

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ
Fakulta elektrotechnická
katedra radioelektroniky

Návrh řídicí jednotky pro smarthome

Diplomová práce

Vedoucí práce: Ing. Máša Pavel, Ph.D.

Diplomant: Bc. Petr Cuchý

květen 2014

Anotace:

Práce se zabývá možnostmi inteligentního domu a jeho komponentami. V práci je kompletní hardwarový i softwarový návrh řídicí jednotky s procesorem ATXmega, který umožňuje regulaci osvětlení, měření a řízení teploty, ovládání zásuvek, elektrických žaluzií, zabezpečovací techniky a solárního ohřevu vody. Kromě hlavní řídicí jednotky práce řeší i návrh rozšiřujících slave jednotek. Obslužný program je napsán v prostředí AVR Studia 6 v jazyku C++. Jednotka disponuje grafickým displejem, pomocí kterého můžeme nastavovat velké množství parametrů a programů, které ovlivňují celodenní řízení celého systému. Komunikuje s PC, které slouží jako webový server, a díky tomu máme možnost ovládání přes webové rozhraní v rámci lokální sítě.

Klíčová slova:

Řídicí jednotka pro inteligentní domy, inteligentní elektroinstalace, smarthome, regulace teploty, regulace osvětlení, elektronické zabezpečovací systémy, procesor ATXMega, procesor ATMEGA, Eagle, AVR Studio6, grafický displej, 1-wire, RS485, RS232, MYSQL, C# aplikace pro sériovou linku a databázi, PHP, HTML stránky.

Summary:

The work deals with the possibilities of the smart building and its components. The thesis covers complete hardware and software design of the control unit with processor ATXMega, which provides lighting control, temperature measurement and control, electric outlet control, electric slat control, electronic security systems and control system for solar water heating. The thesis also covers the design of the expansion slave units. The firmware is written in the AVR Studio6 environment in the C++ programming language. The control unit has a graphical display, by which we can set up a large number of parameters and control programs. It communicates with the PC, which serves as the Web server, and thanks to this, we are able to control the unit via a web interface in the local network.

Index Terms:

Control unit of smart home, smart wiring, smart home, temperature control, lighting control, electronic security systems, ATXMega processor, ATMega, Eagle, AVR Studio6, graphic display, 1-wire, RS485, RS232, MYSQL, C# applications for serial port and database, PHP, HTML pages.

Poděkování

Rád bych touto cestou poděkoval Ing. Pavlu Mášovi, Ph.D, za odborné vedení a cenné rady při tvorbě této práce. Dále bych chtěl poděkovat mé rodině za všestrannou podporu při studiu na ČVUT v Praze.

Prohlášení

Prohlašuji, že jsem diplomovou práci Návrh řídicí jednotky pro smarthome vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k diplomové práci.

Nemám námitky proti půjčování, zveřejnění a dalšímu využití práce, pokud s tím bude souhlasit katedra radioelektroniky.

.....

podpis studenta

V Praze dne

Obsah

Obsah	1
1 Úvod	4
2 Inteligentní budova	4
2.1 Definice inteligentní budovy	4
2.2 Komponenty a možnosti v inteligentní budově	5
2.2.1 Vytápění, ventilace a klimatizace	5
2.2.2 Ohřev teplé vody	7
2.2.3 Osvětlení	7
2.2.4 Bezpečnost	8
2.2.5 Zahrada a garáž.....	8
3 Stávající nabídky Inteligentních instalací.....	8
3.1 ABB	8
3.2 InsightHome.....	10
3.3 Savant	11
3.4 Teco a.s.	11
3.5 Elkoep	12
3.6 Fibaro.....	13
4 Návrh nového řešení	15
4.1 Ovládání světel	15
4.2 Zabezpečovací technika	17
4.3 Elektroventily a servoventily pro topení.....	18
4.4 Datové sběrnice.....	19
4.4.1 Sběrnice RS485	19
4.4.2 Sběrnice CAN	20
4.4.3 1-Wire	22
4.4.3.1 Bus driver DS2482-100.....	23
4.4.4 Opentherm	24
4.5 Měření teploty	24
4.5.1 Čidlo Pt100	25
4.5.2 Digitální čidlo TC74	26
4.5.3 DS18B20	26
4.6 Rozšiřující moduly - Slave	27

5	Hardwarový návrh.....	27
5.1	Procesor.....	27
5.1.1	Procesory Atmel ATXmega.....	27
5.1.2	Procesory Atmel ATmega.....	29
5.1.3	Procesory PIC.....	29
5.2	Grafický displej 128x64.....	29
5.3	Napájecí zdroj.....	31
5.4	Zapojení procesoru.....	32
5.5	Vstupy výstupy.....	33
5.6	Výstupy pro regulaci osvětlení.....	34
5.7	Slave – rozšiřující jednotka.....	35
6	Softwarový návrh.....	37
6.1	Řídicí jednotka MCU ATXmega256A3.....	38
6.1.1	Grafický displej.....	38
6.1.1.1	Knihovna LCDLIB.h.....	38
6.1.1.2	Funkce pro ovládání pinů.....	39
6.1.1.3	Funkce pro zobrazování znaků o velikosti 8 pixelů.....	40
6.1.1.4	Funkce pro zobrazování znaků o velikosti 14 pixelů.....	41
6.1.1.5	Funkce pro pixel.....	42
6.1.1.6	Funkce pro vykreslení úsečky a kružnice.....	42
6.1.1.7	Funkce pro zobrazení obrázku.....	44
6.1.2	USART.....	44
6.1.3	Tlačítka.....	44
6.1.4	Vstupy a výstupy.....	45
6.1.5	Časovače pro stmívání světla.....	45
6.1.6	Řídicí program.....	46
6.1.7	Regulace topení.....	48
6.1.8	Ovládání a menu nabídky.....	48
6.2	Aplikace C# pro sériový port a databázi.....	50
6.3	MYSQL.....	51
6.4	Webové uživatelské rozhraní – Html, Php.....	51
7	Závěr.....	53
8	Zdroje.....	55

Seznam použitých symbolů a zkratek

EZS Elektronický zabezpečovací systém

VoIP Voiceover Internet Protocol

IGBT InsulatedGateBipolar Transistor

RTC Real timecounter

USB Universal serial bus

PWM Pulse widthmodulation

USART Universal Synchronous / AsynchronousReceiver and Transmitter

Seznam příloh

Příloha [1] Zobrazení menu řídicí jednotky – ovládání světel

Příloha [2] Zobrazení menu řídicí jednotky – nastavení teploty

Příloha [3] Zobrazení menu řídicí jednotky – zabezpečovací systém

Příloha [4] Zobrazení menu řídicí jednotky – nastavení režimu

Příloha [5] Zobrazení menu řídicí jednotky – nastavení

Příloha [6] Zobrazení menu řídicí jednotky – servis

Příloha [7] Zobrazení menu řídicí jednotky – komunikace

Příloha [8] Okno aplikace napsané ve Visual Studiu v C#

Příloha [9] Fotografie hlavní řídicí jednotky

Příloha [10] Fotografie hlavní řídicí jednotky

Příloha [11] Webová stránka Smart Home – ovládání světel

Příloha [12] Webová stránka Smart Home – nastavení teplot

Příloha [13] Webová stránka Smart Home – EZS

Příloha [14] Webová stránka Smart Home – volba režimu

Příloha [15] Webová stránka Smart Home – budík

Příloha [16] Plošný spoj hlavní řídicí jednotky- TOP

Příloha [17] Plošný spoj hlavní řídicí jednotky - BOTTOM

Příloha [18] Plošný spoj slave jednotky

Příloha [19] Slave jednotka

1 Úvod

Tato práce pojednává, co vlastně znamená inteligentní budova, jaké jsou možnosti využití oproti klasické elektroinstalaci. Přiblížíme si jednotlivé funkce a prvky. Inteligentní budovy se dnes těší veliké oblibě a to přináší i nespočet firem, které se touto problematikou zabývají. Zaměříme se na stávající řešení a nabídky některých firem a jejich produkty. Dále se zaměříme na vlastní návrh řídicího systému pro inteligentní budovu typu rodinný dům, hlavním cílem je co nejlevnější řešení, samozřejmě ne na úkor kvality. Zvážíme si možnosti jednotlivých možností nového návrhu a zhodnotíme výhody a nevýhody. Uvažujeme, že budeme stavět rodinný dům a nebude pro nás problém natáhnout nějaký ten kabel, jak datový tak silový navíc a přitom navrhnout řešení přímo na míru s možností lehkých dodatečných úprav. Tím máme na mysli rozšiřující moduly a externí prvky. Budeme se zajímat o měření a regulaci teploty v jednotlivých místnostech, ovládání světel, okenních žaluzií, spínání zásuvek, řízení solárního ohřevu užitkové vody a podobné aplikace. Řídicí jednotka bude vybavena displejem pro nastavení a monitorování stavu systému. Bude zcela autonomní, s rozšířením o komunikaci s PC pro pohodlné ovládání přes webový prohlížeč. Tento systém by měl zpříjemnit pobyt v rodinném domě, být užitečný, zautomatizovat některé části budovy. Určitě nám bude příjemné, když se ráno probudíme pomalým roztažením rolet a lehkým rozsvícením lampičky, nebo si pohodlně nastavíme teplotu v jednotlivých místnostech na dobu, když přijdeme z práce.

2 Inteligentní budova

2.1 Definice inteligentní budovy

Inteligentní dům pro nás znamená automatizovanou nadstavbu klasické instalace jak silnoproudé elektřiny, tak datových sítí i audio-video prostředí. Jedná se o komplexní systém, který nám zpříjemní a ulehčí pobyt v domě. Například, když odcházíme z domu a zapomene zhasnout nějaké světlo, tak pouhým jedním tlačítkem vypneme všechna světla i třeba varnou konvici nebo elektrický sporák a můžeme si být jistí, že při naší nepřítomnosti v domě se nic nestandardního nestane. Inteligentní dům se za nás stará o regulaci topení, nemusíme chodit ke každému radiátoru a nastavovat na něm požadovanou teplotu. Může nás ráno

vzbudit oblíbenou písničkou, vytáhnout rolety a otevřít ventilační okno, zatímco se nám v kuchyni začne vařit voda na ranní čaj. Nestará se jenom o naše pohodlí, ale i naši bezpečnost, může přes noc monitorovat pohyb na určitých místech, nebo spustit požární alarm v případě požáru. Také možnost úspory energie vhodným řízením je velice příjemná. Systém nám dává komplexní orientaci nad naším rodinným domkem.

Rozlišujeme různé stupně inteligence těchto řídicích systémů: [1]

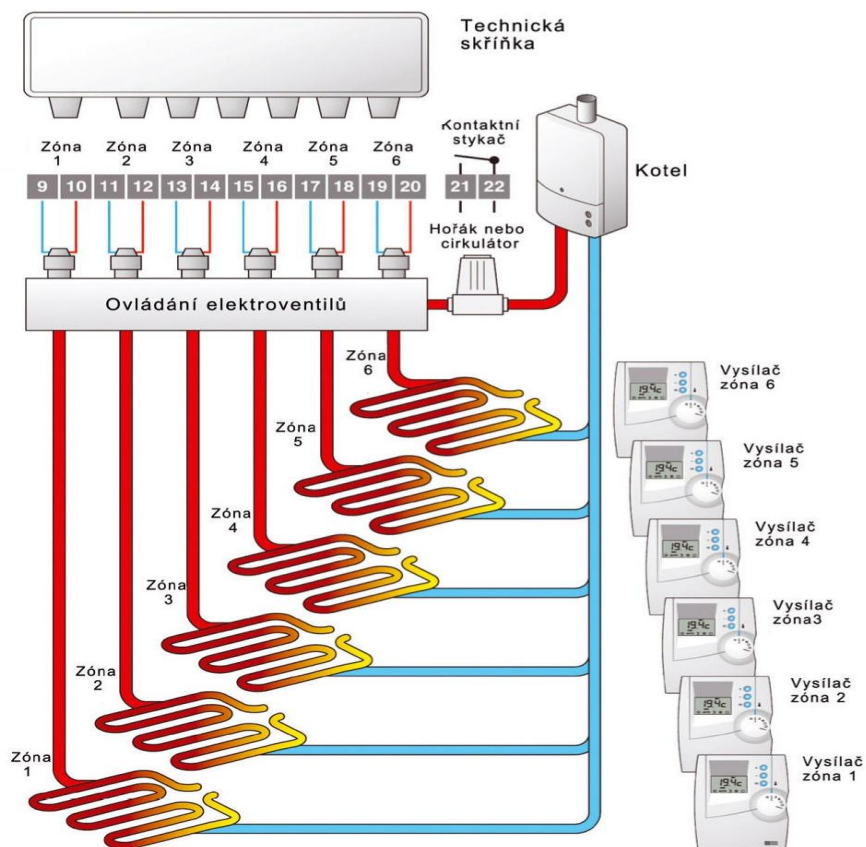
- **Obsahující inteligentní zařízení a systémy** – jedná se o malé samostatné řídicí struktury, které spolu nijak nekomunikují
- **Obsahující inteligentní komunikující zařízení a systémy** – budova obsahuje inteligentní systémy, které již spolu komunikují a snaží se celkově zdokonalit činnost.
- **Propojený dům** - obsahuje jak vnitřní, tak vnější komunikační síť, poskytuje interaktivní ovládání z domu i mimo něj.
- **Učící se dům** – systém monitoruje a zaznamenává stavy systému a pohyb lidí a tyto data používá k samočinnému ovládání podle předvídaných potřeb.
- **Pozorný dům** – zde jsou vyhodnocovány okamžité aktivity a polohy lidí, nevyhodnocuje historická data, ale vše probíhá v reálném čase.

2.2 Komponenty a možnosti v inteligentní budově

Zde se dostáváme k popisu a přiblížení možností funkcí v inteligentní budově, samozřejmě že jsou dnes k dostání různé komponenty od elektricky ovládaných rolet až po automatizovaný bar na pití. Všechno je otázka peněz a dostupnosti, proto se zaměříme spíše na standardní komponenty.

2.2.1 Vytápění, ventilace a klimatizace

Z hlediska vytápění se dostáváme k otázce, jak pohodlně regulovat teploty v jednotlivých místnostech bez manuálního zásahu. V praxi se hovoří o „zónové regulaci“, celý dům máme rozdělen do zón, které jednotlivě regulujeme např. třicestným ventilem, takže v každé zóně nám proudí voda s odlišnou teplotou. Schéma vidíme na obrázku 1.



Obrázek 1 Zónová regulace.[20]

Každá ze zón má svůj termostat, který ovládá určitý elektroventil. Máme-li možnost např. ovládat elektrický kotel, nebo máme klasický kotel, který je vybaven třicestným ventilem, a jsme schopni regulovat vstupní teplotu vody před zónovou regulací a máme informaci o venkovní teplotě, hovoříme o ekvitermní regulaci. Tato regulace je schopna upravit teplotu vstupní vody do systému podle venkovní teploty, nebo rozdílu aktuální a nastavené teploty v místnosti. Možnosti zdrojů tepla jsou také různorodé, od klasického kotle na tuhá paliva, plynových a elektrických kotlů až po tepelná čerpadla.

Ovládat můžeme i ventilaci, nebo detekovat otevřené okno. V případě, kdy je otevřené okno, vypneme topení pod ním a tím šetříme energii a zamezíme např. dlouhodobému topení s otevřeným oknem. Další komponenta pro ventilaci jsou stropní větráky, které můžeme regulovat, nebo měnit směr otáčení, v závislosti na roční době.

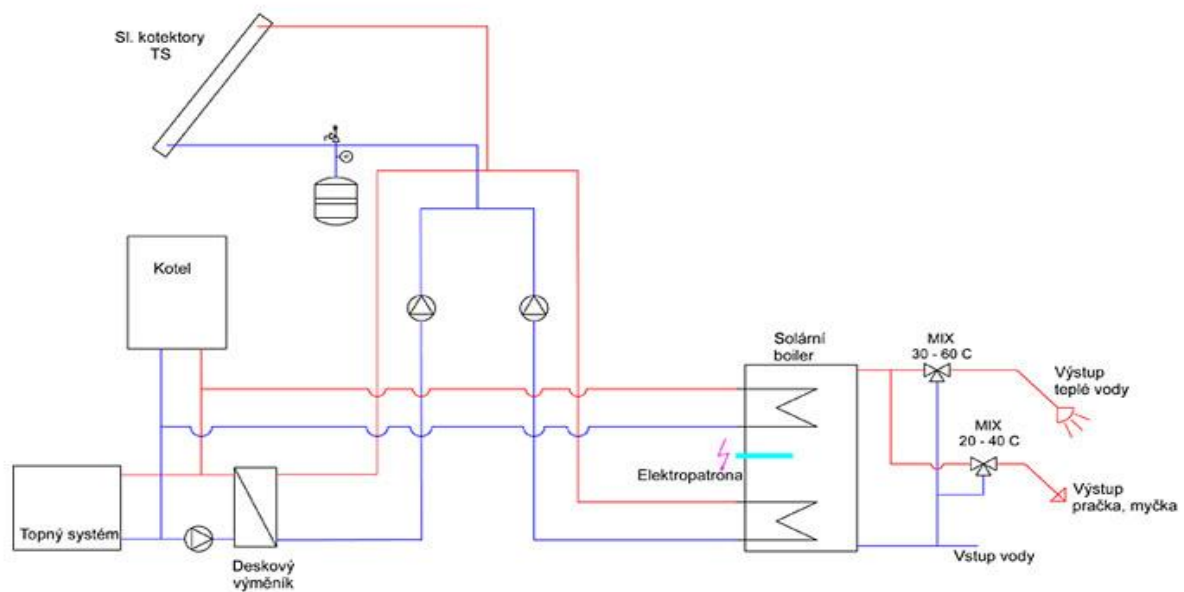
Klimatizace se určitě využije v letních měsících. Můžeme zde použít teplotní čidla a termostaty pro topení, nejvýhodnější ochlazování je při instalaci stropního nebo stěnového

topení, kdy je efektivita nejvyšší. Dále pak můžeme regulovat žaluzie a zamezit tím osvětlení ploch domu zejména na jižní straně domu.

2.2.2 Ohřev teplé vody

Zde již existuje mnoho systémů na solární ohřev užitkové vody, často kombinovaný právě s vytápěním. Jako poslední možnost je brán ohřev elektrickou topnou spirálou. Na obrázku 2 vidíme schéma systému, kdy teplá užitková voda je primárně ohřívána solárními kolektory, když není dostatečná sluneční intenzita, můžeme vodu ohřát kotlem, nebo jako poslední možnost elektrické těleso.

Solární panely mají vysokou účinnost a vyrábí se v klasické nebo ve vakuové variantě. Tyto panely jsou významným ušetřením pro ohřev vody, dosahuje se až 75–80% úspory energií. Na každých 100l užitkové vody se počítá jeden solární panel cca 2m². [2]



Obrázek 2 Realizace topné soustavy a ohřevu teplé užitkové vody.[21]

2.2.3 Osvětlení

Dostáváme se k nejpoužívanější komponentě. Ovládání osvětlení v málo nebo krátkodobě používaných místnostech můžeme vyřešit například detektorem pohybu. V ostatních případech můžeme pohodlně regulovat veškeré osvětlení v domě. Nastavíme si budík a ráno nás probudí lehce rozsvícená lampička, nebo při odchodu z domu se všechna zapomenutá světla automaticky zhasnou. Veliký komfort nám udělá nastavení světelných scén, například

v době dívání na televizor nebo projekční plátno si upravíme hlavní a ostatní osvětlení pouhým zmáčknutím jediného tlačítka. Můžeme měřit intenzitu osvětlení v pracovně a automaticky dorovnávat deficit denního světla, nebo za použití RGB barevných led můžeme zajímavě měnit barevné scény pokoje. Při dlouhodobé nepřítomnosti v domě, můžeme naopak simulovat rozsvěcováním světel naši přítomnost.

2.2.4 Bezpečnost

Inteligentní dům nám nabízí prvky zabezpečení jako například elektronický vrátný, detekci pohybu v okolí nebo uvnitř domu, požární hlásiče, hlásiče CO₂, hlásiče úniku plynu, detektory rozbití skla, otřesový detektor, nebo dveřní a okenní kontakty. Včetně dalších monitorovacích zařízení jako jsou IP-kamery. Možnost 24hodinového nahrávání a monitorování našeho domu a okolí je jistě zajímavá. Při odchodu z domu můžeme chtít, aby se také stáhly rolety a žaluzie, naopak při vyhlášení poplachu budeme chtít všude rozsvítit a všechny rolety vytáhnout.

2.2.5 Zahrada a garáž

Při pořízení domácí meteostanice můžeme automaticky zalévat květiny a jiné, můžeme měřit vlhkost půdy a zjistit jak dlouho již nepršelo, a podle toho nastavit závlahu. Spouštět naši oblíbenou fontánku na naší přírodní skalce. Pro garáž se nabízí automatické otevírání vrat a měření CO, nebo detekce požáru.

3 Stávající nabídky Inteligentních instalací

3.1 ABB

„ABB je přední světová společnost poskytující technologie pro energetiku a automatizaci, které umožňují energetickým a průmyslovým podnikům zvyšovat výkonnost při současném snížení dopadu jejich činnosti na životní prostředí. ABB má 150 000 zaměstnanců ve více než 100 zemích.“ [3]

Tato firma přichází s komplexním řešením inteligentních budov. Všechny komponenty, které nabízí, pracují na mezinárodním standartu KNX, který je založen na decentralizované sběrníkové technologii.

