

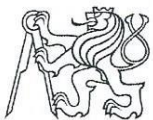
**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**  
**FAKULTA ELEKTROTECHNICKÁ**  
**Katedra měření**

**DIPLOMOVÁ PRÁCE**

**Měření a regulace osvětlení v interiéru**

**2014**

**Bc. IVO TRNKA**



## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Ivo Trnka**

Program: **Kybernetika a robotika**  
Obor: **Senzory a přístrojová technika**

Název tématu česky: **Měření a regulace osvětlení v interiéru**

Název tématu anglicky: **Interior Illumination Measurement and Control**

### Pokyny pro vypracování:

Navrhněte a realizujte systém pro vyhodnocení světelných podmínek a následnou regulaci světelných zdrojů v monitorované místnosti v závislosti na venkovních světelných podmínkách. Systém bude monitorovat a vyhodnocovat osvětlení interiéru a zajišťovat např. minimální úroveň osvětlení, která je dána normou pro světelnou hygienu. Venkovní senzor bude napájen bateriově a dobíjen z fotovoltaického článku. Součástí řídicí jednotky bude ovládací panel, umožňující zásah uživatele, a zobrazovač. Přenos naměřených a kalibračních dat bude probíhat bezdrátově. Systém bude dále podporovat stávající sběrníkové systémy, komunikující s prvky pro ovládání světelných zdrojů.

### Seznam odborné literatury:

- [1] Šubrt, V.: Mikrokontroléry Atmel AVR - vývojové prostředí. BEN, Praha 2002.
- [2] Vedral, J., Fischer, J.: Elektronické obvody pro měřicí techniku. ČVUT, Praha 1999.
- [3] Česká technická norma ČSN EN 12464-1: Světlo a osvětlení. 2004.


Vedoucí diplomové práce: **Ing. Jakub Svatoš**

Datum zadání diplomové práce: **1. října 2013**

Platnost zadání do<sup>1</sup>: **23. ledna 2015**

  
Prof. Ing. Vladimír Haasz, CSc.  
vedoucí katedry



  
Prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 1. 10. 2013

<sup>1</sup> Platnost zadání je omezena na dobu tří následujících semestrů.

## **Čestné prohlášení**

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a použil k tomu úplný výčet citací z níže uvedené literatury. Nemám námitek proti půjčování nebo zveřejňování mé diplomové práce nebo jejích částí a využití výsledků této práce elektrotechnickou fakultou ČVUT se souhlasem katedry měření.

Datum .....

Podpis .....



## **Anotace**

Úkolem diplomové práce je vytvořit systém pro vyhodnocení světelných podmínek a na základě těchto údajů regulovat osvětlení v místnosti, případně objektu. Požadavkem na systém je, aby jej bylo možné zasadit do stávajících instalací inteligentních budov s ohledem na minimální stavební úpravy.

První část práce je zaměřena na studium, teoretickou přípravu a analýzu možností sběru vstupních dat, jejich interpretaci, vyhodnocení a následnou regulaci světelných zdrojů. V této části je také věnována větší pozornost na komunikaci celku se systémy moderních osvětlovacích zařízení.

Následující část se zabývá vývojem hardwaru a softwaru. Kapitoly, popisující vývoj, jsou zaměřeny na výběr vývojových prostředků a jejich aplikování na postupy z teoretické části. Obsahem těchto kapitol jsou také elektrická schémata a vývojové diagramy.

Závěr práce je dedikován kalibraci celého systému, popisu uživatelského ovládání a hodnocení použitých prostředků a postupů.

## **Annotation**

The goal of the thesis is to develop a system for evaluating the data from current light conditions and control lighting in a room or building based on this data. One requirement for the system is the ability to be fitted into an existing installation of intelligent building with minimum construction work.

The first section is focused on the study of theoretical preparation and analysis of the possibilities of input data collection, interpretation, evaluation and subsequent regulation of light sources. This section also includes a description of the communication with modern systems of lighting control.

The following section is about hardware and software development. These chapters describing the development are focused on the selection of development tools and their application to the theoretical piece. Electrical schemas and flowcharts are also included.

The final portion of this thesis is dedicated to the calibration of the entire system, description of the user interface, and evaluation of used resources and procedures..



## Poděkování

V první řadě chci poděkovat vedoucímu práce Ing. Jakubu Svatošovi za konzultaci námětu, odborné vedení v průběhu práce, trpělivost při vysvětlování problematiky, ochotu a přátelské jednání.

Děkuji také Ing. Pavlu Hrzinovi, Ph.D. za věcné rady, připomínky a konzultace ohledně fotovoltaiky a nabíjení baterií. Za konzultace na téma Inteligentní budovy a jejich sběrníkové systémy, děkuji dvojici Ing. Jan Včelák, Ph.D. a Ing. Pavel Mlejnek, Ph.D.. Poděkování také patří Ing. Tomáši Vítkovi za zapůjčení programátoru *AVR Dragon* a Ing. Janu Zálešákovi za zapůjčení *DALI* předřadníku.

Velké díky patří mojí rodině, která mi poskytla výborné podmínky při tvorbě práce a podporovala mne po celý průběh studia.

V neposlední řadě děkuji svým kolegům a přátelům za podporu a pomoc v mém studijním úsilí.



## OBSAH

<b>1. ÚVOD.....</b>	<b>6</b>
<b>2. ROZBOR ŘEŠENÍ .....</b>	<b>7</b>
2.1. POPIS ŘEŠENÍ .....	7
2.2. METODY ZÍSKÁVÁNÍ VSTUPNÍCH DAT .....	8
2.2.1. Měření intenzity osvětlení .....	8
2.2.2. Měření okolní teploty.....	9
2.2.3. Měření napětí fotovoltaického článku.....	9
2.3. BEZDRÁTOVÁ KOMUNIKACE .....	10
2.3.1. Výběr komunikačních modulů.....	10
2.3.2. Uvažované komunikační prostředky .....	10
2.3.3. IQRF .....	11
2.4. KOMUNIKACE SE ZDROJÍ SVĚTLA .....	11
2.4.1. Výběr komunikace s akčními prvky.....	12
2.4.2. Digital Addressable Lighting Interface .....	12
2.5. NAPÁJENÍ SENZORU .....	13
2.5.1. Výběr a dobíjení baterií.....	13
2.5.2. Možnosti napájení senzoru .....	14
2.5.3. Zvolená možnost napájení senzoru.....	14
<b>3. SBĚR VSTUPNÍCH DAT A SPECIFIKACE DALI .....</b>	<b>16</b>
3.1. PROSTŘEDKY ZÍSKÁVÁNÍ VSTUPNÍCH DAT .....	16
3.1.1. Avago APDS-9301 .....	16
3.1.2. Dallas DS18B20 .....	17
3.1.3. Měření napětí fotovoltaického článku.....	18
3.2. DIGITAL ADDRESSABLE LIGHTING INTERFACE .....	18
3.2.1. Fyzické parametry DALI sběrnice .....	18
3.2.2. Fyzická vrstva DALI protokolu.....	20
3.2.3. Komunikace po DALI sběrnici.....	21
3.2.4. Adresace a typy příkazů.....	23
<b>4. HARDWARE .....</b>	<b>27</b>
4.1. VÝVOJOVÉ PROSTŘEDKY .....	27
4.1.1. Softwarové prostředky .....	27
4.1.2. Použité součástky.....	27
4.2. VÝVOJOVÉ MODULY .....	27
4.2.1. Napájecí modul.....	28
4.2.2. APDS modul .....	28
4.2.3. DALI modul .....	29
4.3. NAPÁJENÍ SENZORU.....	31
4.3.1. Výběr baterií.....	31
4.3.2. Výběr fotovoltaického článku .....	33
4.3.3. Schéma napájení senzoru .....	33
4.4. PERIFERIE .....	34
4.4.1. Mikroprocesor .....	35
4.4.2. Senzor .....	35
4.4.3. Ústředna .....	35
<b>5. SOFTWARE.....</b>	<b>37</b>
5.1. VÝVOJOVÉ PROSTŘEDKY .....	37
5.1.1. Softwarové prostředky .....	37
5.2. OBSLUHA DALI SBĚRNICE.....	37
5.2.1. Základní princip obsluhy sběrnice.....	38
5.2.2. Inicializace komunikace .....	39
5.2.3. Přijímání dat.....	40
5.2.4. Odesílání dat .....	41



5.2.5.	Obsluha programové knihovny.....	42
5.3.	ZÍSKÁVÁNÍ DAT ZE SNÍMAČŮ .....	43
5.3.1.	Obsluha snímače Avago APDS-9301 .....	43
5.3.2.	Obsluha snímače Dallas DS18B20.....	46
5.3.3.	Měření napětí fotovoltaického článku.....	46
5.4.	BEZDRÁTOVÁ KOMUNIKACE.....	47
5.4.1.	Obsluha modulů IQRF.....	47
5.4.2.	Obsluha MCU.....	48
5.4.3.	Přenášený rámec .....	48
5.5.	KALIBRACE A VYHODNOCOVÁNÍ DAT ZE SNÍMAČŮ .....	50
5.5.1.	Získávání dat ze snímačů .....	50
5.5.2.	Kalibrace snímačů.....	51
5.5.3.	Výpočet aktuálních hodnot na základě kalibrace .....	52
5.6.	REGULACE.....	53
5.6.1.	Použitý regulátor .....	54
5.6.2.	Časování regulátoru .....	55
5.6.3.	Výstup regulátoru .....	55
5.6.4.	Omezení regulátoru .....	56
<b>6.</b>	<b>UŽIVATELSKÉ OVLÁDÁNÍ A TESTOVÁNÍ.....</b>	<b>58</b>
6.1.	SPOUŠTĚNÍ SYSTÉMU .....	58
6.2.	UŽIVATELSKÉ MENU .....	59
6.2.1.	Výběr režimu.....	60
6.2.2.	Nastavení .....	61
6.2.3.	Měření .....	63
6.2.4.	Kalibrace .....	63
6.3.	KALIBRACE SNÍMAČŮ OSVĚTLENÍ Z POHLEDU UŽIVATELE.....	64
6.3.1.	Rozmístění periferií.....	64
6.3.2.	Proces kalibrace .....	65
6.3.3.	Výběr místa měření .....	65
6.4.	TESTOVÁNÍ .....	66
6.4.1.	Testování komunikace po DALI sběrnici .....	66
6.4.2.	Testování bezdrátové komunikace .....	67
6.4.3.	Testování napájení senzoru .....	67
6.4.4.	Zkušební provoz systému .....	67
<b>7.</b>	<b>ZÁVĚR .....</b>	<b>69</b>
<b>8.</b>	<b>POUŽITÁ LITERATURA.....</b>	<b>71</b>
<b>9.</b>	<b>SEZNAM OBRÁZKŮ A TABULEK.....</b>	<b>74</b>
<b>10.</b>	<b>POPIS PŘÍLOH .....</b>	<b>76</b>
<b>PŘÍLOHA A.</b>	<b>ELEKTRICKÁ SCHÉMATA .....</b>	<b>I</b>
A1.	SENZOR .....	I
A2.	ÚSTŘEDNA.....	II
<b>PŘÍLOHA B.</b>	<b>DESKY PLOŠNÝCH SPOJŮ .....</b>	<b>III</b>
B1.	NAPÁJECÍ MODUL.....	III
B2.	APDS MODUL .....	III
B3.	DALI MODUL .....	III
B4.	SENZOR .....	IV
B5.	ÚSTŘEDNA.....	V
<b>PŘÍLOHA C.</b>	<b>SEZNAM SOUČÁSTEK.....</b>	<b>VII</b>
C1.	NAPÁJECÍ MODUL.....	VII
C2.	APDS MODUL .....	VII
C3.	DALI MODUL .....	VII



---

C4.	SENZOR .....	VIII
C5.	ÚSTŘEDNA.....	IX
<b>PŘÍLOHA D. VÝVOJOVÉ DIAGRAMY .....</b>		<b>X</b>
D1.	FUNKCE RECEIVE TICK() .....	X
D2.	FUNKCE GET BIT VALUE() .....	XI
D3.	FUNKCE SEND TICK() .....	XII
D4.	INICIALIZACE A ČTENÍ DAT ZE SNÍMAČE APDS-9301.....	XIII
<b>PŘÍLOHA E. UŽIVATELSKÉ MENU .....</b>		<b>XIV</b>





## 1. ÚVOD

V současné době trendy moderního bydlení ukazují, že pojem automatizovaná domácnost není pouze slovní spojení používané autory vědeckofantastických knih. Čím dál častěji se setkáváme s prvky domácí automatizace v běžném životě. Garáže se otevírají dálkovým ovládním, topení si zapneme odesláním textové zprávy, bezpečnostní systémy ovládáme přes chytré telefony a další. V některých případech se jedná o pohodlí lidí, kteří prostory obývají, v jiných případech jsou prvky domácí automatizace ryze bezpečnostní či doslova nezbytné pro přežití. K takovýmto zařízením patří například automatické požární systémy a různé systémy pro podporu a monitorování životních funkcí.

Nejen problematikou domácí automatizace se zabývá obor nazvaný *Inteligentní budovy*. Tento obor zahrnuje nejen domácnosti, ale i průmyslové prostory, nemocnice, letiště a další. Snahou návrhářů inteligentních budov je, aby části nebo všechny systémy v daném objektu pracovaly s nějakou logikou a tím ušetřili energie, zlepšili životní podmínky, zabezpečily objekt, prováděly kontrolu a monitoring, usnadnili ovládním, umožnili vzdálený přístup nebo se postaraly o automatizaci určitého procesu.

Tato práce je zaměřena na odvětví inteligentních budov, zabývající se Osvětlením a Světelnými podmínkami. Cílem práce je navrhnout systém, který vyhodnotí venkovní a vnitřní světelné podmínky určitého prostoru a na základě těchto dat provede automatickou regulaci vnitřních světelných zdrojů. Jako světelný zdroj si lze představit umělé osvětlení místnosti nebo venkovní osvětlení regulované žaluziemi či roletami.

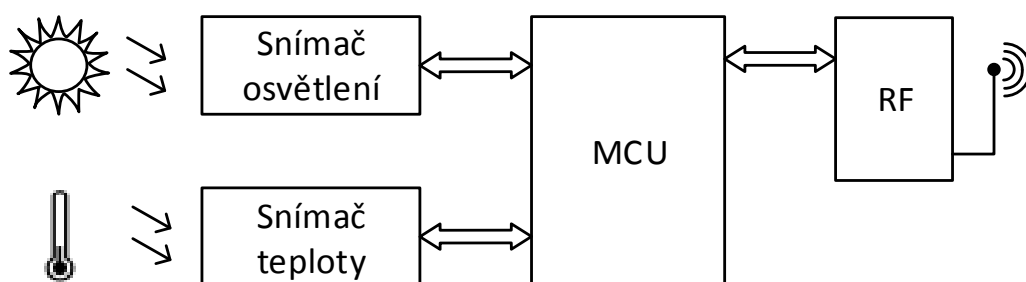
Systém je navržený tak, aby se dal zasadit jak do nově vznikajících prostor, tak i do stávajících objektů. Z tohoto důvodu senzor komunikuje s řídicí jednotkou bezdrátově, je bateriově napájen a dobíjen z fotovoltaického článku.

## 2. ROZBOR ŘEŠENÍ

Před rozvržením celého systému bylo nejprve nutné vyřešit logiku celého systému. To zahrnovalo volbu snímačů neelektrických veličin, komunikačních prostředků, řídicích jednotek a v neposlední řadě také způsobů napájení. Po té, co byly stanoveny a konzultovány stěžní body celého systému, přišly na řadu dílčí části. Do této kategorie se dá zahrnout obvodové řešení, hardwarová realizace a programové vybavení.

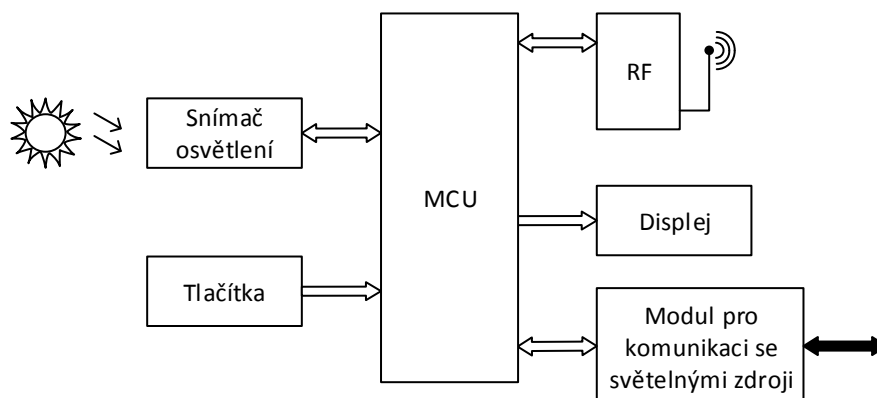
### 2.1. POPIS ŘEŠENÍ

Úkolem práce je navrhnout dvě periferie – *senzor* a *ústřednu*. Úkolem senzoru je sběr vstupních dat a jejich odeslání do ústředny. Blokové schéma senzoru znázorňuje Obr. 2.1. Další podstatnou částí senzoru je systém jeho napájení, jemuž je věnována kap. 2.5.



Obr. 2.1, Blokové schéma senzoru

Data ze senzoru jsou v ústředně dále zpracována a vyhodnocena podle aktuálního režimu nastavení. Dalším úkolem této periferie je sběr vstupních dat v místě, kde je ústředna instalována. Ústředna má na starosti také logiku celku. Uchovává všechny stavové veličiny a řídí proces regulace. Napojení a komunikace s inteligentními osvětlovacími systémy jsou výstupem celého systému. Blokové schéma na Obr. 2.2 popisuje dílčí části ústředny.



Obr. 2.2, Blokové schéma ústředny

## 2.2. METODY ZÍSKÁVÁNÍ VSTUPNÍCH DAT

Pro správnou funkci systému, je nejprve nutné získat informaci o okolním osvětlení. Zkoumaná veličina je v tomto případě *intenzita osvětlení* a rozborem metod jejího měření se zabývá kap. 2.2.1. Pro regulaci světelných zdrojů je údaj o intenzitě osvětlení stěženi. Pro budoucí rozšíření systému disponuje senzor schopností měření okolní teploty, kap. 2.2.2, a napětí na fotovoltaickém článku, kap. 2.2.3.

### 2.2.1. Měření intenzity osvětlení

Měření intenzity osvětlení  $E$  lze provést několika způsoby. Zadání práce specifikuje, že systém má umět udržovat minimální úroveň osvětlení, která je daná normou pro světelnou hygienu [3]. Minimální úrovně osvětlení jsou definovány v jednotkách *Lux* ( $lx$ ). Jedná se o fotometrickou veličinu a je definována jako podíl *svítivého toku*  $\Delta\Phi$  a *obsahu plochy*  $\Delta S$ , na kterou tento svítivý tok dopadá (2.1).

$$E = \frac{\Delta\Phi}{\Delta S} \left( \frac{lm}{m^2} = lx \right) \quad (2.1)$$

Pro měření této fotometrické veličiny se využívá *luxmetr*. Aby byla dodržena podmínka, že intenzita osvětlení v monitorované místnosti bude s danou přesností nastavena na hodnotu v luxech, je nutné provádět měření intenzity osvětlení, na jehož výstupu je hodnota v luxech.

Z počátku bylo zamýšleno použití některých odporových či polovodičových prvků, jejichž specifické vlastnosti se mění v závislosti na intenzitě osvětlení. Konstrukčně nejjednodušší se zdálo být zapojení s fotorezistory. Problém fotorezistoru je, že



závislost jeho elektrického odporu na intenzitě osvětlení není lineární. Tento problém by se dal řešit samostatným zapojením fotorezistorů do děliče napětí s různě odstupňovanými předřadnými odpory.

Vznikly by zde však další problémy s přesností součástek, teplotní závislostí a kalibrací celého systému měření. Výsledným řešením se stal integrovaný obvod *APDS-9301* [19], který provádí měření intenzity osvětlení a jeho číslicový výstup se nechá přepočítat do jednotek luxů.

### **2.2.2. Měření okolní teploty**

Měření teploty je prováděno pouze na senzoru. Důvodem je možné budoucí rozšíření systému. Předpokladem je, že senzor bude umístěn vně budovy. Množství a intenzita slunečního záření je v zimních měsících nižší. To má za následek pokles výkonu fotovoltaického článku. S klesající teplotou se také snižuje výkon baterií. Proto, aby se v budoucnu nechaly sledovat trendy těchto změn v závislosti na teplotě, je zapotřebí teplotu snímat. Dalším rozšířením systému na základě informace o teplotě je v jeho možném bezdrátovém připojení k systémům regulujícím teplotu. V takovém případě by stačila úprava pouze softwaru systému.

Podobně jako u měření intenzity osvětlení v kap. 2.2.1, je okolní teplota měřena integrovaným obvodem s číslicovým výstupem. Použitý snímač teploty typu *DS18B20* [20] provádí měření v rozsahu od  $-55^{\circ}\text{C}$  do  $+125^{\circ}\text{C}$ . Přesnost měření v rozsahu  $-10^{\circ}\text{C}$  až  $+85^{\circ}\text{C}$  je výrobcem garantovaná a je  $\pm 0,5^{\circ}\text{C}$ , což pro je pro danou aplikaci postačující.

### **2.2.3. Měření napětí fotovoltaického článku**

Měření napětí na fotovoltaickém článku tvoří spolu s měřením teploty základ pro budoucí rozšíření systému. Cílem měření napětí na článku je zjištění aktuální hodnoty výkonu článku. Ten lze určit na základě vztahu mezi napětím článku a intenzitou záření, který udává výrobce prostřednictvím grafu. Na základě množství dodaného výkonu článkem za jednotku času je například možné lépe navrhnout kapacitu baterií.

Při výpočtu výkonu článku je zapotřebí také zohlednit jeho zatížení, které se nechá odvodit z celkové spotřeby senzoru.



### **2.3. BEZDRÁTOVÁ KOMUNIKACE**

Jedním z úkolů práce bylo, aby systém bylo možné použít i ve stávajících objektech bez nutnosti stavebních úprav. Tento úkol klade požadavky zejména na senzor a to konkrétně na napájení senzoru, kap. 2.5, a komunikaci s ústřednou. Běžné sběrnice, využívající přenos informací po vodičích, nešlo v tomto případě použít. Ve většině případů by to totiž znamenalo nutnost stavebního zásahu do objektu. Přenos dat mezi oběma periferiemi tedy probíhá bezdrátově.

#### **2.3.1. Výběr komunikačních modulů**

Prostředků, jak bezdrátově komunikovat mezi dvěma periferiemi se nabízí mnoho. V úvahu je potřeba brát několik faktorů. Prvním z uvažovaných faktorů je vzdálenost, na kterou komunikace bude probíhat. Jelikož je systém navržený tak, aby venkovní senzor byl umístěn vedle oken místnosti, ve které se bude nacházet ústředna, je předpoklad takový, že obě periferie od sebe nebudou vzdáleny více jak 100 m. Dalším prvkem, který je nutno zvážit a souvisí s předchozím, jsou překážky mezi oběma periferiemi. Uvažováno je, že tuto překážku bude tvořit zeď nebo okno. V tomto případě je brána v potaz vzdálenost, kterou výrobce uvádí, jako dosah v zástavbě.

Výše uvedené parametry byly při volbě způsobu bezdrátové komunikace stěžejní. Další faktory, které byly vzaty v potaz při výběru komunikačních modulů, byly spotřeba elektrické energie, cena, náročnost obsluhy, velikost a potřeba další obslužné elektroniky.

#### **2.3.2. Uvažované komunikační prostředky**

Po zvážení všech podmínek pro výběr komunikačních modulů, uvedených v kap. 2.3.1, byly uvažovány následující prostředky pro bezdrátovou komunikaci. U každého prostředku jsou v Tab. 2.1 popsány vybrané parametry dané volby spolu s výhodami daného použití.



Tab. 2.1, Porovnání bezdrátových komunikačních prostředků [4]

Název	Wi-Fi	Bluetooth	ZigBee	IQRF
Standard	802.11b	802.15.1	802.15.4	IQRF
Vhodné pro	Web, Video	Hlas	Monitorování	Aut. budov
Výdrž na baterii (dny)	0,5 – 5	1 – 7	100 – 1000+	1000+
Počet zařízení v síti	32	7	65 000	65 000
Rychlost přenosu (Kb/s)	11 000+	720	20 – 250	600
Dosah (m)	1 – 100	1 – 10+	1 – 100+	700
Výhody	Rychlost, Flexibilita	Cena, Dostupnost	Spolehlivost, Cena	Spolehlivost, Jednoduchost

### 2.3.3. IQRF

Výslednou volbou bezdrátové komunikace se stalo *IQRF*. Tato česká platforma byla zvolena, neboť se jedná o cenově dostupný, spolehlivý a parametry více než dostačující způsob bezdrátové komunikace. Výhodou této volby je snadná implementace, nízká proudová spotřeba, flexibilita a snadná rozšiřitelnost.

Platforma využívá frekvence v bezplatném pásmu *ISM*, tj. v EU 868 MHz. Je kladen důraz na velmi nízkou spotřebu, proto je v nabídce mnoho úsporných režimů kontroléru. Moduly komunikují s *MCU* prostřednictvím sběrnic *UART*, *SPI* a *I<sup>2</sup>C*. Další výhodou je, že výrobce poskytuje mnoho příkladů použití včetně zdrojových kódů, tudíž bylo velmi snadné s *IQRF* začít.

## 2.4. KOMUNIKACE SE ZDROJI SVĚTLA

Ovládání akčních prvků je výstupem celého systému. Pro účely práce je vhodné si světelné zdroje rozdělit na *vnější* a *vnitřní*. Toto rozdělení se vztahuje na místnost či objekt, ve kterém se provádí regulace osvětlení. Vnější zdroje světla mohou být jak *přírodní*, tak i *umělé*. Jedná se o světlo, které proniká do objektu skrz otvory v konstrukci, jako jsou okna. Například se jedná o sluneční svit nebo pouliční osvětlení. Vnitřní světelné zdroje se nachází přímo v daném objektu a jsou jimi například lampy a stropní osvětlení.

Důvodem rozdělení je zejména odlišný princip regulace. Zatímco u vnitřních zdrojů světla se reguluje přímo výkon zdroje, u vnějších zdrojů se výkon ovlivnit nedá. Regulace tak probíhá formou clon, jakými jsou žaluzie, rolety a další stínící prvky.



### **2.4.1. Výběr komunikace s akčními prvky**

Celý systém byl navrhován tak, aby ho by možné zasadit do moderních instalací inteligentních budov. Tento fakt vyžaduje použití takových komunikačních prostředků, jež jsou podporovány výrobci zařízení pro inteligentní budovy. Na trhu je však výrobců mnoho a ne všechny výrobky jsou vzájemně kompatibilní, proto byla potřeba najít co nejuniverzálnější způsob komunikace s ohledem na aplikaci. Velká část výrobců se snaží používat standardizované sběrnice systémy, jako jsou například *KNX*, *LON* či *iNels*. Jednou z nejrozšířenějších sběrnic je *KNX*. Tato sběrnice je navržena tak, aby byla nezávislá na hardwarové platformě, a je sdružením tří technologií [5]:

- *EHS* (European Home System)
- *EIB* (European Installation Bus)
- *BâtiBUS*

Přenosovým médiem je převážně kroucený dvoužilový kabel, ale standard je definován také pro komunikaci po elektrické síti, radiový přenos, infračervenou komunikaci a ethernet. Výhodami *KNX* jsou jednoduchost instalace, snadná rozšiřitelnost a robustnost.

Komunikační protokol zvládne obsloužit mnoho prvků s rozdílnými funkcemi. Na sběrnici je možno připojit systémy řízení vytápění a osvětlení, elektrické zabezpečovací systémy, elektrické požární systémy a další. Díky této univerzálnosti nachází sběrnice systém *KNX* uplatnění v široké škále aplikací. Jeho nevýhodou je, že komunikační protokol má pro jednoduché aplikace složitý rámec. Jednoduchou aplikací se rozumí například nastavení úrovně teploty nebo osvětlení, která probíhá výhradně mezi ústřednou a řízeným prvkem.

