



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

PROGRAMOVATELNÝ ČASOVÝ SPÍNAČ KONFIGUROVATELNÝ PŘES USB

PROGRAMMABLE TIMER CONFIGURABLE VIA USB

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jan Dresler

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Ivo Lattenberg, Ph.D.

BRNO 2017

Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

Student: Jan Dresler

ID: 173640

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Programovatelný časový spínač konfigurovatelný přes USB

POKyny PRO VYPRACOVÁNÍ:

Realizujte programovatelný časový spínač. Zařízení bude napájené ze sítě a bude obsahovat obvod reálného času a čidla teploty a intenzity osvětlení (aby bylo možné podmínit sepnutí či rozepnutí nejen časem, ale i teplotou či intenzitou okolního osvětlení). Programovatelné bude prostřednictvím USB portu. Realizujte jej s použitím mikrokontroléru ATMEL. Vytvořte aplikaci pro PC, pomocí které bude možno časový spínač pohodlně konfigurovat.

DOPORUČENÁ LITERATURA:

[1] BRTNÍK, B., MATOUŠEK, D. Mikroprocesorová technika BEN - technická literatura, Praha 2011, 152 stran, ISBN 978-80-7300-406-4.

[2] MATOUŠEK, D. Práce s mikrokontroléry Atmel AVR. Nakladatelství BEN - technická literatura, Praha 2006, 376 stran, ISBN 80-7300-209-4.

Termín zadání: 1.2.2017

Termín odevzdání: 8.6.2017

Vedoucí práce: doc. Ing. Ivo Lattenberg, Ph.D.

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá návrhem desky plošných spojů a naprogramováním mikrokontroleru AVR, který komunikuje s číslicovým teploměrem Dallas, dále s fotorezistorem a obvodem RTC. Pomocí čidel a relé umožňuje spínání různých zařízení. Spínač umožňuje přímé konfigurování pomocí počítače přes USB, k tomu slouží pomocný obvod UART.

KLÍČOVÁ SLOVA

Časový spínač, UART, mikrokontroler, fotorezistor, teploměr, RTC.

ABSTRACT

This thesis describes the design of printed circuit boards and programming AVR microcontroller, that communicates with the digital thermometer Dallas, also with photoresist circuit and RTC. Using sensors and relay allows switching of different devices. The switch allows direct configuration with computer via USB using the auxiliary circuit UART.

KEYWORDS

Time switches, UART, microcontroller, photoresist, thermometer, RTC.

DRESLER.J *Programovatelný časový spínač konfigurovatelný přes USB*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 38 s., 4 s. příloh. Bakalářská práce Vedoucí práce: doc. Ing. Ivo Lattenberg, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svoji Bakalářskou práci na téma Programovatelný časový spínač konfigurovatelný přes USB jsem vypracoval samostatně pod vedením vedoucího Bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené Bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto semestrálního projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Tímto bych chtěl poděkovat vedoucímu Bakalářské práce doc. Ing. Ivu Lattenbergovi, Ph.D. za důležité, věcné a odborné rady, které pomohly k realizaci této práce. Dále bych rád poděkoval svým rodičům, kteří mi svou podporou umožnili studovat na této škole.

V Brně dne.....

.....

(podpis autora)

OBSAH

Úvod	1
1 Návrh desky	2
1.1 Blokové schéma	2
1.2 Popis funkce zařízení	3
2 Srovnání a výběr součástek	4
2.1 Fotorezistory	4
2.1.1 Typ VT93N2	4
2.1.2 Typ VT43N1	4
2.1.3 Typ A906012	4
2.1.4 Výběr typu fotorezistoru	5
2.2 Teploměry	5
2.2.1 Typ DS18S20+	5
2.2.2 Typ DS18B20	6
2.2.3 Typ LM35DZ	6
2.2.4 Výběr typu teploměru	7
2.3 Výběr typu pro obvod reálného času	7
2.3.1 Typ DS1302	7
2.3.1.1 Schéma zapojení	7
2.3.2 Typ PCF8593P	8
2.3.2.1 Vlastnosti pinů pro zapojení	8
2.3.3 Typ DS1306+	9
2.3.3.1 Schéma zapojení	9
2.3.4 Výběr pro obvod reálného času	10
2.4 Výběr mikrokontroleru	10
2.4.1 Typ mega328P-PU	10
2.4.2 Typ mega8L-8PU	10
2.4.3 Typ mega16A-PU	11
2.4.4 Výběr mikrokontroleru pro spínač	11
3 Fotorezistor	12

3.1	Princip fungování fotorezistoru	12
3.2	Vlastnosti fotorezistoru	12
3.3	Rychlost odezvy fotorezistoru	12
3.4	Fotorezistor VT93N2	13
4	Teploměr	14
4.1	Popis teploměru	14
4.2	Připojení teploměru	14
4.2.1	Připojení pomocí napěťové úrovně	14
4.2.2	Připojení pomocí externího napájení	15
4.2.3	Výběr zapojení a jeho výhody a nevýhody	15
5	RTC real-time clock	16
5.1	Časování	16
5.2	Sériové RTC	16
5.3	Funkce	16
6	USB UART	17
7	Návrh zapojení	18
7.1	Blok napájení	18
7.2	Blok USB	18
7.3	Blok reálného času	19
7.4	Blok zapojení teploměru	19
7.5	Blok fotorezistoru	20
7.6	Blok spínání	20
7.7	Blok mikrokontroleru	21
8	Návrh desky plošných spojů	22
9	Použitý software	23
9.1	Arduino IDE	23
9.1.1	Základní struktura programu	23
9.1.2	Arduino zavaděč	24
9.2	Microsoft Visual Studio	24
10	Vytvořené aplikace	25
10.1	Aplikace pro spínač	25
10.1.1	Definice všech funkcí	25

10.1.2	Ukládání do paměti EEPROM.....	28
10.1.3	Podmínky spínání	29
10.2	Aplikace pro konfiguraci pomocí počítače	29
11	Ovládání programu pro konfiguraci spínače	32
11.1	Návod pro konfigurační program	32
11.1.1	Zóna 1	32
11.1.2	Zóna 2	32
11.1.3	Zóna 3	32
11.2	Nepřipojený stav zařízení	33
12	Testování spínače	34
12.1	Testovací prostředí.....	34
12.2	Výsledky testování.....	35
12.2.1	První testování	35
12.2.2	Druhé testování	35
13	Nová verze spínače	36
13.1	Opravené nedostatky.....	36
13.2	Vylepšení pro provoz.....	36
14	Obal pro spínač	37
14.1	Výroba obalu.....	37
15	Závěr	38
	Literatura	39
	Seznam symbolů, veličin a zkratk	41
A	Návrh zařízení	42
A.1	Deska plošného spoje řídicího modulu – top (strana součástek)	42
A.2	Deska plošného spoje řídicího modulu – bottom (strana spojů).....	43
B	Seznam součástek	44
C	Testovací funkce příjem	46
D	Vývojový diagram	47

SEZNAM OBRÁZKŮ

Obrázek 1: Blokové schéma časového spínače	3
Obrázek 2: Schéma zapojení typu DS1302	8
Obrázek 3: Rozmístění pinů na RTC	9
Obrázek 4: Schéma zapojení RTC typu DS1306+	9
Obrázek 5: Popis fotorezistoru VT93N2 [15].....	13
Obrázek 6: Připojení napájení k teploměru pomocí parazitního napájení [16]	15
Obrázek 7: Zapojení pomocí externího napájení [16]	15
Obrázek 8: Schéma zapojení převodníku USB UART.....	17
Obrázek 9: Návrh zapojení napájecího bloku	18
Obrázek 10: Zapojení USB UART	19
Obrázek 11: Zapojení obvodu pro reálný čas	19
Obrázek 12: Schéma zapojení teploměru DS1820	20
Obrázek 13: Zapojení fotorezistoru	20
Obrázek 14: Schéma zapojení spínacího obvodu	21
Obrázek 15: Zapojení mikrokontroleru	21
Obrázek 16: Pohled na top stranu navržené desky	22
Obrázek 17: Pohled na bottom stranu navržené desky	22
Obrázek 18: Příklad kódu pro blikání LED diody [19]	24
Obrázek 19: Ukázka funkce fotor	25
Obrázek 20: Ukázka funkce sendset	26
Obrázek 21: Ukázka funkce pro zápis času do RTC	27
Obrázek 22: Funkce pro kontrolu času	27
Obrázek 23: Funkce pro převod času na sekundy	28
Obrázek 24: Funkce pro načtení dat pro světlo a teplotu	28
Obrázek 25: Ukázka funkce pro výpis aktuální teploty.....	28
Obrázek 26: Ukázka funkce pro ukládání do paměti EEPROM	29
Obrázek 27: Ukázka funkce odešli	30
Obrázek 28: Ukázka funkce obnov	30
Obrázek 29: Funkce pro nastavení času	31
Obrázek 30: Funkce pro příjem aktuálního nastavení	31
Obrázek 31: Oznamovací okno po nastavení aktuálního času	32

Obrázek 32: Rozdělení aplikace do zón	33
Obrázek 33: Oznamovací okno pro nepřipojený stav	33
Obrázek 34: Ukázka programu při nepřipojeném stavu	33
Obrázek 35: Umístění LED osvětlení pro testování spínače	34
Obrázek 36: Umístění spínače během testování	34
Obrázek 37: Výsledky prvního testování.....	35
Obrázek 38: Výsledky druhého dne testování	35
Obrázek 39: Spínač umístěný v obalu	37

SEZNAM TABULEK

Tabulka 1: Provozní vlastnosti teploměru VT93N2	4
Tabulka 2: Provozní vlastnosti teploměru VT43N1	4
Tabulka 3: Provozní vlastnosti teploměru A906012	5
Tabulka 4: Srovnání všech teploměrů.	5
Tabulka 5: Popis vlastností pinů	6
Tabulka 6: Popis vlastností pinů.	6
Tabulka 7: Popis vlastností pinů.	6
Tabulka 8: Srovnání všech teploměrů.	7
Tabulka 9: Vlastnosti RTC typu DS1302	7
Tabulka 10: Vlastnosti RTC typu PCF8593P	8
Tabulka 11: Umístění pinů	8
Tabulka 12: Vlastnosti RTC typu DS1306+	9
Tabulka 13: Srovnání RTC obvodů	10
Tabulka 14: Vlastnosti mega328P-PU	10
Tabulka 15: Vlastnosti mega8L-8PU	11
Tabulka 16: Vlastnosti mega16A-PU	11
Tabulka 17: Srovnání MCU	11
Tabulka 18: Funkce jednotlivých pinů u teploměru Dallas [4]	14
Tabulka 19: Připojení vodičů mezi UART a mikrokontrolerem.	17

ÚVOD

Moje práce se zabývá návrhem a konstrukcí časového spínače, který je přímo konfigurovatelný z počítače.

Celá práce je rozdělena do několika částí. Kde první část je teoretický úvod, výběr součástek a návržení desky. Druhou částí je pak již samotná konstrukce. Poslední část je program, který řídí činnost spínače a podpůrný program na počítači, kterým je možné konfigurovat spínač.

V první části při návrhu je důležité zpracovat i obvod měření teploty a okolního světla. Proto bude moci spínač používat více proměnných a podle toho regulovat sepnutí. Přímou konfigurací umožňuje obvod USB (Universal Serial Bus) UART (Universal asynchronous receiver/transmitter), který umožňuje připojení spínače k počítači.

V další části byla pro spínač navržena deska plošných spojů tak, aby měla potřebné součástky a zároveň měla přijatelné rozměry. Pro přesné rozměry desky byla vyrobena krabička, která je z plexiskla a chrání spínač před vnějším okolím. Byla navržena tak, aby splňovala všechny požadavky např. mřížka pro chlazení, prostor pro svorkovnici.

Poslední část se věnuje programování hlavního programu pro funkci spínače. Program je psaný jazykem C a byl napsán tak, aby sbíral data od všech čidel a potom je porovnával v podmínkách, podle kterých potom spíná. Podpůrný program pro počítač slouží pro přímou konfiguraci spínače přes počítač, a tak nám umožňuje snadněji konfigurovat proměnné pro spínání. Podpůrný program je napsaný v jazyce C#.

1 NÁVRH DESKY

Samotnou desku je vhodné navrhovat v nějakém programu, který je přizpůsoben pro navrhování desky plošných spojů (dále jen DPS). Program nám ušetří práci s rýsováním a výpočty tloušťky cest. Před návrhem desky je potřeba mít navržené schéma výrobku. Schéma lze taktéž navrhnout v programu, který se nám snaží práci co nejvíce ulehčit, a to díky knihovnám a různými podpůrnými doplňky. V programu je vždy mnoho součástek, respektive knihoven, které představují jednotlivé komponenty desky. Pro návrh schématu je důležité vědět, co chceme konstruovat a jak by daná část měla fungovat. V návrhu je dobré si celé schéma rozdělit do několika funkčních celků, jako např. napájení, komunikace, senzory. Když si schéma rozdělíme do jednotlivých celků, bude pozdější orientace ve schématu jednodušší a případné změny nebo opravy budou snadněji proveditelné. Před rozdělením do celků je dobré si navrhnout blokové schéma, které popisuje funkci a zapojení práce. Práci pak rozdělíme na celky podle blokového schématu, a tak můžeme snáze popsat jejich funkci, nebo je snadněji, jak už bylo řečeno, opravit nebo pozměnit.

