ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ, FAKULTA ELEKTROTECHNICKÁ, KATEDRA ŘÍDICÍ TECHNIKY

Bakalářská práce

Bezdrátové řízení motorových jednotek modelové železnice

2009 Václav Rychnovský

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Václav Rychnovský**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Bezdrátové řízení motorových jednotek modelové železnice**

Pokyny pro vypracování:

Smyslem práce je navrhnout elektrické zapojení a zprovoznit komunikační protokol, který bude umožňovat jak povelování motorových jednotek modelové železnice, tak zpětný přenos informací o aktuálních veličinách z těchto jednotek do nadřízeného počítače. Jedná se především o řízení napětí, momentu, případně otáček motoru a měření hodnot odometrických veličin a momentu.

1. Prostudujte dostupné protokoly vhodné pro povelování jednotek, zaměřte se především na možnost obousměrné a bezdrátové komunikace, které jsou z důvodů požadavků zadání a mobility jednotek pravděpodobně nejvhodnější.
2. Zvolte vhodný hardware, nejspíše založený na vhodném mikrokontroléru, který bude obstarávat řízení a komunikaci na straně motorových jednotek. Hlavním omezením jsou rozměry vestavného prostoru.
3. S využitím dostupných softwarových komponent realizujte jednoduchý protokol pro komunikaci s nadřízeným počítačem.
4. Zkuste navrhnout vhodné řešení pro odometrické měření polohy.

Seznam odborné literatury:

TinyOS: open-source operating system designed for wireless embedded sensor networks, http://tinyos.net/
ZigBee Alliance home page, http://www.zigbee.org/

Vedoucí: Ing. Pavel Píša

Platnost zadání: do konce zimního semestru 2009/10

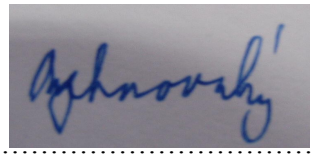
prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 27. 2. 2009

Prohlášení

Prohlašuji, že jsem svou bakalářskou / diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne 8.7.2009 

…….
podpis

Abstrakt

Hlavní záměr této práce je volba způsobu, jak řídit motorové jednotky modelové železnice pomocí transceiveru na palubě motorové jednotky, který komunikuje s nadřazeným řídicím transceiverem. Jako nejperspektivnější se ukázal protokol ZigBee, konkrétně Z-Stack, který je implementovaný firmou Texas Instruments na procesoru MSP430 s transceiverem CC2520. Tato práce popisuje ZigBee protokol, transceiver CC2520, Z-Stack a jeho jednoduchou úpravu pro činnost na desce, kterou jsem navrhl. Obsahuje také metody pro optické měření pohybu optickou myší.

Abstract

Main intention of this work is choice of method, how to control traction unit of model railroad by way of transceiver on the board of traction unit, which communicates with master control unit. The most promising is ZigBee protocol, specifically Z-Stack, which is implemented by Texas Instruments on processor MSP430 with transceiver CC2520. This work describes ZigBee protocol, CC2520 transceiver, Z-Stack package and its simple adaptation for proper work on board, which I designed. It contains methods for optical measuring of motion with optical mouse too.

Poděkování

Děkuji za spolupráci
vedoucímu práce Ing. Pavlu Píšovi
zadavateli Ing. Květoslavu Beldovi, Ph. D.
panu Guiseppe Mazzolenimu za zapůjčení hardwaru

Obsah

1 Zadání práce....4
1.1 Požadavky na řešení....4
1.2 Požadavky týkající se motorové jednotky....4
1.3 Rozsah práce....4
2 Úvod do řízení modelové železnice....4
2.1 Běžně používané způsoby řízení....4
2.1.1 Řízení rychlosti pomocí velikosti napětí mezi kolejnicemi....4
2.1.2 Řízení rychlosti pomocí velikosti napětí a oddělených kolejových úseků....4
2.1.3 Řízení rychlosti pomocí standardu DCC....5
2.2 Nekonvenční použitý způsob řízení....5
2.2.1 Bezdrátové řízení....5
2.2.2 Optické snímání pohybu MJ....6
3 Seznámení se ZigBee protokolem....6
3.1 Zkratky používané v popisu protokolu....6
3.2 Představení....7
3.3 Fyzická vrstva....8
3.3.1 Vysílací část....8
3.3.2 Interface....9
3.3.3 Rámec fyzické vrstvy....10
3.4 MAC vrstva....10
3.4.1 Úvod....10
3.4.2 Interface....10
3.4.3 Beacon a non-beacon síť....11
3.4.4 MAC datový rámec....12
3.4.5 Potvrzovací rámec....13
3.4.6 Řídicí rámec(command frame)....13
3.4.7 Beacon rámec (synchronizace posílání paketů)....13
3.5 Síťová vrstva....14
3.5.1 Úvod....14
3.5.2 Interface....14
3.5.3 Rámce síťové vrstvy....16
3.6 Aplikační vrstva....16
3.6.1 Úvod....16
3.6.2 APS (Application support sublayer)....16
3.6.3 Rámec APS....17
3.6.4 AF (Application framework)....17
3.6.5 ZDO (ZigBee device objects)....18
4 Hardwarové řešení....18
4.1 Úvod....18
4.2 Výběr obvodů Texas Instruments....18
4.3 Optický snímač polohy....19
4.4 CC2520....19
4.4.1 Seznámení s obvodem....19
4.4.2 Externí součástky nutné pro činnost obvodu....20
4.4.3 Registry/paměť a instrukce pro práci s nimi....22
4.5 MSP430F2x1x....22
4.6 DRV 8800....23
4.7 Použité zapojení....24
4.7.1 Propojení mikrokontroléru a transceiveru....24

4.7.2 Volba antény....25
4.7.3 Propojení ovladače motorku a optického senzoru s mikrokontrolérem....27
4.7.4 Senzor optické myši....27
5 Softwarové řešení....27
5.1 Doprovodný software Z-Stack....27
5.1.1 Úvod....27
5.1.2 Implementace ZigBee protokolu....28
5.1.3 Referenční aplikace....28
5.1.4 SmartRF Studio....29
5.1.5 Z-Tool....29
5.2 Vlastní program....29
5.2.1 Kompilační parametry....29
5.2.2 Úpravy v implementaci Z-Stack od TI....29
5.2.3 Změny v aplikaci na straně koncového uzlu....30
5.2.4 Změny v aplikaci na straně koordinátoru....30
6 Závěr....31
7 Zdroje dalších informací....31

1 Zadání práce

1.1 Požadavky na řešení

Cílem práce je navrhnout vhodný způsob pro řízení více motorových jednotek (dále MJ) modelového kolejiště současně. Současně má být i měřena poloha MJ v kolejišti. Informace o poloze se bude předávat řídicí jednotce celého kolejiště, která má plnou kontrolu nad nezávislým pohybem všech MJ. Řídicí jednotka má být schopna komunikace s osobním počítačem, který bude v konečné verzi realizovat vlastní řídicí algoritmus pro modelové kolejiště.

1.2 Požadavky týkající se motorové jednotky

Řízeným objektem je v této práci myšlen model motorové jednotky (lokomotiva, článková tramvaj, vícenásobná jednotka) v měřítku H0 (1:87) poháněný stejnosměrným motorkem s permanentními magnety napájeným z pojezdových kolejnic. Typické napájecí napětí je 9 V a proud do 100 mA pro jednu jednotku. Volné prostory pro desku plošných spojů s komunikačním rozhraním uvnitř MJ jsou velmi omezené. Při použití v modelu osobního vagónu, který bývá nejprostornější, se jedná o šířku do 30 mm, délku do 150 mm. Výška volného prostoru bývá do 20 mm.

1.3 Rozsah práce

Tato práce se soustředí na implementaci části přenosového řetězce mezi MJ a řídicí jednotkou kolejiště, která bude předávat a vysílat zpět data z nadřazeného počítače. Současně se zaměřuje na způsob měření polohy MJ v kolejišti, který je vhodný k použití s tímto přenosovým řetězcem. První část práce představuje komerčně používané způsoby řízení modelového kolejiště a uvádí proč je nelze použít. Následuje kapitola volby nekonvenčního bezdrátového řízení kolejiště – ZigBee protokol, který splňuje zadaná kritéria a navíc je univerzálně použitelný i v jiných aplikacích. Další část se věnuje zevrubnému popisu vlastního ZigBee protokolu s popisem používaných zkratek a významem jednotlivých vrstev ISO/OSI modelu. Část věnovaná hardwarovému řešení podává zprávu o výběru vhodného hardwaru a popisuje klíčové obvody vybrané k činnosti na palubě MJ. Těžištěm další kapitoly je softwarové řešení úlohy, které vychází ze softwaru poskytovaného spolu s použitým hardwarem. Poslední kapitolou je výčet dosažených výsledků při oživování hardwaru a zhodnocení výstupů práce.

2 Úvod do řízení modelové železnice

2.1 Běžně používané způsoby řízení

2.1.1 Řízení rychlosti pomocí velikosti napětí mezi kolejnicemi

Nejstarším, nejjednodušším a nejběžnějším způsobem řízení je změna velikosti napájecího napětí mezi kolejemi. všechny MJ se tak pohybují stejnou rychlostí a stejným směrem na celém kolejišti. Platí to však pokud by všechny jednotky byly identické. Obecně se každá MJ pohybuje jinou rychlostí. Řízení více jednotek je tedy nemožné. Měření polohy MJ se neprovádí.

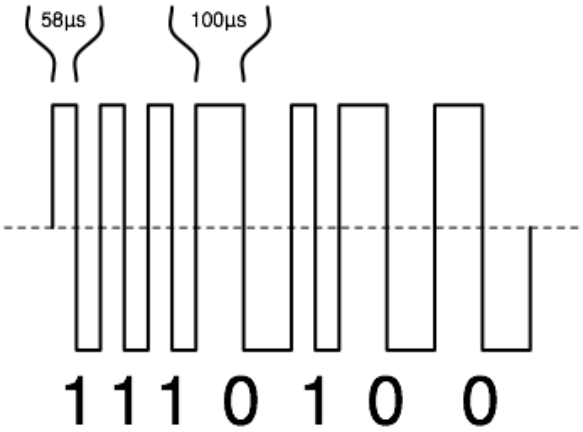
2.1.2 Řízení rychlosti pomocí velikosti napětí a oddělených kolejových úseků

Rozdělením kolejiště na několik menších úseků vzniká kolejiště s možností měnit rychlost a směr MJ téměř nezávisle na ostatních MJ. Důvodem pro slovo „téměř“ je způsob přejezdu mezi úseky. Musí existovat předávací mechanismus zjišťující jednak číslo úseku s MJ a jednak řešení konfliktů při přejetí z jednoho úseku do druhého. Najíždí-li MJ do nového úseku, musí být tento úsek již rezervován právě pro tuto MJ. MJ je totiž obvykle napájena ze všech svých náprav. Kdyby se v novém úseku náhodou nacházela jiná MJ, tak pojede stejným směrem, za předpokladu, že by nezkratovala další úsek. Pro regulérní jízdu N jednotek je potřeba minimálně 2N traťových úseků. Zjišťování polohy MJ je také v běžném provedení příliš hrubé – traťové úseky bývají dlouhé v násobcích délky samotné MJ. Pokud bychom požadovali plně nezávislé řízení, bylo by nutné zkrátit úseky na vzdálenost mezi nápravou a spřáhlem, což by vedlo ke stovkám spínačů, drátů, senzorů a přerušení

kolejí. Toto řešení na komerční bázi v oblasti modelové železnice pochopitelně neexistuje. Další nevýhodou i při rozumné délce úseků je nutnost modifikovat stávající trať – izolovat úseky, instalovat magnetické, či jiné snímače a všechno ovládat a vyhodnocovat. Nepopíratelnou výhodou by však bylo automatické, přesné a spolehlivé určení polohy MJ, které žádný jiný systém nenabízí.

2.1.3 Řízení rychlosti pomocí standardu DCC

Jako jediné řízení, které by splňovalo požadavek nezávislého řízení více MJ, lze označit právě DCC (Digital Command Control – řízení pomocí digitálního povelování). Pracuje na principu přijímače spolu s dekodérem signálu na palubě MJ, který sleduje napájecí napětí a vyhodnocuje změnu jeho polarity. Pokud se polarita mění rychleji (např. každých 58 µs) vyhodnotí ji jako logickou 0, pokud se polarita mění pomaleji (každých 100 µs) vyhodnotí ji jako logickou 1. Přijatou sekvenci bitů pak dekodér porovná s uloženým identifikačním číslem jednotky a pokud se shoduje, tak přijme a splní



Obr 1: Princip DCC modulace

zaslanou zprávu. V takovém kolejišti nelze ovšem provozovat MJ bez usměrňovače, neboť střední hodnota napájecího signálu v kolejišti je nulová. Toto lze však i pomocí DCC vyřešit. Vyhodnocovat pouze délku setrvání napájení buď pouze při kladné nebo záporné polaritě. Druhá polovina periody bude doplňkem do pevné periody pulzu. Střední hodnota napájení je nenulová a lze tak provozovat MJ bez dekodéru stejnou rychlostí. Výhodou tohoto řešení je funkčnost bez nutnosti investovat do úpravy vlastní trati. Ovšem hlavní nevýhodou je stále jednosměrnost komunikace. Modulaci napájecího napětí zde může provádět pouze napájecí zdroj. Informace o poloze či rychlosti MJ tak nelze pomocí DCC získat.

2.2 Nekonvenční použitý způsob řízení

2.2.1 Bezdrátové řízení

Jako další způsob řízení lze jmenovat řízení bezdrátové. Pravé bezdrátové řízení MJ na komerční bázi, pokud je mi známo, neexistuje. Komerčně se pouze vyrábějí bezdrátové ovladače k DCC napájecím zdrojům [17], což vlastně bezdrátové řešení není. Plně bezdrátové řešení je pouze bezdrátový transceiver na palubě MJ spolu se stacionárním transceiverem realizujícím výpočet akčního zásahu nebo napojeným na jiné zařízení tento výpočet realizující např. PC. Další výhodou je to, že lze řízení použít na stávající trať pouze s úpravou MJ. Stávající způsob řízení může být zachován obdobně jako MJ bez DCC dekodéru. Hlavní nevýhodou je zejména spolehlivost spojení, která je vždy o něco menší než v případě metalického propojení. Výhody bezdrátového řízení však jasně převažují. Navíc je to řešení univerzální, které lze použít i pro řízení jiných objektů, což se o DCC říci nedá. Předpokladem je ovšem použití již existujícího a osvědčeného standardu, kterých v poli bezdrátových aplikaci existuje celá řada. Jako nejvýhodnější se jeví standard IEEE 802.15.4 a jeho nadstavba vytvořená ZigBee Alliance známá pod souhrnným názvem ZigBee. Vykazuje nad jinými standardy parametry, které se nejvíce DCC přibližují, i když je samozřejmě o dost složitější a univerzálnější. Dále je to

standard velice perspektivní a moderní, ale prověřený mnoha aplikacemi s mírou rizika daleko větší než je řízení několika MJ modelové železnice.

