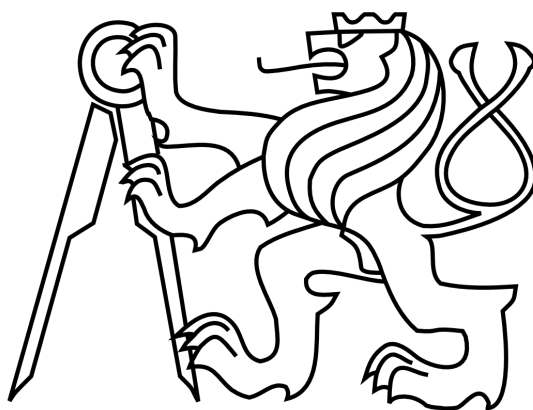


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE FAKULTA ELEKTROTECHNICKÁ
KATEDRA ŘÍDICÍ TECHNIKY

BAKALÁŘSKÁ PRÁCE

Převodníková deska pro komunikaci s autodráhovou
platformou



Jan Hakl

Květen 2015

Vedoucí práce: Ing. Dan Martinec

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jan Hakl**

Studijní program: Kybernetika a robotika

Obor: Systémy a řízení

Název tématu: **Převodníková deska pro komunikaci s autodráhovou platformou**

Pokyny pro vypracování:

1. Realizujte DPS pro převod mezi komunikačními standardy Zigbee, Nordic, Wifi a Bluetooth.
2. Naprogramujte DPS desku pro komunikaci s připraveným autodráhovým vozidlem.
3. Realizujte jednoduché grafické testovací rozhraní.

Seznam odborné literatury:

- [1] Richard Zuravski et al., Embedded systems handbook [Vol. 1] Embedded systems design and verification, CRC Press, 2009
- [2] Pavel Herout, Učebnice jazyka C, KOPP, 2009
- [3] Sarangapani, Jagannathan, Wireless ad hoc and sensor networks protocols, performance, and control, CRC Press, 2007

Vedoucí: Ing. Dan Martinec

Platnost zadání: do konce letního semestru 2015/2016

L.S.

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 12. 12. 2014

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne

.....

Podpis

Poděkování

Chtěl bych poděkovat svému vedoucímu bakalářské práce Ing. Danovi Martincovi za odborné vedení a také Ing. Ivo Hermanovi za rady a připomínky. Dále bych chtěl poděkovat rodičům za podporu při studiu.

Abstrakt

Cílem této práce je navrhnout a realizovat DPS pro převod mezi komunikačními standardy ZigBee, Nordic, Wifi a Bluetooth. Tato deska by měla sloužit ke komunikaci mezi autíčky autodráhy a uživatelem. Komunikovat by pak mělo jít nejen pomocí komunikačních rozhraní Nordic a ZigBee, které jsou přímo v autíčku, ale také pomocí Bluetooth a Wi-Fi.

Klíčová slova

Wi-Fi, Bluetooth, Nordic, ZigBee

Abstract

Goal of this bachelor thesis is to design and to realize PCB for conversion between ZigBee, Nordic, Wi-Fi and Bluetooth communicating standards. The board is providing communication with slot cars and user. Slot cars have only ZigBee and Nordic communicating standards and with this board it will be possible to communicate also through Bluetooth and Wi-Fi.

Key words

Wi-Fi, Bluetooth, Nordic, ZigBee

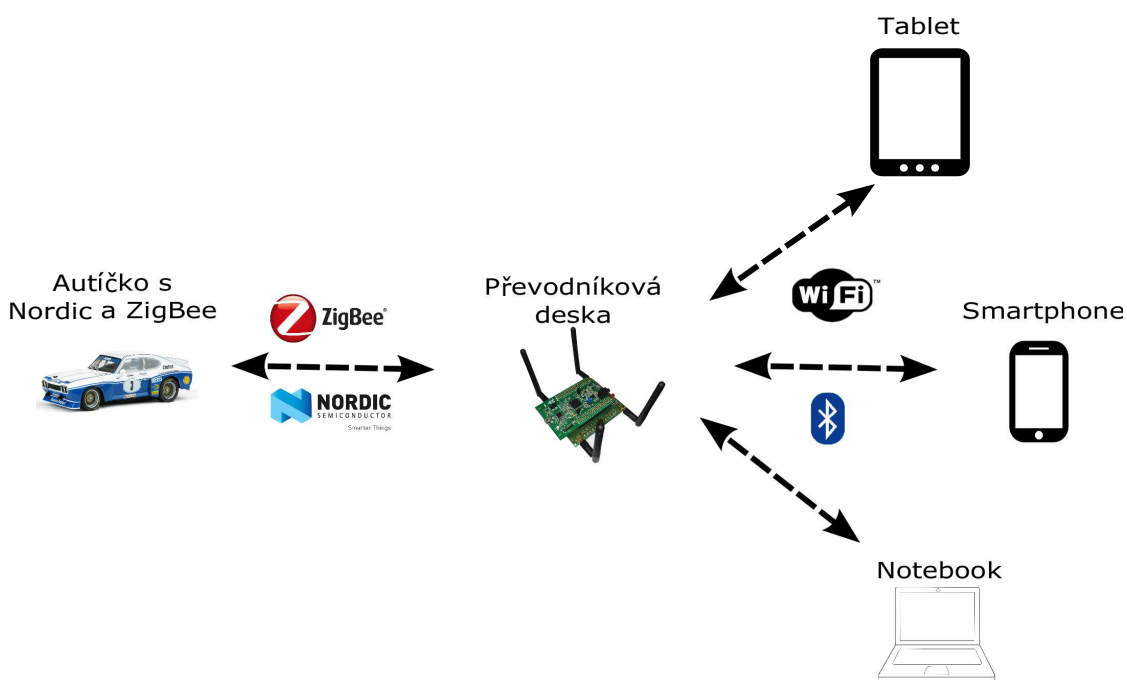
Obsah

1	Úvod	1
2	Popis použitých součástek	2
2.1	Volba vývojového kitu STM32F4 DISCOVERY	2
2.2	Procesor STM32F407VGT7	3
2.3	Wifi čip CC3100	3
2.4	Bluetooth čip CC2541	4
2.5	Zigbee čip CC2530	4
2.6	Nordic čip nRF24L01	4
2.7	Spínaný zdroj LMR10515YMF	5
2.8	Vývojové moduly	5
3	Návrh schématu a plošného spoje	7
3.1	Vývojové prostředí.....	7
3.2	Zdroj napájení a LED diody	7
3.3	Zapojení Wi-Fi čipu CC3100	9
3.4	Zapojení Bluetooth čipu CC2541	10
3.5	Zapojení ZigBee čipu CC2530	11
3.6	Zapojení Nordic nRF24L01.....	12
3.7	Návrh plošného spoje.....	12
4	Programování převodníkové desky.....	14
4.1	Programování STM32F4 DISCOVERY	14
4.2	Bluetooth	16
4.3	ZigBee	19
4.4	Nordic	21
4.5	Programování a testování čipů CC2541 pro Bluetooth a CC2530 pro ZigBee na vývojové desce SmartRF05 Rev. 1.8.1.....	22
4.6	Aplikace Bluetooth Converter Board	24
5	Závěr	27
6	Reference.....	28
	Seznam Zkratek	30
	A Kompletní schéma převodníkové desky	31

B Rozmístění součástek na horní a dolní straně převodníkové desky	32
C Obsah přiloženého CD	33

1 Úvod

Tato bakalářská práce je součástí projektu Slotcar platooning, který má za cíl vytvořit platformu pro distribuované řízení kolony autíček. Projekt převodníkové desky by měl rozšířit komunikační možnosti autíček na autodráze. Autíčka mají v sobě zabudované komunikační standardy ZigBee a Nordic. Komunikační deska by měla rozšířit možnosti komunikace s autíčky také o Wi-Fi a Bluetooth, což by umožnilo komunikovat s autíčky i pomocí tabletu, smartphone nebo notebooku (Obr.1). Tento projekt řeší jak HW část návrhu desky a její realizace, tak i SW část pro obsluhu jednotlivých komunikačních standardů a přeposílání zpráv mezi nimi. V rámci projektu jsem vytvořil i Java aplikaci, která umožňuje vyzkoušet funkčnost převodníkové desky.

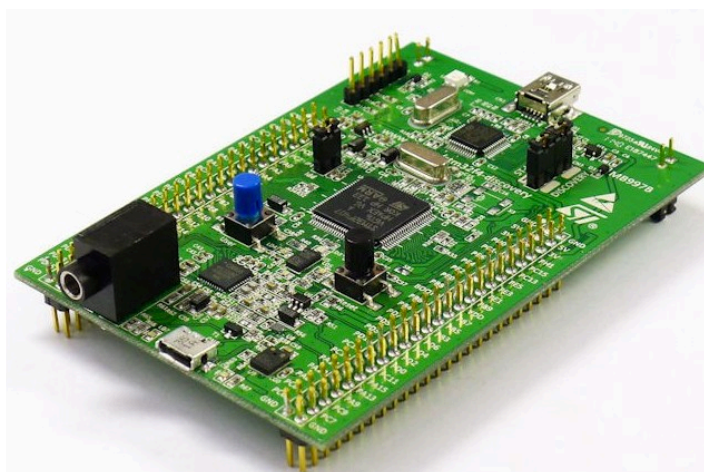


Obrázek 1. Schéma popisující funkci převodníkové desky [16]

2 Popis použitých součástek

2.1 Volba vývojového kitu STM32F4 DISCOVERY

Pro propojení a řízení komunikace mezi standardy Zigbee, Nordic, Bluetooth a Wifi jsem zvolil vývojový kit STM32F4 DISCOVERY (Obr.2) s čipem STM32F407VGT. Při výběru tohoto kitu, na kterém bude tento projekt postaven jsem vybíral především podle výkonu procesoru, počtu pinů a ceny. Procesor architektury ARM byl tedy vhodnou volbou především kvůli velmi příznivému poměru výkonu a ceny. Dalším důvodem pro volbu tohoto kitu bylo to, že stejný procesor je již umístěn i v autíčkách pro jejich řízení a komunikaci. Výhodou použití tohoto čipu na vývojovém kitu STM32F4 DISCOVERY je jeho snadné programování a debugování krok po kroku pomocí nástroje ST-LINK. Kit je dále vybaven 3-osým akcelerometrem, DAC převodníkem a audio výstupem, tlačítkem a čtyřmi programovatelnými diodami. Výhodou vývojového na kitu je, že k němu lze jednoduše pomocí pinů připojit jednotlivé vývojové moduly Bluetooth, Zigbee a Wifi. Dá se tak vyzkoušet práce s nimi ještě dříve, než je hotový návrh výsledné desky převodníku a předejít tak některým chybám v návrhu DPS. Kit může být napájen 5V z USB nebo externě 3V nebo 5V pomocí pinů. DISCOVERY kit je k převodníkové desce připojen 40-ti piny, ze kterých bude převodníková deska napájena.



Obrázek 2. Vývojový kit STM32F4 DISCOVERY [1]

2.2 Procesor STM32F407VGT7

Discovery kit [2] je vybaven ARM čipem typu STM32F407VGT7 s jádrem Cortex™-M4 pracujícím na frekvenci až 168 MHz a má jednotku plovoucí desetinné čárky. Na procesoru se nachází 1MB flash paměť a 192 kB statická RAM paměť. Spotřeba při práci s flash pamětí při vypnutých perifériích je velmi nízká, až 230 μ A/MHz při 168 MHz. V procesoru jsou tři 12-bit D/A převodníky, dva 12-bit D/A převodníky, 17 časovačů o rozsahu 16-bit a 32-bit a časovač pro PWM řízení motorů. Komunikačních rozhraní je na procesoru celkem 16, je na něm umístěno 6x USART s přenosovou rychlostí až 10,5 Mbit/s, 3x SPI s rychlostí až 42Mbit/s, 3x I2C, 2x CAN, USB, ETHERNET a SDIO. Procesor pracuje při napětí 1,8 V až 3,6 V a je umístěn v LQFP100.