1.1 Blokové schéma

Níže můžeme vidět návrh blokového schématu. Schéma je možné navrhnout více způsoby podle toho, jakou funkci od výrobku vyžadujeme. Jak již bylo řečeno níže uvedené blokové schéma je navržené přímo pro potřeby výrobku. Funkce jednotlivých celků si nyní popíšeme.

MCU: Z anglického microcontroller unit. Je hlavní stavební jednotkou a plní tu funkci mozku.

RTC: Z anglického Real-time clock = počítačové hodiny, které zajišťují, aby MCU měl vědomí o aktuálním čase a mohla v případě potřeby řídit aplikace podle reálného času.

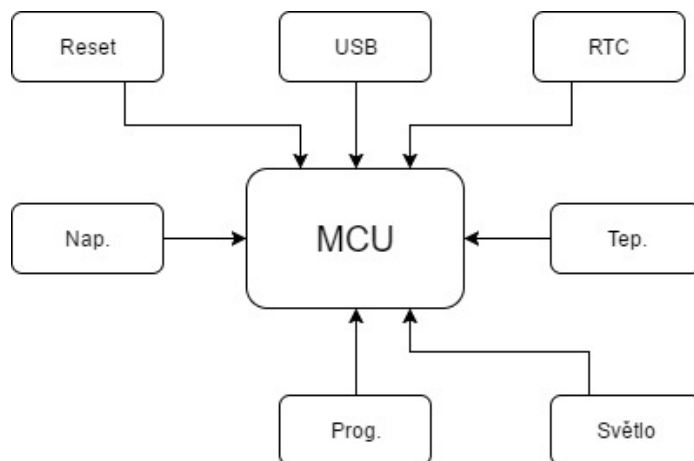
Tep.: Je zkratka pro teploměr. Teploměr plní funkci, jak je již podle názvu je patrné, měří teploty.

Světlo: Tato část blokového schématu plní funkci čidla na měření úrovně světla, díky které můžeme zjistit, jestli je noc, nebo den.

Prog.: Je zkráceně program. Plní funkci portu, který nám umožňuje nahrát základní či testovací program do paměti MCU.

Nap.: Zkráceně napájení, je část celku, která zajišťuje správné napětí pro MCU.

Reset: Slouží nám k restartování MCU.



Obrázek 1: Blokové schéma časového spínače

1.2 Popis funkce zařízení

Zařízení funguje jako časový spínač, který spíná pomocí relé. Spínač bude spínat buďto podle času, nebo je schopen vzít v potaz další proměnné, a to světlo a teplotu. Ve spínači bude nainstalován program, který bude schopen vzít v potaz všechny proměnné a různě je kombinovat. Dále by měl spínač umožňovat ovládání pomocí USB. Pomocí počítače budeme moci upravit dobu spínání podle toho, co chceme ovládat. Samotný spínač je osazený svorkovnicí, která slouží pro připojení spínacího kontaktu libovolného zařízení, které umožňuje sepnutí. Spínač má dva vývody, a proto dokáže spínat dvě zařízení zároveň. Tudiž nebude potřeba vyrábět rozdvojku nebo podobné zařízení, abychom mohli připojit dvě zařízení. Spínač je napájen pomocí externího zdroje o parametrech 12 V/1 A, který je připojen pomocí power jacku do spínače. Díky tomu je v budoucnu umožněno napájet zařízení pomocí baterie. Dále spínač signalizuje připojené funkční napájení pomocí LED (light-emitting diode) diody červené barvy. Ovládání z počítače je umožněno pomocí USB UART, které je popsáno viz Kapitola USB UART.

2 SROVNÁNÍ A VÝBĚR SOUČÁSTEK

V následující kapitole se budu zabývat srovnáním součástek na základě jejich vlastností a srovnáním mezi sebou, kde zdůvodním a popíšu výběr jedné z nich. Zohledním jejich vlastnosti, dostupnost a cenu.

2.1 Fotorezistory

V práci jsem použil fotorezistor, díky čemuž zohledním další fyzikální faktor, a to množství světla. Díky tomu bude moci spínač např. ovládat světla, protože je zbytečné svítit ve dne. Všechny následující typy lze použít a případně vyměnit dle vlastní volby.

2.1.1 Typ VT93N2

Tento typ nabízí celkem pět možností, jak ho zapojit do obvodu, a to měření okolního světla, DC (stejnoseměrné napětí) relay, AC (střídavé napětí) relay, snímání objektů a mostní obvod. Důležité vlastnosti budou zobrazeny v následující tabulce. [1]

Tabulka 1: Provozní vlastnosti teploměru VT93N2

Provozní teplota :	-40 °C až 75 °C
Maximální napětí :	100 V
Ztrátový výkon :	80 mW
Odpor při úplné tmě :	500 kΩ
Cena :	52 Kč

2.1.2 Typ VT43N1

Tento typ je od stejného výrobce jako předchozí, a proto je možné ho zapojit stejně jako typ VT93N2. Jeho vlastnosti si zobrazíme v následující tabulce

Tabulka 2: Provozní vlastnosti teploměru VT43N1 [2]

Provozní teplota :	-40 °C až 75 °C
Maximální napětí :	250 V
Ztrátový výkon :	400 mW při 25 °C
Odpor při úplné tmě :	300 kΩ
Cena :	79 Kč

2.1.3 Typ A906012

Tento teploměr je narušil od předešlých odlišný tím, že má epoxidové zapouzdření. Teploměr je ze série 9060 a konkrétně jsem vybral 906012, jehož vlastnosti jsou zobrazeny v následující tabulce. [3]

Tabulka 3: Provozní vlastnosti teploměru A906012

Provozní teplota :	-20 °C až 70 °C
Maximální napětí :	150 V
Ztrátový výkon :	90 mW
Odpor při úplné tmě :	150 kΩ
Cena :	19,99 Kč

2.1.4 Výběr typu fotorezistoru

Jako spínač byl vybrán fotorezistor typu VT93N2. Jeho výhodou je poměrně velké rozmezí provozních teplot a malý ztrátový výkon. Pro práci není potřeba velkého pracovního napětí, a proto je 100 V vysoce dostačujících. Jeho cena je středem z vybraných typů. Dalším kritériem byla jeho dostupnost a tento typ lze sehnat v mnoha obchodech, jako např. GME nebo TME.

Tabulka 4: Srovnání všech teploměrů

Vlastnosti	VT93N2	VT43N1	A906012
Provozní teplota :	-40 °C až 75 °C	-40 °C až 75 °C	-20 °C až 70 °C
Maximální napětí :	100 V	250 V	150 V
Ztrátový výkon :	80 mW	400 mW při 25 °C	90 mW
Odpor při úplné tmě :	500 kΩ	300 kΩ	150 kΩ
Cena :	52 Kč	79 Kč	19,99 Kč

2.2 Teploměry

Odečítáním hodnot z teploměru bude moci spínač využít další fyzikální veličinu a to teplotu, a proto bude moci spínat také na základě aktuální teploty. Např. je zbytečné použít ostříkovače zahrady, pokud se teplota blíží k nule.

2.2.1 Typ DS18S20+

Digitální teploměr DS18S20+ poskytuje 9-bitové měření teploty, které ukazuje teplotu zařízení. Informace jsou odeslány z teploměru přes 1-Wire rozhraní tak, že pouze jeden drát musí být připojen z centrálního mikroprocesoru k DS18S20+. Napájení pro čtení, zápis a odečítání teploty může být uskutečněno z datového vedení bez nutnosti externího zdroje energie. Každý DS18S20 obsahuje jedinečné číslo křemíku, díky čemuž může být na 1-Wire lince připojeno více teploměrů DS18S20+. To umožňuje umístění snímačů teploty na mnoha různých místech. Aplikace, kdy je tato funkce užitečná, je využívána např. u enviromentální kontroly, snímání teploty uvnitř budov, zařízení nebo stroje a monitorování a řízení technologických procesů. [4]

Tabulka 5: Popis vlastností pinů

Pin	Symbol	Popis
1	GND	Zem.
2	DQ	Datový vstupně/výstupní pin komunikující přes 1-Wire.
3	Vdd	Napájecí pin pro použití externího napájení

Rozmezí měřitelných teplot je $-10\text{ }^{\circ}\text{C}$ až $+85\text{ }^{\circ}\text{C}$ s přesností na $0,5\text{ }^{\circ}\text{C}$ v celém rozsahu.

2.2.2 Typ DS18B20

Digitální teploměr DS18B20 umožňuje 9 až 12-ti bitové měření teploty Celsia a má funkci alarmu. DS18B20 komunikuje přes sběrnici 1-Wire, která ze své podstaty vyžaduje pouze jednu datovou linku pro komunikaci. Stejně jako předchozí typ má každý DS18B20 jedinečný 64-ti bitový sériový kód, který umožňuje vícenásobné zapojení. [5]

Tabulka 6: Popis vlastností pinů

Pin	Symbol	Popis
1	GND	Zem.
2	DQ	Datový vstupně/výstupní pin komunikující přes 1-Wire.
3	Vdd	Napájecí pin pro použití externího napájení

Jeho rozmezí měřitelných teplot je v rozsahu $-10\text{ }^{\circ}\text{C}$ až $+85\text{ }^{\circ}\text{C}$ s přesností $0,5\text{ }^{\circ}\text{C}$ v celém rozsahu.

2.2.3 Typ LM35DZ

Typy LM35 jsou přesné teploměry integrovaných obvodů, jejichž výstupní napětí je přímo úměrná teplotě Celsia. LM35 má výhodu oproti čidlům lineárních teploměrů, které se musí kalibrovat. LM35 nevyžaduje žádné externí kalibrování nebo ořezávání přesnosti. LM35 má nízkou výstupní impedanci, lineární výstup a přesnou vlastní kalibraci. [6]

Tabulka 7: Popis vlastností pinů

Pin	Symbol	Popis
1	+Vs	Napájení v rozmezí 4 V až 20 V
2	Output	Datový vstupně/výstupní pin komunikující přes 1-Wire.
3	GND	Zem.

Rozmezí měřitelných teplot je v rozsahu $-40\text{ }^{\circ}\text{C}$ až $110\text{ }^{\circ}\text{C}$ s přesností $\pm 10\text{ }^{\circ}\text{C}$.

2.2.4 Výběr typu teploměru

Jako spínač byl vybrán typ DS18B20. Jeho měřitelné teploty jsou více než dostačující a s výbornou přesností. Jeho cena přiměřená a je dostupný téměř ve všech prodejnách s elektro zbožím.

Tabulka 8: Srovnání všech teploměrů

Typ	Měřitelné teploty	Přesnost	Cena
DS18S20+	-10 °C až +85 °C	0,5 °C	101 Kč
DS18B20	-10 °C až +85 °C	0,5 °C	78 Kč
LM35DZ	-40 °C až 110 °C	10 °C	65 Kč

2.3 Výběr typu pro obvod reálného času

Díky obvodu pro reálný čas (RTC) bude mít mikrokontroler neustále informaci o tom kolik je hodin. Když mikrokontroler ví tuto informaci, dokáže ji patřičně využít. Pokud budeme chtít použít spínač vždy v přesně daný světový čas, je to díky RTC možné. Mikrokontroler se dotazuje, kolik je hodin, a pokud naskočí čas, který jsme zvolili pro sepnutí, tak spínač sepne.

2.3.1 Typ DS1302

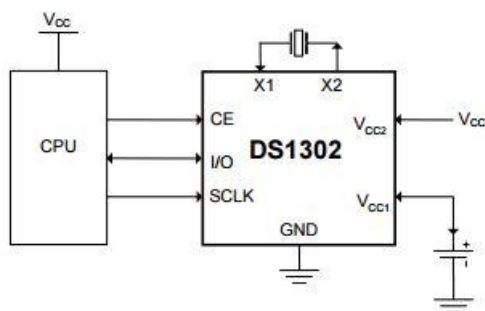
S mikrokontrolerem komunikuje ten to typ pomocí jednoduchého sériového rozhraní. RTC poskytuje sekundy, minuty, hodiny, dny, měsíc a rok. Konečný den pro měsíc s méně než 31 dny se automaticky upraví, včetně oprav u přestupného roku. Hodiny pracují buď ve 24-ti hodinovém nebo 12-ti hodinovém formátu s indikátorem AM/PM. [7]

Tabulka 9: Vlastnosti RTC typu DS1302

RAM	31 x 8 bajtů
Napětí pro napájení	2,0 V až 5,5 V
Komunikace	3-Wire

2.3.1.1 Schéma zapojení

Obvod RTC je další integrovaný obvod a je patrné, že bude potřebovat pomocné součástky mimo obal integrovaného obvodu. Proto je na následujícím obrázku vidět schéma zapojení typu DS1302.



