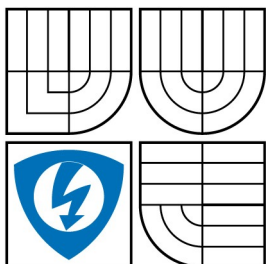


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ
ÚSTAV MIKROELEKTRONIKY



FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF MICROELECTRONICS

BEZDRÁTOVÝ ZVONEK S DIGITÁLNÍM PŘENOSEM HLASU

WIRELESS DOORBELL WITH VOICE TRANSMISSION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. JOSEF PAPEŽ

VEDOUcí PRÁCE
SUPERVISOR

ING. RADEK VÁCLAVÍK

BRNO 2008

uzavřená mezi smluvními stranami:

(dále jen „autor“)

a

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 602 00, Brno
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
Ing. Edita Hejátková

(dále jen „nabyvatel“)

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:	<u>Bezdrátový zvonek s digitálním přenosem hlasu</u>
Vedoucí/ školitel VŠKP:	<u>Ing. Radek Václavík, ON Semiconductor</u>
Ústav:	<u>Ústav mikroelektroniky</u>
Datum obhajoby VŠKP:	

VŠKP odevzdal autor nabyvateli v^{*}:

- | | | |
|--|---|-------------------|
| ☒ tištěné formě | – | počet exemplářů 2 |
| ☒ elektronické formě | – | počet exemplářů 2 |

* hodící se zaškrtněte

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.

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: 26.5.2008

.....
Nabyvatel

.....
Autor

Abstrakt:

V této práci je ukázána možnost realizace bezdrátového domovního zvonku s poloduplexním přenosem hlasu a s bateriovým napájením. Cílem práce je navrhnout zapojení daného systému, návrh DPS a její výroby včetně vytvoření obslužného firmware pro řídicí mikrokontrolér.

Abstract:

This diploma thesis deals with wireless doorbell realisation. The doorbell is battery-powered and supports half duplex voice transferring. The general aim is to design system circuitry, PCB and to create microcontroller firmware.

Klíčová slova:

Bezdrátový zvonek, nRF9E5, přenos hlasu, poloduplexní přenos, 433 MHz

Key words:

Wireless doorbell, nRF9E5, voice transfer, half duplex transmission, 433 MHz

Bibliografická citace díla:

PAPEŽ, J. *Bezdrátový zvonek s digitálním přenosem hlasu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 57 s. Vedoucí diplomové práce Ing. Radek Václavík, ON Semiconductor.

Prohlášení autora o původnosti díla:

Prohlašuji, že jsem tuto vysokoškolskou kvalifikační práci vypracoval samostatně pod vedením vedoucího diplomové práce, s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury. 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 26.5.2008

Poděkování:

Děkuji vedoucímu diplomové práce Ing. Radku Václavíkovi za všechnu pomoc, kterou mě dovedl ke zdárnému dokončení tohoto díla a vedoucímu Ing. Jiřímu Stehlíkovi za věnovaný čas.

S díky se obracím také ke svojí rodině, která mi poskytovala potřebné zázemí.

Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

Ústav mikroelektroniky

Popisný soubor závěrečné práce

Autor: Bc. Josef Papež

Název závěrečné práce: Bezdrátový zvonek s digitálním přenosem hlasu

Název závěrečné práce ENG: Wireless doorbell with voice transmission

Anotace závěrečné práce: V této práci je ukázána možnost realizace bezdrátového domovního zvonku s poloduplexním přenosem hlasu a s bateriovým napájením. Cílem práce je navrhnout zapojení daného systému, návrh DPS a její výroby včetně vytvoření obslužného firmware pro řídicí mikrokontrolér.

Anotace závěrečné práce ENG: This diploma thesis deals with wireless doorbell realisation. The doorbell is battery-powered and supports half duplex voice transferring. The general aim is to design system circuitry, PCB and to create microcontroller firmware.

Klíčová slova: Bezdrátový zvonek, nRF9E5, přenos hlasu, poloduplexní přenos, 433 MHz

Klíčová slova ENG: Wireless doorbell, nRF9E5, voice transfer, half duplex transmission, 433 MHz

Typ závěrečné práce: diplomová práce

Datový formát

elektronické verze: pdf

Jazyk závěrečné práce: čeština

Přidělovaný titul: Ing.

Vedoucí závěrečné práce: Ing. Radek Václavík, ON Semiconductor

Škola: Vysoké učení technické v Brně

Fakulta: Fakulta elektrotechniky a komunikačních technologií

Ústav / ateliér: Ústav mikroelektroniky

Studijní program: Elektrotechnika, elektronika, komunikační a řídicí technika

Studijní obor: Mikroelektronika

Obsah

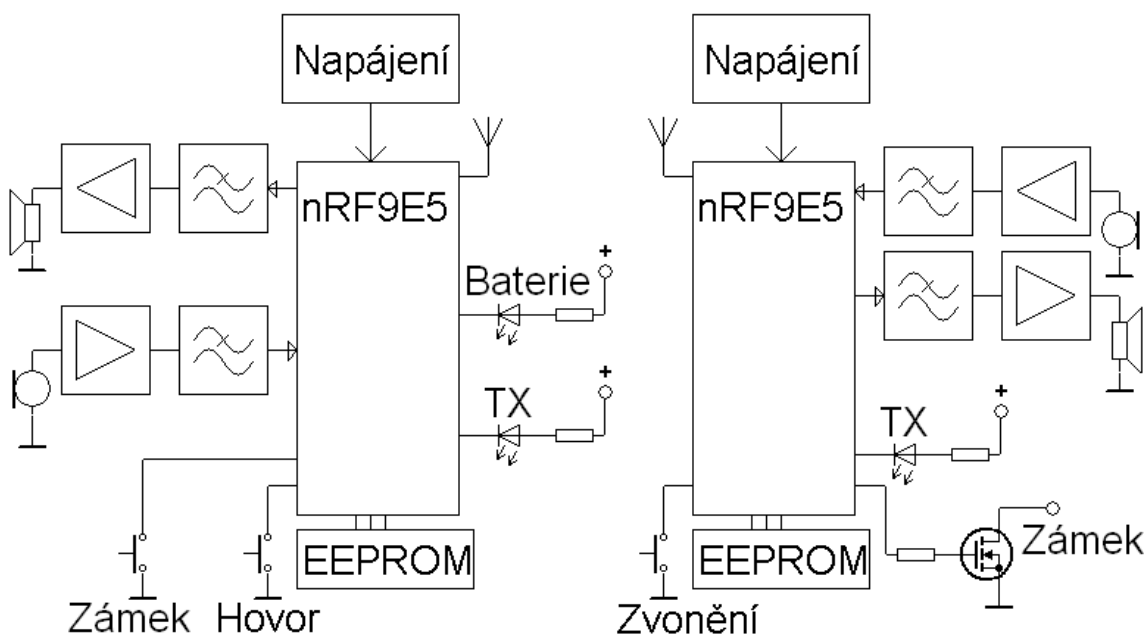
Zadání diplomové práce:	9
Rozbor zadání:	10
Využívání kmitočtové oblasti 433 a 868 MHz:	12
Možnosti obvodu nRF9E5:	13
SPI a konfigurační registry:	15
Transceiver:	19
A/D převodník:	21
D/A převodník PWM:	23
Obecné I/O porty:	24
Zkušební moduly transceiveru nRF9E5:	25
Popis vnitřní stanice:	26
Mikrofonní předzesilovač:.....	27
Antialiasingový filtr.....	29
Antiimaging filtr.....	31
Zesilovač pro reproduktor.....	32
Řídící mikrokontrolér vnitřní stanice:	33
Rozvod napájení.....	35
Vzorkovací kmitočet audiosignálu a přenosová rychlost	36
Popis vnější stanice:	38
Popis ovládacího programu:	39
Naměřené výsledky:	40
Závěr:	42
Seznam použité literatury:	43
Příloha A – bezdrátový zvonek:.....	44
Příloha B – modul s nRF9E5:	48
Příloha C – výpis zdrojového programu pro nRF9E5:	50

Zadání diplomové práce:

Zkonstruujte bezdrátový zvonek pracující na kmitočtu 433 či 886 MHz, který mimo signalizace zvonění dokáže plnit i funkci poloduplexního interkomu (domácího telefonu). Realizace by měla být na bázi mikroprocesoru s vysokofrekvenčním čipem nRF9E5 firmy Nordic, který celou konstrukci velmi zjednodušuje. Digitalizace hlasu bude provedena vestavěným A/D převodníkem s rozlišením 8bitů, rekonstrukce audia pak pomocí PWM modulu a aktivního filtru. Student by měl mít zkušenosti s programováním mikroprocesorů x51 nebo AVR v jazyce C a s návrhem a realizací plošných spojů.

Rozbor zadání:

Konstrukce bezdrátového zvonku je rozdělena na vnější a vnitřní stanici. Zjednodušení konstrukce dosažené použitím obvodu nRF9E5 spočívá v přítomnosti všech důležitých obvodů na jednom společném čipu. Protože obvod obsahuje obecné vstupní i výstupní linky, vysílač i přijímač a A/D i D/A převodník, může být stejný obvod použit v obou blocích telefonu, ve vysílači i přijímači. Vnitřní stanice může být napájena ze sítě pomocí adaptéru, u vnější stanice se počítá s nezávislým napájením z baterie. Blokové schéma celého zařízení ukazuje obr. 1.



Obr. 1: Blokové schéma bezdrátového zvonku

Obě části bezdrátového zvonku jsou vzájemně propojeny bezdrátovým vysokofrekvenčním přenosem na kmitočtu 433 MHz. Přenos dat má dvě části. Stiskem tlačítka na vnější stanici se začne vysílat signál zvonění (bim - bam). Pokud jej vnitřní stanice zachytí, ozve se signál zvonku v reproduktoru. Druhá část datového přenosu se spustí stiskem tlačítka „hovor“ na vnitřní stanici, přenáší se digitalizovaný zvukový signál mezi vnitřní a vnější stanicí jedním nebo druhým směrem, podle stisku ovládacího tlačítka.

Pro digitalizaci i rekonstrukci zvukového signálu vystačíme s převodníky na čipu nRF9E5, pro zajištění přijatelné jakosti zvuku prochází rekonstruovaný zvuk aktivním filtrem typu dolní propust s operačním zesilovačem. Vnější stanice obsahuje výstupní tranzistor NMOS v zapojení s otevřeným kolektorem, kterým může být ovládána závora dveří (zámek). Sepnutí zámku je řízeno stiskem tlačítka Lock ve vnitřní stanici během jejího vysílání, tím se vyšle mezi hlasovými daty řídicí bit, který vnější stanice vyhodnotí jako příkaz k sepnutí tranzistoru zámku. Ovládací tlačítka jsou připojena přímo k I/O portům mikrokontroléru. Obě stanice obsahují svítivou diodu LED, která se rozsvěcuje, pokud daná stanice právě vysílá. Vnitřní stanice navíc druhou diodou oznamuje, že napájecí napětí vnější stanice kleslo pod přijatelnou úroveň. Takto je obsluha zvonku upozorněna na nutnost vyměnit včas baterie.

Diplomová práce nejprve seznamuje s podmínkami využití kmitočtové oblasti 433 a 868 MHz, dále představuje možnosti a nastavování řídicího obvodu nRF9E5 a nakonec po jednotlivých blocích seznamuje s realizací systému bezdrátového zvonku, včetně naměřených vlastností realizovaného systému.

Využívání kmitočtové oblasti 433 a 868 MHz:

Vysílání s krátkým dosahem na kmitočtech 433 MHz a 868 MHz je možno provádět pouze v souladu s všeobecným oprávněním č. VO-R/10/08.2005-24, které vydal Český telekomunikační úřad v červenci 2005 [3]. Tato zařízení mohou být používána bez individuálního oprávnění a pouze s takovou anténou, která zaručuje splnění podmínky maximálního vyzářeného výkonu. Vyzářený výkon je omezen z důvodu zamezení vzájemného rušení v případě, kdy by na této frekvenci vysílalo více vysílačů. Ze stejného důvodu je také zakázáno používat přídavné zesilovače a převaděče a navržená stanice nesmí být dále měněna. Vysílače jsou provozovány na sdílených kmitočtech, proto není zaručena ochrana před škodlivým rušením dalšími stanicemi na stejné frekvenci. Pokud dochází k rušení i při dodržení maximálního vysílaného výkonu, řešení závisí na vzájemné dohodě mezi uživateli. Na kmitočtu 433MHz lze využít pásmo 433,05 až 434,79 MHz a podle vyhlášky je možné vysílat s výkonem 10 mW max. 6 minut za hodinu (klíčovací poměr je max. 10%), nebo při výkonu 1 mW nepřetržitě. V oblasti 868 MHz lze vysílat v celém pásmu 868,0 MHz až 868,6 MHz a pak by zařízení mělo vysílat max. 1% času, tj. max 36s během jedné hodiny. Úsek vyhlášky týkající se bezdrátového zvonku je uveden v tab. 1.

Tab. 1: Úsek vyhlášky pro bezdrátový přenos:

Označení	Kmitočtové pásmo	Vyzářený výkon	Kanálová rozteč	Klíčovací poměr 1)
f	433,050 – 434,790 MHz	10 mW e.r.p.	2)	< 10 %
f1	433,050 – 434,790 MHz	1 mW e.r.p.	2)	až 100 %
f2	433,050 – 434,790 MHz	10 mW e.r.p.	max. 25 kHz	až 100 %
g	868,000 – 868,600 MHz	25 mW e.r.p.	2)	< 1,0 %
h	868,700 – 869,200 MHz	25 mW e.r.p.	2)	< 0,1 %
i	869,300 – 869,400 MHz	25 mW e.r.p.	max. 25 kHz	-
j	869,400 – 869,650 MHz	500 mW e.r.p.	max. 25 kHz	< 10 %
k	869,700 – 870,000 MHz	5 mW e.r.p.	2)	až 100 %

- 1) Podíl času, kdy vysílač vysílá na nosném kmitočtu, v rámci jedné hodiny.
- 2) Pro přenos signálů může být použito celé uvedené kmitočtové pásmo.

Možnosti obvodu nRF9E5:

Tento obvod sdružuje na jednom čipu transceiver, mikrokontrolér, A/D převodník a D/A převodník s šířkovou modulací PWM. Kompletní přehled vlastností obvodu je v katalogu [1].

