

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



BAKALÁŘSKÁ PRÁCE

Návrh čidla pro řízení hydroponického systému

Jan Štefan

Vedoucí práce: Ing. František Vacek



Katedra řídicí techniky

Pardubice, 2009

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

V Praze dne _____

podpis

Poděkování

Děkuji především vedoucímu bakalářské práce Ing. Františku Vackovi, Ing. Pavlu Píšovi a Ing. Petru Smolíkovi za pomoc při realizaci projektu.

Abstrakt

Cílem tohoto dokumentu je popsat průběh vývoje monitorovacího a řídicího systému, který je v praxi aplikován na chod hydroponického akvária. Práce obsahuje rozbor komunikace po sběrnici μ Lan a její fyzické vrstvy RS-485. Dále uvádí návrh čidel pro měření a řízení akvária, seznamuje s ovládacími deskami uLAD31 a s postřehy z průběhu vývoje aplikací pro jednotlivé periferie.

Klíčová slova: průmyslová sběrnice, řídicí jednotka, čidlo, měření, μ Lan.

Abstract

The goal of this project is to describe developing process of monitoring and control of system, which is in practice used for hydroponic aquarium. An analysis of communication μ Lan bus and its physics layer RS-485 is made within this project. In next part of the project is preceded design of sensors used for the aquarium monitoring and control, identification of control boards uLAD31 and remarks from single stages of development of applications for particular circumferences.

Keywords: industrial bus, control unit, sensor, measurement, μ Lan.

Obsah

1	Úvod	1
1.1	Motivace: Hydroponické akvárium	1
1.2	Stanovení cílů	2
2	Protokol μLan	3
2.1	Fyzická vrstva - RS 485	3
2.1.1	Obecný popis	3
2.1.2	Charakteristika sběrnice	4
2.2	Popis sběrnice μ Lan	5
2.2.1	Formát dat	5
2.2.2	Datový rámec	5
2.2.3	Přenos dat	6
2.2.4	Arbitrace sběrnice	6
2.3	Registrace do sítě	7
2.4	OI - Objektový slovník a jeho vytvoření	8
2.5	SDO - Service data object	9
2.6	PDO - Process data object	9
2.6.1	Princip CID-SLOT	10
2.6.2	Heart-beat	11
3	Realizace systému	13
3.1	Blokové schéma a popis systému	13
3.2	Desky uLAD31 a software	15
3.2.1	Hlavní smyčka (main loop)	16
3.2.2	Typy zařízení	16
3.2.2.1	Zařízení senzor	16
3.2.2.2	Zařízení control	18

3.2.2.3	Zařízení regulátor	18
3.3	Návrh hardwaru	19
3.3.0.4	Napájení systému	19
3.3.0.5	Teplotní čidlo	20
3.3.0.6	Vlhkostní čidlo	22
3.3.0.7	Jednotka digitálních vstupů/výstupů	25
4	Závěr	27
	Literatura	I
	A Obrázky	III
	B Software	VII

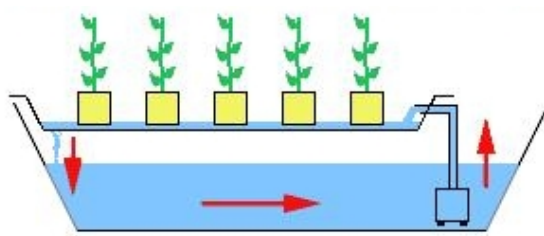
Kapitola 1

Úvod

1.1 Motivace: Hydroponické akvárium

Pod pojmem akvárium si představme uměle vytvořené prostředí nahrazující reálné podmínky v přírodě. Aby mohla rostlina růst v akváriu, musíme jí životní podmínky zabezpečit, tzn. světlo(teplo), vzduch a živiny. V přírodě se vyskytuje mnoho druhů rostlin. Každá z nich má odlišné vegetační požadavky. Náročně rostliny, jako například orchideie, rostou pouze v určitých částech naší planety za velmi specifických podmínek, zatímco jiné jsou nenáročné.

Akvárium označujeme jako hydroponické, protože rostliny pěstujeme bez půdy v živném roztoku. V těchto systémech voda cirkuluje, vrácený roztok se filtruje, doplní se živiny, upraví pH a teplota. Podle systému a účelu pěstování se přidávají hnojiva. Sluneční světlo nahradíme zdrojem, který svým světelným spektrem nejvíce podporuje průběh fotosyntézy.



Obrázek 1.1: Hydroponický systém

Cílem projektu je co nejvíce zautomatizovat tento náročný proces s možností monitorování všech potřebných informací nutných k ideálnímu růstu a s možností případného zásahu přes webové rozhraní. V praxi by se systém uplatnil například jako dekorace do reprezentativních místností, kde by v proskleném boxu rostly exotické rostliny.

1.2 Stanovení cílů

Myšlenka zautomatizování projektu nejprve vyvolala několik zásadních otázek. Bylo třeba

- rozmyslet, jakým způsobem zprostředkovat přenos informace mezi systémem a webovým rozhraním
- vybrat pro periferie konkrétní hardware kompatibilní s vybranou sběrnicí
- naimplementovat software pro jednotlivé periferie
- zkonstruovat jednotlivé ovládací a převodní prvky

Čidla a výstupy jsou připojena k jednotlivým periferiím. K propojení periferií a PC používáme sběrnicí μ Lan. μ Lan jsme zvolili, protože bez problémů umožňuje chod několika desítek zařízení najednou. Není pevně určen master, který by sběrnicí arbitroval. Všechny tyto přístroje jsou rovnocenné. Protokol nevyžaduje neustálý provoz všech periferií a funguje na poměrně velkou vzdálenost. Software je napsaný v jazyce C, poněvadž jsem byl závislý na již napsaných knihovnách. Kompletní popis návrhu webového monitorovacího systému najdeme v bakalářské práci Martina Boháčka, "Návrh software pro řízení a monitoring hydroponického systému."

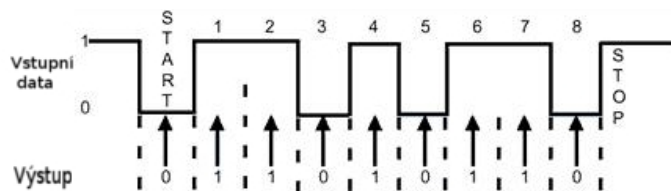
Kapitola 2

Protokol μ Lan

2.1 Fyzická vrstva - RS 485

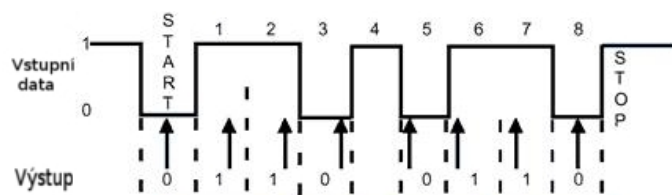
2.1.1 Obecný popis

Protokol RS-485 [4] se v průmyslu používá především pro přenos dat na velkou vzdálenost. Data se po něm přenášejí sériovým způsobem, bez nutnosti použití modulace. Sběrnice nemá vyvedený samostatný hodinový signál, který by sloužil pro synchronizaci vysílaných a přijímaných dat. Proto se veškeré zprávy na sběrnici posílají asynchronně. Pro přenos dat se používají pouze dva signálové vodiče. Lze ještě přidat třetí vodič, který určuje signálovou nulu. Veškeré řízení přenosu dat je prováděno programově, většinou na základě softwarového handshakingu. Přijímač a vysílač musí pracovat na stejné předem určené frekvenci, aby nedošlo k chybě přenosu rámce. Ta vzniká, když se vzorkuje na přechodu mezi dvěma bity. Názorně to můžeme vidět na obrázku 2.1 a 2.2 ¹.