Obrázek 2: Schéma zapojení typu DS1302

2.3.2 Typ PCF8593P

Obvod RTC typu PCF8593P používá technologii CMOS (Complementary Metal–Oxide–Semiconductor) obvodu pro hodiny a kalendář. Díky tomu má nízkou spotřebu energie. Adresy a údaje jsou přenášeny sériově přes dvouřádkový obousměrný I2C-bus. Vestavěný adresový registr je zvýšen automaticky po každém čtení nebo zapisování dat. Vestavěný 32,768 kHz oscilátor a prvních 8 bajtů paměti RAM (Random-access memory) je použito pro hodiny, kalendář a počítadlo funkce. [8]

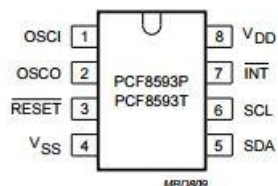
Tabulka 10: Vlastnosti RTC typu PCF8593P

RAM	8 bajtů
Napětí pro napájení	2,5 V až 6,0 V
Komunikace	I2C-bus

2.3.2.1 Vlastnosti pinů pro zapojení

Tabulka 11: Umístění pinů

Symbol	Pin	Popis
OSCI	1	Vstup oscilátoru
OSCO	2	Výstup oscilátoru
RESET	3	Vstup resetu
Vss	4	Záporné napájení
SDA	5	Sériový vstupně-výstupní pin
SCL	6	Sériový vstup hodin
INT	7	
Vdd	8	Kladné napájení



Obrázek 3: Rozmístění pinů na RTC

2.3.3 Typ DS1306+

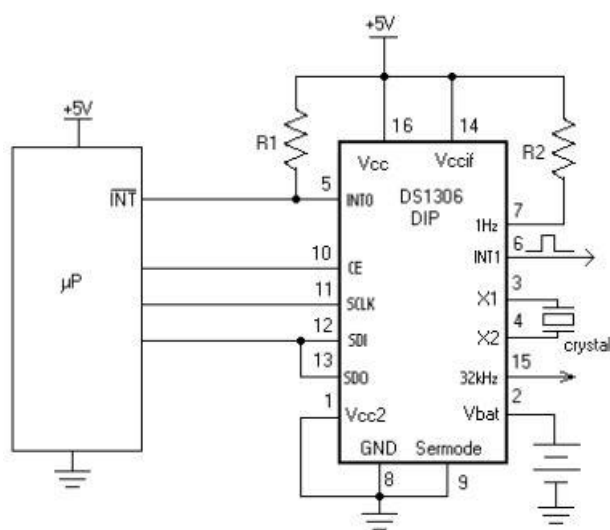
Obvod reálného času poskytuje plně binární překlad do decimálního času. Hodiny a kalendář jsou přístupné pomocí jednoduchého sériového rozhraní. Poslední den měsíce se automaticky upraví pro měsíce s méně než 31 dny, včetně oprav u přestupného roku. Hodiny působí buď ve 24-ti hodinovém nebo 12-ti hodinovém formátu s ukazatelem AM/PM. Kromě toho 96 bajtů NV RAM je k dispozici pro datový úložný prostor. [9]

Tabulka 12: Vlastnosti RTC typu DS1306+

RAM	96 bajtů
Napětí pro napájení	2,0 V až 5,0 V
Komunikace	3-wire

2.3.3.1 Schéma zapojení

Obvod RTC typu DS1306+ pro správnou funkci potřebuje externí součástky, které zajišťují jeho správnou funkci. Proto si na následujícím obrázku ukážeme schéma zapojení.



Obrázek 4: Schéma zapojení RTC typu DS1306+

2.3.4 Výběr pro obvod reálného času

V tabulce níže můžeme vidět srovnání všech zmíněných RTC obvodů. Pro spínač jsem vybral obvod DS1302. Jeho vlastnosti jsou dostačující pro potřebu spínače a komunikace je prováděna po sériové lince. Jeho cena je přiměřená výkonu.

Tabulka 13: Srovnání RTC obvodů

Typ	Paměť	Komunikace	Cena
DS1302	31 x 8 bajtů	3-wire	103 Kč
PCF8593P	8 bajtů	I2C-bus	83 Kč
DS1306+	96 bajtů	3-wire	146 Kč

2.4 Výběr mikrokontroleru

Výběr správného mikrokontroleru je velmi důležitý pro funkci spínače. Při výběru je potřeba dbát na to, aby vybrané MCU mělo dostatečný počet pinů pro ovládání. Dále je také zbytečné kupovat MCU, které má pinů dostatek, ale přitom polovinu z nich nevyužijeme. Takto velké MCU jsou často i drahé a celý projekt by byl dražší jen kvůli nevhodně zvolenému MCU. Při výběru budeme zohledňovat faktory: počet pinů, cena, dostupnost.

2.4.1 Typ mega328P-PU

Vysoce výkonný Atmel kombinuje 32 kb ISP flash paměť s funkcemi pro čtení, zatímco zapisuje data. Disponuje 1024 B EEPROM (Electrically Erasable Programmable Read-Only Memory), 2 KB SRAM (Static Random Access Memory) a třemi flexibilními časovači a čítači s porovnávacími režimy. Vnitřní a vnější přerušování, sériové programování USART, bajtově orientované 2-drátové sériové rozhraní, SPI sériový port, 6-ti kanálový a 10-ti bitový A/D převodník, programovatelný watchdog, časovač s vnitřním oscilátorem a pět softwarových úsporných režimů. Zařízení pracuje mezi 1,8-5,5 voltů. [10]

Tabulka 14: Vlastnosti mega328P-PU

Flash paměť	32 kB
CPU speed	20 MHz
RAM	2 kB
Velikost jádra	8 bit

2.4.2 Typ mega8L-8PU

Mikrokontroler kombinuje 512 K EEPROM a 6 nebo 8 kanálů s 10-ti bitovým převodníkem A/D. Zařízení podporuje propustnost 16 MIPS při 16 MHz a pracuje v rozmezí 2,7 až 5,5 V. Disponuje vysokým výkonem a nízkou spotřebou. Jedná se o 8-bitový mikrokontroler s energeticky nezávislými paměťovými segmenty a interním kalibrovaným RC oscilátorem. [11]

Tabulka 15: Vlastnosti mega8L-8PU

Flash paměť	8 kB
CPU speed	16 MHz
RAM	1 kB
Velikost jádra	8 bit

2.4.3 Typ mega16A-PU

Mega16A-PU je 16-ti bitový CMOS mikrokontroler s nízkou spotřebou. MCU je založený na Atmel AVR a rozšířené architektuře RISC. Přístroj je vybaven 16 kB flash pamětí, 1024 B SRAM, 512 bajtů EEPROM a 8 kanály 10-ti bitového A/D převodníku. MCU má rozhraní JTAG pro on-chip ladění. Přístroj dosahuje 16 MIPS propustnost na 16MHz při 2,7 až 5,5 V. [12]

Tabulka 16: Vlastnosti mega16A-PU

Flash paměť	16 kB
CPU speed	16 MHz
RAM	1 kB
Velikost jádra	8 bit

2.4.4 Výběr mikrokontroleru pro spínač

V tabulce níže můžeme vidět srovnání všech MCU. Pro spínač jsem vybral typ mega328P-PU. Jeho vlastnosti jsou výborné a cena je průměrná. Jeho počet pinů je dostačující a velikost velmi kompaktní, díky čemuž může být deska menší, než kdybychom použili mega16A.

Tabulka 17: Srovnání MCU

Typ	Flash paměť	CPU speed	RAM	Velikost Jádra	Cena
Mega328P-PU	32 kB	20 MHz	2 kB	8 bit	103 Kč
Mega8L-8PU	8 kB	16 MHz	1 kB	8 bit	58 Kč
Mega16A-PU	16 kB	16 MHz	1 kB	8 bit	95 Kč

3 FOTOREZISTOR

3.1 Princip fungování fotorezistoru

Fotorezistor je polovodičová součástka, která využívá energii dopadajícího elektromagnetického záření, včetně denního světla, ke zmenšení svého odporu. Energie světla ve viditelném spektru má energii větší, než je práce nutná k odtržení elektronu z atomu. Po dopadu fotonu světla vzniká pár elektron - díra a tím se zvětšuje vlastní vodivost polovodiče. S nárůstem vodivosti klesá odpor fotorezistoru. Tento jev je vnitřní fotoefekt. [13]

3.2 Vlastnosti fotorezistoru

Vztah odporu R a osvětlení E je nelineární. Při promítnutí grafu v logaritmickém měřítku je to přímka. Charakteristiky reálných provedení fotorezistoru jsou trochu odlišné. Odpor s osvětlením klesá.

Fotorezistory jsou vysoce závislé na teplotě. Při nižší úrovni osvětlení je teplotní závislost větší, teplotní závislost je také větší u vyšších vlnových délek. Teplota způsobuje teplotní šum.

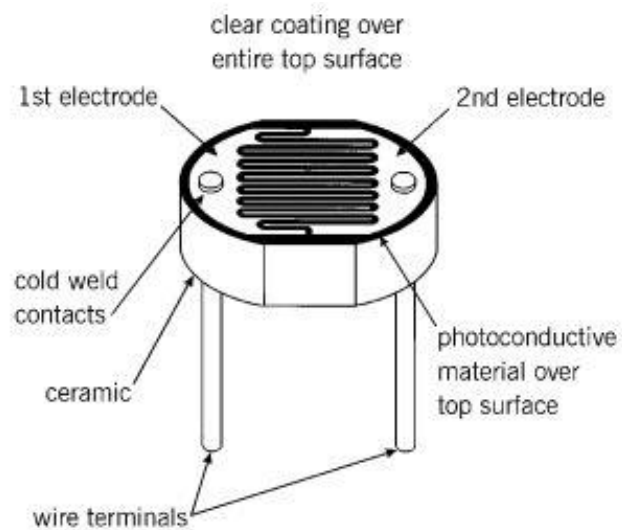
Fotorezistor je nejčastěji vyráběn z materiálu na bázi olova, kadmia, india, a jejich slitin. [14]

3.3 Rychlost odezvy fotorezistoru

Fotorezistory mají velmi pomalou odezvu na změnu osvětlení. Ta se mění s jeho intenzitou. S většími vlnovými délkami a s velikostí osvětlení se odezva fotorezistoru zrychluje. Rychlost odezvy závisí také na použitém materiálu. Nejpomalejší jsou slitiny india a antimonu. [14]

3.4 Fotorezistor VT93N2

Fotorezistor je tenkovrstvé zařízení vytvořené nanesením vrstvou fotovodivého materiálu na keramický podklad. Kovové kontakty jsou roztavené přes povrch fotorezistoru a vnější elektrické připojení je vytvořené těmito kontakty. Tato tenká vrstva fotovodivého materiálu má vysoký odpor. Proto prostor mezi dvěma kontakty je vytvořený na úzko. viz. Obrázek 5.



Obrázek 5: Popis fotorezistoru VT93N2 [15]

4 TEPLOMĚR

4.1 Popis teploměru

Vybraný číslicový teploměr od firmy Dallas je přednastavený pro sběr teploty pomocí 12 bitů. Rozlišení se dá změnit podle příslušného datasheetu.

U teploměru jsou 3 piny: DQ, VCC a GND.

Tabulka 18: Funkce jednotlivých pinů u teploměru Dallas [16]

Pin	Funkce pinu
V_{DD}	Tento pin slouží jako napájecí vývod, na který je možno připojit napětí v rozmezí 3 V – 5,5 V. Lze napájet dvěma způsoby, buď přes port u mikrokontroleru za parazitního napětí, nebo pomocí externího zdroje.
GND	Společný zemnicí potenciál.
DQ	Vstupně-výstupní datový vývod sloužící ke komunikaci s mikrokontrolerem.

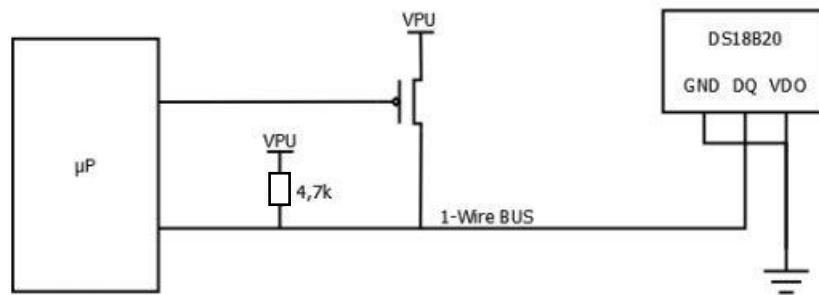
4.2 Připojení teploměru

Připojení číslicového teploměru k mikrokontroleru jde uskutečnit dvěma způsoby, a to s využitím logické úrovně nebo pomocí externího napájení.

