
Tab.2. 1: Část tabulky binárních hodnot značících úroveň hladiny

Binární čísla značící úroveň hladiny					Změna hladiny
0	0	0	0	0	31 cm
0	0	0	0	1	30 cm
0	0	0	1	0	29 cm
0	0	0	1	1	28 cm
0	0	1	0	1	27 cm
0	0	1	0	0	26 cm
...					
1	0	1	1	0	6 cm
1	0	1	0	0	5 cm

1	0	1	0	1	4 cm
1	0	0	0	1	3 cm
1	0	0	1	1	2 cm
1	0	0	1	0	1 cm
1	0	0	0	0	0 cm



Obr.2. 8: Nákres realizace mechanického výtahu



Obr.2. 9: Detail z polystyrenovou koulí

2.7 Výběr idla pro realizaci

Z výše uvedených informací plyne, že bude vybíráno ze těchto typů idel. Prvním uvedeným typem je tlakové idlo, u kterého by bylo měření jednoduché, ovšem v těchto problémech nastávají při realizaci. Idlo je možné umístit na dno studny, ale zde je problém s dobrým odizolováním od kapaliny, což by mohlo přinést značné potíže. Dále by idlo mohlo být umístěno také na dně, ale bylo by umístěno v nádobě s membránou a byl by měřen tlak vzduchu, toto řešení by bylo na realizaci také dosti složité. Tlakové idlo by šlo také použít tak, že by bylo umístěno na hadičku i do uzavřené trubky, jak již bylo zmíněno výše, kde by se také měřil tlak vzduchu, který by byl úměrný výšce hladiny, ale z dostupných informací není tato realizace příliš vhodná. Tlakové idlo je sice relativně levné, od firmy FREESCALE je dokonce možné dostat jako student vzorky zdarma, ale celkově vzato není pro naši realizaci vhodné.

Druhým typem je idlo ultrazvukové. U tohoto idla je také jednoduchá spolupráce s mikroprocesorem. Při realizaci by neměly nastat v těchto problémech, protože při použití tohoto idla není potřeba kontaktu kapaliny s idlem. Přímě s idly (vysílá a přijíma) bude zavšně ve studni nad hladinou. Problém může nastat pouze při odesílání dat z mikroprocesoru do vysíláče, protože přímě bude muset být umístěn zhruba 2 m nad hladinou a to je 4 m pod úrovní země, kde bude umístěn vysíláč na odesílání dat. Ultrazvukový vysíláč i přijímač jsou cenově dostupné stejně jako idla tlaková. I přes některé uvedené problémy **bude celý systém realizován pomocí ultrazvukových idel.**

Těmto typem idel je idlo mechanické, stejně jako předcházející idla i toto by perfektně spolupracovalo s mikroprocesorem, jednalo by se vždy pouze o rozsvícení 5 LED diod a přechod vstupů, na které jsou připojeny opto leny. Největší nevýhodou tohoto idla je poněkud složitější realizace, protože by bylo potřeba celé idlo vyrobit. Ze zde uvedených informací je zřejmé, že idlo by šlo perfektně použít. V tomto případě však použito nebude.

3 Idlo pro měření teploty

Jak plyne z kapitoly o ultrazvukových idlech, je potřeba při použití tohoto způsobu měření hladiny vody, umístit do studny také teplotní idlo. Teplotu je v tomto případě velice důležité měřit, protože k měření se využívá rychlost zvuku, na který má teplota dosti značný vliv. Proto je potřeba tyto změny kompenzovat na základě změny teploty. Ovšem i u měření mechanickým idlem i tlakovým je dobré teplotu ve studni znát, aby bylo možné se teplotou přizpůsobit a to například kontrolou idel při teplotách pod 0°C.

3.1 Výběr teplotního idla

Při výběru teplotního idla není třeba přílišného průzkumu vhodných idel na trhu. Jako nejlepší řešení se jeví výběr idla SMT 160. Toto idlo má rozsáhlé využití v automatizační

a m ěcí technice. Pro m ění teploty ve studni je jeho funkce naprosto dosta ující. Další výhodou v našem p řípád ě je jednoduché p řipojení a komunikace řídla s použitým procesorem ATmega162.

Toto řídlo sta ěí p řipojit pouze na jeden pin procesoru, na kterém se m ěří st řída, která je p řím o výstupem z řídla a je lineárn ě závislá na teplot ě. St řídou se rozumí pom ěr logické „1“ a „0“. Mikroprocesorem se tedy m ěří pouze st řída, která je výstupem řídla a ta je p řepo řítána na teplotu. Výrobce udává vzorec pro p řepo řet:

$$DC = 0,32 + 0,0047 \cdot t . \quad (5.1)$$

Kde t je teplota [°C] a DC je st řída výstupního signálu.

řídlo je kalibrováno p řím o z výroby, proto není t ěeba řídlo již nijak kalibrovat, sta ěí jej pouze zapojit a m ěřit [7].

4 Obvod reálného řasu

Jak již bylo n ěkolikrát zmín ěno, hlavním požadavkem je nízká spot ěba celého za řízení. Z tohoto požadavku plyne, že veřkeré sou řástky, které v za řízení budou, je pot ěeba uvád ět do režim ů snižené spot ěby, tzv. režimy spánku. V p řípád ě této realizace bude do t ěchto režim ů uvád ěn mikroprocesor a vysíla ř RC1280HP. Pomocí mikroprocesoru je možné uvést do režimu spánku vysíla ř a zároveň jej z n ěj op ět uvést do normálního stavu. Mikroprocesor je sám o sob ě schopen uvést do režimu spánku sám sebe, dokonce je schopen sám sebe i probudit, ale pokud je v režimu, ze kterého je toto možné, je spot ěba pon ěkud vyšří. V tomto p řípád ě bude mikroprocesor uveden do úřporného režimu Power-down, kdy je spot ěba na nejniřřší možné hodnot ě, která je v p řím ěru menří než 1 μ A.

Jelikoř v tomto režimu není mikroprocesor schopen se sám probudit, je zapot ěbí použít n ějaké za řízení, které mikroprocesor bude probouzet. Z režimu Power-down je možné mikroprocesor probudit impulsem na vn ějš í p řeruření. Probouzení mikroprocesoru by vřak nem ělo být nijak řasté, protože m ění hladiny vody ve studni není t ěeba provád ět p říliš řasto. Ideálním intervalem je jedna hodina, pop řípád ě as kratří.

Za takto daných podm ěnek se jako ideální jeví obvod reálného řasu RTC62421. Pokud je použito další za řízení, je na m ěst ě zvážít zda nebude spot ěba tohoto za řízení vyšří, než jiný režim spánku mikroprocesoru, který by byl schopen se sám probouzet. Obvod RTC62421 má vřak spot ěbu v řádu jednotek mikroamp ěr a úřpora energie bude řpo řívat také v m ěn řastém probíhání m ění. Z t ěchto d ěvod ů bude tento obvod použít.

4.1 RTC62421

Jedná se o obvod reálného času, který lze přímo připojit ke sběrnici mikroprocesoru. Obvod používá svůj vnitřní oscilátor. Nabízí celou sadu funkcí, které se týkají práce s reálným časem, jako je například kompletní kalendář včetně přestupných roků. V tomto případě však bude využita pouze jedna funkce. Jedná se o funkci odeslání impulsu na port STD.P v určitých intervalech. V tomto případě se bude jednat o interval hodinový. Tento impuls probudí mikroprocesor a tím bude spuštěno celé měření hladiny vody ve studni.

Mikroprocesor komunikuje s obvodem reálného času pomocí 4 bitové sběrnice. Údaje hodin a kalendář jsou uloženy v registrech obvodu, viz literatura [8].

5 MAX232

Mikroprocesor bude odesílat data do vysílače pomocí sériové linky USART. Vysílač však pracuje s logikou standardu RS232 a mikroprocesor pracuje s napájecí úrovní TTL. K přechodu mezi těmito dvěma úrovněmi bude sloužit velice známý a rozšířený převodník MAX232[9]. Ke správné funkci obvodu je třeba jej zapojit tak, jak je uvedeno v jeho katalogovém listu [9] nebo v celkovém schématu uvedeném v příloze.

Mikroprocesor generuje na svém výstupu (USART) signál, který reprezentuje odesílaná data. Tento signál vstupuje do převodníku jako vstupní úroveň TTL. Výstupem převodníku jsou poté totožná data, jako byla na vstupu, pouze jsou převedena na logiku dle standardu RS232. V této logice odpovídá logické jedničce -12V a logické nule 12V. Převodník je schopen tyto úrovně převádět i opačně, toho však v této práci nebude využito.

6 Mikroprocesor

6.1 Požadavky na procesor

Požadavky na mikroprocesor jsou v tomto případě zcela jasné. Je potřeba aby měl co nejmenší odběr proudu a zároveň aby podporoval sériovou komunikaci RS232/485, pomocí kterého bude komunikovat s vysílačem.

Práce bude zaměřena na mikroprocesory firmy Atmel, kvůli předchozím zkušenostem a to s typem AVR.

Jelikož nejsou požadavky na mikroprocesor příliš velké, bude možno použít jeden z nejzákladnějších mikroprocesorů, který lze programovat pomocí programátoru AVR Dragon, který je k dispozici a to procesor ATmega162 [10].

6.2 Mikroprocesor ATmega162

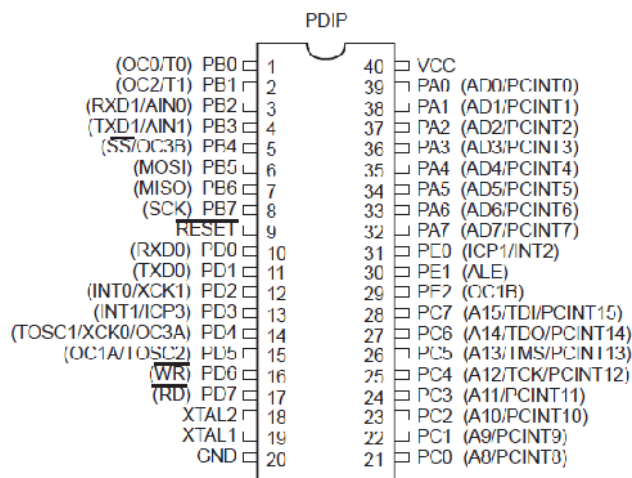
Jedná se o osmibitový mikroprocesor založený na RISC architektuře [11]. Veškeré instrukce jsou vykonávány pomocí aritmeticko-logické jednotky ALU s pomocí 32 registrů

z paměti SRAM. Paměť programu typu FLASH má 16KByt. Tento procesor podporuje komunikaci přes sériový kanál USART, který bude využit při komunikaci s vysílačem RC1280HP [2]. Mikroprocesor je napájen 5V a není potřeba k němu připojovat externí krystal. V tomto případě je však krystal použit, protože interní krystal se vyznačuje značnou nepřesností. Požadavek na přesnou taktovací frekvenci je zde z toho důvodu, že je potřeba generovat přesný 40kHz signál, který bude odesílán do ultrazvukových převodníků. Tento signál, jak je níže popsáno, je generován pomocí zpoždění, které musí být přesné. Další funkce, která bude u procesoru využita, je úsporný režim. Mikroprocesor ATmega162 jich má hned několik, Idle, Power-save, Power-down, Standby a Extended Standby[8]:

- Idle – režim zastaví CPU, ale SRAM, časovače, SPI port, a přerušení dále fungují.
- Power-save – v tomto režimu je usnávána většina čipu, kromě asynchronního čítače, což uživateli dovoluje hlídat časovou základnu.
- Power-down – režim zachová obsah registrů, ale zastaví oscilátor, čímž vypne všechny další funkce čipu až do následujícího přerušení, nebo resetu.
- Standby – v tomto režimu běží jen krystalový oscilátor, zbytek zařízení je usnává, to dovoluje velmi rychlý start, ale zároveň šetří energii.
- Extended Standby – v tomto režimu běží hlavní oscilátor i asynchronní čítač.