Obrázek 2.1: Formát dat u RS485 bez chyby

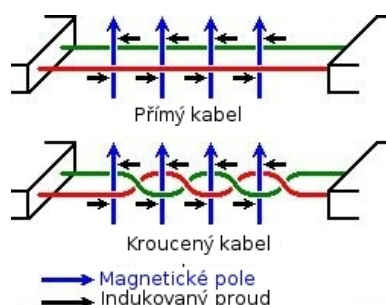
¹Zdroj: <http://www.root.cz/clanky/sberrnice-rs-422-rs-423-a-rs-485/>



Obrázek 2.2: Formát dat u RS485 s chybou

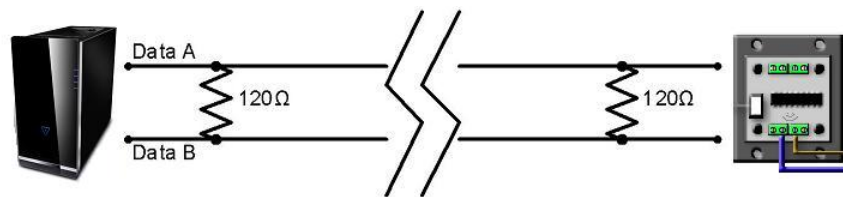
2.1.2 Charakteristika sběrnice

Přenos dat po dvoudrátovém vedení zjišťuje stav bitu podle rozdílu napěťového potenciálu mezi vodiči (diferenciální přenos). Výhoda dvoudrátového přenosu spočívá v možnosti použití kroucené dvojlinky, která umožňuje velkou přenosovou rychlost bez většího vyzařování signálu do okolí nebo naopak (menší zatížení dat šumem). Každý delší vodič totiž vysílá a přijímá elektromagnetické vlnění. U kroucené dvojlinky jsou oba nosiče informace vlněním ovlivněny stejně, takže přijímač může být velmi citlivý. Na rozeznání jiné logické úrovně stačí napětí 200mV. Na obrázku 2.3² je ukázán vliv magnetického pole na rušení. Přenos může být uskutečněn až na vzdálenost 1200m s přenos obou rychlostí 100 kbps. Na vzdálenost do 15 metrů můžeme docílit rychlosti až 10Mbps. Při realizaci RS-485 musíme na oba konce sběrnice připojit rezistory o velikosti přibližně 120Ω, což by mělo odpovídat impedanci kroucené dvojlinky. Typické zapojení vidíme na obrázku 2.4.



Obrázek 2.3: Vliv magnetického pole na kabel

²Zdroj: <http://www.root.cz/clanky/sberrnice-rs-422-rs-423-a-rs-485/>

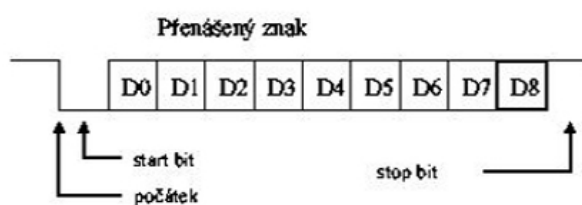


Obrázek 2.4: Zapojení kroucené dvojlinky

2.2 Popis sběrnice μ Lan

2.2.1 Formát dat

μ Lan [1] je 9ti bitový asynchronní komunikační protokol, který je postaven na fyzické vrstvě RS-485. Znak začíná start bitem, následuje sekvence devíti bitů a končí stop bitem (viz. obr. 2.5³). Znaky, které mají na tučně zvýrazněném bitu D8 hodnotu 1, jsou takzvané speciální znaky. V datovém rámci se mohou tyto znaky vyskytovat pouze na přesně definovaných pozicích. Zároveň slouží k označení přenášných dat a rozhodují o stavu obsazení sběrnice.



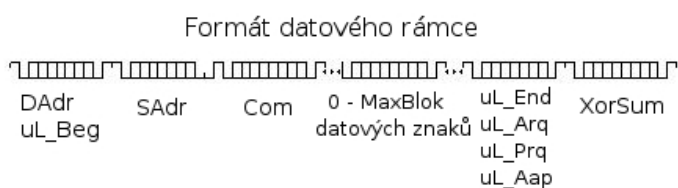
Obrázek 2.5: Formát znaku u μ Lanu

2.2.2 Datový rámec

Datový rámec se skládá ze sekvence znaků. První znak určuje adresu příjemce $DAdr$ nebo neadresný znak uL_Beg , který má bit D8 nastaven na 1. Zpráva je vyslána na sběrnici, každé zařízení ji obdrží a podle prvního znaku se rozhodne, zda-li zpráva byla určena pro něj. Následují znaky s adresou odesílatele $SAdr$ a kódem služby Com (command). Poté

³zdroj: http://cmp.felk.cvut.cz/pisa/ulan/dipl_kom.html

jsou odeslána data. Speciální bit D8 mají nastaven na 0. Není s nimi odesílána délka dat. Ta je určena maximální povolenou délkou. Maximální délka dat je stanovena podle konkrétní aplikace. Záleží na počtu periférií v síti, na nejdelsí přípustné odezvě mezi nimi a na rychlosti komunikace. Poslední znak dat má nastaven bit D8 a rozhodne o následujícím průběhu komunikace. Jsou zde čtyři možnosti. Rámec se ukamžitě ukončí - *uL_End*, zašle se kontrola o doručení rámce - *uL_Arq*, rámec je příjemcem okamžitě zpracováno - *uL_Prq* nebo se potvrdí přijetí a následně je bezprostředně zpracována *uL_Aap*. Jako poslední znak je odeslán kontrolní součet XorSum. Příklad rámce je uveden viz. obr 2.6 ³.



Obrázek 2.6: Datový rámec

2.2.3 Přenos dat

Zařízení odesílající zprávu musí čekat na uvolnění sběrnice. Jakmile je sběrnice volná připojí se na ni, odešle rámec a případně odešle potvrzení přijetí. Následně je sběrnice uvolněna. Pokud není detekována chyba, zpráva se označí jako doručená. Aby zařízení mohlo zprávu přijmout, musí mít nastavenou adresu a musí umět rozeznat znaky, které mají nastavený bit D8.

2.2.4 Arbitrace sběrnice

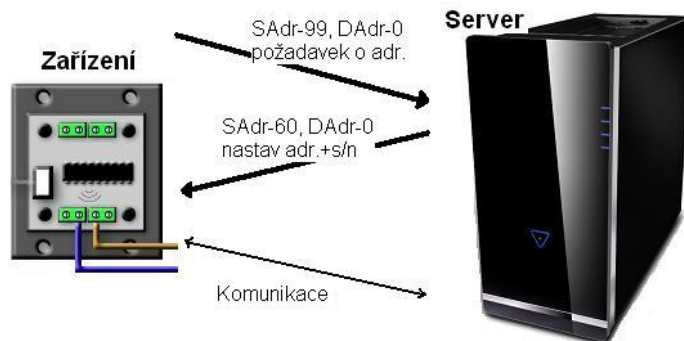
Všechna zařízení v μ Lan síti jsou si rovna. Než dostane zařízení právo vysílat, proběhne přípravná fáze (tzv. arbitrace). Při arbitraci se zaručí, že může vysílat pouze jedno zařízení. To vše je doplněno o rotování priority mezi přístroji. Je určen minimální čas, během kterého se nesmí přístroj připojit, v závislosti na adrese předcházejícího přístroje s oprávněním vysílat. Zařízení jsou schopna přijímat znaky, které mají nastaven bit D8. To vše znamená, že každý zná adresu zařízení (LAdr), které opustilo sběrnici. Každé zařízení si spočítá dobu, po kterou bude testovat klid na sběrnici na základě své adresy a adresy posledního zařízení podle vzorce:

$$t = (LAdr - Adr - 1) \text{ mod } 16 + 4.$$

Všechna zařízení, která chtějí přistoupit na sběrnici, na základě této doby začnou vysílat kombinaci prázdných intervalů a 000h znaků. Vyslání prázdného znaku znamená odpojení výstupního budiče a po tuto dobu je sběrnice v logické 1. Pokud zařízení vyše prázdný znak a sběrnice je v logické 0 znamená to, že se na sběrnici pokouší připojit jiné zařízení. Zařízení, které vysílalo prázdný znak, považuje sběrnici za obsazenou. Díky době vypočtené podle vztahu uvedeného výše, je kombinace vysílaných signálů nastavena tak, že pouze jedno zařízení najde při poslání prázdného znaku sběrnici v logické 1. To se opakuje ještě dvakrát, než přístroj získá sběrnici. Tento způsob arbitrace zajišťuje synchronizaci maximálně mezi 64 přístroji. Aktuální stav sběrnice se sleduje na bitu D7. Zde se definuje příznak obsazenosti *uLF_NB*. Po příjmu jakéhokoliv znaku se příznak vynuluje. K testu aktivity sběrnice slouží přístrojům funkce *uL_STROKE*. Ta sleduje, jestli mezi jejím voláním byl přijat alespoň jeden znak. Maximální prodleva během komunikace nesmí překročit dobu, za kterou se přenesou tři znaky.