4.2.1 Připojení pomocí napěťové úrovně

Napájení lze připojit k teploměru pomocí parazitního napětí přivedeného na datový pin DQ. To je možné realizovat pomocí tranzistoru, u kterého je báze přivedena na pin mikrokontroleru, ten musí napájet datový kabel tehdy, kdy neprobíhá komunikace mezi teploměrem a mikrokontrolerem. Napájení v době, kdy neprobíhá komunikace zároveň nabíjí vnitřní integrovaný kondenzátor, který zajišťuje napájení během datového přenosu. Mezi napájení je připojen pull-up rezistor o hodnotě 4,7 k Ω podle datasheetu. Poté je nutné napájecí pin připojit k zemi. Viz obrázek 3. [16]

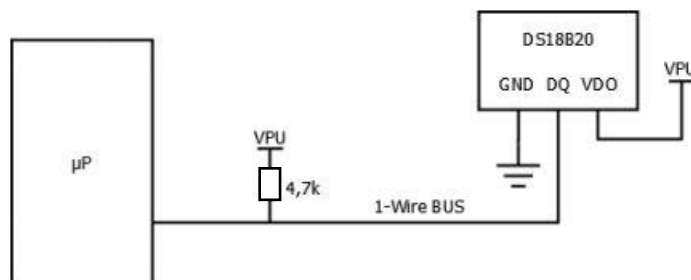
Vzhledem k zapojení je potřeba více součástek a je obsazen další pin na mikrokontroleru.



Obrázek 6: Připojení napájení k teploměru pomocí parazitního napájení [16]

4.2.2 Připojení pomocí externího napájení

V dalším zapojení je použito externí napájení teploměru, které přivedeme na napájecí pin teploměru (viz obr. 2). Dále připojíme pull-up rezistor o hodnotě 4,7 k Ω mezi napájecí napětí podle datasheetu. Toto zapojení je lepší, protože napájecí napětí má velmi podobné napětí jako je napětí mikrokontroleru. Viz obrázek 4. [16]



Obrázek 7: Zapojení pomocí externího napájení [16]

4.2.3 Výběr zapojení a jeho výhody a nevýhody

Pro zapojení použijeme externí zdroj, a proto nebude potřeba tolik součástek. Jeho nevýhodou je, že musíme k teploměru přivést tři vodiče, ale vzhledem k velikosti desky je to zanedbatelná nevýhoda. Tím ušetříme i jeden vstup na mikrokontroleru.

5 RTC REAL-TIME CLOCK

Hodiny reálného času, nebo obvod reálného času, či hodinový obvod jsou hardwarové, nebo méně často softwarové hodiny měřící fyzikální čas. Používají se v počítačích, ale například i ve videorekordérech, domácích spotřebičích apod. Většinou jde o hardware, konkrétně integrovaný obvod. Nejběžnější jsou hodiny reálného času na základní desce. Je to zařízení, které sleduje skutečný čas a udržuje čas i při vypnutí počítače. Obvod je zpravidla napájený samostatnou záložní baterií. Po zapnutí počítače načítá operační systém skutečný – reálný čas z obvodu. Bez tohoto zařízení bychom museli nastavovat reálný čas a datum při každém spuštění počítače. [17]

5.1 Časování

Většina RTC používá krystalový oscilátor, ale některé také používají vedení frekvence. V mnohých případech je na oscilátoru frekvence 32,768 kHz. Jedná se o stejnou frekvenci, jako se používá v krystalových hodinách a hodinkách a ze stejných důvodů má frekvenci přesně 215 cyklů. Proto je ho výhodné použít s jednoduchým čítačem binárních obvodů. [17]

5.2 Sériové RTC

Většina sériových RTC má k sobě začleněno propojení baterie z důvodu ochrany obvodu, ale také mnoho mikroprocesorů obsahuje kontrolní funkce, jako jsou Power-On, reset a Watchdog. Pro aplikace, které nevyžadují zálohu, nebo potřebují jen krátkou dobu zálohy pomocí kondenzátoru, je možné použít dva sériové RTC. [17]

5.3 Funkce

RTC počítá sekundy až roky ve formě BCD na vysílacím čipu obvodu se záložní baterií a s permanentní kontrolou SRAM. Pokud je použito méně než 500 mA na hodiny v zálohovacím režimu, tak nastane RESET mikroprocesoru nezávislým strážným časovačem, který programovatelným časem v období energetického selhání vypíše varování. Také ho vyhodí pro programovatelný budík v režimu záložní baterie a programovatelný přerušení při slabé baterii. [17]

6 USB UART

Dříve se jednalo o převodníky, které obsahovaly konektor CANON pro přímé připojení sériového rozhraní, ale dnešní počítače mají tento konektor zřídka. Proto se dnes používají převodníky s konektorem USB. Tyto převodníky simulují sériové rozhraní, se kterým lze poté pracovat, např. ve Visual Studiu pomocí komponentů SerialPort. Je možné udělat několik variant různých převodníků. S převodníkem se můžeme setkat na desce plošných spojů a nebo přímo s vytvořenou kabelovou redukcí.

Některé převodníky mohou obsahovat přepínač pro změnu napájecího napětí z 5 V na 3,3V a nebo ho lze změnit pomocí přepájení vodiče Vcc.

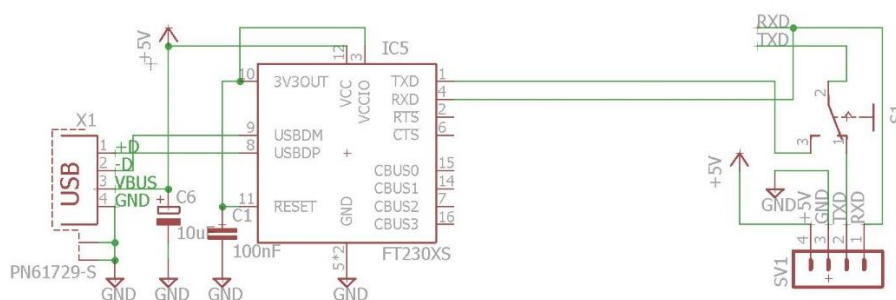
Použití převodníků je jednoduché. Při prvním připojení převodníku do USB v počítači dojde k instalaci ovladače. Při úspěšné instalaci ovladače dostane číslo sériového portu podle toho, jaké mu přidělí sám počítač. [18]

Mezi převodníkem a počítačem jsou čtyři vodiče, které můžeme vidět v tabulce níže.

Tabulka 19: Připojení vodičů mezi UART a mikrokontrolerem

Převodník	Mikrokontroler
Vcc (5 V)	Vcc (5 V)
GND	GND
TXD	RXD
RXD	TXD

Pro spínač jsme použili převodník přímo na desce plošných spojů. Na desce je vypínač, který přepíná vstup TXD mezi převodníkem a jumper lištou. Ta slouží pro budoucí rozšíření, a to tak, že se připojí Bluetooth modul, kterým se bude moci deska také ovládat a bude tak umožněno ovládání z mobilního telefonu. Zapojení tohoto převodníku můžeme vidět na obrázku níže. Na pravé straně můžeme vidět posuvný přepínač, kterým volíme mezi převodníkem a jumper lištou. Na levé straně je vidět samotný převodník.



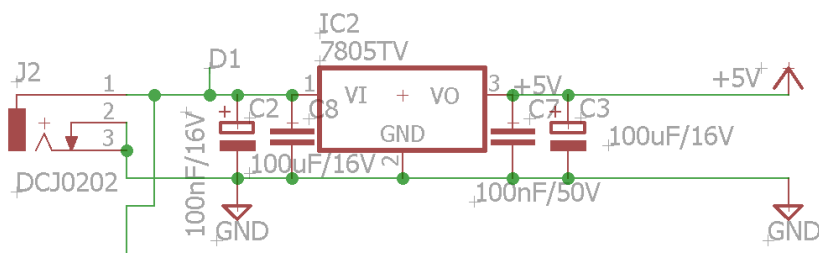
Obrázek 8: Schéma zapojení převodníku USB UART

7 NÁVRH ZAPOJENÍ

Navržené schéma je rozděleno do několika samostatných celků, které jsou společně propojeny přes blok MCU, který byl popsán v podkapitole 1.1. Všechny celky se dají navrhnout více způsoby, přičemž bude popsán ten, který byl použit.

7.1 Blok napájení

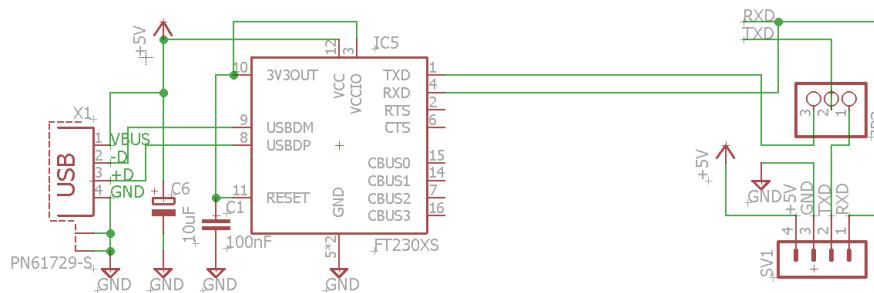
Napájecí blok byl navržen tak, aby bylo možné ho napájet pomocí 12 V, tudíž bude možné použít síťový zdroj, který má výstupní napětí 12 V, nebo může být použita 12 V baterie. Základní funkční jednotkou je usměrňovač napětí 7805TV, jehož hlavní vlastností je, že má pevně dané výstupní napětí 5 V. Před a za tento usměrňovač byly dány kondenzátory tak, že blíže k usměrňovači jsou kondenzátory 100 nF/16 V a dále od usměrňovače 100 μ F/16 V. Pro připojení síťového zdroje byl použit powerjack. Popsané zapojení můžeme vidět na obrázku níže.



Obrázek 9: Návrh zapojení napájecího bloku

7.2 Blok USB

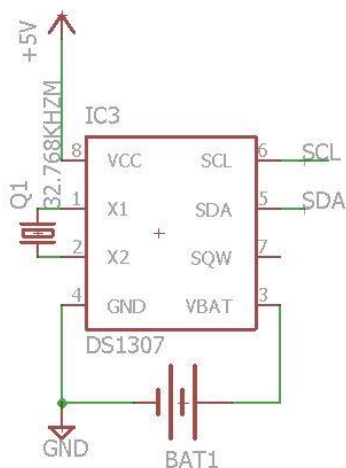
Pro přímou konfiguraci z počítače byl použit převodník UART, který při připojení k počítači vytvoří COM port. Hlavní součástí obvodu UART je mikrokontroler FT230XS, který je napájen 3,3 V, tudíž se napájení 5 V filtruje pomocí dvojice kondenzátorů o hodnotě 100 nF a 10 μ F. Blok komunikuje s hlavním mikrokontrolerem po sériové lince. Přesné zapojení je znázorněné na obrázku níže, přičemž je sériová linka přepínatelná mezi jumper lištou, která slouží pro budoucí připojení Bluetooth modulu a USB UART.



Obrázek 10: Zapojení USB UART

7.3 Blok reálného času

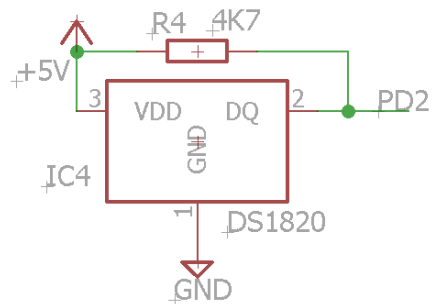
Pro blok reálného času byl použit integrovaný obvod DS1307, který je připojený po lince I2C. Obvod má pro správnou funkci připojené další komponenty, a to krystal o hodnotě 32 768 kHz a baterii, která se stará o to, aby při odpojení časovače od sítě měl RTC obvod stálé napětí pro udržení reálného času. Zapojení je znázorněné na obrázku níže.



Obrázek 11: Zapojení obvodu pro reálný čas

7.4 Blok zapojení teploměru

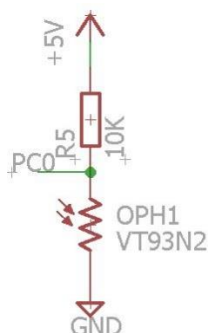
Zapojení vybraného teploměru je jednoduché. Vybraný teploměr je číslicový, a proto posílá přímou hodnotu do mikrokontroleru tak je zapojen podle obrázku níže. Na pin Vdd se přivede napětí 5 V a pin GND se připojí k zemi. Datový pin DQ se připojí k mikrokontroleru a paralelně přes odpor 4K7 k pinu Vdd.



Obrázek 12: Schéma zapojení teploměru DS1820

7.5 Blok fotorezistoru

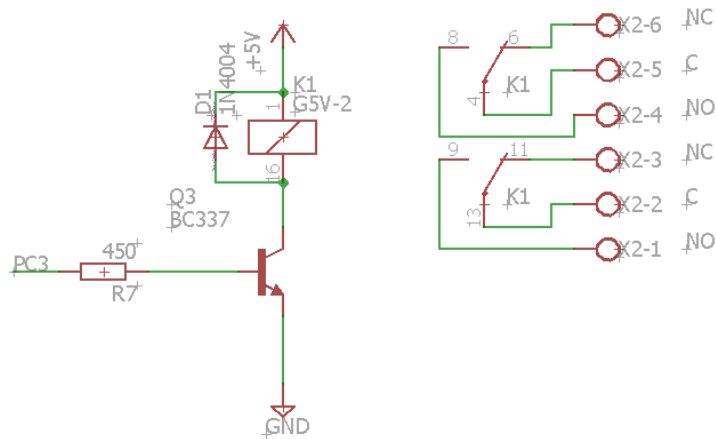
Zapojení fotorezistoru je velice jednoduché. Zapojení fotorezistoru je vlastně odporový dělič, přičemž fotorezistor mění svůj odpor v závislosti na okolním osvětlení. Mezi odpory se připojí pin mikrokontroleru, který odečítá hodnoty napětí a podle toho určí hodnotu okolního světla. Zapojení je na obrázku níže.