2.2.2 Optické snímání pohybu MJ

Pro dodržení nezávislého určení pohybu je třeba mít toto měření zabezpečeno opět z MJ. Je tedy jasné, že musí jít o snímač bezkontaktní. Nejdostupnější řešení na komerční bázi je senzor optické myši. Umožňuje zcela bezkontaktní měření polohy na rovných površích s běžně dosahovanou přesností kolem 0.25 mm. Jeho největší nevýhodou je špatná funkčnost při změně vzdálenosti od podložky. Uvádí se, že funkčnost je zaručena pouze v rozmezí ±0.1 mm od střední hodnoty. Nutnou úpravou tratě je v tomto případě zakrytování pražců, které je v případě výhybek problematické. Další možností je magnetický snímač polohy. Využívá magnetický proužek s podélně se střídajícími magnetickými póly. Existují integrované kodéry [13] disponující přesností v řádu mikrometrů, které realizují celé měření v jednom pouzdře spolu s pokročilým komunikačním rozhraním. Bohužel tento způsob lze realizovat pouze na rovné trati. V zatáčkách MJ vybočuje a navíc magnetická páska nejde ohnout v požadovaném směru. V případě křížení trati by MJ musela být vybavena dvěma kodéry. Proto se jako výhodnější jeví použití optického snímače.

3 Seznámení se ZigBee protokolem

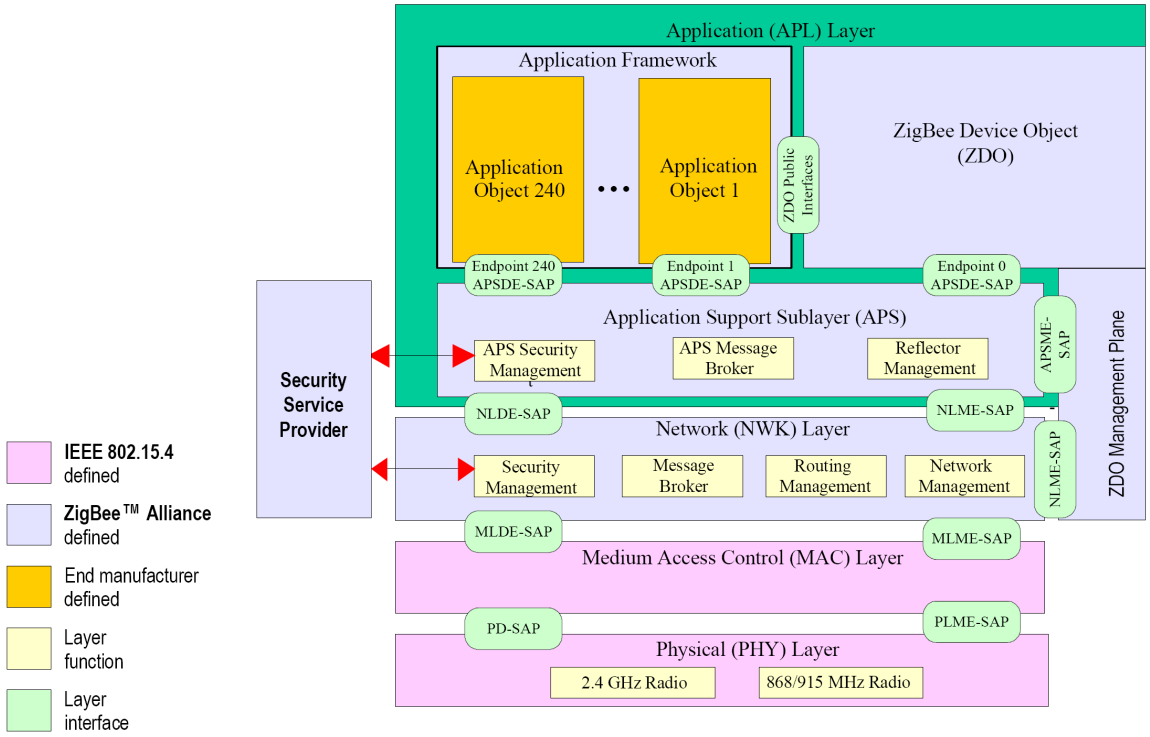
3.1 Zkratky používané v popisu protokolu

AIB	Application support layer information base
AF	Application framework
APDU	Application support sub-layer protocol data unit
APL	Application layer
APS	Application support sub-layer
APSDE	Application support sub-layer data entity
APSDE-SAP	Application support sub-layer data entity . service access point
APSME	Application support sub-layer management entity
APSME-SAP	Application support sub-layer management entity . service access point
ASDU	APS service data unit
BRT	Broadcast retry timer
BTR	Broadcast transaction record
BTT	Broadcast transaction table
CAP	Contention access period
CCA	Clear channel assessment
CFP	Contention free period
CCM*	Enhanced counter with CBC-MAC mode of operation
CSMA-CA	Carrier sense multiple access . collision avoidance
EPID	Extended PAN ID
FFD	Full function device
GTS	Guaranteed time slot
HDR	Header
IB	Information base
LQI	Link quality indicator
LR-WPAN	Low rate wireless personal area network
MAC	Medium access control
MCPS-SAP	Medium access control common part sub-layer . service access point
MIC	Message integrity code
MLME-SAP	Medium access control sub-layer management entity . service access point
MSC	Message sequence chart
MSDU	Medium access control sub-layer service data unit
MSG	Message service type
NBDT	Network broadcast delivery time

NHLE	Next higher layer entity
NIB	Network layer information base
NLDE	Network layer data entity
NLDE-SAP	Network layer data entity . service access point
NLME	Network layer management entity
NLME-SAP	Network layer management entity . service access point
NPDU	Network layer protocol data unit
NSDU	Network service data unit
NWK	Network
OSI	Open systems interconnection
PAN	Personal area network
PD-SAP	Physical layer data . service access point
PDU	Protocol data unit
PHY	Physical layer
PIB	Personal area network information base
PLME-SAP	Physical layer management entity . service access point
POS	Personal operating space
QOS	Quality of service
RFD	Reduced function device
RREP	Route reply
RREQ	Route request
RN	Routing node
SAP	Service access point
SKG	Secret key generation
SKKE	Symmetric-key key establishment
SSP	Security services provider
SSS	Security services specification
WPAN	Wireless personal area network
XML	Extensible markup language
ZB	ZigBee
ZDO	ZigBee device object

3.2 Představení

ZigBee je levný, energeticky nenáročný, obousměrný, bezdrátový komunikační standard vhodný zejména do distribuovaných měřících a automatizačních systémů nižšího výkonu. Jeho hlavní předností oproti obdobným standardům (zejména Bluetooth) je jednodušší implementace (zhruba 64 KB kódu oproti 720 KB u Bluetooth), velká dynamika sítě (až 65536 členů v jedné síti), dostatečný dosah (stovky metrů) a rychlé vytvoření sítě po restartu (30 ms). Základní nevýhodou je pak omezená datová propustnost (teoreticky 250 kb/s pro 2.4 GHz a DSSS modulaci, prakticky méně – záleží na poměru režijního a datového obsahu vysílaného rámce a počtu vysílacích pokusů před úspěšným doručením zprávy atd.).



Obr 2: Přehled vrstev ZigBee protokolu

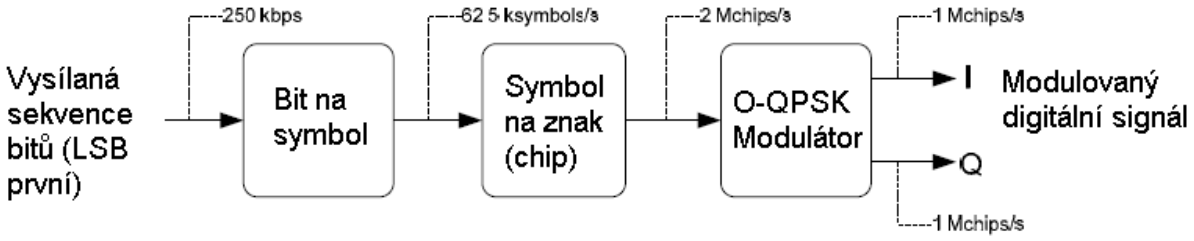
Obr 2 představuje uspořádání jednotlivých vrstev protokolu, tak jak jsou definovány v standardu [1]. Jako vlastní ZigBee by správně měla být označována část definovaná ZigBee Alliance, ale běžně se používá pro označení celého protokolu včetně IEEE 802.15.4-2006 [2], který obsahuje fyzickou a MAC vrstvu. Existuje i novější verze IEEE 802.15.4a-2007, která přidává další fyzické vrstvy, jmenovitě 2.4 GHz chirp spread spectrum a vyšší pásma (3244–4742 MHz a 5944–10 234 MHz). Potřebný hardware se však na trhu zatím nevyskytuje. Pro bližší seznámení uvádím dále přehled jednotlivých vrstev. Především čím se zabývají, co obsahují a jak vypadá datový rámec, případně řídicí rámec, který vrstva vytvoří a pošle vrstvě nižší. Vůbec se například nezabývám otázkou zabezpečení. Stručněji je vypracován i popis aplikační vrstvy, neboť nepředpokládám její hlubší využití pro spolupráci s komerčními zařízeními a její struktura je ze všech vrstev nejkomplexnější. Pro úplný přehled slouží oficiální specifikace [1][2], jednodušší výklad o struktuře ZigBee je v pracích [3] [4].

3.3 Fyzická vrstva

3.3.1 Vysílací část

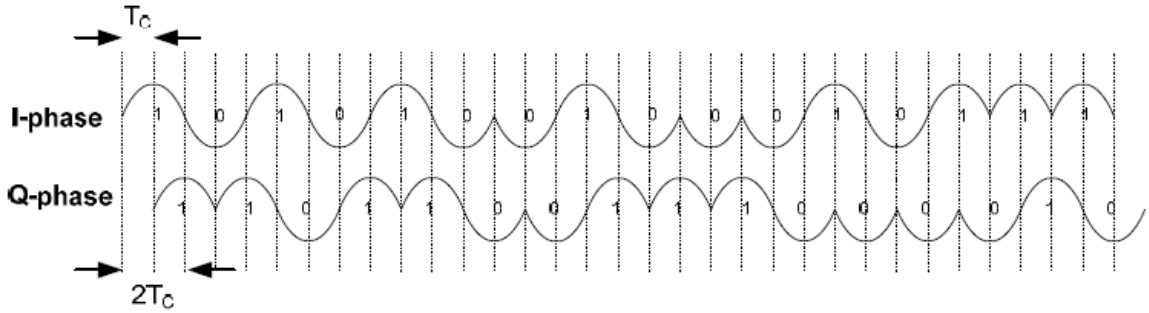
Fyzické vysílací médium je definováno v popisu IEEE 802.15.4 [2]. Existuje několik modulačních a frekvenčních variant komunikace. Základní rozdělení je na pásmo 868/915 (reálně 868-868.6 a 902-928 MHz) a pásmo 2450 (2400-2450 MHz), kterým jsou přidělena čísla kanálů 0, 1 až 10 a 11 až 26. Předchozí specifikace IEEE 802.15.4-2003 používala na místě pásma 868/915 modulování pomocí BPSK (binary phase shift keying) a verze 2006 zavedla možnost používat ASK (amplitude shift keying) a O-QPSK (offset-quadrature phase shift keying). K rozlišení těchto modulačních technik byl proto zaveden ještě identifikátor stránky kanálů. Pásmo 868/915 může přenášet data rychlostí od 20 kbit/s (868.3 MHz BPSK) do 250 kbit/s (906 MHz O-QPSK). Pásmo 2.4 GHz využívá výlučně modulaci O-QPSK pro dosažení přenosové rychlosti až 250 kbit/s. Jelikož je toto pásmo zatím nejperspektivnější a používá ho zvolené hardwarové řešení, zaslouží si podrobnější popis.

Rozdělení kanálů v pásmu 2.4 GHz je pro střední frekvence kanálů f_c následující $f_c = 2405 + 5(k-11),\ k = 11, 12, \ldots, 26$. Samotné kódování a modulace probíhá podle sekvence na Obr 3. Jedná se o DSSS (direct sequence spread spectrum). Nejprve jsou data převedena na 1 ze 16



Obr 3: Způsob kódování zprávy

čtyřbitových symbolů. Každý symbol je poté převeden na sekvenci 32 znaků (chips) definovaných ve standardu. Znaky jsou zvoleny záměrně tak, aby se jevily jako bílý šum – bez jakékoliv pravidelnosti. Tyto znaky jsou následně rozděleny na sudé a liché podle pořadí v sekvenci. Liché znaky jsou pak vysílány větví I a sudé větví Q (namodulovány na nosnou a vyvedeny na anténové výstupy), přičemž jsou vzájemně posunuty o polovinu délky znaku (offset). Praktická realizace 1 znaku je buď kladná sinusová půl vlna pro znak 1 nebo záporná půl vlna pro znak 0, jak je zřejmé z obrázku znázorňujícího přenos jednoho symbolu. Správně se tedy jedná o modulaci MSK (minimum shift keying), která

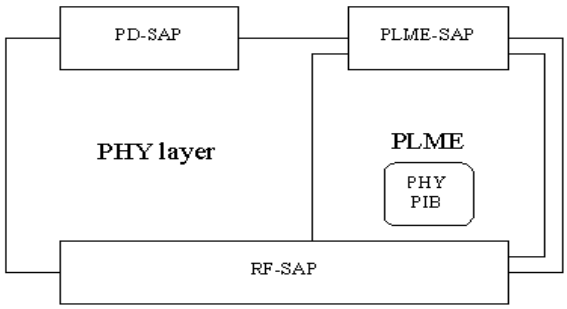


Obr 4: Fáze signálu před namodulováním na nosnou frekvenci

používá místo obdélníkového tvaru pulzu sinusovou půl vlnu. Perioda znaku je $2T_C = 1$ μs.

3.3.2 Interface

Fyzická vrstva implementuje rozhraní pro styk s vyšší MAC vrstvou skládající se ze dvou částí: vlastního datového přenosu PD-SAP a rozhraní pro přenos řídících příkazů PLME-SAP. Datová část rozhraní s MAC vrstvou zajišťuje celkem 3 služby: odeslání dat, potvrzení odeslání dat a indikaci příchozích dat. Řídící příkazy slouží k nastavení a čtení atributů PIB (PAN information base),



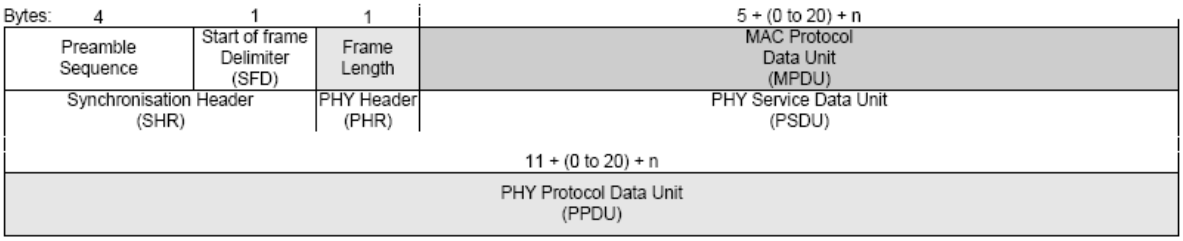
Obr 5: Schéma fyzické vrstvy

nastavení a zjištění výsledku CCA, provedení detekce úrovně signálu, zapnutí a vypnutí vysílání. Všechny tyto signály jsou opět generovány MAC vrstvou a po odeslání následuje jejich potvrzení s předáním výsledku. Na spodní straně má fyzická vrstva přímý přístup k vlastnímu vysílacímu a

přijímacímu hardwaru pomocí rozhraní RF-SAP.

