

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

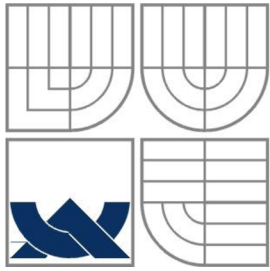
**PROGRAMOVÉ VYBAVENÍ PRO**  
**VÍCEJEDNOTKOVOU BEZDRÁTOVOU SPORTOVNÍ**  
**ČASOMÍRU**

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

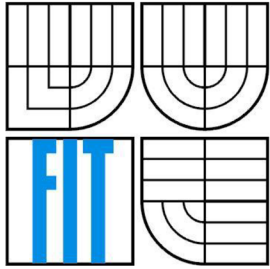
**AUTOR PRÁCE**  
AUTHOR

**DAVID GÁL**

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# PROGRAMOVÉ VYBAVENÍ PRO VÍCEJEDNOTKOVOU BEZDRÁTOVOU SPORTOVNÍ ČASOMÍRU

FIRMWARE FOR MULTI-UNIT WIRELESS SPORTS TIMEKEEPING DEVICE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DAVID GÁL

VEDOUCÍ PRÁCE

SUPERVISOR

ING. JOSEF STRNADEL, PH.D.

BRNO 2015

## **Abstrakt**

Tato práce se zabývá návrhem a následnou realizací programového vybavení pro vícejednotkovou bezdrátovou sportovní časomíru. Použitý hardware vychází z jiné bakalářské práce a je složen z řídicí, primární jednotky, a z koncových, sekundárních, jednotek. Oba typy jsou k řízení vybaveny 16bitovými mikrokontroléry PIC24F a bezdrátovou komunikaci obstarávají RF moduly vysílající v pásmu 868MHz. Návrh je zaměřen především na bezdrátové využití sekundárních jednotek k měření času a na eliminaci chyb měření spojených s bezdrátovým přenosem. Návrh se také zabývá snížením spotřeby jednotek a použitím SD karty k ukládání naměřených časů. Realizace je provedena v jazyce C.

## **Abstract**

This paper describes the design and subsequent implementation of software for multi-unit wireless sports timekeeping device. Used hardware is based on a different thesis and is composed of a control, the primary unit, and end, secondary, units. Both types are equipped with 16-bit PIC24F microcontrollers and wireless communication is catered by RF modules broadcasting in the 868MHz band. The design is aimed primarily at wireless use of secondary units to measure time and to eliminate measurement errors associated with wireless transmission. It also deals with reducing the consumption of units and the use SD cards to store the measured times. Implementation is done in C.

## **Klíčová slova**

měření času, bezdrátová komunikace, časová synchronizace, Microchip, MCU, PIC, firmware

## **Keywords**

time measurement, wireless communication, time synchronization, Mircochip, MCU, PIC, firmware

## **Citace**

Gál David: Programové vybavení pro vícejednotkovou bezdrátovou sportovní časomíru, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Programové vybavení pro vícejednotkovou bezdrátovou sportovní časomíru

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Josefa Strnadela, Ph.D. Další informace mi poskytl autor hardwaru časomíry Marek Gál.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
David Gál  
19.05.2015

## Poděkování

Děkuji vedoucímu bakalářské práce Ing. Josefu Strnadelovi, Ph.D. za jeho ochotnou pomoc při řešení práce. Dále děkuji svému bratrov, autorovi hardwarové části, Marku Gálovi za poskytnuté rady.

© David Gál, 2015

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem defnovaných případů..*

# Obsah

1	Úvod.....	2
2	Současný stav.....	3
2.1	Primární jednotka.....	3
2.2	Sekundární jednotka .....	11
2.3	Vývojové prostředky .....	14
2.4	Shrnutí požadavků na software.....	15
3	Návrh programového vybavení.....	17
3.1	Měření času.....	17
3.2	Bezdrátová komunikace.....	18
3.3	Uživatelské rozhraní .....	20
3.4	Snížení spotřeby.....	23
4	Implementace.....	24
4.1	Měření času.....	24
4.2	Bezdrátová komunikace.....	25
4.3	Primární jednotka.....	28
4.4	Sekundární jednotka .....	32
5	Ověření funkčnosti.....	34
5.1	Přesnost bezdrátového měření .....	34
5.2	Spotřeba a výdrž na akumulátoru .....	35
5.3	Ověření v praxi .....	36
6	Závěr .....	37

# 1 Úvod

Tato práce se zabývá realizací programového vybavení pro vícejednotkovou sportovní bezdrátovou časomíru, která vychází z bakalářské práce Marka Gála [1], studenta Fakulty elektrotechniky a komunikačních technologií VUT v Brně.

Časomíra je navržena jako systém složený z primární jednotky, která funguje jako mozek celého systému, a z jednotek sekundárních, které poskytují informace z měřících senzorů. Primární jednotka použitá v této práci je modifikovaná verze původního návrhu. Sekundární jednotka z původního návrhu nevychází. Původní verze sekundární jednotky nebyla upravena, což znesnadňuje její použití v kombinaci s modifikovanou primární jednotkou. Proto zde nebude použita.

V rámci zadání původní práce bylo i vytvoření programového vybavení demonstrujícího funkčnost časoměrného zařízení. Návrh není pro další použití příliš vhodný.

Hlavním požadavkem na programové vybavení časomíry je zajištění dostatečně přesného měření času a eliminace zpoždění při přenosu časové informace. Komunikace musí být také dostatečně spolehlivá. Velkým problémem by bylo nedoručení paketu obsahujícího informaci o zastavení měřeného času z důvodu zarušení přenosového pásma, nebo kvůli kolizím při použití většího počtu sekundárních jednotek. S tím musí návrh bezdrátové komunikace počítat a eliminovat tyto rizika.

Dále musí firmware, vzhledem k napájení jednotek z vestavěných akumulátorů, vhodně hospodařit s energií a zajistit tak co nejdelší výdrž. Neméně důležité je i vytvoření přívětivého uživatelského rozhraní a s tím spojenou správu naměřených časů. K tomu by měla napomoc možnost ukládat časy na externí paměť v podobě odnímatelné SD karty.

Náplň této práce ani zdaleka nezahrnuje všechny možnosti, které hardware řešení umožňuje. Vzhledem k modularitě s jakou je časomíra vytvořena, je možné do systému fyzicky přidávat další zařízení, např. externí zobrazovač naměřených časů. Primární jednotka je také vybavena pro komunikaci po USB sběrnici, takže může být propojena s PC. Proto by měl být firmware realizován tak, aby jej bylo možné v budoucnu dále rozšiřovat.

## 2 Současný stav

V této kapitole jsou blíže popsány zařízení použité v této práci. Popis je zaměřen především na prvky důležité pro návrh programového vybavení. Jsou zde také zmíněny rozdíly současného řešení jednotek oproti jejich původním verzím. Zahrnutý jsou i dostupné vývojové prostředky.

### 2.1 Primární jednotka

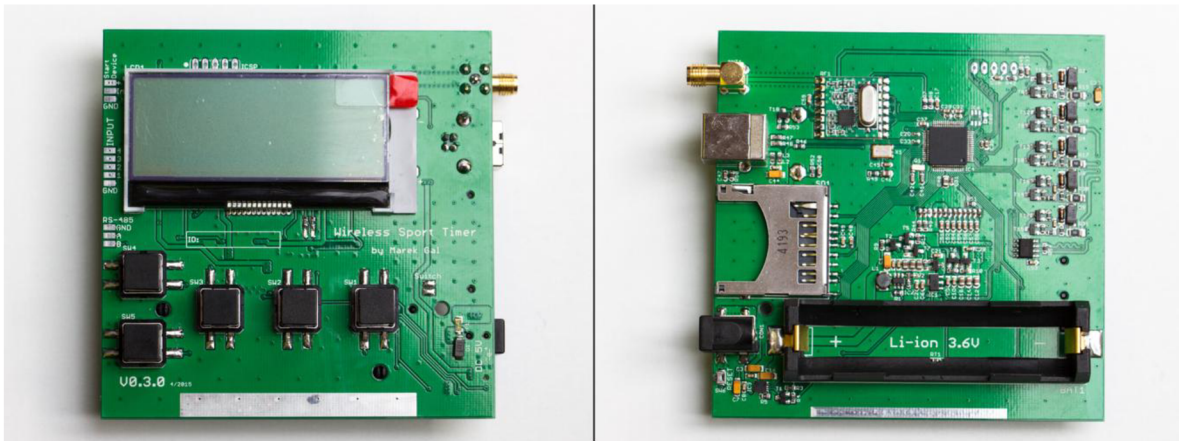
Primární jednotka zaznamenala oproti původní verzi řadu změn. Na první pohled nejvýraznější změnou je úprava fyzického provedení jednotky. Z plastové krabičky byla jednotka přesunuta do kovového pouzdra. Rozdíl v provedení jednotek je zobrazen na obrázku 1. Z obrázku je také patrné, že došlo k přeuspořádání ovládacích tlačítek. Tato změna, i když zdánlivě bezvýznamná, bude mít vliv v návrhu uživatelského rozhraní.



Obrázek 1-Původní verze primární jednotky vlevo, vpravo aktuální verze

Hlavní změny, které jsou podstatné pro tuto práci, se ovšem odehráli uvnitř. Úplně nahrazen byl alfanumerický displej a zdroj hodinových impulzů. Došlo zde také k řadě změn v propojení periférií s mikrokontrolérem, tak aby se dalo lépe využít jejich možností. Některé části byly kompletně odstraněny. Konkrétně byla odstraněna jedna ze dvou sběrnic RS485.

Aktuální konfigurace primární jednotky je nyní: MCU, bezdrátový modul, ovládací tlačítka, displej, napájecí modul, zdroje hodinových impulzů, sada vstupních konektorů a slot pro SD kartu. Obrázek 2 ukazuje rozmístění těchto prvků na DPS. Jednotlivé části jsou blíže popsány dále v této kapitole.



Obrázek 2-Deska primární jednotky

## 2.1.1 Mikrokontrolér

K řízení je jednotka vybavena mikrokontrolérem PIC24FJ256GB206 [2] od firmy Microchip. Jedná se o 16bitový mikrokontrolér z rodiny PIC24F. Tento MCU disponuje 96KB paměti pro data a 256KB pro program.

MCU PIC24F jsou založeny na upravené harvardské architektuře s instrukční sadou typu RISC. Zpracování jedné instrukce trvá 2 periody vstupního hodinového signálu. Jejich maximální taktovací frekvence je 32MHz, maximální počet zpracovaných instrukcí je tak 16MIPS.

Takt lze zvolit ze 4 zdrojů. Mikrokontrolér má zabudovány 2 interní zdroje hodin - rychlý RC oscilátor o frekvenci 8 MHz a nízko příkonový 31kHz oscilátor. Další 2 zdroje hodinových impulzů je možné připojit externě (primární oscilátor OSC a sekundární oscilátor SOSC). Externí hodiny použité v primární jednotce jsou blíže popsány v kapitole 2.1.2.

Pro připojení externích periférií disponuje MCU 52 vstupně výstupními piny. U každého z těchto pinů lze zapnout vyvolání přerušení pro změnu stavu. Na část pinů lze softwarově namapovat některý z jeho vnitřních modulů. Významné pro tuto práci jsou zejména moduly SPI, I2C ADC, a externí přerušení. Kompletní propojení I/O pinů MCU s perifériemi lze vyčíst ze schématu přiloženého v příloze.

Řadič přerušení rozlišuje mezi 7 programově nastavitelnými prioritami. Priorita udává, které přerušení bude vykonáno přednostně. To výrazně usnadňuje implementaci firmware využívajícího zanořující se přerušení.

K měření časových intervalů je MCU vybaven pěti 16bitovými časovači. Časovače 2/3 a 4/5 je možné sloučit do dvou 32bitových. Všechny časovače jsou opatřeny předděličkou snižující frekvenci čítání v poměru 1/1,8,64 a 256.

Spotřeba mikrokontroléru se odvíjí od frekvence a módu, ve kterém se nachází. K dispozici jsou 3 režimy běhu. Jedná se o „run“ mód, „sleep“ mód, a „standby“ mód. Tabulka 1 shrnuje dopad využití jednotlivých módů na běh MCU a zobrazuje spotřebu energie při jejich použití.



režim	spotřeba při 3,3V [ $\mu$ A]	popis
run	800 $\mu$ A/MIPS	zařízení je plně aktivní
standby	22 (32KHz oscilátor, nejnižší možná spotřeba)	zastaveno vykonávání instrukcí, zdroj systémových hodin a periferní moduly zůstávají aktivní, mohou být ale manuálně vypnuty
sleep	20	veškeré prvky, s výjimkou těch, které jsou taktovány sekundárním oscilátorem, jsou vypnuty

Tabulka 1-Režimy běhu mikrokontroléru PIC24FJ256GB206

## 2.1.2 Zdroj hodinových impulzů

V původní verzi primární jednotky byl jako hlavní zdroj hodinových impulzů použit 4MHz krystal. Udávaná odchylka tohoto krystalu byla 30 ppm [3] (parts per milion, odchylka frekvence na 1MHz). Při testování provedeném v původní práci se však ukázalo, že je odchylka ještě větší, a to až 110 ppm. Chyba měření času na 60 sekundách při této hodnotě by mohla být až 6,6ms. Tento krystal navíc ani není teplotně kompenzován, takže se chyba může měnit v závislosti na teplotě okolí. Krystal byl proto nahrazen přesnějším, teplotně kompenzovaným oscilátorem s udávanou chybou 1,5 ppm [4]. Změněna byla také frekvence hodinového signálu ze 4 na 20MHz, což přineslo aditivní výpočetní výkon MCU.

Oscilátor není napájen ihned po spuštění jednotky, MCU musí nejdříve zapnout jeho napájení a až poté jej přepnout jako svůj hlavní hodinový zdroj. Řízení napájení oscilátoru je společné i pro ostatní periférie (displej, bezdrátový modul).

Pro potřeby modulu reálného času na MCU je jednotka opatřena i pomalejším 32,768kHz oscilátorem. Ten běží bez zásahu z MCU.

## 2.1.3 Bezdrátový modul

K bezdrátovému přenosu dat slouží jednotce modul RFM23b od firmy HOPE microelectronic [5]. Moduly jsou dostupné pro přenos dat ve třech frekvenčních pásmech. V této aplikaci byla použita frekvence 868MHz. Podporovány jsou 3 typy modulace: FSK, GFSK a OOK. Maximální vysílací výkon je 13dBm. Přenosovou rychlost lze nastavit v rozsahu od 0,125kb/s po 256kb/s. Modul obsahuje dvojici bufferů o velikosti 64B sloužící pro příjem a odeslání dat.

Odesílání dat je možné provádět 2 způsoby. Jedna možnost je ovládání přenosu manuálně, krok po kroku, z mikrokontroléru. Takové řešení je ale náročné na výkon řídicího MCU a také výrazně ztěžuje návrh a implementaci firmwaru. Vhodnější je využití automatizovaného přenosu dat prostřednictvím FIFO (first in, first out). Při tomto přístupu jsou data automaticky balena do paketů. Obrázek 3 zobrazuje strukturu takového paketu. Kromě dat jsou odesílány i jiné informace. Některé z nich jsou odesílány vždy, jiné jsou volitelné.



Obrázek 3- Struktura paketu RF modulu [5]

## Struktura paketu

Preamble je v paketu povinná. Jedná se o posloupnost střídajících se nul a jedniček. Tato část je zde kvůli tomu, aby byly naslouchající jednotky schopny rozeznat začátek příjmu validních dat od náhodného šumu.

Následující položka v paketu, opět povinná, je synchronizační klíč. Jelikož se v jedné oblasti může nacházet více, vzájemně nesouvisejících, prvků využívajících jedno frekvenční pásmo, mohlo by docházet k příjmu nechtěných informací. A právě k snížení toho rizika slouží synchronizační klíč. Hodnota umístěná v paketu musí být stejná jako hodnota nastavená v jednotce, pro kterou je určen.

Paket může obsahovat také volitelnou, 4B dlouhou, hlavičku. Hlavičku lze nastavit tak, aby filtrovala zprávy na základě hodnoty, kterou nese. Kontrolu je možné zapnout i pouze na část dat, (např. pouze na první byte, další už mohou obsahovat cokoli).

Délka odesílaného paketu může být různá. Proto je přibalena také informace o délce přenášených dat. Za délkou následují samotná data.

Poslední položka paketu je opět volitelná. Jedná se o kontrolní součet dat (CRC), sloužící k odhalení případné chyby při přenosu. V případě neshody CRC je paket zahozen.

Pro zjednodušení obsluhy je RFM23b schopný (např. přijetí validního paketu, úspěšné odeslání paketu) generovat signál přerušení. Další významná funkce, kterou modul nabízí je sledování hladiny RSSI (received signal strength indication). K dispozici jsou i jiné nástroje, jako např. ADC převodník (analog-to-digital converter), teplotní čidlo, nebo wake-up timer. Využít jich lze přes čtyři nastavitelné GPIO piny (general purpose input output).

Spotřeba modulu se odvíjí od režimu ve kterém se právě nachází. Tabulka 2 zobrazuje spotřebu při odesílání, přijímání a 2 hraničních idle stavech. Zobrazen je i čas potřebný k přepnutí do režimu odesílání a přijímání.