Obrázek 13: Zapojení fotorezistoru

7.6 Blok spínání

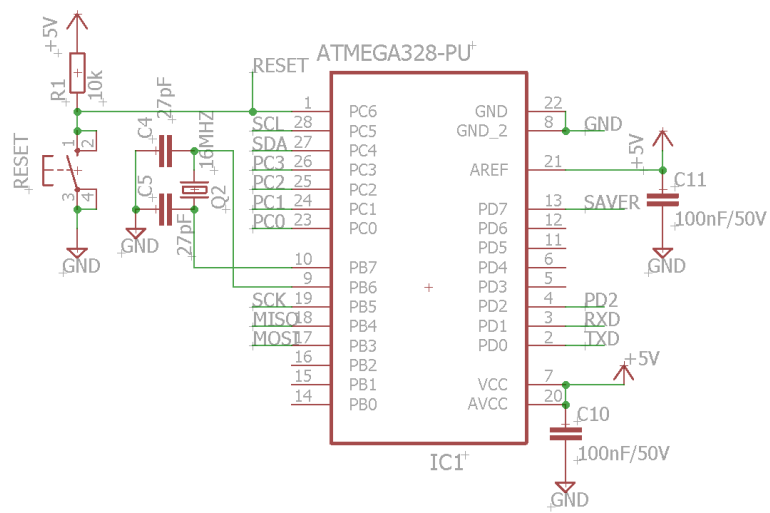
Pro spínání bylo použito relé s cívkou, které je řízeno NPN tranzistorem. Před tranzistor byl umístěn odpor 470 Ω , aby šel na tranzistor vhodný proud. Vybrané relé může spínat i 230 V/2 A. Zapojení je na obrázku níže.



Obrázek 14: Schéma zapojení spínacího obvodu

7.7 Blok mikrokontroleru

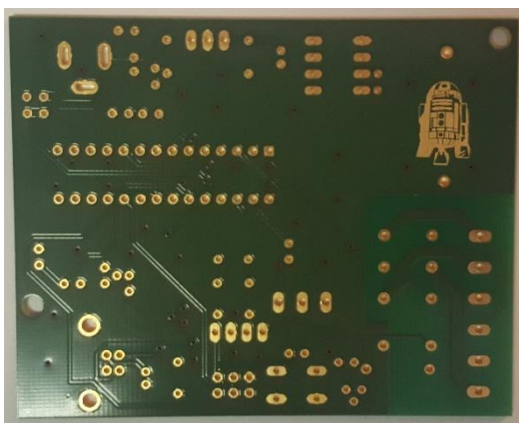
Pro hlavní blok byl použit mikrokontroler mega328, který byl popsán v kapitole Srovnání a výběr součástek. K tomuto bloku jsou připojeny všechny ostatní bloky. K MCU je připojen krystal o hodnotě 16 MHz a dva kondenzátory o hodnotě 27 pF. Dále je k MCU připojený spínač, který plní funkci reset. Zapojení je vidět na obrázku níže.



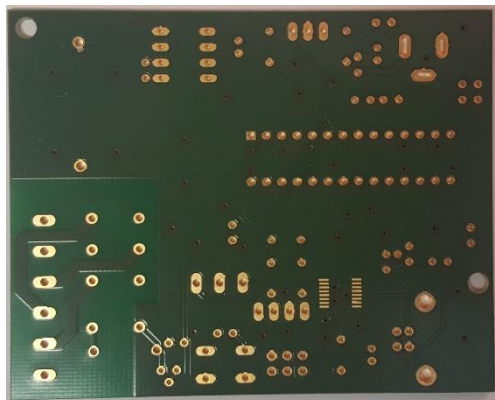
Obrázek 15: Zapojení mikrokontroleru

8 NÁVRH DESKY PLOŠNÝCH SPOJŮ

Deska byla navržena tak, že když máme desku delší stranou k sobě, tak na jedné straně je napájení a USB a na druhé straně je svorkovnice pro připojení prvku, který chceme spínat. Jelikož jsou součástky v THT (Through-Hole Technology), tak jsou osazeny z vrchní strany, až na integrovaný obvod UART. Ten je v provedení SMD (Surface Mount Device) a osazen na spodní straně. Všechny bloky, co byly navrženy a popsány v kapitole 7 Návrh zapojení, jsou na desce osazeny tak, že součástky bloku jsou pohromadě. Navržení si můžeme prohlédnout na obrázcích níže nebo v příloze.



Obrázek 16: Pohled na top stranu navržené desky



Obrázek 17: Pohled na bottom stranu navržené desky

9 POUŽITÝ SOFTWARE

Mezi použitý software pro tuto bakalářskou práci patří:

- Arduino IDE
- Visual Studio

9.1 Arduino IDE

Arduino IDE (Integrated Development Environment) je software určený k programování vývojových desek Arduino. Program se píše do souborů, které se v programu Arduino nazývají Sketch a jsou ukládány do tzv. Sketchbook. I přesto, že je IDE navrženo pro vývojové desky Arduino, mohou se pomocí (tohoto programu) programovat i jiné desky, které obsahují jeden z mikrokontrolerů obsažený na vývojových deskách Arduino. Je možné vypálit bootloader do flash paměti a poté programovat i kompatibilní mikrokontrolery. Tato metoda je vhodná u projektu, kde není žádoucí použití vývojové desky Arduino. Díky tomu lze navrhnout vlastní desku obsahující kompatibilní mikrokontroler s vypáleným zavaděčem. Takový mikrokontroler lze potom programovat úplně stejně jako vývojovou desku Arduino.

Velkou výhodou platformy Arduino je velké množství předpřipravených knihoven, které programátorovi usnadní vývoj programu. Mezi základní prvky Arduino IDE patří tlačítka pro ověření programu, nahrání, založení nového sketchu, otevření souboru, nebo uložení. V nabídce nalezneme i tlačítka pro otevření sériového monitoru. Tlačítka pro ověření programu pomůže programátorovi snadněji vyhledat chyby, které může program obsahovat. V případě, že program najde chyby (v programu), nedovolí programátorovi nahrát program do mikrokontroleru, díky čemuž nedojde k nechtěnému zničení desky. Sériový monitor může sloužit pro výstup, ve kterém lze zobrazovat hodnoty proměnných, výstupy ze senzorů nebo jiná data.

Vývojové prostředí Arduino IDE používá jako primární jazyk Wiring, který sám o sobě není plnohodnotným programovacím jazykem, nýbrž poskytuje dodatečné funkce nad jazykem C++. Jazyk C++ slouží jako základ tohoto frameworku. Použití jazyka Wiring není povinností a lze použít čistou podobu jazyka C++, např. z důvodu zrychlení programu.

9.1.1 Základní struktura programu

Programy psané pro platformu Arduino obsahují dvě základní části:

- Setup-sekce slouží pro nastavení vstupů, výstupů nebo inicializaci proměnných.
- Loop-v této sekci se vykonává samotný program, který je v nekonečné smyčce.

Jednoduchý program pro blikání LED diody může vypadat následovně:

```
4 void setup() {
5   // nastavení pinu 13 jako výstupní
6   pinMode(13, OUTPUT);
7 }
8
9 void loop() {
10  digitalWrite(13, HIGH); // zapne LED
11  delay(1000);           // vyčká 1 sekundu
12  digitalWrite(13, LOW); // vypne LED
13  delay(1000);           // vyčká 1 sekundu
14 }
```

Obrázek 18: Příklad kódu pro blikání LED diody [19]

9.1.2 Arduino zavaděč

Každá deska Arduino obsahuje mikrokontroler od firmy Atmel, ve které je nahrán zavaděč Arduina. Zavaděč je umístěn ve flash paměti a slouží ke komunikaci mezi vývojovou deskou a prostředím Arduino IDE. Po připojení napájení se nejprve spustí zavaděč. [20]

9.2 Microsoft Visual Studio

Microsoft Visual Studio slouží jako vývojové prostředí (IDE). Používá se pro vývoj konzolových aplikací a aplikací s grafickým rozhraním, včetně aplikací Windows Forms a mnohých dalších.

Visual Studio obsahuje editor kódů podporující IntelliSense a refraktování. Obsahuje integrovaný debugger, který pracuje jak na úrovni kódu, tak i na úrovni stroje. Další nástroje slouží pro design formulářů, pro tvorbu aplikací s GUI a mnoho dalšího. Důležitou vlastností je možnost přidávat rozšíření, což vylepšuje funkčnost na téměř každé úrovni.

Jazyky, které Visual Studio podporuje prostřednictvím jazykových služeb, umožňují, aby editor kódů a debugger podporovaly jakýkoliv programovací jazyk. Do vestavěných jazyků patří C/C++, VB.NET a další. Podpora dalších jazyků (např. Python) může být přidána pomocí jazykových služeb, které musí být nainstalované zvlášť. [21]

10 VYTVOŘENÉ APLIKACE

Pro správnou funkci spínače a možnosti ho konfigurovat pomocí počítače bylo zapotřebí vytvořit celkem dvě aplikace:

- Aplikace pro spínač
- Jednoduchá aplikace pro konfiguraci pomocí PC

10.1 Aplikace pro spínač

Aplikace pro spínač slouží jako hlavní prostředek pro ovládání všech komponentů na desce spínače. Slouží pro sběr dat od všech senzorů a porovnává je s daty uloženými v paměti EEPROM. Dále na základě nastavených parametrů, podle kterých má spínat, pošle řídicí impuls na bázi NPN tranzistoru, který způsobí sepnutí relé a tím i sepnutí spínače. Aplikace by se dala rozdělit do několika bloků:

- Definice všech funkcí
- Ukládání do paměti EEPROM (viz. 10.1.2)
- Podmínky spínání

10.1.1 Definice všech funkcí

Definice funkcí slouží pro usnadnění a ovládání všech senzorů. Aplikace obsahuje mnoho funkcí, které mají za účel ovládat senzory či nastavit hodnoty pro spínání a uložit je do paměti.

Funkce fotor

Tato funkce slouží pro fotorezistor, který je jeden ze dvou senzorů na spínači. Fotorezistor může nabývat hodnot od 30 Ω do 1000 Ω v závislosti na intenzitě osvětlení. Toto rozmezí se funkcí „fotor“ rovnoměrně rozdělí celkem na 4 úrovně intenzity, které jsou definované jako tma, šero, jasný den, den. Pokud si vybereme jednu z úrovní a okolní intenzita světla dosáhne námi nastavené hodnoty, dojde k sepnutí.

```
byte fotor()
{
    return map(analogRead(A0), 1000, 30, 0, 3);
}
```

Obrázek 19: Ukázka funkce fotor

Funkce sendset

Funkce „sendset“ slouží pro načtení všech uložených hodnot pro spínání, jako je intenzita světla, teplota nebo čas. Jelikož jsou všechna data pro sepnutí uložena v paměti EEPROM, funkce nejprve načte hodnoty z paměti a potom je vypíše na sériový monitor. Tato funkce je klíčová i pro aplikaci v počítač, jelikož bude možné si vypsat aktuálně nastavené hodnoty pro sepnutí a případně je změnit. Hodnoty jsou uloženy pomocí adres paměti. Použité adresy jsou od 0 do 8 a ke každé adrese se přičte hodnota aktuálního dne, díky čemuž je každý den umožněno spínat v odlišném režimu.

```
void sendset(){
    byte den = Serial.readStringUntil(';').toInt() * 10;

    Serial.print(EEPROM.read(0 + den)); //start time
    Serial.print(";");
    Serial.print(EEPROM.read(1 + den));
    Serial.print(";");
    Serial.print(EEPROM.read(2 + den));
    Serial.print(";");

    Serial.print(EEPROM.read(3 + den)); //Stop time
    Serial.print(";");
    Serial.print(EEPROM.read(4 + den));
    Serial.print(";");
    Serial.print(EEPROM.read(5 + den));
    Serial.print(";");

    Serial.print(EEPROM.read(6 + den)); //light
    Serial.print(";");

    Serial.print(EEPROM.read(7 + den)); //min temp
    Serial.print(";");
    Serial.print(EEPROM.read(8 + den)); //max temp
    Serial.println(";");
}
```

Obrázek 20: Ukázka funkce sendset

Funkce settime

Funkce pro nastavení aktuálního času do paměti RTC se jmenuje „settime“ a slouží pro zapsání data a času do paměti RTC. Toto zapsání je možné provést dvěma způsoby. První z nich je pomocí sériového monitoru, kde se do řádku píše datum a čas ve správném pořadí a jednotlivé údaje se oddělují středníkem. Druhou možností je tlačítko umístěné v aplikaci pro počítač, které nastaví čas podle počítače. Tlačítko provádí obdobnou funkci, jako bychom čas zadávali ručně. Čas a datum převede počítač do řetězce ve správném pořadí, kde jsou jednotlivé údaje odděleny středníkem, a pošle je do paměti spínače.

```

void setTime(){
    int year = Serial.readStringUntil(';').toInt();
    int month = Serial.readStringUntil(';').toInt();
    int day = Serial.readStringUntil(';').toInt();
    int hour = Serial.readStringUntil(';').toInt();
    int min = Serial.readStringUntil(';').toInt();
    int sec = Serial.readStringUntil(';').toInt();
    DS1307.adjust(DateTime(year,month,day,hour,min,sec));
    timekontrol();
}