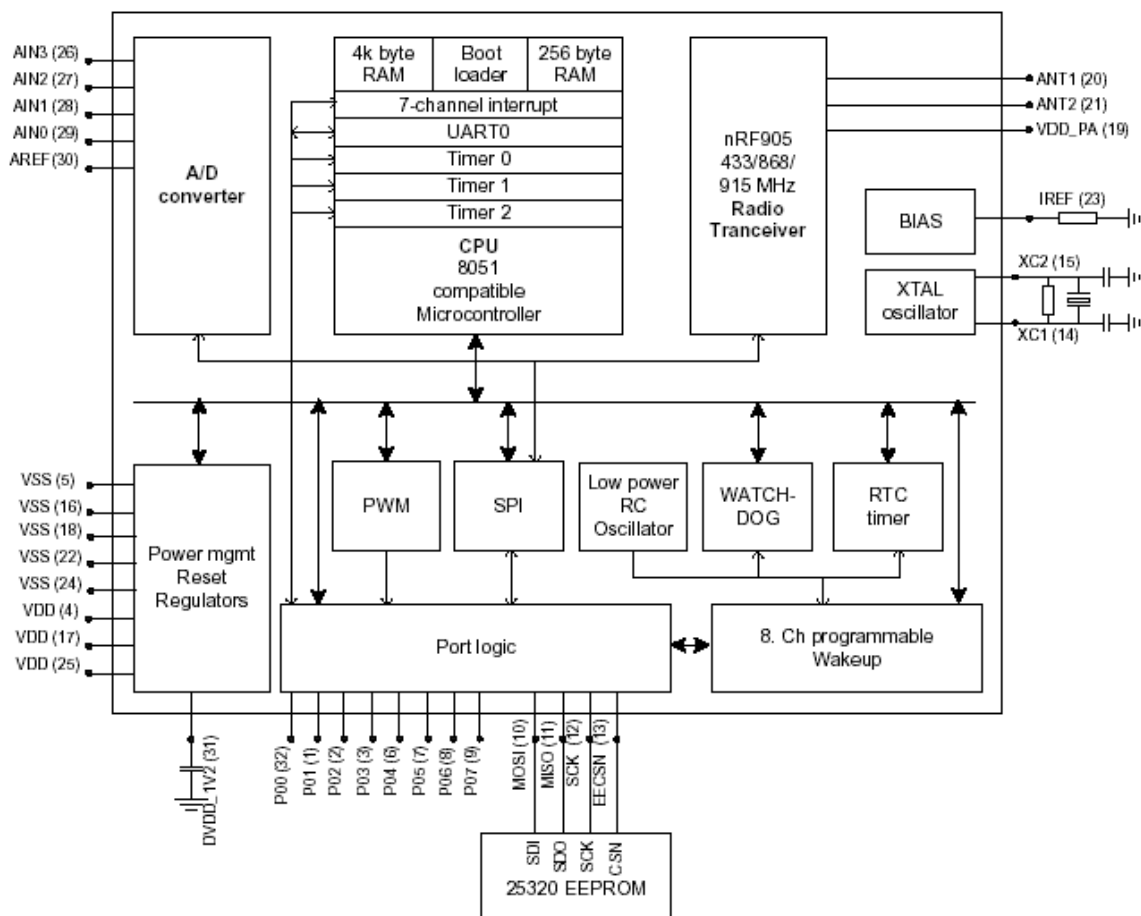
Přehled dostupných funkcí v obvodu nRF9E5:

- Transceiver na frekvenci 433/868/915 MHz
- Max. přenosová rychlost transceiveru 6 kS/s v režimu kontinuálního vysílání
- Mikrokontrolér slučitelný se standardem 8051
- 4-vstupový A/D převodník s rozlišením 10 bitů, vzorkovací frekvence 80 kS/s
- D/A převodník s šířkovou modulací výstupního signálu PWM, rozlišení 8 bitů
- Napájení 1,9 až 3,6 V
- Malé 32-pinové pouzdro QFN 5x5 mm
- Vnitřní hlídání úrovně napájecího napětí
- Odběr v pohotovostním režimu 2,5 μ A s probuzením pomocí časovače nebo vnějším napětím
- Nastavitelný vysílací výkon až do 10 dBm
- Čas přepnutí vysílacího kanálu menší než 650 μ s
- Nízký napájecí proud vysílače, typicky 9 mA při -10 dBm
- Nízký napájecí proud přijímače, typicky 12,5 mA ve špičce
- Nízký napájecí proud mikrokontroléru, typicky 1 mA při 4 MHz a 3 V
- Vhodný pro vysílání v režimu frequency hopping
- Detekce nosné pro indikaci volného vysílacího pásma

Jádro mikrokontroléru je instrukčně slučitelné se standardem 8051, časování instrukcí je však vylepšeno. Výkon jedné instrukce trvá 4 až 20 taktů hodin, což je méně proti 12 až 48 cyklům u standardu 8051. Taktovací frekvence se může pohybovat mezi 4 a 20 MHz při použití krystalového oscilátoru.

Funkce mikrokontroléru je dána ovládacím programem. Paměťový prostor je tvořen pouze pamětí typu RAM, ovládací program je tedy nutné mít ve vnější napěťově nezávislé paměti EEPROM typu 25C320. Začátek obsahu paměti má přesně definovanou strukturu a slouží pro nastavení frekvence rozhraní SPI a určení kmitočtu systémového oscilátoru. Mikrokontrolér s pamětí komunikuje pomocí sériové sběrnice SPI. K přenesení programu do vnitřní paměti po zapnutí napájení je určen vestavěný zaváděcí program.

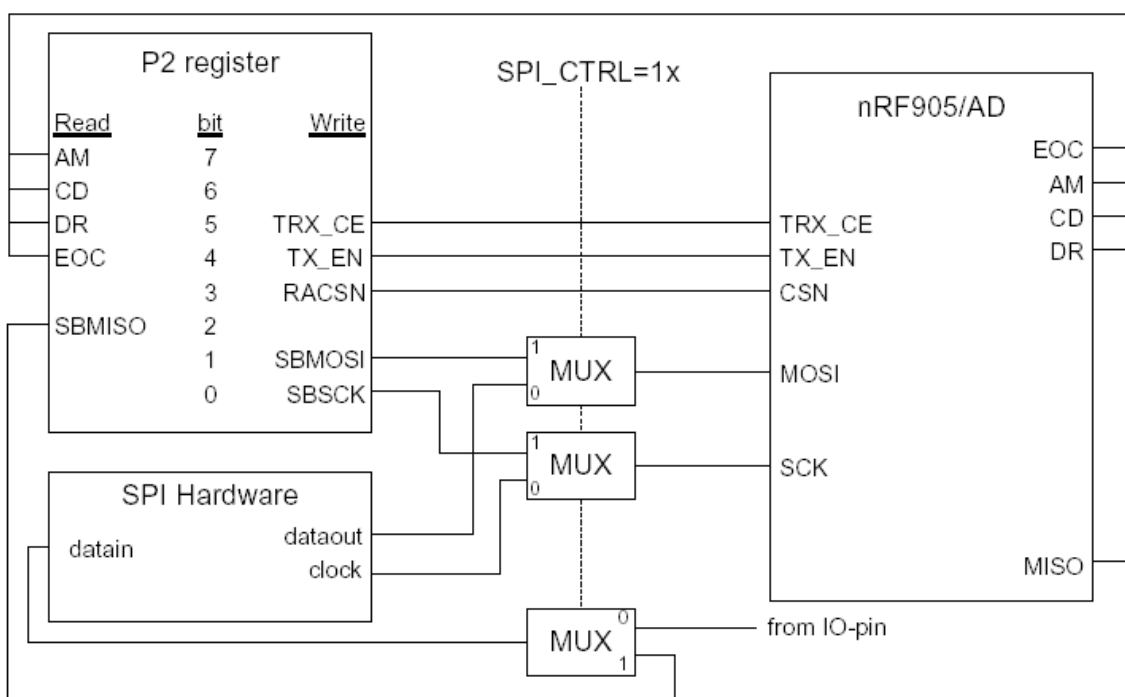
Blokové zapojení obvodu nRF9E5 je na obr. 2.



Obr. 2: Blokové schéma obvodu nRF9E5 (převzato z [1]).

SPI a konfigurační registry:

Jak je vidět z obr. 2, mikrokontrolér má přímý přístup ke všem hlavním blokům na čipu, pouze AD převodník a transceiver jsou odděleny a přístup k nim je prováděn pomocí vnitřního SPI rozhraní. Pro správné pochopení způsobu, kterým je transceiver a AD převodník ovládán, je nutné seznámit se s architekturou obvodu nRF9E5. Obr. 3 ukazuje propojení mikrokontroléru s transceiverem a A/D převodníkem.



Obr. 3: Připojení SPI rozhraní k transceiveru (převzato z [1]).

Pro ovládání sběrnice slouží registry P2 a SPI_CTRL. Registr SPI_CTRL určuje propojení SPI sběrnice mezi mikrokontrolérem, pamětí programu (připojena na P1) a subsystémem transceiveru s AD převodníkem (tab. 2).

Tab. 2: Funkce registru SPI_CTRL:

SPI_CTRL [1:0]	Funkce SPI rozhraní
00	SPI hardware není použit
01	SPI hardware je připojen k P1 (bootování programu z paměti EEPROM)
10	SPI hardware je připojen k transceiveru a AD převodníku
11	SPI rozhraní transceiveru je připojeno k bitům portu P2

Pokud je $\text{SPI_CTRL}[1:0] = 00$, SPI rozhraní je zakázáno. Stav 01 určuje, že je jednotka SPI spojena s externí pamětí programu EEPROM pomocí vývodů portu 1. Tento stav je typicky používán pouze během automatického načtení programu do vnitřní paměti RAM. Stav 10 je používán k běžné komunikaci jednotky SPI s řídicími registry transceiveru a AD převodníku, komunikace je řízena bity TRX_CE, TXEN a RACSN na portu P2. Poslední volbou je stav 11, při kterém je rozhraní transceiveru ovládáno pouze portu P2, včetně vývodů MISO, MOSI a SCK. Tato volba však nemá praktický význam z důvodu mnohem jednodušší komunikace pomocí jednotky SPI.

Port P2 není vyveden na vnější vývody pouzdra jako je tomu u standardu 8051, ale obsahuje bity pro řízení chodu transceiveru a také jeho stavové bity (tab. 3).

Tab. 3: Funkce portu P2:

Bit	P2 čtení	P2 zápis
7	Příznak příjmu platné adresy: Adress Match (AM)	Není použit
6	Příznak zachycení nosné: Carrier Detect (CD)	Není použit
5	Příznak přijatých dat: Data Ready (DR)	Spuštění transceiveru (TRX_CE)
4	Dokončení převodu AD převodníku (EOC)	Volba vysílání/příjem transceiveru (TXEN)
3	Není použit	Chip Select transceiveru a ADC (RACSN)
2	Výstup SPI transceiveru a ADC (MISO)	Není použit
1	Není použit	Vstup SPI transceiveru a ADC (MOSI)
0	Není použit	Hodiny SPI transceiveru a ADC (SCK)

SPI rozhraní může být aktivováno ($RACSN = 0$) pouze tehdy, pokud je transceiver klidovém režimu ($TRX_CE = 0$). Funkce transceiveru a AD převodníku se nastavuje konfiguračními registry. K jejich nastavování slouží SPI instrukce (tab. 4). Konfiguračních registrů je 7, z toho 3 registry jsou pouze pro čtení a další 4 je možno měnit. Obsah stavového registru je transceiverem vyslán na SPI sběrnici vždy během příjmu každé instrukce (první byte).

SPI registry určené pouze pro čtení:

- STATUS-REGISTER
- ADC-DATA-REGISTER
- RX-PAYLOAD

Programem měnitelné registry:

- ADC-CONFIGURATION-REGISTER
- RF-CONFIGURATION-REGISTER
- TX-ADDRESS
- TX-PAYLOAD

STATUS-REGISTER obsahuje příznaky Carrier Detect (CD), Address Match (AM), Data Ready (DR), ADC_End_Of_Conversion (EOC) a ADC_Ready_Flag (ADC_RFLAG). Příznaky CD, AM, DR a EOC můžeme číst také přímo z registru P2.

V registru ADC-CONFIGURATION-REGISTER se nastavuje pracovní režim A/D převodníku a ADC-DATA-REGISTER obsahuje výsledek A/D převodu. Obsah RF-CONFIGURATION-REGISTER řídí funkci transceiveru, adresa vysílače je dána obsahem registru TX-ADDRESS. Registry TX-PAYLOAD a RX-PAYLOAD obsahují přenášená data vysílače, resp. přijímače.

K datové komunikaci po rozhraní SPI slouží datový registr SPI_DATA. SPI instrukce se provádí zápisem hodnoty do datového registru SPI_DATA při aktivním rozhraní SPI ($RACSN = 0$). Každou novou instrukci musí předcházet sestupná hrana na signálu RACSN.

Pokud má po odeslání instrukce následovat čtení z transceiveru, je nutné spustit další přenos zápisem libovolného byte (např. 0) do registru SPI_DATA, počkat na přenesení dat a poté jsou čtená data dostupná v registru SPI_DATA. Pro každý další čtený byte je znovu nutné začít přenos zápisem do SPI_DATA.

Tab. 4: SPI instrukce:

Instrukce	Formát instrukce	Funkce
W_RF_CONFIG (WRC)	0000 AAAA	Zápis do konfiguračního registru. AAAA označuje, od kterého byte zápis začíná.
R_RF_CONFIG (RRC)	0001 AAAA	Zápis do konfiguračního registru. AAAA označuje, od kterého byte zápis začíná.
W_TX_PAYLOAD (WTP)	0010 0000	Zápis dat k vysílání: jeden až 32 byte (podle TX_PW). Zápis začíná vždy od byte 0.
R_TX_PAYLOAD (RTP)	0010 0001	Čtení dat k vysílání: jeden až 32 byte (podle TX_PW). Čtení začíná vždy od byte 0.
W_TX_ADDRESS (WTA)	0010 0010	Zápis adresy TX: jeden až 4 byte (podle TX_AFW). Zápis začíná vždy od byte 0.
R_TX_ADDRESS (RTA)	0010 0011	Čtení adresy TX: jeden až 4 byte (podle TX_AFW). Čtení začíná vždy od byte 0.
R_RX_PAYLOAD (RRP)	0010 0100	Čtení přijatých dat: jeden až 32 byte (podle RX_PW). Čtení začíná vždy od byte 0.
R_ADC_DATA (RAD)	0100 000A	Čtení dat z AD převodníku (jeden nebo oba byte). „A“ označuje, od kterého byte čtení začne.
W_ADC_CONFIG (WAC)	0100 0100	Zápis do konfiguračního registru ADC: jeden až 3 byte. Zápis začíná vždy od byte 0. Byte 2 není použit.
R_ADC_CONFIG (RAC)	0100 0110	Čtení z konfiguračního registru ADC: jeden až 3 byte. Čtení začíná vždy od byte 0. Byte 2 není použit.
CHANNEL_CONFIG (CC)	1000 pphc cccc cccc	Instrukce pro rychlé nastavení transceiveru (CH_NO = cccccccc, HFREQ_PLL = h, PA_PWR = pp).
START_ADC_CONV (SAV)	1100 ssss	Speciální instrukce pro spuštění AD převodu na vybraném kanálu (CHSEL = ssss).
STATUS REGISTER	není	Obsah stavového registru je vyslán na MISO po každém přechodu RACSN do nuly při zápisu instrukce do registru SPI_DATA.

Transceiver:

Mikrokontrolér obsahuje na čipu periferní obvod transceiveru, který je shodný s obvodem nRF905 od téhož výrobce Nordic Semiconductor. Ucelený přehled možností transceiveru nRF9E5 je tedy nutné hledat v katalogu obvodu nRF905 [2].

Nordic Semiconductor vyrábí podobný jednočipový systém mikrokontroléru s transceiverem pod označením nRF24E1. Tento obvod má vyšší přenosovou rychlost, která by umožnila přenášet audiosignál v lepší kvalitě a s větší šířkou přenášeného pásma. Obvod nRF24E1 však pracuje v mikrovlnném pásmu 2,4 GHz, které vykazuje mnohem větší útlum při průchodu zdmi budov. Protože předpokládáme, že vnitřní stanice bude od vnější oddělena minimálně jednou stěnou, je v konstrukci použit obvod nRF9E5, pracující v pásmu 433 MHz.

Transceiver nRF9E5 nedovoluje současné vysílání a příjem dat, proto mohou být data jednoduše přenášena pouze poloduplexně. Bitová rychlost vysílače je 50 kbit/s. Tato rychlost je snížena o přenášené řídicí informace a o čas potřebný ke změnám směru vysílání. Při použití k přenosu zvukové informace to znamená maximální přenášenou frekvenci přibližně 3 kHz. To je hodnota srovnatelná s kvalitou telefonní linky. Vysílač používá modulaci GFSK a kódování Manchester. Zabraná šířka pásma vysílače je 100 kHz v pásmu 433 MHz a 200 kHz v oblasti 868 a 915 MHz. Vyhláška ČTÚ [3] stanovuje maximální klíčovací poměr vysílače na 10 % v pásmu 433 MHz a 1 % pro 868 a 915 MHz. Pokud však budeme vysílat pouze s výkonem 1 mW v pásmu 433 Mhz (nebo 5 mW v pásmu 868 MHz), není doba vysílání omezena.

Pro řízení vysílání a příjmu transceiveru slouží bity TRX_CE a TX_EN. Jejich funkci ukazuje tab. 5.

Tab. 5: Pracovní režimy transceiveru:

TRX_CE	TX_EN	Funkce SPI rozhraní
0	X	Režim Standby a přenos po SPI
1	0	Transceiver aktivován – probíhá příjem
1	1	Transceiver aktivován – probíhá vysílání

Po spuštění transceiveru signály TX_EN a TRX_CE se nejprve provede jeho inicializace. Tato doba činí podle katalogu 650 μ s. Teprve po této době se data začnou odesílat nebo začne příjem. V režimu vysílání je zapotřebí držet TRX_CE = 1 a TX_EN = 1 minimálně 10 μ s, poté je možno signály změnit, aniž by došlo k přerušení vysílání. Pokud po této době nastavíme TRX_CE = 0, dokončí se odeslání paketu a transceiver poté přejde do úsporného režimu. Pokud je nastaven TRX_CE = 0 v režimu příjmu, přeruší se přijímání okamžitě a transceiver přejde do úsporného režimu.

Přenášená sekvence se skládá z několika částí (viz obr. 4). Nejprve je vysílána úvodní synchronizační posloupnost, která má pevnou délku 10 bitů a je automaticky generována řídicími obvody transceiveru. Následuje vyslání adresy, její délka je 1 až 4 byte (8 až 32 bitů). Dále jsou vysílána uživatelská data s délkou 1 až 32 byte (8 až 256 bitů). Sekvence může být ukončena 0 až 2 byte automaticky generovaného kontrolního kódu CRC.