3.3.3 Rámec fyzické vrstvy

Fyzická vrstva přidává k rámci předanému MAC vrstvou synchronizační hlavičku (SHR) a údaj o velikosti rámce MAC vrstvy PSDU. Synchronizační hlavička se skládá ze synchronizační sekvence a značky počátku rámce (A7h). Pro přenosové médium 2400–2483.5 MHz O-QPSK trvá synchronizační sekvence 128 µs. Údaj o velikosti datové části rámce PSDU, proto i jeho velikost je maximálně 127 B.



Obr 6: Zpracování rámce na úrovni fyzické vrstvy

3.4 MAC vrstva

3.4.1 Úvod

MAC vrstva (medium access control - „střední přístupová vrstva“) slouží jako rozhraní mezi fyzickou vrstvou a vrstvou síťovou (např. definovanou ZigBee Alliance). Jedná se o klíčovou vrstvu celého standardu IEEE 802.15.4, která se v jednodušších aplikacích používá jako konečná vrstva pro bezdrátovou komunikaci. Tato vrstva zabezpečuje fyzické spojení mezi komunikačními zařízeními v síti (dále uzly), neboť síťová vrstva jí předává adresu cílového uzlu. K vytvoření spojení MAC vrstva specifikuje role jednotlivých uzlů v síti na koordinátor a koncový (end device). Koordinátor je vlastním tvůrcem sítě, do které se připojují koncové uzly. Dále provádí fyzické řízení komunikace v síti buď jako náhodné (non-beacon) nebo pravidelné (beacon enabled). Nevytváří však vlastní strukturu sítě (nezná router), což obstarává vrstva síťová.

3.4.2 Interface

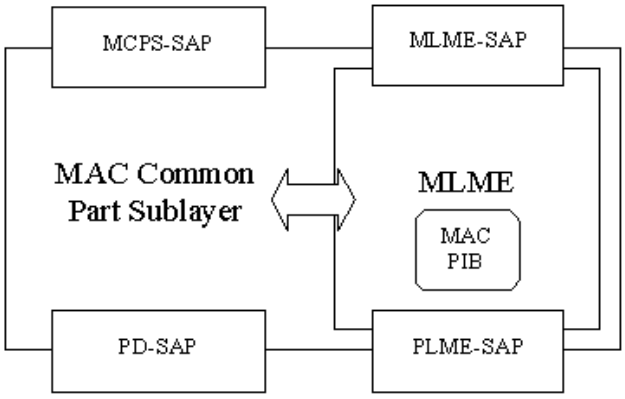
Rozhraní MAC vrstvy obsahuje, stejně jako rozhraní fyzické vrstvy, část datovou a část řídicí. Část rozhraní pro přenos dat ze síťové vrstvy je znázorněna jako MCPS-SAP. Datová část opět obsahuje struktury pro vyslání, potvrzení vyslání a indikaci příjmu zprávy. Požadavek na vyslání zprávy má formát:

MCPS-DATA.request (

SrcAddrMode,	- 16 b nebo 64 b adresa
DstAddrMode,	- 16 b nebo 64 b adresa
DstPANId,	- identifikátor cílové sítě 16 b
DstAddr,	- adresa cílového uzlu 16 b nebo 64 b
msduLength,	- délka zprávy 8 b
msdu,	- zpráva max. kolem 80 B
msduHandle,	- identifikátor zprávy (pro potvrzení)
TxOptions,	- 3 b , povolit potvrzování, GTS/CAP, přímý/nepřímý přenos
SecurityLevel,	- nastaveni zabezpečení
KeyIdMode,	- parametry zabezpečení,
KeySource,	- jsou ignorovány,
KeyIndex	- pokud je nastaveno zabezpečení úrovně 0

).

Pokud MAC vrstva požadavek přijme, vyšle síťové vrstvě potvrzení vysílání MCPS-DATA.confirm. Indikace dat MCPS-DATA.indication má formát velmi podobný MCPS-DATA.request, navíc obsahuje indikátor kvality spojení a časové razítko zprávy.



Obr 7: schéma MAC vrstvy IEEE 802.15.4

Řídicí část rozhraní MLME-SAP obsluhuje režijní komunikaci v síti. Jedná se zejména o vytvoření sítě, připojení, odpojení, ztrátu spojení a synchronizaci na koordinátor sítě. Samozřejmě zabezpečuje i přímé nastavení a čtení parametrů informační báze MAC PIB. Pro ilustraci rozhraní uvedu položku pro zařazení (asociaci) uzlu do sítě:

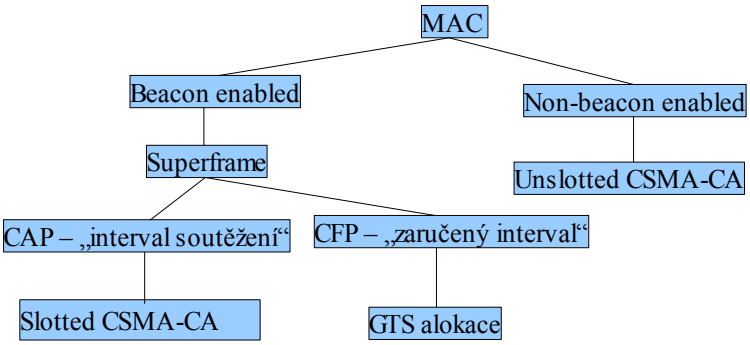
MLME-ASSOCIATE.request (

LogicalChannel,	- kanál, na kterém vysílá koordinátor sítě
ChannelPage,	- stránka kanálů
CoordAddrMode,	- 16 b nebo 64 b mód adresy koordinátoru
CoordPANId,	- identifikátor cílové sítě 16 b
CoordAddress,	- adresa cílového uzlu 16 b nebo 64 b
CapabilityInformation,	- dovednosti připojovaného uzlu (koncový,vypínající se, zabezpečený)
SecurityLevel,	- zabezpečení zprávy
KeyIdMode,	- a jeho parametry
KeySource,	
KeyIndex	

).

MAC vrstva požadavek odešle cílové MAC vrstvě běžící na koordinátoru sítě, která pošle zpět zprávu s odpovědí. Zdrojová MAC vrstva po příjmu této zprávy následně síťové vrstvě předá odpověď MLME-ASSOCIATE.response.

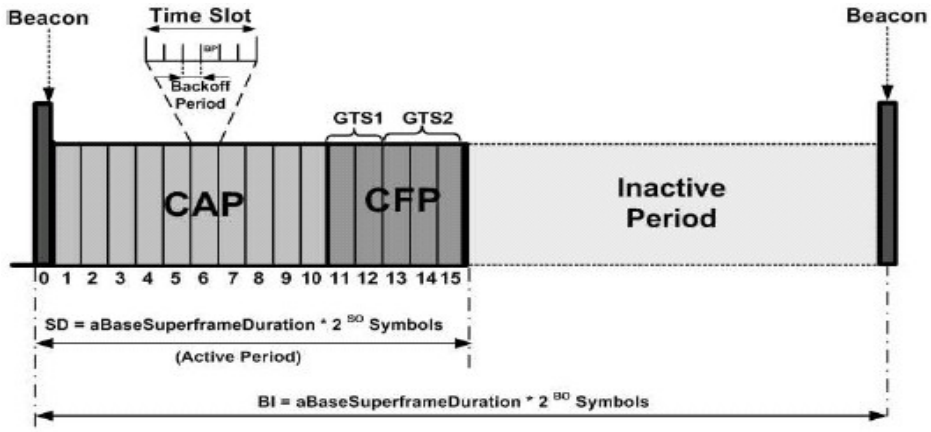
3.4.3 Beacon a non-beacon síť



Obr 8: Rozdíl mezi beacon a non-beacon komunikací MAC vrstvy

Rozhodování o způsobu spolupráce s fyzickou je určeno typem sítě, ve kterém má uzel pracovat.

MAC vrstva rozlišuje, zda pracuje v beacon enabled nebo non-beacon enabled síti. Beacon lze chápat jako startovací značku (synchronizaci) pro takzvaný superframe, ve kterém jsou vymezeny časové sloty pro vysílání a příjem jednotlivých uzlů. S beacon resp. non-beacon síti je svázán přístupový mechanismus CSMA-CA, který je slotted resp. unslotted. Beacon vysílá vždy koordinátor sítě a dává tak komunikaci v síti větší pravidelnost. Je to vhodné zejména při synchronním vzorkování z většího počtu bezdrátových senzorů. Spíše však platí, že při větším počtu uzlů ani non-beacon síť efektivně použít nejde, tedy pokud uzly nevysílají jeden datový rámec za hodinu.
U non-beacon sítě působí po celou dobu komunikace mechanismus CSMA-CA v unslotted podobě. Konflikty se řeší odmlčením uzlů, které zaznamenaly konflikt na náhodnou dobu v násobcích základního intervalu tzv. backoff (20 symbolů … 0, 32 ms). Jakmile uzel zjistí, že je kanál volný, začne vysílat, jinak se znovu odmlčí. Slotted CSMA-CA, používaná v beacon síti během CAP periody superframe, se liší pouze tím, že uzly se zde odmlčí všechny hned po přijmutí beacon rámce a po náhodném čekání se provede test obsazenosti kanálu CCA. Provádí se však vždy na počátku backoff

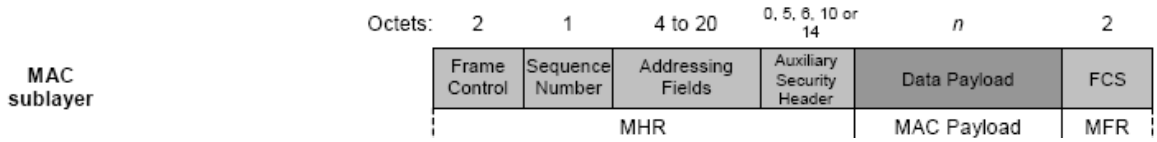


Obr 9: Příklad superframe

intervalu a typicky ve dvou po sobě následujících backoff intervalech. Pokud by odeslání zprávy připadlo mimo interval CAP, odloží se komunikace do dalšího superframe. Po úseku CAP může následovat úsek CFP obsahující několik GTS (garantovaných časových slotů). Časové sloty jsou úseky času, kdy je komunikace vyhrazena vždy jednomu uzlu a CSMA-CA se nepoužívá. Trvání GTS se měří v násobcích backoff periody. Časové sloty přiřazené konkrétnímu zařízení jsou specifikovány v beacon rámci, který může obsahovat rozdělení slotů až pro 7 uzlů. Platí, že vždy po beacon rámci následuje CAP minimální délky 440 symbolů (beacon je součástí CAP, navíc musí probíhat i jiná komunikace např. připojení nového uzlu do sítě). Nakonec může následovat interval spánku, kdy je celá síť, spadající pod řídicí uzel, nečinná. Každý superframe obsahuje 16 časových slotů. Délka jednoho slotu závisí na hodnotě SO – superframe order jako $3 \cdot 2^{SO}$ - vyjádřeno v násobcích backoff intervalu. Celý superframe tak může trvat od 48 do 786432 backoff intervalů, neboť SO je maximálně 14.

3.4.4 MAC datový rámec

Datový rámec realizuje vlastní přenos zpráv mezi uzly. Obsahuje 2 řídicí bajty, potvrzovací číslo (sequence number), adresu, případné šifrování, vlastní data a kontrolní součet (FCS). Řídicí bajty obsahují všechny typy rámců MAC vrstvy. Skládá se zejména z identifikátoru typu rámce a standardu, který splňuje (verze 2003 nebo novější), povolení šifrování a potvrzovacího rámce a druh adresování (16 nebo 64 b). Potvrzovací číslo je číslo, které odpovídající uzel odešle zpět v potvrzovacím rámci. Adresní část se skládá ze zdrojových a cílových adres sítě a uzlů. Kontrolní součet je standardního formátu 16-bit ITU-T CRC.



Obr 10: MAC rámec dat

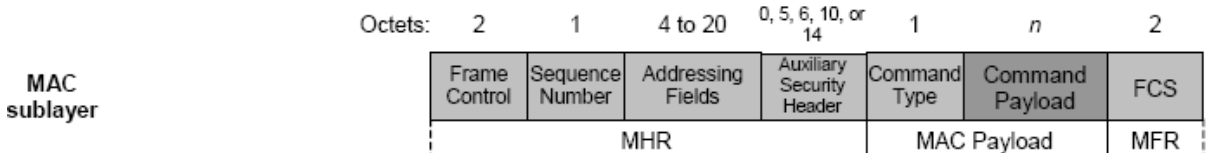
3.4.5 Potvrzovací rámec

Potvrzovací rámec je zjednodušenou verzí datového rámce. Obsahuje pouze řídicí bajty, číslo zprávy, kterou potvrzuje a kontrolní součet.

Obr 11: MAC rámec potvrzení příjmu datového rámce

3.4.6 Řídicí rámec(command frame)

Řídicí rámec se velmi podobá datovému rámci. Místo dat ale obsahuje identifikátor příkazu a k němu



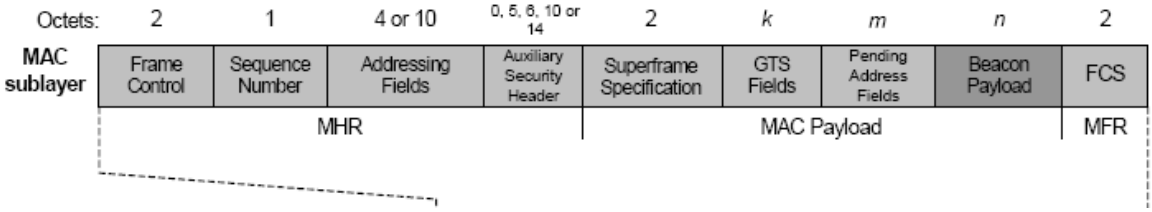
Obr 12: MAC řídící rámec

příslušné parametry. Existuje celkem 9 příkazů, jako příklad mohu jmenovat příkaz "Association request command", který je výsledkem interface příkazu MLME-ASSOCIATE.request. Tento příkaz má pouze 1 oktet parametrů. Parametry obsahují informace o schopnostech uzlu, který žádá zařazení do sítě, a požadavek na přidělení krátké adresy od koordinátoru. Jedná se o:

- schopnost stát se koordinátorem,
- plná nebo částečná funkčnost uzlu v síti (FFD nebo RFD),
- bateriové nebo síťové napájení,
- vypínání transceiveru během periody spánku,
- schopnost dekódovat šifrované rámce.