V této realizaci bude nejvhodnější použít úsporný režim „Standby“, protože spotřeba v tomto režimu je minimální a procesor se bude zapínat pouze při změně vzdálenosti vodní hladiny od idla, resp. hloubku studny a následném odesílání dat. Z tohoto režimu bude mikroprocesor probouzen pomocí obvodu reálného času, který bude generovat v určeném časovém intervalu signál na portu vnějšího přerušení procesoru.

Další funkcí mikroprocesoru, která bude využita, je analogový komparátor, ten bude sloužit k detekci odražené vlny od hladiny vody. Tzn., že bude měřit čas od vyslání impulzu až k jeho návratu k idlu. K měření tohoto času bude využit jeden ze tří 16 bitových čítačů / časovačů.



Obr.6. 1: Pouzdro mikroprocesoru ATmega162 (p evzato z katalogového listu[10])

6.3 Programování procesoru

P i výb ru zp sobu programování procesoru se nabízí celá ada možností, ovšem ne všechny jsou tím optimálním ešením. P i výb ru bylo p ihlíženo hlavn k dostupnosti vývojového prost edí, se kterým je programátor schopen spolupracovat a také jaké množství a jaké typy procesor je schopen programovat. Proto byl zvolen práv programátor AVR Dragon, který je dražší, než programátory, které lze sestrojít pomocí velkého množství návod , které jsou k dispozici na internetu. K programátoru je však voln dostupné programovací prost edí AVR Studio. V AVRStudiosu se výborn orientuje a jsou zde vypsány veškeré registry použité u daného procesoru, což je obrovskou výhodou, zvlášt pokud chceme používat debugger, který slouží k simulaci b hu programu.

P i výb ru programovacího jazyka se nabízejí dv možnosti, bu programovací jazyk ASSEMBLER, nebo jazyk C. K ešení této úlohy byl vybrán programovací jazyk C. V porovnání s ASSEMBLEREM se ve zdrojovém kódu jazyku C daleko lépe orientuje a celkov práce v n m je pohodln jší.

6.3.1 Programátor AVR Dragon

Jedná se o programátor, který je velice oblíbený pro svou jednoduchost a výkonnost. Velkou výhodou je, že m že fungovat jako JTAG rozhraní pro lad ní aplikace p ímo v mikroprocesorech. Tzn., že je možné nahrát zdrojový kód do procesoru a necháme jej p ipojen v patici programátoru. Pomocí programu AVR Studio se k n mu p es programátor p ipojíme a m žeme po jednotlivých strojových cyklech sledovat jak se naprogramovaný procesor chová, zároveň také m žeme pomocí AVR Studia m nit hodnoty veškerých registr a sledovat chování procesoru. Tímto nám celé programovací za ízení m že simulovat nap .

stisknutí tlačítka, aniž bychom nějak mohli apod. Tato funkce je však možná jen pro mikroprocesory, které mají FLASH paměť do 32kB.

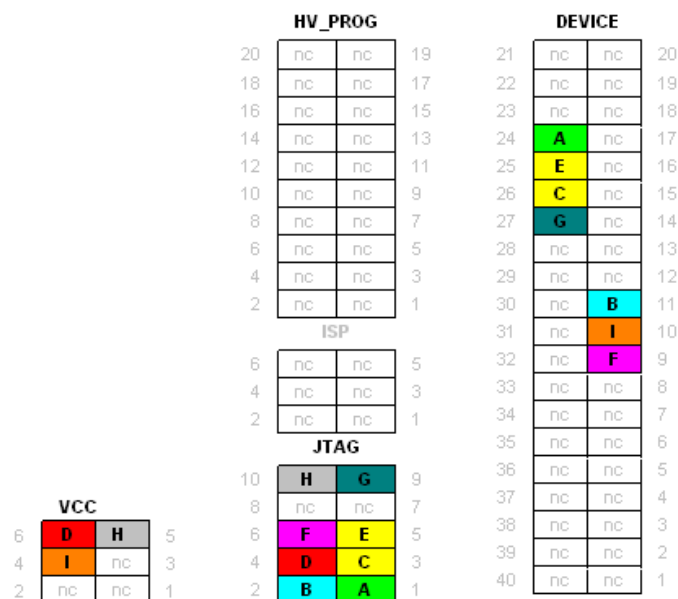
Programátor podporuje následující programovací a ladící rozhraní:

- ISP – sériové programování přímo v aplikaci, 3 vodičové (plus napájecí vodiče).
- JTAG – 4 vodičové programování (plus napájecí vodiče).
- Sériové programování vyšším napětím.
- Paralelní programování.
- JTAG ladění pro procesory s FLASH pamětí do 32kB.
- DebugWIRE jednovodičové ladící rozhraní pro procesory s malým počtem pinů.

S počítačem programátor komunikuje přes USB rozhraní, přes které je zároveň i napájen, a není zapotřebí žádných dalších zdrojů. Zároveň je zde možnost připojení externího napájení v případě, že zařízení bude odebírat větší proud, než je schopno USB rozhraní dodat. Tento problém může však nastat jen v případě, že budeme používat programátor v režimu JTAG ladění a budeme k mikroprocesoru připojovat zařízení, která mají vyšší odběr proudu.

Pro programování bylo vybráno rozhraní JTAG, protože jeho zapojení je stejné, jako když chceme použít JTAG ladění.

Zapojení programátoru na JTAG rozhraní je na obrázku 6.2:



Obr.6. 2: Zapojení programu na JTAG rozhraní

Programátor je plně kompatibilní s programovacím prostředím AVR Studio4, které je využito k programování. Je zde i návod na jeho zapojení a možnosti jeho využití.

Pokud však chceme programovat v AVR Studiu v jazyce C, tak je zapotřebí si nainstalovat i kompilátor WinAVR, se kterým AVR Studio spolupracuje a pomocí něho vytváří hexadecimální program, který se nahrává do paměti programu procesoru.

Při použití těchto dvou programů je zapotřebí nejprve nainstalovat WinAVR a až poté AVR Studio, protože AVR Studio vyhledává vhodné nástroje pro spolupráci a pokud již máme nainstalované WinAVR, zvolí ho jako výchozí a bude s ním spolupracovat.

Realizace

V předchozích kapitolách bylo pojednáno o výběru jednotlivých částí zařízení. Na základě tohoto rozboru byly pro realizaci vybrány tyto komponenty:

- idlo pro měření hladiny - ultrazvukové převodníky UST-40R a UST-40T (kap. 2.5.2).
- idlo pro měření teploty - idlo SMT160 (kap. 3.1).
- řídící prvek - mikroprocesor ATmega162 (kap. 6.2).
- Vysílač - vysílač RC1280HP (kap. 1.4).

V následujících kapitolách bude popsána konkrétní realizace částí i celkového pojetí celého zařízení. Zejména nastavení registrů mikroprocesoru a části programu.