2.3 Wifi čip CC3100

Pro komunikaci pro rozhraní wifi jsem vybral nově na trh uvedený čip CC3100 od výrobce Texas Instruments [3]. CC3100 je řešením pro připojení mikroprocesoru k wifi síti, je to první průmyslově certifikovaný Wifi čip. Jedná se o čip z Simple-Link Wifi rodiny, která velmi zjednodušuje implementaci připojení k internetu. Čip má sám v sobě integrovány všechny potřebné protokoly pro komunikaci po Wifi a připojení k internetu. Výhodou volby tohoto čipu je jeho nízká spotřeba.

Na čipu je umístěn ARM procesor Cortex®-M4, který funkcí předbíhá MCU. Tento subsystém zahrnuje 802.11 b/g/n rádio, MAC a wifi driver. Zařízení podporuje režimy Station a Access Point se zabezpečením pomocí klíčů WPA2 a WPS 2.0. CC3100 má rozhraní pro připojení k MCU pomocí SPI nebo UART. Požadavky na napájení jsou od 2,1 V do 3,6 V. Spotřeba je v režimu hibernace 4 μ A, v režimu deep sleep 115 μ A, při příjmu signálu RX 53mA a při vysílání TX 223 mA. Vysílaný výkon je 18,00 dBm a citlivost přijmače je -95,7 dBm. To by spolu s anténou mělo zajistit dostatečnou sílu signálu pro komunikaci například s počítačem nebo tabletem. Čip je vyroben v pouzdře QFN-64.

2.4 Bluetooth čip CC2541

Pro komunikaci po rozhraní Bluetooth jsem vybral opět čip od výrobce Texas Instruments a to CC2541[4]. Čip je kombinací kvalitního RF vysílače a standardního průmyslového procesoru 8051. Jedná se o nízkospotřebový System-on-chip CC2541 s 256 kB flash paměť a 8kB RAM. Do flash paměti lze nahrát vlastní program pro obsluhu Bluetooth a periférií. Na čipu se také nachází 8-bit a 16-bit časovač, dvě výkonné USART, SPI a I2C rozhraní. Čip je napájený 2V- 3,6V, pro snížení spotřeby má čip 3 různé módy, při režimu „external interrupts“ je spotřeba 0,5 μ A, v režimu „Sleep timer on“ 1 μ A a v režimu „4- μ s Wake-Up“ je to 270 μ s. V aktivním módu, při příjmu RX je spotřeba 17,9 mA a při vysílání TX 18,2 mA. Čip je vyroben v pouzdře QFN-40. Citlivost přijímače je -93dBm.

2.5 Zigbee čip CC2530

Pro komunikaci pomocí Zigbee jsem použil stejný chip jako ten, který je již umístěn v autíčku a se kterým bude potřeba komunikovat.

Jedná se o CC2530 [5] nízkospotřebový System-on-chip od Texas Instruments, který je vlastnostmi velmi podobný Bluetooth čipu CC2541. Na čipu se nachází kvalitní RF vysílač a 8051 MCU. Dále je v něm umístěn 8-bit a 16-bit časovač, dvě výkonné USART a SPI rozhraní. Napájen je 2V- 3,6V a pro snížení spotřeby má tři různé módy, při režimu „external interrupts“ je spotřeba 0,4 μ A, v režimu „Sleep timer on“ 1 μ A a v režimu „4- μ s Wake-Up“ je to 200 μ A. V aktivním módu, při příjmu RX je spotřeba 24 mA a při vysílání TX 29 mA. Čip se vyrábí stejně jako CC2541 v pouzdře QFN-40. Citlivost přijímače je -97 dBm.

2.6 Nordic čip nRF24L01

Pro komunikaci po standardu Nordic je použit chip nRF24L01 od výrobce Nordic Semiconductor. Jedná se o stejný čip, který je umístěn v autíčku.

Nordic nRF24L01 [6] je ultra nízkospotřebový čip napájený z 1,9-3,6 V se maximální spotřebou v aktivním módu při vysílání nebo přijímání až 14 mA. V normálním režimu, kdy se nevysílá, má spotřebu v řádu μ A. Je to vhodné řešení do zařízení, které mají omezený zdroj napájení nebo jsou napájeny z baterie. Vyrábí se v pouzdru QFN-20. Pro komunikaci má tento čip rozhraní SPI, které je schopné

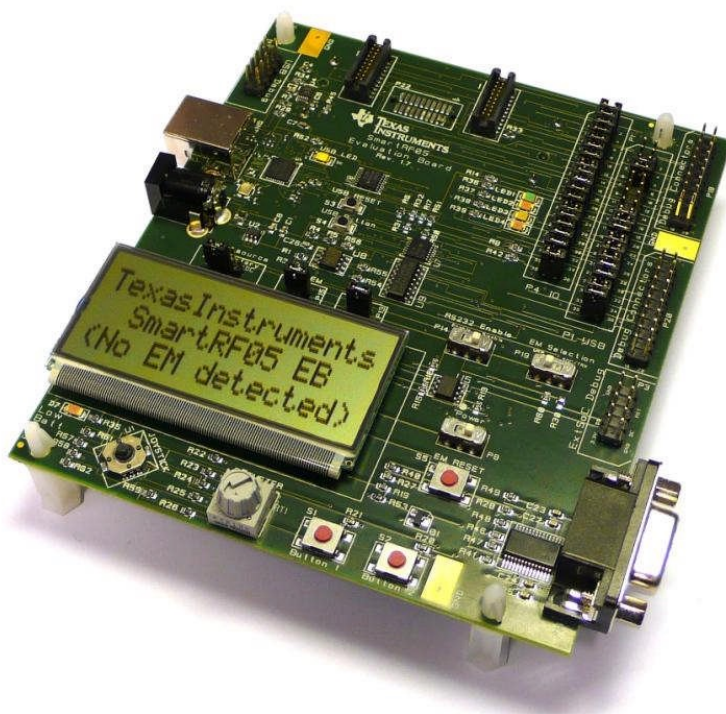
komunikovat až 10 Mbps. Bezdrátová přenosová rychlost je 1Mbps nebo 2Mbps. Citlivost přijímače je při přenosové rychlosti 1Mbps -85dBm a při 2Mbps -82dBm.

2.7 Spínaný zdroj LMR10515YM

Jedná se o vysokofrekvenční spínaný zdroj step-down od Texas Instruments [7]. Zdroj převádí 5V vstupního napětí na 3,3V výstupního napětí s maximálním proudovým odběrem až 1,5A což by měla být dostatečná rezerva i při špičkových odběrech Wi-Fi čipu.

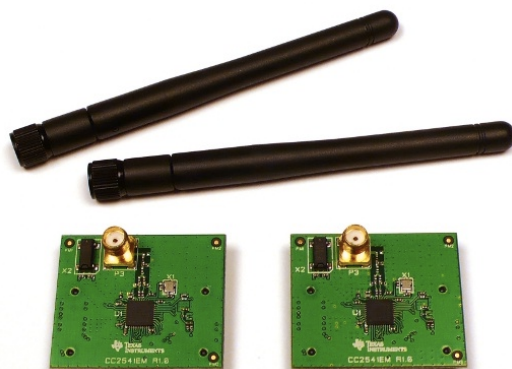
2.8 Vývojové moduly

Krom vývojového kitu STM32F4 DISCOVERY jsem dále používal vývojové moduly, které nejsou součástí převodníkové desky. Tyto moduly sloužily k odzkoušení všech aplikací, které by měli fungovat i na desce.



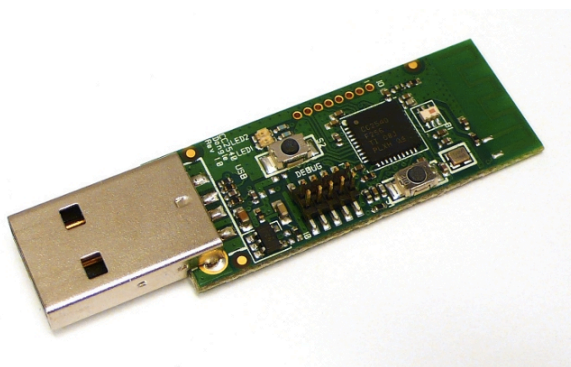
Obrázek 3. Vývojová deska SmartRF05 EB [8]

Moduly CC2541EM pro Bluetooth (Obr.4) a CC2530EM pro ZigBee je možno vyzkoušet pomocí vývojové desky SmartRF05 (Obrázek č.3), se kterou je možné čipy i naprogramovat. Na desce smartRF05b [8] se nachází LCD display, tlačítka, LED diody a jsou zde vyvedené a přístupné všechny piny čipu a rozhraní RS-232. S pomocí této desky a příkladů aplikací, které se dají do čipů nahrát lze vyzkoušet několik možností jejich využití.



Obrázek 4. Vývojový modul CC2541EM [12]

Pro komunikaci mezi Bluetooth modulem a počítačem jsem používal CC2540 USB Dongle (Obr.5) což je téměř stejný čip jako CC2541 s tím rozdílem, že má USB rozhraní místo UART nebo SPI.



Obrázek 5. CC2540 USB Dongle [13]

3 Návrh schématu a plošného spoje

3.1 Vývojové prostředí

Pro návrh schématu a plošného spoje jsem zvolil studentskou verzi programu Eagle 7.1.0. pro Mac OS X. Omezení Lite verze byly maximální rozměry desky plošného spoje a pouze dvouvrstvý plošný spoj. Výhodou Eagle naopak je, že vzhledem k jeho rozšíření se dají nalézt již hotové knihovny některých součástek.

3.2 Zdroj napájení a LED diody

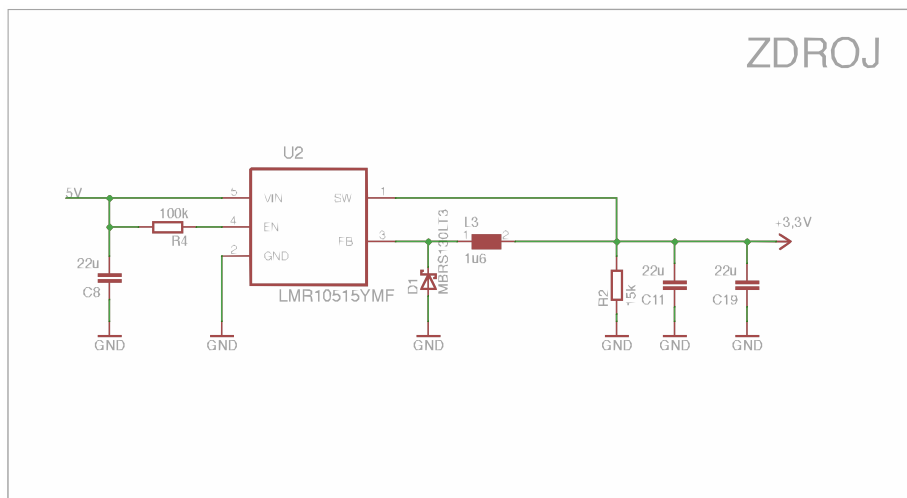
Všechny čipy pro bezdrátovou komunikaci jsou k procesoru připojeny pomocí SPI a jejich napájení zajišťuje spínaný 3,3V zdroj LMR10515YMF. Pro signalizaci různých stavů (zapnutí, komunikace, chybové stavy, testovací stavy), je na desce umístěno několik diod. Stavy, které diody signalizují popisuje tabulka č.1.

