VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ŘÍZENÍ ZÁVLAHOVÉHO SYSTÉMU
CONTROL OF IRRIGATION SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Marek Farba**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Zdeněk Bradáč, Ph.D.**
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**
Ústav automatizace a měřicí techniky

Student: Bc. Marek Farba ***ID:*** 164262
Ročník: 2 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Řízení závlahového systému

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a vytvořte elektronický systém pro řízení závlahového systému. Navrhněte systém založený na vhodné mikrokontrolérové platformě s komunikací přes Ethernet/GSM. Navrhněte a zrealizujte elektronický systém, který vybavte nezbytnými rozhraními, senzory a akčními členy. Vytvořte nezbytné programové vybavení včetně vizualizace. Otestujte a demonstrujte plnou funkčnost.

1. Zpracujte literární rešerši.
2. Navrhněte a popište vlastní koncepci systému.
3. Realizujte vlastní obvodové řešení, realizujte DPS, oživte.
4. Vytvořte funkční programové vybavení.
5. Ověřte a demonstrujte funkčnost systému a vyhodnoťte výsledky.

DOPORUČENÁ LITERATURA:

1. ZEZULKA, F. Prostředky průmyslové automatizace. VUTIUM. VUTIUM. Brno: VUTIUM, 2004. 176 s. ISBN: 80-214-2610-1.

2. Dle doporučení vedoucího práce.

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 14.5.2018

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.
Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

Abstrakt

Témou diplomovej práce je riadenie závlahového systému. V úvode sa nachádza teoretický rozbor zavlažovacích jednotiek, navrhnutého systému a použitých programov. Jadrom práce je popis vyvinutého software riadiaceho programu a vizualizácie . Následne je popísaný postup návrhu dosky plošných spojov. Záver práce tvorí overenie funkčnosti zavlažovacieho systému a zhrnutie dosiahnutých výsledkov.

Kľúčové slová

Elektronický zavlažovací systém, riadiaca jednotka, Raspberry Pi, python, webová stránka.

Abstract

This thesis deals with controlling the irrigation system. In the introduction there is a theoretical analysis of the irrigation units, the designed system and the programs used. The core of the paper is a description of the software development and visualization software developed. Subsequently, the process of designing printed circuit boards is described. The conclusion of the work is to verify the functionality of the irrigation system and the summary of the results achieved.

Keywords

Electronic irrigation system, control unit, Raspberry Pi, python, web page.

Bibliografická citácia

FARBA, M. *Řízení závlahového systému* . Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 93 s. Vedoucí diplomové práce doc. Ing. Zdeněk Bradáč, Ph.D..

Prehlásenie

Prohlašuji, že svou semestrální práci na téma „Lineární jednotka s krokovým motorem“ jsem vypracoval samostatně pod vedením vedoucího semestrální práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brne dňa: ....

podpis autora

Poďakovanie

Ďakujem vedúcemu diplomovej práce doc. Ing. Zdeňku Bradáčovi, Ph.D. za odborné vedenie a cenné rady pri vypracovaní tejto práce.

V Brne dňa: ....

podpis autora

Obsah

Zoznam obrázkov

1 ÚVOD

Predmetom tejto práce je navrhnúť a vytvoriť elektronický systém pre riadenie závlahového systému. Tento systém bude založený na vhodnej mikrokontrolérovej platforme s komunikáciou cez Ethernet alebo GSM. Navrhnutý elektronický systém bude vybavený potrebnými rozhraniami, senzormi a akčnými členmi. Súčasťou elektronického závlahového systému bude nevyhnutné programové vybavenie vrátane vizualizácie. Súčasťou práce bude demonštrácia plnej funkčnosti programu a vizualizácie.

Teoretická časť práce popíše internetový prieskum a literárny rešerš. Následne sa zameria na koncepciu závlahového systému.

Praktická časť práce bude obsahovať navrhnuté elektronické zapojenie elektronického závlahového systému, postup tvorby DPS a určenie hodnôt použitých súčiastok. Ďalej popíše vytvorené programové vybavenie a vizualizáciu. V poslednej časti budú zhrnuté dosiahnuté výsledky a demonštrácia funkčnosti celého programu.

2 TEORETICKÝ ROZBOR

Úvodná časť kapitoly obsahuje základné informácie o zavlažovacích systémoch a podobnejšie sa zaoberá elektronickými komponentmi týchto systémov. Ďalšia časť kapitoly sa venuje návrhu konceptu elektronického zavlažovacieho systému a výberu vhodného mikrokontroléru na riadenie elektronickej závlahy. Na záver sú vysvetlené princípy požitých zberníc a stručne charakterizované programy a programovacie jazyky, ktoré boli pri tvorbe práce využité.

2.1 Zavlažovanie

Počas môjho štúdia na vysokej škole sa brat zamestnal v záhradníckej firme. Ako to tak býva, časom začal niektoré veci z práce realizovať aj na našom trávnatom dvore. Najprv to bolo častejšie kosenie, následne kompletné presadenie trávnika až sa prepracoval k manuálne ovládanej závlahe pomocou ventilov.

Pri pestovaní krásneho trávniku je dôležitých niekoľko faktorov. Ako prvé je potrebné mať vhodný substrát (zeminu), následne zvoliť vhodný typ trávniku. Počas rastu je taktiež v prípade potreby použiť hnojivo ale nevyhnutnou súčasťou je výdatná a pravidelná závlaha.

Použitím automatizovanej závlahy sa urýchli rast novo vysiateho trávniku a zaistia sa mu ideálne zavlažovacie podmienky. Taktiež je možné optimalizovať spotrebu vody.

2.2 Druhy zavlažovacích systémov

Na trhu je momentálne už veľmi veľa závlahových systémov s rôznymi funkciami, senzormi a príslušenstvom. V tejto podkapitole sa pokúsim o stručné zhrnutie hlavných vlastností dostupných zavlažovacích systémov.

Ktorékoľvek zavlažovacie hodiny vám predstavu o automatickom zavlažovacom systéme naplnia. Či už budete doma alebo nie, nebudete sa musieť zaoberať tým, či záhrada zostane bez vody, pretože všetky zavlažovacie jednotky sú veľmi spoľahlivé. [1]

2.2.1 Riadiace jednotky

Ovládacie jednotky spoľahlivo ovládajú celý systém automatickej závlahy pomocou elektromagnetických ventilov, ktoré otvárajú a uzatvárajú jednotlivé zóny - sekcie. [1]

Ovládacia jednotka má za úlohu riadiť celé zavlažovanie bez akejkoľvek obsluhy. Pretože vzhľadom ku kapacite čerpadla nemožno zavlažovať celú záhradu, park, alebo ihrisko naraz, rozdeľuje sa zavlažovaná plocha spravidla do viacerých zavlažovacích sekcií, ktoré sú zavlažované postupne. Interiérové zavlažovacie hodiny sa zvyčajne umiestňujú v technickom zázemí domu, alebo v pivnici, v garáži či v

záhradnom domčeku. Exteriérové hodiny sú vhodné pre montáž do vonkajšieho prostredia. [2]

Dôležité parametre pri výbere riadiacej jednotky sú napríklad :

- prostredie (interiér/exteriér);
- kompatibilita (kombinovanie ventilov a čidiel rôznych značiek);
- počet a typ vstupov pre čidla;
- maximálny počet riadených sekcií (ventilov);
- rozšíriteľnosť modulmi (meteo stanica, zrážkomer, anemometer);
- spôsob ovládania (tlačidlá, PC, smartfón);
- konektivita (WiFi, Ethernet);
- užívateľské rozhranie (jazyk, zložitosť, upraviteľnosť);
- prispôsobivosť (možnosť upravovať čas a typ závlahy);
- spoľahlivosť;
- bezpečnosť;
- cena.

Pri výbere riadiacej jednotky je potrebné taktiež skontrolovať, či sú pre danú riadiacu jednotku dostupné kompatibilné čidlá a doplňujúce rozšírenia. Nie každé čidlo vlhkosti alebo napríklad teploty má rovnaké napájacie napätie, typ konektoru alebo spôsob komunikácie. Taktiež je potrebné sa uistiť, či má riadiaca jednotka dostatočný počet vstupov pre potrebné čidlá a dostatočný počet výstupov pre riadené zavlažovacie sekcie.

Niektoré, väčšinou drahšie a komplexnejšie, riadiace jednotky ponúkajú možnosť komunikácie s meteo stanicami, zrážkomermi alebo napríklad anemometrami. Komunikácia prebieha buď klasicky pomocou vodičov alebo bezdrôtovo. Niektoré zavlažovacie systémy ponúkajú dokonca možnosť komunikovať so servermi výrobcu, ktoré v spolupráci s verejnými meteo stanicami komplexne plánujú všetky parametre zavlažovania.

Najjednoduchší a najčastejší spôsob ovládania elektronickej závlahy je pomocou tlačidiel a prepínačov umiestnených priamo na riadiacej jednotke. Mnohé sofistikované zavlažovacie systémy podporujú fyzickú komunikáciu priamo s PC alebo dokonca ovládanie cez mobilné aplikácie alebo programy v PC pomocou internetu.

Dôležitým faktorom môže byť pre užívateľa aj rozhranie ktoré daný typ riadiacej jednotky poskytuje. Niektorý užívatelia uprednostňujú jednoduché ovládanie, ktoré je ideálne v rôznych jazykoch. Iný užívatelia preferujú komplexné možnosti úprav parametrov zavlažovania.

Pre rôzne typy špeciálnych zavlažovacích plôch ako sú napríklad veľké športové areáli je určite dôležitá garancia spoľahlivosti a bezpečnosti zavlažovacích systémov. Pre tieto prípady je vhodné mať riadiacu jednotu podporujúcu pripojenie

prietokomerov, ktoré slúžia ako spätná väzba a dokážu detegovať a odstrániť prípadné poruchy.

2.2.2 Porovnanie vlastností vybraných riadiacich jednotiek

Aktuálne sú na trhu najrozšírenejšie dve veľké značky zavlažovacích systémov. Firma Hunter sa špecializuje na jednoduchšie a cenovo dostupnejšie riadiace jednotky ale ponúka taktiež moderné a komplexné zavlažovacie systémy. Firma Toro má väčšiu škálu dostupných produktov, ktoré majú často špecifickejšie vlastnosti a sú drahšie.

Od spomínaných výrobcov porovnám nasledujúce riadiace jednotky:

1. TORO DDC4 - 4 sekcie - interná;
2. TORO DDCWP 2 sekcie;
3. HUNTER XCORE 201 I E - 2 sekcie - interná;
4. HUNTER HYDRAWISE HC 601i-E 6 sekcií.