činnost	spotřeba	čas potřebný k přepnutí do	
		TX	RX
odesílání (TX, max. výkon)	30 mA	-	200us
přijímání (RX)	18,5 mA	200us	-
idle (tune mode)	8,5 mA	200us	200us
idle (standby mode)	450 nA	800us	800us

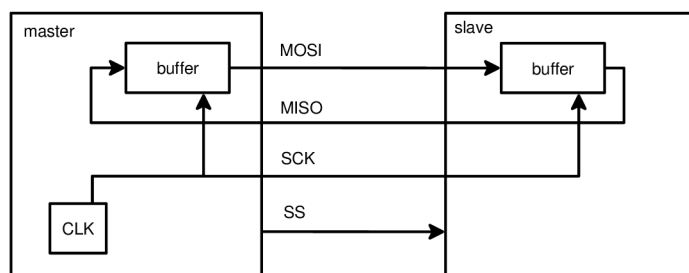
Tabulka 2-Spotřeba RFM23b v různých režimech a doba potřebná k přepnutí do TX a RX

Ovládání modulu je řešeno zápisem hodnot do řídicích registrů modulu přes rozhraní SPI. Kromě SPI rozhraní je k MCU přiveden i pin s přerušením a jeden konfigurovatelný pin GPIO.

## SPI

SPI (Serial Peripheral Interface) je sériová sběrnice sloužící k vzájemnému propojení dvou a více prvků. Přenos po sběrnici je plně duplexní. Složená je ze čtyř vodičů. Jsou to: MISO (Master In, Slave Out), MOSI (Master Out, Slave In), SCK (Serial Clock) a SS (Slave Select). Místo MISO a MOSI se používají i jiná označení. V příložených schématech to jsou SDI (data input) a SDO (data output). Na sběrnici je vždy jedno zařízení, které komunikaci řídí (master) a ostatní zařízení (slave).

Přenos dat je synchronizován hodinovým signálem z řídicího zařízení. Tento signál je veden po vodiči SCK. Zařízení, se kterým se komunikuje, určuje vodič SS. Všechna slave zařízení musí být připojena samostatným SS vodičem. Výběr je proveden nastavením SS do aktivní logické úrovně. Samotný přenos dat je prováděn po vodičích MISO a MOSI. Master i slave vysílají data souběžně v závislosti na SCK. Každou periodu SCK je přenesen 1 bit. K přenesení dat bývá používán 8bitový rotační registr. Obrázek 4 zobrazuje princip komunikace na sběrnici SPI.



Obrázek 4- Princip komunikace na sběrnici SPI

Počet zařízení komunikujících po sběrnici SPI v této práci nepřesahuje počet SPI modulů, které jsou v mikrokontrolérech k dispozici. Proto bude použita pouze verze s jedním zařízením slave.

### 2.1.4 Vstupy

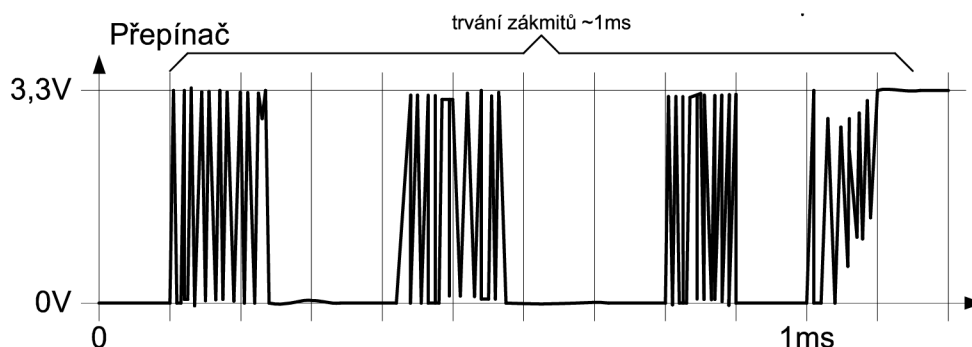
Kromě bezdrátového získávání informací z koncových prvků byla zachována i možnost drátového připojení snímacích prvků. Napřímo, drátově, lze připojit až 5 snímačů, a to na jeden konektor typu XLR a jeden 4-vstupý konektor M12. Důvod použití jednoho konektoru typu XLR je ten, že bývá často používán startovacími zařízeními, jako je startovací pistole. Z pohledu zpracování v MCU se tyto konektory nijak neliší – koncová událost (aktivace senzoru) je indikována uzemněním vodiče (koncové prvky připojené k prim. jednotce mohou být pouze tzv. pasivního typu).

Napájet jednotku je možné přes USB kabel (USB konektor typu B) proudem 500mA nebo také přes standardní DC napájecí vidlici, která umožňuje rychlejší nabíjení proudem až 1A. USB konektor zde není pouze pro účely napájení, jeho prostřednictvím lze jednotku propojit např. s PC. Tato práce se ovšem touto možností nezabývá a proto není USB blíže popsáno.

Ovládání, jak bylo již výše zmíněno, je řešeno 5tíci mechanických tlačítek. Tlačítka a ani vstupy nejsou fyzicky ošetřena (např. dolní propustí) proti možným zákmitům.

## Ošetření vstupů

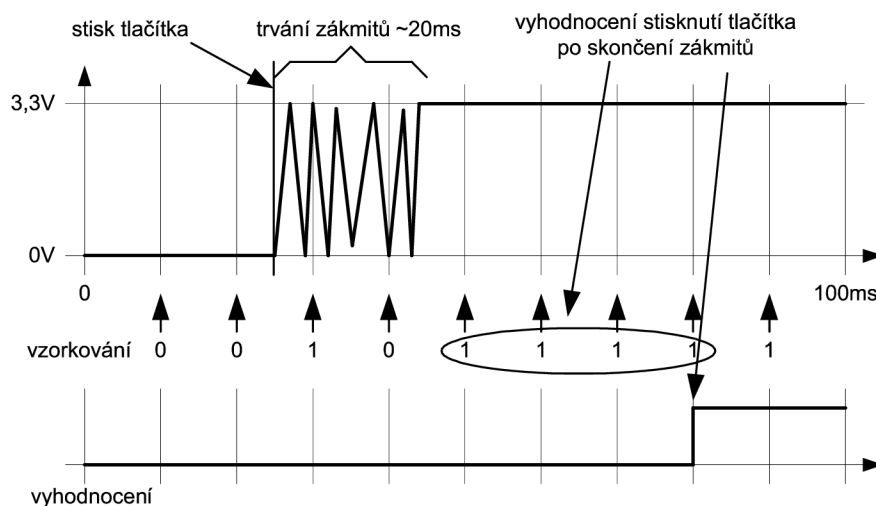
U mechanických tlačítek, ale i jiných spínačů je nutné počítat s tím, že při jejich sepnutí nedochází pouze k jedné změně logické úrovně. Kvůli nedokonalosti jejich provedení dochází většinou k celé řadě logických skoků, k tzv. zákmitům. K zákmitům nedochází jenom při sepnutí tlačítka, stejný efekt má i jeho rozeptnutí. Doba trvání zákmitu není vždy stejná. Ovlivněna je zejména kvalitou provedení snímačů a jejich opotřebením. Udávaná doba trvání zákmitu je většinou pod 10-15ms [13].



Obrázek 5- Průběh signálu při zmáčknutí tlačítka [13]

Kdyby zůstal takovýto vstup neošetřen, mikrokontrolér by vyhodnotil jedno sepnutí snímače jako několik po sobě opakovaných sepnutí. Proto je nutné tyto nechtěné stavy eliminovat. Způsobů jak to provést je několik.

Jedna z možností je kontrola stavu snímače v určitých časových intervalech, v tzv. vzorcích. Jakmile je v několika po sobě jdoucích vzorcích detekována logická úroveň značící sepnutí, je snímač vyhodnocen jako sepnutý. Řešení má však řadu nevýhod. Tou nejzásadnější, pro aplikaci měřící čas na základě vstupních impulzů, je neschopnost určit přesný okamžik sepnutí snímače. Takto získaný čas by byl značně nepřesný.



Obrázek 6-Ochrana proti zákmitům pomocí vzorkování [13]

Další možné řešení je principem podobné prvnímu. Místo vzorkování je použito přerušení vyvolané změnou stavu na vstupu. Jakmile přerušení nastane, je nastaven časovač, který po předem definované době znovu zkontroluje stav snímače. Pokud jsou sepnuty, vykoná se patřičná operace. Problém nepřesného měření času s touto metodou, díky využití přerušení, odpadá. Ideální řešení to

ovšem není. Metoda spoléhá na to, že snímač zůstane po určitou dobu sepnut. U tlačítek obsluhovaných lidskou rukou na to lze spoléhat, ale v případě připojení snímačů z koncových prvků, které by mohly signalizovat sepnutí krátkým impulzem, by nastal problém.

Správného chování lze dosáhnout modifikací předchozí metody. K detekci sepnutí je opět použito přerušení a časovač. Snímač je však nyní vyhodnocen jako sepnutý už při první změně. Základní je eliminován „z necitlivěním“ patřičného vstupu po předpokládaný čas jeho trvání.

## 2.1.5 Slot pro SD kartu

Úložný prostor jednotky lze rozšířit paměťovou kartou SD. Karta je s MCU propojena přes rozhraní SPI. Přivedeny jsou i piny CD (card detect) a WP (write protect).

## 2.1.6 Displej

O zobrazení se stará 4 řádkový alfanumerický LCD displej [6]. Na jeden řádek se vejde 20 znaků. Rozměry jednoho znaku jsou 5x8 bodů. Displej je podsvícený.

Původní verze primární jednotky obsahovala parametry podobný displej. Ten byl ovšem řízen řadičem HD44780 [7], nyníjší displej je ovládán řadičem SSD1803A [8]. Odlišné je i propojení displeje s mikrokontrolérem. Zatímco v původní verzi bylo propojení realizováno přes 4 bitové paralelní rozhraní, aktuální verze je ovládána po sběrnici I2C. Podsvícení je ovládáno zvlášť, samostatným pinem.

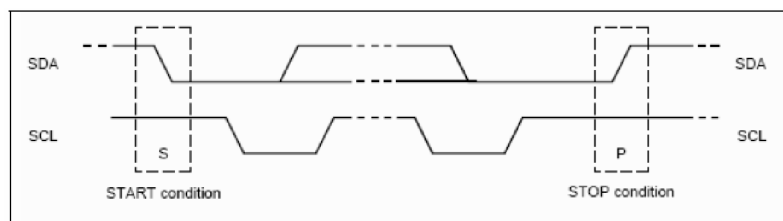
Řadič SSD1803A se, alespoň z pohledu ovládání a funkcionality, od HD44780 příliš neliší. Aktuální obsah displeje je zapisován do paměti DDRAM (Display Data RAM). Vypisované znaky jsou uloženy v paměti CGROM, celkem je na výběr 255 znaků (v SSD1803A lze přepínat mezi 3 sadami znaků s 255 pozicemi, první polovina ve všech sadách je stejná). Na prvních 8 pozic CGROM jsou namapovány znaky z paměti CGRAM, do které lze zapsat uživatelsky definované znaky. Mimo zápisu a čtení do/z těchto pamětí je možné provádět i operace, jako např. vyčištění displeje, rotace znaků na řádku, zobrazení/skrytí kurzoru.

## I2C

I2C, někdy také IIC (Inter-Integrated Circuit) je synchronní sériová sběrnice. Přenos dat je poloduplexní. Komunikace je realizována dvojicí vodičů. Jsou to SDA (Serial Data) a SCK (Serial Clock). Stejně jako na sběrnici SPI i zde vystupují zařízení ve dvou rolích master a slave. Sběrnice je však typu multi-master – všechna propojená zařízení mohou ovládat sběrnici.

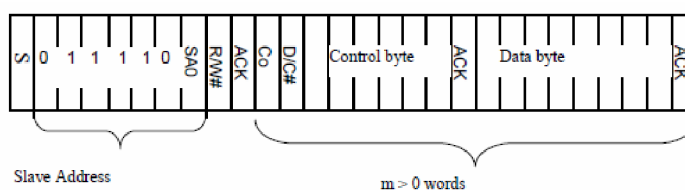
K adresaci se používá 7 bitová, případně i 10 bitová adresa. Teoreticky je možné připojit 128 resp. 1024 zařízení. Adresa zařízení bývá přiřazena výrobcem. Níže je popsána pouze 7 bitová verze komunikace.

Přenos dat je zahájen tzv. start start condition – log. hodnota datového vodiče SDA je nastavena do nuly, SCK je ponechán v log. 1 (defaultně je na obou vodičích log. 1). Po start bitu je odeslána adresa požadovaného zařízení. Další odeslaný bit označuje, zda půjde o zápis nebo čtení (1 – čtení, 0 - zápis). Zařízení, které rozpozná svoji adresu, nastaví na SDA log. 0 (tzv. ACK bit). Ukončení komunikace, stop condition, je provedeno nastavením SCK a následně SDA na vysokou úroveň. Start a stop condition jsou zobrazeny na obrázku 7.



Obrázek 7-I2C zahájení a ukončení komunikace [8]

V případě zápisu jsou po navázání spojení po bytech odesílána data. Každý přenesený byte je příjemcem potvrzen ACK bitem. První odeslaný byte bývá řídicí, udává jak má být s daty naloženo. Jakmile jsou všechna data odeslána, je přenos ukončen.



Obrázek 8-ukázka zápisu dat po I2C [8]

Počátek operace čtení je stejný jako zápis. Nejdříve je zahájena komunikace start bitem, adresou periferie a bitem udávajícím, že se jedná o zápis. Poté je odeslána adresa registru, který chce master číst. Následuje opětovné odeslání start bitu, včetně adresy periferie a bitem, tentokrát už oznamujícím čtení. Zařízení slave nato začne odesílat požadovaná data. Čtení ukončuje master.

## 2.1.7 Napájení

Jednotka je napájena z Li-Ion akumulátoru s kapacitou 2200mAh [9]. Napětí z akumulátoru je zvýšeno step-up měničem MCP1640 na 5V a dále pro potřeby MCU a dalších periférií sníženo a stabilizováno pomocí stabilizátoru MCP1700 na hodnotu 3.3V.

Součástí napájení je i nabíjecí obvod MCP73837 [10]. Nabíjení akumulátoru je ovládáno (nabíjení povoleno/zakázáno) z mikrokontroléru. Defaultně, když je zařízení fyzicky vypnuto, není dobíjení povoleno. Z tohoto důvodu nemá primární jednotka vypínač, který by odpojil MCU od energie. Místo toho je opatřena pouze přepínačem, který oznamuje MCU, že má být „vypnut“. Firmware se musí postarat o vypnutí periférií a převést mikrokontrolér do úsporného režimu.

Z nabíjecího obvodu jsou dále pro potřeby řízená nabíjení přivedeny vstupy oznamující: „obvod má dostatek energie k dobíjení“, „baterie se nabíjí“, „nabíjení dokončeno“; a také aktuální napětí z baterie (napětí je sníženo děličem na polovinu, aby nedošlo k poškození MCU).

Ovládání nabíjení z MCU bylo zvoleno kvůli zvýšení životnosti akumulátoru. Li-Ion baterie by neměly být nabíjeny na 100% své kapacity. MCP73837 bez vnějšího zásahu dobíjí baterii do maximální kapacity. Proto byla volena možnost manuálního ovládání nabíjení, které je řešeno odpojením jednoho z nastavujících odporů nabíjecího obvodu. Tento stav přivede bateriový kontrolér do stand-by režimu.

Při nabíjení Li-Ion akumulátorů by jejich napětí nemělo překročit hodnotu 4,2V [14]. Dobíjení by také nemělo být opakovaně spouštěno při první detekci neúplně nabitě baterie, ale mělo by být spouštěno, až když překročí určitou mez (např. nabito z 90%).

## 2.1.8 Spotřeba

Spotřeba jednotlivých prvků je shrnuta v tabulce 3. Hodnoty pro MCU a bezdrátový modul jsou udány v tabulce 1 a v tabulce 2.

Přibližná teoretická výdrž jednotky při uvedených hodnotách, MCU run mode, RFM RX, bez SD karty s 2200mAh akumulátorem je 20h.

prvek	spotřeba [mA]
displej	1
podsvícení displeje	48
10MHz oscilátor	6
SD karta SanDisk [15] (zápis)	100

Tabulka 3-Spotřeba prvků v primární jednotce

## 2.2 Sekundární jednotka

Principem se nová sekundární jednotka neliší od té staré. Její provedení a také využití z hlediska sportovních aktivit jsou však značně odlišné. Nová jednotka je navržena speciálně pro potřeby požárního útoku, což je jedna z disciplín požárního sportu [22]. Jednotka je určena k upevnění na požární terč, kde detekuje sepnutí čidla a zároveň slouží jako signalizace. Disciplína vyžaduje 2 koncové prvky. Obrázek 9 ukazuje způsob použití v daném sportu.