2.3 Registrace do sítě

Funkce pro obsluhu dynamického přidělování adres jsou implementovány v knihovně *uldy_base*. Pro registraci do sítě je v protokolu vyhrazena adresa 99. Adresu mají přístroje defaultně přiřazenou. Na této adrese začne zařízení voláním funkce *send_sn* vysílat broadcastem žádost o přiřazení nové adresy. Žádost se vysílá s periodou tři vteřiny. Odesílaná zpráva má příznak *UL_CMD_NCS* (network control servise). Server, který tuto zprávu obdrží, na základě příkazu ví, že v prvním bytu zprávy je uložen upřesňující požadavek. V našem případě se jedná o *ULNCS_ADR_RQ* (address request), což je požadavek o adresu. Na následujících čtyřech bytech zprávy je uloženo sériové číslo (s/n). Server sériové číslo porovná s čísly již připojených přístrojů a na adresu 99 zařízení pošle zprávu s příznakem *UL_CMD_NCS*, ve které je na prvním bytu uložena *ULNCS_SET_ADDR* (set address). Na další bytech je vložena nová adresa, kterou server vygeneruje. Jediná možnost, jak zařízení pozná, že zpráva o nastavení adresy byla pro něj, je podle sériového čísla s/n. Dále již server komunikuje s přístrojem na nové adrese. Přístroj přijetím zprávy rozpozná příznak, nastaví si novou adresu a zjistí adresu serveru, čímž přestane zasílat žádost o přidělení adresy. Průběh komunikace při registraci do sítě viz. obrázek 2.7.



Obrázek 2.7: Průběh registrace - server-device

2.4 OI - Objektový slovník a jeho vytvoření

Objektový slovník je vrstva, ve které může mít každé zařízení definované proměnné viditelné v rámci μ Lan sítě. Přístup k nim nám zajišťuje knihovna *uloi_base*. Zde jsou naimplementované funkce, o kterých se zmíním v kapitole SDO.

Každý slovník musí mít nadefinovanou minimální sadu objektů (viz. tabulka 2.1) a dále může mít libovolný počet nově nadefinovaných.

objekt	funkce
ULOI_DOII	obsahuje vlastnosti objektů určených pro zápis
ULOI_DOIO	obsahuje vlastnosti objektů určených pro čtení
ULOI_QOII	obsahuje identifikační čísla objektů pro zápis
ULOI_QOIO	obsahuje identifikační čísla objektů pro čtení
ULOI_RDRQ	zjišťuje hodnoty objektů
ULOI_STATUS	pouze pro čtení
ULOI_ERRCLR	pouze pro zápis, nastavuje status na 0

Tabulka 2.1: Tabulka objektů

Definice každé proměnné musí nezbytně obsahovat několik věcí. Název, identifikační číslo, typ proměnné a informace o tom, jestli je proměnná pro zápis a/nebo čtení. Pro představu uvádím příklad:

```
ULOI_GENOBJDES(VARNAME,I_VARNAME,"type/additional info",
uloi_rdfnc,&oi_varname,uloi_wrfnc,&oi_varname).
```

Název je označení, kterým se proměnná prezentuje v rámci sítě. Identifikační číslo musí být unikátní. Při deklaraci je třeba, aby po sobě jdoucí objekty měly rostoucí identifikátory. Posloupnost může obsahovat mezery. Typ proměnné je řetězec, podle kterého se určuje skutečný datový typ. Do řetězce je možné vložit další pomocné informace. Tyto informace se vkládají za lomítko. Objekt je určen pro čtení, pokud má nadefinovaný ukazatel na data a funkci, která tato data čte. Knihovna μ Lan obsahuje funkce pro čtení základních datových typů. Celá čísla se znaménkem čte funkce *uloi_int_rdfnc* a hodnoty bez znaménka čte *uloi_uint_rdfnc*. Tyto funkce berou jako parametr ukazatel na data, která se mají přečíst. Podobně to je pro objekty určené pro zápis. Zde se volají funkce *uloi_int_wrfnc* a *uloi_uint_wrfnc*. Pokud chceme provést některé operace, jež se mají vykonat během zápisu nebo čtení, je možné objektu předat jméno vlastní funkce. V rámci této funkce naimplementujeme požadovaný kód a zavoláme jednu ze čtyř výše uvedených funkcí.

2.5 SDO - Service data object

Jedná se o potvrzované synchronní zprávy. To znamená, že zařízení odešle data pouze na základě požadavku vyslaného mastrem sběrnice. Synchronní zprávy se používají pro komunikaci s objektovým slovníkem, ze kterého čte v našem projektu výhradně monitorovací systém. Monitorovací zařízení schraňuje informace o adresách všech zařízení připojených v síti. Systém odešle dotaz na konkrétní objekt a čeká na potvrzení. Při tomto dotazu se volají funkce pro zápis nebo pro čtení, které byly popsány výše.

2.6 PDO - Process data object

PDO [2] jsou asynchronní nepotvrzované zprávy. Zařízení vysílají tyto zprávy nezávisle formou broadcastu. Z toho důvodu byl vytvořen nový příznak *UL_CMD_PDO*, který určuje asynchronnost datového rámce. Samotný rámec zprávy se nemění a zpracovává ji stejná funkce jako SDO. Pouze je definován formát dat, který je ukázán v tabulce 2.2.

			CID	data len	data	CID ...
1 byte	1 byte	1 byte	2 bytes LE	1 bytes	0-n bytes	2 bytes LE

Tabulka 2.2: Formát PDO zprávy

První tři byty jsou rezervovány pro případné pozdější rozšíření. Následuje identifikační číslo dat CID, délka dat a posléze samotná data.

Protože se zprávy posílají formou broadcastu, můžou je číst všechna zařízení připojená na sběrnici. Pokud objekt má nastavený nulový CID, nejsou data do PDO zprávy přidána. Velká výhoda je, že systém navržený na principu asynchronních zpráv může pracovat autonomě. K tomu, aby to tak bylo, jsem musel navrhnout, jakým způsobem si zařízení budou informace předávat.

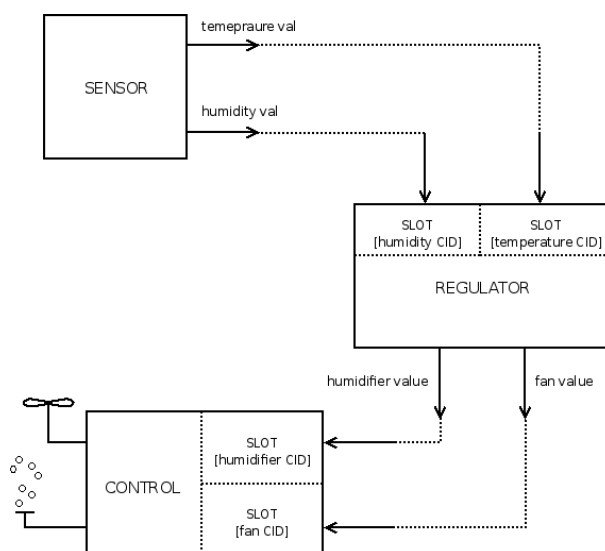
2.6.1 Princip CID-SLOT

Princip cid-slot je vlastně vytvoření mapy propojení producentů zpráv (CID) na spotřebitele zpráv (SLOT). Každé zařízení, které chce přijímat procesní zprávy, musí mít předem definované objekty typu SLOT, do kterých lze uložit CIDy dat, která budou ze sběrnice filtrována a dále zpracovávána. Na každou zprávu s příznakem *UL_CMD_PDO* je zavolána funkce *process_PDO* a *check_PDO*. *Process_PDO* vezme zprávu a vyseparuje všechny CIDy a data k nim náležící. Ty předá jako parametry funkci *check_PDO*, která zjistí jestli, jsou data pro zařízení zajímavá. Porovnájí se hodnoty uložené v objektech typu SLOT s vyseparovanými CIDy. Pro názorné vysvětlení zde uvedu příklad.