1. TORO DDC4 - 4 sekcie - interná

Špecifikácia a výhody [3]:

- určená pre inštaláciu v interiéri (dom, garáž, záhradný domček);
- možnosť ovládania maximálne 4 vetiev - sekcií;
- jednotka má 3 nezávislé programy;
- v každom programe je možné zvoliť 4 štartovacie časy;
- možnosť napojenia dažďového senzora a pôdneho čidla;
- programovateľná dažďová pauza;
- možnosť programovania senzorov;
- rýchla kontrola systému pomocou testovacieho programu;
- trvalá pamäť;
- záložný zdroj 9 V batéria - zálohovanie nastavenia programov;
- možnosť nastavenia zavlažovacích dní;
- možnosť ovládania hlavného ventilu alebo relé čerpadla;
- funkcia prezerania programov;
- istič s funkciou vlastnej diagnostiky;
- dodáva sa s transformátorom;
- typická spotreba 12 VA (riadiaca jednotka + 1 ventil);
- cena 74,95 €.

2. TORO DDCWP 2 sekcie

Špecifikácia a výhody [4]:

- vodotesná batériová riadiaca jednotka série DDC;
- napájanie 2 x 9 V batériami (nie sú súčasťou dodávky);
- úplne vodotesná, možnosť ponorenia až do hĺbky 2 m;
- súčasťou príslušenstva je montážny držiak pre jednoduchú inštaláciu;
- spolupracuje s dažďovým senzorom a ďalšími pevne pripojiteľnými senzormi;
- umožňuje ovládať výhradne elektroventily s 9 V DC cievkou;
- cena 144,95 €.

3. HUNTER XCORE 201 I E - 2 sekcie - interná

Špecifikácia a výhody [5]:

- určená pre inštaláciu v interiéri (dom, garáž, záhradný domček);
- možnosť ovládania maximálne 2 vetiev - sekcií;
- jednotka má 3 nezávislé programy;
- v každom programe je možné zvoliť 4 štartovacie časy;
- možnosť napojenia dažďového senzora a pôdneho čidla;
- zabudovaný senzor Solar Sync;
- možnosť odpojenia senzora pre jednotlivé vetvy (napr. vetva pre skleník);
- programovateľná dažďová pauza;
- trvalá pamäť;
- rýchla kontrola systému pomocou testovacieho programu;
- pamäť Easy Retrive;
- systém QuickCheck - umožňuje zistiť problémy s elektroinštaláciou na mieste inštalácie stlačením jediného tlačítka ;
- automatická ochrana proti skratu;
- sezónne nastavenie - dlhodobé alebo automatické aktualizácie so senzorom Solar Sync;
- pauza medzi sekciami;
- možnosť programovania senzorov;
- možnosť ovládania hlavného ventilu alebo relé čerpadla;
- možnosť pripojiť diaľkové ovládanie ROAM KIT;
- BYPASS dažďového senzora;
- dodáva sa s transformátorom;
- typická spotreba 20 VA (riadiaca jednotka + 1 ventil);
- cena 61,95 €.

4. HUNTER HYDRAWISE HC 601i-E 6 sekcií

Špecifikácia a výhody [6]:

- internetová ovládacia jednotka Hunter HC 6 s webovým softvérom Hydrawise;
- jednotka ponúka vzdialenú správu závlahového systému a prehľadný farebný dotykový displej s podsvietením;
- intuitívne ovládanie a programovanie pomocou dotykového displeja;
- ovládanie a nastavanie jednotky cez PC, tablet, smartfón;
- ovládanie a nastavenie pomocou cloudového sofwéru alebo aplikácie;
- pripojenie k internetu prostredníctvom WiFi siete;
- jednotka vyhodnocuje informácie o počasí z meteorologických staníc;
- zavlažovanie sa upravuje na základe prognózovanej teploty, pravdepodobnosti zrážok, vetra a vlhkosti;
- úspora vody;
- prietokomer automaticky upozorní, ak sa poškodí potrubie alebo postrekovač;
- dodáva sa s transformátorom;
- cena 284,95 €.

Licencia Enthusiasm je predplatený ročný program umožňujúci prístup k nadštandardným funkciám ovládacej jednotky. V cene jednotky je licencia Enthusiasm na 1 rok zadarmo.

Rozšírené funkcie licencie Enthusiasm:

- okamžitá "Push" notifikácia a SMS správy na mobilný telefón (iOS alebo Android) so správami o alarmoch;
- prístup k vlastnej meteostanici alebo k najbližším staniciam vo vašom okolí (max. 5 staníc);
- z nameraných hodnôt je stanovanie priemer (medián) a následne je potom upravený závlahový program v závislosti od aktuálnych klimatických podmienkach;
- aktualizácia závlahových programov po jednej hodine na základe vývoja počasia
- možnosť tvorby reakcií ovládacej jednotky (štart / stop / oneskorenie) na aktiváciu senzorov;
- história zavlažovania za 12 mesiacov;
- možnosť pripojiť viac jednotiek pod jedným účtom;
- priradenie obrázkov k jednotlivým sekciám;
- zasielanie upozorňujúcich SMS správ;
- pokročilé nastavenia senzorov.

Zavlažovací systém s Raspberry

- určená pre inštaláciu v interiéri (dom, garáž, záhradný domček);
- ovládanie a nastavanie jednotky cez PC, smartfón... pomocou webovej stránky;
- možnosť ovládania maximálne 2 vetiev - sekcií;
- v každom programe je možné zvoliť 4 štartovacie časy;
- možnosť napojenia dažďového senzora a pôdneho čidla;
- BYPASS dažďového senzora a pôdneho čidla
- možnosť nastavenia prahu senzorov;
- jednotka zobrazuje teplotu a vlhkosť vzduchu
- funkcia prezerania nastavení;
- záloha nastavení a času;
- napájanie riadiacej jednotky pomocou 5V/1A DC microUSB konektoru;
- svorkovnice na pripojenie ventilov a externého napájania ventilov;
- typická spotreba 6,54 VA (riadiaca jednotka + 1 ventil);
- cena približne 100€.

Z uvedených parametrov vyplýva že mnou navrhnutý zavlažovací systém má absenciu niekoľko bežných funkcií ako je napríklad uloženie nastavení do niekoľkých programov, kontrola poruchy systému a ochrana proti skratu. Na rozdiel od všetkých dostupných jednotiek v svojej cenovej kategórii(jednotky 1, 2, 3) ponúka ovládanie a nastavanie zavlažovania pomocou webovej stránky, čo bolo hlavným dôvodom tvorby tohto systému. Ďalej ponúka zobrazenie teploty a vlhkosti vzduchu a taktiež možnosť pripojiť elektromagnetické ventily rôznych typov a značiek (vždy rovnaká dvojica ventilov). Posledná jednotka HUNTER HYDRAWISE ponúka širšie spektrum funkcií a ako jediná má funkciu ovládania pomocou webovej stránky, čo sa samozrejme odráža na niekoľko násobne vyššej cene.

Väčšina zavlažovacích systémov vyžaduje pripojenie priamo do elektrickej siete. Druhá popísaná jednotka *TORO DDCWP* je napájaná dvojicou 9V batérií. Môj systém je taktiež možné napájať pomocou batérie, za predpokladu že budú použité ventile so solenoidmi ovládanými jednosmerným napätím. Typická spotreba uvedených komerčných jednotiek sa pohybuje v rozmedzí 12-20VA, čo je niekoľko násobne viac ako spotreba jednotky s Raspberry(6,54 VA). Výpočet spotreby je v kapitole *4.1.6*.

2.2.3 Akčné členy

"Elektromagnetické ventily sú hlavné ovládacie prvky automatického zavlažovacieho systému. Každý automatický zavlažovací systém je podľa veľkosti a členitosti zavlažovanej plochy, použitých postrekovačov a výdatnosti zdroja vody rozdelený do niekoľkých zavlažovacích sekcií.

Elektromagnetický ventil slúži na uzatváranie prietoku vody, hlavným cieľom využitia elektromagnetických ventilov sú najmä rozvody vody. Účinný zavlažovací

systém sa nezaobíde bez ventilov, ktoré sú spoľahlivé a funkčné aj pri kontakte so znečistenou vodou, odolávajú vysokému tlaku a sú energeticky úsporné.

Elektromagnetické ventily sú uložené vo ventilových šachtách pod terénom. Ventily sú spínané ovládacou jednotkou, ktorá na záhradách rodinných domov ovláda obyčajne 4 - 12 ventilov (závlahových sekcií). Každá sekcia je ovládaná samostatným elektromagnetickým ventilom podľa pokynov riadiacej jednotky. Automatický zavlažovací systém postupne, podľa programu nahranom v riadiacej jednotke, spína jednotlivé elektromagnetické ventily tak, aby v danom okamihu zavlažovala vždy iba v jednej sekcia." [7]

Pre elektronické zavlažovacie systémy sa ako akčné členy takmer výhradne používajú elektromagnetické ventily ovládané pomocou 9/12/24 V solenoidu.

Pri výbere ventila sú dôležité tieto parametre :

- napájanie (solenoid - 9V/12/24V; AC/DC);
- veľkosť a typ závitu (3", 1", 3/4" ; vonkajší/vnútorný);
- prevádzkový tlak (typicky 10-12 bar, max. 15 bar);
- prietok vody (typicky 1-120 l/min);
- regulácia prietoku (áno/nie);
- zobrazovač výstupného tlaku (áno/nie);
- manuálne otvorenie ventilu (áno/nie);
- ručné odvzdušnenie (áno/nie);
- materiál;
- cena.

Z hľadiska elektrickej kompatibility s riadiacou jednotkou je dôležitá hlavne hodnota a typ napájacieho napätia použitého solenoidu.

Podľa rozlohy a tvaru zavlažovaného priestoru sa rozvrhne počet a veľkosť postrekovačov, na základe čoho je možné určiť priemer elektromagnetického ventilu.

Pre sekcie s kvapôčkovou závlahou je vhodné použiť ventil s reguláciou prietoku aby nedochádzalo k príliš veľkej závlahe. Výstupný tlak vody je možné u niektorých špeciálnych ventilov priamo odčítať na zabudovanom tlakomeri.

Pri prvom testovaní jednotlivých sekcií je výhodné manuálne otvorenie ventilu, ktoré je zvyčajne umožnené pootočením solenoidu o 90°. Pootočením sa elektromagnetický ventil otvorí bez nutnosti pripojenia k riadiacej jednotke, čo nám umožní skontrolovať funkčnosť a tesnosť závlahového systému.