Řešením je použití jednoduššího sběrnice systému, který se na *KNX* sběrnici nechá připojit. Ve své podstatě se jedná o jednoúčelovou sběrnici, která je navržena pouze pro regulaci a řízení specifické veličiny nebo subsystému. Takovým řešením je, pro systémy ovládající osvětlení, sběrnice systém *Digital Addressable Lighting Interface* (dále *DALI*).

### **2.4.2. Digital Addressable Lighting Interface**

Tento komunikační protokol je možno od roku 2002 využívat bezplatně a není tedy vázán žádnou licencí. To napomohlo k jeho rychlému rozšíření mezi výrobce



osvětlovací systémů. Požadavkem těchto výrobců bylo vytvořit jednotný komunikační protokol pro řízení světelných zařízení [7].

Jedná se o dvouvodičovou sběrnici typu *master-slave*, kde *master* zařízení ovládají *slave* prvky pomocí 2bytových zpráv. Na sběrnici může být připojeno více *master* zařízení, jejichž počet je omezen na 64, a komunikace mezi nimi probíhá prostřednictvím 3bytových zpráv. Rychlost této sběrnice je 1200 bit/s a využívá bifázové kódování. Stejně jako počet *master* zařízení je počet *slave* zařízení na jedné síti omezen na 64. Ty mohou být ovládány buď jednotlivě anebo skupinově.

## 2.5. NAPÁJENÍ SENZORU

Jedním z požadavků na senzor je, aby byl schopný dlouhodobě samostatně fungovat bez potřeby připojení napájecího zdroje. Ke splnění požadavku musí být napájecí zdroj součástí senzoru. Napájení také musí zajistit provoz bez ohledu na okolní světelné podmínky.

Z výše uvedených požadavků vyplývá, že je nutné použít akumulátor energie, tedy baterie. Z hlediska dlouhodobého fungování je také potřeba vyřešit dobíjení baterií. Předpokladem je, že senzor bude umístěn vně budovy a tudíž na něj bude dopadat sluneční záření. Jako vhodné řešení napájení senzoru jsou tedy baterie dobíjené z fotovoltaického článku.

### 2.5.1. Výběr a dobíjení baterií

Pro každý typ bateriových článků existuje vhodná metoda jejich nabíjení. V současnosti jsou v podobných aplikacích nejvíce využívány baterie typu *NiCd*, *NiMH* a *Li-ion*. Pro každý z těchto uvažovaných typů baterií bylo nutné uvážit její životnost, samovybíjení, náročnost nabíjení a citlivost na přebíjení.

- NiCd a NiMH baterie:

Jako nejvhodnější způsob nabíjení baterií, při nabíjení pomocí fotovoltaického článku, je pomalé nabíjení přibližně deseti procenty jmenovitého proudu baterie. U této metody se využívá přerušení nabíjení časovým omezením, které by mělo být přibližně 12 hodin od započetí nabíjení, při úplném vybití článku.





- Li-ion baterie:

Nejčastějším způsobem nabíjení je konstantním napětím, kde je nutno dodržet hodnotu konečného nabíjecího napětí s přesností  $\pm 1\%$ , jinak dochází k výraznému zkrácení životnosti baterie.

Porovnání vlastností bateriových článků se nachází v Tab. 2.2.

Tab. 2.2, Hodnocení vybraných parametrů bateriových článků

Typ baterie	Samovybíjení (%/den)	Počet nabíjecích cyklů
NiCd	1	1000
NiMH	1,5	500
Li-ion	0,5	400

### **2.5.2. Možnosti napájení senzoru**

Postupů jak vyřešit napájení periferie a současně nabíjení baterií je několik. Při výběru metody bylo zejména nutné zvážit složitost návrhu, životnost baterií, velikost fotovoltaického článku a spolehlivost dodávání energie pro správnou činnost senzoru.

V úvahu byla varianta s řízeným nabíjením baterií. Výhodou tohoto způsobu nabíjení je delší životnost baterií, nevýhodou je konstrukční složitost. Další metodou, jak řešit napájení senzoru je metoda časování. V této metodě by byl senzor napájen pouze v pravidelných cyklech, což by přineslo snížení spotřeby, ale problém může nastat při potřebě napájení senzoru po delší dobu, například při jeho kalibraci.

### **2.5.3. Zvolená možnost napájení senzoru**

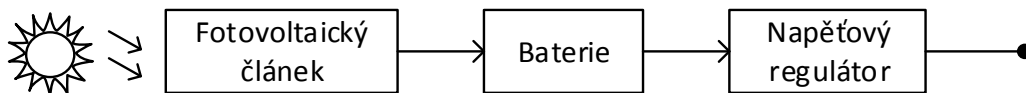
S ohledem na konstrukční jednoduchost a tím pádem i na složitost návrhu byla vybrána varianta napájení senzoru, kdy je fotovoltaický článek permanentně připojen k bateriím. Zdrojem napětíového regulátoru jsou tedy primárně baterie, které jsou, při dostatečném výkonu článku, neustále dobíjeny. Blokové schéma je na Obr. 2.3.

Varianta byla také vybrána, jelikož nároky na velikost a výkon fotovoltaického článku nejsou velké a jsou úměrné předpokládané velikosti celého senzoru.

Nevýhodou metody jsou nároky na volbu baterií. V tomto případě nelze použít baterie, které nesnesou přebíjení a vyžadují přesné odpojení od zdroje nabíjení při dosažení úplné kapacity baterie.



Vhodné baterie pro tuto aplikaci jsou akumulátory s alkalickým elektrolytem, tedy *NiCd* nebo *NiHM*. Nabíjení probíhá metodou nabíjení konstantním proudem. Výpočtem kapacity baterií a také velikosti fotovoltaického článku se zabývá kap. 4.3.



Obr. 2.3, Blokové schéma zvolené metody napájení

### 3. SBĚR VSTUPNÍCH DAT A SPECIFIKACE DALI

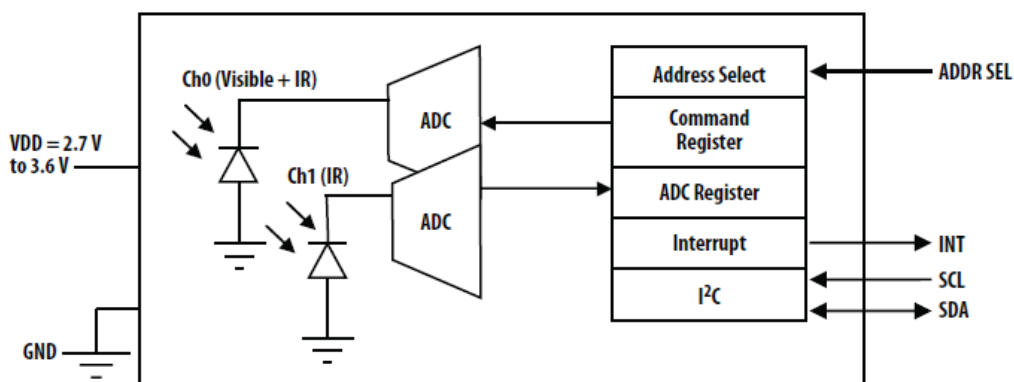
V této části diplomové práce jsou popsány použité metody při získávání vstupních dat. Další část kapitoly je věnována bližšímu popisu komunikace po *DALI* sběrnici. Závěr kapitoly se zabývá použitou metodou napájení senzoru. Výběr použitých prostředků byl založen na teoretickém rozboru, jímž se zabývá kap. 2.

#### 3.1. PROSTŘEDKY ZÍSKÁVÁNÍ VSTUPNÍCH DAT

Tato kapitola se zabývá použitými metodami získáváním vstupních dat a přenosem informací mezi snímači neelektrických veličin a řídicí jednotkou. Vstupní data jsou pak následně vyhodnocována příslušnými bloky programu, které jsou popsány kap. 5.

##### 3.1.1. Avago APDS-9301

Jedná se o digitální snímač intenzity osvětlení, který převádí intenzitu osvětlení na digitální výstup rozhraní *I<sup>2</sup>C*. Zařízení se skládá z jedné širokopásmové (pásmo viditelné a infračervené) a jedné infračervené fotodiody. Dva integrační A/D převodníky zajišťují převod proudu fotodiody, který je úměrný intenzitě záření každého kanálu, na digitální výstup. Tento digitální výstup pak může být vstupem do mikroprocesoru, kde je intenzita osvětlení v luxech odvozena na základě empirického vzorce, popsaného v katalogovém listu [19].



Obr. 3.1, Blokové schéma integrovaného obvodu *APDS-9301*

Integrační A/D převodníky, viz Obr. 3.1, provádějí integraci obou kanálů současně. Po dokončení cyklu převodu jsou výsledky konverze přesunuty do datových registrů (každý kanál má svůj datový registr). Aby nedocházelo ke čtení neplatných dat během

přenosu, jsou zde k dispozici dvě vyrovnávací paměti. Po přenosu dat do datových registrů začíná automaticky další integrační cyklus A/D převodníku.

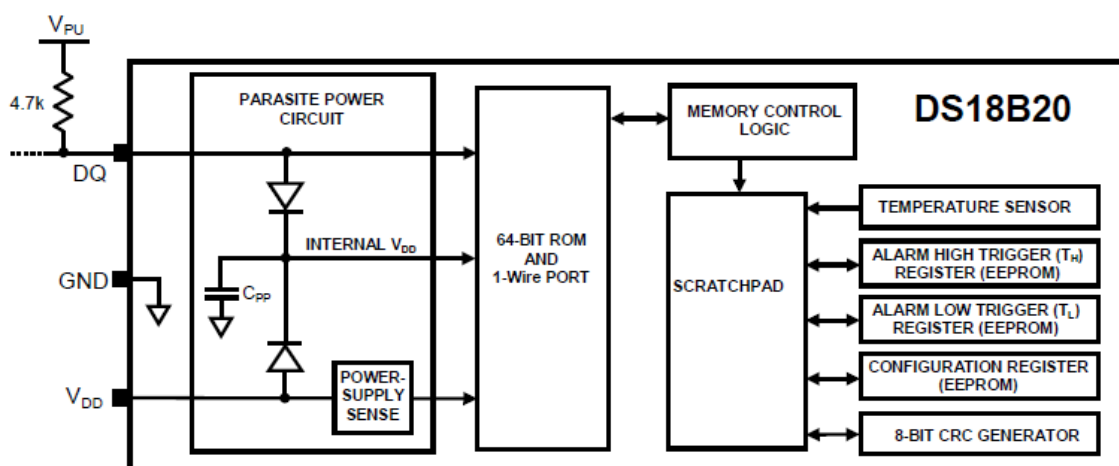
Zařízení je schopno pracovat v rozmezí teplot od  $-30^{\circ}\text{C}$  do  $+85^{\circ}\text{C}$ , je tak přeurlčen pro použití jak uvnitř, tak i vně místností.

Komunikace a ovládání se provádí přes dvou vodičové sériové rozhraní. Rozhraní poskytuje přístup k registrům, jenž slouží k řízení funkcí zařízení a vyčítání výstupních dat. Toto sériové rozhraní je kompatibilní s rychlou  $I^2C$  sběrnici.

### 3.1.2. Dallas DS18B20

Tento digitální teploměr nabízí 9ti až 12ti bitové měření teploty ve stupních Celsia. Zařízení komunikuje po jednovodičové *master-slave* sběrnici *1-Wire bus*. Výhodou u některých aplikací je, že je celý integrovaný obvod možno napájet po této jednovodičové sběrnici, viz. katalogový list [20].

Jádrem integrovaného obvodu je snímač, který přímo převádí změřenou teplotu do digitální podoby. Výchozí rozlišení senzoru je 12ti bitové, což odpovídá teplotnímu rozlišení  $0,0625^{\circ}\text{C}$ . Nejnižší, 9ti bitové rozlišení, odpovídá teplotnímu rozlišení  $0,5^{\circ}\text{C}$ . Po digitální konverzi se naměřená data uloží do 2bytových registrů teploty, nacházejících se v zápisníkové paměti (*scratchpad memory*). Blokové schéma senzoru popisuje Obr. 3.2.



Obr. 3.2, Blokové schéma integrovaného obvodu *DS18B20*

Údaje na výstupu jsou kalibrovány ve stupních Celsia. Teplotní data jsou spolu se znaménkem ukládána v 16ti bitové podobně jako rozšířený dvojkový doplněk čísla



v registrech teploty. Záznam je názorně popsán na Obr. 3.3, kde znak *S* označuje teplotní znaménko.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	$2^3$	$2^2$	$2^1$	$2^0$	$2^1$	$2^2$	$2^3$	$2^4$
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	$2^8$	$2^5$	$2^4$

Obr. 3.3, Příklad interpretace výstupních dat snímače *DS18B20*

### 3.1.3. Měření napětí fotovoltaického článku

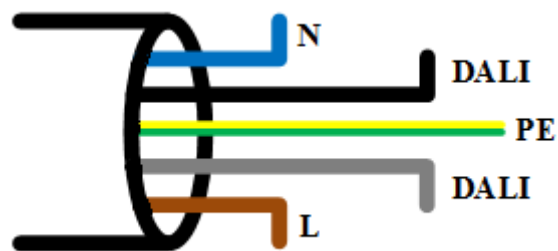
Pro měření napětí fotovoltaického článku je využit vnitřní A/D převodník řídicí jednotky. Jako reference napětí pro převod je použito napájecí napětí. Jelikož je v čase maximálního výkonu fotovoltaického článku jeho napětí vyšší než napětí referenční, je pro měření použit napěťový dělič se stejnými hodnotami rezistorů. Napětí na vstupu A/D převodníku je tudíž poloviční.

## 3.2. DIGITAL ADDRESSABLE LIGHTING INTERFACE

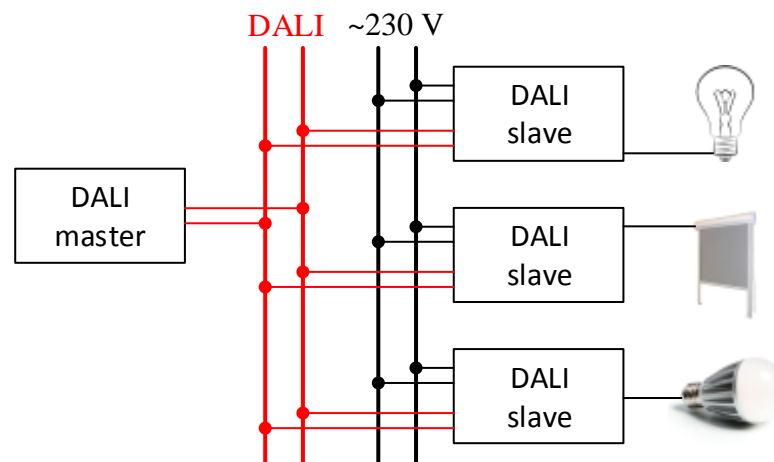
V současnosti není na trhu univerzální řadič *DALI* sběrnice. Je to dáno její specifickou aplikací a také hardwarovými nároky. K sehnání jsou různé moduly, které s *DALI* sběrnici komunikují, ale ty jsou buď drahé, nepoužitelné pro potřeby této práce nebo neumožňují napájení sběrnice. Z důvodu nedostupnosti integrovaného obvodu, který řeší hardwarové a softwarové požadavky tohoto komunikačního standardu, bylo nutné takový modul zkonstruovat. Bylo tedy zapotřebí sestavit *master* modul včetně programového vybavení. Největší inspirací se stala diplomová práce [16], kde bylo za úkol *DALI* protokol implementovat pro mikrokontrolér STM8.

### 3.2.1. Fyzické parametry *DALI* sběrnice

Napěťové požadavky pro *DALI* sběrnici jsou 9,5 V až 22,5 V pro úroveň log. 1 a pro úroveň log. 0 je napěťová úroveň mezi -6,5 V a 6,5 V. Každé zařízení připojené na sběrnici musí snést proud až 250 mA. Dalším požadavkem je galvanické oddělení sběrnice od řídicích a dalších obvodů. Je to dáno způsobem elektroinstalace.

Obr. 3.4, Typická kabeláž *DALI* sběrnice

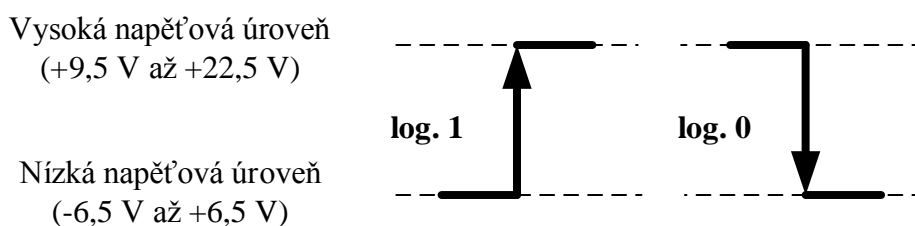
Na Obr. 3.4 je typická kabeláž pro *DALI* zařízení. Jedná se o 5-žilový kabel, který kromě klasických napájecích a ochranných přívodů, obsahuje dva sběrnicové vodiče. Tyto sběrnicové vodiče nemusí být stíněné ani kroucené. Důvodem galvanického oddělení obvodů sběrnice od zbytku systému je možné nesprávné připojení jednotlivých vodičů k zařízení. Je tedy možné, že důsledkem špatného zapojení, se na sběrnici vyskytne fázové napětí, což by mělo za následek zničení nízkonapěťových obvodů systému.

Obr. 3.5, Příklad připojení zařízení na *DALI* sběrnici

Výhoda této elektroinstalace oproti stávajícím “neintelligentním” instalacím spočívá v úspoře kabeláže a nákladech na údržbu. Zatím co u konvenčních světelných zdrojů je potřeba, aby každé zařízení, jehož výkon je potřeba regulovat, mělo přivedeno svojí samostatnou fází, u inteligentní instalace stačí jedna, vhodně dimenzovaná, fáze. Prvky se pak mohou libovolně připojovat na sběrnici a pomocí jejich adres regulovat. Příklad takového zapojení je na Obr. 3.5.

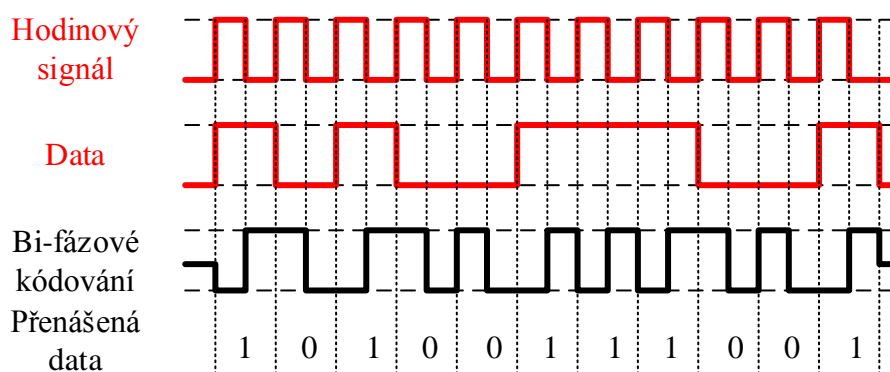
### 3.2.2. Fyzická vrstva DALI protokolu

Komunikační protokol sběrnice využívá bi-fázové (tzv. Manchesterské) kódování, který se řídí normou *IEEE 802.3* [8]. Jedná se o sériový asynchronní protokol, kde každý bit je bi-fázově kódován, vyjma dvou stop bitů, u kterých nedochází ke změně logické úrovně. Úroveň log. 1 je brána jako přechod z nízké do vysoké napěťové úrovně a log. 0 je brána opačně, čili se jedná o přechod z vysoké do nízké napěťové úrovně, přičemž trvání náběžných a sestupných hran by se mělo pohybovat mezi 10  $\mu$ s a 100  $\mu$ s. Příklad logických úrovní se nachází na Obr. 3.6.



Obr. 3.6, Příklad logických úrovní na *DALI* sběrnici

Signálová hrana, nesoucí datovou informaci, se u bi-fázového kódování nachází v každé půlperiodě bitu. V případě, že mají dva po sobě jdoucí bity stejnou logickou hodnotu, je na začátku druhého bitu nutné provést změnu logické úrovně, která slouží k nastavení správné signálové úrovně a nikoliv k přenosu informace. Na Obr. 3.7 je princip tohoto kódování ilustrován pomocí hodinového a datového signálu.

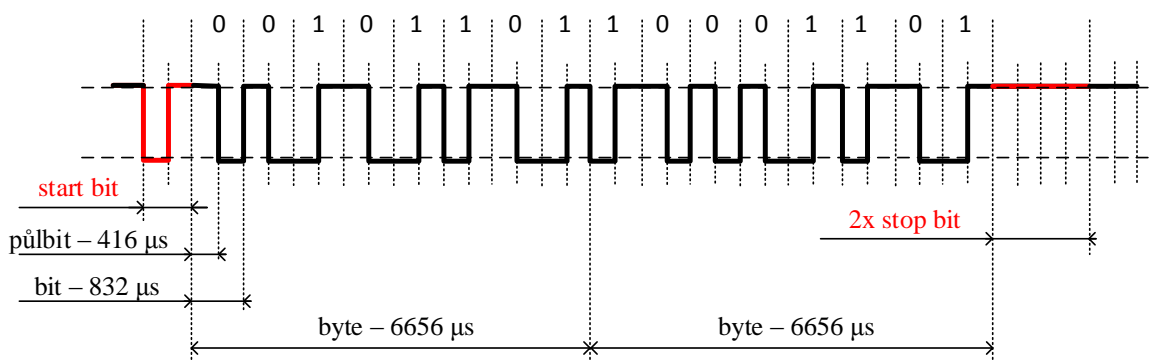


Obr. 3.7, Princip bi-fázového kódování

Změna napěťové úrovně, která neslouží k přenosu informace, se nazývá *fázová hrana*. Případu, kdy dochází ke změně napěťové úrovně z důvodu přenosu informace, se říká *bitová hrana*. Zjednodušeně se tak dá říci, že v každé půlce periody jednoho

bitu, pokud dochází k přenosu dat, se vyskytne bitová hrana. K výskytu fázové hrany dochází pouze mezi dvěma bity se stejnou logickou hodnotou.

Pokud na sběrnici neprobíhá komunikace, tzv. klidový stav, je na sběrnici vysoká napěťová úroveň. Začátek komunikace vždy inicializuje *master* zařízení. Děje se tak pomocí *start bitu*. Tato inicializace spočívá ve vygenerování spádové hrany. Některá zařízení připojená na *DALI* sběrnici, fungují v různých režimech úspory energie a je tak nutné po odeslání *start bitu* chvíli počkat, než bude adresované zařízení schopno komunikovat. Jelikož je doba trvání jednoho bitu  $832 \mu\text{s}$ , je nutné, aby tato rezerva byla minimálně půlka doby trvání jednoho bitu.



Obr. 3.8, Časový průběh 2bytové zprávy na *DALI* sběrnici

Na Obr. 3.8 je znázorněn časový průběh 2bytové zprávy. Po odeslání dat je komunikace ukončena odesláním dvou stop bitů. Každý stop bit má dobu trvání stejnou jako datový bit, tudíž  $832 \mu\text{s}$ . Po ukončení komunikace se sběrnice opět navrácí do klidového stavu.

### 3.2.3. Komunikace po *DALI* sběrnici

Zařízení připojená na sběrnici spolu mohou komunikovat prostřednictvím tří typů komunikačních rámců. Každý komunikační rámec se odlišuje délkou (počtem bitů), dobou ustálení a použitím. Použitím se rozumí, že ne všechna zařízení mohou daný typ rámce vysílat, resp. přijímat. Základní dělení komunikačních rámců je na *Forward* a *Backward*. Rámec nazvaný *Forward* vysílá vždy *master* a je určený pro další *master* zařízení nebo pro zařízení *slave*. Druhý typ rámce, *Backward*, vysílá *slave* a obsahuje

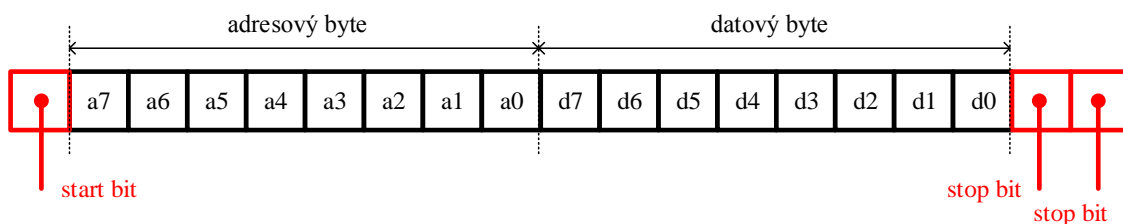




odpověď na požadavek *master* zařízení. Následuje výčet všech tří komunikačních rámců a jejich specifikace.

1. Master → Slave Forward rámeček:

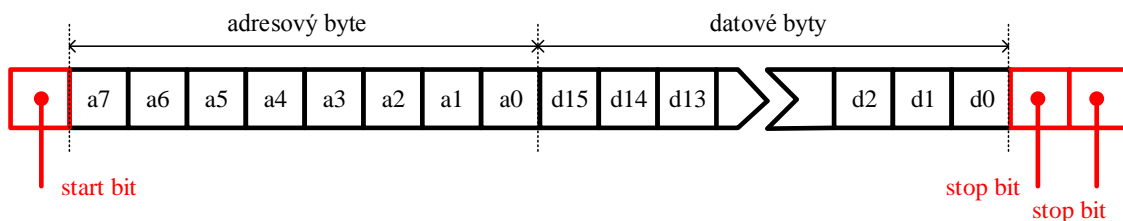
- počet bitů: 19 (1x start bit, 8x adresa, 8x data, 2x stop bit)
- délka trvání: 15,83 ms



Obr. 3.9, *Master → Slave Forward* rámeček

2. Master → Master Forward rámeček:

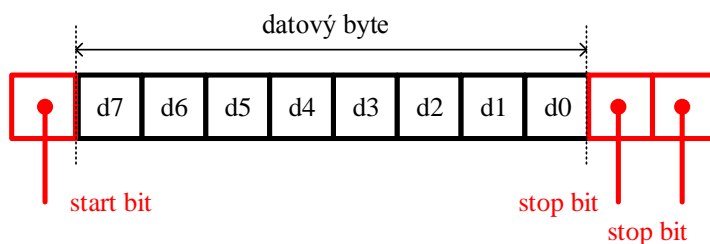
- počet bitů: 27 (1x start bit, 8x adresa, 16x data, 2x stop bit)
- délka trvání: 22,5 ms



Obr. 3.10, *Master → Master Forward* rámeček

3. Slave → Master Backward rámeček:

- počet bitů: 11 (1x start bit, 8x adresa, 2x stop bit)
- délka trvání: 15,83 ms



Obr. 3.11, *Slave → Master Backward* rámeček

Doba ustálení mezi komunikačními rámci závisí na typech jednotlivých rámci a počítá se od konce posledního datového bitu po první bit adresový následujícího rámce. Dalším kritériem pro stanovení doby ustálení je přiřazená přístupová priorita *Forward* rámci, která se stará o minimalizaci kolizí a o zajištění správného pořadí přístupů na sběrnici. Celkem jsou tedy tři případy posloupností dvou rámci, pro které je nutné rozlišit doby ustálení, viz. Obr. 3.12.