Vysílač umožňuje opakované vysílání datových paketů. V tomto režimu je po vyslání posledního byte zařazena synchronizační posloupnost následujícího vysílaného paketu. Data jsou vysílána tak dlouho, dokud je signál TRX_CE držen v nízké úrovni.

Uživatelská data pro vysílání se zapisují do registru TX-PAYLOAD, přijatá data se ukládají do registru RX-PAYLOAD. Adresa vysílače a přijímače se nastavují zvlášť, pro přijímač v registru RF-CONFIGURATION-REGISTER a adresa vysílače je uložena v registru TX-ADDRESS.

Synchronizace 10 bitů	Adresa vysílače 1 až 4 Byte	Přenášená uživatelská data 1 až 32 Byte	Kód CRC 0 až 2 Byte
--------------------------	--------------------------------	--	------------------------

Obr. 4: Formát přenášeného datového paketu

A/D převodník:

A/D převodník má rozlišení 6, 8, 10 nebo 12 bitů. V aplikaci bezdrátového telefonu je použito rozlišení 8 bitů z důvodu kompromisu mezi kvalitou přenášeného zvuku a možnostmi nízké přenosové rychlosti transceiveru. Vzorkovací rychlost je programově nastavitelná až do 80 kS/s. Převod může být spouštěn automaticky po dokončení předchozího převodu, pro účely snímání zvuku v bezdrátovém zvonku je použit režim jednoduchých převodů a spouštění převodu je řízeno pomocí přerušení od časovače.

Výsledek AD převodu se nachází v registru ADC-DATA-REGISTER. Nastavení funkcí AD převodníku se provádí SPI zápisem požadované hodnoty do registru ADC-CONFIGURATION-REGISTER (tab. 6).

Tab. 6: ADC-CONFIGURATION-REGISTER – registr pro nastavení AD převodníku :

Číslo byte	Bitový obsah [7:0]
0	Chsel [7:4], VFSEL, PWR_UP, ADCRUN, CSTARTN
1	Reserved [7:4], ADC_RL_JUST, DIFFMODE, RESCTRL[1:0]
2	Reserved [7:0]

Převodník má vstup multiplexovaný mezi 4 vývody (AIN0 – AIN3, tab. 7), napětí se může snímat vždy pouze z jednoho. Převodník umožňuje měřit velikost napájecího napětí mikrokontroléru, pak je jako reference automaticky připojena vnitřní bandgap reference 1,22 V. V tomto případě je aktuální napájecí napětí určeno vztahem:

$$V_{dd} = 3,66 \cdot \frac{ADCDATA}{2^N}$$

N označuje nastavené rozlišení převodníku v bitech (podle obsahu RESCTRL [1:0]).

Tab. 7: ADC-CONFIGURATION-REGISTER, Chsel [7:4] – volba vstupu AD převodníku :

Chsel [7:4]	Vstup AD převodníku
0000	AIN0
0001	AIN1
0010	AIN2
0011	AIN3
1xxx	Měření Vdd/3

Referenční napětí převodníku 0,8 až 1,5 V se přivádí na vstup AREF nebo je možné využít vnitřní stabilizované referenční napětí 1,22 V. Volba reference se provádí bitem VFSEL, pokud je nulový nebo když je měřeno napájecí napětí, je použita vnitřní reference 1,22 V, jinak je použito vnější referenční napětí.

Pokud je bit PWR_UP nastaven, jsou aktivovány obvody AD převodníku. Odběr AD převodníku je přibližně 1 mA, před přechodem do úsporného režimu je tedy nutné převodník deaktivovat vynulováním bitu PWR_UP.

Nastavením bitu ADCRUN přejde převodník do režimu automatického spouštění po dokončení převodu. V tomto případě stačí ke spuštění první náběžná hrana bitu CSTARTN. Ke spuštění následujícího jednoduchého převodu se používá další náběžná hrana bitu CSTARTN nebo lze kdykoliv použít speciální SPI instrukci SAC.

D/A převodník PWM:

Vývod P0.7 má jako alternativní funkci výstup vestavěného D/A převodníku s šířkovou modulací výstupního signálu PWM. Přesnost převodníku může být programově nastavena na 6, 7 nebo 8 bitů (tab. 8). Kmitočet signálu PWM je odvozen ze systémového krystalového oscilátoru 6-bitovou předděličkou (tab. 9). Při frekvenci krystalového oscilátoru 8 MHz a rozlišení 8 bitů je maximální výstupní frekvence 30 kHz. Střída signálu je nastavitelná od 0 do 100 %. Pro správnou funkci přenosu zvukové informace musí být výstupní signál filtrován dolní propustí s dělicím kmitočtem na polovině kmitočtu přenosu vstupních vzorků.

Tab. 8 Nastavení rozlišení PWM pomocí registru PWMCON :

PWMCON [7:6]	Funkce PWM
00	Zastavení PWM modulu
01	Rozlišení 6 bitů
10	Rozlišení 7 bitů
11	Rozlišení 8 bitů

Bity PWMCON [5:0] tvoří předděličku, která nastavuje frekvenci PWM. Střída výstupního signálu PWM je určena obsahem registru PWMDUTY. Nastavení frekvence pro různá rozlišení PWM a nastavení střidy signálu PWM ukazuje tab. 2.

Tab. 9: Nastavení frekvence PWM pomocí registru PWMCON :

PWMCON [5:0]	Frekvence PWM	Střída PWM signálu
00	0 (PWM neaktivní)	0
01	$f_{xo} \cdot \frac{1}{63 \cdot (PWMCON[5:0] + 1)}$	$\frac{PWMDUTY[5:0]}{63}$
10	$f_{xo} \cdot \frac{1}{127 \cdot (PWMCON[5:0] + 1)}$	$\frac{PWMDUTY[5:0]}{127}$
11	$f_{xo} \cdot \frac{1}{255 \cdot (PWMCON[5:0] + 1)}$	$\frac{PWMDUTY[5:0]}{255}$

Obecné I/O porty:

Mikrokontrolér nRF9E5 má 2 vstupně-výstupní porty pro obecné použití. Směr každého bitu je zvlášť nastavitelný příslušným bitem `_DIR`. Některé bity mají alternativní funkce jednotlivě spouštěné aktivní úrovní příslušného bitu `_ALT`. Funkce vývodů jsou zobrazeny v tab. 10.

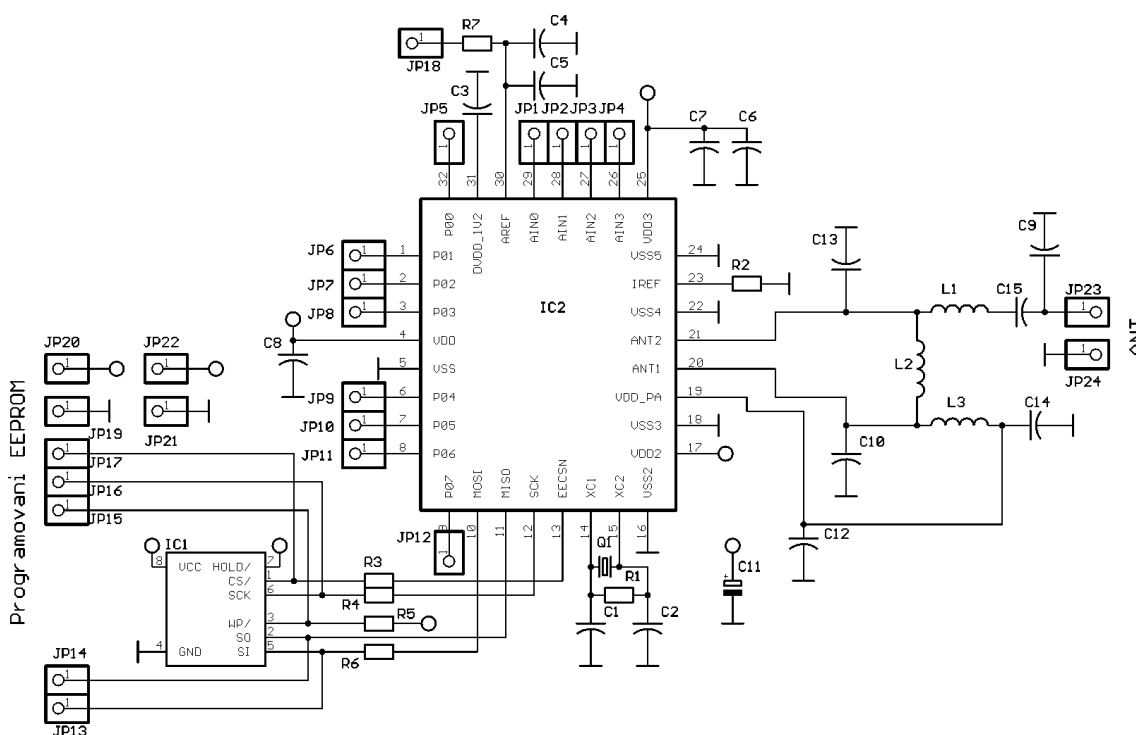
Tab. 10: Funkce I/O portů mikrokontroléru nRF9E5:

Číslo vývodu	Název	Výchozí funkce	<code>_ALT = 1</code>	<code>SPI_CTRL != 01</code>
13	EECSN	P1.3		P1.3
11	MISO	SPI.datain		P1.2
12	SCK	SPI.clock	T2 (timer2 output)	P1.1
10	MOSI	SPI.dataout		P1.0
32	P00	P0.0	GTIMER	
1	P01	P0.1	RXD (UART)	
2	P02	P0.2	TXD (UART)	
3	P03	P0.3	INT0_N (interrupt)	
6	P04	P0.4	INT1_N (interrupt)	
7	P05	P0.5	T0 (timer1 interrupt)	
8	P06	P0.6	T1 (timer1 interrupt)	
9	P07	P0.7	PWM	

Vývod P00 je v bezdrátovém zvonku použit pro ovládání napájecího napětí analogového systému, P01 řídí spínací tranzistor dveřního zámku, vývodem P02 se volí vnější nebo vnitřní stanice, na P03 a P04 jsou připojena ovládací tlačítka, P05 a P06 ovládají LED indikaci a P07 je použit jako výstup audiosignálu v PWM tvaru.

Zkušební moduly transceiveru nRF9E5:

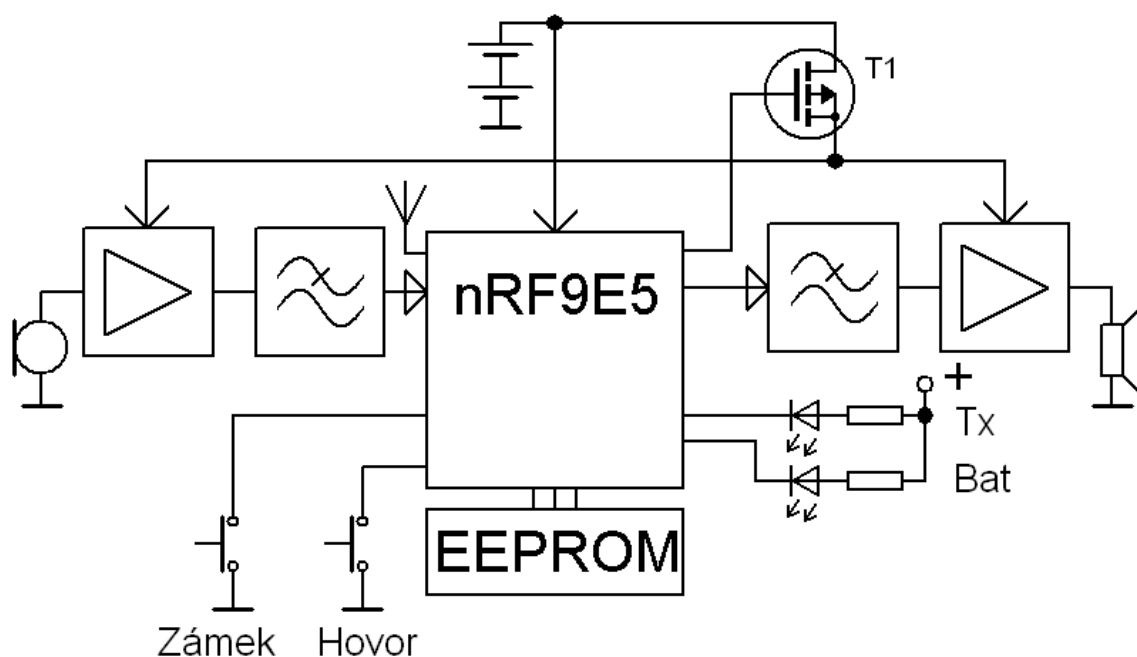
Mikrokontrolér nRF9E5 je dodáván v malém pouzdře QFN. Na ploše 5 x 5 mm je rozmístěno po čtvercovém obvodu 32 vývodů. Protože není jednoduché takto malé pouzdro zapájet do cílové aplikace, budou při vývoji použity zkušební moduly [5], které velmi usnadňují návrh DPS výsledného systému. Schéma modulu je na obr. 5. Modul sdružuje obvod nRF9E5, paměť programu typu EEPROM, hodinový krystal, filtrační kondenzátory v napájecí cestě, přizpůsobovací obvody pro připojení antény a vývody pro připojení do cílové aplikace. Modul je realizován na oboustranné desce plošných spojů s rozměry 25x30mm, podklady pro jeho zhotovení jsou v příloze B.



Obr. 5: Zapojení modulu transceiveru s obvodem nRF9E5 [5]

Popis vnitřní stanice:

Blokové zapojení vysílače je ukázáno na obr. 6. Základem zapojení je mikrokontrolér nRF9E5. Signál z elektretového mikrofону je zesílen pomocí předzesilovače a pomocí vnitřního A/D převodníku převeden do digitální podoby. Získaná data jsou transceiverem posílána přijímači. Zpětně získaná hlasová data jsou pomocí D/A převodníku převedena na analogový signál a ten je po zesílení přiveden na reproduktor. Pro řízení funkce jsou k mikrokontroléru připojena tlačítka Hovor a Zámek. Napájecí napětí je vedeno přímo na mikrokontrolér, analogová část však má napájení vypínáno, aby se zvýšila životnost baterií. Vnitřní stanice by mohla být upravena pro napájení ze síťového adaptéru, tím bychom eliminovali použití baterií, ve vnější stanici se však použití baterií nevyhne.

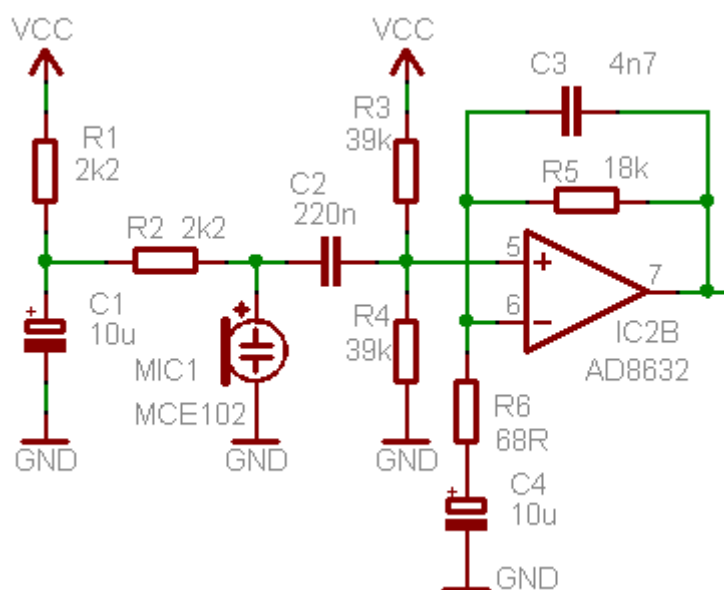


Obr. 6: Blokové zapojení vnitřní stanice

Následující popis přibližuje funkci a zapojení jednotlivých bloků bezdrátového zvonku. Kompletní schéma obvodu je v příloze A.

Mikrofonní předzesilovač:

Konstrukce mikrofonního předzesilovače pro obvod příbuzný s nRF9E5 je ukázána v [4]. Mikrofonní předzesilovač použitý v bezdrátovém zvonku je z tohoto obvodu odvozen, z důvodu zjednodušení konstrukce však byl vynechán stupeň s proměnným zesílením. Schéma mikrofonního předzesilovače ukazuje obr. 7.



