

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Diplomová práce

System pro měření a záznam fyzikálních parametrů prostředí

Plzeň, 2014

Milan Bárta

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 11. 5. 2014

Milan Bárta

Abstract

This paper discusses the system for measuring of physical parameters of the environment. The system is divided into three main parts – microcontroller, wireless module and sensors. Microcontroller is in charge of managing the entire system, collects and processes the sensor data and forwards the data to the wireless module that transmits them on the selected channel to the measuring station. Sensors measure the actual parameters of the environment. The sensor measured data are send to the microcontroller for further processing. The wireless module is used for wireless data transfer to the measuring station. The user has the option to configure the program as needed – what sensors will measure data, how big will be the delay between measurements or how precise will be the measuring from specified sensors. An important factor is the energy consumption since the system can be located for example somewhere in the countryside, where there is no access to electricity. All components can be switched to power saving mode when inactive.

Abstrakt

Tato práce pojednává o systému na měření fyzikálních parametrů prostředí. Systém se skládá ze 3 hlavních částí – mikrokontrolér, bezdrátový modul a senzory. Mikrokontrolér se stará o řízení celého systému, sbírá a zpracovává data ze senzorů a tyto data předává bezdrátovému modulu, který je vysílá na zvoleném kanálu do měřicí stanice. Senzory měří samotné parametry prostředí. Veškerá naměřená data předává ke zpracování mikrokontroléru. Bezdrátový modul slouží k bezdrátovému přenosu dat do měřicí stanice. Uživatel má možnost nakonfigurovat program dle potřeby – z jakých senzorů se budou měřit data, jaká bude prodleva mezi měřeními či jak přesně mají některé senzory měřit. Důležitým faktorem systému je spotřeba energie, jelikož systém může být umístěn např. někde v přírodě, kde není přístup k elektrické síti. Všechny komponenty se mohou při neaktivitě přepnout do úsporného režimu pro šetření energie.

Poděkování

Tímto bych chtěl poděkovat vedoucímu diplomové práce Ing. Jiřímu Novotnému za cenné rady, připomínky a vedení práce.

Obsah

1	Úvod.....	1
2	Výběr komponent	2
2.1	Srovnání mikrokontrolérů	2
2.1.1	EFM 32 Tiny Gecko	2
2.1.2	Kinetis KL2.....	3
2.1.3	NXP LPC110x	4
2.2	Výběr mikrokontroléru.....	5
2.3	Srovnání bezdrátových modulů.....	5
2.3.1	Analog Devices ADF7242.....	5
2.3.2	Microchip MRF24J40.....	6
2.3.3	Texas Instruments CC2520.....	7
2.4	Výběr bezdrátového modulu	8
3	Popis mikrokontroléru	9
3.1	Co je to mikrokontrolér?	9
3.2	Popis periferií	9
3.2.1	Jádro a paměť	9
3.2.2	Řízení periferií	13
3.2.3	Správa hodin (CMU)	13
3.2.4	Správa napájení (EMU)	16
3.2.5	I/O Porty	18
3.2.6	Časovače a triggery.....	19
3.2.7	Analogová rozhraní.....	24
3.2.8	Sériová rozhraní	26
3.2.9	Bezpečnost.....	33
4	Popis bezdrátového modulu.....	35
4.1	Základní popis.....	35
4.2	Paměť zařízení.....	35
4.3	Funkční popis	36
4.3.1	Reset.....	36
4.3.2	Inicializace	36
4.3.3	Přerušení	36
4.3.4	Výběr kanálu.....	37

4.3.5	Vysílání dat	37
4.3.6	Příjem dat	38
5	Popis seznorů	39
5.1	Akcelerometr Freescale MMA7660FC	39
5.2	Magnetometr Freescale MAG3110	40
5.3	Teploměr a vlhkoměr Sensirion SHT21	41
5.4	Tlakoměr BOSCH BMP180	42
6	Spotřeba	44
6.1	Spotřeba MCU	44
6.2	Spotřeba bezdrátového modulu	44
6.3	Spotřeba senzorů	44
6.3.1	Akcelerometr	44
6.3.2	Magnetometr	45
6.3.3	Teploměr a vlhkoměr	45
6.3.4	Tlakoměr	45
6.4	Teoretický odhad spotřeby na základě navrženého programového vybavení	46
7	Programová realizace	52
7.1	Účel programu	52
7.2	Popis programu	52
7.2.1	i2c.c	52
7.2.2	i2c.h	52
7.2.3	i2c_temp_sensor.c	52
7.2.4	i2c_temp_sensor.h	54
7.2.5	i2c_pres_sensor.c	54
7.2.6	i2c_pres_sensor.h	55
7.2.7	i2c_accel_sensor.c	55
7.2.8	i2c_accel_sensor.h	56
7.2.9	i2c_mag_sensor.c	57
7.2.10	i2c_accel_sensor.h	58
7.2.11	config.h	58
7.2.12	spi.c	58
7.2.13	spi.h	59
7.2.14	usart.c	59

7.2.15	usart.h.....	59
7.2.16	i2c_sensors_main.c	59
7.3	Konfigurace.....	60
8	Závěr	61
	Použité zkratky a termíny	62
	Literatura.....	65
	Příloha A – Schéma zapojení systému.....	66
	Příloha B – Uživatelská příručka	70

1 Úvod

Tato diplomová práce pojednává o systému na měření fyzikálních parametrů prostředí. Systém je rozdělen na 3 hlavní části.

Mikrokontrolér celý systém řídí, dává podněty sensorům, sbírá a zpracovává data ze sensorů a tyto data předává bezdrátovému modulu, který je vysílá na zvoleném kanálu do měřicí stanice. Jestliže mikrokontrolér neprovádí žádnou činnost, přepne se do úsporného režimu, aby šetřil energií.

Senzory měří samotné parametry prostředí. Systém obsahuje senzory na měření teploty, vlhkosti, atmosférického tlaku, změnu polohy a magnetického pole. Senzory jsou s mikrokontrolérem připojeny přes sériovou sběrnici I²C. Senzory jsou v neaktivním režimu do té doby, než dostanou podnět od mikrokontroléru na zahájení měření. Po vyžádání jsou naměřená data poslána mikrokontroléru a senzor se přepne opět do neaktivního režimu.

Bezdrátový modul slouží k bezdrátovému přenosu dat do měřicí stanice. Mikrokontrolér po sesbírání všech dat od sensorů vytvoří rámec, který pošle modulu a dá mu příkaz na odvysílání. Ten si tyto data vezme a odvysílá je na nastaveném kanálu. Měřicí stanice musí být na stejném kanálu jako vysílaná data, aby data mohla přijímat.

Vzhledem k tomu, že systém slouží pro měření fyzikálních parametrů prostředí, může být umístěn např. někde v přírodě. Z tohoto důvodu je nutné, aby měl systém co nejnižší spotřebu elektrické energie. Mikrokontrolér obsahuje několik režimů pro úsporu energie. Jestliže není používán, je přepnut do nízkopříkonového režimu, ve kterém jsou odpojené určité periferie či hodiny, které v tu dobu ale stejně nejsou potřeba. Ve zjednodušené formě je to i v případě sensorů, které jsou buď v aktivním režimu, kdy měří danou veličinu nebo v režimu spánku či neaktivním režimu, kdy mají minimální spotřebu. Bezdrátový modul má nejvyšší spotřebu v režimu vysílání nebo přijímání. V ostatních případech je automaticky v neaktivním režimu, ve kterém má minimální spotřebu.

2 Výběr komponent

2.1 Srovnání mikrokontrolérů

Základním prvkem modulu je mikrokontrolér. Mikrokontrolér řídí všechna připojená zařízení. Následuje srovnání a výběr mikrokontroléru pro potřeby systému.

2.1.1 EFM 32 Tiny Gecko

Mikrokontroléry od firmy Silicon Labs řady EFM32™ Tiny Gecko se vyznačují autonomními a energeticky nenáročnými periferiemi a výkonem díky použití procesoru postaveném na architektuře ARM® Cortex®-M3. Velikost mikrokontroléru se pohybuje již od 4 x 4 mm a spotřeba v aktivním režimu se pohybuje okolo 150 μ A/MHz a 900nA v úsporném režimu se zapnutým RTC a uchováním dat v registru a RAM. Toto je vhodné ke snížení odběru elektrického proudu v případech, kdy mikrokontrolér nemusí provádět žádnou činnost [3].

V tabulce 1 je vidět srovnání 3 mikrokontrolérů řady Tiny Gecko.

Tabulka 1 Srovnání mikrokontrolérů řady Tiny Gecko

	RFM32 Tiny Gecko		
	EFM32TG110	EFM32TG210	EFM32TG822
Procesor	32-bit do 32MHz		
Správa napětí	20 nA Shutoff Mode 0,6 uA Stop Mode 1,0 uA Deep Sleep Mode 51 uA/MHz Sleep Mode, přibl. 1,6 mA 150 uA/MHz Run Mode, přibl. 5 mA		
Hodiny	32 kHz nízkofrekvenční RC a krystalový oscilátor 32 MHz vysokofrekvenční RC a krystalový oscilátor 1 kHz Extrémně nízkofrekvenční RC oscilátor		
FLASH	4/8/16/32 KB	8/16/32 KB	
RAM	2/2/4/4 KB	2/4/4 KB	
I/O	17 pinů Nastavení umístění IO pinů periferií 11 asynchronních externích vstupů	24 pinů Nastavení umístění IO pinů periferií 14 asynchronních externích vstupů	37 pinů Nastavení umístění IO pinů periferií 16 asynchronních externích vstupů
DMA	7 kanálů	8 kanálů	9 kanálů
PRS	7 kanálů pro autonomní posílání signálů mezi periferiemi	8 kanálů pro autonomní posílání signálů mezi periferiemi	9 kanálů pro autonomní posílání signálů mezi periferiemi
AES (Advanced Encryption Standard)	128/256-bit klíče v 54/75 cyklech		

Čítače/Časovače	2x 16-bit čítač/časovač Funkce Compare/Capture/PWM 16-bit časovač s nízkým odběrem proudu 24-bitový RTC WatchDog
Komunikační rozhraní	2x Universal Synchronous Asynchronous Receiver/Transmitter (USART) Low Energy UART - autonomní operace s DMA řadičem v Deep Sleep módu Rozhraní I ² C
Analogová rozhraní	12-bit ADC převodník 12-bit DAC převodník 2x Analogový komparátor s 12 vstupy Porovnávač napájecího napětí
Teploty	-40 – 84 °C
Napájení	1,85 – 3,8V
Ostatní	Nízkonapěťové autonomní rozhraní pro monitorování senzorů v Deep Sleep módu

2.1.2 Kinetis KL2

Mikrokontroléry od firmy Kinetis řady KL2x mají již v základu integrované rozhraní USB 2.0. Zároveň jsou kompatibilní s mikrokontroléry řady Kinetis K20, které jsou na jádře ARM® Cortex®-M4 a se všemi dalšími mikrokontroléry řad KL1x – KL4x. Zařízení začínají na 32 KB paměti FLASH s velikostí 5 x 5 mm. Každé zařízení kombinuje vysoký výkon s ohledem na co nejnižší spotřebu se všemi základními periferiemi [4].

V tabulce 2 je srovnání 2 mikrokontrolerů řady KL2.

Tabulka 2 Srovnání mikrokontrolerů řady KL2

	Kinetis KL2	
	KL24	KL25
Procesor	32-bit do 48MHz	32-bit do 48MHz
Správa napětí	127 - 800 nA - Very low leakage stop mode, 0 current 4.4 uA - Very low power stop mode 350 uA - Stop mode 3 mA - Wait mode 5 mA - Run mode	
Hodiny	32 kHz oscilátor v low power módu 32 MHz krystalový oscilátor Víceúčelový zdroj hodin	
FLASH	až 64 KB	až 128 KB
RAM	až 8 KB	až 16 KB

I/O	Víceúčelový I/O	Víceúčelový I/O Nízkonapěťový dotykový snímač (TSD)
DMA	4 kanály	5 kanálů
Čítače/Časovače	6 kanálový časovač/PWM 2x 2-kanálový časovač/PWM 16-bit časovač s nízkým odběrem proudu Hodiny reálného času WatchDog	7 kanálový časovač/PWM 2x 2-kanálový časovač/PWM 16-bit časovač s nízkým odběrem proudu Hodiny reálného času WatchDog
Komunikační rozhraní	2x Universal Asynchronous Receiver/Transmitter (UART) Low power UART modul USB plná/nízká rychlost 2x 8-bit SPI modul 2x I ² C modul	
Analogová rozhraní	12-bit SAR ADC převodník Analogový komparátor obsahující 6-bit DAC převodník s programovatelným vstupem	16-bit SAR ADC převodník 12-bit DAC převodník Analogový komparátor obsahující 6-bit DAC převodník s programovatelným vstupem
Teploty	-40 – 105 °C	
Napájení	1,71 – 3,6V	
Ostatní	80-bit unikátní identifikační číslo pro každý čip	

2.1.3 NXP LPC110x

Mikrokontroléry od firmy NXP jsou postaveny na architektuře ARM® Cortex®-M0. Procesor je 32 bitový s maximální frekvencí až 50 MHz. Nabízejí vysoký výkon, nízkou spotřebu, jednoduchou instrukční sadu a paměťové adresování [5].

V tabulce 3 je srovnání 2 mikrokontrolérů řady LPC111x.

Tabulka 3 Srovnání mikrokontrolérů řady LP110x

	NXP LPC110x	
	LPC1112FDH28	LPC1115FBD48
Procesor	32-bit do 50MHz	32-bit do 50MHz
Správa napětí	Integrovaný PMU (Power Management Unit) 220 nA - Deep power down mode 6 uA - Deep sleep mode 2 mA - Sleep mode 3 - 9 mA - Active mode	
Hodiny	25 MHz krystalový oscilátor 12 MHz interní RC oscilátor PLL - umožňuje procesoru běžet na vysoké frekvenci bez potřeby vysokofrekvenčního	

	oscilátoru	
FLASH	16 KB	64 KB
RAM	4 KB	8 KB
I/O	22 víceúčelových I/O pinů Vysoký proud 20 mA na výstupní pin	42 víceúčelových I/O pinů Vysoký proud 20 mA na výstupní pin
Čítače/Časovače	4 víceúčelové čítače/časovače WatchDog	
Komunikační rozhraní	Universal Asynchronous Receiver/Transmitter (UART) SPI kontrolér I ² C rozhraní	Universal Asynchronous Receiver/Transmitter (UART) 2x SPI kontrolér I ² C rozhraní
Analogová rozhraní	10-bit ADC převodník s multiplexováním až nad 5 piny	10-bit ADC převodník s multiplexováním až nad 8 piny
Teploty	-40 – 85 °C	
Napájení	1,8 – 3,6 V	
Ostatní	In-system programming (ISP) In-application programming (IAP)	

2.2 Výběr mikrokontroléru

Z výše uvedených MCU jsem se rozhodl vybrat mikrokontrolér EFM 32 Tiny Gecko EFM32TG210. Hlavním důvody jsou:

- O něco nižší spotřeba než u ostatních mikrokontrolérů.
- Volně stažitelné vývojové prostředí (po free registraci)
- Nižší cena

2.3 Srovnání bezdrátových modulů

Bezdrátový modul bude sloužit k poslání dat, naměřených ze senzorů, do měřicí stanice.

2.3.1 Analog Devices ADF7242

Zařízení ADF7242 je vysílač a přijímač s vysokým výkonem a nízkou spotřebou. Zařízení operuje v pásmu 2,4 GHz. Je navrženo s důrazem na flexibilitu, odolnost, jednoduchost použití a nízkou spotřebu. Podporovaným standardem je IEEE 802.15.4. Dále jsou podporovány GFSK/FSK/GMSK/MSK modulace jak pro přenos paketů, tak pro streamovaný přenos. Přenosové rychlosti se pohybují mezi 50 kbps a 2 Mbps [6].

V tabulce 4 jsou uvedeny podrobnější informace o ADF7242.

Tabulka 4 Vlastnosti ADF7242

	Analog Devices
	ADF7242
Frekvenční rozsah	2400 - 2483.5 MHz
Rychlost přenosu	250 kbps 50 - 2000 kbps modulační rozsah
Modulace	GFSK/FSK/GMSK/MSK
Spotřeba	19 mA - mód přijímač 21,5 mA - mód vysílač (3 dBm) 1,7 uA - použití 32kHz krystalového oscilátoru
Citlivost	-95 dBm při 250 kbps
Výstupní výkon	-20 dBm až +4.8 dBm, kroky po 2 dBm
Napájení	1,8 - 3,6 V
Funkce	<ul style="list-style-type: none"> - Digitální měření RSSI - Integrovaný PLL filtr smyček, přepínač mezi vysílačem/přijímačem, sledování stavu baterie, teplotní senzor, 32 kHz RC a krystalový oscilátor - Rychlá automatická kalibrace VCO - Automatický RF syntetizér - Integrovaný Low-dropout regulátor - Podpora správy paketů - IEEE 802.15.4-2006 filtrování rámců - IEEE 802.15.4-2006 CSMA/CA - Flexibilní 256-byte vysílací/přijímací datový buffer

2.3.2 Microchip MRF24J40

MRF24J40 je zařízení kompatibilní se standardem IEEE 802.15.4 a operuje ve frekvenčním pásmu 2,4 GHz. MRF24J40 podporuje protokoly ZigBee™ a MiWi™ díky čemuž je ideálním řešením pro bezdrátové senzorní sítě, monitorování budov a spotřební aplikace. Vedle výše zmíněných protokolů obsahuje jednoduché 4 vodičové SPI rozhraní, integrované 20 MHz a 32 kHz krystalové oscilátory, hardwarovou podporu CSMA/CA a zabezpečení AES-128 [7].

V tabulce 5 jsou uvedeny vlastnosti MRF24J40.

Tabulka 5 Vlastnosti MRF24J40

	Microchip
	MRF24J40
Frekvenční rozsah	2405 - 2480 MHz
Rychlost přenosu	250 kbps 625 kbps Turbo mód
Modulace	O-QPSK

Spotřeba	19 mA - mód přijímač 23 mA - mód vysílač 2 uA - Sleep mód
Citlivost	-95 dBm
Výstupní výkon	+0 dBm typická hodnota s 36 dBm pracovním rozsahem
Napájení	2,5 - 3,6V
Funkce	<ul style="list-style-type: none"> - Integrovaný RSSI ADC převodník a I/Q DAC převodník - Integrovaný VCO s nízkým rušením - Frekvenční syntetizér a PLL filtrem smyček - Digitální kalibrace VCO a filtru - Diferenciální RF vstup/výstup s integrovaným prepínačem vysílač/přijímač - Integrovaný Low-dropout regulátor - Hardwarový CSMA/CA mechanismus, automatické poslání potvrzení a kontrola, zda nenastala chyba při přenosu - Schopnost automatického opětovného vyslání paketu - Hardwarová podpora AES-128 - Podporuje šifrování a dešifrování pro MAC vrstvu

2.3.3 Texas Instruments CC2520

CC2520 od firmy Texas Instruments je zařízení druhé generace s podporou protokolu ZigBee®. Zařízení je kompatibilní se standardem IEEE 802.15.4 a operuje ve frekvenčním pásmu 2,4 GHz. Díky vysokému rozmezí provozních teplot a nízké spotřebě lze zařízení použít v různých aplikacích. Dále obsahuje hardwarovou podporu pro manipulaci s rámci, kódování dat, ověřování dat a výběr volného kanálu. Tyto vlastnosti snižují zátěž mikrokontroléru [8].

V tabulce 6 jsou vidět vlastnosti CC2520.

Tabulka 6 Vlastnosti CC2520

	Texas Instruments
	CC2520
Frekvenční rozsah	2394 - 2507 MHz
Rychlost přenosu	250 kbps
Modulace	O-QPSK
Spotřeba	18,5 mA - mód přijímač 33,6 mA - mód vysílač +5 dBm 25,8 mA - mód vysílač +0 dBm < 1uA - power down mód
Citlivost	-98 dBm
Výstupní výkon	až +5 dBm

Napájení	1,8 - 3,8V
Funkce	<ul style="list-style-type: none"> - Digitální RSII/LQI podpora - Automatické určení volného kanálu v CSMA/CA Filtrování rámců Automatický kontrolní součet - 768 bytů RAM pro bufferování - Podpora AES-128 - Plně podporované zabezpečení MAC - Generátor přerušení - Generátor náhodných čísel

2.4 Výběr bezdrátového modulu

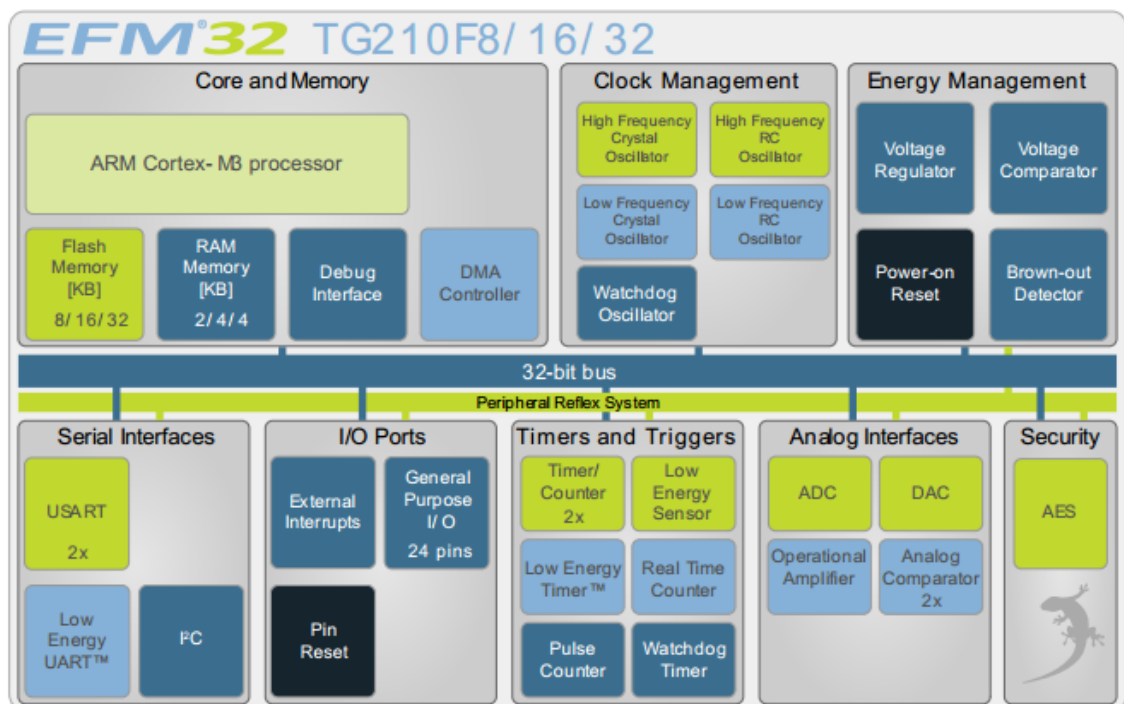
Vybral jsem si modul Microchip MRF24J40. Modul ADS7242 jsem i přes o něco málo nižší spotřebu zavrhl proto, že nepodporuje 128 bitové zabezpečení AES. Modul Texas Instruments CC2520 toto zabezpečení podporuje, ale má vyšší spotřebu při vysílání. Pro modul Microchip MRF24J40 také nahrává cena, která je nižší než u konkurence

3 Popis mikrokontroléru

3.1 Co je to mikrokontrolér?

Mikrokontrolér (dále MCU) obsahuje na rozdíl od obyčejného procesoru i některé integrované periferie. Může se jednat například o kódovou a datovou paměť, periferní obvody či generátor hodin. Klasický procesor používá pouze vnější obvody (vnější paměť apod.).