1. *Forward* rámec → *Backward* rámec:

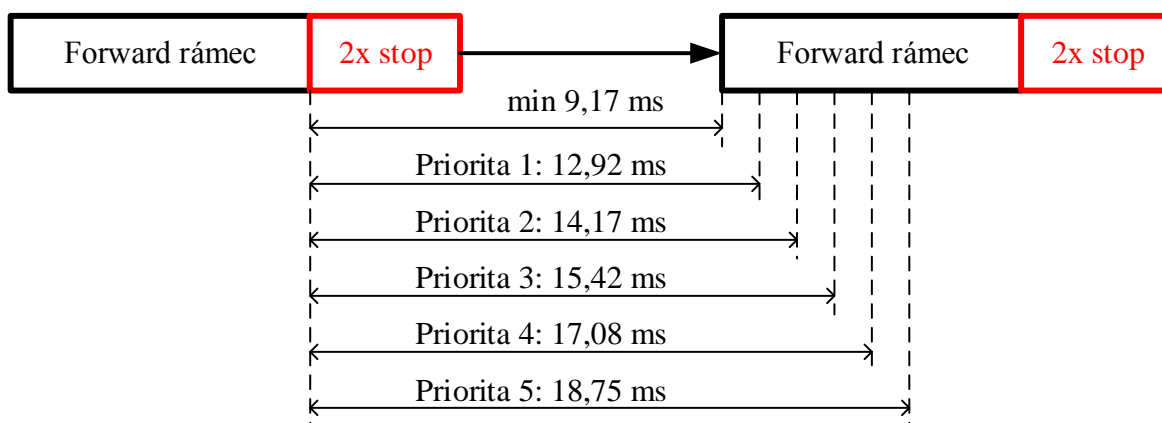
- doba ustálení: 2,92 ms až 9,17 ms  
(prodleva delší jak 9,17 ms je považována za žádnou nebo zápornou odpověď, v závislosti na dotazu)

2. *Backward* rámec → *Forward* rámec:

- doba ustálení: minimálně 9,17 ms

3. *Forward* rámec → *Forward* rámec:

- doba ustálení: minimálně 9,17 ms
- dále se řídí přístupovou prioritou, viz. Obr. 3.12



Obr. 3.12, Doby ustálení mezi *Forward* rámci na *DALI* sběrnici

### 3.2.4. Adresace a typy příkazů

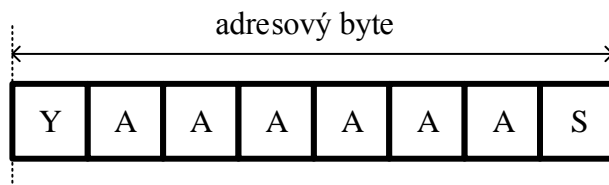
Každý *Forward* rámec, při pomnutí start bitu, začíná adresou zařízení, s nímž chce *master* komunikovat. Z předchozí kap. 3.2.3 vyplývá, že *master* může komunikovat po sběrnici jak se zařízeními typu *slave*, tak s dalšími *master* zařízeními. Zatímco *master*

má svoji adresu pevně stanovenou, *slave* prvkům je adresa přidělována a nazývá se *short adresa*. Celkem existují tři způsoby jak přiřadit adresu *slave* zařízení.

- Přednastavením
- Fyzickým zvolením
- Náhodným zvolením

První typ přiřazení adresy, přednastavením, probíhá pomocí adresových přepínačů přímo na *slave* zařízení. Pro další dva způsoby je již nutný zásah *master* zařízení. Pokud je adresa přiřazena fyzickým zvolením, jedná se o přiřazení konkrétní *short adresy* danému zařízení v *DALI* síti. Takovéto zařízení po té musí být vynecháno z procesu přiřazování adres náhodným zvolením. U toho procesu se nejprve všechna *slave* zařízení přepnou do adresacího módu příkazem *INITIALISE*. Dalším příkazem *RANDOMISE* všechny *slave* prvky vygenerují 24bitové náhodné číslo. Následně *master* hledá zařízení s nejmenším náhodným číslem, jemuž po té přiřadí první *short adresu*, po přiřazení je *slave* vyřazen z procesu adresace příkazem *WITHDRAW*. Po přiřazení adres všem *slave* prvkům proces adresace končí příkazem *TERMINATE*.

Obecný tvar adresového bytu je znázorněn na Obr. 3.13.



Obr. 3.13, Obecný tvar adresového bytu *DALI* zařízení

kde:

bit **Y** = 0 ..... značí že *short adresa* **AAAAAA** patří jedinému *slave* prvku

bit **Y** = 1 ..... adresa náleží skupině *slave* prvků, skupina *slave* prvků má vždy tvar adresy **100AAAAA** (maximálně 16 skupin)

bit **S** = 0 ..... následující byte nastavuje přímo hladinu osvětlení

bit **S** = 1 ..... následující byte obsahuje jiný příkaz

Komunikační protokol ještě umožňuje i jiné než obecné tvary adres. Jedním takovým je *Broadcast*, na který musí reagovat všechna *slave* zařízení a jeho tvar



je **1111111S**. Adresový byte ve formě **101CCCC1** nebo **110CCCC1** je určen pro speciální skupinu příkazů, kde písmena **C** určují bity speciálních příkazů.

Standardní příkazy, které následují za adresovým bytem ve *Forward* rámci, se rozdělují do několika skupin podle charakteru příkazu a očekávané reakci *slave* prvku.

### 1. Příkazy nastavující přímo úroveň osvětlení:

Tento typ příkazů předpokládá strukturu adresového bytu **YAAAAAA0**. Následuje datový byte s požadovanou hladinou osvětlení v rozsahu 1 až 254. Číslo v tomto rozsahu udává procentuální výkon světelného zdroje, jenž je dán logaritmickou křivkou, která vychází ze vztahu (3.1), kde  $X(k)$  udává procentuální výkon zdroje pro požadovanou hodnotu  $k$  z rozsahu.

$$X(k) = 10^{\frac{k-1}{253/3}-1} (\%) \quad (3.1)$$

Příkladem takového příkazu je *Forward* rámec **00000010 10010001**, kde se jedná o příkaz pro *slave* zařízení se *short adresou* **000001**, pro přechod do úrovně osvětlení 145 (**10010001**), což odpovídá 5,099 % jeho výkonu.

### 2. Příkazy nastavující nepřímou úroveň osvětlení

Tento typ příkazů nastavuje úroveň osvětlení jinak než zadáním číselné hodnoty osvětlení. Jedná se například o povely zvýšení či snížení výkonu světelného zdroje bez udání přesné procentuální hodnoty nebo o požadavek maximálního výkonu.

Rozmezí příkazů této kategorie je 0 až 31 a je to například příkaz *STEP UP* s číslem 3 (**0000 0011**), který zvyšuje výkon světelného zdroje o jednu úroveň osvětlení  $k$ , danou vztahem (3.1).

### 3. Konfigurační příkazy

Příkazy v této kategorii jsou v rozmezí hodnot 32 až 143. Nastavují se jimi chování *slave* zařízení, ukládání dat a také se pomocí nich vytvářejí skupiny zařízení. Odesílání konfiguračních příkazů se provádí dvakrát s časovým odstupem 100 ms. Pokud v tomto intervalu dojde k odeslání jiného příkazu na dané *slave* zařízení, je tato zpráva ignorována. Děje se tak z důvodu zvýšení pravděpodobnosti přijetí konfiguračního příkazu.



Příkladem této kategorie je příkaz *REMOVE FROM SCENE* s číslem 80 - 95, jenž má podobu **0101 XXXX**. Tento příkaz odstraní dané zařízení z definované scény. Část příkazu **XXXX** uvádí číslo scény.

#### 4. Dotazové příkazy

Po odeslání dotazového příkazu v rozsahu 144 až 255 je očekávána odpověď, tedy *Backward* rámeček, kterým *slave* odpovídá na dotaz od *master* zařízení. Existují tři druhy relevantních odpovědí:

- “ANO“ (**1111 1111**)
- “NE“ (žádná reakce)
- DATA (**DDDD DDDD**)

Příkaz z této kategorie je například číslo 160 (**1010 0000**) s názvem *QUERY ACTUAL LEVEL*, kde se očekává odpověď s aktuální nastavenou úrovní osvětlení.

#### 5. Příkazy speciální

Poslední kategorií příkazů v rozmezí 256 až 272 jsou příkazy speciální. Většina příkazů z této kategorie slouží k adresování *slave* prvků. Jak již bylo popsáno v kapitole, k identifikaci těchto příkazů slouží speciální tvar adresového bytu.



## 4. HARDWARE

Kapitola má za úkol popsat principy, na kterých pracuje fyzická realizace celého systému. Je provázána s kap. 5 zabývající se programovým vybavením systému. Předpokladem pro fyzickou realizaci je využití vývojových prostředků, kterými se zabývá kap. 4.1. Vývojové prostředky umožnily realizaci dílčích částí systému, jejichž popis se nachází v dalších podkapitolách.

### 4.1. VÝVOJOVÉ PROSTŘEDKY

Vývoj a výroba *HW* probíhal ve dvou fázích. První fází bylo navržení a výroba modulů napájení, měření světelných podmínek a *DALI*. Každý z modulů byl navržen tak, aby bylo snadné ho připojit do nepájivého pole, to především znamenalo dodržení patřičných roztečí přípojovacích konektorů. Druhá fáze byla vývoj a výroba periférií.

Důvodem rozdělení na dvě fáze bylo testování vyrobených modulů v první fázi na nepájivém poli a předpokládaná prodleva při odlaďování chyb.

#### 4.1.1. Softwarové prostředky

Pro návrh veškerého *HW* bylo využito vývojového prostředí *Cadence 16.6*. Elektrická schémata byla vytvořena v programu *OrCAD Capture*, kontaktní plošky součástek v programu *PAD Designer*, pouzdra součástek a desky plošných spojů (dále *DPS*) v programu *PCB Editor*.

#### 4.1.2. Použité součástky

Většina částí systému byla nejprve zapojena na nepájivém poli a po té byly na této konfiguraci vyzkoušeny dílčí části programu. Bylo tomu tak, aby se zamezilo případným chybám v návrhu elektrického schématu celého systému a volbě součástek.

Při volbě součástek byl brán zřetel na jejich dostupnost. Byly vybírány spíše ty, které bylo možné v rámci vybavení katedry měření osadit na desky plošných spojů.

### 4.2. VÝVOJOVÉ MODULY

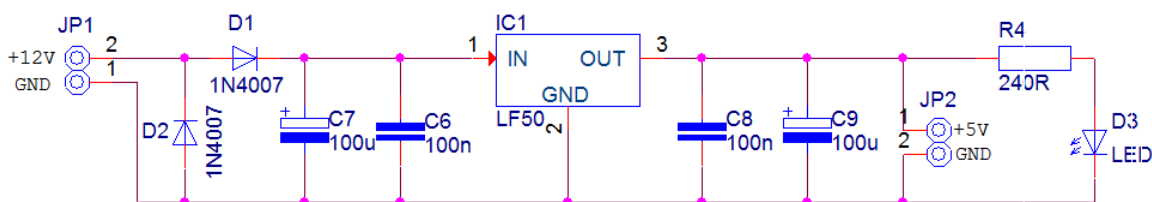
Důležitým faktorem při vývoji modulů bylo jejich začlenění do druhé fáze vývoje a výroby *HW*, jež byl návrh elektrického zapojení a především *DPS* obou periférií. Všechny tři moduly byly vytvořeny zejména z důvodu testování použitých technologií.

Po té, co byla odzkoušena jejich funkčnost, jak po hardwarové, tak po softwarové stránce, byly moduly použity jako dílčí části ústředny.

#### 4.2.1. Napájecí modul

Vstupním napájení ústředny je 12 V zdroj. Celá ústředna pracuje na napětí 5 V. Důvodem použití 12 V zdroje je potřeba napájení pro sběrnici *DALI*. Napájecí modul, Obr. 4.1, tedy zajišťuje stabilní 5 V napětí na výstupu při vstupním 12 V napájení.

Jádrem modulu je lineární regulátor napětí *LF50*. Výstupní proud regulátoru 0,5 A je plně dostačující pro napájení celé ústředny. Napájecí modul disponuje signalizací činnosti, kterou zajišťuje zelená *LED* dioda. *DPS*, viz. příloha B1.



Obr. 4.1, Elektrické schéma napájecího modulu

#### 4.2.2. APDS modul

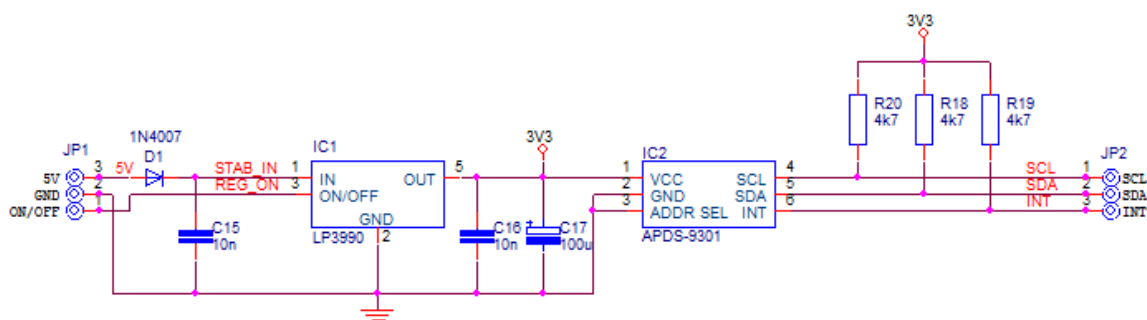
Úkolem tohoto modulu, Obr. 4.2, je poskytnout údaj o intenzitě osvětlení. Základním vstupem modulu je 5 V napájení a výstupem je sběrnice *I<sup>2</sup>C*. Měření intenzity okolního osvětlení provádí integrovaný obvod *APDS-9301*, popsáný v kap. 3.1.1. Tento integrovaný obvod, mimo jiné, disponuje výstupem přerušení, které je vždy vysláno, dle aktuálního nastavení obvodu. Výstup přerušení je také výstupem modulu.

Katalogový list integrovaného obvodu *APDS-9301* [19] uvádí, že je schopen pracovat s rozsahem vstupního napájení obvodu mezi 2,7 V a 3,6 V. Jelikož je ústředna napájena z modulu popsaného v kap. 4.2.1, na jehož výstupu je 5 V, je nutné nejprve provést úpravu vstupního napětí. Dalším důvodem úpravy napětí jsou výstupní napěťové nároky obvodu *APDS*. Výrobce uvádí maximální hodnotu napětí log. 1 na sběrnici 3,6 V.

Jelikož tento modul komunikuje přímo s mikroprocesorem, který je napájen 5 V, bylo nutné zajistit, aby udávaná minimální hodnota napěťové úrovně log. 1 nepřekročila maximální udávanou hodnotu napětí na sběrnici, udávanou výrobcem *APDS*.

V případě použitého mikroprocesoru, *ATmega16* [21], je minimální napěťová úroveň logické jedničky  $0,6 \cdot U_{CC}$ , což je při daném napájecím napětí 3 V. Aby tedy byly splněny oba požadavky na napětí logických úrovní, bylo nutné snížit napájení na hodnotu mezi 3 V až 3,6 V.

O snížené napětí modulu se stará lineární napěťový regulátor *LP3990*, jež na výstupu poskytuje napětí 3,3 V a proud 150 mA. Tento regulátor má ještě jeden vstup a tak vedle své hlavní funkce, umožňuje i regulaci napětí vypínat, resp. zapínat. Tento vstup je také dalším vstupem *APDS* modulu, jehož *DPS* se nachází v příloze B2.



Obr. 4.2, Elektrické schéma *APDS* modulu

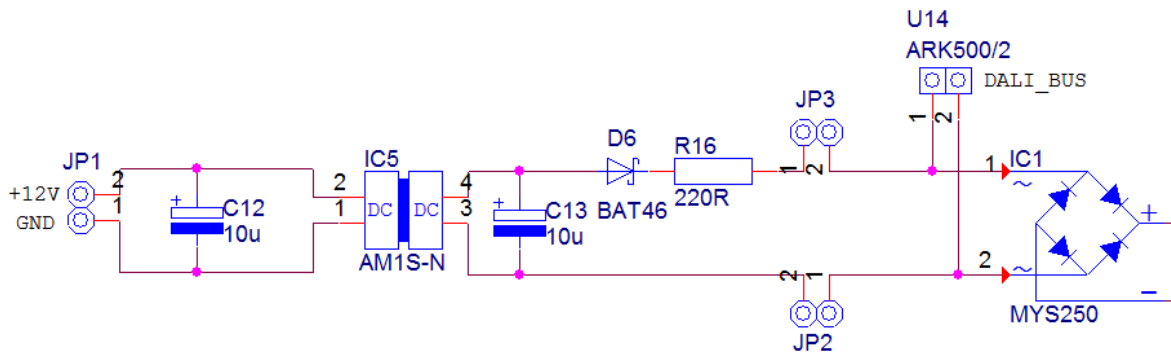
### 4.2.3. *DALI* modul

Tento modul, jehož *DPS* se nalézá v příloze B3, slouží ke komunikaci mikroprocesoru po *DALI* sběrnici. Do modulu jsou přivedeny dvě hodnoty napětí. Je tomu tak z důvodu, že ústředna bude na sběrnici v roli *master* zařízení, tudíž bylo nutné vyřešit napájení sběrnice. Napětí 12 V slouží jako zdroj pro *DALI* sběrnici. Napětí 5 V umožňuje komunikaci modulu s mikroprocesorem. Samotnou komunikaci modulu zajišťují konektory *DALI\_RX* a *DALI\_TX*, viz. Obr. 4.4.

Pro snadnější popis se nechá elektrické schéma modulu rozdělit do dvou částí. První část, Obr. 4.3, se skládá z napájení sběrnice, konektoru *DALI* sběrnice a usměrňovacího můstku. Inspirací pro tvorbu modulu se staly aplikace [16] a [6].

Aby bylo zamezeno zničení ostatních částí systému při případném nesprávném připojení *DALI* sběrnice, je napájení sběrnice odděleno od napájení zbytku ústředny DC/DC měničem *IC5*. Dalším ochranným prvkem je Schottkyho dioda *D6* za měničem, která zabraňuje průchodu proudu do měniče nesprávným směrem.



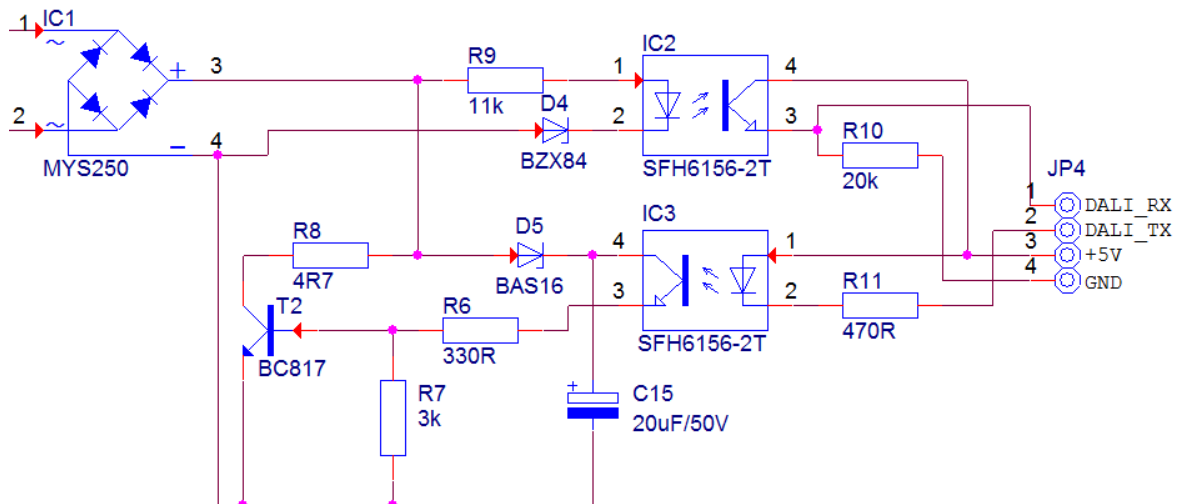
Obr. 4.3, Elektrické schéma *DALI* modulu, první část

Univerzálnost modulu zajišťují 2 jumpery *JP2* a *JP3*, které slouží k připojení, resp. odpojení napájení *DALI* sběrnice. Při vyndání propojovacích kolíků obou jumperů dojde k odpojení napájení sběrnice a modul se nechá využít pro komunikaci jako *slave* zařízení.

Posledním prvkem této části *DALI* modulu je diodový můstek *IC1*. Sběrnice připojená k modulu nemá standardem pevně daný způsob portace. V praxi to znamená, že při připojení dvoužilového vedení sběrnice nezáleží na pořadí vodičů v konektoru. Z tohoto důvodu je použit diodový můstek, který zajistí správnou polaritu signálu pro další zpracování.

Druhá část *DALI* modulu, Obr. 4.4, začíná za diodovým můstkem. Řeší se v ní vysílání a přijímání dat. Oba směry toku dat probíhají přes optočleny *SFH6156-2T*. Důvodem použití optočlenů je nutnost galvanicky oddělit sběrnici od zbytku systému. Optočleny mají izolační pevnost 5,3 kV a doba sestupné/náběžné hrany je 2 µs, což je pro časování sběrnice postačující.

Přijímání dat probíhá přes optočlen *IC2*. Pokud se na sběrnici vyskytne hodnota odpovídající log. 0, tranzistor optočlenu zůstane zavřený a tím dojde ke spojení výstupu *DALI\_RX* se zemí. V opačném případě prochází proud přes resistor *R9* a stabilizační diodu *D4*, což má za následek otevření tranzistoru optočlenu. Otevření tranzistoru způsobí připojení 5 V napětí na výstup *DALI\_RX*.

Obr. 4.4, Elektrické schéma *DALI* modulu, druhá část

Vysílání dat funguje obdobně. Vychází se z předpokladu, že sběrnice je výchozím stavu na hodnotě log. 1. Diodou optočlenu *IC3* protéká proud, což má za následek otevření jeho tranzistoru. Po příchodu log. 0 dojde k uzavření tranzistoru optočlenu, což vede k sepnutí bipolárního tranzistoru *T2*. To způsobí pokles napětí na *DALI* sběrnici.

Při dané konfiguraci a použitých součástkách jsou napěťové úrovně logických hodnot následující: log. 1 = +12,5 V a log. 0 = +1,76 V.

Obě napěťové úrovně vyhovují požadavkům fyzické vrstvy *DALI* sběrnice, které jsou popsány v kap. 3.2.1.

### 4.3. NAPÁJENÍ SENZORU

Uvažované metody napájení senzoru jsou diskutovány v kap. 2.5. Vybrána byla konstrukčně nejjednodušší metoda, kdy je fotovoltaický článek nastálo připojen k bateriím. K tomuto účelu bylo nutné vybrat vhodné baterie a na základě výběru baterií byl navržen vhodný fotovoltaický článek. Při výběru byly vzaty v potaz statistické údaje, které se vážou k České republice [18].

#### 4.3.1. Výběr baterií

Závěr kap. 2.5.3 uvádí, že vhodným typem bateriových článků pro použitou metodu napájení je *NiCd* nebo *NiMH*. Jako konstrukčně vhodnější byly vybrány *NiCd* baterie. Mají delší životnost a lépe snášejí přebíjení. Jelikož tyto články obsahují kadmium, které je škodlivé pro životní prostředí, je jejich prodej na území Evropské unie zakázán.



S ohledem na tuto skutečnost byly vybrány *NiMH* články, které dosahují horších parametrů.

Při návrhu elektrického schématu senzoru byl z důvodu bateriového napájení kladen důraz na co nejnižší spotřebu celé periferie. Senzor tedy disponuje součástkami, které pracují na sníženém napětí a mají nižší proudovou spotřebu, než je tomu v případě ústředny. Další úsporu energie přineslo použití *MOSFET* tranzistorů jako spínačů, které v době, kdy daný integrovaný obvod není používán, odpojí obvod od napájení.

S ohledem na elektrické parametry použitých obvodů senzoru bylo napájecí napětí zvoleno 3,3 V. Pro výpočet potřebné kapacity článků bylo nutné odhadnout celkovou proudovou spotřebu periferie. Odhad probíhal na základě proudových spotřeb obvodů, které jsou na napájení připojeny trvale s ohledem na čas, kdy jsou v aktivním módu. V úvahu byla brána 10ti hodinová pracovní doba, ve které jsou data ze senzoru odesílána každé 3 minuty, viz. kap. 5.6.2. Celková proudová spotřeba senzoru byla odhadnuta na  $C_0 = 25$  mAh za den. Statistika udává, že z dlouhodobého průměru je počet dní ( $D_N$ ) v řadě, kdy výkon slunečního záření nedosáhl republikového průměru, přibližně  $D_N = 20$ .

Pokud má být senzor schopen pracovat v těchto světelných podmínkách musí být, dle vztahu (4.1), minimální kapacita bateriových článků  $C = 500$  mAh.

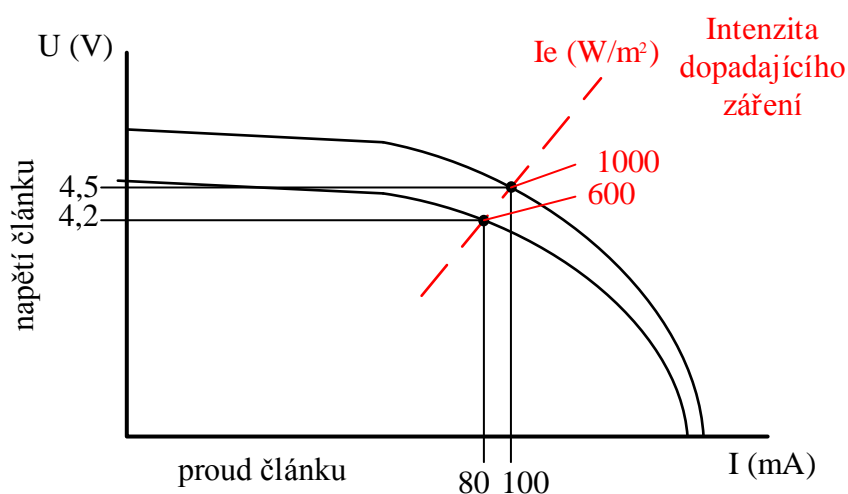
$$C = C_0 D_N (\text{mAh}) \quad (4.1)$$

Nastat však mohou případy, kdy vzroste spotřeba senzoru, například z důvodu kalibrace. Autor práce tyto skutečnosti zvážil a po konzultaci vybral články o dvojnásobné kapacitě, než je minimální vypočtená.

Aby byla zajištěna stálá hodnota napětí pro napájení jednotlivých součástí senzoru, jsou baterie připojeny na lineární napěťový regulátor s nízkou vlastní spotřebou, jenž dodává napětí celé periférii. Použitý napěťový regulátor způsobí 0,15 V pokles napětí mezi jeho výstupem a vstupem. S ohledem na dostupné baterie, bylo vybráno sériové zapojení 3 *NiMH* článků o jmenovité hodnotě 1,2 V / 1 Ah. Celkově je tak na vstup napěťového regulátoru připojený bateriový set o napětí 3,6 V a kapacitě 1 Ah.

### 4.3.2. Výběr fotovoltaického článku

Vhodný způsob nabíjení baterií v této aplikaci je nabíjení proudem v rozmezí  $1/12C$  až  $1/10C$ , kde  $C$  udává kapacitu baterie. Nabíjecí proud by se tedy měl pohybovat mezi 83 mA až 100 mA. Dalším faktorem při výběru vhodného fotovoltaického článku je průměrná intenzita slunečního záření, která je v ČR  $I_e = 600 \text{ W/m}^2$ . Výrobci většiny fotovoltaických článků udávají jmenovité hodnoty napětí a proudu článku při  $I_e = 1000 \text{ W/m}^2$ . Následující graf udává přibližný pokles výkonu článku při poklesu intenzity dopadajícího záření na článek.