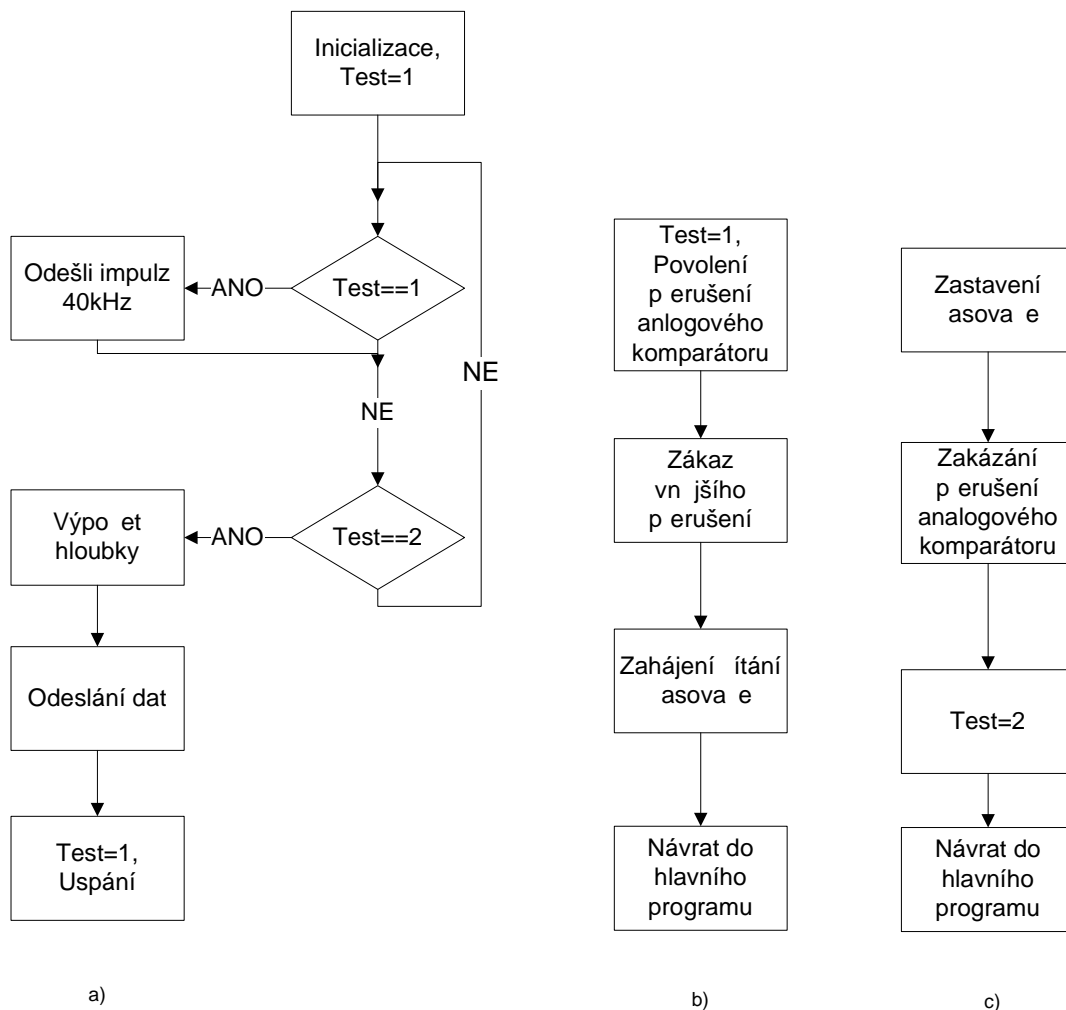
7 Napájecí napětí

Ultrazvukové převodníky budou připojeny k mikroprocesoru tak, jak je uvedeno v celkovém schématu, které je uvedeno v příloze. Oba převodníky budou umístěny ve stejné výšce vedle sebe. Tato měřicí část i s mikroprocesorem bude umístěna na zhruba 32 cm nad vodní hladinou, protože při větších vzdálenostech by docházelo k odrazům a ke zkreslení měření, popř. také k úplné ztrátě signálu. Bylo by možné převodníky umístit výše nad hladinu, ovšem bylo by zapotřebí použít vyšší napájecí napětí na napájení převodníku. Toto by bylo možné udělat tak, že by byla použita napájecí baterie s vyšším napájecím napětím (např. 9V, 12V). Mikroprocesor je však potřeba napájet napětím o hodnotě maximálně 5,5 V. Tento problém by byl vyřešen použitím stabilizátoru, který by napájecí napětí snížil. Vhodný stabilizátor je např. 7805[12]. Musí být bráno v potaz, že stabilizátor zvýší odběr celého zařízení i v případě, že budou všechna zařízení v úsporném režimu, proto bude tato varianta záviset na konzultaci se zákazníkem.

8 Konfigurace mikroprocesoru a periférií

V této kapitole bude pojednáno o konkrétním nastavení mikroprocesoru a jeho periférií, tak aby plnily funkce, které jsou požadovány.

8.1 Vývojové diagramy a popis programu



Obr.8. 1: a) hlavní program, b) vn jší p erušení, c) analogový komparátor

Jak je vidět z vývojových diagramů, celý program lze rozdělit na tři části a to hlavní program, obsluhu vn jšího p erušení a obsluhu p erušení analogového komparátoru. Hlavní smyčka není žádným způsobem složitá, jde pouze o testování jedné proměnné, která se nastaví podle toho, v jaké části programu se právě nachází. Zjednodušeně lze říci, že hlavní program obsahuje dvě smyčky. V první se provádí odesílání signálu a měří se čas, za jak dlouho se odražená vlna vrátí zpět, tento čas je úměrný vzdálenosti hladiny od idla. Ve druhé smyčce se provádí měření času na vzdálenost a data se odesílají. Jednotlivé části programu jsou podrobně popsány v následujících kapitolách.

8.2 Vn jší p erušení

Vn jší p erušení má v této realizaci významnou roli, protože celé měření je zahájeno právě obsluhou programu vn jšího p erušení, které je vyvoláno impulzem od obvodu reálného času.

P ed p íchodem tohoto impulsu (log0), na port INC0 je celý systém uspán a tímto impulsem se systém probudí a spustí m ění. Nastavení ídících registr ů vn ějšího p ěrušení p i inicializaci je uvedeno v následující tabulce (viz tab.8.1), v registru GICR, kde se vn ější p ěrušení povolují je nastaven do log1 bit INT0, který povolí vn ější p ěrušení 0, ostatní bity v tomto p ípad ě nemají význam.

Registr MCUCR je t ěeba nastavit tak, aby bylo vn ější p ěrušení vyvoláváno v p ípad ě, že na portu INT0 bude log0.

Tab.8. 1: Nastavení registru GICR

Bit:	7	6	5	4	3	2	1	0
GICR	INT1	INT0	INT2	PCIE1	PCIE0	-	IVSEL	IVCE
Hodnota:	-	1	-	-	-	-	-	-

Tab.8. 2: Nastavení registru MCUCR

Bit:	7	6	5	4	3	2	1	0
MCUCR	SRE	SRW10	SE	SM1	ISC11	ISC10	ISC01	ISC00
Hodnota:	-	-	-	-	-	-	0	0

Inicializace vn ějšího p ěrušení programov ě :

```
void Int0INIT()
{
    GICR|=
    (1<<INT0);           //povoleni preruseni 0
    MCUCR&=~           //preruseni se vyvola log0 na vstupu
    (1<<ISC00)|
    (1<<ISC01);
}
```

Pokud má být ve vn ějším p ěrušení zahájeno m ění, je pot ěeba, aby se zde provedly p íslušné kroky.

Prvním z nich je nastavení prom ěnné *test* na hodnotu 1, aby v hlavním programu po návratu z p ěrušení mohlo dojít k okamžitému generování 40kHz obdělníkověho signálu, který bude vysílán ultrazvukověm p ěvodníkem. Druhým krokem je povolení p ěrušení

analogového komparátoru, který bude takto oekávat impuls od odražené vlny. V poslední ad je třeba spustit íání íta e, abychom mohli m ít as, za který se odražená vlna vrátí a pomocí toho posléze vypo íst vzdálenost hladiny od ídel.

Obslužný program vn jšího p erušení je následující:

```
ISR(INT0_vect)
{
    sleep_disable();    //zakaz uspavani
    ACSR|=(1<<ACIE);   //povoleni preruseni komparatoru
    test=0x01;         //testovani v hlavnim programu
    TCCR1B|=(1<<CS10); //zapnuti casovace bez deleni
}
```

8.3 Generování m ícího signálu

V kapitole 4.5.2 je popsán ultrazvukový p evodník, který je použit v této realizaci. Z této kapitoly plyne, že tyto p evodníky fungují práv p í signálu, který má frekvenci 40kHz. Tento signál je třeba generovat pomocí mikroprocesoru.

Signál je generován ve smy ce, kdy se opakuje odesílání dvou period 40kHz signálu za sebou. Sta ilo by odeslat tyto dv perody pouze jednou, ovšem takovéto ešení bylo jednodušší. Signál se generuje až do té doby, dokud nedojde k p eklopení analogového komparátoru, který je nastaven na hodnotu v intervalu 50 – 100mV, což znamená, že došlo k p íchodu odražené vlny k p íjímacímu p evodníku.

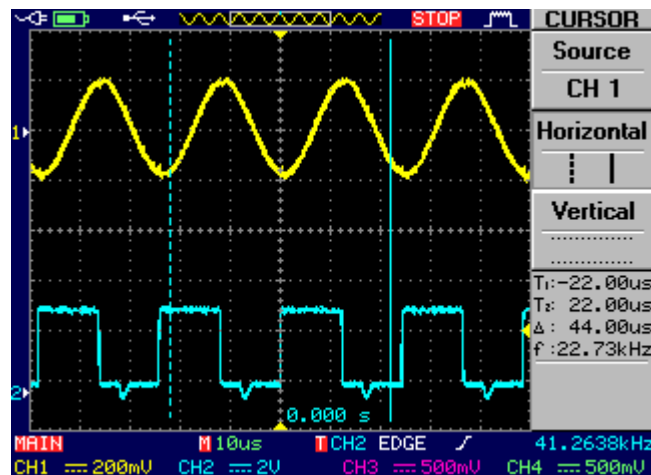
Generování signálu probíhá jednoduše st ídavým nastavováním logické jedni ky a nuly s ur ítým zpožd ním tak, aby vznikl obdélníkový signál o frekvenci 40kHz.

P í m ení zapojení p evodník , které bylo provád no za ú elem zjišt ní funk nosti tohoto zapojení, které lze vid t na obrázku 8.5 nebo v literatu e[6], byl signál generován p esným generátorem signálu. V tomto p ípad byl výstupní signál z p evodníku ideální. P í p ípojení mikroprocesoru docházelo k pom rn ílnému rušení na vysílacím í p íjímacím p evodníku, což komplikovalo m ení, zvlášt p í vyšší vzdálenosti p ekážky od p evodník , protože amplituda odraženého signálu již byla nižší a docházelo k zarušení tohoto signálu. Problém byl vy ešen p ídáním kondenzátoru o kapacit 4,7 μ F mezi napájecí nap tí a zem.

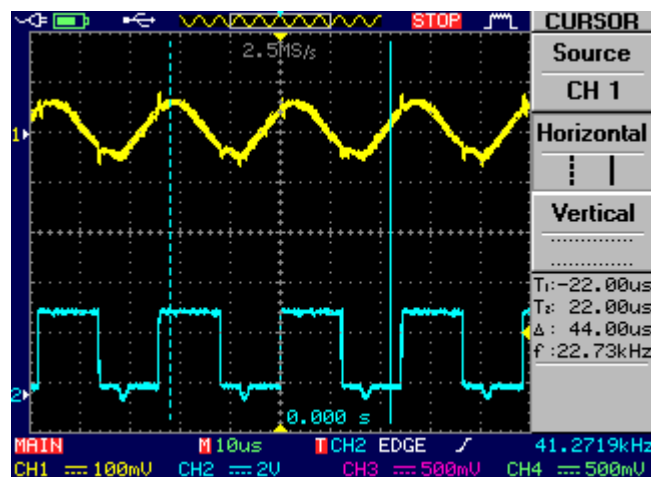
P í m eních byl také zjišt n ideální vysílací kmito et, p í kterém je odražený p íjímaný signál nejsiln jší. Tento kmito et je zhruba 41,26kHz, pokud je generován p esný 40kHz signál dochází k výraznému snížení p íjímané amplitudy. V p ípad generování frekvence takovéto hodnoty mikroprocesorem m že dojt k potížím s nastavením této hodnoty p esn . Proto je třeba tomuto nastavení a testování v novat pozornost. P í použití 8Mhz krystalu je možné generovat signál o frekvenci 41,24kHz. Toto je pom rn ídosta ující, pokud by však

bylo potřeba generovat frekvenci přesněji, musel by se použít krystal s vyšší frekvencí. Vysílací a přijímací charakteristiky lze vidět na obrázcích 8.2 – 3. Na obrázku 8.4 lze vidět, že je i při nepřijímaném signálu přítomno rušení, se kterým se musí počítat při nastavování hodnoty napětí na analogovém komparátoru (viz kap. 8.5).