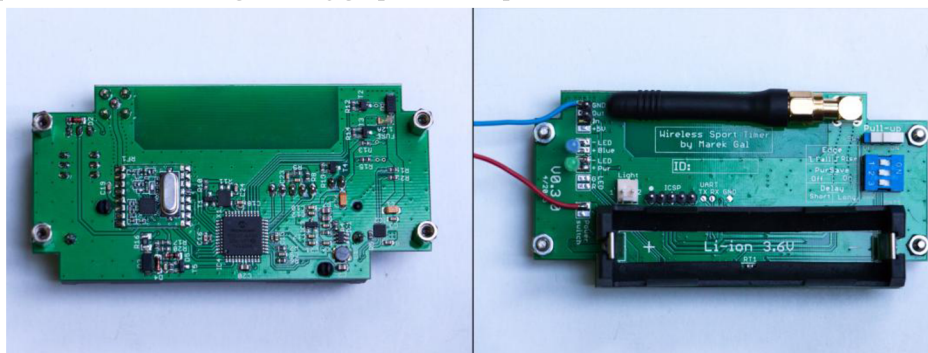
Obrázek 9- Koncové zařízení pro požární sport

Jednotka je vsazena do uzavřeného, vodotěsného pouzdra. K signalizaci je přední panel opatřen sadou červených led diod. Připojit lze pouze jeden snímací prvek. Obrázek 10 ukazuje srovnání hardwarového provedení nové a staré jednotky.



Obrázek 10-Srovnání staré (vlevo) a nové (vpravo) sekundární jednotky

Jednotka je vybavena mikrokontrolérem, bezdrátovým modulem, konektory k propojení čidla, měřicí jednotky a napájení, několika konfiguračními spínači a sadou signalizačních led diod. Obrázek 11 ukazuje vnitřek sekundární jednotky. Část schématu zobrazující mikrokontrolér a k němu propojené periferie sekundární jednotky je přiložen v příloze.



Obrázek 11-Deska sekundární jednotky

## 2.2.1 Mikrokontrolér

Mikrokontrolér použitý v této jednotce je, stejně jako v primární jednotce, z 16bitové řady PIC24F od firmy Microchip. Konkrétně se jedná o kontrolér PIC24fj64GA004 [20]. Celkově jsou parametry tohoto MCU podobné parametrům mikrokontroléru použitému v primární jednotce. Zásadní odlišnosti jsou ve velikosti paměti - maximální velikost programu je 64kB, RAM paměť pojme maximálně 8kB dat a v počtu pinů (44).

## 2.2.2 Zdroj hodinových impulzů

K taktování CPU je použit externí 20MHz oscilátor s přesností 30 ppm [21]. I když není sekundární jednotka určena k provádění výpočetně náročných operací vyžadujících relativně vysoký výkon a tudíž by mohl být použit pomalejší zdroj hodin, byl oscilátor zvolen tak, aby jeho frekvence odpovídala oscilátoru v primární jednotce. Důvodem je zvolený způsob měření času v původní práci. Způsob je blíže popsán v kapitole 3.1.

## 2.2.3 Vstupy

Pro připojení snímacího čidla je jednotka vybavena standardním průmyslovým konektorem M12. Vstupní modul je navržen tak, aby přes něj bylo možné připojit jak pasivní čidla, která jsou

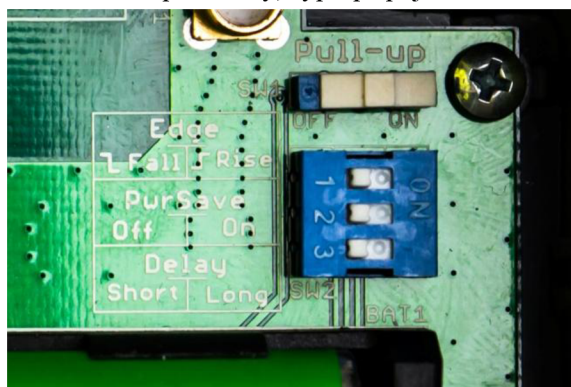


aktivována uzemněním, tak také aktivní čidla. Vstup musí být podobně jako v primární jednotce ošetřen proti zákmitům.

Aby program dokázal správně určit, jaká hrana signálu má být považována za aktivní, je jednotka opatřena dvou-stavovým přepínačem.

Mimo tento přepínač jsou zde ještě další dva. Jejich funkce jsou definovány popiskem přímo na desce plošných spojů a program se jimi proto bude řídit. Jedná se o volbu doby minimálního zpoždění mezi impulzy indikujícími validní sepnutí čidla (0,5s – 5s) a o způsob signalizace sepnutí čidla signalizačními diodami (při zapnutí PwrSave by měla jednotka začít po určité době blikat).

Na obrázku 12 je kromě zmíněných přepínačů vidět ještě jeden přepínač. Ten slouží k hardwarovému přepnutí (pro firmware nepodstatný) typu připojeného snímače (aktivní/pasivní).



Obrázek 12-Nastavení sekundární jednotky

## 2.2.4 Výstupy

Jednotka je vybavena i jedním konektorem XLR k přímé signalizaci sepnutí měřicího čidla hlavnímu měřicímu zařízení (primární jednotce, nebo i jinému kompatibilnímu prvku). Sepnutí je provedeno uzemněním vodiče. O to se stará MCU sepnutím N-MOSFET tranzistoru.

## 2.2.5 Světelná signalizace

Jak bylo výše zmíněno, je na předním panelu umístěna sada LED diod určených k signalizaci sepnutí propojeného čidla.

Na jednotce jsou ještě další 3 samostatné LED diody. Jedna je ovládána automaticky hardwarem (signalizuje dobíjení), zbylé dvě jsou volně k dispozici mikrokontroléru.

## 2.2.6 Napájení

Napájení je, stejně jako u primární jednotky, řešeno Li-Ion akumulátorem s kapacitou 2200 mAh. Jednotka také obsahuje nabíjecí obvod, ten zde ale není řízen mikrokontrolérem. Nabíjení řídí sám obvod. K MCU je, pouze informačně, přivedeno aktuální napětí na baterii a stavové výstupy o průběhu napájení (podobně jako u primární jednotky).

## 2.2.7 Spotřeba

Spotřeba bezdrátového modulu a MCU je stejná jako v primární jednotce (MCU - tabulka 1, RFM - tabulka 2). Další významné energii odebírající prvky shrnuje Tabulka 4.

prvek	spotřeba [mA]
přední signalizační LED panel	230
LED dioda (3x)	10
10MHz oscilátor	5

**Tabulka 4-Spotřeba prvků v sekundární jednotce**

Přibližná teoretická výdrž jednotky při uvedených hodnotách (MCU run mode, RFM RX) s rozsvíceným předním panelem a s 2200mAh akumulátorem je 5,5h.

## 2.3 Vývojové prostředky

Microchip poskytuje ke svým zařízením velkou škálu vývojových nástrojů. Programovat lze buďto přímo v jazyce symbolických instrukcí, nebo je možné využít některý z kompilátorů jazyka C. Pro mikrokontroléry PIC24F je od Microchipu určen překladač XC16 [16]. Jeho základní verze je dostupná zdarma ke stažení a je možné ji použít pro libovolné účely. Kromě volně dostupné verze jsou k dispozici i placené verze „standard“ a „pro“. Od základní verze se liší pouze mírou optimalizace výsledného kódu. Jinak není základní verze nijak limitována. XC16 je založen na překladači GCC a vyhovuje standardu ANSI C. Dostupné jsou i překladače třetích stran a to nejen pro jazyk C ale i pro jiné programovací jazyky. Tyto prostředky jsou však často pouze placené, nebo je jejich neplacená verze výrazně limitována.

Přeložený program je do paměti MCU nahráván přes rozhraní ICSP (In-Circuit Serial Programming). K nahrání programu je použito zařízení PICKIT3 [18] (obrázek 13). K počítači je PICKIT propojen USB kabelem. Jedná se o oficiální programátor od Microchipu. Kromě zápisu a čtení paměti poskytuje i podporu pro ladění programu za běhu.



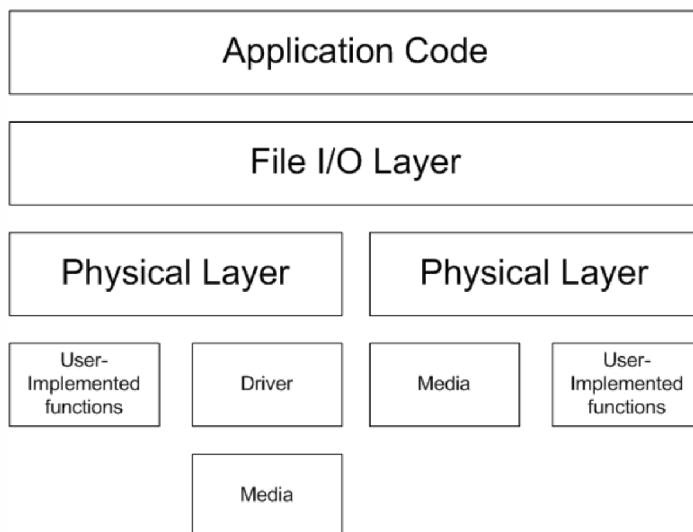
**Obrázek 13-Programovací a ladící nástroj PICKIT3 [18]**

Nejen pro zjednodušení obsluhy těchto nástrojů poskytuje Microchip také integrované vývojové prostředí MPLAB X IDE. Jedná se o prostředí vybudované na vývojovém prostředí NetBeans IDE od firmy Oracle. V kombinaci s programátorem poskytuje přehledné rozhraní k překladu, naprogramování i debugování programu (breakpointy, krokování instrukcí, čtení obsahu paměti ...).

Microchip také nabízí volně ke stažení řadu hotových řešení v podobě knihoven pro jazyk C. Zahrnutý jsou mimo jiné USB framework, grafická knihovna, TCP/IP stack a také knihovna pro práci s externími úložnými zařízeními (File I/O). Právě tato knihovna je významná pro tuto práci.

### 2.3.1 File I/O knihovna

Řešení od firmy Microchip umožňuje ukládání dat v souborovém systému FAT (FAT12, FAT16, FAT32) na různé druhy paměťových médií. Přímou podporována je i SD karta s přístupem po rozhraní SPI. Souborová knihovna je seskládána z několika vrstev. Struktura je zobrazena na obrázku 14.



Obrázek 14-Struktura souborové knihovny

Application Code vrstva představuje samotný uživatelský program využívající File I/O vrstvy.

File I/O vrstva realizuje API pro práci se soubory (čtení souboru, zápis apod.) a složkami ve FAT systému. Na výběr jsou dvě verze uživatelských rozhraní. Rozhraní pro práci s krátkými názvy souborů (dále jen KN) a rozhraní pro práci s dlouhými názvy souborů (DN). Kromě rozdílu vyplívajícího z názvu se liší i řadou dalších věcí. API DN poskytuje více možností, ale je náročnější na zdroje mikrokontroléru. Maximální délka názvu souboru pro KN je 8 znaků plus „.“ a 3 znaky přípony souboru. Použít lze pouze alfanumerické znaky a „! # \$ % & ' ( ) - @ ^ \_ ` { } ~“. Maximální délka pro DN je 255 znaků a jejich výběr je téměř neomezen. DN navíc umožňuje naformátování média.

File I/O využívá služeb fyzické vrstvy. Ta už přímo, nebo přes ovladač přistupuje k samotnému médiu. V tomto případě k SD kartě.

Od uživatele vyžaduje knihovna implementaci několik funkcí. Fyzická vrstva požaduje funkce pro práci s piny, na kterých se nachází médium a File I/O vyžaduje funkci dodávající časová razítka pro datum uložení souboru.

## 2.4 Shrnutí požadavků na software

Bezdrátový modul použitý v obou jednotkách umožňuje automatizované odesílání zpráv, nabízí i určitou ochranu proti chybám v podobě CRC součtu. To však nezaručuje spolehlivý přenos dat. O to se musí postarat programové vybavení. Protože je k jednomu měření potřeba komunikovat s více sekundárními jednotkami, musí být bezdrátová komunikace schopná získávat data z více sekundárních jednotek. Aktuálně použité sekundární jednotky mají pevně daný účel, v budoucnu by

se ale mohly rozrůst o další typy. Aby je mohla primární jednotka využívat k měření, musí mít možnost dynamického výběru jednotek.

Dále musí být v obou jednotkách ošetřeny vstupy z tlačítek i pro drátové propojení. Software sekundární jednotky musí být schopný rozlišit mezi aktivní sestupnou a náběžnou hranou na základě nastavení přepínačů.

V primární jednotce je potřeba realizovat uživatelské rozhraní celého systému. K tomu je jednotka vybavena 4 řádkovým displejem a sadou tlačítek. Výsledné časy mohou být získány z bezdrátově, ale i z kabelového připojení. Rozhraní by s těmito možnostmi mělo počítat. S uživatelským rozhraním je spojeno i využití SD karty k případnému zaznamenání naměřených časů. Pro práci s SD kartou existují volně dostupná hotová řešení, je však nutné také určit vhodný formát pro zápis časů.

Primární jednotka také nemá defaultně povolené dobíjení akumulátoru. Proto musí být realizována detekce zdroje napájení a povolení dobíjení. Jednotce také chybí centrální vypínač, který by ji odpojil od napájení. Místo toho má přepínač připojený v mikrokontroléru. Software tedy musí zajistit přechod jednotky do režimu šetřícího energii.

## 3 Návrh programového vybavení

Kapitola popisuje návrh programového vybavení pro časoměrný systém. Návrh je zaměřen na přesné měření času, spolehlivou bezdrátovou komunikaci, uživatelské rozhraní a snížení spotřeby primární i sekundární jednotky.

### 3.1 Měření času

Zásadním požadavkem na časoměrný systém je přesnost měřeného času. Při použití drátového připojení měřících senzoru lze jednoduše změřit čas mezi přichozími impulzy. Zpoždění způsobené přenosem po vodiči je téměř nulové. Tento princip však u bezdrátového přenosu použít nelze. Odesílání dat nějakou dobu trvá, takže by byl sejmутý čas na měřícím prvku zpožděný. Za předpokladu konstantní doby přenosu, by šel problém odstranit odečtením této doby od zaznamenané hodnoty. Zpoždění přenosu však není jediným problémem. Při přenosu dat hrozí riziko poškození zprávy. I kdyby zdrojové zařízení dokázalo ztrátu detekovat a znovu zprávu odeslat, výsledný čas by byl nesprávný. Tento způsob je tedy značně nespolehlivý.

Možný způsob řešení je provedení časové synchronizace mezi jednotkami. V případě, že na všech jednotkách běží stejný čas, stačí koncové jednotce zaznamenat čas sepnutí senzoru a ten odeslat řídicí jednotce. Řídicí jednotka získá výsledný čas prostým odečtením zaznamenaného ukončovacího a startovacího času. Měření již není závislé na okamžiku přijetí zprávy a jejich opoždění nepředstavuje problém. Tento způsob byl úspěšně, s uspokojivými výsledky, odzkoušen v původní verzi časoměry a proto není důvod jej zde měnit.

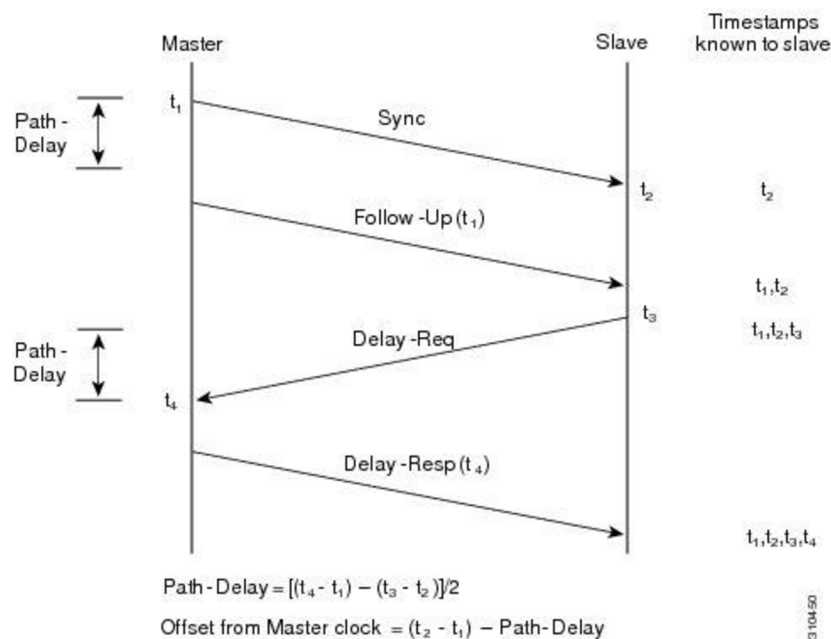
#### 3.1.1 Časová synchronizace

Aby bylo možné využít výše zmíněný způsob měření, je nutné provést synchronizaci času co možná nejpřesněji.

Nejjednodušší způsob synchronizace je prosté odeslání aktuálního času z primární jednotky jednotce sekundární. Opět zde ale hraje velkou roli zpoždění způsobené přenosem. I kdyby bylo zpoždění konstantní a dalo se odečíst od získaného času, byl by čas zneřádněn už jeho samotným přepočtem.