Obr. 4.5, Pokles výkonu FV článku v závislosti na intenzitě dopadajícího záření

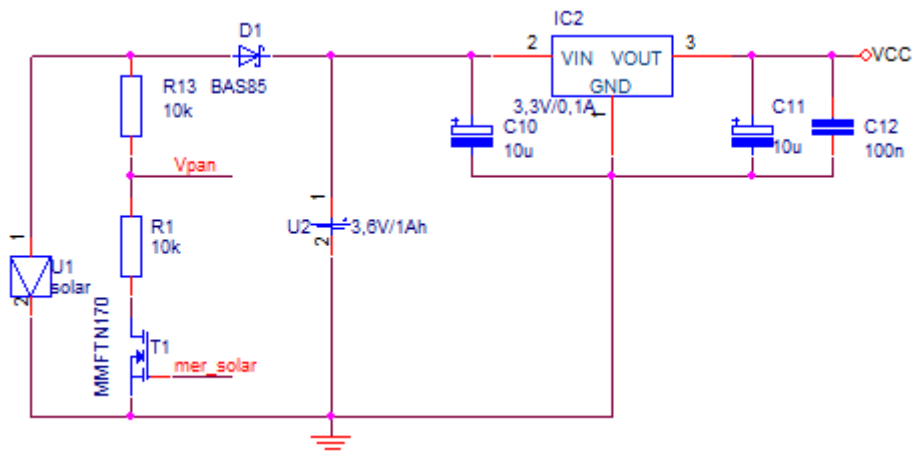
Obr. 4.5 je orientační a udává, jaké by přibližně měly být jmenovité hodnoty fotovoltaického článku, aby byl článek schopen dosáhnout potřebného výkonu pro nabíjení baterií. Napětí, dodávané článkem musí být vyšší, než je napětí soustavy baterií a to z důvodu úbytku napětí Shottkyho diody za fotovoltaickým článkem

Použitý fotovoltaický článek je monokrystalický se jmenovitými hodnotami 4,5 V / 100 mA.

### 4.3.3. Schéma napájení senzoru

Jak již bylo popsáno výše, napětí periferie je 3,3 V. Aby veškerá energie baterií byla soustředěna na výrobu napájecího napětí, je mezi fotovoltaický článek a baterie zařazena Shottkyho dioda, jež zabraňuje průchodu proudu směrem k fotovoltaickému

článku a zároveň umožňuje průchod nabíjecího proudu ze článku směrem k bateriím viz. Obr. 4.6.



Obr. 4.6, Elektrické schéma napájení senzoru

Paralelně k fotovoltaickému článku je umístěn napěťový dělič, jež má za úkol sprostředkovat měření napětí na článku. Aby nedocházelo k průchodu proudu a tím poklesu nabíjecího výkonu, je napěťový dělič sepnut pouze při měření napětí článku pomocí *MOSFET* tranzistoru *T1*. Měření napětí nemůže probíhat přímo, jelikož je potřeba napětí snížit kvůli A/D převodníkům, jejichž referenční napětí je rovno napětí z regulátoru a předpokládá se, že při plném výkonu článek bude dodávat napětí vyšší než referenční. Změřené napětí na fotovoltaickém článku odpovídá dodávanému napětí do soustavy při aktuálním zatížení článku.

#### 4.4. PERIFERIE

Obě periferie systému využívají mnoho společných částí. Výhoda využití stejných nebo podobných součástek, principů a postupů usnadnila návrh a přispěla k přehlednosti systému jako celku. Další výhodou společných částí je při případné sériové výrobě, jelikož se tak nechají ušetřit náklady na výrobu.

Výhoda použití stejného *MCU* u obou periferií spočívá zejména ve snazší přenositelnosti kódu mezi oběma periferiemi, jelikož se některé části u obou periferií shodují.



#### 4.4.1. Mikroprocesor

Jádrem celého systému se staly u obou periférií mikrokontroléry společnosti *ATMEL*, konkrétně *AVR ATmega16*. Venkovní senzor disponuje variantou této jednotky *ATmega16L*, která pracuje na nižším napájecím napětí a má nižší spotřebu. Jedná se o 8bitové mikroprocesory (*MCU*) *RISC* architektury s 16 KByty programovatelné *flash* paměti [21].

Při volbě mikroprocesoru byl brán také zřetel na dostupnost komunikačních portů a možnost připojení externích součástek [1]. Jelikož data na výstupu *IQRF* jsou posílána do *MCU* přes sběrnici *SPI*, bylo zapotřebí, aby tento komunikační standard mikroprocesor podporoval. Obdobně je tomu i u měření intenzity osvětlení, kde jsou využity porty pro připojení *I<sup>2</sup>C* sběrnice. Dalšími kritérii byly možnost připojení krystalu reálného času, minimálně dva vstupy externího přerušení a jeden A/D převodník.

Propojení *MCU* s programátorem přes *JTAG* přináší velký komfort při odladování programu. Starší typ připojení přes *ISP* nenabízí možnost krokování v *debug* módu a proto je těžší program testovat a vyhledávat případné chyby v kódu.

#### 4.4.2. Senzor

Elektrické schéma senzoru se nachází v příloze A1, výsledná *DPS* v příloze B4. Řídící jednotka senzoru pracuje z důvodu snížení spotřeby na frekvenci 4 MHz. Pro získávání vstupních dat využívá výše popsané integrované obvody *APDS-9301*, kap. 3.1.1 a *DS18B20*, kap. 3.1.2. Oba integrované obvody jsou připojeny na napájení pouze v průběhu měření pomocí *MOSFET* tranzistorů *T3*, resp. *T2*. Dalším vstupním údajem je napětí fotovoltaického článku. Způsob měření napětí článku je popsán v kap. 4.3.3.

Modul pro bezdrátovou komunikaci *IQRF* má v případě senzoru označení *TR-55D* [22]. Na rozdíl od modulu použitého v ústředně je tento modul schopný pracovat na nižším napájecím napětí.

Z důvodu ladění senzor disponuje *LED* diodou, vstupním a resetovacím tlačítkem.

#### 4.4.3. Ústředna

Jelikož je ústředna napájena ze síťového adaptéru, odpadla potřeba vybírat prvky, pracující na nižším napětí. Schéma elektrického zapojení se nachází v příloze A2.



Frekvence oscilátoru mikroprocesoru je 16 MHz. Deska plošných spojů ústředny, příloha B5, byla navržena tak, aby bylo možné využít testovacích modulů z kap. 4.2.

Pro bezdrátovou komunikaci *IQRF* je využit modul *TR-52DA* [23]. Modul pracuje na stejném napětí jako ústředna, tedy 5 V.

Tlačítka *S2*, *S3*, *S4* a *S5* na ovládacím panelu jsou rozmístěna tak, aby umožňovala intuitivní pohyb v menu na zobrazovací jednotce. Zobrazovací jednotkou je *LCD* displej *ATM1602B*. Byl zvolen z hlediska nízké ceny a dostupné programové knihovny pro jeho ovládání. Součástí displeje je řadič *S6A0069*. Dvouřádkový displej používá standardní znakovou sadu a je modře podsvícený.



## 5. SOFTWARE

Tvorba programového vybavení probíhala v několika fázích. Nejprve bylo zapotřebí vytvořit nebo upravit programové knihovny pro komunikaci s jednotlivými integrovanými obvody. Následně byly vytvořeny programové knihovny, které ovládají větší celek. Do této kategorie se řadí například programové knihovny, které mají na starosti zpracování dat ze snímačů, kalibraci, regulaci nebo bezdrátovou komunikaci. V poslední řadě bylo vytvořeno uživatelské prostředí a celý systém zkompletován.

### 5.1. VÝVOJOVÉ PROSTŘEDKY

Při vývoji softwaru bylo využito některých programových knihoven, které jsou volně k dispozici na internetu. Jednalo se výlučně o programové knihovny pro obsluhu displeje a sběrnic. Všechny přejeté knihovny musely být upraveny, aby splňovaly potřeby aplikace.

#### 5.1.1. Softwarové prostředky

Programování *MCU* probíhalo ve vývojovém prostředí *Atmel Studio 6.1*. Kromě editoru zdrojového kódu tento vývojový nástroj po překladu kódu umožňuje nahrát projekt přímo do zařízení, které je po té možno spustit v *debug* módu, což usnadňuje ladění programu.

Jelikož vývojové prostředí pro programování *IQRF* modulů nenabízí editor zdrojového kódu, byl pro tuto činnost využíván *PSPad* a pro menší zásahy do kódu poznámkový blok. Překlad a nahrávání kódů do modulů *IQRF* probíhal v prostředí *IQRF IDE4*.

### 5.2. OBSLUHA DALI SBĚRNICE

Popisem principu činnosti *DALI* sběrnic se zabývá kap. 3.2. Programová knihovna, která by poskytovala ucelený nástroj pro komunikaci s touto sběrnicí, bohužel není pro mikrokontroléry *ATmega* k dispozici. Bylo proto nutné ji vytvořit. Z důvodu náročnosti na implementaci a testování byla této části diplomové práce věnována větší časová dotace. Pro potřeby této práce bylo nutné vytvořit program pro obsluhu *master* zařízení. Implementace však probíhala tak, aby bylo možné, bez velkých úprav, použít programovou knihovnu i pro zařízení typu *slave*.

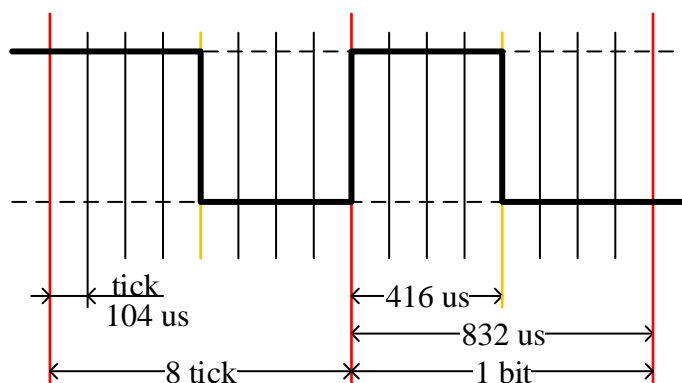


### 5.2.1. Základní princip obsluhy sběrnice

Přijímání i odesílání jednotlivých dat po sběrnici probíhá v cyklech s pevně danou periodou. Signál je vzorkován. Konec periody každého cyklu je okamžik, kdy je možné číst logickou úroveň na sběrnici, zapisovat logickou úroveň na sběrnici nebo obojí. O správné časování se stará interní 16bitový časovač *Timer1*. Každé přerušení časovače znamená konec jedné periody cyklu, který se nazývá *tick*.

Pro správné fungování sběrnice bylo nutné zvážit kolik takovýchto přerušení, neboli *ticků*, se má provést během jednoho bitu. Z důvodu bi-fázového kódování je při odesílání nutné logickou úroveň kontrolovat na začátku a uprostřed bitu (*fázová* a *bitová* hrana). Z toho vyplývá, že minimální počet *ticků* pro odesílání jsou dva.

V případě přijímání dat je situace jiná. Nelze se totiž spolehnout na přesné časování sběrnice, jež má na starosti jiné zařízení. Je tedy nutné odečíst více hodnot během doby trvání jednoho bitu a na základě statistiky určit hodnotu daného bitu. S ohledem na rychlost mikrokontroléru, použité součástky a výsledky testování, byla časová prodleva mezi jednotlivými *ticky* stanovena na hodnotu 104  $\mu$ s. Znamená to, že celkový počet *ticků* v jednom bitu je 8. Situace je znázorněna na posloupnosti dvou bitů, Obr. 5.1.



Obr. 5.1, Vzorkování signálu na *DALI* sběrnici

Celý proces řízení komunikace na sběrnici je dán stavovou proměnnou *flag*. Jedná se o proměnnou, která určuje, v jakém stavu se program pro obsluhu komunikace nachází. Přípustných je 6 stavů komunikace, tudíž proměnná *flag* může nabývat následujících hodnot.



<i>NO_ACTION</i> .....	na sběrnici neprobíhá komunikace
<i>SENDING_DATA</i> .....	probíhá odesílání dat na sběrnici
<i>RECEIVING_DATA</i> .....	probíhá přijímání dat ze sběrnice
<i>REPEAT_SEND</i> .....	opakované odesílání dat
<i>ERROR</i> .....	chyba v přijímání dat
<i>CD_ERROR</i> .....	detekována kolize při odesílání dat

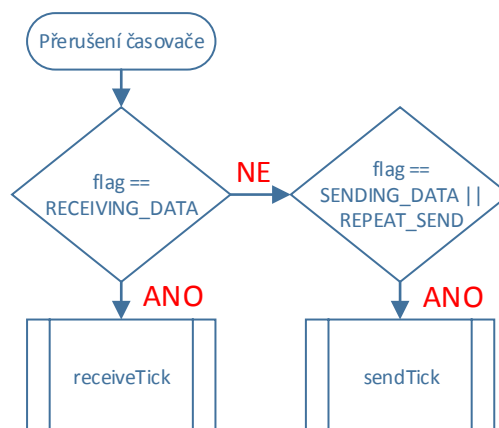
### 5.2.2. Inicializace komunikace

Při práci s programovou knihovnou se nejprve provede inicializace komunikace po sběrnici. Funkce, která má na starosti tento krok se nazývá *DALInit()*. Pomocí této funkce dojde k inicializaci a aktivaci externího přerušení, také hodnota proměnné *flag* se nastaví na *NO\_ACTION*. Jelikož se *DALI* sběrnice nachází v klidovém stavu na hodnotě log. 1, je externí přerušení aktivní na sestupnou hranu.

Pokud dojde k externímu přerušení, znamená to, že nějaké zařízení se snaží komunikovat po sběrnici. Sestupná hrana je tedy vyhodnocena jako *start bit* příchozího rámce. Pokud na sběrnici právě neprobíhá jiné zpracování dat, algoritmus spustí proces přijímání dat funkcí *receiveData()*, kap. 5.2.3.

V případě, že uživatel programové knihovny zvolí odesílání rámce, zavolá se funkce *sendData()*, kap. 5.2.4. Aby nedocházelo ke kolizím, tato funkce deaktivuje externí přerušení a aktivuje ho po skončení odesílání dat.

V obou případech dojde k aktivování časovače *Timer1*, který se stará o vzorkování signálu. Při každém nastalém přerušení časovače, bez ohledu na to, zda jsou data přijímána nebo odesílána, dochází k inkrementaci proměnné *tick\_count*, která udává informaci o poloze právě zpracovávaného vzorku v rámci. Zda se jedná o vysílání či přijímání dat, je rozhodováno při každém přerušení časovače na základě stavové proměnné *flag*, podle následujícího digramu, Obr. 5.2.

Obr. 5.2, Vývojový digram přerušení časovače *Timer1*

### 5.2.3. Přijímání dat

Po té, co je na sběrnici přijat start bit příchozího rámce, je deaktivováno externí přerušení, spuštěn časovač *Timer1*, hodnota proměnné *flag* nastavena na *RECEIVING\_DATA* a vynulovány pomocné proměnné. To vše zajišťuje funkce *receiveData()*, volaná externím přerušením.

Po spuštění časovače *Timer1* nastává periodické vzorkování příchozího signálu. O čtení logických úrovní a vyhodnocení přijatých dat se stará funkce *receiveTick()*, jejíž vývojový digram se nachází v příloze D1.

Na začátku každé periody vzorkování dojde k uložení logické úrovně na sběrnici do pole *actual\_bit*, jehož velikost odpovídá počtu vzorků v jednom bitu, tedy 8. Toto pole uchovává všechny vzorky aktuálně přijímaného bitu. O uložení vzorku na správné místo v poli *actual\_bit* se stará proměnná *internTick*, která inkrementuje vždy při načtení logické úrovně.

Pokud proměnná *internTick* nabude hodnoty odpovídající pořadí posledního vzorku, tedy 7, předá algoritmus pole *actual\_bit* funkci *getBitValue()*, která vrátí hodnotu aktuálního bitu.

Výsledná hodnota bitu se vyhodnocuje na základě statistiky počtu logických úrovní v obou polovinách bitu. Funkce je detailně popsána vývojovým diagramem v příloze D2. Pokud dojde k selhání algoritmu, je návratová hodnota funkce chybová. V takovém případě je stavová proměnná *flag* nastavena na hodnotu *ERROR* a proces přijímání dat je tím přerušen. K přerušení přijímání dat může také dojít při chybě *start bitu*, který je kontrolován po přijetí prvního bitu.



Po skončení přijímání je hodnota stavové proměnné *flag* vrácena do stavu *NO\_ACTION*, časovač *Timer1* je vypnut a externí přerušení aktivováno.

#### 5.2.4. Odesílání dat

Při zápisu dat na sběrnici je využit podobný princip jako při čtení dat ze sběrnice, kap. 5.2.3. Hlavním rozdílem je počet manipulací (čtení/zápis) se sběrnici v průběhu jednoho bitu. Při odesílání dat probíhá manipulace se sběrnici pouze na začátku a v půlce každého bitu.

Algoritmus odesílání dat začíná zavoláním funkce *sendData()*. Tato funkce přijme vstupní rámec, jež nastaví jako aktuálně odesílaný. Dále se funkce stará o výchozí nastavení pomocných proměnných, deaktivaci externího přerušení a převod délky rámce do jednotek *tick*. Na závěr ještě dochází k určení, zda aktuálně odesílaný rámec neobsahuje některý ze speciálních příkazů, u kterého je nutné odesílání zopakovat. Po té je zapnut časovač *Timer1*, který periodicky volá funkci *sendTick()*.

Funkce *sendTick()* nejprve zjistí, v jaké části rámce se proces odesílání nachází. Pokud se proces nachází uvnitř odesílaného rámce, pak je postup následující. Na začátku každého bitu je přečtena logická úroveň na sběrnici (*fázová hrana*). Pokud se logická úroveň shoduje s hodnotou právě odesílaného bitu, je na sběrnici zapsaná opačná logická úroveň. Důvod změny, resp. ponechání logické úrovně z předchozího bitu vychází z principu bi-fázového kódování, popsáno v kap. 3.2.2.

Hodnota bitu je na sběrnici zapisována v každé polovině bitu (*bitová hrana*), je doprovázena inkrementací proměnné *bit\_count*, která určuje pořadí právě odesílaného bitu. Každý zápis na sběrnici (na začátku i v polovině bitu) je kontrolován pro případnou kolizi na sběrnici. Ke kolizím může docházet zejména tehdy, pokud se nepovede zápis log. 0 na sběrnici. V takovém případě je odesílání přerušeno nastavení stavové proměnné *flag* na hodnotu *CD\_ERROR*.

Po skončení odesílání rámce se provede odeslání stop bitů. Protokol *DALI* má pevně danou časovou prodlevu mezi koncem stop bitů a příchozí odpovědí. Hodnota prodlevy je 2,92 ms a po jejím uplynutí algoritmus aktivuje externí přerušení a vrátí stavovou proměnnou *flag* na hodnotu *NO\_ACTION*.

Na začátku této podkapitoly bylo uvedeno, že před odesláním dat proběhne kontrola, zda odesílaný příkaz není z kategorie speciálních. Pokud je tomu tak, je nastavena prodleva odpovídající prioritě rámce a proces odesílání dat je zopakován.



Stejná prodleva před opětovným odesláním dat je nastavena, i pokud dojde k detekci kolize na sběrnici.

Algoritmus funkce *sendTick()* je detailně popsán vývojovým diagramem v příloze D3. Po skončení funkce je vypnut časovač *Timer1*.

### 5.2.5. Obsluha programové knihovny

Výsledkem vývoje softwaru pro obsluhu *DALI master* zařízení je programová knihovna *DALImaster.h*. Před započítím jakékoliv komunikace po sběrnici je nejprve nutné provést inicializaci příkazem *DALIInit()*.

#### Odeslání rámce:

Pro odeslání jednoho rámce je v programové knihovně připraven datový typ s názvem *FRAME*. Je tedy nutné vytvořit proměnnou tohoto datového typu, která odpovídá struktuře rámce. Datový typ *FRAME* je struktura, která obsahuje 6 položek, které jsou popsány v Tab. 5.1.

Tab. 5.1, Struktura datového typu *FRAME*

Název položky	Datový typ	Popis
address	uint8_t	1. byte rámce, adresa <i>slave/master</i> zařízení
command	uint8_t	2. byte rámce, příkaz pro adresované zařízení
data	uint8_t	3. byte rámce, data pro <i>master</i> zařízení
priority	uint16_t	priorita rámce
bytes_to_send	uint8_t	počet bytů v rámci
frame_length	uint16_t	délka rámce (nevyplňuje se)

Pro vytvoření nového rámce a vyplnění všech položek se data nechají odeslat funkcí *sendData()*. Vstupním parametrem této funkce je vytvořený rámec.

#### Přijetí rámce:

Jelikož je při inicializaci a po skončení odesílání aktivní externí přerušení, tak přijetí rámce proběhne automaticky bez potřeby zásahu. Systém je tak citlivý na příchozí start bit zařízení, které se pokouší komunikovat s *master* zařízením.

Příchozí zpráva se ukládá do pole *received\_answer*. V případě odpovědi od *slave* zařízení prvních 8 položek *received\_answer* odpovídá 8 bitům dat. Pokud je přijata



odpověď od jiného *master* zařízení, potom se v prvních 8 položkách nachází adresa a v následujících 8 položkách data, viz. kap. 3.2.3. V obou případech se jedná o rámce bez *start* a *stop* bitů.

### 5.3. ZÍSKÁVÁNÍ DAT ZE SNÍMAČŮ

Na rozdíl od programové knihovny pro obsluhu *DALI* sběrnice, jsou knihovny pro komunikaci po sběrnicích *I<sup>2</sup>C* a *1-Wire* volně dostupné na internetu, např. [11]. Bylo tudíž snadnější začít pracovat se snímačem intenzity osvětlení a teploty. V obou případech byly využity programové knihovny, které byly následně upraveny pro danou aplikaci. Původy knihoven jsou uvedeny ve zdrojích práce, konkrétně [12] a [15].

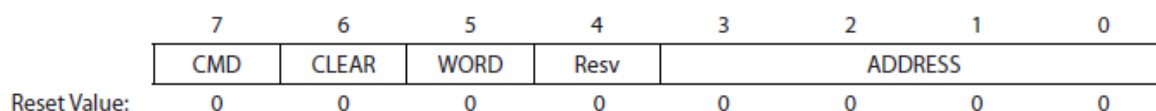
#### 5.3.1. Obsluha snímače Avago APDS-9301

Prostředkem pro komunikaci po sběrnici *I<sup>2</sup>C* se stala programová knihovna *i2cmaster.h*. Programová knihovna nabízí několik základních funkcí pro komunikaci *master* zařízení se zařízeními typu *slave*. Pro účely komunikace s *APDS-9301* byly využity následující funkce, které jsou popsány v Tab. 5.2.

Tab. 5.2, Použité funkce programové knihovny *i2cmaster*

Název funkce	Popis
<i>i2c_init</i>	inicializace sběrnice (volána jednou před začátkem komunikace)
<i>i2c_stop</i>	ukončení komunikace; uvolnění sběrnice
<i>i2c_start</i>	start komunikace; určení směru komunikace
<i>i2c_start_wait</i>	start komunikace; určení směru; čekání na potvrzení komunikace
<i>i2c_write</i>	zápis jednoho bytu dat do zařízení
<i>i2c_readAck</i>	čtení jednoho bytu dat; očekává se více příchozích bytů
<i>i2c_readNak</i>	čtení jednoho bytu dat; ukončení komunikace

Při jakémkoliv čtení nebo zápisu dat do integrovaného obvodu je vždy nejprve nastaven *command* registr.



Obr. 5.3, Struktura *command* registru *APDS-9301*

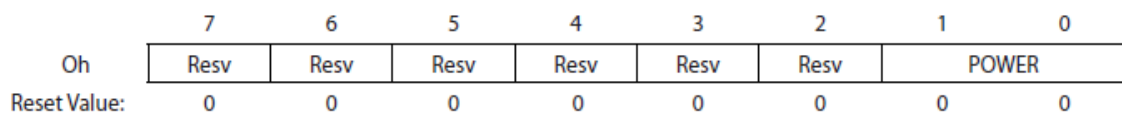


Nastavení *command* registru udává, s jakým registrem se bude pracovat v dalším kroku. Tento 8bitový registr proto nemá žádnou adresu a má následující strukturu, Obr. 2.1. Významem jednotlivých bitů registru se zabývá následující tabulka Tab. 5.3.

Tab. 5.3, Význam jednotlivých bitů *command* registru APDS-9301

Název bitu	Číslo bitu	Význam
<i>CMD</i>	7	Výběr <i>command</i> registru, musí být log. 1
<i>CLEAR</i>	6	Vymaže všechny čekající přerušení
<i>WORD</i>	5	Následovat bude čtení/zápis dvou bytů
<i>Resv</i>	4	Rezerva, musí být log. 0
<i>ADDRESS</i>	3:0	Adresa registru, do kterého se bude zapisovat/číst

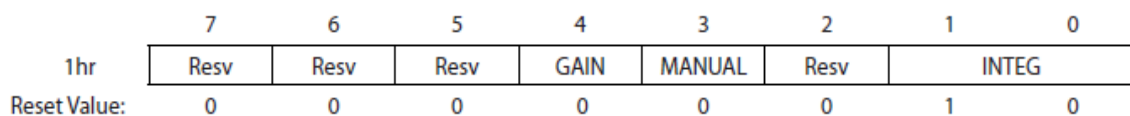
Celý proces inicializace, počátečních nastavení a čtení dat probíhá podle vývojového digramu v příloze D4. Po inicializaci a zkušebním zápisu je zařízení nastaveno do provozního stavu. Provozním stavem se rozumí stav, kdy je obvod přepnut ze stavu nízké spotřeby do normálního režimu. Změna se provádí pomocí *control* registru, jež má adresu 0 a jeho struktura je následující, Obr. 5.4.



Obr. 5.4, Struktura *control* registru APDS-9301

Pokud je do *control* registru zapsána hodnota 03h je zařízení uvedeno do normálního pracovního režimu.

Dalším registrem, který je potřeba nastavit je *timing* registr. Volbami nastavení se dá docílit požadované doby integrace obou A/D převodníků. Registr také umožňuje volbu analogového zesílení a obsahuje následující bity, Obr. 5.5.

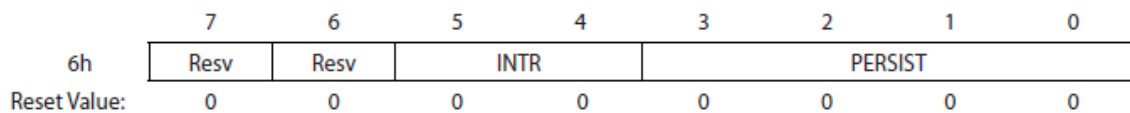


Obr. 5.5, Struktura *timing* registru APDS-9301



Pro získání co nejpřesnější informace o okolním osvětlení je potřeba nastavit co nejdelší dobu integrace. Do registru je tedy zapisována hodnota  $12h$ , což má za následek aktivování zesílení pomocí bitu *GAIN* a nastavení doby integrace na hodnotu  $402ms$ , pomocí bitů *INTEG*. Jelikož výsledný systém není založen na potřebě rychlého opakovaného čtení informace o okolním osvětlení, je maximální doba integrace přípustná v zájmu kvality zjišťovaných dat.

Poslední registr, který je před vyčítáním dat z obvodu nutné nastavit je *interrupt control* registr. Jeho adresa je  $6h$  a nastavuje se pomocí bitů *INTR* a *PERSIST*, viz Obr. 5.6.



Obr. 5.6, Struktura *interrupt* registru *APDS-9301*

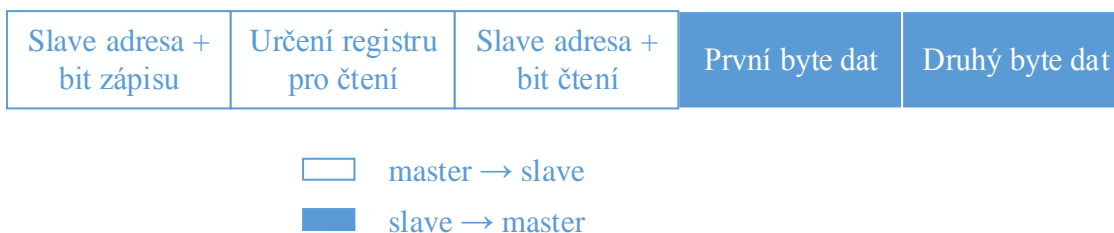
Bity *INTR* slouží k aktivaci přerušení, pokud je do nich zapsána hodnota  $01h$  je přerušení aktivní. K určení, kdy má přerušení nastat, je potřeba nastavit bity *PERSIST*. Přerušení může nastat při každém dokončení převodu A/D převodníků nebo pokud je aktuální úroveň osvětlení mimo hranice nastavené v *interrupt threshold* registru. Mimo tyto dvě hlavní události vyvolávající přerušení je ještě možné nastavit po kolika dokončení převodů A/D převodníků má přerušení nastat.