Každé KNX zařízení má svojí řídicí jednotku, komunikace na sběrnici je multi-master, takže v případě výpadku hlavní řídicí jednotky, mohou jednotlivá zařízení spolu dále komunikovat. Sběrníkové napětí je 24V, v případě že toto napětí klesne pod hranici 20V zařízení se od sběrnice odpojí. Přístup na sběrnici nebo řešení kolizí je provedeno pomocí protokolu CSMA/CA. [4]

K dispozici máme i bezdrátové varianty některých prvků, určeno primárně například na prosklené zdi, nebo podobné prvky. Systém obsahu centrální jednotku, která může být rozšířena síťovým nebo GSM modulem, pro komunikaci pomocí sms zpráv, nebo přes internet. V základním stupni této instalace ovládáme celý systém z dotykových panelů, nebo popř. dálkovým ovládáním. Všechny komponenty jsou pak spojeny na společnou sběrnici. Tento systém umožňuje řídit a ovládat opravdu cokoli. Od základních funkcí jako řízení a regulace osvětlení, ovládání žaluzií, rolet, zásuvek, topení, až po audio-video a EZS.



Obrázek 3 Komponenty od firmy ABB. [22]

Na obrázku 3 vidíme některé z mnoho prvků, které firma ABB nabízí. Jedná se o instalaci pro malé a rodinné domky, kterou nazývají Ego-n. Vlevo vidíme dotykový barevný ovládací panel, kde si můžeme nastavit jednotlivé scény a funkce světla a veškeré další techniky. Dále vidíme menší verzi ovládacího panelu s grafickým displejem, jehož funkčnost je úplně stejná. A nakonec vpravo se nachází pokojový termostat. Všechny tyto komponenty jsou provedením velice líbivé. Jelikož všechny komponenty i např. vypínače komunikují po sběrnici, tak musíme počítat s vyšší cenou jednotlivých komponent.

3.2 InsightHome

Další velmi povedený systém pro kompletní ovládání prvků inteligentní budovy, včetně zabezpečení a zábavy. Jedná se o prvky od firmy AMX. Vše je modulární, takže případné úpravy, nebo dodatečné pořízení není problém, závisí jen na předem dobře navržené kabeláži. Můžeme tu opět ovládat vytápění, ventilaci, klimatizaci, stínění, ohřev teplé vody, zavlažování, zabezpečovací techniku, kamery, osvětlení, provoz spotřebičů, komunikaci, distribuci TV a video signálu, rádio a hudbu a telefony. Na obrázku 4 vidíme základní přístupovou aplikaci.



Obrázek 4 GUI systému InHome. [23]

InHome používá americký zabezpečovací systém HoneywellGalaxy, který je spolu s moduly AMX a zabezpečovacími komponenty EZS propojen ethernetem. Pohybová čidla mohou jednak spustit poplach neoprávněným přístupem do domu, ale také mohou rozsvěcet světla např. na chodbách. Dále jsou použity IP kamery s nočním IR přisvícením, které nahrávají 30denní provoz, obraz z těchto kamer si můžeme prohlédnout např. na televizi. Pomocí VoIP technologie, můžeme komunikovat mezi jednotlivými komunikátory, nebo si zobrazit záběr kamery u vstupní branky, nebo se nám zašle v době nepřítomnosti zpráva, že nám někdo zvoní u dveří spolu s fotkou nebo videem. Jako centrální síťová jednotka je použita dvojice mirrorovaných serverů, která se stará i o bezpečný vzdálený přístup. O automatizaci vytápění se stará systém SIEMENS Syncoliving. Centrály SIEMENS a AMX jsou propojeny pomocí KNX / EIB gateway. Určitě zajímavá možnost je připojení spotřebičů v domácnosti, které podporují

rozhraní IP ready zařízení. Tento systém je velice obsáhlý a umožňuje velké možnosti využití. [5]

3.3 Savant

Opět další velmi rozsáhlý systém pro centrální řízení komplexního vybavení domácnosti. Opět od řízení osvětlení až po multimedia a zabezpečení. Tento systém komunikuje a je možno ho ovládat ze zařízení Apple – iPad, iPhone a iPod. Disponuje pak velmi dobře vybaveným ovladačem, který můžeme vidět na obrázku 5.



Obrázek 5 Dálkový ovladač od firmy Savant. [24]

Dále bychom mohli zmínit, že tento systém využívá také bezdrátové Wifi komunikace. Zvukový systém SmartAudio SSA-4000 nabízí až 12 zónové ozvučení z 6 audiozdrojů. Tento systém je specifikován svojí platformou, zastane opravdu veškeré funkce a možnosti, které bychom si mohli přát. [6]

3.4 Teco a.s.

Tato firma se zabývá průmyslovou automatizací a inteligentními budovami. Jednotlivé komponenty se dají mezi sebou připojovat několika rozhraními, jako je ethernet, RS485, RS232, RS485-Profibus, TLC2, CIB, RS422, M-bus a CAN. Opět nám nabízí široké možnosti, např. na obrázku 6 vidíme řešení multiroom audio/video.



Obrázek 6 Multiroom audio/video system Foxtrot. [25]

Další zajímavá komponenta je rozšiřující modul pro komunikace se standardem OpenTherm.

3.5 Elkoep

Tato firma se řadí mezi přední výrobce komplexních inteligentních instalací. Mají kompletní sběrníkovou a bezdrátovou variantu. Nechybí zabezpečovací technika spolu s detektory. O audio-video techniku se stará Inels-multimedia spolu se systémem LARA. Můžeme přehrávat filmy z datového úložiště, poslouchat internetová rádia nebo zobrazit video z interkomu. Na obrázku 7 vidíme strukturu audio/video systému.



Obrázek 7 Audio/video systém firmy Elkoep.[26]

3.6 Fibaro

Zde se dostáváme k systému, který je bezdrátově koncipován a je zaměřen spíše na ne-multimediální část elektroinstalace. K bezdrátovému přenosu dat využívá technologii Z-Wave, která využívá topologii sítě typu mesh. Nejatraktivnější na tomto systému je možnost instalace systému Fibaro do již realizované elektroinstalace bez jakýkoliv zásahů do omítek a kabelů. Každý modul komunikuje bezdrátově a má svoji napájecí baterii, což nás nutí na druhou stranu přemýšlet, jak často budeme muset tyto baterie měnit a celkově nám to zneprůjemňuje servis a starání se o celkový systém. [8]

Fibaro nabízí centrální jednotku Home center 2, která disponuje procesorem Intel Atom 1,6 Ghz a dále s dalšími komponenty již komunikuje bezdrátově. K této jednotce je nabízeno internetové rozhraní pro ovládání celého inteligentního domu, které můžeme vidět na obrázku 8.



Obrázek 8 GUI systému Fibaro. [27]

Máme možnost ovládat jak klasické světla, tak bílé a barevné led světla, můžeme si nastavit světelné scény a to i barevně. Například pro ovládání světla jen stačí našroubovat objímku s přijímačem a instalace je hotová.

Další komponenta je magnetický kontakt, který disponuje čtyřmi funkcemi: magnetickým kontaktem pro detekci otevření dveří nebo okna, binární vstup (např. pro vypínač), teplotní senzor a modul scén, tzn. zmáčknutím jednoho tlačítka si okamžitě nastavíme uloženou scénu v celém pokoji.

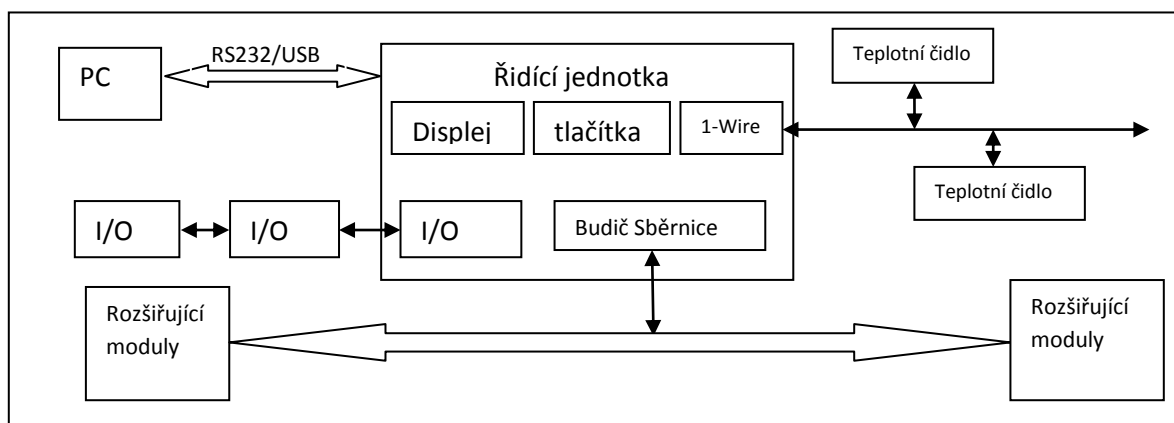
Spínání zásuvek je vyřešeno mezimodulem, který můžeme zasunout přímo do zásuvky. Disponuje tlačítkem pro připojení komunikace do systému. Barevné led podsvícení nám indikuje aktuální spotřebu připojeného spotřebiče.

Detektor zaplavení nám umožňuje jak detekovat přítomnost vody, tak například nadměrný nárůst nebo pokles teploty, dále pak neoprávněné otevření a manipulaci s komponentou. Detektor zaplavení je možné napájet z baterie, nebo ze zdroje 12/24V.

A nakonec zde máme termostatické hlavice pro topení. Jejich instalace je opět velmi jednoduchá, jen vyměníme původní hlavice za nové s akumulátory a můžeme bezdrátově řídit topení. Tento celý systém je velice šikovný v případě, že nechceme nijak zasahovat do původní instalace. Systém má i velmi zdařilé a praktické grafické ovládání.

4 Návrh nového řešení

Zde se již dostávám k návrhu nového řešení pro řídicí jednotku inteligentního domu. Zaměřím se prioritně na ovládání světel, zásuvek, elektrických rolet, ventilů pro topení, pohybová čidla, magnetická čidla apod. Základním konceptem je jednoznačně minimalizovat náklady na tuto řídicí jednotku. Uvažujeme, že tato jednotka bude vhodná do novostaveb, nebo do rozsáhlých rekonstrukcí rodinných domů. Budu se tedy snažit co nejvíce funkčních bloků a komponent integrovat do centrálního rozvaděče. S případným rozšířením dalších modulů po sběrnici. Např. vypínače budou propojeny každý zvlášť přímo do rozvaděče apod. Celý systém pak budeme schopni ovládat pomocí jednotky s grafickým displejem a tlačítky, zobrazit si stav systému, nastavovat režimy a programy. A nakonec bude implementována komunikace s PC pro snadnější a praktičtější ovládání.



Obrázek 9 Základní struktura řídicího systému

Na obrázku 9 vidíme návrh struktury řídicího systému. Řídicí jednotka obsahuje mikroprocesor, který komunikuje pomocí RS232 nebo USB s PC. Jednotka disponuje grafickým displejem a ovládacími tlačítky. Přímou na desce řídicí jednotky jsou k dispozici vstupy a výstupy, které je možno libovolně rozšířit dalšími I/O moduly. Na společnou sběrnici můžeme připojit rozšiřující moduly, např. I/O, teplotní čidlo apod. Lze použít samostatnou sběrnici pro teplotní čidla 1-Wire.

4.1 Ovládání světel

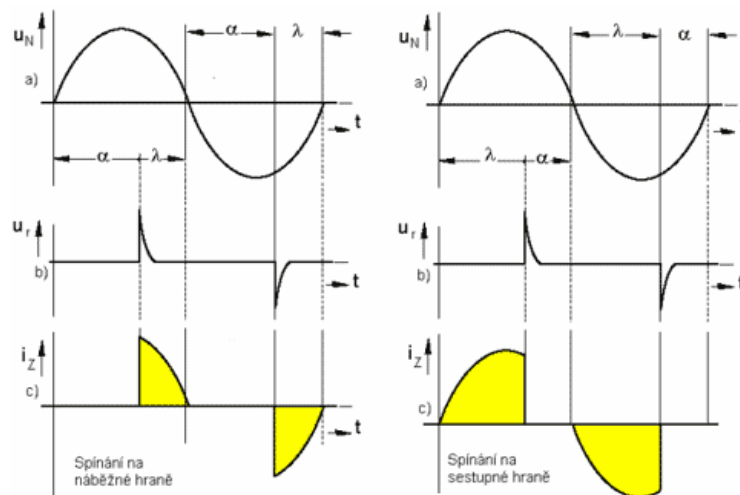
Ovládání světel je dnes s příchodem halogenových, úsporných a led žárovek trochu více problematický. Jedná se tedy o různé typy zátěží, hlavně z hlediska regulace svitu. Ke spínání

nám může posloužit obyčejný optotriak, ale jestli budeme chtít nejenom spínat, ale i regulovat jas těchto žárovek, tak se dostáváme do složitější problematiky.

Existují tři možnosti regulace 230V 50Hz:

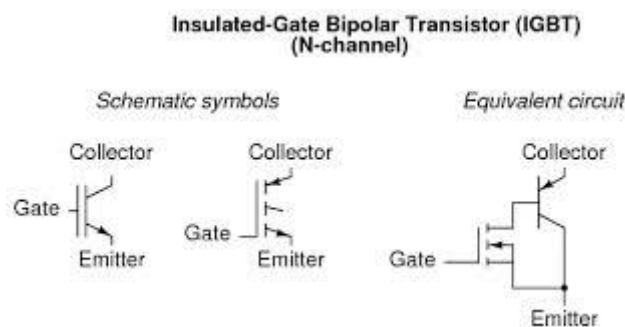
- Změna amplitudy harmonického proudu
- Střídavým propouštěním a zadržováním půlvln
- Změnou tvaru střídavého proudu

Pro osvětlení je vhodná právě metoda změny tvaru střídavého proudu. Realizováno je to pomocí spínání triakem, jedná se o fázové řízení. Jak vidíme na obrázku 10, je možné tento princip aplikovat na náběžnou nebo sestupnou hranu.[9]



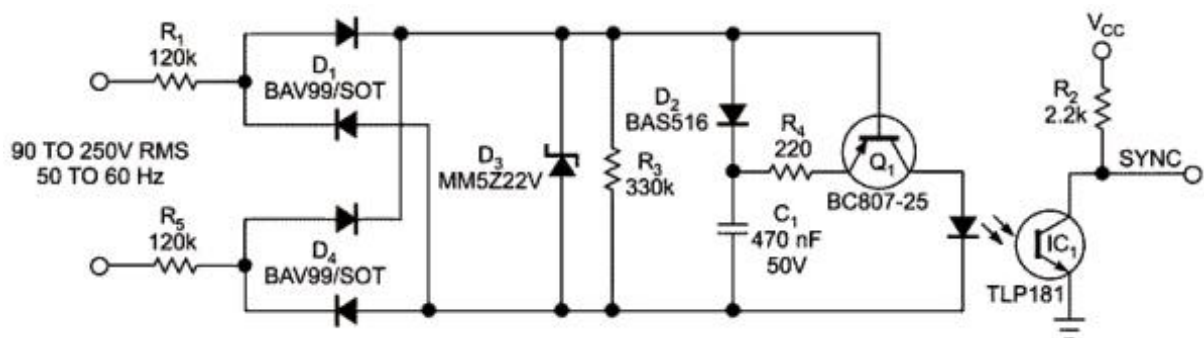
Obrázek 10 Fázové řízení. [28]

Ve starších stmívačích se používají klasické triaky nebo dvojice tyristorů, které spínají na náběžné hraně a rozpínají se průchodem nulou. Tyto součástky můžeme nahradit IGBT tranzistory, které se skládají z dvojice bipolárního a unipolárního tranzistoru obrázek 11.



Obrázek 11 IGBT tranzistor. [29]

Takže se nám zde nabízí možnost spínat zátěž po průchodu nulou a rozepínat v době, kdy je to pro nás žádoucí, tedy spínání na sestupné hraně. Tento princip je složitější, ale umožní nám větší možnosti regulací různých zátěží. Snížíme tak například proudové špičky při spínání kapacitní zátěže, nebo napěťové při rozepínání indukční zátěže, a tím i vyšší vznik vyšších harmonických složek. Důležitá je tedy detekce průchodu napětí nulou. Na internetu je mnoho zapojení na tuto problematiku, na obrázku 12 vidíme jednu možnost zapojení.



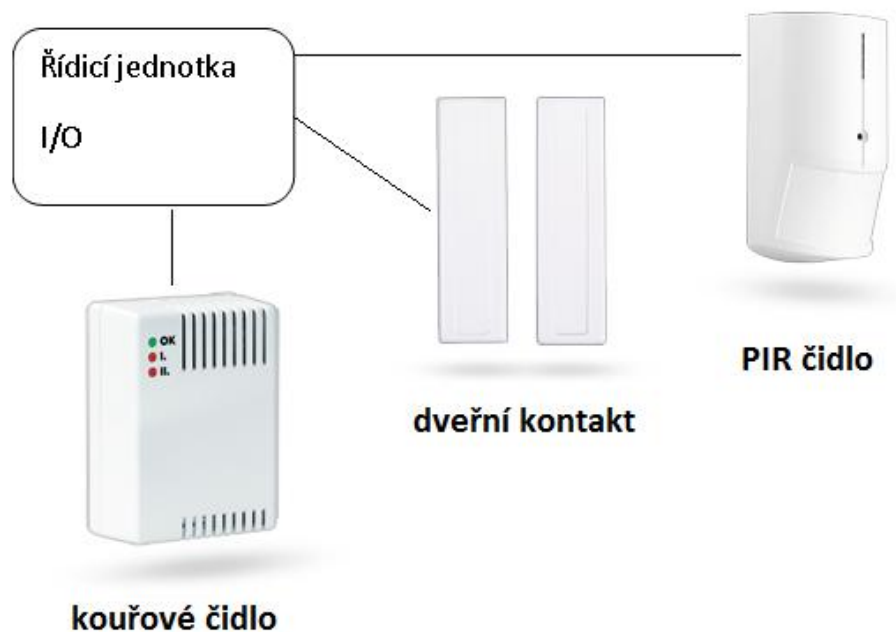
Obrázek 12 Detektor průchodu napětí nulou. [30]

Síťové napětí je zde usměrněno a stabilizováno na napětí zenerovy diody 22V. Ve chvíli, kdy klesne napětí na diodě D2 téměř k nule, se otevře tranzistor a kondenzátor se vybije do optotranzistoru a tím se na malou chvíli otevře.

4.2 Zabezpečovací technika

I zde se dostávám k řešení drátového provedení. Rád bych použil jednak pohybové, kouřové senzory a magnetické kontakty pro detekci otevření dveří nebo oken. Většina prvků tohoto typu je napájena 12V, a mají své pomocné spínací relé. Například detektor od firmy Jablotron JS-25 v sobě integruje pohybové PIR čidlo, detekci rozbití skla a neoprávněnou manipulaci s čidlem. Tyto senzory bych využil prioritně k zabezpečovací technice a dále, když není nutno monitorovat vstup cizích osob do prostoru domu, tak tyto senzory využít například na chodbách pro pohybem spínané osvětlení. Využili bychom tak maximálně funkce těchto komponentů. Dále se dostáváme k detektorům kouře a detektoru zaplavení. Ať se jedná o detektor kouře, nebo použijeme detektor úniku hořlavých plynů a jiné, stále využíváme drátovou verzi těchto zařízení a připojení přes vstupně výstupní porty centrální řídicí jednotky. Podobně pak můžeme využít detektor zaplavení, který se nám hodí například

v koupelně nebo kuchyni. Jedná se o detektor opět od firmy Jablotron Ld-12. Na obrázku 13 vidíme návrh struktury se zabezpečovacími prvky.



Obrázek 13 Struktura zapojení EZS komponentů

4.3 Elektroventily a servoventily pro topení

Dostáváme se k regulaci a ovládání topení. O zónové regulaci jsem se již zmínil. V zásadě se nám nabízejí varianty dvou a více-cestných ventilů. Nás budou zajímat elektrohlavice ke každému topnému tělesu, nebo dvoucestný ventil např. pro celou místnost. Druhá varianta by byla v podobě zónové regulace, kdy bychom využili třícestných ventilů a regulovali teplotu vody v jednotlivých sekcích.

K dispozici jsou různé typy z hlediska funkce otevírání a uzavření ventilu. Například ventil s vratnou pružinou, které je ve variantách „Normally open“ nebo „Normally close“. To znamená, že bez přivedeného napětí je buď otevřený, nebo uzavřený. Například dvoucestný ventil od firmy RegulusSF15-2 M1 má příkon 5-6W dostupný ve variantách 220-110-24 V. Doba otevírání je 10s a doba uzavírání 4s. Firma Jablotron nabízí přímo termohlavici na topné těleso TH-80, které má příkon 1,8W. Je jasné, že zde bude nutná určitá spotřeba elektrické energie pro řízení tohoto typu ventilů.