```

Obrázek 21: Ukázka funkce pro zápis času do RTC

Funkce pro ověření času

Tato funkce slouží pro kontrolu nastaveného času v paměti RTC. Funkce vezme datum a čas z paměti a vypíše ho na sériový monitor nebo po nastavení času pomocí aplikace v počítači na oznamovací okno. Funkce je poměrně důležitá. Pokud si přejeme přesně spínat, je dobré mít přesný čas, jelikož RTC typu DS1307 se měsíčně o několik sekund odchýlí. Proto se doporučuje čas pravidelně kontrolovat a aktualizovat.

```

void timekontrol(){
    DateTime datumCas = DS1307.now();
    Serial.print(datumCas.hour());
    Serial.print(':');
    Serial.print(datumCas.minute());
    Serial.print(':');
    Serial.print(datumCas.second());
    Serial.print(", ");
    Serial.print(datumCas.day());
    Serial.print('.');
    Serial.print(datumCas.month());
    Serial.print('.');
    Serial.println(datumCas.year());
}

```

Obrázek 22: Funkce pro kontrolu času

Převod času na sekundy

Tři funkce „startinsec“, „stopinsec“ a „actualinsec“ slouží pro převod času na sekundy. Tento převod se provádí pro větší přesnost a zjednodušení podmínky pro spínání. Funkce funguje na principu, když se hodiny vynásobí číslem 3600, minuty číslem 60 a potom se vše sečte včetně sekund a provede se u všech tří funkcí a tím dostaneme tři velká čísla, která už se dají jednodušeji porovnávat.

```

long startinsec() {DateTime datumCas = DS1307.now(); return EEPROM.read(0 + (datumCas.dayOfTheWeek()*10))*3600
+ EEPROM.read(1 + (datumCas.dayOfTheWeek()*10))*60 + EEPROM.read(2 + (datumCas.dayOfTheWeek()*10));}

long stoptinsec() {DateTime datumCas = DS1307.now(); return EEPROM.read(3 + (datumCas.dayOfTheWeek()*10))*3600
+ EEPROM.read(4 + (datumCas.dayOfTheWeek()*10))*60 + EEPROM.read(5 + (datumCas.dayOfTheWeek()*10));}

long actualinsec() {
    DateTime datumCas = DS1307.now();
    return datumCas.hour()*3600 + datumCas.minute()*60 + datumCas.second();
}

```

Obrázek 23: Funkce pro převod času na sekundy

Načtení dat pro světlo a teplotu

Funkce na obrázku níže slouží pro načtení proměnných z paměti EEPROM v závislosti na dnu v týdnu. Funkce je velice jednoduchá, porovná se datum s aktuálním a vrátí hodnotu proměnných pro aktuální den.

```

byte mintemp() {DateTime datumCas = DS1307.now(); return EEPROM.read(7 + (datumCas.dayOfTheWeek()*10));}

byte maxtemp() {DateTime datumCas = DS1307.now(); return EEPROM.read(8 + (datumCas.dayOfTheWeek()*10));}

byte svetlo() {DateTime datumCas = DS1307.now(); return EEPROM.read(6 + (datumCas.dayOfTheWeek()*10));}

```

Obrázek 24: Funkce pro načtení dat pro světlo a teplotu

Výpis aktuální teploty

„Aktteplota“ je velice jednoduchá funkce, která vypíše na sériový monitor aktuální teplotu naměřenou pomocí senzoru od firmy Dallas.

```

int aktteplota(){
    senzoryDS.requestTemperatures();
    return senzoryDS.getTempCByIndex(0);
}

```

Obrázek 25: Ukázka funkce pro výpis aktuální teploty

Testovací funkce příjem

Funkce „příjem“ slouží pouze pro účely testování správné funkce programu a na správný běh programu nemá žádný vliv. Funkce má vlastnosti pouze pro kontrolu či úpravu programu. Jedná se o funkci „switch“, pomocí které je možné vypsat nebo zapsat proměnné či si vypsat některé údaje nebo ověřit funkci relé. Funkce je v příloze.

10.1.2 Ukládání do paměti EEPROM

Zápis proměnných do paměti EEPROM

Zápis do paměti EEPROM je proveden pomocí pěti funkcí, které můžeme vidět na obrázku níže. Každá funkce má za úkol uložit data, která pošle aplikace z počítače. První z nich je funkce „light“, která zapisuje úroveň světla, podle které má spínač spínat. Funkce „starttime“ zapisuje čas, od kterého má být spínač sepnutý. Obdobně funguje „stoptime“,

který uloží čas ukončení sepnutého stavu. Poslední dvě funkce „starttemp“ a „stoptemp“ určují teplotní rozpětí, ve kterém může být spínač sepnutý. Každá funkce se ukládá na vlastní adresu 0-8, ke které je přičten den vynásobený deseti. Tím docílíme, že např. u „starttime“ se data uloží na adresy 10, 11, 12.

```
void light(){
    EEPROM.write(6 + den2*10,Serial.readStringUntil(';').toInt());
}

void starttime(){
    EEPROM.write(0 + den2*10,Serial.readStringUntil(';').toInt());
    EEPROM.write(1 + den2*10,Serial.readStringUntil(';').toInt());
    EEPROM.write(2 + den2*10,Serial.readStringUntil(';').toInt());
}

void stoptime(){
    EEPROM.write(3 + den2*10,Serial.readStringUntil(';').toInt());
    EEPROM.write(4 + den2*10,Serial.readStringUntil(';').toInt());
    EEPROM.write(5 + den2*10,Serial.readStringUntil(';').toInt());
}

void starttemp(){
    EEPROM.write(7 + den2*10,Serial.readStringUntil(';').toInt());
}

void stoptemp(){
    EEPROM.write(8 + den2*10,Serial.readStringUntil(';').toInt());
}
```

Obrázek 26: Ukázka funkce pro ukládání do paměti EEPROM

10.1.3 Podmínky spínání

Klíčovou částí programu jsou podmínky pro spínání, které zajišťují přečtení a porovnání všech proměnných s daty z paměti EEPROM. Dále zajišťují sepnutí či rozepnutí spínacího relé a tím sepnutí spínače na základě načtených dat. Podmínky jsou vytvořené podle vývojového diagramu, který je zobrazen v příloze. Do podmínek jsou vepsány komentáře, které mají za účel jednodušší orientaci. Jedná se o definování podmínek, za kterých spínač sepne, např. spínat podle času.

10.2 Aplikace pro konfiguraci pomocí počítače

Aplikace sestává z mnoha funkcí, ale budou popsány pouze ty nejdůležitější pro správnou funkci aplikace. Celý kód bude uveden v příloze na CD, nebo v elektronické podobě.

Funkce odešli

Tato funkce má za úkol poslat zadané údaje pomocí aplikace do paměti spínače prostřednictvím USB UART. Funkce funguje na jednoduchém principu. Nastavení lze provést i prostřednictvím sériového monitoru tak, že např. napíšeme číslo dvě a funkce potom očekává čas pro sepnutí, ve kterém jsou hodiny, minuty a sekundy odděleny středníkem. Funkce odešli otevře sériový port pro komunikaci a potom začne postupně odesílat data tak, že nejprve pošle číslo odkazující na funkci „příjem“, kde každé číslo

slouží pro jiný typ dat. Funkce postupně pošle všechna data.

```
void odesli(byte den)
{
    byte light = 0;

    if (checkBox2.Checked) light += 1;
    if (checkBox3.Checked) light += 2;
    if (checkBox4.Checked) light += 4;
    if (checkBox5.Checked) light += 8;

    serialPort1 = new SerialPort(comboBox1.SelectedItem.ToString(), 115200, Parity.None, 8, StopBits.One);
    serialPort1.Open();
    serialPort1.WriteLine("12;" + den + ";");
    serialPort1.WriteLine("2;" + Convert.ToInt32(numericUpDown1.Value) + ";" + Convert.ToInt32(numericUpDown2.Value) + ";" + Convert.ToInt32(numericUpDown3.Value) + ";");
    serialPort1.WriteLine("3;" + Convert.ToInt32(numericUpDown4.Value) + ";" + Convert.ToInt32(numericUpDown5.Value) + ";" + Convert.ToInt32(numericUpDown6.Value) + ";");
    serialPort1.WriteLine("4;" + light + ";"); //light
    serialPort1.WriteLine("5;" + Convert.ToInt32(numericUpDown7.Value) + ";"); //min temp
    serialPort1.WriteLine("6;" + Convert.ToInt32(numericUpDown8.Value) + ";"); //max temp
    serialPort1.Close();
}
```

Obrázek 27: Ukázka funkce odešli

Funkce obnov

Funkce obnov slouží pro kontrolu připojení spínače k počítači pomocí USB. Pokud je spínač připojený, aplikace se spustí. V případě nepřipojeného stavu dojde k zobrazení okna, které je popsáno v kapitole 11.2 Nepřipojený stav zařízení, a dojde k deaktivaci tlačítek.

```
private void obnov()
{
    comboBox1.Items.Clear();
    if (SerialPort.GetPortNames().Length == 0)
    {
        comboBox1.Items.Clear();
        MessageBox.Show("Nebylo nalezeno žádné zařízení, zkontrolujte připojení a opakujte");
        button1.Enabled = false;
        time.Enabled = false;
        pracdny.Enabled = false;
        celytyden.Enabled = false;
        pondeli.Enabled = false;
        utery.Enabled = false;
        streda.Enabled = false;
        ctvrtek.Enabled = false;
        patek.Enabled = false;
        sobota.Enabled = false;
        nedele.Enabled = false;
        combo.Enabled = false;
    }
    else
    {
        foreach (string currentPort in SerialPort.GetPortNames())
        {
            comboBox1.Items.Add(currentPort);
        }
        comboBox1.SelectedIndex = 0;
        button1.Enabled = true;
        time.Enabled = true;
        pracdny.Enabled = true;
        celytyden.Enabled = true;
        pondeli.Enabled = true;
        utery.Enabled = true;
        streda.Enabled = true;
        ctvrtek.Enabled = true;
        patek.Enabled = true;
        sobota.Enabled = true;
        nedele.Enabled = true;
        combo.Enabled = true;
    }
}
```

Obrázek 28: Ukázka funkce obnov

Funkce pro nastavení času

Funkce „time_click“ slouží pro kalibraci času v paměti RTC. Tento čas je nastaven podle času počítače. Nejprve se otevře komunikace po sériové lince. Potom se odešle číslo jedna, které je podle funkce „příjem“ pro nastavení času. Potom aplikace odešle datum a čas ve formátu „rok, měsíc, den, hodina, minuta, sekunda“, kde jsou jednotlivé hodnoty odděleny středníkem. Po odeslání nového času se zobrazí oznamovací okno, které informuje o nově nastaveném čase, ale vypíše ho již z paměti RTC.

```
private void time_Click(object sender, EventArgs e)
{
    DateTime time = DateTime.Now;

    serialPort1 = new SerialPort(comboBox1.SelectedItem.ToString(), 115200, Parity.None, 8, StopBits.One);
    serialPort1.Open();
    serialPort1.Write("1;" + time.Year + ";" + time.Month + ";" + time.Day + ";" + time.Hour + ";" + time.Minute + ";" + time.Second + ";");
    serialPort1.DataReceived += new SerialDataReceivedEventHandler(timekontrol);
}

private void timekontrol(object sender, SerialDataReceivedEventArgs e)
{
    SerialPort sp = (SerialPort)sender;
    MessageBox.Show("Aktuální čas spínače: " + sp.ReadExisting());
    sp.Close();
}
```

Obrázek 29: Funkce pro nastavení času

Funkce pro příjem aktuálního nastavení

Funkce „DataReceivedHandler“ slouží pro načtení nastavených dat z paměti spínače a zapsání do políček pro čas, teplotu a intenzitu osvětlení. Funkce funguje na stejném principu jako funkce „odešli“, nicméně údaje načítá a potom zapíše do příslušného pole. V pořadí šestá je hodnota intenzity osvětlení na čtyři úrovně. Pokud chceme spínat podle více úrovní, funkce si je opět rozdělí na příslušné hodnoty a deska pošle součet.

```
private void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
{
    SerialPort sp = (SerialPort)sender;
    string indata = sp.ReadExisting();
    string[] kody = indata.Split(';');

    numericUpDown1.Value = Convert.ToInt32(kody[0]);
    numericUpDown2.Value = Convert.ToInt32(kody[1]);
    numericUpDown3.Value = Convert.ToInt32(kody[2]);
    numericUpDown4.Value = Convert.ToInt32(kody[3]);
    numericUpDown5.Value = Convert.ToInt32(kody[4]);
    numericUpDown6.Value = Convert.ToInt32(kody[5]);
    //checkbox
    int bin = Convert.ToInt32(kody[6]);
    bool[] bins = new bool[4];
    for (int i = 0; i < 4; i++)
    {
        bins[i] = Convert.ToBoolean(bin % 2);
        bin = bin / 2;
    }
    checkBox2.Checked = bins[0];
    checkBox3.Checked = bins[1];
    checkBox4.Checked = bins[2];
    checkBox5.Checked = bins[3];
    numericUpDown7.Value = Convert.ToInt32(kody[7]);
    numericUpDown8.Value = Convert.ToInt32(kody[8]);
    sp.Close();
}
```

Obrázek 30: Funkce pro příjem aktuálního nastavení

11 OVLÁDÁNÍ PROGRAMU PRO KONFIGURACI SPÍNAČE

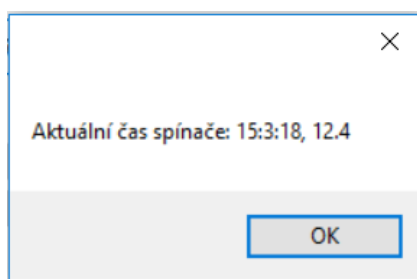
Pro pohodlné ovládání programu byla vytvořena aplikace pro platformu Windows, která umožňuje snadné nastavení všech proměnných a uložení do paměti spínače. Její ovládání je velice snadné a návod k obsluze je zobrazen v následující kapitole.