2.2.4 Senzory

Senzory aj celé meteo stanice zaisťujú chod automatického závlahového systému presne podľa počasia. Ani najlepšie ovládacie jednotky nezaistia taký závlahový režim, ktorý by zodpovedal premenlivosti počasia. Na tento účel je potrebné automatické ovládanie

doplniť o senzory a čidlá zrážok, reagujúce na zmeny počasia. Čidlo zrážok je možné použiť pre akýkoľvek automatický závlahový systém, pracujúci s kompatibilným napätím. Čidlá sa umiestňujú vo voľnom priestore, tak aby správne reagovali na padajúce zrážky a vplyv vetra. [1]

Čidlo zrážok, teploty alebo pôdnej vlhkosti pracuje v spolupráci s ovládacou jednotkou zavlažovacieho systému a v prípade prirodzených zrážok naprogramované závlahové cykly pozastaví. [8]

Súčasťou automatického zavlažovacieho systému je čidlo zrážok. Umiestňuje sa tak, aby zachytilo dážď. Potom v prípade dostatočnej vlhkosti pôdy vplyvom prirodzených zrážok samostatne vypne zavlažovanie a po vyschnutí opäť podľa potreby zapne. Tým sa šetrí voda a predovšetkým sa zabraňuje podmáčaniu trávnika. [8]

Teplotné čidlo slúži primárne na detekciu nízkej vonkajšej teploty, čím zamedzuje spustenie závlahy pri poklese teploty pod užívateľom nastavenú hodnotu. V prípade zavlažovania pri nízkych teplotách hrozí zamrznutie vody na steblách trávy a poškodenie trávnika. Toto čidlo sa ale v takmer vôbec nepoužíva, pretože v jarnom a jesennom období je dostatok prirodzených zrážok a cez leto nebezpečne nízke teploty nehrozia.

Senzor pôdnej vlhkosti je umiestnený pod povrchom zeme. Keď systém deteguje suchý stav pred normálnym cyklom zavlažovania cyklus je povolený, keď je vlhkosť nad nastavenú prahovú hodnotu cyklus sa pozastaví, aby sa neplytvalo vodou. [9]

Niektoré riadiace jednotky automatických závlah podporujú po prepojení jednotky s prietokomerom funkciu snímania prietoku. Takéto systémy dokážu detegovať automaticky problémovú sekciu a izolovať ju zatvorením príslušného ventilu.

2.2.5 Doplnky

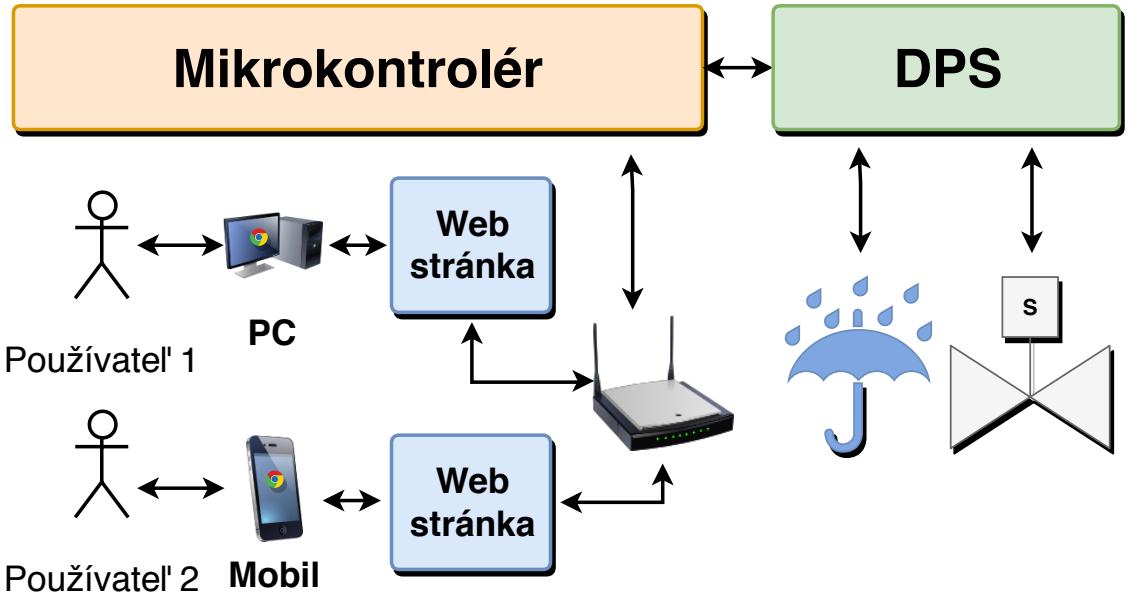
K závlahe neodmysliteľne patrí ešte veľké množstvo doplňujúcich častí, bez ktorých by závlaha nefungovala. V tejto práci sa nebudem venovať popisu záhradníckych komponentov ale zameriam sa na elektronickú časť zavlažovacieho systému.

Stručný súhrn ostatných častí závlahy :

- postrekovače;
- filtrácia;
- trysky;
- šachty;
- čerpadlá;
- spojovací materiál;
- manuálne ventily;
- spojovací materiál;
- potrubia;
- hadice.

2.3 Návrh systému

Blokové schéma mnou navrhnutého zavlažovacieho systému je znázornené na nasledujúcom obrázku.



Obr. 1 Koncept zavlažovacieho systému

Cieľom práce je navrhnúť elektronický systém, ktorý prostredníctvom webového prehliadača bude schopný ovplyvňovať parametre riadiacej jednotky zavlažovacieho systému realizovaného mikrokontrolérom. Riadiaca jednotka bude taktiež spolupracovať so senzorom vlhkosti a teploty. Na základe automatického programu zavlažovania alebo manuálnych pokynov užívateľa sa budú riadiť jednotlivé elektroventile ktoré budú zavlažovať príslušné zavlažovacie sekcie.

2.4 Mikrokontrolér

Z požiadaviek formulovaných v zadaní práce vyplýva, že základom elektronického zavlažovacieho systému má byť mikrokontrolér podporujúci komunikáciu cez Ethernet alebo GSM. Prvotnou myšlienkou je vytvoriť jednoduchú webovú aplikáciu komunikujúcu s mikrokontrolérom cez Ethernet port. Následne je potrebné pomocou mikrokontroléru na základe vstupných signálov od senzorov a web aplikácie ovládať niekoľko digitálnych výstupných portov, ktoré budú riadiť jednotlivé zavlažovacie elektroventile.

Uvedené požiadavky splňujú mimo iné aj dve najrozšírenejšie mikrokontrolérové platformy Arduino a Raspberry Pi. S mikrokontrolérom od spoločnosti Arduino som pracoval v mojej bakalárskej práci. S Raspberry Pi som ešte nepracoval, preto je to pre mňa tak trochu výzva, ktorú hodlám zdolať. Ako je uvedené nižšie, Raspberry Pi B+ má zabudovaný Ethernet port, taktiež všetky ostatné potrebné porty a zbernice. Taktiež má možnosť pripojenia LCD dotykového displeja.

2.4.1 Raspberry Pi 1 Model B+

Keďže nepredpokladám, že výsledná aplikácia bude príliš náročná na výpočtový výkon, stačí použiť aj staršiu verziu Raspberry Pi. Tento model sa mi podarilo zohnať od spolužiaka ale je možné využiť ktorýkoľvek model s možnosťou pripojenia k domácej sieti, potrebnými zbernicami a dostatočným počtom GPIO výstupov.

Základné parametre:

Architektúra	ARMv6Z (32-bit);
SoC	Broadcom BCM2835;
Procesor	700 MHz single-core ARM1176JZF-S;
Video	Broadcom VideoCore IV @ 250 MHz, OpenGL ES 2.0, MPEG-2, 1080p30 H.264 / MPEG-4 AVC;
Pamäť	512 MB (SDRAM, zdieľaná s GPU);
USB 2.0 porty	4 (cez vstavaný 5-portový USB hub);
Video vstup	1 x MIPIM kamerové rozhranie (CSI);
Video výstup	HDMI (rev 1.3), kompozitné video (3.5 mm TRRS jack), MIPIM display interface (DSI);
Zvukový vstup	Cez I²S rozhranie;
Zvukový výstup	Analógový, 3.5 mm jack, Digitálny, HDMI a I²S;
Interná pamäť	MicroSDHC;
Integrovaná sieť	10/100 Mbit / s Ethernet (8P8C) USB adaptér cez USB hub;
Periférie	17 × GPIO plus rovnaké funkcie a HAT ID zbernice;
Menovitý výkon	200 mA (1 W) pohotovostný režim, maximálne 350 mA, (1.75 W) pri zaťažení (monitorom, klávesnica a myš);
Rozmery	85,60 mm × 56,50 mm (bez konektorov);
Hmotnosť	45 g.

Všetky parametre boli prevzaté z internetovej stránky predajcu [10]. Na Obr. 46-A Popis GPIO konektoru [11] sa nachádza popis konektoru umiestneného na tomto modeli Raspberry Pi.

2.5 Zbernice

V praktickej časti práce som využíval zariadenia, ktoré sa pripájali pomocou dvojice nižšie spomenutých zberníc. Pre správne zapojenie a komunikáciu s týmito zariadeniami bolo potrebné naštudovať a pochopiť princíp ich fungovania, oboznámiť sa s ich možnosťami, výhodami a nevýhodami a následne tieto informácie zohľadniť pri ich používaní.

Zbernica je elektrický vodič, alebo častejšie sústava vodičov, ktoré prepájajú viaceré elektrické či elektronické zariadenia a slúžia na vzájomný prenos energie, alebo údajov vo forme elektrických impulzov, medzi nimi. Často sa pojem zbernica (a názov konkrétneho druhu zbernice) používa aj na označenie komunikačného rozhrania a príslušnej normy či komunikačného protokolu, ktorými sa riadi prenos údajov. [12]

Počet vodičov sa mení v závislosti na druhu zbernice. V mojej práci som sa stretol s použitím zbernice označenej I^2C a SPI.

Na stránke kiwiki je zmysluplne, názorne a detailne popísaný princíp oboch zberníc. Taktiež sú uvedené výhody, nevýhody, použitie a porovnanie uvedených zberníc. Z tohto dôvodu som sa rozhodol informácie o SPI prevziať a upraviť zo stránky [13] a informácie o I^2C zo stránky [14].

2.5.1 Komunikačná zbernica SPI

"SPI (Serial Peripheral Interface) je štvorvodičová synchrónna sériová zbernica pracujúca v móde full duplex slúžiaca na prepojenie periférií s mikropočítačmi. Protokol bol definovaný firmou Motorola v produktovej línii mikroradičov MC68HCxx. Zariadenia komunikujú spôsobom master/slave, kde master iniciuje dátový rámec. SPI zbernica sa dá použiť v prípadoch, kedy je k jednému zariadeniu master pripojených viacero zariadení slave. Na ich realizáciu musí master pri bežnom zapojení disponovať potrebným počtom slave select (chip select) vývodov. SPI sa niekedy nazýva "four wire" sériová zbernica ako opak ku three, two a one wire sériovým zberniciam.