Druhou variantou jsou servoventily, které zastupují stejnou funkci jako předchozí typ ventilů, spotřebovávají energii jen pro změnu svého stavu. Například servoventil LK525 G od firmy Regulus s napájením 230V a příkonem 7VA. Na obrázku 14 vidíme příklady těchto komponent.[10]



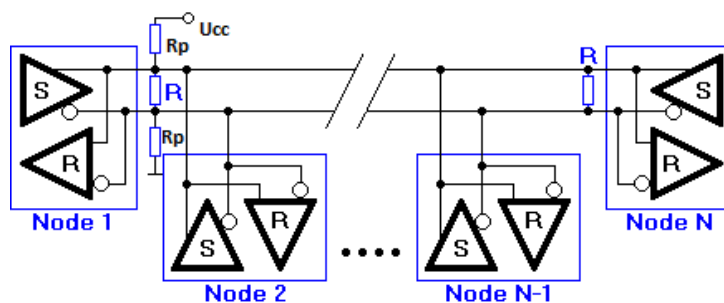
Obrázek 14 Příklady dvoucestných ventilů. [31]

4.4 Datové sběrnice

Zde jsme nuceni řešit jak způsob komunikace, tak např. fyzickou vrstvu. Celou komunikaci bude ovládat Master (v řídicí jednotce), každá rozšiřující komponenta se svým procesorem bude Slave. Z hlediska rychlosti sběrnice nás nic neomezuje, většinou přenášíme velice málo dat, které se navíc ještě velmi pomalu mění. Z toho plyne, že nám postačí datový tok jen několik Kb/s. Jelikož většinu modulů plánujeme v hlavní rozvaděči na řídicí jednotce, tak případný počet rozšiřujících modulů očekáváme velmi malý. Rozšiřujícími moduly máme na mysli například modul vstupů a výstupů, měření teploty apod. Chceme tím docílit, že v případě nefunkčnosti sběrnice z nějakého důvodu, se nesesype ovládání celého domu, ale jen některé rozšiřující bloky.

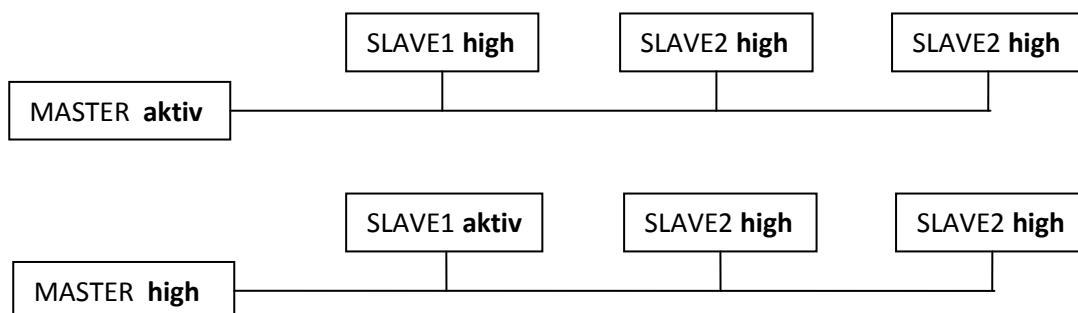
4.4.1 Sběrnice RS485

Jedná se o definici fyzické vrstvy sběrnice. Zapojení pomocí dvou nebo čtyř drátů, s diferenčním napětím. Pomocí dvou drátů můžeme komunikovat polovičním duplexem. Maximální přenosová rychlost je až 10 Mbit/s, maximální délka sběrnice je 1200m. Strukturu vidíme na obrázku 15 včetně nutných zakončovacích odporů R a upínacích pull-up a pull-down rezistorů R_p .



Obrázek 15 Sběrnice RS485. [32]

Všechny stanice se musí nacházet ve stavu vysoké impedance, aby neovlivňovaly komunikaci ostatních, rozhraní RS485 nic neříká o způsobu řízení komunikace ani o způsobu řešení kolizí nebo arbitrází. Pro naši aplikaci by princip byl následovný podle obr. 16.



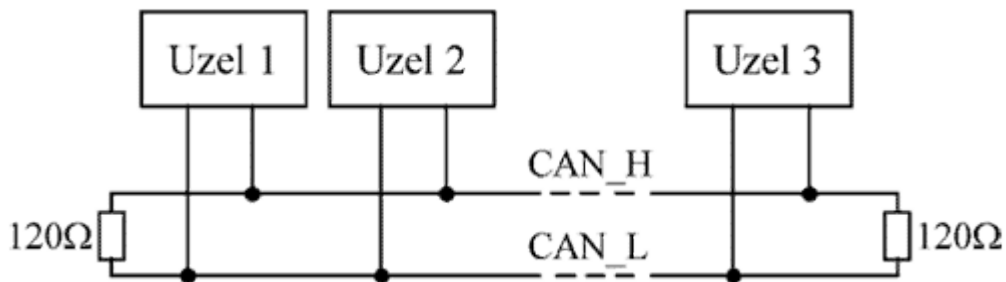
Obrázek 16 Komunikace Master-Slave

Jak vidíme z obrázku 16, komunikaci řídí Master, tzn. master požádá Slave1 o zaslání dat, Master se přepne do přijímacího režimu (**high**) a Slave1 pošle data a Master přijímá. Takhle by Master dostal všechny potřebná data od všech Slave Mcu. Od tohoto řešení si budeme slibovat velice velkou přenosovou rychlost, tak malou časovou náročnost pro Master Mcu.

4.4.2 Sběrnice CAN

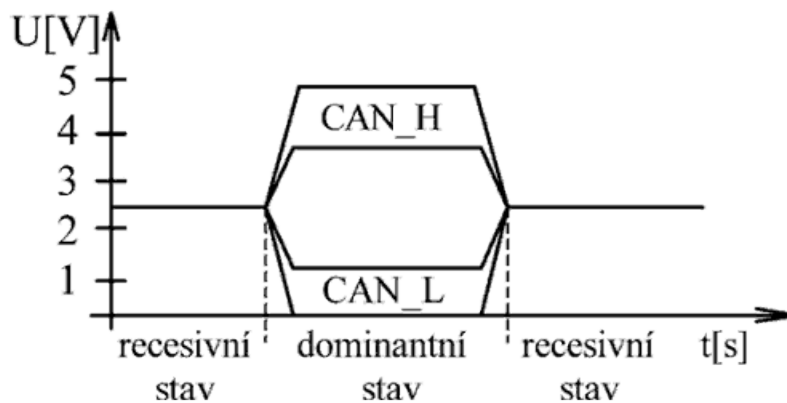
Tato sběrnice je koncipována jako Multi-Master. Jedná se o sériový komunikační protokol s vysokou ochranou proti chybám. Když vznikne chyba v jednom uzlu, tak se komunikace na sběrnici nezhroutí a je schopna komunikovat dál. Vlastní komunikace je založena na posílání zpráv – datová zpráva, žádost o data a chybové zprávy. Vlastní data se vysílají pomocí

recesivních a dominantních bitů, pomocí kterých se řeší kolize na sběrnici nebo arbitráže o počátek vysílání. Na obrázku 17 vidíme blokové schéma CAN.



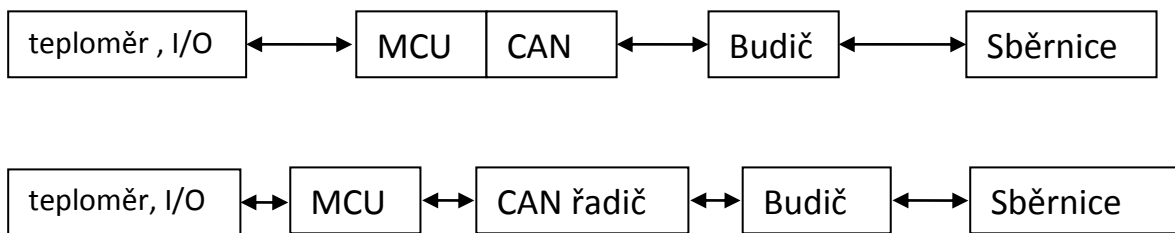
Obrázek 17 CAN sběrnice. [33]

Opět se zde používá diferenční zapojení logických úrovní. Tato sběrnice je schopna přenášet data na malé vzdálenosti cca 40m rychlostí až 1Mbit/s, na větší vzdálenosti cca 100m v řádech desítek 10kbit/s. Na obrázku 18 vidíme jednotlivé logické stavy, které sběrnice může použít.



Obrázek 18 CAN stavy sběrnice. [34]

Z hlediska naší aplikace při realizaci CAN sběrnice bychom čelili problému, buď použít MCU s přímou podporou CAN sběrnice, nebo ještě použít řadič CAN. Tato situace je nastíněna na obrázku 19.

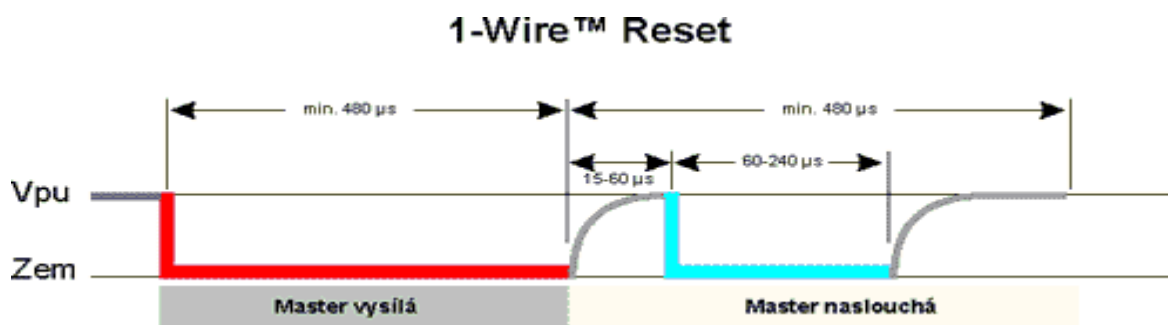


Obrázek 19 Struktura MCU - CAN

Výhodou použití CAN řadičů nám zcela odpadá řízení vlastní sběrnice, řešení kolizí a přístupu na sběrnici. Tato varianta, podobně jako řešení pomocí RS485, by nám samozřejmě opět umožňovala připojit na tuto sběrnici mnoho jiných zařízení. Avšak, jak již bylo zmíněno, jedná se o multi-master komunikaci, takže princip zasílání dat by byl odlišný, například rozšiřující modul by nezasílal periodicky všechny své data, ale poslal by data centrální jednotce, jen když změnil svůj stav.

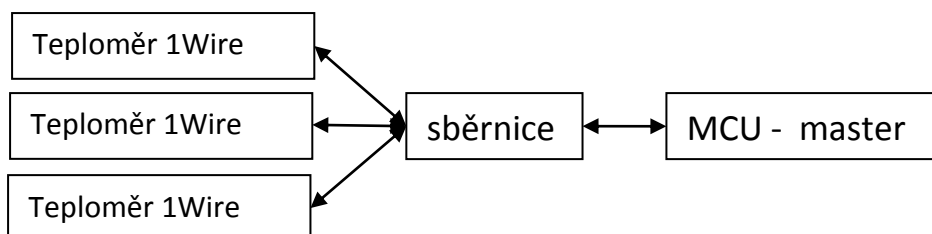
4.4.3 1-Wire

Tato sběrnice je určena pro malou datovou komunikaci. Skládá se z jednoho Master a několik Slave zařízení. Teoretická rychlost této sběrnice je okolo 15Kbit/s, pokud neuvažujeme např. reset sběrnice a adresování zařízení. Každé Slave zařízení má unikátní sériové označení, pomocí kterého se adresuje a probíhá vlastní komunikace. Speciálně tuto sběrnici využívá i teplotní čidlo od firmy Dallas – DS18B20. Výhody jsou například takové, že ke komunikaci po sběrnici nepotřebuje žádný budič, realizace je možná po dvou vodičích, kde je za potřeby přidat budící tranzistor pro proudové posílení (v případě více připojených zařízení), nebo po třech – napájení, zem a datový vodič. Maximální délka mezi masterem a posledním zařízením je okolo 750m. Komunikace používá časové kódování bitových úrovní, bude tedy nutné např. v MCU zakázat dočasně všechna přerušení apod. Jako příklad viz. obrázek 20, který demonstruje reset celé sběrnice.



Obrázek 20 1-Wire reset [38]

U této varianty bychom se dostali k řešení sběrnice jen pro teplotní čidla, jak vidíme na obrázku 21. Použijeme teplotní čidla DS18B20, které můžeme přímo připojit na společnou sběrnici, nepotřebujeme žádný budič ani řadič pro sběrnici. Teplotní čidlo připojíme přímo k MCU. Avšak je zde problém, s přesným časováním této sběrnice, které nám zkomplikuje výpočetní výkon Masteru. A také zamyšlení, že tato sběrnice není nijak chráněna proti chybám přenosu nebo detekcemi chyb, to znamená, že mohou nastat např. nepředvídatelné chyby na dlouhých vedeních.



Obrázek 21 1-Wire

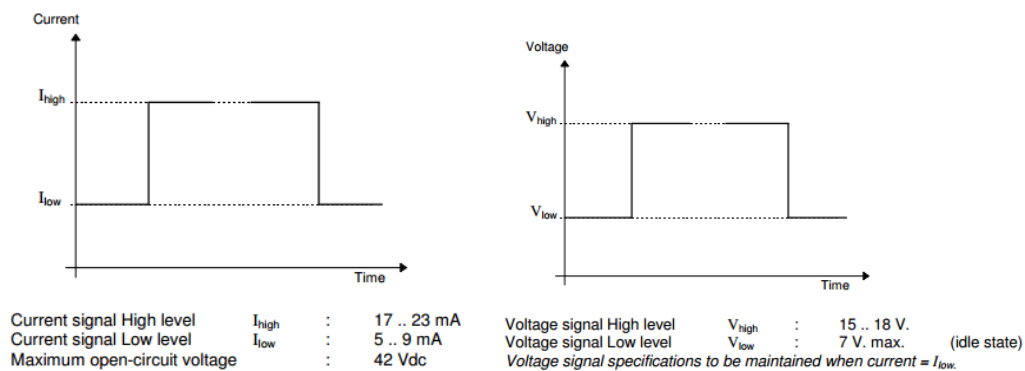
4.4.3.1 Bus driver DS2482-100

Jedná se o sběrnicev řadič 1-Wire řízený I2C rozhraním. Jednalo by se o velice zajímavý prvek, který by nám vyřešil hned několik problémů. Nejdříve bychom připomněli, že v předchozím řešení byla 1-Wire sběrnice připojena přímo k MCU, takže riziko zničení vstupních pinů MCU není vyřešeno, nebo jinak galvanicky upraveno. Tento řadič by nám také zjednodušil ovládání 1-Wire sběrnice, nemuseli bychom vypínat všechna přerušení a přesně časovat jednotlivé příkazy a následně data na 1-Wire sběrnici. Komunikace přes I2C je velmi rychlá a z hlediska MCU se nepotýká s přesným měřením času apod. Další výhodu

spatřujeme v tom, že tento obvod se vyrábí ve verzi DS2482-800, který obsahuje 8krát 1-Wire na sobě nezávislé sběrnice, to by bylo výhodné hlavně pro rozsáhlé budovy a složité větvení celé sběrnice. Dále spatřujeme další výhody, jaké jsou např. řešení napájení teploměrů posíleným proudovým výstupem, řízení strmosti hran kvůli odrazům a EMC a impedančnímu přizpůsobení.

4.4.4 Opentherm

Jedná se o protokol, který využívají některé z kotlů ke komunikaci např. s termostaty nebo řídicími jednotkami. Jedná se o Master-slave komunikaci po nekroucené dvojlince. Maximální délka vedení 50m, elektrický odpor 2x5 Ω. Na obrázku 22 jsou znázorněny logické stavy komunikační sběrnice.[7]



Obrázek 22 OpenTherm logické stavy sběrnice. [35]

4.5 Měření teploty

Z hlediska způsobu měření teploty nejsme nijak zvlášť omezeni, nebo nuceni přizpůsobovat se různým faktorům (např. rychlost a intervaly měření...). Jelikož se jedná o měření pokojové teploty, tak nepotřebujeme rychlé měření – teplota se mění v pokoji v řádu minut. Přesnost měření by byla žádoucí na 1°C a přesněji. Rozsah měření si stanovíme mezi -20°C až 100°C pro potřebu měření jak venkovní teploty tak např. pro měření teploty topné vody.

4.5.1 Čidlo Pt100

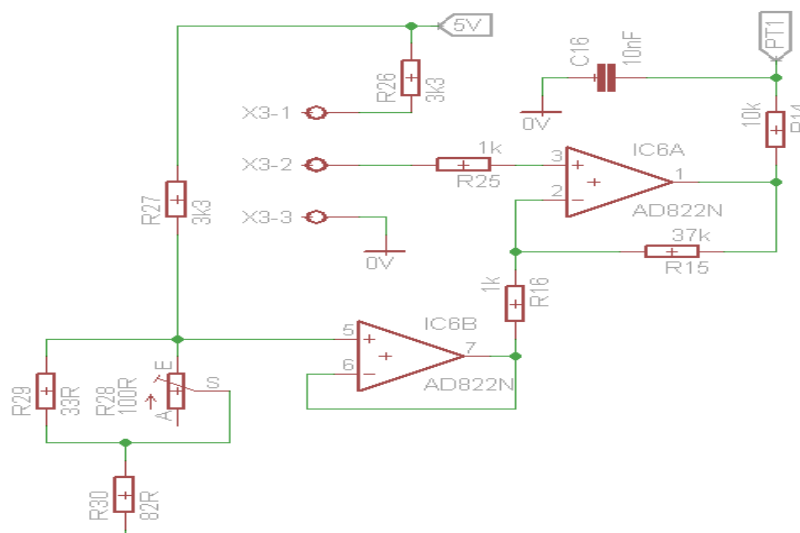
Nejprve se dostáváme na analogová teplotní čidla, konkrétně platinové odporové čidlo PT100.

°C	Ω
0.00	100.00
0.00	100.00
10.00	103.90
20.00	107.79
30.00	111.67
40.00	115.54
50.00	119.40
60.00	123.24
70.00	127.08
80.00	130.90
90.00	134.71
100.00	138.51



Obrázek 23 Teplotní čidlo Pt100 [39]

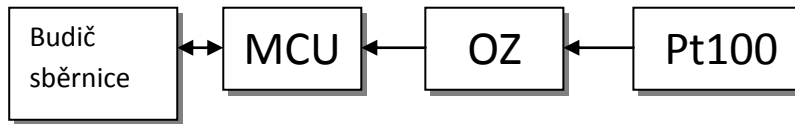
Na obrázku 23 vidíme závislost odporu na teplotě, při použití těchto čidel, bychom museli implementovat ke každému čidlu ještě MCU s A/D převodníkem a zapojení, které bude převádět změnu hodnoty elektrického odporu na napětí. Na obrázku 24 vidíme návrh zapojení pro připojení čidla Pt100.



Obrázek 24 Zapojení pro Pt100

Teplotní čidlo bychom připojili mezi spojené svorky X3-1 X3-2 a na X3-3. Trimrem R28 bychom mohli nastavit offset pro OZ IC6A, který je zapojený jako neinvertující zesilovač. Tento offset nám nastaví nejnižší teplotu, kterou bychom chtěli měřit. Odpor R16 a R15 nám nastaví zesílení, které nám ovlivní rozsah měřených hodnot. MCU připojíme na vývod

PT1. U tohoto řešení bychom museli pro každé čidlo provést dvoubodovou kalibraci, protože rezistory mají toleranci hodnoty, a tím pádem výsledné zapojení i trochu jiné zesílení a nastavený offset.

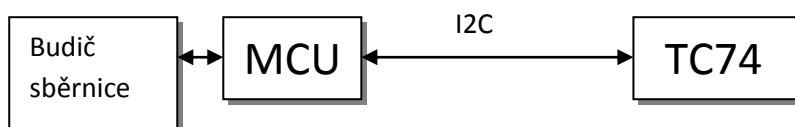


Obrázek 25 Blokové zapojení čidla Pt100

Na obrázku 25 vidíme blokové schéma pro celý modul připojitelný na sběrnici. Hlavní nevýhodou je složitá kalibrace každého odporového čidla.

4.5.2 Digitální čidlo TC74

Jedná o teplotní čidlo od firmy Microchip s digitální rozhraním I2C. Maximální přesnost výrobce udává +/-2°C v rozsahu -25°C až 100°C. Samozřejmě existují přesnější, ale to přináší i vyšší cenu. Oproti prvnímu řešení tu zde máme zjednodušení, nemusíme každé čidlo kalibrovat, ani konstruovat pomocné vyhodnocovací obvody. Blokové zapojení by tedy vypadalo podle obrázku 26.



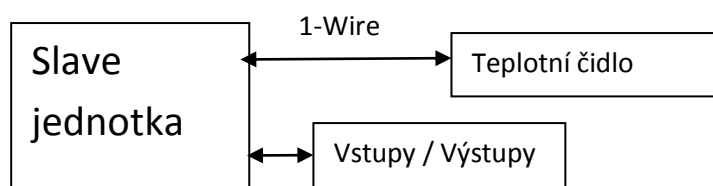
Obrázek 26 Blokové zapojení čidla TC74

4.5.3 DS18B20

Jedná se digitální čidlo teploty od firmy Dallas, disponuje měřícím rozsahem od -55°C do 125°C. Udávaná přesnost se pohybuje až na +/- 0,5°C v rozsahu -10°C až 85°C. Rozsah vnitřního převodníku je od 9-12 bitů, můžeme tak docílit rozlišení až na 0.0625°C. Doba převodu hodnoty teploty při 12-bitovém rozlišení je 750ms, což se nám může zdát celkem dlouhá doba, ale počítáme s tím, že pošleme příkaz pro převod a pak např. za 1 sekundu budeme číst hodnotu z čidla.