Popis signalizací LED1 až LED13	
LED1 - Zelená	Signalizuje zapnutý Bluetooth
LED2 - Zelená	Signalizuje zapnutou WiFi
LED3 - Zelená	Signalizuje zapnutý ZigBee
LED4 - Zelená	Signalizuje zapnutý Nordic
LED5 - Modrá	Signalizuje komunikaci po Bluetooth
LED6 - Modrá	Signalizuje komunikaci po WiFi
LED7 - Modrá	Signalizuje komunikaci po ZigBee
LED8 - Modrá	Signalizuje komunikaci po Nordic
LED9 - Červená	Signalizuje chybový stav Bluetooth
LED10 - Červená	Signalizuje chybový stav WiFi
LED11 - Červená	Signalizuje chybový stav ZigBee
LED12 - Červená	Signalizuje chybový stav Nordic
LED13 - Žlutá	Signalizuje běh ZigBee čipu

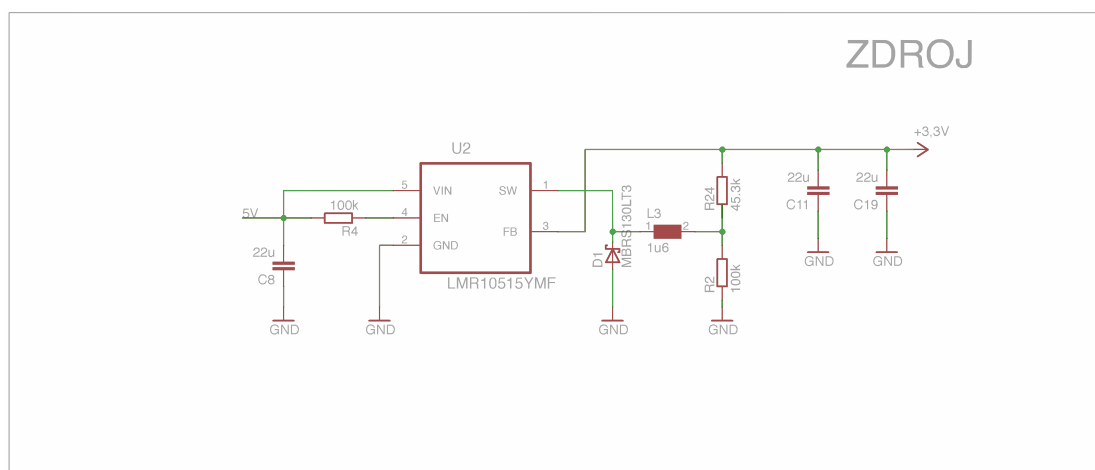
Tabulka 1. Popis signalizačních LED diod

Celá deska je napájena 5V pomocí USB, které se nachází na vývojovém kitu STM32F4 DISCOVERY. Toto napětí je přes piny přivedeno na spínaný zdroj LMR10515YMF, který toto napětí převádí na 3,3V pro napájení všech čtyř čipů. Zapojení zdroje [7] se skládá z odporu, cívky, schottkyho diody a třech kondenzátorů, které by měli zajistit dostatečný výkon zdroje i pro proudové špičky.

Při prvním návrhu schématu, podle kterého byl vytvořen první plošný spoj (Obr.6), došlo k chybě z důvodu výměny spínaného zdroje (původně jsem chtěl použít MCP16301). Druhá opravená verze je na Obr.7.



Obrázek 6. První návrh schématu spínaného zdroje

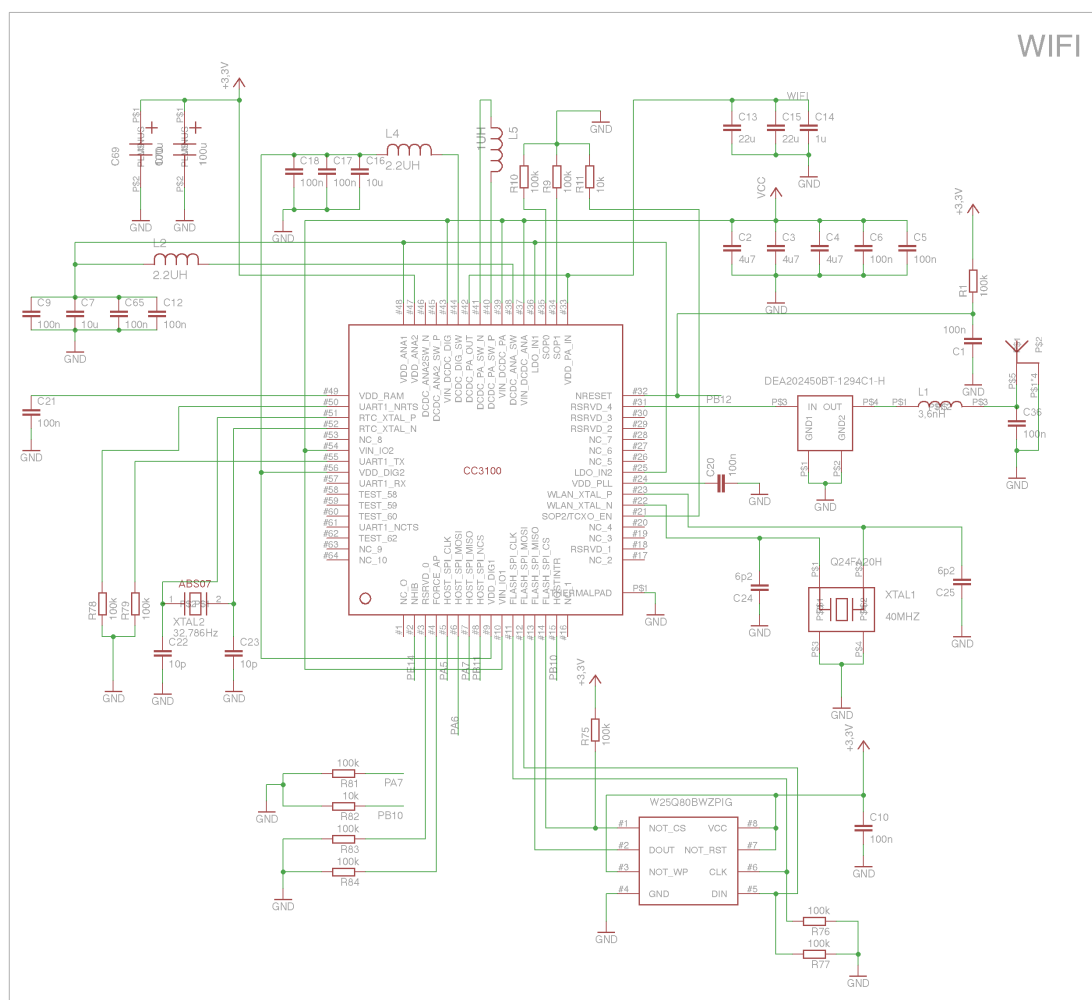


Obrázek 7. Schéma spínaného zdroje

Cívka L3 o hodnotě 1,6 μH byla na plošném spoji nahrazena cívkou o hodnotě 1 μH . To může způsobit o něco nižší výkon zdroje, ale ten by měl být i tak více než dostačující.

3.3 Zapojení Wi-Fi čipu CC3100

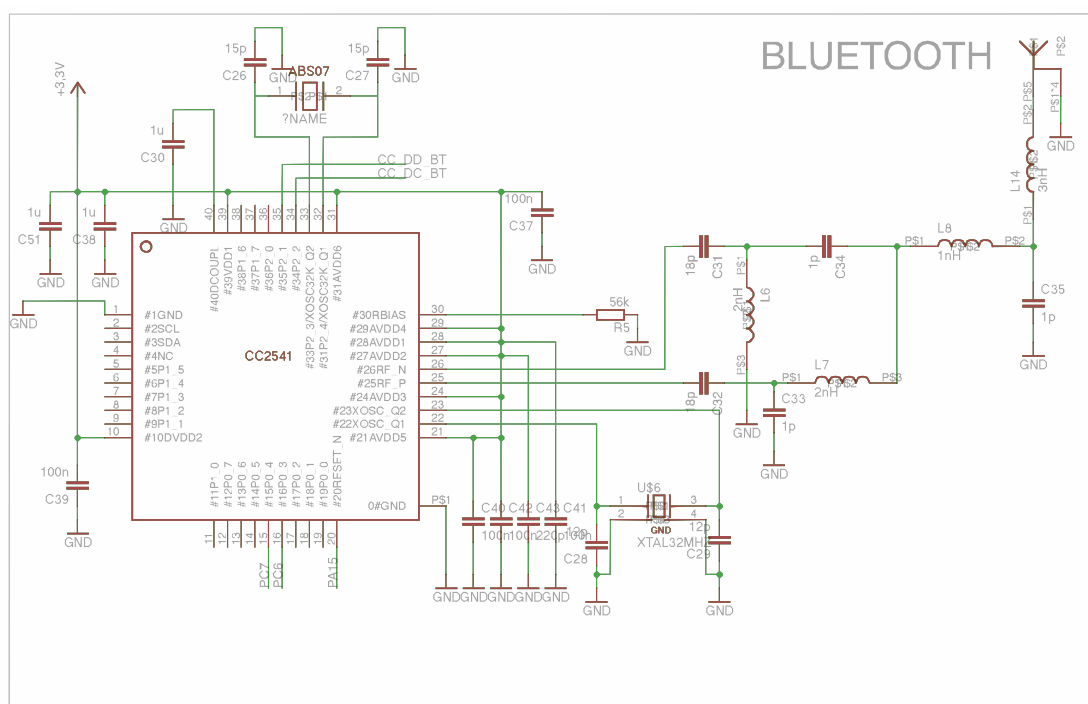
Zapojení (Obr. 8) Wi-Fi čipu CC3100 [3] vychází z doporučeného zapojení a do schématu byly přidány další blokační kondenzátory pro proudové špičky. Do procesoru STM32F4 je Wi-Fi čip připojen pomocí SPI rozhraní a signálových vodičů pro nRESET pro resetování čipu, nHIB pro nastavení stavu hibernace a vyvolání přerušení HOSTINT. Čip má dva oscilátory. Krystal XTAL1 zajišťuje taktovací frekvenci pro ARM procesor a Wi-Fi subsystem čipu. Krystal XTAL2 zajišťuje hodinový signál reálného času s frekvencí 32,768 kHz. K čipu je také připojena 8 Mb paměť, do které si čip ukládá nastavení sítě. Ta je k čipu připojena pomocí rozhraní SPI a to pomocí pinů FLASH_SPI_CSa, FLASH_SPI_DIN, FLASH_SPI_DOUT a FLASH_SPI_CLK.



3.4 Zapojení Bluetooth čipu CC2541

Zapojení Bluetooth čipu CC2541 vychází z doporučeného zapojení Keyfob [9] což je výrobek od Texas Instruments. Tento výrobek obsahuje stejný čip jako ten, který je použit zde a jeho schéma je volně přístupné na stránkách Texas Instruments. Čip je k hlavnímu procesoru STM32F4 připojen pomocí SPI, kde SCLK je na pinu P0_5, CS na pinu P0_4, MISO na pinu P0_3 a MOSI na pinu P0_2. K procesoru je také připojen signálový vodič pro reset, který je na čipu na pinu RESET_N. K čipu jsou připojeny dva oscilátory. Krystal XTAL3 s frekvencí 32,768-kHz zajišťuje hodiny reálného času pro sleep časovač. Krystal XTAL3 s frekvencí 32-MHz slouží k pro taktování procesoru a rádia.

V první verzi chyběly k čipu připojené konektory k pinům P2_2 pro CC, P2_1 pro DC, a RESET_N pro naprogramování čipu. V druhé verzi je čip připojen pomocí rozhraní UART, které se nachází na pinech P0_3 pro TX a P0_4 pro RX. Druhá verze schématu je na obrázku č.9.