3.4.7 Beacon rámec (synchronizace posílání paketů)

Beacon rámec je opět podobný rámci datovému. Obsahuje řídicí pole, identifikační číslo, adresy, případné zabezpečení a vlastní datovou část. Datová část obsahuje specifikaci superframe a časových slotů, čekajících adres a případných dalších dat síťové vrstvy. Specifikace superframe obsahuje délku aktivní části a celého superframe, délku CAP a jestli je možné do sítě přiřadit nový uzel. Pole identifikující časové sloty obsahuje celkový počet použitých slotů, povolení alokace dalších slotů, směr vysílání při jejich použití a vlastní seznam slotů. Tento seznam se skládá z krátké adresy uzlu, kterému byl přidělen, pozice počátku slotu a doby jeho trvání. Čekající adresy je seznam adres uzlů, pro něž má koordinátor data.



Obr 13: MAC beacon rámec

3.5 Síťová vrstva

3.5.1 Úvod

Jak bylo zmíněno v případě MAC vrstvy, tak síťová vrstva slouží především ke správě struktury, neboli topologie sítě. Využívá k tomu pokročilejší koncový uzel – router. Rozlišují se tyto druhy sítí:

- hvězdicová (star) – jeden centrální uzel, přes který probíhá veškerá komunikace, nemá routery
- pavučina (mesh) – uzly s funkcí routeru jsou si rovnocenné, komunikace probíhá přes řadu routerů, zpravidla možnost několika cest mezi koncovými uzly
- strom (cluster tree) – několik hvězdicových sítí s routerem jako "koordinátorem" spojených dohromady ve vyšší hvězdicovou síť, která může být spojena s dalšími hvězdicovými sítěmi atd.

Další službou, kterou síťová vrstva v návaznosti na správu sítě provádí, je posílání zpráv s různým počtem příjemců:

konkrétní uzel – unicast

všechny uzly v sítí/podsíti – broadcast

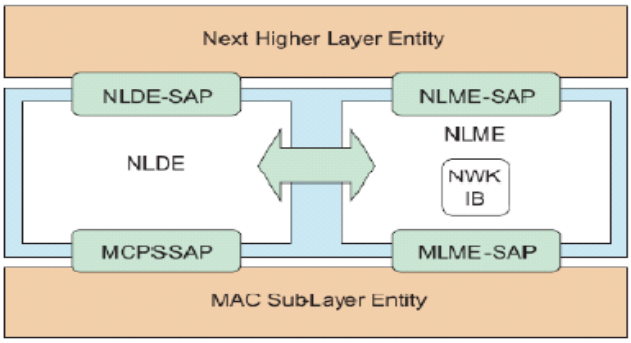
vybrané uzly – multicast

vybraná cesta – source routed

3.5.2 Interface

Obdobně jako vrstva MAC má i síťová vrstva část pro zpracování dat NLDE a část pro zpracování řídicích příkazů NLME. Aplikační vrstva používá obě části prostřednictvím interface NLDE-SAP a NLME-SAP. Část rozhraní pro přenos dat využívá strukturu

```
NLDE-DATA.request {
        DstAddrMode,            - typ adresování (16 nebo 64 b)
        DstAddr,                - cílová adresa
        NsduLength,             - délka zprávy
        Nsdu,                   - zpráva
        NsduHandle,             - identifikátor zprávy
        Radius,                 - maximální počet přeposlání
        NonmemberRadius,        - maximální počet přeposlání uzly, kterým není určena
                                  skupinová zpráva (multicast)
        DiscoverRoute,          - provést/neprovést prozkoumání struktury sítě
        SecurityEnable          - nastavení zabezpečení
}.
```

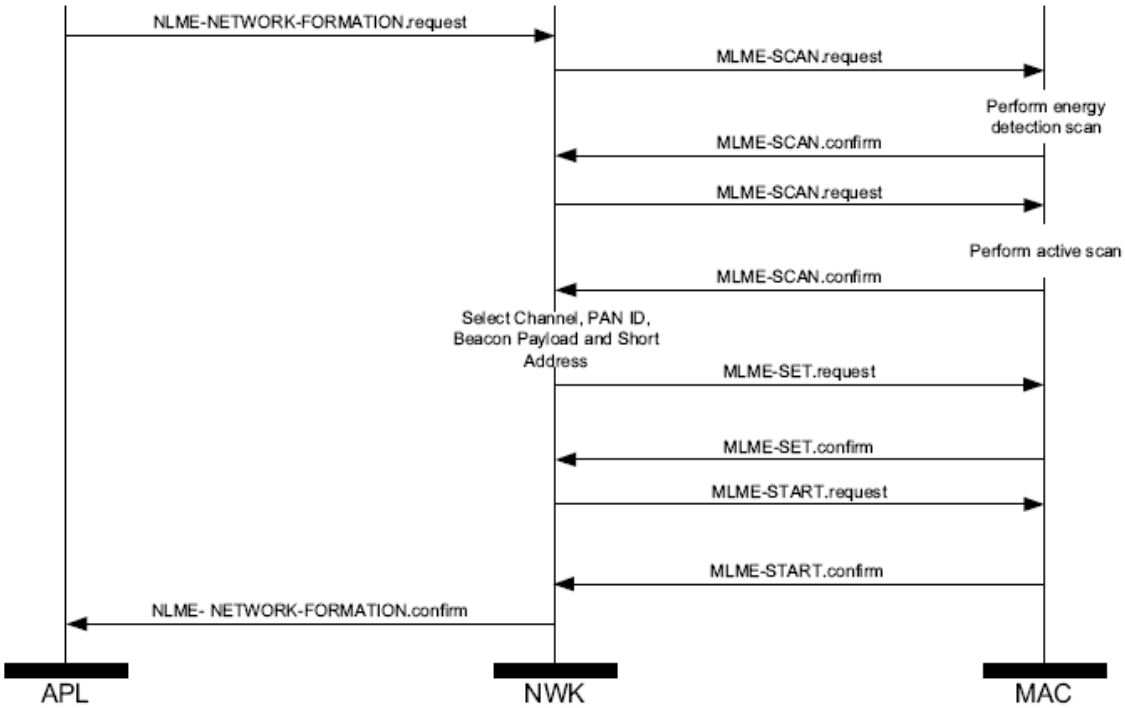


Obr 14: Schéma síťové vrstvy

Rozhraní síťové vrstvy tuto zprávu potvrzuje pomocí NLDE-DATA.confirm a případně indikuje příchozí zprávu NLDE-DATA.indication.
Těžištěm činnosti řídicí části rozhraní síťové vrstvy NLME-SAP je založení a udržování topologie sítě. Jako příklad ilustrující použití rozhraní síťové vrstvy mohu uvést použití příkazu na založení sítě:
NLME-NETWORK-FORMATION.request {

ScanChannels,	- bitová maska kanálu, které se mají prozkoumat
ScanDuration,	- doba strávená skenováním kanálu
BeaconOrder,	- délka superframe
SuperframeOrder,	- délka aktivní části superframe
BatteryLifeExtension	- šetřící mód

}.
Sekvenční diagram ilustrující použití tohoto příkazu je na Obr 15. MAC vrstva je nejprve instruována provést detekci výkonu všech kanálu a případné zjištění již fungujících sítí. Poté je provedeno vlastní založení sítě nastavením parametrů MAC vrstvy jako je zvolený kanál, identifikátor sítě, krátká adresa koordinátoru v síti a beacon/non-beacon enabled parametrů. Po založení sítě dostává aplikační vrstva potvrzení o úspěšném založení sítě.



Obr 15: Diagram funkce síťové vrstvy při založení sítě

3.5.3 Rámce síťové vrstvy

Rámec síťové vrstvy obsahuje podobné údaje jako rámec MAC vrstvy. Význam některých částí se ale liší. Pole Frame control obsahuje typ rámce (datový, řídicí), verzi síťového vrstvy, prozkoumání cesty

Octets: 2	2	2	1	1	0/8	0/8	0/1	Variable	Variable
Frame control	Destination address	Source address	Radius	Sequence number	Destination IEEE Address	Source IEEE Address	Multicast control	Source route subframe	Frame payload
NWK Header									Payload

Obr 16: Formát rámce síťové vrstvy

(discover route) a nastavení související s počtem příjemců. Radius je povolený počet přeposlání. Multicast control je pole obsahující řízení přeposílání skupinové zprávy mezi členy a nečleny cílové skupiny. Source route subframe je spojový seznam cesty od zdroje k cíli zprávy, je přítomen pouze při source routed zprávě. Tuto společnou část využívají jak datové, tak řídicí rámce síťové vrstvy. Je to obdobné jako v případě datových a řídicích rámců MAC vrstvy. Payload datového rámce jsou data vyšší vrstvy – aplikační. Payload řídicího rámce je identifikátor řídicího příkazu a jeho parametry tvořící datovou část řídicího rámce.

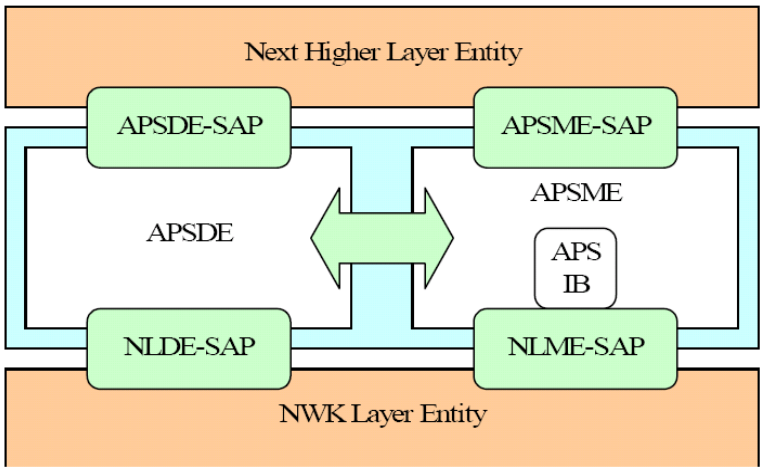
3.6 Aplikační vrstva

3.6.1 Úvod

Aplikační vrstva je nejvyšší vrstvou modelu ZigBee standardu. Skládá se z APS (Application support sublayer), která se specializuje na komunikaci se síťovou vrstvou, ZDO (ZigBee device objects) a AF (Application framework). Hlavním účelem celé aplikační vrstvy je umožnit dorozumění mezi uzly se stejnou funkcí (aplikací, kterou realizují – např. snímače mechanického namáhání stavební konstrukce). Jde tedy o spojení zařízení se specializovanou funkcí, která mají pracovat v jedné síti současně se zařízeními s jinou specializací.

3.6.2 APS (Application support sublayer)

Primárním účelem APS je styk se síťovou vrstvou a službami poskytovanými ZDO resp. AF. Model APS je v podstatě totožný s modelem síťové vrstvy. Opět používá rozhraní s datovou a řídicí částí. Jedinou změnou oproti síťové vrstvě je to, že přidává do hlavičky rámce údaje o profilu a clusteru, ke kterému je datová část určena.



Obr 17: Schéma APS

Mezi zodpovědnost této vrstvy patří i fragmentace dat poskytovaných AF (pouze unicast zprávy). Maximální délka rámce předaného fyzické vrstvě je totiž 127 B. MAC vrstva i síťová vrstva přidávají k datům své vlastní hlavičky (SHR,PHR,MHR,NHR, APS header), které velikost dat dále zmenšují. Přidávaná hlavička ale může mít různou délku. MAC vrstva může přidávat maximálně 25 B svých dat, takže délka MAC payload může být nejvýše 102 B. Síťová vrstva přidává minimálně 8 B své hlavičky, nicméně zjednodušuje MAC header neboť nepoužívá plné 64 bitové IEEE adresování, takže je možno počítat s MAC headerem maximální délky 11 B. Samotná APS používá 12 B header, což znamená, že fragment může mít délku 127 - 11 - 8 - 12 = 96 B.

3.6.3 Rámec APS

Přidání profilů a clusterů do specifikace rozhraní se pochopitelně projeví i na formátu rámce APS

Octets: 1	**0/1**	**0/2**	**0/2**	**0/2**	**0/1**	**1**	**0/ Variable**	**Variable**
Frame control	Destination endpoint	Group address	Cluster identifier	Profile identifier	Source endpoint	APS counter	Extended header	Frame payload
	Addressing fields							
APS header								APS payload

Obr 18: Formát rámce APS

vrstvy. Místo síťových adres používá popis cíle pomocí endpointů, které jsou objekty aplikace s přiřazenou síťovou adresou. Za zmínku dále stojí pole APS counter, které zabraňuje přijmutí stejné zprávy vícekrát.

3.6.4 AF (Application framework)

Application framework je vrcholem celé aplikační vrstvy. Rozděluje komunikaci s APS na samostatné funkční jednotky - endpointy, kterých může obsahovat až 240. Application framework tak definuje třídění zpráv podle parametrů identifikujících fyzickou funkci aplikace běžící jako objekt uvnitř něj. Jedná se zejména o ZigBee profily určující třídu aplikace např. domovní automatizace. Profily jsou buď veřejné nebo definované výrobcem. Jakmile je znám profil aplikace, provádí se další dělení podle identifikátoru zařízení (device descriptor) a podle služby poskytované tímto zařízením (cluster). Pokud má nové zařízení komunikovat s komerčním zařízením, musí použít stejný profil, identifikátor i cluster. Další součástí AF je device descriptor, definující dovednosti daného uzlu. Tyto dovednosti jsou k dispozici pro přístup ostatním uzlům v síti, které používají svoje ZDO k prohledání možností okolních uzlů v síti.

3.6.5 ZDO (ZigBee device objects)

V závislosti na typu uzlu (koordinátor, router, enddevice) existují v Application framework různé objekty zabezpečující třídy činností při zapojení uzlu v síti. Nemusí nutně přidávat nové parametry do sítě, pouze si přerozdělují jednotlivé úlohy při správě sítě. Řada parametru těchto správců sítě je nepovinná.