Jelikož není potřeba hlídat úrovně osvětlení, je pro potřeby celého systému přerušení vyvoláno vždy po skončení převodu A/D převodníků. Do registru je tedy zapsána hodnota  $10h$ .

Po této počáteční proceduře mohou být data obsahující informace o okolním osvětlení z integrovaného obvodu postupně odečítána. Jelikož obvod disponuje dvěma kanály, je výsledná informace obsažena ve dvou 2bytových zprávách.

Na Obr. 5.7 jsou zobrazeny jednotlivé byty, které jsou vyměňovány mezi *master* a *slave* zařízeními při čtení informace z jednoho kanálu. Postupně jsou takto přečteny informace z obou kanálů.



Obr. 5.7, Příklad čtení 2bytové zprávy z *APDS-9301*

Následuje výpočet intenzity osvětlení v jednotkách lux. Pro tuto operaci byla knihovna *i2cmaster.h* doplněna o funkci *countLux()*. Jejími vstupními parametry jsou 16bitové informace z obou kanálů. To, jakým způsobem je výsledná hodnota vypočtena, určuje podíl hodnot prvního a druhé kanálu, resp. podíl informací přečtených na obou diodách. Algoritmus výpočtu je uveden v katalogovém listu integrovaného obvodu [19].

### 5.3.2. Obsluha snímače Dallas DS18B20

Jelikož je integrovaný obvod pro měření teploty *DS18B20* velmi rozšířený, existuje mnoho literatury a příkladů, jak s tímto obvodem pracovat [13]. V případě *APDS-9301* bylo nutné nejprve nastudovat katalogový list [19] a na jeho základě sestavit sekvenci příkazů, které vedou k přečtení dat. V případě snímače teploty tato potřeba odpadla, jelikož byla k dispozici programová knihovna zajišťující celou komunikaci s tímto snímačem [12].

Pro čtení informace o okolní teplotě bylo tedy nutné pouze importovat knihovnu *DS18B20.h* a využít funkce *read\_temperature()*, která výslednou hodnotu ve stupních celsia ukládá do výstupních proměnných *digit* a *decimal*.

### 5.3.3. Měření napětí fotovoltaického článku

Napětí fotovoltaického článku jsou spolu s teplotou veličiny, které přímo neovlivňují chod celého systému a nemají proto při výsledné regulaci osvětlení zásadní význam. Obě hodnoty jsou spíše informativní a hodí se pro budoucí rozšíření systému. Pro účel zjišťování napětí fotovoltaického článku byla vytvořena knihovna *solar.h*. Knihovna v podstatě obsluhuje integrovaný A/D převodník a zpracovává vstupní data.

Nejprve je nutné provést inicializaci pomocí funkce *solarInit()*. Funkce, která vrací napětí článku, se nazývá *solarMeasure()*. Napětí článku je měřeno na dvě desetinná



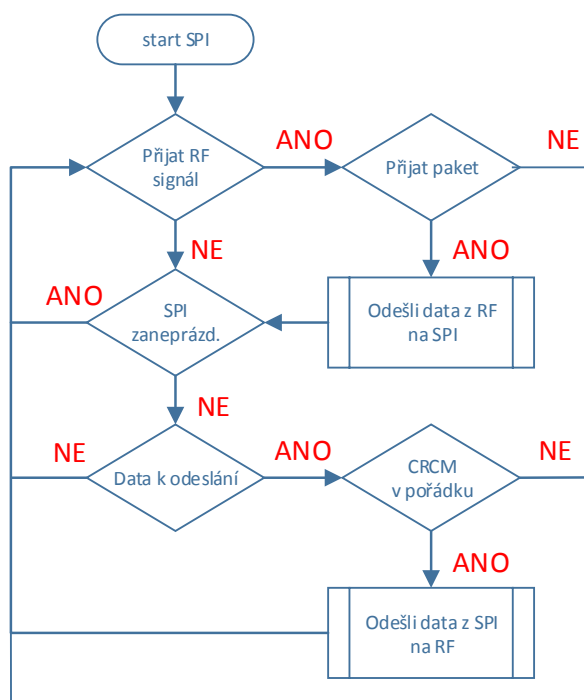
místa. Jelikož by uchování hodnoty napětí v desetinném formátu přineslo zvýšené paměťové nároky a zkomplikovalo další zpracování, tak funkce *solarMeasure()* vrací hodnotu napětí vynásobenou konstantou 100.

#### 5.4. BEZDRÁTOVÁ KOMUNIKACE

Software pro obsluhu bezdrátové komunikace se dá rozdělit do dvou okruhů. Prvním je program pro *IQRF* moduly a druhým je program pro řídicí jednotky, které zajišťují komunikaci s *IQRF*. Společné pro oba okruhy je komunikace po sběrnici *SPI*, která slouží jako prostředek pro výměnu dat mezi *MCU* (v roli *master*) a *IQRF* (v roli *slave*). V případě programového vybavení *IQRF* modulů byl využit upravený příklad pro komunikaci po *SPI* sběrnici dodávaný výrobcem. Pro komunikaci *MCU* s *IQRF* moduly byla využita a upravena dostupná knihovna *iqrf.h*.

##### 5.4.1. Obsluha modulů *IQRF*

Modul *IQRF* bezdrátově odesílá data přijatá ze sběrnice *SPI* a stejně tak na tuto sběrnici zapisuje data bezdrátově přijatá. Ve svém jádru tvoří bezdrátový most pro *MCU* komunikující po *SPI* sběrnici. Činnost *IQRF* modulů popisuje vývojový diagram na Obr. 5.8.



Obr. 5.8, Vývojový diagram obsluhy *IQRF* modulů



Nejprve je aktivována komunikace po sběrnici *SPI*. Pokud je detekován signál z jiného modulu, nastane příjem rámce. Po přijetí celého rámce jsou data odeslána na sběrnici. Jestliže není sběrnice zaneprázdněna a vyskytnou se na ní data k odeslání, pak algoritmus zkontroluje kontrolní součet *CRCM* a data odešle. Celý proces se následně opakuje.

#### 5.4.2. Obsluha MCU

Pro komunikaci řídicí jednotky s modulem *IQRF* lze využít standardních knihoven pro práci se sběrnici *SPI*. Jelikož je technologie *IQRF* velmi populární [9], nebylo těžké najít upravenou knihovnu pro práci s těmito moduly. Knihovna použitá v této práci se nazývá *iqrf.h* [10].

Ke komunikaci s bezdrátovými moduly stačí 3 funkce:

*SPI\_MasterInit()* ..... inicializace zařízení v *master* módu  
*IQRF\_GetData()* ..... příjem dat z *IQRF*  
*IQRF\_SendMessage()* ..... odeslání jednoho bytu dat do *IQRF*

#### 5.4.3. Přenášený rámec

Jelikož celý systém pracuje pouze se dvěma *IQRF* moduly, odpadla zde nutnost adresace zařízení. Základní rámec přenášený z ústředny do senzoru, na který ústředna očekává odpověď, je 2bytový a jedná se o sekvenci znaků datového typu *char*. Odpověď senzoru je 3bytová a obsahuje sekvenci znaků a čísel v závislosti na charakteru odpovědi. Speciální 3bytový rámec, který odesílá ústředna bez očekávané odpovědi, je nastavení doby nečinnosti senzoru na konci pracovní doby. Strukturou základního rámce odesílaného ústřednou, na který očekává odpověď, se zabývá Tab. 5.4.

Tab. 5.4, Základní bezdrátově přenášený rámec - ústředna

1. byte rámce	2. byte rámce	Význam
'S'	'A'	Žádost o potvrzení při navazování spojení
	'U'	Žádost o vrácení hodnoty napětí FV článku
	'L'	Žádost o vrácení hodnoty intenzity osvětlení
	'T'	Žádost o vrácení hodnoty teploty



Struktura speciálního rámce je popsána v Tab. 5.5. Speciální rámec ústředna vysílá vždy na konci pracovní doby a udává, na jak dlouho se má senzor přepnout do úsporného režimu. Hodnota *HH* ve 3. bytu určuje čas v hodinách, po který se senzor odmlčí a nebude s ním možno komunikovat.

Tab. 5.5, Speciální bezdrátově přenášený rámec - ústředna

1. byte rámce	2. byte rámce	3. byte rámce	Význam
'S'	'N'	<i>HH</i>	Nastavení doby nečinnosti senzoru

Odesláním základního rámce je ústředna schopna si vyžádat ze senzoru 4 druhy informací a tudíž existují 4 typy odpovědí, které odesílá senzor v závislosti na přijatém požadavku od ústředny.

Potvrzení spojení:

Při startu systému nejprve probíhá kontrola komunikace mezi ústřednou a senzorem. Ústředna odesílá sekvenci znaků 'SA' a očekává na ní odpověď, viz Tab. 5.6.

Tab. 5.6, Struktura odpovědi potvrzující bezdrátové spojení

Odpověď na rámec	1. byte rámce	2. byte rámce	3. byte rámce
'SA'	'I'	'V'	'O'

Pokud je obdržena odpověď s touto strukturou, je proces kontroly bezdrátové komunikace ukončen. V případě, že senzor neodpovídá nebo je odpověď chybná, pak se proces navazování spojení ještě 5x zopakuje. Pokud se ani poslední pokus nepovede, je ostatní chod systému zastaven.

Vrácení naměřené hodnoty:

Pokud senzor obdrží rámec, na který má odpovědět hodnotou dotazované veličiny, bude struktura odpovědi následující, viz. Tab. 5.7.

Tab. 5.7, Struktura odpovědi navracející dotazovanou veličinu

Odpověď na	1. byte	2. byte	3. byte
'SU'	'X'	napětí FV článku ( $V \cdot 100$ )	'N'
'SL'	'Y'	16bitová hodnota intenzity osvětlení (lx)	
'ST'	'Z'	teplota ( $^{\circ}\text{C}$ )	'N'



Hodnoty napětí fotovoltaického článku a teplota jsou 8bitové. Proto je poslední byte odpovědi doplněn znakem 'N'. Odesílaná hodnota teploty je zaokrouhlena na celé stupně celsia.

## 5.5. KALIBRACE A VYHODNOCOVÁNÍ DAT ZE SNÍMAČŮ

Kalibrace snímaných veličin by měla být vždy provedena po spuštění systému. Z pohledu uživatele je kalibrace velmi jednoduchá a stačí k ní luxmetr, v případě kalibrace snímačů intenzity osvětlení, nebo teploměr pro kalibraci snímače teploty. Po kalibraci systém upravuje změřená data ze snímačů na základě dat zadaných při procesu kalibrace.

### 5.5.1. Získávání dat ze snímačů

Pro účely odečítání aktuálních hodnot snímaných veličin byla vytvořena programová knihovna *measure.h*. Tato knihovna poskytuje ucelený nástroj pro odečítání dat ze snímačů umístěných na senzoru a ústředně. Základní funkce knihovny a jejich návratové hodnoty jsou popsány v Tab. 5.8.

Tab. 5.8, Základní funkce knihovny *measure*

Název funkce	Návratová hodnota	Popis
<i>measureInit()</i>		Inicializace měření, volaná pouze jednou při startu systému
<i>APDSmeasure()</i>	<i>uint16_t</i>	Vrací hodnotu v luxech ze snímače intenzity osvětlení na ústředně
<i>senzorAPDSmeasure()</i>	<i>uint16_t</i>	Vrací hodnotu v luxech ze snímače intenzity osvětlení na senzoru
<i>senzorDALLASmeasure()</i>	<i>uint8_t</i>	Vrací hodnotu ve stupních celsia ze snímače teploty na senzoru
<i>senzorSOLARmeasure()</i>	<i>uint8_t</i>	Vrací hodnotu napětí na fotovoltaickém článku ve voltech*100

Vrácená hodnota ze snímačů je vždy aritmetickým průměrem 3 po sobě jdoucích měření z důvodů minimalizace chyby při odečítání dat ze snímačů.



### 5.5.2. Kalibrace snímačů

Pro potřeby kalibrace byla vytvořena programová knihovna s názvem *calibration.h*. Ta má za úkol zpracovávat a uchovávat kalibrační data. Dalším úkolem této programové knihovny je výpočet relevantních dat na základě dat změřených. Jelikož odpadá potřeba kalibrace systému pro odečítání napětí na fotovoltaickém článku, nabízí knihovna celkem 6 + 1 základních funkcí. Pro každou snímanou veličinu je tedy k dispozici funkce pro kalibraci a funkce pro vrácení aktuální hodnoty snímané veličiny na základě kalibračních dat. Poslední funkce vrací příkaz pro *DALI* zařízení na základě požadované intenzity osvětlení.

Pro každou kalibrovanou veličinu je v paměti připraveno dvourozměrné pole schopné pojmout až 15 záznamů. Uložení záznamu provádí uživatel zadáním hodnoty do systému prostřednictvím uživatelského menu, viz. 6.2.4. Příklad záznamu je demonstrován na kalibraci snímače intenzity osvětlení, viz Tab. 5.9.

Tab. 5.9, Příklad záznamu kalibračních dat

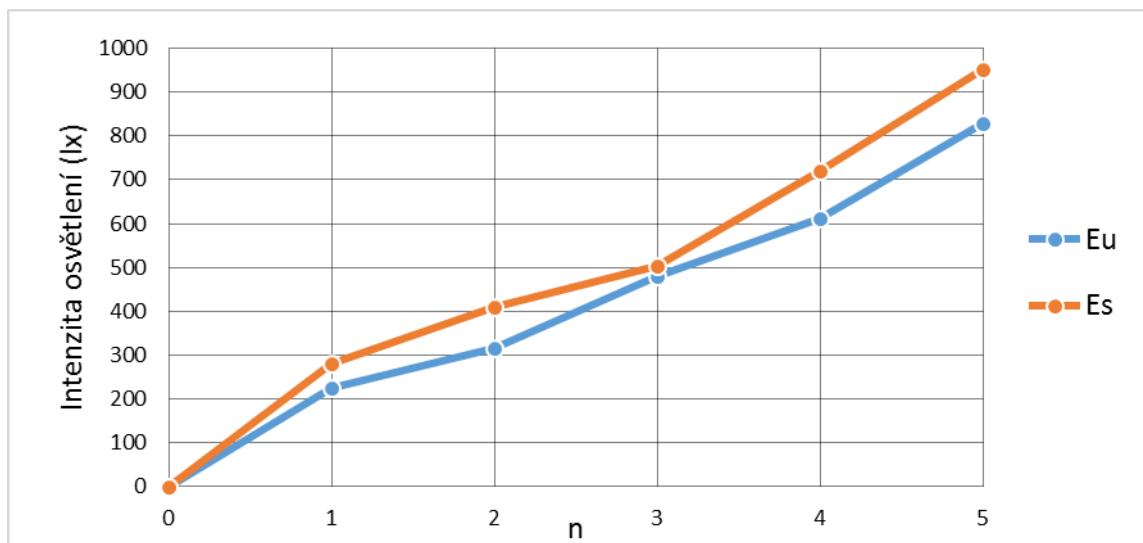
$n$	0	1	2	3	4	5
$E_S(n)$ (lx)	0	225	315	480	611	829
$E_U(n)$ (lx)	0	282	411	503	721	950

V prvním řádku dvourozměrného pole  $E_S$  se nachází hodnoty intenzity osvětlení odečtené z příslušného snímače dané veličiny. V řádku druhém  $E_U$  jsou uloženy hodnoty zadané uživatelem při kalibraci. Položka  $n$  udává pořadí záznamu v poli. Přehlednější náhled na příklad z Tab. 5.9 umožňuje grafické znázornění dat na Obr. 5.9.

Funkce knihovny *calibration.h* pro záznam kalibračních dat jsou následující:

*setUstrednaLux()*..... snímač intenzity osvětlení na ústředně  
*setSenzorLux()*..... snímač intenzity osvětlení na senzoru  
*setSenzorTemp()*..... snímač teploty na senzoru

Při kalibraci snímače intenzity osvětlení na ústředně se provádí taktéž záznam do pole pro následnou regulaci *DALI* zařízení, viz. Tab. 5.10. Ke každé zadané hodnotě v luxech je zaznamenána příslušná hodnota výkonu světelného zdroje, která je následně využita pro regulaci tohoto světelného zdroje.



Obr. 5.9, Demonstrace ukázkových kalibračních dat

### 5.5.3. Výpočet aktuálních hodnot na základě kalibrace

Pokud je systém kalibrován a některá z částí programu zažádá o vrácení snímané veličiny (vyjma procesu kalibrace) je navrácená hodnota vypočtena podle rovnice (5.1). V případě, že kalibrace neproběhla, vrácená hodnota odpovídá hodnotě přímo odečtené na snímači.

Výpočet probíhá na základě dat získaných z procesu kalibrace, kap 5.5.2. Algoritmus výpočtu je podobný lineární aproximaci a začíná zavoláním některé z následujících funkcí z knihovny *calibration.h*:

*getUstrednaLux()*.....intenzita osvětlení na ústředně  
*getSenzorLux()*.....intenzita osvětlení na senzoru  
*getSenzorTemp()*.....teplota na senzoru  
*setDaliPower()*.....nastavení DALI zařízení na požadovaný výkon

U všech výše uvedených funkcí je výstupem požadovaná veličina, vyjma funkce *setDaliPower()*, která nemá žádný výstup a naopak očekává vstupní argument odpovídající intenzitě osvětlení v luxech.

Algoritmus výpočtu výsledné hodnoty je demonstrován na funkci *getSenzorLux()*:

- 1) Po zavolání funkce *getSenzorLux()* je odečtena hodnota na snímači intenzity osvětlení  $E_S(x)$  na ústředně.

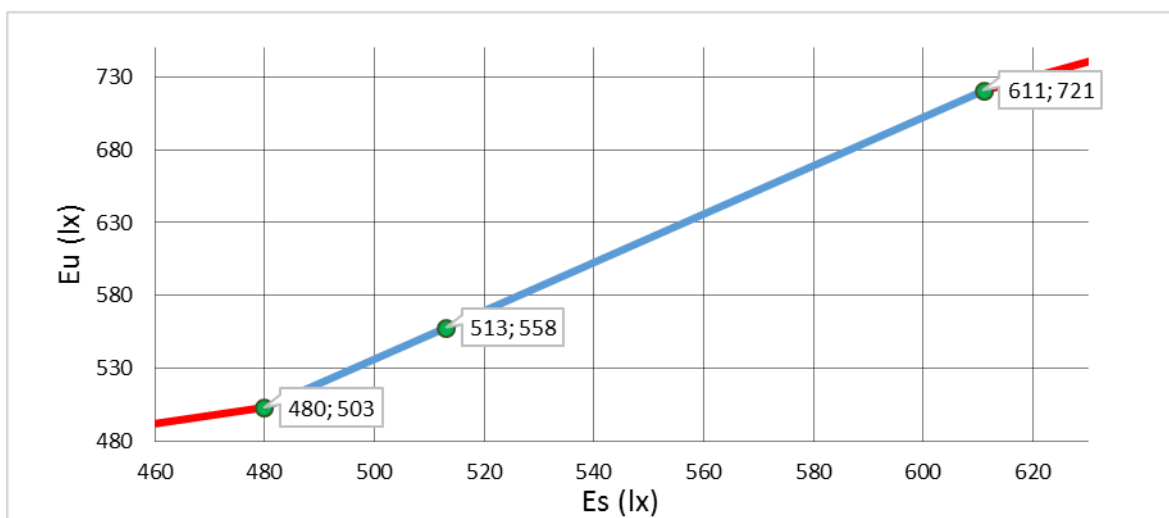


- 2) Dále algoritmus vyhledá, mezi jakými hodnotami  $E_S$  se hodnota  $E_S(x)$  nachází a uloží je do dočasných proměnných spolu s odpovídajícími hodnotami  $E_U$ .
- 3) Na základě výše uvedených dat lze vypočítat požadovanou hodnotu  $E_U(x)$ , dle vztahu (5.1).

$$E_U(x) = E_U(n) + \left( (E_U(n+1) - E_U(n)) \left( \frac{E_S(x) - E_S(n)}{E_S(n+1) - E_S(n)} \right) \right) \quad (5.1)$$

Metodu výpočtu  $E_U(x)$  lze názorně demonstrovat na příkladu z Tab. 5.9 a grafu na Obr. 5.9. Jako příklad, lze uvažovat, že hodnota odečtená na snímači intenzity osvětlení je  $E_S(x) = 513 \text{ lx}$ .

Tato hodnota se nachází mezi  $E_S(n) = 480 \text{ lx}$  a  $E_S(n+1) = 611 \text{ lx}$ , tomu odpovídá  $E_U(n) = 503 \text{ lx}$  a  $E_U(n+1) = 721 \text{ lx}$ . Po dosazení do vztahu (5.1) je hledaná hodnota  $E_U(x) = 558 \text{ lx}$ . Což odpovídá lineárnímu proložení obou bodů, viz Obr. 5.10.



Obr. 5.10, Lineární proložení bodů  $E_S$  a  $E_U$

Pokud je odečtená hodnota  $E_S(x)$  větší než poslední kalibrovaná hodnota  $E_S(n)$ , pak jsou do vztahu (5.1) dosazeny poslední dvě známe hodnoty  $E_S(n)$  a  $E_S(n+1)$ , resp.  $E_U(n)$  a  $E_U(n+1)$ .

## 5.6. REGULACE

Pro potřeby této části systému byla vytvořena programová knihovna *control.h*. Knihovna umožňuje přístup a uchování stavových veličin důležitých pro hlavní chod

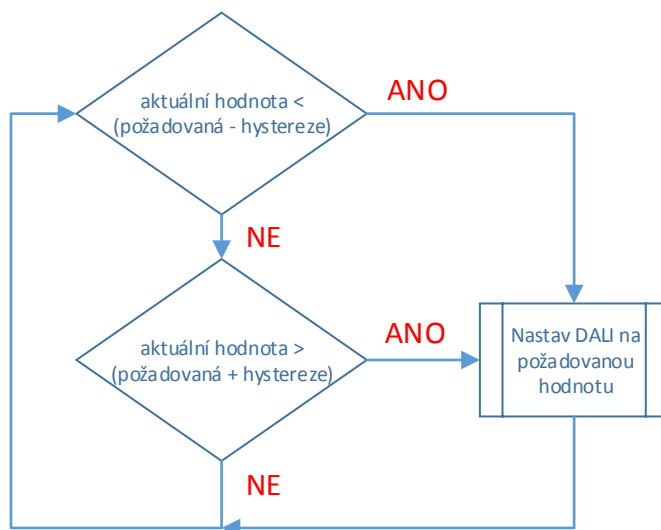


systemu. Jsou jimi například aktuálně nastavený režim, místo měření, pracovní doba, hystereze a jiné. Funkce, která má na starosti algoritmus regulace světelných zdrojů se nazývá *regulate()*.

### 5.6.1. Použitý regulátor

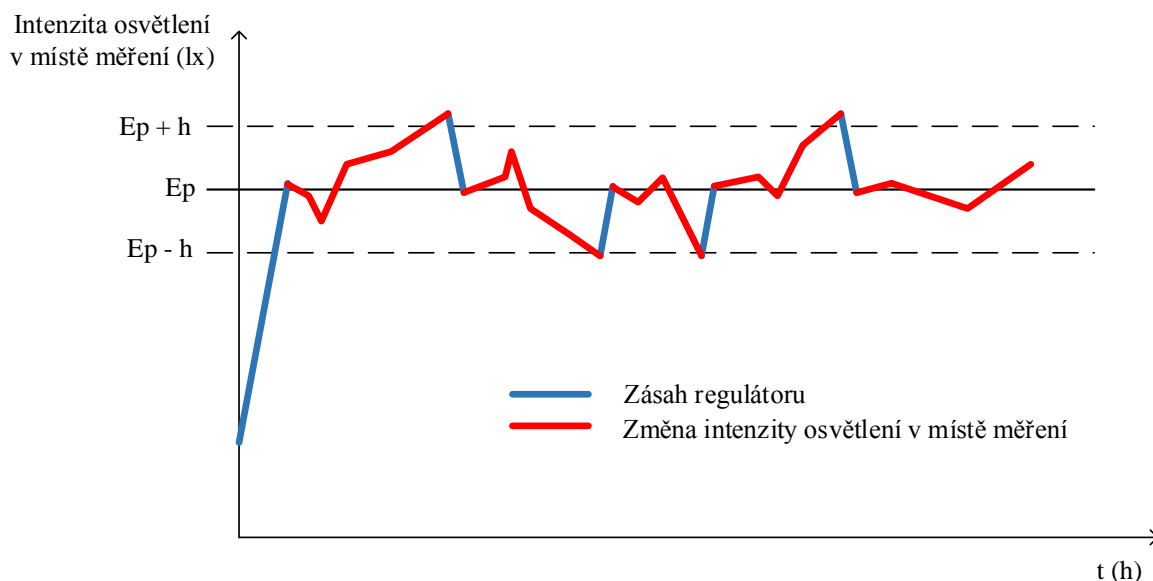
Vstupní data, která tvoří základ pro vyhodnocení regulace, jsou získávána z knihovny *calibration.h*. Algoritmus regulace je řízen aktuálním časem, který má na starosti knihovna *time.h*. Výstupem regulace je řízení světelných zdrojů prostřednictvím knihovny *DALImaster.h*.

Pro účely regulace byl navržen jednoduchý regulátor využívající principu nespojitě regulace. Aktivní je pouze v pracovní době a popisuje jej následující vývojový diagram, Obr. 5.11.



Obr. 5.11, Vývojový diagram regulátoru

Pokud je aktuální hodnota mimo rozsah hystereze  $h$ , pak regulátor nastaví světelné zdroje na požadovanou hodnotu  $E_p$ , resp. na hodnotu nejbližší, viz. kap. 5.6.4. Situace je demonstrována na příkladu trendu vývoje intenzity osvětlení v místě měření, Obr. 5.12.



Obr. 5.12, Příklad vývoje intenzity osvětlení v místě měření

### 5.6.2. Časování regulátoru

Časováním regulátoru se rozumí, jak často se odečítají data ze senzoru při regulaci a na jejich základě je regulace prováděna. Prodleva mezi odečítáním dat je věc testování a přizpůsobení systému dané aplikaci. Úzce souvisí se spotřebou senzoru a kvalitou regulace.

Zvolená prodleva mezi odečítáním dat je kompromisem mezi spotřebou senzoru a reakční dobou a byla pevně stanovena na 3 minuty. Kratší interval by mohl způsobit nepříjemné prudké změny osvětlení, delší interval naopak necitlivost regulátoru na rychlejší změny osvětlení.

Pauza mezi odečítáním dat ze senzoru byla také vzata v potaz při odhadování spotřeby senzoru při výběru baterií, kap. 4.3.1. Pokud by došlo k její změně, bude zapotřebí provést nový odhad.

### 5.6.3. Výstup regulátoru

Výstupem regulátoru je nastavení požadované hodnoty výkonu zdroje světla. Při procesu kalibrace snímače intenzity osvětlení na ústředně je zároveň vytvářena převodní tabulka mezi intenzitou osvětlení a výkonem světelného zdroje na *DALI* sběrnici, viz. kap. 5.5.2. Příklad záznamu je uveden v Tab. 5.10.

Tab. 5.10, Příklad převodní tabulky mezi  $E_S(n)$  a  $k(n)$ 

$n$	0	1	2	3	4	5
$E_S(n)$ (lx)	0	225	315	480	611	829
$k(n)$ [-]	1	41	66	84	102	139

Veličina  $k$  (3.1), kap. 3.2.4., je úměrná procentuální výkonu světelného zdroje. Při regulaci je hodnota  $k$  vypočítávána stejným algoritmem, který je popsán v kap. 5.5.3.