11.1 Návod pro konfigurační program

Ovládání programu je pomyslně rozděleno do tří zón. Každá zóna má svůj vlastní úkol.

11.1.1 Zóna 1

První ze zón má za úkol načíst COM port, přes který komunikuje aplikace se spínačem. V případě více připojených spínačů je možné si vybrat, který COM port bude obsluhován. Dále má za úkol nastavit aktuální čas podle hodin v PC. Po zapsání nového času se zobrazí oznamovací okno s aktuálním časem zapsaným do RTC viz obrázek níže.



Obrázek 31: Oznamovací okno po nastavení aktuálního času

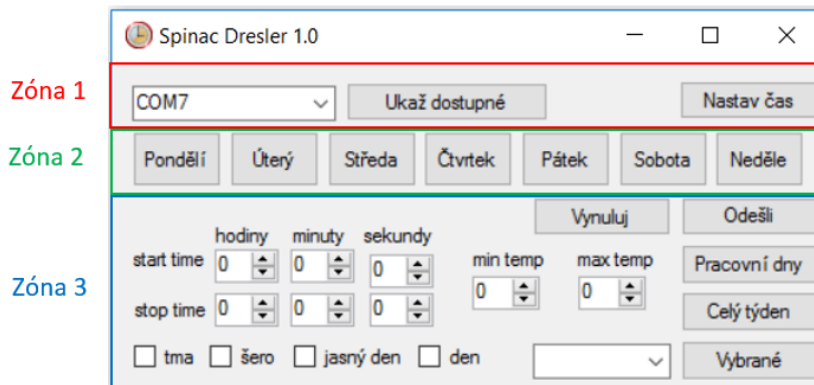
11.1.2 Zóna 2

V zóně číslo dvě si uživatel může vybrat, jaký den chce nastavovat. Při kliknutí na vybraný den se tlačítko toho dne zbarví do světle modra tak, aby uživatel věděl, jaký den nastavuje.

11.1.3 Zóna 3

Poslední zóna obsahuje ty nejpodstatnější komponenty. Pomocí těchto prvků se nastavují proměnné pro spínání, nebo se pomocí nich načítají či odesílají do paměti spínače. Tlačítko „Vynuluj“ slouží pro rychlé vynulování nastavených polí, jako je např. starttime, stoptime atd. Tlačítko „odešli“ slouží pro zapsání nastavených proměnných do paměti spínače. Tlačítka „Pracovní dny“ a „Celý týden“ zkopírují aktuální nastavení vybraného dne na pracovní dny, nebo na celý týden. Pole „combobox“ slouží společně s tlačítkem „Kopírovat“ ke zkopírování nastavení aktuálního dne na vybrané dny. Položky za nadpisem „start time“ slouží pro nastavení přesného času, odkdy bude deska spínat. Obdobně fungují položky za nadpisem „stop time“, nicméně nastavují čas rozepnutí desky. Podobnou funkci jako „start time“ a „stop time“ mají položky „min temp“ a „max

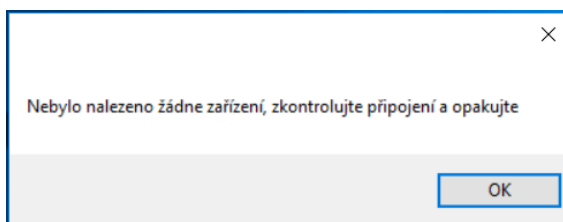
temp“, ale nastavují maximální a minimální teplotu, při které bude deska spínat. Poslední položkou jsou „check boxy“, pomocí kterých uživatel může nastavit spínání podle úrovně okolního osvětlení.



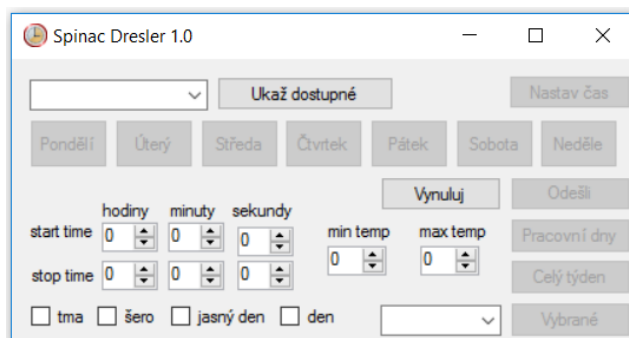
Obrázek 32: Rozdělení aplikace do zón

11.2 Nepřipojený stav zařízení

Pokud zapneme aplikaci, i když nemáme zařízení připojené, zobrazí se před spuštěním programu oznamovací okno, ve kterém je napsáno „Nebylo nalezeno žádné zařízení, zkontrolujte připojení a opakujte“. Pokud hlášku odklikneme tlačítkem „OK“, program se spustí, nicméně většina tlačítek je „zešedlá“, což signalizuje stav nepřipojení, a tlačítka nelze použít. Všechna tato tlačítka slouží k načítání nebo odesílání dat do spínače, proto je omezeno jejich použití. Stav programu můžeme vidět na obrázku níže.



Obrázek 33: Oznamovací okno pro nepřipojený stav



Obrázek 34: Ukázka programu při nepřipojeném stavu

12 TESTOVÁNÍ SPÍNAČE

12.1 Testovací prostředí

Spínač byl testován pro spínání LED osvětlení na terase. Zařízení bylo umístěné v elektrickém rozvaděči navazující přímo na terasu. V rozvaděči je suché prostředí a je použit pouze pro spínání několika zařízení, nikoliv pro umístění jističů apod. Spínač byl napájen pomocí 12 V zdroje. Rozvaděč a LED osvětlení je vidět na fotografiích níže.

Při testování spínače byly použity podmínky pro čas a teplotu, nikoliv pro světlo, a to z důvodu, že je spínač umístěn uvnitř rozvaděče, kde je trvalá tma. Během testování byly změřeny proudové odběry, kdy při nesepnutém stavu má spínač odběr 26,4 mA a při sepnutém stavu 138,4 mA. Proudový odběr při sepnutém stavu je již poměrně velký. V případě potřeby snížení proudového odběru by bylo možné vyměnit stávající elektromagnetické relé za jiný typ relé s DC cívkou, které bude schopno, jak spínat větší proud, tak bude mít i menší proudový odběr.



Obrázek 35: Umístění LED osvětlení pro testování spínače



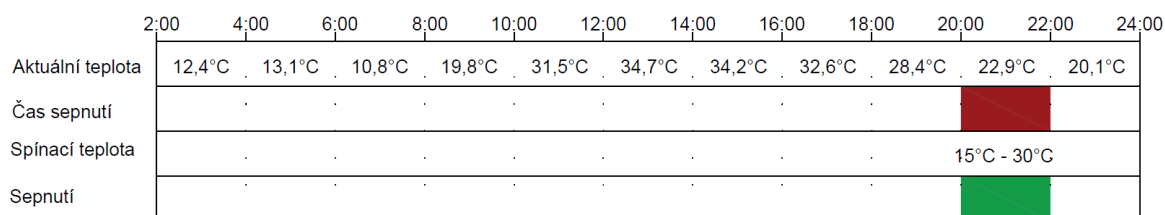
Obrázek 36: Umístění spínače během testování

12.2 Výsledky testování

Spínač byl testován dva dny. Jak již bylo řečeno, testoval se pouze pomocí času a teploty. Pro každý den byla vytvořena časová osa s nastavenými hodnotami, průběhem teploty a časem.

12.2.1 První testování

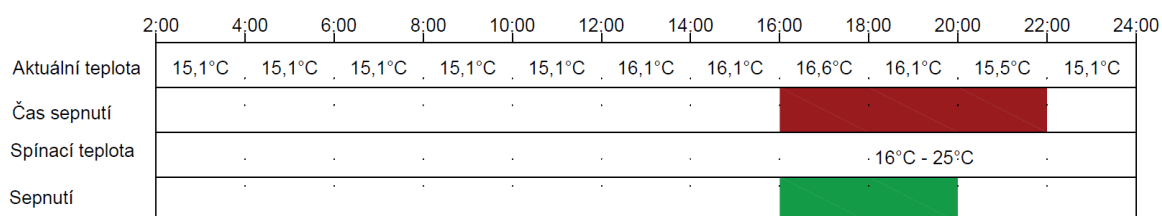
Během prvního testování byl spínač nastavený na čas sepnutí od 20:00 do 22:00 a při teplotě od 15 °C do 30 °C. Během časového úseku byla teplota 22,9 °C. Jelikož teplota byla v daném rozmezí, byl spínač sepnutý.



Obrázek 37: Výsledky prvního testování

12.2.2 Druhé testování

Při druhém testování bylo časové rozmezí nastaveno od 16:00 do 22:00 při spínací teplotě od 16 °C do 25 °C. Na začátku testování spínač sepnul přesně v 16:00, ale ve 20:00 se teplota snížila pod 16 °C, čímž nebyla splněna podmínka pro sepnutí, a spínač se rozeplnul.



Obrázek 38: Výsledky druhého testování

13 NOVÁ VERZE SPÍNAČE

Původní spínač, navržený pro semestrální projekt, vykázal několik malých nedostatků. Proto bylo potřeba uskutečnit několik úprav (a vylepšení) pro kvalitnější funkci spínače.

13.1 Opravené nedostatky

Při výrobě desky se objevil problém se špatným zapojením RTC a USB UART. Pro původní spínač byly tyto chyby opraveny pomocí externích propojení, aby spínač fungoval. Nová verze žádný z těchto nedostatků nemá. Dalším nedostatkem bylo umístění teploměru Dallas, který byl nízko a blízko jiných součástí, které ovlivňovaly okolní teplotu. Tento nedostatek byl vyřešen umístěním teploměru výše nad desku.

13.2 Vylepšení pro provoz

Nová verze byla vylepšena několika drobnými úpravami. První z nich je přidání dalšího přepínače, který má za úkol umožnit spuštění úsporného režimu. při kterém spínač vypne svoji funkci a každých třicet sekund přeskóčí do normálního režimu. Při něm odečte data ze senzorů a zkontroluje, jestli se nezměnily podmínky pro sepnutí, a vykoná potřebné kroky. Tento mód umožňuje delší životnost baterie v případě, že bude spínač napájen pomocí externí baterie místo zdroje na 230 V.

Dalším vylepšením je vysunutí USB portu a power jacku mimo desku. Toto vysunutí bylo provedeno z důvodu umístění spínače do krabičky z plexiskla. Porty se vysunuly tak, aby nebyly „utopené“ uvnitř krabičky, a tudíž se mohly dobře používat.

Poslední vylepšení se věnuje oběma přepínačům, které jsou nahrazeny jumper lištami, a to z důvodu umístění přepínačů na povrch krabičky k snadnějšímu ovládání. Přepínače budou s jumper lištami propojeny pomocí vodičů. Tím bude možné krabičku rozdělat a spínač vyjmout bez složitého demontování spínačů.

14 OBAL PRO SPÍNAČ

Obal je zhotovený z čirého plexiskla o tloušťce 5 mm. V obalu bude několik otvorů, a to z důvodu připojení napájení, USB a připojení zařízení pro spínání. Obal je složen ze šesti částí. Části jdou navrženy tak, aby se složili do sebe bez použití lepidla. Vrchní část je navržena tak, aby se dala otevírat.

14.1 Výroba obalu

Obal, jak již bylo zmíněno, je z plexiskla a je vyřezán do potřebných rozměrů pomocí laserové rezačky, která zaručí přesný řez, aniž by deformovala plexisklo. Rozměry byly zvoleny podle rozměrů desky. Do plexiskla jsou vyřezány otvory pro připojení kabelů a mřížka pro lepší chlazení stabilizátoru. Hotový obal je na obrázku níže. Spínač je uvnitř obalu přišroubován na spodní stranu tak, aby při otočení obalu nedošlo k poškození.