ZigBee device objects jsou tyto:

Device and service discovery	- obnovování sítě – správa adres a endpoint deskriptorů
Network manager	- správce síťové vrstvy
Binding manager	- vytvoření a skončení spojení („soketu“)
Security manager	- správce zabezpečení, práce s klíči, identifikace uzlů
Group manager	- utvoření a zrušení skupin endpoint objektů
Node manager	- správce popisu uzlu (vysílací výkon, citlivost)

4 Hardwarové řešení

4.1 Úvod

Hlavním problémem, který je nutno v souvislosti s hardwarem řešit je omezená velikost celého hardwarového řešení. Pro univerzálnost se hodí deska plošných spojů s malými rozměry i v rámci vlastní MJ. Je tedy nezbytné zvolit součástky s co možná nejmenším pouzdrem a vysokou mírou integrace, patrně v SMD provedení. Pro činnost v modelovém kolejišti bude třeba nasadit transceiver s integrovanou, čipovou nebo anténkou na desce plošných spojů. Případná ztráta výkonu/citlivosti na úkor ušetření místa nemusí být podstatná, neboť běžné kolejiště bývá v jedné místnosti. Naopak mikrokontrolér a samotný transceiver bude lépe nemít v jednom pouzdře kvůli případnému zachytávání komunikace při odlaďování programu. Spínání motorku MJ bude nejvýhodnější pomocí integrovaného MOSFET můstku, spínaný výkon je velmi malý, takže i výkon ke chlazení bude malý. K další úspoře místa určitě dojde při použití jednotného napájecího napětí v celé digitální části zapojení. Stabilizátor je však nutností, protože napájecí zdroj kolejiště bývá pouze transformátor s usměrňovačem. Na druhou stranu spotřeba zapojení nehraje žádnou roli. Samostatnou kapitolou je výběr vhodného optického snímače polohy. Na přesnost nejsou kladeny zvláštní nároky – spolehlivě stačí přesnost běžně používané optické myši. Problémem je spíše kompatibilita s ostatním hardwarem – běžné mikrokontroléry nemají podporu rozhraní myši (PS2).

4.2 Výběr obvodů Texas Instruments

Klíčové součástky splňující požadavky na výkon, velikost a kompatibilitu jsou obvody Texas Instruments. Jedná se o mikrokontrolér rodiny MSP430F2x1x, transceiver CC2520 a MOSSFET ovladač DC motorků DRV8800. Rozeberu jednotlivé body, které vedly ke zvolení hardwaru právě od Texas Instruments. Zmíním však i drobné nedostatky, které hodnocení obvodů snižují.

- dostupnost obvodů – TI poskytuje vzorky obvodů zdarma
- dostupnost vývojového kitu s podrobnou dokumentací – TI má v nabídce hotové komplexní řešení s kompletními periferiemi , navíc se ho podařilo zapůjčit (CC2520DK skládající se ze 3 modulů transceiveru CC2520EM, dvou modulů s mikrokontrolérem MSP430F2418 a 3 modulů periférií)
- dostupnost několika řešení vf části včetně antének na plošném spoji

- velká citlivost/vysílací výkon
- modulární koncepce – transceiver je připojen k mikrokontroléru přes SPI
- jednotné napájecí napětí – 3 V
- dostupnost softwaru – pro vývojové účely poskytuje TI svoji implementaci všech vrstev protokolu spolu s jednoduchou funkčností volně ke stažení pro nekomerční účely a použití na TI obvodech, obsahuje navíc i komunikační protokol s PC s výukovými aplikacemi. Vynikající je i přenositelnost kódu, neboť je podrobně popsán a pečlivě rozčleněn na ovladače funkcí mikrokontroléru.
- malá pouzdra – můstek 4x4 mm, transceiver 5x5 mm, mikrokontrolér 12x12

Nedostatky

- software a mikrokontrolér pouze pro komerční kompilátor, implementace části ZigBee kódu v přeložených knihovnách
- nutnost externích vf součástek – bez přesného dodržení referenční PCB klesá citlivost/vysílací výkon

4.3 Optický snímač polohy

Jako nejvhodnější snímač polohy vychází předpokládaná optická myš. Je snadno dostupná, levná a funguje prakticky na jakémkoliv rovném povrchu s malou odrazivostí. Nezanedbatelnou výhodou je také to, že tvoří samostatný funkční modul se všemi součástkami potřebnými ke své činnosti. Bohužel nevýhodou je, že stejný typ myši může obsahovat různé typy senzorů, o čemž jsem se sám přesvědčil. Jmenovitě typ NetScroll 100 firmy Genius. Obsahuje obvod PAN3101 [12]. Jednoduchou úpravou je možné jeho nasazení s obvody TI, neboť používá sériové rozhraní s obousměrným datovým vodičem a hodinami řízenými mikrokontrolérem. Bohužel obvod PAN3101 je ve starších (listopad 2006) kusech této myši. Stejná myš NetScroll 100 zakoupená v současné době (červen 2009) používá obvod PAN3401, který nabízí přímo podporu počítačového rozhraní PS2 [15]. Protokol PS2 je sice pro spolupráci s mikrokontrolérem méně vhodný (frekvence hodin pouze 10 – 16 kHz oproti až 10 MHz u PAN3101), je však univerzálně rozšířený. Po napsání ovladače lze tento ovladač využít s jakoukoliv PS2 myší. Další výhodou obvodu PAN3401 je menší počet externích součástek. Obsahuje přímý výkonový výstup pro osvětlovací LED diodu. Společnou nevýhodou obou obvodů zůstává nekompatibilita napájecího napětí s obvody od TI.

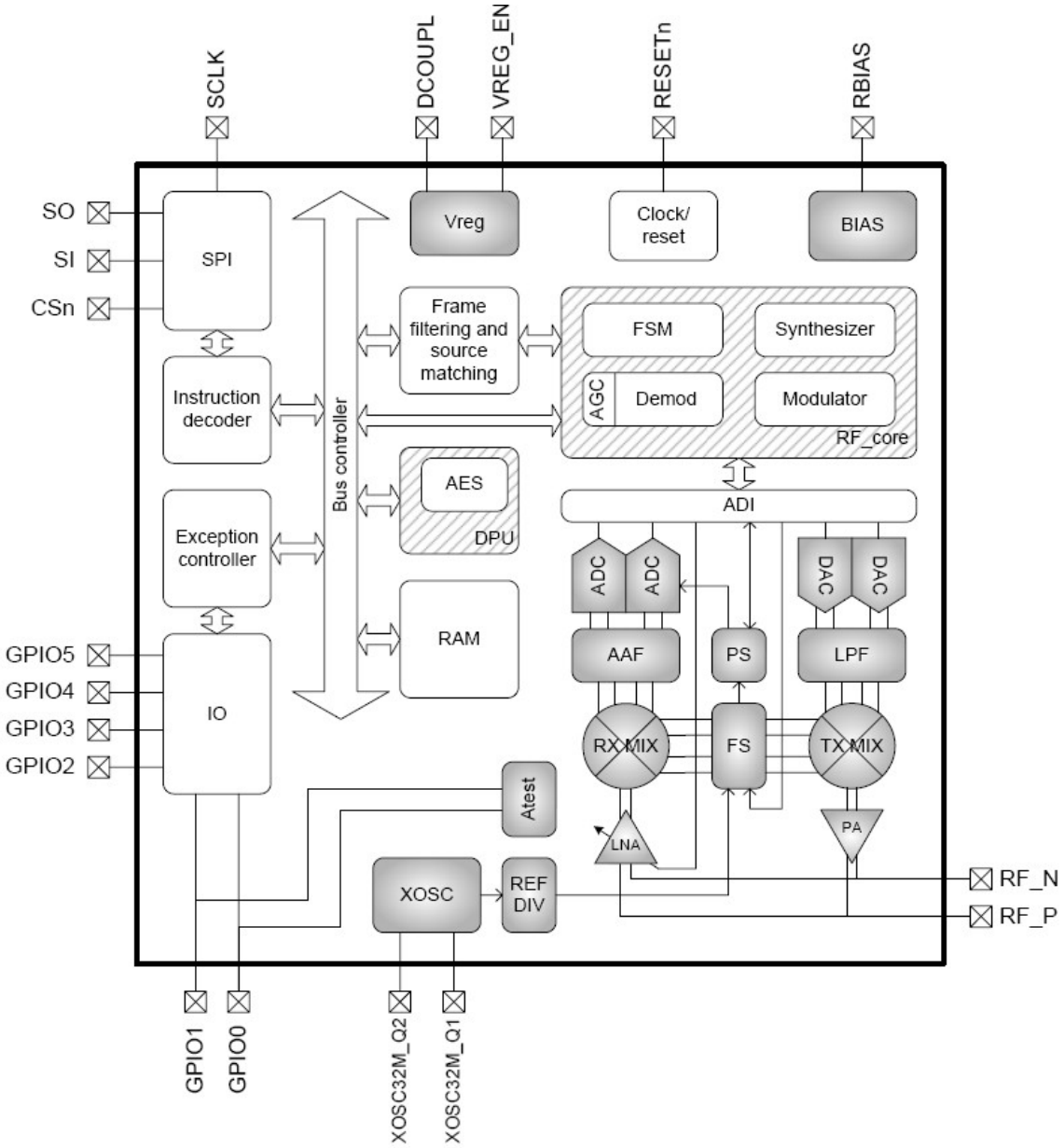
4.4 CC2520

4.4.1 Seznámení s obvodem

CC2520 je 2.4 GHz IEEE 802.15.4 kompatibilní transceiver od Texas Instruments 2. generace (dále v popisu zmiňován jako rádio nebo transceiver). Mezi jeho hlavní přednosti patří:

- rozsáhlá podpora IEEE 802.15.4 rámce:
  - a) úplná podpora fyzické vrstvy
  - b) podpora MAC vrstvy – detekce a filtrování podle FCF (frame control field) a adresy, automatická kontrola podle FCS (frame check sequence), automatické posílání potvrzovacích rámců, CSMA-CA, AES-128 kódování (bez dekódování) – vše volitelné
- regulovatelný vysílací výkon až + 5 dBm, citlivost až – 98 dBm
- základní komunikace s MCU pomocí 4 vodičového SPI + 6 dalších volitelných GPIO
- vyhrazená paměť pro vysílaný a přijímaný rámec spolu s maskou adres (celkem 768 B)
- rozsáhlý systém výjimek pro zachytávání významných událostí při činnosti obvodu

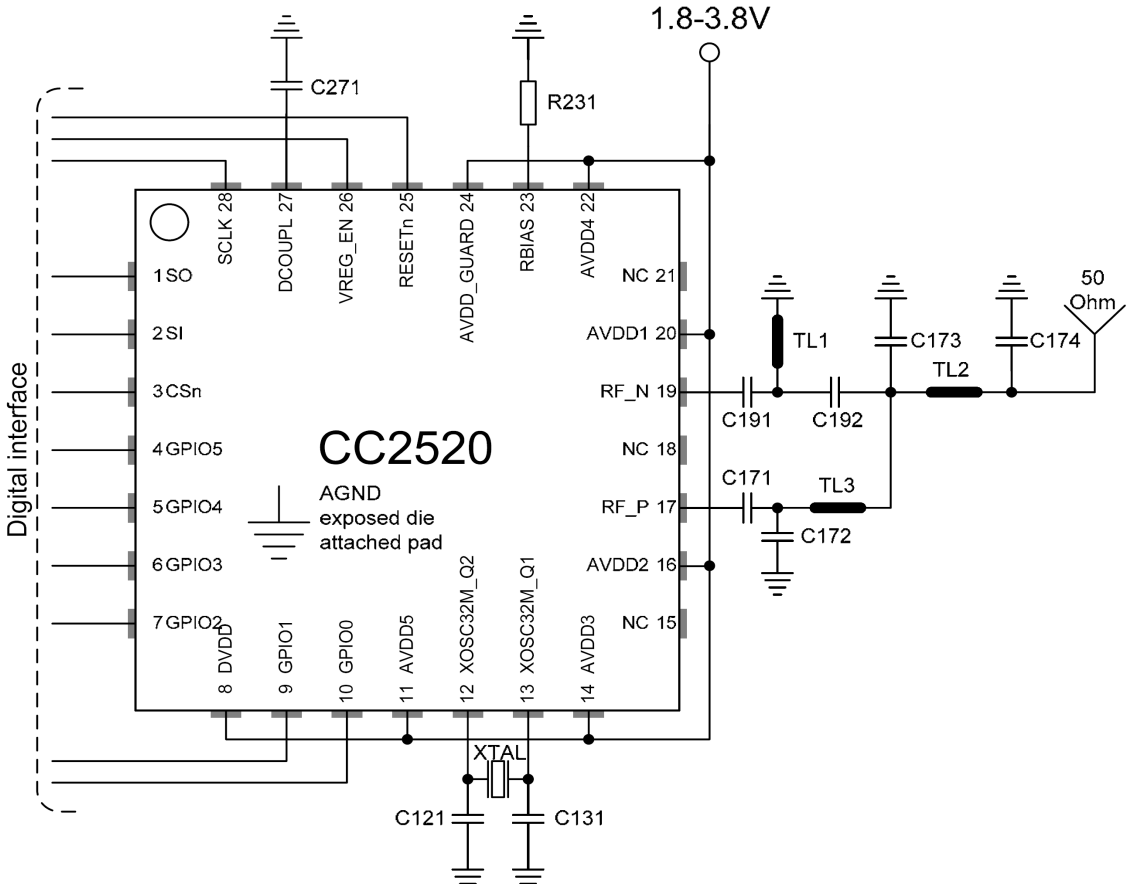
- maximální spotřeba 34 mA, pod 1 µA v úsporném režimu
- přídavné funkce jako externí nastavitelný hodinový výstup a teplotní senzor



Obr 19: Blokové schéma transceiveru CC2520

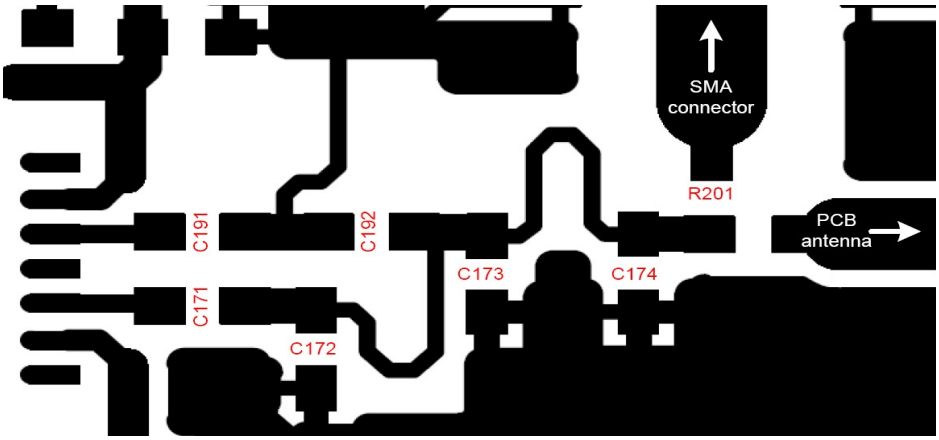
4.4.2 Externí součástky nutné pro činnost obvodu

Pro správnou činnost obvodu je v první řadě nutný krystal či oscilátor (připojen na XOSC32M_Q1) o frekvenci 32 MHz. Nejběžněji dostupný typ nevyhoví, neboť osciluje na 3. harmonické, proto je třeba zvolit typ pracující ve fundamentálním módu. Další kritickou součástkou je R231 na vývodu RBIAS, který musí být 56 kΩ s tolerancí 1 %, neboť slouží k regulaci proudu v rádiu. Na místě kondenzátoru C271 vyhoví běžný keramický 100 nF.