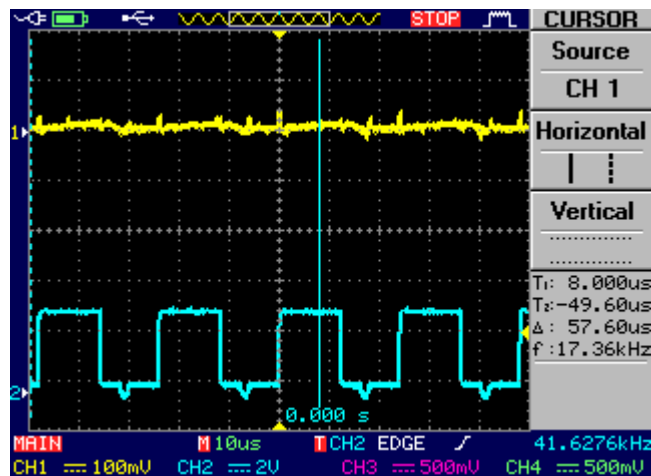
Přiblížení vstupního 41,26kHz signálu, který je vysílán do vysílacího převodníku a přichází odražená vlna na přijímacím převodníku jsou znázorněny na obrázku 8.2. Na obrázcích je vidět, že přijímaný signál má sinusový tvar.



Obr.8. 2: Vysílaný a přijímaný signál převodník při vzdálenosti 15cm

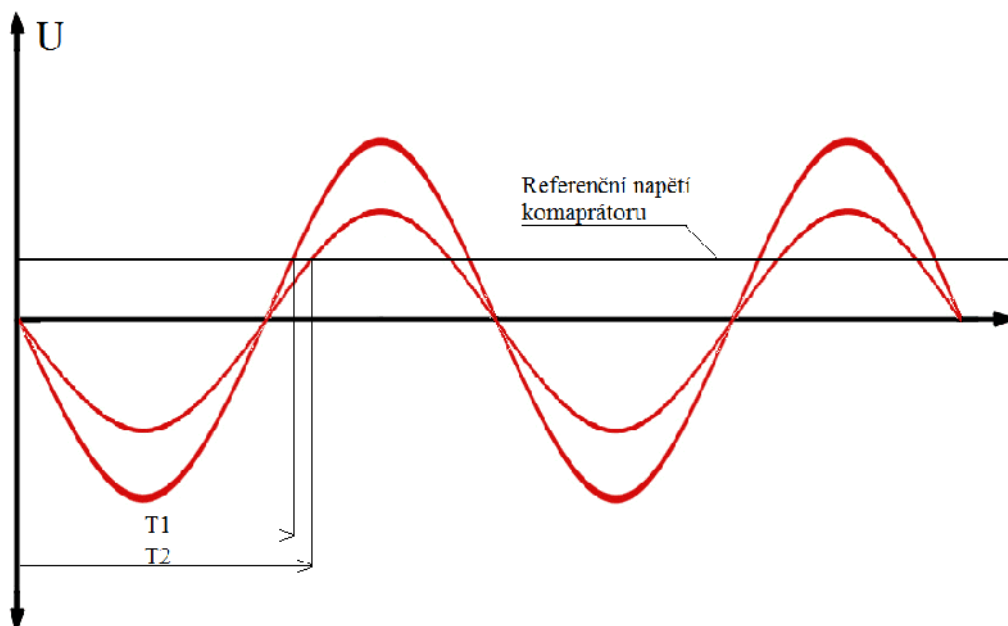


Obr.8. 3: Vysílaný a přijímaný signál převodník při vzdálenosti 30cm



Obr.8. 4: Vysílaný a přijímaný signál pro evodník, pokud není v jejich trase žádná překážka

Jak je vidět z přijímaného signálu, tak u vzdálenosti 30 cm je amplituda odraženého signálu nízká, což způsobuje problémy s překlopením analogového komparátoru při vyšších vzdálenostech. Pokud dojde k nastavení referenčního napětí analogového komparátoru na příliš nízkou hodnotu, dojde k okamžitému překlopění vlivem rušení. Proto je nastavení referenčního napětí určitý kompromis mezi přesností a maximální možnou měřenou vzdáleností. Stejně tak vzniká i chyba měření při větších vzdálenostech, kdy analogový komparátor přepíná až těsně u vrcholu amplitudy odraženého signálu. Toto způsobí delší měřicí dobu, než je tomu u vzdáleností nižších, resp. dosažení určité hodnoty napětí u větších vzdáleností dojde později, než u nižších. Tato chyba může dosahovat až 0,85 cm. Chyba je způsobena delší dobou měřicí asova (viz obr. 8.5).



Obr.8. 5: Chyba vznikající při měření různých vzdáleností

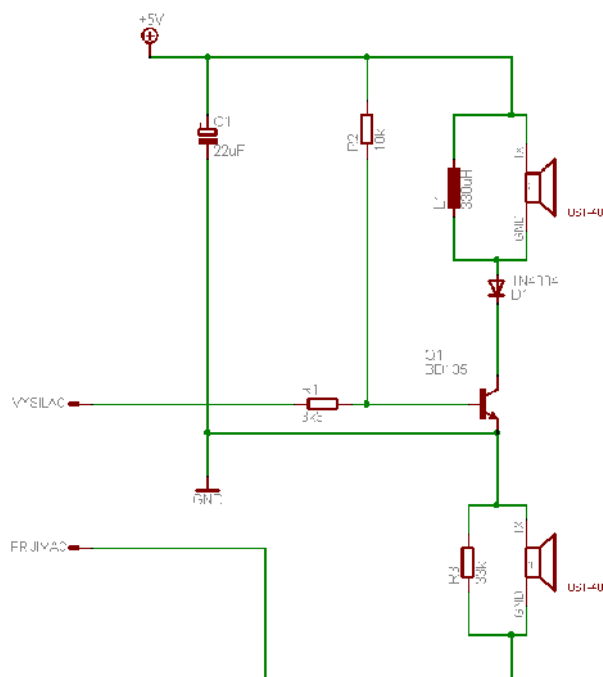
Programové vyjádření generování signálu je velice jednoduché, jediným problémem je nastavení správného zpoždění mezi impulzy, protože po odeslání dvou period dojde k návratu programu na začátek a znovu se testuje proměnná *Test*. Tzn., že zpoždění po posledním impulzu je třeba udělat kratší, aby zde byly započítány i další instrukce, které potom proběhnou.

```

if (test==0x01){           //odeslání impulsu
setb(PORTD,3);           //PORTD|=(1<<CidloOUTpin);
_delay_us(12.125);
clrb(PORTD,3);           //PORTD&=~(1<<CidloOUTpin);
_delay_us(12.125);
setb(PORTD,3);           //PORTD|=(1<<CidloOUTpin);
_delay_us(12.125);
clrb(PORTD,3);           //PORTD&=~(1<<CidloOUTpin);
_delay_us(12.125);
}

```

Funkce *clrb* a *setb* jsou zde použity na základě definic na začátku programu, mají stejnou funkci jako příkazy, které jsou za nimi uvedeny v komentářích.



Obr.8. 6: Schéma zapojení ultrazvukových převodníků

8.4 Měření času

K přesnému určení vzdálenosti hladiny vody od idla je třeba přesně změřit čas, za který se odražený signál vrátí k přijímacímu převodníku. K tomuto měření je využit jeden z 16 bitových řádů, který je integrován v mikroprocesoru.

Rychlost šíření je dána frekvencí oscilátoru mikroprocesoru, tato frekvence je 8MHz, což znamená, že řád odpovídá jedné klapce za 125ns, to je pro tento případ ideální. Pokud bude vzato v úvahu, že vzdálenost hladiny od idla je 1cm, tedy nejmenší možná měřitelná vzdálenost a bude-li po šíření s rychlostí zvuku zhruba $300\text{m}\cdot\text{s}^{-1}$, tak řád napočítá zhruba do hodnoty 500. Pokud bude vzdálenost maximální (30cm), hodnota řádu bude okolo 15000. Tzn., že 16 bitový řád, u kterého není použita žádná předlika je ideální.

8.4.1 Nastavení řádu

Nastavení registrů řádu pro tento případ je následující:

Tab.8. 3: Nastavení registru TCCR1A

Bit:	7	6	5	4	3	2	1	0
TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
Hodnota:	0	0	0	0	0	0	0	0

Tab.8. 4: Nastavení registru TCCR1B

Bit:	7	6	5	4	3	2	1	0
TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
Hodnota:	0	0	0	0	0	0	0	0

Oba řídicí registry zstanou nastaveny do logických nul, protože je třeba mít časovař nastaven do základního módu, kdy šíří impulsy od krystalu. Nastavení bitů WGM1-3 do logických nul je právě určení základního řádicího módu.

Bity CS10-3 určují, zda bude použita předlika a zároveň se časovař pomocí bitu CS10 zapíná. Při inicializaci však zůstává časovař stále vypnutý, zapínat se bude až při odeslání impulsu na vysílací ultrazvukový převodník.

Funkce pro inicializaci časovaře, která je provedena po restartu, která takto nastaví časovař je programově provedena takto:

```
void Timer1INIT()
{
```

```

TCCR1A&=~
(1<<WGM10)|          //mod 0
(1<<WGM11)|
(1<<COM1A1)|          //zadna reakce vystupu na preruseni
(1<<COM1A0);

TCCR1B&=~
(1<<WGM12)|          //mod 0
(1<<WGM13)|
(1<<CS10)|           //casovac vypnut
(1<<CS11)|
(1<<CS12);
}

```

Zapnutí asova se provede nastavením bitu CS10 do log1, toto bude provedeno v p erušení, které bude vyvoláno obvodem reálného asu a bude tím spušt no celé m ení. Nastavení do log1 nebo-li nahození, bitu se programov provede takto:

```
TCCR1B|=(1<<CS10); //zapnuti casovace bez deleni
```

8.4.2 Vyhodnocení asu

Jak již bylo e eno asova se spouští ve vn jším p erušení INTO a zastavuje se v p erušení od analogového komparátoru ANA_COMP. Vždy je t eba dbát na to, aby program v p erušení byl co nejkratší, proto se v p erušení od komparátoru asova pouze zastaví. Zpracování dat íta e probíhá poté v hlavním programu v rámci výpo tu hladiny.

Jelikož se jedná o asova , který je 16 bitový, hodnota která se má zpracovat je dvoubajtová. Hodnotu, která je uložena ve dvou registrech (TCNT1L, TCNT1H) je t eba uložit do jedné prom nné. Toto je možné provést pomocí ukazatele na prom nnou, kterým uložíme byte TCNT1L do prvních osmi bit výsledné prom nné a hodnotu registru TCNT1H do druhých osmi bit . Tím vznikne výsledný as jako 16 bitové íslo, se kterým je možno dále pracovat.