4.6 Rozšiřující moduly - Slave

Tyto moduly budou komunikovat s hlavní řídicí jednotkou po sběrnici a budou sloužit k rozšíření periférií, jako jsou vstupy, výstupy a čidla teploty. Anebo budou mít úplně jiný charakter jako například modul pro ovládání kotle, modul pro řízení solárního ohřevu užitkové vody nebo jenom informační displej.



Obrázek 27 Blokové schéma rozšiřující jednotky

Slave jednotka bude posílat informace o teplotě a zasílat data vstupů a výstupů. Podle obrázku 27 vidíme, že teplotní čidlo komunikuje pomocí 1-Wire sběrnice s čidlem, výhoda je taková, že teplotní čidlo nemusí být v bezprostřední blízkosti procesoru, ale můžeme ho natáhnout např. několik metrů k topení. Dále zde máme část vstupů a výstupů.

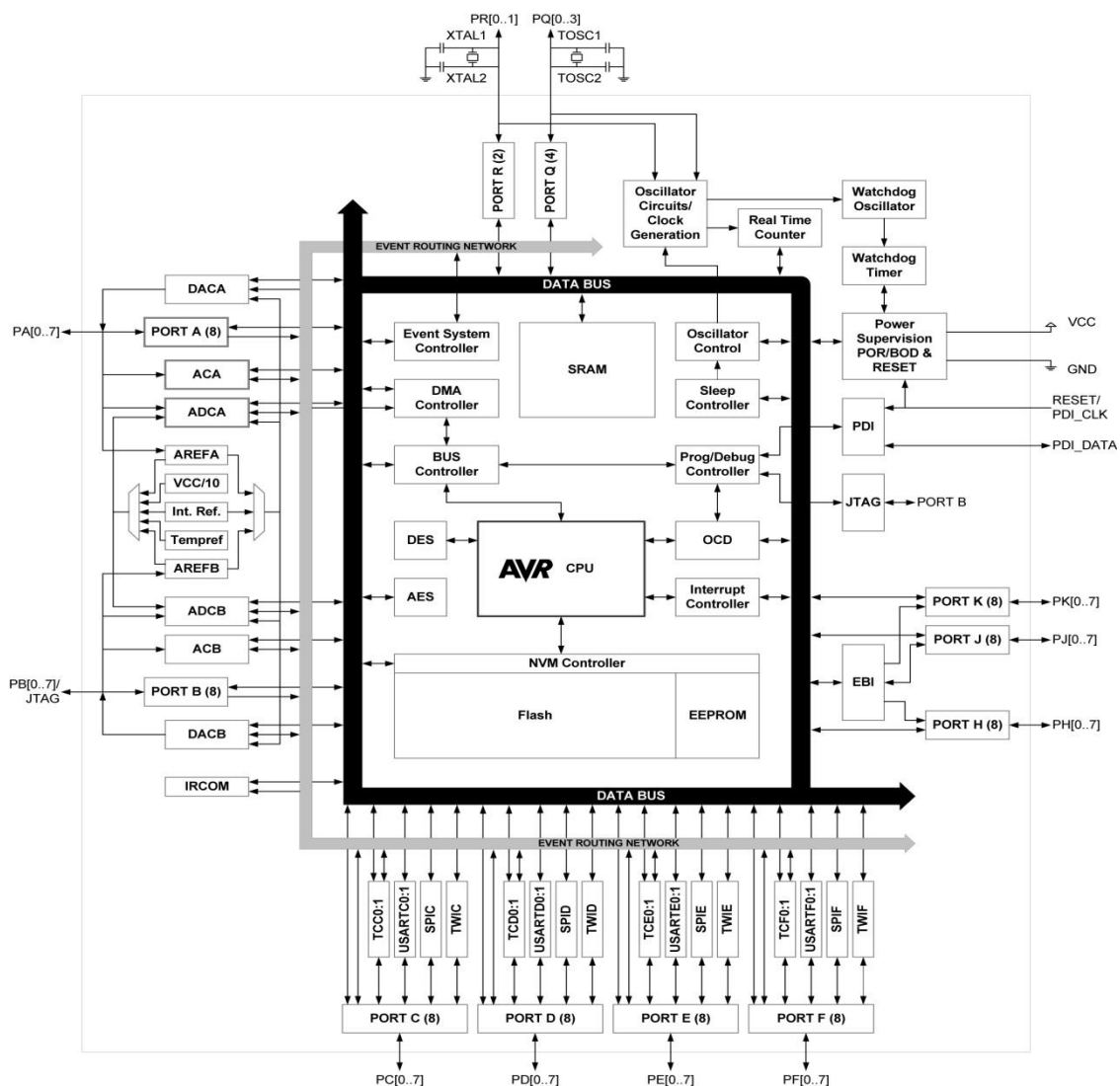
5 Hardwarový návrh

5.1 Procesor

Jelikož již nějakou tu dobu pracuji s procesory od firmy Atmel, tak jsem se nijak zvlášť nerozhodoval, jaký typ procesoru si vybrat. Jedná se o procesory řady Mega a Xmega.

5.1.1 Procesory Atmel ATXmega

Jedná se o 8bitové procesory, které pracují od napájecího napětí 1,6V až 3,6V. Zajímavá je použitá druhá generace technologie PicoPower, která velice ovlivňuje spotřebu procesoru a tím je spojeno jejich široké využití v bateriově napájených aplikacích. Mimo jiné tyto procesory mají implementované bohaté periferie, které si ještě popíšeme. Na obrázku 28 vidíme blokové schéma procesoru ATXmega řady A1.



Obrázek 28 Blokové schéma ATXmega. [11]

Ze zdrojů hodinového taktu můžeme využít jednak nízkopříkonový interní oscilátor 32kHz, které slouží k řízení Watchdogu a obvodů hlídající výpadky napájení. Dále pak interní kalibrovaný RC oscilátor 32768Hz, který může být použit jako zdroj systémového taktu nebo taktu pro RTC obvody. K dispozici máme dále dva RC oscilátory 2MHz a 32MHz, a samozřejmě vstup pro externí krystal s frekvencí až 16Mhz a externí hodinový signál.

Procesor ATXmegaA1 obsahuje hned 8 čítačů/časovačů, každý z časovačů podle typu obsahuje 2 nebo 4 jednotky compare/capture. Časovače s označením 0 disponují systémem AWeX, což je systém pro nastavení času přechodu z logických úrovní na výstupu. Další šikovná věc je například využití čítačů s přímou podporou kvadraturního kodéru.

Event systém je princip řízení událostí, které obcházejí nutnost být řízeny jádrem procesoru. Docílíme tím, že mnoho funkcí můžeme realizovat mimo hlavní výpočetní výkon procesoru, a

tím odlehčit a uvolnit výkon pro další aplikace. Celkem zde máme 8 kanálů, které můžeme propojovat mezi časovači, čítači, a/d převodníky a porty.

Procesor obsahuje další prvky, jako: přímý přístup do paměti DMA, obvody reálného času RTC, TWI - TwoWire Interface, SPI – SerialPeripheral Interface, USART – Universal SynchronousAsynchronousReceiverTranceiver, případně přímá podpora USB. [11]

Zajímavé jsou zde 12-bitové A/D převodníky, které zvládají až 2 miliony převodů za sekundu a také 12-bitové D/A převodníky s rychlostí 1 milionu převodů za sekundu. Další vybavení jsou analogové komparátory, u kterých si můžeme téměř libovolně nastavit zesílení vstupních kanálů.

5.1.2 Procesory Atmel ATMega

Opět se jedná o 8-bitové procesory, akorát o straší řadu. Nemáme zde až tolik bohaté periferie. Rád bych zmínil o základním modelu ATmega8. Podporuje například Atmel QTouch® library, která slouží pro pohodlný návrh kapacitních tlačítek a klávesnic. Obsahuje dva 8-bitové a jeden 16-bitový čítač/časovač, třemi PWM kanály, odděleným RTC obvodem, osmi kanálovým A/D převodníkem s 10-bitovým rozlišením, Two-wireSerialrozhraním, USART a SPI rozhraním, napájecí napětí 2.7 - 5.5V při taktu 0 - 16MHz. Jedná se o velice dostupný procesor a určitě se nám hodí na jednodušší aplikace a zapojení.

5.1.3 Procesory PIC

S tímto druhem procesorů bohužel nemám žádné zkušenosti, vybral jsem si na porovnání řadu PIC18F. Tyto procesory operují s napájecím napětím 2V - 3,6V s možností taktování procesoru až 40 Mhz, lze použít i další interní oscilátory 32KHz. Zatížitelnost každého výstupu je 25mA. Procesor obsahuje standardní rozhraní: SPI, I2C, USART(s podporou RS-485, RS-232). Dále 13-kanalové A/D převodníky s rozlišením 10-bitů. K dispozici jsou 4 výstupy PWM kanálů.

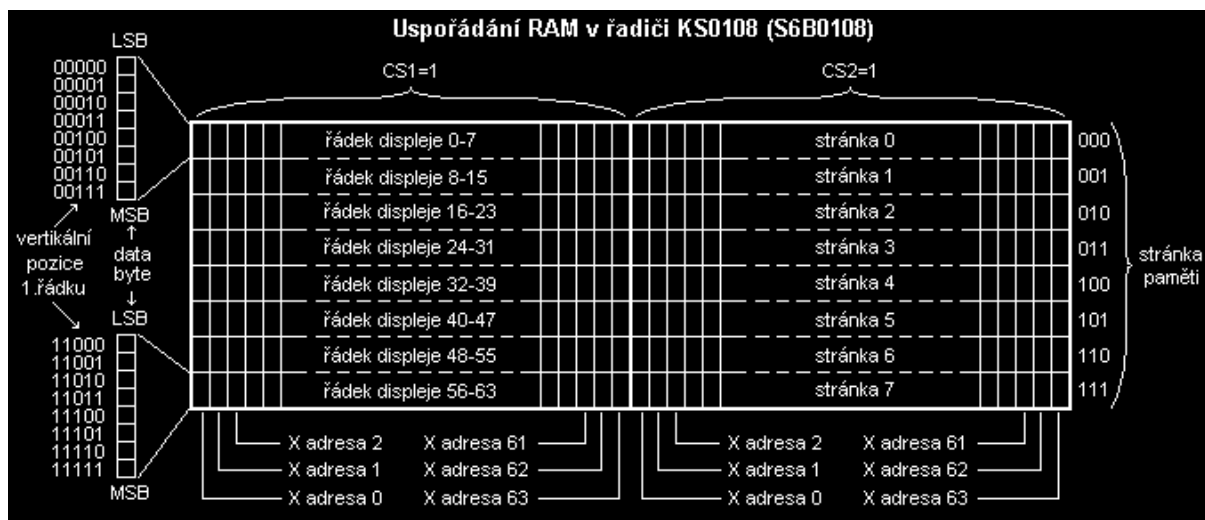
5.2 Grafický displej 128x64

Tyto displeje používají nejčastěji řadiče KS0108. Jedná se monochromatický displej s rozlišením 128x64 pixelů. Vidíme ho na obrázku 29.



Obrázek 29 Gafický displej. [36]

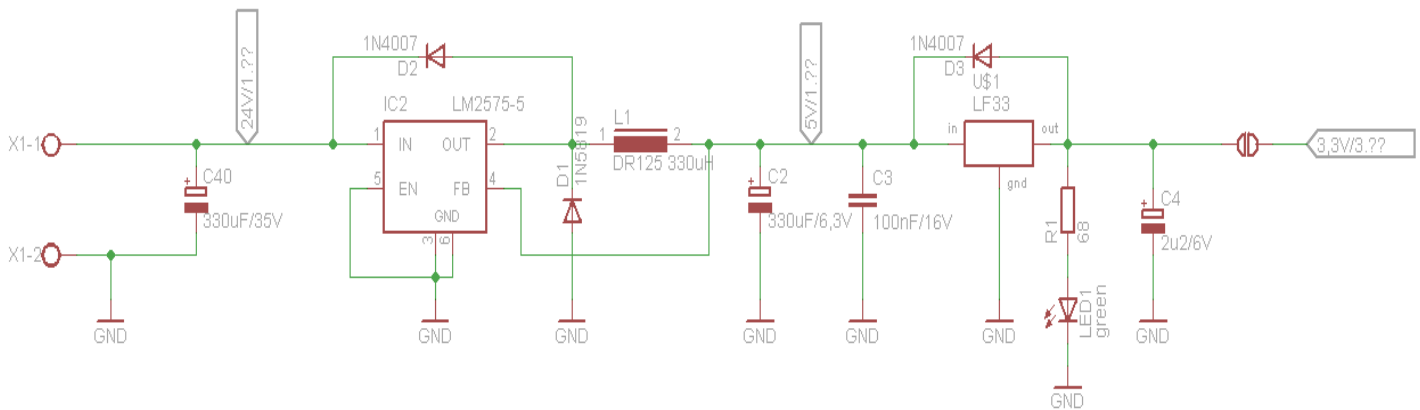
Z hlediska struktury jednotlivých pixelů vidíme, že displej je rozdělený na dvě části, každá má svůj řadič - obrázek 30. Pomocí datových a ovládacích pinů můžeme například číst nebo zapisovat (pin RW), vybírat levou nebo pravou polovinu řadiče (pin CS1 a CS2), dále určovat zda jde o instrukci nebo data (pin RS), to vše potvrzeno hodinovým signálem pin E. Jak vidíme z obrázku, obě poloviny displeje se adresují pozicí stránky a pozicí X. Pokaždé zapisujeme nebo čteme všech 8bitů najednou. Po každém čtení nebo zápisu se automaticky inkrementuje pozice X, když přeteče přes pozici 63, tak se vynuluje a začíná opět od 0. Jsou definovány struktury příkazů např. pro zapnutí/vypnutí displeje, nastavení adresy stránky a pozice x. Musíme také respektovat rychlost a dobu signálů pro řadič, to je dáno přesnými hodnotami ve výrobním listu.



Obrázek 30 Struktura displeje. [37]

5.3 Napájecí zdroj

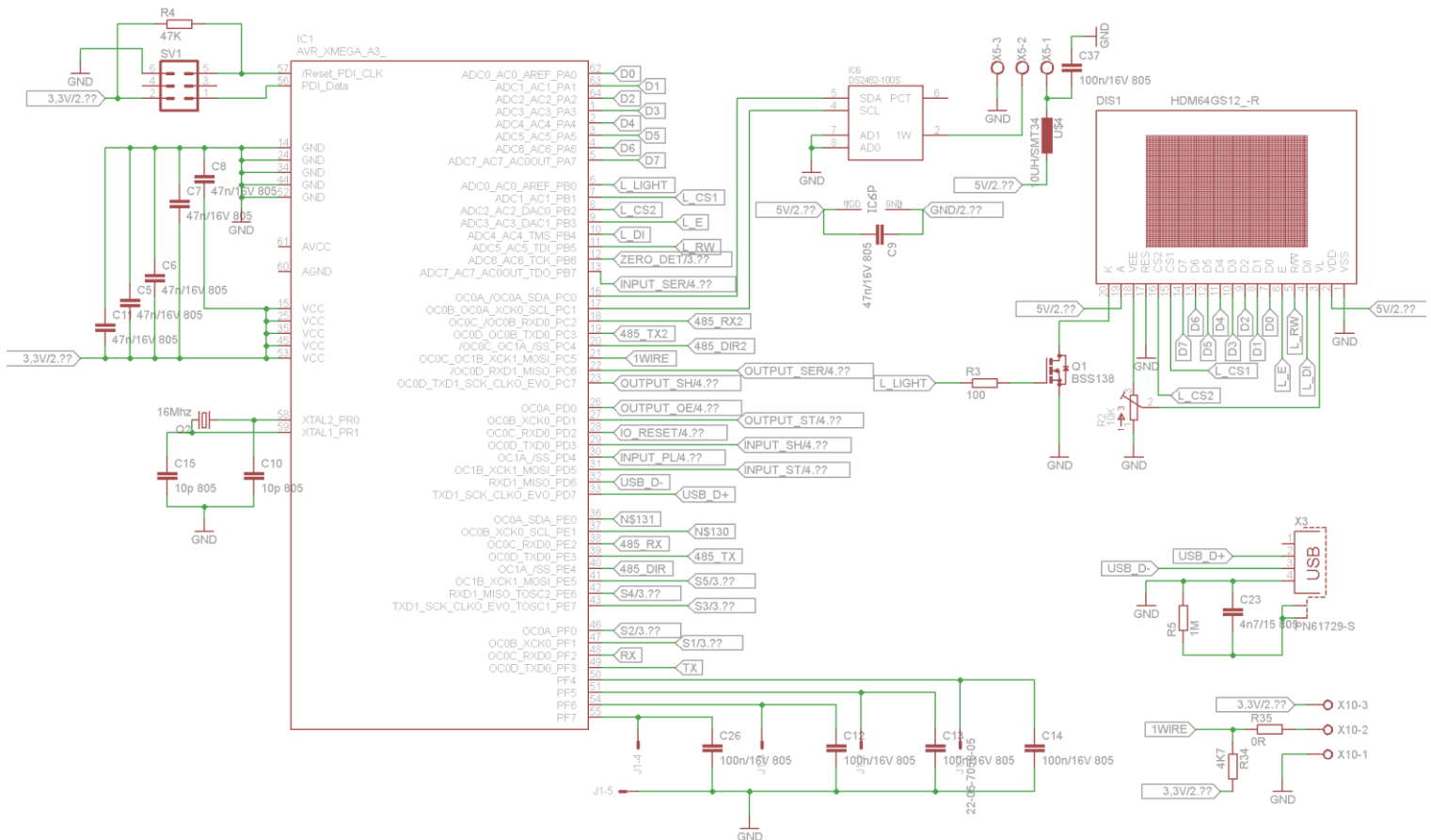
K napájení systému budu používat 24V, je tedy nutné použít spínané zdroje pro snížení napětí vhodného pro mikroprocesor. Zapojení vidíme na obrázku 31.



Obrázek 31 Zapojení zdrojů řídicí jednotky

Použil jsem spínaný zdroj LM2575-5, který je zapojen v režimu Step-down. Pracuje na spínací frekvenci 52kHz s výstupním napětím 5V a proudem 1A. Je přidána blokovácí dioda D2. Napájecí napětí 5V využijeme pro podsvícení displeje nebo napájení budičů sběrnic. Následuje lineární stabilizátor LF33 na napětí 3,3V, který má maximální výstupní proud 0,5A. [12][13]

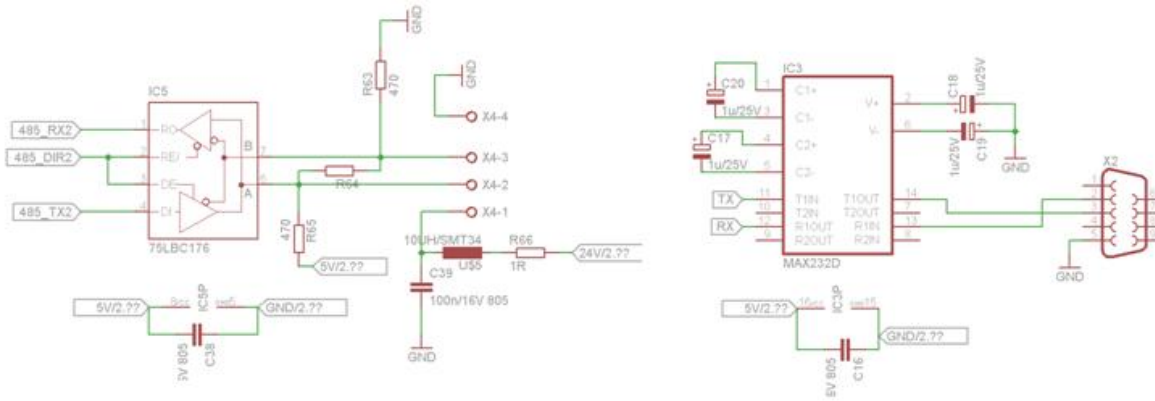
5.4 Zapojení procesoru



Obrázek 32 Zapojení procesoru ATxmega256A3

Použil jsem procesor ATxmega256A3, který má 256Kb programovatelné flash paměti. Ke každému páru napájecích pinů procesoru jsem dal blokovací kondenzátory 47n. Je použit externí krystal 16MHz spolu s dvojicí doporučených kondenzátorů C15 a C10. Jelikož tento procesor hardwarově podporuje USB zařízení, tak je připojeno přímo k procesoru, spolu s doporučeným RC filtrem připojeným na stínění USB konektoru. Pro spínání podsvícení grafického displeje je použit unipolární tranzistor BSS138. Trimrem R2 nastavujeme kontrast displeje, který je bohužel také teplotně závislý. Dále jsem použil budič sběrnice 1-Wire s komunikačním rozhraním I2CDS2482-100. Konektor J1 je použit pro připojení čtveřice tlačítek, každé má svůj blokovací kondenzátor kvůli kompenzaci zákmitů, které vznikají při sepnutí tlačítek.