Pokud v průběhu regulace nastane potřeba změny výkonu světelného zdroje, pak je vypočtená hodnota  $k$  odeslána přímo na *DALI* sběrnici.

#### 5.6.4. Omezení regulátoru

Celý proces regulace má svá omezení, na které je potřeba dát pozor při instalaci systému, kalibraci, volbě velikosti hystereze a výběru světelných zdrojů. Každé z omezení má odlišný charakter, ale vzájemně se ovlivňují.

Pro dosažení co nejpřesnějších výsledků regulace je zapotřebí, aby byl systém správně kalibrován. S tím souvisí vhodné umístění ústředny, senzoru i místa kalibrace. Touto problematikou z hlediska uživatele se zabývá kap. 6.3.3. Kvalita kalibrace má tedy zásadní vliv na přesnost regulace.

Proces kalibrace naráží na další omezení regulátoru, kterým je maximální a minimální výkon světelného zdroje. Pokud například bude v monitorovaném místě instalován slabý zdroj světla, který ani při maximálním výkonu nebude schopný dosáhnout požadované hodnoty intenzity osvětlení, pak regulátor bude shora omezen maximálním výkonem zdroje světla. V případě, kdy regulovaným zařízením bude stínící technika, bude regulátor zdola omezen minimální hodnotou intenzity osvětlení, která se bude nacházet v monitorovaném prostoru při úplném zastínění.

Dalším omezením souvisejícím s předchozím je nárůst intenzity osvětlení při jednotkovém navýšení hodnoty  $k$  ze vztahu (3.1) na *DALI* zařízení. Pokud má například světelný zdroj maximální výkon  $P_{MAX} = 1000$  W a požadovaná hodnota  $k$  při regulaci bude 211, pak nastavený výkon zdroje  $P_Z$  bude dán vztahem (5.2).

$$P_Z(k) = \frac{P_{MAX} X(k)}{100} = \frac{1000 \cdot 10^{\frac{211-1}{253/3}-1}}{100} = 309,1 \text{ W} \quad (5.2)$$



Při jednotkovém navýšení  $k$  na hodnotu  $k+1$  se výkon zdroje v tomto případě zvýší o 8,57 W tedy na hodnotu 317,68 W. Obě hodnoty  $k$  i  $k+1$  jsou úměrné intenzitě osvětlení v monitorovaném místě. Jednotková změna výkonu světelného zdroje a jí úměrná hodnota intenzity osvětlení je tedy dalším omezením přesnosti regulace. Za zmínku stojí, že se přesnost regulace v závislosti na změně  $k$  se nemění lineárně. Je to dáno zejména logaritmickou závislostí procentuální změny výkonu na  $k$ .

Volba velikosti hystereze navazuje na předchozí omezení regulátoru. Pokud je zvolená hystereze menší než změna intenzity osvětlení odpovídající jednotkovému nárůstu, resp. poklesu výkonu světelného zdroje při jednotkovém nárůstu, resp. poklesu hodnoty  $k$  v daném rozsahu, pak na výstupu regulátoru bude cyklicky nastavována hodnota  $k+1$ , resp.  $k-1$ . Proto je zapotřebí volit alespoň 3x větší hodnotu hystereze než je změna intenzity osvětlení odpovídající nárůstu, resp. poklesu součtu změn výkonů světelného zdroje při nárůstu, resp. poklesu 3 po sobě jdoucích  $k$ , v daném rozsahu. Pokud je tedy například maximální výkon zdroje  $P_{MAX} = 1000$  W, nejedná se o stínící techniku a požadovaná intenzita osvětlení bude odpovídat  $k = 211$ , pak na základě vztahu (5.2) se nechá vypočítat součet změn  $P_Z(k)$  pro potřebná  $k$ .

$$\sum_{k=211}^{213} P_Z(k+1) - P_Z(k) = 8,57 + 8,76 + 9,04 = 26,37W \quad (5.3)$$

Ze vztahu (5.3) vyplývá, že minimální hystereze by měla odpovídat změně intenzity osvětlení při nárůstu výkonu světelného zdroje o 27 W od požadované hodnoty.

Pokud se požadovaná hodnota intenzity osvětlení v průběhu regulace mění, je potřeba určit minimální hystereze pro obě hraniční hodnoty z rozsahu požadovaných hodnot. Větší z těchto dvou hodnot je chápána jako celková minimální hystereze.

V praxi je náročné odvodit potřebné převodní vztahy mezi výkonem světelného zdroje a intenzitou osvětlení v místě měření. Jistou chybu také vnáší přesnost nastaveného výkonu na samotném světelném zdroji, která se liší výrobek od výrobku. Pro regulaci osvětlení v běžných prostorách, jako jsou školy, kanceláře apod., jsou běžně využívány světelné zdroje do 1000 W. V takovém případě lze obecně říci, že by se hystereze měla pohybovat mezi 50 lx až 70 lx. Předpokladem je však pečlivě provedená kalibrace, kap. 6.3.2.



## 6. UŽIVATELSKÉ OVLÁDÁNÍ A TESTOVÁNÍ

Kapitola má za úkol přiblížit celý systém z pohledu uživatele. Zejména pak pohyb v uživatelském menu a režimy ve kterých systém pracuje. Další částí této kapitoly je kalibrace snímačů intenzity osvětlení z pohledu uživatele. Poslední část je věnována testování. Jsou zde diskutovány výsledky provedených testů, a jakým způsobem bylo výsledků dosaženo.

### 6.1. SPOUŠTĚNÍ SYSTÉMU

Start systému provází několik procedur, které musí systém vykonat před tím, než pustí uživatele do uživatelského menu. Aby byl zajištěn správný chod systému, je nutné, aby při startu systému proběhla kontrola, zda je funkční komunikace mezi senzorem a ústřednou a také zda je k systému připojeno *DALI* zařízení. O průběhu těchto kontrolních fází je uživatel informován prostřednictvím dvouřádkového displeje [14].

Nejprve proběhne pokus o navázání spojení, viz. Obr. 6.1.

S	E	N	Z	O	R	.	.	.							
N	A	V	A	Z	U	J	I		S	P	O	J	E	N	I

Obr. 6.1, Pokus o navázání spojení

Pokud je spojení úspěšně navázáno, systém zobrazí následující zprávu, Obr. 6.2.

S	E	N	Z	O	R	.	.	.							
S	P	O	J	E	N	I		N	A	V	A	Z	A	N	O

Obr. 6.2, Úspěšně navázané spojení

Při neúspěšném pokusu o navázání spojení se na displeji zobrazí zpráva viz. Obr. 6.3 a celý proces navazování spojení se pětkrát opakuje.

S	E	N	Z	O	R	.	.	.							
C	H	Y	B	A		S	P	O	J	E	N	I			

Obr. 6.3, Chyba při pokusu o navázání spojení



Pokud ani po pátém pokusu o navázání spojení senzor nereaguje, obrazovka Obr. 6.3 setrvá a uživatel není vpuštěn do uživatelského menu, kap. 6.2.

Proces navazování spojení s *DALI* zařízením je provázen stejnými obrazovkami a postupem pouze s rozdílem prvního řádku displeje, kde namísto nápisu “SENZOR...” se uživateli zobrazí nápis “DALI...”.

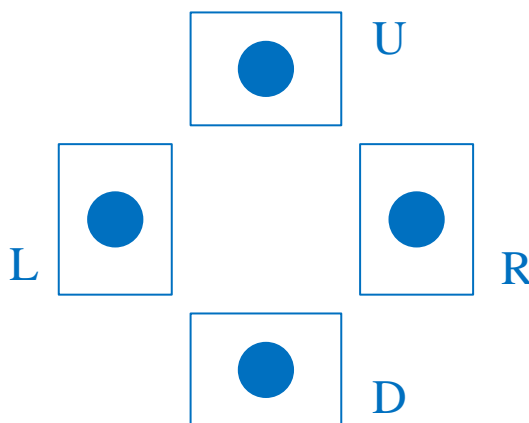
## 6.2. UŽIVATELSKÉ MENU

Uživatelské menu slouží uživateli ke změně aktuálního režimu, nastavení veličin ovlivňujících chod systému, přímému odečítání hodnot a kalibraci. V základu obrazovka poskytuje informaci o aktuálním režimu, čase a nastavené hodnotě osvětlení, viz. Obr. 6.4. Pro pohyb a zobrazení menu slouží tlačítka.

R	E	Z	I	M	:		M	A	N	U	A	L	N	I	
1	9	:	2	5			L	U	X	:		3	5	0	

Obr. 6.4, Základní obrazovka

Rozmístění tlačítek na ovládacím panelu je ilustrováno na Obr. 6.5.



Obr. 6.5, Rozmístění tlačítek

Kompletní menu se nachází v příloze E. Vstup do menu se provádí pomocí stisku klávesy **D**. Pro listování v menu jsou určena tlačítka **U** a **D**, přičemž tlačítko **U** slouží pro listování směrem nahoru a tlačítko **D** opačným směrem. Celkem je na výběr ze 4 základních položek menu a jsou jimi:



- REZIM
- NASTAVENI
- MERENI
- KALIBRACE

Příklad zobrazení se nachází na Obr. 6.6.

N	A	S	T	A	V	E	N	I												

Obr. 6.6, Příklad zobrazení položky v základním menu

Výběr dané položky se provádí stiskem tlačítka **R**. Pro návrat je zapotřebí stisknout tlačítko **L**. Obsahem jednotlivých položek se zabývají následující kapitoly.

### 6.2.1. Výběr režimu

Po výběru položky REZIM v základním menu má uživatel na výběr 3 režimy chodu systému. V podstatě se jedná o nastavení hodnoty v luxech, které chce uživatel dosáhnout ve zvoleném místě.

#### 1) *OKAMZITY*

Tento režim umožňuje uživateli okamžité nastavení hodnoty v luxech v daném místě měření.

#### 2) *AUTOMATICKY*

Automatický režim umožňuje udržovat osvětlení v místnosti dle normy pro světelnou hygienu. Na výběr je ze 4 typů prostor (Prodejna – 300 lx, Provozovna – 600 lx, Kancelář – 500 lx, Škola – 500 lx).

#### 3) *MANUALNI*

Režim Manuální umožňuje uživateli nastavit hodnotu v luxech pro každou hodinu dne. K dispozici jsou 3 scény, kam je možné si uložit své nastavení. Nastavení jednotlivých scén probíhá v menu NASTAVENI, kap. 6.2.2.

Příklad zobrazení položky v menu REZIM se nachází na Obr. 6.7.



R	E	Z	I	M													
A	U	T	O	M	A	T	I	C	K	Y							

Obr. 6.7, Příklad zobrazení položky v menu REZIM

Po výběru daného režimu následuje specifikace režimu a jeho potvrzení. Příklad na Obr. 6.8 ilustruje nastavení režimu AUTOMATICKY a jeho varianty SKOLA.

S	K	O	L	A													
N	A	S	T	A	V	E	N	O									

Obr. 6.8, Příklad nastavení položky v režimu AUTOMATICKY

### 6.2.2. Nastavení

Po přechodu do druhé položky v základním menu je uživateli umožněno měnit nastavení scén, času, pracovní doby, místa měření a hystereze. Dále je možné přímo zvyšovat či snižovat výkon zdroje světla připojeného na *DALI*.

Zobrazení jednotlivých položek v menu NASTAVENI je obdobné jako v menu REZIM na Obr. 6.7. První 3 položky menu jsou určeny pro nastavení jednotlivých scén a jejich konfigurace probíhá následovně.

Stiskem tlačítka **R** uživatel provede výběr scény, kterou si přeje nastavit. Opětovným stiskem tlačítka **R** probíhá výběr daného časového rozmezí ve scéně. Nastavení hodnoty v luxech pro dané časové rozmezí probíhá pomocí tlačítek **U** a **D**. Příklad na Obr. 6.9 znázorňuje nastavení hodnoty 425 lx v časovém rozmezí mezi 14:00 a 15:00. Návrat do předchozího menu se provádí stiskem tlačítka **L**.

1	4	h	-	1	5	h											
L	U	X	:		4	2	5										

Obr. 6.9, Nastavování časových úseků ve scéně

Čtvrtou položkou, kterou je možno nastavit je čas. Po výběru položky se uživateli zobrazí obrazovka s možností nastavení aktuálního času, viz. Obr. 6.10. Nastavení času se provádí tlačítkem **U** a **D**, přičemž tlačítko **U** přidává k zobrazenému času jednu





minutu a tlačítko **D** jednu minutu ubírá. Pokud jsou minuty na nule, potom stiskem tlačítka **D** se k nastavenému času přidá jedna hodina.

N	A	S	T	A	V	E	N	I											
1	2	:	2	4															

Obr. 6.10, Nastavení aktuálního času

Další položkou je nastavení pracovní doby. Počátkem pracovní doby se zapíná proces regulace a na jejím konci je proces regulace vypnut. Její nastavení probíhá obdobně jako při nastavení aktuálního času. Pracovní doba se nechá nastavit na celé hodiny a obrazovka pro její nastavení je zobrazena na Obr. 6.11. Při návrhu napájení senzoru byla uvažovaná pracovní doba 10ti hodinová, viz. kap. 4.3.1.

O	D		-		D	O													
1	0		-		1	8													

Obr. 6.11, Nastavení pracovní doby

V pořadí 6. položka umožňuje výběr místa měření intenzity osvětlení. Zobrazení volby výběru místa měření je podobné výběru režimu na Obr. 6.8. Na výběr je ze dvou míst měření a to senzor a ústředna. Touto volbou uživatel nastavuje, zda si přeje, aby pro výslednou regulaci bylo využito dat ze snímače intenzity osvětlení v místě senzoru nebo ústředny.

Předposlední položka, je věnována nastavení výkonu *DALI* zařízení. Uživateli je zobrazena obrazovka na Obr. 6.12. Stiskem tlačítka **U** proběhne zvýšení světelného výkonu o jednotku  $k$  odpovídající procentuálního výkon světelného zdroje, který je dán vztahem (3.1). Snížení je možné provést stiskem tlačítka **D**.

U	P																		
D	O	W	N																

Obr. 6.12, Nastavení světelného výkonu *DALI* zařízení



Poslední hodnotou, kterou je možno nastavit je hystereze. Lze buďto zvyšovat pomocí tlačítka **U** nebo snižovat tlačítkem **D**. Na Obr. 6.13 se nachází obrazovka, která je zobrazena při nastavování hystereze.

L	U	X	:		5	0												

Obr. 6.13, Nastavení hystereze

### 6.2.3. Měření

Menu měření slouží k přímému odečítání hodnot z jednotlivých snímačů. Pokud je systém po kalibraci, pak hodnoty zobrazené v tomto menu odpovídají již kalibrovaným hodnotám. Uživatel si může postupně nechat zobrazit 4 snímané veličiny:

- Intenzita osvětlení v místě ústředny
- Intenzita osvětlení v místě senzoru
- Teplota v místě senzoru
- Napětí na fotovoltaickém článku

Pohyb v tomto menu je stejný jako v předešlých. Příklad na Obr. 6.14 znázorňuje měření hodnoty intenzity osvětlení v místě ústředny (stejná obrazovka i v místě senzoru).

L	U	X	:		5	5	7											

Obr. 6.14, Příklad měření intenzity osvětlení

### 6.2.4. Kalibrace

Poslední položkou v základním menu je KALIBRACE. Při potvrzení výběru je uživateli umožněno kalibrovat oba snímače intenzity okolního osvětlení a snímač teploty. Proces kalibrace je detailně popsán v kap. 5.5.

Pohyb v menu je stejný jako v předchozích případech. Zadání skutečné hodnoty do systému probíhá obdobně jako nastavení hystereze, viz. kap. 6.2.2. V případě kalibrace je nutné navíc hodnotu potvrdit stiskem tlačítka **R**. Uživateli se následně zobrazí obrazovka z Obr. 6.15 (příklad kalibrace snímače intenzity osvětlení).



L	U	X	:	6	2	2								
N	A	S	T	A	V	E	N	O						

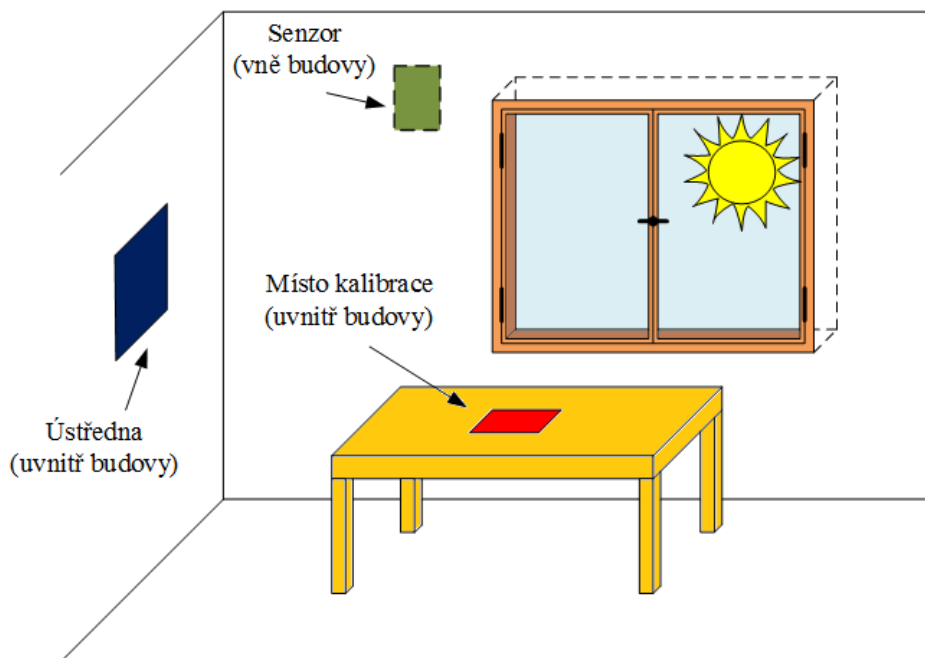
Obr. 6.15, Kalibrace snímače intenzity osvětlení

### 6.3. KALIBRACE SNÍMAČŮ OSVĚTLENÍ Z POHLEDU UŽIVATELE

Kalibrací a zvolením místa měření by měl uživatel začít pokaždé po uvedení nebo znovuuvedení systému do provozu. Pokud u systému není provedena kalibrace, nebude pracovat správně, jelikož na těchto datech závisí jak vyhodnocování naměřených dat, tak i regulace světelných zdrojů.

#### 6.3.1. Rozmístění periferií

Myšlenka kalibrace snímačů intenzity osvětlení spočívá v možnosti uživatele přizpůsobit si měřenou veličinu podmínkám, ve kterých má být udržována daná hladina osvětlení. Příkladem je umístění senzoru, ústředny a místa kalibrace na Obr. 6.16.



Obr. 6.16, Ilustrace příkladu kalibrace snímačů intenzity osvětlení

Pokud bude senzor umístěn vně budovy, měl by být umístěn co nejblíže oknům nebo otvorům, přes která k požadovanému místu regulace proniká nejvíce světla



z vnějších světelných zdrojů. Pokud se uživatel rozhodne instalovat senzor uvnitř místnosti, měl by jej instalovat co nejbližší k požadovanému místu regulace.

Ústředna by se měla nacházet v místě, kde se nachází zdroje světla ovlivňující intenzitu osvětlení v místě kalibrace. Nejlépe je ji připevnit na stěnu tak, aby snímač osvětlení zabíral celý monitorovaný prostor.

### **6.3.2. Proces kalibrace**

Ke kalibraci je nejlépe využít cejchovaný luxmetr. Přístroj je zapotřebí umístit do místa, pro které chce uživatel celý systém kalibrovat. Na pořadí kalibrace snímačů intenzity osvětlení nezáleží, potřebné je však provést kalibraci u obou snímačů.

Po té co uživatel nastaví světelné podmínky pomocí menu NASTAVENI - DALI, viz. kap. 6.2.2, zadá do ústředny hodnotu z luxmetru. Zadávání kalibračních dat do systému je blíže popsáno v kap. 6.2.4. Čím větší bude počet a rozsah kalibračních dat, tím lépe bude systém regulovat na požadovanou hodnotu. Nejlépe je provést kalibraci v rozsahu od 0 až do maximální intenzity osvětlení v místě regulace.

Pokud je jako místo měření zvolen senzor, pak zadávané hodnoty při procesu kalibrace snímače intenzity osvětlení na ústředně by se měly pohybovat v rozmezí předpokládané požadované hodnoty intenzity osvětlení při regulaci. Pokud je například předpoklad, že rozsah požadovaných hodnot v průběhu regulace bude 300 lx až 600 lx, měl by v podobném rozsahu být kalibrován i snímač intenzity osvětlení na ústředně.

Kalibrační data se ztrácí při odpojení ústředny od napájení. To lze využít například při potřebě opětovného kalibrování systému.

### **6.3.3. Výběr místa měření**

Z pohledu uživatele má nastavení místa měření výhodu v podobě možnosti volby výběru, kde se má místo regulace nacházet. Spolu s vhodně umístěným senzorem je tak možno zajistit, aby regulace probíhala v jiném místě, než je instalována ústředna. Ústředna tak může být umístěna v prostoru, kde je vyvedeno síťové napájení, avšak měla by se nacházet v místě, kde se nachází zdroje světla ovlivňující intenzitu osvětlení v monitorovaném prostoru.

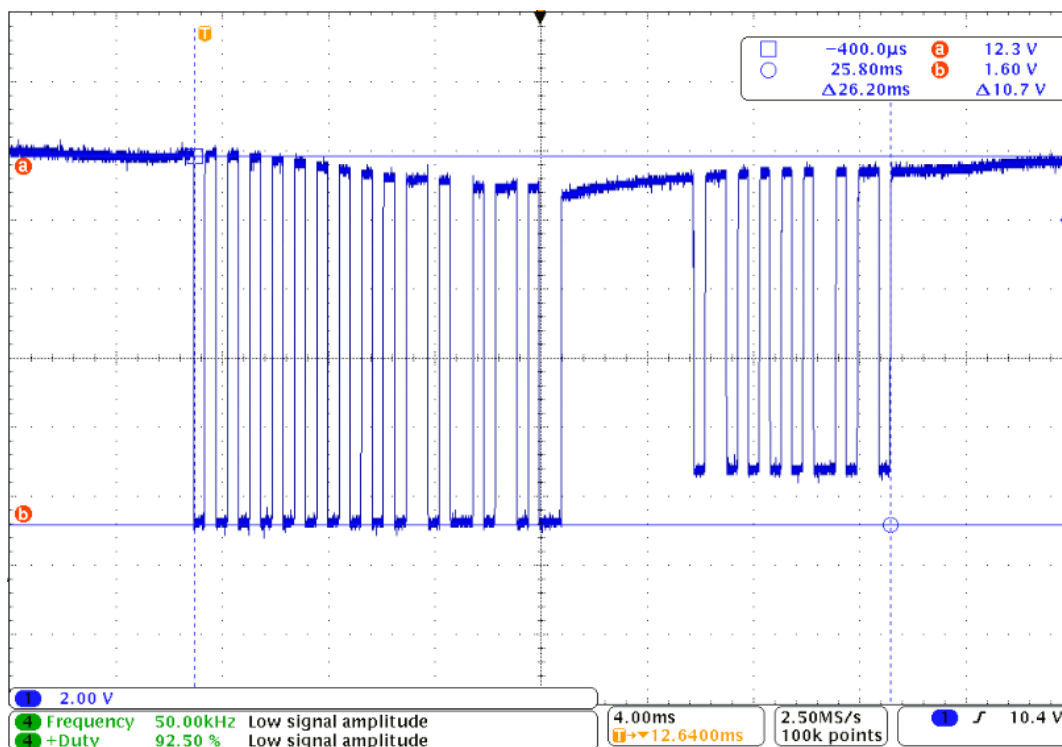
Pokud uživatel zvolí jako místo měření ústřednu, pak jsou data ze senzoru v průběhu regulace ignorována (resp. nejsou odečítána).

## 6.4. TESTOVÁNÍ

Prvotní testy probíhaly na nepájivém poli a po výrobě a osazení desek plošných spojů ústředny a senzoru byly zopakovány na hotových přípravných. Jedny z prvních testů byly věnovány komunikaci integrovaných obvodů s řídicí jednotkou. Následovaly testy *DALI* sběrnice a bezdrátové komunikace, jímž je věnován začátek této kapitoly. Ve druhé části kapitoly jsou diskutovány testy napájení senzoru a výsledky nasazení systému do zkušebního provozu.

### 6.4.1. Testování komunikace po *DALI* sběrnici

Časově nejnáročnější testování proběhlo po výrobě *DALI* modulu, kap. 4.2.3. Nejprve byly provedeny softwarové testy zaměřené na správné časování bitů v rámci. Po odladění chyb byly provedeny hardwarové testy, kde byla největší pozornost věnována napájení sběrnice. Bylo zapotřebí zjistit, zda při komunikaci DC/DC konvertor, použitý jako zdroj napětí, udrží požadované velikosti napětí definované normou [8]. Testy ukázaly, že při odesílání klesá hodnota napětí log. 1. Pro úplnou komunikaci je zapotřebí odesílat maximálně 3bytové zprávy. Pokles napětí na konci rámce však není v rozporu se standardem, tudíž tento pokles napětí není závadou.



Obr. 6.17, Odeslaný a přijatý rámec na *DALI*



Na Obr. 6.17 je záznam komunikace na *DALI* sběrnici. Rámec odeslaný z ústředny je adresován všem zařízením (adresový byte = 1111111) a odeslaný příkaz se nazývá “*QUERY DEVICE TYPE*“ (datový byte = 10011001). Odesláním toho rámce je očekávána odpověď, která obsahuje informaci o typu zařízení. Testy byly provedeny na předřadníku pro *LED* svítidla do 25 W *FOXTRON proLED25*, který v případě na Obr. 6.17 odpoví bytem 00000110, což odpovídá předřadníkům tohoto typu.

#### **6.4.2. Testování bezdrátové komunikace**

Jak již bylo zmíněno v kap. 5.4, moduly *IQRF* komunikují s řídicí jednotkou prostřednictvím *SPI* sběrnice. Sběrnice funguje obdobně jako kruhový registr. Problém při testování vznikl při přijímání dat, a to na straně řídicí jednotky ústředny. Moduly *IQRF* totiž sledují, zda data přijatá na sběrnici *SPI* jsou aktuální a v tomto případě je odešle. V případě ústředny bylo zapotřebí zajistit, aby data ze senzoru, vrácená některou z funkcí knihovny *measure.h*, byla také aktuální. Po přijetí poslední bezdrátové zprávy totiž tato data zůstávají na sběrnici a je tedy obtížné určit, zda data byla právě přijata, nebo zda se jedná o data z posledního vysílání. Také doba měření závisí na charakteru měřené veličiny.

Závěrem testování byla úprava algoritmu přijímání zpráv. Funkce, která kontroluje přijetí zpráv je volána v cyklu s omezením počtu průchodů na 20. Pokud v této době nejsou, po odeslání požadavku, data přijata, pak funkce vrátí chybovou hodnotu. V opačném případě vrátí aktuální hodnotu požadované veličiny.

#### **6.4.3. Testování napájení senzoru**

Jelikož byly jmenovité hodnoty fotovoltaiického článku a baterií spočítány na základě odhadované spotřeby senzoru, kap. 4.3, bylo provedeno testování napájení senzoru. Před provedením testu byly baterie plně nabity. Senzor byl umístěn vně oken, kde byl vystaven odraženému slunečnímu záření. Po dobu 25ti dní z něj byly odečítány hodnoty každé 3 minuty v rozmezí pracovní doby 8:00 až 18:00. Během testu nedošlo k výpadku napájení senzoru. Po ukončení testu byla kapacita baterií 84 %.