Máme zařízení *X* a *Y*. Zařízení *X* sbírá data, na základě kterých zařízení *Y* generuje graf. Zařízení *X* má definované objekty *VAL* a *VAL_CID*. V objektu *VAL* je uložena měřená veličina a v objektu *VAL_CID* je uloženo identifikační číslo veličiny unikátní v rámci celé sítě. Zařízení *X* tyto hodnoty posílá broadcastem na sběrnici jako PDO zprávu. Zařízení *Y* má nedefinovaný pouze objekt *VAL_SLOT*, jehož hodnota je prozatím nenastavena. Zařízení *Y* přijímá zprávy od zařízení *X*, ale zatím je nijak nezpracovává. Zmapování zařízení *X* a *Y* provedeme vložením hodnoty do objektu *VAL_SLOT*, která musí být rovna hodnotě objektu *VAL_CID*. Tímto způsobem můžeme namapovat větší množství periférií.

Příklad je znázorněn na obrázku 2.8⁴. Samotné namapování se dělá prostřednictvím webového rozhraní, které navrhl Martin Boháček. Screenshot webového rozhraní je připojen v příloze A.

Díky použití tohoto principu můžeme za běhu měnit architekturu celé sítě.



Obrázek 2.8: Princip CID-SLOT

2.6.2 Heart-beat

Speciální typ PDO zprávy je heart-beat. Je to jediná zpráva, kterou zařízení vysílá na konkrétní adresu. Zpráva serveru říká, že je zařízení připojeno a v provozu. Když má server ve své tabulce adresu, ze které nepřijde žádná zpráva typu heart-beat, server dynamických adres přístroj smaže ze svého seznamu. Perioda odesílání je jedna vteřina. CID heart-beatu je 1023 a místo dat se odesílá hodnota proměnné *status_val*.

⁴zdroj: Martin Boháček

Kapitola 3

Realizace systému

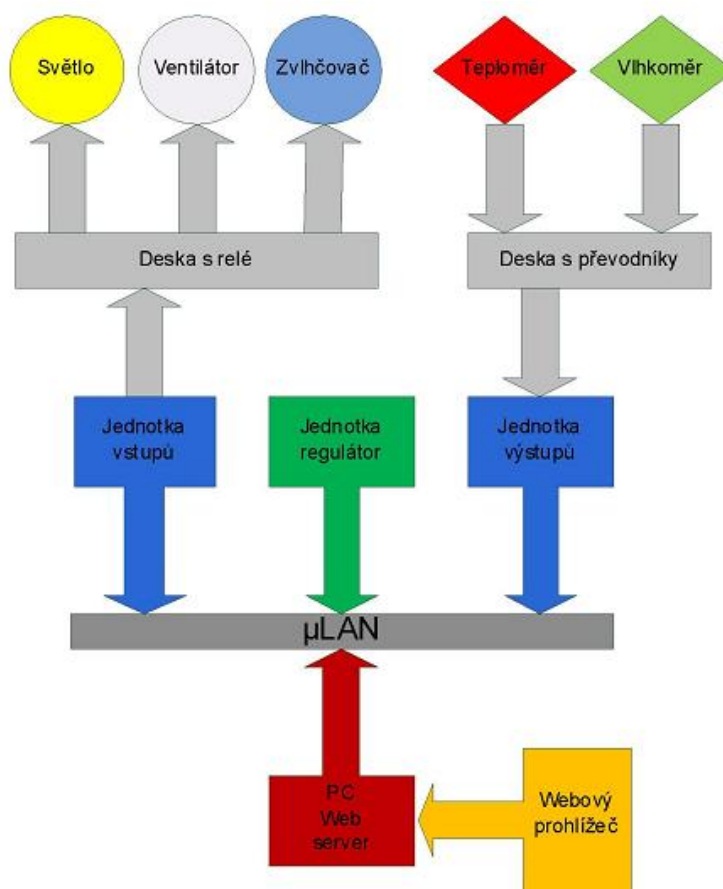
3.1 Blokové schéma a popis systému

Hydroponické akvárium je umístěno v dřevěném boxu. Rám akvária je vyroben ze smrkových hranolů a spojený kovovými spojnicemi. Spodní deska je složená ze dvou dřevěných desek. Přední stěnu tvoří ze $\frac{2}{3}$ 4mm silný makrolon a zbytek je vyplněn dřevěnou překližkou. Bočnice, víko a zadní kryt jsou vyrobeny z 5mm dřevěné překližky. Celkové rozměry boxu jsou 860x800x500mm.



Obrázek 3.1: Pěsticí box

Přívod elektrické energie zajišťuje prodlužovací kabel s pěti vstupy. V akváriu je nutné napájet ze sítě světlo, ventilátor, čerpadlo na cirkulaci živného roztoku, zvlhčovač vzduchu a zdroj nízkého napětí pro řídicí jednotku. Na jedné z bočnic boxu je umístěna rozvodní skříňka, ve které je umístěna elektronika. Zapojení elektroniky je patrné z blokového schématu na obrázku 3.2, tzn. napájecí zdroj pro mikroelektroniku, tři desky uLAD31 a deska se zapojením potřebných elektrických obvodů. Fotografie jednotlivých komponent jsou přiloženy v příloze A.



Obrázek 3.2: Blokové schéma systému

Na druhé bočnici boxu je v dolní části prostor na odtah vzduchu z akvária a v horní části je ventilátor pro přívod chladnějšího vzduchu z okolí. V akváriu je umístěna pěstební vana. Skládá se ze dvou částí. Větší část slouží jako rezervoár pro živný roztok. V menší části vany jsou položeny kostky z minerální vaty, ve které se nachází rostlina. Kořeny rostliny prorostou srkz vatu a popnou se po dně kudy protéká roztok. Vedle vany je umístěna nádoba s vodou, ve které je ponořen zvlhčovač.

3.2 Desky uLAD31 a software

Na desce je osazen procesor firmy Phillips typ LPC2148 [3]. Procesor je založen na 32bitové ARM7TDMI-S architektuře s 256kB vnitřní vysokorychlostní flash paměti. Má 128bitové vnitřní paměťové rozhraní a unikátní akcelerátor umožňující vykonávání 32-bitových instrukcí na nejvyšším možném kmitočtu procesoru. Procesor má 64 kontaktů, nízkou spotřebu elektrické energie. Samotný procesor je napájen napětím v rozsahu 1.65-1.95V. Výstupy jsou napájeny v rozmezí 3.0-3.6V. Pracuje na kmitočtu 60MHz. Dále obsahuje dvanáct externích přerušovacích kontaktů, čtyři kanály s 10bitovými analogově-digitálními převodníky a čtyřicet šest obecných vstupně/výstupních kontaktů. LPC2148 umí sériová rozhraní UART, rychlou I²C sběrnici a SPI rozhraní.

