

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

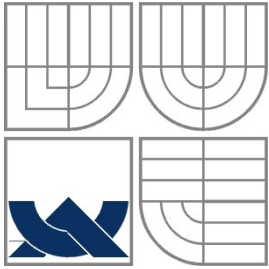
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

LOKALIZACE AKUSTICKÝCH ZDROJŮ METODOU TDOA

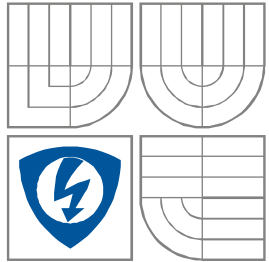
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JAN BERNKOPF
AUTHOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A
KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

LOKALIZACE AKUSTICKÝCH ZDROJŮ METODOU TDOA ACOUSTIC SOURCES LOCALIZATION USING TDOA METHOD

SEMESTRÁLNÍ PROJEKT
BACHELOR'S PROJECT

AUTOR PRÁCE
AUTHOR

Jan Bernkopf

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Jiří Šebesta, Ph.D.

BRNO, 2008

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Jan Bernkopf
Bytem: Na Jámě 15, Mikulov, 69201
Narozen/a (datum a místo): 4. listopadu 1985 ve Valticích

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
 - diplomová práce
 - bakalářská práce
 - jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Lokalizace akustických předmětů metodou TDOA

Vedoucí/ školitel VŠKP: Ing. Jiří Šebesta, Ph.D.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- v tištěné formě – počet exemplářů: 2
- v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy
(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 6. června 2008

.....
Nabyvatel

.....
Autor

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Lokalizace akustických předmětů metodou TDOA jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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 6. června 2008

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Jiřímu Šebestovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne 6. června 2008

.....
podpis autora

ABSTRAKT:

Práce se zabývá teoretickým a praktickým řešením akustické lokalizace metodou TDOA. Rozvádí návrh schémat pro vysílač a přijímač. Dále uvádí způsob komunikace mezi procesorem, který vyhodnocuje přijatý signál, a počítačem, jež výsledek zobrazuje ve srozumitelné podobě. A nakonec se zabývá návrhem plošných spojů.

The project deals with theoretical and practical resolution of acoustic localization using TDOA method. It specifies designing circuitry for receiver and transmitter. Furthermore it introduces the way of communication between microprocessor that analyzes received signal and computer that displays the result in understandable shape. And finally it is concerned with designing printed wirings.

Klíčová slova: TDOA
Goldovy kódy
PN sekvence
Pseudo náhodná sekvence

Key words: TDOA
Gold codes
PN sequence
Pseudo noise sequence

BERNKOPF, J. *Lokalizace akustických zdrojů metodou TDOA*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 38 s.
Vedoucí bakalářské práce Ing. Jiří Šebesta, Ph.D..

1 Úvod.....	2
1.1 Pasivní radar.....	2
1.2 Směrové systémy.....	2
1.3 Časoměrné systémy.....	2
1.4 Cíle práce.....	2
2 TDOA	3
2.1 Výpočet polohy.....	3
2.2 Přesnost určení polohy	6
2.3 Goldovy kódy.....	6
3.1 Vysílač.....	11
3.1.2 Generátor Goldova kódu.....	11
3.1.2 Reproduktor.....	12
3.2 Přijímač	13
3.2.1 Akustický měnič.....	16
3.2.2 Aplikační rozhraní pro ATmega2560-16AU	19
3.2.3 Sériový USB modul UMS2	22
3.3 Použití a rozšíření systému.....	23
4 Desky plošných spojů.....	24
4.1 Vysílač.....	24
4.2 Přijímač.....	25
4.3 Aplikační rozhraní pro ATmega2560-16AU.....	28
4.4 Sériový Modul UMS2.....	29
5 Závěr.....	30
6 Seznam použité literatury.....	31
Příloha č. 1: Zdrojový kód pro ATtiny26.....	32
Příloha č. 2: Zdrojový kód pro ATmega2560.....	34

1 Úvod

1.1 Pasivní radar

Pasivní radary nevysílají žádný signál, pouze sledují signály vyzařované sledovaným objektem nebo signály od objektu odražené (elektromagnetické, infračervené, akustická, atd.). Vývoj pasivních radarů začal již začátkem padesátých let dvacátého století. K rozsáhlejšímu využití došlo až v letech sedmdesátých. V civilní sféře se pasivních radarů začalo využívat až v devadesátých letech minulého století.

Pasivní radary je možné rozdělit podle zdroje použitého signálu a podle způsobu určování polohy. Z hlediska metody určení polohy lze tyto systémy rozlišit na ty které měří směr příchodu signálu, směrové systémy a ty, které měří čas příchodu signálu, časoměrné systémy.

1.2 Směrové systémy

DOA (direct of arrival), AOA (angle of arrival) používají k lokalizaci cíle sklon dopadající vlnoplochy signálu. Tímto zjišťují směr příchodu signálu (minimálně dvě stanoviště) a pomocí triangulace určí místo, kde se nachází zdroj signálu.

1.3 Časoměrné systémy

Určují polohu z okamžiku příchodu signálu vysílaného cílem na minimálně tři stanice systému. Čas vyslání přijatého signálu radarový systém nezná, pro určení polohy je nezbytné ho určit nebo eliminovat. Při určování polohy využívají měření času příchodu impulsu přijímaného signálu na tři stanoviště. Z časových rozdílů a rychlosti šíření signálu (konstantní, rychlost světla nebo rychlost zvuku) se určí dráhový rozdíl mezi vzdálenostmi, které musel signál urazit k jednotlivým stanovištím. Pomocí známých poloh a vzdáleností mezi místy příjmu se určí poloha zdroje. Pro měření polohy ve 3D je potřeba minimálně čtyř stanovišť. Na principu měření časových rozdílů příchodu signálu na jednotlivá stanoviště (TDOA) pracuje například systém Tamara nebo Věra. Anglicky se tento princip označuje jako TDOA (time difference of arrival).

Aktivní a pasivní princip lokalizace, stejně jako směrový a časoměrný, jsou často zkombinovány v jediném zařízení.

1.4 Cíle práce

Cílem této bakalářské práce je prostudovat princip metody TDOA pro lokalizaci aktivních předmětů v prostoru. Navrhnout kompletní systém pro lokalizaci touto metodou, což znamená navrhnout schéma zapojení převodníků pro akustické měniče. Navrhnout, osadit a otestovat plošné spoje převodníků. Sestavit řídicí programy mikrokontrolerů a odladit jejich funkčnost. Nakonec je nutné sestavit aplikační rozhraní pro vyhodnocení akustických signálů pomocí počítače.

2 TDOA

Metoda TDOA (také multilaterace, časoměrně hyperbolický princip) je založena na určení přesného časového rozdílu příjmu signálu vyslaného od objektu ke třem a více přijímačům.

Metoda TDOA se dnes běžně využívá ve vojenských i civilních aplikacích ke stanovení přesné polohy vozidla, letadla nebo nehybného objektu.

Hyperbolické vymezení polohy je rozděleno do dvou stupňů. První stupeň zahrnuje určení TDOA signálu zdroje mezi párem přijímačů pomocí techniky určení časového rozdílu. Ve druhém stupni jsou změřené TDOA signály transformovány do řady rozdílných měření mezi základní stanice vedoucí v sadu nelineárních, hyperbolických rovnic. Následně je použito efektivních algoritmů pro jednoznačné stanovení těchto nelineárních rovnic. Výsledek řešení těchto rovnic potom určí přesnou polohu zdroje.

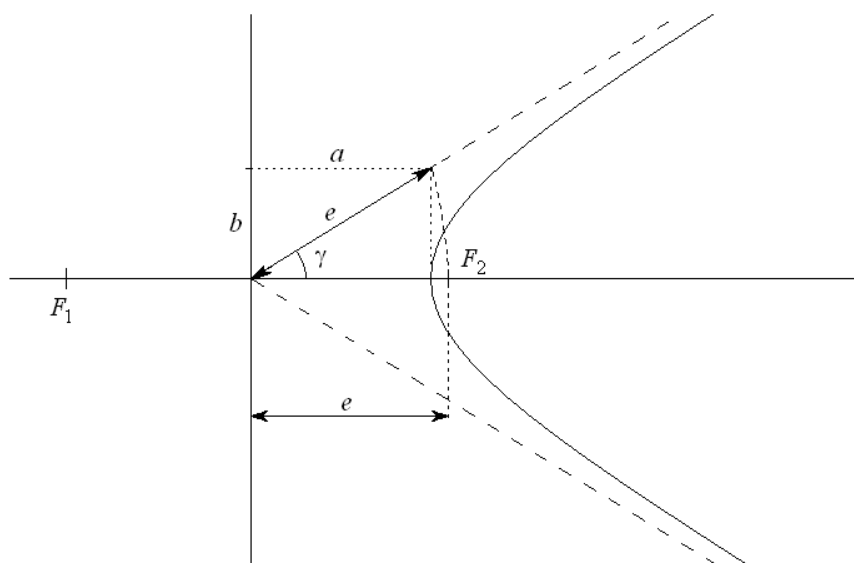
2.1 Výpočet polohy

Časoměrný systém získá po vyslání signálu cílem časy příchodu tohoto signálu na jednotlivé stanice systému. Poloha jednotlivých stanic systému je známá a pro určení polohy cíle je nutné kromě trojice jeho prostorových souřadnic určit (nebo eliminovat) čas vysílání signálu cílem. Nejčastěji je čas vysílání cíle eliminován použitím rozdílů naměřených časů příchodu.

Potřebný počet stanic lze snadno určit z počtu měřených veličin. V prostoru je nutné určit tři prostorové souřadnice a čas vysílání signálů, k jejich určení musí být v systému alespoň čtyři stanice. Pokud budeme určovat polohu cíle v rovině, postačí trojice stanic. Počet stanic je možné snížit i získáním některé ze souřadnic jiným způsobem. Je-li například známá výška cíle, klesá počet určovaných proměnných o jednu.

Geometricky popisuje uvedenou situaci křivka s konstantním rozdílem vzdáleností $\Delta\rho_{12} = c \cdot \Delta\tau_{12}$ od dvou pevných bodů (stanice systému), hyperbola. Ohniska této hyperboly leží v místech příjmu signálu. Uvedenou hyperbolu popisuje rovnice (2.1). Vztah dalších parametrů hyperboly s konfigurací měřicího systému popisuje tabulka 2.1 a obrázek 2.1.

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \quad (2.1)$$



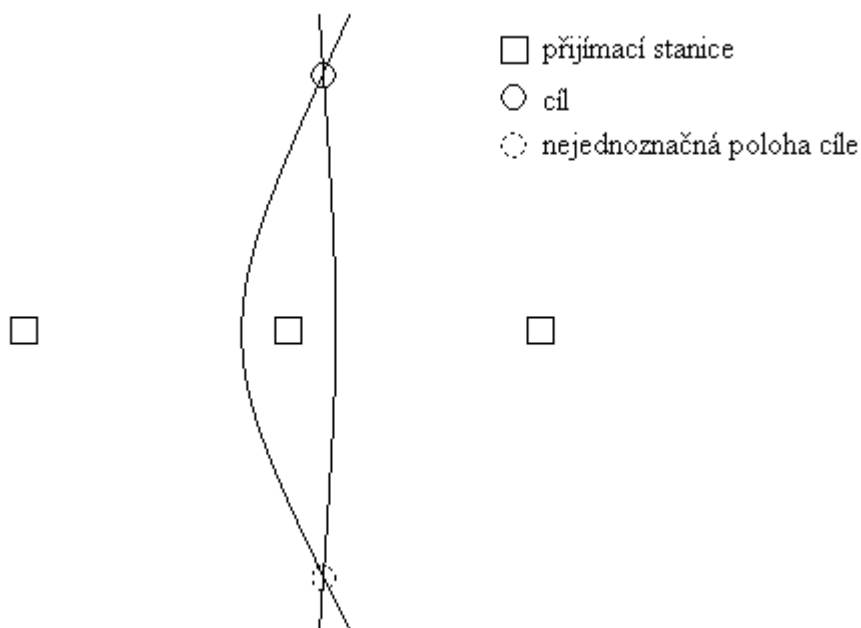
Obr. 2.1: Hyperbola – popis parametrů

Tabulka 2.1: Parametry hyperboly

	Hyperbola	Radar
a	Reálná poloosa Hyperboly	$2a = c \cdot \Delta\tau_{12} = \Delta\rho_{12}$
b	Imaginární poloosa Hyperboly $b = \sqrt{e^2 - a^2}$	
e	Délka výstřednosti hyperboly	polovina vzdálenosti stanic
γ	Úhel asymptoty $\text{tg}(\gamma) = b/a$	
F_1, F_2	Ohniska hyperboly	stanoviště přijímačů

Z každého naměřeného rozdílu časů získáme jednu hyperbolu, o které víme, že na ní leží cíl. Protože určujeme polohu cíle jako bod, potřebujeme v rovinném případě průsečík dvou hyperbol a v prostorovém pak průsečík tří rotačních hyperboloidů, jejichž osa rotace prochází ohnisky (místa s anténami přijímačů). Pro získání dvou, resp. tří rozdílů časů příchodu signálu je nutné získat tři, resp. čtyři časy příchodu.

Dostatečný počet stanic ještě nemusí zajišťovat jednoznačnost určení polohy. Pokud jsou stanice pro lokalizaci cíle v rovině rozmístěny na přímce nebo pro prostorovou lokalizaci na rovině, mají hyperboly, resp. hyperboloidy dva průsečíky. U 2D systémů se jednoznačnosti obvykle dosahuje použitím směrových antén, které omezí příjem signálu na jednu z polorovin vymezených linií instalace. Systém poté není schopen přijímat signál „za anténami“ a určení polohy je jednoznačné. U 3D systémů není nejednoznačnost určení polohy problémem, průsečíky vymezují jeden bod nad zemí (rovinou instalace systému) a druhý pod zemí. Protože jsou signály přijímány od cílů nad povrchem Země, je možné průsečík pod povrchem ignorovat.

**Obr. 2.2:** Nejednoznačnost určení polohy cíle TDOA

V pasivních TDOA radarových systémech je na vstupu algoritmu výpočtu polohy n -tice známých vektorů s polohou stanic a vektor časů příchodů signálu na jednotlivé stanice. Rovnice měření jsou zapsány analogicky s rovnicemi měření časoměrných družicových navigačních systémů, pro tzv. pseudovzdálenosti. Časové údaje (absolutní čas příchodu signálu na i -tou stanicí t_i) jsou převedeny na vzdálenost násobením rychlostí šíření:

$$\rho_i = c \cdot t_i = \|\mathbf{x}_i - \mathbf{x}_t\| + c \cdot t_t \quad (2.2)$$

kde ρ_i je vzdálenost i -té stanice, t_i je čas příjmu signálu na i -té stanici, t_t je čas vyslání signálu cílem, $\mathbf{x}_i = [x_i \ y_i \ z_i]^T$ je vektor souřadnic i -té stanice, $\mathbf{x}_t = [x_t \ y_t \ z_t]^T$ je vektor souřadnic cíle a c je rychlost šíření. Uvedená rovnice platí pro všechna N (všechny stanice).