Obr. 2 SPI zbernica - jeden master a jeden slave [13]

Zbernica SPI špecifikuje štyri logické signály:

- SCLK/SCK/CLK — Serial Clock (výstup zariadenia master);
- MOSI/SDI/DI/SI — Master Out, Slave In (výstup zariadenia master);
- MISO/ SDO/DO/SO — Master In, Slave Out (výstup zariadenia slave);
- SS/nCS/CS/CSB/nSS/STE — Slave Select (active low - logickú jednotku reprezentuje nižšia úroveň napätia; výstup zariadenia master).

Zbernica SPI vie pracovať s jedným zariadením master a jedným alebo viacerými zariadeniami slave. SS pin môže byť zafixovaný na nižšiu úroveň napätia, ak sa používa iba jedno zariadenie slave a toto zariadenie dovoľuje zmienenú možnosť.

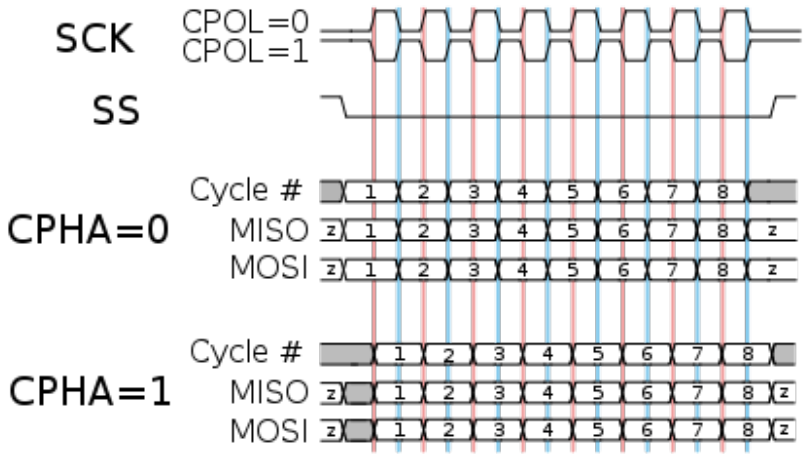
Niektoré zariadenia vyžadujú na pine SS hranu prechodu napätia z vyššej úrovne na nižšiu pre iniciáciu akcie, napríklad AD prevodník Maxim MAX1242. Ak je pripojených viacero zariadení slave, je potrebný samostatný SS signál zo zariadenia master do každého zariadenia slave. Väčšina zariadení slave má trojstavový výstup, takže ich signál MISO do stavu hi-Z (vysoko-impedančný, "odpojený" - neovplyvňuje obvod, do ktorého je zapojený), keď zariadenie nie je vybraté. Zariadenia bez trojstavového výstupu nemôžu zdieľať segmenty SPI zbernice s ostatnými zariadeniami. Iba jedno takéto zariadenie môže komunikovať so zariadením master a iba jeho SS môže byť aktivované.

Pred začatím komunikácie master najskôr nastaví hodinový signál SCLK na hodnotu frekvencie menšej alebo rovnej maximu, ktoré slave podporuje. Tieto frekvencie sú väčšinou v rozsahu 1-70 MHz. Master potom vyberie požadovaný slave nastavením jeho signálu SS na úroveň logickej jednotky. Ak je nutná čakacia doba (napríklad pri AD prevode), potom master musí čakať minimálne po tento čas, kým začne vykonávať hodinové cykly. Počas každého hodinového cyklu SPI sa vykoná full duplexový prenos dát:

- master pošle bit po linke MOSI, slave ho z tejto linky prečíta;
- slave pošle bit po linke MISO, master ho z tejto linky prečíta.

Nie všetky prenosy vyžadujú všetky štyri operácie, ale vykonajú sa, aj keď nemajú význam. Pri prenosoch sa väčšinou používajú dva posuvné registre veľkosti danej dĺžkou slova, napríklad osem bitov, jeden v zariadení master, druhý v zariadení slave. Sú spojené do kruhu. Dáta sa väčšinou vyberajú od najsignifikantnejšieho bitu a do rovnakého registra sa presunie nový najmenej signifikantný bit. Po tom, čo je tento register vybratý, si master a slave navzájom vymenili hodnoty v registroch. Následne každé zariadenie vezme túto hodnotu a nejako ju spracuje, napríklad si ju zapíše do pamäte. Ak existujú ďalšie dáta, ktoré majú byť vymenené, posuvné registre sa naplnia novými hodnotami a proces sa opakuje. Prenos môže trvať ľubovoľný počet hodinových cyklov. Keď už nie sú žiadne dáta, ktoré majú byť prenesené, master prestane vysielať hodinový signál. Väčšinou potom odznačí slave. Prenos sa často skladá z 8-bitových slov. Bežné sú však aj iné dĺžky slov, napríklad 16-bitové pre touchscreeny alebo audio kodeky (TSC2101 od Texas Instruments) alebo 12-bitové pre mnohé DA a AD prevodníky. Každý slave na zbernici, ktorý nebol aktivovaný cez SS linku, musí ignorovať vstupujúci hodinový a MOSI signál a nesmie vysielať cez MISO. Master musí vybrať v danom čase iba jeden slave.

Okrem frekvencie hodinového signálu musí master nastaviť aj jeho polaritu a fázu vzhľadom na prenášané dáta. Príručka SPI Block Guide [15] od Freescale Semiconductor nazýva tieto dve možnosti CPOL a CPHA a väčšina predajcov toto pomenovanie používa.



Obr. 3 Časový diagram rôznych kombinácií polarít a fáz hodinového signálu [13]

Opis časového diagramu zobrazeného na obrázku (platí pre zariadenie master aj slave):

- Pri CPOL=0 je základná hodnota hodinového signálu nula.
  - Pri CPHA=0 sú dáta čítané pri nábežnej hrane signálu (prechod 0 -> 1) a menia sa pri dobežnej hrane (prechod 1 -> 0).
  - Pri CPHA=1 sú dáta čítané pri dobežnej hrane signálu a menia sa na nábežnej hrane.
- Pri CPOL=1 je základná hodnota hodinového signálu jedna.
  - Pri CPHA=0 sú dáta čítané pri dobežnej hrane signálu a menia sa pri nábežnej hrane.
  - Pri CPHA=1 sú dáta čítané pri nábežnej hrane signálu a menia sa pri dobežnej hrane.

Takže CPHA=0 znamená vzorkovanie na úvodnej hrane hodinového signálu a CPHA=1 znamená vzorkovanie na koncovej (druhej) hrane signálu, pričom nezáleží na tom, či je hrana nábežná alebo dobežná. Všimnite si, že pri CPHA=0 musia byť dáta stabilné polovicu cyklu pred prvým cyklom hodinového signálu. Pri všetkých módoch CPOL a CPHA musí byť počiatočná hodnota hodinového signálu nastavená pred zvolením zariadenia slave. Tiež si všimnite, že "dáta sú prečítané" v tomto dokumente zvyčajne znamená "dáta môžu byť prečítané". Signály MOSI a MISO sú zvyčajne stabilné (v bode ich prijatia) polovicu cyklu až do ďalšieho prenosu hodinového signálu. Zariadenia master a slave na SPI zbernici môžu vzorkovať dáta v iných bodoch tejto polovice cyklu. To dáva komunikačnému kanálu medzi zariadeniami väčšiu možnosť flexibility. Niektoré produkty používajú inú konvenciu názvov. Napríklad TI MSP430 používa namiesto CPOL názov UCCKPL a UCCKPH namiesto CPHA. Pri spájaní dvoch čipov dajte pozor na inicializačné hodnoty fázy hodinového signálu.

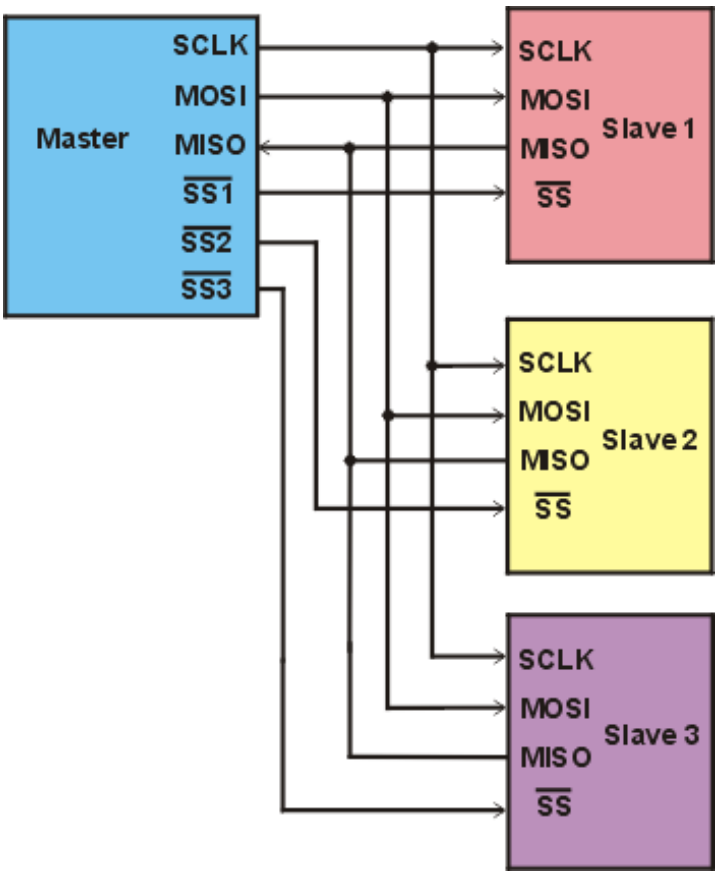
Kombinácie hodnôt polarity a fázy sa často označujú ako módy, ktoré sú bežne očíslované podľa pravidla, že CPOL označuje vyšší bit čísla a CPHA nižší bit. Ďalším

bežne používaným spôsobom reprezentácie módu je usporiadaná dvojica (CPOL,CPHA), napríklad hodnota "(0,1)" by znamenala CPOL=0 a CPHA=1.

Tabuľka 1 Čísla módov SPI zbernice [13]

Mód	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

V konfigurácií s nezávislými zariadeniami slave existuje samostatná linka SS pre každé zo zariadení. Toto je bežný spôsob použitia zbernice SPI. Keďže sú MISO piny zariadení slave spojené, musia to byť trojstavové piny.