Obr 20: Referenční zapojení transceiveru CC2520

Pasivní prvky připojené na diferenciální výstupy vysílací části rádia (RF_N a RF_P) nejsou povinné, nicméně ve většině zapojení nutné. Charakteristická impedance vysílací části rádia je totiž $Z_0 = \infty$, ale impedance většiny antén pracujících v požadovaném pásmu (monopólů) je $Z_0 = 50\,\Omega$. Pasivní prvky tvoří přizpůsobovací člen, který zamezuje zbytečné ztrátě výkonu vysílacího řetězce. Je však možno použít přímo anténu s nekonečnou impedancí – skládaný dipól. V každém případě je nutno se přísně držet referenčních návrhů symetrizačního členu i samotné antény, jinak může být vysílací výkon a citlivost rádia zcela promrhána. Příklad provedení části desky plošných z modulu CC2520EMK, který je součástí vývojového kitu CC2520DK je na následujícím obrázku. Modul umožňuje použít buď externí nebo PCB anténku. V druhém případě je třeba přepájet rezistor R201. Rozteč vývodů rádia, které je na levé stráně, je 0.5 mm. Další informace o výběru anténky jsou v [7].



Obr 21: Referenční řešení PCB s rozprostřenými indukčnostmi modulu CC2520EM

Jako Digital interface je míněno propojení s mikrokontrolérem. Základ komunikace tvoří čtyřvodičové SPI. Rádio je při komunikaci vždy podřízená jednotka, na straně mikrokontroléru tvoří výstup MO (master output) a vstup MI (master input), u rádia pak vstup SI (slave input) a výstup SO (slave output). Mikrokontrolér vysílá SPI hodiny do vstupu (maximálně 8 MHz) SCLK a aktivuje komunikaci signálem do vstupu CSN. Propojení GPIO s mikrokontrolérem nejsou teoreticky nutná, avšak významně ulehčují jeho činnost, neboť se s jejich pomocí dají přímo vyvolat přerušení např. při přijmutí platného rámce do paměti rádia. Zvláště výhodné je zapojení GPIO0, jelikož po resetu rádia poskytuje hodinový signál 1 MHz, což ušetří krystal pro mikrokontrolér. Vstup VREG_EN slouží k přechodu do „ hlubokého spánku“ (LPM2 - odběr 30 nA, jedná se fakticky o vypnutí celého obvodu). Pokud je VREG_EN připojen na napájení, rádio zůstává stále aktivní a lze použít mód „lehkého spánku“ (LPM1 – odběr 0.2 mA). Vstup RESETn slouží k resetu rádia. Opět při trvalém napojení na napájení zůstává rádio stále v činnosti. I tak je však možno provést ekvivalentní reset pomocí dvou SPI příkazů (krystal se sice odpojí avšak SPI modul zůstává v činnosti).

4.4.3 Registry/paměť a instrukce pro práci s nimi

Struktura paměti

Obvod CC2520 obsahuje celkem 768 bajtů operační paměti a 128 B registrů nastavení obvodu. Pro detailní popis registrů slouží datasheet [3]. K bližšímu seznámení je vhodná i aplikace SmartRF Studio poskytující přehled o funkci jednotlivých registrů (kromě testovacích). Možnosti aplikace rozebírá samostatná sekce o doprovodném softwaru. Registry tvoří část paměti obvodu od adresy 0h po 7Fh, přičemž ne všechny adresy v tomto rozsahu obsahují registr. Část od adresy 100h po 3FFh pak tvoří 768 B RAM obvodu s úseky implicitně určenými pro určitou činnost např. přijímací buffer. Všechny instrukce jsou pochopitelně do rádia nahrávány přes SPI, takže je záhodno upřednostnit kratší instrukci před delší, pokud se taková možnost nabízí.

Přístup k paměti

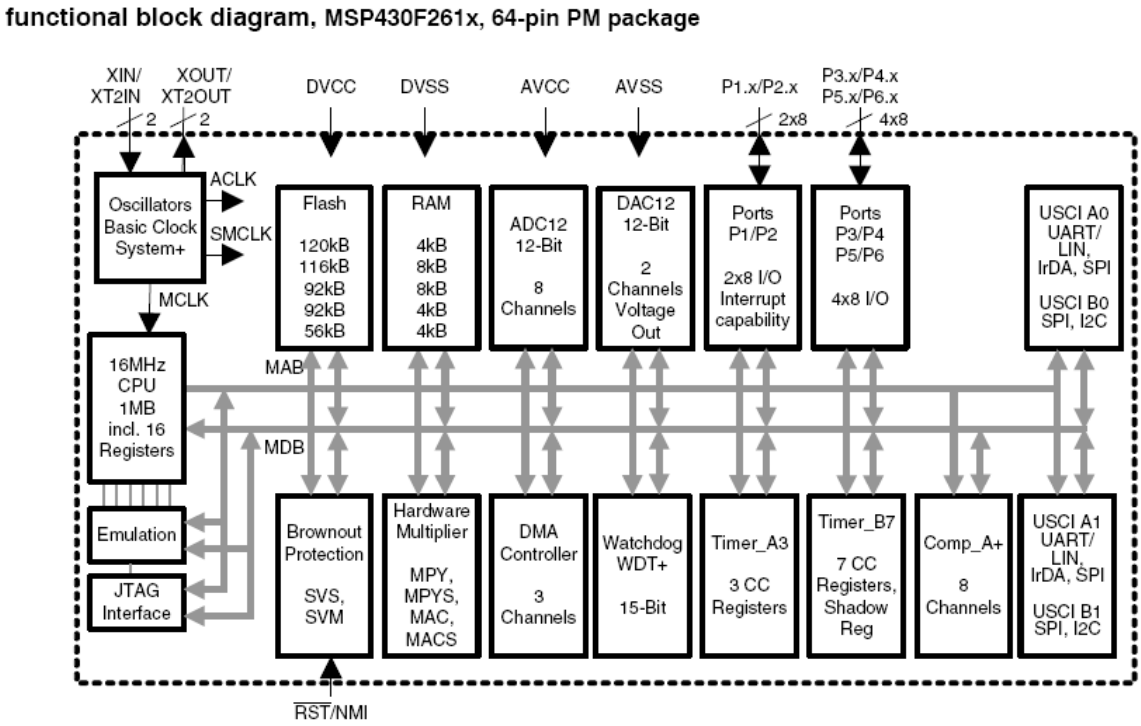
Základní instrukcí pro přístup k paměti je instrukce MEMRD resp. MEMWR. Obě pracují s 12bitovou adresou, která pokrývá veškerou paměť obvodu. Primárně se však používá pro přístup k registrům v rozsahu 40h až FFh. Pro adresy 0 až 3Fh totiž existuje instrukce REGRD resp. REGWR. Tyto instrukce jsou o 1 bajt kratší v porovnání s předchozími instrukcemi. Registry na adresách 0 až 1Fh lze pak také obsluhovat přímo instrukcí měnící 1 bit – BSET resp. BCLR.

4.5 MSP430F2x1x

Rodina mikrokontrolérů MSP430F2x1x je postavena na 16 bitovém RISC procesoru MSP430X s adresní sběrnicí rozšířenou na 20 bitů. V uvažovaném zapojení je použít mikrokontrolér MSP430F2618, který je použit i ve vývojovém kitu CC2520DK [6]. Je ovšem možné použít i obdobné typy MSP430F2617, MSP430F2418 a MSP430F2417. Typy se 2x17 se liší menší flash pamětí (92 kB

místo 116 kB). Typy 241x se liší absencí DMA a DAC převodníku. Společné je pro všechny 4 mikrokontroléry to, že mají ze všech MSP430, které TI vyrábí, největší paměť RAM (8 kB). Mikrokontroléry jsou vybaveny celou řadou modulů, pro činnost v uvažovaném zapojení lze vyzdvihnout např.

- 2 čítače/časovače s celkem 10 záchytnými/porovnávacími (capture/compare) registry (3+7)
- 2 moduly univerzální sériové komunikace (SPI, I2C, UART)
- 12 bitový A/D převodník s 8 vstupy, multiplexovaný
- celkem 16 vstupů s přerušovací schopností (po 1 vektoru přerušení od portu P1 a P2)
- celkem 3 možné zdroje hodinového signálu včetně interního oscilátoru

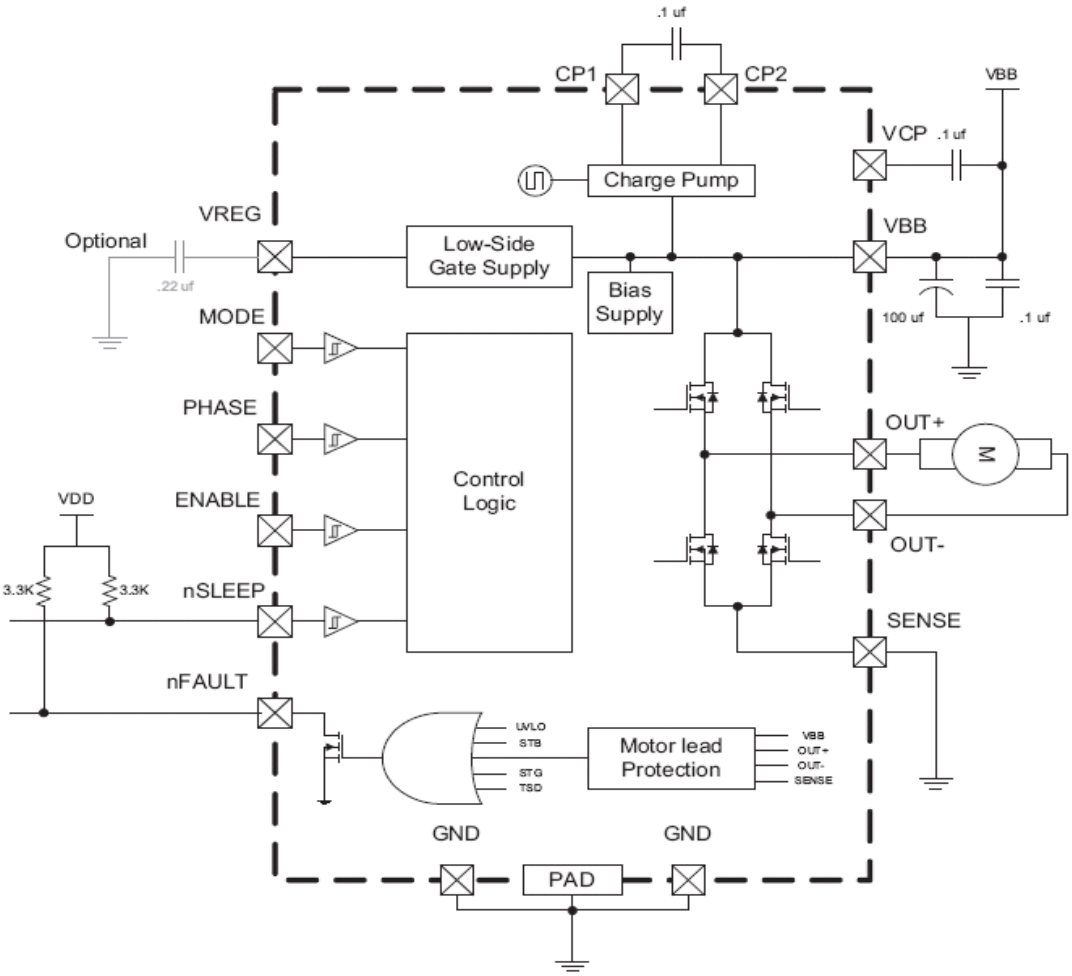


Obr 21: Blokové schéma MSP430F261x

K popisu mikrokontroléru z hlediska programátora je určen průvodce programátora[5]. Obsahuje podrobný popis práce se všemi moduly MSP430, které lze v mikrokontroléru této řady nalézt. Co se týče CPU, tak je popsána pod částí MSP430X (extended – 20 bitová adresní sběrnice). Mnohé popisy jsou však „device specific“, takže pro popis konkrétních přerušovacích vektorů a konfiguračních registrů periférií je nutné mít při ruce datasheet [4].

4.6 DRV 8800

DRV 8800 je DMOS integrovaný můstkový ovladač DC motorků. Hodí se pro spínání stejnosměrného napětí až 36 V při proudu ±2.8 A. Zahrnuje v sobě klasický tranzistorový můstek s měřením protékajícího proudu a spořící a ochranné funkce se signalizací. Vstupy ENABLE, PHASE, MODE a nSLEEP ovládají vlastní činnost můstku – ENABLE a PHASE rychlost a směr, MODE rychlost útlumu (pomalý a rychlý) a SLEEP odpojí výstupy můstku, čímž klesne jeho spotřeba pod 10 µA. Výstupem je naopak pin nFAULT signalizující poruchu činnosti. Jedná se o zkratování výstupu, přehřátí obvodu nebo pokles výkonového napětí pod 8 V. Dalším výstupem je SENSE. Pokud se na něj připojí malý odpor vůči zemi, je napětí na něm úměrné proudu protékajícímu zátěží můstku. Maximální hodnota tohoto napětí je 500 mV, při vyšší hodnotě se můstek vypíná.



Obr 22: Blokové schéma DRV8800

4.7 Použité zapojení

4.7.1 Propojení mikrokontroléru a transceiveru

Základní koncepce je shodná se zapojením modlu transceiveru CC2520EM s modulem mikrokontroléru EM430F2618 vývojového kitu CC2520DK. Rádio je propojeno s mikrokontrolérem pomocí čtyřvodičového SPI a všech GPIO. Propojení GPIO je nicméně trochu odlišné, uvádím zde popis propojení na mé desce spolu s přiřazenou funkcí na straně transceiveru:

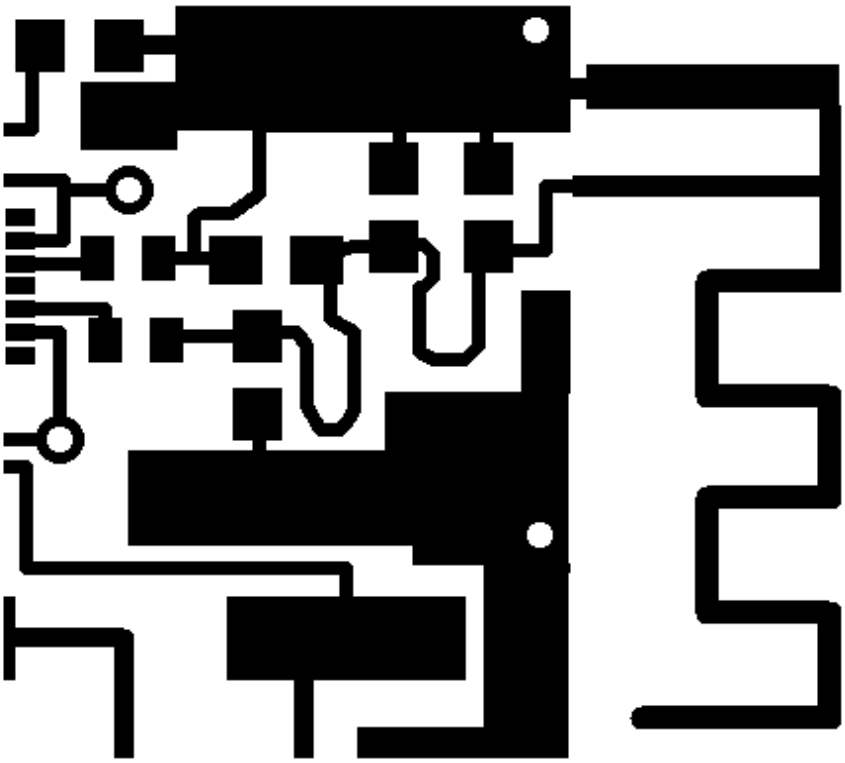
GPIO0	XT1IN	hodinový signál 8 MHz	(1 MHz po restartu obvodu)
GPIO1	P2.7	TX_FRM_DONE_BIT | TX_ACK_DONE_BIT (EXCEPTION_CHANNEL_A)	
GPIO2	P2.6	EXCEPTION_RFC_FIFOP	
GPIO3	P4.5	EXCEPTION_RFC_SNIFFER_CLK	(v původní implementaci GPIO5)
GPIO4	P4.6	EXCEPTION_RFC_SFD_SYNC	
GPIO5	P4.7	EXCEPTION_RFC_SNIFFER_DATA	(v původní implementaci GPIO3)

Původní funkce pinu GPIO1 – EXCEPTION_RFC_FIFO byla zrušena, aby uvolnila místo hodinovému signálu. Obecně se nedoporučuje provozovat mikrokontrolér na interní oscilátor, neboť má příliš nepřesnou a nestabilní frekvenci. Propojení VREG_EN a RESETn s mikrokontrolérem nejsou nutná, neboť neuvažuji režim šetření energií a resetovat lze pomocí SPI příkazu SRES [3].