Model měření (2.2) je nelineární a hledaná pozice cíle \mathbf{x}_t (a čas vyslání signálu t_t) se obvykle získává iteračním řešením linearizace tohoto modelu (2.3) ve zvoleném bodě počátečního odhadu $\hat{\mathbf{x}}_t$ a \hat{t}_t .

$$\Delta \boldsymbol{\rho} = \boldsymbol{\rho} - \hat{\boldsymbol{\rho}} = \underbrace{\begin{bmatrix} \frac{\partial \rho_1}{\partial \hat{x}_t} & \frac{\partial \rho_1}{\partial \hat{y}_t} & \frac{\partial \rho_1}{\partial \hat{z}_t} & \frac{\partial \rho_1}{\partial c \cdot \hat{t}_t} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \rho_N}{\partial \hat{x}_t} & \frac{\partial \rho_N}{\partial \hat{y}_t} & \frac{\partial \rho_N}{\partial \hat{z}_t} & \frac{\partial \rho_N}{\partial c \cdot \hat{t}_t} \end{bmatrix}}_{\mathbf{H}} \cdot \begin{bmatrix} \Delta \hat{\mathbf{x}}_t \\ c \cdot (t_t - \hat{t}_t) \end{bmatrix} \quad (2.3)$$

kde $\Delta \hat{\mathbf{x}}_t = \mathbf{x}_t - \hat{\mathbf{x}}_t$, $\boldsymbol{\rho}$ je vektor naměřených pseudovzdáleností a $\hat{\boldsymbol{\rho}}$ odpovídá zvoleným hodnotám vektoru $\hat{\mathbf{x}}_t$. Převodní maticovou konstantu \mathbf{H} lze v bodě $\hat{\mathbf{x}}_t$ určit podle vztahu (2.4)

$$\mathbf{H} = \begin{bmatrix} \frac{x_1 - \hat{x}_t}{\hat{r}_1} & \frac{y_1 - \hat{y}_t}{\hat{r}_1} & \frac{z_1 - \hat{z}_t}{\hat{r}_1} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_N - \hat{x}_t}{\hat{r}_N} & \frac{y_N - \hat{y}_t}{\hat{r}_N} & \frac{z_N - \hat{z}_t}{\hat{r}_N} & 1 \end{bmatrix} \quad (2.4)$$

$$\text{kde} \quad \hat{r}_i = \|\mathbf{x}_i - \hat{\mathbf{x}}_t\| = \sqrt{(x_i - \hat{x}_t)^2 + (y_i - \hat{y}_t)^2 + (z_i - \hat{z}_t)^2} \quad (2.5)$$

Postup určení polohy cíle lze rozdělit do následujících kroků:

- Ověření přijatých signálů – všechny pseudovzdálenosti (časy příchodu) použité k výpočtu musí patřit k jedinému vyslanému signálu
- Přepočítání naměřených časů příchodu na pseudovzdálenosti
- Volba počátečního odhadu $\hat{\mathbf{x}}_t$ a výpočet odpovídajících pseudovzdáleností $\hat{\boldsymbol{\rho}}$
- Výpočet korekce odhadu polohy $\Delta \mathbf{x}_t$ z rozdílu naměřených pseudovzdáleností a pseudovzdáleností $\hat{\boldsymbol{\rho}}$

Postup určení korekce odhadu polohy $\Delta \mathbf{x}_t$ závisí na počtu stanic systému. Pokud není systém předurčený (matice \mathbf{H} je čtvercová) a \mathbf{H} je regulární (pro obvyklé konfigurace systému splněno), lze získat řešení násobením rovnice (2.3) zleva maticí \mathbf{H}^{-1} (2.6). Pokud je systém předurčený, pak lze korekci odhadu polohy $\Delta \mathbf{x}_t$ získat metodou nejmenších čtverců (2.7)

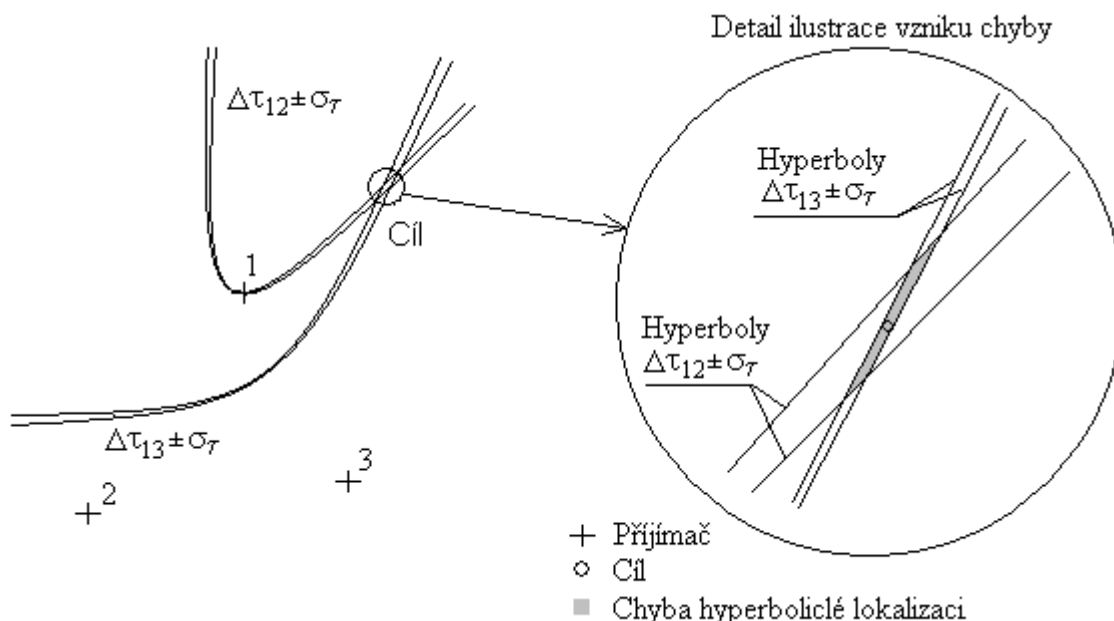
$$\Delta \mathbf{x}_t = \mathbf{H}^{-1} \Delta \boldsymbol{\rho} \quad (2.6)$$

$$\Delta \mathbf{x}_t = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \Delta \boldsymbol{\rho} \quad (2.7)$$

(Bezoušek, Šedivý, 2004: 3-5).

2.2 Přesnost určení polohy

Protože měření rozdílů časů je jako každé měření zatíženo chybou, je chybou zatížena i určená poloha cíle. Vznik této chyby lze snadno ilustrovat rozštěpením hyperboly na oblast vymezenou dvojicí hyperbol (oblast mezi hyperbolami odpovídá rozdílu pseudovzdáleností $\Delta\rho_i \pm$ chyba; hodnota odpovídá zvolené pravděpodobnosti údaje – například 99 %). Průsečíkem dvou takových oblastí již není bod, ale jistá oblast. Velikost této oblasti závisí na velikosti chyby měřených rozdílů časů a na umístění stanic. Závislost chyby určení polohy na chybě měření časů a poloze stanic ilustruje obrázek 2.3



Obr. 2.3: Ilustrace chyby při hyperbolické lokaci

Chybu měření lze potlačit i použitím více stanic. Větší množství vstupních hodnot s nezávislými chybami umožní jejich zprůměrování a zvýšení celkové přesnosti. Přesnost určení polohy metodou TDOA je závislá zejména na:

- Geometrickém uspořádání přijímacích stanic
- Přesnosti synchronizace přijímacích stanic
- Délce vysílaného impulzu
- Nejistotě poloze přijímačů

(Bezoušek, Šedivý, 2004: strany 3-6).

2.3 Goldovy kódy

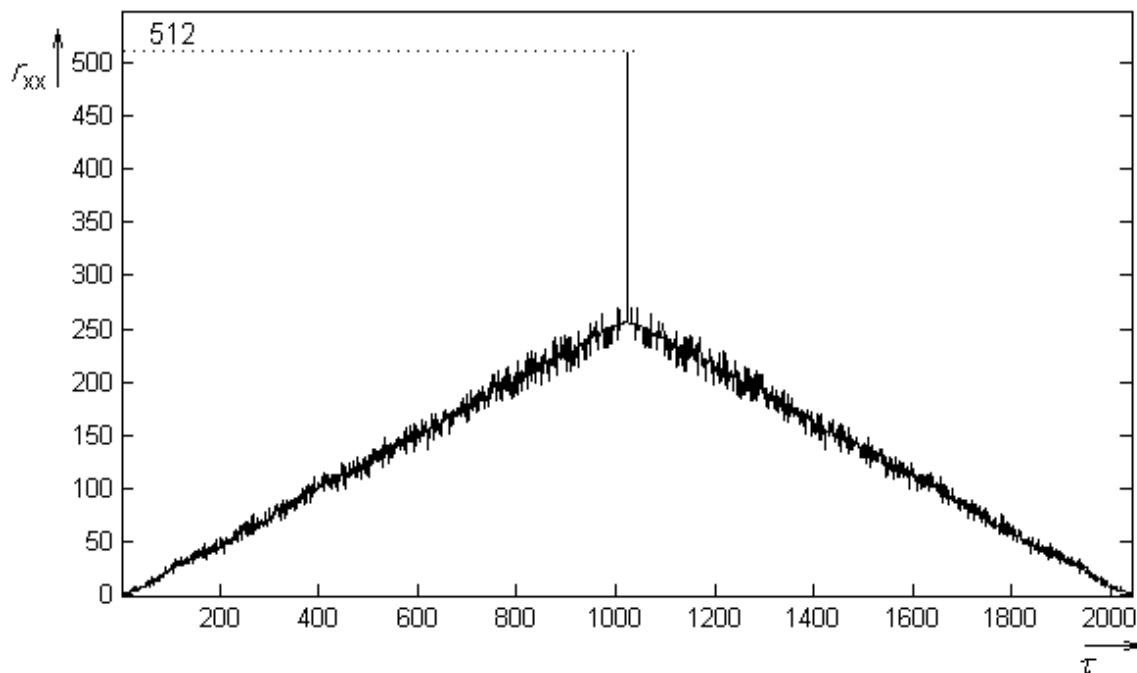
Abychom mohli měřit rozdíly příchodu časů na jednotlivá stanoviště, potřebujeme stanovit přesný začátek signálu. K tomu použijeme Goldovy kódy, které disponují dobrými korelačními vlastnostmi.

Goldovy kódy patří mezi PN sekvence (Pseudo Noise, Pseudo Náhodná sekvence). Goldovy kódy se používají v systémech s rozprostřeným spektrem. Tyto systémy jsou schopny jednotlivé účastníky rozlišit pomocí různých PN kódů. Když je přenosový kanál sdílen technikou rozprostřeného spektra, všichni uživatelé mohou vysílat současně na stejných frekvencích i ve stejný čas. Stejná PN sekvence slouží jak pro rozprostření signálu ve vysílači, tak k derozprostření na přijímači (inverznímu rozprostření). Každému uživateli je

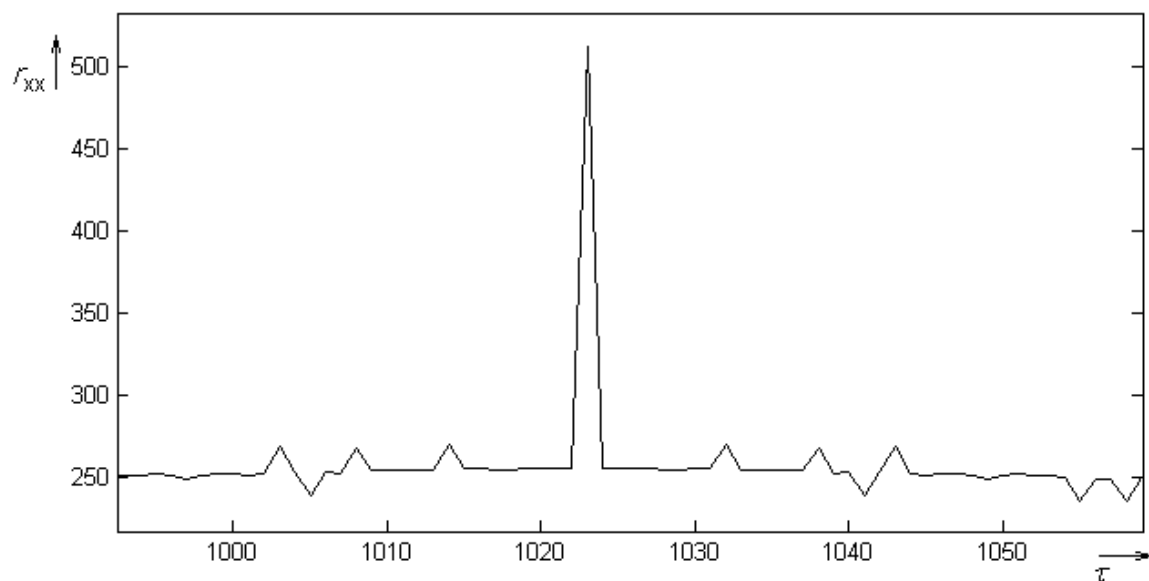
přidělen odlišný PN kód a tím se odliší od ostatních. Cílem systému s rozprostřeným spektrem (pro systém mnohonásobného přístupu) je najít sadu rozprostřených kódů nebo signálů tak, aby co nejvíce uživatelů mohlo používat pásmo kmitočtu s co nejmenší vzájemnou interferencí. Ideální PN sekvence je ortogonální, což znamená, že má nulovou vzájemnou korelaci. Další vlastností je impulzní autokorelační funkce. Díky ortogonalitě pseudo-náhodné sekvence (PN sekvence) jsme schopni odlišit jednotlivé uživatele a díky impulzní autokorelační funkci přesně určit začátek vysílaného signálu. Autokorelační funkce je dána rovnicí (2.8).

$$r_{xx}(\tau) = \sum_{i=0}^M x(i)x(i-\tau) = \sum_{i=0}^M x(i+\tau)x(i) \quad (2.8)$$

kde τ je zpoždění a $x(i)$ je Goldova posloupnost



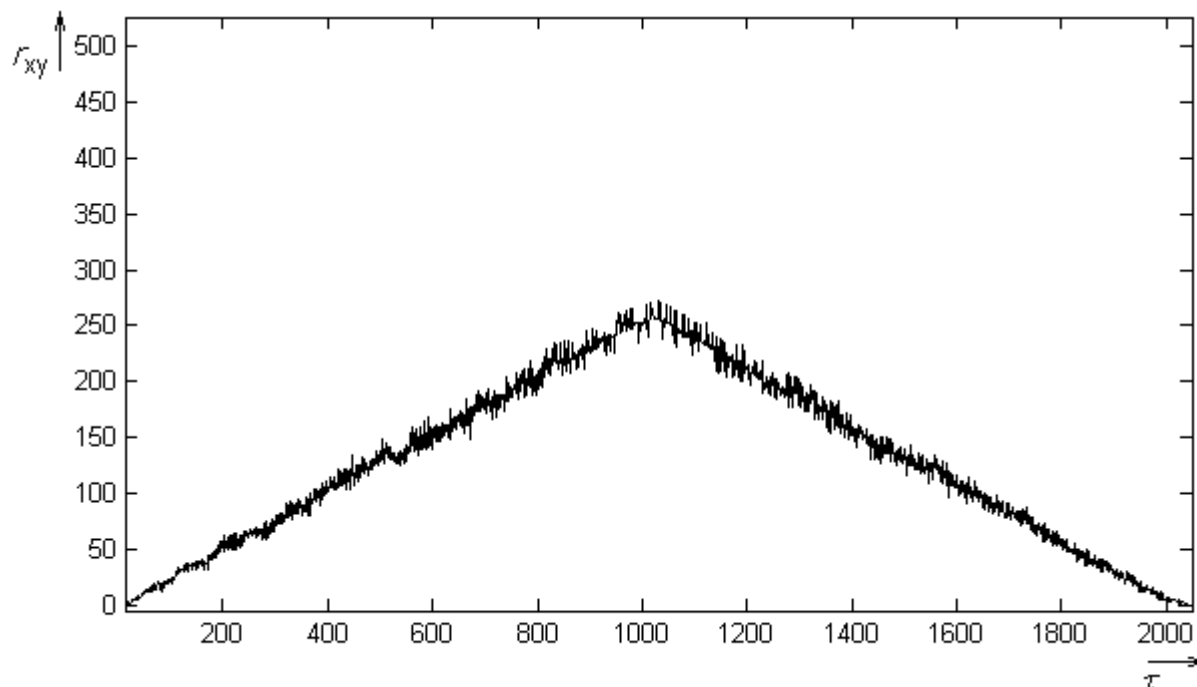
Obr. 2.4: Autokorelace Goldovy sekvence



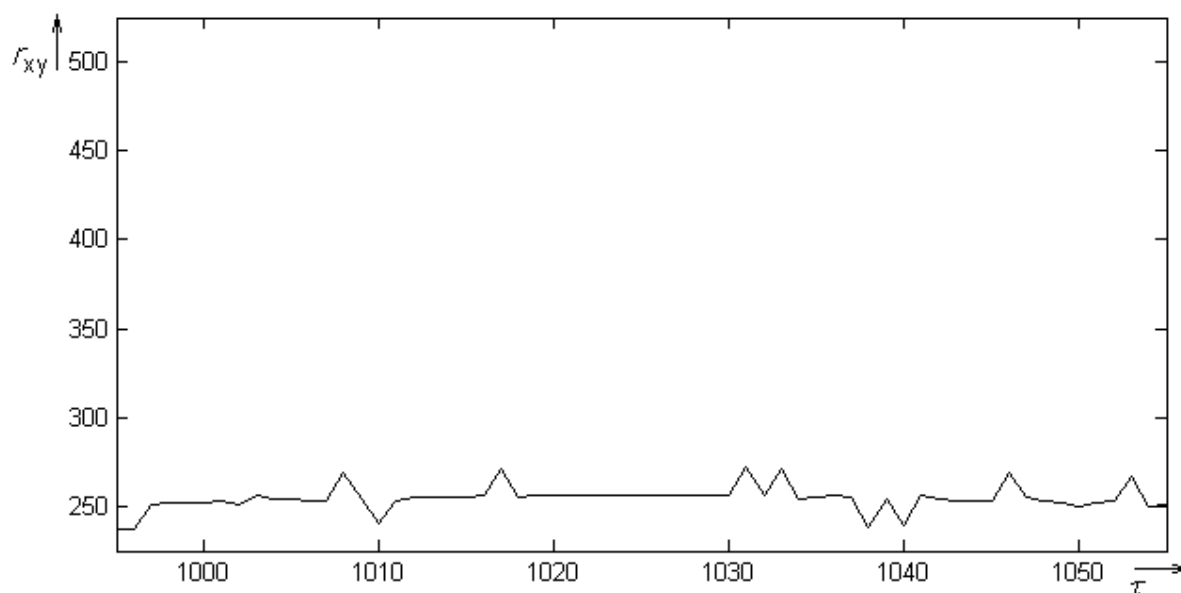
Obr. 2.5: Detail maxima autokorelace Goldovy sekvence