Zde je t eba dbát na to, jestli ukládání do pam ti probíhá tak, že se nejprve uloží byte s vyšší váhou nebo nižší. Toto se je dáno typem MCU. M že se jednat o typ ukládání „Little Endian“ – od nižšího, nebo „Big Endian“ – od vyššího. U mikroprocesoru ATmega162 se jedná o ukládání typu „Little Endian“, proto uložení prob hne takto:

```
char *pchPtr=(char*)&cas; //definice ukazatele na promennou cas, do ktere se
                          budou //ukladat dva byty casovace
```

```

unsigned int cas;           //definice promenne

...

pchPtr[0]=TCNT1L;         //dolni byte casu
pchPtr[1]=TCNT1H;         //horni byte casu
TCNT1L=0x00;              //vynulovani casovace
TCNT1H=0x00;

```

8.5 Analogový komparátor

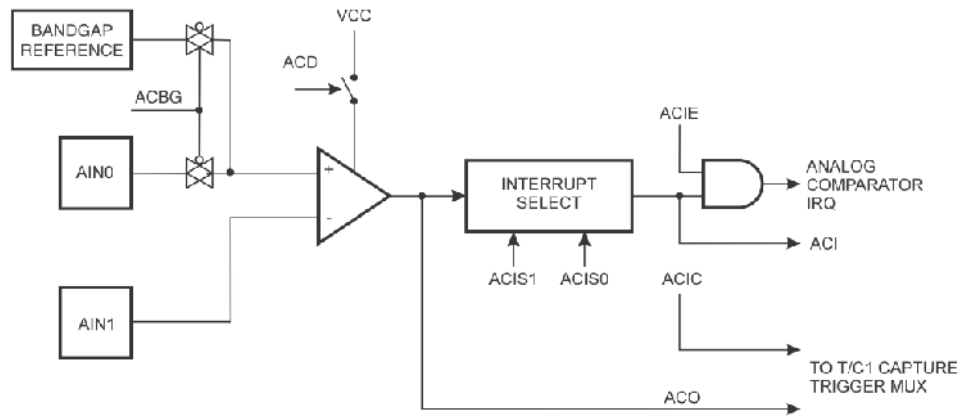
Analogový komparátor je v této realizaci, jak již bylo řečeno, použit k vyhodnocení příchodu odražené vlny od hladiny vody z přímého měření.

Analogový komparátor funguje na principu porovnávání dvou napětí. Je možné vybrat si zdroj napětí, které chceme porovnávat s měřeným. Schéma analogového komparátoru je znázorněno na obrázku 8.6. Jak je vidět, je zde možnost výběru mezi vnitřním referenčním napětím, nebo napětím na vstupu AIN0. V tomto případě bude použito napětí na vstupu AIN0, protože je třeba porovnávat napětí nižší, než je vnitřní referenční.

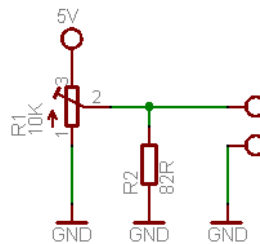
Napětí, které bude nastaveno na vstupu AIN0, bude zhruba 80mV. Toto napětí je zvoleno tak, aby nedocházelo k překlápění analogového komparátoru i v případě nepřijímaného signálu, tedy vlivem rušení, ale zároveň je nastaveno tak, aby bylo překlopení co nejrychlejší v případě příchodu odražené vlny. Nastavení napětí je provedeno pomocí napájecího děliče, jehož zapojení je na obrázku 8.7. Výpočet odporů děliče byl proveden pomocí následující rovnice.

$$U_2 = \frac{R_2}{R_1 + R_2} U_1 = \frac{82}{82 + 10000} \cdot 5 = 0,04V \quad (11.1)$$

Díky trimru, který byl použit v zapojení děliče je možné nastavovat napětí, které bude porovnáváno s napětím na příchodím ultrazvukovém převodníku. Při použití hodnot trimru a odporů tak, jak je uvedeno ve schématu na obrázku 8.7, je možné nastavovat hodnoty tohoto napětí od 40mV až do hodnoty téměř 5V. V tomto případě použití je požadovaná hodnota napětí 40-80mV. Velikost tohoto napětí se nastaví v závislosti na rušivých signálech, které budou na výstup přijímacího ultrazvukového převodníku přibíhat.



Obr.8. 7: Blokové schéma analogového komparátoru (Obrázek p evzat z katalogového listu mikroprocesoru [10])



Obr.8. 8: D li nap tí

Nastavení registr analogového komparátoru, které prob hne po resetu v inicializaci je uvedeno v následující tabulce.

Tab.8. 5: Nastavení registru ACSR

Bit:	7	6	5	4	3	2	1	0
ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
Hodnota:	0	0	0	0	0	0	1	1

Logická jedni ka v ACD by znamenala zakázání funkce analogového komparátoru. Bitem ACBG se volí již zmi ované nap tí, se kterým bude porovnáváno m ené nap tí. Pokud je bit nastaven do log1, je vybráno vnit ní referen ní nap tí. V tomto p ípad je t eba, aby bylo vybráno nap tí p ivedené na vstup AIN0, proto je tento bit nastaven do log0. Bit ACO je výstupem analogového komparátoru, není v tomto p ípad podstatný. Bit ACI je p íznakem p erušení, které nastane p í stavu, který je nastaven bity ACIS1 a ACIS0. Pomocí bitu ACIE se nastavuje povolení p erušení analogového komparátoru, v této realizaci je toto p erušení použito, ovšem p í inicializaci není povoleno. Toto p erušení se povoluje až v obslužném programu pro vn jší p erušení, které je popsáno dále. Bit ACIC slouží k zachycení stavu asova e1 do záchytného registru, v tomto p ípad tato funkce není využita,

proto je bit nastaven do log0. Bity ACIS1 a ACIS2 se, jak již bylo řečeno, nastavuje podmínka, při jejímž splnění dojde k vyvolání obslužného programu analogového komparátoru. V tomto případě, kdy jsou oba bity nastaveny do log1 je podmínka nastavena tak, že při erušení komparátoru se vyvolá, pokud napětí na portu AIN1 překročí hodnotu na portu AIN0. Toto překročení v této realizaci znamená návrat odražené vlny k přijímacímu převodníku.

Inicializace analogového komparátoru je programově provedena takto:

```
void ACINIT()
{
  ACSR|=
  (0<<ACD)| //zapnutí komparatoru
  (0<<ACBG)| //kladny vstup je AIN0
  (0<<ACIE)| //zakazani preruseni komparatoru, aby nedoslo k jeho vyvolani při
  // nastavovani ACIS
  (0<<ACIC)| //zakazani spusteni casovace
  (1<<ACIS1)| //preruseni vyvolano prekrocenim hodnoty na ref vstupu
  (1<<ACIS0);
}
```

Obslužný program pro přerušení analogového komparátoru je následující:

```
ISR(ANA_COMP_vect)
{
  TCCR1B&=~(1<<CS10); //vypnutí casovace
  ACSR&=~(1<<ACIE); //vypnutí komparatoru
  ACSR&=~(1<<ACI); //vynulovani priznaku
  test=0x02;
  sleep_enable();
}
```

Jak lze v programu pro obsluhu přerušení komparátoru vidět, dojde nejprve k zastavení časovače, jehož hodnota koresponduje se vzdáleností hladiny od idla. Dále je samotný komparátor vypnut, aby nebylo přerušení vyvoláno vícekrát vlivem více odeslaných vln. Z tohoto důvodu je také vynulován příznak komparátoru, protože s největší pravděpodobností, vlivem dalších odeslaných impulzů došlo k jeho dalšímu nastavení dříve, než byl komparátor vypnut. Pokud by byl takto příznak nastaven, došlo by při dalším zapnutí komparátoru k okamžitému vyvolání tohoto přerušení a měření by bylo nepřesné. V této části programu se také nastaví proměnná *test* na hodnotu 2. Tato hodnota je testována v hlavní programové smyčce, a pokud je takto nastavena program zahájí po návratu z tohoto přerušení

výpočet vzdálenosti. Také je zde opět povoleno uspání procesoru, který bude uspán, po provedení měření a odeslání dat.

8.6 Výpočet vzdálenosti

Princip výpočtu vzdálenosti hladiny vody od čidla, resp. hloubka hladiny je zřejmý již z výše uvedených kapitol. Nyní bude uveden přesný postup.

Samotnému výpočtu předchází změna prostředí, za který ultrazvuková vlna urazí vzdálenost od vysílacího převodníku k přijímacímu. Z tohoto údaje bude celý výpočet vycházet.

Nejprve je však nutné spočítat si přesnou rychlost zvuku, která odpovídá dané teplotě, na které je rychlost zvuku nejvíce závislá. Popis změny teploty je uveden v následující podkapitole. V první řadě je tedy změna teplota a na základě této hodnoty je možno vypočítat přesnou rychlost zvuku ze vzorce 4.2. Pokud je takto vypočtená rychlost zvuku, přejde na řadu samotný výpočet vzdálenosti:

$$s = \frac{(331,8 + 0,61 \cdot \tau) \cdot H}{160000} = \frac{(331,8 + 0,61 \cdot 20) \cdot 14000}{160000} = 29,225 \approx 29 \text{ cm}$$

V uvedené rovnici je proveden výpočet vzdálenosti čidla od hladiny. Teplota okolí je zde označena τ a její hodnota je v tomto případě 20°C . Symbol H zde značí hodnotu času a v době příchodu odražené vlny, tuto hodnotu lze chápat jako násobek 125ns , perioda měření se

z frekvence krystalu vypočte jako $\frac{1}{8000000}$. Vše je podléno hodnotou 160000 . Tato hodnota

vychází z předpokladu, že hodnota H , která má v tomto případě hodnotu 14000 , odpovídá času $0,00175\text{s}$. Proto ji musíme podlélit 8000000 a to dvakrát, protože vlna urazila cestu právě dvakrát. Ovšem takto bychom hodnotu dostali v metrech, pro tuto realizaci je výhodnější výpočítat hodnotu v centimetrech, proto je celé číslo děleno právě 160000 .

Programově je tento výpočet proveden následovně :

```
unsigned char Vzdalenost(long int a)
{
    unsigned int PresTeplota, SprRychl;
    float rychl=331.8, rychlK=0.61;
    teplota=Merto();
    PresTeplota=(teplota-1000)/10;
    SprRychl=(unsigned int)(rychl+((float)(rychlK)*(float)(PresTeplota)));
    vzdalenost=(unsigned int)((SprRychl*a)/160000);
    return vzdalenost;
}
```

8.7 Měření teploty

Měření teploty se u tohoto zařízení provádí z již zmíněného důvodu vlivu teploty na rychlost zvuku. V tomto případě, kdy je přesnost měření 1cm, není příliš velký rozdíl ve vzdálenosti pro různé teplotách. A však ten který z uživatel by mohl potřebovat měření přesnější a v tomto případě by se již jednalo pro měření hladiny bez ohledu teplotního neobešlo. Toto ohledu zde však může fungovat i jako zdroj informací o teplotě ve studni.

Měření teploty je prováděno pomocí teplotního ohledu SMT160[7], jehož základní princip je popsán v kapitole 3.1.