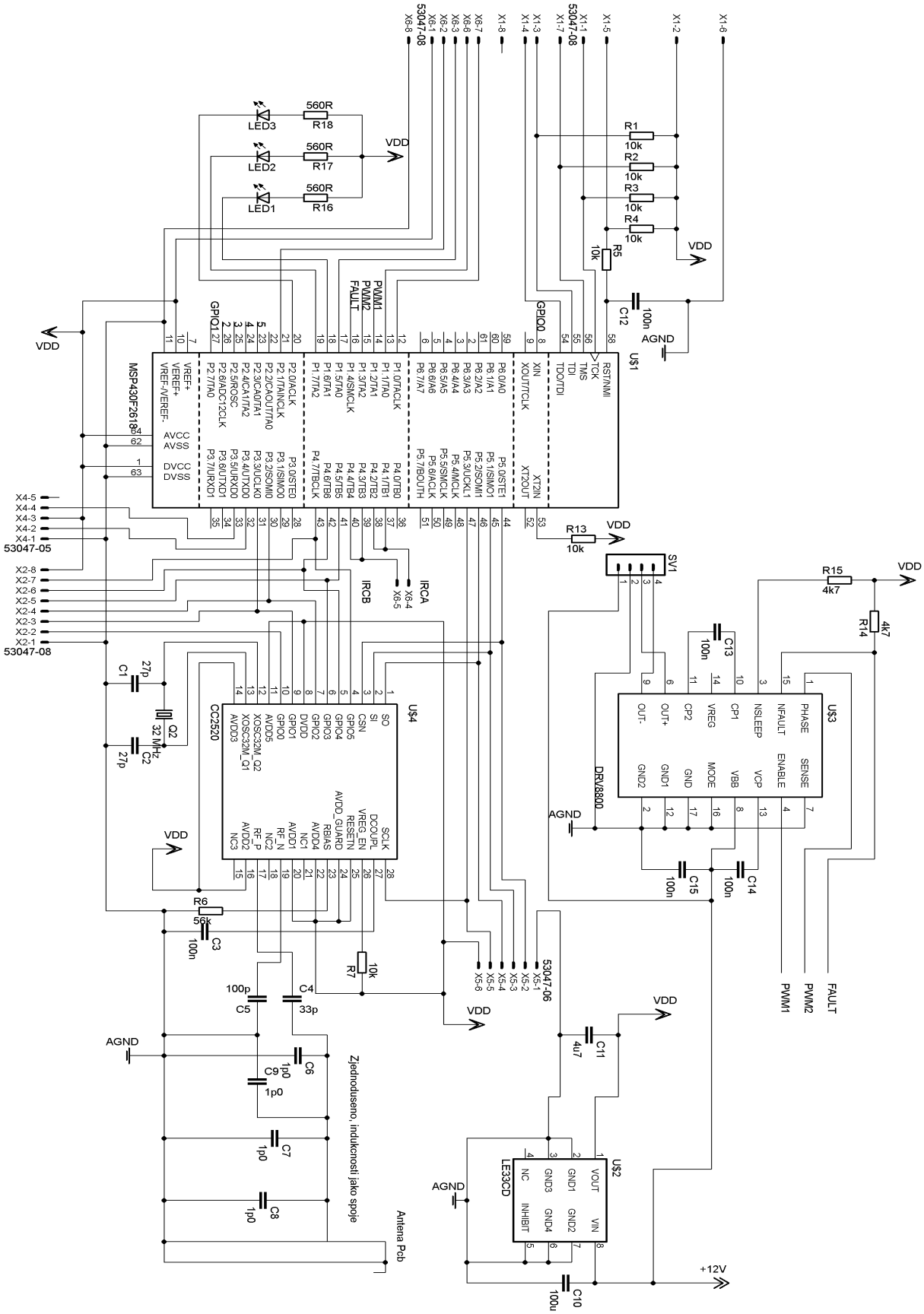
Řešení symetrizačního členu vf části se bohužel zcela neshoduje s modulem CC2520EM vlivem momentální nedostupnosti potřebných kondenzátorů. Rovněž plošné spoje představující indukčnost nejsou identické s referenčním návrhem. Existují rovněž malé diskrétní indukčnosti. S jejich dostupností je však situace ještě horší než v případě kondenzátorů. Komunikační dosah tak činí pouhých 5 metrů při plném vysílacím výkonu. Celkové schéma zapojení včetně vyvedení konektorů je na další stránce. Jedná se o konečnou verzi i s případnými opravami, proto jsou GPIO piny připojeny k více vývodů mikrokontroléru.

4.7.2 Volba antény

Texas Instruments nabízí celou řadu referenčních návrhů antén pro pásmo 2.4 GHz [7]. Protože je důležitým parametrem hardwaru především jeho velikost, zvolil jsem nejmenší typ antény – monopól s impedancí 50 ohmů. Lepším řešením, co se kvality vysílání/příjmu týče, by byl skládaný dipól, který nepotřebuje symetrizační člen. Zabral by však podstatnou část desky plošných spojů a omezil možnosti při umístění dalších součástek.



Obr 23: Detail antény a symetrizačního členu (velké pájecí body jsou pro kondenzátory 1p0)



Obr 24: Celkové schéma zapojení

4.7.3 Propojení ovladače motorku a optického senzoru s mikrokontrolérem

Pro generování PWM je vhodné využít časovače mikrokontroléru. Jelikož je potřeba jeden časovač pro čítání backoff period (320 µs), jsou obsazeny oba časovač obvodu. Perioda PWM 320 µs je totiž pro PWM dlouhá – jsou slyšitelné pískavé zvuky. Proto bylo zvoleno řešení, kdy je PWM generována časovačem A a backoff perioda časovačem B. Z principu je k činnosti můstku nutné propojit s mikrokontrolérem vstupy PHASE a ENABLE. V mém řešení je vstup ENABLE řízen právě časovačem a PHASE může být potom na libovolném pinu mikrokontroléru. Dále jsem provedl propojení s diagnostickým vývodem nFAULT můstku, pro případnou detekci zkratu výstupu neb nízkého napájecího napětí [14]. Měření proudu můstkem u takto malého motorku nepředpokládám.

Vstup od senzoru pohybu je možné přivést na vstupy časovače B (IRCA hodiny, IRCB data) kdyby bylo třeba dekódovat přímý signál od kvadraturního inkrementálního čidla. Jeden vstup může sloužit pro získání času náběžné a druhý spádové hrany jednoho IRC signálu. Nicméně zvolený senzor optické myši (PAN3101) takový výstup nemá, má pouze synchronní sériové komunikační rozhraní a to navíc s obousměrnou datovou linkou. Mikrokontrolér MSP tímto rozhraním nedisponuje (SPI má 2 jednosměrné datové linky), 2 neobsazené piny se tedy použijí jako běžné vstupy/výstupy a komunikace se implementuje softwarově. Pokud je třeba implementovat PS2 protokol (PAN3401), lze využít např. piny P1.0 a P1.1, které jsou vybaveny přerušovací schopností. Zahájení komunikace ze strany myši je indikováno stop bitem na datovém vodiči. Následných 9 datových bitů (8 hodnota + lichá parita) a stop bit je vzorkováno na spádové hraně hodinového signálu. Hlavním důvodem pro řešení s přerušením je malá rychlost přenosu (frekvence hodin 10 – 16 kHz), která by jinak plně zaměstnala procesor. Více o PS2 je například v [16].

4.7.4 Senzor optické myši

Samotný senzor optické myši je specializovaný obvod PAN3101 nebo PAN3401 pravidelně porovnávající obraz povrchu na kterém se pohybuje s obrazem předchozím. Pomocí pohybu obrazu tak určí rychlost a směr pohybu. K tomu ovládá osvětlovací diodu kvůli lepším světelným podmínkám snímače. Obvod PAN3101 je vybaven pouze sériovým komunikačním rozhraním napojeným na další specializovaný obvod. Tento obvod zaznamenává nejen pohyb myši, ale i stisknutí tlačítek. Realizuje také finální komunikaci pomocí protokolu PS2 s PC. Pokud se využívá původní desky myši, tak je nutno tento obvod odpájet. Další nepříjemností je napájení senzoru, který vyžaduje 5 V, zatímco obvody TI pracují při 3.3 V. Na desku myši je nutné přidat stabilizátor snižující napětí na 5 V. Z odlišného napájecího napětí vyplývá i nutnost upravit komunikační rozhraní. Na libovolný vývod mikrokontroléru nemůže být přivedeno napětí větší o 0.3 V než je napájecí napětí. Opačným směrem se problém nevyskytuje, jelikož obvod PAN3101 akceptuje jako log. 1 už napětí 2 V. Hodinový signál generuje vždy mikrokontrolér, je třeba ošetřit pouze propojení datového vodiče. Datový vodič musí být vybaven buď děličem nebo pouze pull down rezistorem s hodnotou kolem 3 kΩ. U PAN3101 se při logické jedničce udává možný odběr 2 mA při napětí 2.4 V, zatímco MSP430 vykazuje dostatečné napětí 2.7 V při odběru 6 mA. Typické schéma zapojení senzoru na desce myši je v datasheetu [12]. V případě použití obvodu PAN3401 s rozhraním PS2, jehož typické zapojení je uvedeno v datasheetu [15], je úprava ještě jednodušší. Stabilizátor zůstává a samotnou myš ani není potřeba rozebírat. Datový a hodinový vodič mají pouze pull down rezistor 6700 Ω nebo Zenerovu diodu 3.3 V s malou parazitní kapacitou. Sběrnice PS2 je totiž s otevřeným kolektorem s rezistorem 5 kΩ uvnitř senzoru myši. Klidové napětí na sběrnici je v případě rezistoru 2.9 V.

5 Softwarové řešení

5.1 Doprovodný software Z-Stack

5.1.1 Úvod

Texas Instruments poskytuje na svých stránkách [**] ke stažení celý softwarový balík Z-Stack určený ke spolupráci s vývojovým kitem CC2520DK a dalšími obvody jeho výroby. Tento balík se skládá z vlastní implementace ZigBee protokolu, ukázkových aplikací, podpůrného software pro práci s transceiverem (SmartRF Studio) a komunikačního software k zobrazení datových přenosů (Z-Tool).

5.1.2 Implementace ZigBee protokolu

Vlastní implementace ZigBee protokolu tvoří řada zdrojových souborů ovladačů spolu s knihovními funkcemi realizujícími vnitřní algoritmus vrstev protokolu. Obecně platí, že knihovní funkce obsluhují vnitřní schéma vrstvy – informační bázi a algoritmy funkcí vrstev. Zdrojové soubory obsahují hardwarově závislé funkce, které knihovny volají a hlavičkové soubory popisující struktury a proměnné pro pro práci s vrstvami protokolu a někdy i část kódu samotné vrstvy. Fyzická a MAC vrstva, kde transceiver přímo realizuje některé její funkce, je implementována převážně ve zdrojovém kódu. Naopak síťová vrstva je celá implementována v knihovně. Podrobněji je zdrojový kód Z-Stacku rozdělen na tyto části (shodné s adresářem Components na CD):

- hal – ovladače periférií mikrokontroléru optimalizované pro vývojový kit
- mac – ovladače transceiveru a styku s ním, systémové hodiny, rozhraní MAC vrstvy s vrstvou síťovou
- mt – monitor činnosti pro komunikaci s vývojovou deskou, ovladač komunikace se Z-Tool
- osal – operating system abstraction layer, rozvrhovač úloh, správa paměti, meziprocesní zprávy
- services – nevýznamné (porovnávání adres)
- stack – vlastní rozhraní ke knihovním funkcím realizujícím vrstvy protokolu (obsahuje i implementaci ZigBee cluster knihoven – profilů)
- zmac – interní úpravy v rozhraní mezi síťovou a MAC vrstvou

5.1.3 Referenční aplikace

Překladač IAR Embedded Workbench for MSP430

Texas Instruments preferuje při vývoji softwaru pro ZigBee protokol překladač jazyka C firmy IAR. V současné době (červen 2009) existuje jako další varianta Code Composer Essentials [18] s kompilátorem MSP430X přímo od Texas Instruments. Testovací 30denní verzi IAR EW430, která obsahuje téměř všechny prvky plné verze, lze, navzdory registraci, stáhnout ze stránek IAR vícekrát. Vývojové prostředí přichází již se zakomponovanými knihovními funkcemi, podporou debuggování, rozsáhlou dokumentaci a kompilačními profily. Zásadní nevýhodou je závislost na platformě Win XP, na kterou jsou určeny ovladače programátoru/debuggeru od TI a uzavřenost podpůrných knihoven pro procesor.

Vlastní aplikace

Texas Instruments poskytuje ke své implementaci ZigBee několik ukázkových projektů demonstrujících využití ZigBee. Jedná se o samostatné projekty v IAR Embedded Workbench for MSP430 verze 4.10A a vyšší, které mají nastavené všechny potřebné cesty ke zdrojovým souborům, knihovnám, konfiguračním souborům a nastavení kompilačního profilu. Kompilační profily jsou volbou typu uzlu (koordinátor, router, koncový uzel) při překladu zdrojového kódu, který je univerzální pro všechny typy uzlů.. Jako ukázkové úlohy jsou poskytovány tyto:

- GenericApp – posílá jednoduchou zprávu mezi kity každých 5 s
- SerialApp – bezdrátové prodloužení sériové linky RS232
- TransmitApp – testování přenosové rychlosti
- SampleLight / SampleSwitch – demonstrace použití cluster knihoven, použití komerčního profilu
- Home Automation – využití ZigBee cluster knihoven pro spolupráci s komerčními uzly

Součástí Z-Stacku je i rozsáhlá dokumentace použitých řešení a knihovních funkcí [**].