Funkce vzájemné korelace popisuje rovnice (2.9)

$$r_{xy} = \frac{1}{M} \sum_{i=1}^M x(i)y(i-\tau) = \frac{1}{M} \sum_{i=1}^M x(i+\tau)y(i) \quad (2.9)$$



Obr. 2.6: Vzájemná korelace Goldových sekvencí



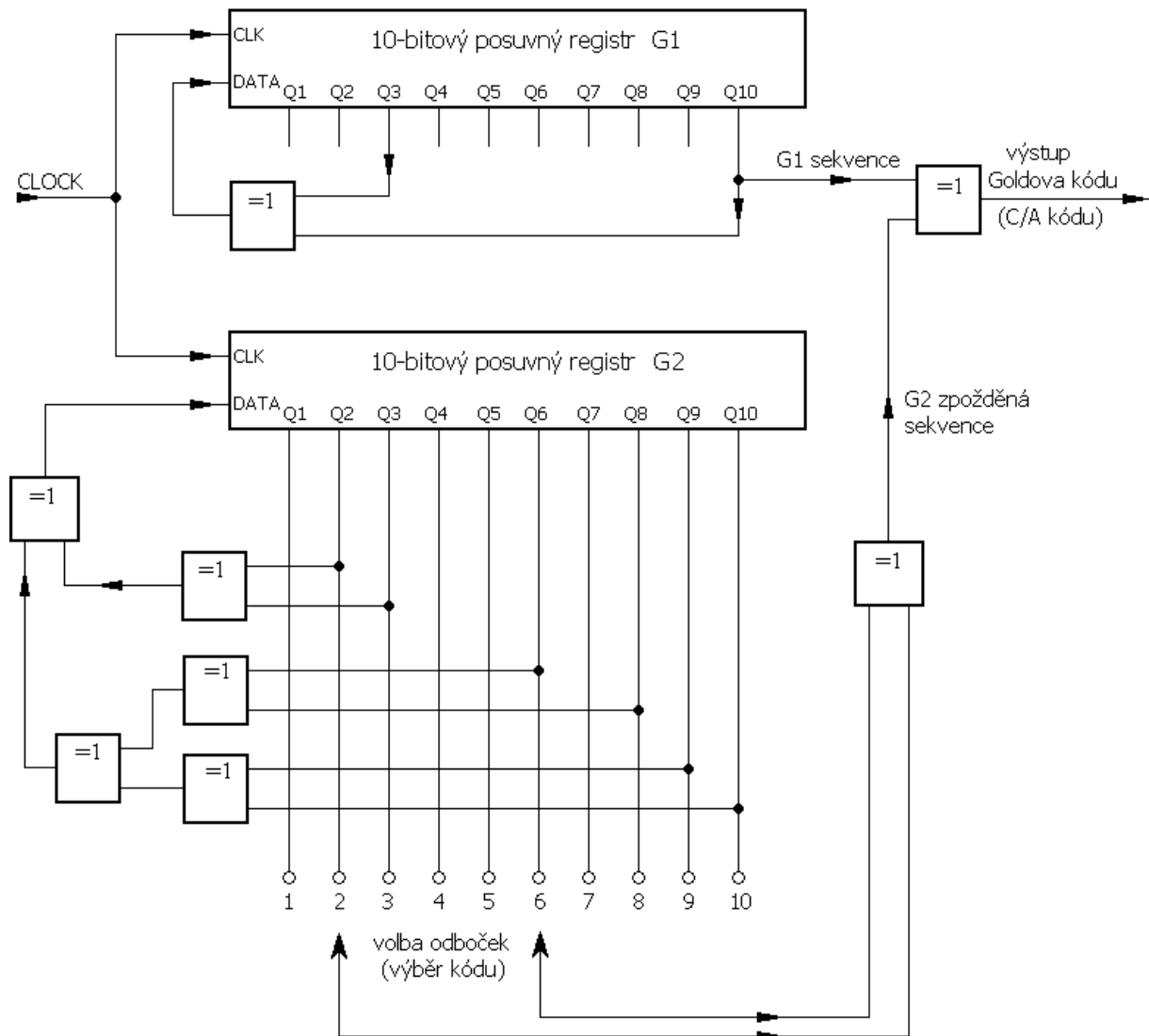
Obr. 2.7: Detail maxima vzájemné korelace Goldových sekvencí

Pro vzájemnou korelaci mezi požadovanou a nepožadovanou PN sekvencí je zaručený přeslech menší než 21,6dB. Z obrázku 2.6 a 2.7 je patrné, že při příjmu jiné PN sekvence než je očekávána, je na výstupu korelátoru pouze šum .

Dekódování (derozprostírání) spočívá nejprve v určení počátku signálu. Tedy v korelaci přijímaného signálu a PN sekvence, kterou má přijímač k dispozici. Je to stejná PN

sekvence, kterou byl rozprostřen signál na vysílací straně. Známe-li počátek signálu, můžeme provést samotné derozptření. Tato operace spočívá v násobení (v případě funkcí), nebo sčítání modulo 2 (v případě sekvencí). V ideálním případě se přijímaný signál, který byl rozprostřen jiným kódem, nedekóduje (nekoreluje) a způsobí tak minimální interferenci očekávaného signálu. Stupeň interference od jiného rozprostřeného kódu je dán vzájemnou korelací mezi dvěma rozprostřenými kódy. Relativně velká skupina Goldových kódů má přesně dané vlastnosti vzájemné korelace.

Blokové schéma generátoru C/A kódu pro GPS je vyobrazeno na obrázku 2.8 a obsahuje dva desetibitové posuvné registry, každý s oddělenou zpětnovazební sítí vytvořenou z hradel XOR. Obě zpětnovazební sítě jsou vybrány tak, aby generovali sekvenci o maximální délce 1023 bitů. Oba posuvné registry jsou při spuštění naplněny samými jedničkami.



Obr. 2.8: Generátor Goldova kódu, blokové schéma

Goldovy kódy jsou vytvářeny pomocí operace sčítání modulo-2 výstupů dvou registrů G1 a G2. Perioda Goldova kódu je 1023 bitů, $(2^{10} - 1)$. Různé kódy můžeme získat změnou vzájemné fáze registrů G1 a G2. Namísto resynchronizace posuvných registrů je pro vytvoření různých kódů jednodušší zpoždit jeden z výstupů registrů (G2). Toto zpoždění je dosaženo dalším součtem modulo-2 (exkluzivní or) mezi dvěma odbočkami registru G2. Sekvence vytvořená zpětnovazebním, xor-rovým, posuvným registrem má tu vlastnost,

že sčítání modulo-2 mezi sekvencí a její zpožděnou replikou vyprodukuje tu samou sekvenci, ale zpožděnou o rozdílný počet hodinových cyklů (clks).

Zvolením dvou odboček registru G2 je možno generovat 45 různých zpoždění, tedy i 45 různých Goldových kódů s dobrou autokorelací a vzájemnou korelací. Ze 45 možných kódů je systému GPS přiřazeno 32, jak je uvedeno v tabulce 2.2. Pro naši aplikaci jsme vybrali kód GPS PRN 1 (Global Positioning System Pseudo Random Number).

Tab. 2.2: Používané kombinace odboček registru G2 u GPS a jejich zpoždění

Číslo C/A kódu	Odbočka registru G2	zpoždění G2 sekvence
GPS PRN 1	2 & 6	5 clks
GPS PRN 2	3 & 7	6 clks
GPS PRN 3	4 & 8	7 clks
GPS PRN 4	5 & 9	8 clks
GPS PRN 5	1 & 9	17 clks
GPS PRN 6	2 & 10	18 clks
GPS PRN 7	1 & 8	139 clks
GPS PRN 8	2 & 9	140 clks
GPS PRN 9	3 & 10	141 clks
GPS PRN 10	2 & 3	251 clks
GPS PRN 11	3 & 4	252 clks
GPS PRN 12	5 & 6	254 clks
GPS PRN 13	6 & 7	255 clks
GPS PRN 14	7 & 8	256 clks
GPS PRN 15	8 & 9	257 clks
GPS PRN 16	9 & 10	258 clks
GPS PRN 17	1 & 4	469 clks
GPS PRN 18	2 & 5	470 clks
GPS PRN 19	3 & 6	471 clks
GPS PRN 20	4 & 7	472 clks
GPS PRN 21	5 & 8	473 clks
GPS PRN 22	6 & 9	474 clks
GPS PRN 23	1 & 3	509 clks
GPS PRN 24	4 & 6	512 clks
GPS PRN 25	5 & 7	513 clks
GPS PRN 26	6 & 8	514 clks
GPS PRN 27	7 & 9	515 clks
GPS PRN 28	8 & 10	516 clks
GPS PRN 29	1 & 6	859 clks
GPS PRN 30	2 & 7	860 clks
GPS PRN 31	3 & 8	861 clks
GPS PRN 32	4 & 9	862 clks

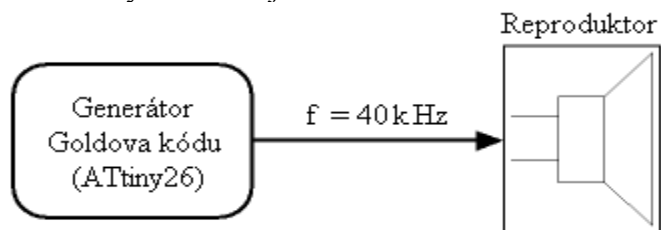
Zpoždění G2 sekvence nám udává po kolika hodinových cyklech se rozběhne generování Goldova kódu a tím se kód stane periodickým.

Goldovy kódy byly vyvinuty v roce 1967 společností Magnavox, která se specializuje výhradně na systémy mnohonásobného přístupu s rozprostřeným spektrem. Goldovy kódy jsou hojně využívány při určování polohy amerického systému GPS a ruského GLONASS.

3 Koncepce lokalizačního systému

3.1 Vysílač

Abychom mohli ověřit systém pro lokalizaci akustických zdrojů je nejdříve nutné zdroj vytvořit. Jako lokalizovaný kmitočet jsme zvolili 40 kHz.

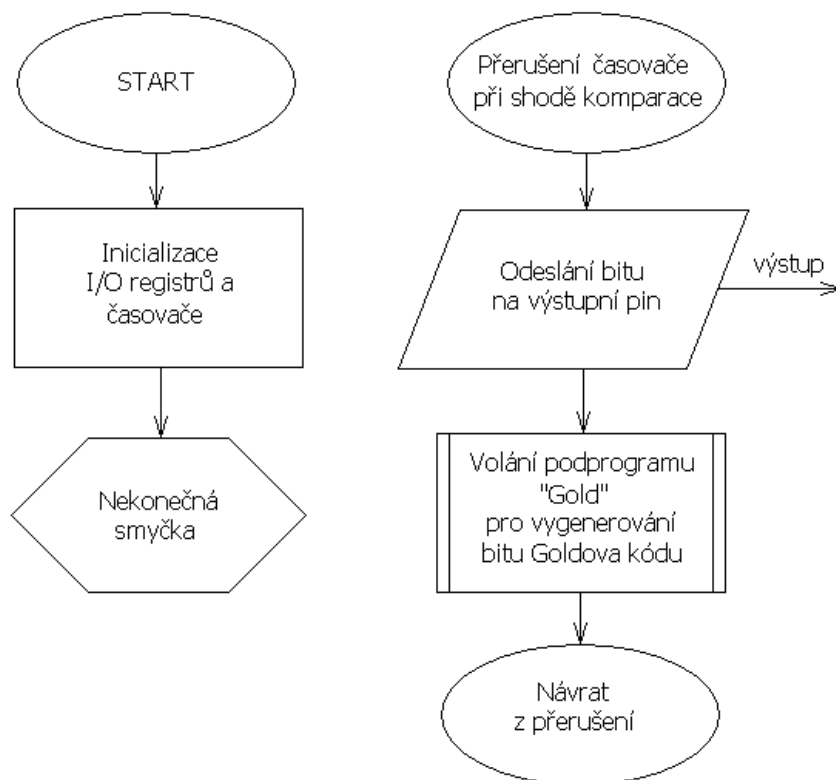


Obr. 3.1: Blokové schéma vysílače

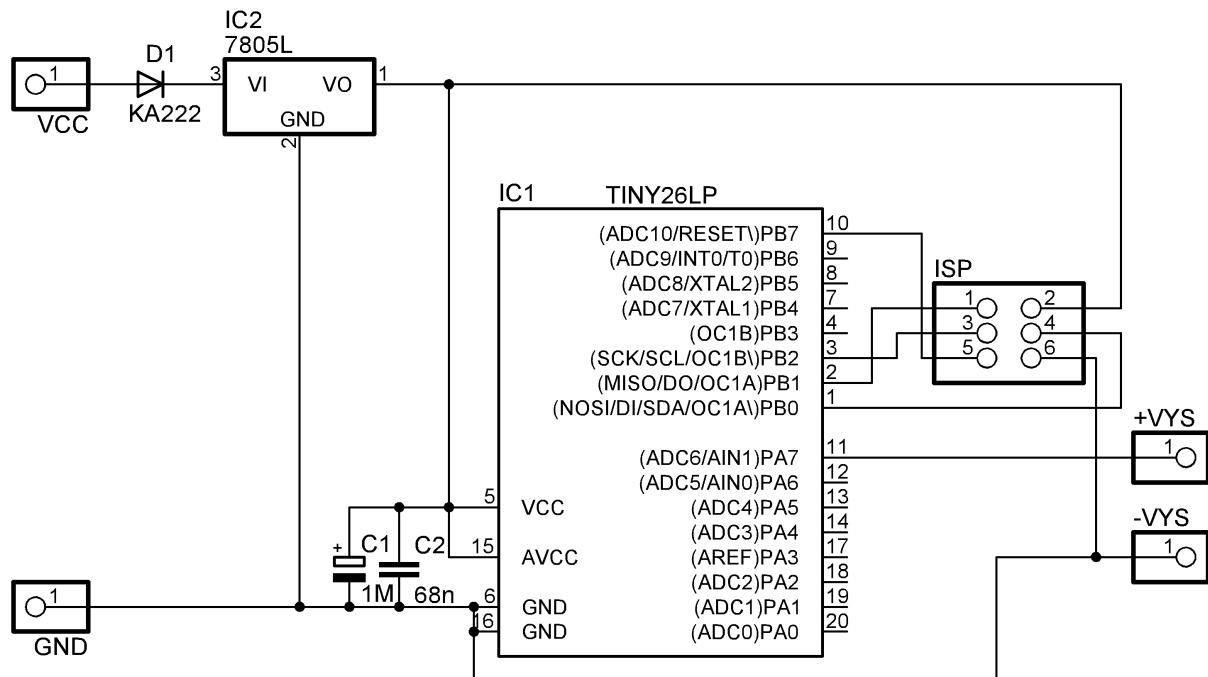
První blok představuje programovatelný mikrokontrolér ATtiny26. Do něj je nahrán pomocí rozhraní ISP program, jež generuje Goldovu posloupnost rychlostí 40 kHz. Signál přiváděný do reproduktoru je pravoúhlý a má základní frekvenci 40 kHz. Vstupní obvody vysílače (reproduktoru) oscilují na 40 kHz se šířkou pásma 2 kHz. Díky tomu vyberou první harmonickou, čímž poněkud zkruslí signál. To však není problém, protože jsme na přijímací straně schopni tento signál obnovit. Reproduktor představuje vysílač akustických vln.