Každá firma dodává MCU dle potřeb zákazníka. Zákazník si může vybrat, jaké veškeré obvody na MCU potřebuje a také velikost pamětí či frekvenci procesoru. Výkonnost MCU se volí podle toho, pro jaké potřeby zákazník daný MCU potřebuje. Na obrázku 1 je vidět schéma mikrokontroléru Energy Micro EFM32 TinyGecko.



Obrázek 1 Schéma MCU Energy Micro EFM32 TG210, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26]

<<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

3.2 Popis periférií

Popis periférií se bude vztahovat k výše uvedenému MCU. Informace v této kapitole byly použity z [1].

3.2.1 Jádro a paměť

Procesor

Procesor použitý v EFM32 TG210 je postaven na architektuře ARM®. Samotné jádro je poté z rodiny Cortex®-M, zde konkrétně Cortex®-M3 [9]. Procesory ARM® Cortex®-M3 jsou navrženy pro velice rychlou odezvu na požadavky. Pro propojení s

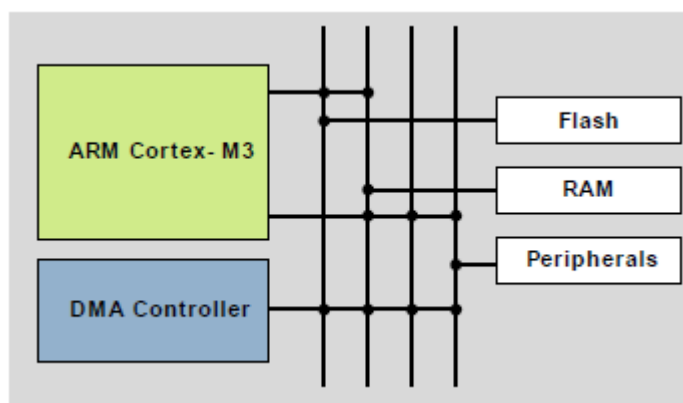
ostatními periferiemi se využívá 32 bitová APB (Advanced Peripheral Bus). Všechny periferie jsou navrženy tak, aby měli co nejmenší možný odběr energie.

Vlastnosti:

- Harvardská architektura
 - Rozdělení datové a kódové paměťové sběrnice (není úzké místo pro systémy s 1 sběrnicí)
- Možnost vykonávání 3 instrukcí současně
- Thumb-2 instrukční set – Vylepšená výkonnost a snížení odběru
- Efektivní instrukce pro násobení a dělení
 - Manipulace se 32 bity v jednom cyklu
 - Znaménkové a bezznaménkové operace mezi 2 a 12 cykly
- Atomická bitová manipulace
 - Přímý přístup k jednotlivým datovým bitům
 - Atomické operace nemohou být přerušeny ostatními aktivitami na sběrnicí
- Výkon procesoru je 1,25 DMIPS/MHz
- 24 bitový systémový časovač pro systémy reálného času (RTOS)
- Vhodné pro migraci systémů navržené na 8 nebo 16 bitové architektuře
- Integrované módy pro řízení spotřeby
 - Sleep mód pro okamžité uvedení MCU do režimu nízkého odběru
 - Sleep on exit mód pro uvedení MCU do režimu spánku po obslužení daného přerušení
 - Možnost snížení odběru řízením napájení jednotlivých systémových komponent

DMA řadič

DMA řadič (Direct Memory Access) provádí operace s pamětí nezávisle na procesoru. Hlavní výhodou je snížení spotřeby a zátěže na procesor a umožňuje systému zůstat v režimu nízké spotřeby například při přesunu dat z USART do RAM paměti. DMA řadič používá řadič PL230 μ DMA od ARM®. Každý z PL230 kanálů může být připojen do libovolné EFM32 periferie. Na obrázku 2 je vidět zapojení DMA 2 řadiče vzhledem k procesoru a pamětím.



Obrázek 2 Zapojení DMA řadiče, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Vlastnosti:

- Možné přenosy dat zahrnují
 - Přenos z RAM/FLASH do periferie
 - Přenos z RAM do FLASH
 - Přenos z periferie do RAM
 - Přenos z RAM/FLASH do RAM
- DMA řadič má 8 nezávislých kanálů
- Každý kanál má buď 1 (primární) nebo 2 (primární a sekundární) deskriptory
- Konfigurace pro každý kanál zahrnuje
 - Přenosový mód
 - Priorita
 - Počet slov
 - Velikost slova (8, 16, 32 bitů)
- Přenosový mód zahrnuje
 - Použití primárního nebo sekundárního DMA deskriptoru
 - Možnost přepínání za běhu mezi primárním a sekundárním deskriptorem
 - Použití primárního deskriptoru pro konfiguraci sekundárního
- Každý kanál má programovatelnou délku přenosu
- DMA kanál může být aktivovaný
 - Komunikačním modulem (USART, LEUART)
 - Časovačem
 - Analogovým modulem (DAC, ACMP, ADC)
 - Programem
- Programovatelné mapování mezi číslem kanálu a periferií – jakýkoli DMA kanál může být aktivován jakýmkoliv dostupným zdrojem
- Přerušování po úspěšném přenosu

Debug Interface (DBG)

DBG se používá k programování a ladění EFM32 TG zařízení. Pro připojení se používá 2 pinové sériové rozhraní.

Vlastnosti:

- DPB (Flash Patch and Breakpoint) jednotka
 - Poskytuje jednoduchý breakpoint (bod zastavení) nebo přemapování instrukce přístupu z FLASH na jiné umístění v RAM
- DWT (Data Watch point and Trace) jednotka
 - Používá se k zastavení procesoru nebo k vygenerování informací o sledovaných datech
- ITM (Instrumentation Trace Macrocell)
 - Do paměťového registru ITM se zapíše data. Debugger tyto data může sesbírat a zpracovat nebo zobrazit

FLASH paměť

FLASH paměť dokáže uchovat jakákoliv data na delší dobu. Většinou se jedná o zdrojové kódy aplikace, speciální uživatelská data a bezpečnostní informace. FLASH paměť se většinou programuje přes DBG. Je však možné jí smazat a nebo do ní zapisovat přímo z programu. Toho využívá například bootloader.

Vlastnosti:

- Velikost až 32 kB
- Velikost stránky 512 bajtů (minimální velikost pro mazání)
- Minimálně 20 000 mazacích cyklů
- Uchování dat po dobu minimálně 10 let při 85°C
- Vyhrazené bity pro uzamčení paměti před nepovoleným zápisem či mazáním
- Uchování dat v jakémkoliv stavu

SRAM paměť

Hlavním úkolem SRAM paměti je uchovávat data běžící aplikace. Navíc je možné vykonání instrukcí přímo ze SRAM a DMA řadič může být použit pro přenos dat mezi SRAM, FLASH a periferiemi.

Vlastnosti:

- Velikost až 4 kB
- Umožňuje tzv. bit-banding, což je mapování celého slova v paměti na 1 bit v bit-banding oblasti
- Bloky o velikosti 4 kB mohou být nezávisle na sobě odpojeny od napájení, jestliže nejsou zrovna používány
- Uchování dat ve všech napájecích módech vyjma Shutoff módu

3.2.2 Řízení periferií

Peripheral Reflex Systém (PRS)

PRS je síť, která umožňuje přímou komunikaci různých periferních modulů mezi sebou bez zásahu procesoru. Periferní modul, který vysílá reflexní signál, se nazývá producent. PRS směřuje tento signál ke konzumentovi, který po přijetí provede příslušné akce.

Vlastnosti:

- 8 konfigurovatelných propojovacích kanálů
 - Každý kanál může být připojen k jakémukoliv producentovi
 - Konzumenti si mohou vybrat, na jakém kanálu budou poslouchat
 - Lze nastavit, jaké hrany detekovat (rostoucí, klesající nebo obě)
- Programem kontrolovaný výstup kanálu

3.2.3 Správa hodin (CMU)

Clock Management Unit (CMU) se stará o řízení oscilátorů a hodin integrovaných v procesoru. CMU umožňuje zapínání a vypínání oscilátorů včetně jejich konfigurace. Dále umožňuje vypínat a zapínat hodiny individuálně všem periferním modulům. Velký stupeň flexibility umožňuje aplikaci minimalizovat spotřebu. To lze docílit vypnutím energie pro nepotřebné periferie nebo neaktivní oscilátory.

Vlastnosti:

- Několik zdrojů hodin
 - 1 – 28 MHz Vysokofrekvenční RC oscilátor (HFRCO)
 - 4 – 32 MHz Vysokofrekvenční krystalový oscilátor (HFXO)
 - 32 768 Hz Nízkofrekvenční RC oscilátor (LFRCO)
 - 32 768 Hz Nízkofrekvenční krystalový oscilátor (LFXO)
 - 1 kHz Extrémně nízkofrekvenční RC oscilátor (ULFRCO)
- Nízká spotřeba jednotlivých oscilátorů
- Nízké startovací časy
- Oddělené obvody u vysokofrekvenčních hodin pro jádro a periferní hodiny
- Individuální hodiny pro každou periferii
- Některé hodiny mohou být vyvedeny na výstupní piny pro externí použití
- Přídavný 1 – 28 MHz RC oscilátor pro programování FLASH paměti

Hodiny

High Frequency Clock (HFCLK)

Tyto hodiny jsou používány CMU pro řízení 2 nezávislých hodin HFCORECLK a HFPERCLK. HFCLK mohou být řízeny vysokofrekvenčním oscilátorem (HFRCO nebo HFXO) nebo nízkofrekvenčním oscilátorem (LFRCO nebo LFXO). Jako výchozí

je vybraný HFRCO oscilátor. Pro většinu aplikací je vysokofrekvenční oscilátor lepší volba.

High Frequency Core Clock (HFCORECLK)

Tyto hodiny řídí hlavní moduly. Jedná se o procesor a další moduly úzce spjaté s procesorem např. DMA řadič. Některé moduly řízené těmito hodinami je možné odpojit od hodin úplně, jestliže nejsou zrovna používány. Jakékoliv změny v nastavení jsou hned aktualizovány.

High Frequency Peripheral Clock (HPPERCLK)

Tyto hodiny řídí vysokofrekvenční periferie. Všechny periferie řízené těmito hodinami je možné kdykoliv od hodin odpojit, jestliže nejsou zrovna používány. Jakékoliv změny v nastavení jsou hned aktualizovány.

Low Frequency A Clock (LFACLK)

Tyto hodiny se používají pro nízkonapěťové periferie A. Hodiny mohou být řízeny 4 různými zdroji: LFRCO, LFXO, HFCORECLK/2 a ULFRCO a mohou být také kdykoliv vypnuty. Implicitně jsou tyto hodiny řízeny LFRCO, avšak LFRCO je implicitně vypnut, proto jsou i tyto hodiny vypnuty. Při použití HFCORECLK/2 mohou být nízkonapěťové periferie A použity jako vysokofrekvenční periferie.

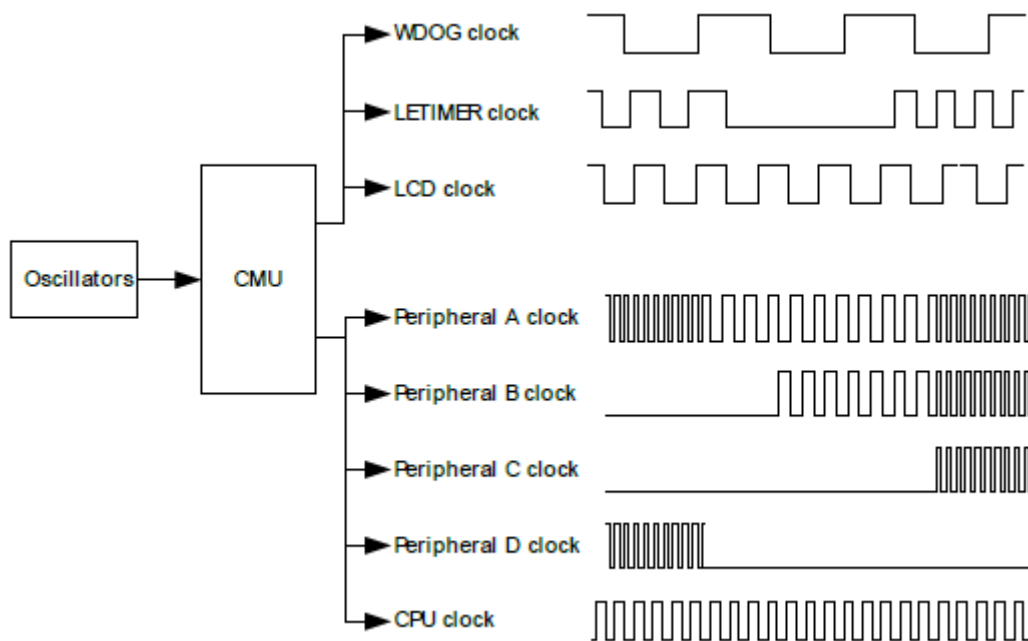
Low Frequency B Clock (LFBCLK)

Tyto hodiny se používají pro nízkonapěťové periferie B. Hodiny mohou být řízeny 4 různými zdroji: LFRCO, LFXO, HFCORECLK/2 a ULFRCO a mohou být také kdykoliv vypnuty. Implicitně jsou tyto hodiny řízeny LFRCO, avšak LFRCO je implicitně vypnut, proto jsou i tyto hodiny vypnuty. Při použití HFCORECLK/2 mohou být nízkonapěťové periferie B použity jako vysokofrekvenční periferie

Watchdog Time Clock (WDOGCLK)

Watchdog může být řízen: LFRCO, LFXO nebo ULFRCO. Více o Watchdog viz. kapitola 3.2.6.

Rozvod hodin je vidět na obrázku 3.



Obrázek 3 Rozvod hodin, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Oscilátory

Oscilátor je obvod, který produkuje opakující se a kmitající signál, často se jedná o sinusoidu. Oscilátor převádí stejnosměrný proud na střídavý. Zde slouží ke generování hodinového signálu

Každý oscilátor má různý čas spuštění. Pro RC oscilátory je tento čas neměnný, avšak pro krystalové oscilátory je možné tento čas nastavit. Na konci spuštění je nastaven daný příznak značící že se oscilátory spustily a je možné používat hodiny. Rychlejší spuštění je vhodné použít v případě externích hodin, naopak pomalejší spuštění je vhodné v případě, kdy hodinový signál pochází přímo z krystalu. Každému oscilátoru jsou přiřazeny jednotlivé bity značící stav daného oscilátoru:

- ENABLED – oscilátor je zapnut
- READY – oscilátor byl spuštěn a běží
- SELECTED – oscilátor byl spuštěn, běží a je použit jako zdroj hodinového signálu

Jako výchozí oscilátor pro HFCLK je použit HFRCO. Hlavní výhodou tohoto oscilátoru je velice rychlý čas spuštění. Výchozí frekvence po restartu je 14 MHz. Pro použití jiného oscilátoru je nejprve nutné tento oscilátor spustit. Kdyby nastalo přepnutí na vypnutý oscilátor, procesor by se vypnul a bylo by nutné provedení restartu MCU. Jakmile oscilátor, na který chceme přepnout, je spuštěn, lze na něj přepnout. Je nutné pamatovat na to, že krystalové oscilátory mají delší dobu spuštění než RC oscilátory.

3.2.4 Správa napájení (EMU)

Energy Management Unit (EMU) spravuje jednotlivé módy (EM) pro snížení spotřeby elektrického proudu v MCU EFM32TG. Každý mód spravuje, zda jsou procesor a různé periferie k dispozici. Módy jsou značeny od EM0 po EM4.

Jednotlivé módy se různí v:

- Spotřeba energie
- Aktivita procesoru
- Reakční čas
- Impulsy probuzení
- Aktivní periferie
- Dostupné zdroje hodin

Do nízkonapěťových módů EM1 – EM4 je možné se přepnout kdykoliv příslušným příkazem v průběhu programu. Z módů EM1 – EM3 je možné se vrátit za pomoci několika impulsů probuzení. Z módu EM4 se lze vrátit pouze za pomoci power-on-reset (viz níže), restartu přivedeného na externí pin nebo EM4 GPIO požadavku na probuzení. Níže je uveden popis jednotlivých módů.

EM0

- Vysokofrekvenční oscilátor je aktivní
- Vysokofrekvenční hodiny pro hlavní moduly jsou aktivní
- Funkce všech periférií jsou k dispozici

EM1

- Vysokofrekvenční oscilátor je aktivní
- Vysokofrekvenční hodiny pro hlavní moduly jsou neaktivní
- Vysokofrekvenční hodiny pro periferie jsou aktivní
- Funkce všech periférií jsou k dispozici

EM2

- Vysokofrekvenční oscilátor je vypnutý
- Vysokofrekvenční hodiny pro hlavní moduly jsou neaktivní
- Nízkofrekvenční oscilátor je aktivní
- Nízkofrekvenční hodiny pro hlavní moduly jsou zapnuty
- Funkce nízkonapěťových periférií jsou k dispozici
- Probuzení na základě asynchronního přerušení z nějaké periferie
- Hodnoty v RAM a registru jsou zachovány

EM3

- Nízko i vysokofrekvenční oscilátory a hodiny jsou vypnuty

- Probuzení na základě asynchronních přerušení nebo I²C rozpoznání adres
- Watchdog a některé nízkonapěťové periferie jsou k dispozici za použití ULFRCO
- Funkce ostatních periférií jsou vypnuty
- Hodnoty v RAM a registru jsou zachovány

EM4

- Všechny oscilátory a hodiny jsou vypnuty
- Hodnoty v RAM a registru nejsou zachovány
- Možnost probuzení při příchodu signálu reset na externí pin nebo piny, které podporují EM4 probuzení (probuzení bez úplného restartu)

Voltage regulator

Úkolem stabilizátoru napětí je stabilizovat hodnotu napětí na svém výstupu, tedy zabezpečit, aby se tato hodnota měnila co možná nejméně při změnách některých veličin, např. vstupního napětí, výstupního proudu či teploty. Stabilizátory také snižují zvlnění, tedy střídavou složku stejnosměrného napětí.

Voltage comparator (VCMP)

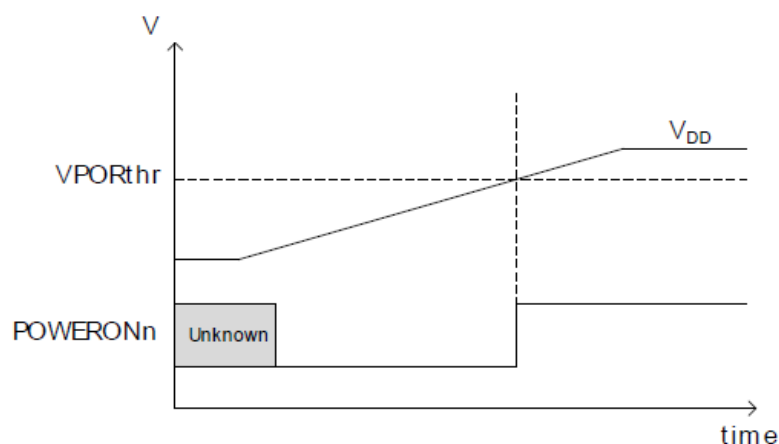
VCMP slouží k monitorování vstupního napětí z běžícího programu. Je možné generovat přerušení, jestliže se napětí dostane mimo dané mezní hodnoty. Samotný VCMP generuje pouze přerušení.

Vlastnosti:

- Monitorování vstupního napětí
- Nastavitelný čas odpovědi
- Generování asynchronního přerušení na hranách
 - Rostoucích
 - Klesajících
 - Rostoucích i klesajících
- Funkční v módech EM0 – EM3
- Nastavitelný výstup, jestliže je VCMP neaktivní z důvodu zabránění nechtěných přerušení

Power-On Reset (POR)

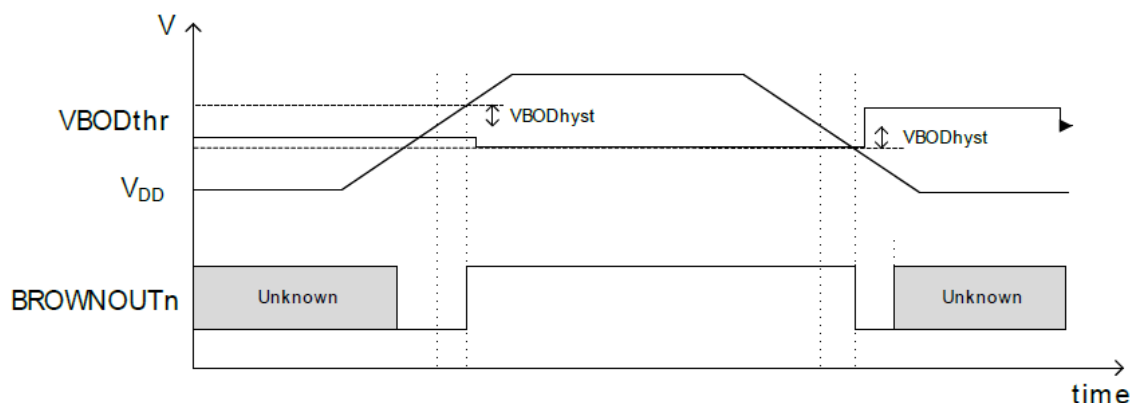
Úkolem POR je zajistit to, aby se MCU nespustil dříve, než se napájecí napětí (V_{DD}) dostane přes minimální povolenou hodnotu (V_{PORthr}). Předtím, než V_{DD} této hodnoty dosáhne, je MCU uchovávan ve stavu reset. Na obrázku 4 je vidět funkce POR. Dokud napětí nedosáhne požadované úrovně, je signál $POWERONn$ udržován v nízké úrovni. Po dosažení hranice se přepne do vysoké úrovně.



Obrázek 4 Power-On Reset, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Brown-Out Detector Reset (BOD)

BOD slouží k neustálému monitorování napětí. Kdykoliv se sledované napětí dostane pod hodnotu V_{BODthr} , příslušný BROWNOUTn vodič je držen v nízké úrovni. MCU EFM32TG má 4 BOD. Na obrázku 5 je vidět funkce BOD.