5.1.4 SmartRF Studio

SmartRF Studio je další částí balíku Z-Stack. Je to Win aplikace komunikující s kitem CC2520DK pomocí USB za předpokladu přímého propojení modulu s transceiverem s modulem periférií (bez použití modulu procesoru). Umožňuje bližší seznámení s funkcemi registrů radia a provádění některých příkazů. Dále je možné si zde vyzkoušet jednoduché posílání dat mezi dvěma jednotkami pouze na úrovni fyzické vrstvy. Podrobnější informace jsou v [6][8].

5.1.5 Z-Tool

Další Win aplikací v balíku Z-Stack je komunikační nástroj Z-Tool. Spolupracuje s modulem periférií, tentokrát v plném zapojení s modulem mikrokontroléru a transceiveru, pomocí sběrnice RS232. Jedná se o jednoduché ovládací rozhraní komunikace vývojového kitu. Zobrazuje příchozí zprávy a parametry nastavené v ZigBee aplikaci. Rovněž umožňuje změnit parametry ZigBee protokolu [9].

5.2 Vlastní program

5.2.1 Kompilační parametry

Překladač MSP430 od IAR umožňuje vytvořit vlastní kompilační profil pro implementaci koordinátoru, routeru a koncového zařízení. Součástí profilu je i nastavení případných hardwarových zvláštností, proto jsem vytvořil vlastní kompilační profily. Nicméně změny v ovladačích desky jsou natolik významné, že jsem nakonec vytvořil novou verzi Z-Stacku. Pro odlišení koncového uzlu na palubě MJ a koordinátoru jsem zavedl profily vencasBoard (deska MJ) a modelBoardCoord (deska z CC2520DK).

5.2.2 Úpravy v implementaci Z-Stack od TI

Jak již bylo řečeno dodaný software je určen exkluzivně ke spolupráci s kitem CC2520DK. Můj návrh zapojení ačkoliv z tohoto kitu vychází, liší se v prohození propojení mikrokontroléru s rádiem, jak bylo zmíněno v předchozí sekci. Uvedu zde pouze seznam změněných zdrojových souborů spolu s popisem. Kompletní původní (pro koordinátor) a upravená (pro MJ) implementace je na přiloženém CD.

hal_board_cfg.h – vynechána sekvence kalibrace hodin HAL_CLOCK_INIT(x), která kalibrovala interní oscilátor MSP430 na frekvenci externího krystalu. Změněno makro HAL_CPU_CLOCK_MHZ na 8. Nastavení definičních maker displeje, 2. časovače (ovládán až aplikací), klávesnice na zakázáno. Přemístění LED na P1.6, P1.7 a P2.0. Ostatní části, které se nepoužívají (tlačítka) byly ponechány. Přidán jednoduchý test blikání ke kontrole frekvence hodinového signálu – LED na P1.6 a připojení myši – LED na P1.7.

DCO_calibrate.h, DCO_calibrate.s43 – odstraněny, jednalo se o kalibrace interních hodin mikrokontroléru na násobek ACLK

hal_mac_config.h – hlavní úpravy spočívaly v přemístění GPIO pinů na nové pozice. Pin s funkcí HAL_MAC_FIFO_GPIO_BIT byl odstraněn, neboť není funkčně kritický a jeden GPIO pin je použit jako zdroj hodin, což v původní implementaci nebylo. Změněn přerušovací vektor z PORT1_VECTOR na PORT2_VECTOR. SPI piny zůstaly zachovány. Přidáno makro HAL_MAC_SPI_ACLK_CLOCK() nastavující zdroj hodin na ACLK, které jsou ze XIN. Dále jsou zde všechny činnosti časovače A převedeny na časovač B. Konkrétně TACCTL0 … TBCCTL0 ,TACCTL2 … TBCCTL5, TACCTL … TBCCTL6. Obdobná záměna je s korespondujícími registry pro uchování času capture/compare TACCRx.

hal_mac_cfg.c – odstraněn HAL_MAC_FIFO_GPIO_BIT

hal_timer.c – převedeno na časovač A, nicméně v hal_board_cfg.h je vypnutý, takže se tento kód nikdy nepoužívá

mac_dualchip.c – nastavení GPIO funkcí na straně rádia, zapnutí rádia v dualchipTurnOnRadioPowerVREG() s ekvivalentním resetem pomocí SPI. Dále je zde přidáno přepnutí MCLK (Master clock) mikrokontroléru na externí zdroj 8 MHz z transceiveru. V

macDualchipTurnOffRadioPower() se vypnutí (reset) neprovádí, neboť by pak bylo nutné znovu nastavovat zdroj u MCLK (MSP430 detekuje chybu oscilátoru a přepne se zpět na interní oscilátor).

mac_mcu_timer.h a mac_mcu_timer.c – zdroje přerušení časovače A upraveny na odpovídající zdroje přerušení časovače B. Úprava detekce zdroje přerušení – původní zdroje v registru zdroje přerušení TAIV měly kód 0x02 a 0x04. Nové zdroje v registru TBIV pak mají 0x0C a 0x0A. V nové implementaci je zjištění zdroje přerušení pomocí " if (TBIV & 0x0C) {}; if (TBIV & 0x0A) {}" nefunkční. Je nutno porovnávat pomocí " if (TBIV == 0x0C) {}; if (TBIV == 0x0A) {}".

mac_radio_defs.h – vynechání HAL_MAC_FIFO_GPIO_BIT z maker. Prohození funkcí GPIO3 a GPIO5 (záměna GPIOCTRL3 za GPIOCTRL5)

mac_radio.c – trvalé nastavení na maximální vysílací výkon
MAC_RADIO_SET_TX_POWER(0xFF);

5.2.3 Změny v aplikaci na straně koncového uzlu

Koncová aplikace je úpravou GenericApp.c z balíku aplikací dodávaných TI. Původní funkce GenericApp každých 5 sekund posílá zprávu instanci aplikace běžící na jiném uzlu v síti. Úprava spočívá v zavedení druhého typu zprávy v rámci aplikace – jiné číslo clusteru zprávy. Koncový uzel (na palubě MJ) ihned po zařazení do sítě – přijmutí zprávy ZDO_STATE_CHANGE požádá o spojení ZDP_MatchDescReq všechny jednotky v síti se stejným číslem profilu a číslem clusteru. Jedná se o automatické spojení. Existuje ještě spojení ruční ZDApp_SendEndDeviceBindReq(), které se ale musí spustit na obou spojovaných uzlech ve vymezené časovém intervalu. Po navázání spojení startuje softwarový časovač osal_start_timerEx, který aplikaci za určený čas pošle zprávu, kterou mu nastavila. Spustí se tak vlastní odeslání dat pomocí funkce AF_DataRequest.

Druhou základní součástí aplikace je OSAL_GenericApp.c (OSAL – operating system abstraction layer). Slouží k uchování a inicializaci všech úloh, které tvoří jednotlivé vrstvy ZigBee protokolu, a také k inicializaci vlastní aplikace. Rovněž si registruje ukazatele na obslužné rutiny přijímající z OSAL meziprocesní zprávy – předávání dat mezi vrstvami protokolu a koncovou aplikací. Podle identifikačního čísla úlohy pak hlavní smyčka celého programu implementovaná v OSAL.c rozvrhuje jednotlivé úlohy na procesor.

Třetí částí aplikace je má implementace PWM a práce s optickým senzorem myši MeasExtension.c. Generování PWM probíhá pomocí časovače A běžícího s periodou 40 KHz. Při frekvenci hodin 8MHz čítá časovač do hodnoty 200. Funkce setSpeed() pak nastaví rychlost otáčení motorku (hodnoty ±200 odpovídají plné rychlosti). Jako optický senzor je preferován obvod PAN3401 s PS2 komunikaci. Po úvodním resetu a nastavení rozlišení na 8 jednotek/mm, je povoleno přerušení spádovou hranou od hodinového signálu senzoru. Po resetu je totiž myš v tzv. stream módu, kdy při pohybu nebo stisknutí tlačítka vyšle standardní PS2 datový rámec. Obslužná rutina přerušení postupně přijímá celé 3 bajty dat ze senzoru myši a po 33. přerušení prohlásí data ze senzoru za platná a aktualizuje polohu jednotky. Pokud by se vyskytla komunikační chyba, obslužná rutina odmítne po nastavenou dobu přijímat data. Senzor optické myši totiž vysílá data s periodou 10 ms. Vlastní přenos dat trvá pžibližně 4 ms, neboť mezi bajty rámce bývá prodleva 300 µs. Jakmile dostane GenericApp zprávu o vypršení softwarového časovače, odešle již aktualizovanou polohu MJ.

5.2.4 Změny v aplikaci na straně koordinátoru

Na straně koordinátoru má aplikace stejnou formu jako na straně koncového uzlu. Jedinou změnou v části GenericApp.c je, že svázání s koncovou jednotkou probíhá po stisknutí tlačítka, nikoliv po připojení do sítě, v které je koordinátor od počátku její existence. Od tohoto okamžiku jsou koncovému uzlu zasílány zprávy o poloze potenciometru a případné další. Část MeasExtension.c obsahuje pouze čtení hodnoty napětí na potenciometru vývojové desky. K převodu je využit již implementovaný ovladač 12 bitového AD převodníku definovaný v hal_adc.c. Hodnota je pouze upravena tak, aby střední poloze potenciometru odpovídala nulová rychlost motorku.

6 Závěr

Díky správné volbě hardwaru a kvalitnímu poskytovanému softwaru se podařilo realizovat univerzální zapojení vhodné pro bezdrátové zpětnovazební řízení motorových jednotek i mnoha jiných zařízení. Konečné rozměry desky plošných spojů jsou 65x26 mm. Omezujícími faktory při nasazení mého zapojení budou dosah komunikace (5 m) a rychlost komunikace. Použitím optické myši jako senzoru pohybu se prokazatelně vzorkovací frekvence posunula pod hodnotu 100 Hz. Vyšší vzorkovací frekvence není potřeba. Implementace ZigBee Z-Stack v současné podobě rychlejší komunikaci ani neumožňuje, neboť nastávají konflikty mezi vysíláním koncového uzlu a koordinátoru a dochází tak ke zpoždění přenosu dat. Mám odzkoušeno, že spolehlivé dodržení pravidelnosti lze zaručit pro periodu zasílání zpráv 20 Hz v obou směrech. Pro důsledné řešení problému řízení více motorových jednotek by bylo třeba založit beacon-enabled síť. Rovněž by plynulosti přenosu prospělo zrušení přenosu MAC potvrzovacích rámců.

Současná implementace komunikace mezi motorovou jednotkou a jednotkou řídicí je nicméně funkční a po zastavění desky spolu s optickým senzorem do motorové jednotky může sloužit svému účelu. Po zprovoznění komunikačního rozhraní RS232 mezi koordinátorem a PC může být modelové kolejiště ovládáno přímo z osobního počítače. Samotný Z-Stack ovladač sériového portu již obsahuje (hal_uart.c).

Dalším námětem na zlepšení softwarového řešení řízení motorových jednotek je použití volně šiřitelné implementace ZigBee a volně šiřitelného kompilátoru C kódu. Texas Instruments totiž poskytuje svou implementaci pouze pro výzkumné účely a použití na svých obvodech. Patrně by se jednalo o řešení na základu TinyOS. Volně šiřitelný kompilátor pro rodinu MSP430X zatím neexistuje.

7 Zdroje dalších informací

[1] ZigBee Alliance (2008), „ZigBee Specification“ ZigBee Document 053474r17, January 17, 2008 11:09 am, http://www.zigbee.org/

[2] IEEE-TG15.4, „Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR- WPANs)" IEEE Std 802.15.4™-2006, IEEE standard for Information Technology, 2006.

[3] Texas Instruments, „CC2520 Datasheet”, http://focus.ti.com/lit/ds/symlink/cc2520.pdf

[4] Texas Instruments, „MSP430F2618 Datasheet”, http://focus.ti.com/lit/ds/symlink/msp430f2618.pdf

[5] Texas Instruments, „MSP430 User guide“, http://focus.ti.com/lit/ug/slau144e/slau144e.pdf

[6] Texas Instruments, „CC2520DK user guide“ , http://focus.ti.com/lit/ug/swru138/swru138.pdf

[7] Texas Instruments, „Antenna selection guide”, http://focus.ti.com/lit/an/swra161a/swra161a.pdf

[8] Texas Instruments, „SmartRF® Studio User Manual“, http://focus.ti.com/lit/ug/swru070h/swru070h.pdf

[9] Texas Instruments, „Z-Tool“, http://focus.ti.com/lit/an/swra119/swra119.pdf

[10] Ricardo Augusto Rodrigues da Silva Severino, „On the use of IEEE 802.15.4/ZigBee for Time-Sensitive Wireless Sensor Network Applications“, **P**olytechnic Institute of Porto School of Engineering, říjen 2008, http://www.cooperating-objects.eu/fileadmin/dissemination/2009-thesis-award/severino.pdf

[11] Mario Naval Escartín, „INTRA-VEHICULAR SENSOR ZIGBEE NETWORK“, ČVUT – FEL , prosinec 2008, http://www3.fs.cvut.cz/web/fileadmin/documents/12241-BOZEK/publikace/2008/2008_067_01.pdf

[12] PixArt Imaging Inc., „PAN3101 LOW COST CMOS OPTICAL MOUSE SENSOR”, http://www.pixart.com.tw/upload/PAN3101_V10_20051121170653.pdf

[13] Austriamicrosystems AG, „AS5311 High Resolution Magnetic Linear Encoder“, http://www.austriamicrosystems.com/eng/content/download/17954/325059/file/AS5311_Datasheet_v1_00.pdf

[14] Texas Instruments, „DRV8800, DRV8801 DMOS FULL-BRIDGE MOTOR DRIVERS“, http://focus.ti.com/lit/ds/symlink/drv8800.pdf

[15] PixArt Imaging Inc., „PAN3401 PS/2 OPTICAL MOUSE SINGLE CHIP”, http://www.pixart.com.tw/upload/PAN3401_SPEC_V13_20081009155108.pdf

[16] Adam Chapweske, „The PS/2 Mouse/Keyboard Protocol”, http://www.computer-engineering.org/ps2protocol/

[17] Hornby, „The Hornby Elite” DCC controller,

http://www.hornby.com/digital/the-hornby-select,53,HAR.html

[18] Texas Instruments, “Code Composer Essentials Professional v3.1 SR1 - Unrestricted IDE”, http://focus.ti.com/docs/toolsw/folders/print/msp-cce430pro.html

[**] Texas Instruments, „Z-Stack” – dokumentace ,

http://focus.ti.com/docs/toolsw/folders/print/z-tack.html, kopie je přiložena na disku