3.1.2 Generátor Goldova kódu

Generátor Goldova kódu naprogramujeme do mikrokontroléru ATtiny26 použitím dvou posuvných, kolujících registrů a osmi operacemi sčítání modulo-2 (exkluzivní or). Zdrojový kód pro mikrokontroler ATtiny26 je uveden v příloze 1.



Obr. 3.2: Vývojový diagram generátoru Goldova kódu



Obr. 3.3: Blokové schéma generátor Goldova kódu

Na svorky VCC a GND je připojeno napětí 9 V. ATtiny však může pracovat pouze v rozsahu 2,7 - 5,5 V, proto je na napájecích svorkách připojen stabilizátor 7805, který nám stabilizuje výstupní napětí na 5 V.

Dioda D1 plní funkci ochrany proti přepólování napájení. Kondenzátory C1 a C2 jsou blokovací kapacity obvodu ATtiny26. Výsledná Goldova sekvence s frekvencí 40 kHz je posílána na výstupní pin PA7, který je připojen na vstup reproduktoru 400ST160. Perioda Goldova kódu je rovna:

$$T = (2^n - 1) \cdot T_b \quad (3.1)$$

kde T_b je doba jednoho bitu.

V tomto případě je perioda rovna: $T = (2^n - 1) \cdot T_b = (2^{10} - 1) \cdot T_b = 1023 \cdot T_b$, Goldův kód je tedy periodický po 1023 bitech. Při generující frekvenci 40 kHz je $T_b = 1/40$ kHz a maximální dosah systému odvoditelný z rovnice (3.2)

$$d = v \cdot T \quad (3.2)$$

kde d je dosah, v je rychlost zvuku ve vzduchu (~ 340 m.s⁻¹) a T je perioda goldova kódu. Dosah je tedy zhruba 8,5 metru.

3.1.2 Reprodukter

Jako reproduktor použijeme součástku 400ST160.

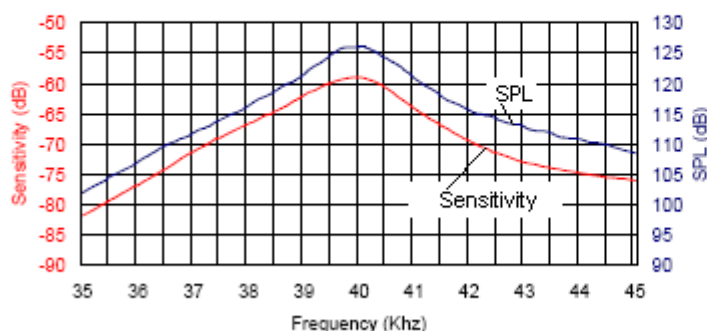
Důležité specifikace 400ST160:

- Střední frekvence $40,0 \pm 1,0$ kHz
- Šířka pásma 2,5 kHz
- Max. připojené napětí $20 V_{RMS}$
- Hladina akustického tlaku při 40 kHz a $10V_{RMS}$ 120 dB min.

Na obrázku 3.4 můžeme vidět charakteristiky reproduktoru. Tyto charakteristiky jsou platné i pro mikrofon.

Sensitivity/Sound Pressure Level

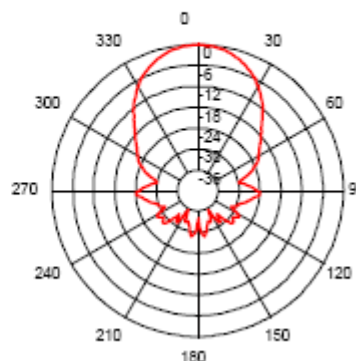
Tested under 10Vrms @30cm



a)

Beam Angle:

Tested at 40.0Khz frequency



b)

Obr. 3.4: a) Citlivost mikrofону, hladina akustického tlaku reproduktoru,
b) Vyzařovací/přijímací charakteristika
[10]

3.2 Přijímač

Blokové schéma přijímače je zobrazeno na obrázku 3.5.

Přijímače 1 až 4 představují mikrofony 400SR160. Tyto různé přijímané signály jsou přiváděny na oddělené zesilovače, potom na komparátory.

Zesilovač má za úkol zesílit signál na takovou hodnotu napětí aby byl komparátor schopen obnovit obdélníkový průběh signálu. Na vytvoření zesilovačů použijeme operační zesilovač integrovaný v obvodu LM324N, ve kterém jsou v jednom pouzdře integrovány čtyři zesilovače. Zesilovače musejí být dvoustupňové, aby malý vstupní signál (řádově milivolty, desítky milivoltů) byl zesílen na zhruba 1V.

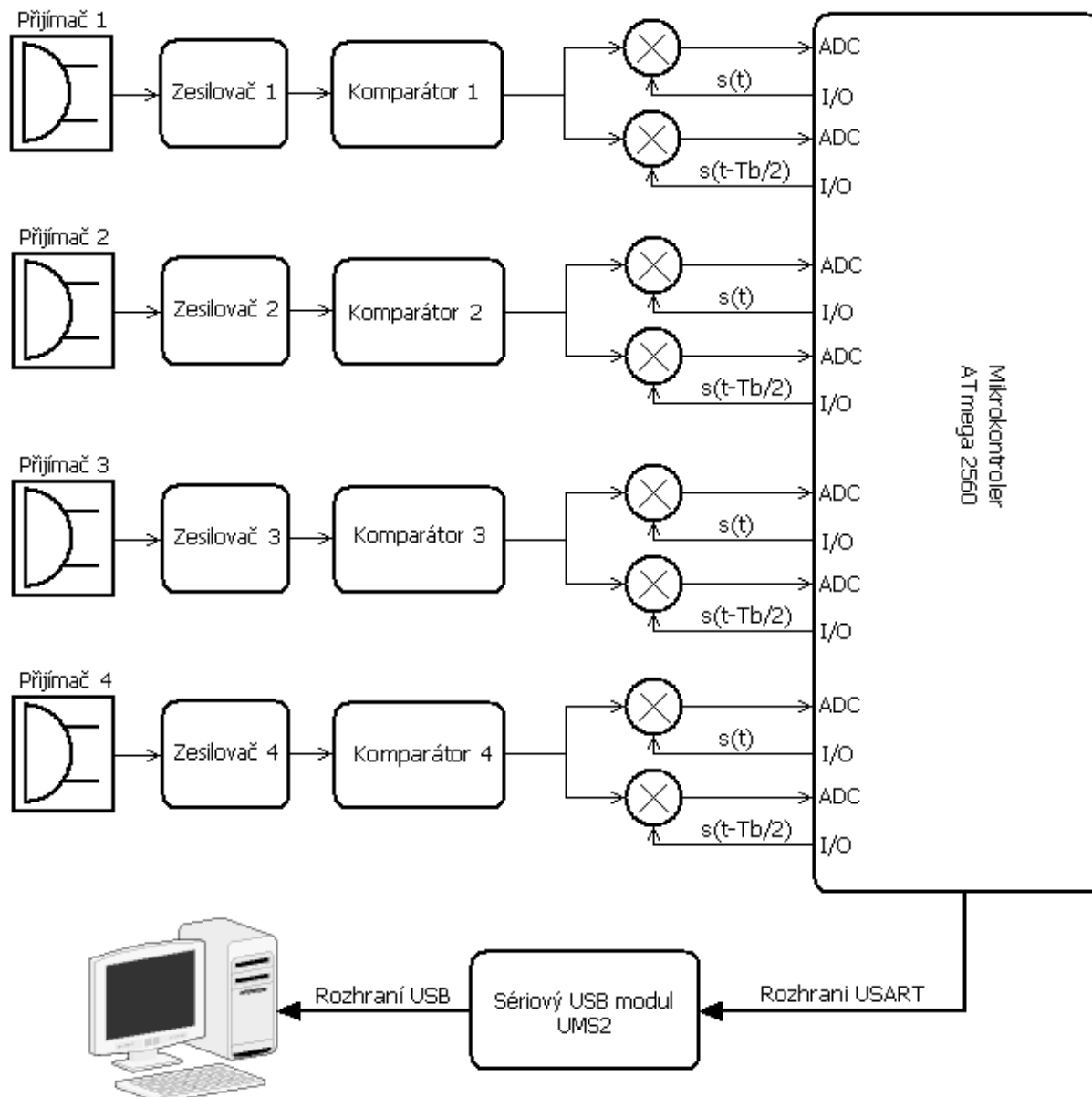
Komparátor nám slouží k obnovení původního vysílaného Goldova kódu vytvořením příslušných logických úrovní (pravoúhlého signálu) a pro jeho konstrukci použijeme obvod LM339, který má v sobě opět integrovány čtyři komparátory.

Dále signál přivádíme současně na dvě analogové násobičky realizované jako integrovaných logických členů exkluzivního OR. Na druhé vstupy XORu přivádíme signál z mikrokontroléru ATmega 2560 a to tak, že na první z dvojice XORu přivádíme Goldův kód nezpožděný a na druhou přivádíme Goldův kód zpožděný o půl bitové periody ($1/2T_b$). Za logické členy XOR jsou připojeny integrační články a tímto realizujeme obvod autokorelace. Je-li na obou dvou integračních člancích stejné napětí (přibližně stejné), je přijímaný Goldův kód v synchronismu s vysílaným Goldovým kódem. Pokud tedy máme přijímací větev, myšleno větev pro jeden mikrofon, zasynchronizovanou se signálem z mikrokontroléru, můžeme u ostatních větví zjistit rozdíly příchodu času. Jednotlivé zasynchronizování provádíme postupným posouváním (zpoždováním) signálu z mikrokontroléru o jeden bit. Nejistota určení polohy je potom dána dráhou, kterou urazí akustický signál za bitovou periodu, rovnicí (3.3).

$$n = T_b \cdot v \quad (3.3)$$

kde n je nejistota v metrech a v je rychlost zvuku ve vzduchu ($\sim 340 \text{ m}\cdot\text{s}^{-1}$). Nejistota vychází zhruba 8,5 mm.

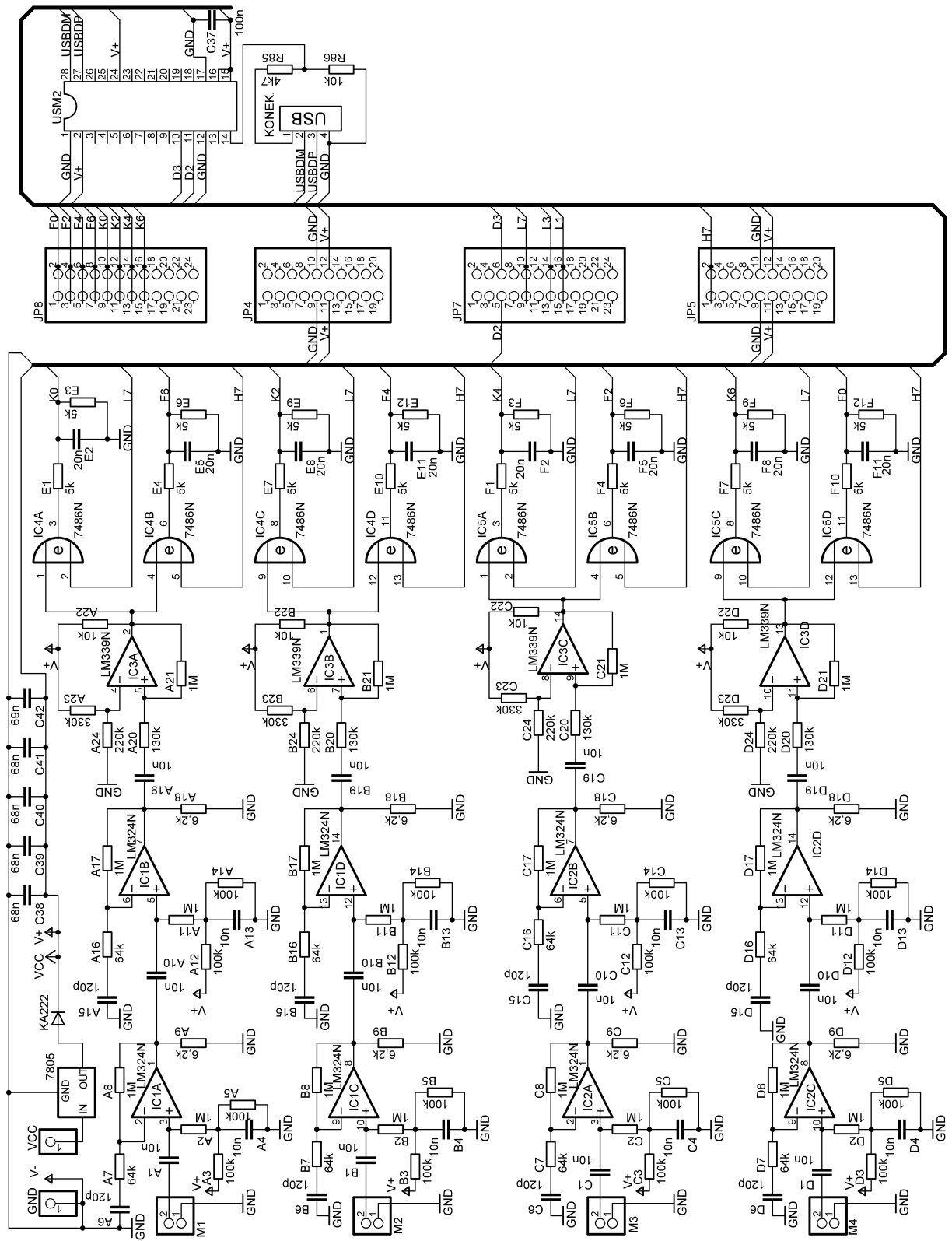
Informaci o časovém rozdílu příjmu signálu předává ATmega (signálu TDOA) USB modulu UMS2 prostřednictvím sběrnice UART.



Obr. 3.5: Blokové schéma přijímače

Sériový modul USB UMS2 je převodník mezi sběrnici USB a UART s možností plně hardwarového řízení toku dat. Modul UMS2 umožňuje přenos dat rychlostí až 1 MBd po sběrnici RS232 a až 3 MBd po sběrnici RS485.

UMS2 je přes USB kabel připojen k počítači, který má za úkol závěrečné vyhodnocení například pomocí programovacího prostředí C++.



Obr. 3.6: Celkové schéma zapojení přijímače

3.2.1 Akustický měnič

Mikrofon:

Jako mikrofon použijeme součástku 400SR160.

Důležité specifikace 400SR160:

- Střední frekvence $40,0 \pm 1,0$ kHz
- Šířka pásma $2,0$ kHz
- Citlivost
při 40 kHz a 1 V/ μ bar -65 dB min.

Mikrofon je přes vazební kondenzátor připojen na zesilovač.