Pro svou realizaci jsem z procesoru využil dva analogově-digitální převodníky a tři výstupní kontakty. 10bitový A/D převodník převádí napětí v rozsahu 0-3.3V. Rozlišení převodníku se určí ze vztahu

$$A/D \text{ resolution} = \frac{3.3}{2^{10}} = 0.0032V.$$

Po zapojení napájení do desky se z paměti procesoru nahraje úvodní program, tzv. bootloader. Je to zaváděcí program, který spouští nahranou aplikaci. Doba, po kterou je bootloader spuštěn, je nastavena u konkrétních desek na 5 sekund. Tuto dobu můžeme prodloužit na neurčito použitím příkazu:

```
$ ul_sendhex -o0 -m0
```

V průběhu bootloderu je možné nahrát do paměti procesoru jinou aplikaci prostřednictvím utility *sendhex*. Do desky je možné aplikaci nahrát třemi způsoby. Pomocí μ Lan nebo sériové linky je proces nahrávání poměrně zdlouhavý. Voláním *make-load* se spustí utilita *ul_sendhex*. Sekvence příkazů, kterou spustíme zavoláním této funkce, je definována v souboru *config.omk*. Uvádím konzolový příkaz, kterým se nahrávání spustí.

```
$ make MOD=device_adr load-name-app
```

Lepší a hlavně rychlejší je možnost nahrát aplikaci přes rozhraní USB. Na to slouží utilita *usb_sendhex*, která ale musí být volána s právy superuživatele.

```
$ usb_sendhex -d 0xdead -i 0x2263 -s 0x20000 -l 0x20000 -e;  
$ sleep 2;  
$ usb_sendhex -d 0xdead -i 0x2263 -s 0x20000 -f binary name-app.bin  
$ usb_sendhex -d 0xdead -i 0x2263 -r;
```

První příkaz vymaže paměť procesoru. Pak počká a nahraje nový binární kód. V posledním kroku se procesor restartuje.

Architektura procesoru určila, že se aplikace pro jednotlivé desky budou vyvíjet v jazyce C. Pro překlad aplikace používám linuxový crosscompiler gcc-arm-elf. Compilátor vytvoří binární kód přeložitelný procesorem řady LPC21xx.

3.2.1 Hlavní smyčka (main loop)

V prvním cyklu si zařízení nejprve zažádá o adresu voláním funkce *send_sn*. Po obdržení adresy program přejde do druhého cyklu, kde začne posílat zprávu typu heart-beat funkcí *ul_send_command*. Další odesílání zpráv na sběrnici se odvíjí podle nastavení objektového slovníku přístroje. Pokud není nastavena perioda odesílání dat a/nebo CIDy proměnných, je odesílán pouze heart-beat. Pokud je perioda odesílání shodná s periodou heart-beatu, zprávy se sloučí a je odeslána pouze jedna zpráva. To má drobnou nevýhodu v tom, že se nevyhneme poslání heart-beatu broadcastem. Perioda a CIDy musí být nastaveny prostřednictvím webového rozhraní. Zařízení během své neaktivity přijímá všechny zprávy ze sběrnice a kontroluje, jestli není nějaká ze zpráv určena pro něj. Ve stejném časovém intervalu je přístroj schopný na požádání vyčítat data, která potřebuje webové rozhraní pro monitoring. Systém obsahuje tři typy zařízení, které odesílají různá data a mají jinak definované objektové slovníky.

3.2.2 Typy zařízení

3.2.2.1 Zařízení senzor

Senzor je vstupní zařízení, které sbírá vstupní data potřebná pro regulaci. V případě hydroponického akvária měří teplotu a vlhkost. Na analogově-digitální převodníky procesoru jsou přivedena napětí odpovídající měřené veličině. Pro získání dat z převodníku slouží funkce *read_ADC*. Funkce je volána tolikrát, z kolika naměřených hodnot chceme

udělat průměr. Funkci musíme předat parametr s datovou strukturou, do které je výsledek uložen. Před začátkem převodu se zavolá funkce *init_ADC*, kterou vybereme požadovaný A/D převodník. Na výběr slouží registr *PINSEL1*. Zde má procesor uloženou informaci o tom, pro jaký účel je který vývod. Senzor využívá A/D převodníky na adresách P0.28 a P0.29. Před každým převodem je nutné nastavit registr *AD0CR* (kontrolní registr). V něm je řečeno, který z převodníků bude použit ke vzorkování, je zde nastavena frekvence hodin, počet cyklů potřebných k provedení konverze, nastavení operačního módu a bit pro spuštění konverze. Výsledek a informace o konci převodu se ukládají do *ADGDR* (AD global data register). Mód může být buď automatický (*BURST* = 1) nebo softwarově řízený (*BURST* = 0). Automatický mód se obsluhuje formou přerušení. Po dokončení konverze je vygenerováno přerušení, které se odchytil a výsledek převodu je zpracován. Senzor využívá softwarově řízený převod. Při něm se na začátku konverze nastaví *START* bit v *AD0CR* a konec se sleduje na *DONE* bitu *ADGDR*.

Před uložením navzorkované hodnoty do objektového slovníku je volána funkce *adc2oi*, která přepočítává zprůměrovanou hodnotu z převodu na reálnou veličinu a ukládá ji do objektového slovníku. Velikost veličiny se spočítá s přesností na jedno desetinné místo. Protože objekt pro uložení veličiny je celočíselného typu, vynásobí se hodnota 10. Z toho důvodu je u objektů uvedena informace o pozici desetinné čárky. Využil jsem možnost napsat doplňující informace za lomítko u řetězce určující datový typ při definici objektu. V tabulce 3.1 uvádím seznam objektů definovaných pro toto zařízení.

objekt	popis
TEMPERATURE	aktuální hodnota teploty
TEMPERATURE_CID	identifikační číslo objektu TEMPERATURE v rámci sítě
HUMIDITY	aktuální hodnota vlhkosti
HUMIDITY_CID	identifikační číslo objektu HUMIDITY v rámci sítě
PERIOD	perioda posílání PDO zpráv

Tabulka 3.1: Tabulka objektů zařízení senzor

3.2.2.2 Zařízení control

Control je zařízení digitálních výstupů. Ovládá spínání akčních členů systému: světlo, ventilátor a zvlhčovač. Obsloužení zapínání a vypínání akčních členů je vyřešeno v rámci *uloj_wrfncí*. Pokud je v objektech *FAN*, *HUMIDIFIER* a *LIGHT* uložena hodnota 1, nastaví se pomocí funkce *SET_OUT_PIN(OUT_PORT,P1_XX_DEV)* výstup do logické 1. Při jakékoli jiné hodnotě se volá funkce *CLR_OUT_PIN(OUT_PORT,P1_XX_DEV)*, která logickou úroveň resetuje. Tyto funkce jsou definovány v headru *system_def.h*. V tomto souboru je nutné nastavit ovládané vývody jako výstupní.

Přístroj přijímá zprávy ze sítě a reaguje pouze na ty, které obsahují CIDs rovné hodnotám uložených ve slotech. *FAN_CID*, *HUMIDIFIER_CID* a *LIGHT_CID* nejsou použity a jsou definovány pro budoucí použití. Informace o tom, kdy a v jakém stavu se nachází výstupní proměnná, se může hodit jako zpětná vazba pro kontrolu chodu řízení. V tabulce 3.2 je uveden seznam objektů tohoto zařízení.

objekt	popis
FAN	aktuální stav ventilátoru
FAN_CID	identifikační číslo objektu FAN v rámci sítě (nepoužívá se)
FAN_SLOT	hodnota určuje CID objektu řídící ventilátor
HUMIDIFIER	aktuální stav zvlhčovače
HUMIDIFIER_CID	identifikační číslo objektu HUMIDIFIER v rámci sítě (nepoužívá se)
HUMIDIFIER_SLOT	hodnota určuje CID objektu řídící zvlhčovač
LIGHT	aktuální stav světla
LIGHT_CID	identifikační číslo objektu LIGHT v rámci sítě (nepoužívá se)
LIGHT_SLOT	hodnota určuje CID objektu řídící světlo
PERIOD	perioda posílání PDO zpráv

Tabulka 3.2: Tabulka objektů zařízení control

3.2.2.3 Zařízení regulátor

Regulátor přijímá od zařízení sensor PDO zprávy, které obsahují *TEMP_CID* a *HUM_CID* s aktuální teplotou a vlhkostí. Na základě nastavení hodnot požadované vlhkosti a teploty (*WANT_TEMP*, *WANT_HUM*) regulátor vypočítá potřebný akční zásah. Tzn. že na sběrnici odešle PDO zprávu obsahující CIDs *FAN_CID*, *HUMIDIFIER_CID* a *LI-*