Princip měření u tohoto ohledu je změření střídy signálu, který ohledu vysílá. Toto je provedeno tak, že signál je po určité dobu testován, zda je v log1, pokud ano přičemž do určité proměnné, v tomto případě *sh*, jedničku. Stejně dlouhou dobu, resp. stejným po tem vzorkování testujeme signál na log0, pokud je signál v log0, přičemž jedničku do jiné proměnné, v tomto případě *sl*. Po dokončení tohoto vzorkování spočítáme hodnotu korespondující naměřené teplotě *t*, pomocí vzorce 5.1 a to takto:

$$t = \frac{\frac{sh}{sh+sl} - 0,32}{0,00047}$$

Jak je vidět z výpočtu, dělení je provedeno číslem o řád menším, než je uvedeno v rovnici 5.1. Je to z důvodu přesnějšího výsledku, proto je třeba tuto hodnotu při použití ve výpočtech podílit deseti. Toto měření je provedeno pětkrát a výsledek je vždy přičten k předchozí hodnotě odpovídající teplotě, proto je třeba na konci měření tuto hodnotu vydělit pěti. Aby nemusela být teplota uložena jako typ float (datový typ s plovoucí čárkou), je k hodnotě teploty přičtena hodnota 1000. Tzn., že například hodnotě 23,5°C bude odpovídat hodnota 1235.

Program pro měření teploty je následující:

```
unsigned int Merto() {
    unsigned int sh, sl, opakovani, vystup;
    unsigned char i=5, th=0x80;

    float teplota=0;
    cli();
    while(i--) {
        sh=0;
        opakovani=0xFFFF;
        while(opakovani--) if(PINA&th) sh++;
    }
}
```

```

        sl=0;
        opakovani=0x0FFFF;
        while(opakovani-- if(!(PINA&th)) sl++;
        teplota+=(int)((((float)sh/(float)(sh+sl))-0.32)/0.00047);
    }
    sei();
    vystup=(int)(teplota/5)+1000;
    if((vystup<700) || (vystup>2200)) vystup=0;
    return vystup;
}

```

8.8 Režimy spánku

Vzhledem k tomu, že základním požadavkem na měření je nízká spotřeba celého zařízení, je potřeba, aby měření probíhalo pouze v určitých intervalech. Zároveň budou části zařízení v době neinnosti usnány. V režimu spánku je spotřeba za řízení několikrát menší.

Komponenty, které byly vybírány na konstruování zařízení na měření hladiny vody, byly voleny tak, aby měly co nejmenší spotřebu. Toho lze docílit právě různými režimy spánku. U mikrokontroléru ATmega162 je hned několik režimů spánku[8], pro naše účely je zvolen režim power-down, který je popsán v kapitole mikrokontroleru.

Nastavení režimu spánku mikrokontroleru:

V tomto případě byl jako nejvhodnější režim spánku vybrán mód „Power down“, který má spotřebu nižší než $1\mu\text{A}$. Z tohoto režimu se mikroprocesor do režimu měření dostane po impulsu na vnější přerušení. Tento impuls je generován pomocí obvodu reálného času RTC63421[8].

Jelikož je použita knihovna pro režimy spánku, tak je nastavení režimu power down u mikroprocesoru následující:

```

#include <avr/sleep.h>          //knihovna rezimu spanku

...
//nastaveni rezimu spanku
void SleepINIT()
{
    sleep_enable();             //povoleni uspani
    set_sleep_mode(SLEEP_MODE_PWR_DOWN); //rezim uspani Power-Down
}

```

Pokud je režim spánku mikroprocesoru takto nastaven sta í potom v programu tam, kde chceme mikroprocesor uspat napsat p íkaz `sleep_cpu()`, který procesor uspí. Procesor bude v režimu spánku až do té doby, dokud nep ijde na jeho vstup vn jšího p erušení impuls od obvodu reálného asu. Samoz ejmostí je, že port, na kterém je vn jší p erušení, musí být nastaven jako vstupní a vn jší p erušení musí být povoleno.

Nastavení režimu spánku RC1280HP:

U vysílacího za ízení RC1280HP je nastavení režimu spánku také velmi d ležité, protože jeho spot eba v aktivním režimu je relativn vysoká. P í vysílání dosahuje odebíraný proud hodnoty až 600mA. Vysíla RC1280HP je však velice inteligentní za ízení, které samo p ejde do režimu spánku, pokud nemá na vstupu žádná data k vysílání. V tomto režimu má spot ebu 0,9mA. Tuto hodnotu proudu však je možno ješt snížit, pokud za ízení úpln vypneme. To lze jednoduše provést nastavením jednoho pinu za ízení [2], tímto se spot eba sníží na úplné minimum, které je 0,003 μ A. Nastavením tohoto pinu za ízení op t zapneme a je možno odesílat data.

8.9 Frekvence m ení

M ení bude probíhat jednou za hodinu, což bude pro m ení výšky hladiny vody ve studni zcela dosta ující. Tento interval bude možno také zkrátit, nastavení delšího intervalu by však bylo pon kud komplikovan jší, protože by bylo t eba použít jiného obvodu, než je obvod reálného asu RTC62421, pomocí kterého je hodinový interval m ení nastaven.

Nastavení obvodu reálného asu se provádí zápisem do systémových registr , konkrétn se jedná o registr E. V tomto registru se nastavuje perioda generování impulsu na port STD.P. Tento interval lze nastavit maximáln na jednu hodinu, ehož bude v tomto p ípad využito.

Nastavení tohoto registru je následující:

Tab.8. 6 Nastavení registru obvodu reálného asu C_E

Bit:	7	6	5	4	3	2	1	0
C_E	A3	A2	A1	A0	T1	T0	INTRP	MASK
Hodnota:	1	1	1	0	1	1	1	0

Toto nastavení prob hne pouze jednou p í prvním p ípojení napájecího nap tí k mikroprocesoru, pop ípad p í resetu.

8.10 Odesílání dat

Data, která budou jednou za hodinu zmeřena, budou odeslána do vyhodnocovací stanice. Odesílání probíhá pomocí vysílače RC1280HP, který bude k mikroprocesoru připojen pomocí sériové linky RS232, jak je uvedeno ve schématu v příloze. Tento vysílač bude umístěn u povrchu země, tzn., že propojovací kabel bude dlouhý 4 m, což by neměl být problém.

Tento vysílač není potřeba žádným způsobem nastavovat, veškerá data, která jsou do něj po sériové lince posílána, jsou ihned odesílána.

Z důvodu vysoké ceny nebyl vysílač zakoupen k otestování. Navíc vysílač kopíruje data, která přijme a odesílá je do vzdálené stanice, takže nebylo zapotřebí jej nijakým způsobem testovat.

8.10.1 Komunikační protokol

Jelikož se jedná o systém, který bude obsahovat více měřících stanic, je potřeba navrhnout komunikační protokol, kterým se jednotlivé stanice od sebe odliší. Data, která bude třeba odeslat, budou mít velikost dvakrát 7 bitů. Vysílat se však bude dvakrát 8 bitů, aby se vysílání nekomplikovalo. Odesílané naměřené hodnoty s číslem židla budou tedy sedmi bitové a nejvyšší osmý bit bude mít hodnotu nula, se kterou se nebude počítat.

Z důvodu, že v tuto chvíli není známo, jakým způsobem bude přijímací stanice data vyhodnocovat, je použit na odesílání dat samoopravný kód, který bude schopen odhalit a opravit jednu chybu ve čtyřech bitech. Jedná se o hammingův kód typu (7,4), což znamená, že se přenáší 4 bity dat a 3 nadbytečné bity, které slouží k nalezení a opravě chyby. Tento kód byl zvolen proto, že je v tuto chvíli předpokládáno, že komunikace mezi židlem a přijímací stanicí bude jednosměrná. Tzn., že nebude možno vyžádat špatně přijatá data od židla znovu. Proto byl zvolen právě hammingův kód, který i při chybovosti přenosového kanálu, která bude při přenosu dat určité procenta, zaručí opravu jedné chyby ve 4 bitech.

Hodnoty kontrolních bitů p_1 - p_3 , které jsou uvedeny v tabulce 8.7, byly vypočteny z následujících rovnic:

$$p_1 = b_4 \oplus b_3 \oplus b_1 \quad (8.1)$$

$$p_2 = b_4 \oplus b_2 \oplus b_1 \quad (8.2)$$

$$p_3 = b_3 \oplus b_2 \oplus b_1 \quad (8.3)$$

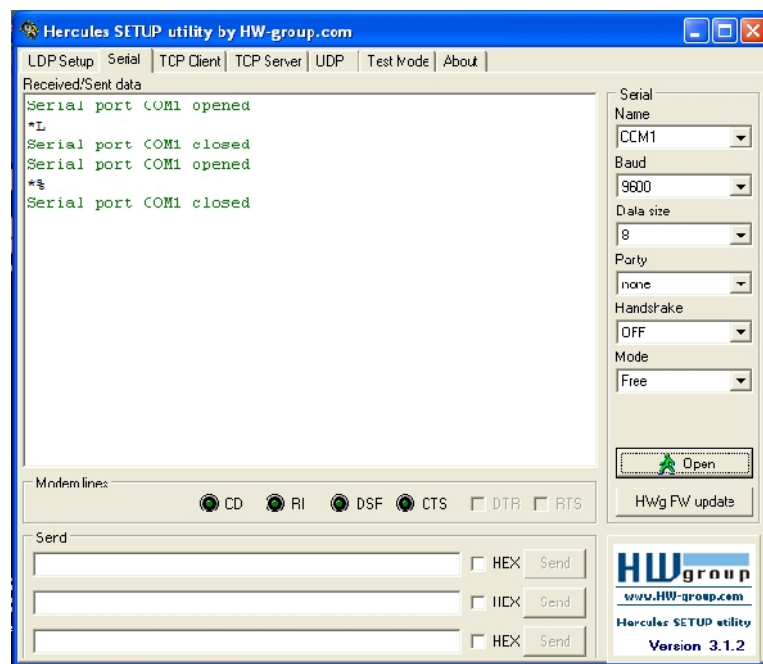
Pro vytvoření protokolu a uspořádání dat se bude vycházet z mírně zjednodušené formy návrhu. V předchozích kapitolách bylo uváděno, že měření bude možno provádět na zhruba deseti bezdrátových židlech pro měření hladiny. Tento fakt by znamenal vyšší složitost při vytváření protokolu, avšak princip zůstane stejný, proto bude v tomto návrhu protokolu

uvažováno maximálně 8 bezdrátovýchidel. Tento počet vychází z počtu bitů, které budou kódovány. Je-li totiž rozsah naměřených dat 0-30 (rozsah změny hladiny vody), bude tato hodnota kódována 5 bity, pokud jsou přidány další 3 bity identifikující idlo, výsledkem je 8 bitů, které je třeba zakódovat.

Princip kódování je jednoduchý, 8 bitů je nejprve rozděleno na dvě části, každá o 4 bitech, která je každá zvlášť zakódována. Z toho předpoklad vyplývá, že bude použit hamming v kódu typu (7,4).