Zesilovač:

Je realizován jako dvoustupňový, neinvertující operační zesilovač. Značení kondenzátorů a odporů jsem volil jako A1, A2...; B1, B2... a pod., abych si usnadnil návrh plošného spoje a věděl, ke které větvi jaká součástka patří. Zesílení potřebujeme pomocí zpětné vazby nastavit tak, aby bylo konstantní do kmitočtu $f = 40$ kHz. Toto zesílení jsme mohli buď odečíst z grafu v datasheetu [12], což je nepřesné, nebo spočítat s průsečíků v grafu datasheetu:

$$\text{při } f_T = 10 \text{ kHz je } A = 40 \text{ dB} = 1$$

$$\text{při } f = 100 \text{ kHz je } A = 20 \text{ dB} = 10$$

přímka zesílení je sice lineární, ale měřítko osy frekvence je logaritmické, proto by bylo nepřesné odečítat zesílení pro 40 kHz. Pro názorné odvození si představíme že 10 kHz je „1“ a 100 kHz je „10“ potom 40 kHz leží v lineárním měřítku mezi nimi:

$$\log(1) = 0$$

$$\log(4) = 0,6021$$

$$\log(10) = 1$$

nyní již stačí aplikovat trojčlenku na známé zesílení:

$$1 = 40 \text{ dB}$$

$$0,6021 = A_U[\text{dB}]$$

$$A_U[\text{dB}] = \frac{0,6021}{1} \cdot 40 = 24,08 \text{ dB}$$

$$A_U[\text{dB}] = 24,08 \text{ dB} \Rightarrow A_U = 16$$

Známe-li zesílení, zbývá určit odpory, které dané zesílení nastaví. Je-li zesílení neinvertujícího operačního zesilovače dáno vztahem:

$$A = 1 + \frac{A8}{A7} \quad (3.4)$$

Odpor A8 volíme klasicky $A8 = 1 \text{ M}\Omega$, potom odpor A7 vypočítáme jako:

$$A7 = \frac{A8}{A - 1} = \frac{1 \cdot 10^6}{16 - 1} = 66,6 \text{ k}\Omega$$

Odpor A7 volíme z řady E24, a to nejbližší vyšší hodnotu z důvodu zajištění konstantního zesílení do 40 kHz, $A7 = 68 \text{ k}\Omega$. Zesílení je o něco menší než vypočítané:

$$A = 1 + \frac{1 \cdot 10^6}{68 \cdot 10^3} = 15,7$$

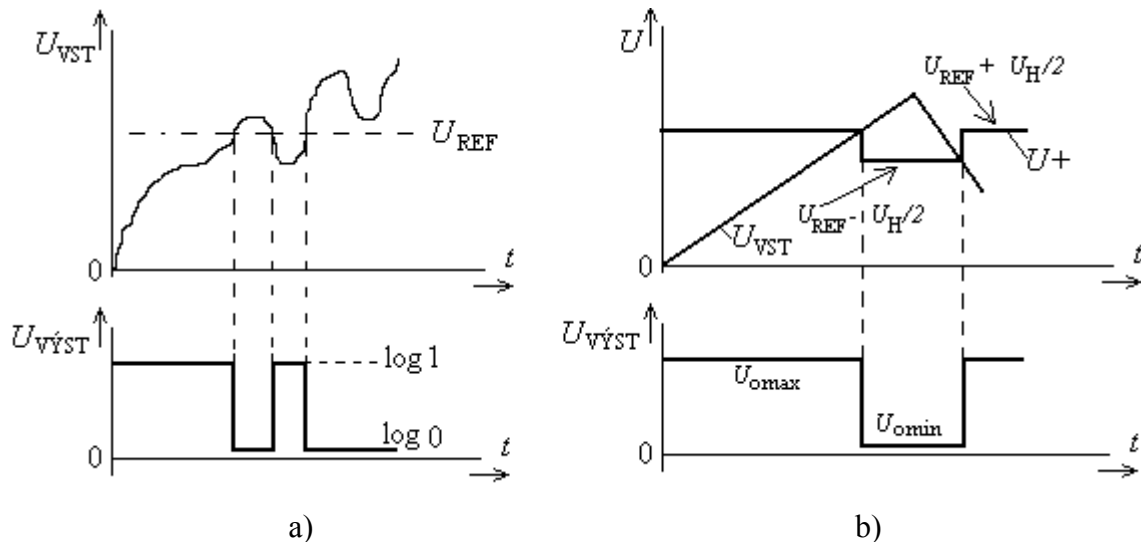
$$A_{[\text{dB}]} = 20 \log(A) = 23,9 \text{ dB}$$

Společné zesílení obou stupňů je zesílení jednoho stupně krát zesílení druhého stupně:

$$A_{\text{celkové}}[\text{dB}] = 23,9^2 = 571 \text{ dB}$$

Komparátor:

Jako komparátor jsme použili obvod LM339N. Pokud je komparátor zapojený bez hystereze, je jeho výstup změněn při každém průchodu signálu úrovní U_{REF} . Toto nebývá často žádoucí, zvláště je-li na užitečném signálu namodulovaná rušivá složka, ať už síťové napětí nebo jiné rušení, dochází potom k několikanásobnému překlopení (Obr. 3.7 a)).



Obr. 3.7 Průběhy na komparátoru a) bez hystereze b) s hysterezí [14]

Naše zapojení komparátoru je navrženo na referenční napětí $U_{REF} = 2\text{ V}$ s hysterezí $U_H = 0,5\text{ V}$ nastavené ve větvi „A“ Odporů A20, A21, A23 a A24.

Odporů A23, A24 nastavují referenční napětí a přivádí ho na invertující vstup komparátoru. Referenční napětí U_{REF} je dáno vztahem:

$$U_{REF} = U_{CC} \frac{A24}{A23 + A24} \quad (3.5)$$

Napětí U_H je dáno vztahem:

$$U_H = (U_{0max} - U_{0min}) \cdot \frac{A20}{A20 + A21} \quad (3.6)$$

Doporučení výrobce je:

$$A20 \approx A24 \parallel A23 = \frac{A24 \cdot A23}{A24 + A23} \quad (3.7)$$

Odpor A21 zvolíme klasicky $1\text{ M}\Omega$. Pokud chceme, aby referenční napětí bylo $U_{REF} = 2\text{ V}$ a hystereze $U_H = 0,5\text{ V}$, musíme řešit soustavu dvou rovnic o dvou neznámých. U_{0max} a U_{0min} známe z datasheetu, $U_{0max} = 4,7\text{ V}$ a $U_{0min} = 0,3\text{ V}$.

Po úpravě rovnice (3.6) můžeme spočítat odpor A20

$$A20 = \frac{U_H \cdot A21}{U_{0max} - U_{0min} - U_H} = \frac{0,5 \cdot 1 \cdot 10^6}{4,7 - 0,3 - 0,5} = 128\text{ k}\Omega$$

Z rovnice (3.5) vyjádříme odpor A23 a dosadíme známé veličiny. Vyjde nám:

$$A23 = \frac{U_{CC} \cdot A24 - U_{REF} \cdot A24}{U_{REF}} = \frac{5 \cdot A24 - 2 \cdot A24}{2} \Rightarrow \Rightarrow A23 = 1,5 \cdot A24 \quad (3.8)$$

Výsledek dosadíme do upravené rovnice (3.7):

$$A_{20} = \frac{A_{24} \cdot A_{23}}{A_{24} + A_{23}}$$

$$A_{20} \cdot A_{24} + A_{20} \cdot A_{23} = A_{24} \cdot A_{23}$$

$$A_{20} \cdot A_{24} + A_{20} \cdot 1,5 \cdot A_{24} = A_{24} \cdot 1,5 \cdot A_{24}$$

Do této odvozené rovnice můžeme dosadit hodnotu odporu A20 a vypočítat odpor A24:

$$128000 \cdot A_{24} + 128 \cdot 10^3 \cdot 1,5 \cdot A_{24} = 1,5 \cdot A_{24}^2$$

$$A_{24} \cdot (128 \cdot 10^3 + 1,5 \cdot 128 \cdot 10^3) = 1,5 \cdot A_{24}^2$$

$$\frac{320 \cdot 10^3}{1,5} = A_{24}$$

$$\Rightarrow A_{24} = 213 \text{ k}\Omega$$

Ted' už zbývá jen dopočítat odpor A23 z rovnice (3.8):

$$A_{23} = 1,5 \cdot 213 \cdot 10^3$$

$$A_{23} = 320 \text{ k}\Omega$$

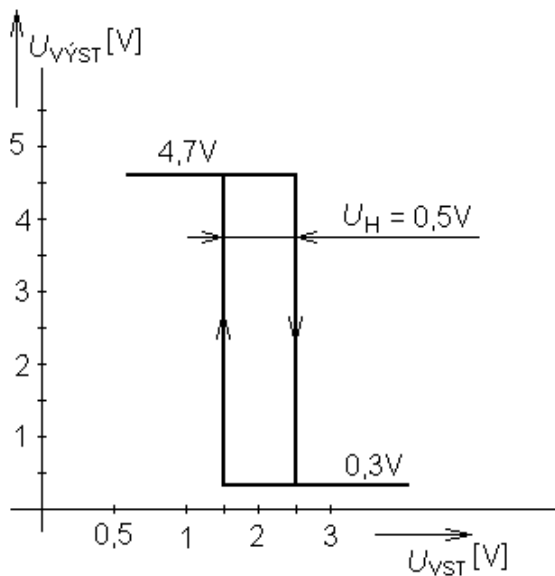
Odpor A22 je většinou volen jako 10 k Ω . Ostatní vypočítané odpory (A20, A23, A24) musíme zvolit z řady E24, čili:

$$A_{20} = 130 \text{ k}\Omega$$

$$A_{23} = 330 \text{ k}\Omega$$

$$A_{24} = 220 \text{ k}\Omega$$

Při zavedení hystereze do komparátoru (při zavedení kladné zpětné vazby) odporem A21 nám komparátor reaguje na dvě různé úrovně napětí, viz. obrázek 3.7 b). Při růstu vstupního napětí dojde k překlopení komparátoru na hodnotě $U_{REF} + U_H/2$. Při následném poklesu vstupního napětí dojde ke zpětnému překlopení na hodnotě $U_{REF} - U_H/2$.



Obr. 3.8: Převodní charakteristika komparátoru s hysterezí

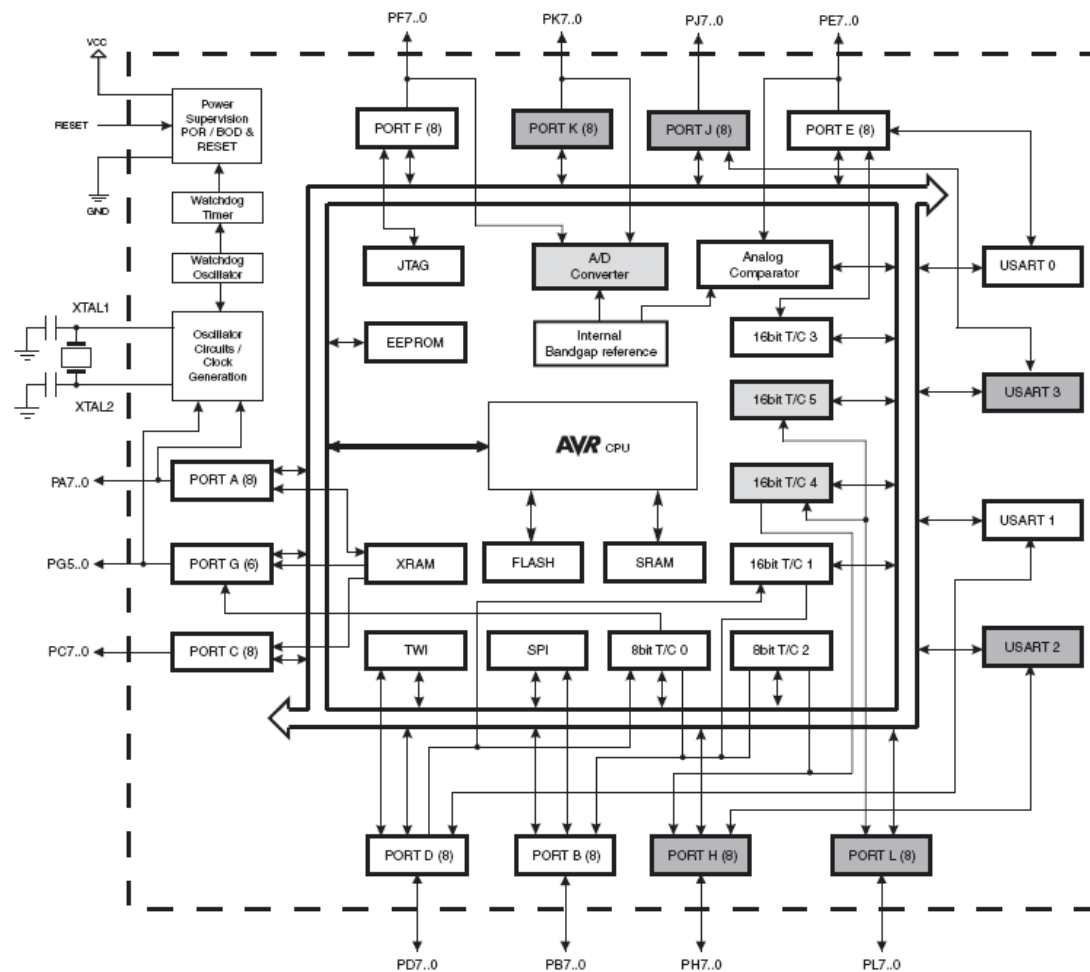
Z komparátoru je signál přiváděn na logické členy XOR.

3.2.2 Aplikační rozhraní pro ATmega2560-16AU

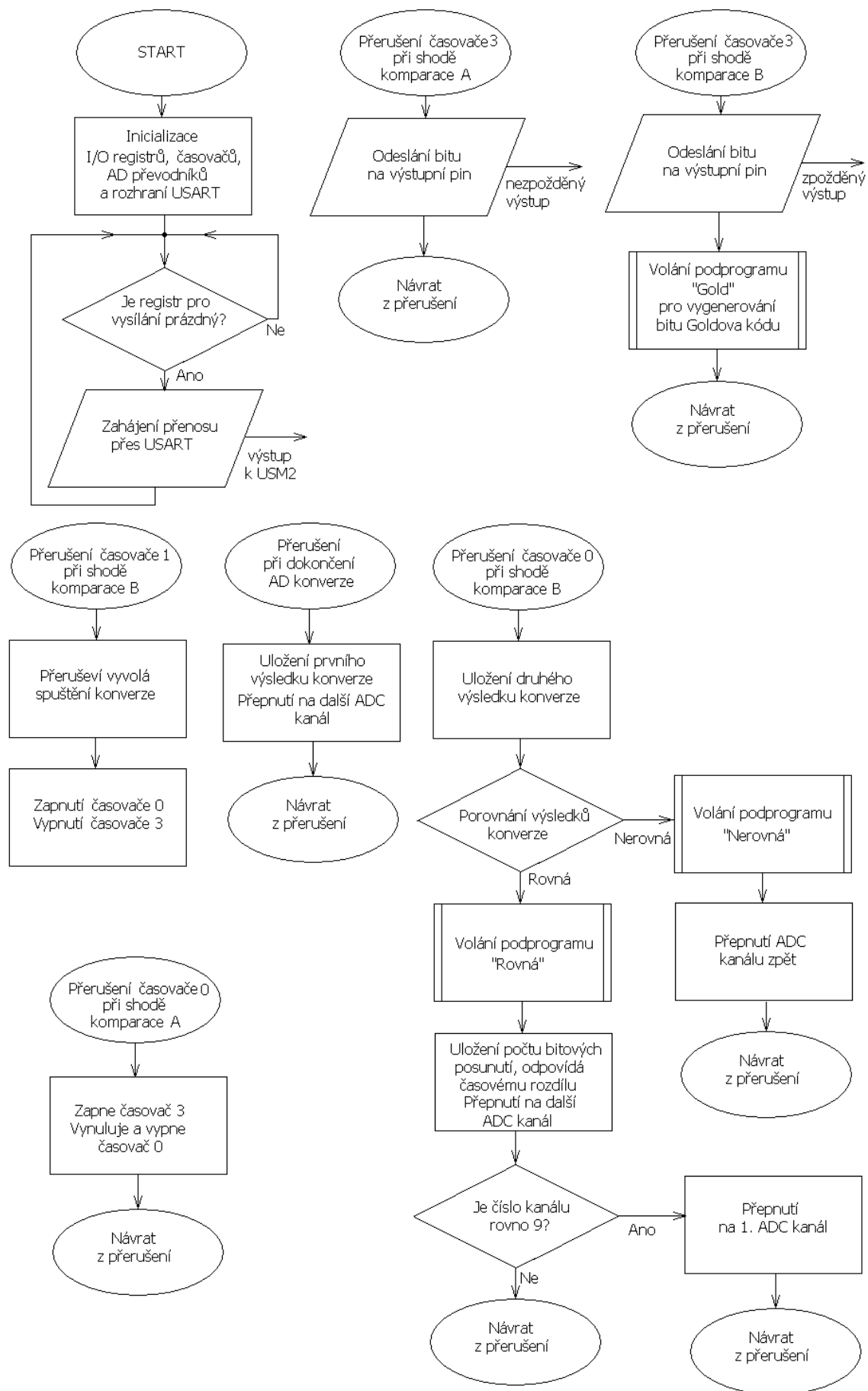
Mikrokontrolér ATMEGA patří do rodiny mikroprocesorů AVR firmy Atmel. Struktura mikrokontroléru AVR je navržena tak, aby co nejvíce vyhovovala překladačům vyšších programovacích jazyků, zejména široce používaného jazyka C. Takto optimalizované jádro s harvardskou architekturou nese hlavní charakteristiky mikroprocesorů s redukovanou instrukční sadou.

V tomto projektu využíváme mikrokontroler řady ATmega2560. Tento mikrokontrolér disponuje poměrně velkou vnitřní pamětí typu Flash, kterou je možné programovat přímo v aplikaci. Takový způsob programování je označován jako In-System Programming (ISP). To znamená, že čip může být osazen přímo na desce plošných spojů a není problémem obsah jeho paměti nahradit jiným obsahem. K tomu je pouze zapotřebí, aby byl patřičný mikrokontrolér vybaven možností sériového programování nebo tzv. SPI-rozhraním, což je v případě této rodiny samozřejmostí. Rozhraní SPI (Serial Peripheral Interface) poskytuje vysokorychlostní synchronní přenos dat mezi mikrokontrolérem a okolními prostředky nebo dvěma mikrokontroléry.