Vhodnější je synchronizace popsána standardem IEEE 1588 [11] (jinak také PTP - precision time protocol). Tento protokol byl navržen speciálně pro aplikace vyžadující vysokou časovou přesnost. Udávaná přesnost, které je možné běžně dosáhnout je v jednotkách mikrosekund (ve specializovaných zařízeních i přesnější). Stejně jako princip měření vychází i způsob synchronizace z původní práce.

Synchronizace probíhá v několika krocích. Nejdříve je odeslán aktuální čas ( $T_1$ ) ze zdrojové jednotky (master) cílové jednotce (slave). Slave si v okamžiku přijetí času  $T_1$  zaznamená svůj čas ( $T_2$ ) a následně odešle další čas ( $T_3$ ) jednotce master. Ta, obdobně jako předtím slave, si zaznamená čas přijetí  $T_3$  ( $T_4$ ) a odešle jej zpět jednotce slave. Princip PTP je zobrazen na obrázku 15.



Obrázek 15- časová synchronizace PTP [11]

Slave jednotka má nyní k dispozici 4 časy. Z nich je schopna spočítat posunutí svého času oproti času v zařízení master. Tento výpočet se provede podle vzorce 1. Získanou hodnotu už jenom přičte ke svému času.

$$\Delta t = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}$$

K dosažení maximální přesnosti je důležité sejmout časová razítka co nejbližší samotnému přenosu dat. Spoléhá se také na konstantní dobu odeslání. Samotný výpočet ani doba, kdy se slave jednotka rozhodne provést úpravu svého času na synchronizaci nemají vliv.

Při synchronizaci je nutné počítat i s nepřesností oscilátorů. Hodnoty na synchronizovaných jednotkách se mohou časem rozcházet. Proto by měl být proces synchronizace průběžně opakován. Oscilátor s přesností 2,5 ppm v primární jednotce a oscilátory v sekundární jednotce s přesností 30 ppm se teoreticky, v extrémním případě, mohou rozcházet o 32,5 ppm (chyba  $32,5 \times 10^{-6}$ s – 32,5μs každou sekundu). Na minutě vznikne chyba 1,195ms. Pro dosažení přesnosti 1ms by se měla synchronizace obnovovat alespoň jednou za 30s.

## 3.2 Bezdrátová komunikace

Bezdrátová komunikace musí kromě časové synchronizace a přenosu koncových časů umožnit i dynamické využití sekundárních jednotek. K měření by měla primární jednotka moci použít libovolný počet sekundárních jednotek.

### 3.2.1 Adresace

Aby mohly komunikovat více než dvě jednotky, je potřeba ke každé zprávě přidat identifikátor příjemce i odesílatele. Všechny jednotky tak musí být opatřeny jedinečným ID. 16 bitů na identifikátor s možností vytvořit až 65536 zařízení by mělo být více než dostatečné. Kromě

identifikátoru jednotek musí kvůli možnosti detekce jednotek existovat i jeden identifikátor společný pro všechny (broadcastová adresa). Typicky se používá poslední hodnota z rozsahu.

### 3.2.2 Dynamické využití jednotek

Jednotky nesmí být napevno spárovány. Proto musí existovat prostředek, kterým jednotky zjistí, s kým mají komunikovat. Od sekundární jednotky se v jeden okamžik očekává komunikace pouze s jednou řídicí jednotkou. Stačí jí tedy vyčkat, až bude oslovena primární jednotkou a poté adresovat všechny zprávy právě jí. Aby ovšem primární jednotka věděla, které jednotky jsou dostupné, musí sekundární jednotky oznamovat svoji přítomnost. Ideálně by to měly vykonávat až na vyžádání.

Protože mohou být v budoucnu přidány nové druhy sekundárních jednotek (více snímacích čidel, ale i úplně nové druhy jako např. externí LED displej), měly by kromě svého ID oznamovat také svůj typ a dodatečné informace spojené s typem (počet připojených měřicích čidel).

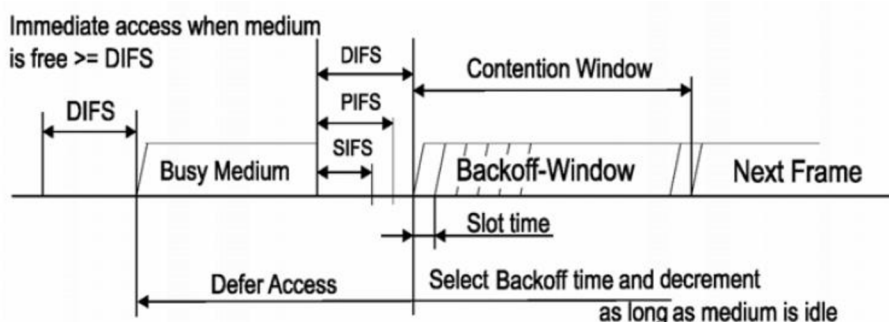
Sekundární jednotky by také neměly zůstat spárované napořád. Mohou být uvolněny na vyžádání nebo se po určitém čase sami uvolnit. Kdyby totiž byla primární jednotka nečekaně vypnuta, musely by být restartovány i sekundární jednotky.

### 3.2.3 Přístup k médiu

Kdyby mohly všechny prvky odesílat svá data v libovolný okamžik, bez ohledu na to zda již nějaká komunikace neprobíhá, mohlo by docházet k nežádoucím kolizím a tím ke ztrátě dat. I kdyby jednotky dokázaly detekovat neúspěšný přenos a data by odeslali znovu, pravděpodobně by došlo k další kolizi. Určité zlepšení by přineslo náhodného zpoždění před pokusem o znovuodeslání, ale i tak by v případě přístupu většího množství jednotek stále hrozilo úplné zablokování přenosu dat. Navíc by tento přístup značně zneprůjemnil návrh komunikace na vyšší vrstvě. Ta by musela počítat s každou eventualitou (např. příjem dat během libovolného stavu probíhající synchronizace).

Možností jak přistupovat k médiu je celá řada. Jedná se např. o metodu „polling“, při které smí vysílat pouze zařízení, které k tomu bylo vyzváno primární jednotkou. Primární jednotka střídavě vyzývá všechny účastníky. Další možná metoda je použití časového multiplexu TDMA (time division multiple access). Každá jednotka může vysílat v přesně vymezeném čase. Metoda je vhodná v sítích kde se přenáší velké objemy dat. Obě metody navíc počítají s tím, že v daném přenosovém pásmu není jiný vysílací prvek. Výhodnější je použití metody aktivního vyhýbání se kolizím CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance).

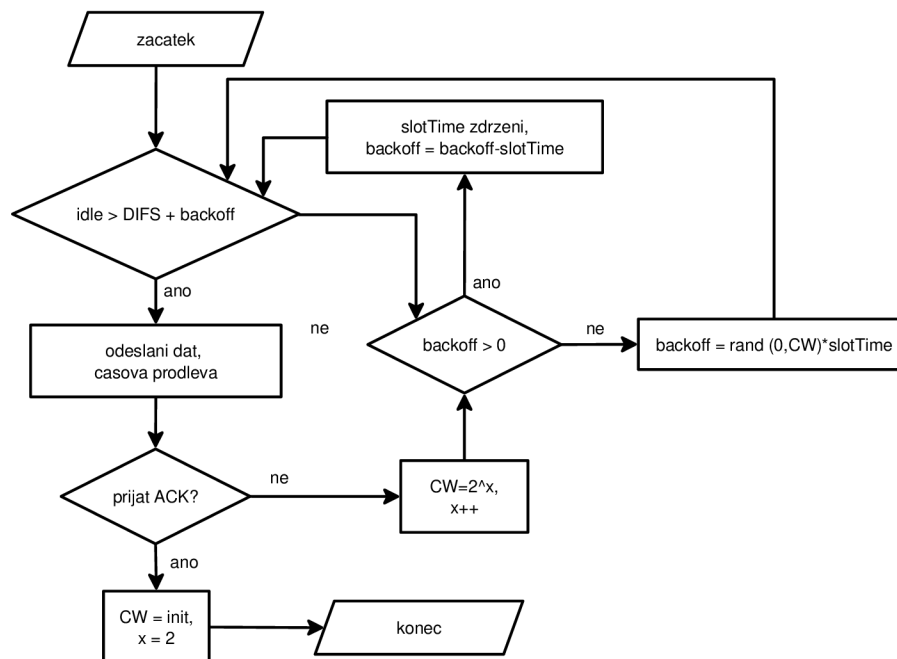
Princip fungování CSMA/CA je čerpán ze standardu 802.11(WiFi) [12]. MAC standard 802.11 definuje základní provedení CSMA/CA označenou jako DCF (Distribute Coordination Function) a její nadstavby poskytující různou úroveň QoS (Quality of services) a jiná rozšíření. Pro tento systém je podstatná pouze základní verze.



Obrázek 16-Základní přístupová metoda [12]

Když chce zařízení využívající CSMA/CA začít vysílat, musí nejprve po určitý čas (DIFS – DCF Interframe Space) naslouchat zda právě neprobíhá jiný přenos. Pokud zůstalo médium nepoužité, může jednotka zahájit přenos. Jinak jednotka čeká na ukončení probíhajícího přenosu. Po čekání musí znovu, po čas DIFS, sledovat médium. Tentokrát je ale před samotné odeslání přidáno náhodné zpoždění (backoff). Zpoždění je rozdělenou časových slotů a jeho maximální velikost omezuje CW (contention window). CW začíná na nízké hodnotě a s každým neúspěšným odesláním (viz níže) se exponenciálně zvyšuje (do určité hranice). Zpoždění je potřeba pro případ kdy 2 a více stanic čekají na ukončení přenosu.

Tento proces do značné míry snižuje počet kolizi, úplně je ale nevylučuje. Proto jsou odesílané zprávy ještě pozitivně potvrzovány (ACK). Jestliže po odeslání dat nepříjde ACK, jsou data přeposlána. Přeposlání je opět zdrženo o náhodný čas. Algoritmus znázorňuje obrázek 17.



Obrázek 17- použitý algoritmus k odeslání dat

Při potvrzování přijatých zpráv se stav média nesleduje, ACK je odeslán jen s malým zpožděním SIFS (Short Interframe Space).

Hodnoty jsou definovány následovně:

- slotTime = doba potřebná k přenosu jednoho rámce
- DIFS = SIFS + 2 × slotTime
- SIFS = doba potřebná ke zpracování a odeslání rámce
- backoff = random(0, CW) × slotTime

### 3.3 Uživatelské rozhraní

Pojem uživatelské rozhraní v sobě nezahrnuje jenom zobrazení dat a ovládání jednotky, ale také vhodné nabídnutí všech funkcí, které systém nabízí.



### 3.3.1 Zobrazení a ovládání

Čtyř řádkový displej neposkytuje příliš volnosti ke způsobu zobrazení a ovládání. Vhodné je navrzení do podoby stromově strukturovaného menu. Na každý řádek se vejde dostatečný počet znaků k vypsání právě jedné položky. Pro posun v menu nahoru a dolů se pro intuitivní ovládání hodí vertikálně umístěná tlačítka. K výběru položky pravé tlačítko a návrat o úroveň výš levé tlačítko.

V časomíře je potřeba provádět také různá nastavení. Přímý vstup od uživatele, např. v podobě zadávání číslic nebo znaků je s danými tlačítky nepraktický. Lepší je výběr z předdefinovaného seznamu, ve kterém uživatel vybírá listováním mezi položkami. Návratové tlačítko by mělo zůstat zachováno, zbývá tak pravé a prostřední tlačítko (další/předchozí položka).

Podstatný je i formát zobrazovaných časů. Standardně je používán formát „hh:mm:ss:ee“ (hodiny:minuty:sekundy:setiny) a není důvod jej měnit. Způsob výpisu měření časů je popsán dále.

### 3.3.2 Konfigurace měření

Navržený způsob měření časů a bezdrátové komunikace v kombinaci s drátovou variantou měření umožňuje využití velkého množství senzorů. Jednotlivé senzory nemají pevně přiřazenou funkci a je tedy možné je libovolně používat. Sekundární jednotka může sloužit jako koncový ale i jako startovací prvek. Systém je také možné využít bez snímačů, jenom s primární jednotkou, při čemž je čas stopován tlačítky (jako klasické stopky). Kromě měření více časů (drah) najednou je v různých sportech využíváno i stopování mezičasů, což by měl systém také umožňovat. Při stopování mezičasů bývá využívána i možnost, kdy je na jeden vstup zaznamenáno více časů (např. klasické stopky). U měřicích senzorů, kde může dojít k nežádoucímu vícenásobnému protnutí čidla by mohl nastat nežádoucí efekt. Proto by měly být tyto mezičasy odlišeny.

Aby byl systém opravdu univerzálně použitelný, měl by uživateli nabízet relativní volnost v nastavení měření. Dostatečně flexibilní řešení je přiřazení jednotlivých senzorů (drátových, bezdrátových a tlačítek) k měřeným časům s tím, že je také konfigurovatelný počet měřených času a funkce těchto časů (start, stop, mezičas a kumulativní mezičas).

Někdy se může stát, že koncové čidlo nelze protnout a tím zastavit čas. Časy by tedy měly jít zastavit i z primární jednotky. Aby nedocházelo k stopnutí měření omylem, nesmí být tato možnost dostupná přímo ze zobrazení měření, ale spíše jako volba v menu. Takto zastavené časy ovšem nejsou validní a měly by být zvlášť označeny. Buď jako čas s hodnotou 0, nebo lépe např. pomlčkami místo číslic.

Pro přiřazení je potřeba mít všechna čidla jednoznačně identifikována. Využit lze ID zařízení z bezdrátové komunikace a číselné označení senzoru. Primární jednotka může být označena zvlášť, textově (tlačítko a přímé vstupy).

Dynamická správa senzorů vyžaduje i dynamický výpis časů. Prosté seřazení podle číselného označení času by mělo být dostatečně přehledné.

### 3.3.3 Stavové informace a signalizace

Obě jednotky by měly informovat uživatele o svém stavu, minimálně o tom zda jsou zapnuty a jestli nepotřebují dobít baterii.

U primární jednotky je zapnutý stav zřejmý z displeje. K poskytnutí dalších informací je vyhrazen první řádek displeje. Na něm se zobrazuje stav energie na akumulátoru, aktuální poloha v menu a informace o přítomnosti SD karty.

Základní stav sekundární jednotky udávají LED diody. Zelená LED dioda určuje, zda je zařízení zapnuté. Její rozblikání značí nízký stav na baterii. Další podstatná informace, která lze signalizovat LED diodou je připravenost jednotky k měření (jednotka je synchronizována s primární). Sepnutí připojeného čidla signalizuje přední panel, který je po této akci na určitou dobu rozsvícen. Panel by mohl být na chvíli rozsvícen i po spárování s primární jednotkou. Uživatel si tak nebude muset pamatovat číselné označení jednotky.

### 3.3.4 Využití SD karty

I když jsou naměřené časy, obzvlášť na trénincích, často pouze orientačně zkontrolovány, někdy se hodí mít je zaznamenány k pozdějšímu využití. Tomu by měla napomocť možnost logovat změřené časy na SD kartu. K jejich prohlížení, nebo snad i k editaci se zařízení moc nehodí. Kromě nepohodlí spojeného s 4 řádkovým displejem by mikrokontroléru nemusela pro otevření větších souborů stačit paměť.

Aby mohly být záznamy čitelné z PC, měly by být ukládány do souborového systému FAT, který je u SD karet běžně používán. Formát ukládaných časů by neměl být příliš složitý. Jak kvůli požadavkům na software a výkon MCU, tak také kvůli snadnému porozumění dat při otevření textovém editorem. Zachována by však měla být i možnost editace tabulkovým procesorem, jakým je např. MS Excel. Těmto požadavkům vyhovuje formát CSV (Comma-separated values). Jak vyplývá z názvu, ukládané položky jsou odděleny čárkami (znak „,“). Jednotlivé záznamy jsou odděleny znakem konce řádku.

V souboru by měl být uložen změřený čas (stejný formát jako při výpisu na displej), informace identifikující jeho původ (ID zařízení a sensor), označení v rámci primární jednotky a typ času (mezičas/koncový čas). Převedení do formátu CSV by byl záznam uspořádán např. takto: „označení v prim. j., ID zařízení, sensor, čas, typ\r\n“ (\r\n – nový řádek). Tabulka 5 ukazuje příklad vytvořeného CSV souboru se dvěma provedenými měřeními.

textový editor	tabulkový procesor (MS Excel)				
01, 123, 2, 00:00:22:34, k	1	123	2	00:00:22:34	k
02, 022, 1, 00:00:30:01, k	2	22	1	00:00:30:01	k
03, 022, 2, 00:00:08:49, m	3	22	2	00:00:08:49	m
01, 123, 2, 00:00:18:98, k	1	123	2	00:00:18:98	k
02, 022, 1, 00:00:21:43, k	2	22	1	00:00:21:43	k
03, 022, 2, 00:00:07:22, m	3	22	2	00:00:07:22	m

**Tabulka 5-Příklad ukládaného CSV souboru, vlevo textové zobrazení, vpravo zobrazení v MS Excel**

Název souboru s časy může být datum dne jejich pořízení (např. měsíc\_den\_rok.CSV). Soubor by měl být umístěn na viditelné místo, kořenový adresář souborového systému FAT není nejvhodnější, lepší je vytvoření nové složky s názvem jasně oznamujícím k jakému účelu byla vytvořena.