Obrázek 5 Brown-Out Detector Reset, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

3.2.5 I/O Porty

General Purpose Input/Output

U MCU EFM32TG jsou General Purpose Input/Output (GPIO) piny organizovány do portů s až 16 piny u každého portu. Tyto piny mohou být individuálně nakonfigurovány jako vstupní nebo jako výstupní. Některé piny GPIO mohou být překryty piny nějaké periferie, např. USART komunikační rozhraní. GPIO podporuje až 16 vnějších asynchronních přerušení na pinech a umožňuje volat přerušení z kteréhokoliv pinu na zařízení. Také vstupní hodnota na nějakém pinu může být přeposlána jiným periferiím. Zařízení EFM32TG210 obsahuje 24 vstupně/výstupních pinů.

Vlastnosti:

- Každý pin lze libovolně nakonfigurovat jako
 - Třístavový vstup
 - Výstup s Push-pull
 - Výstup s otevřeným kolektorem
 - Výstup s otevřeným kolektorem a pull-up
 - Výstup s otevřeným kolektorem a pull-down
 - Nastavitelný maximální výstupní proud
 - 0,5 mA
 - 2 mA
 - 6 mA
 - 20 mA
- Vzbuzení na vybraných GPIO pinech v EM4 módu
- Vstupní filtr pro omezení poruch
- Přerušení
 - 2 přerušovací vodiče pro až 16 očekávaných zdrojů
 - Lze nastavit pro libovolné GPIO piny
 - Oddělené funkce pro povolení, stav, nastavení a smazání registrů
 - Generování přerušení na rostoucích, klesajících nebo na obou hranách
 - Vzbuzení z módů EM0 – EM3
- Konfiguraci lze uzamknout proti nechtěným změnám

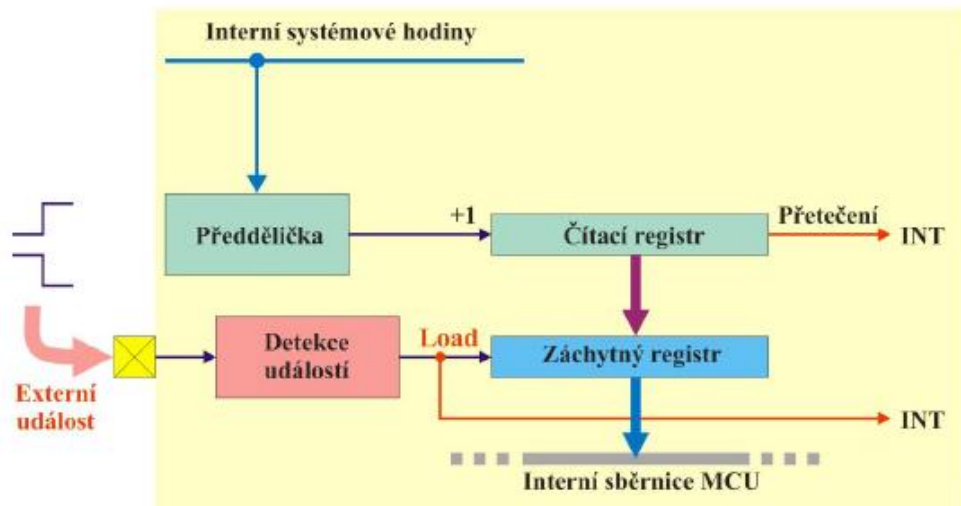
Pin reset

Jakmile se pin RESETn uvede do nízké úrovně, vyvolá se reset celého zařízení. Pin RESETn také zahrnuje pull-up odpor a není ho tudíž nutné nikam zapojovat, jestliže nechceme vyvolávat reset z žádného vnějšího zdroje. K tomuto pinu je také připojen filtr zabraňující chybám, které by mohly vyvolat náhodný reset zařízení.

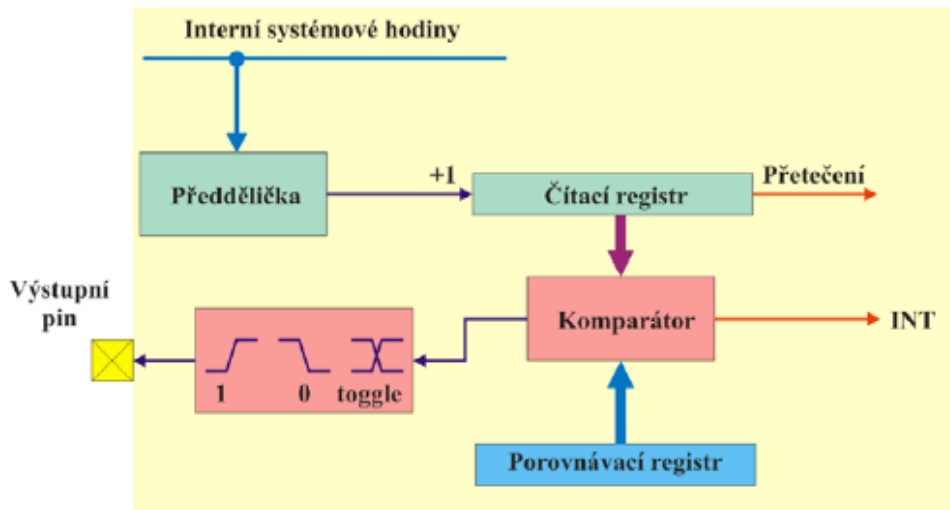
3.2.6 Časovače a trigger

Časovač si udržuje informace o času a počítá události, generuje výstupní průběh a při dosažení příslušné meze spouští událost v jiných perifériích. EFM32TG obsahuje tři 16 bitové časovače s capture kanálem pro vstup a compare/PWM kanály pro výstup.

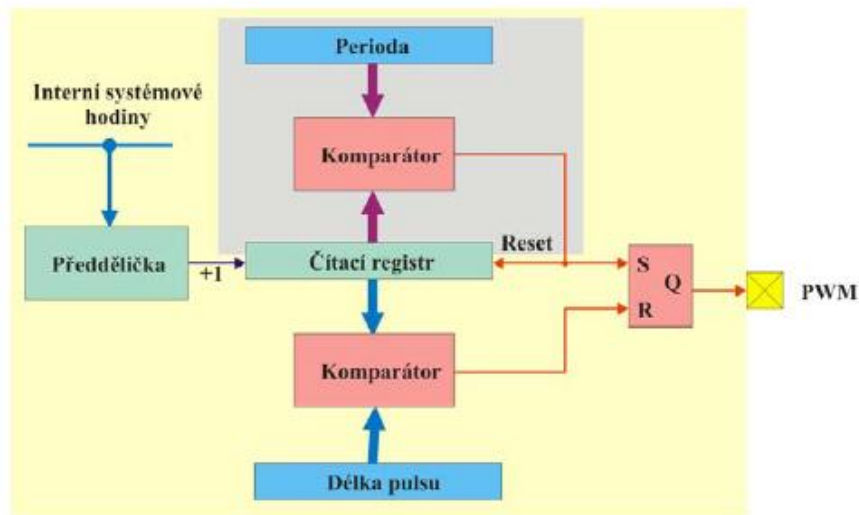
Na obrázku 6 je vidět funkce capture, která zjistí přesný čas dané události. Obrázek 7 znázorňuje funkci compare, která porovnává čítací registr s porovnávacím registrem a nastavuje výstup na nízkou či vysokou hranu. Obrázek 8 ukazuje funkci PWM, která generuje periodický obdélníkový průběh s nastavitelnou šířkou. Šířka závisí na délce pulzu a na periodě [2].



Obrázek 6 Funkce capture, DUDÁČEK Karel. Navrhování mikro počítačových systémů. [online] [cit. 2014-03-27] <<http://home.zcu.cz/~dudacek/NMS/PDF/>>



Obrázek 7 Funkce compare, DUDÁČEK Karel. Navrhování mikro počítačových systémů. [online] [cit. 2014-03-27] <<http://home.zcu.cz/~dudacek/NMS/PDF/>>



Obrázek 8 Funkce PWM, DUDÁČEK Karel. Navrhování mikropočítačových systémů. [online] [cit. 2014-03-27] <<http://home.zcu.cz/~dudacek/NMS/PDF/>>

Vlastnosti:

- 16 bitový čítač s automatickým plněním registru
 - Vlastní 16 bitový plnicí registr který slouží jako maximum čítače
- 3 compare/capture kanály
 - Individuální konfigurace pinu buď jako capture vstupní nebo compare/PWM výstupní
- Několik čítacích módů
 - Přičítání
 - Odčítání
 - Přičítání/odčítání
 - Kvadrurní dekodér
 - Správa a čítání z externích pinů
- 2 čítací módy
- Řízení časovače z PRS nebo externího pinu
 - Start
 - Stop
 - Opětovné nahrání a start
- Zdroje hodin
 - HFPERCLK
 - Externí pin
 - PRS
- Ladící mód
 - Lze nakonfigurovat, aby čítač běžel, i když je procesor zastaven
- Generování přerušování při:
 - Přetečení
 - Podtečení
 - Událost při capture/compare

Real Time Counter (RTC)

RTC obsahuje 24 bitový čítač a je časován buď 32 768 Hz krystalovým oscilátorem a 32 768 Hz RC oscilátorem nebo 1 kHz RC oscilátorem. Vedle módů EM0 a EM1 je k dispozici i v módu EM2. Toto dělá RTC ideální k udržování informací o aktuálním času, jelikož RTC je zapnut v módu EM2 zatímco je většina jiných zařízení vypnuta. Při použití 1 kHz ULFRCO jako vstupních hodin lze RTC použít pro udržení času i v módu EM3.

RTC používá 2 porovnávací kanály. Ty mohou být použity k vyvolání přerušení, které může vzbudit zařízení z módu nízké spotřeby.

Vlastnosti:

- 24 bitový čítač reálného času
- 2 porovnávací registry
 - Při shodě může být zařízení vzbuzeno z módů EM1 a EM2
 - Oba porovnávací registry mohou spustit LETIMER
 - Události při shodě jsou k dispozici ostatním perifériím pomocí PRS

Low Energy Timer (LETIMER)

LETIMER je 16 bitový časovač, který je vedle módů EM0 a EM1 k dispozici i v módech EM2 a EM3. Díky tomu může být použit k měření času kdy je většina zařízení vypnuta. Tím je možné vykonávání jednoduchých úloh zatímco je odběr systému udržován na minimu.

Vlastnosti:

- 16 bitový časovač
- 2 porovnávací registry
- 8 bitový opakovací registr se 2 zásobníky
- Stejný zdroj hodin jako pro Real Time Counter (RTC)
- LETIMER může být spuštěn programem nebo RTC událostí
- Generování přerušení při:
 - Shodě při porovnání
 - Podtečení časovače
- Lze spustit při ladění

Low Energy Sensor Interface (LESENSE)

LESENSE je rozhraní s nízkou spotřebou pro měření dat ze senzorů. Naměřené výsledky mohou být zpracovány LESENSE dekodérem. Výsledky mohou být také uloženy ve vyrovnávací paměti a využity procesorem nebo DMA řadičem k dalšímu zpracování.

LESENSE může operovat v módech EM0 – EM2 a může být probuzen procesorem při předem nastavitelných událostech.

Vlastnosti:

- Až 16 senzorů
- Autonomní monitorování senzorů v módech EM0 – EM2
- Generování přerušení při událostech na senzorech
- Konfigurovatelné signály pro externí senzory
- Cyklická vyrovnávací paměť pro uchování až 16 výsledků
- Podpora různých druhů senzorů

Pulse Counter (PCNT)

PCNT se používá pro počítání příchozích pulzů na jednom vstupu nebo pro dekódování kvadraturních vstupů. Jako zdroj hodin lze použít LFACLK (při módu EM0 – EM2). Počet pulsů se poté počítá na pinu PCNTn nebo lze pin použít jako zdroj externích hodin (EM0 – EM3), který poté zajišťuje počítání pulsů na PCNT a také přístup do registru.

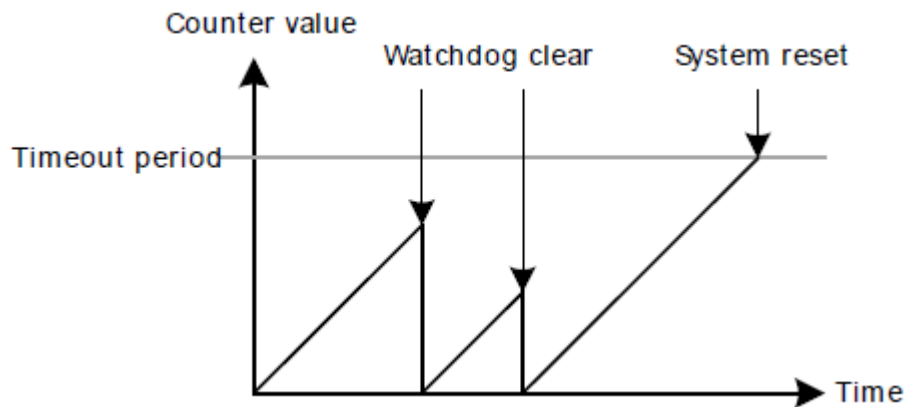
Vlastnosti:

- 16 bitový čítač s automatickým plněním registru
- Čítací mód pro jeden vstup (EM0 – EM2)
- Čítací mód za použití externích hodin (EM0 – EM3)
- Kvadraturní dekódování při použití externích hodin (EM0 – EM3)
- Generování přerušení při:
 - Podtečení
 - Přetečení
 - Změně směru při kvadraturním dekódování

Watchdog

Účel Watchdog časovače je generování signálu reset v případě chyby v systému. Chyby může nastat nějakou událostí zvenčí nebo chybou při vykonávání programu.

Funkce Watchdog časovače je vidět na obrázku 9. Watchdog zvyšuje s každým tikem hodin danou hodnotu. Tato hodnota musí být v průběhu vykonávání programu opakovaně nulována. Jakmile k vynulování nedojde, Watchdog vyvolá reset celého systému.



Obrázek 9 Funkce Watchdog, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

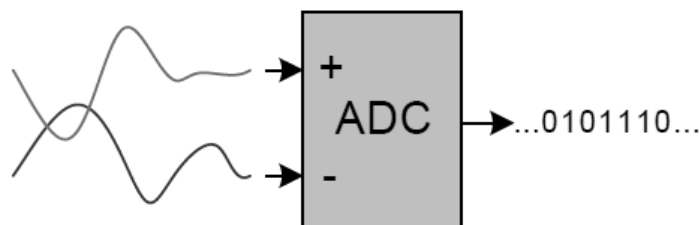
Vlastnosti:

- Vstup hodinového signálu z následujících oscilátorů
 - Vnitřní 32 768 Hz RC oscilátor
 - Vnitřní 1 kHz RC oscilátor
 - Externí 32 768 Hz XTAL oscilátor
- Nastavení časového limitu mezi 9 a 256k hodinovými signály
- Možnost zmrazení nebo ponechání běhu při přechodu do ladícího módu
- Možnost zablokovat procesoru přechod do EM4
- Možnost zablokovat CMU aby vypnul hodiny aktuálně použité pro Watchdog.

3.2.7 Analogová rozhraní

Analog to Digital Converter (ADC)

ADC převodník se používá pro převod analogových signálů na digitální data. Jednoduché schéma je vidět na obrázku 10. Rozlišovací schopnost ADC je až 12 bitů a umožňuje snímkování až 1M snímků za sekundu. Jako vstup lze nastavit až 8 externích pinů nebo 6 interních signálů.



Obrázek 10 ADC převodník, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

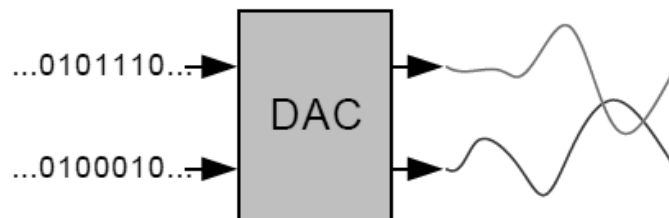
Vlastnosti:

- Programovatelná rozlišovací schopnost(6/8/12 bitů)
 - Maximálně 1 MSPS při 12 bitech

- Maximálně 1,86 MSPS při 6 bitech
- 18 vstupních kanálů
 - 8 externích kanálů
 - 6 interních kanálů
 - Včetně teplotního čidla
 - 4 externí diferenciální kanály
- Integrovaný vstupní filtr
- Programovatelná konverze posloupnosti
 - Až 8 konfigurovatelných vzorků v jedné posloupnosti
 - Výběr pinů obsažených v posloupnosti
 - Spuštěno programem nebo vstupem z PRS
 - Lze provést pouze jednou nebo opakovaně
 - Možnost převzorkování
 - Při přepsání ještě nepřečtených dat se nastaví příznak přetečení
- Programovatelná samostatná konverze
 - Spuštěna programem nebo vstupem z PRS
 - Lze provést pouze jednou nebo opakovaně
 - Možnost převzorkování
 - Při přepsání ještě nepřečtených dat se nastaví příznak přetečení
- Podpora HW převzorkování
- Volitelné napětí pro snímání
- Generování přerušení a/nebo DMA požadavky
 - Dokončení převodu
 - Nastání přetečení

Digital to Analog Converter (DAC)

DAC převodník slouží k převodu digitální hodnoty na analogovou hodnotu reprezentovanou příslušným napětím. Rozlišovací schopnost DAC je 12 bitů. DAC obsahuje 2 výstupní kanály, které mohou být zkombinované do jednoho diferenciálního výstupu. Na obrázku 11 je vidět jednoduché schéma.



Obrázek 11 DAC převodník, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

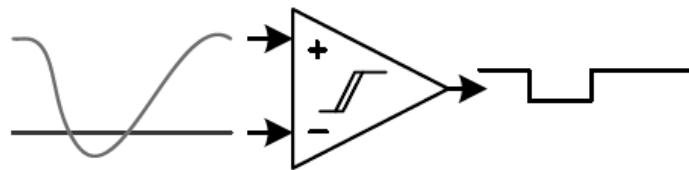
Vlastnosti:

- 500k vzorků za sekundu
- 2 výstupní kanály

- Lze zkombinovat do 1 diferenciálního výstupu
- Volitelné napětí
- Generování přerušení po skončení převodu
 - Každý kanál má vlastní identifikátor
- DMA požadavek po skončení převodu
 - Oddělené pro každý kanál

Analog Comparator (ACMP)

ACMP se používá pro porovnání napětí na 2 analogových vstupech. Výstupem je digitální signál, který značí, jaké napětí je vyšší. Vstupem může být nějaký vnitřní signál nebo může být čten z externího pinu. Obrázek 12 znázorňuje jednoduché schéma.



Obrázek 12 Analogový porovnávací obvod, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Vlastnosti:

- 8 volitelných externích pozitivních vstupů
- 8 volitelných externích negativních vstupů
- 5 volitelných interních vstupů
- Mód nízkého odběru pro vnitřní napájecí napětí V_{DD}
- Volitelný čas odpovědi
- Asynchronní generování přerušení na volitelných hranách
 - Rostoucí hrana
 - Klesající hrana
 - Rostoucí i klesající hrana
- Lze použít v módech EM0 – EM3

3.2.8 Sériová rozhraní

Universal Synchronous Asynchronous Receiver/Transmitter (USART)

USART je velmi flexibilní vstupně/výstupní sériové rozhraní. Podporuje plně duplexní UART komunikaci, rozhraní SPI, MicroWire a 3-wire. Toto rozhraní dovoluje komunikaci s různými externími zařízeními. Je zde na výběr velký počet pracovních módů, formátů přenášeného rámce a přenosových rychlostí. Trojitá vyrovnávací paměť a podpora DMA řadiče dovoluje vysokou přenosovou rychlost při minimální zátěži procesoru a je možné vysílat a přijímat velké rámce v módu EM1.

Vlastnosti:

- Asynchronní a synchronní (SPI) komunikace
- Full duplex a half duplex
- Oddělený vysílací a přijímací buffer s oddělenými posuvnými registry
- Programovatelná baudová rychlost
- Maximální bitová rychlost
 - SPI master mód, rychlost periferních hodin/2
 - SPI slave mód, rychlost periferních hodin/8
 - UART mód, rychlost periferních hodin/16, 8, 6 nebo 4
- Asynchronní mód podporuje
 - Detekce falešného start-bitu
 - Multi-processor mód
 - Generování a detekce přerušení
- Synchronní mód podporuje
 - Pro všechny 4 SPI hodiny nastavení polarity a fáze
 - Master a slave mód
- Nastavitelný počet datových bitů, 4 – 16 (plus bit polarity, jestli je povolen)
- Nastavitelný počet stop bitů v asynchronním módu – 0,5; 1; 1,5; 2
- Detekce HW kolize

Operační módy

USART může operovat buď v synchronním nebo asynchronním módu.

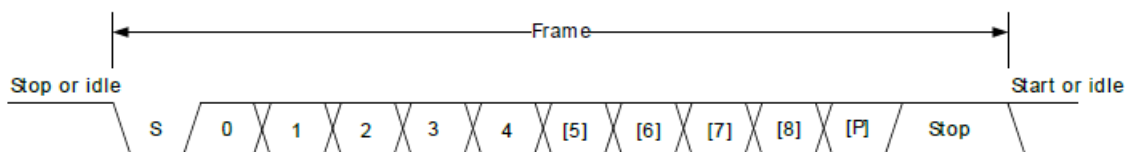
V synchronním módu je oddělený hodinový signál přenášen spolu s vysílanými daty. Signál je generován aktuálním master zařízením a jak master, tak i slave zařízení vzorkují a přenášení data vzhledem k těmto hodinám. Synchronní komunikace je kompatibilní s Serial Peripheral Interface Bus (SPI).

V asynchronním módu není vysílán žádný oddělený hodinový signál. Přijímač musí tudíž sám rozeznat, kdy daná data vzorkovat. Aby se toho docílilo, přidávají se synchronizační bity do přenášených dat. Toto však vede ke zvýšení režie.

Asynchronní operace

Formát rámce

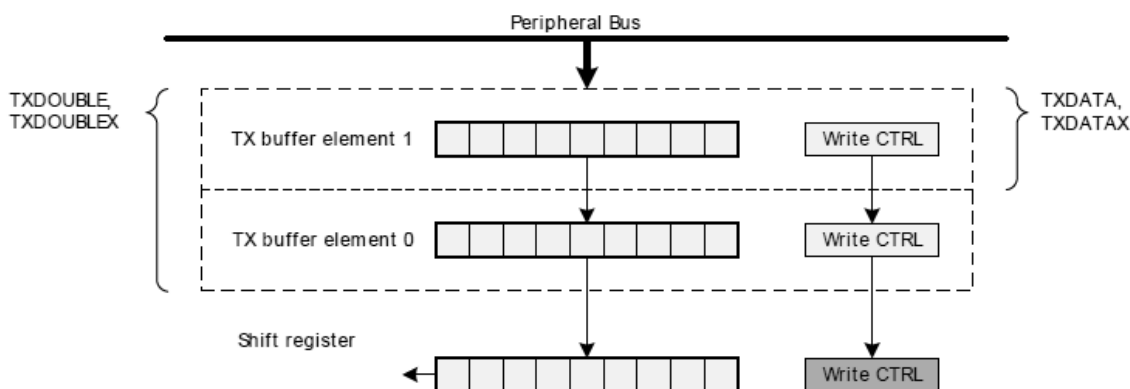
Formát rámce použitého v asynchronním módu se skládá z množiny datových bitů, ke kterým jsou připojeny bity pro synchronizaci a volitelného bitu parity. Rámec začíná jedním start-bitem, při kterém se signál nastaví na nízkou úroveň na dobu jednoho bitu. Tento bit je použit zároveň pro synchronizaci. Následuje sekvence 4 – 16 datových bitů a volitelný bit parity. Na konci je několik stop-bitů, kde je signál nastaven na vysokou úroveň. To také značí konec rámce. Příklad rámce je vidět na obrázku 13.