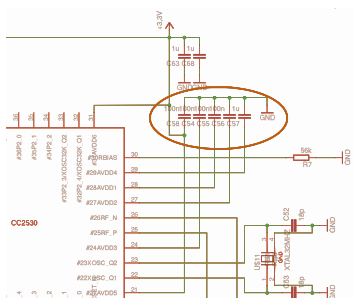


Obrázek 9. Schéma Bluetooth

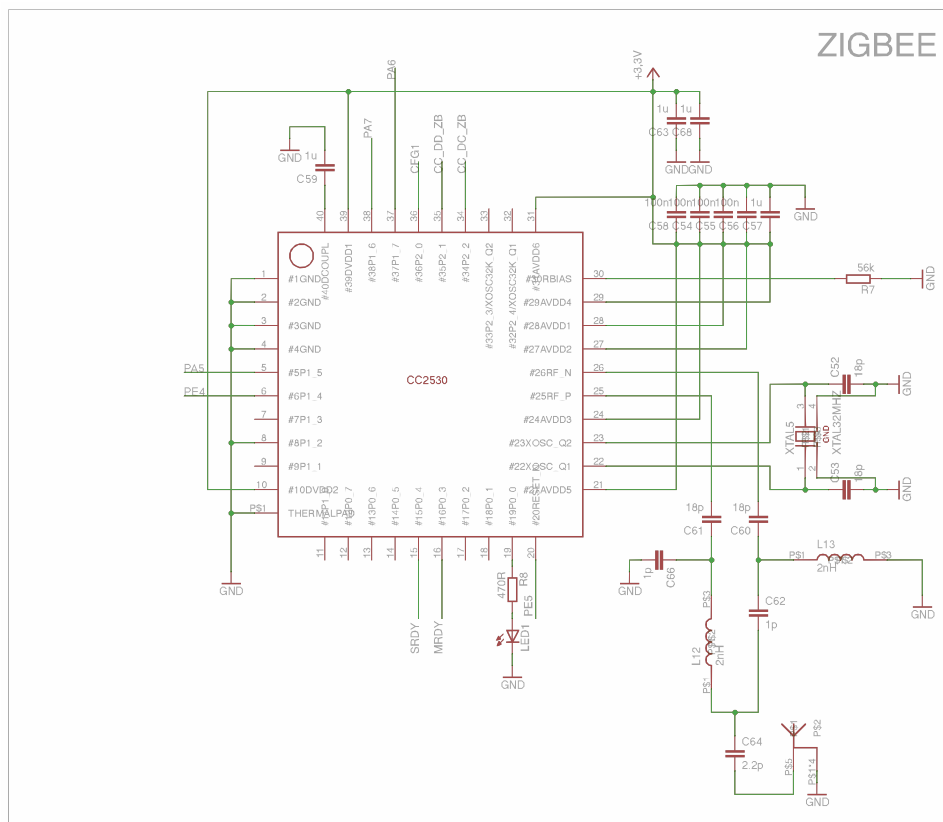
3.5 Zapojení ZigBee čipu CC2530

Schéma ZigBee čipu CC2530 je navrženo podle doporučeného zapojení z dokumentace k čipu od Texas Instruments[5]. K čipu je připojen jeden krystal XTAL5 o frekvenci 32-MHz.

V první verzi schématu, podle které byla vyrobena první deska, chyběly vyvedené piny pro naprogramování Bluetooth čipu CC2541. Jsou to piny P2_2 pro CC, P2_1 pro DC, a RESET_N. Také byli špatně propojené blokační kondenzátory (Obr.10). Druhá verze (Obr.11) tyto piny vyvedení již má.



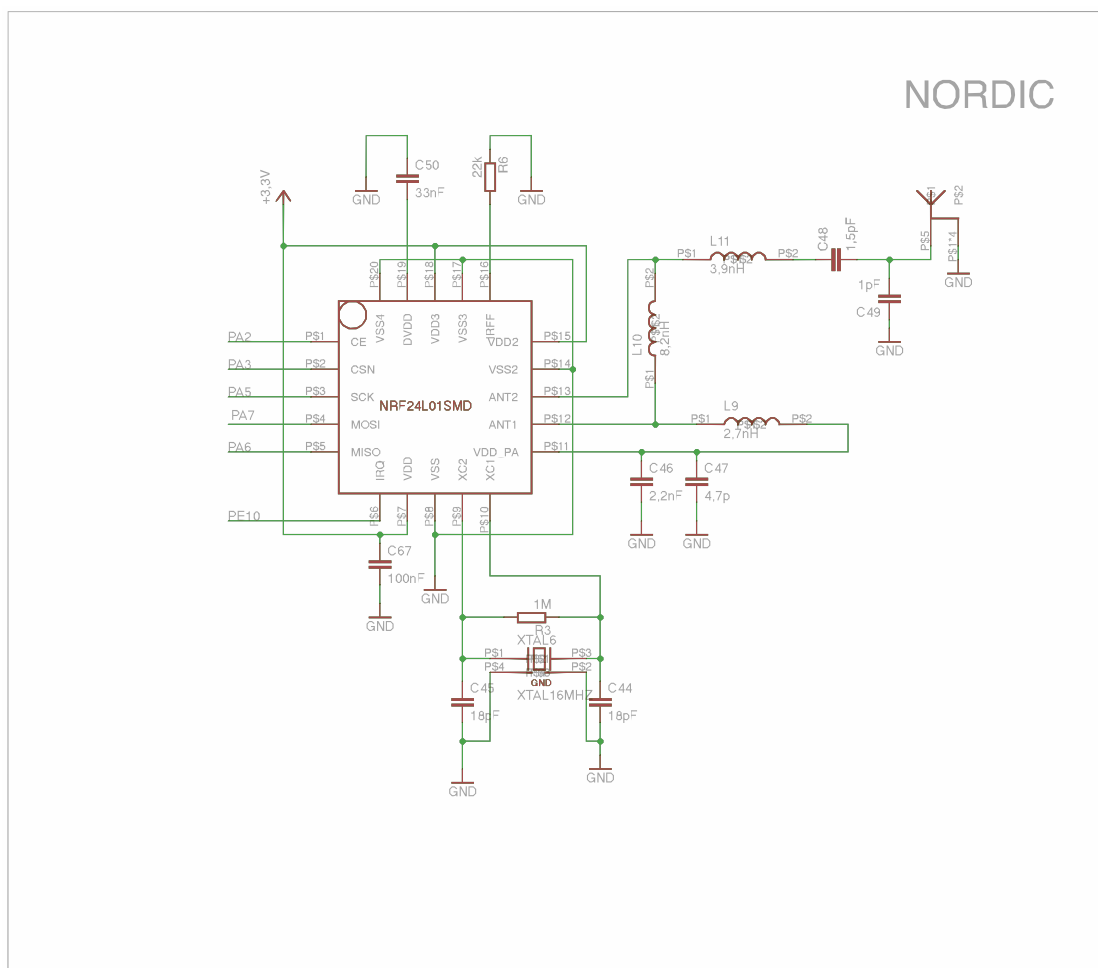
Obrázek 10. První verze se špatně zapojenými kondenzátory



Obrázek 11. Schéma ZigBee

3.6 Zapojení Nordic nRF24L01

Schéma (Obr.12) zapojení Nordic čipu nRF24L01 [6] je navrženo podle doporučeného zapojení z dokumentace k čipu od Nordic Semiconductor. K čipu je připojen jeden krystal XTAL6 s frekvencí 32-MHz.

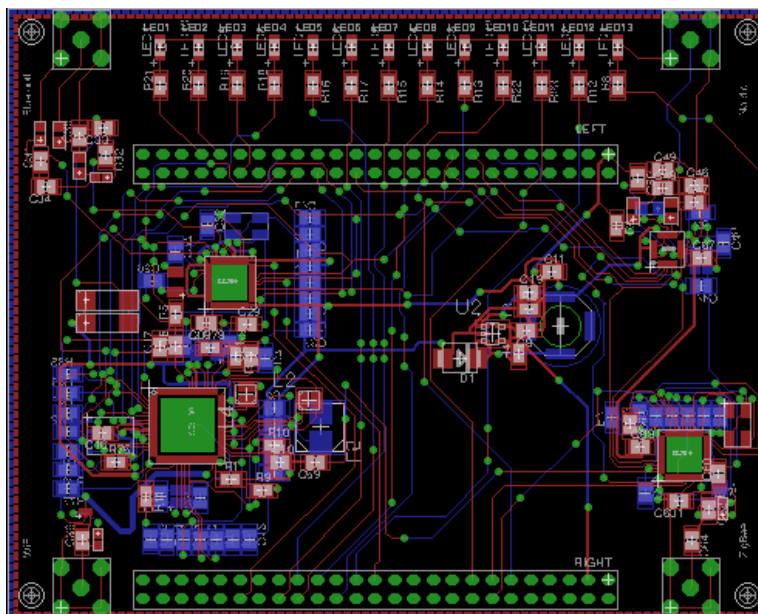


Obrázek 12. Schéma Nordic

3.7 Návrh plošného spoje

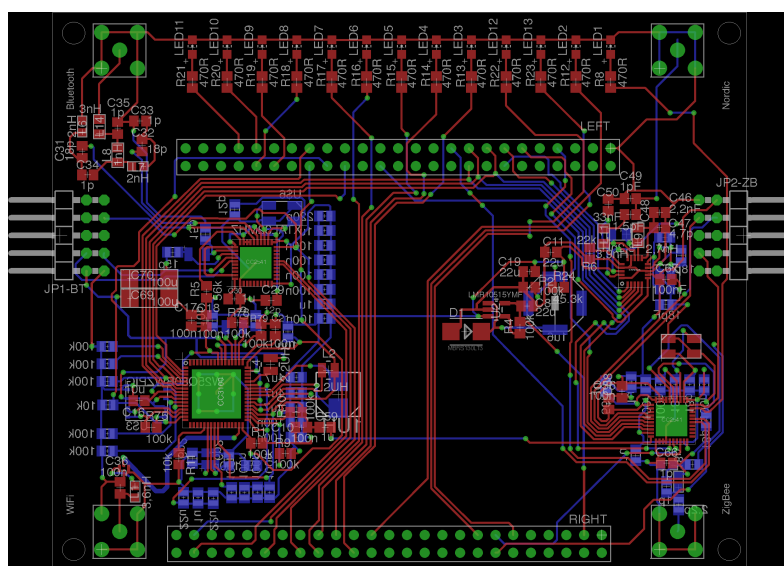
Poté, co jsem měl hotové schéma zapojení včetně vytvořených pouzder ke všem součástkám, jsem začal s návrhem desky plošných spojů (Obr.13). Rozvržení součástek na desce a její rozměr vyplývá především z toho, jak k ní bude připojena deska STM32F4 DISCOVERY kitu. Minimální rozměr desky byl daný DISCOVERY kitem (97x56 mm), aby byly dobře vidět signalizační diody a byl dobrý přístup k programovacím konektorům. Deska má velikost 100x80 mm což je rozměr, který povoluje Lite verze Eagelu. V každém rohu desky je umístěn konektor pro připojení

antény pro Wifi, Bluetooth, Zigbee a Nordic. Po horní straně desky je umístěno 13 LED diod pro signalizaci komunikace a různých stavů tak, aby byli vidět, když bude k desce připojen kit STM32F4 DISCOVERY.



Obrázek 13. První verze DPS převodníkové desky

V prvním návrhu schématu bylo několik chyb, které se promítly i do prvního návrhu DPS, ve které byla také špatná vzdálenost řad pinů, kterými je k desce připojen DISCOVERY kit. Tyto chyby jsem opravil přímo desce prvního návrhu ručně a pro připojení DISCOVERY kitu jsem vyrobil desku pouze s konektory, přes kterou je možné připojit. V druhé verzi DPS (Obr.14) jsou již tyto chyby opraveny, ale na její výrobu už nebyl dostatek času.