Z rovnic 8.1 – 3 jsou vypočteny paritní bity, pomocí kterých je vytvořena následující tabulka (8.7). V tabulce je možno vidět, jakým hodnotám naměřené hodnoty odpovídají data, která budou odesílána.

Na obrázku 8.8 jsou vidět vysílaná data z mikroprocesoru. V tomto případě je zadáno idlo s identifikačním číslem 1 (binárně 001) a naměřená vzdálenost je v prvním případě 4 cm a ve druhém 5 cm. Při vzdálenosti 4 cm bude mít vysílaný bajt tvar 00100100, kde první tři bity označí idlo a zbylých pět bitů je naměřená vzdálenost. Program Hercules, který byl k zachycení vysílaných dat použit, zobrazuje data jako ASCII kódy. Pokud bude bajt rozdělen na dvě části, výše zmíněným postupem, vyjdou dvě čísla a to 2 a 4. Tato jsou dále převedena na samoopravný kód podle tabulky 8.7 a odesílána. Z tabulky je zřejmé, že hodnotám 2 a 4 odpovídají znaky „*“ a „L“, pokud bude vzdálenost 5, budou odpovídající znaky „*“ a „%“.



Obr.8. 9: Zachycená odesílaná data

Tab.8. 7: Hodnoty hammingova kódu a jejich znaky ASCII

Hodnota	Odesílaná data							Dec	Znak
	p ₁	p ₂	b ₄	b ₃	p ₃	b ₂	b ₁		
0	0	0	0	0	0	0	0	0	
1	1	1	0	1	0	0	1	105	i
2	0	1	0	1	0	1	0	42	*
3	1	0	0	0	0	1	1	67	C
4	1	0	0	1	1	0	0	76	L
5	0	1	0	0	1	0	1	37	%
6	1	1	0	0	1	1	0	102	F
7	0	0	0	1	1	1	1	15	
8	1	1	1	0	0	0	0	112	P
9	0	0	1	1	0	0	1	25	
10	1	0	1	1	0	1	0	90	Z
11	0	1	1	0	0	1	1	51	3
12	0	1	1	1	1	0	0	60	<
13	1	0	1	0	1	0	1	85	
14	0	0	1	0	1	1	0	22	
15	1	1	1	1	1	1	1	127	

8.10.2 USART

Data jsou z mikroprocesoru odesílána, jak již bylo zmíněno přes sériovou linku RS232, přes sériový kanál USART.

Pro nastavení sériového kanálu na požadovanou funkci slouží tři řídicí registry UCSR0A, UCSR0B a UCSR0C. Dále je zde registr UDR, do kterého se ukládají data, která mají být odeslána.

V registru UCSR0A jsou pro tuto realizaci dležité dva bity, prvním z nich je pátý bit UDRE, který se nastaví do log1, pokud je vysílací bajt UDR prázdný, tohoto je využito při kontrole zda je možné odeslat další bajt dat. Druhým bitem, který je dležitý je bit U2X, kterým je nastavena dvojnásobná rychlost přenosu, ovšem v této realizaci tohoto není využito. Nastavení registru UCSR0A je uvedeno v tabulce 8.8. Další bity v tomto registru slouží k nastavování příznaků chyb a poslední dva bity RXC a TXC se nastaví jako příznak odeslaných a přijatých dat.

Tab.8. 8: Nastavení registru UCSR0A

Bit:	7	6	5	4	3	2	1	0
<u>UCSR0A</u>	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0
Hodnota:	0	0	1	0	0	0	0	0

Registr UCSR0B obsahuje devátý bit příjmu při vysílání, pokud je použit devítibitový formát dat (TXB8, RXB8). Bit UCSZ02 slouží společně s bity UCSZ01 a UCSZ00, které jsou v registru UCSR0C, právě k nastavení formátu dat, takže se jedná o 5-9 bitů dat. V tomto případě bude použit 8 bitový formát. Bity RXEN a TXEN se povoluje vysílání a příjem dat. Posledními třemi bity se nastavuje povolení přerušení, při vyslání dat, při přijetí dat a při vyprázdnění registru UDR. Nastavení registru v této realizaci ukazuje tabulka 8.9.

Tab.8. 9: Nastavení registru UCSR0B

Bit:	7	6	5	4	3	2	1	0
<u>UCSR0B</u>	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80
Hodnota:	0	0	0	0	1	0	0	0

V registru UCSR0C je nejdležitější bit URSEL0, kterým se určí, zda se budou nastavovat data v registru UCSR0C, nebo v UBRR0H. Jelikož je potřeba nastavit registr UCSR0C, tak je tento bit nastaven do log1. Další bity, které se nastaví, jsou již zmínované UCSZ01 a UCSZ00. Pokud má být zvolen 8 bitový formát dat, bude nastavení registru UCSR0C takové jako ukazuje tabulka 8.10.

Tab.8. 10 Nastavení registru UCSR0C

Bit:	7	6	5	4	3	2	1	0
UCSR0C	URSEL0	UMSEL0	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0
Hodnota:	1	0	0	0	0	1	1	0

Nastavení registr programov je následující:

```
void Init_USART0()
{
    UCSR0A&=~
    (1<<U2X0)|
    (1<<MPCM0);
    UCSR0B|=
    (1<<TXCIE0)|
    (1<<RXEN0)|
    (1<<TXEN0);
    UCSR0C|=
    (1<<URSEL0)|
    (1<<UCSZ00)|
    (1<<UCSZ01);
    UBRROH=0x00;
    UBRROL=0x33;    //rychlosst 9600 Bd
}
```

Jak je z části programu vidět, musela být nastavena i přenosová rychlost. Nastavení se provádí pomocí registr UBRROH a UBRROL, jejich nastavení pro různé frekvencích krystalu je uvedeno v katalogovém listu procesoru [10]. V tomto případě je přenosová rychlost, které je uvedeno v části programu, je nastavena rychlost 9600Bd/s.

Vysílání dat je potom velice jednoduché, protože mikroprocesor automaticky odesílá data, která mu jsou vložena do registru UDR. V tomto případě jsou, jak již bylo zmíněno, odesílány dva bajty dat. Vysílání tedy probíhá tím způsobem, že je kontrolován příznak vyprázdnění registru UDR, což je bit UDRE0 v registru UCSR0A. Funkce pro odeslání jednoho bajtu dat je uvedena v následující části programu. Pokud mají být odeslány dva bajty dat, stačí tuto funkci zavolat dvakrát po sobě, s příslušnými vstupními daty.

```
void u_putc( char data )
{
```

```
while ( !( UCSRA & (1<<UDRE0)) ); // cekej do vyprázdnění bufferu
UDR0 = data;
}
```

9 Závěr

Cílem diplomové práce byl návrh a realizace bezdrátového měřidla pro měření hladiny vody.

Jako měřidla pro měření hladiny byly zvoleny ultrazvukové převodníky, prostřednictvím kterých mikroprocesor měří výšku hladiny. Takto navržené měřidlo měří vzdálenost hladiny od měřidel, pokud by mělo udávat výšku hladiny, muselo by se tomu měřidlo přizpůsobit. Z provedených měření vyplývá, že měření vzdálenosti od hladiny, provedené tímto způsobem se hodí spíše na vzdálenosti nižší, než je 30 cm. Při 30 cm vzdálenosti hladiny od měřidel docházelo k problémům měření a vznikala poměrně vysoká chyba měření.

Naměřená data jsou odesílána do vzdálené přijímací stanice v intervalu 1 hodiny. Jelikož náplní této práce nebylo řešení přijímací stanice a není znám způsob přijmu dat, jsou odesílaná data kódována samoopravným kódem.

Data jsou vysílána vysílačem RC1280HP, který nebyl v realizaci použit, kvůli vysoké ceně. Vysílač však nebylo třeba testovat, protože pouze kopíruje přijatá data.

Pomocí tohoto systému bude možno připojit několik měřidel, která budou měřit hladinu vody a všechna budou odesílat data do jedné přijímací stanice, která může být od měřidel, při přímé viditelnosti, vzdálena až 2,5km.

Cíl diplomové práce bylo dosaženo, ovšem pouze v laboratorních podmínkách.

Seznam literatury

- [1] Spezial Electronic [online]. 2006 , 10.2007 [cit. 2009-12-02]. Dostupný z WWW: <http://www.spezial.cz/pdf/em_ds_oemspa_311_331.pdf>.
- [2] VOJÁ EK, Antonín. Bezdrátová komunikace z RS-232/485 - modul RC1280HP | HW.cz [online]. 2007 , 7. únor 2007 [cit. 2009-12-05]. Dostupný z WWW: <<http://hw.cz/Produkty/Obecne-produkty/ART1810-Bezdratova-komunikace-z-RS-232-485--modul-RC1280HP.html>>.
- [3] A O, Stanislav, BEJ EK, Ludvík, PLATIL, Antonín. Měření průtoku a výšky hladiny. 1. vyd. Praha : BEN, 2005. 448 s. ISBN 80-7300-156-X.
- [4] VOJÁ EK, Antonín . Senzory tlaku Fujikura | Automatizace.HW.cz [online]. 2005 , 22 Kv ten, 2005 [cit. 2009-12-14]. Dostupný z WWW: <<http://automatizace.hw.cz/mereni-a-regulace/ART155-senzory-tlaku-fujikura.html>>.
- [5] VOJÁ EK, Antonín. Senzory pro měření tlaku v integrovaném provedení | Automatizace.HW.cz [online]. 2005 , 14 Srpen, 2005 [cit. 2009-12-14]. Dostupný z WWW: <<http://automatizace.hw.cz/mereni-a-regulace/ART176-senzory-pro-mereni-tlaku-v-integrovanem-provedeni.html>>.
- [6] Www.ges.cz [online]. c2010 [cit. 2010-05-20]. UST-40R-GES-ELECTRONICS. Dostupné z WWW: <<http://www.ges.cz/sheets/u/ust40.pdf>>.
- [7] Pdf1.alldatasheet.net [online]. c2010 [cit. 2010-05-20]. SMT 160. Dostupné z WWW: <<http://pdf1.alldatasheet.net/datasheet-pdf/view/90332/ETC/SMT160.html>>.
- [8] Pdf1.alldatasheet.net [online]. c2010 [cit. 2010-05-20]. RTC62421. Dostupné z WWW: <<http://pdf1.alldatasheet.net/datasheet-pdf/view/89187/EPSON/RTC62421.html>>.
- [9] Pdf1.alldatasheet.net [online]. c2010 [cit. 2010-05-20]. MAX232. Dostupné z WWW: <<http://pdf1.alldatasheet.net/datasheet-pdf/view/73047/MAXIM/MAX232.html>>.
- [10] Atmel corporation [online]. c2009 [cit. 2009-12-12]. Datasheet k procesoru ATmega 162 . Dostupný z WWW: <http://www.atmel.com/dyn/resources/prod_documents/doc2513.pdf>.
- [11] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR*. 2. vyd. Praha : Nakladatelství BEN – technická literatura, 2006. 375 s., 1 CD-ROM. ISBN 80-7300-209-4.
- [12] Pdf1.alldatasheet.net [online]. c2010 [cit. 2010-05-20]. 7805. Dostupné z WWW: <<http://pdf1.alldatasheet.net/datasheet-pdf/view/106291/ETC/7805.html>>.