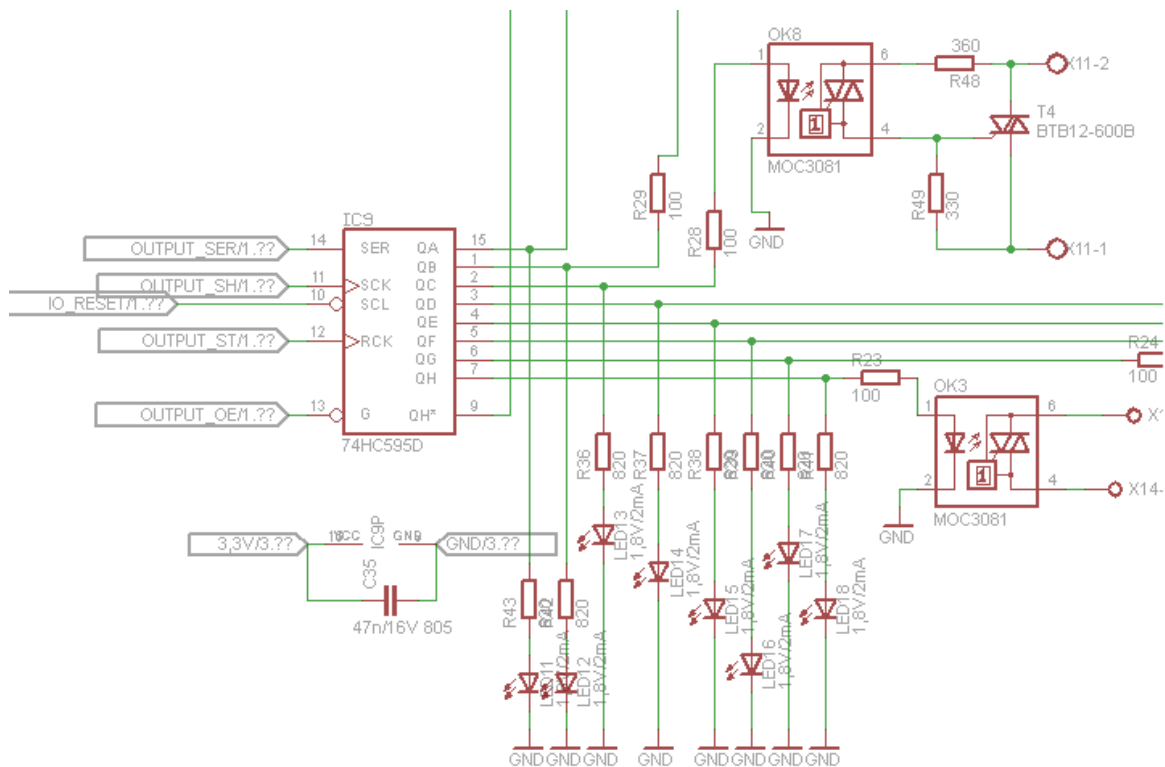
Na obrázku 33 vidíme zapojení budiče pro sběrnici RS485 a převodník na rozhraní RS232. U sběrnice RS485 je zapojen zakončovací odpor 120Ω, dále pak upínací odpory 470Ω pro počáteční definici logického stavu na sběrnici. Oba tyto obvody jsou napájeny 5V.



Obrázek 33 Zapojení RS485 a RS232

5.5 Vstup výstupy

Veškeré vstupy a výstupy na desce hlavní řídicí jednotky jsou řešeny pomocí 8-bitových posuvných registrů. Tento přístup je výhodný pro neomezené rozšiřování počtu vstupů a výstupů pomocí zřetězení sériových vstupů a výstupů těchto posuvných registrů. Jak vidíme na obrázku 34, využil jsem posuvný registr 74HC595, který ovládá optotriaky MOC3081. Tento optotriak má v sobě funkci spínání v nule, takže tím ošetříme nevhodné sepnutí síťového napětí například v maximu periody, a tím omezíme vznik rušivých složek.

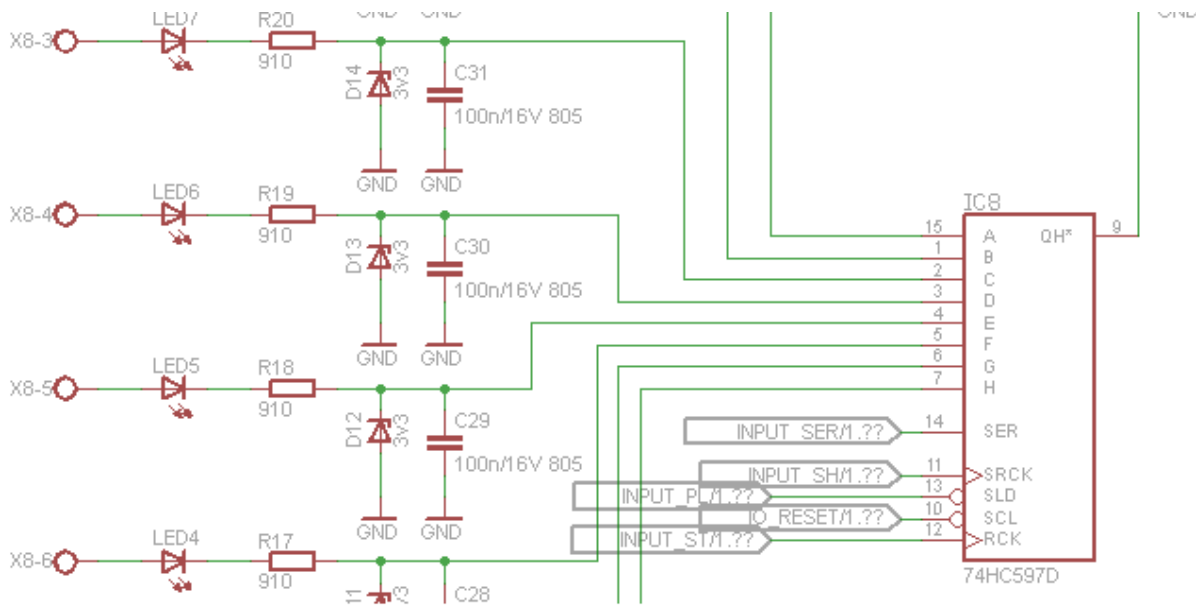


Obrázek 34 Zapojení výstupů 230V

Optotriak MOC3081 je schopen spínat napětí 230V a maximální proud 1A, to se nám hodí pro ovládání ventilů a nízko příkonových komponent. Druhá varianta výstupu je posílána o

přidaný triak BTB12, který je schopen spínat až 12A. To využijeme pro spínání světel nebo zásuvek. Každý výstup je indikován led diodou s malým proudovým příkonem 2mA.

Zapojení vstupů vidíme na obrázku 34. Při návrhu jsem počítal s tím, že tyto vstupy budou například od vypínačů nebo požárních čidel apod. Zde je použit posuvný registr 74HC957.

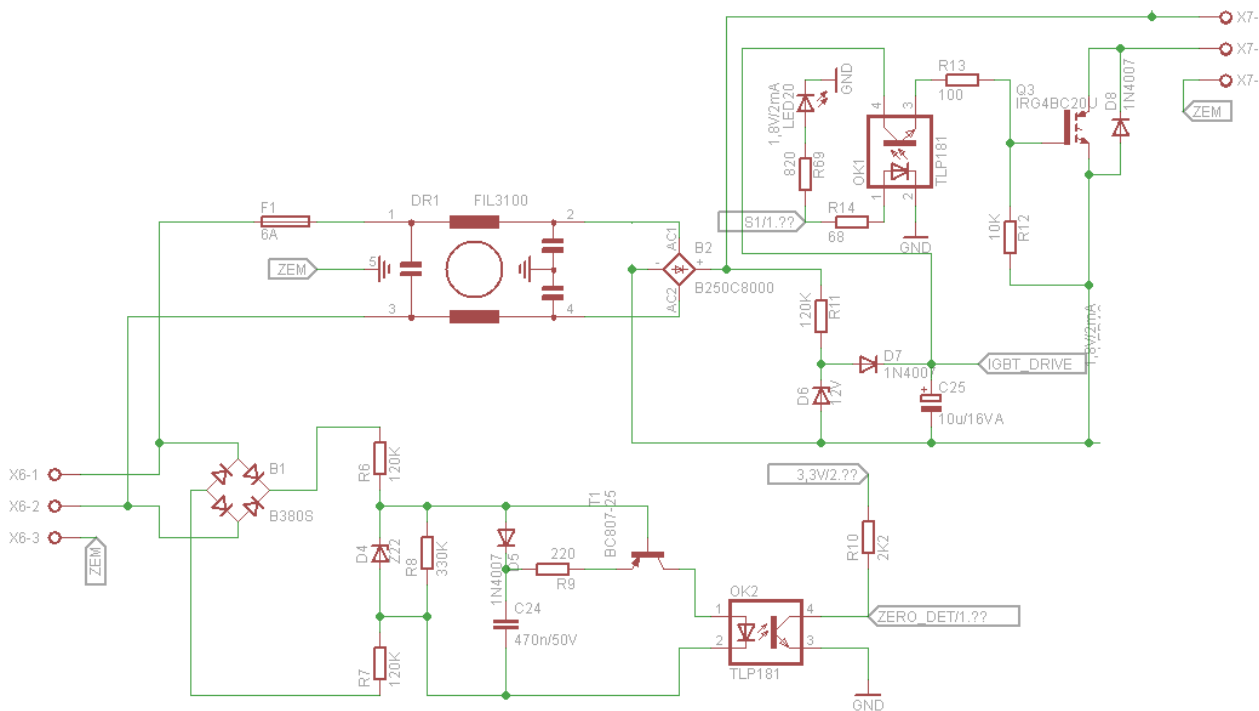


Obrázek 35 Zapojení vstupů 24V

K indikaci stavu jsou opět použity led diody. Zenerova dioda na napětí 3,3V snižuje napětí logické 1 pro posuvný registr. Sériový vstup QH' je pak připraven na konektor pro připojení dalších posuvných registrů. Obnáší to jen úprava programu obsluhující tyto vstupy nebo výstupy.

5.6 Výstupy pro regulaci osvětlení

Princip spínání jsem již popsal v předchozích kapitolách. Na obrázku 36 vidíme zapojení jednoho výstupu pro regulovatelné světlo. První část je s diodovým můstkem B1, posléze zapojení detekce průchodu nulou, které jsem již vysvětloval. Optotranzistor OK2 pak vysílá procesoru detekci průchodu nulou. Dále zde máme druhý výkonový usměrňovací můstek B2, před kterým je pojistka na 6A a síťový odrušovací filtr, který nám bude filtrovat vyšší harmonické složky proudu, které by nám mohly rušit celý síťový rozvod 230V.

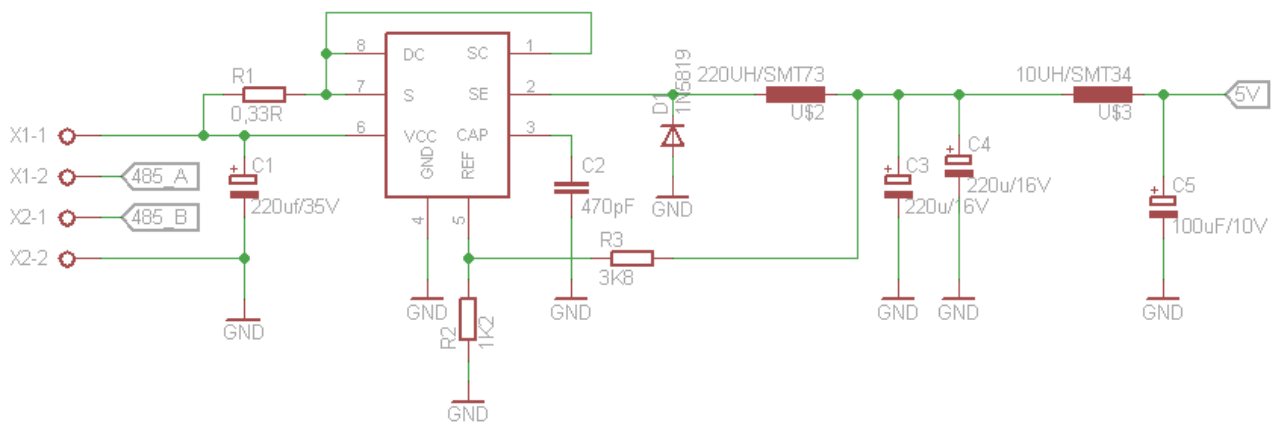


Obrázek 36 Zapojení modulu regulace osvětlení

Jak již bylo zmíněno, použil jsem výkonový IGBT tranzistor pro spínání zátěže. K ovládání jeho řídicí elektrody slouží kombinace diody D6 D7 a kondenzátoru C25, na kterém máme přibližně 11V. Toto napětí pomocí optotranzistoru OK1 přivádíme právě na řídicí hradlo tranzistoru Q3. Zátěž tedy připojíme mezi horní dvě svorky X7.

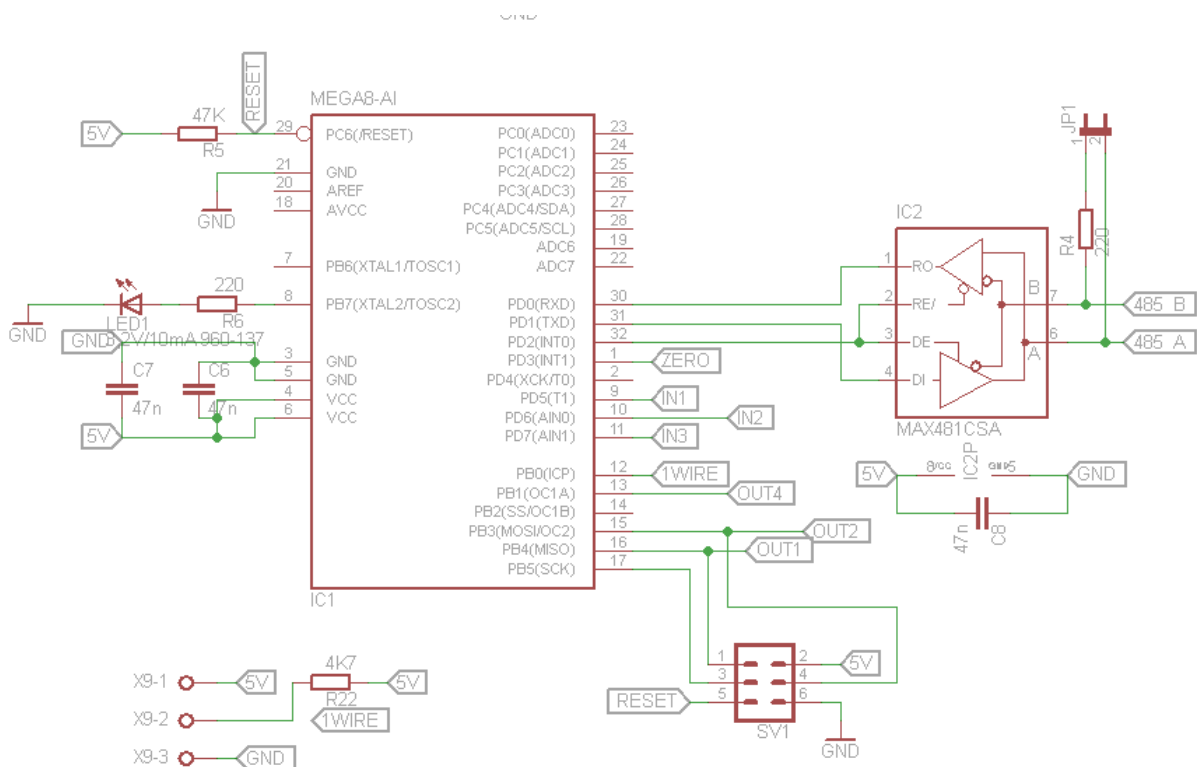
5.7 Slave - rozšiřující jednotka

Tyto jednotky jsou připojeny na sběrnici RS485, dále pak i na napájení 24V. Jsem si vědom, že nejsou galvanicky oddělené, tudíž jsou zde jistá rizika například nadpětí nebo zkratu s jiným napájením. V nejhorším případě by však odešel budič sběrnice a ne celá procesorová jednotka. Napájecí napětí 24V volím také proto, aby po vedení, které bude dlouhé desítky metrů, procházel co nejmenší proud, a tím pádem byly co nejmenší úbytky napětí na vedení. Na obrázku 37 vidíme zapojení spínaného zdroje MC33063A. Pomocí rezistorů R1 a R3 nastavíme výstupní napětí step-down měniče na 5V. Výpočet je podle datasheetu. Zařadil jsem zde také LC vyhlazující filtr.



Obrázek 37 Zdroj MC33063 pro rozšiřující modul

A nakonec zapojení procesoru a budiče sběrnice vidíme na obrázku 38. Použil jsem procesor ATmega8-16au. Svorkovnice X9 složí k připojení teplotního čidla. Přidal jsem také jumper JP1 pro libovolné zapojení/odpojení zakončovacího odporu pro RS485.

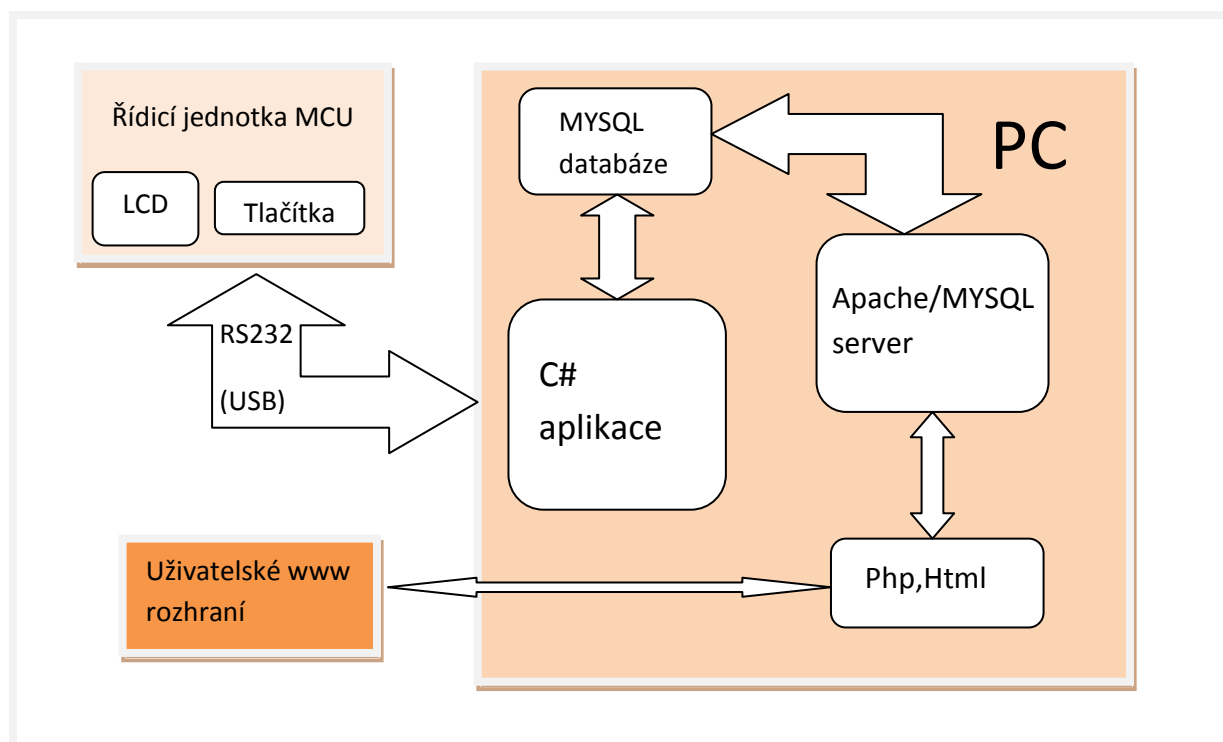


Obrázek 38 Zapojení ATMegy a budiče sběrnice

Tato rozšiřující jednotka disponuje třemi vstupy 24V, jeden výstup 230V/8A pro regulaci osvětlení, výstup 230V/10A, dva výstupy 230V/1A a libovolným počtem teplotních čidel.

6 Softwarový návrh

Celý systém je primárně koncipován tak, abychom byli schopni ovládat řídicí jednotku pomocí tlačítek a grafického displeje. Můžeme zde nastavovat intenzity jednotlivých světel včetně světelných scén, nastavit teplotu v pokoji, zapínat a vypínat vlastní regulaci topení, nastavit zóny zabezpečovacích prostor. Dále pak nechybí volba režimů, jako jsou třeba: normální, večer, noc a nepřítomnost. Tyto scény mají za úkol jak nastavovat a zapínat světla, tak i ovládat zabezpečovací techniku. K tomu se pak dále ještě dostanu. Dále pak zde máme možnost veškeré tyto věci nastavovat pomocí webového rozhraní např. pomocí PC nebo telefonu, a to jak v rámci domácí Wifi nebo Lan sítě, tak i přístupu zvenčí. K tomu bude za potřeby domácího serveru. Na obrázku 39 vidíme celou strukturu komunikace řídicí jednotky a PC.



Obrázek 39 Blokové schéma komunikace jednotky a uživatelem

Veškeré záležitosti ohledně řízení má na starost hlavní řídicí jednotka, která vysílá a přijímá různé parametry, data o stavech vstupů výstupů, nastaveném aktuálním programu, konstant

a jiných. Řídicí jednotka komunikuje s PC pomocí sériového rozhraní RS232. Připravena je i varianta pro připojení pomocí USB. Pomocí aplikace, kterou jsem napsal v jazyce C#, implementuji data poslaná/přijata pomocí RS232 do MySQL databáze. V této databázi jsou dvě tabulky – pro přijatá a odeslaná data. Pomocí MySQL a Apache serveru přistupuji k datům v databázi a zobrazuji je pomocí PHP a HTML jazyka ve webovém prohlížeči. Takže v rámci sítě, můžeme pomocí stolního počítače nebo tabletu, telefonu s přístupem na bezdrátovou síť ovládat celý systém. Tímto si slibuji velikou pohodlnost a praktičnost tohoto ovládání.