Obrázek 14. Druhá verze DPS převodníkové desky

4 Programování převodníkové desky

4.1 Programování STM32F4 DISCOVERY

Pro programování čipu na DISCOVERY kitu je možné použít několik prostředí, ale podporované ze strany STM jsou pouze Keil nebo IAR. Nejprve jsem pro programování čipu chtěl použít program Eclipse, který je volně ke stažení. Bohužel pro Eclipse jsou podporované pouze nové knihovny typu HAL a ne standartní, pro které je připravena většina příkladů, které se dají buď přímo od výrobce nebo z internetu stáhnout. Do Eclipse lze standartní knihovny manuálně importovat a nastavit, ale ne automaticky. Rozhodl jsem se tedy, že je pro tuto aplikaci vhodnější použít prostředí μ Vision od Keil, které je podporované ze strany STM a lze do něj jednoduše importovat jak knihovny HAL tak i standartní knihovny. Prostředí μ Vision od Keil má u verze, která je volně ke stažení omezenou maximální velikost programu na 32 KB a nemá optimalizaci kódu. I přes tyto omezení je volná verze μ Vision pro tento projekt dostačující. Pro programování a debugování čipu na desce pomocí USB bylo potřeba nainstalovat utilitu ST-LINK.

Pro obsluhu rozsvícení jednotlivých signalizačních LED diod jsem vytvořil soubor *converter_LED.c* a k němu hlavičkový soubor *converter_LED.h*. Tento soubor obsahuje funkci pro inicializaci LED diod *converter_LedInit(void)*, pro jejich rozsvícení *converter_LedOn(port, led)*, pro zhasnutí *converter_LedOff(port, led)*, pro přepnutí z rozsvícené na zhaslou nebo naopak *converter_LedToggle(port, led)* a pro zjištění zda je LED dioda rozsvícená je zde funkce *converter_LedIsOn(port, led)*. Při volání těchto funkcí, zapíšeme jako parametr název dané LED diody a port, na kterém se nachází. Definice těchto diod:

```
#define LED_1_GREEN    GPIO_Pin_11
#define LED_2_GREEN    GPIO_Pin_9
#define LED_3_GREEN    GPIO_Pin_8
#define LED_4_GREEN    GPIO_Pin_10
#define LED_1_BLUE     GPIO_Pin_2
#define LED_2_BLUE     GPIO_Pin_0
#define LED_3_BLUE     GPIO_Pin_1
#define LED_4_BLUE     GPIO_Pin_3
#define LED_1_RED      GPIO_Pin_6
#define LED_2_RED      GPIO_Pin_6
#define LED_3_RED      GPIO_Pin_2
#define LED_4_RED      GPIO_Pin_7
```

Diody *LED_2_RED* a *LED_3_RED* jsou na portu *GPIOE*, všechny ostatní jsou na portu *GPIOD*.

Dále jsem pro snadnější diagnostiku programu implementoval UART, přes který lze jednoduše vypisovat stavy programu přímo do terminálu na PC. Pro připojení k počítači jsem použil převodník ze sériové linky RS-232 do USB. Mezi převodníkovou deskou a sériovou linkou v počítači je potřeba také vyřešit rozdílné napěťové úrovně. UART převodníkové desky používá TTL úroveň kde 0V je logická nula a 3,3V je logická jednička. Standartní sériová linka RS-232 používá 0 - 15V pro logickou nulu a -15 - 0V pro logickou jedničku. Přizpůsobení výstupu UART jsem vyřešil jednoduchým převodníkem s obvodem MAX 3221.

Funkce pro obsluhu UART se nachází v souboru *uart_communication.c*. V tomto souboru je funkce *init_usart(void)* pro inicializaci UART. UART se inicializuje na pinech PA2 a PA3 s přenosovou rychlostí 9600 bit/s, 8-bit délkou slova, jedním stop bitem, bez paritního bitu a bez kontroly datového toku. Dále jsou zde funkce *fgetc()* pro přečtení znaku nebo *fputc(char c)* pro odeslání znaku.

Pro komunikaci po sériové lince s počítačem jsem používal program Docklight, který umožňuje přednastavit zprávy k odeslání a to jak v ASCII podobě nebo i hexadecimálně po 8 bitech.

Rozhraní SPI které je potřeba ke komunikaci se ZigBee čipem CC2530 a Nordic nRF24L01 zajišťuje soubor *SPI.c*. Pro inicializaci čipu je zde funkce *spi_Init()*, ta inicializuje SPI na pinech, které popisuje tabulka č.2.

Pin	Funkce pinu
PA5	SCLK - hodinový signál
PA6	MISO - master vstup, slave výstup
PA7	MOSI - master výstup, slave vstup
PE4	CS - ZigBee
PA3	CS - Nordic

Tabulka 2. Inicializace pinů pro SPI

Další piny, které jsou potřeba ke komunikaci s čipem ZigBee, jsou inicializovány ve funkci *znpSignalsInit()* popisuje tabulka č.3.

Pin	Funkce pinu
PC2	CFG1
PA1	CFG2
PC1	SRDY – Slave ready
PC5	MRDY – Master ready
PE5	RESET

Tabulka 3. Inicializace signálových pinů pro ZigBee

Rozhraní UART pomocí kterého se komunikuje s Bluetooth čipem CC2541 má funkce potřebné pro inicializaci a komunikaci v souboru *bluetooth.c*. Pro inicializaci je zde funkce *init_USART6()*, která inicializuje pin PC6 pro TX a PC7 pro RX a nastaví UART6 na přenosovou rychlost 115200 bit/s, počet datových bitů 8, žádný paritní bit, 1 stop bit, bez řízení datového toku. Dále je k procesoru přiveden pin pro reset, který je se inicializuje na pinu PA15.

4.2 Bluetooth

Pro vyzkoušení funkčnosti Bluetooth modulu CC2541EM bylo nejprve nutné čip naprogramovat. Texas Instruments nabízí ke stažení několik již hotových aplikací ve formátu .hex souborů. Tyto již přeložené aplikace lze nahrát přímo do čipu pomocí programu SmartRF Flash Programmer. Programování čipů a testování jejich aplikací je dále popsáno v kapitole 4.4.

Kromě již hotových aplikací, jsou ke stažení také vzorové projekty. Vzorové projekty jsou vytvořeny pro vývojové prostředí IAR. Pro jejich editaci je zapotřebí nainstalovat program IAR Workbench s kompilérem pro průmyslové čipy typu 8051. IAR nabízí volnou verzi se zkušební lhůtou 30-ti dní, což je pro úpravu vzorového projektu dostatečně dlouhá doba.

Pro převodníkovou desku jsem vybral vzorový projekt SerialBLEBridge [9], který funguje jako náhrada sériového komunikačního portu, ten Bluetooth Smart (Low Energy) nepodporuje. Tento projekt funguje tak, že po zapnutí se zařízení chová jako slave a master by mělo být zařízení, které se k němu připojí, což by měl být smartphone, tablet nebo počítač s USB Dongle. Roli, ve které zařízení komunikuje

zajišťuje GATT profil, může to být klient nebo server. S čipem se komunikuje pomocí sériového rozhraní UART které zajišťuje soubor *bluetooth.c*.

Projekt SerialBLEBridge umožňuje přeposílat data do hostitelského MCU (v tomto případě STM32F4 DISCOVERY) skrze GATT profil, což je zajištěno implementací simpleGATTprofile_bridge v projektu. Ke komunikaci se zde využívá jednoduchý profil Characteristic 3, na který je možno zapsat až 20-byte zprávu. Data k odeslání by měla být ve formátu: první bajt je délka zprávy a za ním následuje zpráva odpovídající této délce. Data, která pak přijdou po UART z CC2541 jsou ve formátu podle tabulky č. 4. Přijímaná data využívají profil Characteristic 4. Pro příjem je zapotřebí zapnout notifikace, data zde přijdou ve formátu: data[0]... data[LEN].

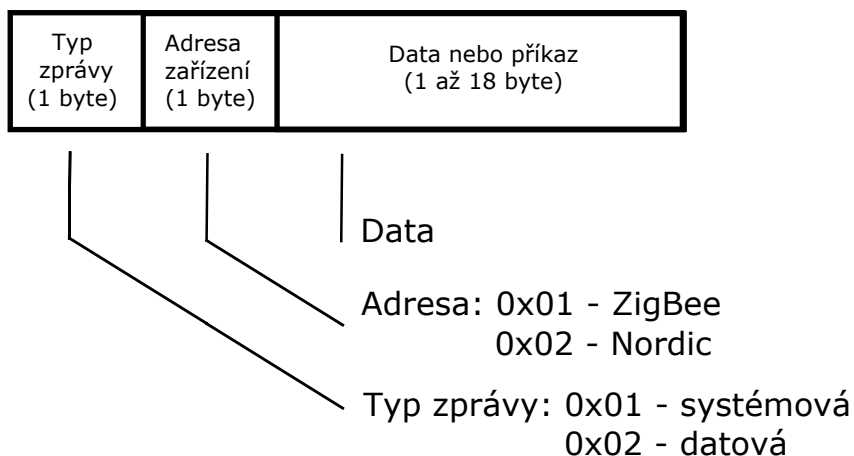
0xAD	0xAB	Data[0] ... Data[LEN]
------	------	-----------------------

Tabulka 4. Formát přichozích dat z CC2541 po UART

Při práci s tímto projektem jsem narazil na problém, že komunikace fungovala pouze jedním směrem a to pouze z čipu ke kterému jsem se připojil. Tento problém byl způsoben tím, že projekt SerialBLEbridge, který Texas Instruments nabízí ke stažení, obsahoval soubor simpleGATTprofile místo simpleGATTprofile_bridge a v tomto GATT profilu nebylo umožněno klientovi (zařízení které se připojuje k čipu) zapisovat na Characteristic 3. Problém jsme po komunikaci s pracovníkem Texas Instruments vyřešili a správný soubor mi byl zaslán.

Pro vyzkoušení komunikace nabízí Texas Instruments dva programy pro počítač, BLE Device Monitor a Btool verze 1.40.5. Oba programy používají CC2540 USB Dongle, který se po připojení do počítače chová jako COM port s vlastnostmi přenosu: přenosová rychlost 115200 bit/s, počet datových bitů je 8, žádný paritní bit, 1 stop bit, bez řízení datového toku. Pro vyzkoušení komunikace jsem si vybral program Btool, který umožňuje více možností než BLE Device Monitor a vypisuje jak vypadá komunikace pomocí HCI příkazů s CC2540USB Dongle po COM portu. Dále jsem také používal aplikaci LightBlue pro mobilní zařízení s operačním systémem iOS, pomocí které se lze jednoduše připojit k zařízení Bluetooth Smart, zapisovat data na Characteristic 3 k odeslání, číst přijímaná data z Characteristic 4. a měřit úroveň signálu bez potřeby CC2540 USB Dongle.

Datový protokol pro nastavení převodníkové desky, výběr komunikačního standardu (ZigBee nebo Nordic) a odesílání dat popisuje Obr.15. Formát jednotlivých příkazů popisuje tabulka č.5.



Obrázek 15. Datový protokol převodníkové desky

Typ zprávy	Adresa zařízení	Data nebo příkaz	Popis
0x01	0x01	0x01	Příkaz pro zjištění zda je zapnuté ZigBee
0x01	0x02	0x01	Příkaz pro zjištění zda je zapnutý Nordic
0x01	0x01	0x02	Příkaz pro zapnutí a inicializaci ZigBee
0x01	0x02	0x02	Příkaz pro zapnutí a inicializaci Nordic
0x01	0x01	0x03 XX..XX	Příkaz pro nastavení adresy XX..XX zařízení ke kterému se má ZigBee připojit
0x01	0x01	0x03 XX..XX	Příkaz pro nastavení adresy XX..XX zařízení ke kterému se má Nordic připojit
0x02	0x01	XX..XX	Příkaz pro odeslání dat XX..XX přes Zigbee
0x02	0x02	XX..XX	Příkaz pro odeslání dat XX..XX přes Nordic

Tabulka 5. Seznam příkazů převodníkové desky

Typ zprávy	Adresa zařízení	Data nebo příkaz	Popis
0xFF	0x01	0x01	Zpráva, že je zapnuté ZigBee
0xFF	0x01	0x02	Zpráva, že je vypnuté ZigBee
0xFF	0x01	0x03	Zpráva, že je chyba na ZigBee
0xFF	0x02	0x01	Zpráva, že je zapnutý Nordic
0xFF	0x02	0x02	Zpráva, že je vypnutý Nordic
0xFF	0x02	0x03	Zpráva, že je zapnuté ZigBee
0x01	0x01	XX..XX	Příchozí data XX..XX od Zigbee
0x01	0x02	XX..XX	Příchozí data XX..XX od Nordic

Tabulka 6. Seznam příchozích zpráv z převodníkové desky

4.3 ZigBee

Pro vyzkoušení funkčnosti ZigBee modulu bylo nejprve nutné čip CC2530 naprogramovat stejně jako u Bluetooth čipu CC2541 pomocí desky SmartRF05 a programu SmartRF Flash Programmer. Texas Instruments nabízí pro tento čip několik již hotových aplikací ve formě hex souboru. Pro komunikaci s autíčkem je zde použita aplikace Z-Stack Home 1.2.1, která je naprogramovaná i v čipu, který je umístěn v autíčku, se kterým bude komunikovat.