Obr. 4 Konfigurácia SPI s nezávislými zariadeniami slave [13]

Klady:

- Komunikácia full duplex;
- Vyššia priepustnosť než pri I^2C alebo SMBus;
- Flexibilita prenosu bitov;
    - Nie je obmedzená na osembitové slovo;
    - Ľubovoľný výber veľkosti správy, obsahu a účelu;
- Extrémne jednoduché hardvérové rozhranie;
    - Väčšinou nižšie energetické nároky než pri I^2C alebo SMBus;
    - Žiadna nejednoznačnosť alebo pridružené chybové stavy;
    - Slave používa hodinový signál zariadenia master a nepotrebuje presný oscilátor;
    - Slave nepotrebuje unikátnu adresu ako je to pri I^2C, GPIB alebo SCSI
    - Nie sú potrebné vysielače;
- Používa oveľa menej káblov než paralelné rozhrania;
- Nanajvýš jeden signál pre každé zariadenie zvlášť (SS), ostatné sú zdieľané všetkými zariadeniami;
- Signály sú jednosmerné, čo umožňuje galvanickú izoláciu.

Zápory:

- Potrebuje viac pinov na integrované obvody než I^2C aj pri variante "3-Wire";
- Žiadna hardvérová kontrola toku (ale master môže oddialiť ďalšiu hranu hodinového signálu aby spomalil prenosovú rýchlosť);
- Žiadne hardvérové potvrdenie od zariadenia slave (master by nemusel "hovoriť s nikým" bez toho, aby si to všimol);
- Podporuje iba jedno zariadenie master;
- Nie je definovaný protokol kontroly chýb;
- Vo všeobecnosti náchylná na chyby spôsobené šumom;
- Bez formálneho štandardu nie je možná kontrola zhody;
- V porovnaní s RS-232, RS-485 alebo CAN zvláda iba malé vzdialenosti.

Použitie:

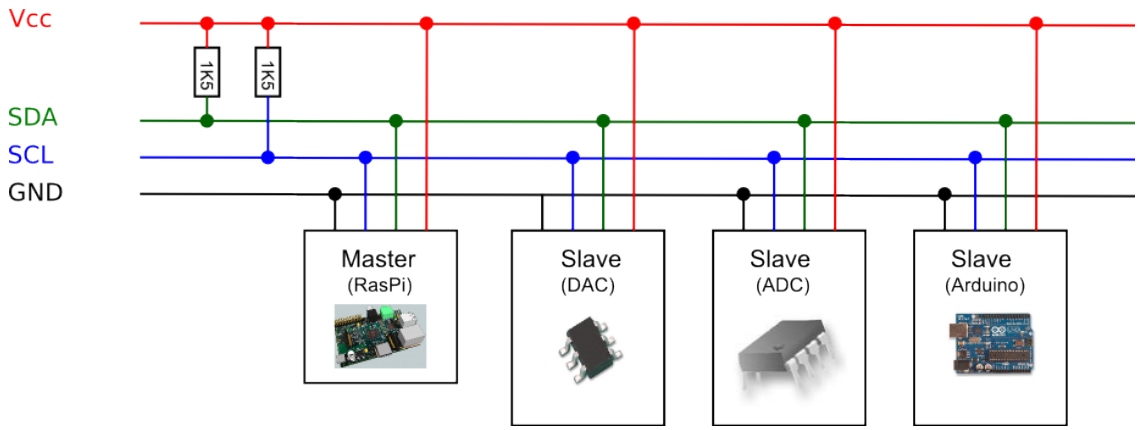
- snímače teploty, tlaku, dotykové obrazovky, AD prevodníky;
- riadiace zariadenia: audio kodeky, digitálne potenciometre, DA prevodníky;
- šošovky fotoaparátov, kamier;
- komunikačné zariadenia: Ethernet, USB, USART, CAN, IEEE 802.15.4/802.11;
- pamäte: flash a EEPROM;
- hodiny reálneho času;
- LCD displeje, MMC a SD karty." [13]

2.5.2 Komunikačná zbernica I^2C

"Skratka pre zbernicu vznikla z anglického názvu “Inter-Integrated Circuit”. Protokol a zbernicu vyvinula firma Philips Semiconductor pôvodne pre svoje TV prijímače v roku 1980 za účelom komunikácie medzi IO na jednej DPS pri použití minimálneho množstva pinov. Špecifikácia zbernice I2C je založená na jednoduchých hardvérových štandardoch (nie sú potrebné špeciálne konektory alebo kabeláž) a rovnako jednoduchého softvérového štandardu pre komunikačný protokol. Obvody, ktoré používajú I2C protokol zahŕňajú pamäte EEPROM a RAM, senzory teploty, expandéry portov, hodiny reálneho času, atď. Používa sa tiež ako riadiaca zbernica v obvodoch spracovania signálov, ktoré majú oddelenú dátovú zbernicu, napr. RF tunery, video dekodéry a enkodéry, audio procesory a kodeky. Zbernica I^2C môže pracovať pri troch prenosových rýchlostiach:

- slow (pod 100 Kbps);
- fast (400 Kbps);
- high-speed (3.4 Mbps) – protokol označený ako I2C v.2.0.

Vzdialenosť komunikujúcich zariadení je limitovaná z dôvodu udržania komunikačnej rýchlosti na približne 4m. Maximálna kapacitancia prenosového vedenia je 400pF. Zbernica používa dva vodiče – Serial Data (SDA) a Serial Clock (SCL). I^2C je synchrónna (signál SCL), multimasterová (aj slave môže byť konfigurovaný tak, aby mohol začať komunikáciu) zbernica s polovičným duplexom. Každý IO na zbernici je identifikovaný svojou adresou, ktorá je v rámci siete jedinečná, preto zbernica I2C nevyžaduje signál CS (chip select) ani ďalšiu logiku. Linky SDA aj SCL sú pripojené na napájacie napätie pomocou tzv. pullup rezistorov. Každé zariadenie na zbernici môže stiahnuť danú linku na nízku úroveň pomocou tranzistorov s otvoreným kolektorom.



Obr. 5 Hardvérová architektúra I^2C [16]

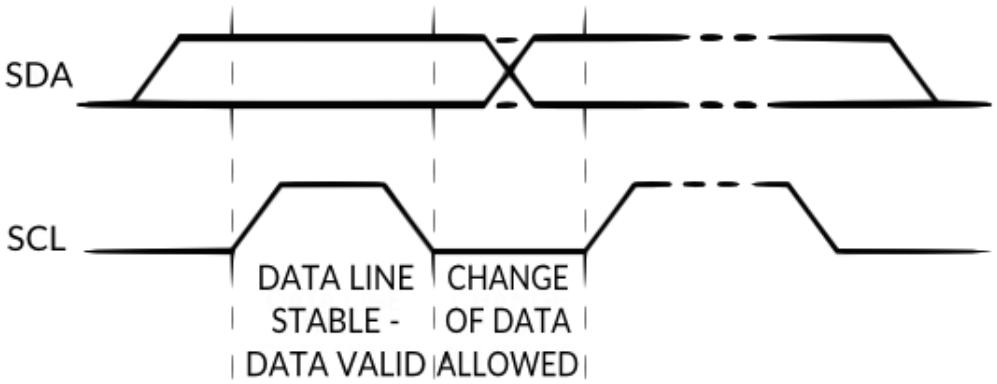
Master:

- začína a končí dátový prenos generovaním štart bitu a stop bitu;
- generuje hodinový signál;
- vysiela adresu podriadeného IO, pre ktorý budú dáta určené;
- určuje smer prenosu dát.

Slave:

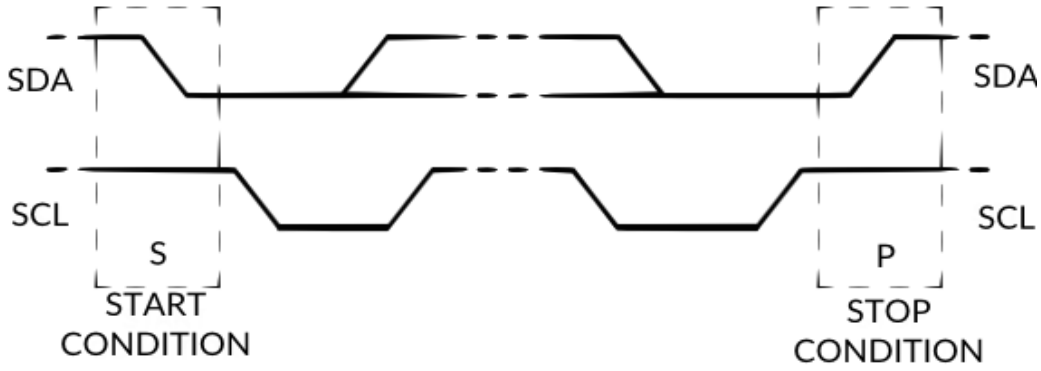
- odpovedá iba v prípade, že rozpoznal svoju adresu;
- časovanie prenosu je riadené hodinovým signálom.

Počas dátového prenosu sa stav na dátovom vodiči bitu mení iba keď je hodinový signál na nízkej úrovni.



Obr. 6 Prenos bitu na I^{2}C zbernici [17]

Začiatok alebo koniec prenosu je definovaný prechodom dátovej linky z vysokej na nízku úroveň (štart bit) alebo z nízkej na vysokú úroveň (stop bit) kým hodinový signál je na vysokej úrovni. Po štart bite považujú všetky zariadenia zbernicu za zaneprázdnenú. Po príchode stop bitu čakajú zariadenia istý čas a potom považujú zbernicu za voľnú.



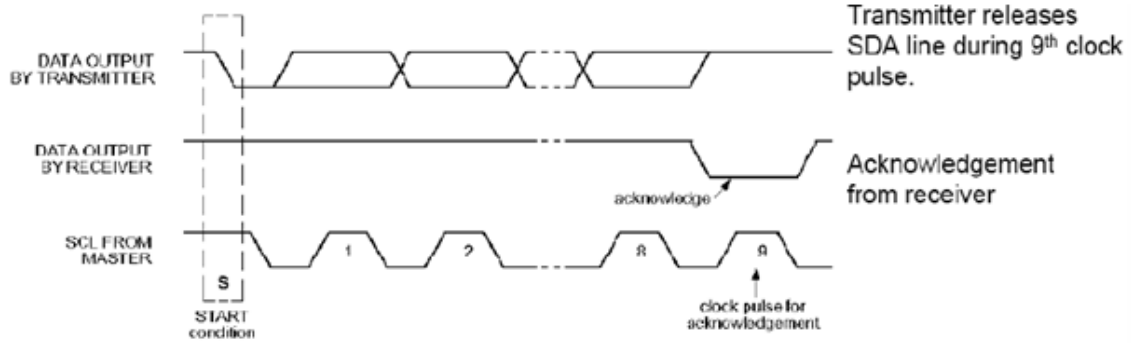
Obr. 7 Štart bit a stop bit signály I^{2}C zbernice [17]

Každé zariadenie má 7 (10) bitovú adresu. Adresa sa skladá z fixnej a programovateľnej časti. Programovateľná časť adresy umožňuje použiť na danej zbernici viacero rovnakých vstupno-výstupných zariadení. Niektoré adresy sú vyhradené pre špeciálne účely, preto je možné celkovo teda pripojiť na zbernicu pri 7-bitovom adresovaní maximálne 119 vstupno-výstupných zariadení.