Seznam použitých zkratek

h – hloubka hladiny

– hustota kapaliny

g – tíhové zrychlení

P – hydrostatický tlak

c – rychlost sv tla

– adiabatická konstanta

R – plynová konstanta

T – termodynamická teplota

LED – sv tlo vyza ující dioda (Light-Emitting Diode)

TTL – tranzistorov tranzistorová logika

ALU – aritmeticko-logická jednotka (Arithmetic Logic Unit)

SRAM – statická pam (Static Random Access Memory)

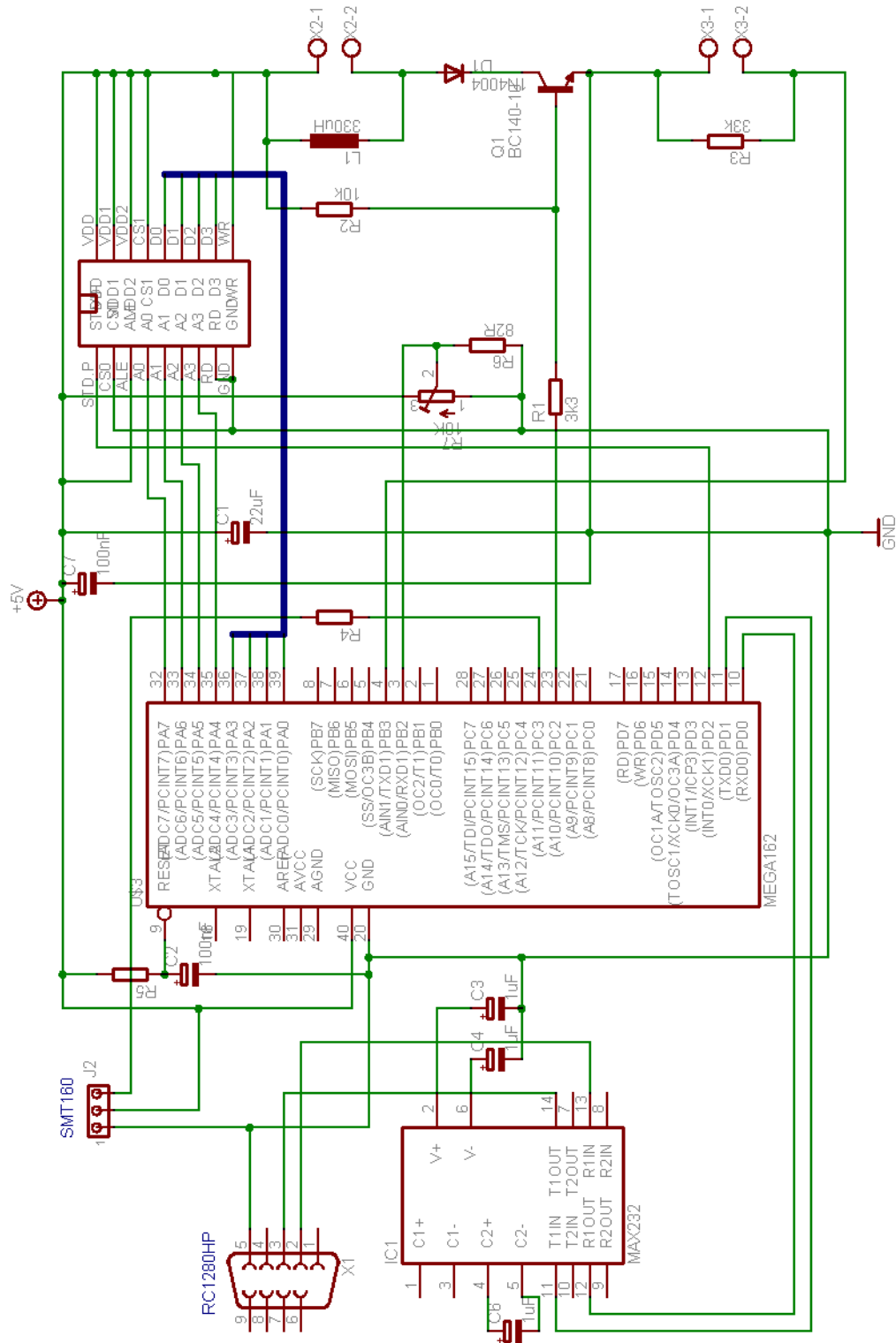
CPU – ídíčí jednotka (Central Processor Unit)

SPI – sériové programovací rozhraní (Serial Interface Programming)

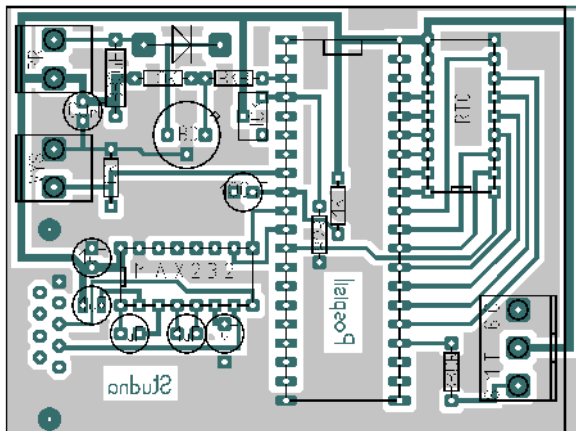
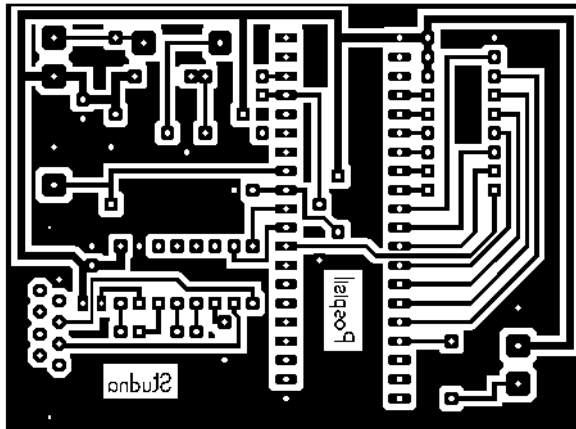
Seznam příloh

Celkové schéma.....	56
Deska plošných spojů a rozložení součástek.....	57
Realizace zařízení na nepájecím poli	58
Obsah příloženého CD	59

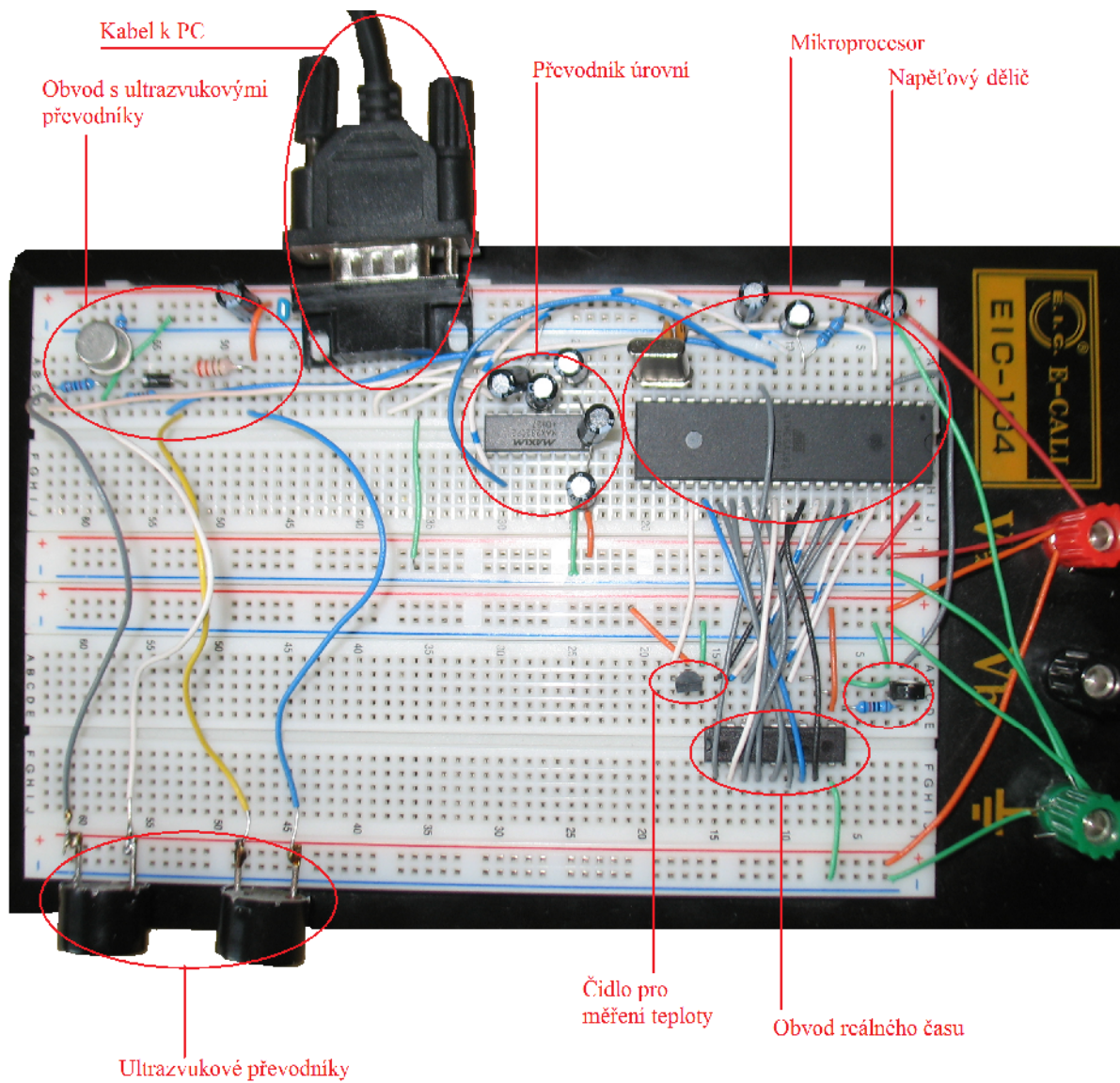
Celkové schéma



Deska plošných spoj a rozložení součástek



Realizace za řízení na nepájivém poli



Obsah přiloženého CD

Elektronická verze diplomové práce

Programové vybavení mikroprocesoru ATmega162

Program Hercules – program pro příjem dat po sériové lince RS232

Katalogový list mikroprocesoru ATmega162

Katalogový list obvodu reálného času RTC62421

Katalogový list čidla pro měření teploty SMT160