Pro komunikaci po standardu ZigBee [11] existují podle ZigBee Alliance dvě různá nastavení stack profilu a to ZigBee nebo ZigBee PRO. Tyto profily se rozlišují pomocí ID a zařízení s rozdílným nastavením profilu spolu nemohou komunikovat. Nastavení stack profilu, který se v tomto projektu pro komunikaci s autíčkem používá je ZigBee PRO a má ID = 0x02. Zařízení jsou adresovány pomocí 64-bit IEEE MAC adresy a 16-bit síťové adresy. Při nastavení stack profilu ZigBee PRO jsou zařízení v síti adresována náhodně a v případě konfliktu adres si zařízení se stejnou adresou vytvoří náhodně novou. Síť vytváří zařízení s nastavením koordinátora a ostatní zařízení se k němu připojují jako routery nebo koncová zařízení. Koordinátor může vytvořit síť na jednom z několika různých kanálů a nastavuje identifikátor sítě PAN ID. Převodníková deska by měla být v tomto případě koordinátorem a vytvářet síť na kanálu č. 14 s PAN ID = 0x0003.

Pro počáteční nastavení sítě je nejprve nutné inicializovat čip pomocí funkce *SPI_startup()*, poté odeslat příkaz s nastavením stack profilu ZigBee PRO a provést znovu startup. Poté se odešle příkaz s nastavením daného zařízení jako koordinátor, PAN ID=0x0003 a kanál číslo 14. Dále se odesílá příkaz REGISTER_REQUEST pomocí funkce *spiAppRegRequest()* pro registraci koncového bodu. Po tomto příkazu následuje příkaz START_REQUEST pomocí funkce *spiStartRequest()* po něm se čeká na jeho potvrzení, po kterém se může zahájit komunikace.

Formát paketu, pomocí kterého se mezi zařízeními komunikuje, popisuje tabulka č.5.

Octets: 1	2	1	2	2	2	5	Variable	2
Length Byte	Frame Control	Sequence number	Dest. PAN ID	Dest. Address	Source Address	Aux.Sec. Header	Frame payload	FCS

Tabulka 7. Datový paket ZigBee [4]

Komunikace mezi autíčky probíhá skrze funkci *spiSendDataRequest(DstAdr, len, DataToSend)* jejíž parametry jsou síťová adresa zařízení, na které se mají data odeslat, délka dat v bajtech a data k odeslání. Pokud chceme odeslat broadcast zprávu všem zařízením v síti, pak bude hodnota *DstAdr = {0xFF, 0xFF}*. Pro přijetí příchozích dat je zde funkce *znpProcessPoll(*data)*, které se jako parametr předává pointer na pole, do kterého chceme data uložit.

Pro ověření správné komunikace a její sledování lze použít program Packet Sniffer od Texas Instruments, který je s pomocí CC2531 USB Dongle schopný zachytávat jednotlivé pakety. Příklad komunikace po ZigBee mezi převodníkovou deskou a autíčkem popisuje Obr.16 z Packet Snifferu. Na obrázku jsou pakety po vytvoření sítě a broadcast paket, který zašle autíčko s informací o tom, že se připojilo

Pnbr RX 2

Time (ms) +12924

Length 10

Frame control field
Type Sec Pnd Ack.req PAN_compr
CMD 0 0 0 0

Sequence number 0xB4

Dest. PAN 0xFFFF

Dest. Address 0xFFFF

Beacon request

LQI 68

FCS OK

Coordinator vytvořil síť

Pnbr RX 3

Time (ms) +12927

Length 28

Frame control field
Type Sec Pnd Ack.req PAN_compr
BCN 0 0 0 0

Sequence number 0x36

Source PAN 0x0003

Source Address 0x0000

Superframe specification
BO SO F.CAP BLE Coord Assoc
15 15 15 0 1 1

GTS fields
Len Permit 0 0

Beacon payload
00 22 84 01 B9 CD 03 00
4B 12 00 FF FF FF 00

Stk_Prof 0x2

P.Ver 0x2

Rtr_Cap 0x0

Dev.Depth 0x1

Dev.Cap 0x1

Ext.PANID 0x00124B0003CDB381

LQI 120

FCS OK

Beacon Payload (NWK Layer Decoded)

Pnbr RX 4

Time (ms) +509

Length 21

Frame control field
Type Sec Pnd Ack.req PAN_compr
CMD 0 0 0 1

Sequence number 0xB5

Dest. PAN 0x0003

Dest. Address 0x0000

Source PAN 0xFFFF

Source Address 0x00124B00041A9E21

Association request
alt.coord FFD Power Idle RX Sec Allor.addr
0 1 1 1 0 1

LQI 68

FCS OK

Proces připojení autíčka

Pnbr RX 5

Time (ms) +13437

Length 5

Frame control field
Type Sec Pnd Ack.req PAN_compr
ACK 0 0 0 0

Sequence number 0xB5

LQI 123

FCS OK

Pnbr RX 6

Time (ms) +494

Length 18

Frame control field
Type Sec Pnd Ack.req PAN_compr
CMD 0 0 0 1

Sequence number 0xB6

Dest. PAN 0x0003

Dest. Address 0x00124B00041A9E21

Data request

LQI 65

FCS OK

Pnbr RX 7

Time (ms) +0

Length 5

Frame control field
Type Sec Pnd Ack.req PAN_compr
ACK 0 1 0 0

Sequence number 0xB6

LQI 123

FCS OK

Pnbr RX 8

Time (ms) +13934

Length 27

Frame control field
Type Sec Pnd Ack.req PAN_compr
CMD 0 0 0 1

Sequence number 0x37

Dest. PAN 0x0003

Dest. Address 0x00124B00041A9E21

Source Address 0x00124B0003CDB381

Short Addr Assoc Status
Short_addr Assoc_status
0xFF45 Successful

LQI 123

FCS OK

Pnbr RX 9

Time (ms) +13935

Length 5

Frame control field
Type Sec Pnd Ack.req PAN_compr
ACK 0 0 0 0

Sequence number 0x37

LQI 65

FCS OK

Pnbr RX 10

Time (ms) +334

Length 36

Frame control field
Type Sec Pnd Ack.req PAN_compr
DATA 0 0 0 1

Sequence number 0xB8

Dest. PAN 0x0003

Dest. Address 0xFFFF

Source Address 0xFF45

MAC payload
08 00 FF FF 45 FF 02 84 08 08 00 00 0D
BF 05 01 3C 05 64 6A 01 45 FF 6A 01

APS Payload
3C 05 64 6A 01 45 FF 6A 01

LQI 70

FCS OK

Pnbr RX 11

Time (ms) +15964

Length 36

Frame control field
Type Sec Pnd Ack.req PAN_compr
DATA 0 0 0 1

Sequence number 0x3A

Dest. PAN 0x0003

Dest. Address 0xFFFF

Source Address 0x0000

MAC payload
08 00 FF FF 45 FF 01 84 08 08 00 00 0D
BF 05 01 3C 05 64 6A 01 45 FF 6A 01

APS Payload
3C 05 64 6A 01 45 FF 6A 01

LQI 126

FCS OK

Autíčko zasílá paket s informací, že se připojilo a adresou

Obrázek 16. Pakety zachycené pomocí Packet Sniffer

4.4 Nordic

Čip Nordic nRF24L01 je k procesoru připojen pomocí rozhraní SPI. Při inicializaci se nastaví rozhraní Nordic pomocí struktury Interface, kde se nastaví způsob předávání přijatých a odesílaných dat. Dále se pomocí struktury *nordic_settings* nastaví třída, adresa a kanál zařízení. Samotná inicializace proběhne zavoláním funkce *rf_nordic_init(nordic_settings)*, které se jako parametr předá struktura s nastavením. Pro odeslání zprávy je potřeba zprávu uložit do datové struktury *Packet*. Paket se zprávou se pak předá funkci *add_packet_to_transmission_queue()* pro odeslání a odešle se po zavolání funkce *transmit_packets_in_queue()*. [14] Formát paketu s daty jejichž maximální délka je 28 bajtů popisuje tabulka č.6.

Byte 0	Byte 1	Byte 3	Byte 4	Byte 4 + length
SERVICE	LEN+CTR	APPL TYPE	SOURCE ADDR	PAYLOAD

Tabulka 7. Datový paket Nordic

Servisní bajt 0 určuje o jaký typ zprávy se jedná, jednotlivé typy zpráv popisuje tabulka č. 7.

Hodnota	Typ zprávy
0x01	Odeslání zprávy autíčku
0x0A	Žádost o data z autíčka
0x10	Reset zařízení
0x30 – 0x36	Příkazy pro Bootloader

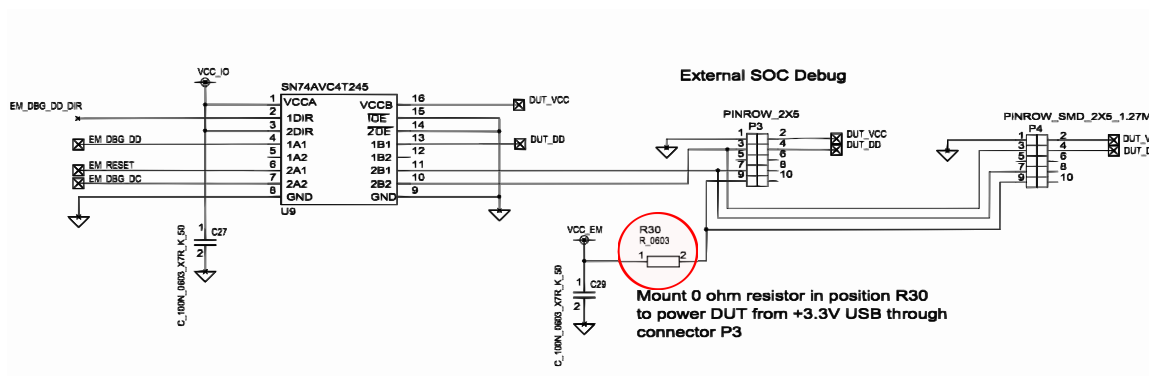
Tabulka 8. Popis možných hodnot servisního bajtu

Bajt LEN+CTR udává číslo paketu a jeho délku, kde první 3 bity jsou číslo a zbývající délka zprávy - PAYLOAD. Bajt APPL udává typ platformy v tomto případě autíčko s hodnotou 0xAA. Bajt SOURCE ADDRESS obsahuje adresu odesílatele zprávy. Od 4 bajtu začíná samotná zpráva, která má maximální délku 28 bajtů.

Při ověřování komunikace s čipem jsem narazil na problém, že s čipem nešlo komunikovat po SPI. Při zjišťování, proč čip nekomunikuje, jsem vyzkoušel změřit signál na oscilátoru, při čemž jsem zjistil, že oscilátor čipu nefunguje. Důvodem byl vadný čip nRF24L01, který jsem vyměnil. Při kontrole komunikace se s novým čipem se mi podařilo úspěšně vyčíst hodnoty z registru. Komunikaci s autíčkem pomocí rozhraní Nordic se mi však zprovoznit nepodařilo. Pravděpodobně se jedná o chybu v sestavování spojení s autíčkem, bohužel již nemám dostatek času pro zjištění této chyby.