Postupnosť pri prenose dát je nasledovná. Master generuje štart bit a hodinový signál. Master posiela adresu podriadeného IO + generuje bit R/\W. Podriadený IO potvrdí prijatie bitom ACK. Vysielacie zariadenie (master alebo slave) vyšle jeden bajt dát. Prijímacie zariadenie vloží bit ACK, čím potvrdí prijatie bajtu. Opakujú sa predchádzajúce dva body ak je potrebné vyslať viac bajtov. Pri zápise (master vysiela), master vloží stop bit po prenesení posledného bajtu dát. Pri prijímaní (master prijíma), master nepotvrdzuje posledný bajte bitom ACK, ale priamo vloží stop bit, aby oznámil podriadenému IO, že prenos bol dokončený.

Potvrdenie príjmu (Acknowledge) sa vykonáva počas deviateho impulzu hodinového signálu a je povinný. Vysielajúce zariadenie uvoľní linku SDA (umožní jej „plávať“ na vysokej úrovni). Prijímajúce zariadenie stiahne linku SDA na nízku úroveň (linka SCL musí byť na vysokej úrovni). Ak nedošlo k potvrdeniu, prenos je ukončený.

Natiahnutie hodín (Clock Stretching) nastáva, keď slave (prijímač) potrebuje viac času na spracovanie bitu alebo práve vykonáva iné funkcie. Môže tak stiahnuť a podržať SCL na nízkej úrovni. Master potom čaká kým slave uvoľní linku SCL predtým, než vyšle ďalší bit." [14]



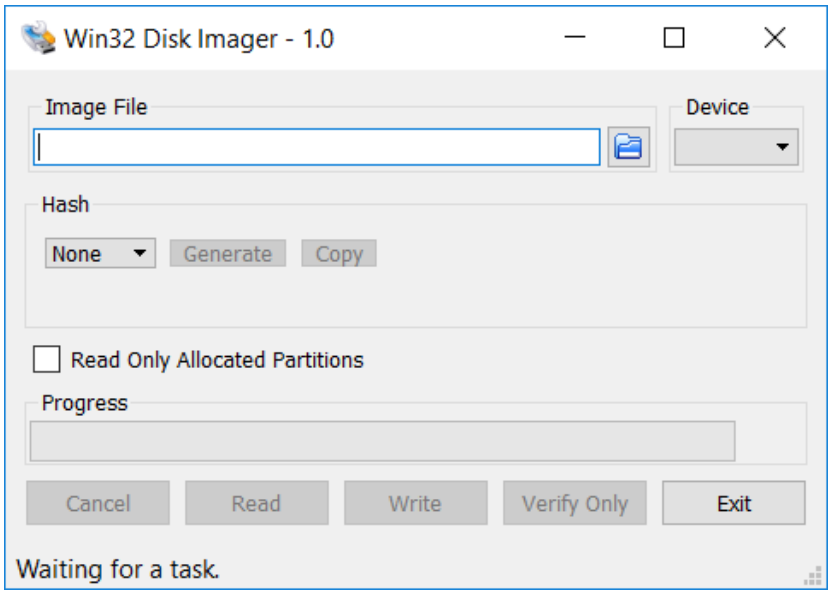
Obr. 8 I²C prenos dát na I²C zbernici [14]

2.6 Použité programy

Pri tvorbe automatického zavlažovacieho systému som použil niekoľko rôznych programov. Či už na nainštalovanie operačného systému, komunikáciu s Raspberry Pi, tvorbu samotného programu a vizualizácie alebo návrh dosky plošných spojov. V nasledujúcich podkapitolách sa nachádza zoznam najdôležitejších programov a popis ich funkcií.

2.6.1 Win32DiskImager

Tento program je navrhnutý na zápis obrazu disku na vymeniteľné zariadenie alebo zálohovanie súborov vymeniteľného zariadenia do nespracovaného (raw) obrazu, čo je veľmi užitočné pre embedded vývoj, menovite Arm vývojové projekty (Android, Ubuntu na Arm, atď). [18]



Obr. 9 Win32DiskImager [18]

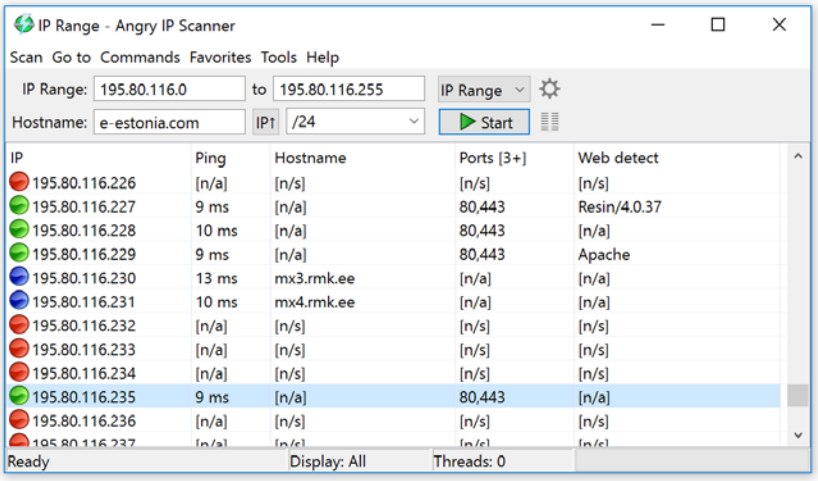
2.6.2 Angry IP scanner

"Angry IP scanner je veľmi rýchly scanner IP adries a portov s otvoreným zdrojovým kódom (open source). Môže skenovať IP adresy v ľubovoľnom rozsahu, ako aj všetky ich porty. Je priehľadný a jednoduchý.

Angry IP scanner zistí odozvu (ping) každej IP adresy a ak daná adresa odpovedá, potom podľa voliteľných nastavení rozpozná názov hostiteľa, určí MAC adresu, skenuje porty atď. Množstvo zhromaždených dát o každom hostiteľovi môže byť rozšírené pluginmi.

Obsahuje aj ďalšie funkcie, ako sú informácie o systéme NetBIOS (názov počítača, názov pracovnej skupiny a aktuálne prihlásený používateľ systému Windows), obľúbené rozsahy adries IP, detekcia webového servera atď.

Výsledky skenovania je možné uložiť do CSV, TXT, XML alebo IP-Port súborov. Aby sa zvýšila rýchlosť skenovania, používa sa viacvláknový prístup: pre každú naskenovanú IP adresu sa vytvorí samostatné skenovacie vlákno." [19]



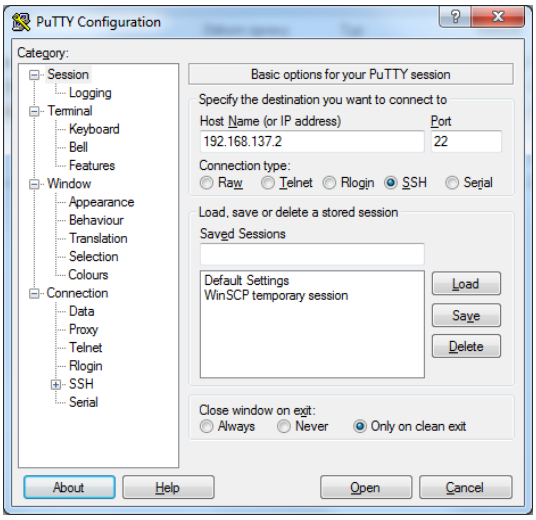
Obr. 10 Angry IP scanner [20]

2.6.3 PuTTY

PuTTY je populárny klient protokolov ako napríklad SSH a terminál pre spojenia cez sériový port. Pôvodne bol dostupný iba pre systém Windows a v súčasnosti existuje aj pre rôzne UNIXové platformy. PuTTY je open source softvér, šírený pod MIT licenciou, ktorý je k dispozícii so zdrojovým kódom. Pôvodne ho napísal Simon Tatham a momentálne je vyvíjaný malou skupinou dobrovoľníkov. [21]

Niektoré z vlastností PuTTY [21]:

- ukladanie informácií o serveroch a nastaveniach;
- výber protokolu SSH a šifrovacieho kľúča;
- klient príkazového riadku SCP a SFTP;
- možnosť forwardovania portov.



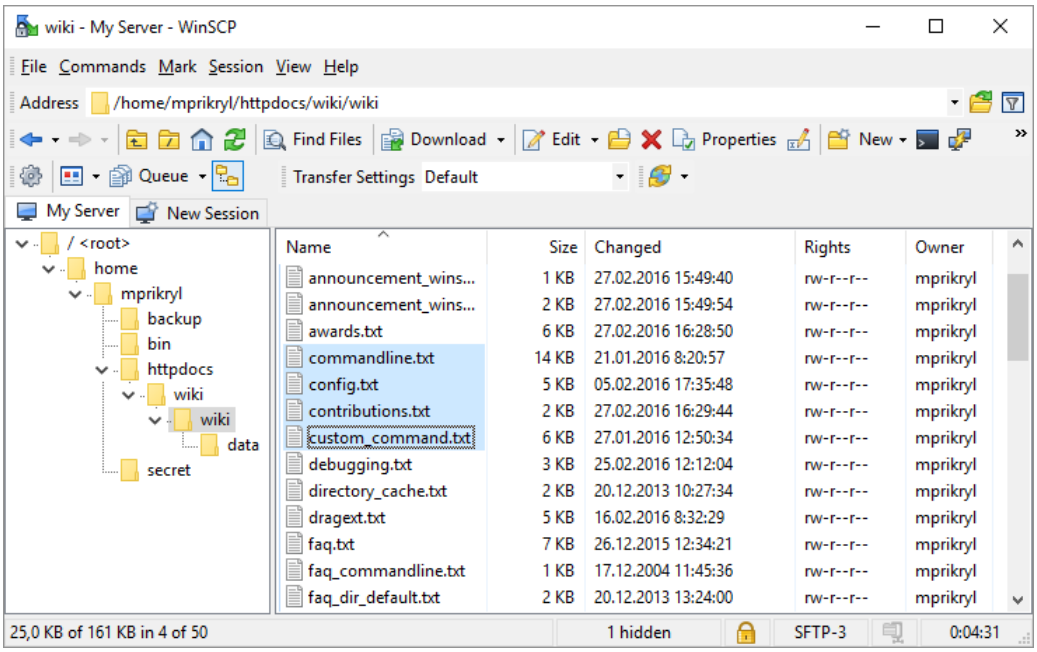
Obr. 11 PuTTY [21]

2.6.4 WinSCP

WinSCP 5.13 je SFTP klient a FTP klient s otvoreným zdrojovým kódom (open source) pre systém Windows. Zároveň podporuje aj starší protokol SCP. Jeho hlavnou funkciou je bezpečné prenášanie súborov medzi vaším počítačom a vzdialeným serverom.