## 3.4 Snížení spotřeby

Samotná volba hardwarový komponent obou jednotek byla provedena s ohledem na co nejnižší spotřebu. Výdrž je ale možné ještě prodloužit vhodným návrhem softwaru.

### 3.4.1 Primární jednotka

Jak vyplývá z kapitoly 2.1.8, nejvíce energie z celé jednotky odebírají operace s SD kartou. Pro snížení spotřeby je proto vhodné neprovádět zápis času ihned po jeho zastavení, ale počkat až jsou zastaveny všechny běžící časy.

Prvek s druhou největší spotřebou je podsvícení displeje. Radikální způsob jak tuto spotřebu snížit je nechat podsvícení vypnuté. Tím by se ovšem za určitých podmínek drasticky snížila jeho čitelnost. Lepší varianta je snížení jasu použitím PWM (pulse width modulation). Ideální je ponechání této volby samotnému uživateli prostřednictvím variabilního nastavení PWM z menu. Další možná úspora je jeho úplné vypnutí při delší nečinnosti jednotky (např. po 30s), nebo také nastavitelně na základě uživatelského přání.

Nezanedbatelný odběr má také bezdrátový modul. Největší odběr z jeho režimů má odesílání. Navržená bezdrátová komunikace odesílá pouze ty nejnnutnější informace potřebné k měření. Určitou úsporu by přineslo sloučení datových zpráv do jedné větší. Měřený čas by nebyl ovlivněn, negativně ovlivněna by ale byla uživatelská přívětivost. Zobrazený běžící čas na primární jednotce by byl zastaven s viditelným zpožděním (čas by nejprve překročil výslednou hodnotu a následně by byl „vrácen“). Snížení odběru příjmu dat, i když není extrémně velký, by také přinesla zvýšení výdrže. Možného snížení lze dosáhnout přenosem dat pouze v určitých časech. Mezi nimi by byly všechny moduly uspány. Tato metoda má ovšem stejný problém jako kumulované odesílání – velká latence. Proto je bezdrátový přenos ponechán bez dalších zásahů. V případě, že by chtěl uživatel používat drátovou verzi komunikace, nebo samostatnou primární jednotku (stopky) by mělo být možné RFM modul uspat manuálně, volbou v menu.

Samostatné omezení ostatních prvků by na spotřebu velký přínos nemělo. Uspání celé jednotky by se však projevilo. Aby mělo uspání nějaký účinek, musel by být zastaven také hlavní čítač času. Časová synchronizace je ale závislá na jeho kontinuálním běhu. Větší výpadek by také znamenal ztrátu spárovaných sekundárních jednotek.

### 3.4.2 Sekundární jednotka

Na výdrž sekundární jednotky má největší vliv spotřeba předního LED panelu. Ten ale slouží pouze ke krátkodobé signalizaci sepnutí čidla. Navíc je na jednotce přepínač, který je určen k dalšímu snížení odběru (rozblíknání panelu). V případě potřeby lze snížit jas PWM modulem.

Jinak jednotka, kromě bezdrátového modulu popsáno výše, nemá další významný energii konzumující prvek.

## 4 Implementace

Kapitola shrnuje provedenou implementaci. Jednotlivé části jsou rozděleny do samostatných celků. Způsob měření času a velká část bezdrátové komunikace jsou společné pro obě jednotky. Další části jsou popsány odděleně pro každou z jednotek.

### 4.1 Měření času

K měření času je v obou jednotkách použit jeden z 16 bitových časovačů. Vstupní frekvence 10MHz je předděličkou snížena 64x na hodnotu 156250Hz. Při nastavení komparačního registru časovače na hodnotu 15624 (čítání začíná na 0) dojde k vyvolání přerušení přesně za 0,1s. Desetiny sekundy jsou střádány do 32bitové celočíselné proměnné (long). Kvůli synchronizačnímu způsobu měření běží časovač nepřetržitě od zapnutí jednotky. Kdyby byl pro desetiny sekundy použit menší datový typ, docházelo by k jeho častému přetečení a tím k možným chybám měření. Maximální čas na 32 bitech je přibližně 6,8 let.

Aby bylo dosaženo dostatečného rozlišení měřeného času, je při měřené události zaznamenáno kromě desetinného čítače i samotná hodnota z časovače (vznikne tím tzv. časové razítko). Tím se dosáhne rozlišení pouhých 64 $\mu$ s. Zaznamenané časy jsou uchovávány ve struktuře.

```
TCapturedTime {
    long time;                // desetiny
    unsigned short counter;  // hodnota časovače
}
```

Pro potřeby práce se synchronizací je vytvořena i struktura se znaménkovým flagem.

#### 4.1.1 Operace s časem

Měření časového úseku je provedeno odečtením koncového a počátečního časového razítka. Synchronizace také potřebuje provádět výpočetní operace. Časová razítka sestávají z desetinného čítače a hodnoty časovače. Nelze je tedy prostě sečíst nebo odečíst. K tomu jsou vytvořena knihovny (pro znaménkovou i bezznaménkovou strukturu) se základními matematickými operacemi zahrnující součet, rozdíl a dělení dvěma. Zahrnuta je i funkce pro převod časového razítka na řetězec do formátu určeného v kapitole 3.3.1. Jiné operace nejsou potřeba.

#### 4.1.2 Odebrání časových razítek

Časová razítka jsou odebírána pouze v přerušeních z bezdrátového modulu a při změně stavu na pinu. Změna stavu na pinu je použita k zaznamenání času z drátového propojení. Toto přerušení má přiřazenu nejvyšší prioritu (6) a nemůže tedy být přerušeno ničím jiným. Přerušení z bezdrátového modulu má druhou největší prioritu a je používáno pouze k synchronizaci, která se s případným narušením musí vypořádat.

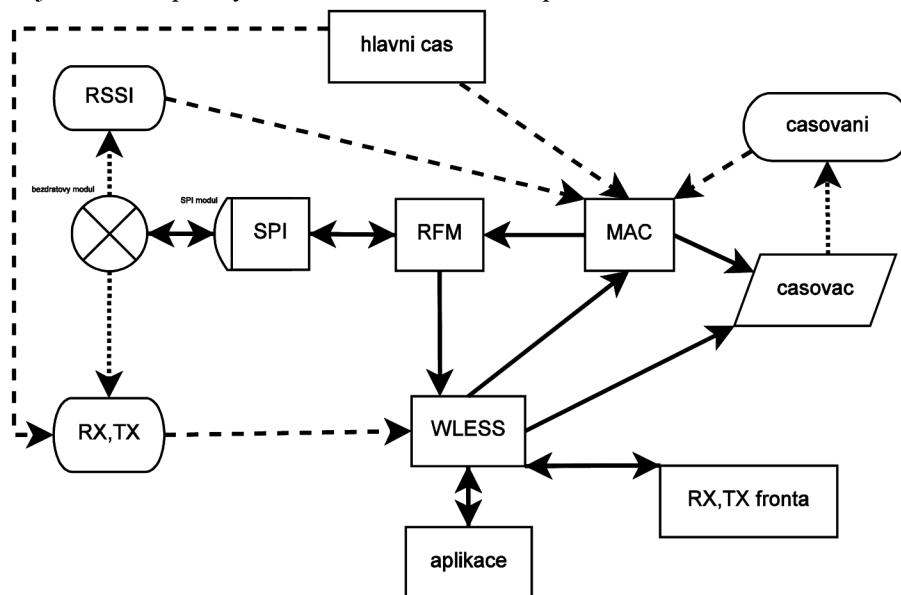
Přerušení z hlavního časovače má nastavenou prioritu nižší než přerušení z bezdrátového modulu. Při odebírání časových razítek by se mohlo stát, že dojde k automatickému vynulování

čítače, ale proměnná s desetinnými sekundami zůstane nezměněna. Proto je pro odebrání razítek implementována funkce, která tuto situaci hlídá. Kontroluje flag přerušení hlavního časovače. Je-li nastaven, zvýší lokální návratovou hodnotu desetinného záznamu.

## 4.2 Bezdrátová komunikace

Rozvržení implementace bezdrátové komunikace je zobrazeno na obrázku 18. Všechny části, kromě WLESS jsou stejné pro primární i sekundární jednotku. Úplně totožné jsou pouze RFM a MAC, implementace ostatních částí je závislá na hardwaru, logice a rozhraní se ale nemění.

Přeškrtnutý kruh představuje bezdrátový modul. K modulu je přistupováno přes SPI knihovnu. Ta implementuje rozhraní pro bytové a dávkové čtení a zápis na libovolnou adresu modulu.



Obrázek 18-Rozvržení implementace bezdrátové komunikace

RFM v obrázku přímo poskytuje služby bezdrátového modulu. Implementováno je odeslání a čtení dat a přepnutí do úsporného režimu. Zajišťuje také inicializaci bezdrátového modulu (parametry přenosu, složení paketu).

Přístup k médiumu hlídá část MAC. Aby dokázala určit, zda je médium volné, používá údaj RSSI. Modul je nastaven tak, aby na svém pinu GPIO provedl změnu stavu, když hladina RSSI přesáhne hodnotu 40dB (experimentálně určeno). GPIO je propojen s remapovatelným pinem, na který je namapováno externí přerušení. Při jeho každém vyvolání dojde k odebrání časového razítka. Podle toho vyhodnocuje dobu volného média. MAC má k dispozici i jeden 16 bitový časovač. Ten používá k časování DIFS, SIFS, backoff a limitu do kdy má přijít potvrzení. Logika časování je implementována stavovým automatem podle algoritmu v kapitole 3.2.3. Ovládací rozhraní k odeslání dat v MAC umožňuje kromě dat, cílové adresy a typu také zvolit, zda má být odesílaný paket potvrzen a jaký přístup k médiumu má být zvolen (DIFS, SIFS – odesláno bez ohledu na stav média). Deklarována je následovně.

```
int MAC_Send (
    int dest, WLessPackType type, char *data,
    int len, bool acked, eDelayAction ifs);
```

MAC žádné zprávy nepřijímá a tudíž ani nezastavuje nastavený časovač pro přeposlání v případě nepřijetí dat.

Přijatá data vyhodnocuje část WLESS. Bezdrátový modul je nastaven ke generování přerušení v případě příjmu a odeslání zprávy. Pin MCU připojený k bezdrátovému modulu je remapovatelný. Podobně jako při RSSI je na něj namapováno externí přerušení (jiné než u RSSI). Zpracování přijatých dat probíhá právě v tomto přerušení. Pro potřeby synchronizace je při každém přijetí a odeslání dat okamžitě, na začátku přerušovací procedury, sejmuto aktuální časové razítko. Zprávy obsahující data a zprávy typu ACK jsou vyhodnoceny stejně v primární i sekundární jednotce. Data jsou vloženy do přijímacího bufferu(RX), odkud si je vyzvedává hlavní program. Při přijetí ACK, ale i jiných potvrzovacích zpráv je zastaven časovač používaný MAC částí (jinak by došlo k znovuodeslání). Je zde i logika časové synchronizace a dynamického propojení jednotek.

WLESS se nestará jenom o přijaté zprávy. Zprostředkovává také odesílání zpráv. Ukládá si je a následně odesílá z odesílacího (TX) bufferu. Oba buffery jsou implementovány jako kruhové FIFO fronty s kapacitou 20 zpráv. Aplikace provádí odeslání, čtení dat, synchronizaci i využití okolních jednotek právě prostřednictvím těchto bufferů. Výjimku tvoří detekce jednotek.

## 4.2.1 Parametry přenosu

Zvolená modulace je GFSK s frekvenční odchylkou 50kHz. Přenosová frekvence 868MHz (jinou zvolit nelze) a přenosová rychlost 125kb/s (hodnota byla zvolena experimentálně a může se v budoucnu změnit).

K odesílání je využito automatického FIFO režimu bezdrátového modulu. V hlavičce je odesílána cílová adresa a typ zprávy. Dohromady jsou využity 3 ze 4 bytů. Zdrojová adresa je odesílána na začátku datové části. Při příjmu je hlavička nastavena k zahazení všech zpráv obsahujících na prvních 2 bytech jiný identifikátor než má daná jednotka přiřazený firmwarem (každá jednotka má vlastní). Propouští také zprávy definované s hodnotou  $FFFF_{HEX}$  (broadcast). Pakety jsou automaticky zabezpečeny CRC součtem.

## 4.2.2 Synchronizace

Časová synchronizace má definovány 4 typy zpráv. Jsou to SYNC, SYNC\_RE, SYNC\_ACK a SYNC\_END. SYNC označuje odebrání časového razítka. Příjemce si zaznamená čas příchodu této zprávy. SYNC\_RE označuje zprávu nesoucí čas přijetí SYNC (odesíláno z primární do sekundární jednotky). SYNC\_ACK slouží k potvrzení přijetí SYNC\_RE. SYNC\_END nese čas odeslání první zprávy SYNC (T1) a oznamuje validní provedení odesílací části.

Tyto zprávy nejsou potvrzovány paketem ACK, protože jsou odeslány v jednom sledu a mohou sami sloužit jako potvrzení přijetí. Platí u nich také stejná časovací pravidla k přístupu k médiu jako u ACK (kromě zahájení synchronizace). SYNC\_END není potvrzován. Pokud by nebyl doručen, došlo by k přeposlání SYNC\_ACK.

Při odesílání aktuálního času se může stát, že jeho odebrání bude znehodnoceno jiným přerušením. Aby tento stav bylo možné detekovat je v každém přerušení nastavena globální proměnná značící jeho vyvolání. Proměnná je před odebráním časového razítka vynulována a posléze zkontrolována. Došlo-li k přerušení je nastaven flag chyby synchronizace.

Synchronizace je prováděna bezstavově. Tabulka 6 popisuje reakci obou jednotek při přijmutí jednotlivých synchronizačních zpráv.

příjem	chyba	primární jednotka	sekundární jednotka
SYNC	ano	odesláni SYNC	odesláni SYNC
	ne	odesláni SYNC_RE	
SYNC_RE	ano	-	odesláni SYNC
	ne		odesláni SYNC_ACK
SYNC_ACK	ano	odesláni SYNC_END	-
	ne		
SYNC_END	ano	-	výpočet rozdílu časů, úprava vlastního času
	ne		

**Tabulka 6-Bezstavová implementace přenosu časových razítek**

Při úspěšném dokončení je v sekundární jednotce ihned vypočítán rozdíl času vůči primární jednotce. Úprava času je provedena až v přerušení hlavního časovače. Kdyby byla tato operace provedena ihned a časovač by byl těsně před dočítáním do nastavené hranice, mohlo by dojít k nechtěným následkům (čas je totiž měněn za běhu).

### 4.2.3 Detekce a využití jednotek

Pro detekci a spárování jednotek jsou vyhrazeny typy DISCOVER, IMHERE, SEIZE, RELEASE. Zprávu DISCOVER odesílá broadcastem primární jednotka (doručení není potvrzováno). Nepoužívané sekundární jednotky, které ji přijmou, odpoví zprávou IMHERE. Tato zpráva je adresována konkrétně primární jednotce. Její tělo nese dva byty. Konkrétně jde o typ jednotky (SENZOR) a počet snímačů (1).

Při vyžádání nalezení jednotek je zpráva DISCOVER odeslána několikrát po určitých intervalech. Odpovědi jsou kumulovány do aplikací definovaného ukazatele na pole. Vyhledání jednotek je implementováno blokujícím způsobem, po odesláni DISCOVER se 1s vyčká na příchozí odpovědi. Bezdrátová část primární jednotky si zabrané jednotky nijak nehlídá, pamatovat si je musí vyšší vrstva aplikace.

SEIZE slouží k spárování sekundární jednotky. Sekundární jednotka si při jejím obdržení napevno nastaví adresu primární jednotky, kterou automaticky používá k adresaci všech zpráv. Na rozdíl od primární jednotky si tedy vyšší vrstva aplikace nemůže vybrat adresáta.

Zpráva RELEASE danou jednotku převede do výchozího stavu. RELEASE může být adresován i broadcastem. V takovém případě jsou uvolněny všechny spárované jednotky náležící jejímu odesílateli.

Spárovaná jednotka po pěti minutách bez příchozí zprávy sama přechází do výchozího stavu, využívá globální proměnnou inkrementovanou hlavním časovačem. Podobně jednotka hlídá čas od poslední synchronizace. Ten je nastaven na 30 sekund (podle maximální teoretické odchylky, viz 3.1.1) Po této době jednotka zakáže odesílání všech zpráv s výjimkou těch synchronizačních.

## 4.3 Primární jednotka

Výchozí nastavení mikrokontrolerů PIC se odvíjí od obsahu jeho konfiguračních registrů. Pomocí nich lze zvolit např. i hlavní zdroj hodinových impulzů, kterým je pro tuto práci externí 10MHz oscilátor. Napájení tohoto oscilátoru je však ovládáno z MCU a v defaultním stavu je vypnutý. Proto musí MCU ze začátku běžet na interní zdroj hodin (8MHz). Konfigurační registry jsou ponechány téměř beze změny oproti defaultnímu nastavení, vypnut je pouze watchdog.