Obr. 7: Mikrofonní předzesilovač

Přenášený hlas je snímán elektretovým mikrofonom. Elektretový mikrofón je speciální typ kondenzátorového mikrofónu, který má ve snímacím kondenzátoru elektret (dielektrikum s trvalou polarizací). Díky tomu není nutné přivádět k mikrofónu externí polarizační napětí, které je například u kondenzátorových mikrofónů používaných ve studiové audiotechnice 48 V. Běžně používané elektretové mikrofóny mají vestavěný zesilovač s tranzistorem FET. Mají však pouze 2 vývody, a proto je zesilovač napájen proudem, který z napájecího napětí systému odvodí zatěžovací rezistor R2.

Sériové spojení mikrofону a zatěžovacího rezistoru se pak chová jako dělič napětí, na jehož výstupní napětí je superponován signál z mikrofonu. Signál je potom veden přes oddělovací kondenzátor C2 na operační zesilovač IC2B. Zesílení tohoto stupně v propustném pásmu je dáno poměrem $R5/R6$. Pro hodnoty $R5 = 18k$ a $R6 = 68R$ je výsledné zesílení přibližně 48 dB. $R6$ a $C4$ tvoří ve zpětné vazbě operačního zesilovače horní propust, jejíž mezní kmitočet určuje nejnižší přenášený kmitočet hlasového signálu. Horní propust je prvního řádu, pokles pod mezním kmitočtem je tedy 20 dB/dek. Při hodnotách $R6 = 68R$ a $C4 = 10\mu$ je mezní kmitočet:

$$f_d = \frac{1}{2 \cdot \pi \cdot R6 \cdot C4} = \frac{1}{2 \cdot \pi \cdot 68 \cdot 10 \mu} = 230 \text{ Hz}$$

Mikrofonní předzesilovač je částí systému nejchoulostivější na rušení, proto je napájecí napětí elektretového mikrofonu filtrováno RC článkem tvořeným $R1$ a $C1$.

Rezistory $R3$ a $R4$ nastavují stejnosměrnou úroveň v následujících stupních. Úroveň je volena tak, aby v klidovém stavu byla na vstupu AD převodníku přibližně polovina referenčního napětí, které je dáno napájecím napětím analogové části. Pokud přijde z mikrofonu hlasový signál, bude vstupní napětí převodníku vychylováno směrem k úrovni referenčního napětí a směrem dolů k nulovému potenciálu.

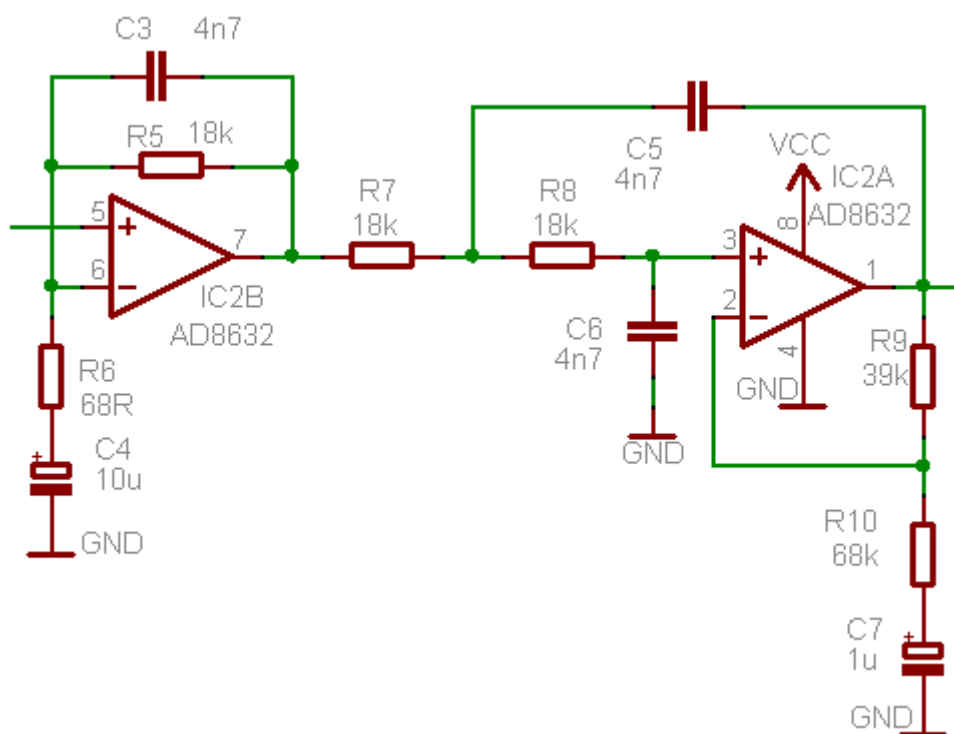
IC2B také patří k antialiasingovému filtru, o němž je psáno níže.

Antialiasingový filtr

Součástí předzesilovače je antialiasingový filtr, který před vstupem signálu do A/D převodníku odstraní složky signálu nad polovinou vzorkovacího kmitočtu.

Protože je hlasový signál z mikrofону před odesláním digitalizován, je nutné před vstupem signálu do A/D převodníku jeho kmitočtové omezení na polovinu vzorkovacího kmitočtu. Jak bude ukázáno v kapitole „Vzorkovací kmitočet audiosignálu a přenosová rychlost“, mezní kmitočet audiosignálu je max. 2100 Hz. Protože mají použité filtry konečnou strmost, musí být mezní kmitočet dolní propusti ještě nižší, aby byly dostatečně potlačeny kmitočty nad polovinou kmitočtu vzorkovacího. Z řady E12 byly vybrány hodnoty rezistoru 18k a kondenzátoru 4n7, s nimiž je mezní kmitočet filtru:

$$f = \frac{1}{2 \cdot \pi \cdot R \cdot C} = \frac{1}{2 \cdot \pi \cdot 18 \cdot 10^3 \cdot 4,7 \cdot 10^{-9}} = 1880 \text{ Hz}$$



Obr. 8: Antialiasingový filtr

Výstup antialiasingového filtru je veden přímo na AD převodník, proto je použita aktivní dolní propust s operačním zesilovačem a signál je nejprve zesílen zesilovačem IC2B.

Vlastní antialiasingový filtr je třetího řádu (nad dělicím kmitočtem pokles 60 dB na dekádu) a tvoří jej operační zesilovače IC2B a IC2A, rezistory R7 a R8 a kondenzátory C5 a C6. Filtrační systém s IC2A je filtr druhého řádu s Butterworthovou aproximací (podle [6]). IC2B přidává k celkovému filtru ještě jeden řád, díky R5 a C3 ve zpětné vazbě.

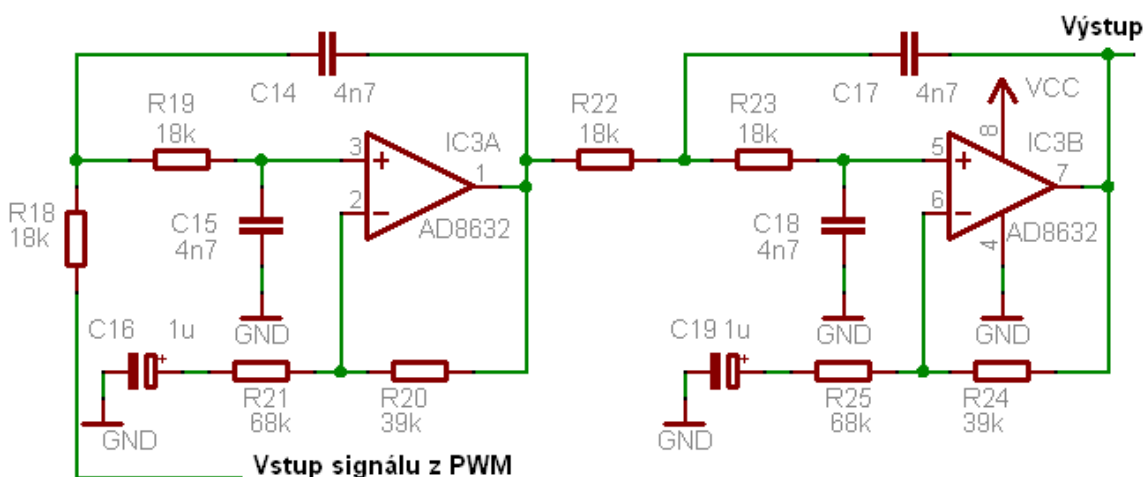
Celý systém bezdrátového zvonku je napájen napětím 3 V, proto jsou v zapojení použity operační zesilovače určené pro práci s nízkým napájecím napětím. Použité OZ AD8632 (Analog Devices) mohou pracovat od 1,8 V. S nízkým napájecím napětím souvisí i schopnost operačního zesilovače pracovat v režimu Rail-to-Rail. Speciální vnitřní návrh obvodu dovoluje, aby se rozkmit vstupního a výstupního napětí zesilovače blížil až k úrovním kladného a záporného napájecího napětí OZ. To je důležité zejména pro maximální využití dynamického rozsahu A/D převodníku.

Napájecí napětí je nesymetrické, proto nemohou být střídavé signály stejnosměrně vztaženy k nulovému potenciálu. Stejnosměrná úroveň je posunuta elektrolytickými kondenzátory C4 a C7, napětí na nich i na výstupu OZ se ustálí přibližně na polovině rozdílu napájecího napětí OZ. C4 a R6 tvoří horní propust s mezním kmitočtem 230 Hz, tím je tedy přenosové pásmo filtru omezeno zdola proti rušivým vlivům na nízkých kmitočtech (poryvy větru, nárazy vzduchu při velmi blízkém hovoru do mikrofonu, atd.).

Antiimaging filtr

Výstup audiosignálu z mikrokontroléru je ve formě obdélníkového signálu s pulzně šířkovou modulací, pro získání původního hlasového signálu je tedy nutné odfiltrovat nosnou frekvenci signálu PWM a také spektrální složky původního signálu zrcadlené kolem kmitočtu obnovování PWM.

Rekonstrukční filtr je tvořen Butterworthovou dolní propustí druhého řádu s dělicím kmitočtem 2 kHz, stejně jako antialiasingový filtr. Pro důkladné potlačení zrcadlových spektrálních složek jsou zapojeny dva filtry v kaskádě (celkový pokles 80 dB na dekádu). Stejnoseměrný posuv na polovinu napájecího napětí je podobně jako v antialiasingovém filtru proveden kondenzátory C16 a C19. RC články R21, C16 a R25, C19 tvoří horní propust s mezním kmitočtem 2 Hz. Tento kmitočet je mimo zvukovou oblast, C16 a C19 tedy nemají vliv na přenos hlasového signálu.



Obr. 9: Antiimaging filtr

Výstup PWM má rozkmit 0 V až V_{dd}, proto musí být jeho úroveň snížena pomocí odporového děliče. Každý ze dvou bloků dolní propusti druhého řádu má nastavené napěťové zesílení 4 dB a zesilovač reproduktoru TDA7050 má v můstkovém režimu zesílení 32 dB. Celkové zesílení je tedy 40 dB a dělič napětí musí vstupní signál snížit o 40 dB, tedy 100x zeslabit. Protože TDA7050 vyžaduje vstupní signál vztažený k nulovému potenciálu, bude odporový dělič i za antiimaging filtrem. Dělič R28 (1k) a R29 (68R) potlačuje o 24 dB, zeslabení děliče R26 (18k) a R27 (2k2) je 19 dB.

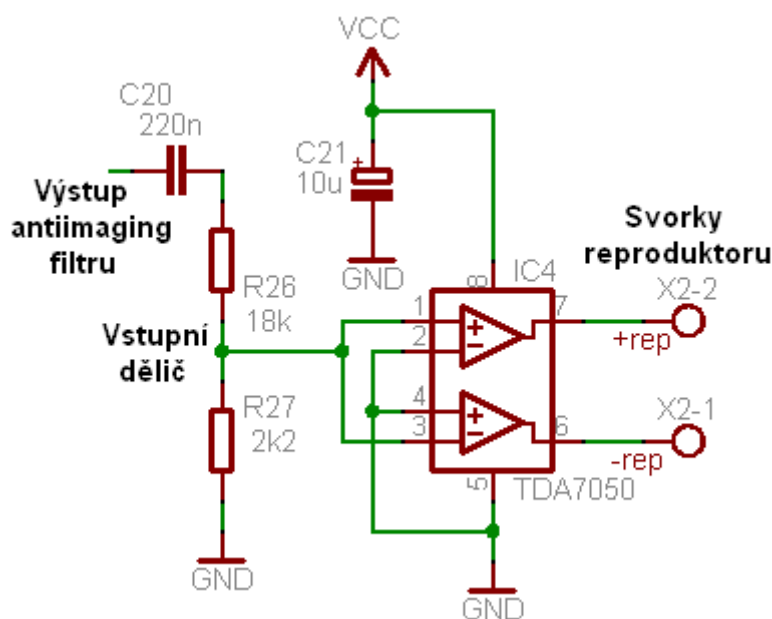
Zesilovač pro reproduktor

Zesilovač je obvodová součást bezdrátového zvonku nejnáročnější na příkon. Protože se předpokládá napájení z baterií a obvod nFR9E5 je konstruován pro napájecí napětí max. 3,6 V, je systém napájen ze dvou článků napětím 3 V.

Výstupní zesilovač musí být schopen dodat dostatečný výkon do reproduktoru i při použitém nízkém napájecím napětí 3 V. Tuto podmínku splňuje obvod TDA7050. Výhodou obvodu je jeho integrita, protože nepotřebuje žádné další vnější součástky, a jeho schopnost pracovat od napájecího napětí 1,6 V. Jedná se o stereofonní zesilovač, ale v případě použití jediného kanálu může být reproduktor připojen v můstkovém zapojení a tím se zvýší efektivita přenosu výkonu do reproduktoru. Protože se snažíme dosáhnout co největší hlasitosti reproduktoru, použijeme právě můstkový režim.

Aby byl zvuk vycházející z reproduktoru dostatečně hlasitý a aby přitom odběr zesilovače nebyl příliš velký, je nutné použít reproduktor s větší citlivostí. Z toho však vyplývají větší rozměry reproduktoru, který bude svými rozměry určovat velikost výsledné mechanické konstrukce.

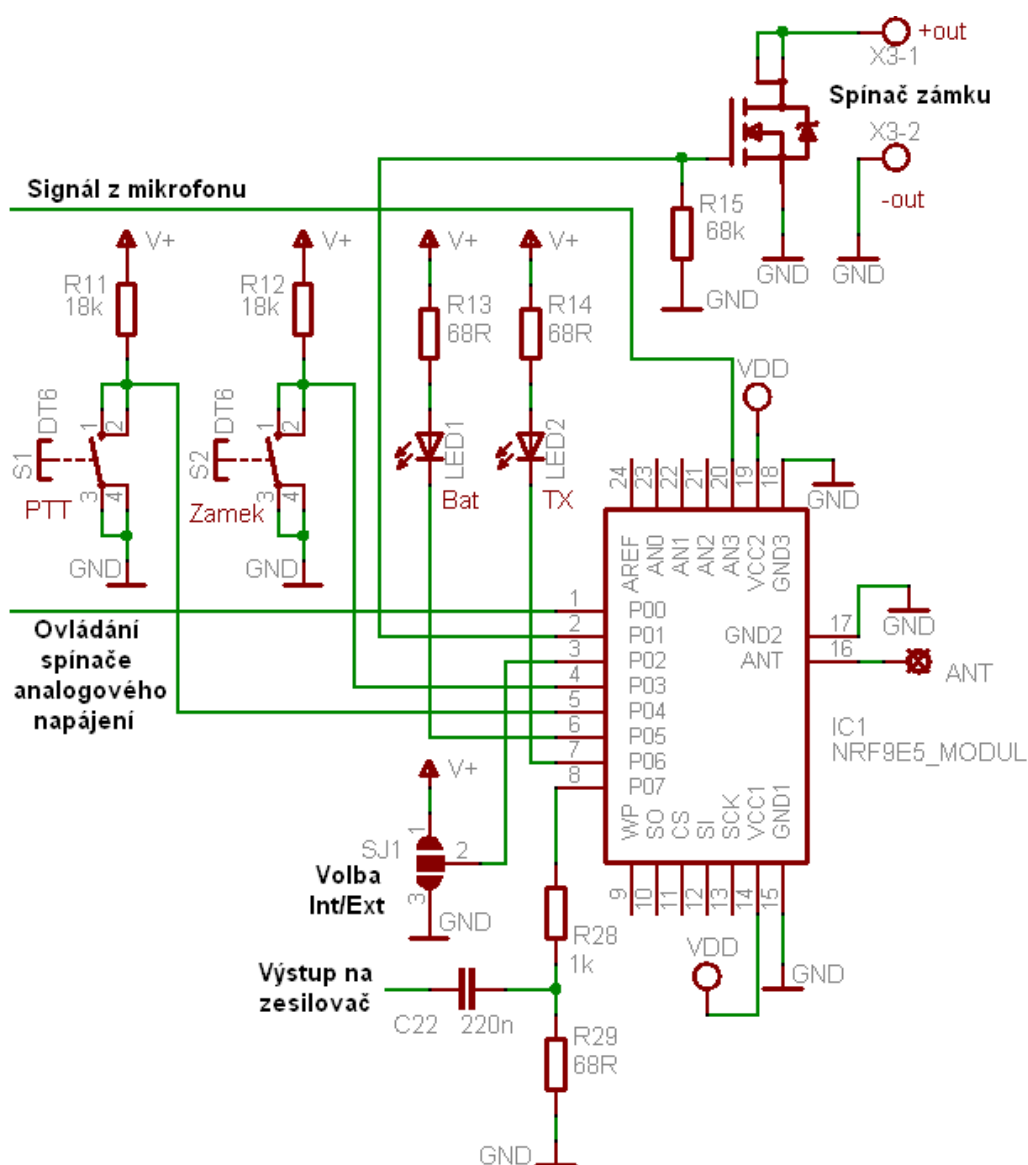
Vstupní napětí zesilovače TDA7050 je vztaženo vůči zemi, proto musí být od antiimaging filtru oddělen vazebním kondenzátorem C22.