Program WinSCP zvláda všetky základné operácie so súbormi, ako je sťahovanie a nahrávanie. Taktiež je možné súbory a priečinky premenovávať, vytvárať nové priečinky, meniť vlastnosti súborov a priečinkov a vytvárať symbolické odkazy.

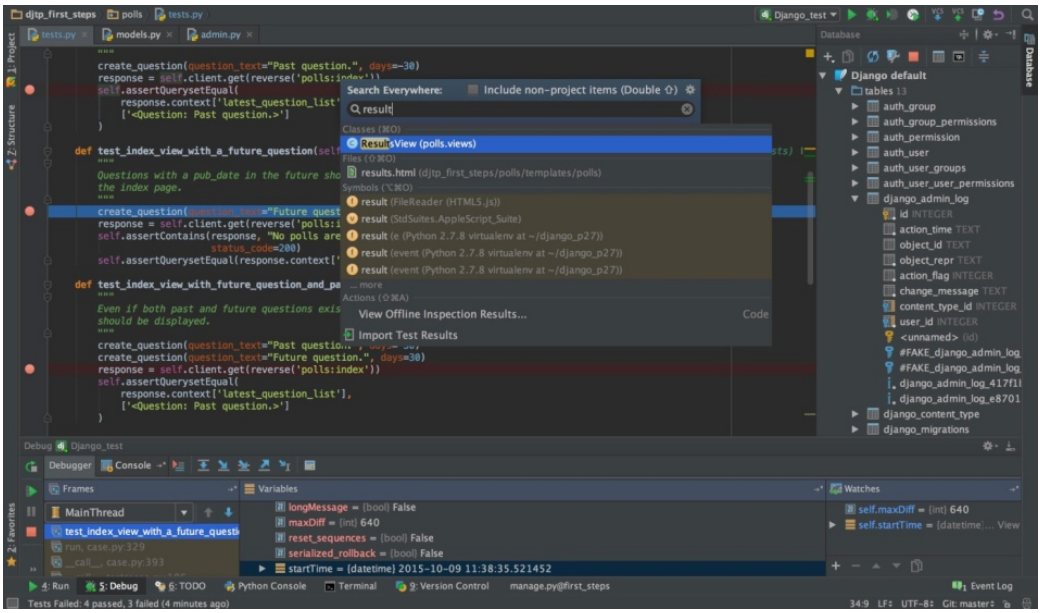
Pomocou programu WinSCP je možné pripojiť sa k serveru SSH (Secure Shell) so službou SFTP (SSH File Transfer Protocol) alebo SCP (Secure Copy Protocol) alebo k serveru FTP (File Transfer Protocol). [22]



Obr. 12 WinSCP [23]

2.6.5 Pycharm

"PyCharm 4.0.6 je profesionálne prispôsobiteľné viacplatformové vývojové prostredie (IDE) určené pre programovanie v jazyku Python 2 alebo 3 a taktiež umožňuje vývoj webových stránok v jazykoch HTML, JavaScript a iné.



Obr. 13 Pycharm [24]

Poskytuje inteligentné dokončenie kódu, kontrolu syntaxe, zvýraznenie chýb a rýchle opravy chýb, spolu s automatickými úpravami kódu (refactor) a bohatými navigačnými schopnosťami.

PyCharm ponúka špecifickú podporu pre moderné webové vývojové rámce ako Django, Flask, Google App Engine, Pyramid a web2py.

Okrem Pythonu podporuje jazyky JavaScript, CoffeeScript, TypeScript, Cython, SQL, HTML / CSS, jazyky šablón, AngularJS, Node.js a ďalšie.

Podporuje spúšťanie, ladenie, testovanie a nasadzovanie aplikácií na vzdialených počítačoch alebo virtuálnych počítačoch, diaľkovými tlmočníkmi, integrovaným terminálom SSH a integráciou Docker-u a Vagrant-u.

PyCharm poskytuje obrovskú kolekciu nástrojov: integrovaný debugger, Python profiler, vstavaný terminál a integráciu s hlavnými VCS a vstavanými databázovými nástrojmi." [24]

2.6.6 Eagle

"Editor plošných spojov EAGLE je užívateľsky prívetivý a výkonný nástroj pre návrh dosiek plošných spojov (DPS, PCB). Názov EAGLE je skratka, pochádzajúca z pôvodného názvu Easily Applicable Graphical Layout Editor.

Program sa skladá zo štyroch hlavných modulov:

- editor spojov;
- editor schém;
- autorouter;
- CAM Processor.

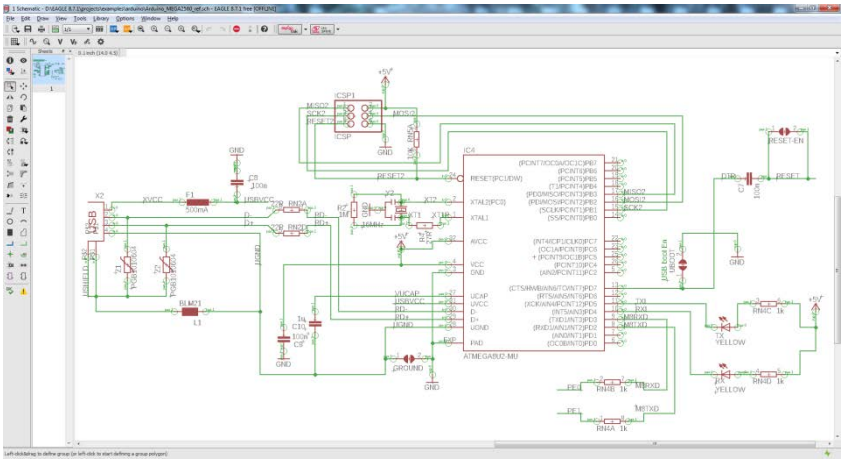
Tieto moduly sú ovládané z jedného užívateľského prostredia. Preto nie je treba konvertovať netlisty medzi schémami a doskami.

Vlastnosti programu:

- dopredná a spätná anotácia v reálnom čase;
- nápoveda orientovaná podľa obsahu;
- žiadna hardvérová ochrana programu;
- viacnásobné okná pre dosku, schému a knižnicu;
- výkonný užívateľský jazyk;
- integrovaný textový editor;
- dostupný pre Windows, MAC a Linux.

Editor schém:

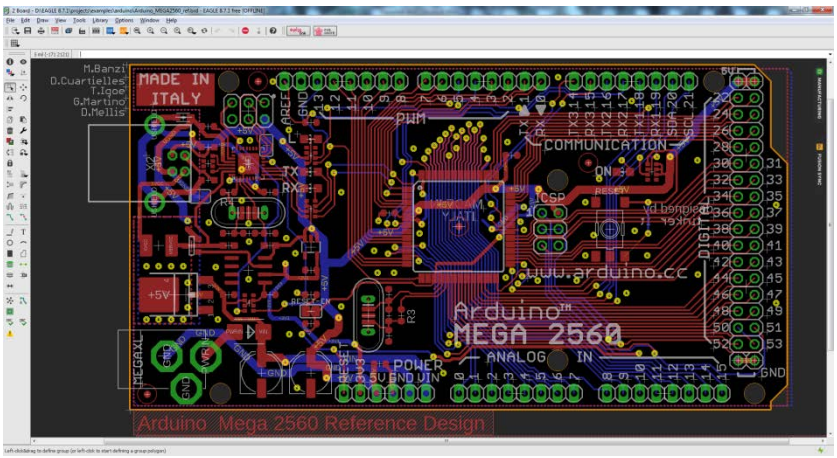
- až 99 listov jednej schémy;
- kontrola elektrických pravidiel zapojenia;
- prehadzovanie hradiel a pinov;
- vytvorenie dosky zo schémy jediným príkazom.



Obr. 14 Eagle - editor schém

Editor spojov:

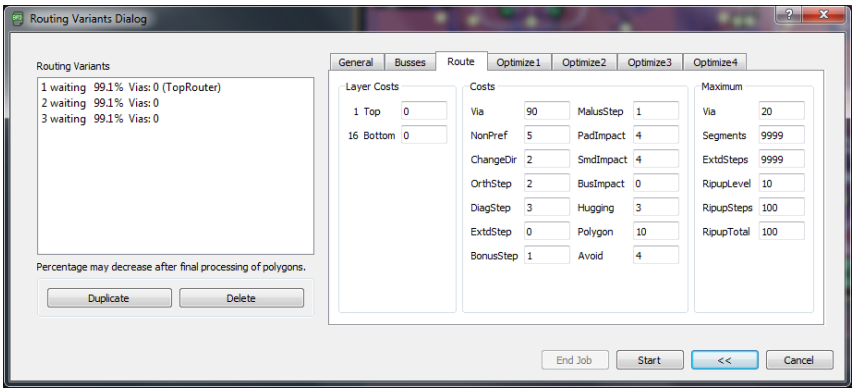
- najväčší rozmer výkresu 1.6 x 1.6m (64 x 64 inch);
- rozlíšenie 1 / 10.000 mm (0,1 mikrónu);
- až 16 signálových vrstiev;
- klasické i SMD súčiastky;
- dodáva sa s plnou sadou knižníc súčiastok;
- jednoduché vytváranie vlastných súčiastok v plne integrovanom editore knižníc;
- funkcie vpred / vzad pre ľubovoľnú editačný príkaz, do ľubovoľnej hĺbky;
- skriptové súbory pre dávkové spracovanie príkazov;
- pomedenie plôch;
- funkcie kopírovať a vložiť pre kopírovanie kompletných častí výkresu;
- kontrola pravidiel návrhu;
- oblúky v spojoch;
- meandre pre rýchle obvody.



Obr. 15 Eagle - editor spojov

Autorouter má:

- ripup&retry router;
- až 16 signálových vrstiev;
- stratégiu prepojovania nastaviteľnú užívateľom pomocou váhových faktorov.