4.5 Programování a testování čipů CC2541 pro Bluetooth a CC2530 pro ZigBee na vývojové desce SmartRF05 Rev. 1.8.1

Čip na vývojovém modulu lze naprogramovat snadno pouze připojením k desce SmartRF05 pomocí dvou konektorů. Vývojový modul má na konektory, kterými je k desce SmartRF05 připojen, vyvedeny i další piny, které jsou připojeny k displeji, tlačítkům, diodám nebo RS-232 konektoru. To umožňuje snadné vyzkoušení několika aplikací bez nutnosti návrhu vlastní desky. Pro programování a debugování čipu na vlastní desce je potřeba mít vyvedené piny DC, DD, RESET a GND. Tyto piny se připojí k desce SmartRF05 na konektor pro externí programování a debugování. V případě, že deska není napájena, pak ji lze napájet z konektoru pro externí programování, ale je potřeba na desce SmartRF05 připojit nulový odpor R30 jako propojku pro přivedení 3,3V ke konektoru (Obr. č.17).



Obrázek 17. Schéma s piny pro externí programování a debugování na SmartRF05 Rev 1.8.1 [8]

Pro naprogramování čipu je nutné mít propojené jumperry na pinech 19-20 pro DBG_DC, 21-22 pro DBG_DD na řadě pinů P1. Dále je potřeba propojit piny 35-36 na řadě pinů P10. Pro nahrání programu do čipu je zapotřebí mít nainstalovaný program SmartRF Flash Programmer od Texas Instruments.

Pro otestování komunikace s čipem po SPI je zapotřebí propojit jumperry piny 23-24 pro CS, 25-26 pro MISO, 27-28 pro MOSI a 29-30 pro SCLK.

Pro otestování komunikace po UART je potřeba propojit piny 5-6 pro UART_RX, 7-8 pro UART_TX na řadě pinů P1 v případě, že není nastavena kontrola toku dat. Pro případ, že je zapnutá kontrola toku je zapotřebí propojit i piny 1-2 UART_RTS a 3-4 UART_CTS na řadě pinů P1.

4.6 Aplikace Bluetooth Converter Board

Pro psaní aplikace pro připojení a ovládání převodníkové desky pomocí CC2540 USB Dongle jsem zvolil jazyk Java kvůli jeho multiplatformnosti. Pro komunikaci s COM portem bylo potřeba stáhnout a importovat knihovnu RXTX, protože základní JDK tento komunikační standard nepodporuje. Sun podporuje knihovnu JavaComm, ale neoficiální knihovna RXTX je podporována více platformami a měla by fungovat lépe. Komunikace mezi aplikací a probíhá pomocí HCI příkazů. [15] Formát příkazu vypadá takto: první bajt je typ zprávy, další dva bajty jsou Opcode, poté jeden bajt délka zprávy a další bajty jsou samotná data.

Type (1 byte)	Opcode (2 bytes)	Length (1 byte)	Data (length)
------------------	---------------------	--------------------	------------------

Tabulka 9. Formát příkazů pro komunikaci CC2540 USB Dongle [15]

Způsob řazení bajtů je Little-endian, na místo s nejnižší adresou se uloží nejméně významný bajt LSB a jako poslední se uloží nejvíce významný bajt MSB.

Pro zahájení komunikace je nejprve třeba odeslat inicializační příkaz *GAP_DeviceInit*, na který přijde odpověď, že inicializace proběhla v pořádku s informací o adrese zařízení, identifikačním klíči IRK a podpisujícím klíčem CSRK. Tím by měl být CC2541 USB Dongle inicializovaný. Dále je potřeba nalézt Bluetooth zařízení převodníkové desky, pro to je příkaz *GAP_DeviceDiscoveryRequest* pro skenování dostupných zařízení, na který by měla přijít potvrzovací zpráva a adresy nalezených zařízení. Bluetooth převodníkové desky má veřejnou adresu : 84:DD:20:E6:9D:E7 a připojíme se k němu pomocí příkazu *GAP_EstablishLinkRequest*, po úspěšném připojení by měla přijít potvrzovací zpráva s adresou zařízení ke kterému jsme se připojili. Pro funkčnost preposílání dat je ještě potřeba povolit notifikace pro Characteristic 4. v GATT profilu. To se provede odesláním hodnoty 01:00 na handler 0x2F pomocí příkazu *GATT_WriteCharValue*, po jeho povolení by měl Dongle přijímat data ze zařízení, ke kterému je připojen. Odesílání dat probíhá zápisem až 20-ti bajtové zprávy na Characteristic 3. V GATT profilu pomocí příkazu *GATT_WriteCharValue* s handlerem 0x2B. Formát používaných příkazů popisuje tabulka č.6.

Formát přijímaných zpráv z Dongelu popisuje tabulka č.7.

Název příkazu	Typ	Opcode	Data length	Data
<i>GAP_DeviceInit</i>	0x01	0xFE00	0x26	08:05:00:00:00:00:00:00:0 0:00:00:00:00:00:00:00:00 :00:00:00:00:00:00:00:00: 00:00:00:00:00:00:00:00:0 0:01:00:00:00
<i>GAP_DeviceDiscovery Request</i>	0x01	0xFE04	0x03	03:01:00
<i>GAP_EstablishLink Request (kde XX je adresa zařízení)</i>	0x01	0xFE09	0x09	00:00:00:XX:XX:XX:XX:XX:XX
<i>GATT_WriteCharValue (Povolení notifikací z Characteristic 4.)</i>	0x01	0xFD92	0x06	00 00 2F 00 01 00
<i>GATT_WriteCharValue (Odeslání dat na Characteristic 3.)</i>	0x01	0xFD92	0x03 +data	00 00 2B XX..XX

Název příkazu	Typ	Event Code	Data length	Data
<i>GAP_DeviceInitDone</i>	0x04	0xFF	0x2C	00:06:00:(6 byte Adress): (2byte data packet length): (16 byte IRK): (16 byte CSRK)
<i>GAP_DeviceDiscovery Done</i>	0x04	0xFF	0xFF	01:06:00:(NumDevices): 00:(Address Type): (Adress)
<i>GAP_EstablishLink</i>	0x04	0xFF	0x13	05:06:00:00:(Adress): 00:00:50:00:00:00:D0:07:00
<i>ATT_HandleValue Notification</i>	0x04	0xFF	0x08+ data	00:00:00:04:00:2E:XX:XX..XX

Po spuštění aplikace se objeví okno (Obr.17), ve kterém v se v horní části textově zobrazuje stav připojení. Pod tímto textem je výběr z nalezených COM portů, kde by měl uživatel vybrat COM port, ke kterému je připojen CC2540 USB Dongle. Pro připojení se k Dongle je zde tlačítko „Connect to dongle“. V případě, že připojení k Dongle proběhlo v pořádku, pak se může uživatel připojit k převodníkové desce pomocí tlačítka „Connect“. Pokud je připojení úspěšné v horní části okna se objeví text: „Connected OK“, pokud ne, tak se objeví dialogové okno: „Connection was not established! Try it again!“. Neúspěšné připojení může být způsobené tím, že převodníková deska není zapnutá nebo je k desce již připojené jiné zařízení. Po úspěšném připojení se k převodníkové desce může uživatel inicializovat a zapnout buď ZigBee nebo Nordic. Po inicializaci se na Převodníkové desce rozsvítí zelená led dioda k danému rozhraní a zaškrtně se políčko „ON“ v rámečku ZigBee nebo Nordic. V případě, že dojde k chybě zaškrtně se políčko „Error“. Pod rámečky ZigBee a Nordic lze vybrat s jakým zařízením chce uživatel komunikovat, nastavení po zapnutí je ZigBee. Dále je zde textové pole, do kterého může uživatel napsat zprávu v hexadecimálním formátu zakončenou ukončovacím znakem „0A“, kterou chce odeslat do autíčka skrze předem vybraný komunikační standard. Přijímané zprávy se zobrazují v textovém poli pod nápisem „Received data:“. Pro zjednodušení jsou zde příkazy pro rozjezd autíčka a jeho zastavení předdefinované pod tlačítky „Start car“ a „Stop car“.



Obrázek 17. Java aplikace pro komunikaci s převodníkovou deskou

5 Závěr

V první části této práce jsem vybral čipy CC2541 pro Bluetooth, CC 3100 pro Wi-Fi, pomocí kterých se uživatel připojuje k převodníkové desce. Pro připojení desky k autíčku jsem vybral čipy nRF24L01 pro Nordic a CC2530 pro ZigBee. Následně jsem podle parametrů těchto čipů vybral další potřebné součástky pro zapojení čipů a spínaný zdroj s dostatečnou rezervou maximálního proudového odběru, tak aby byl schopný pokrýt i proudové špičky. Poté jsem navrhl schéma celého zapojení převodníkové desky a plošný spoj, který se poté nechal vyrobit.

Po osazení desky součástkami jsem narazil na několik chyb v návrhu desky. Byl zde problém se zdrojem, u kterého byla chyba v zapojení, chyběli konektory pro programování čipů Bluetooth a ZigBee.. Všechny tyto chyby byly opraveny na desce prvního návrhu a jsou popsány v kapitole 3. Další chybou byly špatně umístěné konektory pro připojení DISCOVERY kitu k převodníkové desce. To jsem vyřešil pomocí propojovací desky s konektory, která tuto rozdílnou vzdálenost mezi konektory vyrovnává.

Před programováním samotného čipu STM32F4 na DISCOVERY kitu jsem musel naprogramovat čipy pro rozhraní Bluetooth a ZigBee. Pro Bluetooth jsem použil vzorový projekt od Texas Instruments, ve kterém jsem musel provést úpravy pro jeho zprovoznění. Do čipu pro ZigBee jsem nahrál již hotový program od Texas Instruments. Poté jsem naprogramoval obsluhu čipu Bluetooth po rozhraní UART a ZigBee a Nordic po rozhraní SPI. Komunikace po rozhraní Bluetooth a odesílání příkazů autíčku po rozhraní ZigBee funguje spolehlivě. Obsluhu Nordic čipu jsem ověřil na úrovni vyčítání dat z registru, ale komunikace s autíčkem nefunguje. Na realizaci programu pro obsluhu Wi-Fi čipu jsem z důvodu problémů, které se vyskytly při oživování ostatních čipů neměl dostatek času.

V poslední části této práce jsem vytvořil Java aplikaci pro počítač s CC2540 USB Dongle pro nastavení převodníkové desky a komunikaci s autíčky.