Obr. 10: Zesilovač reproduktoru

Řídící mikrokontrolér vnitřní stanice:

Jádrem celého systému je obvod nRF9E5. Obr. 11 znázorňuje jeho připojení k ostatním částem systému. Na vstup AN3 je přiváděn zesílený signál z mikrofonu, zvukový výstup ve formě PWM je odváděn z děliče na výstupu P07. Na vývodu P00 je ovládací signál pro spínač analogového napájecího napětí, sepnutí se provede přivedením log. 0. Na P01 je výstup spínače dveřního zámku, signál je aktivní v log. 1.



Obr. 11: Zapojení systému kolem mikrokontroléru

Protože jsou vnitřní i vnější stanice realizovány na jednotné DPS, pájecím přepínačem SJ1 se na vstup P02 přivede log. 0 nebo log. 1, a tím se rozhodne, zda má být systém konfigurován jako vnitřní nebo vnější stanice. Na vstup P03 a P04 jsou připojena tlačítka Zámek a PTT (hovor). Vstupy nemají funkci pull-up, proto je realizována externími rezistory R11 a R12. Na výstupy P05 a P06 jsou připojeny svítivé diody Bat a TX, které indikují vybitou baterii ve vnější stanici a aktivovaný vysílač.

Anténa je realizována jako čtvrtvlnný monopól tenkým izolovaným měděným lankem. Vysílání probíhá na kmitočtu 433 MHz, délka antény je tedy:

$$\frac{\lambda}{4} = \frac{1}{4} \cdot \frac{c}{f} = \frac{1}{4} \cdot \frac{3 \cdot 10^8}{433 \cdot 10^6} = 0,17 \text{ m} = 17 \text{ cm}$$

Transceiver používá z důvodu redukce spotřeby nejnižší vysílací výkon 100 μW . S tímto výkonem byl naměřen dosah na volném prostranství 50 m, stanice umístěné v oddělených místnostech udržovaly spojení i při oddělení třemi cihlovými zdmi tloušťky 15 cm.

V případě potřeby většího dosahu je možné výkon zvýšit, dostupné jsou výkony -10, -2, 6 a 10 dBm, je však potřeba počítat s vyšším oděrem proudu z baterií a větším rušením v přenášeném hlasu.

Každá stanice odebírá z baterií přibližně 30 mA, pokud je přenášen hlasitý zvuk, stoupá odběr až na 80 mA. Vnější stanice odebírá v úsporném režimu 3 μA .

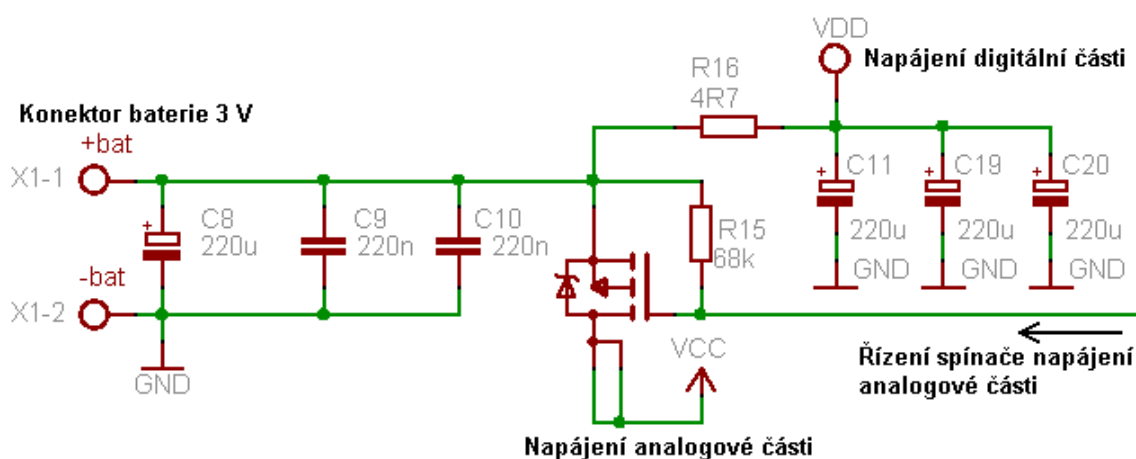
Rozvod napájení

K napájení systému je využito sériové spojení dvou tužkových článků, výsledné napájecí napětí je tedy 3 V. Řešení napájecího obvodu je na obr. 12.

Při bateriovém napájení je rozhodujícím faktorem životnost baterií, obvod je tedy postaven tak, aby v klidovém režimu odebíral minimální proud. Analogová část systému, zahrnující mikrofonní předzesilovač s aktivním filtrem a zesilovač mikrofonu s filtrem, má napájecí napětí ovládáno tranzistorem PMOS. Tento tranzistor je ovládán mikrokontrolérem tak, aby byla analogová část napájena pouze během přenosu hlasového signálu. Během úsporného režimu systému tedy analogová část neodebírá z baterií proud.

Digitální část zahrnuje mikrokontrolér s transceiverem a indikační diody LED. Při vysílání transceiveru vznikají v napájecím rozvodu špičky odběru, napájení digitální části je tedy odděleno od analogových obvodů pomocí rezistoru R16 a filtrováno kondenzátory C11, C19 a C20. Pro dostatečnou účinnost filtrace musí mít tyto kondenzátory nízký sériový odpor (ESR).

Hodnota rezistoru R16 je volena kompromisem tak, aby na něm nevznikal zbytečně velký úbytek napětí, ale aby potlačoval přenos výkyvů napětí z části digitální do analogové. Pokud budeme uvažovat odběr digitální části v aktivním režimu přibližně 20 mA, je úbytek na rezistoru o hodnotě 4R7 necelých 100 mV.



Obr. 12: Zapojení napájecí části systému

Vzorkovací kmitočet audiosignálu a přenosová rychlost

Bezdrátový zvonek bude bezdrátově přenášet hlasové informace. Pro srozumitelný přenos je zapotřebí přenášet kmitočtové složky přibližně do 3 kHz. Původní analogová telefonie pracovala s šířkou pásma 3 kHz, tomu při kvantování 8 bitů na vzorek a vzorkovací frekvenci 8 kHz odpovídá datový tok 64 kbps. Transceiver nRF9E5 však má maximální přenosovou rychlost 50 kbps a ta je dále snížena o přenášené řídicí informace. Čas potřebný pro odeslání jednoho bloku dat při kontinuálním vysílání je udán rovnicí:

$$TOA = T_{\text{preamble}} + \frac{N_{\text{address}} + N_{\text{payload}}}{BR}$$

TOA značí čas přenosu bloku, T_{preamble} je čas potřebný k odesílání synchronizační hlavičky bloku (její trvání je 0,2 ms), N_{address} je počet bitů adresy (8 až 32), N_{payload} je počet odesílaných bitů (8 až 256) a BR je bitová rychlost přenosu transceiveru (50 kbps). Při kontinuálním vysílání v blocích 256 bitů (32 bajtů) s nejkratší adresou (8 bitů) je tedy čas přenosu jednoho bloku:

$$TOA = 0,2 \cdot 10^{-3} + \frac{8 + 256}{50 \cdot 10^3} = 5,48 \cdot 10^{-3} \text{ s}$$

Z doby přenosu vypočítáme maximální přenosovou rychlost kontinuálního vysílání:

$$R = \frac{N_{\text{payload}}}{TOA} = \frac{256}{5,48 \cdot 10^{-3}} = 48485 \text{ bps} = \frac{48485}{8} \text{ Bps} = 6061 \text{ Bps}$$

Toto je maximální využitelná přenosová rychlost transceiveru, při použití opakovaného vysílání stejného paketu, nejkratší adresy, bez přenosu CRC a s využitím všech 32 byte v paketu. Pro zabezpečení správnosti přijatých dat je vysílán i 8-bitový kód CRC (automaticky generovaný obvody transceiveru). Protože směr přenosu řídí vnitřní stanice, po každém odeslaném paketu se musí vnější stanice přepnout do režimu příjmu a zkontrolovat, zda může pokračovat ve vysílání. Musíme tedy ke každému paketu přičíst zpoždění při přepínání do režimu příjmu a zpět do režimu vysílání ($2 \times 550 \mu\text{s}$).

Výsledný čas odeslání paketu a přenosová rychlost tedy jsou:

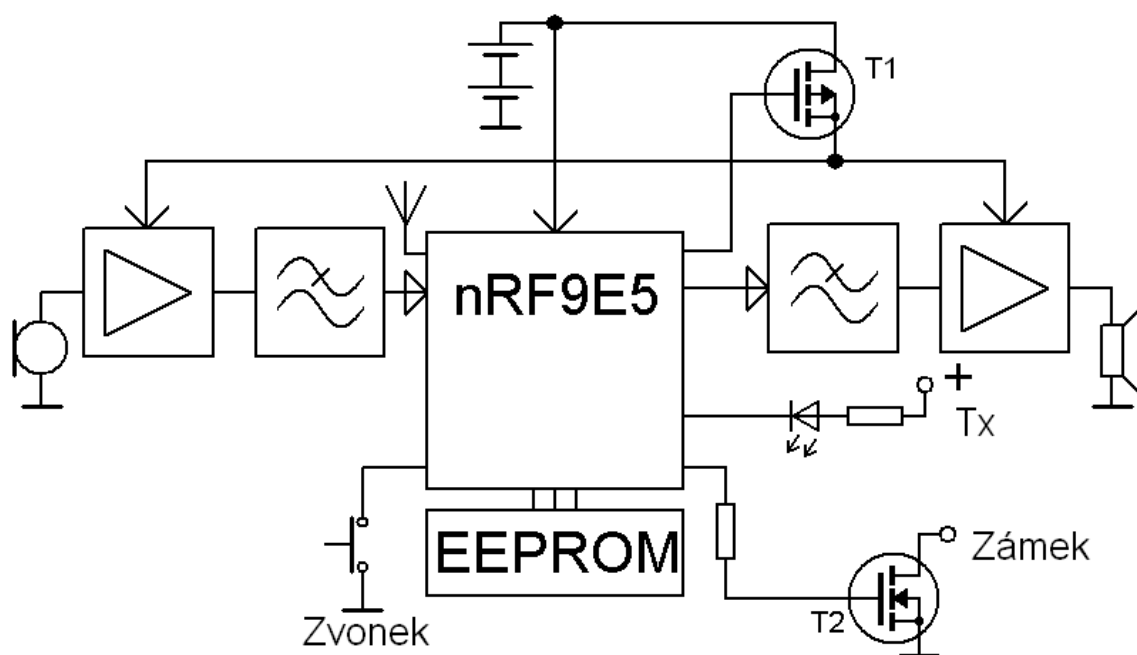
$$TOA = T_{TX > RX} + T_{RX > TX} + T_{preamble} + \frac{N_{address} + N_{payload} + N_{CRC}}{BR}$$
$$TOA = (0,55 + 0,55 + 0,2) \cdot 10^{-3} + \frac{8 + 256 + 8}{50 \cdot 10^3} = 6,74 \cdot 10^{-3} \text{ s}$$
$$R = \frac{N_{payload}}{TOA} = \frac{31 \cdot 8}{6,74 \cdot 10^{-3}} = 36795 \text{ bps} = \frac{36795}{8} \text{ Bps} = 4599 \text{ Bps}$$

Z důvodu konečné rychlosti obslužného programu mikrokontroléru musí být přidána do vysílání časová rezerva, program spolehlivě fungoval při vzorkovací rychlosti do 4246 Hz. Audiosignál je vzorkován s rozlišením 8 bitů na vzorek (každý byte nese jeden vzorek), jeho mezní kmitočet je tedy max. polovina přenosové rychlosti, tj. přibližně 2100 Hz.

Popis vnější stanice:

Blokové zapojení vnější stanice je na Obr. 13. Zapojení je totožné s vnitřní stanicí, liší se pouze ovládacím tlačítkem, vynechanou LED pro indikaci vybité baterie a obvod navíc obsahuje výstup na ovládání elektromagnetického zámku dveří. Výstup má otevřený kolektor, napájení zámku tedy musí být přiváděno z vnějšího zdroje.

Vnitřní stanice musí být stále na příjmu, aby byla připravena zachytit vyzváněcí signál z vnější stanice. Vnější stanice však obsahuje časovač, který uvede jádro mikrokontroléru do úsporného režimu, pokud nebyl po 25 s. zachycen signál z vnitřní stanice. Obnovení činnosti se provede stiskem tlačítka Zvonek, tím se vyšle signál zvonění a komunikace může pokračovat.



Obr. 13: Blokové zapojení vnější stanice

Vnitřní A/D převodník dovoluje měřit vlastní napájecí napětí. Tuto vlastnost využíváme pro hlídání stavu baterií ve vnější stanici. Mikrokontrolér měří napětí baterie a posílá v každém paketu stavové hlášení přijímači. V případě nízkého napětí baterií vnitřní stanice upozorní rozsvícením červené LED na nutnost vyměnění baterie ve vnější stanici.

Popis ovládacího programu:

Řídící program je napsán v jazyku C, kompilace byla provedena pomocí programu Keil uVision 3, verze V3.60. Program obsahuje hlavní smyčku main(), ve které je napřed inicializován transceiver pomocí InitTRX() a poté je podle logické úrovně na vývodu P0.2 provedeno rozdělení programu na vnější a vnitřní stanici. Příslušné funkce External() a Internal() jsou realizovány nekonečnou smyčkou.

Funkce Internal() nejprve nastaví port P0, AD převodník, PWM a časovač T2, potom spustí příjem transceiveru a čeká na příjem dat. Kdykoliv při stisku tlačítka Hovor pak přejde do režimu vysílání.

Funkce External() začíná nastavením portu P0, nastaví probuzení sestupnou hranou na P03 (stisk tlačítka Zvonek) a přejde do úsporného režimu. Při stisku tlačítka Zvonek se obnoví činnost mikrokontroléru, nastaví se port P0, AD převodník, PWM, časovače T2 a T0, odešle se signál zvonění a obvod čeká na příjem dat z vnitřní stanice. Časovač T0 odměřuje 25 sekund do vypnutí, pokud nenastane odezva z vnitřní stanice. Přerušení od T2 probíhá v obou stanicích ve stejných intervalech 235 μ s (kmitočet 4246 Hz). Pokud je stanice právě nastavena jako vysílač, provede se spuštění AD převodu, jinak se zjišťuje, zda má přijímač aktuální data. Pokud jsou data aktuální, pošle se na PWM aktuální vzorek, jinak se PWM neobnovuje.

Komunikace po SPI není z důvodu potřeby maximální rychlosti realizována pomocí univerzální funkce, ale vloženými příkazy. Použitá rychlost komunikace po SPI 4 MHz je maximum, kterého lze dosáhnout při použitém řídicím krystalu 8 MHz. Při této rychlosti trvá odeslání byte 2 μ s, tedy 4 instrukční cykly.

Pro ukládání vysílaných nebo přijímaných vzorků slouží vyrovnávací paměť s kapacitou 32 byte, do které se zapisují data hromadně při příjmu a postupně se čtou, nebo při vysílání se postupně ukládají vzorky z AD převodníku a hromadně se odešlou.