6.1 Řídicí jednotka MCU ATXmega256A3

Jak jsem již zmínil, tato jednotka obsahuje grafický displej, ovládací tlačítka, rozšiřitelné porty vstupů a výstupů, rozhraní USB, RS232, RS485 a 1-Wire rozhraní. Programové vybavení obsluhuje jednotlivé periférie, můžeme nastavit uživatelské konstanty nebo nastavovat aktuální režim řízení. Dále popíši jednotlivé bloky a programy. Všechno jsem programoval pomocí jazyka C++, v prostředí AVR Atmel Studia 6, který využívá Gcc překladač.

6.1.1 Grafický displej

6.1.1.1 Knihovna LCDLIB.h

Vytvořil jsem si vlastní knihovnu pro obsluhu grafického displeje 128x64 pixelů. Princip obsluhy jsem již přiblížil. Tato knihovna obsahuje veškeré funkce, které budeme potřebovat pro obsluhu. Struktura této knihovny je naznačena na obrázku 40.

```
#ifndefLCDLIB // nutná hlavička pokud není definován tak připoj
#defineLCDLIB // jenom jednou
#include<avr/io.h> // přidání knihoven pro Mcu
#defineLCDdataportPORTA // definice proměných např. port pro data
....

voidinicializace(); //deklarace všech funkcí
....

constunsignedcharfont14[]PROGMEM={...} //definicekonstant
.... //uložených v programové paměti

voidLCDm(char*q,uint8_tx,intstr,uint8_tcolor=0){
//vlastní kód funkcí
}
....

#endif //ukončení knihovny
```

Obrázek 40 Struktura knihoven pro grafický displej LCDLIB.h

6.1.1.2 Funkce pro ovládání pinů

Zde jsem vytvořil jednoduché funkce pro ovládání jednotlivých pinů pro displej. Příklad jedné funkce vidíme na obrázku 41.

```
void RS(int a){
    //define LCD_RSPIN4_bm bitová maska
    //pro pin 4
    if(a==0){PORTB_OUTCLR=LCD_RS;} // pin 4 portu B na log 0
    else PORTB_OUTSET=LCD_RS; // pin 4 portu B na log 1
}
```

Obrázek 41 Příklad funkce pro ovládání pinu RS (data/příkaz)

Podobným přístupem jsem vytvořil funkce:

- void RW(int a) - nastavení čtení nebo zápisu z/do displeje
- void E(int a) - pin pro hodinový signál
- void CS1(int a) - pin pro výběr levého řadiče
- void CS2(int a) - pin pro výběr pravého řadiče

Nejdříve je nutná inicializace celého displeje. Jedná se o přesně definovanou sekvenci příkazů a data pro správnou funkčnost obou řadičů. Přesný popis je uveden v každém datasheetu. Důležité je např. zpoždění po naběhnutí napájení. Vytvořil jsem funkci `void inicializace()`.

Funkce `void LCDDATA(uint8_t a)` zapíše vlastní data do displeje a vypadá názorně:

```
void LCDDATA(uint8_t a){
    RW(0); //nastaveni pro zápis
    _delay_us(1); //zpoždění
    RS(1); //nastavení pro posílání dat
    _delay_us(2);
    E(1); //hodinový signál
    _delay_us(2);
    LCDdataport.OUT=a; //poslání vlastních dat na datový port
    _delay_us(2); //nutné ustálení log stavů
    E(0); } //při sestupné hraně odeslání dat
```

Dále pak funkci `void LCDX(uint8_t a)` pro nastavení pozice ve směru X v hodnotě 0-63:

```
void LCDX(uint8_t a){ //hodnota a v rozmezí 0-63
    RW(0); //zápis
    RS(0); //instrukce
    E(1); //hodinový signál
    _delay_us(1);
    LCDdataport.OUT=(0b01000000+a); //01 instrukce + 6bitů pozice
    _delay_us(1); //ustálení hodnot
    E(0); //sestupná hrana hodinového signálu
}
```

Na podobném principu jsou koncipovány funkce:

- `voidstranka(uint8_t a)` -nastavení pozice ve směru Y v rozmezí 0-7
- `voiddelLCD()` -smazání celého displeje
- `voidLCDY(int a)` -tato funkce umožňuje rolování ve směru Y 0-7

6.1.1.3 Funkce pro zobrazování znaků o velikosti 8 pixelů

Jak jsem již poznamenal, pole konstant, které nám definuje jednotlivé alfanumerické znaky, je uloženo v programové paměti procesoru, docílíme toho, že ušetříme datovou paměť procesoru. Klíčové slovo pro umístění dat do programové paměti je `PROGMEM`, musíme použít knihovnu `<avr/pgmspace.h>`. Nevytvářel jsem si svůj vlastní font, ale použil jsem již definované pole. [14]

Toto písmo má velikost 8pixelů. Definoval jsem toto pole jako

`constunsignedcharfont8[]PROGMEM`. Dále jsem si vytvořil vlastní funkci pro zobrazení poslaného řetězce na displeji.

```
constunsignedcharfont8[]PROGMEM={
    0x00,0x00,0x00,0x00,0x00, // SPACE
    0x00,0x00,0x5F,0x00,0x00, // !
    0x00,0x03,0x00,0x03,0x00, // "
    0x14,0x3E,0x14,0x3E,0x14, // #
    0x24,0x2A,0x7F,0x2A,0x12, // $
    0x43,0x33,0x08,0x66,0x61, // %
    0x36,0x49,0x55,0x22,0x50, // &
    0x00,0x05,0x03,0x00,0x00, // '
}
```

Zde vidíme příklad uložených dat. Každý znak má 8 bytů. Jejich pozice v tomto poli odpovídá hodnotě ASCII, tzn. první znak – mezera má dekadickou hodnotu podle tabulky „32“. Podle této první pozice pak budeme přepočítávat veškeré další znaky.

```
voidLCDm(char*q,uint8_tx,intstr,uint8_tcolor=0){
uint8_t a=0; //do této funkce vstupují parametry:
LCDX(x); //vstupní řetězec, pozice x,y
stranka(str); //a případná inverze textu
uint8_t pocet=0;
while(1){
if(q[a]!='\0')break; //čekání na konec řetězce
else{ for(int b=0;b<5;b++){ //každý znak má 5 bytů
if((pocet+x)<64){CS1(1);CS2(0);} //volba řadiče 0-64 a 64-128
elseif((pocet+x)>127)break;
else{CS1(0);CS2(1);LCDX(x+pocet-64);}
if(color==0){LCDDATA(((uint8_t)*(&(font[(((q[a])-32)*5+b]))));)}
else{LCDDATA(~((uint8_t)*(&(font[(((q[a])-32)*5+b]))));)}
pocet++; //prepočítání pozice znaku v poli
}
if(color==0){LCDDATA(0x00);} //vlození mezery za písmenem
else{LCDDATA(0xFF);}
pocet++;
a++;}
if((pocet+x)>=126)break;
else;
}}
```

Do této funkce vstupují parametry jako jsou: řetězec znaků pro zobrazení, pozice začátku ve směru X (0-127) a pozice Y (0-7), dále pak parametr, který invertuje zobrazení písmen (využijeme např. při zobrazování položek v menu). Nejdříve nastavíme výchozí pozici x a y. Cyklus while() kontroluje konec řetězce, který je ukončen vždy znakem „\0“. Pro každý znak vyhledáme jeho příslušnou pozici v definovaném poli fontu. Dále po každém odeslání hodnoty dat musíme kontrolovat, jestli jsme nepřekročili pozici prvního řadiče displeje (0-63 pixelů). Každý znak je definovaný 8 byty, proto cyklus for(). Zbývá jen přidání mezery mezi jednotlivými znaky a ošetření podmínky, kdy velikost řetězce je příliš dlouhá, aby se celý vešel na displej.

6.1.1.4 Funkce pro zobrazování znaků o velikosti 14 pixelů

Zde jsem řešil podobnou problematiku, jako u předchozího případu. Opět jsem si stáhl definovaný font o velikosti 14 pixelů. [15]

Vytvořil jsem si funkci `voidLCD(char*q,uint8_tx,uint8_tstr,uint8_tcolor=0)`

Zde opět vstupuje řetězec znaků, pozice x a y, případně inverze zobrazení znaků. Nejsložitější bylo přiřazení jednotlivých znakům data v definovaném poli dat, protože zde má každý znak jiný počet bytů. Na obrázku 42 vysvětlím, jak jsou tyto data uložena.

Pozice0	Pozice1	PoziceN	Data0	Data1(pozice Data0 + delka znaku0)
Délka znaku0	Délka znaku1		Délka znakuN	Data prvního znaku „!“ ASCII hodnota 32	Data prvního znaku „!“ ASCII hodnota 33

Obrázek 42 Struktura uložení dat v poli FONT14

Jelikož písmo je o velikosti 14 pixelů, tak pro jeden sloupec znaku potřebujeme 2 byty (8 + 6 bitů). Takže hodnota v poli „délka znaku0“ udává přesně dvounásobnou délku vlastního znaku. Kvůli rychlosti vykresluji celou horní část řetězce a pak se cyklus vrátí na počáteční místo a vykreslí spodní polovinu znaků. Pro příklad máme zobrazit znak „!“, ASCII dekadická hodnota je 33, první znak v poli má hodnotu 32, tím pádem budeme vykreslovat znak od

pozice: PoziceN+délka znaku0, do pozice: PoziceN+ délka znaku0 +délka znaku1. Celé toto pole jsem opět uložit do programové paměti.

6.1.1.5 Funkce pro pixel

Tato funkce adresuje jeden pixel v rozmezí souřadnic X(1-128) a Y(1-64). Vstupními hodnotami jsou právě tyto souřadnice a parametr rozsvícení a zhasnutí pixelu.

```
voidLCD_pixel(uint8_tx,uint8_ty,uint8_tcolor=0){
    uint8_tdata;
    unsignedintw=(unsignedint)((y-1)/8);    //spocitanicislostranky
    stranka(w);                               //nastavení požadované stránky
    if(x<65){CS1(1);CS2(0);LCDX(x-1);}      //zvolení řadiče 1-64 nebo 64-128
        else{CS1(0);CS2(1);LCDX(x-65);}
    LCDdataport.DIR=0x00;                     //port pro data jako vstupní
    RW(1);                                    //čtení z displeje
    RS(1);                                    //čtení dat
    E(1);                                     //hodinovýcykl
    E(0);
    E(1);
    data=LCDdataport.IN;                      //ulození dat z LCD
    LCDdataport.DIR=0xFF;                    //nastavení portu dat jako výstupní
    RW(0);                                    //čtení z displeje
    if(x<65){CS1(1);CS2(0);LCDX(x-1);}      //vracení se na adresu o 1 zpet
    else{CS1(0);CS2(1);LCDX(x-65);}
    RS(1);                                    //posílání dat
    E(1);
        if(color==0
            {LCDDATA(data|(1<<(y-8*w-1)));    //nastavení určeného pixelu a
            //přičtení k původním datum
        }
        else{LCDDATA(data|(~(1<<(y-8*w-1)))));}
    E(0);                                     //hodinový cyklus pro zaspání dat
}
```

Tuto základní funkci dále využiji při vykreslování úseček a obdélníků a kružnic. Z hlediska rychlosti vykreslování složitější tvarů tento přístup není ideální, protože samotný řadič displeje je celkem pomalý, takže výhodnější přístup by byl například, nezapisovat data přímo do RAM paměti displeje, ale vytvořit si paměťové místo přímo v procesoru a všechny výpočty a pozice pixelů počítat a adresovat v paměti procesoru a ve finále tento buffer poslat celý najednou do paměti displeje. Tento přístup by byl výrazně rychlejší.

6.1.1.6 Funkce pro vykreslení úsečky a kružnice

Pro vykreslení obecné úsečky jsem vytvořil funkci `voidLCDprimka(uint8_txstart,uint8_tystart,uint8_txend,uint8_tyend)`, do této funkce vstupují počáteční hodnoty (x,y) a koncové hodnoty úsečky (x,y). V této funkci jsem nejprve

ošetřil speciální případy úseček jakou jsou vertikální a horizontální směry a směr pod 45°, všechny ostatní případy je nutno řešit jedním z několika způsobů rasterizace.

Vybral jsem si Bresenhamův algoritmus, který je možno dále použít jak pro úsečku, tak pro výpočet kružnice. Tento algoritmus používá celočíselný výpočet rasterizace. Základní princip je, že ve směru jedné osy se inkrementuje pozice a pozice ve druhém směru záleží na velikosti chybového členu a podle toho se určí bližší pozice druhé souřadnice k ideální přímce.[16]

$$\Delta x = x_2 - x_1 \quad \Delta y = y_2 - y_1 \quad k = \frac{\Delta y}{\Delta x} \quad (1)$$

$$e_i = 2 \times \Delta y - \Delta x \quad (2)$$

$$e_i < 0 \quad [x_i + 1, y_i] e_{i+1} = e_i + 2 \times \Delta y \quad (3)$$

$$e_i \geq 0 \quad [x_i + 1, y_i + 1] e_{i+1} = e_i \times (2 \times \Delta y - \Delta x) \quad (4)$$

Podle rovnice (1) vypočteme přírůstky v obou osách, a podle rovnice (2) první chybový člen pro další pozici pixelu. Podle hodnoty chybového členu inkrementujeme, nebo necháváme pozici y (3) (4). Je jasné, že nejdříve před těmito výpočty musíme zjistit, jestli budeme pozici y inkrementovat nebo naopak. Tímto algoritmem jsem vyřešil funkci pro vykreslení obecné úsečky.

Pro vykreslení kružnice jsem si vytvořil funkci `voidLCDkruh(uint8_t xstred, uint8_t ystred, uint8_t polomer)` jako vstupní parametry jsou pozice středu x a y a poloměr kružnice. Opět jsem použil Bresenhamův algoritmus.

$$p_1 = 3 - 2 \times r \quad (5)$$

$$p_i < 0 \quad [x_i + 1, y_i] p_{i+1} = p_i + 4 \times x_i + 6 \quad (6)$$

$$p_i \geq 0 \quad [x_i + 1, y_i - 1] p_{i+1} = p_i + 4(x_i - y_i) + 10 \quad (7)$$

Tímto algoritmem vypočteme jen druhý oktant kružnice, zbytek symetricky doplníme. První chybový člen vypočteme podle vzorce (5), r – je poloměr kružnice. Poté podobně jako v případě úsečky rozhodujeme, jestli chybový člen je menší než nula, nebo je větší nebo roven nule. Podle rovnic (6) a (7) pak vypočteme nadcházející chybový člen.[16]

6.1.1.7 Funkce pro zobrazení obrázku

Jedná se o funkci `voidLCDpicture(constuint8_tobrazek[],uint8_txstart,uint8_tystart, uint8_txsize,uint8_tysize,uint8_tcolor=0)`, do které vstupuje vlastní pole s daty obrázku, počáteční souřadnice obrázku x a y , a velikost obrázku v pixelech x a y . Opět tu máme možnost celý obrázek invertovat.

Pro pohodlné zpracování již existujícího obrázku, jsem použil program `Image2GLCD`. Jedná se o jednoduchý program, kterým otevřeme jakýkoliv obrázek, zadáme velikost displeje, nebo vlastního obrázku a program nám vypočte potřebné pole s daty. Můžeme nastavit práh jasu, nebo nastavení uspořádání dat v jednom bytu. [17]

6.1.2 USART

Pro komunikaci s PC používám rozhraní RS232 a dále pro komunikaci pro rozšiřující moduly dva kanály RS485. Aktuální program používá jako zdroj taktovacího kmitočtu integrovaný RC oscilátor 2 Mhz. Podle toho musíme nastavit přenosovou rychlost sériové linky. Na stránkách [18], je možno pohodlně vypočítat hodnoty registrů pro zvolený zdrojový kmitočet a přenosovou rychlost. Sériový kanál jsem nastavil na rychlost 9600baud, délku dat 8 bitů, bez paritního bitu a jeden stop bit. Využitý je například sériový kanál na portu F.

```
USARTF0_CTRLC=(uint8_t)USART_CHSIZE_8BIT_gc|USART_PMODE_DISABLED_gc;
//nastavení délky dat 8 bitů a zakázání bitu parity
USARTF0_CTRLB|=USART_RXEN_bm|USART_TXEN_bm; //povolení přijímače a vysílače
USARTF0_BAUDCTRLA=(uint8_t)12; //nastavení rychlosti na 9600
USARTF0.CTRLA=0x20; //nastavení přijímače na střední úroveň přerušení
```

Tímto nastavíme jeden sériový kanál, pro každou linku jsem si vytvořil dva zásobníky – buffery, pro příjem a vysílání. Každý buffer má velikost 20 bytů.

6.1.3 Tlačítka

Zde jsem opět využil možností přerušovacího systému procesoru. Nastavil jsem všechny 4 piny, ke kterým jsou připojeny tlačítka tímto způsobem:

```
PORTF_PIN4CTRL|=PORT_OPC_PULLUP_gc|PORT_ISC_FALLING_gc;
//nastavení pinu 4 s pullup rezistorem a reakce na sestupnou hranu
PORTF_INTCTRL=PORT_INT0LVL_LO_gc; //nastavená nízká úroveň přerušení
PORTF_INT0MASK|=0b11110000; //přerušení INT0 portu F na pinech 4-7
```

Tím pádem se bude generovat přerušení na nízké úrovni a jen při sestupné hraně napětí. Spolu s přidávanými blokovacími kondenzátory se domnívám, že je to celkem dobrá ochrana

proti záskmitům tlačítek. Další velice dobrá možnost je využití událostního systému procesoru ATXmega. Disponuje možností nastavení události změny stavu hodnoty na kterémkoli pinu, opět s možností nastavením náběžné, sestupné hrany nebo úrovně. Největší výhodou je pak využití až 10ti vzorkového digitálního filtru, který tuto událost může zpracovávat. Tím bychom docílili maximální ochrany proti záskmitům tlačítek.

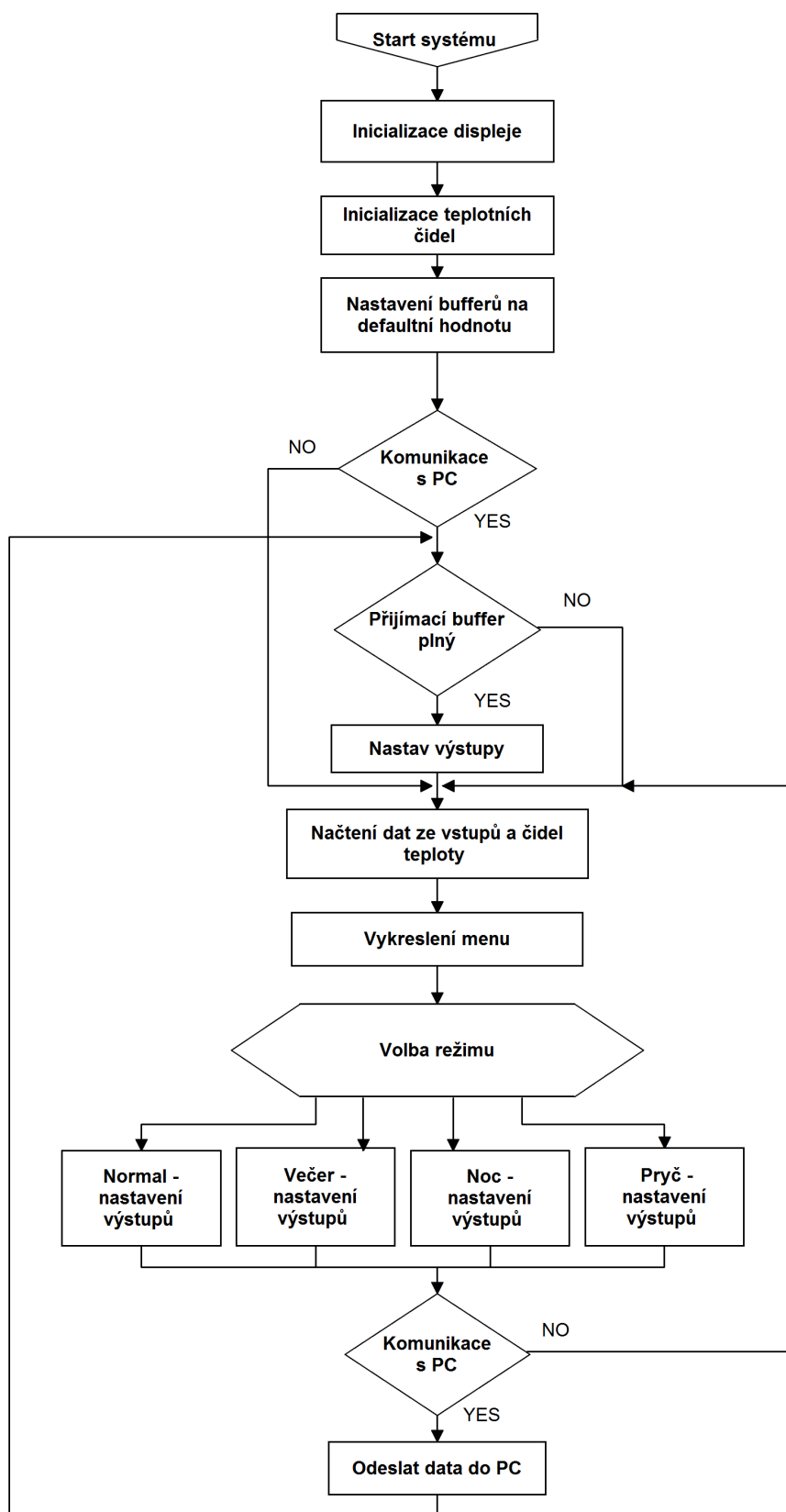
6.1.4 Vstupy a výstupy

Jak jsem již uvedl v návrhu, hlavní řídicí jednotka obsahuje dva posuvné registry pro vstupy, které jsou na 24V a výstupy, ty jsou koncipovány na spínání 230V s maximálním spínaným proudem 1A nebo 10A. Jak vstupy, tak výstupy je možné libovolně rozšířit dalšími posuvnými registry. Vytvořil jsem si funkci `charstaticIO_IN()` a `voidstaticIO_OUT(chara)`, zatím bez parametru počtu posuvných registrů, tyto funkce načítají nebo posílají stavy vstupů nebo výstupů vždy po 8 bitech. Do budoucna tyto funkce budou rozšířeny právě o parametr počtu posuvných registrů.