GHT_CID, ve kterých je uložen akční zásah pro jednotlivé výstupy. Pro řízení systému jsem použil dvoustavový regulátor s hysterezí. Pokud je požadovaná hodnota veličiny nastavena na hodnotu x , akční zásah se provede až ve chvíli, kdy bude skutečná hodnota $x \pm h$. Spojité řízení nebylo v systému vyřešeno. Světlo není řízeno na základě vstupních informací. Zapíná se v periodě shodné s periodou světelného záření Slunce v letním období (tzn. 16 hodin světla a 8 hodin tmy). V tabulce 3.3 uvádím seznam objektů s popisem.

objekt	popis
FAN_CID	identifikační číslo PDO objektu generujícího akční zásah - zapnutí nebo vypnutí ventilátoru
HUMIDIFIER_CID	identifikační číslo PDO objektu generujícího akční zásah - zapnutí nebo vypnutí zvlhčovače
LIGHT_CID	identifikační číslo PDO objektu generujícího akční zásah - zapnutí nebo vypnutí světla
TEMPERATURE_SLOT	hodnota určuje CID objektu teplota, vstup regulátoru
HUMIDITY_SLOT	hodnota určuje CID objektu vlhkost, vstup regulátoru
PERIOD	perioda posílání PDO zpráv
WANT_TEMP	nastavení požadované teploty v systému
WANT_HUM	nastavení požadované vlhkosti v systému

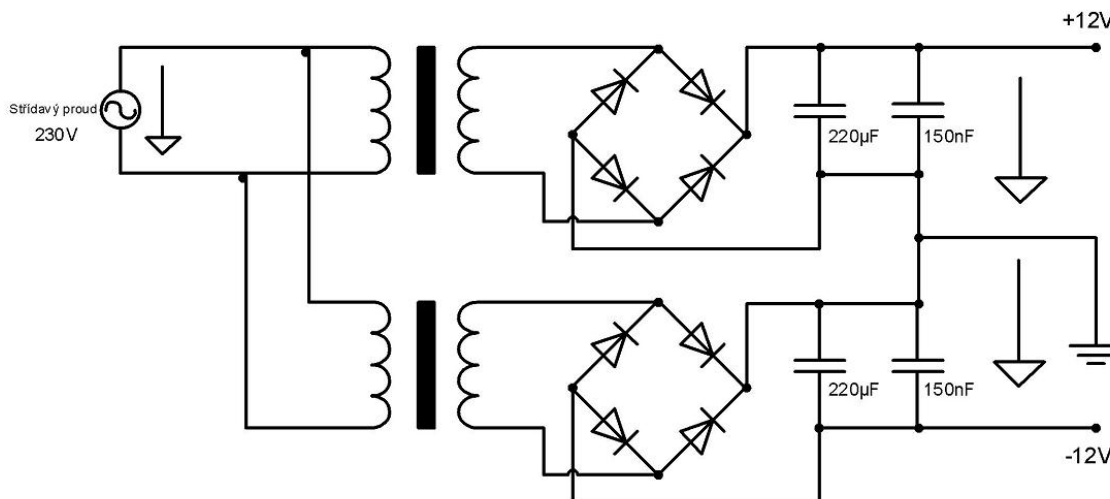
Tabulka 3.3: Tabulka objektů zařízení regulátor

3.3 Návrh hardwaru

3.3.0.4 Napájení systému

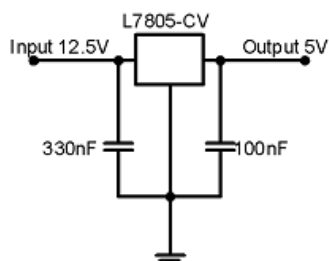
Celé akvárium je napájeno ze sítě střídavého napětí 230V, 50Hz. Pro napájení mikroelektroniky a elektrických obvodů jsem vyrobil symetrický zdroj stejnosměrného napětí, které je potřeba hlavně kvůli napájení operačních zesilovačů. Zdroj se skládá ze dvou transformátorů, usměrňovače a kondenzátorů pro filtraci nízkofrekvenčního a vysokofrekvenčního šumu. Napětí se transformuje z 230V na 10V střídavého napětí, které je přivedeno na usměrňovač tvořený Gratzovým můstkem a dvěma kondenzátory. Díky kapacitě použitých kondenzátorů je na výstupu usměrňovače $\pm 12V$ stejnosměrného napětí.

Celé zapojení můžeme vidět na obrázku 3.3. Světlo, ventilátor a zvlhčovač jsou zapojeny do sítě 230V.



Obrázek 3.3: Zapojení zdroje a usměrňovače

Desky uLAD31 jsou napájeny 5V prostřednictvím USB konektoru. Pro napájení desek jsem použil stabilizátor L7805-CV, zapojený podle schématu na obrázku 3.4¹.



Obrázek 3.4: Stabilizátor napětí pro napájení desek uLAD31

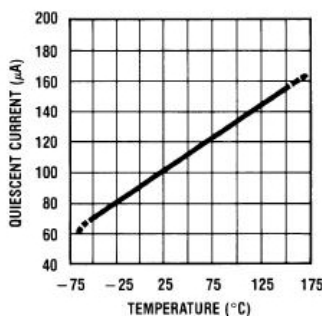
3.3.0.5 Teplotní čidlo

Pro měření teploty jsem použil integrovaný obvod LM35. Výstupní napětí senzoru je lineárně úměrné teplotě ve stupních Celsia. Rozlišení čidla je 10mV/°C. Senzor nepotřebuje externí kalibraci a jeho nelinearita je maximálně $\frac{1}{4}$ °C. Garantovaná přesnost

¹http://www.datasheetcatalog.com/datasheets_pdf/L/7/8/0/L7805CV.shtml

čidla je 0.5°C . V plném rozsahu měří teplotu v rozmezí -55°C - 150°C . Je možné ho napájet v rozsahu 4V - 30V . Proudový odběr čidla je $60\mu\text{A}$. Díky malému odběru proudu se čidlo zahřívá minimálně. Senzor je umístěn v plastickém pouzdře.

Závislost klidového proudu na teplotě je znázorněna na obrázku 3.5 ².



Obrázek 3.5: Závislost klidového proudu na teplotě u senzoru LM35

Na analogově-digitální převodník zařízení senzor jsem přivedl napětí z výstupu integrovaného obvodu LM35. Vzhledem k tomu, že rozlišení převodníku jsou 3mV , jsem schopný určit teplotu s přesností na třetiny $^{\circ}\text{C}$. Velikost rezistoru je vypočítána podle vztahu:

$$R = \frac{U}{I_{LOAD}} = \frac{12.5}{50 \cdot 10^{-6}} = 250\text{k}\Omega,$$

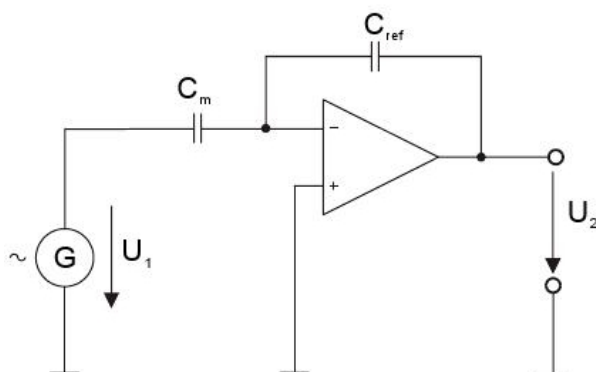
kde velikost I_{LOAD} je uvedena v datasheetu. V tabulce 3.4 uvádím výstupní napětí senzoru v závislosti na teplotě.

Napětí U_{out}	teplota
1500mV	150°C
250mV	25°C
-550mV	-55°C

Tabulka 3.4: Tabulka závislosti výstupního napětí senzoru na teplotě

²http://www.datasheetcatalog.com/datasheets_pdf/L/M/3/5/LM35.shtml

Aby senzor byl použitelný v hydroponickém akváriu, musel jsem před analogově-digitální vstup zapojit převodník kapacita/napětí. Zapojení využívá zpětnovazební zapojení operačního zesilovače. Na invertující vstup je přivedena měřená kapacita a ve zpětné vazbě je zapojen referenční kondenzátor. Princip zapojení viz. obrázek 3.8 ⁴.