#### **6.4.4. Zkušební provoz systému**

Během prvního dne testování napájení senzoru, kap. 6.4.3, bylo simulováno zkušební nasazení systému do provozu. Jako zdroj světla byl použit 4 W *LED* diodový



pás řízený napětím přes *DALI* předřadník použitý v kap. 6.4.1. Zdroj světla byl umístěn 1 m od snímače ústředny. Po provedení kalibrace byl nastaven režim “Škola“ s hysterezí 50 lx. Jako místo měření byl zvolen senzor. Ostatní nastavení a okolnosti testování odpovídají kap. 6.4.3.

Po prvním dni testování bylo potřeba doplnit kalibrační data obou snímačů intenzity osvětlení, jelikož při spouštění systému nebylo možné navodit dostatečné světelné podmínky pro další fungování systému. Důvodem bylo špatné počasí, při kterém intenzita venkovního osvětlení nedosahovala takových hodnot, jako ve dni druhém, kdy naopak svítilo slunce.

Do místa kalibrace byl umístěn luxmetr *Lutron LX-105* a druhý den testování byla data v místě kalibrace vyhodnocována. Data získaná z měření se nachází v Tab. 6.1, kde  $E_k$  označuje intenzitu osvětlení v místě kalibrace.

Tab. 6.1, Naměřená data při zkušebním provozu

Čas měření	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00
$E_k$ (lx)	495	484	472	460	532	526	485	533	511

Hladina intenzity osvětlení při volbě režimu “Škola“ je 500 lx. Zvolená hystereze byla 50 lx. Na základě naměřených dat v Tab. 6.1 je vidět, že ani jedna z naměřených hodnot intenzity osvětlení v místě kalibrace nepřekračuje požadovanou hladinu osvětlení včetně hystereze, tedy  $500 \text{ lx} \pm 50 \text{ lx}$ .



## 7. ZÁVĚR

Hlavní cíl této práce, tj. navrhnout a realizovat systém pro vyhodnocení světelných podmínek s následnou regulací světelných zdrojů v monitorované místnosti, v závislosti na venkovních světelných podmínkách, byl splněn. Systém umožňuje uživateli nastavit minimální úroveň osvětlení dané normou pro světelnou hygienu v závislosti na typu prostoru. Na výběr je ze čtyř typů prostor a to: škola, kancelář, provozovna a prodejna. Uživatel si také může libovolně měnit úroveň osvětlení v jednotlivých hodinách dne, přičemž svá nastavení si může ukládat do třech scén.

Výstupem systému je jeho napojení na moderní sběrnici pro osvětlovací techniku *DALI*. Tato sběrnice je v současnosti nejpoužívanějším médiem ve sféře inteligentního bydlení pro komunikaci mezi světelnými zdroji a prvky, které je regulují.

Ve fázi návrhu elektrických schémat, zejména desek plošek spojů, bylo zapotřebí si nejprve osvojit základy práce s vývojovým prostředím *Cadence 16.6*, což bylo jedním z dílčích úkolů práce. Navzdory předchozí zkušenosti autora práce s vývojem desek plošných spojů v prostředí *Eagle*, vyžaduje práce v prostředí *Cadence 16.6* zkušenějšího uživatele, což způsobilo některé nedostatky v návrhu výsledných desek plošných spojů. Nedostatky však nemají zásadní vliv na funkčnost systému.

Použití stejných mikroprocesorů u obou periférií zjednodušilo vývoj programového vybavení, zejména v podobě snadné přenositelnosti kódu. Při další výrobě periférií by stálo za zvážení použití jednodušší verze MCU na senzoru, jelikož schopnosti použité řídicí jednotky převyšují požadavky této periferie, zejména počet vstupů/výstupů, velikost paměti a rychlost čipu. V případě ústředny je stávající velikost paměti použitého mikroprocesoru sice dostačující, ale pro další rozšíření nevyhovující, jelikož je při současném programovém vybavení z 93 % zaplněna.

Zapojení systému do moderních sítí inteligentního bydlení jej předurčuje pro jeho další rozšíření. Proto je systém navržený tak, aby jeho případné rozšíření nevyžadovalo velký zásah v podobě rozsáhlé změny elektrických schémat a programového vybavení. Jednou z možností jak systém obohatit je bezdrátově ho připojit k systémům pro regulaci teplot.

Jako další pokračování této práce by bylo vhodné upravit desku plošných spojů ústředny. Vyrobené moduly by se tak mohly stát součástí ústředny. Dále by bylo dobré provést úpravu systému měření napětí fotovoltaického článku, tak aby měřené napětí článku nebylo zatížené spotřebou celé periferie.





Uvedení systému do reálného provozu by znamenalo získání dalších údajů pro případné vylepšení celku. Získané údaje by se tak nechaly využít například pro zlepšení procesu kalibrace snímačů či zkvalitnění regulace světelných zdrojů. Z uživatelského hlediska by zlepšení spočívalo v ovládní a struktuře systémového menu.

Navzdory některým nedostatkům a možným zdokonalením systému, práce autorovi posloužila ke zdokonalení se v oblasti návrhu elektrických schémat a desek plošných spojů. Nemalou měrou také přispěla k prohloubení znalostí o sběrníkových systémech a systémech pro inteligentní řízení osvětlení. Dalším přínosem byla zkušenost s bezdrátovými technologiemi a výběrem fotovoltaického článku.



## 8. POUŽITÁ LITERATURA

- [1] ŠUBRT, Vladimír. *Mikrokontroléry ATMEL AVR: vývoj aplikací*. 1. vyd. Praha: BEN - technická literatura, 2002, 95 s. ISBN 80-730-0055-5.
- [2] VEDRAL, Josef a Jan FISCHER. *Elektronické obvody pro měřicí techniku: vývoj aplikací*. Vyd. 2. Praha: Vydavatelství ČVUT, 2004, 340 s. ISBN 80-010-2966-2.
- [3] ČSN EN 12464-1. *Světlo a osvětlení: Osvětlení pracovních prostorů*. ČESKÝ NORMALIZAČNÍ INSTITUT, Praha: XEROX CR, s.r.o, 2004.
- [4] KRAJÍČEK, Tomáš. *Moderní bezdrátová technologie - ZigBee*. Brno, 2009. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Michal Skořepa.
- [5] KNX (standard). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2014 [cit. 2014-04-16]. Dostupné z: [http://en.wikipedia.org/wiki/KNX\\_\(standard\)](http://en.wikipedia.org/wiki/KNX_(standard)).
- [6] INFINEON TECHNOLOGIES AG. *White LED Card: For XMC1000 Family*. 1.0. Munich, Germany, 2013, 16 s. Edition 2013-03-08. Dostupné z: [http://www.infineon.com/dgdl/Board\\_Users\\_Manual\\_White\\_LED\\_Card\\_R1.pdf?folderId=db3a30433580b3710135a47f3eb76c98&fileId=db3a30433dd42dcf013dd458419d005c&ack=t](http://www.infineon.com/dgdl/Board_Users_Manual_White_LED_Card_R1.pdf?folderId=db3a30433580b3710135a47f3eb76c98&fileId=db3a30433dd42dcf013dd458419d005c&ack=t).
- [7] ZVEI. *AG DALI: Oficiální sdružení DALI* [online]. [cit. 2014-04-16]. Dostupné z: <http://www.dali-ag.org/>.
- [8] NEMA. *Nema Standard Publication 243 – 2004: General Requirements*. 2004. Dostupné z: [http://www.archenergy.com/lrp/lightingperf\\_standards](http://www.archenergy.com/lrp/lightingperf_standards).
- [9] SULC, Vladimír. Malé, chytré, české... Bezdrátová platforma IQRF. *HW server s.r.o.* [online]. 2010 [cit. 2014-04-16]. Dostupné z: <http://www.hw.cz/rf/male-chytre-ceske-bezdratova-platforma-iqrf.html>.
- [10] KARAS, Ondřej. IQRF Prakticky V - SPI. *HW server s.r.o.* [online]. 2008 [cit. 2014-04-16]. Dostupné z: <http://www.hw.cz/teorie-a-praxe/konstrukce/iqrf-prakticky-v-spi.html>.
- [11] OLEJÁR, Martin. Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroléru PIC16F877. *HW server s.r.o.* [online]. 2000, 2005-05-20 [cit. 2014-04-16]. Dostupné z: <http://www.hw.cz/navrh->



obvodu/strucny-popis-sbornice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-  
eeprom-24lc256.

- [12] SVADBÍK, Vít. Ovládání teplotního čidla Dallas DS18B20 pro uC AVR. *VS* [online]. 2013, 2013-08-19 [cit. 2014-04-16]. Dostupné z: <http://www.vsvadbik.cz/products/ovladani-teplotniho-cidla-dallas-ds18b20-pro-uc-avr/>.
- [13] JEDLIČKA, Jan. Komunikace s více teplotními čidly DS18B20 na společné sběrnici 1wire. *JedlaTech* [online]. [cit. 2014-04-16]. Dostupné z: <http://jedla.4fan.cz/programovaniAVR/1wire&ds18B20/1wire&ds18B20.php>.
- [14] FLEURY, Peter. LCD library for HD44870 based LCD's: AVR-GCC libraries. *AVR-Software* [online]. 2006, 2006-01-30 [cit. 2014-04-16]. Dostupné z: [http://homepage.hispeed.ch/peterfleury/group\\_\\_pfleury\\_\\_lcd.html](http://homepage.hispeed.ch/peterfleury/group__pfleury__lcd.html).
- [15] FLEURY, Peter. I2C Master Interface: AVR-GCC libraries. *AVR-Software* [online]. 2006, 2005-03-29 [cit. 2014-04-16]. Dostupné z: [http://homepage.hispeed.ch/peterfleury/group\\_\\_pfleury\\_\\_ic2master.html](http://homepage.hispeed.ch/peterfleury/group__pfleury__ic2master.html).
- [16] KNOTEK, Martin. *Implementace D.A.L.I. protokolu pro STM8 mikrokontroler*. Praha, 2009. Diplomová práce. České vysoké učení technické v Praze. Vedoucí práce Ing. Jan Fischer, CSc..
- [17] ZÁHLAVA, Vít. *Přednášky předmětu A0B34PPN: Principy a pravidla elektronického návrhu* [online]. Praha, 2013, 2013-03-21 [cit. 2014-04-16]. Dostupné z: <http://www.micro.feld.cvut.cz/home/zahlava/ppn/prednasky/>.
- [18] Sluneční energie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2014 [cit. 2014-04-16]. Dostupné z: [http://cs.wikipedia.org/wiki/Slune%C4%8Dn%C3%AD\\_energie](http://cs.wikipedia.org/wiki/Slune%C4%8Dn%C3%AD_energie).
- [19] AVAGO TECHNOLOGIES. *Katalogový list APDS-9301: Miniature Ambient Light Photo Sensor with Digital (I2C) Output*. Spojené státy americké, 2010. Dostupné z: <http://www.avagotech.com/docs/AV02-2315EN>.
- [20] MAXIM INTEGRATED. *Katalogový list DS18B20: Programmable Resolution 1-Wire Digital Thermometer*. Spojené státy americké, 2008. Dostupné z: <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
- [21] ATMEL. *Katalogový list ATmega16(L): 8-bit Microcontroller with 16K Bytes In-System Programmable Flash*. Spojené státy americké, 2010. Dostupné z: <http://www.atmel.com/Images/doc2466.pdf>.



- [22] MICRORISC S.R.O. *Katalogový list IQRF TR-55D: Transceiver Module*. Jičín, 2013. Dostupné z: <http://iqrf.org/weben/downloads.php?id=239>.
- [23] MICRORISC S.R.O. *Katalogový list IQRF TR-52D: Transceiver Module*. Jičín, 2013. Dostupné z: <http://iqrf.org/weben/downloads.php?id=213>.



## 9. SEZNAM OBRÁZKŮ A TABULEK

Obr. 2.1, Blokové schéma senzoru .....	7
Obr. 2.2, Blokové schéma ústředny .....	8
Obr. 2.3, Blokové schéma zvolené metody napájení .....	15
Obr. 3.1, Blokové schéma integrovaného obvodu <i>APDS-9301</i> .....	16
Obr. 3.2, Blokové schéma integrovaného obvodu <i>DS18B20</i> .....	17
Obr. 3.3, Příklad interpretace výstupních dat snímače <i>DS18B20</i> .....	18
Obr. 3.4, Typická kabeláž <i>DALI</i> sběrnice .....	19
Obr. 3.5, Příklad připojení zařízení na <i>DALI</i> sběrnici .....	19
Obr. 3.6, Příklad logických úrovní na <i>DALI</i> sběrnici .....	20
Obr. 3.7, Princip bi-fázového kódování .....	20
Obr. 3.8, Časový průběh 2bytové zprávy na <i>DALI</i> sběrnici.....	21
Obr. 3.9, <i>Master</i> → <i>Slave Forward</i> rámec .....	22
Obr. 3.10, <i>Master</i> → <i>Master Forward</i> rámec .....	22
Obr. 3.11, <i>Slave</i> → <i>Master Backward</i> rámec .....	22
Obr. 3.12, Doby ustálení mezi <i>Forward</i> rámci na <i>DALI</i> sběrnici .....	23
Obr. 3.13, Obecný tvar adresového bytu <i>DALI</i> zařízení .....	24
Obr. 4.1, Elektrické schéma napájecího modulu .....	28
Obr. 4.2, Elektrické schéma <i>APDS</i> modulu.....	29
Obr. 4.3, Elektrické schéma <i>DALI</i> modulu, první část.....	30
Obr. 4.4, Elektrické schéma <i>DALI</i> modulu, druhá část .....	31
Obr. 4.5, Pokles výkonu FV článku v závislosti na intenzitě dopadajícího záření .....	33
Obr. 4.6, Elektrické schéma napájení senzoru.....	34
Obr. 5.1, Vzorkování signálu na <i>DALI</i> sběrnici .....	38
Obr. 5.2, Vývojový digram přerušení časovače <i>Timer1</i> .....	40
Obr. 5.3, Struktura <i>command</i> registru <i>APDS-9301</i> .....	43
Obr. 5.4, Struktura <i>control</i> registru <i>APDS-9301</i> .....	44
Obr. 5.5, Struktura <i>timing</i> registru <i>APDS-9301</i> .....	44
Obr. 5.6, Struktura <i>interrupt</i> registru <i>APDS-9301</i> .....	45
Obr. 5.7, Příklad čtení 2bytové zprávy z <i>APDS-9301</i> .....	46
Obr. 5.8, Vývojový diagram obsluhy <i>IQRF</i> modulů.....	47
Obr. 5.9, Demonstrace ukázkových kalibračních dat.....	52
Obr. 5.10, Lineární proložení bodů $E_S$ a $E_U$ .....	53



Obr. 5.11, Vývojový digram regulátoru .....	54
Obr. 5.12, Příklad vývoje intenzity osvětlení v místě měření .....	55
Obr. 6.1, Pokus o navázání spojení .....	58
Obr. 6.2, Úspěšně navázané spojení .....	58
Obr. 6.3, Chyba při pokusu o navázání spojení.....	58
Obr. 6.4, Základní obrazovka .....	59
Obr. 6.5, Rozmístění tlačítek .....	59
Obr. 6.6, Příklad zobrazení položky v základním menu .....	60
Obr. 6.7, Příklad zobrazení položky v menu REZIM.....	61
Obr. 6.8, Příklad nastavení položky v režimu AUTOMATICKY.....	61
Obr. 6.9, Nastavování časových úseků ve scéně .....	61
Obr. 6.10, Nastavení aktuálního času .....	62
Obr. 6.11, Nastavení pracovní doby .....	62
Obr. 6.12, Nastavení světelného výkonu <i>DALI</i> zařízení .....	62
Obr. 6.13, Nastavení hystereze .....	63
Obr. 6.14, Příklad měření intenzity osvětlení .....	63
Obr. 6.15, Kalibrace snímače intenzity osvětlení .....	64
Obr. 6.16, Ilustrace příkladu kalibrace snímačů intenzity osvětlení .....	64
Obr. 6.17, Odeslaný a přijatý rámec na <i>DALI</i> .....	66
Tab. 2.1, Porovnání bezdrátových komunikačních prostředků [4].....	11
Tab. 2.2, Hodnocení vybraných parametrů bateriových článků.....	14
Tab. 5.1, Struktura datového typu <i>FRAME</i> .....	42
Tab. 5.2, Použité funkce programové knihovny <i>i2cmaster</i> .....	43
Tab. 5.3, Význam jednotlivých bitů <i>command</i> registru <i>APDS-9301</i> .....	44
Tab. 5.4, Základní bezdrátově přenášený rámec - ústředna .....	48
Tab. 5.5, Speciální bezdrátově přenášený rámec - ústředna.....	49
Tab. 5.6, Struktura odpovědi potvrzující bezdrátové spojení.....	49
Tab. 5.7, Struktura odpovědi navracející dotazovanou veličinu .....	49
Tab. 5.8, Základní funkce knihovny <i>measure</i> .....	50
Tab. 5.9, Příklad záznamu kalibračních dat.....	51
Tab. 5.10, Příklad převodní tabulky mezi $E_s(n)$ a $k(n)$ .....	56
Tab. 6.1, Naměřená data při zkušebním provozu .....	68



## 10. POPIS PŘÍLOH

Přílohy přiložené na konec tohoto dokumentu:

- PŘÍLOHA A. ELEKTRICKÁ SCHÉMATA
- PŘÍLOHA B. DESKY PLOŠNÝCH SPOJŮ  
(*uvedené rozměry DPS jsou v palcích*)
- PŘÍLOHA C. SEZNAM SOUČÁSTEK
- PŘÍLOHA D. VÝVOJOVÉ DIAGRAMY
- PŘÍLOHA E. UŽIVATELSKÉ MENU

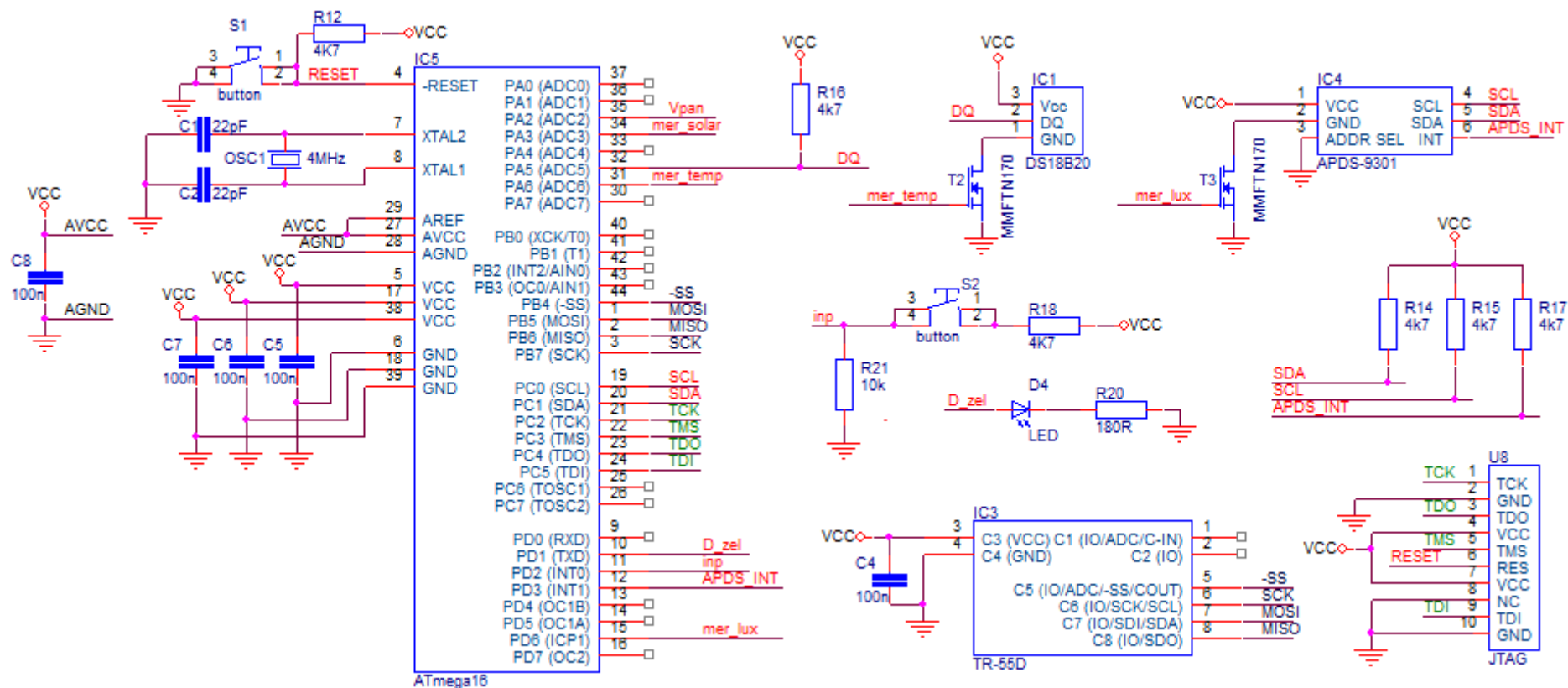
Obsah přiloženého CD je následující:

`\DP_TrnkaI_SvatosJ_2014`

`\Cadence`.....soubory s el.schématy a DPS  
`\Boards`.....desky plošných spojů (\*.brd)  
`\Libraries`.....knihovny součástek (\*.olb)  
`\Masterboards`.....předlohy pro výrobu DPS (\*.brd)  
`\Packages`.....pouzdra součástek (\*.dra)  
`\Pads`.....kontaktní plošky (\*.dra)  
`\Projects`.....elektrická schémata (\*.dsn)  
`\Datashets`.....katalogové listy (\*.pdf)  
`\DP_TrnkaI_SvatosJ_2014.pdf`.....tento dokument  
`\IQRFmodul\IQRFmodul.c`.....sw pro obsluhu IQRF modulů  
`\Senzor`.....sw pro obsluhu senzoru  
`\SENZOR.atsln`.....spustitelný projekt v *Atmel Studiu 6.1*  
`\SENZOR`.....zdrojové soubory senzoru (\*.h, \*.c)  
`\Ustredna`.....sw pro obsluhu ústředny  
`\USTREDNA.atsln`.....spustitelný projekt v *Atmel Studiu 6.1*  
`\USTREDNA`.....zdrojové soubory ústředny (\*.h, \*.c)