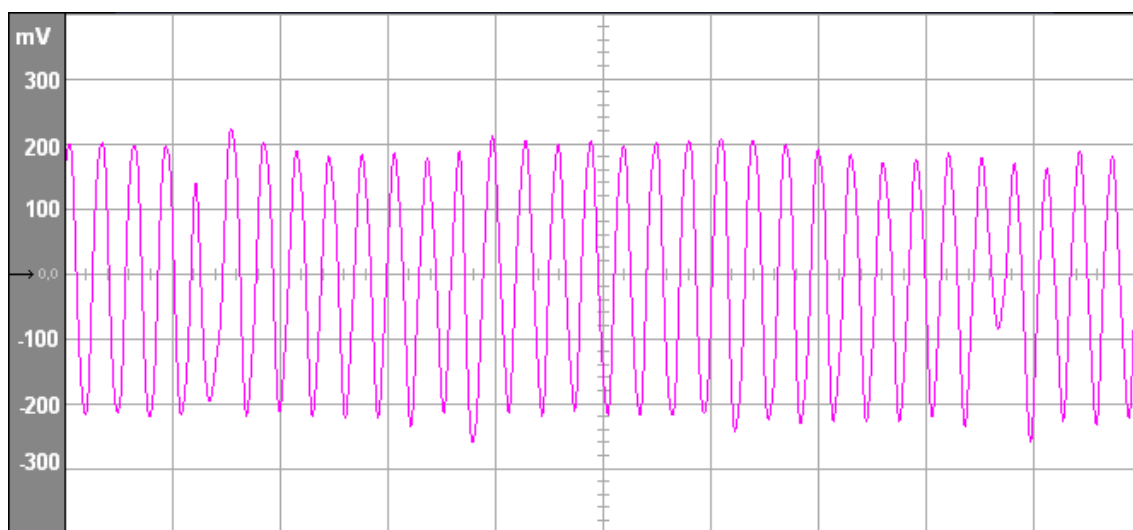
Signál zvonku je realizován vysíláním v paměti uloženého navzorkovaného sinusového kmitočtu. Pro dosažení efektu „bim bam“ je vysílán 0,5 s vyšší tón a navazující 0,5 s nižší tón.

Klíčovou částí programu je úsek ve funkci External(), který při každém novém vzorku z AD převodníku detekuje nosnou a stará se tak o to, aby při začátku vysílání vnitřní stanice přešla vnější stanice z režimu vysílání na příjem.

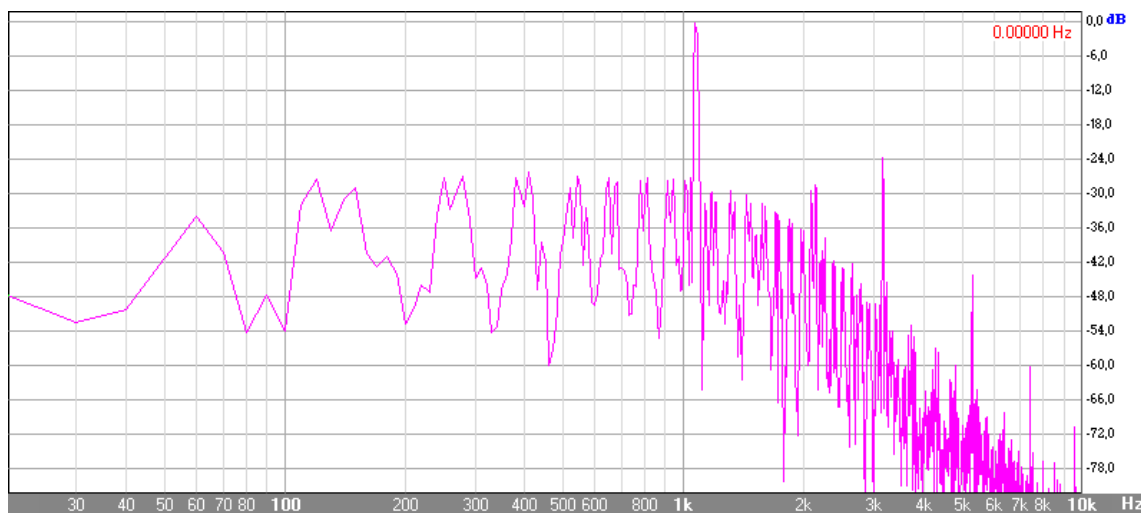
Kompletní výpis programu s komentáři je uveden v Příloze C.

Naměřené výsledky:

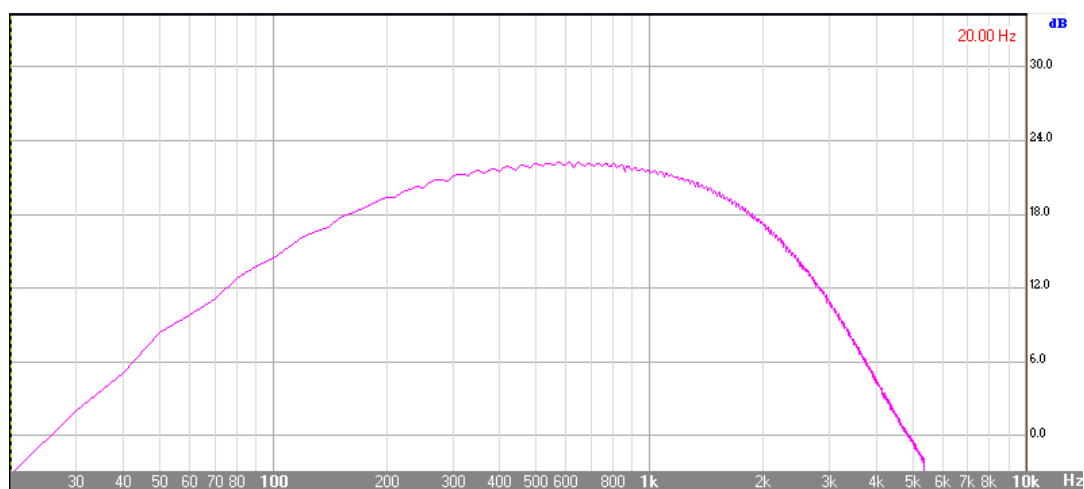
Následující obrázky ukazují frekvenční charakteristiky celého systému. Nejprve je ukázán časový průběh a spektrum při přenosu sinusového signálu 1 kHz z vnitřní stanice do vnější, následující 3 obrázky ukazují spektrální charakteristiky mikrofonního předzesilovače a filtru pro zesilovač reproduktoru.



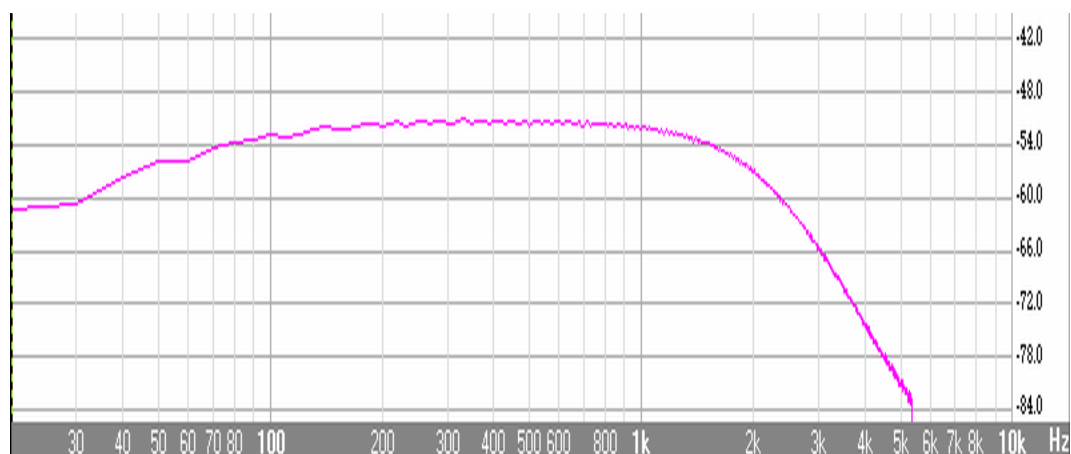
Obr. 14: Časový průběh přeneseného signálu 1 kHz



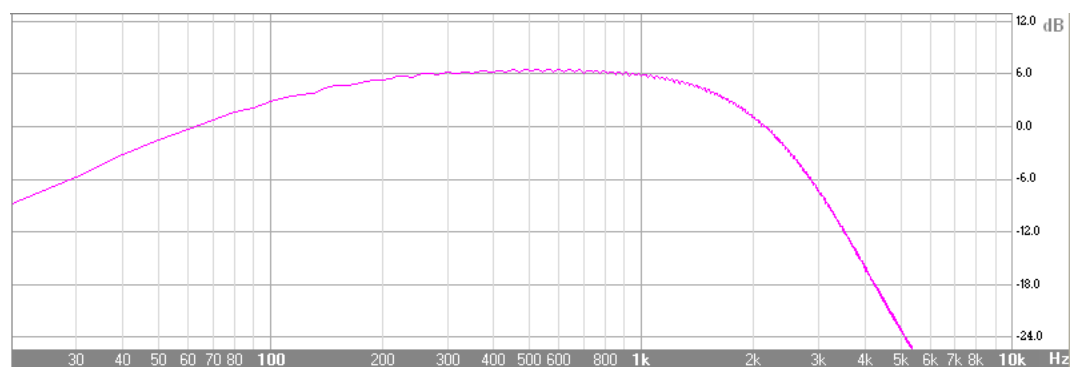
Obr. 15: Spektrum přeneseného signálu 1 kHz



Obr. 16: Přenosová charakteristika mikrofonního zesilovače s antialiasingovým filtrem



Obr. 17: Přenosová charakteristika antiimaging filtru od PWM před TDA7050



Obr. 18: Přenosová charakteristika jádra antiimaging filtru

Závěr:

Zadáním diplomové práce je konstrukce bezdrátového zvonku. Nejprve bylo navrženo blokové zapojení, které tuto funkci umožňuje. Postupně byly realizovány jednotlivé funkce systému. Jako první byl zprovozněn jednostranný přenos dat mezi dvěma stanicemi, poté výstup zvuku na reproduktor, mikrofonní zesilovač a nakonec byl program upraven pro oboustranný přenos hlasu a signalizaci zvonění. Nakonec byla navržena deska plošných spojů obou stanic a po jejím zhotovení a osazení byly desky umístěny do krabičky. Frekvenční rozsah přenášeného hlasu pro pokles o 3 dB je cca 150 Hz až 1500 Hz.

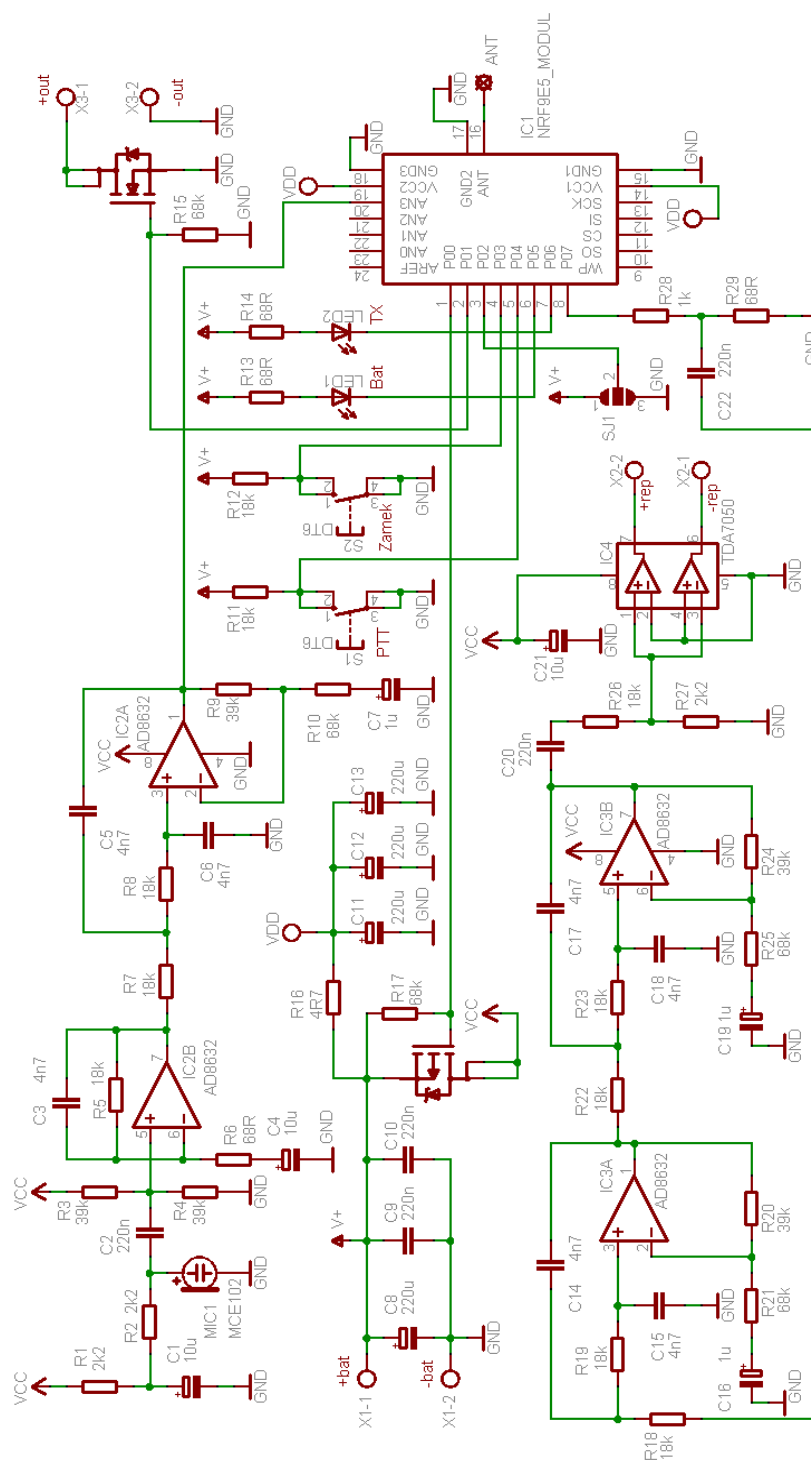
Výsledkem diplomové práce je tedy funkční systém realizující všechny požadované funkce a také grafické průběhy naměřených frekvenčních a časových charakteristik systému.

Seznam použité literatury:

- [1] http://www.nordicsemi.no/files/Product/data_sheet/Product_Specification_nRF9E5_v1.5.pdf – katalog použitého řídicího obvodu nRF9E5, oficiální stránky Nordic Semiconductor (květen 2008)
- [2] http://www.nordicsemi.no/files/Product/data_sheet/Product_Specification_nRF905_v1.5.pdf – katalog transceiveru nRF905, který je integrován na čipu nRF9E5, stránky Nordic Semiconductor (květen 2008)
- [3] http://www.ctu.cz/1/download/Opatreni%20obecne%20povahy/VO_R_10_08_2005_24.pdf - Všeobecné ustanovení Českého telekomunikačního úřadu stanovující možnosti provozu zařízení využívajících kmitočty 433MHz a 868MHz (květen 2008)
- [4] http://www.nordicsemi.no/files/Product/applications/nAN24-05_rev1_2.pdf - Wireless hands-free using nRF24E1, white paper, Nordic VLSI ASA (květen 2008)
- [5] <http://www.sysala.cz/Elektro/Nordic/index.htm> – Bezdrátový modul pro pásma 433 a 868 MHz, Radek Václavík (květen 2008)
- [6] LÁNÍČEK, R. *Elektronika*. Praha: BEN – technická literatura, 2000. 479 stran.

Příloha A – bezdrátový zvonek:

Schema zapojení bezdrátového zvonku:



Seznam součástek pro bezdrátový zvonek:

Název:	Hodnota:	Pouzdro:	Pozn.:
IC1	Modul nRF9E5		Popsán níže
IC2, IC3	AD8632	SO8	
IC4	TDA7050	DIL8	
IC5	IRF7343	SO8	2x MOS
LED2	Červená LED	LED 5 mm	
R1, R2, R27	2k2	SMD 1206	
R3, R4, R9, R20, R24	39k	SMD 1206	
R5, R7, R8, R11, R12, R18, R19, R22, R23, R26	18k	SMD 1206	
R6, R13, R14, R29	68R	SMD 1206	
R10, R15, R17, R21, R25	68k	SMD 1206	
R16	4R7	SMD 1206	
R28	1k	SMD 1206	
C1, C4, C21	10 μ	SMD B	
C2, C9, C10, C20, C22	220n	SMD 1206	
C3, C5, C6, C14, C15, C17, C18	4n7	SMD 1206	
C7, C16, C19	1 μ	SMD A	
C8, C11, C12, C13	220 μ	SMD D	
MIC1	MCE102		Elektretový mikrofon
S1		DT6	Tlačítko
X1, X2	ARK300V-2P		Svorkovnice

Pro jednodušší manipulaci s modulem při pájení a kvůli možnosti výměny modulu je vhodné modul osazovat do zásuvné patice. Z důvodu nesymetrického rozmístění vývodů modulu je vhodné využít precizní lámací kolíkové a dutinkové lišty.