Obrázek 3.8: Princip převodníku C/U

Pro napětí U_1 a U_2 platí,

$$\frac{U_1}{\frac{1}{j\omega C_m}} = -\frac{U_2}{\frac{1}{j\omega C_{ref}}}$$

a po úpravě,

$$U_2(j\omega) = -U_1(j\omega) \frac{C_m}{C_{ref}}$$

z toho vyplývá, že kapacita je přímo úměrná amplitudě napětí. Zpětnovazební odpor snižuje stejnosměrné zesílení, aby se operační zesilovač nedostal do saturace integrací vlastního offsetu. Abych mohl přivést výstup z C/U převodníku na A/D převodník, signál jsem musel usměrnit a filtrovat od šumu.

Převodník je potřeba napájet zdrojem střídavého napětí. Napětí z transformátoru je nepoužitelné, protože kmitočet zdroje neleží v rozmezí, kdy vlhkostní čidlo funguje. Nejlepší možností byla použít zapojení Wien-Bridgeova oscilátoru. Oscilátor generuje střídavé napětí bez vstupního zdroje. Zapojení se skládá ze čtyř rezistorů, dvou kondenzátorů a operačního zesilovače. Pro zesílení signálu jsou na invertujícím vstupu zapojeny dva rezistory. Na neinvertující vstup jsou připojeny dvě větve RC článků. Hodnoty kapacity a rezistance se u obou větví se nesmí lišit.

⁴zdroj:measure.feld.cvut.cz/groups/edu/sz/pdfs/uloha8.pdf

Frekvence oscilátoru se vypočítá podle vztahu

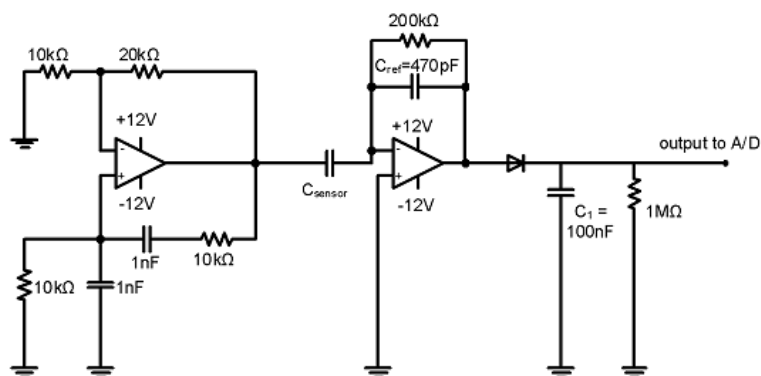
$$f = \frac{1}{2\pi RC}$$

V případě zapojení použitého v akváriu je frekvence

$$f = \frac{1}{2\pi 10^4 10^{-9}}$$

$$f = 15.92 \text{ kHz}$$

Kombinací dvou výše uvedených zapojení dostaneme kompletní zapojení vlhkostního čidla. Ke konstrukci jsem použil nominální hodnoty součástek viz. obrázek 3.9. Operační zesilovač jsem vybral UA741.



Obrázek 3.9: Zapojení vlhkostního čidla

Protože Wien-Bridgeův oscilátor generuje střídavé napětí o amplitudě, která je rovna napájecímu napětí operačního zesilovače, musel jsem převodník C/U zkalibrovat tak, aby výstupní napětí bylo v měřitelném rozsahu A/D převodníku a aby po převodu zůstalo optimální rozlišení. Amplituda střídavého signálu je $U_1=11,8V$, rozsah kapacity vlhkostního senzoru je 118pF pro vlhkost 30% a 137pF pro vlhkost 80%. Referenční kapacitu jsem zvolil 470pF.

Vycházím ze vztahu odvozeného výše,

$$|U_2(j\omega)| = |U_1(j\omega)| \frac{C_{sensor}}{C_{ref}}$$

$$U_{2min} = 11,8 \cdot \frac{118}{470}, \quad U_{2max} = 11,8 \cdot \frac{137}{470}$$

$$U_{2min} = 2.832V, \quad U_{2max} = 3.435V$$

U_{2min} a U_{2max} je amplituda výstupního signálu z převodníku. Když oba signály usměrníme diodou a vyfiltrujeme kondenzátorem, dostaneme na výstupu napětí o velikosti amplitudy signálu zmenšené o úbytek napětí na diodě. Rozdíl napětí stačí vydělit rozlišením A/D převodníku a rozdílem krajních hodnot vlhkostí. Vybití kondenzátoru C_2 je zajištěno paralelně připojeným rezistorem.

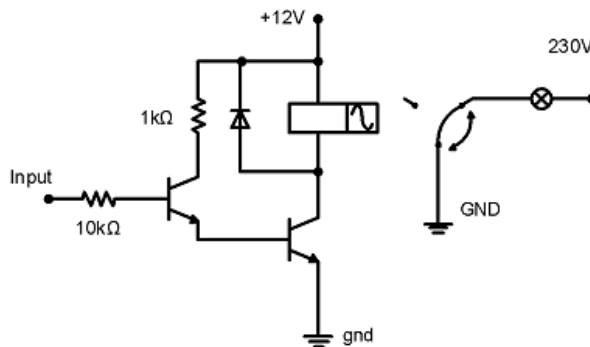
$$sensitivity = \frac{U_{2max} - U_{2min}}{AD_{resolution} \cdot (hum_{max} - hum_{min})} = \frac{3.434 - 2.832}{0.003 \cdot 50}$$

$$sensitivity = 4.01 \quad accuracy = \frac{1}{sensitivity} = 0.25$$

Převrácené hodnota citlivosti nám řekne, s jakou přesností jsme schopni vlhkost určit.

3.3.0.7 Jednotka digitálních vstupů/výstupů

Všechny výstupní jednotky systému (zátěže) jsou napájeny ze sítě 230V. K jejich spínání jsem použil zapojení s elektromagnetickým spínačem čili relé. Toto řešení pro ovládání výstupů nám neumožňuje systém řídit jinak než hysteretně. Schéma zapojení viz. obrázek 3.10.



Obrázek 3.10: Zapojení pro ovládání výstupů

Pokud připojíme na vstup systému logickou 1, projde přes předřadný odpor o velikosti $10k\Omega$ proud přes bázi tranzistoru tzn., že tranzistor sepne. Sepnutí prvního tranzistoru zapříčiní sepnutí i toho druhého. Kolektorem druhého tranzistoru proteče proud dostatečný pro sepnutí elektromagnetického spínače. Pro konstrukci obvodu jsem vybral relé od firmy Takamisawa JS12N-K⁵ a tranzistor BC547⁶.

⁵<http://us.100y.com.tw/PNoInfo/52404.htm>

⁶http://www.datasheetcatalog.com/datasheets_pdf/B/C/5/4/BC547.shtml

Kapitola 4

Závěr

Práce na projektu pro mě byla ohromným přínosem. Při realizaci jsem se seznámil s architekturou procesoru ARM a s jazykem C. Psaním své práce jsem získal zkušenost, jak vytvořit obsáhlejší dokument v sázecím systému L^AT_EX s pomocí manuálových stránek¹. Vývojem softwaru jsem se naučil pracovat s verzovacím systémem Git. Pro projekt byl vytvořen repozitář běžící na školním serveru. Dále jsem získal zkušenosti s pokročilým makeovacím systémem OMK, který je svou strukturou uživatelsky přátelštější než klasické Makefily. Při tvorbě hardwaru jsem nabyl spoustu praktických znalostí z okruhu sestavování a pájení elektrických obvodů. Příjemné byly i truhlářské práce nutné k sestavení boxu, ve kterém rostliny porostou.