Analogicky lze programovat i vnitřní paměť EEPROM, kterou je rovněž každý mikrokontrolér vybaven. Kromě výše uvedených pamětí je k dispozici ještě paměť SRAM, která slouží běžící aplikaci v mikrokontroléru k odkládání “nepotřebných” dat. Velikosti jednotlivých pamětí samozřejmě záleží na daném typu mikrokontroléru. Tyto obvody mohou pracovat na frekvenci až 40 MHz, disponují až třemi časovači, watchdogem, sběrnici I2C, hodinami reálného času, A/D převodníkem, modulací PWM a řadou dalších vylepšení ([16]: 19).

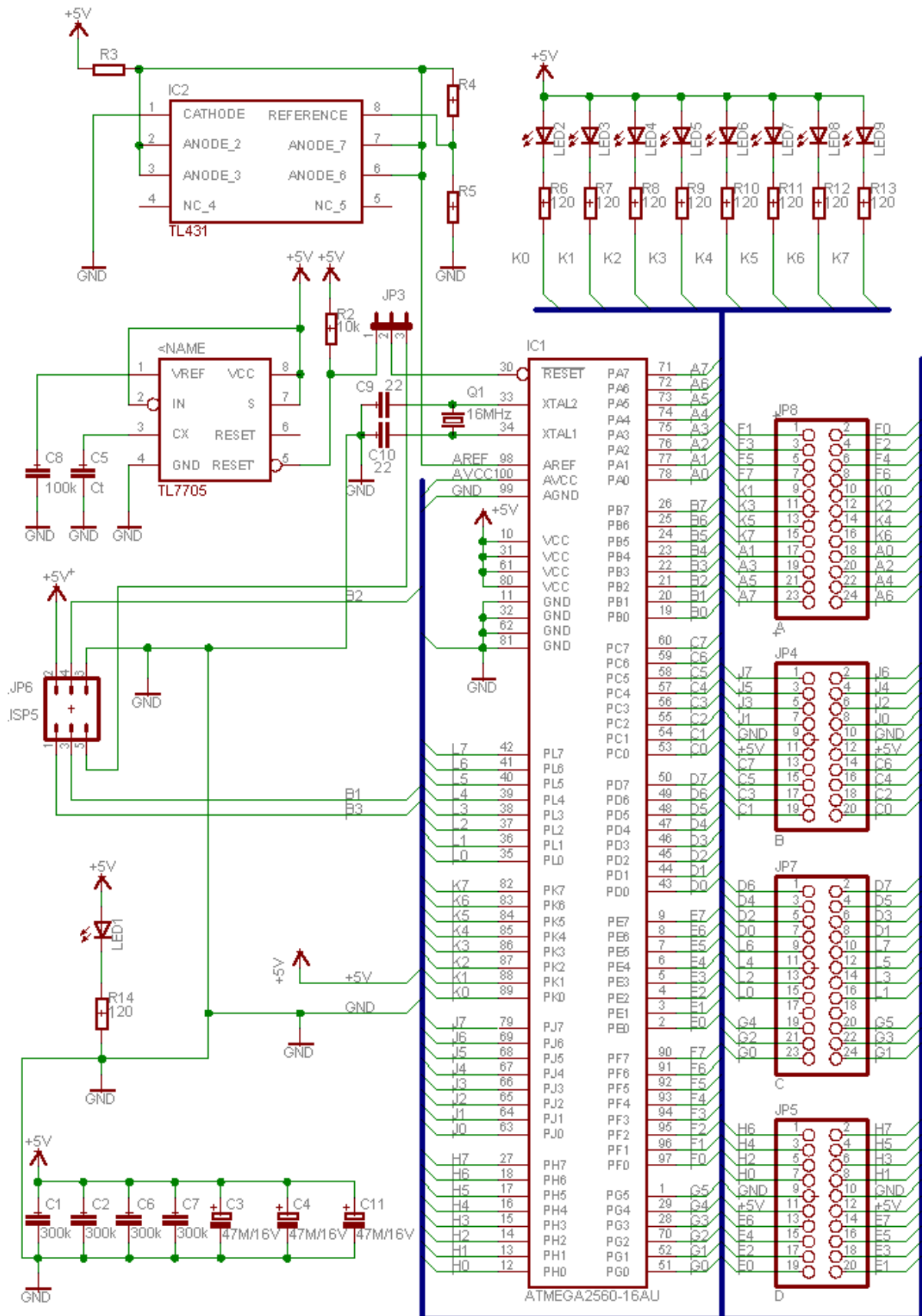


Obr. 3.9: Blokovoé schéma mikrokontroléru ATmega2560-16AU [3]



Obr. 3.10: Vývojový diagram programu pro ATmega 2560-16AU

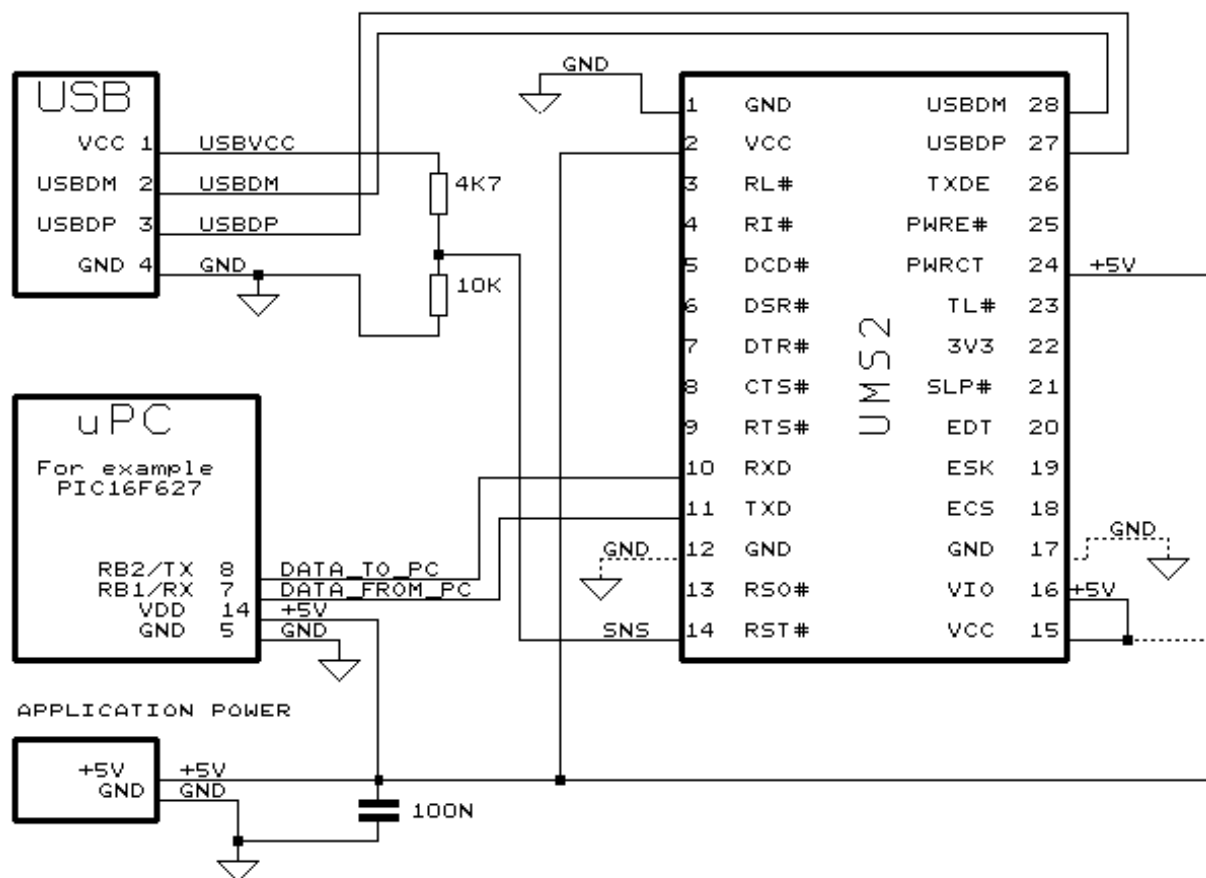
Zdrojový kód mikrokontroleru ATmega 2560 je uveden v příloze 2.



Obr. 3.11: Schéma zapojení aplikačního rozhraní pro ATmega 256

3.2.3 Sériový USB modul UMS2

UMS2 je modul založený na integrovaném obvodu FT8U232BM firmy FTDI, který umožňuje velmi snadné připojení uživatelské aplikace k PC pomocí sběrnice USB, aniž by byly nutné jakékoli předchozí znalosti o problematice USB.



Obr. 3.12: Zapojení UMS pro napájení 5V

UMS2 je převodník mezi sběrnici USB a UART s možností plně hardwarového řízení toku dat. Modul UMS2 umožňuje přenos dat rychlostí až 1 MBd po sběrnici RS232.

Modul je řešen v provedení DIP28. Modul UMS2 umožňuje také provoz v režimu "BitBang mode", kdy se datové vstupy/výstupy chovají jako 8-bitová paralelní sběrnice - data vyslaná z PC budou na sběrnici ihned trvale přítomna. Tento režim umožňuje například konfiguraci programovatelného logického obvodu přímo z PC po sběrnici USB, není ovšem vhodný pro aplikace s vysokými nároky na časování signálů. V modulu UMS2 je osazena paměť 93LC56 o kapacitě 128x16 bitů, z které je pro potřeby USB využita spodní polovina paměti (64x16 bitů). Pomocí pinů EEDATA, EESK a EECS může aplikace využít horních 64x16 bitů pro svoje potřeby.

Vlastnosti modulu UMS2

- Datové vstupy/výstupy 3,0 až 5 V - CMOS kompatibilní
- Napájení modulu ze sběrnice USB nebo externí
- Přenosová rychlost až 1 Mb/s - sběrnice RS232, až 3 Mb/s - sběrnice RS485
- Možnost plně hardwarového řízení toku dat
- Vyrovnávací paměť pro příjem 384 byte
- Vyrovnávací paměť pro vysílání 128 byte
- Podpora řízení převodníku úrovně RS485
- Integrovaný frekvenční násobič 6 MHz - 48 MHz
- Protokol USB 1.1, USB 2.0 kompatibilní
- Podpora OEM aplikací (VID a PID produktu uloženo v paměti EEPROM, která je součástí modulu)
- BitBang režim
- Signál PWREN# určený k řízení tranzistoru MOSFET s P-kanálem u aplikací s odběrem proudu z USB nad 100 mA (max. 500 mA)
- Možnost hardwarové optimalizace toku dat pomocí změny signálu CTS, DSR, DCD nebo RI (více informací v aplikační poznámce č.4)
- Možnost využití volné kapacity v EEPROM (64x16 bitů volné kapacity)
- Ovladače pro WIN 98/2K/ME/XP/CE, Mac OS8/OS9/OS X a Linux zdarma na internetu v sekci „technická podpora“
- Modul je určen pro zasunutí do patice DIL28 nebo pro montáž do PCB ([7]: 22-23).

3.3 Použití a rozšíření systému

Systém můžeme s výhodou použít pro lokalizaci například malých robotů v jedné místnosti. Nutná podmínka pro lokalizaci je přímá viditelnost jelikož používáme ultrazvukový akustický signál a ten by byl téměř zcela utlumen například zdí.

Tato navržená koncepce lokalizačního systému může být rozšířena například:

Pro určení přesného postavení předmětu, dejme tomu podvozku, použijeme dva vysílače, jež budou vysílat dva různé Goldovy kódy.

Můžeme lokalizovat několik zdrojů najednou. počet lokalizovaných zdrojů by potom byl určen celkovou úrovní interferencí C/A sekvencí.

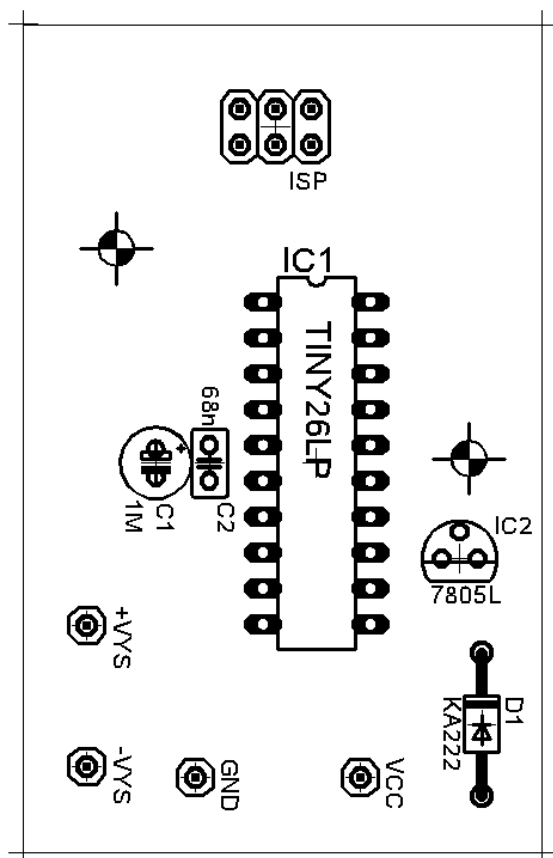
Pro lokalizaci objektů například v celém domě bychom museli signál modulovat na nosnou frekvenci, která by už nebyla tolik tlumena předměty v domě

4 Desky plošných spojů

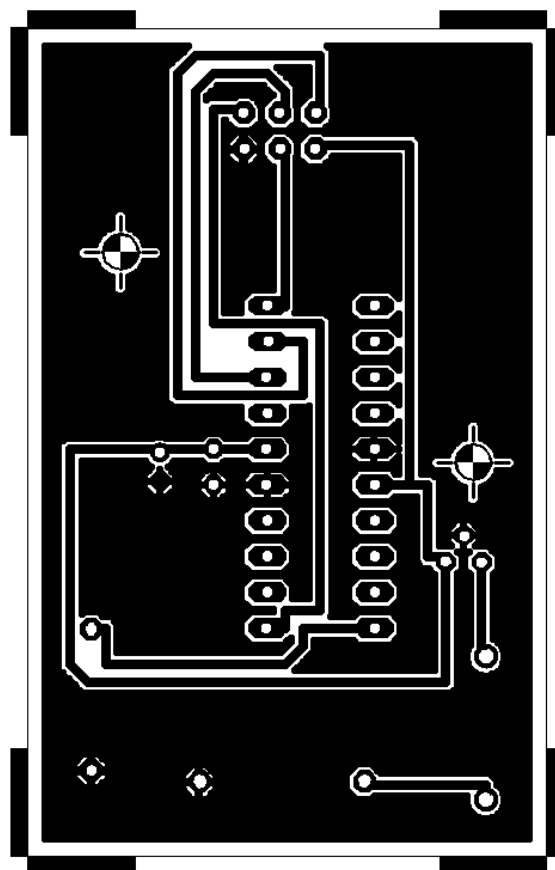
4.1 Vysílač

Rozměry:

- Výška 59mm
- Šířka 36mm



a)



b)