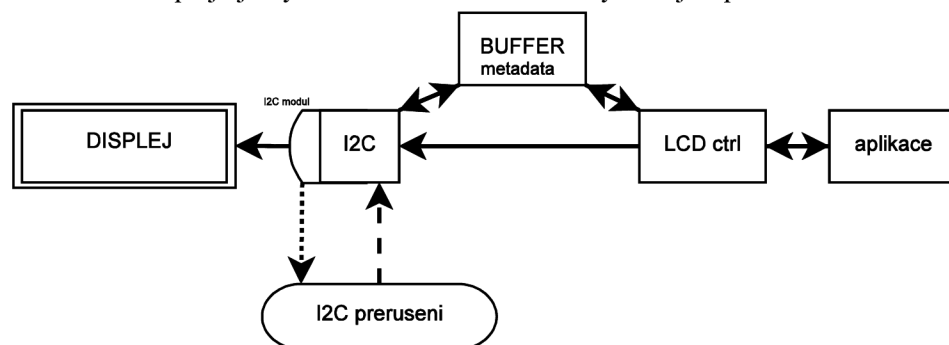
Ihned po startu jednotky jsou nakonfigurovány porty MCU a je zapnuto napájení periférií, mezi nimiž je i 10MHz oscilátor. Následně se určitou chvíli, v řádu desítek  $\mu$ s počká na jeho naběhnutí. Poté je provedeno přepnutí zdroje hodinových pulzů. Toho je dosaženo zápisem do kontrolního registru oscilátoru a následným vyčkáním na signalizaci dokončení přepnutí. Aby bylo možné do registru zapisovat, musí se nejdříve provést určitý sled instrukcí. Překladač xc16 však k tomuto účelu poskytuje zabudované funkce, které sami provedou odemknutí a následný zápis.

Po přepnutí oscilátoru je provedena inicializace ostatních částí zařízení (časovače, přerušení, LCD, bezdrátová komunikace ...). Následně přechází vykonávání programu do hlavní programové smyčky (`while(1) { ... }`).

### 4.3.1 Výpis znaků na LCD

Ovládání displeje je rozvrženo do dvou částí. Kromě příkazů k zápisu znaků a řetězců na displej je zahrnuta i podpora pro menu a stavový řádek.

Na nejnižší úrovni je knihovna I2C. Ta zprostředkovává zadávání dat do LCD displeje. Čtení dat není zahrnuto. Čtení by bylo potřeba pouze k určení, zda je displej připraven zpracovat další příkaz, ale protože je komunikace po I2C značně pomalá (frekvence 400kHz), bylo zjištěno, že displej zvládá zpracovávat příkazy bez dalších prodlev. Aby zápis na displej nezdržoval rychle běžící mikrokontrolér, je komunikace implementována za pomoci přerušení z I2C modulu. Ta však úzce spolupracuje s vyšší vrstvou a umožňuje pouze zápis znaků a změnu řádku a sloupce vypisovaného znaku. Kvůli inicializaci displeje je vytvořena i komunikace nevyužívající přerušení.



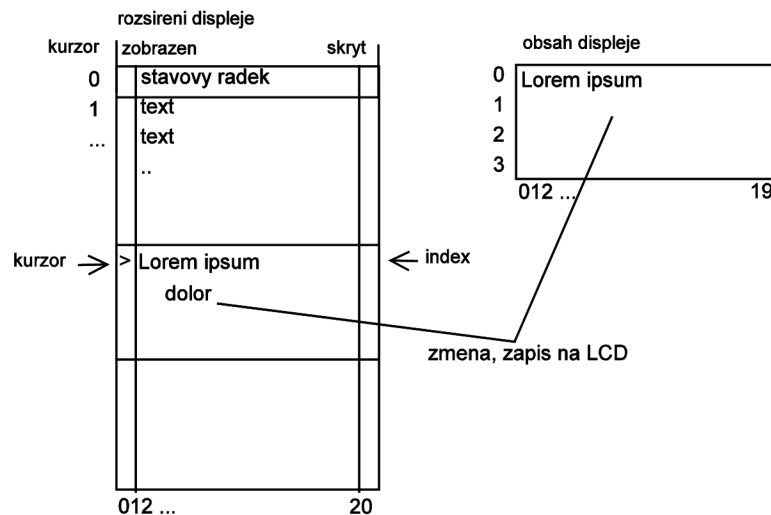
Obrázek 19-Rozvržení implementace využití LCD displeje

Další část realizuje inicializaci displeje a poskytuje funkce k vypsání znaku a řetězce na zvolenou pozici displeje. Přidána je i vlastní sada znaků představující různé úrovně nabití akumulátoru, znak představující SD kartu a znak oznamující povolenou bezdrátovou komunikaci. Kvůli volbě zápisu dat pomocí přerušení jsou vypisované znaky nejprve vloženy do mezi-paměti, odkud je I2C čerpá.



Protože má displej paměť pouze na 4 řádky textu, nelze smysluplně využít zabudovaného rolování řádků. Proto je přidána část emulující větší počet řádků. Ta je realizována dvou-dimenzionálním polem o rozměrech 20 sloupců 40 řádků (výše zmíněná mezi-paměť). K němu je přidán buffer uchovávající fakticky vypsané znaky na LCD a proměnná uchovávající index řádku, který je na displeji vypsán. Buffer je zde kvůli tomu, aby se nepřepisoval při každé změně celý displej, ale pouze jednotlivé znaky. Index lze zvyšovat a snižovat, tím se docílí požadovaného rolování textu.

Kromě řádkového rozšíření je přidána i další podpora pro menu. Ta zahrnuje volitelně zobrazitelný kurzor, který lze posouvat, podobně jako index řádku. Při zapnutí kurzoru jsou všechny sloupce posunuty o jeden znak doprava a do prvního sloupce na řádku s kurzorem je vložen znak šipky (kurzor). Kurzoru lze také nastavit minimální pozici (řádek od shora), na kterou smí vstoupit. Řádky odshora spadající pod minimální pozici navíc nejsou posunuty o jeden znak doprava. To se hodí zejména pro potřeby stavového řádku. Rozšíření je načrtnuto na obrázku 20.



Obrázek 20-Řádkové rozšíření displeje

### 4.3.2 Záznam měření

K ukládání záznamů j vytvořen samostatný modul. Modul uchovává jak konfigurační informace o měření tak také samotné časy. Zajišťuje i celkový start a stopnutí probíhajícího měření. Časy jsou uchovávány v poli struktur s kapacitou 20 záznamů. Struktura obsahuje položky: typ záznamu, id zařízení, pořadové číslo senzoru, zaznamenaný čas, údaj jestli je čas zastaven a jestli je validní. Typy údajů mohou být: start, stop mezičas a kumulativní mezičas. Přičemž start může být jen jeden a je napevno přiřazen prvnímu záznamu. Stop položek může být libovolné množství, celkové měření zastaví až poslední zastavený čas typu stop. Mezičasy nemají na zastavení měření vliv. Pokud přijde žádost o zastavení již zastaveného kumulativního mezičasu, je za poslední záznam přidána nová položka, do které je přichodzí čas zastaven. Minimálně lze mít v poli 2 záznamy. Bez alespoň jednoho koncového času nelze měření spustit.

Časy jsou vkládány přes rozhraní deklarované následovně:

```
void REC_StopTime(int deviceID, int senzor, TCapturedTime
time);
```

Záznam obsahující časový údaj nemůže být do vynulování přepsán. Pokus o vložení času z jiného než nakonfigurovaného senzoru nemá žádný vliv.

Zastavení měření má za následek převod záznamu do CSV formátu a pokus o uložení dat na SD kartu.

Časy z přímého drátového propojení jsou zaznamenány na začátku přerušení vyvolaném změnou stavu na pinu. Ochrana proti záskmitům je provedena způsobem popsáním v kapitole 2.1.4. K znečitlivění vstupu je použit hlavní časovač, který postupně dekrementuje nastavené proměnné. Jakmile dosáhnou nuly, může být přijat další impuls. Doba znečitlivění je nastavena na 0,1s. Zaznamenané časy mají napevno přiřazeny identifikátor s hodnotou 2 (DEVICE\_THIS\_IN) a pořadové číslo, kterým se identifikují výše popsanému modulu.

Časy z bezdrátového spojení jsou zasílány datovou cestou. Přijatá data jsou průběžně vyzvedávána a zpracovávána hlavní programovou smyčkou. Kvůli možnosti budoucího rozšíření obsahují všechna odesílaná data typ a samotný obsah. Momentálně existuje pouze jeden typ. Typ oznamující přenašení času. S časem je přenašeno i číslo senzoru, ze kterého byl čas sejmuto. Aktuální sekundární jednotky mají pouze jeden senzor. Tyto údaje jsou předávány k měření.

### 4.3.3 SD karta

Zápis na SD kartu je proveden pomocí souborové knihovny od Microchipu. Použita je verze s krátkými názvy souborů. Pro účely knihovny je zprovozněn RTCC modul a jsou implementovány funkce pro přístup k potřebným pinům. Inicializační hodnoty pro RTCC modul jsou brány z předdefinovaných maker překladače `__TIME__` a `__DATE__` [16] (čas zkompileování kódu).

Celá SD knihovna je ještě abstrahována do vlastního modulu. Ten automatizuje inicializační operace při prvním vložení SD karty („mount“). Zároveň se zjistí aktuální datum a převede se do řetězce s příponou .CSV, který slouží k pojmenování zapisovaných souborů. Odebrání karty zapříčiní vyvolání operace „unmount“. Vyhledání karty je umístěno v hlavní programové smyčce. Připravenost karty k použití lze zjistit z funkce `SD_IsReady()`.

Zápis je prováděn posloupností operací otevření souboru, zápisu a zavření souboru. Bez uzavření by nemuselo dojít k fyzickému uložení dat na SD kartu. Zapisovaná data musí volající vložit do bufferu s kapacitou 500 znaků a vyžádat si zápis. Samotná zapisovací procedura je volána v hlavní programové smyčce.

### 4.3.4 Nabíjení a stav akumulátoru

Dobíjení akumulátoru je řešeno podle popisu v kapitole 2.1.7. Stav napětí je čten pomocí 10bitového AD (Analog to digital) převodníku. Převod je prováděn na manuální vyžádání z programu. K tomu je použit třetí časovač. Frekvence čítání je snížena předděličkou 256×. To snižuje frekvenci dostatečně pro nastavení časování 1s intervalů. Nejedná se o přesnou sekundu, ale pro potřeby převodníku plně postačuje.

Vyhodnocení stavu akumulátoru probíhá v hlavní programové smyčce. 10bitová hodnota převodníku (1024) je vztažena na napájecí napětí mikrokontroléru (3,3V). Nabitá baterie má napětí 2,05V, což na 1024 stupnici odpovídá hodnotě 636. Klesne-li hodnota z převodníku pod hodnotu 589 a je detekován zdroj nabíjení (vstup pinu PG), povolí se dobíjení.

Kromě dobíjení je hodnota z převodníku použita i k indikaci stavu akumulátoru. Získané napětí je rozděleno do 6 intervalů udávajících různý stav nabití, každý má přiřazen vlastní znak. Hodnoty byly určeny přibližně podle vybíjecí charakteristiky Li-Ion baterií [14]. Znak akumulátoru je

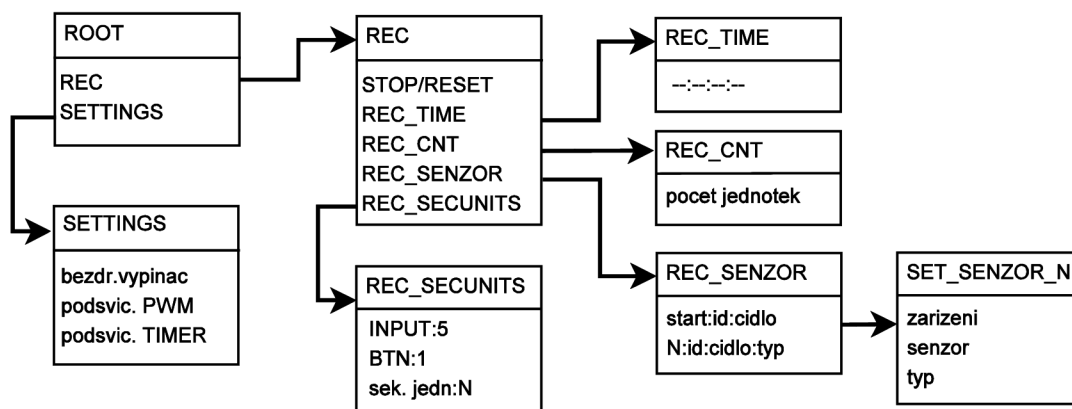
vypisován na poslední pozici prvního řádku. Pokud je spuštěno dobíjení, jsou znaky baterie postupně, po sekundách iterovány (od znaku vybité po znak dobité baterie), takže vznikne jednoduchá animace dobíjení.

### 4.3.5 Ovládání a zobrazení

Menu je realizováno pomocí stavového automatu. Stisknutí tlačítek je zaznamenáváno v přerušeních pro změnu stavu na pinu. Stejně přerušení jako vstupy drátového připojení koncových prvků. Toto přerušení má nastavenou nejvyšší prioritu. Aby se nezdržoval chod všech ostatních částí programu, jsou v něm události zmáčknutí tlačítka pouze zaznamenány do globálních proměnných a jejich vyhodnocení stavovým automatem je provedeno až v hlavní smyčce programu.

První řádek je ponechán jako stavový. Na poslední pozici je znak baterie, předposlední ukazuje, zda je zapnutý bezdrátový modul a třetí od konce je ponechán pro indikaci připravenosti SD karty. Při každé změně stavu dojde k aktualizaci stavového řádku.

Hlavní stavy představující jednotlivé obrazovky jsou: ROOT, REC, SETTINGS, REC\_TIME, REC\_CNT, REC\_SENZOR, REC\_SECUNITS a REC\_SET\_SENZOR\_N. ROOT, kořenový stav, vede k REC a SETTINGS. Obrázek 21 zobrazuje rozvržení menu.



Obrázek 21-Rozvržení menu

SETTINGS nabízí 3 položky k nastavení zařízení. Je to zapnutí/vypnutí bezdrátové komunikace, nastavení jasu podsvícení displeje (v % po 10 krocích od 0 do 100) a nastavení doby, po které bude zhasnuto podsvícení. Pro úpravu jasu je použit modul PWM, který je nemapován na pin podsvícení displeje. Jako vstup hodinových pulzů PWM je použit časovač 5, nebo spíše frekvenční předdělička časovače dělicí hodnotou 256. Perioda je nastavena na hodnotu 99, uživatelem definovaná hodnota 0-100 je přímo použita jako střída. Ke zhasnutí displeje po určité době je použit 3 časovač (jako u ADC baterie – 1s) dekrementující proměnnou počínající na uživatelem definované hodnotě. Jakákoliv akce v menu (stisk tlačítek) re-inicializuje dekrementovanou hodnotu a pokud je displej zhasnutý, rozsvítí jej. Pokud je detekováno probíhající měření, zůstává displej svítit.

REC je rozcestí pro REC-TIME/CNT/SENZOR/SECUNITS. Kromě toho je v něm detekováno, zda právě není spuštěno měření, nebo jestli bylo dokončeno, ale není vynulováno. Pokud právě probíhá, je na první pozici možnost jeho zastavení. Tím se také znehodnotí doposud nezastavené časy. Bylo-li měření zastaveno, je místo zastavení volba vynulování časů.

REC\_TIME zobrazuje samotné časy. Při probíhající měření je na prvním řádku zobrazen běžící čas, obnovující se po desetínách sekundy (použit časovač 1). Jinak jsou zde vypsané nastavené snímače. Prostřední a pravé tlačítko zde předávají podnět k zastavení času (levé = senzor 1, pravé 2).

V REC\_CNT se nachází nastavení počtu snímačů/drah. Jejich výčet je v REC\_SENZOR. Při vstupu do tohoto stavu dojde k vyhledání sekundárních jednotek (pokud je zapnuta bezdrátová komunikace a žádné jednotky nebyly doposud nalezeny), které jsou následně nabídnuty k použití. Nastavení jednotlivých snímačů/drah je v další podúrovni (REC\_SET\_SENZOR\_N). Výběr bezdrátově propojeného snímače má za následek jeho spárování a synchronizaci. Při odebrání je sekundární jednotka uvolněna.

Poslední zbývající stav je REC\_SECUNITS. V něm jsou zobrazeny lokálně dostupné snímače, ale hlavně jsou zde zobrazeny dostupné sekundární jednotky. Pokaždé (při zapnutí bezdr. kom.) jsou jednotky znovu vyhledány.

### 4.3.6 Úsporný režim – vypnutí

Vypnutí je řešeno přepnutím mikrokontroléru do režimu spánku. Je-li detekován vypnutý stav na přepínači, přechází jednotka do režimu spánku. Před samotným usmáním jsou uvolněny všechny spárované jednotky, následuje přepnutí hlavního zdroje hodin na interní 8MHz RC oscilátor. Potom jsou de-inicializovány veškeré výstupní a remapovatelné piny, vypnuty použité periferie a jsou zakázána všechna přerušení. De-inicializace pinů před přepnutím oscilátoru by znamenala odpojení energie hlavnímu oscilátoru v době kdy je ještě používán. Před samotným usmáním musí být zajištěno jeho probuzení. Probuzení oscilátoru může způsobit jakékoliv přerušení. Proto je k přepínači namapováno externí přerušení. Aby mohla být jednotka dobíjena i ve vypnutém stavu, je přerušení namapováno i k pinu detekujícímu připojený zdroj energie. Následuje již samotné usmáním pomocí předdefinovaného makra `Sleep()`. RTCC zůstává aktivní.