Obrázek 13 Formát asynchronního rámce, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Vysílání dat

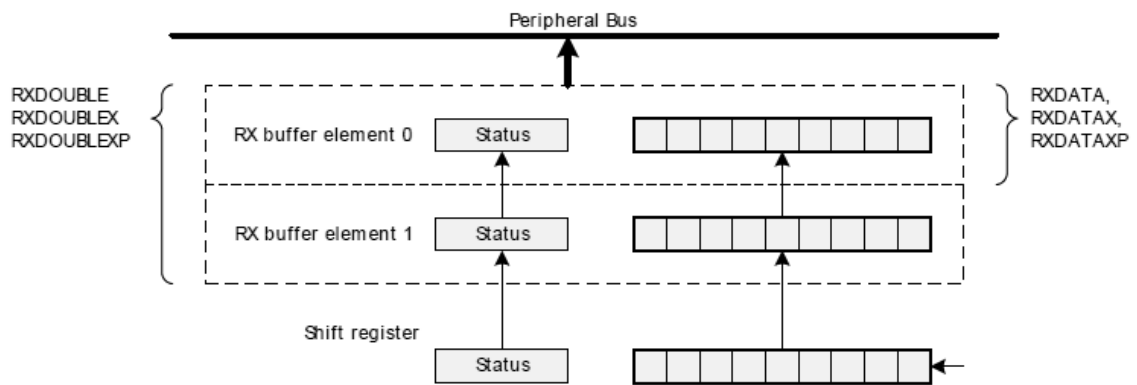
Vysílání je zahájeno zápisem dat do vysílacího bufferu. Když je vysílací posuvný registr prázdný a čeká na nová data, rámec z vysílacího bufferu je nahrán do posuvného registru a jestliže je vysílač povolen, přenos je zahájen. Poté, co byl rámec poslán, nový rámec (pokud je nějaký další na řadě) je nahrán do posuvného registru a přenos pokračuje. Jestliže je vysílací buffer prázdný, vysílač přejde do nečinného stavu a čeká, dokud nebude nějaký další rámec k dispozici. Na obrázku 14 je vidět předávání dat mezi buffery.



Obrázek 14 Předávání dat k vysílání, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Příjem dat

Jestliže je přijímač povolen, aktivně vzorkuje vstup a zjišťuje, zda nenarazí na přechod z vysokého signálu na nízký značící start-bit nějakého rámce. Jestliže je start-bit nalezen, začne příjem daného rámce, jestliže je posuvný registr prázdný a je připraven přijímat nová data. Poté, co byl rámec přijat, je přesunut z posuvného registru do přijímacího bufferu (obrázek 15). Tím se posuvný registr vyprázdní a je připraven pro nová data. Jestliže je přijímací buffer plný, přijatý rámec zůstane v posuvném registru do té doby, dokud není místo v přijímacím bufferu. Jestliže je detekován nový rámec době, kdy je přijímací buffer i posuvný registr plný, rámec v posuvném registru bude přepsán a nastaví se příznak RXOF značící, že došlo k přepsání dat.



Obrázek 15 Předávání přijatých dat, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Synchronní operace

Formát rámce

Rámce používané při synchronním přenosu nemusí mít start a stop bit, jelikož je použit nezávislý hodinový signál. Také bity parity nemohou být použity v synchronním módu. Délka rámce může být mezi 4 – 16 bity. Posílání větších rámců se řeší rozdělením těchto rámců na několik menších.

Master mód

V master módu má USART plnou kontrolu nad tokem dat a synchronní sběrnici. Ve full duplex módu nemůže slave vysílat data k master bez toho, aniž by master vysílal do slave jako první. Komunikace začíná tehdy, je-li vysílač zapnut a ve vysílacím bufferu se nachází nějaká data. Poté se spustí USART hodiny a master poté vysílá jednotlivé bity z vysílacího posuvného registru za použití vnitřních hodin. Jestliže se ve vysílacím bufferu nenachází žádné další rámce a vysílací posuvný registr je prázdný, hodiny se zastaví a komunikace končí.

Jestliže je zapnut přijímač, vzorkuje data za použití vnitřních hodin, pokud vysílač poslal nějaká data. Operace nad vysílacími a přijímacími buffery jsou stejné jako u asynchronního módu.

Slave mód

Jestliže je USART v módu slave, přenos dat není kontrolován USART, ale nějakým externím master. USART tudíž nemůže vyvolat přenos a nemá žádnou kontrolu nad počtem bytů zapsaných do master. Pro vysílání dat ve slave módu, slave musí nahrát data do vysílacího bufferu a povolit vysílač. Data zůstanou v USART do té doby, dokud master nezačne přenos vyzváním slave k vysílání dat. Za každý rámec, který master přenesl na slave je jiný rámec přenesen ze slave na master.

Low Energy Universal Asynchronous Receiver/Transmitter (LEUART)

LEUART zajišťuje obousměrnou UART komunikaci v situacích, kdy je vyžadována minimální spotřeba. LEUART může čekat na příchozí UART rámce i v módu EM2 při velmi nízké spotřebě. Poté, co je UART rámec přijat, procesor může být vzbuzen. Je ale také možné přenést několik rámců pomocí DMA modulu do paměti RAM a procesor vzbudit až následně. Přijatá data mohou být také blokována do doby, než je detekován nastavitelný start rámec. Lze také nakonfigurovat tzv. signální rámec pro generování přerušení (např. že se jedná o konec přenosu).

Vlastnosti

- Asynchronní sériová komunikace s nízkou spotřebou
- Full/half duplex komunikace
- Oddělitelné povolení přijímače a vysílače
- Oddělený vysílací a přijímací buffer
- Programovatelná baudová rychlost (300 – 9600 baud/s)
- Lze použít i vysokofrekvenční hodiny pro zvýšení baudové rychlosti
- Konfigurovatelný počet datových bitů – 8 nebo 9 (plus bit parity, jestli je povoleno)
- Konfigurovatelný počet stop bitů – 1 nebo 2
- Možnost probuzení ze sleep módu při přijetí rámce
 - Probuzení při přijetí libovolného bytu
 - Probuzení při přijetí specifikovaného start nebo signálního rámce
- Podpora vysílání i přijímání v módech EM0 – EM2 s
 - Plnou DMA podporou
 - Specifikovaný start-byte může zapnout příjem automaticky
- Multi-procesor mód

Inter-Integrate Circuit Intergace (I²C)

I²C modul poskytuje rozhraní mezi MCU a sériovou I²C sběrnici. Může se nastavit jako master nebo slave a zároveň je podporováno i více master. Jsou podporovány všechny 3 módy: standard-mode, fast-mode a fast-mode plus. Rychlost přenosu se pohybuje mezi 10 kbit/s – 1 Mbit/s.

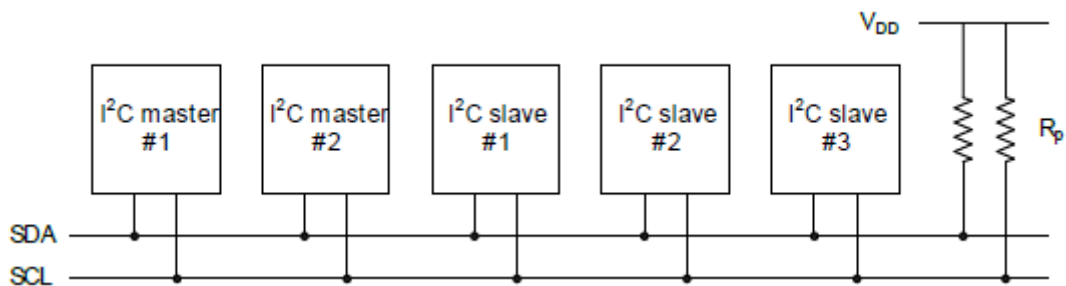
Vlastnosti:

- Podpora více master
- Podpora různých módů
 - Standard-mode (Sm) – rychlost až 100 kbit/s
 - Fast-mode (Fm) – rychlost až 400 kbit/s
 - Fast-mode plus (Fm+) – rychlost až 1 Mbit/s
- Hodinová synchronizace
- HW rozpoznání adres
 - 7 bitová adresa

- Aktivní ve všech módech vyjma EM4
- Podpora 10 bitových adres
- Dvojitý buffer pro data
- Plná podpora DMA

Propojení stanic

I²C používá pro komunikaci mezi stanicemi 2 sériové vodiče – sériový datový vodič (SDA) a sériový hodinový vodič (SCL) viz obrázek 16. Sběrnice také umožňuje detekci kolizí a řešení situací, kdy chce v jednu chvíli vysílat několik master data, aby nedošlo ke ztrátě dat.

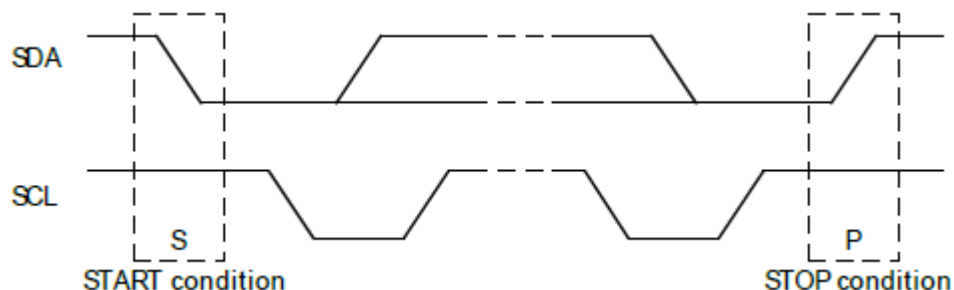


Obrázek 16 I²C sběrnice, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Každé zařízení na sběrnici lze adresovat pomocí unikátní adresy. Master může adresovat jakékoliv zařízení a to i ostatní master.

START a STOP podmínky

Start a stop podmínky se používají k zahájení a ukončení přenosu. Jakékoliv přenosy za sběrnici začínají se START podmínkou (S) a končí se STOP podmínkou (P). Jak je vidět na obrázku 17, START podmínka je generována přechodem SDA vodiče z vysoké úrovně do nízké a držením SCL vodiče ve vysoké úrovni. STOP podmínka je generována přechodem SDA vodiče z nízké úrovně do vysoké a držením SCL vodiče ve vysoké úrovni.



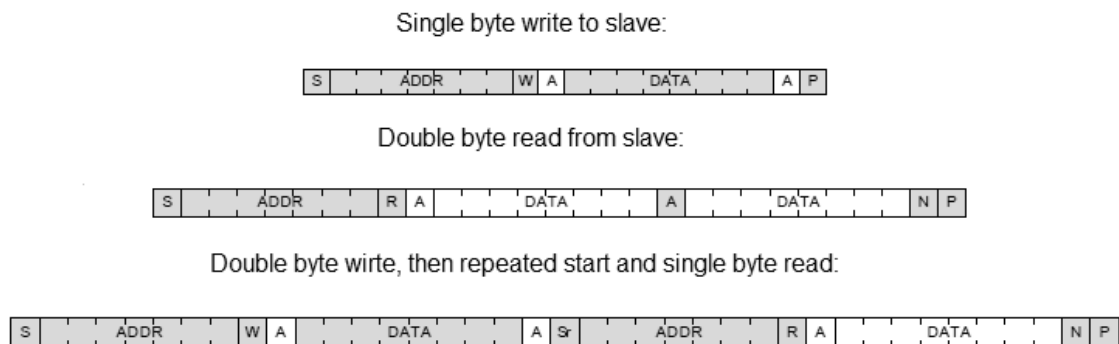
Obrázek 17 START a STOP podmínky, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Přenos dat

Jestliže chce master vyvolat přenos, čeká dokud se sběrnice nedostane do nečinného stavu. Poté vyšle START podmínku. Master dále vyšle adresu slave stanice, se kterou chce komunikovat a bit R/W podle toho, jestli chce ze slave číst data (R/W bit nastaven na 1) nebo do slave zapisovat data (R/W bit nastaven na 0). Poté co master pošle 7 bitovou adresu a R/W bit, uvolní sběrnici a čeká na potvrzení od slave. Slave nastaví SDA na nízkou úroveň (ACK) při kladném potvrzení nebo na vysokou úroveň (NACK) při záporném potvrzení.

Po kladném potvrzení začne master nebo slave vysílat data podle hodnoty v R/W bitu. Po každých 8 poslaných bitech po SDA vodiče uvolní vysílací stanice sběrnici a čeká na potvrzení dat od přijímače (ACK nebo NACK).

Master ukončí přenos vysláním STOP podmínky po (N)ACK. Po STOP podmínce může jakýkoliv master vyvolat přenos na sběrnici. Jestliže chce aktuální master vyvolat další přenos bez uvolnění sběrnice, může místo STOP podmínky a následné START podmínky vyslat opakovanou START podmínku (Sr). Příklady přenosů jsou vidět na obrázku 18.



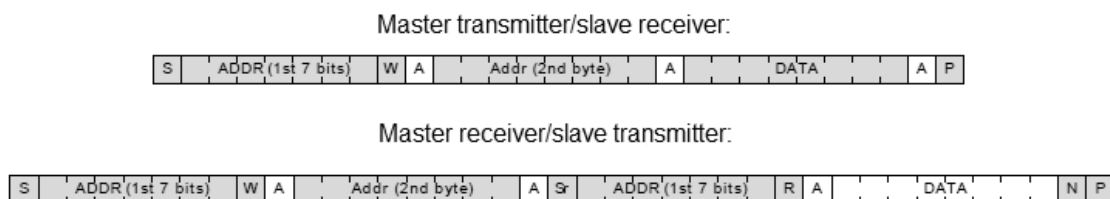
Obrázek 18 Příklady jednoduchá přenosů, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Adresování

I²C podporuje jak 7 bitové adresování, tak 10 bitové adresování. Při použití 7 bitové adresy, první byte vyslaný po START podmínce značí adresu slave stanice, kterou chce master kontaktovat.

Pokud chceme slave adresovat pomocí 10 bitové adresy, je nutné poslat 2 byty místo 1. Prvních 7 bitů musí být ve tvaru 1111 0XX, kde 11110 značí, že se jedná o 10 bitovou adresu a XX jsou první 2 bity z adresy. 8. bit stejně jako u 7 bitové adresy značí, zda chce master vysílat nebo přijímat data od slave. Druhý byte obsahuje zbylých 8 bitů z dané adresy.

Na obrázku 19 je vidět zápis a čtení ze slave při použití 10 bitových adres. V případě čtení ze slave je nutné nejdříve vytvořit spojení pro zápis, poslat slave 2 byty obsahující adresu, vyvolat nový start pro čtení a až následně data číst.



Obrázek 19 Použití 10 bitové adresy, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

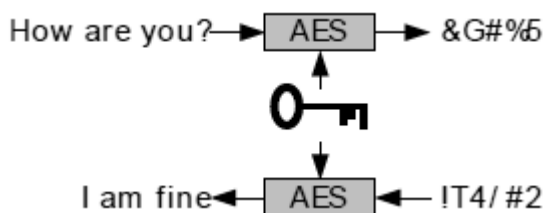
Arbitrace

Jestliže je arbitrace zapnutá, jakýkoliv master, který chce vyvolat přenos, musí zároveň číst, jaká data se na sběrnici aktuálně nacházejí. Jestliže je hodnota na sběrnici různá od hodnoty, kterou se master snaží vyslat, znamená to že na sběrnici je více přenosů a tento master se musí odpojit.

3.2.9 Bezpečnost

Advanced Encryption Standard Accelerator (AES)

AES je symetrická bloková šifra operující nad 128 bitovými bloky dat a používá 128, 192 nebo 256 bitové klíče. AES Accelerator provádí kódování a dekódování pomocí 128 nebo 256 bitových klíčů. Kódování nebo dekódování jednoho 128 bitového bloku dat trvá 54 cyklů HFCORECLK hodin při použití 128 bitového klíče a 75 cyklů HFCORECLK hodin při použití 256 bitového klíče. Na obrázku 20 je vidět jednoduché schéma symetrické šifry.



Obrázek 20 AES šifra, Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>

Vlastnosti:

- AES hardwarové kódování dekódování
 - 128 bitový klíč (54 cyklů HFCORECLK)
 - 256 bitový klíč (75 cyklů HFCORECLK)
- Efektivní podpora procesoru a DMA řadiče
- Generování přerušeni po skončení kódování/dekódování
- DMA požadavek po skončení kódování/dekódování

- Konfigurovatelné pořadí bytů

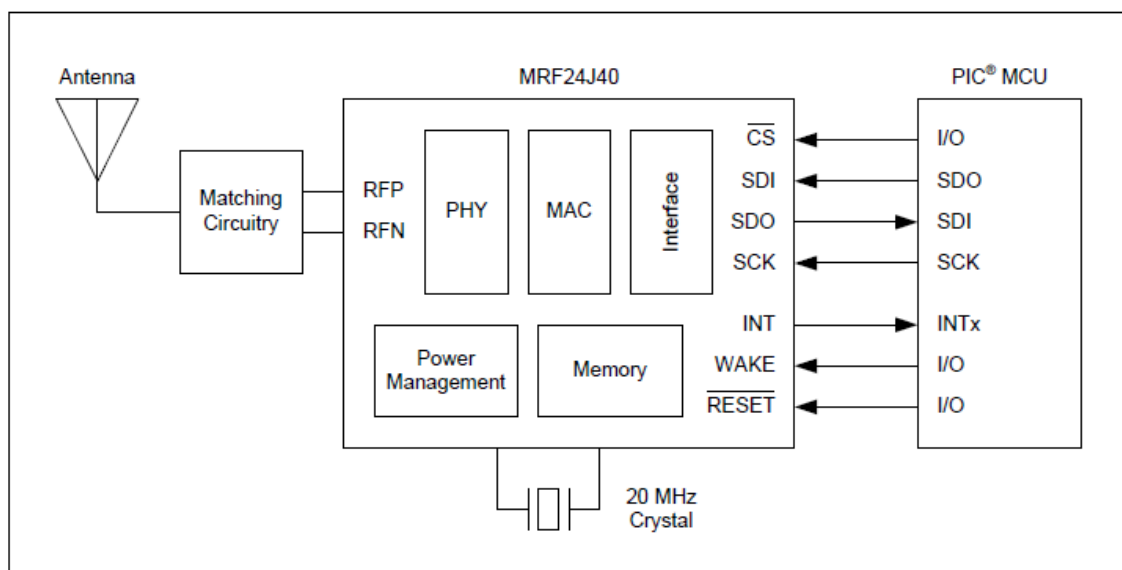
4 Popis bezdrátového modulu

4.1 Základní popis

MRF24J40 je vysílač/přijímač operující v pásmu 2,4 GHz a podporující standard IEEE 802.15.4. Integruje fyzickou a síťovou vrstvu v jednom zařízení. Na obrázku 21 je vidět zjednodušené schéma modulu. Hlavní výhody zařízení je nízká cena, nízká spotřeba a možnost nastavení rychlost přenosu (250 nebo 625 kbps). K propojení s MCU se používá 4 vodičové sériové rozhraní SPI.

Pro generování 2,4 GHz signálu se používá tzv. frekvenční syntetizér, který využívá 20 MHz krystalového oscilátoru. Zařízení obsahuje 6 GPIO (General Purpose Input Output) pinů, které mohou být nakonfigurovány pro řídicí nebo monitorovací účely. Spotřeba v režimu spánku se pohybuje okolo 2 μ A. Pro časování v režimu spánku lze použít interní 100 kHz oscilátor nebo externí 32 kHz krystalový oscilátor.

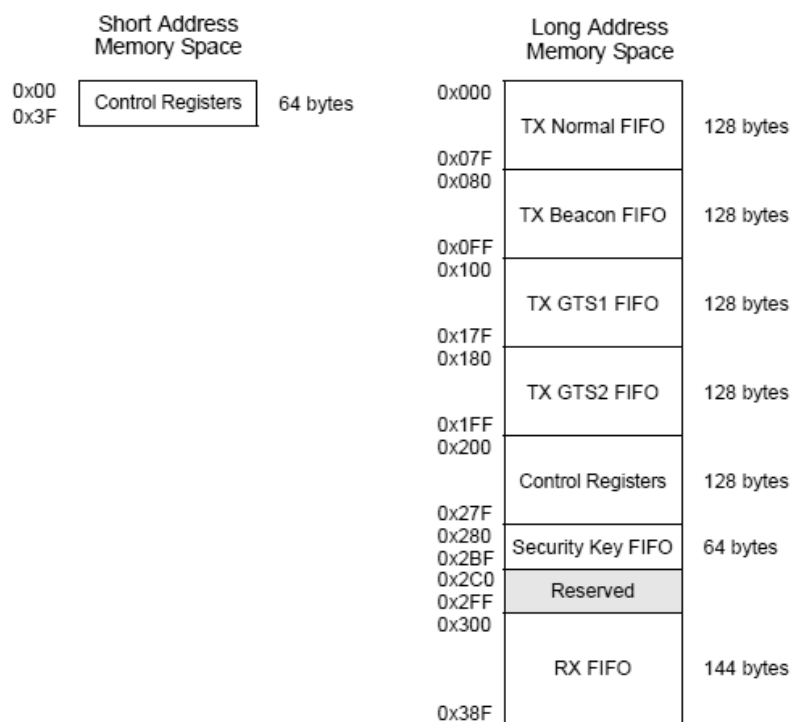
Obvod Media Access Controller (MAC) ověřuje příjem a formátuje packety pro vysílání, aby odpovídaly standardu IEEE 802.15.4. Data jsou ukládány do vysílacích a přijímacích bufferů. Zabezpečení lze provádět pomocí integrovaného hardwarového AES-128.



Obrázek 21 Zjednodušené schéma, MRF24J40 Microchip. MRF24J40 Data Sheet. [online] [cit. 2014-04-10] <<http://ww1.microchip.com/downloads/en/DeviceDoc/39776C.pdf>>

4.2 Paměť zařízení

Paměť v zařízení MRF24J40 je organizována jako statická RAM a je přístupná pomocí SPI rozhraní. Paměť je rozdělena na kontrolní registry a datové buffery (viz obrázek 22). Kontrolní registry slouží k řízení zjišťování stavu a adresování. Buffery slouží jako dočasné úložiště pro vysílaná nebo přijímaná data. K paměti lze přistupovat pomocí krátkých nebo dlouhých adres. Podrobný popis registrů lze najít v [7] kapitola 2.15.