6.1.5 Časovače pro stmívání světla

Princip stmívání byl už vysvětlen, takže k programové realizaci jen zmíním nastavení. Ke každému stmívatelnému světlu jsem přiřadil jeden časovač. Perioda jedné usměrněné půlvlny je 10ms. Nastavil jsem předděličky časovačů ze zdroje taktu interního oscilátoru 2000000 Mhz na poměr 64, tím jsem dostal interval 0,032 ms. Princip celé regulace je v tom, že když obvod detekce průchodu nulou signalizuje střídavé napětí velmi blízké nule, tak se zapne časovač spolu s logickým výstupem světla a podle nastavené délky přetečení časovače nastavujeme intenzitu světla, např. pro 100 % je hodnota přetečení čítače zhruba 315. Pro 50 % výkon pak polovina. Tímto řešením se nabízí variabilita způsobu spínání. Můžeme si vybrat možnost spínání na náběžnou nebo sestupnou hranu a tím i druh zátěže - RLC.

6.1.6 Řídicí program

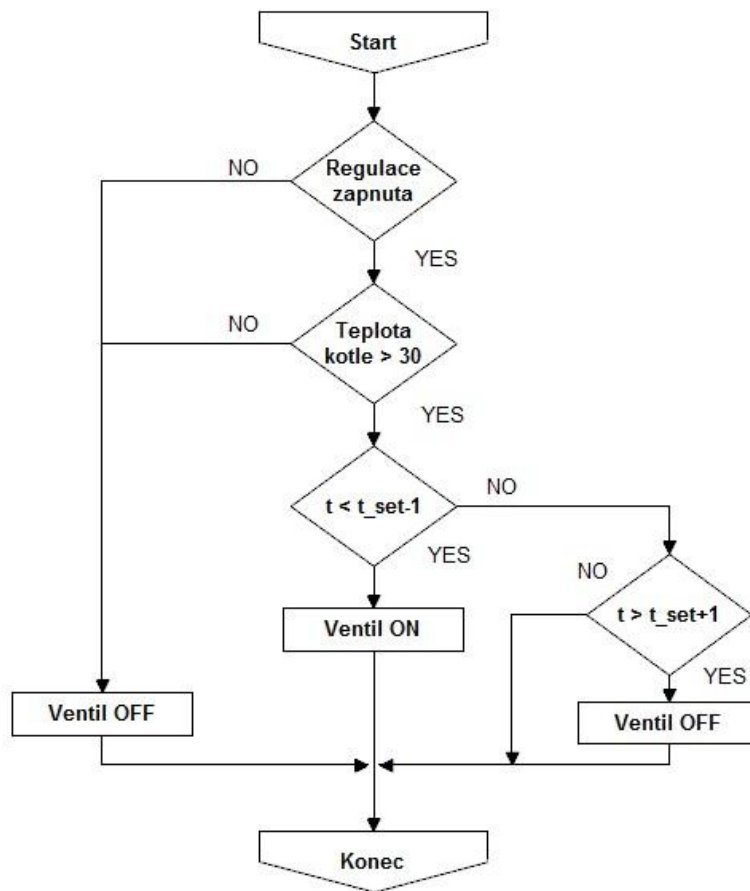


Obrázek 43 Struktura hlavního programu v jednotce master

Z obrázku 43 vidíme celou strukturu hlavního řídicího programu v řídicí jednotce. Po zapnutí jednotky se nejprve inicializuje grafický displej, pak dojde k inicializaci teplotních čidel na 1Wire sběrnice a k jejich identifikaci, poté dojde k načtení defaultních hodnot do obou bufferů. Řídicí jednotka, jak již bylo řečeno, může celý systém ovládat i bez připojení k PC a posléze i k databázi. Komunikace s PC slouží pouze k předávání nastavujících parametrů, nikoli k hlavnímu řídicímu procesu. To znamená, že všechny parametry nebo volby programů a podprogramů, můžeme volit jak přímo na řídicí jednotce, tak i přes PC. Po veškeré inicializaci všech potřebných komponent zjišťuje program, jestli je zapnutá komunikace s PC, jestliže ano, pošle všechny parametry a data po sériové lince nebo USB. Podobně detekujeme, jestli je přijímací buffer plný, jestli přišla nějaká data. Pokud ano, program zpracuje přijatá data a podle toho nastaví parametry systému a výstupů. Dále program zjišťuje hodnoty teplotních čidel a vstupů. Poté se program dostává k vykreslení menu a zpracování úprav vstupujících z ovládacích tlačítek. Dále se dostáváme k větvení podle zvoleného programu. Máme k dispozici 4 základní programy: Normální, Večer, Noc a Pryč (nepřítomnost). Každý z těchto programů má nastavené své specifické parametry, které popíši níže. Nakonec, po průchodu nastaveným programem se opět detekuje, jestli je zapnutá komunikace s PC, v případě že ano, vyšlou se data a parametry do PC – databáze. A posléze se nastaví výstupy.

Na volbě programu závisí nastavení několika parametrů, ať se to týká spínání osvětlení, nebo nastavení aktuální teploty až po zapnutí nebo využití např. zabezpečovací techniky. Např. v režimu „Normální“ budou nastaveny určité teploty v každé topné zóně, zabezpečovací technika (např. detektory pohybu) budou vypnuty, světla se budou zapínat jen manuálně a na původní intenzitu. V programu „Večer“ využijeme detektorů pohybu, které budou rozsvěcet světla na chodbách apod. Budeme mít uložené jiné teploty pro večerní režim. A v režimu „Noc“ budeme mít nastavení opět jiné teploty, zabezpečovací technika bude aktivní jen v neobytných prostorách (garáž, dílna, zahrada...), všechna světla se budou rozsvěcet automaticky na poloviční výkon. Toto jsou hlavní záměry využití režimů v celodenním provozu.

6.1.7 Regulace topení



Obrázek 44 Regulace topení

Na obrázku 44 vidíme princip regulace topení v jedné zóně, nebo jednom pokoji. První parametr je, jestli máme zapnutou regulaci topení, v případě kdy není topná sezóna apod. Další podmínka hlídání teploty kotle, v případě kdy klesne teplota pod nastavenou hodnotu, vypne se také regulace. Následující dvě podmínky nastavují hysterezi spínání ventilu, která je jako jedním z parametrů, které si můžeme nastavit. V uvedeném příkladu je nastavena na +/- 1°C.

6.1.8 Ovládání a menu nabídky

Jak již bylo řečeno, jakékoli nastavení přímo na řídicí jednotce se projeví i ve webovém prohlížeči a naopak. Poskytuje nám to velké pohodlí a variabilitu. Na obrázku 45 vidíme vyfocené menu přímo z displeje. Položka „Status“ nám zobrazuje, jestli a jak je připojena

jednotka k PC – serveru. Na výběr máme 7 menu položek, ve kterých si můžeme detailně nastavit konstanty, data a režimy celého systému.



Obrázek 45 Menu hlavní řídicí jednotky

Ovládání světla

Zde máme možnost nastavit intenzitu osvětlení pro 4 světla v rozmezí 0-100 %. Nebo je rychlím tlačítkem zapnout nebo vypnout. Je zde přehledný posuvník, který naznačuje stav daného světla viz. Příloha [1].

Nastavení teplot

Tato položka umožňuje zobrazit teplotu v pěti pokojích (popř. okruzích) a dále pak upravit nastavenou teplotu. Příloha[2]

Zabezpečení EZS

Zde je možno nastavit vybrané okruhy zabezpečovacího systému, buď můžeme všechny tři zóny najednou vypnout, nebo si zvolit konkrétní zóny jako aktivní. Příloha[3]

Volba režimu

V této nabídce si můžeme zvolit aktuální režim, ve kterém má celý systém pracovat. Jak již bylo zmíněno, máme na výběr ze čtyř režimů: Normální, Večer, Noc a Pryč. Poslední nastavitelná položka je budík – on/off. V případě, že máme nastavenou nějakou událost, tímto globálně zakazuje nebo povolujeme provedení této události. Příloha[4]

Nastavení

Zde může uživatel nastavit, zapnutí nebo vypnutí regulace vytápění, konstantu pro diferenci spínání jednotlivých ventilů, zapnutí nebo vypnutí podsvícení displeje, nakonec ještě datum a čas. Příloha[5]

Servis

Tato položka umožňuje testování vstupů a výstupů, dále si zde můžeme zobrazit stav vstupního a výstupního bufferu na lince RS232, dále počet připojených slave jednotek na RS485, hodnoty všech teplotních čidel a nakonec test tlačítek. Příloha[6]

Komunikace

V poslední nabídce můžeme nastavit komunikaci s PC, anebo jí úplně zakázat. Je zde možnost vybrat si mezi komunikací přes RS232 nebo USB. Příloha[7]

6.2 Aplikace C# pro sériový port a databázi

Cílem této aplikace je, přijímat a vysílat data po sériovém portu (nebo USB) a dále je pak zpracovat a ukládat do MYSQL databáze. Do budoucna bych tuto aplikační část využil i k časovacím událostem jako jsou například aplikace: „Ráno v 6 hodin zapni lampičku na 40% spolu s rádiem.“ Tento druh aplikací by si žádal rozšíření dosavadní databáze. Pro jazyk C# jsem se rozhodl, protože má velice pohodlné grafické rozhraní a jednoduchý přístup k sériovým portům. Vytvořil jsem si MYSQL databázi s tabulkou „data“ v této tabulce jsou zatím jen dva vložené řádky, které představují data přijatá a data vyslaná. Struktura dat je naznačena na obrázku 46, všechna pole jsou 8bitová.

start	vstupy	výstupy	Světlo1	Světlo2	Světlo3	Světlo4	Set_t1	Set_t2	Set_t3	Set_t4	Set_t5
T1	T2	T3	T4	T5	Data0	Data1	Data2				

Obrázek 46 Struktura dat v databázi

První byte je definován jako hodnota FF, následuje byte stavu vstupů a výstupů, dále pak hodnoty 0-100 pro čtyři regulovatelná světla, pět hodnot požadované teploty a pět hodnot aktuální teploty v místnostech. Data0-2 určují zvolený režim, stav zabezpečovací techniky, zapnutý režim regulace teploty a nastavení budíku. Tato aplikace za použití jednoduchého časovače, každých 100ms zpracuje příchozí data od řídicí jednotky a uloží je do vstupních dat databáze, poté vyhodnotí stav výstupních hodnot, a když jsou odlišná od předchozích (došlo ke změně jakékoli hodnoty), tak po sériové lince vyšle data do řídicí jednotky. Komunikační port mám nastavení na defaultní hodnotu COM5. Příloha [8]

6.3 MYSQL

MySQL je relační databázový typ Database Management System. Každá databáze je tvořena z jedné nebo více tabulek. Tabulka se skládá ze sloupců, které tvoří konkrétní datové typy položek. Každý sloupec má své jedinečné jméno a je definován datovým typem. Každý řádek pak představuje další záznam. K práci s databázemi a tabulkami se využívají textové příkazy – dotazy. [19]

Použil jsem program EasyPHP, který podporuje jak webový tak i MySQL server. Používám přístup k databázi jen pro lokální síť. Jak jsem již zmínil, vytvořil jsem si jednu databázi s jednou tabulkou „data“, ve které jsou dva záznamy všech potřebných dat. První záznam je pro vstupní a druhý pro výstupní data. Tudiž dotazy, které je potřeba použít, pracují vždy s celým řádkem dat. Například dotaz:

```
mysql_query("SELECT * FROM data WHERE id=1");
```

Tento dotaz vybere všechny data z tabulky „data“, kde id (identifikační číslo řádku) je 1. Další příkaz uloží nové data například jen do zadaného sloupce.

```
mysql_query("Update data SET data2='šstav' WHERE id=2");
```

Tento příkaz provede uložení hodnoty z proměnné \$stav do tabulky „data“ sloupce „data2“, kde sloupec „id“ je roven 2. Tímto způsobem se dostaneme k požadovaným datům.

6.4 Webové uživatelské rozhraní – Html, Php

Celý systém můžeme ovládat pomocí hlavní řídicí jednotky. Pro vyšší pohodlí a praktičnost jsem vytvořil webové rozhraní pro monitorování systému, tak i pro pohodlné ovládání. Takže budeme mít možnost ovládání z jakéhokoli počítače, tabletu a telefonu v rámci domácí sítě, případně i z vnějších sítí. Tyto stránky jsem psal pomocí Html a Php jazyka. Náhled stránky vidíme na obrázku 46. Pro zatím zde chybí zabezpečení heslem, které vytvoří relaci na stránkách a ochrání tak systém před cizím nebo neoprávněným přístupem. Jinak veškeré dotazy a vkládání hodnot jsem vyřešil formuláři a následné zpracování pomocí php jazyka, který také ovládá a komunikuje s databází.



Obrázek 47 Uživatelské webové rozhraní pro ovládání domu - světel

Na obrázku 47 vidíme záložku ovládání světel, zde můžeme pohodlně zapnout/vypnout konkrétní světlo, nebo nastavit příslušnou intenzitu. K dispozici jsou také světlené scény, které nám nastaví jedním tlačítkem všechny světla. V dalších záložkách nastavujeme stejné parametry, jako jsem uváděl v možnostech přímo na hlavním panelu řídicí jednotky. Změny provedené ručně v řídicí jednotce se s malým zpožděním zobrazí ve webovém prohlížeči a naopak. Stránky jsou tedy shodné s menu v hlavní řídicí jednotce. Druhá záložka nastavuje a monitoruje teploty včetně možnosti zapnutí a vypnutí vlastní regulace topení. Třetí záložka umožňuje upravit zóny zabezpečovacího systému. V další záložce můžeme nastavit aktuální program pro řízení celého domu. V poslední záložce si můžeme nastavit časovač nebo budík. Tuto funkci nemám bohužel plně funkční, měla by umožňovat pohodlné nastavení času a data provedení události typu: „ráno v 6 hodin rozsvítí lampičku a roztáhne rolety“. Webové rozhraní je velice univerzální a nezáleží na tom, jaký systém a jaký přístroj používáme, tudíž si od toho slibuji vysokou univerzálnost a použitelnost.

7 Závěr

Cílem této práce bylo seznámení s inteligentními budovami a jejich možnostmi. Popsal jsem většinu dostupných komponentů a funkcí inteligentní budovy. Seznámili jsme se s některými významnými firmami, které se zabývají komplexními systémy pro řízení a zabezpečení celého domu. Dále se má práce zabývala nemultimediálními částmi inteligentní instalace, jejich návrhem a realizací. Navrhl jsem hlavní řídicí jednotku pro rodinný dům, která disponuje grafickým displejem a ovládacími tlačítky. Jednotka je schopná plynule regulovat 5 kanálů osvětlení nejen odporového charakteru. Dále disponuje sběrnici 1Wire pro připojení několik digitálních čidel teploty. Základní vybavení desky je 8 digitálních vstupů na 24 V, které budou sloužit pro detekci stavu vypínačů a ovládacích prvků elektrických rolet. Výstupy jsem navrhl všechny na 230V s zatížitelností od 1 do 10 A, s využitím jak pro ovládání elektrických rolet a ventilů, tak i pro spínání zásuvek. Tyto výstupy jsou navrženy tak, abychom je mohli libovolně pomocí dalších rozšiřujících jednotek vstupů a výstupů libovolně rozšířit. Jednotka je vybavena dvěma rozhraními pro sběrnice RS485, jedním RS232 a USB. Pomocí RS485 plánuji komunikaci po sběrnici s rozšiřujícími slave moduly, které zajišťují případné rozšíření systému. Slave jednotky jsou malé a jednoduché. Umožňují měřit teplotu a indikovat stavy třech vstupů a třech výstupů včetně jednoho kanálu pro regulaci osvětlení. Hardwarový návrh jsem dělal v programu Eagle v6.4, včetně obou plošných spojů. Drtivá většina součástek je pro smd montáž, ušetříme tím velikost plošného spoje. Obslužný program jsem programoval v prostředí AvrStudia 6 pomocí C++ jazyka. Pro hlavní řídicí jednotku jsem použil procesor od firmy Atmel konkrétně ATXmega256A3U, který má velice bohaté periférie a možnosti použití. Vytvořil jsem si vlastní knihovnu pro ovládání grafického displeje. Naprogramoval jsem si funkce pro použití dvou velikostí písma – 8 a 14pixelů, dále jednoduché funkce pro kreslení základních tvarů – bod, úsečka, obdélník a kružnice. Přidal jsem funkce pro vykreslení obrázků. Ovládání je možné jak pomocí grafického displeje a tlačítek, tak i pomocí jakéhokoli zařízení s přístupem do lokální sítě přes webové rozhraní. Menu v řídicí jednotce a webovém rozhraní jsou téměř podobné. Můžeme se nastavit jednotlivá světla, zapnout, vypnout nebo nastavit intenzitu 0-100%. Nechybí zde ani nastavení světelných scén. Dále můžeme monitorovat a nastavovat teploty v pěti místnostech a případně zapnout nebo vypnout vlastní regulaci teploty. Máme možnost nastavit si zóny zabezpečovacího systému, aby nebyl celý dům jen jedna velká zóna. Vlastní řízení a regulace závisí na aktuálním zvoleném programu : Normální, Večer, Noc a Pryč. Tyto

programy pak rozhodují o nastavení světel, teplot a zabezpečovací techniky. Rád bych do budoucna dodělal časování událostí, které by velice zpříjemnilo pobyt v domě. Systém je pomocí RS232 rozhraní připojen k PC, který zastává funkci serveru. Pomocí aplikace v C# jsem propojil sériový port s MYSQL databází, z které pomocí PHP jazyka zobrazuji data ve webovém rozhraní. Tímto se stává tento systém celkem uživatelsky pohodlný. Když bude člověk sedět v obývacím pokoji na pohovce, sáhne do kapsy pro mobilní telefon a zhasne si hlavní světlo. Řídicí jednotka je odzkoušena a připravena k použití, rád bych ji použil na našem rodinném domku, který se chystáme rekonstruovat a tím využít tento systém v praxi a v každodenním životě.