Obrázek 39: Spínač umístěný v obalu

15 ZÁVĚR

Tématem bakalářské práce bylo navržení a zkonstruování programovatelného časového spínače konfigurovatelného přes USB a následné navržení programu pro funkci spínače včetně konfiguračního programu pro počítač.

První část práce se zabývá navržením a zkonstruováním programovatelného časového spínače konfigurovatelného přes USB. Při samotném návrhu bylo možné postupovat několika způsoby. Jednou z variant byl výběr typu součástek ze dvou možností, a to THT nebo SMD. Pro práci byl vybrán typ THT. Malým nedostatkem tohoto typu součástek je jejich velikost. Součástka pro USB UART se nevyrábí v provedení THT, a proto se použitá deska musela upravit tak, aby byla oboustranná. Z toho důvodu je součástka pro UART osazena na spodní straně v provedení SMD. Při navrhování bylo velmi důležité zjistit si prodejní dostupnost výše zmíněných součástek.

V další části návrhu byl pro spínač vymyšlen obal, který zajišťuje ochranu spínače a zároveň umožňuje přístup k hlavním ovládacím částem spínače.

Dále se práce zabývá naprogramováním MCU pro správnou funkci spínače. Program musí umět načítat hodnoty od všech senzorů, tak aby mohl správně spínat. Umožňuje tak spínat na základě teploty, času i světla.

Pro splnění všech požadavků bylo potřeba do práce navrhnout také obvod pro teploměr, fotorezistor a obvod pro reálný čas. Bylo velmi obtížné je správně umístit. Kdyby byl fotorezistor umístěn blízko LED diody, která signalizuje napájení, mohla by být ovlivněna jeho funkce.

Závěrečná část práce se věnuje programu pro konfiguraci pomocí počítače. Program je velice jednoduchý a umožňuje snadné nastavení spínání pro každý den. Poskytuje také snadnou kalibraci času. Program umožňuje připojení více zařízení pomocí COM portu.

LITERATURA

- [1] *VT93N2: Photoconductive Cells*. [Waltham, Massachusetts, U.S.]: PerkinElmer, 2001. Dostupné také z: <http://html.alldatasheet.com/html-pdf/84405/PERKINELMER/VT93N2/46355/76/VT93N2.html>
- [2] *VT43N1: Photoconductive Cells*. [Waltham, Massachusetts, U.S.]: PerkinElmer, 2001. Dostupné také z: <http://html.alldatasheet.com/html-pdf/84379/PERKINELMER/VT43N1/46355/76/VT43N1.html>
- [3] *A906012: Photocells*. [Waltham, Massachusetts, U.S.]: PerkinElmer, 2003. Dostupné také z: <http://www.tme.eu/cz/Document/a820537624aad4b6d9dbc11ba592fea4/A906012.pdf>
- [4] *DS18S20: High Precision 1-Wire Digital Thermometer*. [Dallas, Texas, U.S.]: Dallas semiconductor, b.r. Dostupné také z: <https://www.gme.cz/data/attachments/dsh.530-070.1.pdf>
- [5] *DS18B20: Programmable Resolution 1-Wire Digital Thermometer*. [San Jose (CA, U.S.)]: Maxim Integrated, 2016. Dostupné také z: <https://www.gme.cz/data/attachments/dsh.530-067.1.pdf>
- [6] *LM35: Precision Centigrade Temperature Sensors*. [Santa Clara, CA, U.S.]: Texas Instrument, 2000. Dostupné také z: <https://www.gme.cz/data/attachments/dsh.313-909.1.pdf>
- [7] *DS1302: Trickle-Charge Timekeeping Chip*. [San Jose (CA, U.S.)]: Maxim Integrated, 2004. Dostupné také z: <https://www.gme.cz/data/attachments/dsh.433-112.1.pdf>
- [8] *PCF8593: Low poewr clock/calendar*. [Amsterdam, Netherlands]: Philips, 1997. Dostupné také z: <https://www.gme.cz/data/attachments/dsh.433-126.1.pdf>
- [9] *DS1306: Serial Alarm Real-Time Clock*. [San Jose (CA, U.S.)]: Maxim Integrated, 2005. Dostupné také z: <http://www.tme.eu/cz/Document/70fbd5f4c4537801ae6323a51730bb45/ds1306.pdf>
- [10] *ATmega328P: 8-bit Microcontroller*. [San Jose, USA]: Atmel Corporation, 2009. Dostupné také z: <https://www.gme.cz/data/attachments/dsh.432-228.1.pdf>
- [11] *ATmega8L: 8-bit Microcontroller*. [San Jose, USA]: Atmel Corporation, 2003. Dostupné také z: <https://www.gme.cz/data/attachments/dsh.432-193.1.pdf>
- [12] *ATmega16A: 8-bit Microcontroller*. [San Jose, USA]: Atmel Corporation, 2014. Dostupné také z: <https://www.gme.cz/data/attachments/dsh.432-232.1.pdf>
- [13] Encyklopedie Fyziky: Fotorezistor. *Fyzika.jreichl* [online]. ČR: Encyklopedie Fyziky, 2006 [cit. 2016-11-06]. Dostupné z:

<http://fyzika.jreichl.com/main.article/view/270-fotorezistor>

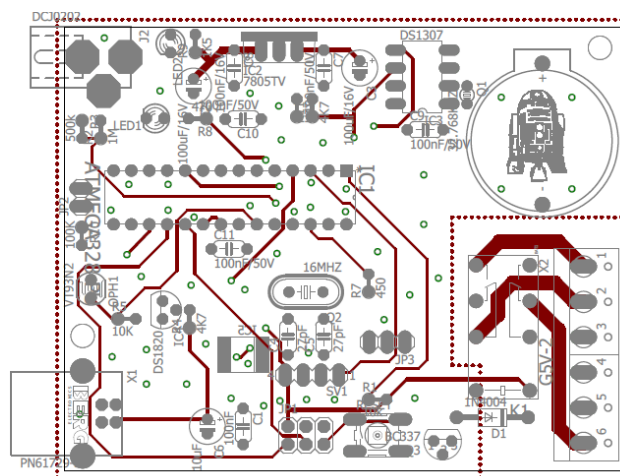
- [14] Fotoodpor: fotorezistor. *ELUC* [online]. Olomouc: ELUC, 2009 [cit. 2016-11-06]. Dostupné z: <https://eluc.kr-olomoucky.cz/verejne/lekce/610>
- [15] NOVÁK, Lukáš. *MĚŘENÍ TEPLoty POMOCÍ DVOU IDENTICKÝCH ČIDEL* [online]. Brno, 2014 [cit. 2016-11-01]. Dostupné z: https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/35613/Bakalarska_prace.pdf?sequence=2&isAllowed=y. Bakalářská práce. VUT Brno. Vedoucí práce Michal Pavlík.
- [16] DS18B20 Programmable Resolution 1-Wire Digital Thermometer. *ALLDATASHEET* [online]. San Jose: Maxim Integrated, 2008 [cit. 2016-11-01]. Dostupné z: <http://www.alldatasheet.com/datasheet-pdf/pdf/433921/MAXIM/DS18B20.html>
- [17] Hodiny reálného času. *Kiwiki* [online]. Trenčín: kiwiki, 2009 [cit. 2016-11-17]. Dostupné z: http://www.kiwiki.info/index.php/Hodiny_re%C3%A1lneho_%C4%8Dasu
- [18] Použití převodníku USB: TTL UART. *Tajned* [online]. tajned, 2013 [cit. 2016-11-17]. Dostupné z: <http://www.tajned.cz/2015/03/pouziti-prevodniku-usb-uart/>
- [19] První program. *Arduino návody* [online]. Havlíčkův Brod: ECLIPSE s.r.o, 2016 [cit. 2017-05-02]. Dostupné z: <http://navody.arduino-shop.cz/zaciname-s-arduinem/prvni-program.html>
- [20] MEDLA, EDUARD. *MODERNÍ ZPŮSOBY PROGRAMOVÁNÍ MIKROKONTROLÉRU*. BRNO, 2015. BAKALÁŘSKÁ PRÁCE. VUT FSI. Vedoucí práce Ing. DANIEL ZUTH, Ph.D.
- [21] Microsoft Visual Studio. *Wikipedia* [online]. Florida (USA): Wikimedia Foundation, 2017 [cit. 2017-05-02]. Dostupné z: https://cs.wikipedia.org/wiki/Microsoft_Visual_Studio

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

MCU	Microcontroller unit
USB	Universal Serial Bus
RTC	Real-time clock
LED	Light-Emitting Diode
UART	Universal Asynchronous Receiver and Transmitter
GND	GrouND
CMOS	Complementary Metal–Oxide–Semiconductor
RAM	Random-access memory
EEPROM	Electrically Erasable Programmable Read-Only Memory.
THT	Through-hole technology
SMD	Surface mount device
COM	Communication port

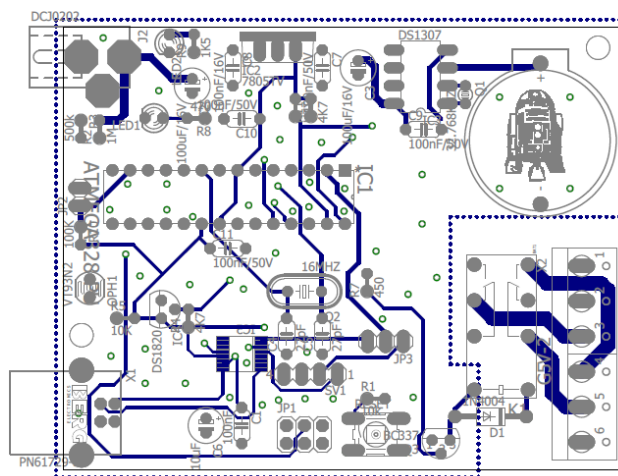
A NÁVRH ZAŘÍZENÍ

A.1 Deska plošného spoje řídicího modulu – top (strana součástek)



Rozměr desky 63 x 79 [mm], měřítko M1:1

A.2 Deska plošného spoje řídicího modulu – bottom (strana spojů)



Rozměr desky 63 x 79 [mm], měřítko M1:1

B SEZNAM SOUČÁSTEK

Označení	Hodnota	Pouzdro	Popis
BAT1		BH2032	Držák baterie
C1	100 nF	C050-024X044	Fóliový kondenzátor
C2	100 μ F/16 V	E2-5	Elektrolytický kondenzátor
C3	100 μ F/16 V	E2-5	Elektrolytický kondenzátor
C4	27 pF	C050-024X044	Keramický kondenzátor
C5	27 pF	C050-024X044	Keramický kondenzátor
C6	10 μ F	E2-5	Elektrolytický kondenzátor
C7	100 nF/50 V	C050-024X044	Fóliový kondenzátor
C8	100 nF/16 V	C050-024X044	Fóliový kondenzátor
C9	100 nF/50 V	C050-024X044	Fóliový kondenzátor
C10	100 nF/50 V	C050-024X044	Fóliový kondenzátor
C11	100 nF/50 V	C050-024X044	Fóliový kondenzátor
D1	1N4007	DO41-10	Dioda
IC1		DIP254P762X406-28P	Mikrokontroler
IC2		TO220V	Regulátor napětí
IC3		DIL08	Obvod RTC
IC4		TO-92	Teploměr
IC5		SSOP16	UART
J1		DCJ0202	Napájecí jack
JP1		2X03	Jumper lišta
K1		G5V-2	Relé
LED1		LED3MM	LED dioda
LED2		LED3MM	LED dioda
OPH1		VT900	Fotorezistor
Q1	32.768 KHZM	TC26V	Krystal
Q2	16 MHZ	HC49U-V	Krystal
Q3	BC337	TO92	NPN tranzistor
R1	10k	0204V	Rezistor
R2	1M	0204V	Rezistor
R3	500k	0204V	Rezistor
R4	4K7	0204V	Rezistor
R5	10k	0204V	Rezistor
R6	820	0204V	Rezistor
R7	820	0204V	Rezistor
R8	470	0204V	Rezistor
R9	1K5	0204V	Rezistor
R10	100k	0204V	Rezistor
R11	4K7	0204V	Rezistor

R12	4K7	0204V	Rezistor
RESET		B3F-10XX	Spínač
S1		255SB	Přepínač
SV1		MA04-1	Jumper lišta
X1		PN61729-S	USB konektor
X2		W237-6	Svorkovnice

C TESTOVACÍ FUNKCE PŘÍJEM

```
void prijem(){
  if(Serial.available()){
    int pom = Serial.readStringUntil(';').toInt();
    switch (pom){
      case 1:
        settime();
        break;
      case 2:
        starttime();
        break;
      case 3:
        stoptime();
        break;
      case 4:
        light();
        break;
      case 5:
        starttemp();
        break;
      case 6:
        stoptemp();
        break;
      case 7:
        sendset();
        break;
      case 8:
        sepni();
        break;
      case 9:
        rozepni();
        break;
      case 10:
        Serial.print("Teplota cidla DS18B20: ");
        Serial.print(aktteplota());
        Serial.println();
        break;
      case 11:
        timekontrol();
        break;
      case 12:
        den2 = Serial.readStringUntil(';').toInt();
        break;
    }
  }
}
```

D VÝVOJOVÝ DIAGRAM