Obr. 4.1: Deska pro plošného spoje pro vysílač

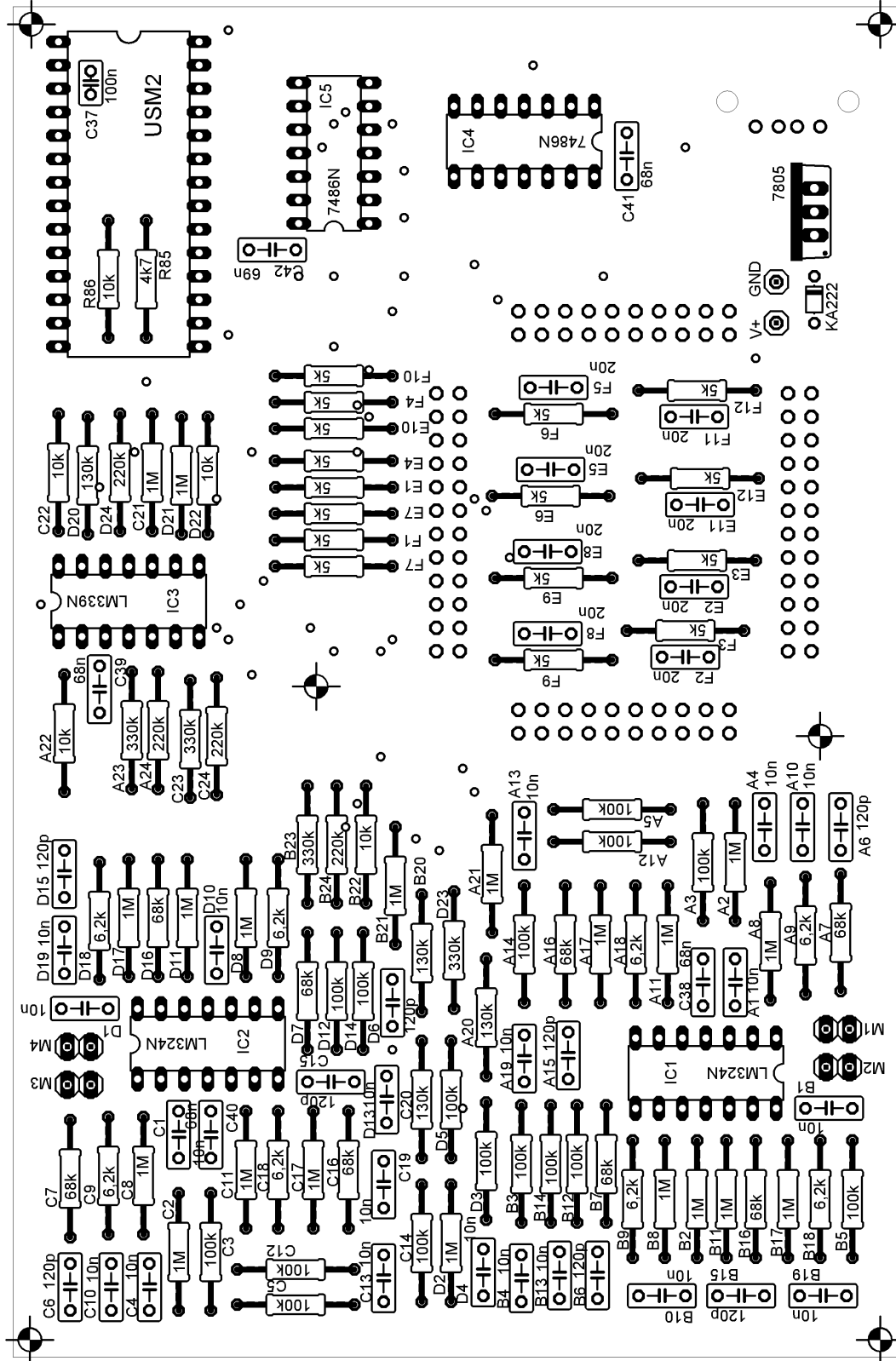
a) rozmístění součástek

b) deska plošného spoje

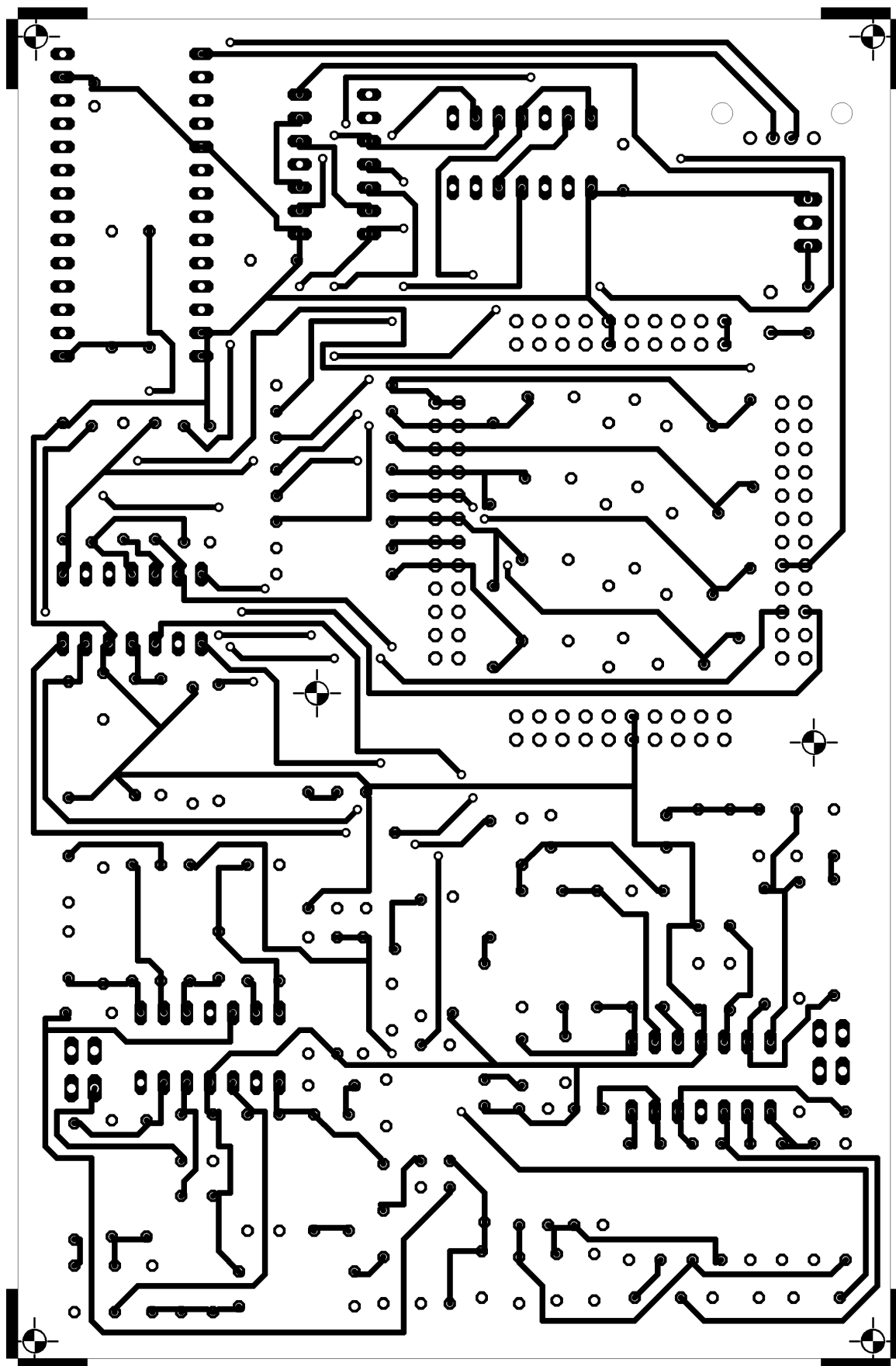
4.2 Přijímač

Deska plošného spoje je vyhotovena jako dvoustranná.

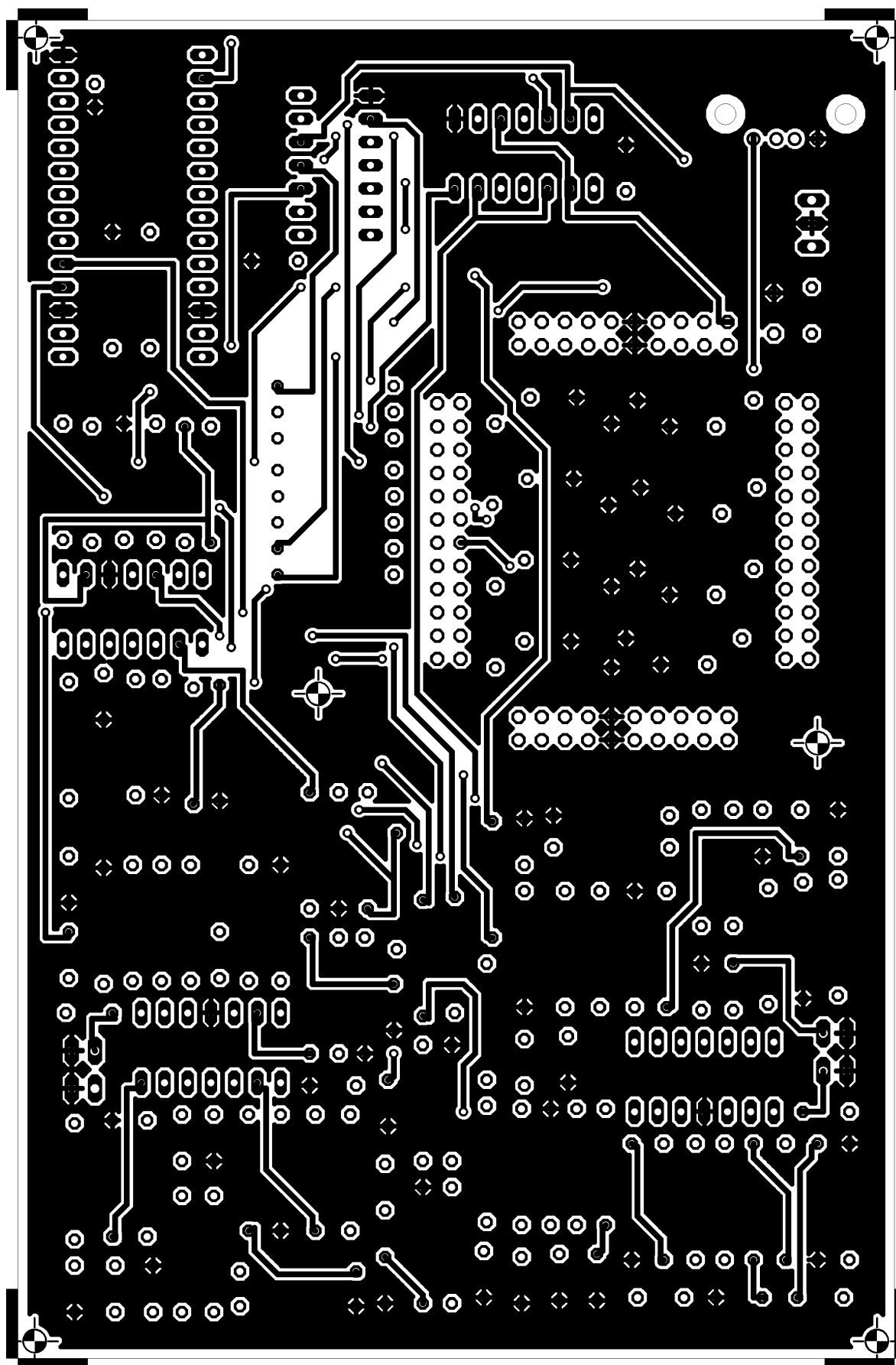
Rozměry: - Výška 146mm
- Šířka 96mm



Obr. 4.2: Rozmístění součástek na desce plošných spojů pro přijímač



Obr. 4.3: Horní vrstva desky plošných spojů pro přijímač

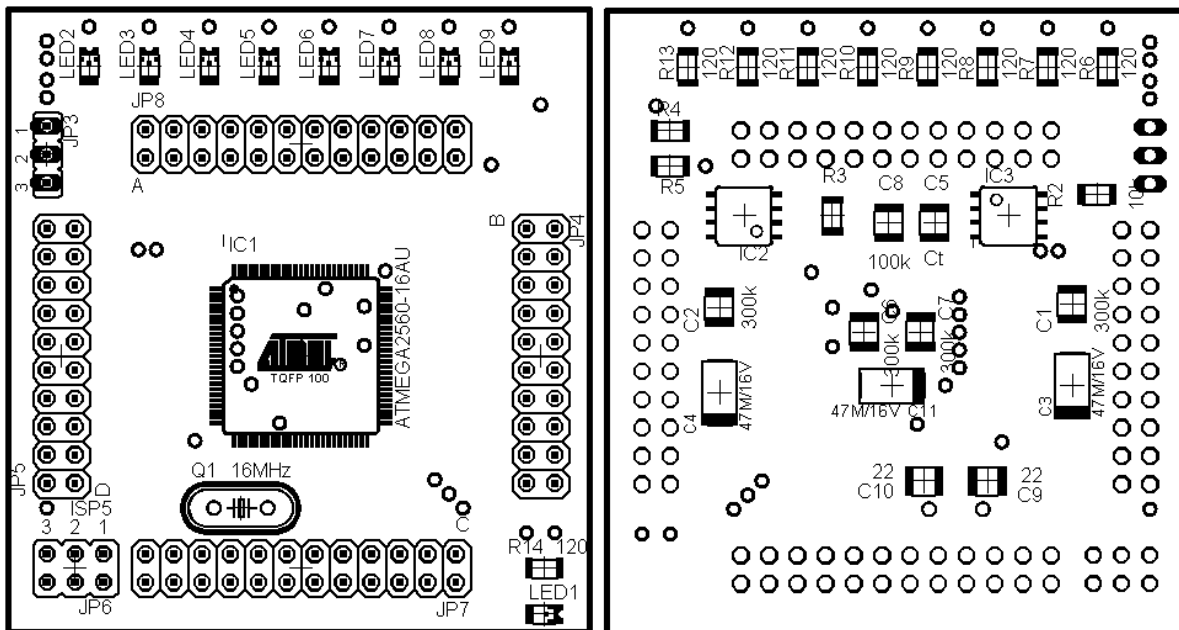


Obr. 4.4: Spodní vrstva desky plošných spojů pro přijímač

4.3 Aplikační rozhraní pro ATmega2560-16AU

Deska plošného spoje je vyhotovena jako dvoustranná.

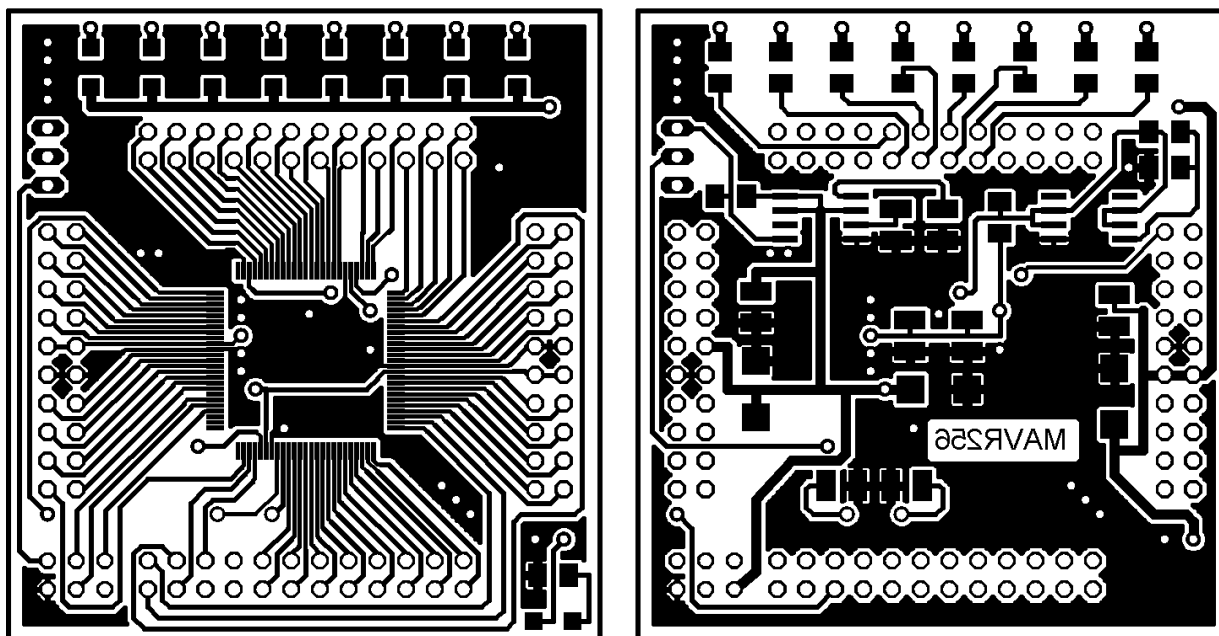
Rozměry: - Výška 56mm
 - Šířka 53mm



a) b)

Obr. 4.5: Rozmístění součástek pro aplikační rozhraní ATmega128

- a) součástky horní vrstvy
- b) součástky spodní vrstvy



a) b)

Obr. 4.6: Deska plošných spojů pro aplikační rozhraní ATmega128

- a) horní vrstva desky
- b) spodní vrstva desky

5 Závěr

V bakalářské práci jsem popsal způsob lokalizace akustických zdrojů metodou TDOA. Navrhl jsem systém pro lokalizaci, rozebral jsem jeho jednotlivé části a navrhl plošné spoje pro praktickou realizaci projektu. Sestavil jsem řídicí programy mikrokontrolerů a osadil navrhnuté desky plošných spojů.