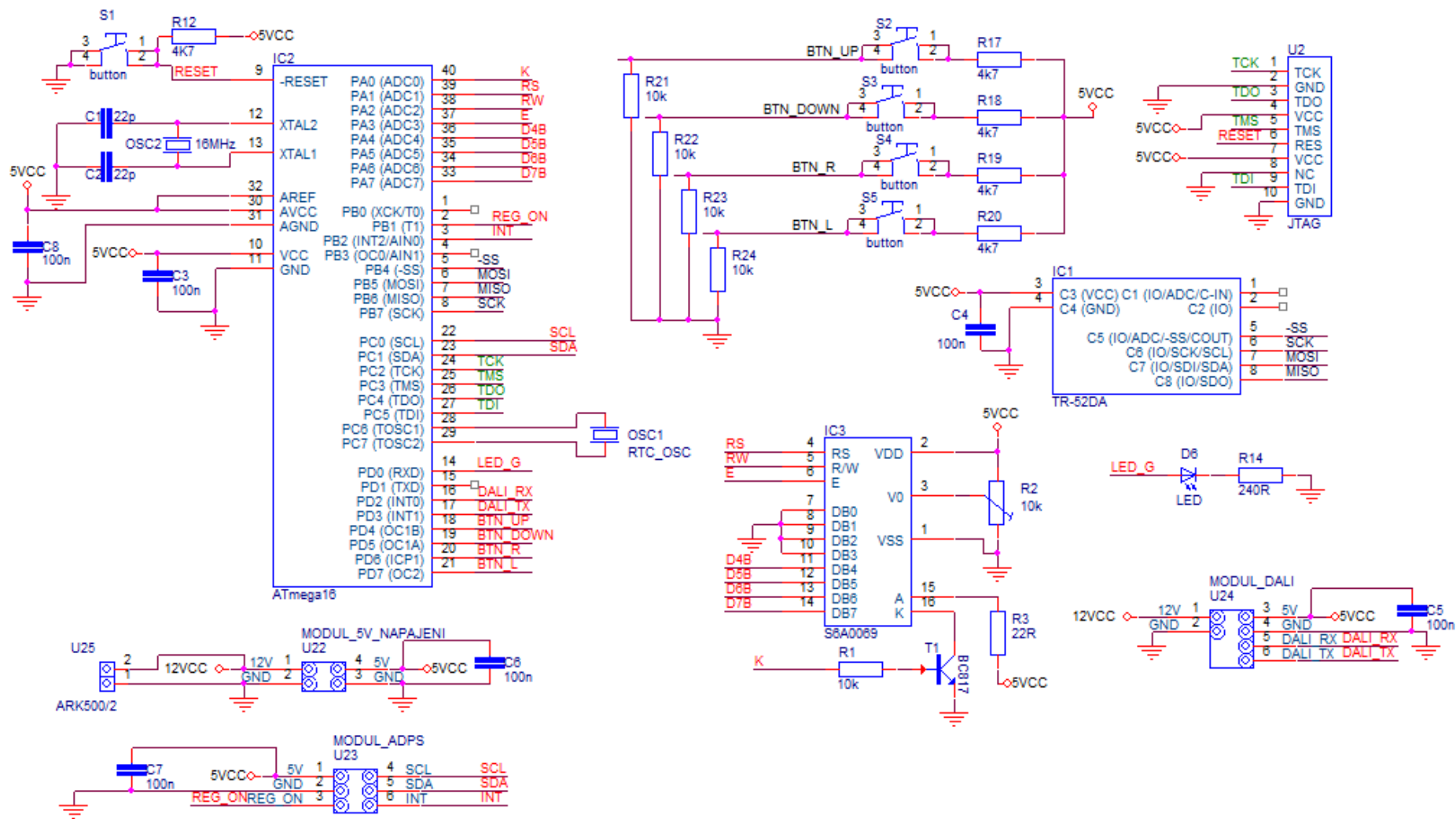
# PŘÍLOHA A. ELEKTRICKÁ SCHÉMATA

## A1. SENZOR





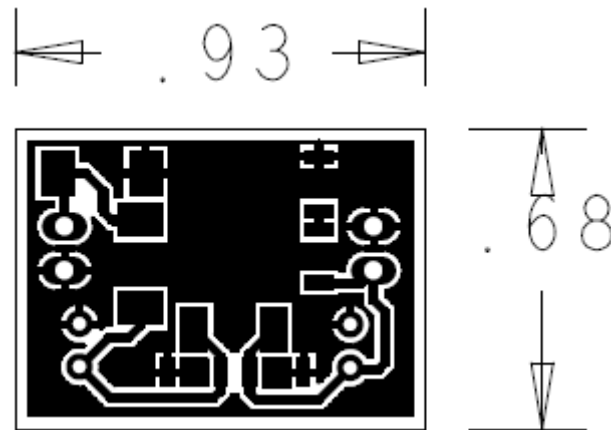
## A2. ÚSTŘEDNA



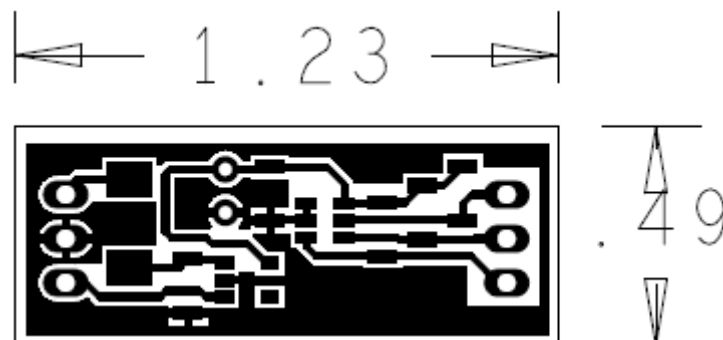


## PŘÍLOHA B. DESKY PLOŠNÝCH SPOJŮ

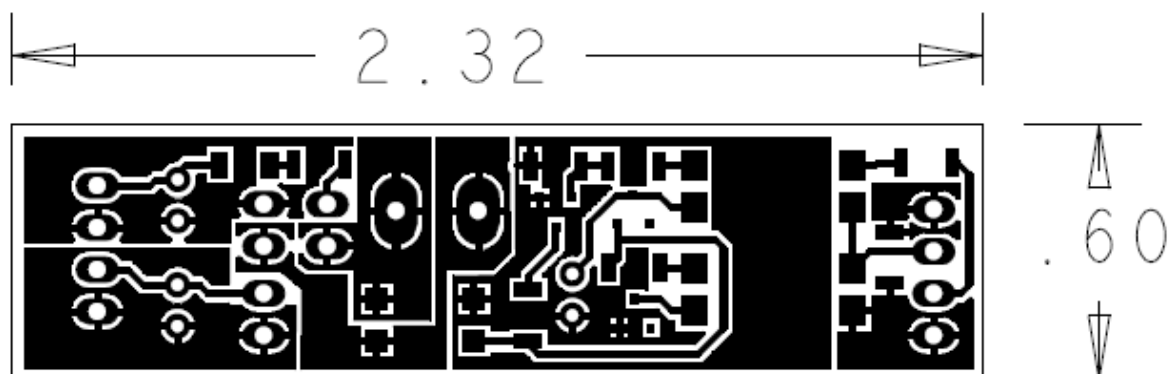
### B1. NAPÁJECÍ MODUL



### B2. APDS MODUL



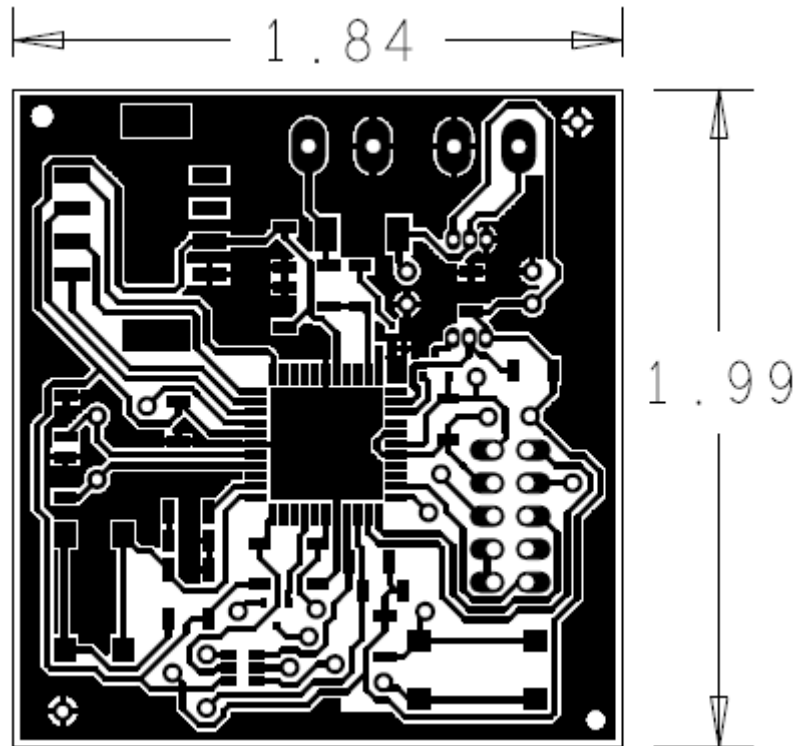
### B3. DALI MODUL



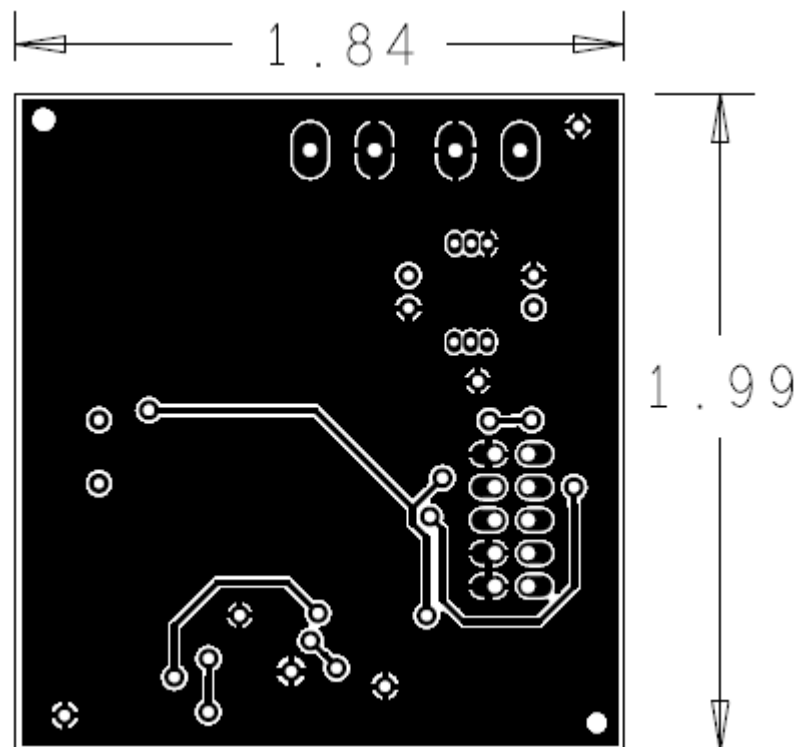


### B4. SENZOR

Pohled na spoje (TOP) :



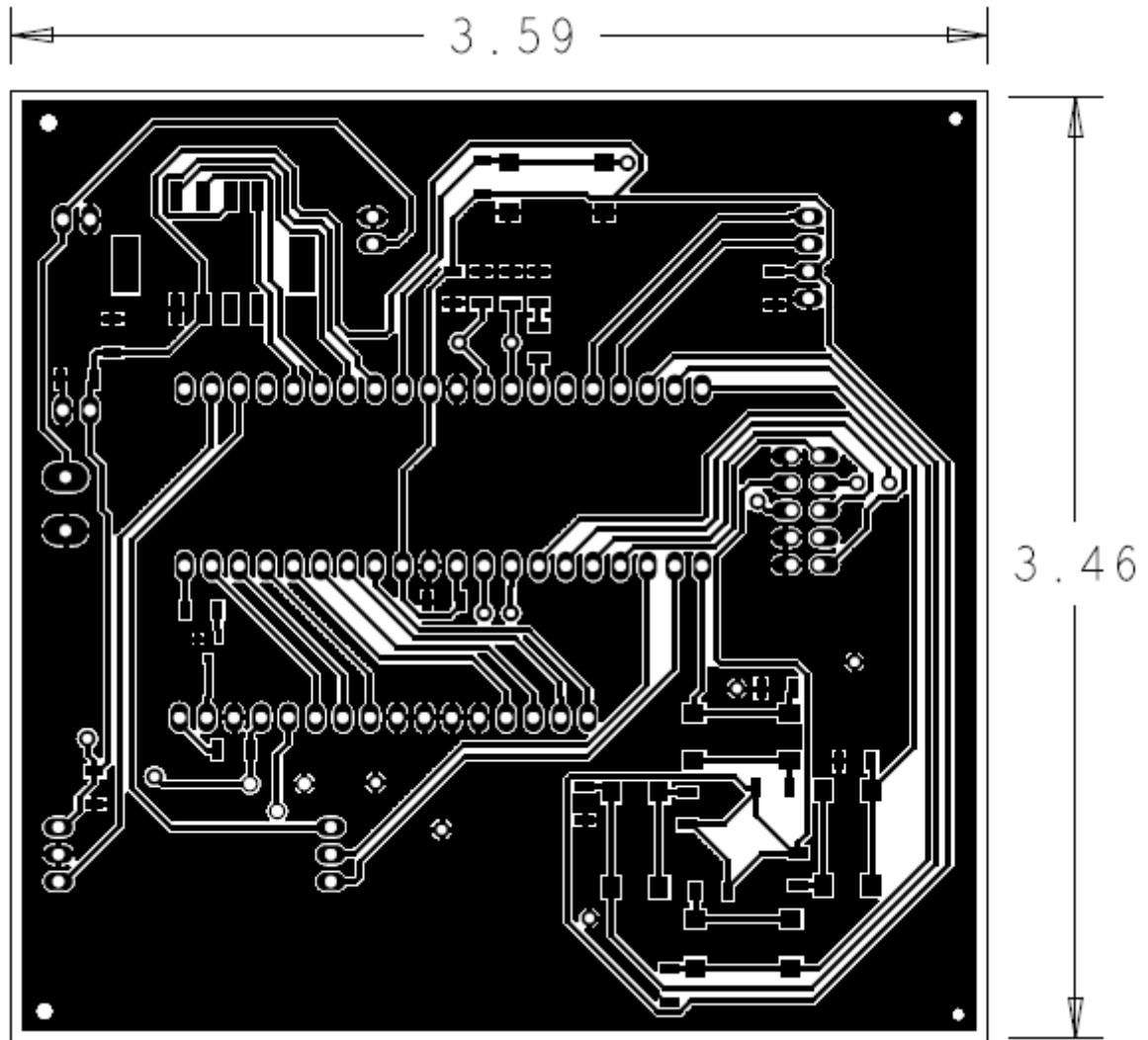
Průhled (BOT) :





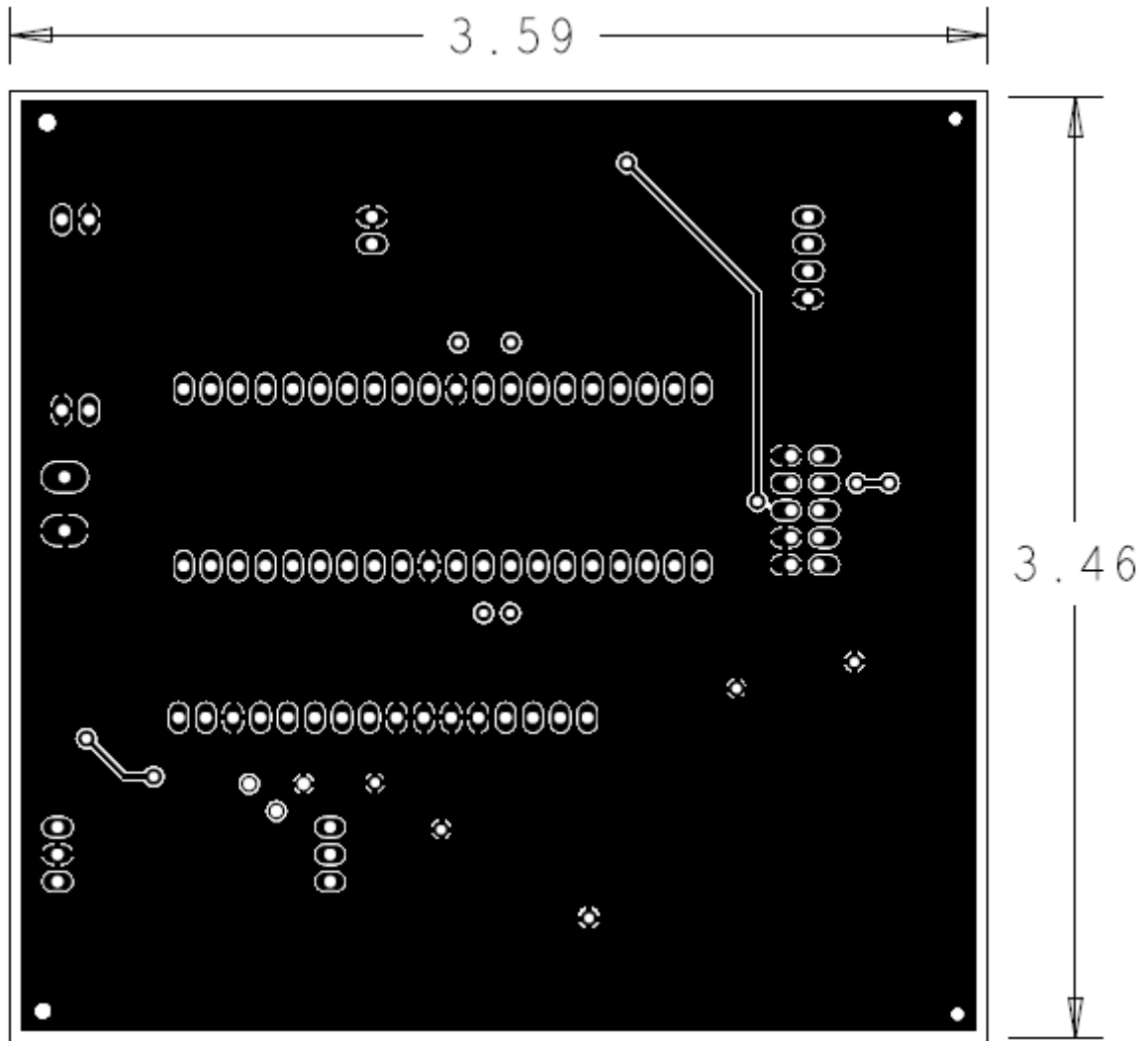
## B5. ÚSTŘEDNA

Pohled na spoje (TOP) :





Průhled (BOT) :



**PŘÍLOHA C. SEZNAM SOUČÁSTEK****C1. NAPÁJECÍ MODUL**

Označení	Hodnota	Pouzdro
R4	240R	SMD 1206
C7, C9	100u / 16 V radiální	E2-5
C6, C8	100n	SMD 1206
D1, D2	1N4007	MELF DO213ab
IC1	LF50	DPAK TO252
D3	LED zelená 20 mA	SMD 1206
JP1, JP2	lámací kolíková lišta – 2 pin	

**C2. APDS MODUL**

Označení	Hodnota	Pouzdro
R18, R19, R20	4k7	SMD 1206
C17	100u / 16 V radiální	E2-5
C15, C16	10n	SMD 1206
D1	1N4007	MELF DO213ab
IC1	LP3990	SOT 23-5
IC2	APDS-9301	
JP1, JP2	lámací kolíková lišta – 3 pin	

**C3. DALI MODUL**

Označení	Hodnota	Pouzdro
R16	220R	SMD 1206
R9	11k	SMD 1206
R8	4R7	SMD 1206
R6	330R	SMD 1206
R7	3k	SMD 1206
R10	20k	SMD 1206
R11	470R	SMD 1206
C12, C13	10u / 25 V radiální	E2-5
C15	20u / 50 V radiální	E2-5



D6	BAT 46	Mini MELF
D4	BZX 84	SOT 23
D5	BAS 16	SOT 23
T2	BC 817	SOT 23
IC1	MYS250	Microdil
IC2, IC3	SFH6156-2T	DIP4
IC5	AM1S-N	SIP4
U14	ARK500/2	
JP1, JP2, JP3	lámací kolíková lišta – 2 pin	
JP4	lámací kolíková lišta – 4 pin	

#### **C4. SENZOR**

<b>Označení</b>	<b>Hodnota</b>	<b>Pouzdro</b>
R12, R14, R15, R17, R18	4k7	SMD 1206
R1, R13, R21	10k	SMD 1206
R20	180R	SMD 1206
C1, C2	22p	SMD 1206
C4, C5, C6, C7, C8, C12	100n	SMD 1206
C10, C11	10u / 16 V radiální	E2-5
D1	1N4007	MELF DO213ab
D4	LED zelená 20 mA	SMD 1206
T1, T2, T3	MMFTN170	SOT 23
IC1	DS18B20	TO 92
IC2	HT7533-1	TO 92
IC3	TR-55D	SIM
IC4	APDS-9301	
IC5	ATmega16L	TQFP
S1, S2	tlačítko P-B1720C	
OSC1	4MHz	HC49US
U8	lámací kolíková lišta – 2x5 pin	
U1	FV článek 4,5V/100mA	
U2	3x NiMH baterie 1,2V/1000mAh	

**C5. ÚSTŘEDNA**

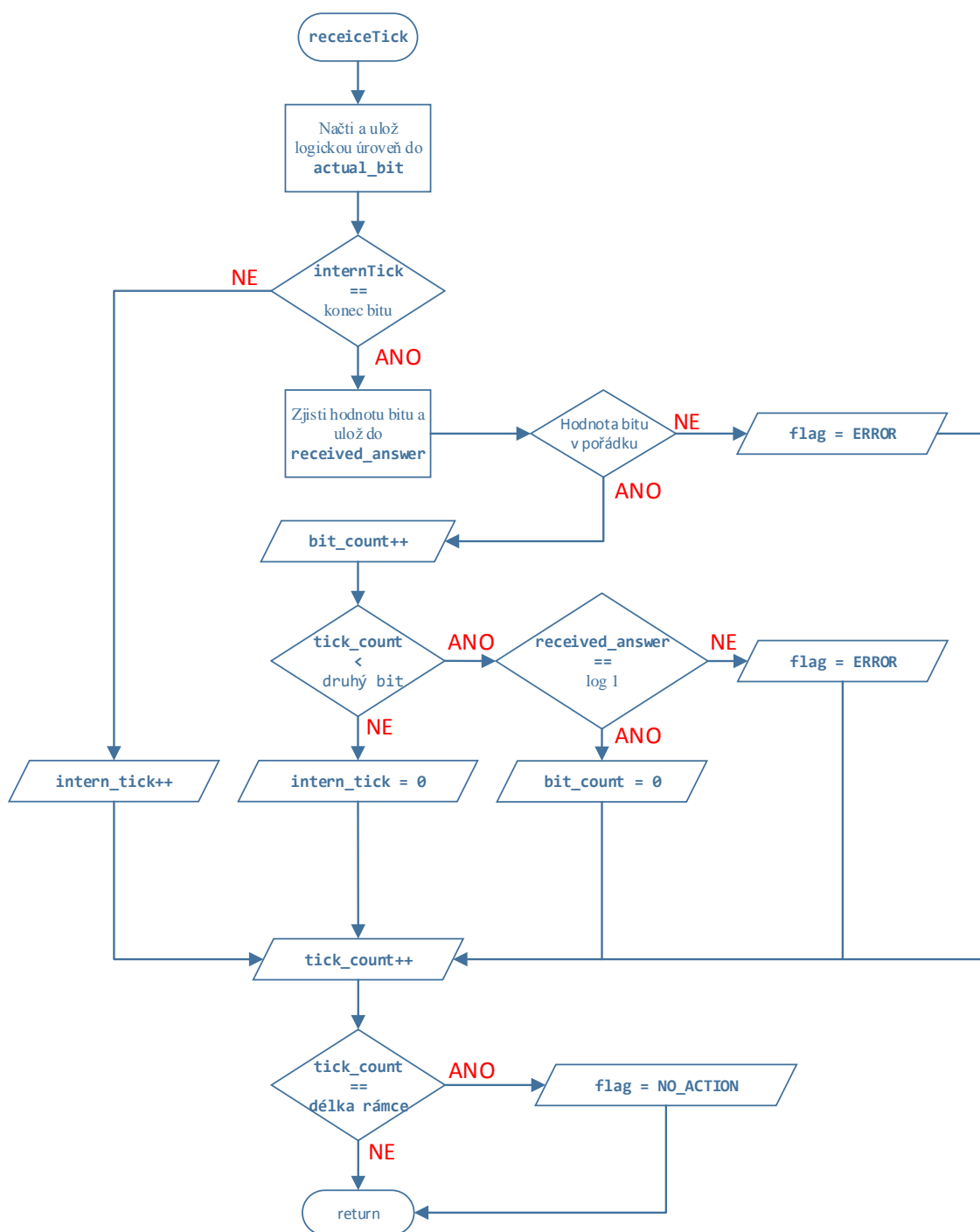
Označení	Hodnota	Pouzdro
R12, R17, R18, R19, R20	4k7	SMD 1206
R14	240R	SMD 1206
R1, R21, R22, R23, R24	10k	SMD 1206
R3	22R	SMD 1206
R2	trimr 10k	64Z
C1, C2	22p	SMD 1206
C3, C4, C5, C6, C7, C8	100n	SMD 1206
D6	LED zelená 20 mA	SMD 1206
T1	BC 817	SOT 23
OSC2	16 MHz	HC49US
OSC1	32,768 kHz	DT26
IC1	TR-52DA	SIM
IC2	ATmega16	PDIP
IC3	ATM1602B	
S1, S2, S3, S4, S5	tlačítko P-B1720C	
U2	lámací kolíková lišta – 2x5 pin	
U25	ARK500/2	

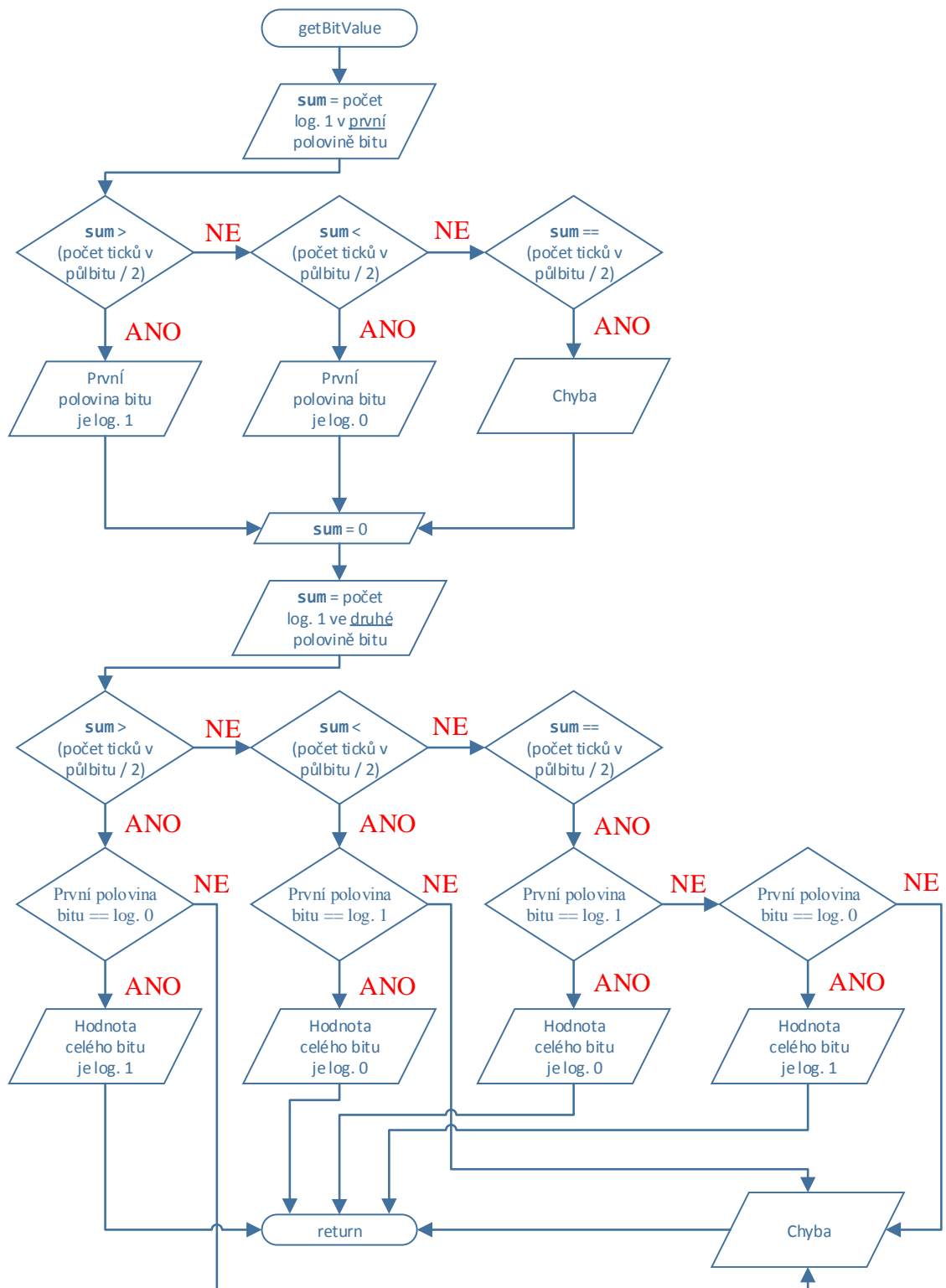




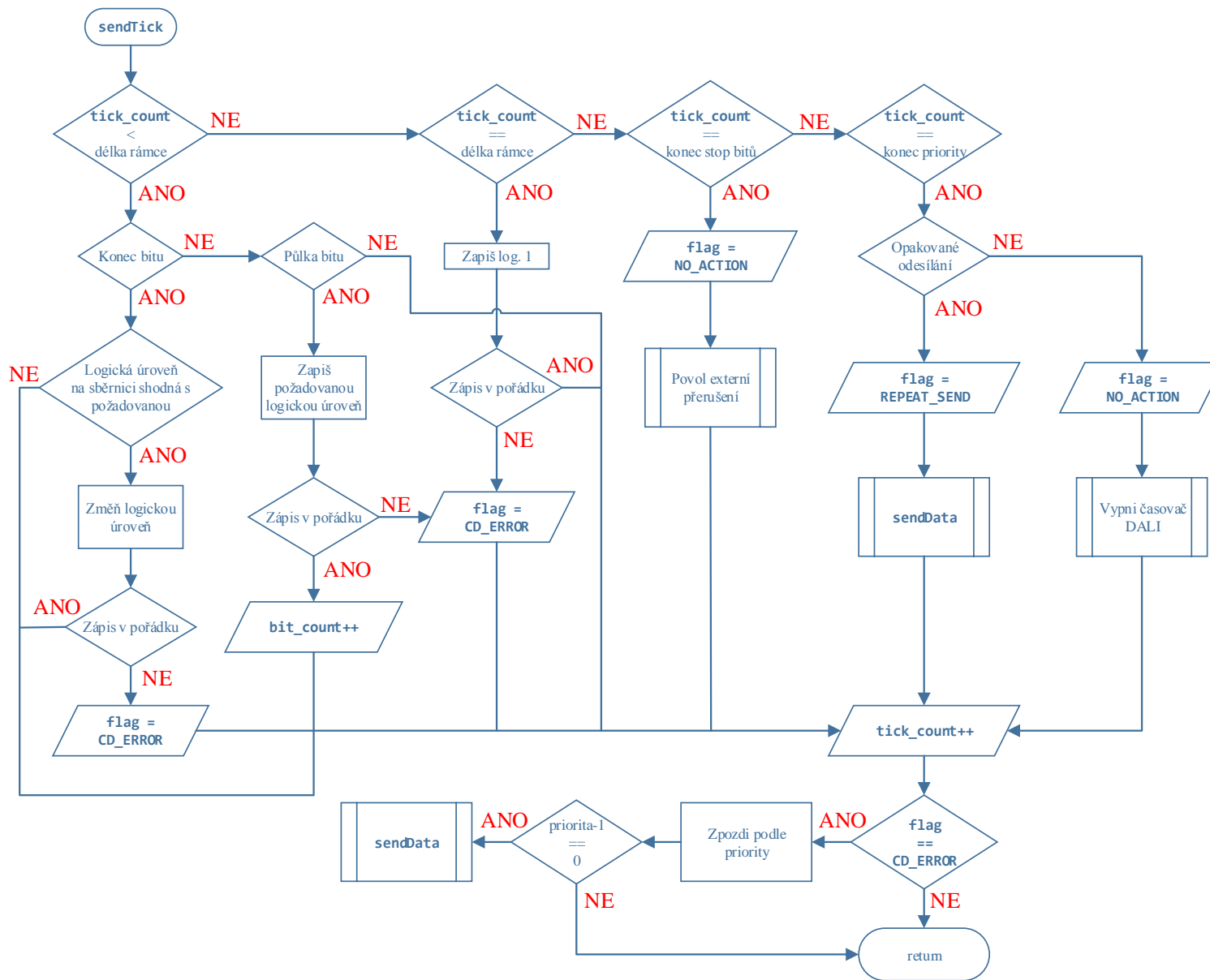
## PŘÍLOHA D. VÝVOJOVÉ DIAGRAMY

## D1. FUNKCE receiveTick()



**D2. FUNKCE `getBitValue()`**

### D3. FUNKCE sendTick()





#### D4. INICIALIZACE A ČTENÍ DAT ZE SNÍMAČE APDS-9301