6 Reference

- [1] Drotek. DISCOVERY KIT WITH STM32F4 [cit. 4.4. 2015] Dostupné z : <http://www.drotek.fr/shop/en/home/127-discovery-kit-stm32f407.html>
- [2] DATASHEET: STM. STM32F4 DISCOVERY. [online]. [cit. 4.4. 2015]. Dostupné z: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/data_brief/DM00037955.pdf
- [3] DATASHEET: Texas Instruments. CC3100 SimpleLink Wi-Fi Network Processor, Internet-of-Things Solution for MCU Applications (Rev. D). [online]. [cit. 4.4. 2015]. Dostupné z: <http://www.ti.com/lit/ds/symlink/cc3100.pdf>
- [4] DATASHEET: Texas Instruments. 2.4-GHz Bluetooth® low energy and Proprietary System-on-Chip (Rev. D). [online]. [cit. 4.4. 2015]. Dostupné z: <http://www.ti.com/lit/ds/symlink/cc2541.pdf>
- [5] DATASHEET: Texas Instruments. CC2530 (Rev. B). [online]. [cit. 4.4. 2015]. Dostupné z: <http://www.ti.com/lit/ds/symlink/cc2530.pdf>
- [6] DATASHEET: Nordic Semiconductors. nRF24L01 Product Specification. [online]. [cit. 4.4.2015]. Dostupné z: http://www.nordicsemi.com/eng/nordic/download_resource/8041/1/9437913
- [7] DATASHEET: Texas Instruments. LMR10515 SIMPLE SWITCHER 5.5Vin, 1.5A Step-Dwn Vltg Reg in SOT-23 & LLP (Rev. C). [online]. [cit. 4. 4. 2015]. Dostupné z: <http://www.ti.com/lit/gpn/lmr10515>
- [8] DATASHEET: Teks Instruments. SmartRF05 Evaluation Board User's Guide. [online]. [cit. 8.4.2015]. Dostupné z: <http://www.ti.com/lit/ug/swru210a/swru210a.pdf>
- [9] REFERENCE DESIGN: Texas Instruments. Bluetooth Smart Keyfob Reference Design Schematic. [online]. [cit. 8.4.2015]. Dostupné z: <http://www.ti.com/lit/pdf/tidr191>
- [10] Texas Instruments. SerialBLEBridge. [online]. [cit. 8.4.2015]. Dostupné z: <http://processors.wiki.ti.com/index.php/SerialBLEbridge>
- [11] Texas Instruments. Z-Stack Developer's Guide. [online]. [cit. 8.4.2015]. Dostupné z: http://software-dl.ti.com/lprf/z_stack
- [12] Texas Instruments. CC2541 Evaluation module kit. [online]. [cit. 3.5.2015]. Dostupné z: <http://www.ti.com/tool/CC2541EMK>

- [13] Texas Instruments. CC2540 USB Evaluation module kit. [online]. [cit. 3.5.2015]. Dostupné z: <http://www.ti.com/tool/CC2540EMK-USB>
- [14] Ivo Herman, Dan Martinec. Interní dokumentace Slotcar platooning.
- [15] Texas Instruments. CC2540/41 Bluetooth® Low Energy Software Developer's Guide v1.3.2 [online]. [cit. 3.5.2015]. Dostupné z: <http://www.ti.com/lit/ug/swru271f/swru271f.pdf>
- [16] Tenamp Slot Cars. Carrera 27295 - Ford Capri RS3100 [online]. [cit. 10.5.2015]. Dostupné z: <http://www.tenamp.com/carrera/carrera-27295-ford-capri-rs3100.html>

Seznam Zkratek

DPS – Deska Plošných Spojů

ARM - Advanced RISC Machine

UART - Universal Synchronous / Asynchronous Receiver and Transmitter

SPI – Serial Peripheral Interface

HCI – Host Controller Interface

GAP – Generic Access Profile

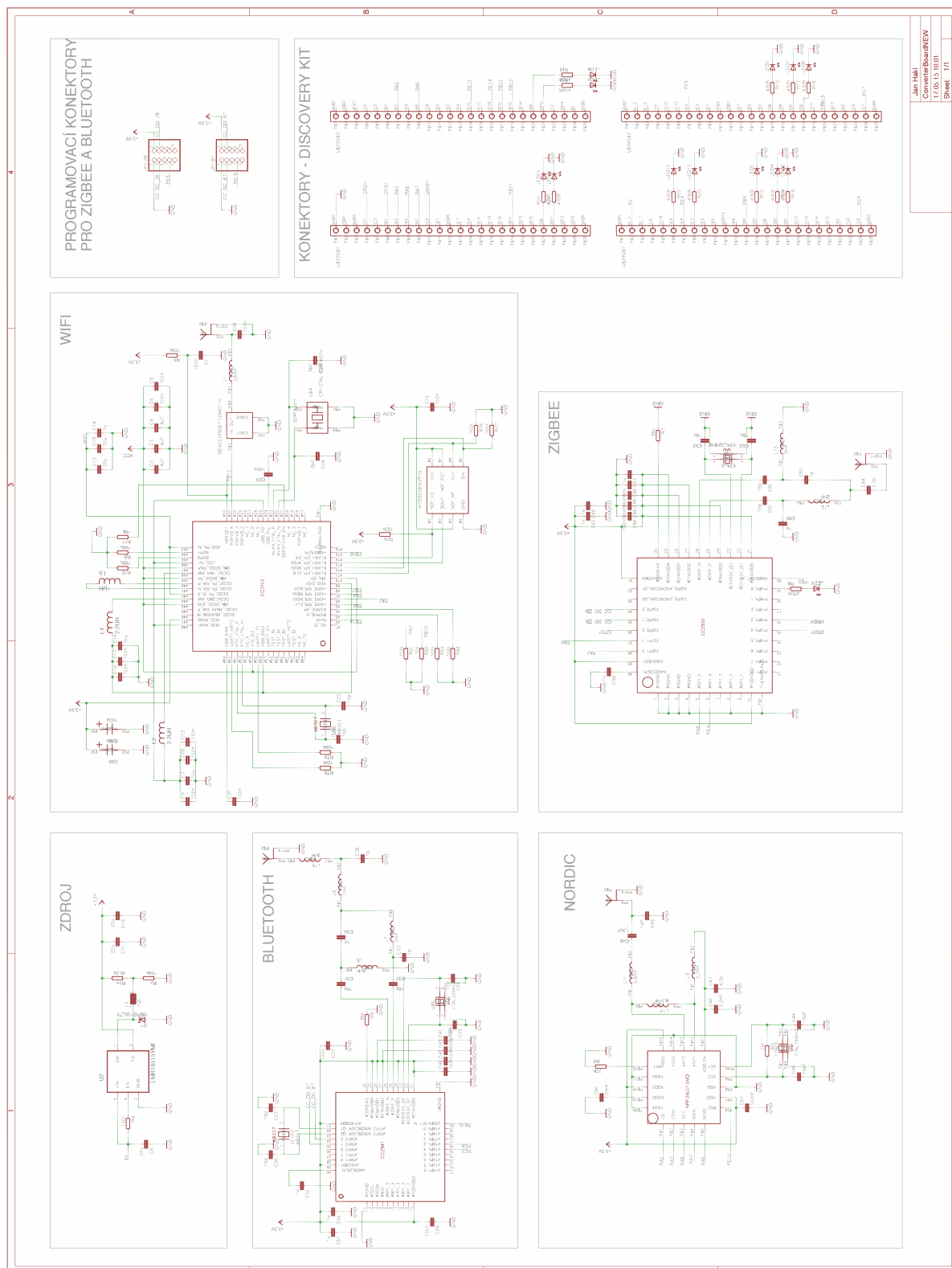
GATT – Generic Attribute Profile

MSB – Most Significant Byte

LSB – Least Significant Byte

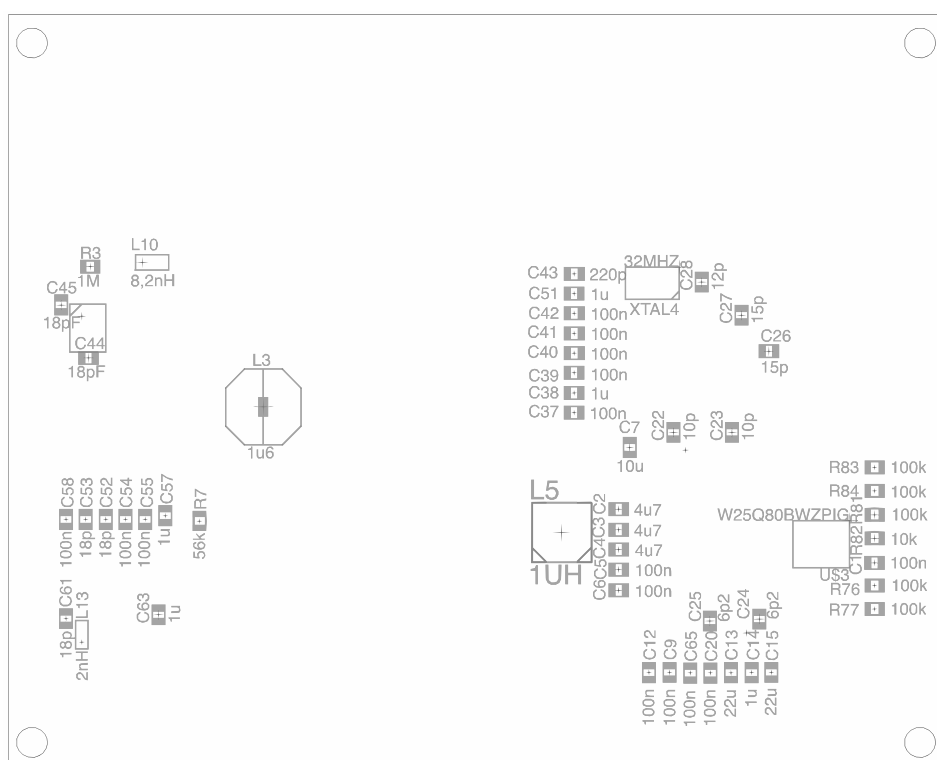
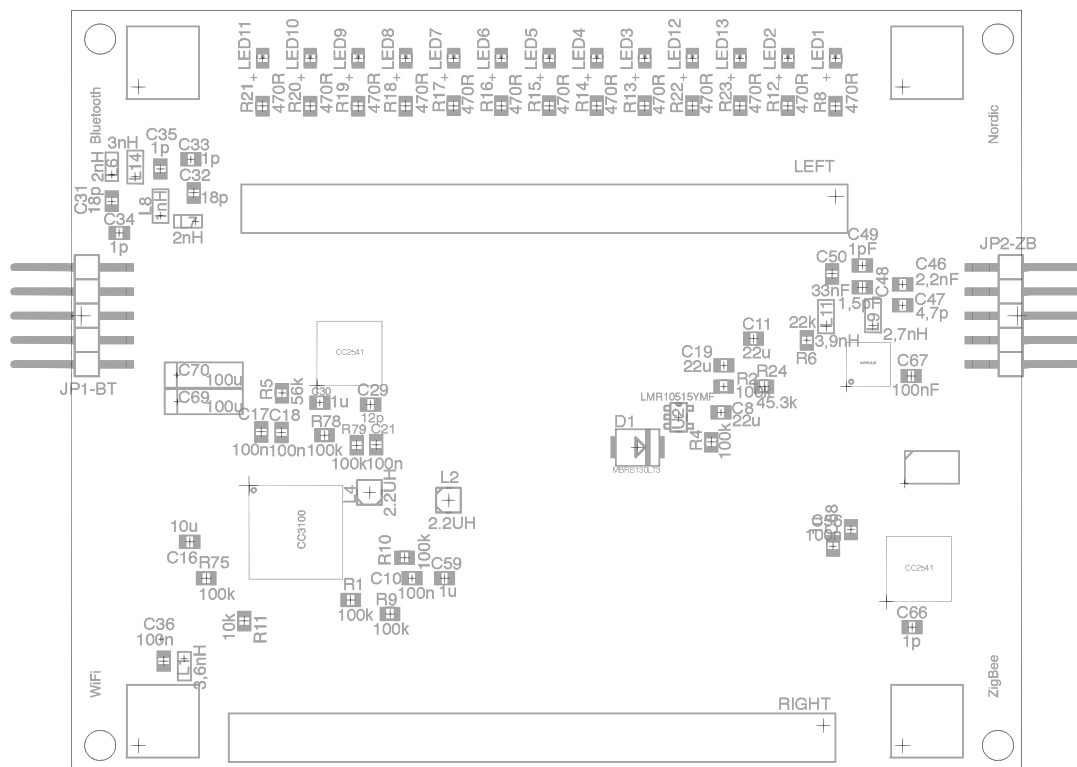
Příloha A

Kompletní schéma převodníkové desky



Příloha B

Rozmístění součástek na horní a dolní straně převodníkové desky



Příloha C

Obsah přiloženého CD

- Bakalářská práce ve formátu PDF
- Návrh první i druhé verze schématu a plošného spoje ve složce /HW
- Zdrojové kódy k programu pro převodníkovou desku ve složce /SW/ConverterBoard
- Zdrojové kódy Java aplikace pro ovládání převodníkové desky ve složce SW/Application