Po probuzení jsou znovu inicializovány piny, je přepnut hlavní zdroj hodin zpět na externí oscilátor a jsou inicializovány displej, tlačítka, časovače a napájecí část. Poté je zjištěn důvod probuzení.

Pokud k němu došlo kvůli připojení zdroje energie, zůstane program běžet v lokální programové smyčce, dokud není zařízení přepínačem zapnuto (pokračuje v inicializaci) nebo dokud není odpojen zdroj energie (znovu se uspí).

Probuzení způsobené zapnutím provede, až na RTCC část, všechny inicializační kroky jako při prvním zapnutí jednotky.

## 4.4 Sekundární jednotka

Kód sekundární jednotky je podstatně jednodušší a kratší než v jednotce primární. Inicializace je prováděna obdobně. Hlavní 10MHz oscilátor je také napájen až na vyžádání z MCU. Po zapnutí jsou tedy nakonfigurovány porty a je zapnuto napájení oscilátoru. Následuje přepnutí hlavního zdroje hodinového signálu. Potom jsou inicializovány ostatní části a program přechází do hlavní programové smyčky.

## 4.4.1 Snímací čidlo – časový záznam

Pin s připojeným snímacím čidlem je nastaven k vyvolání přerušení při změně logické hodnoty. Protože může být připojen snímač signalizující sepnutí v log. 0 i v log. 1, je k zajištění korektního vyhodnocení je porovnávána hodnota z čidla s hodnotou z pinu, na kterém je přiveden výstup uživatelem definovaného přepínače. Zákmity jsou ošetřeny stejně jako v primární jednotce znečitlivěním vstupu. Doba znečitlivění závisí na dalším, uživatelem definovaném, přepínači. Je to buď 0,5s nebo 5s.

Při detekovaném sepnutí čidla dojde k pokusu o odeslání sejmutého časového razítka primární jednotce (nespárovaná jednotka nic neodešle). Ještě před odesláním je ovšem nastavena logická 1 na pin určený k přímému drátovému propojení. Zároveň je rozsvícen přední signalizační panel. Sepnutí drátového připojení i světelné signalizace mají stejnou dobu trvání. Přední panel zůstává rozsvícen minimálně po dobu znečitlivění snímače. Pokud je čidlo sepnuté, zůstává panel rozsvícen. Až jeho rozpojení způsobí zhasnutí světla. V závislosti na stavu dalšího přepínače je nastavena doba 30s, po které se rozsvícené světlo rozbliká. Doba 30s i blikání zajišťuje rutina přerušení z hlavního časovače. Ten dekrementuje přednastavenou proměnnou a po dosažení nulové hodnoty obrátí stav signalizace a nastaví proměnnou, v závislosti na předchozím stavu signalizace na hodnotu 3 resp. 7 (0,3s rozsvíceno, 0,7 zhasnuto).

## 4.4.2 Světelná signalizace

Způsob signalizace při sepnutí čidla je popsán výše. Svit předního panelu je však ještě kvůli snížení spotřeby regulován PWM modulem. Jeho nastavení je stejné jako PWM pro podsvícení displeje v primární jednotce. Střída je prozatím nastavena na 70% periody. Signalizace musí být vidět i na přímém slunečním světle, proto je hodnota zvolena relativně vysoko. Na základě testování může být dále snížena.

Po zapnutí jednotky je rozsvícena zelená LED dioda. Kromě signalizace zapnutého stavu je použita i k signalizaci vybitého akumulátoru. Stav akumulátoru je určen stejně jako v primární jednotce pomocí napětí, které je převedeno 10bitovým ADC. Jakmile je zaznamenána hodnota odpovídající stavu energie pod 30% (527), je zelená dioda rozblikána.

Modrá LED dioda slouží k signalizaci časově synchronizovaného zařízení. Obdobným způsobem jako rozblikání předního panelu je hlídán čas od poslední synchronizace. Po 30 sekundách dojde k rozblikání diody, po dalších 30 sekundách dioda zhasne a v bezdrátové části je zakázáno odesílání jiných než synchronizačních zpráv. Po další minutě bez jakýchkoliv příchozích zpráv je přerušeno bezdrátové spárování jednotky s primární jednotkou.

## 5 Ověření funkčnosti

Jednotlivé části byly postupně ověřovány během vývoje. Vstupy pro drátové propojení na primární jednotce byly ověřeny fyzickým připojením tlačítek. Sekundární jednotka musí být schopná reagovat na vstupní impulzy podle zvolené konfigurace. Odzkoušen byl indukční snímač (aktivní prvek) a tlačítko (pasivní prvek). K sekundární jednotce je tedy možné připojit libovolný snímač (s odpovídajícím konektorem). Primární jednotka funguje korektně s libovolným zařízením udávajícím koncové sepnutí uzemněním vodiče.

Bezdrátová komunikace byla ověřena za použití 3 sekundárních a jedné primární jednotky. Jednotky byly umístěny kousek od sebe, takže nehrozily chyby z důvodu nedostatečného signálu. Vyhledání jednotek proběhlo opakovaně, asi v 50 pokusech. Vždy byly korektně detekovány všechny jednotky. Jednotky byly také bez problému spárovány s primární jednotkou a bylo ověřeno i odesílání koncových časů.

Vizuálně byly ověřeny i ostatní části. Byl zaznamenán problém s využitím SD karty. Kartu je někdy nutné vložit na několikrát, aby byla řádně detekována.

Zapnuté zařízení zobrazuje obrázek 22. Vpravo nahoře, na prvním řádku, je vidět indikace vložené SD karty, zapnutá bezdrátová komunikace a indikátor stavu nabití baterie. Ukázka funkčnosti primární i sekundárních jednotek je zaznamenána na video přiložené na CD. V ukázce použité složení časoměrného systému je stejné jako složení použité k praktickému ověření popsánému níže (5.3).



Obrázek 22-Zapnutá primární jednotka

### 5.1 Přesnost bezdrátového měření

Přesnost bezdrátového měření je závislé především na přesnosti provedené časové synchronizace. Aby bylo možné určit tuto chybu, byl na jeden ze vstupů primární jednotky a na vstup ze sekundární jednotky připojen jeden snímací prvek. Program primární jednotky byl upraven tak, aby se sejmутý čas ze vstupu vypisoval na displej v interní podobě (hodnota desetinného střadače a hodnota z časovače), stejným způsobem byl vypsán čas přijatý ze sekundární jednotky. Navíc bylo přidáno

vykonání synchronizace na vyžádání po zmáčknutí tlačítka. Při měření byla vždy nejdříve vykonána synchronizace a následně bylo, s maximální prodlevou 1s, protnuto měřící čidlo. Hodnoty z časovačů se vždy lišily maximálně o 1, což při nastavení frekvence čítače na 156250MHz znamená chybu 64 $\mu$ s.

Kromě chyby vzniklé synchronizací je však potřeba také počítat s chybou způsobenou nepřesností oscilátorů. Teoreticky, v extrémním případě, se jedná o chybu 1ms na každých 30 vteřin (viz kapitola 3.1.1). Doba obnovy synchronizace je nastavena na 20s. Je však nutné zjistit jestli udávané odchylky odpovídají skutečnosti. K měření byla jedenkrát provedena synchronizace jednotek a následně bylo po 3 minutách sepnuto vstupní čidlo (automatická obnova byla vypnuta). Takto otestovány byly 3 různé sekundární jednotky. Jedna se lišila průměrně o 49 hodnot časovače, další o 182 a poslední o 166. Největší chyba byla 1,1648ms/3min. To je mnohem nižší chyba než teoreticky určená.

## 5.2 Spotřeba a výdrž na akumulátoru

Ke změření spotřeby byly jednotky připojeny na zdroj energie dodávající konstantní napětí 4V, což odpovídá téměř nabitému akumulátoru. Následně byl při různých stavech jednotek zaznamenáván jejich proudový odběr. Tato metoda bohužel nedovoluje měřit chvilkové nárůsty odběru, jako jsou zápis na SD kartu nebo odeslání dat.

Ničím omezená primární jednotka vykazuje odběr 160mA. Snížení jasu podsvícení displeje (PWM) na 50% znamená pokles na 130mA a úplně vypnutí podsvícení jej sníží na 80mA. Vypnutí bezdrátové komunikace znamená pokles o dalších 20mA (tím byla také potvrzena funkčnost vypnutí bezdrátového modulu).

Při vypnutí hlavním vypínačem klesne odběr na 2mA. Udávaná hodnota spotřeby MCU v režimu spánku je mnohem nižší (viz tabulka 1), což indikuje možnou chybu v použité uspávací sekvenci. I tak by ale jednotka s nabitým 2200mAh akumulátorem měla vydržet bez externího napájení přibližně 45 dní.

Sekundární jednotka ve výchozím stavu po zapnutí odebírá 50mA (MCU, bezdrátový modul a indikace zapnutí zelenou LED diodou). Po provedení časové synchronizace a s tím spojeným rozsvícením další indikační diody vzroste odběr na 70mA. Rozsvícením předního panelu (regulovaný pomocí PWM na 70%) vzroste spotřeba na 270mA. Při PWRSERVE módu, kdy po 30 sekundách dojde k rozblíkání panelu v poměru 0,3s svítí, 0,7s nesvítí je jeho odběr v průměru snížen na 130mA.

Výdrž obou jednotek je závislá na způsobu jejich využití. Vzhledem k specifikacím sekundární jednotky je dále uveden příklad použití ze soutěže v požárním útoku.

V nejhorším případě bývají koncová zařízení (zde sekundární jednotky) ponechány v sepnutém stavu okolo 5 minut. Mezi dalším sepnutím bývá prodleva přibližně 30 sekund. Při nastavení rozblíkání světla po 30s, ignorování úvodního inicializačního procesu a případných krátkých prodlev vychází výdrž baterie pro sekundární jednotku kolem 16h. Pokud by bylo čidlo rozepnuto do 30 sekund, výdrž by se zvýšila na 25h.

Primární jednotka by při nastavení podsvícení displeje na 100% jasu, jeho ztmavení po 30 sekundách a za předpokladu, že bude podsvícení zapnuto během měření (30s), následně do doby automatického zhasnutí (30s) a poté zůstat 4,5 minuty neaktivní, měla vydržet 23h. A s podsvícením omezeným na 50% až 31h.

## 5.3 Ověření v praxi

Časomíra byla použita k měření časů na tréninku soutěžního družstva SDH Tvrdonice. Trénink byl zaměřen na disciplínu požárního útoku. Fotodokumentace je přiložena v příloze.

Akumulátor v primární jednotce byl před testem jednotkou nabit na maximální kapacitu (limitovanou softwarem). Pro ukládání záznamů byla vložena SD karta. Jas podsvícení displeje byl nastaven na 100% (bez omezení) a doba automatického zhasnutí byla zvolena na 30s. Na vstup primární jednotky (konektor XLR) bylo navíc připojeno zvukové startovací zařízení.

Vstup sekundárních jednotek byl nastaven ke snímání sestupných hran z pasivního čidla (mechanismus koncového zařízení dává koncový podnět rozepnutím magnetického jazýčkového kontaktu). Doba znečitlivění zde nehraje roli (nastavena byla na 0,5s). Zapnuto bylo také rozblikání rozsvíceného panelu po době delší než 30s.

Na začátku tréninku byly sekundární jednotky připevněny na koncové zařízení a primární jednotka byla umístěna ke startu. Vzdálenost mezi sekundárními a primární jednotkou byla přibližně 95m a nebyly mezi nimi žádné překážky. Sekundární jednotky byly bez problémů detekovány a nastaveny jako zdroje dvou koncových časů. Spárování i časová synchronizace proběhly v pořádku. Problém nastal s SD kartou. Vložená SD karta zpočátku nebyla detekována. Po několika opakovaných vloženích byla korektně rozpoznána. Po úvodním nastavení byly jednotky ponechány v zapnutém stavu až do konce tréninku. Během něj byly korektně dokončeny 4 pokusy (došlo k sepnutí snímačů). Kromě toho byla primární jednotka 3x odstartována a následně bylo měření lokálně ukončeno.

Zařízení zůstala zapnutá po dobu 2 hodin, během níž nebylo zaznamenáno nekorektní chování měřicího systému (nedoručená data, nesprávná synchronizace aj.). Ukazatel akumulátoru na primární jednotce po skončení stále vykazoval nabitý stav. Všechny měření byly správně zaznamenány na vloženou SD kartu do souboru pojmenovaném aktuálním datem (firmware byl aktualizován předchozího dne). Obsah souboru je také v příloze.

Na konci byl ještě proveden test maximálního dosahu bezdrátové komunikace. K testu byla využita detekce jednotek. Již bylo zjištěno, že komunikace funguje na vzdálenost 95m. Ve vzdálenost přibližně 120m přestala být detekce spolehlivá. Často byla vyhledána pouze jedna jednotka, někdy žádná. Ve vzdálenosti ~150m již nebylo možné vyhledat ani jednu jednotku. Výpadky byly nejspíš způsobeny ztrátou nepotvrzovaných zpráv. Pro ujištění byl test proveden ještě jednou, tentokrát ale se spárovanými jednotkami a odesílala se potvrzovaná data. Zprávy se začaly ztrácet až za hranicí 140m.



## 6 Závěr

Práce si kládla za cíl vytvoření programového vybavení pro sportovní vícejednotkovou bezdrátovou časomíru. Časomíra je složena ze dvou druhů jednotek. Oba druhy byly blíže popsány v úvodu práce. Popis byl zaměřen především na části významné pro tvorbu programového vybavení. Řídící prvky v obou jednotkách jsou 16 bitové mikrokontroléry z řady PIC24F. Byly zde také shrnuty vývojové prostředky potřebné k realizaci programového vybavení.

V následující části byl navržen způsob měření času pomocí časové synchronizace mezi jednotkami. Pomocí tohoto způsobu již není měření času závislé na okamžiku přijetí zprávy a bezdrátové měření se tak stává mnohem spolehlivější. Aby mohly být k jedné primární jednotce použity libovolné sekundární jednotky, byl navržen způsob adresace jednotek a s tím související detekce a následné jejich využití. V jednotkách použitý bezdrátový modul umožňuje automatizované odesílání i detekci poškozených zpráv. Nezaručuje však jejich spolehlivé doručení. K řešení tohoto problému byla zvolena přístupová metoda CSMA/CA. Využití bezdrátového propojení poskytuje velkou škálovatelnost měřicího systému. Aby mohl být systém použitý k měření v různých sportovních odvětvích, bylo navrženo rozhraní, které ponechává konfiguraci měření na uživateli. To zahrnuje nastavitelný počet měřených časů a k nim přiřazení dostupných snímacích prvků. Kromě koncových časů byla přidána možnost měření mezičasů. V rámci zadání práce bylo i využití příložitelné SD karty k ukládání naměřených dat. Byl zvolen způsob ukládání dat ve formátu CSV do souborového systému FAT32. Návrh se také zaměřil na snížení spotřeby jednotek. Největší spotřebu mají výstupní prvky – LED diody a displej. Ty byly omezeny PWM moduly. V primární jednotce bylo nastavení ponecháno na uživateli.

Návrh byl posléze realizován. Implementace bezdrátové komunikace byla rozvržena tak, aby se dala snadno využít bez větších zásahů i na případný nový typ sekundárních jednotek. Zahrnuje v sobě podporu pro časovou synchronizaci, detekci i propojení jednotek. Výsledné časy jsou odesílány přes rozhraní univerzálně využitelné i k přenosu jiných dat. Ukládání dat na SD kartu je realizováno za pomoci hotového řešení od firmy Microchip. V primární jednotce byla také implementována správa dobíjení a kvůli absenci fyzického vypínače i převod jednotky do režimu spánku.

Výsledný produkt byl následně otestován. Bezdrátová komunikace byla odzkoušena za pomoci 1 primární a 2 sekundárních jednotek. Na blízkou vzdálenost, do 100m, nebyl zaznamenán žádný problém s jejich detekcí, ani s následným odesláním dat. Ve větší vzdálenosti nastává problém s detekcí. Následná komunikace však funguje bez problémů. Příčina může být v slabé síle signálu. Standardní zprávy jsou totiž potvrzovány a při nedoručeném potvrzení přeposlány, vyhledávací zpráva ovšem potvrzována není. Na vzdálenost vyšší než 140m se již komunikace stává nepoužitelnou.

Byla také změřena přesnost použité časové synchronizace jednotek. Okamžitě po jejím provedení se čas v jednotkách liší maximálně o 64 $\mu$ s. Kvůli nepřesnosti použitých oscilátorů se chyba postupně zvyšuje. Obnova synchronizace po 20s však zaručuje přesnost měření s chybou nižší než 1ms. Změřena byla i spotřeba jednotek. Výdrž obou jednotek je závislá především na jejich použití. Teoreticky by se však měla u obou jednotek pohybovat okolo 1 dne. Měření odhalilo možnou chybu při usnutí primární jednotky. Jednotka odebírá více proudu než by měla, i tak ale vydrží „vypnutá“ bez externího napájení déle než měsíc. Zařízení bude potřeba blíže proměřit a nalézt příčinu zvýšeného odběru. Při testování byl také odhalen problém s detekcí SD karty. Kartu je někdy nutné vložit opakovaně, aby byla úspěšně rozpoznána. Poté již funguje bez problémů.