Obrázek 22 Paměť MRF24J40, MRF24J40 Microchip. MRF24J40 Data Sheet. [online] [cit. 2014-04-10] <<http://ww1.microchip.com/downloads/en/DeviceDoc/39776C.pdf>>

4.3 Funkční popis

4.3.1 Reset

MRF24J40 obsahuje několik typů resetu:

- Power-on Reset – obvod, který automaticky vyresetuje zařízení, jestliže je přivedeno napájecí napětí. Po resetu je nutné počkat minimálně 2 ms než se operace resetu dokončí.
- RESET Pin – jestliže se přivede na RESET pin nízký signál, zařízení se vyresetuje.
- Softwarový Reset – Softwarový reset lze vyvolat přímo v programu nastavením registru pro reset.

4.3.2 Inicializace

Určité kontrolní registry musí být inicializovány pro základní operace. Tyto hodnoty se liší od Power-on Reset hodnot a poskytují vylepšené provozní parametry. Tyto nastavení se normálně provedou po provedení resetu. Po inicializaci mohou být potřebné vlastnosti nastaveny dle potřeb aplikace.

4.3.3 Přerušení

MRF24J40 obsahuje jeden přerušovací pin (INT). Lze signalizovat 8 různých přerušení připojenému MCU. Přerušení lze povolit nebo zakázat zapsáním do INTCON

registru. INTSTAT poté obsahuje hodnotu přerušení, které nastalo a který si může MCU přečíst.

4.3.4 Výběr kanálu

Zařízení MRF24J40 umožňuje vybrat jeden ze 16 dostupných kanálů z frekvenčního pásma 2,4 GHz. Požadovaný kanál lze vybrat zapsáním příslušné hodnoty do registru RFCON0. V tabulce 7 je vidět seznam kanálů s příslušnými frekvencemi a hodnotami.

Tabulka 7 Výběr kanálu pro MRF24J40

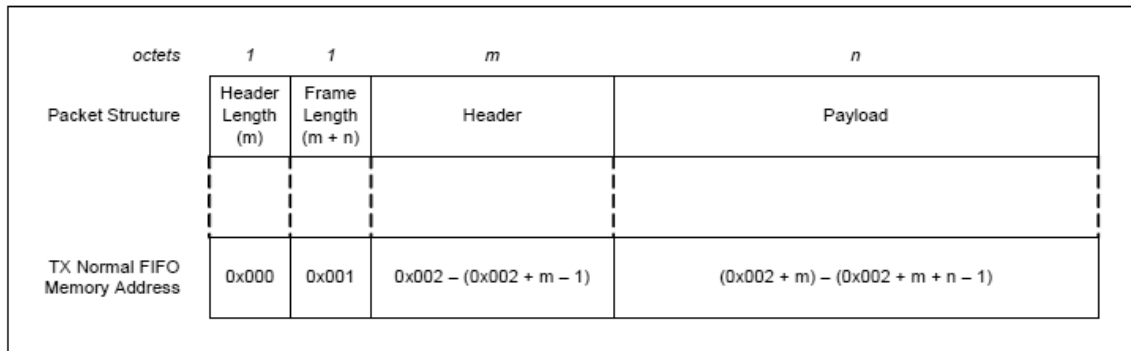
Channel Number	Frequency	Set Value
11	2.405 GHz	0x03
12	2.410 GHz	0x13
13	2.415 GHz	0x23
14	2.420 GHz	0x33
15	2.425 GHz	0x43
16	2.430 GHz	0x53
17	2.435 GHz	0x63
18	2.440 GHz	0x73
19	2.445 GHz	0x83
20	2.450 GHz	0x93
21	2.455 GHz	0xA3
22	2.460 GHz	0xB3
23	2.465 GHz	0xC3
24	2.470 GHz	0xD3
25	2.475 GHz	0xE3
26	2.480 GHz	0xF3

4.3.5 Vysílání dat

Standard IEEE 802.15.4 definuje 4 typy rámců: Acknowledgment, Data, Beacon a MAC Command frame. Pro naše účely je nejdůležitější datový rámec. Pro potřeby datového rámce se používá fronta TX Normal FIFO s adresami od 0x000 do 0x07F.

Formát dat v rámci je vidět na obrázku 23. Obsahuje délku hlavičky, délku rámce, hlavičku a samotná data.

Samotný přenos poté probíhá tak, že se nejdříve tato fronta naplní požadovanými daty. Nastaví se, zda je požadováno potvrzení a zabezpečení přenášeného rámce. Poté se bit TXNTRIG nastaví na '1' a obsah TX Normal FIFO je bezdrátově odeslán na aktuálně nastaveném kanálu.



Obrázek 23 Formát rámce TX Normal FIFO, MRF24J40 Microchip. MRF24J40 Data Sheet. [online] [cit. 2014-04-10] <<http://ww1.microchip.com/downloads/en/DeviceDoc/39776C.pdf>>

4.3.6 Příjem dat

Každý přijatý rámec obsahuje prefix SHR, díky čemuž je možné provádět synchronizaci rámců. MRF24J40 monitoruje příchozí signály a hledá příslušný prefix označující rámec IEEE 802.15.4. Jestliže je tento rámec nalezen, celý paket je demodulován a je spočítáno a ověřeno CRC. Paket je buď přijat nebo odmítnut v závislosti na nastaveném filtru a je umístěn do RXFIFO bufferu. Signalizovaným přerušením se dává na vědomí, že byl přijat rámec.

5 Popis seznorů

5.1 Akcelerometr Freescale MMA7660FC

MMA7660FC je 3-osý akcelerometr se sériovým výstupním rozhraním (I²C) a maximálním zaznamenaným přetížením ±1,5 g. Zařízení má velmi malé rozměry i odběr energie. Naměřené hodnoty převádí do 6 bitové digitální hodnoty. Uživatel si může nastavit rychlost snímkování za sekundu [10].

Vlastnosti:

- Digitální sériové rozhraní (I²C)
- Rozměry: 3 x 3 x 3,9 mm
- Několik pracovních módů:
 - Off mód: 0,4 μA
 - Standby mód: 2 μA
 - Active mód:
 - 47 μA při 1 snímku za sekundu
 - 294 μA při 120 snímcích za sekundu
- Nastavitelné vzorkování od 1 do 120 vzorků za sekundu
- Nízké napětí
 - Analogové napětí: 2,4 V – 3,6 V
 - Digitální napětí: 1,71 V – 3,6 V
- Automatické probuzení/uspání
- Zjištění náklonu
- Nízká cena

Operační módy

Off mód

- Se zařízením nelze komunikovat přes I²C
- I²C lze použít ke komunikaci s jinými zařízeními

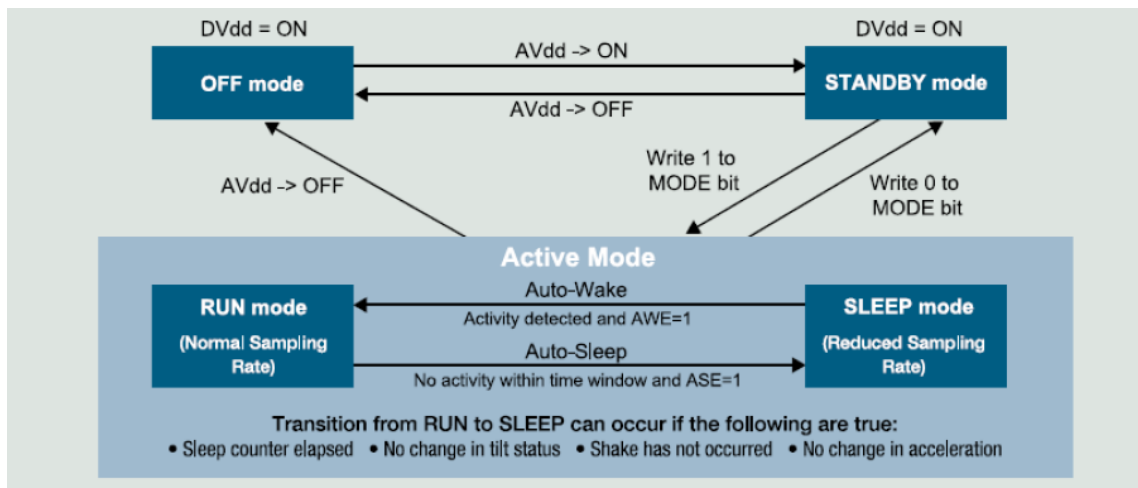
Standby mód

- Se zařízením lze komunikovat přes I²C
- K registrům lze libovolně přistupovat
- Měření ze senzoru je neaktivní

Active mód

- Měření ze senzoru je aktivní v závislosti na uživatelem nastaveném vzorkování
- Možnost automatického uspání/probuzení

Obrázek 24 ukazuje přepínání mezi jednotlivými módy.



Obrázek 24 Přechod mezi módy MMA7660FC, Freescale. 3-Axis Orientation/Motion Detection Sensor. [online] [cit. 2014-02-17] <http://www.freescale.com.cn/files/sensors/doc/data_sheet/MMA7660FC.pdf?fpsp=1>

5.2 Magnetometr Freescale MAG3110

MAG3110 je 3-osý magnetometr. Lze ho použít k měření magnetických polí. Lze také použít společně s 3-osým akcelerometrem pro vytvoření elektrického kompasu, který je nezávislý na orientaci přístroje. Používá standardní sériové rozhraní I²C [11].

Vlastnosti:

- Digitální sériové rozhraní (I²C)
- Rozměry: 2 x 2 x 0,85 mm
- Několik pracovních módů:
 - Standby mód: 2 μ A
 - Active mód:
 - 8,6 μ A při 0,63 snímcích za vteřinu
 - 900 μ A při 80 snímcích za vteřinu
- Nastavitelné vzorkování od 0,63 do 80 vzorků za sekundu
- Nízké napětí
 - Analogové napětí: 1,95V – 3,6 V
 - Digitální napětí: 1,62 V – 3,6 V

Operační módy

Standby mód

- Se zařízením lze komunikovat přes I²C
- K registrům lze libovolně přistupovat
- Měření ze senzoru je neaktivní

Active mód

- Měření ze senzoru je aktivní v závislosti na uživatelem nastaveném vzorkování
- Možnost automatického uspání/probuzení

5.3 Teploměr a vlhkoměr Sensirion SHT21

SHT21 slouží k měření teploty a vlhkosti. Pro komunikaci se používá sériové rozhraní I²C [12].

Vlastnosti:

- Digitální sériové rozhraní (I²C)
- Rozměry: 3 x 3 x 1,1 mm
- Několik pracovních módů:
 - Sleep mód: 0,5 μA
 - Measuring mód: 300 μA
- Napájecí napětí: 2,1V – 3,6 V

Převod hodnot

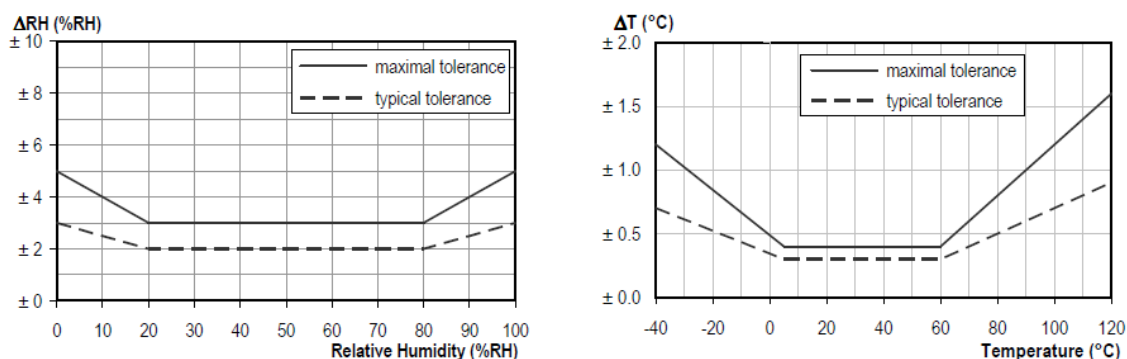
Výsledné naměřené hodnoty je nutné převést pomocí následujících vzorců. S_T značí naměřenou teplotu a S_{RH} naměřenou vlhkost:

$$RH = -6 + 125 * \frac{S_{RH}}{2^{16}}$$

$$T = -46,85 + 175,72 * \frac{S_T}{2^{16}}$$

Odchylka

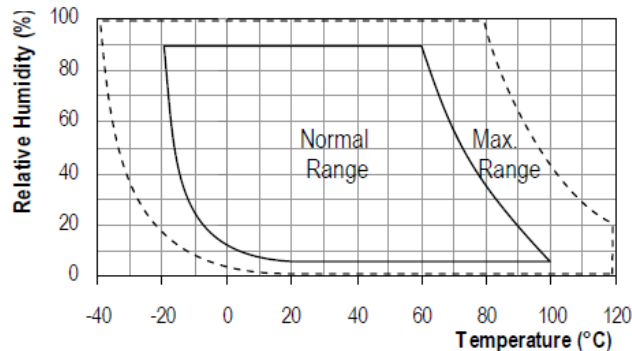
Jak je vidět na obrázku 25, odchylka je v normálních podmínkách minimální od skutečných hodnot. U krajních hodnot se ale odchylka zvyšuje.



Obrázek 25 Možná odchylka naměřených hodnot teploty a vlhkosti od skutečných u senzoru SHT21, Sensirion. Datasheet SHT21. [online] [cit. 2014-02-17]
<http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT21_Datasheet_V3.pdf>

Provozní rozsah

Obrázek 26 ukazuje provozní rozsah senzoru. Jestliže je senzor dlouhou dobu vystaven hodnotám mimo normální rozsah (obzvláště vystavení vlhkosti přes 80%), může se na nějaký čas zvýšit odchylka senzoru. Po vrácení do normálního rozsahu se senzor postupně vrátí do svého původního stavu.



Obrázek 26 Provozní rozsah senzoru SHT21, Sensirion. Datasheet SHT21. [online] [cit. 2014-02-17] <http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT21_Datasheet_V3.pdf>

5.4 Tlakoměr BOSCH BMP180

Senzor BMP180 slouží k měření atmosférického tlaku. Pro komunikaci se používá sériové rozhraní I²C [14].

Vlastnosti:

- Digitální sériové rozhraní (I²C)
- Rozměry: 3,6 x 3,8 x 0,93 mm
- Několik pracovních módů:
 - Standby mód: 0,1 μA
 - Active mód:
 - 3 μA při Ultra low power módu
 - 32 μA při Advanced resolution módu
 - Převod hodnot: 650 μA
- Nastavitelná přesnost měření
- Nízké napětí
 - Analogové napětí: 1,8 V – 3,6 V
 - Digitální napětí: 1,62 V – 3,6 V

Spočítání nadmořské výšky

Z naměřeného tlaku p a hodnoty tlaku na úrovni moře $p_0 = 1013,25$ hPa lze spočítat nadmořskou výšku v v metrech z následujícího vzorce:

$$\text{nadmořská výška} = 44330 * \left(1 - \left(\frac{p}{p_0} \right)^{\frac{1}{5,255}} \right)$$

Z toho nám vyjde, že při změně tlaku o 1 hPa se nadmořská výška změní o 8,43m.

Spočítání tlaku na úrovni moře

Z naměřeného tlaku p a nadmořské výšky lze spočítat tlak na úrovni moře z následujícího vzorce:

$$p_0 = \frac{p}{\left(1 - \frac{\text{nadmořská výška}}{44330}\right)^{5,255}}$$

Z toho nám vyjde, že při změně nadmořské výšky o 10m se tlak na úrovni moře změní o 1,2 hPa.

6 Spotřeba

Spotřeba je nejdůležitější faktor tohoto systému, jelikož poběží ve většině případů na baterii a je požadována co nejdelší výdrž. Každá komponenta systému umožňuje snížení spotřeby na minimum, jestliže není zrovna využívána.

6.1 Spotřeba MCU

MCU umožňuje běžet v několika úsporných módech. V každém módu je dáno, které části jsou napájeny a které ne. Podrobnější informace jsou v kapitole 3.2.4. Tabulka 8 ukazuje srovnání módů a spotřeby.

Tabulka 8 Porovnání spotřeby MCU EFM32TG

Mód spotřeby	Spotřeba
EM0	157 $\mu\text{A}/\text{MHz} \Rightarrow 5 \text{ mA}$ při 32 MHz
EM1	53 $\mu\text{A}/\text{MHz} \Rightarrow 1,7 \text{ mA}$ při 32 MHz
EM2	1 μA
EM3	0,59 μA
EM4	0,02 μA

Spotřeba v módech EM2 – EM4 roste při vysoké teplotě. Např. spotřeba v módu EM2 je při 25 stupních 1 μA , jak je uvedeno v tabulce. Ovšem při 85 stupních Celsia je tato spotřeba již 2,5 μA .

6.2 Spotřeba bezdrátového modulu

Jak je vidět v tabulce 5 v kapitole 2.3.2, bezdrátový modul nabízí 3 pracovní režimy – režim vysílání, přijímání a spánku. Při vysílání nebo přijímání (jak je vidět z tabulky 9) má bezdrátový modul několikanásobně vyšší spotřebu než MCU.

Tabulka 9 Srovnání spotřeby bezdrátového modulu

Mód spotřeby	Spotřeba
Vysílač	23 mA
Přijímač	19 mA
Mód spánku	2 μA

6.3 Spotřeba senzorů

6.3.1 Akcelerometr

Spotřeba u akcelerometru se odvíjí od toho, jak moc časté vzorkování se požaduje. Tabulka 10 ukazuje srovnání spotřeby při různém vzorkování. Jak je vidět, je nutné se rozhodnout, jaké vzorkování je potřeba, protože spotřeba se zvýší několikanásobně.

Tabulka 10 Srovnání spotřeby akcelerometru

Mód spotřeby		Spotřeba
Off mód		0,4 μA
Power down mód		2 μA
Standby mód		2 μA
Active mód	1 vzorek / sekundu	47 μA
	2 vzorky / sekundu	49 μA
	4 vzorky / sekundu	54 μA
	8 vzorků / sekundu	66 μA
	16 vzorků / sekundu	89 μA
	32 vzorků / sekundu	133 μA
	64 vzorků / sekundu	221 μA
	120 vzorků / sekundu	294 μA

6.3.2 Magnetometr

Spotřeba u magnetometru se vyvíjí podobně jako u akcelerometru – je možné nastavit počet vzorků za vteřinu, z čehož se následně vyvíjí spotřeba. Tabulka 11 ukazuje srovnání spotřeby.

Tabulka 11 Srovnání spotřeby magnetometru

Mód spotřeby		Spotřeba
Standby mód		2 μA
Active mód	0,63 vzorku / sekundu	8,6 μA
	1,25 vzorků / sekundu	17,2 μA
	2,5 vzorků / sekundu	34,4 μA
	5 vzorků / sekundu	68,8 μA
	10 vzorků / sekundu	137,5 μA
	20 vzorků / sekundu	275 μA
	40 vzorků / sekundu	550 μA
	80 vzorků / sekundu	900 μA

6.3.3 Teploměr a vlhkoměr

Teploměr a vlhkoměr obsahuje pouze 2 módy – sleep a měření. Přesnost zde nezáleží na spotřebě (ta je při měření vždy stejná), ale na okolních teplotách a vlhkosti (jak je uvedeno v kapitole 5.3). Tabulka 12 ukazuje rozdíl spotřeby v různých módech.

Tabulka 12 Srovnání spotřeby teploměru a vlhkoměru

Mód spotřeby	Spotřeba
Sleep mód	0,15 μA
Měření	300 μA

6.3.4 Tlakoměr

Tlakoměr obsahuje několik módů. Čím se požaduje větší přesnost měření, tím vyšší spotřeba. Tabulka 13 ukazuje srovnání těchto módů včetně možné odchylky od reálné hodnoty.