8 Zdroje

- [1] VALEŠ, Miroslav. *Inteligentní dům*, 1. vyd. Brno: ERA group spol. s.r.o., 2006. 122s. ISBN:80-7366-062-8.
- [2] JHSolar solární termické systémy [online]. citováno [10-3-2014]. Dostupné z: <<http://jhsolar.cz/tepla-voda>>
- [3] ABB [online]. citováno [12-3-2014]. Dostupné z: <<http://www.abb.cz/cawp/czabb014/49db58a47c2d18d4c1257598004151b4.aspx?v=C82&leftdb=GLOBA/CZABB/czabb014.NSF&e=cz&leftmi=cc70b0fdf470bdbcc1256a850029b508>>
- [4] HW.cz Inteligentní instalace: otázky a odpovědi [online]. citováno [11-3-2014]. Dostupné z: <<http://www.hw.cz/automatizace/inteligentni-elektroinstalace-otazky-a-odpovedi.html>>
- [5] Insignhome řešení pro chytré bydlení [online]. citováno [11-3-2014]. Dostupné z: <<http://www.insighthome.eu/CITIB.html>>
- [6] Savant [online]. citováno [11-3-2014]. Dostupné z: <<http://translate.google.cz/translate?hl=cs&sl=en&tl=cs&u=http%3A%2F%2Fwww.savantsystems.com%2F>>
- [7] Domoticaforum [online]. citováno [11-3-2014]. Dostupné z: <<http://www.domoticaforum.eu/uploaded/Ard%20M/Opentherm%20Protocol%20v2-2.pdf>>
- [8] Fibaro [online]. citováno [3-4-2014]. Dostupné z: <<http://mojefibaro.cz/home-center-2/>>
- [9] HW.cz [online]. citováno [8-4-2014]. Dostupné z: <<http://www.hw.cz/teorie-a-praxe/jednoduche-stmivani-svetel.html>>
- [10] Regulus [online]. citováno [8-4-2014]. Dostupné z: <<http://www.regulus.cz/cz/ventily-zonove-a-smesovaci>>
- [11] FK Technics [online]. citováno [10-4-2014]. Dostupné z: <http://www.fkt.cz/cz/clanky/soucastky/art_130/chap_205/blokove-schema-atxmega384a1.aspx>
- [12] Datasheet Lm2575 [online]. citováno [10-4-2014]. Dostupné z: <<http://www.gme.cz/img/cache/doc/935/038/lm2575s-adj-to-263-datasheet-1.pdf>>
- [13] Datasheet LF33 [online]. citováno [10-4-2014]. Dostupné z: <<http://www.gme.cz/img/cache/doc/934/038/lf33cdt-datasheet-1.pdf>>
- [14] MCU.cz [online]. citováno [20-4-2014]. Dostupné z: <<http://mcu.cz/news.php?extend.2721>>
- [15] PRJC.com [online]. citováno [20-4-2014]. Dostupné z: <https://www.pjrc.com/teensy/td_libs_GLCD.html>
- [16] WIKIKNIHY.org [online]. citováno [20-4-2014]. Dostupné z: <<http://cs.wikibooks.org/wiki/Geometrie/Rasterizace>>
- [17] Avrportal.com [online]. citováno [20-4-2014]. Dostupné z: <<http://www.avrportal.com/?page=image2glcd>>
- [18] Wormfood.net [online]. citováno [24-4-2014]. Dostupné z: <<http://www.wormfood.net/avrbaudcalc.php>>
- [19] Wormfood.net [online]. citováno [24-4-2014]. Dostupné z: <<http://www.junext.net/mysql/>>
- [20] Unimars.r.o [online]. © 2014 [vid. 5.3.2014]. Dostupné z: <http://www.unimarcz.cz/DELTA-DORE-zonova-regulace-DELTA-600-BT,2023.html>

- [21] JhSolar [online] @2014. [10.3. 2014] Dostupné z: <<http://jhsolar.cz/pritapeni/>>
- [22] ABB [online] @2014. [11-3-2014]. Dostupné z: <<http://search.abb.com/library/Download.aspx?DocumentID=9AKK105152A4204&LanguageCode=cs&DocumentPartId=&Action=Launch>>
- [23] Insighthome [online] @2014.[11-3-2014]. Dostupné z: <<http://www.insighthome.eu/CITIB.html>>
- [24] Savant [online]. @2014. [11-3-2014]. Dostupné z: <http://www.savantsystems.com/savant_and_apple.aspx>
- [25] Teco a.s. [online] @2014.[11-3-2014]. Dostupné z: <<http://www.tecomat.com/index.php?ID=636>>
- [26] Elkoep[online].[11-3-2014]. Dostupné z: <<http://www.elkoep.cz/produkty/inels-bus-system/multimedia/lara-inels-multimedia/>>
- [27] Fibaro[online].[3-4-2014]. Dostupné z:<<http://mojefibaro.cz/home-center-2/>>
- [28] HW.cz [online].[8-4-2014]. Dostupné z:<<http://www.hw.cz/teorie-a-praxe/jednoduche-stmivani-svetel.html>>
- [29] Allaboutcircuit[online].[8-4-2014]. Dostupné z: <http://www.allaboutcircuits.com/vol_3/chpt_6/12.html>
- [30] Pandatron.cz[online].[8-4-2014]. Dostupné z: <http://pandatron.cz/?3233&detektor_pruchodu_napeti_nulou>
- [31] Regulus[online].[8-4-2014]. Dostupné z: <<http://www.regulus.cz/cz/ventily-zonove-a-smesovaci>>
- [32] Root.cz [online] [8-4-2014]. Dostupné z: <<http://www.root.cz/clanky/sbernice-rs-422-rs-423-a-rs-485/>>
- [33] HW.cz [online] [10-4-2014]. Dostupné z: <<http://www.hw.cz/navrh-obvodu/rozhrani/aplikovani-sbernice-can.html>>
- [34] HW.cz [online] [10-4-2014]. Dostupné z: <<http://www.hw.cz/navrh-obvodu/rozhrani/aplikovani-sbernice-can.html>>
- [35] The Opentherm Protocol [online] [10-4-2014] Dostupné z: <<http://www.domoticaforum.eu/uploaded/Ard%20M/Opentherm%20Protocol%20v2-2.pdf>>
- [36] GME [online].[10-4-2014]. Dostupné z: <<http://www.gme.cz/lcd-graficky-displej-winstar-wg12864a-yyh-v-a-p513-245>>
- [37] Elektronika kvalitně. [online].[10-4-2014]. Dostupné z: <<http://elektronika.kvalitne.cz/ATMEL/necoteorie/LCDmatKS0108.html>>
- [38] HW.cz [online].[5-5-2014]. Dostupné z: <<http://www.hw.cz/navrh-obvodu/rozhrani/sbernice-1-wiretm.html>>
- [39] Your electronic open source [online].[5-5-2014]. Dostupné z: <<http://dev.emcelettronica.com/howto-measure-rtd-resistance-temperature-detectors-over-long-distances>>
- Wikipedie, *Ekvitermní regulace* [online].citováno [10-3-2014]. Dostupné z: <http://cs.wikipedia.org/wiki/Ekvitermn%C3%AD_regulace>

Příloha [1] Zobrazení menu řídicí jednotky – ovládání světel



Příloha [2] Zobrazení menu řídicí jednotky – nastavení teploty



Příloha [3] Zobrazení menu řídicí jednotky – zabezpečovací systém



Příloha [4] Zobrazení menu řídicí jednotky – nastavení režimu



Příloha [5] Zobrazení menu řídicí jednotky – nastavení



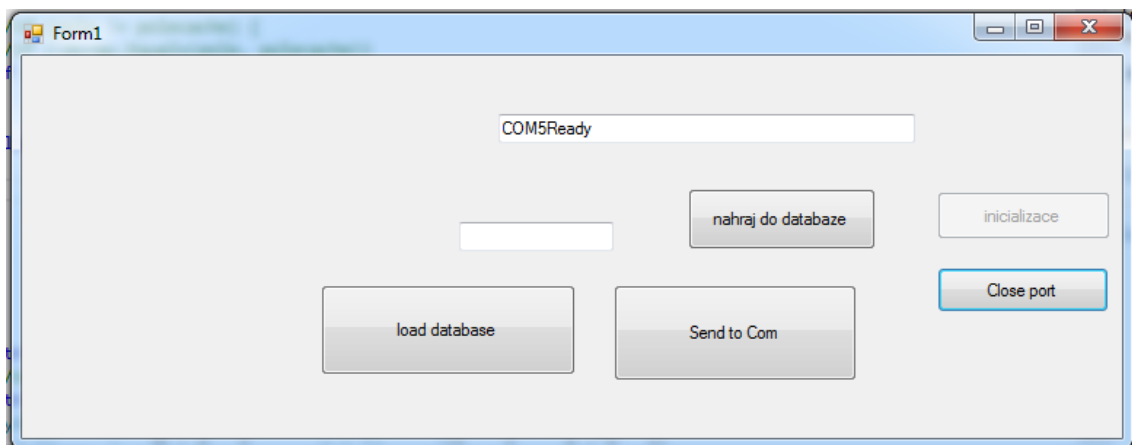
Příloha [6] Zobrazení menu řídicí jednotky – servis



Příloha [7] Zobrazení menu řídicí jednotky – komunikace



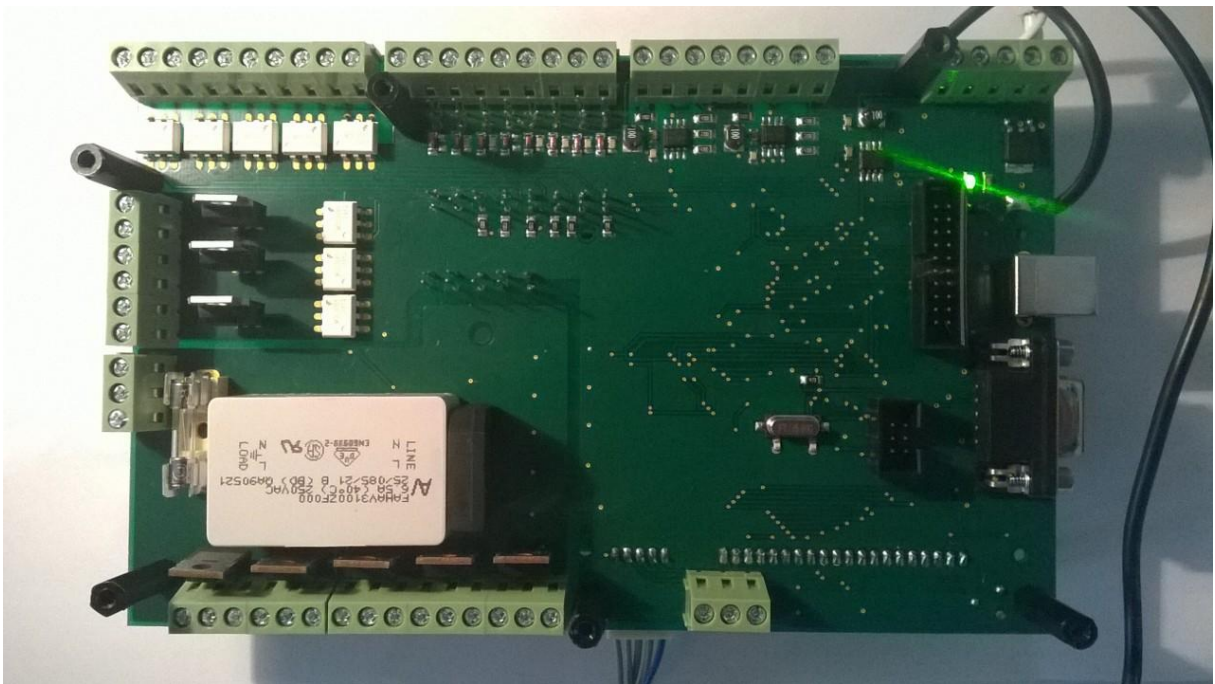
Příloha [8] Okno aplikace napsané ve Visual Studiu v C#



Příloha [9] Fotografie hlavní řídicí jednotky



Příloha [10] Fotografie hlavní řídicí jednotky



Příloha [11] Webová stránka Smart Home – ovládání světel



Příloha [12] Webová stránka Smart Home – nastavení teplot



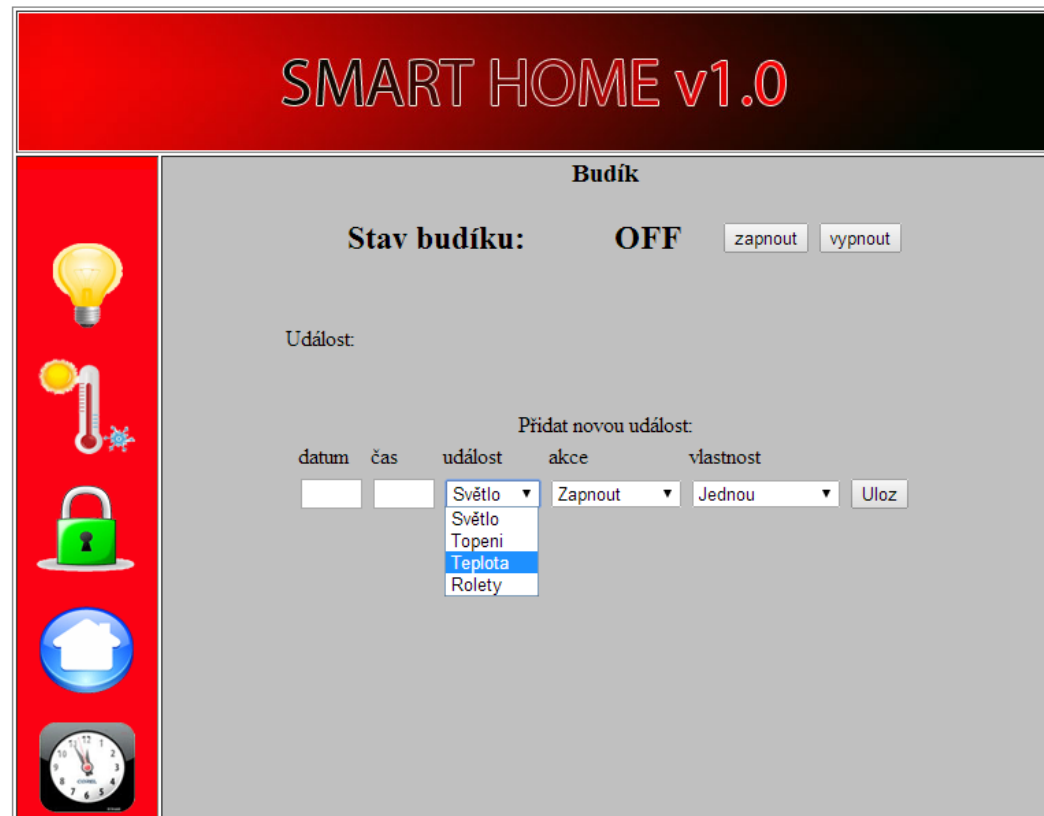
Příloha [13] Webová stránka Smart Home – EZS



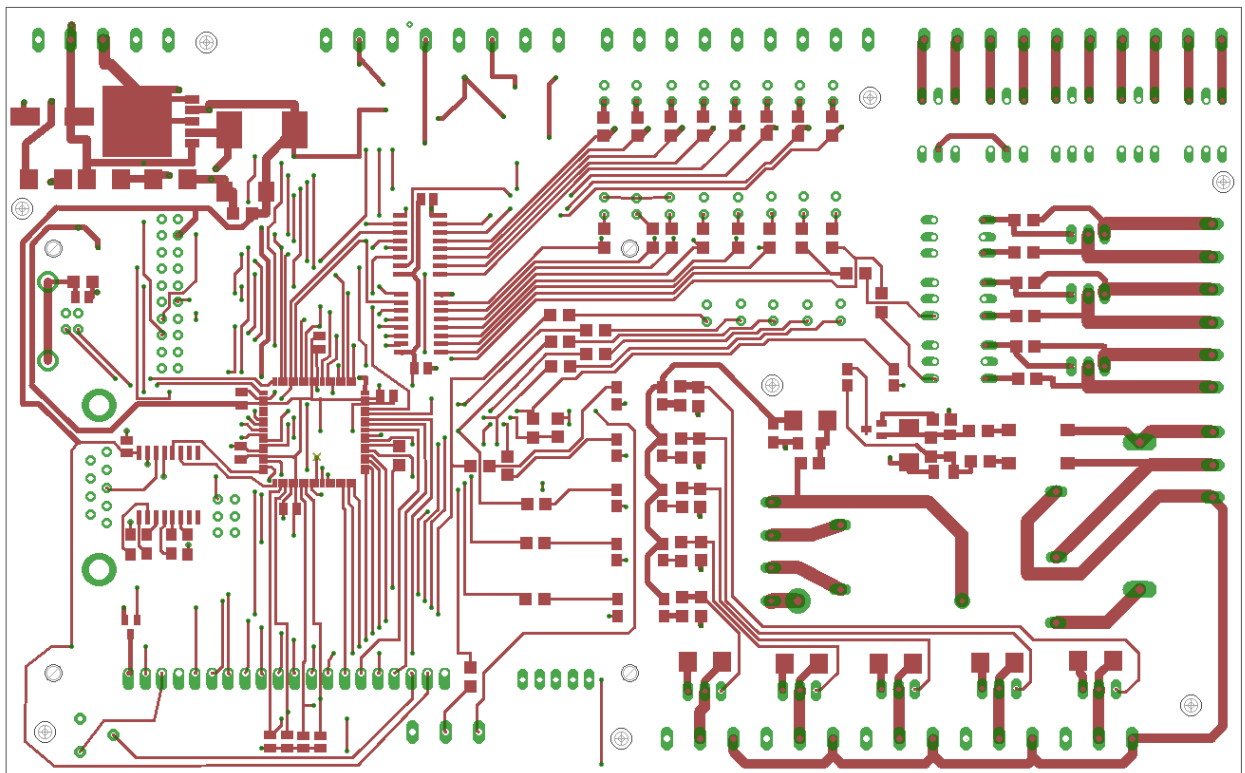
Příloha [14] Webová stránka Smart Home – volba režimu



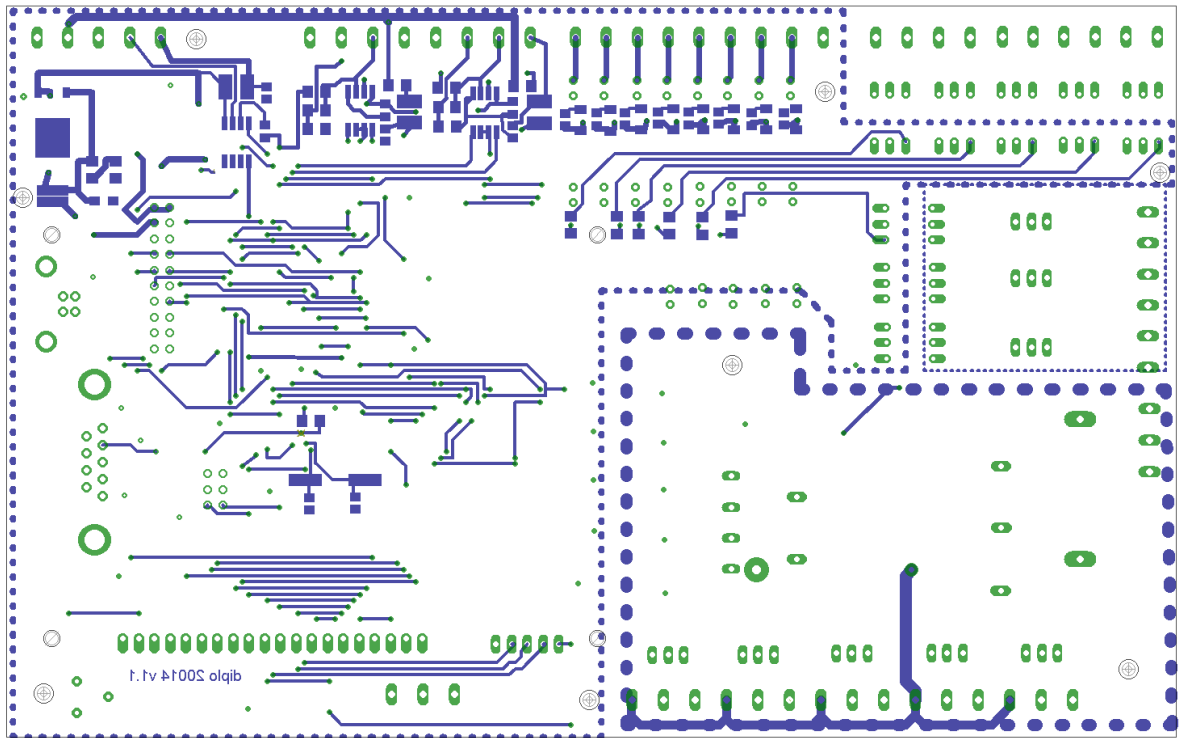
Příloha [15] Webová stránka Smart Home – budík



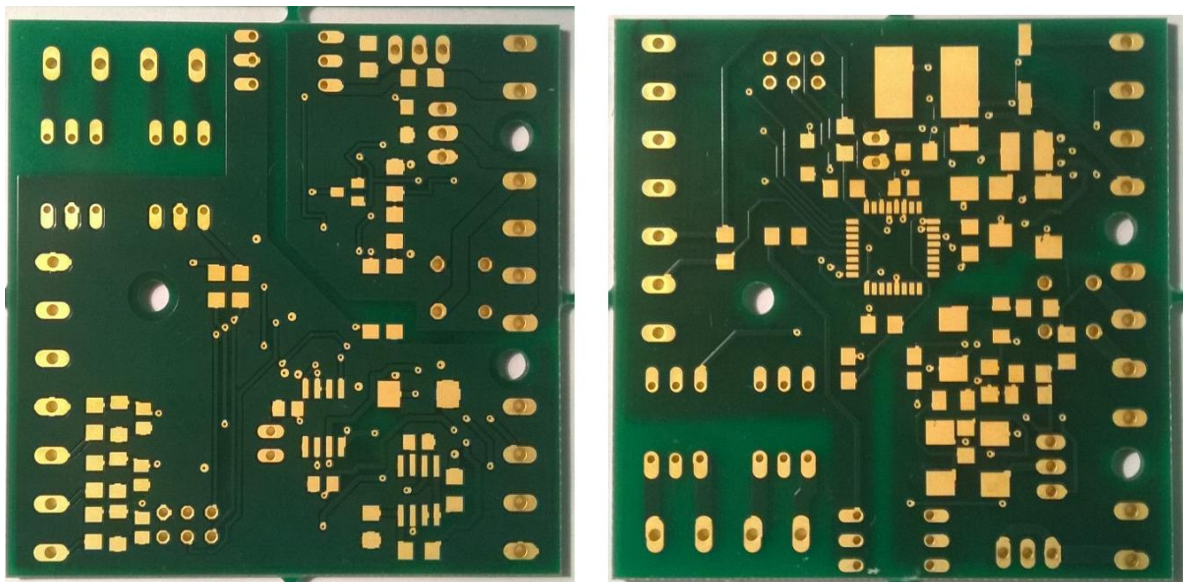
Příloha [16] Plošný spoj hlavní řídicí jednotky– TOP



Příloha [17] Plošný spoj hlavní řídicí jednotky – BOTTOM



Příloha [18] Plošný spoj slave jednotky



Příloha [19] Slave jednotka