Protože jsou vnitřní i vnější stanice realizovány na shodné DPS, nejsou na obou deskách osazeny všechny součástky. Seznam součástek specifických pro každou stanici ukazují následující tabulky. Na DPS vnitřní stanice je potřeba pájkou propojit na SJ1 střední plošku s ploškou blíže k modulu nRF9E5, na vnější stanici se vzdálenější.

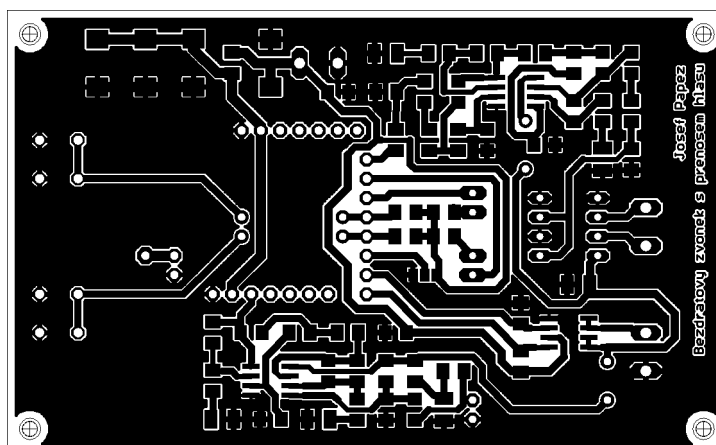
Součástky použité pouze ve vnitřní stanici:

Název:	Hodnota:	Pouzdro:	
LED1	Zelená LED	LED 5 mm	
R13	68R	SMD 1206	
S2		DT6	Tlačítko

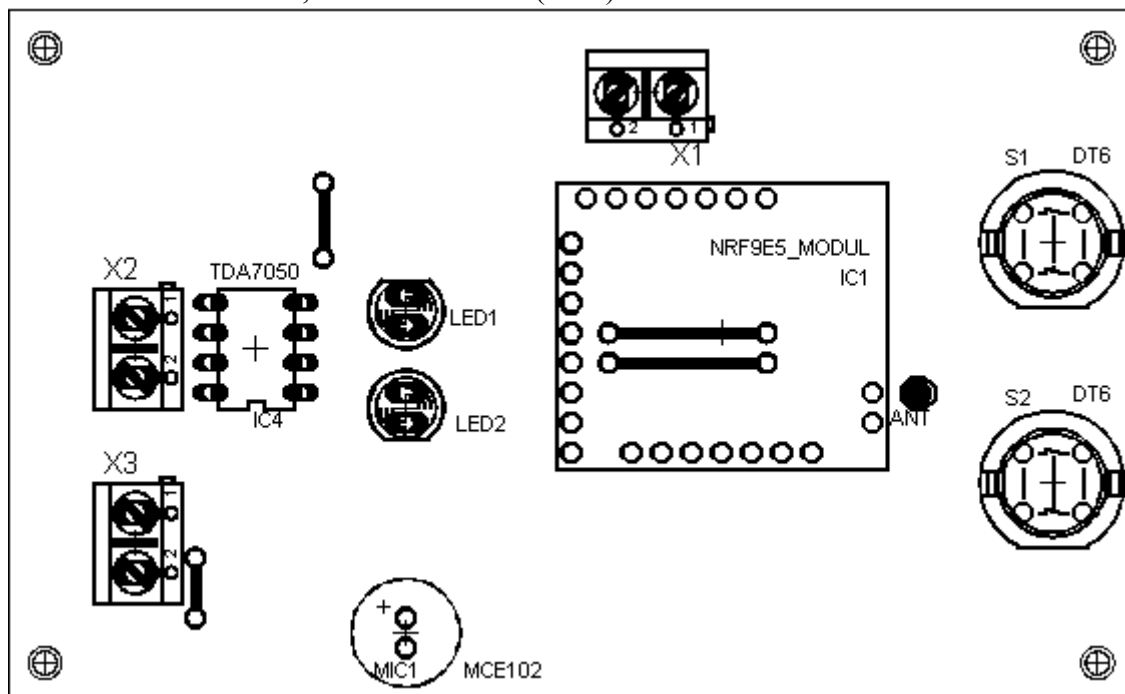
Součástky použité pouze ve vnější stanici:

Název:	Hodnota:	Pouzdro:	Pozn.:
X3	ARK300V-2P		Svorkovnice

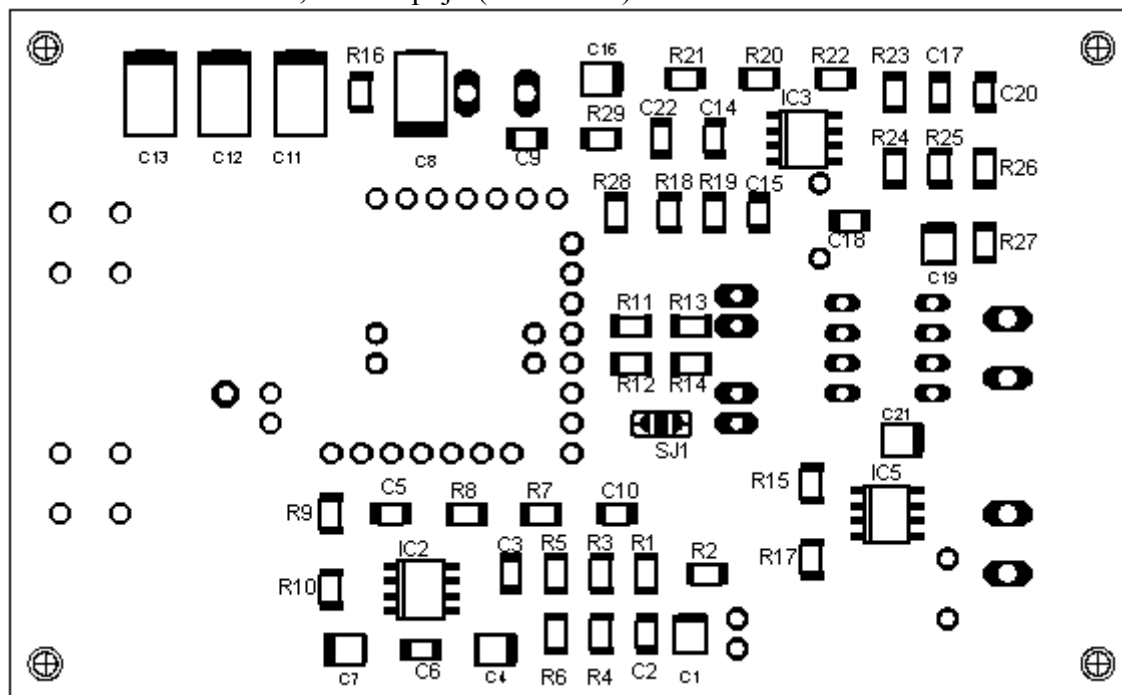
DPS, strana spojů (BOTTOM, M 1:1, rozměry 95 x 58 mm):



Osazení součástek, strana součástek (TOP):

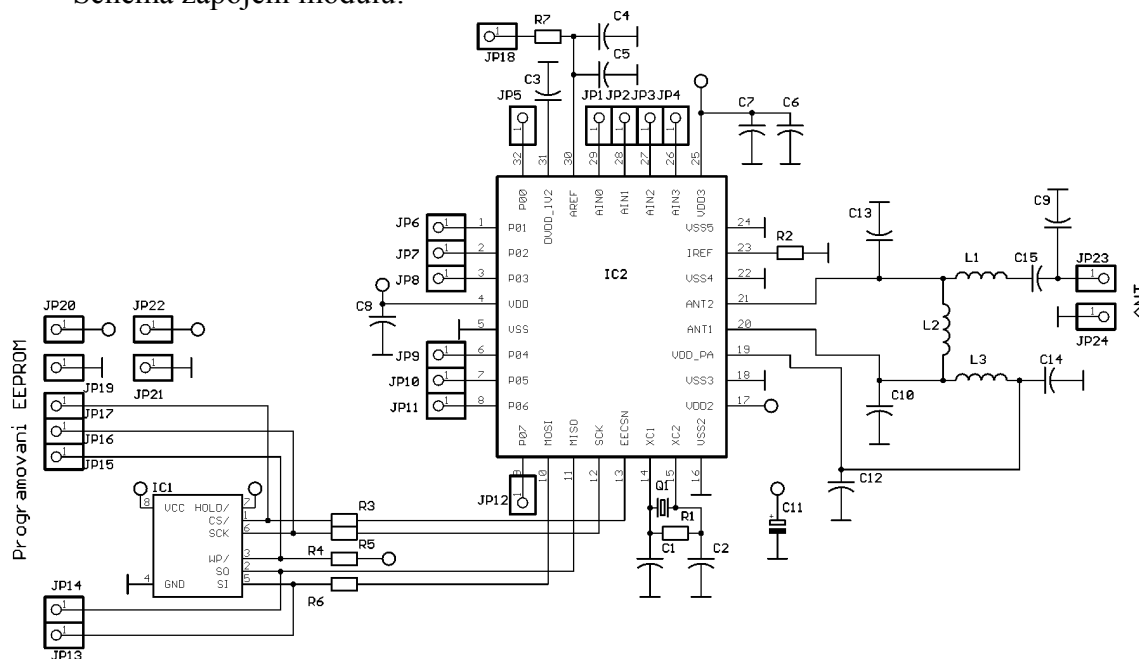


Osazení součástek, strana spojů (BOTTOM):



Příloha B – modul s nRF9E5:

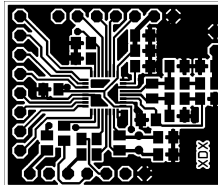
Schema zapojení modulu:



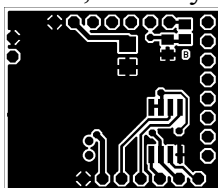
Seznam součástek pro modul:

IC1	25AA320	SO8	
IC2	nRF9E5	QFN32	
Q1	8 MHz		
R1	1M		
R2	22k, 1%		
R3, R4, R5, R6, R7	10k		
C1, C2	22p		
C3	10n		
C4	100n		
C5	1n		
C6, C12	3n3		
C7, C8	33p		
C9	Není použit		
C10, C13	18p		Pro 868 MHz 3,9p
C11	33μ/16 V		
C14	180p		Pro 868 MHz 33p
C15	6p8		Pro 868 MHz 33p
L1, L3	39 nH		Pro 868 MHz 12 nH
L2	12 nH		

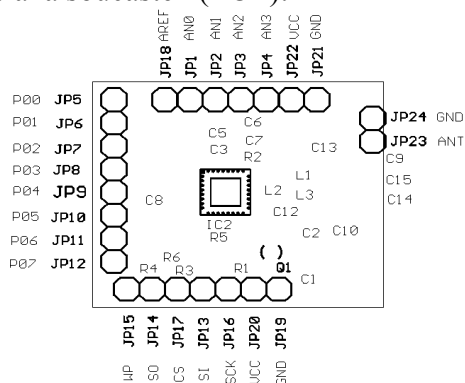
DPS, strana součástek (TOP, M 1:1, rozměry 29 x 25 mm):



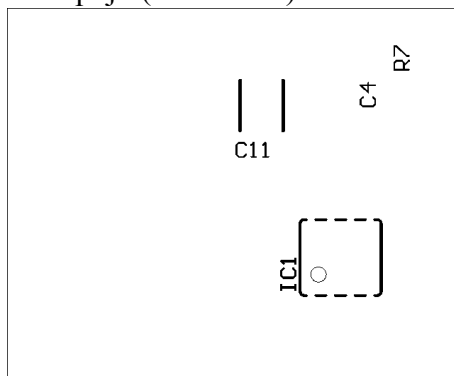
DPS, strana spojů (BOTTOM, M 1:1, rozměry 29 x 25 mm):



Osazení součástek, strana součástek (TOP):



Osazení součástek, strana spojů (BOTTOM):



Příloha C – výpis zdrojového programu pro nRF9E5:

```

/*= xpapez00_MMSM.c =====
*
* Bezdratovy domovni zvonek s digitalnim prenosem hlasu s nRF9E5.
* Zapojeni HW:
* Int: vnitřní stanice, Ext: vnější stanice
* P0.0: PMOS Analog VCC; 0 = VCC on
* P0.1: NMOS Lock out; 1 = otevření zamku (pouze Ext)
* P0.2: Ext/Int switch; 1 = Ext, 0 = Int
* P0.3: Talk switch(Int) / Ring switch(Ext) , external 10k Pull-up
* P0.4: Lock switch, external 10k Pull-up (pouze Int)
* P0.5: Bat LED (pouze Int); 0 = svítí
* P0.6: TX LED; 0 = svítí
* P0.7: Vystup audio (PWM)
* AN3: Vstup audio
*
* Popis SW:
* Casovac T2 odměruje čas pro snímání vzorku napětí z AN3
* Po sejmutí 32 vzorků je odeslán jeden paket se vzorky
* Smer vysílání je řízen tlačítkem Talk ve vnitřní stanici
* V režimu vysílání je poslána log. 0 na P06 (rozsvítí se TX LED)
* Přijímací buffer má 32 byte.
* Pokud buffer nemá aktuální data, PWM se neobnovuje.
* Vysílač je nastaven na minimální výkon (-10 dBm)
* Venkovní stanice začíná s uspořádaným režimem, probouzí se
* tlačítkem Talk/ring (log. 0 na pinu P0.3)
* Ve vnější stanici odměruje Timer0 25 sekund do vypnutí
* Pokud Vdd stanice External klesne pod 2,2 V, je vždy LSB prvního
* odesílaného vzorku v paketu nastaven, jinak je nulován
*
* COMPILER:
* Keil uVision V3.53
*
* 15. 5. 2008
*
*=====
*/
#include "reg9e5.h"
#include "stdio.h"
#include "intrins.h" // Hlavickovy soubor pro funkci _nop_ ();

#define POWER 0 // 0=min power...3 = max power
#define RXRED 0 // 0=normal, 1 = reduced RX power
#define HFREQ 0 // 0=433MHz, 1=868/915MHz
#define CHANNEL 115 // Channel number: f(MHz) =
(422.4+CHANNEL/10)*(1+HFREQ) 433.9MHz (pro kanal 115)

#define ON 1
#define OFF 0

#define SEND_COUNT 32 // Pocet vysilanych znaku

unsigned char sample; // Ukazatel poctu vzorku
unsigned char off_tim; // Casovac vypnuti vnější stanice
unsigned char idata buffer[32]; // Pamet vzorku audiosignalu
bit vysilac; // Priznak smeru prenosu
bit new_sample_flag; // Priznak noveho vzorku
bit ring; // Priznak zvoneni (Ext)

void _nop_(void);

unsigned char const idata zvon1[32] =

```

```

{
    0x7F, 0xF5, 0xD9, 0x4E,          // Pole vzorku sinusoveho signalu
    0x00, 0x4E, 0xD9, 0xF5,          // kmitocet fVZ * 6/32
    0x7F, 0x09, 0x25, 0xB0,
    0xFF, 0xB0, 0x25, 0x09,
    0x7F, 0xF5, 0xD9, 0x4E,
    0x00, 0x4E, 0xD9, 0xF5,
    0x7F, 0x09, 0x25, 0xB0,
    0xFF, 0xB0, 0x25, 0x09,
};

unsigned char const idata zvon2[32] =
{
    0x7F, 0xE9, 0xF5, 0x98,          // Pole vzorku sinusoveho signalu
    0x25, 0x02, 0x4E, 0xC6,          // kmitocet fVZ * 5/32
    0xFF, 0xC6, 0x4E, 0x02,
    0x25, 0x98, 0xF5, 0xE9,
    0x7F, 0x15, 0x09, 0x66,
    0xD9, 0xFC, 0xB0, 0x38,
    0x00, 0x38, 0xB0, 0xFC,
    0xD9, 0x66, 0x09, 0x15,
};

void Delay10us(volatile unsigned char n) // Zpozdeni o (20 + n*20) cyklu
{
    // @8 MHz: (10 + n*10) us
    while(n) // Zpozdeni o n * 20 cyklu
    {
        n--;
        n++; n++; _nop_ (); // Zpozdimе jeste o 10 cyklu
        n--; n--; _nop_ (); // n++, n--: 2 cykly, nop: 1 cyklus
    }
}

void InitPort(void)
{
    P0_DIR = 0x1C; // P0.4, P0.3 and P0.2 jsou vstupy, zbytek
    // P0.2: "0" -> Internal; "1" -> External
    // P0.3: Tlacitko Talk, vnejsi 10k Pull-up
    // P0.4: Tlacitko Lock, vnejsi 10k Pull-up
    // P0.7: PWM vystup
    // P0.6, P0.5: TX LED, Bat LED; obe vypnute
    // P0.1: NMOS otevreny kolektor; 0: zamek
    P0 = 0xFC;
    deaktivovan
}

// P0.0: PMOS Analog VCC; 0: VCC on

void InitTimer0(void)
{
    TMOD = 0x09; // T0 = 16 bit casovac
    TF0 = 0; // Smazeme pripadne cekajici preruseni od T0
    ET0 = 1; // Povolime preruseni od T0
}

void InitTimer2(void)
{
    CKCON |= 0x20; // T2M = 1 (timer clock = 1/4 CPU clk)
    RCAP2L = 0x29; // 4246 Hz tick...
    RCAP2H = 0xFE; // ... = 65536-8e6/(4*4246) = FE29h
    TF2 = 0; // Smazeme pripadne cekajici preruseni od T2
    ET2 = 1; // Povolime preruseni od T2
}

void InitPWM(void)
{
    // Nastaveni a spusteni PWM
    PWMCON = 0xC0; // 8 bitu PWM, fPWM = 31746 Hz @ 8 MHz
    PWMDUTY = 0x80; // vychozi strida = 50% @ 8 bits PWM
}