Tabulka 13 Srovnání spotřeby Tlakoměru

Mód spotřeby	Doba převodu [ms]	Možná odchylka hodnoty tlaku [hPa]	Možná odchylka hodnoty výšky [m]	Spotřeba
Standby mód	-	-	-	0,1 μ A
Ultra low power	4,5	0,06	0,5	3 μ A
Standard	7,5	0,05	0,4	5 μ A
High resolution	13,5	0,04	0,3	7 μ A
Ultra high resolution	25,5	0,03	0,25	12 μ A
Advanced resolution	76,5	0,02	0,17	32 μ A
Konverze hodnot	-	-	-	650 μ A

6.4 Teoretický odhad spotřeby na základě navrženého programového vybavení

Spotřeba systému se v průběhu měření mění. Níže jsou uvedeny spotřeby a doby v jednotlivých fázích měření při výchozím nastavení:

- MCU běží, bezdrátový modul neaktivní, probíhá měření ze senzoru teploty, ostatní senzory vypnuty

Jednotlivé spotřeby:

- MCU – 157 μ A (při frekvenci 1 MHz)
- Bezdrátový modul – 2 μ A (režim spánku)
- Teplotní senzor – 300 μ A (mód měření)
- Akcelerometr – 2 μ A (Standby mód)
- Magnetometr – 2 μ A (Standby mód)
- Tlakoměr – 0,1 μ A (Standby mód)

Výsledná spotřeba:

$$157 + 2 + 300 + 2 + 2 + 0,1 = 463,1 \mu\text{A}$$

Jednotlivé časy:

- Probuzení z EM2 – 2 μ s
- Maximální frekvence přenosu přes I²C = 91 kHz => 10,9 μ s pro 1 přenos, potřeba 1 přenos (požadavek na zahájení měření získání dat)
- Čas měření – 66 ms pro teplotu a 22 ms pro vlhkost

Celkový čas běhu:

$$2 + 2 * (10,9) + 66000 + 22000 = 88044,7 \mu\text{s} = 88,04 \text{ ms}$$

- MCU běží, bezdrátový modul neaktivní, probíhá měření ze senzoru tlaku, ostatní senzory vypnuty

Jednotlivé spotřeby:

- MCU – 157 μA (při frekvenci 1 MHz)
- Bezdrátový modul – 2 μA (režim spánku)
- Teplotní senzor – 0,15 μA (sleep mód)
- Akcelerometr – 2 μA (Standby mód)
- Magnetometr – 2 μA (Standby mód)
- Tlakoměr – 5 μA (Měření Standard mód), 650 μA (převod hodnot)

Výsledné spotřeby:

$$157 + 2 + 5 + 650 + 2 + 2 + 0,1 = 818,1 \mu\text{A}$$

Jednotlivé časy:

- Probuzení z EM2 – 2 μs
- Maximální frekvence přenosu přes I²C = 91 kHz => 10,9 μs pro 1 přenos, potřeba 4 přenosů (požadavek na měření a získání dat ze 3 registrů) => 43,6 μs
- Čas měření – 7,5 ms pro tlak a 4,5 ms pro teplotu

Celkový čas běhu:

$$2 + 2 * (43,6) + 7500 + 4500 = 12089,2 \mu\text{s} = 12,01 \text{ ms}$$

- MCU běží, bezdrátový modul neaktivní, probíhá měření z akcelerometru, ostatní senzory vypnuty

Jednotlivé spotřeby:

- MCU – 157 μA (při frekvenci 1 MHz)
- Bezdrátový modul – 2 μA (režim spánku)
- Teplotní senzor – 0,15 μA (Sleep mód)
- Akcelerometr – 66 μA (8 vzorků / sekundu)
- Magnetometr – 2 μA (Standby mód)
- Tlakoměr – 0,1 μA (Standby mód)

Výsledné spotřeby:

$$157 + 2 + 0,15 + 66 + 2 + 0,1 = 227,25 \mu\text{A}$$

Jednotlivé časy:

- Probuzení z EM2 – 2 μs

- Maximální frekvence přenosu přes I²C = 91 kHz => 10,9 μs pro 1 přenos, potřeba 5 přenosů (požadavek na měření, získání dat ze 3 registrů a přepnutí do Standby módu) => 54,5 μs
- Čas měření – 125 ms
- Čas přechodu do aktivního módu – 12 ms
- Čas přechodu do Standby módu – 1 ms

Celkový čas běhu:

$$2 + 54,5 + 125000 + 12000 + 1000 = 138056,5 \mu s = 138,01 \text{ ms}$$

- MCU běží, bezdrátový modul neaktivní, probíhá měření z magnetického senzoru, ostatní senzory vypnuty

Jednotlivé spotřeby:

- MCU – 157 μA (při frekvenci 1 MHz)
- Bezdrátový modul – 2 μA (režim spánku)
- Teplotní senzor – 0,15 μA (Sleep mód)
- Akcelerometr – 2 μA (Standby mód)
- Magnetometr – 137,5 μA (10 vzorků / sekundu)
- Tlakoměr – 0,1 μA (Standby mód)

Výsledné spotřeby:

$$157 + 2 + 0,15 + 2 + 137,5 + 0,1 = 298,75 \mu A$$

Jednotlivé časy:

- Probuzení z EM2 – 2 μs
- Maximální frekvence přenosu přes I²C = 91 kHz => 10,9 μs pro 1 přenos, potřeba 8 přenosů (požadavek na měření, získání dat ze 6 registrů a přepnutí do Standby módu) => 87,2 μs
- Čas měření – 100 ms
- Čas přechodu do aktivního módu – 12 ms
- Čas přechodu do Standby módu – 1 ms

Celkový čas běhu:

$$2 + 87,2 + 100000 + 12000 + 1000 = 113089,2 \mu s = 113,01 \text{ ms}$$

- MCU běží, bezdrátový modul vysílá, senzory neaktivní

Jednotlivé spotřeby:

- MCU – 157 μA (při frekvenci 1 MHz)
- Bezdrátový modul – 23000 μA (mód vysílač)
- Teplotní senzor – 0,15 μA (Sleep mód)
- Akcelerometr – 2 μA (Standby mód)

- Magnetometr – 2 μA (Standby mód)
- Tlakoměr – 0,1 μA (Standby mód)

Výsledné spotřeby:

$$157 + 23000 + 0,15 + 2 + 2 + 0,1 = 23161,25 \mu\text{A}$$

Jednotlivé časy:

- Probuzení z EM2 – 2 μs
- Doba přenosu přes SPI při rychlosti 1Mbaud/s = $128 / 1\text{M} = 128 \mu\text{s}$
- Doba poslání paketu – není uvedeno přímo v datasheetu, průměrná hodnota na základě měření uživatelů v [14] je 5 ms

Celkový čas běhu:

$$2 + 128 + 5000 = 5130 \mu\text{s} = 5,13 \text{ ms}$$

- MCU uspán, bezdrátový modul neaktivní, senzory neaktivní

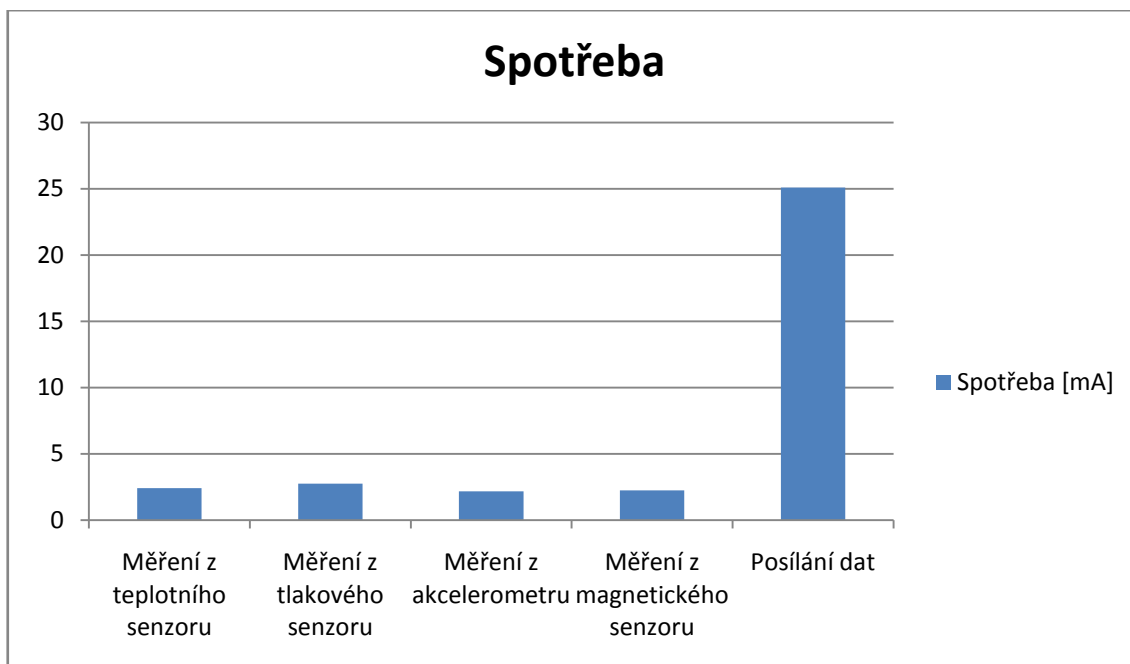
Jednotlivé spotřeby:

- MCU – 1 μA (mód EM2)
- Bezdrátový modul – 2 μA (režim spánku)
- Teplotní senzor – 0,15 μA (Sleep mód)
- Akcelerometr – 2 μA (Standby mód)
- Magnetometr – 2 μA (Standby mód)
- Tlakoměr – 0,1 μA (Standby mód)

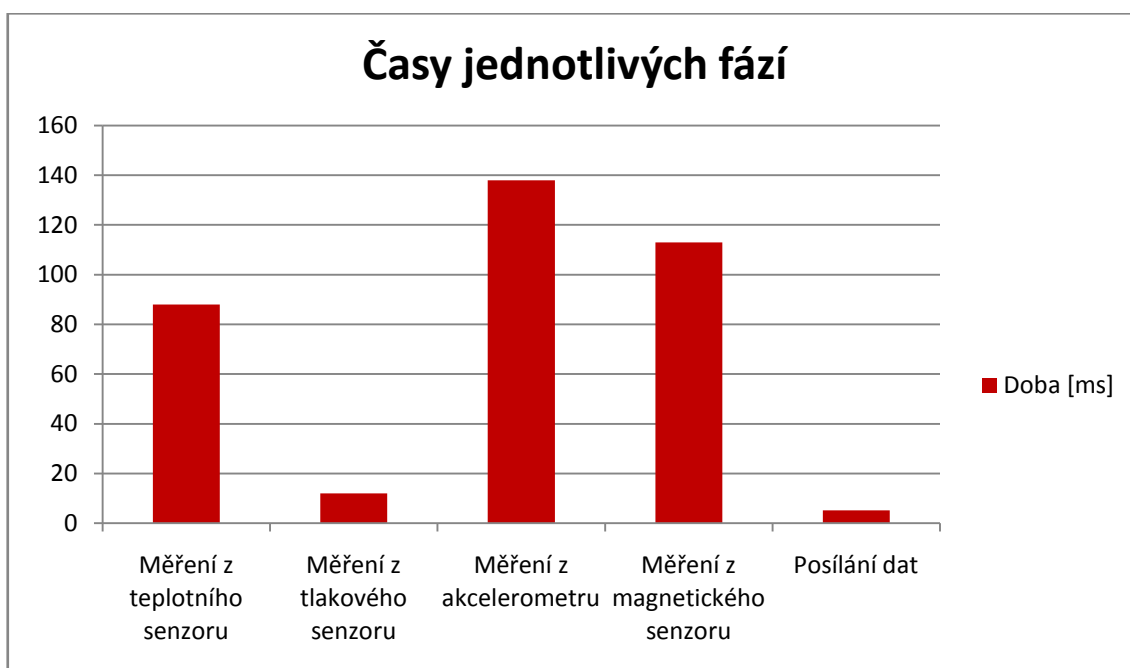
Výsledné spotřeby:

$$1 + 2 + 0,15 + 2 + 2 + 0,1 = 7,25 \mu\text{A}$$

V grafu 1 jsou uvedeny špičky měření v jednotlivých fázích. V grafu 2 jsou uvedeny časy.



Graf 1 Spotřeba v jednotlivých fázích



Graf 2 Časy jednotlivých fází

Celková spotřeba:

Je nutné spočítat, jaká bude spotřeba celého systému 1 hodinu. Z toho lze potom odhadnout, jaká bude výdrž systému na 1 nabití baterie. Vezme se spotřeba při měření, vynásobí se časem (tím dostaneme odběr proudu za vteřinu) a následně se vynásobí 6, jelikož při standardním nastavením se provede 6 měření za hodinu.

Měření z teplotního senzoru:

$$0,4631 * 0,08804 * 6 = 0,245 \text{ mA/h}$$

Měření z tlakového senzoru:

$$0,8181 * 0,01201 * 6 = 0,056 \text{ mA/h}$$

Měření z akcelerometru:

$$0,22725 * 0,13801 * 6 = 0,188 \text{ mA/h}$$

Měření z magnetického senzoru:

$$0,29875 * 0,11301 * 6 = 0,203 \text{ mA/h}$$

Poslání dat:

$$23,16125 * 0,00513 * 6 = 0,713 \text{ mA/h}$$

Neaktivní režim:

$$0,00725 * 6 = 0,044 \text{ mA/h}$$

Celkový odběr proudu za hodinu:

$$0,245 + 0,056 + 0,188 + 0,203 + 0,713 + 0,044 = 1,45 \text{ mAh}$$

Doba běhu na baterii:

Jestli bychom měli např. klasickou baterii LiIon s kapacitou 2000 mAh, mohl by tento modul na ní běžet zhruba 58 dní. Pro zvýšení výdrže by bylo možné odpojit nějaké senzory nebo zvýšit prodlevu mezi měřeními. V případě zvýšení prodlevy na 1 hodinu bychom dostali zhruba rok.

7 Programová realizace

Program byl naprogramován v jazyce C za využití vývojového prostředí IAR Embedded Workshop for ARM a byl testován na platformě Windows.

7.1 Účel programu

Tento program slouží k opakovanému měření fyzikálních parametrů prostředí (teplota, vlhkost, tlak, magnetické pole, zrychlení). Program tyto informace zpracuje, vytvoří paket, který pošle bezdrátovému modulu a ten tento paket bezdrátově odešle na zvoleném kanálu.

Při psaní programu jsem vycházel z ukázkových příkladů pro I2C a SPI a upravil a doplnil je, aby odpovídaly daným pinům a senzorům.

7.2 Popis programu

Program se skládá celkem ze 16 souborů – 8 zdrojových a 8 hlavičkových.

7.2.1 i2c.c

Obsahuje funkci pro inicializaci I²C sběrnice a obsluhu přerušení. Zařízení je inicializované jako master.

Funkce:

I2C0_IRQHandler(void) – Funkce volá funkci *I2C_Transfer*, která kontroluje nastavené flagy a vrací příslušný stav sběrnice

I2C_Sensor_Init(void) – Funkce pro inicializaci I²C sběrnice. Nejdříve se nastaví rychlost sběrnice a spustí se příslušné hodiny. Poté se nastaví piny SDA (pin 5) a SCL (pin 6). V některých případech se může stát (např. při nějakém výpadku), že zařízení slave zůstane v neznámém stavu, proto se poté pošle 9 hodinových signálů. Poté se povolí SDA a SCL piny a volá se knihovní funkce pro inicializaci I²C. Nakonec se povolí přerušení od I²C modulu.

7.2.2 i2c.h

Hlavičkový soubor pro i2c.c. Obsahuje definici funkce *I2C_Sensor_Init(void)*.

7.2.3 i2c_temp_sensor.c

Obsahuje funkce pro čtení a zapsání hodnot do registrů teplotního senzoru a funkce pro zjištění teploty a vlhkosti.

Funkce:

TEMPSENS_RegisterGet() – přečtení hodnoty z daného registru. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení, adresa registru, ze kterého se má číst a ukazatel na proměnou, do které se má zapsat výsledek. Samotný kód probíhá tak, že se nejdříve sestaví požadavek senzoru (adresa + bit R (read) + adresa registru), následně se

odešle a čeká se na přerušení, které nastane až když byl přenos dokončen. Během čekání se MCU přepne do módu EM1. Nakonec se přečte přijatá hodnota a uloží se do dané struktury. Jestliže vše proběhlo v pořádku, vrátí se 0. Jestliže ne, vrátí se hodnota stavu linky I²C, podle které lze zjistit, k jakému problému došlo.

TEMPSENS_RegisterSet() – zápis hodnoty do daného registru. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení, adresa registru, do kterého se má zapisovat a hodnota, která se má do daného registru zapsat. Samotný kód probíhá tak, že se nejdříve sestaví požadavek senzoru (adresa + bit W (write) + adresa registru + hodnota), následně se odešle a čeká se na přerušení, které nastane až když byl přenos dokončen. Během čekání se MCU přepne do módu EM1. Nakonec se vrátí se hodnota stavu linky I²C.

TEMPSENS_TemperatureGet() – zjištění aktuální teploty. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení a ukazatel na strukturu, do které se má zapsat výsledek. Samotný kód probíhá tak, že se volá funkce *TEMPSENS_RegisterGet()* s parametry:

- *i2c* – ukazatel na I²C strukturu
- *addr* – adresa teplotního senzoru
- *triggerTmeasHold* – jméno registru. Čtením z tohoto registru se spustí měření a poté je vrácena výsledná hodnota. Master je po celou dobu měření blokován a čeká se, dokud není měření dokončeno.
- *val* – proměnná, do které se zapíše výsledek

Jestliže vše proběhlo v pořádku, funkce končí. V opačném případě se pomocí vzorce v kapitole 5.3 naměřená hodnota převede na teplotu ve stupních Celsia a uloží se do příslušné struktury.

TEMPSENS_HumidityGet() – zjištění aktuální vlhkosti. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení a ukazatel na strukturu, do které se má zapsat výsledek. Samotný kód probíhá tak, že se volá funkce *TEMPSENS_RegisterGet()* s parametry:

- *i2c* – ukazatel na I²C strukturu
- *addr* – adresa teplotního senzoru
- *triggerRHmeasHold* – jméno registru. Čtením z tohoto registru se spustí měření a poté je vrácena výsledná hodnota. Master je po celou dobu měření blokován a čeká se, dokud není měření dokončeno.
- *val* – proměnná, do které se zapíše výsledek

Jestliže vše proběhlo v pořádku, funkce končí. V opačném případě se pomocí vzorce v kapitole 5.3 naměřená hodnota převede na vlhkost v procentech a uloží se do příslušné struktury.

7.2.4 i2c_temp_sensor.h

Hlavičkový soubor pro `i2c_temp_sensor.c`. Je v něm definována adresa teplotního senzoru `TEMPSENS_ADDR = 0x80`. Dále obsahuje strukturu, ve které jsou definovány registry teplotního senzoru spolu s adresami těchto registrů a také obsahuje strukturu, do které se ukládá naměřená teplota a vlhkost. Nakonec obsahuje definici použitých funkcí.

7.2.5 i2c_pres_sensor.c

Obsahuje funkce pro čtení a zapsání hodnot do registrů tlakového senzoru a funkce pro zjištění atmosférického tlaku a teploty. Lze poté porovnat, zda hodnota teploty tlakového senzoru je stejná jako u teplotního senzoru či zda se liší.

Funkce:

PRESSENS_RegisterGet() – přečtení hodnoty z daného registru. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení, adresa registru, ze kterého se má číst a ukazatel na proměnou, do které se má zapsat výsledek. Samotný kód probíhá tak, že se nejdříve sestaví požadavek senzoru (adresa + bit R (read) + adresa registru), následně se odešle a čeká se na přerušení, které nastane až když byl přenos dokončen. Během čekání se MCU přepne do módu EM1. Nakonec se přečte přijatá hodnota a uloží se do dané struktury. Jestliže vše proběhlo v pořádku, vrátí se 0. Jestliže ne, vrátí se hodnota stavu linky I²C, podle které lze zjistit, k jakému problému došlo.

PRESSENS_RegisterSet() – zápis hodnoty do daného registru. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení, adresa registru, do kterého se má zapisovat a hodnota, která se má do daného registru zapsat. Samotný kód probíhá tak, že se nejdříve sestaví požadavek senzoru (adresa + bit W (write) + adresa registru + hodnota), následně se odešle a čeká se na přerušení, které nastane až když byl přenos dokončen. Během čekání se MCU přepne do módu EM1. Nakonec se vrátí se hodnota stavu linky I²C.

PRESSENS_PressureGet() – zjištění aktuálního atmosférického tlaku. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení a ukazatel na strukturu, do které se má zapsat výsledek. Samotný kód probíhá tak, že se nejdříve volá funkce *PRESSENS_RegisterSet()* s parametry:

- *i2c* – ukazatel na I²C strukturu
- *addr* – adresa tlakového senzoru
- *measure* – jméno registru. Zápisem do tohoto registru se spustí měření.
- *PRES_SENS_PRES_MEAS* – hodnota, která se zapíše. Lze nakonfigurovat v souboru `config.h`. V tomto případě se spustí měření tlaku.

Jestliže vše proběhlo v pořádku, funkce končí. V opačném případě se volá funkce *PRESSENS_RegisterGet()* s parametry:

- *i2c* – ukazatel na I²C strukturu

- *addr* – adresa tlakového senzoru
- *value0* – jméno registru. Čtením z tohoto registru získáme hodnotu nejvyššího bytu.
- *val* – proměnná, do které se zapíše výsledek.

Toto zopakujeme i pro registry *value1* a *value2* pro získání zbylých bytů. Vše se spojí do jedné proměnné a ta se uloží do příslušné struktury.

PRESSENS_TemperatureGet() – zjištění aktuální teploty. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení a ukazatel na strukturu, do které se má zapsat výsledek. Samotný kód probíhá tak, že se nejdříve volá funkce *PRESSENS_RegisterSet()* s parametry:

- *i2c* – ukazatel na I²C strukturu
- *addr* – adresa teplotního senzoru
- *measure* – jméno registru. Zápisem do tohoto registru se spustí měření.
- *PRES_SENS_TEMP_MEAS* – hodnota, která se zapíše. Lze nakonfigurovat v souboru *config.h*. V tomto případě se spustí měření teploty.

Jestliže vše proběhlo v pořádku, funkce končí. V opačném případě se volá funkce *PRESSENS_RegisterGet()* s parametry:

- *i2c* – ukazatel na I²C strukturu
- *addr* – adresa teplotního senzoru
- *value0* – jméno registru. Čtením z tohoto registru získáme hodnotu nejvyššího bytu.
- *val* – proměnná, do které se zapíše výsledek.

Toto zopakujeme i pro registry *value1* a *value2* pro získání zbylých bytů. Vše se spojí do jedné proměnné a ta se uloží do příslušné struktury.

7.2.6 *i2c_pres_sensor.h*

Hlavičkový soubor pro *i2c_pres_sensor.c*. Je v něm definována adresa tlakového senzoru *PRESSENS_ADDR = 0xEE*. Dále obsahuje strukturu, ve které jsou definovány registry tlakového senzoru spolu s adresami těchto registrů a také obsahuje strukturu, do které se ukládá naměřený tlak a teplota. Nakonec obsahuje definici použitých funkcí.

7.2.7 *i2c_accel_sensor.c*

Obsahuje funkce pro čtení a zapsání hodnot do registrů senzoru pro měření změny polohy a funkce pro zjištění aktuálních dat.

Funkce:

ACCELSENS_RegisterGet() – přečtení hodnoty z daného registru. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení, adresa registru, ze kterého se má číst a ukazatel na proměnnou, do které se má zapsat výsledek. Samotný kód probíhá tak, že se

nejdříve sestaví požadavek senzoru (adresa + bit R (read) + adresa registru), následně se odešle a čeká se na přerušení, které nastane až když byl přenos dokončen. Během čekání se MCU přepne do módu EM1. Nakonec se přečte přijatá hodnota a uloží se do dané struktury. Jestliže vše proběhlo v pořádku, vrátí se 0. Jestliže ne, vrátí se hodnota stavu linky I²C, podle které lze zjistit, k jakému problému došlo.

ACCELSENS_RegisterSet() – zápis hodnoty do daného registru. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení, adresa registru, do kterého se má zapisovat a hodnota, která se má do daného registru zapsat. Samotný kód probíhá tak, že se nejdříve sestaví požadavek senzoru (adresa + bit W (write) + adresa registru + hodnota), následně se odešle a čeká se na přerušení, které nastane až když byl přenos dokončen. Během čekání se MCU přepne do módu EM1. Nakonec se vrátí se hodnota stavu linky I²C.

ACCELSENS_AccelerationGet() – zjištění aktuální změny polohy. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení a ukazatel na strukturu, do které se má zapsat výsledek. Samotný kód probíhá tak, že se nejdříve volá funkce *ACCELSENS_RegisterSet()* s parametry:

- *i2c* – ukazatel na I²C strukturu
- *addr* – adresa senzoru pro měření polohy
- *mode* – jméno registru. Zápisem do tohoto registru se spustí měření.
- *ACCEL_SENS_MEAS* – hodnota, která se zapíše. Lze nakonfigurovat v souboru *config.h*. V tomto případě se senzor přepne do aktivního módu.

Jestliže vše proběhlo v pořádku, funkce končí. V opačném případě se volá funkce *ACCELSENS_RegisterGet()* s parametry:

- *i2c* – ukazatel na I²C strukturu
- *addr* – adresa senzoru pro měření polohy
- *xOutput* – jméno registru. Čtením z tohoto registru získáme hodnotu osy x.
- *val* – proměnná, do které se zapíše výsledek.

Toto zopakujeme i pro registry *yOutput* a *zOutput* pro získání hodnot zbylých os. Vše se uloží do příslušné struktury. Nakonec se funkce *ACCELSENS_RegisterSet()* volá ještě jednou pro přepnutí senzoru do neaktivního režimu.

7.2.8 *i2c_accel_sensor.h*

Hlavičkový soubor pro *i2c_accel_sensor.c*. Je v něm definována adresa akcelerometru *ACCELSENS_ADDR = 0x98*. Dále obsahuje strukturu, ve které jsou definovány registry akcelerometru spolu s adresami těchto registrů a také obsahuje strukturu, do které se ukládají naměřené hodnoty jednotlivých os. Nakonec obsahuje definici použitých funkcí.