Díky zachování možnosti drátového připojení lze primární jednotku propojit s libovolným koncovým prvkem dávající koncový impulz uzemněním vodiče. Sekundární jednotku lze obdobně zkombinovat s existujícími měřicími systémy. Bezdrátová komunikace je však specificky navržena pro tento systém a kombinace pomocí ní s dostupnými měřicími zařízeními je vyloučena.

Do budoucna by bylo vhodné rozšířit komunikaci primární a sekundárních jednotek např. o informace ze stavu akumulátoru sekundární jednotky nebo umožnit její vzdálenou konfiguraci. Další rozšíření se může týkat propojení primární jednotky s počítačem přes sběrnici USB.

# Literatura

- [1] GÁL, M. Sportovní bezdrátová časomíra. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 63 s. Vedoucí bakalářské práce Ing. Petr Petyovský.
- [2] MICROCHIP TECHNOLOGY INC. 2010. *PIC24FJ256GB210 Family Data Sheet: 64/100-Pin, 16-Bit Flash Microcontrollers with USB On-The-Go (OTG)* [online]. [cit. 2015-05-12]. ISBN 978-1-60932-209-0. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/39975a.pdf>
- [3] LF XTAL003074. 2004. In: *HC49/4H Crystals* [online]. [cit. 2015-05-12]. Dostupné z: [http://www.tme.eu/cz/Document/34c7f602e8ebb4973e164b0738844184/hc49\\_4h.pdf](http://www.tme.eu/cz/Document/34c7f602e8ebb4973e164b0738844184/hc49_4h.pdf)
- [4] 3.3V Ultra Miniature SMD HCMOS TCXO/VCTCXO. 2007. In: *FOX924 SERIES* [online]. [cit. 2015-05-12]. Dostupné z: <http://www.foxonline.com/pdfs/fox923CH.pdf>
- [5] RFM23B FSK transceiver. HOPE Microelectronic [online]. [2006] [cit. 2015-05-12]. Dostupné z: [http://www.hoperf.com/rf/fsk\\_module/RFM23B.htm](http://www.hoperf.com/rf/fsk_module/RFM23B.htm)
- [6] RAYSTAR OPTRONICS. 2005. *RX2004A-FHW* [online]. 37 s. [cit. 2015-05-12]. Dostupné také z: <http://datasheet.octopart.com/RX2004A-FHW-Raystar-datasheet-34202234.pdf>
- [7] HITACHI, LTD. 1998. *HD44780U: Dot Matrix Liquid Crystal Display Controller/Driver* [online]. 60 s. [cit. 2015-05-12]. Dostupné také z: <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
- [8] SOLOMON SYSTECH LTD. 2011. *SSD1803A: 100 x 34 STN LCD Segment / Common Mono Driver with Controller* [online]. 69 s. [cit. 2015-05-12]. Dostupné také z: [http://www.lcd-module.de/fileadmin/eng/pdf/zubehoer/ssd1803a\\_2\\_0.pdf](http://www.lcd-module.de/fileadmin/eng/pdf/zubehoer/ssd1803a_2_0.pdf)
- [9] SAMSUNG SDI CO., LTD. 2008. *ICR18650-22F: Lithium-ion Rechargeable Cell* [online]. [cit. 2015-05-12]. Dostupné také z: <http://www.tme.eu/cz/Document/71b610093f14a53a8417944008628183/ICR18650-22F.pdf>
- [10] MICROCHIP TECHNOLOGY INC. 2011. *MCP73837/8* [online]. [cit. 2015-05-12]. ISBN 978-1-61341-885-7. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/22071B.pdf>
- [11] CISCO SYSTEMS. 2011. *Cisco Connected Grid Switch Software Configuration Guide* [online]. [cit. 2015-05-12]. Dostupné také z: [http://www.cisco.com/c/en/us/td/docs/switches/connectedgrid/cgs2520/software/release/12\\_2\\_58\\_ey/configuration/cgs\\_2520\\_swcg.pdf](http://www.cisco.com/c/en/us/td/docs/switches/connectedgrid/cgs2520/software/release/12_2_58_ey/configuration/cgs_2520_swcg.pdf)
- [12] *IEEE Std 802.11™-2012* [online]. 2012. [cit. 2015-05-12]. ISBN 978-0-7381-7245-3 STDPD97218. Dostupné z: <http://standards.ieee.org/getieee802/download/802.11-2012.pdf>
- [13] *Sekvenční logika - debouncer, čítače, měření doby stisknutí tlačítka* [online]. 2014. [cit. 2015-05-12]. Dostupné také z: [http://www.umel.feec.vutbr.cz/bdio/cviceni/uloha\\_09.pdf](http://www.umel.feec.vutbr.cz/bdio/cviceni/uloha_09.pdf)

- [14] BU-409: Charging Lithium-ion. 2010. *Battery University* [online]. [cit. 2015-05-12]. Dostupné z: [http://batteryuniversity.com/learn/article/charging\\_lithium\\_ion\\_batteries](http://batteryuniversity.com/learn/article/charging_lithium_ion_batteries)
- [15] SANDISK. 2007. *SanDisk SD Card Product Family: OEM Product Manual* [online]. [cit. 2015-05-12]. Dostupné také z: <http://media.digikey.com/pdf/Data%20Sheets/M-Systems%20Inc%20PDFs/SD%20Card%20Prod%20Family%20OEM%20Manual.pdf>
- [16] MICROCHIP TECHNOLOGY. 2014. *MPLAB® XC16 C Compiler: User's Guide* [online]. [cit. 2015-05-12]. ISBN 978-1-63276-913-8. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/50002071E.pdf>
- [17] MPLAB® XC Compilers. 2014. *Microchip Technology Inc* [online]. [cit. 2015-05-14]. Dostupné z: [http://www.microchip.com/pagehandler/en\\_us/devtools/mplabxc/](http://www.microchip.com/pagehandler/en_us/devtools/mplabxc/)
- [18] PICkit 3 In-Circuit Debugger. 2014. *Microchip Technology Inc* [online]. [cit. 2015-05-14]. Dostupné z: <http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=PG164130>
- [19] MPLAB® X Integrated Development Environment (IDE). 2014. *Microchip Technology Inc* [online]. [cit. 2015-05-14]. Dostupné z: <http://www.microchip.com/pagehandler/en-us/family/mplabx/>
- [20] *PIC24FJ64GA004 FAMILY: 28/44-Pin General Purpose, 16-Bit Flash Microcontrollers* [online]. 2010 [cit. 2015-05-18]. ISBN 978-1-62077-201-0. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/39881e.pdf>
- [21] SILICON LABS. *CMEMS OSCILLATOR* [online]. 2013 [cit. 2015-05-18]. Dostupné také z: <https://www.silabs.com/Support%20Documents/TechnicalDocs/Si504.pdf>
- [22] Požární sport. *HZSČR: Kraj Vysočina* [online]. 2012 [cit. 2015-05-18]. Dostupné z: <http://www.hzscr.cz/clanek/informacni-servis-pozarni-sport-pozarni-sport.aspx>

# Příloha 1

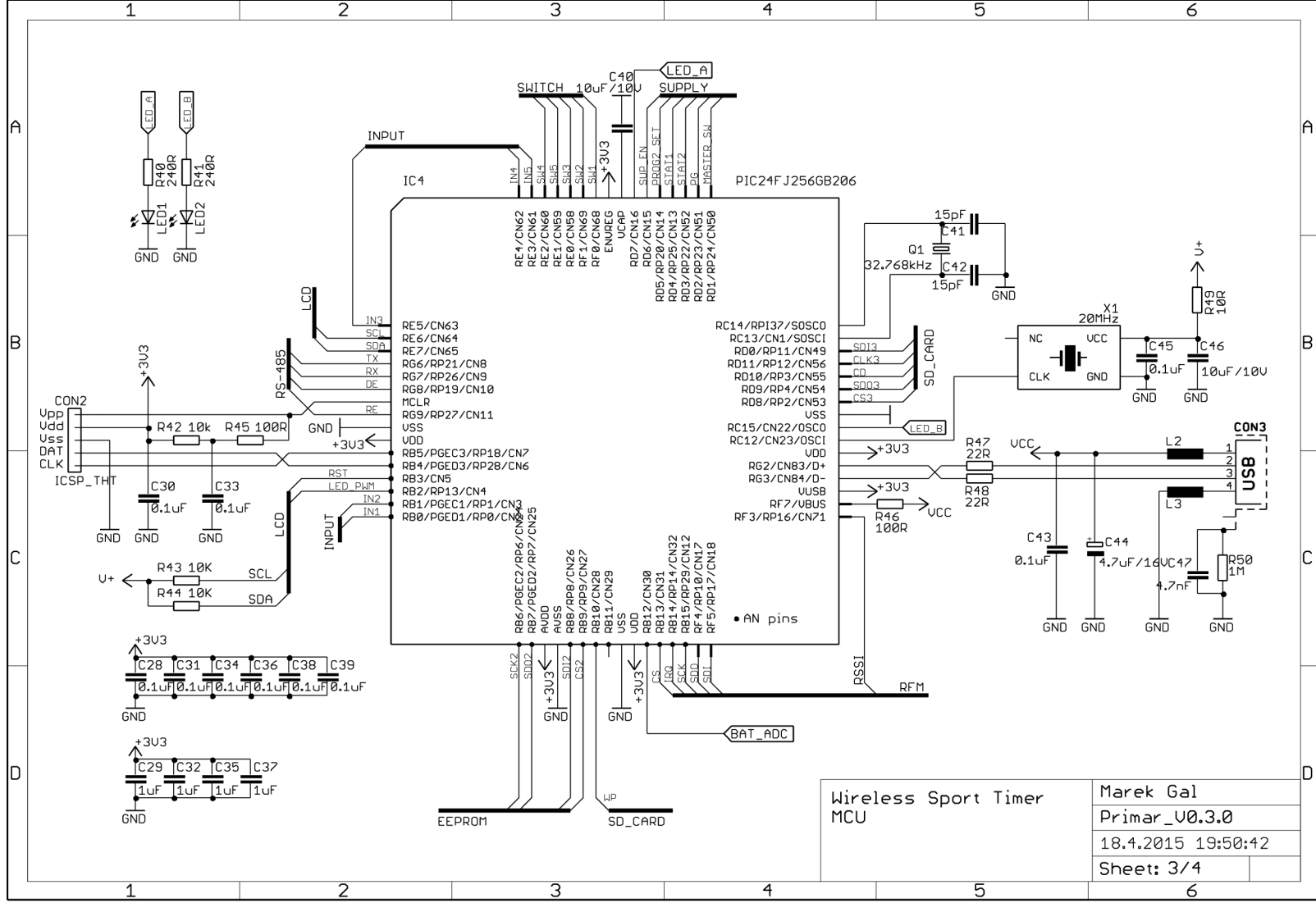
## OBSAH CD

Příložené CD obsahuje:

- Zdrojové kódy pro primární a sekundární jednotku
- Video demonstrující funkčnost realizace
- Zdrojový text bakalářské práce
- Výsledný text bakalářské práce (PDF)
- Fotodokumentace použitých jednotek i praktického ověření

# Příloha 2

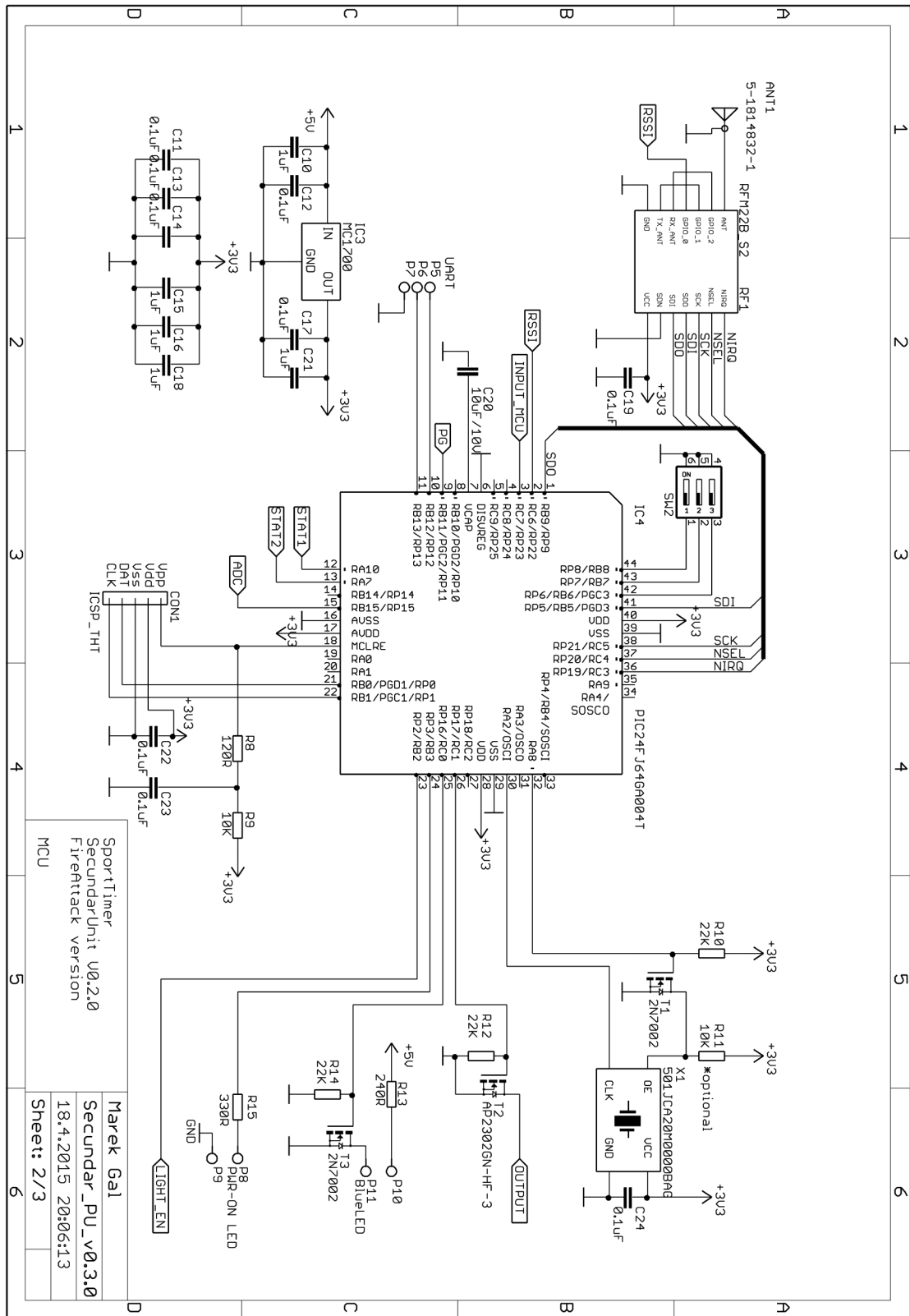
## Část schématu primární jednotky



Wireless Sport Timer	Marek Gal
MCU	Primar_V0.3.0
	18.4.2015 19:50:42
	Sheet: 3/4

# Příloha 3

## Část schématu sekundární jednotky



Sportimer  
 SecunderUnit V0.2.0  
 Firetrack version  
 Marek Gal  
 Secunder\_PU\_v0.3.0  
 18.4.2015 20:06:13  
 Sheet: 2/3

## Příloha 4

### Praktické ověření časoměrného systému



**Příloha 1-Upevnění sekundární jednotky na koncové zařízení**



**Příloha 2-Spouštěcí mechanismus**





**Příloha 3-Primární jednotka připravená k měření**



**Příloha 4-Protnuté koncové zařízení (vpravo)**

The screenshot shows a window titled "15-05-15.CSV (F:\VYSL\_CAS) - GVIM". The menu bar includes "Soubor", "Úpravy", "Nástroje", "Syntaxe", "Buffery", "Okna", and "Nápověda". The toolbar contains various icons for file operations and editing. The main text area displays the following content:

```
01, 04444 , 00, ---:---:---, k
02, 03333 , 00, ---:---:---, k

01, 04444 , 00, ---:---:---, k
02, 03333 , 00, ---:---:---, k

01, 04444 , 00, 00:00:17:90, k
02, 03333 , 00, 00:00:17:49, k

01, 04444 , 00, ---:---:---, k
02, 03333 , 00, ---:---:---, k

01, 04444 , 00, 00:00:18:96, k
02, 03333 , 00, 00:00:18:00, k

01, 04444 , 00, 00:00:18:52, k
02, 03333 , 00, 00:00:18:39, k

01, 04444 , 00, 00:00:23:05, k
02, 03333 , 00, 00:00:22:76, k

~
~
~
"F:\VYSL_CAS\15-05-15.CSV" 21L, 476C
```

**Příloha 5-Záznam měření na SD kartě (otevřeno programem GVIM)**