```

```

    P0_ALT |= 0x80;                // P0.7 (PWM) je vystup PWM
}

void InitADC(void)                // Nastaveni a spusteni ADC
{
    RACSN = 0;
    SPI_DATA = (WAC);             // Zapis do ADC config
    _nop_(); _nop_(); EXIF &= ~0x20;
    SPI_DATA = (0x3D);            // Vyber vstupu AIN3, PWR_UP = 1, VFSEL=1;
    _nop_(); _nop_(); EXIF &= ~0x20;
    SPI_DATA = (0x09);            // 8 bitu, zarovnani vpravo
    _nop_(); _nop_(); EXIF &= ~0x20;
    RACSN = 1;
}

void StopADC(void)               // Zastaveni ADC pro Power down
{
    RACSN = 0;
    SPI_DATA = (WAC);            // Zapis do ADC config
    _nop_(); _nop_(); EXIF &= ~0x20;
    SPI_DATA = (0x31);            // Vyber vstupu AIN3, PWR_UP = 0, VFSEL=0;
    _nop_(); _nop_(); EXIF &= ~0x20;
    RACSN = 1;
}

void PwrDwn(void)
{
    unsigned int i;
    P0 |= 0x61;                  // TX & Bat Led off, Analog Vcc off
    P0 &= ~0x82;                 // PWM & Lock off
    P0_ALT &= ~0x80;             // Zakazeme vystup PWM
    TRX_CE = 0;                  // Vypneme transceiver
    StopADC();                   // Zastavime ADC
    while((REGX_CTRL & 0x10) != 0); // Pockame na uvolneni registru
    REGX_MSB = 0xB0;             // Probuzeni neg. hranou na P03, 64 pulzu CKLF
debounce
    REGX_LSB = 0x00;
    REGX_CTRL = 0x0C;            // Spustime zapis do WWCON0
    while((REGX_CTRL & 0x10) != 0); // Pockame na uvolneni registru
    CK_CTRL = 0x04;              // Spustime rezim Deep Power Down

    InitPort();                  // Nastavime I/O porty
    InitADC();                   // Spustime ADC
    for(i=0;i<32;i++)            // Naplnime buffer vzorky zvoniciho signalu
    {
        buffer[i] = zvonl[i];    // Vysleme dalsi vzorek
    }
    sample = 31;                 // V bufferu jsou platna data
    vysilac = 1;                 // Budeme vysilat
    ring = 1;                    // Zvoneni
    P0_ALT |= 0x80;              // Povolime vystup PWM
    P0 &= ~0x41;                 // TX LED on, Analog Vcc on
}

void Timer0ISR (void) interrupt 1 // Probiha kazdych 100 ms
{
    unsigned int i;
    if(off_tim == 255)
    {
        PwrDwn();               // Prejdeme do usporneho režimu
    }
    off_tim++;                   // Zvysime citac vypnuti
    if(ring)                     // Pokud mame zvonit
    {
        if(off_tim == 5)
        {
            for(i=0;i<32;i++)    // Naplnime buffer dalsimi vzorky zvoniciho

```

```
signalu
{
    buffer[i] = zvon2[i];          // Vysleme dalsi vzorek
}
sample = 32;                      // V bufferu jsou platna data
// new_sample_flag = 1;
}
if(off_tim == 10)
{
    for(i=0;i<32;i++)              // Nulovani bufferu, prestavame zvonit
    {
        buffer[i] = 0x80;          // Vysilame ticho
    }
    sample = 32;                  // V bufferu jsou platna data
}
}
}

void Timer2ISR (void) interrupt 5 // Probiha kazdych 220 us
{
    TF2 = 0;                      // Smazeme priznak preruseni od T2
    if(vysilac)                   // Pokud je vysilac
    {
        new_sample_flag = 1;      // Novy vzorek k zapisu
    }
    else                           // Prijimac
    {
        if(sample < SEND_COUNT)   // Porovname, zda jsme na konci platnych dat
        {
            PWM_DUTY = buffer[sample]; // Zapis vzorku na PWM
            sample++;
        }
    }
}

void Receive(void)
{
    unsigned char i;
    RACSN = 0;
    SPI_DATA = (RRP);              // Preceteme prijaty paket
    sample = 0; _nop_ (); EXIF &= ~0x20; // Nulujeme pocitadlo vzorku
    for(i=0;i<32;i++)
    {
        SPI_DATA = 0;              // Odesleme prazdny byte pro zahajeni prijmu
        SPI
        _nop_ (); _nop_ (); EXIF &= ~0x20;
        buffer[i] = SPI_DATA;      // Zapiseme blok 32 vzorku do horni casti
        bufferu
    }
    RACSN = 1;
}

void Internal(void)               // Program vnitřní stanice
{
    unsigned int i;
    InitPort();                   // Nastavíme I/O porty
    InitADC();                    // Spustíme ADC
    InitPWM();                    // Spustíme PWM
    InitTimer2();                 // Spustíme časovač 2
    sample = 0;                   // Vynulujeme pocitadlo vzorku
    vysilac = 0;                  // Smazeme priznak vysilani
    TXEN = 0;                     // Zvolíme přijímac
    TRX_CE = 1;                   // Spustíme příjem
    Delay10us(64);                // Pockáme na aktivaci přijímac
    TR2 = 1;                      // Spustíme Timer2
    for(;;)                       // Zacyklení
    {
```

```

if(vysilac)                                // Pokud máme vysílat
{
    if(new_sample_flag)                    // Pockame na další sejmутý vzorek
    {
        new_sample_flag = 0;
        // Read Out ADC Value
        RACSN = 0;
        SPI_DATA = (RAD);
        _nop_ (); _nop_ (); EXIF &= ~0x20;
        SPI_DATA = 0;
        buffer[sample] = SPI_DATA; // Uložit převedený vzorek
        sample++;                     // Zvýšit počet zapsaných vzorků
        RACSN = 1;
        RACSN = 0;
        SPI_DATA = (SAV | 0x03); // AD převod ze vstupu AN3
        _nop_ (); _nop_ (); EXIF &= ~0x20; // Pockame na dokončení přenosu
SPI
        RACSN = 1;                     // Ukončíme komunikaci po SPI

        if(sample == SEND_COUNT)        // Pokud máme sejmутý potřebný počet vzorků
        {
            sample = 0;                 // Vynulujeme počítadlo vzorků
            if(P0 & 0x08)                // Pokud nemáme vysílat (Talk switch off)
            {
                vysilac = 0;             // Smazeme příznak vysílání
                TXEN = 0;                 // Zvolíme přijímač
                TRX_CE = 1;               // Spustíme příjem
                P0 |= 0x40;                // TX Led off
                Delay10us(64);            // Pockame na aktivaci příjmu
            }
            else
            {
                RACSN = 0;                // Zčátek komunikace s TRX
                SPI_DATA = (WTP);          // Zápis do vysílání
                for(i=0; i<32; i++)        // Odesleme 32 vzorků do transceiveru
                {
                    EXIF &= ~0x20;
                    SPI_DATA = (buffer[i]); // Vysleme další vzorek
                }
                EXIF &= ~0x20;
                RACSN = 1;                 // Konec komunikace
                TRX_CE = 1;                 // Spustíme vysílání
                Delay10us(0);              // Pulz 10 us
                TRX_CE = 0;
            }
        }
    }
}
else
{
    if (DR)                              // Pokud jsou přijata nová data
    {
        Receive();
        if(buffer[0] & 0x01)              // Pokud je nastaven LSB prvního vzorku
        {
            P0 &= ~0x20;                  // Bat LED on
        }
        else
        {
            P0 |= 0x20;                   // Bat Led off
        }
    }
    if(!(P0 & 0x08))                     // Pokud máme vysílat (stisk Talk)
    {
        vysilac = 1;                     // Nastavíme příznak vysílání
        sample = 0;
        TRX_CE = 0;
    }
}

```

```

        TXEN = 1;                // Zvolíme vysilac
        P0 &= ~0x40;             // TX LED on
    }
}
}

void External(void)              // Program vnejsi stanice
{
    unsigned char i, timeout;
    P0_DIR = 0x1C;               // P0.4, P0.3 and P0.2 are inputs, the rest
    output                        // output
    // Vnejsi stanice zacina prechodem do usporneho rezimu
    PwrDwn();                     // Prejdeme do usporneho režimu
    // Bylo stisknuto tlacitko zvoneni, zacneme vysilat
    InitPWM();                    // Spustime PWM
    InitTimer2();                 // Nastavime casovac 2
    InitTimer0();                 // Nastavime casovac 0
    TR0 = 1;                     // Spustime Timer0
    TR2 = 1;                     // Spustime Timer2
    for(;;)                       // Zacykleni
    {
        if(vysilac)              // Pokud neni zachycena nosna, muzeme vysilat
        {
            if(new_sample_flag)
            {
                new_sample_flag = 0;
                if(CD)             // Pri detekci nosne
                {
                    sample = 32;
                    ring = 0;      // Konec zvoneni
                    vysilac = 0;   // Prestaneme vysilat
                    timeout = 0;   // Casovac vypadku nosne
                    P0 |= 0x40;    // TX Led off
                }
            }
            else
            {
                if(ring == 0)
                {
                    // Read Out ADC Value
                    RACSN = 0;
                    SPI_DATA = (SAV | 0x03); // AD prevod ze vstupu AN3
                    _nop_(); _nop_(); EXIF &= ~0x20;
                    RACSN = 1;
                    Delay10us(0);           // Pockame 10 us na dokonceni prevodu
                    RACSN = 0;
                    SPI_DATA = (RAD);        // Precteme prevedenou hodnotu
                    _nop_(); _nop_(); EXIF &= ~0x20;
                    SPI_DATA = 0;
                    _nop_(); _nop_(); EXIF &= ~0x20;
                    buffer[sample] = SPI_DATA; // Ulozime prevedeny vzorek
                    RACSN = 1;
                    if(sample == 0)         // Prvni vzorek
                    {
                        RACSN = 0;
                        SPI_DATA = (SAV | 0x08); // AD prevod z Vdd/3
                        _nop_(); _nop_(); EXIF &= ~0x20;
                        RACSN = 1;
                        Delay10us(0);         // Pockame 10 us na dokonceni prevodu
                        RACSN = 0;
                        SPI_DATA = (RAD);
                        _nop_(); _nop_(); EXIF &= ~0x20;
                        SPI_DATA = 0;
                        _nop_(); _nop_(); EXIF &= ~0x20;
                        i = SPI_DATA;         // Precteme prevedenou hodnotu napeti
                        RACSN = 1;
                        if(i < 0x93)         // Pokud napajeci napeti kleslo pod 2,1 V

```

```

        {
            //ADCDATA = Vdd_min*2^8 / 3,66
            buffer[sample] |= 0x01; // Nastavíme LSB prvního vzorku
            P0 &= ~0x20;           // Bat LED on
        }
        else
        {
            buffer[sample] &= ~0x01; // Nulujeme LSB prvního vzorku
            P0 |= 0x20;               // Bat Led off
        }
    }
    sample++;                        // Zvýšit počet zapsaných vzorku
    if(sample == 4)
    {
        TXEN = 0;                   // Zvolíme přijímac
    }
    if(sample == SEND_COUNT)
    {
        sample = 0;
        TRX_CE = 0;
        RACSN = 0;                  // Zčátek komunikace s TRX
        SPI_DATA = (WTP);           // Zapis do vysílac
        for(i=0;i<32;i++)           // Odesleme 32 vzorku do transceiveru
        {
            EXIF &= ~0x20;
            SPI_DATA = (buffer[i]); // Vysleme další vzorek
        }
        EXIF &= ~0x20;
        RACSN = 1;                  // Konec komunikace
        TXEN = 1;                   // Zvolíme vysílac
        TRX_CE = 1;                 // Spustíme vysílání
    }
}
}
else                                // Máme přijímat
{
    if (DR)                          // Pokud jsou přijata nová data
    {
        timeout = 0;                // Nulujeme počítadlo výpadku nosu
        off_tim = 0;                // Nulujeme čítec vypnutí
        Receive();                  // Nacteme přijatá data z transceiveru
    }
    else                             // Není zachycena nosná
    {
        timeout++;                  // Zvýšíme počítadlo
        if (timeout == 240)         // Pokud jsme 12 ms nezachytili nosnou
        {
            timeout = 0;
            vysílac = 1;             // Nastavíme příznak vysílání
            sample = 0;
            P0 &= ~0x40;             // TX LED on
        }
        Delay10us(4);              // Zpoždění 50 us
    }
}
}
}

void InitTRX(void)
{
    SPICLK = 1;                     // SPICLK = 1/2 CPU_clk
    SPI_CTRL = 0x02;                // Connect internal SPI controller to Radio

    // Switch to 8 MHz clock:
    RACSN = 0;
    SPI_DATA = (WRC | 0x09);

```

```

        _nop_ (); _nop_ (); // Pockame na odeslani pres SPI (vice nez 2
us)    EXIF &= ~0x20; // Smazeme priznak preruseni od SPI
    SPI_DATA = (0x4F); // 8 MHz clock, 8 CRC
        _nop_ (); _nop_ (); EXIF &= ~0x20; // Smazeme priznak preruseni od SPI
    RACSN = 1;

    CKLFCON = 0x0F; // LF Clk setup

    // Configure Radio:
    RACSN = 0;
    SPI_DATA = (WRC | 0x02); // Write to RF config address 2
        _nop_ (); _nop_ (); EXIF &= ~0x20;

    SPI_DATA = (0x22); // 2 byte RX & 2 byte TX address
        _nop_ (); _nop_ (); EXIF &= ~0x20;

    SPI_DATA = (0x20); // 32 byte RX payload
        _nop_ (); _nop_ (); EXIF &= ~0x20;

    SPI_DATA = (0x20); // 32 byte TX payload
        _nop_ (); _nop_ (); EXIF &= ~0x20;
    RACSN = 1;

    RACSN = 0;
    SPI_DATA = (RRC | 0x01); // Read RF config address 1
        _nop_ (); _nop_ (); EXIF &= ~0x20;

    SPI_DATA = 0; // Odesleme prazdny byte pro zahajeni prijmu
SPI    _nop_ (); _nop_ (); EXIF &= ~0x20;
    sample = SPI_DATA; // Docasne ulozeni
    sample &= 0xf0; // Clear the power and frequency setting bits
    RACSN = 1;

    RACSN = 0;
    SPI_DATA = (WRC); // Write RF config address 0
        _nop_ (); _nop_ (); EXIF &= ~0x20;

    SPI_DATA = (CHANNEL & 0xff); // CHANNEL bit7..0
    // Change power defined by POWER above, to 433 or 868/915MHz defined by HFREQ
and    // bit8 of CHANNEL:
        _nop_ (); _nop_ (); EXIF &= ~0x20;

    SPI_DATA = (sample | (RXRED<<4) | (POWER<<2) | (HFREQ << 1) | ((CHANNEL >> 8)
& 0x01));
        _nop_ (); _nop_ (); EXIF &= ~0x20;
    RACSN = 1;
    EA = 1;
}

void main(void)
{
    InitTRX(); // Nejprve nastavime transceiver
    if(P0 & 0x04) // P0.2: propojka Ext/Int
    {
        External(); // Program vnejsi stanice
    }
    else
    {
        Internal(); // Program vnitřni stanice
    }
}

```