7.2.9 i2c_mag_sensor.c

Obsahuje funkce pro čtení a zapsání hodnot do registrů senzoru pro měření magnetického pole a funkce pro zjištění aktuálního magnetického pole.

Funkce:

MAGSENS_RegisterGet() – přečtení hodnoty z daného registru. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení, adresa registru, ze kterého se má číst a ukazatel na proměnou, do které se má zapsat výsledek. Samotný kód probíhá tak, že se nejdříve sestaví požadavek senzoru (adresa + bit R (read) + adresa registru), následně se odešle a čeká se na přerušení, které nastane až když byl přenos dokončen. Během čekání se MCU přepne do módu EM1. Nakonec se přečte přijatá hodnota a uloží se do dané struktury. Jestliže vše proběhlo v pořádku, vrátí se 0. Jestliže ne, vrátí se hodnota stavu linky I²C, podle které lze zjistit, k jakému problému došlo.

MAGSENS_RegisterSet() – zápis hodnoty do daného registru. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení, adresa registru, do kterého se má zapisovat a hodnota, která se má do daného registru zapsat. Samotný kód probíhá tak, že se nejdříve sestaví požadavek senzoru (adresa + bit W (write) + adresa registru + hodnota), následně se odešle a čeká se na přerušení, které nastane až když byl přenos dokončen. Během čekání se MCU přepne do módu EM1. Nakonec se vrátí se hodnota stavu linky I²C.

MAGSENS_MagneticGet() – zjištění aktuálního magnetického pole. Vstupem této funkce je ukazatel na I²C strukturu, adresa zařízení a ukazatel na strukturu, do které se má zapsat výsledek. Samotný kód probíhá tak, že se nejdříve volá funkce *MAGSENS_RegisterSet()* s parametry:

- *i2c* – ukazatel na I²C strukturu
- *addr* – adresa magnetického senzoru
- *controlReg1* – jméno registru. Zápisem do tohoto registru se spustí měření.
- *MAG_SENS_MEAS* – hodnota, která se zapíše. Lze nakonfigurovat v souboru *config.h*. V tomto případě se spustí měření s frekvencí 5 Hz.

Jestliže vše proběhlo v pořádku, funkce končí. V opačném případě se volá funkce *MAGSENS_RegisterGet()* s parametry:

- *i2c* – ukazatel na I²C strukturu
- *addr* – adresa teplotního senzoru
- *xOutputHigh* – jméno registru. Čtením z tohoto registru získáme hodnotu vyššího bytu osy x.
- *val* – proměnná, do které se zapíše výsledek.

Toto zopakujeme i pro registry *xOutputLow*, *yOutputHigh*, *yOutputLow*, *zOutputHigh* a *zOutputLow* pro získání hodnot zbylých os. Vše se uloží do příslušné struktury.

Nakonec se funkce *MAGSENS_RegisterSet()* volá ještě jednou pro přepnutí senzoru do neaktivního režimu.

7.2.10 *i2c_accel_sensor.h*

Hlavičkový soubor pro *i2c_accel_sensor.c*. Je v něm definována adresa magnetického senzoru *MAGSENS_ADDR = 0xC4*. Dále obsahuje strukturu, ve které jsou definovány registry magnetického senzoru s adresami těchto registrů a také obsahuje strukturu, do které se ukládají naměřené hodnoty jednotlivých os. Nakonec obsahuje definici použitých funkcí.

7.2.11 *config.h*

Obsahuje konfigurační proměnné. Více informací v kapitole 7.3

7.2.12 *spi.c*

Inicializace SPI rozhraní pro komunikaci s bezdrátovým modulem a funkce pro povolení a obsluhu přerušení od SPI sběrnice.

Funkce:

SPI_setup() – nastavení UART rozhraní jako SPI. Vstupem je číslo UART, které se má použít, konfigurace pinů, která se má použít a zda se jedná o master zařízení nebo slave. Samotný kód probíhá tak, že se nejdříve nastaví, o jaké USART zařízení jde (podle číslo UART) a nastaví se baudová rychlost. V samotné konfiguraci se nastaví, že se má jednat o synchronní mód, smažou se staré přenosy (pokud nějaké jsou) a zakážou se přerušení. Poté se povolí příslušné piny. Jestliže se jedná o zařízení master, jsou mu povoleny operace TX a RX (posílání a přijímání dat) a konfigurace GPIO je nastavena jako master. Nakonec se GPIO konfigurace přiřadí konkrétním pinům na zařízení

SPII_setupRXInt() – funkce pro povolení přerušení pro příjem dat z USART1. Jako vstupem je přijímací buffer a počet bytů, které se mají přijmout. Funkce probíhá tak, že se naplní přijímací buffer pro master příslušnými daty, vymaže se přijímací posuvný registr a povolí se příslušné přerušení.

SPII_setupTXInt() – funkce pro povolení přerušení pro vysílání dat přes USART1. Jako vstupem je vysílací buffer a velikost vysílacího bufferu. Funkce probíhá tak, že se naplní vysílací buffer pro master příslušnými daty, vymaže se vysílací posuvný registr a povolí se příslušné přerušení.

SPII_setupInt() – Volá funkce *SPII_setupRXInt()* a *SPII_setupTXInt()*.

USART1_RX_IRQHandler() – obsluha přerušení pro příjem dat. Jestliže je přerušení povoleno a v přijímacím posuvném registru se nacházejí data, ulož tyto data do přijímacího bufferu pro master. Toto prováděj do doby, dokud není přijímací buffer plný.

USART1_RX_IRQHandler() – obsluha přerušení pro vysílání dat. Jestliže je přerušení povoleno, zkontroluj, zda jsou připravena nějaká data pro vysílání. Pokud ano, přesuň je

do vysílacího posuvného registru a zkontroluj, zda jsou připravena další data. Pokud ne, ulož do posuvného registru 0.

7.2.13 spi.h

Hlavičkový soubor pro `spi.c`. Obsahuje definici použitých funkcí.

7.2.14 usart.c

Obsahuje funkci pro posílání dat.

Funkce:

USART1_sendBuffer() – posílání dat za použití USART1. Jako vstupem je vysílací buffer a délka tohoto bufferu. Ve funkci je cyklus, který běží přes všechny byty bufferu. Vždy se čeká, dokud není USART připraven pro vysílání a poté se aktuální byte nahraje do vysílacího posuvného registru a ten je poté odvyslán. Pokud již nejsou žádná data k odvysílání, vysílací registr se vyprázdní. Jestliže se jedná o příjem, místo ukazatele na vysílací buffer se nastaví 0.

7.2.15 usart.h

Hlavičkový soubor pro `usart.c`. Obsahuje definici několika konstant pro USART a definici funkce *USART1_sendBuffer()*.

7.2.16 i2c_sensors_main.c

Hlavní funkce – periodicky sbírá data ze senzorů a předává je bezdrátovému modulu, který je přeposílá dál.

Funkce:

SPI_init(void) – inicializace SPI. Zapnou se hodiny pro USART1 a pro GPIO a volá se funkce *SPI_init(void)* pro inicializaci SPI. Poté se přes SPI pošle bezdrátovému modulu požadavek na reset zařízení. Inicializace modulu se po resetu provede sama. Nakonec se nastaví kanál, implicitně je nastaven kanál 26 (2480 MHz).

main(void) – hlavní funkce. Na začátku se volají funkce pro inicializaci MCU, I²C a SPI. Následuje nekonečná smyčka. Pro každý senzor se nejdřív zkontroluje, zda je povolen, poté se volají funkce pro spuštění měření a získání naměřených dat. Nakonec se tyto data přidají do vysílacího bufferu a čeká se krátký čas před měřením z dalšího senzoru. Po získání dat ze všech senzorů se připraví vysílací rámeček. Na příslušné adresy v bezdrátovém modulu se nahraje velikost hlavičky, velikost rámce, hlavička a data. Hlavička obsahuje MAC adresu zařízení. Data obsahují naměřené hodnoty ze senzorů oddělené '-'. Po nahrání dat se do Transmit Normal FIFO Control registru nahraje hodnota 1 a data jsou odeslána na příslušném kanálu. Na konci se čeká zadaný čas (implicitně 10 minut) před dalším měřením. V neaktivním čase se MCU přepne do módu EM2 pro šetření energie.

7.3 Konfigurace

Soubor `config.h` obsahuje konfigurační proměnné pro nastavení programu. Níže je uveden seznam a popis těchto proměnných.

- *WAIT_TIME* – čas, jak dlouho MCU čeká mezi měřeními. Zadává se v milisekundách. Implicitně je nastavena na 600000 (10 minut).
- *WAIT_TIME_SENS* – čas, jak dlouho MCU čeká mezi měřeními u jednotlivých senzorů. Zadává se v milisekundách. Implicitně je nastavena na 20
- *TEMP_MEAS* – povolení měření z teplotního senzoru. Povolené hodnoty jsou: 0 – měření zakázáno, 1 – měření povoleno. Implicitně je nastavena na 1
- *PRES_MEAS* – povolení měření z tlakového senzoru. Povolené hodnoty jsou: 0 – měření zakázáno, 1 – měření povoleno. Implicitně je nastavena na 1
- *ACCEL_MEAS* – povolení měření ze senzoru polohy. Povolené hodnoty jsou: 0 – měření zakázáno, 1 – měření povoleno. Implicitně je nastavena na 1
- *MAG_MEAS* – povolení měření z magnetického senzoru. Povolené hodnoty jsou: 0 – měření zakázáno, 1 – měření povoleno. Implicitně je nastavena na 1
- *PRES_SENS_PRES_MEAS* – hodnota, která se napíše do registru pro měření tlaku u tlakového senzoru:
 - *0x34* – Ultra Low Power mód
 - *0x74* – Standard mód
 - *0xB4* – High Resolution mód
 - *0xF4* – Ultra High Resolution módImplicitně je nastavena hodnota *0x74*.
- *ACCEL_SENS_MEAS* – hodnota, která se napíše do registru pro měření polohy u senzoru polohy. Implicitně je nastavena hodnota *0x01* (aktivní mód).
- *MAG_SENS_MEAS* – hodnota, která se napíše do registru pro měření magnetického pole u magnetického senzoru:
 - *0x01* – 80x měření / sekundu
 - *0x21* – 40x měření / sekundu
 - *0x41* – 20x měření / sekundu
 - *0x61* – 10x měření / sekundu
 - *0x81* – 5x měření / sekundu
 - *0xA1* – 2,5 měření / sekundu
 - *0xC1* – 1,25 měření / sekundu
 - *0xE1* – 0,63 měření / sekunduImplicitně je nastavena hodnota *0x61*. Více informací viz [11] kapitola 5.5.
- *WIRELESS_CHANNEL* – hodnota kanálu, na kterém vysílá a přijímá bezdrátový modul. Implicitně je nastavena hodnota *F3*, která znamená kanál 26. Seznam kanálů viz kapitola 4.3.5.

8 Závěr

Tato práce pojednávala o systému na měření fyzikálních parametrů prostředí. Při návrhu programu byl kladen velký důraz na minimální spotřebu celého systému. MCU se při nečinnosti přepíná do módu EM2. Sensory se při neaktivitě nachází v neaktivních režimech nebo režimech spánku. U bezdrátového modulu se režim vysílání zapíná pouze ve chvíli, kdy je připraven rámeček na odvysílání, po odvysílání rámce se modul automaticky přepíná do neaktivního režimu.

Program by bylo možné rozšířit o situaci, kdy by měření neprobíhalo periodicky, ale vždy na podnět z vnějšku. Bylo by nutné nastavit, aby bezdrátový modul periodicky poslouchal na daném kanálu, jestli není nějaký rámeček určen tomuto systému. Při zachycení příslušného rámce by vyslal přerušení mikrokontroléru, který by se probudil, získal data ze senzorů, odeslal data bezdrátovému modulu a opět by se uspal. Ten by je následně odeslal pryč. Zde by ale bylo nutné vytvořit nějakou synchronizaci s měřicí stanicí, aby bezdrátový modul na daném kanálu poslouchal co nejkratší dobu, jelikož režim přijímání a vysílání je energeticky nejnáročnější operace celého systému.

Bohužel se vyskytl problém se samotným hardwarem. Na dodaném kitu se vyskytla chyba a opravený hardware se nestihl dodat včas. Získané teoretické spotřeby (vypočtené v kapitole 6) nebylo možné porovnat s reálným zařízením a nebylo možné otestovat funkčnost navrženého řešení. Proto nebyl bod 3 ze zadání splněn.

Použité zkratky a termíny

APB	Pokročilá periferní sběrnice
ACMP	Analogový komparátor, používá se pro porovnávání hodnoty napětí
ADC	Převod analogových signálů na digitální data
AES	Symetrická bloková šifra
Arbitrace	Kontrola, zda je hodnota na I ² C sběrnici stejná, jakou zařízení aktuálně vysílá
ARM	Architektura procesorů
ARM® Cortex®	32-bitová architektura procesorů
Baudová rychlost	Modulační rychlost, počet symbolů přenesených za sekundu
BOD	Monitoruje napětí, při nízkém napětí nastavuje vodič BROWNOUTn na nízkou úroveň
Bootloader	Umožňuje uživateli programovat přes UART i bez potřeby mít debugger
Buffer	Vyrovnávací paměť, slouží k dočasnému uchování dat
CMU	Správa hodin a oscilátorů
CSMA/CA	Patří do třídy protokolů označovaných jako metody s vícenásobným přístupem, snaha vyhnout se kolizím
DAC	Převod digitální hodnoty na analogovou
DBG	Rozhraní, které se používá k programování a ladění
DMA	Provádí operace s pamětí nezávisle na procesoru
EMU	Správa spotřeby MCU
FLASH	Uchovává data na delší dobu
FSK	Frekvenční modulace, digitální informace se přenáší pomocí diskrétních změn frekvence nosné vlny
Full Duplex	Obousměrná komunikace může probíhat současně
GFSK	Typ frekvenční modulace, používá Gaussův filtr
GMSK	Typ frekvenční modulace

Half Duplex	V každý okamžik může probíhat komunikace pouze jedním směrem
I ² C	Sériové 2-vodičové rozhraní
Krystalový oscilátor	Oscilátor, který využívá rezonanci vibrujícího krystalu jako zdroj signálu
LESENSE	Rozhraní s nízkou spotřebou pro měření dat ze senzorů
MAC	Vrstva zajišťující fyzické adresování a přístup k médiu
MCU	Mikrokontrolér, obsahuje integrované periferie
MSK	Typ frekvenční modulace
MSPS	Jednotka pro vzorkování signálu
Oscilátor	Obvod produkující opakující se kmitající signál
Q-QPSK	Typ fázové modulace
Paritní bit	Bit přidávaný ke slovu, označuje počet jedničkových bitů ve slově
PCNT	Počítání přichozích pulzů na jednom vstupu
POR	Označuje, zda napájecí napětí dosáhlo minimální požadované hodnoty
Procesor	Složité číslicový obvod, který vykonává strojové instrukce
PRS	Síť umožňující přímou komunikaci periferních modulů
RAM	Paměť s přímým přístupem, uchovává data pouze na krátkou dobu
RC oscilátor	Zpětnovazební oscilátor, zpětnou větev tvoří RC článek
Reset	Signál, který ukončí chod systému a nastartuje jej znovu od začátku
RTC	Slouží k udržování informací o aktuálním času
RTOS	Operační systémy reálného času – umožňuje reagovat na události v okolí v reálném čase
Senzor	Zařízení, které měří určitou fyzikální nebo technickou veličinu a převádí jí na signál
SPI	4-vodičové sériové periferní rozhraní

USART	Zařízení pro sériovou komunikaci, lze mu nastavit synchronní nebo asynchronní režim
USB	Univerzální sériová sběrnice, slouží k propojení periférií k počítači
VCMP	Slouží k monitorování vstupního napětí
Watchdog	Generuje signál reset v případě chyby v systému

Literatura

- [1] Silicon Laboratories Inc. EFM32TG Reference manual. [online] [cit. 2014-03-26] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG-RM.pdf>>
- [2] DUDÁČEK Karel. Navrhování mikropočítačových systémů. [online] [cit. 2014-03-27] <<http://home.zcu.cz/~dudacek/NMS/PDF/>>
- [3] Silicon Laboratories Inc. EFM32TG210 Data Sheet. [online] [cit. 2014-04-10] <<http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32TG210.pdf>>
- [4] Freescale. KL24 Sub-Family Data Sheet. [online] [cit. 2014-04-10] <http://cache.freescale.com/files/32bit/doc/data_sheet/KL24P80M48SF0.pdf?fasp=1&Parent_nodeId=1339527982286712105629&Parent_pageType=product>
- [5] NXP. LPC1110/11/12/13/14/15 Data Sheet. [online] [cit. 2014-04-10] <http://www.nxp.com/documents/data_sheet/LPC111X.pdf>
- [6] Analog Devices. Low Power IEEE 802.15.4/Proprietary GFSK/FSK Zero-IF 2.4 GHz Transceiver IC ADF7242. [online] [cit. 2014-04-10] <http://www.analog.com/static/imported-files/data_sheets/ADF7242.pdf>
- [7] Microchip. MRF24J40 Data Sheet. [online] [cit. 2014-04-10] <<http://ww1.microchip.com/downloads/en/DeviceDoc/39776C.pdf>>
- [8] Texas Instruments. CC2520 DATASHEET. [online] [cit. 2014-04-10] <<http://www.ti.com/lit/ds/symlink/cc2520.pdf>>
- [9] YIU, Joseph. The definitive guide to ARM Cortex-M3. Second edition. Burlington: Elsevier Inc, 2010. ISBN 978-1-85617-963-8
- [10] Freescale. 3-Axis Orientation/Motion Detection Sensor. [online] [cit. 2014-02-17] <http://www.freescale.com.cn/files/sensors/doc/data_sheet/MMA7660FC.pdf?fp=1>
- [11] Freescale. Xtrinsic MAG3110 Three-Axis, Digital Magnetometer. [online] [cit. 2014-02-17] <http://cache.freescale.com/files/sensors/doc/data_sheet/MAG3110.pdf>
- [12] Sensirion. Datasheet SHT21. [online] [cit. 2014-02-17] <http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT21_Datasheet_V3.pdf>
- [13] Bosh. BMP180, Digital pressure sensor. [online] [cit. 2014-02-17] <<http://www.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>>

Příloha A – Schéma zapojení systému

Schéma mikrokontroléru:

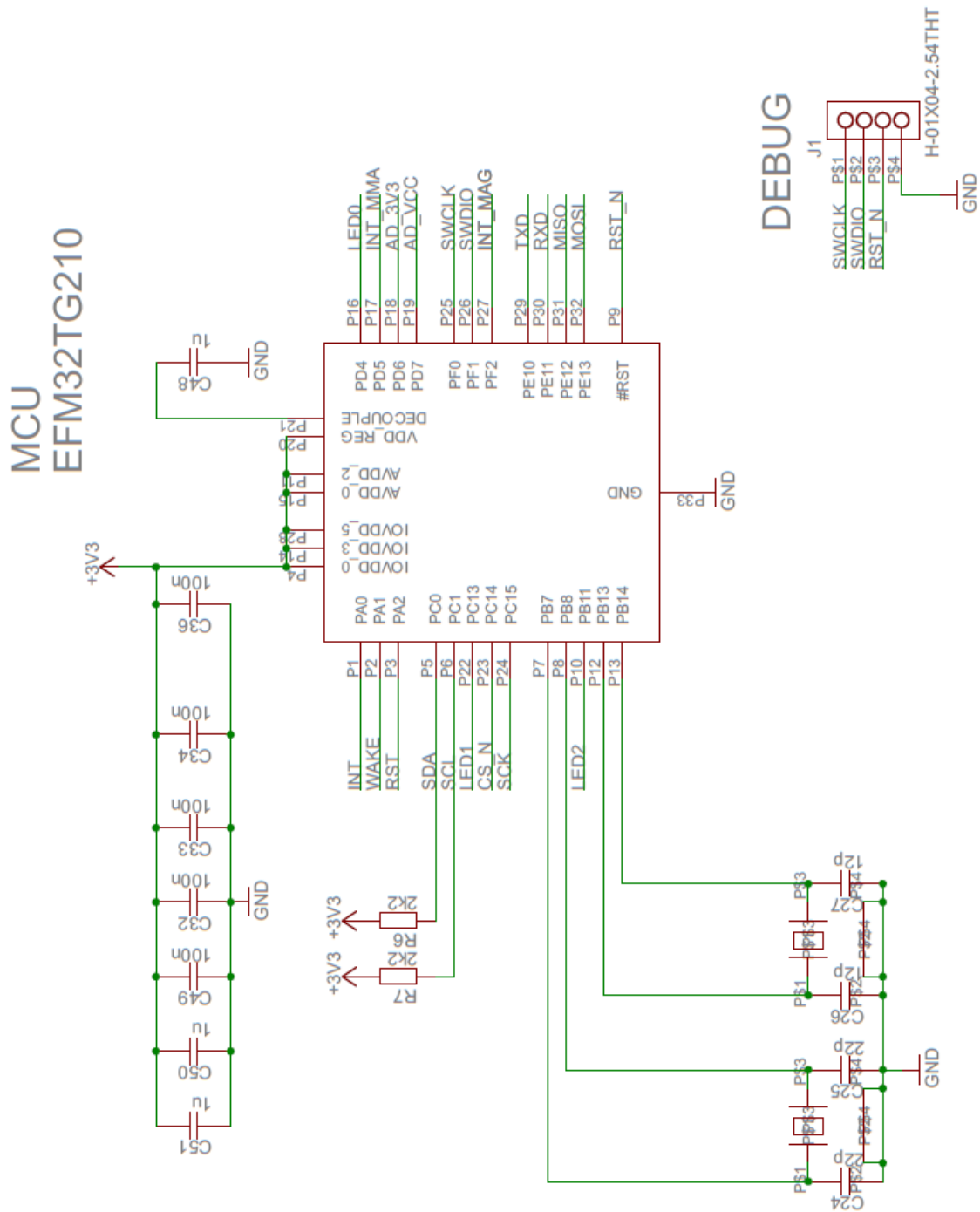


Schéma bezdrátového modulu:

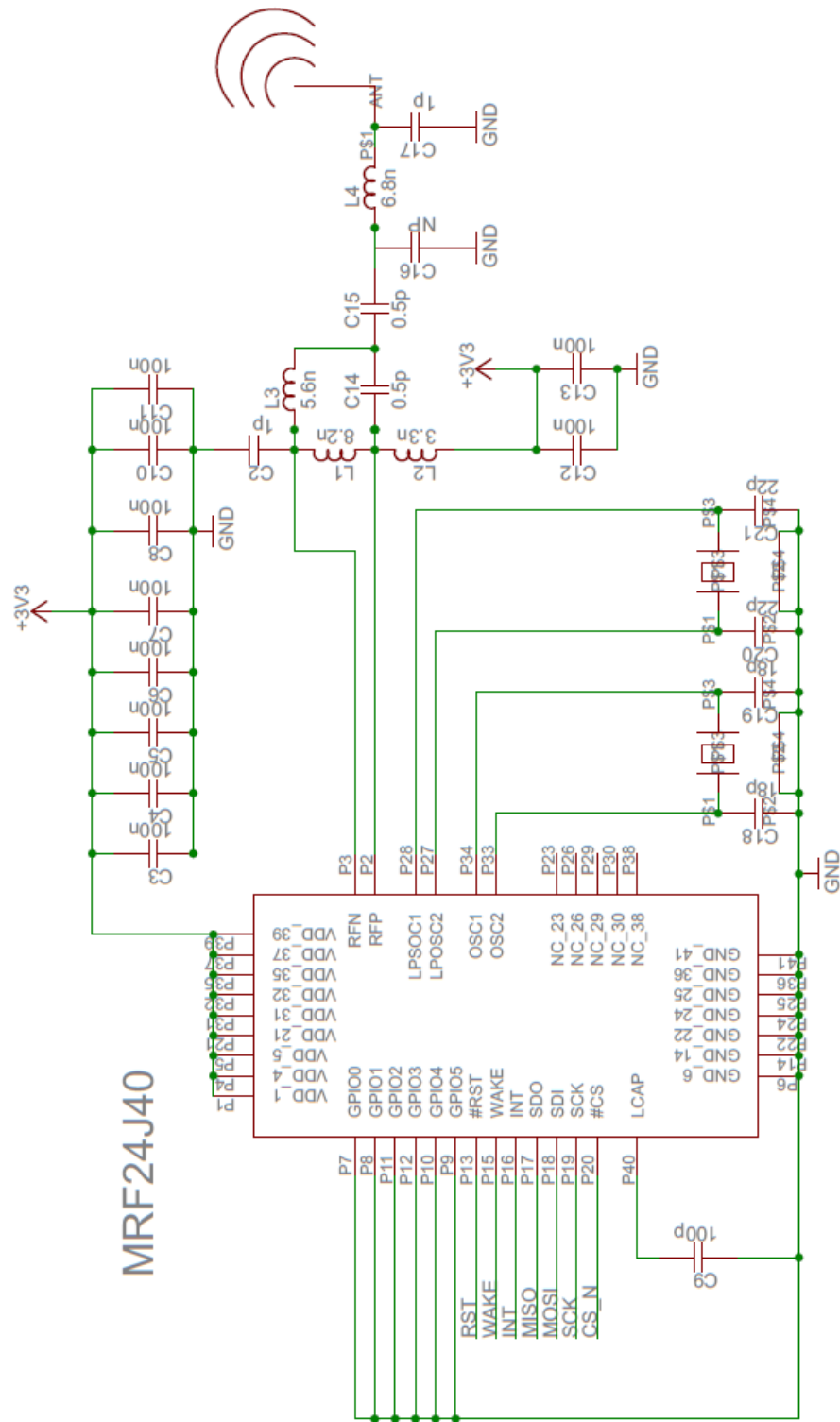
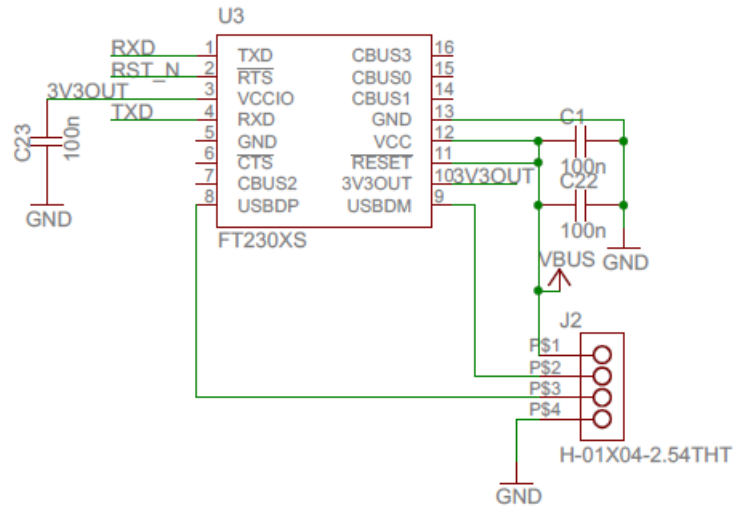


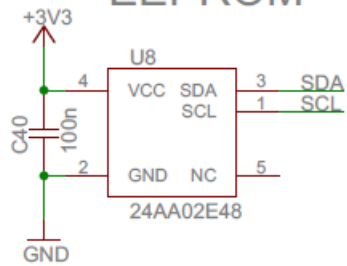
Schéma USB rozhraní, paměti EEPROM a signalizačních diod::

USB-to-UART

UART
BOOTLOADER



EEPROM



LEDs

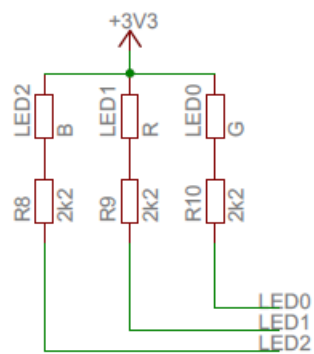
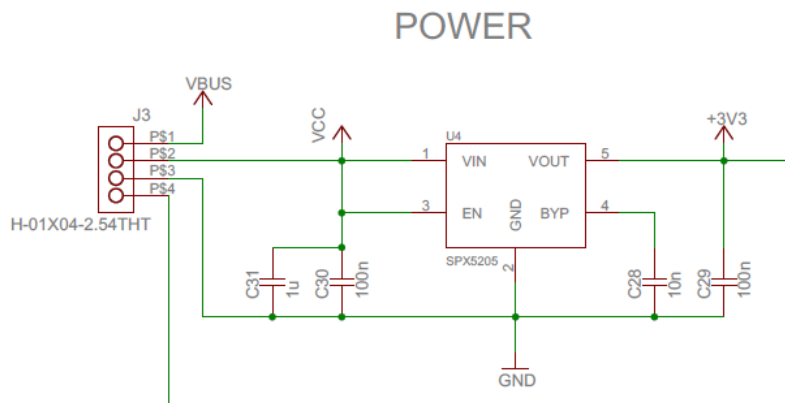
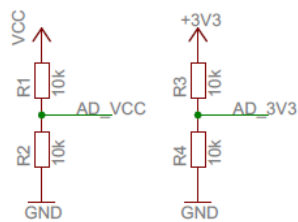


Schéma napájení:



POWER SENSE



RST

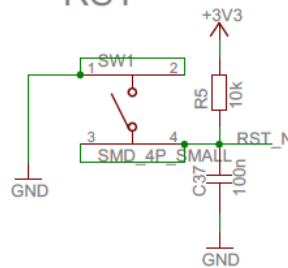
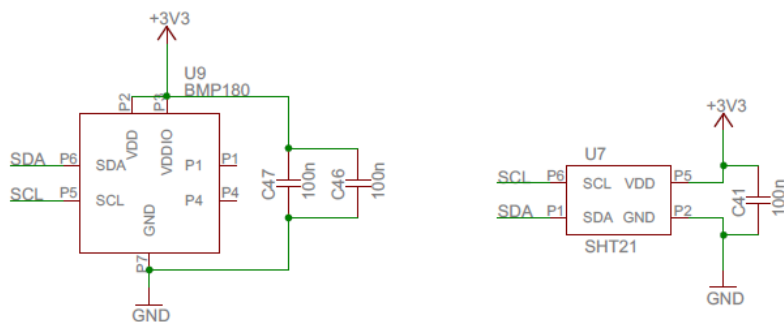
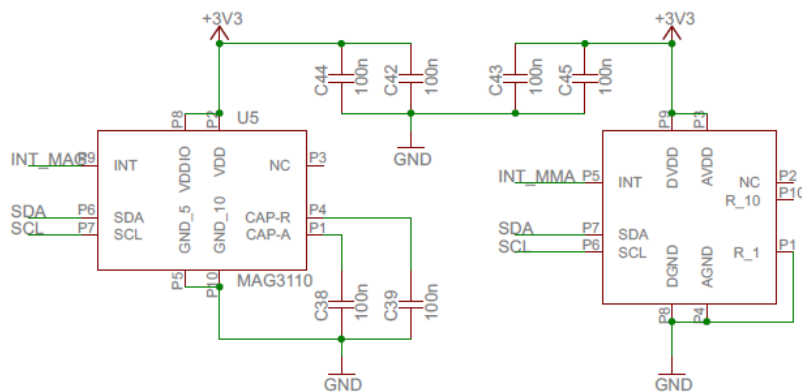


Schéma senzorů:

SENSORS



Příloha B – Uživatelská příručka

1) Stažení a nainstalování potřebného SW

Pro potřeby programu je vhodné stáhnout 2 programy uvedené níže. Nejsou nezbytně nutné, lze použít i nějaké jiné vývojové prostředí. Tyto programy jsem však využíval já a přiložený projekt je určen přímo pro IAR Embedded Workbench. Není pak nutné řešit připojování různých knihoven potřebných pro překlad a ladění. Obecně lze použít jakékoli prostředí s arm překladačem.

- Simplicity Studio – software, který poskytuje instantní přístup ke všem prostředkům, vývojovým nástrojům, dokumentům a zdrojům pro použití mikrokontroléru EFM32 Tiny Gecko.

<http://www.silabs.com/products/mcu/pages/simplicity-studio.aspx>

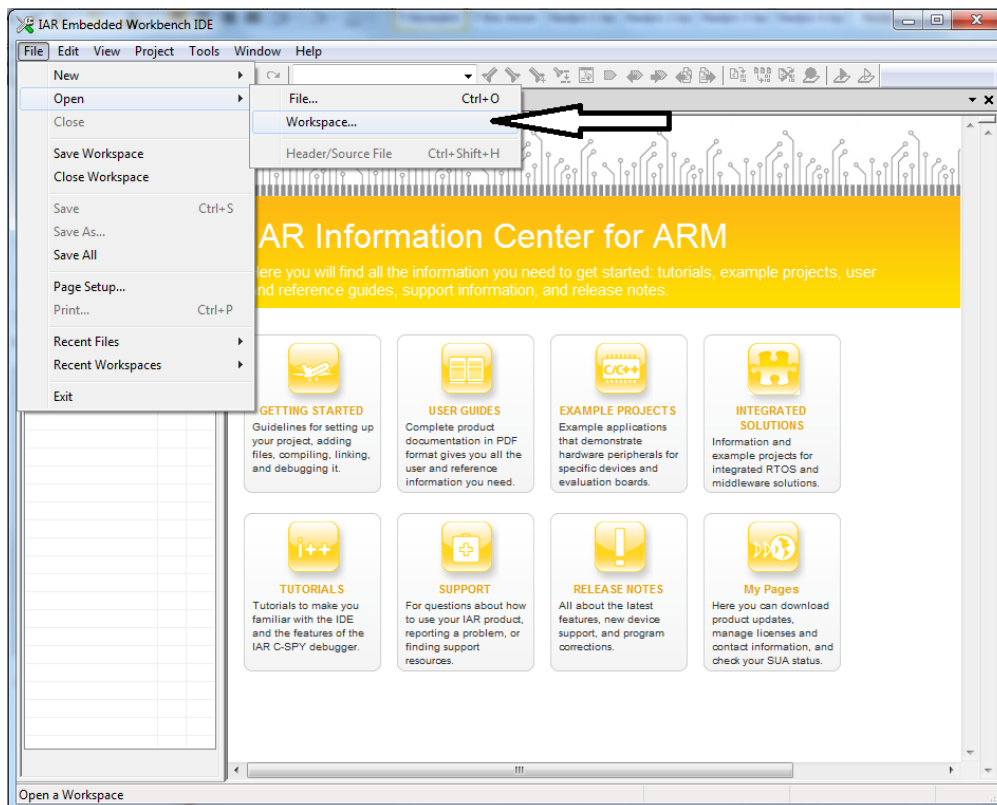
- IAR Embedded Workbench – vývojové prostředí pro tvorbu programů pro mikrokontroléry. Pro spuštění je nutné provést free registraci (stačí vybrat maximální velikost zdrojového souboru 32 KB)

<http://supp.iar.com/Download/SW/?item=EWARM-EVAL>

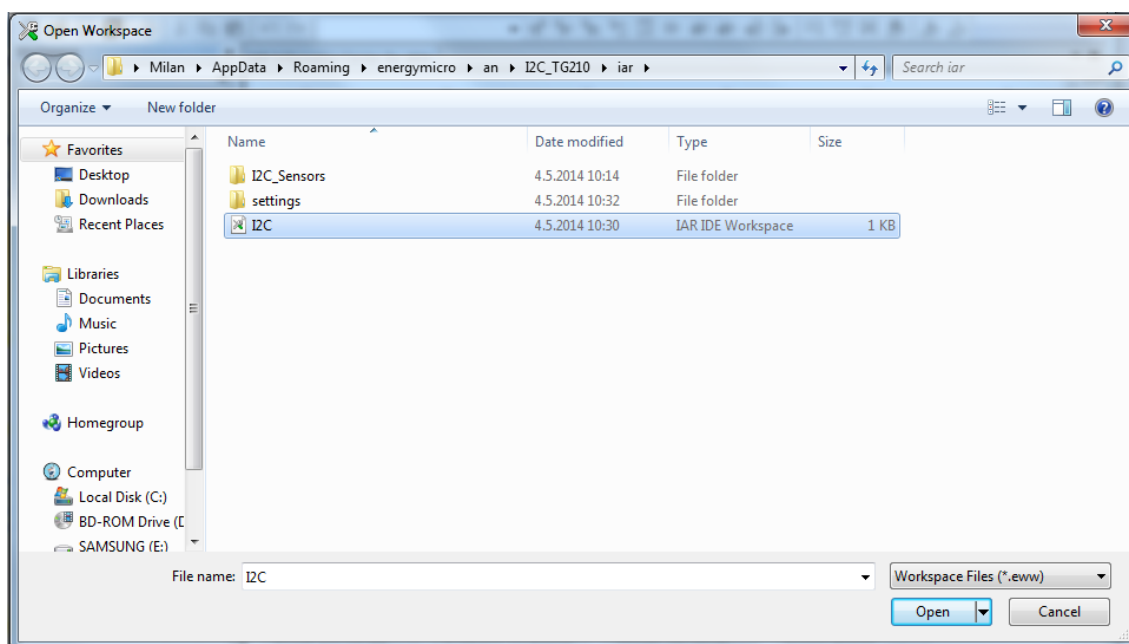
Oba programy lze stáhnout z internetu (viz uvedené odkazy) a zároveň jsou na přiloženém DVD.

2) Otevření projektu

V nabídce File vyberte Open a poté Workspace.



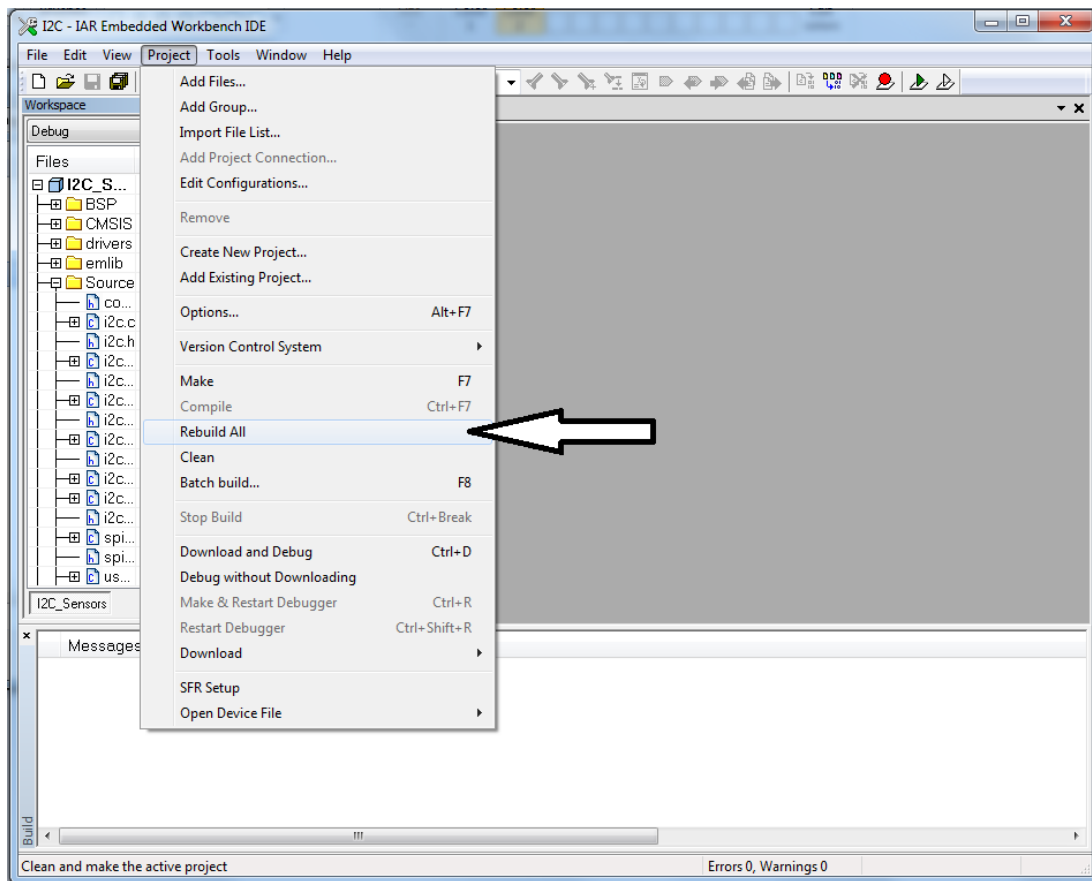
Vyberte složku, kde je umístěn projekt, přejděte do složky iar a vyberte I2C.eww.



Tím se otevře projekt.

3) Přeložení projektu

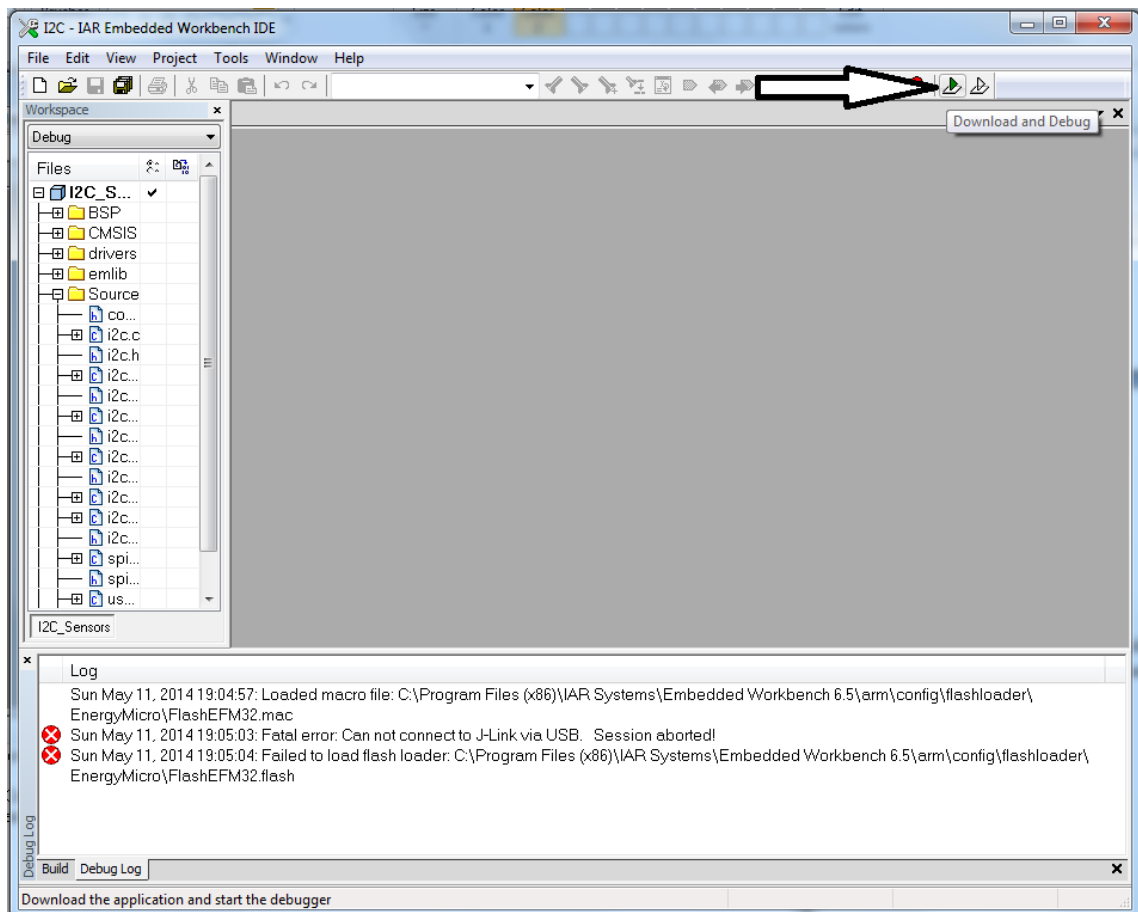
Projekt se přeloží vybráním možnost Project -> Rebuild All. Tím se přeloží kompletně celý projekt.



4) Ladění programu

Pro ladění je nutné mít zařízení J-link od firmy Segger.

- 1) Nejdříve připojte modul přes USB do počítače
- 2) Nainstalujte ovladače pro USB. Lze stáhnout odtud: <http://www.ftdichip.com/Support/Documents/InstallGuides.htm>
- 3) Připojte zařízení J-link k počítači a nechte nainstalovat automatické ovladače
- 4) Propojte J-link s modulem
- 5) V IAR Embedded Workbench vyberte Download and Debug.



- 6) Tím se dostanete do ladícího režimu, ve kterém je možné krokovat, vypisovat si hodnoty proměnných a registrů a testovat funkci programu

Vytvoření vlastního projektu

Pro vytvoření vlastního projektu postupujte podle tohoto odkazu:

<http://community.silabs.com/t5/Microcontroller-How-to-Guides/Create-a-new-project-in-IAR/m-p/101935>

U nového projektu je nutné nastavit složky knihoven a nastavení ladícího modulu. U přiloženého projektu jsou již všechny tyto věci nastaveny

Použití Bootloaderu

Spuštění Bootloaderu

- 1) Pin DBG_SWCLK je nutné nastavit na sysokou úroveň a MCU musí být vyresetován
- 2) UART Bootloader používá pro komunikaci piny E11 a E10. Pro automatické určení přenosové rychlosti se využívá funkce autobaud.
- 3) Do příkazové řádky lze vkládat 1 znakové příkazy

Nahrání aplikace

- 1) Pro nahrání aplikace musí být použit příkaz 'u' nebo 'd'. Je nutné použít terminálový software pro nahrání souboru
- 2) Nahrání má 2 režimy – destruktivní přepíše bootloader, obvyčejný ho ponechá

Vytvoření aplikace a podrobnější popis:

<http://www.silabs.com/Support%20Documents/TechnicalDocs/AN0003.pdf>