Ověřoval jsem funkčnost desky vysílače a jednotlivých částí desky přijímače - bloků zesilovačů, komparátorů a analogových násobiček. Při testování aplikačního rozhraní ATmega2560-16AU se mi nepodařilo dostatečně odladit řídicí program, a proto jsem ani nemohl ověřit fungování převodníku USART na USB.

6 Seznam použité literatury

- [1] BEZOUŠEK, P., ŠEDIVÝ, P. *Radarová technika*. Skriptum ČVUT Praha, 2004
- [2] *Atmel Corporation ATtiny26 8-bit Low Voltage Microcontroller with 4K Bytes In-System Programmable Flash*. Atmel Corporation, 2007, www.atmel.com
- [3] *Atmel Corporation ATmega2560-16AU 8-bit Low Voltage Microcontroller with 4K Bytes In-System Programmable Flash*. Atmel Corporation, 2007, www.atmel.com
- [4] MUHAMMED, A. *Evaluation of TDOA techniques for position location in CDMA systems*. Blacksburg Virginie, 1997
- [5] ŠEBESTA, J. *Radiolokace a radionavigace*. Skriptum VUT Brno, 2003
- [6] MATOUŠEK, D. *USB prakticky*. BEN Praha, 2003
- [7] [Uživatelská příručka modulu UMS2](http://www.asix.cz/a6ums2.htm) <<http://www.asix.cz/a6ums2.htm>>
- [8] HEGARTY, C., KAPLAN, E. *Understanding GPS: Principles and applications*. ARTECH HOUSES London, 2005
- [9] *Low power quad operational amplifier LM324*. ST Microelectronics, 2001, www.datasheetcatalog.com
- [10] *Air Ultrasonic Ceramic Transducers 400ST/R160*. www.datasheetarchive.com
- [11] PUNČOCHÁŘ, J. *Operační zesilovače v elektronice*. BEN Praha, 2002
- [12] *Multilaterace (TDOA)*. <<http://www.czechspace.cz/cs/galileo/aktuality-GPS-Glonass/GNSS-urcovani-polohy/multilaterace>>
- [13] *Single Supply Quad Comparator LM339*. ON Semiconductor, 2007, www.onsemi.com
- [14] *Positive Voltage regulator 7805*. SGS-THOMSON Microelectronics, 1994, www.datasheet4U.com
- [15] *Quadruple 2-input Exclusive-OR Gate SN74HC86*. Texas Instruments Incorporated, 2005, www.ti.com
- [16] Klíma, D., Korbel, S., Jáne, V., (2004): *Zajímavé aplikace mikrokontrolérů AVR firmy Atmel*. <<http://www.stech.cz/articles.asp?ida=323&idk=267>> (cit. 4.3. 2007)

Příloha č. 1: Zdrojový kód pro ATtiny26

```
.include<tn26def.inc>
.def ah=r16
.def al=r17
.def bh=r18
.def bl=r19
.def c=r20
.def d=r21
.def e=r22
.cseg
.org 0x000
    rjmp start
.org 0x003
    rjmp timer

start:
ldi ah,0b00000001
ldi al,0b00000010
ldi bh,0b00000011
ldi bl,0b00101110           //prvotní naplnění registrů

ldi c,0b10000000
out DDRA,c

ldi c,0x58
out OCR1A,c                //nastavení o compare
ldi c,(1<<OCIE1A)
out TIMSK,c                //povolení přerušení čítače při CM
ldi c,(1<<CS10)
out TCCR1B,c               //nastavení Prescale pto Timer/Counter1 = 1CK

gold:

ldi d,0                    //vynulování před dalším cyklem
ror ah                     //posun registru s CARRY
ror al                     //posun registru s CARRY
out sreg,d                //vynulování příznakového bitu CARRY

ror bh                     //posun registru s CARRY
ror bl                     //posun registru s CARRY
out sreg,d                //vynulování příznakového bitu CARRY

ldi c,0
ldi e,0                    //vynulování před dalším cyklem

sbrc bh,0                  //příprava 1.xoru
sbr c,0b00000001          //příprava 1.xoru
sbrc bl,4                  //příprava 1.xoru
sbr d,0b00000001          //příprava 1.xoru   výber odboček 2 & 6

sbrc al,7                  //příprava 5.xoru
sbr c,0b00000100          //příprava 5.xoru
sbrc al,0                  //příprava 5.xoru
sbr d,0b00000100          //příprava 5.xoru

sbrc bl,4                  //příprava 3.xoru
sbr c,0b00001000          //příprava 3.xoru
sbrc bl,2                  //příprava 3.xoru
```


Příloha č.1

```
sbr d,0b00001000 //příprava 3.xoru

sbrc bl,1 //příprava 4.xoru
sbr c,0b00010000 //příprava 4.xoru
sbrc bl,0 //příprava 4.xoru
sbr d,0b00010000 //příprava 4.xoru
eor c,d //naráz provede XOR pro 1. 5. 3. a 4. xor

sbrc c,2 //je-li 2.bit v "c" = "0" přeskočí operaci OR
  rjmp jednicka //je-li 2.bit v "c" = "1" skok na "jednicka"
  sbrc ah,2 //je-li 2.bit v "ah" = "0" přeskočí AND
  andi ah,0b00000011 //vynulování 2.bit v "ah" ostatní nezmění
  rjmp konec //skok na konec
jednicka: ori ah,0b00000100 //2.bit v "c" nastaví na "1"
konec:

ldi d,0

sbrc c,0 //příprava 6.xoru
sbr d,0b10000000 //příprava 6.xoru
sbrc al,0 //příprava 6.xoru
sbr e,0b10000000 //příprava 6.xoru

sbrc c,3 //příprava 7.xoru
sbr d,0b00000010 //příprava 7.xoru
sbrc c,4 //příprava 7.xoru
sbr e,0b00000010 //příprava 7.xoru

sbrc bh,0 //příprava 2.xoru
sbr d,0b00000100 //příprava 2.xoru
sbrc bl,7 //příprava 2.xoru
sbr e,0b00000100 //příprava 2.xoru
eor d,e //naráz provede XOR pro 6. 7. 2. xor

ldi e,0
ldi c,0

sbrc d,2 //příprava 8.xoru
sbr e,0b00000001 //příprava 8.xoru
sbrc d,1 //příprava 8.xoru
sbr c,0b00000001 //příprava 8.xoru
eor e,c //provede XOR pro 8. xor

sbrc e,0 //je-li 2.bit v "c" = "0" přeskočí operaci OR
  rjmp jednicka2 //je-li 2.bit v "c" = "1" skok na "jednicka2"
  sbrc bh,2 //je-li 2.bit v "ah" = "0" přeskočí
  andi ah,0b00000011 //vynulování 2.bit v "ah" ostatní nezmění
  rjmp konec2 //skok na konec2
jednicka2: ori bh,0b00000100 //2.bit v "c" nastaví na "1"
konec2:

sei //povolení globálního přerušeni
loop: //nekonečná smyčka
  rjmp loop //skok na návěští "loop"

timer:
  nop
  out PORTA,d //odeslání výstupního bitu na pin ldi c,0x00
  out TCNT1,c //nulování časovače
  rcall gold //volání pro výpočet dalšího výstupního bitu
  reti
```

Příloha č. 2: Zdrojový kód pro ATmega2560

```
.include<m2560def.inc>

.def ah=r16
.def al=r17
.def bh=r18
.def bl=r19
.def c=r20
.def d=r21
.def e=r22
.def pr1=r23
.def pr2=r24
.def pr3=r25
.def pr4=r26
.def mux=r27
.def posuv=r28

.cseg
.org 0x0000
    rjmp start
.org $0024
    rjmp timer1b
.org $002A
    rjmp timer0a
.org $002C
    rjmp timer0b
.org $003A
    rjmp prevodnik
.org $0040
    rjmp timer3a
.org $0042
    rjmp timer3b
.org $004C
    rjmp usart

start:
ldi ah,0b00000001
ldi al,0b00000010
ldi bh,0b00000011
ldi bl,0b00101110                //prvotní naplnění registrů

ldi c,0b10000000
sts DDRL,c                      //nastavení výstupních pinů v portu L
sts DDRH,c                      //nastavení výstupních pinů v portu H

;Timer Counter0
ldi c,(1<<OCIE0A)|(1<<OCIE0B)   //povolení přerušení čítače0
sts TIMSK0,c                   //při Compare Match A a B

ldi c,52                        //nastavení registru OCR0A pro compare match A
out OCR0A,c                    //zajištění zpoždění přesně o jeden bit

ldi c,26                        //nastavení registru OCR0B pro compare match B
out OCR0B,c                    //zajištění zpoždění přesně o jeden bit

;Timer Counter1
ldi c,0b00000001
sts OCR1BH,c
```

Příloha č.2

```
ldi c,0b10000000 //nastavení compare match B = 384
sts OCR1BL,c //přesné spouštění ADC

ldi c,(5<<CS10)
sts TCCR1B,c //zapnutí TC1 s Prescale = CK/1024

ldi c,(1<<OCIE1B) //povolení přerušení čítače1
sts TIMSK1,c //při shodě komparace(Compare Match B)

;Timer Counter3
ldi c,0xb8 //nastavení registru OCR3AL pro compare match
A =184
sts OCR3AL,c //(zajištění přesného kmitočtu)

ldi c,0b00000001
sts OCR3BH,c
ldi c,0b10000000 //nastavení compare match B = 384
sts OCR3BL,c //(zajištění přesného kmitočtu)

ldi c,(1<<CS30)
sts TCCR3B,c //zapnutí TC3 s Prescale = CK

ldi c,(3<<OCIE3A) //povolení přerušení čítače3
sts TIMSK3,c //při shodě komparace(Compare Match A a B)

;USART
ldi c,0x67
sts 0xcc,c //nastavení baudrate na 9600Baudů

ldi c,(1<<RXEN1)|(1<<TXEN1)
sts UCSR1B,c //zapnutí vysílače a přijímače USART
ldi c,(1<<USBS1)|(3<<UCSZ10)
sts UCSR1C,c //USART: 8data, 2stop bity, bez parity

;ADC
ldi c,(1<<ADEN)|(1<<ADSC)|(1<<ADIF) //zapnutí AD převodníku, auto-
triggeringu a
sts ADCSRA ,c //povolení přerušení

ldi c,(1<<REFS0)|(1<<ADLAR) //nastavení externího ref. napětí (AVCC)
sts ADMUX,c //a ADC left adjust result

ldi c,(1<<MUX5)|(5<<ADTS0)
sts ADCSRB,c //nastavení triggeringu při TC1-CM B

ldi c,0b10101010 //vypnutí vstupů ADC v registru
sts 0x7e,c //DIDR0 a to ADC1,ADC3,ADC5,ADC7
sts 0x7d,c //DIDR2 o to ADC9,ADC10,ADC11,ADC13

sei //povolení globálního přerušení

;nekonečná smyčka
loop:

Transmit:
lds c,UCSR1A
sbrs c,5 //čekání na vyprázdnění bufferu pro vysílání
rjmp Transmit
sts UDR1,c //odelání dat na výstup

rjmp loop //skok na návěští "loop"
```

Příloha č.2

```
gold:

clc                //vynulování příznakového bitu CARRY
ror ah            //posun registru s CARRY
ror al            //posun registru s CARRY
clc                //vynulování příznakového bitu CARRY

ror bh            //posun registru s CARRY
ror bl            //posun registru s CARRY
CLC                //vynulování příznakového bitu CARRY

ldi c,0
ldi d,0
ldi e,0            //vynulování před dalším cyklem

sbrc bh,0         //příprava 1.xoru
sbr c,0b00000001 //příprava 1.xoru
sbrc bl,4         //příprava 1.xoru
sbr d,0b00000001 //příprava 1.xoru výběr odboček 2 & 6

sbrc al,7         //příprava 5.xoru
sbr c,0b00000100 //příprava 5.xoru
sbrc al,0         //příprava 5.xoru
sbr d,0b00000100 //příprava 5.xoru

sbrc bl,4         //příprava 3.xoru
sbr c,0b00001000 //příprava 3.xoru
sbrc bl,2         //příprava 3.xoru
sbr d,0b00001000 //příprava 3.xoru

sbrc bl,1         //příprava 4.xoru
sbr c,0b00010000 //příprava 4.xoru
sbrc bl,0         //příprava 4.xoru
sbr d,0b00010000 //příprava 4.xoru

eor c,d           //naráz provede XOR pro 1. 5. 3. a 4. xor výsledek
                  //je v reg "c" na 0. 2. 3. a 4. bitu

sbrc c,2          //je-li 2.bit v "c" = "0" přeskočí operaci OR
  rjmp jednicka  //je-li 2.bit v "c" = "1" skok na "jednicka"
  sbrc ah,2      //je-li 2.bit v "ah" = "0" přeskočí AND
  andi ah,0b00000011 //vynulování 0. a 1.bit v "ah" ostatní nezmění
  rjmp konec     //skok na konec
  jednicka: ori ah,0b00000100 //2.bit v "c" nastaví na "1"
  konec:

ldi d,0

sbrc c,0          //příprava 6.xoru
sbr d,0b10000000 //příprava 6.xoru
sbrc al,0         //příprava 6.xoru
sbr e,0b10000000 //příprava 6.xoru

sbrc c,3          //příprava 7.xoru
sbr d,0b00000010 //příprava 7.xoru
sbrc c,4          //příprava 7.xoru
sbr e,0b00000010 //příprava 7.xoru

sbrc bh,0        //příprava 2.xoru
sbr d,0b00000100 //příprava 2.xoru
```

Příloha č.2

```
sbrc bl,7 //příprava 2.xoru
sbr e,0b00000100 //příprava 2.xoru
eor d,e //naráz provede XOR pro 6. 7. 2. xor výsledek
//je v reg "d" na 7. 1. 2. bitu

ldi e,0
ldi c,0

sbrc d,2 //příprava 8.xoru
sbr e,0b00000001 //příprava 8.xoru
sbrc d,1 //příprava 8.xoru
sbr c,0b00000001 //příprava 8.xoru
eor e,c //provede XOR pro 8. xor

sbrc e,0 //je-li 2.bit v "c" = "0" přeskočí operaci OR
rjmp jednicka2 //je-li 2.bit v "c" = "1" skok na "jednicka2"
sbrc bh,2 //je-li 2.bit v "ah" = "0"
andi ah,0b00000011 //vynulování 0. a 1.bit v "ah" ostatní nezmění
rjmp konec2 //skok na
jednicka2: ori bh,0b00000100 //2.bit v "c" nastaví na "1"
konec2:

ret

;podprogram pro případ že se rovnají dvě příslušné konverze
rovna:
mov pr1,posuv //zaznamenání "času" příchodu
ldi c,1
add mux,c //přepnutí ADC kanálu
sbrc mux,3
ldi mux,0
ret

;podprogram pro případ že se NErovnejí dvě příslušné konverze
nerovna:
subi mux,1 //přepnutí ADC kanálu zpět o jeden dolů
ret

timer3a:
sts $010b,d //odeslání výstupního bitu na pin PL7
reti

timer3b:

sts $0102,d //odeslání výstupního bitu na pin
ldi c,0x00
sts TCNT3H,c
sts TCNT3L,c //nulování časovače3
rcall gold //volání pro výpočet dalšího výstupního bitu
reti

timer1b:
ldi c,0x00
sts TCCR3B,c //vypnutí TC3 (realizace posunutí o jeden bit)
ldi c,(1<<CS01)
out TCCR0B,c //zapnutí TC0 s Prescale clk/8
ldi c,0
sts TCNT1H,c
sts TCNT1L,c //nulování časovače1

reti
```

Příloha č.2

```
timer0a:
    ldi c, (1<<CS30)           //opětovné zapnutí TC3 s Prescale = CK
    sts TCCR3B,c               //(realizace posunutí o jeden bit)
    ldi c,0
    out TCCR0B,c               //vypnutí TC0
    ldi c,0
    out TCNT0,c                //vynulování TC0
    ldi c,1
    add posuv,c                //počet posunutí Golddova kódu
    reti

timer0b:
    lds e,ADCH                 //uložení výsledku konverze
    andi e,0b11110000         //použití pouze 4 MSB
    cpse d,e
    rjmp ne
    call rovna
ne:
    cpse d,e
    call nerovna
    reti

prevodnik:
    lds d,ADCH                 //uložení výsledku konverze
    andi d,0b11110000         //použití pouze 4 MSB
    ldi c,1
    add mux,c                  //přepnutí ADC kanálu
    ldi c, (1<<ADSC)
    sts ADCSRA,c              //spuštění další konverze
    reti

usart:
    reti
```