Projekt je zajímavý svou pestrostí spočívající v aplikaci různých technologií. Nejvíce náročný byl návrh a implementace komunikace mezi zařízeními. Nejprve jsem si musel prostudovat veškerou dostupnou dokumentaci k protokolu μ Lan a seznámit se se zdrojovými kódy. Přínosem této práce do projektu μ Lan je rozšíření protokolu o asynchronní zprávy (PDO).

Byl vytvořen systém, který má následující vlastnosti:

- možnost připojení velkého množství periférií, které jsou schopny mezi sebou komunikovat bez potřeby nadřazené řídicí jednotky na poměrně velkou vzdálenost
- umožňuje snadnou konfiguraci sítě pouhou změnou dat u objektů definovaných v objektovém slovníku prostřednictvím editace v uživatelsky příjemném http rozhraní
- programy běžící v referenčních zařízeních lze modifikovat pouhou změnou definice objektového slovníku ve zdrojovém kódu a tak vytvořit nové periferie.

¹zdroj: <http://www.cstug.cz/>

Spolu s navazující prací kolegy Martina Boháčka byl vytvořen univerzální řídicí systém, který lze snadno použít pro řízení velkého množství tříd technologických procesů, nejen hydroponického akvária. Velká výhoda návrhu je v tom, že všechna zařízení se do sítě připojují formou plug&play.

Myslím si, že jsem splnil všechny cíle vytyčené v úvodu této práce a že jsem dodržel formální zadání bakalářské práce.

Na projektu hydroponického akvária bych rád pokračoval a dále ho zdokonaloval. Pro řízení bych v budoucnu chtěl zkusit použít řídicí jednotku PimX1 navrženou firmou Pikron. Rád bych také zdokonalil měření jednotlivých veličin použitím přesnějších metod měření. Projekt by se dal velice snadno rozšířit o další vstupy a výstupy. Například důležitý faktor, který je potřeba při řízeném hydroponickém růstu rostlin kontrolovat, je koncentrace minerálů a hnojiv v živném roztoku. Z hlediska uživatele je určitě zajímavé monitorování růstu webovou kamerou.

Mé přání je, aby tento projekt nezapadl a aby byl v praxi využitelný.

Literatura

- [1] Pavel Píša, http://cmp.felk.cvut.cz/pisa/ulan/dipl_kom.html, stránka s dokumentací k μ Lanu
- [2] Pavel Píša, <http://ulan.wiki.sourceforge.net/uLCN>, wikipedie k projektu μ Lan
- [3] Kolektivní autorství, <http://www.datasheetcatalog.com/>, webová stránka se všemi potřebnými datasheety
- [4] Kolektivní autorství, <http://www.root.cz/clanky/sbornice-rs-422-rs-423-a-rs-485/>, článek o sběrnici RS-485
- [5] Katedra měření, measure.feld.cvut.cz/groups/edu/sz/pdfs/uloha8.pdf, laboratorní úloha předmětu Zpracování signálu
- [6] Kolektivní autorství, <http://www.cstug.cz/>, manuál k sázecímu systému L^AT_EX
- [7] Pavel Herout, *Učebnice jazyka C*, nakladatelství KOPP, České Budějovice, říjen 1994, III. upravené a rozšířené vydání
- [8] Kolektivní autorství, <http://en.wikipedia.org/>, světová encyklopedie
- [9] Kolektivní autorství, <http://rttime.felk.cvut.cz/>, interní stránky Real-time System Group

Příloha A

Obrázky

Správa konfigurace

Hydroponie web interface

Home >> Regulator >> Konfigurace zařízení

Dostupná zařízení
Controller
Regulator
Sensor

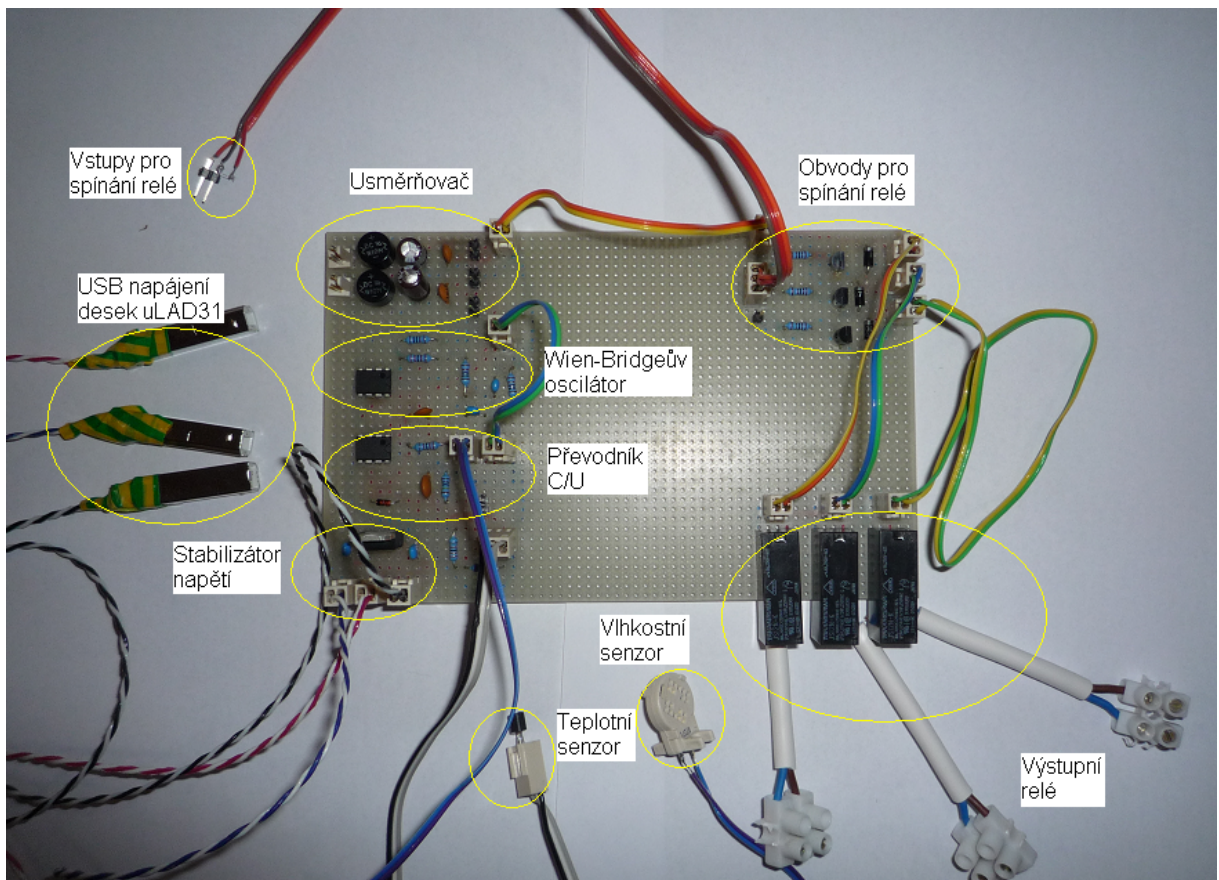
Regulator (konfigurace zařízení)

Název zařízení: Regulator
Uložít

Systemový název	Uživatelský název	Nastavit hodnotu	Aktuální hodnota	Barva	Spol. graf	Detail
FAN_CID	FAN_CID	40	40			
HUMIDIFIER_CID	HUMIDIFIER_CID	30	30			
HUMIDITY_SLOT	HUMIDITY_SLOT	Sensor - HUMIDITY_CID	20			
LIGHT_CID	LIGHT_CID	50	50			
PERIOD	PERIOD	5	5			
TEMPERATURE_SLOT	TEMPERATURE_SLOT	Sensor - TEMPERATURE_CID	10			
WANT_HUM	WANT_HUM	0	0	Zelená		Detail
WANT_TEMP	WANT_TEMP	400	400	Tmavě zelená		Detail

Uložít

Obrázek A.1: Webové rozhraní



Obrázek A.2: Deska s elektronikou



Obrázek A.3: Komponenty

Příloha B

Software

odkaz na aktuální verzi softwaru: <http://rtime.felk.cvut.cz/gitweb/hydro.git>