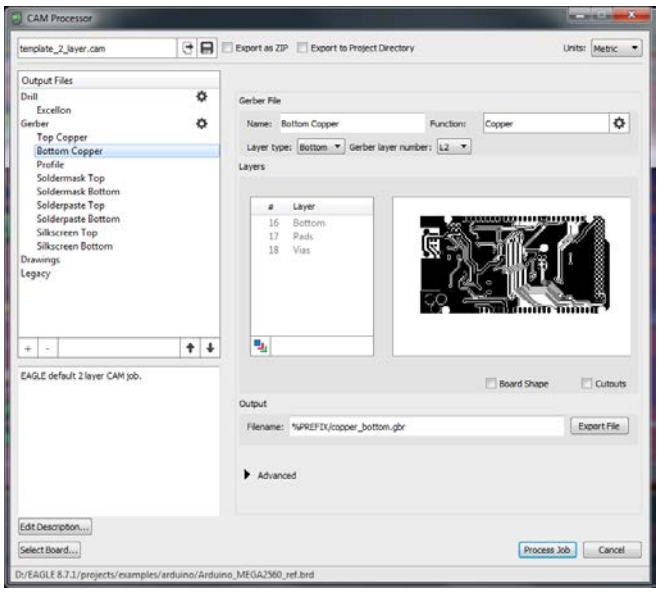


Obr. 16 Eagle - autorouter

CAM Processor podporuje:

- postscript;
- perové plotre;
- plotre Gerber;
- súbory pre vŕtačky Excellon a Sieb&Meyer;
- vlastné výstupné zariadenie, ľahko konfigurovateľné pomocou ASCII súborov.

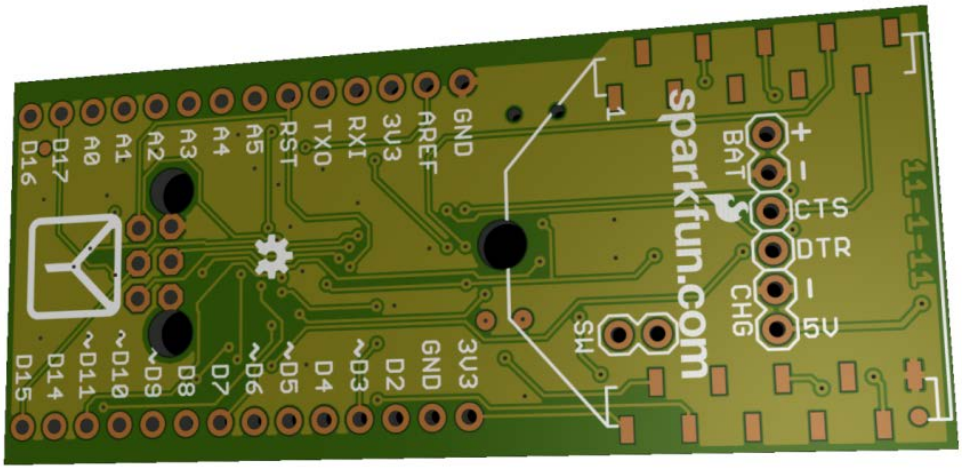
" [25]



Obr. 17 Eagle - CAM Processor

2.6.7 MayhewLab - 3D Gerber Viewer

Pomocou tohto prehliadača je možné nahrať svoje Gerber súbory a mať 360 stupňový 3D pohľad na vzhľad svojej DPS. Program umožňuje základné funkcie ako približovanie a odďaľovanie dosky, vypnutie a zapnutie jednotlivých vrstiev (meď, nepájavá maska...), natočenie dosky a taktiež export obrázku aktuálneho pohľadu na dosku plošných spojov. [26]



Obr. 18 Gerber Viewer - náhľad na DPS [27]

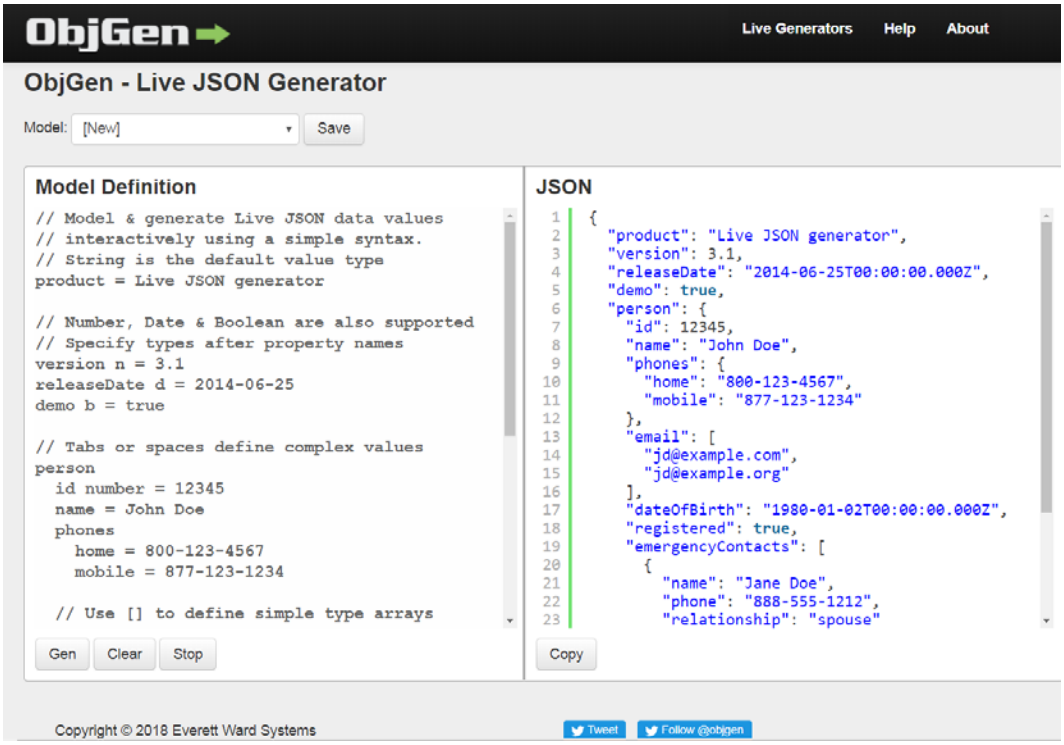
2.6.8 ObjGen - Live JSON Generator

ObjGen Live JSON generátor umožňuje používateľovi interaktívne tvarovať, vytvárať a generovať JSON štruktúry a údaje. Z jednoduchého textového modelu zadaného používateľom sa dáta JSON dynamicky generujú vždy, keď používateľ zmení model.

Použitím jednoduchej syntaxe môže používateľ rýchlo vytvoriť model na generovanie rozsiahlych JSON štruktúr. [28]

Dátové typy sú špecifikované jednoduchou konvenciou [29]:

- s (string) = reťazec, text;
- n (number) = číslo;
- d (date) = dátum / čas;
- b (bool) = logická premenná;
- [] = definícia poľa.



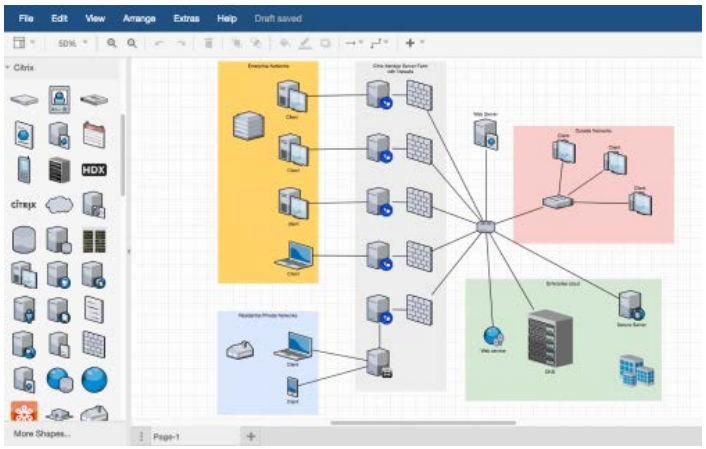
Obr. 19 ObjGen Live JSON generátor

2.6.9 Draw.io

Draw.io je open source technologický zásobník (tech stack) pre aplikácie na vytváranie a grafické spracovanie diagramov. Z hľadiska počtu koncových užívateľov je celosvetovo najrozšírenejšou webovou aplikáciou. Draw.io je spomedzi nástrojov na vytváranie diagramov najflexibilnejší a najviac zameraný na ochranu súkromia. [30]

Hlavné typy diagramov a vizualizácií [30]:

- vývojový diagram
- diagram procesov
- graf signálových tokov
- UML diagram
- stromový diagram
- sieťový diagramy
- myšlienkové mapy
- Vennov diagram
- stĺpcový diagram
- sekvenčný diagram



Obr. 20 Draw.io - ukážka diagramu [31]

2.7 Programovacie jazyky

Pred tvorbou samotného programu zavlažovacieho systému som sa rozhodoval medzi rôznymi programovacími jazykmi. Najrozšírenejšie sú C/C++, python, perl, erlang a iné. Pre tvorbu vizualizácie boli k dispozícii dve základné možnosti. Spraviť vizualizáciu pomoc dotykového panela alebo pomocou webovej stránky. V prípade webovej stránky sa ponúkali ako možné programovacie jazyky HTML pre základný vzhľad s nadstavbami PHP, Javascript alebo Jquery.

2.7.1 Programovací jazyk Python

Cieľom tejto podkapitoly ja stručne charakterizovať programovací jazyk Python a následne popísať vybrané knižnice, ktoré som využíval pri tvorbe zavlažovacieho programu.

"Python je moderný programovací jazyk, ktorého popularita stále rastie. Jeho autorom je Guido van Rossum (vymyslel ho v roku 1989), používajú ho napr. v Google, YouTube, Dropbox, Mozilla, Quora, Facebook, Rasperry Pi atď. Na mnohých špičkových univerzitách sa učí ako úvodný jazyk, napr. MIT, Carnegie Mellon, Berkeley, Cornell, Caltech, Illinois atď. Beží na rôznych platformách, napr. Windows, Linux, Mac. Je to freeware a tiež open source.

Na rozdiel od mnohých iných jazykov, ktoré sú kompilačné (napr. Pascal, C/C++, C#) je Python interpreter. To znamená, že interpreter nevytvára spustiteľný kód (napr. .exe súbor vo Windows), na spustenie programu musí byť v počítači nainštalovaný Python, interpreter umožňuje aj interaktívnu prácu s prostredím." [32]

Hlavné vlastnosti jazyka Python [32]:

- Veľmi jednoduchá a dobre čitateľná syntax a keďže Python je aj vysoko interaktívny, je veľmi vhodný aj pre vyučovanie programovania.
- Na rozdiel od staticky typovaných jazykov, pri ktorých je treba dopredu deklarovať typy všetkých dát, je dynamicky typovaný, čo znamená, že neexistujú deklarácie.
- Python obsahuje pokročilé črty moderných programovacích jazykov, napr. podpora práce s dátovými štruktúrami, objektovo-orientovaná tvorba softvéru...
- Je to univerzálny programovací jazyk, ktorý poskytuje prostriedky na tvorbu moderných aplikácií, ako je analýza dát, spracovanie médií, sieťové aplikácie...
- Python má obrovskú komunitu programátorov a expertov, ktorí sú ochotní svojimi radami pomôcť aj začiatočníkom

2.7.2 Programovací jazyk HTML

"HTML je skratkou pre anglický názov Hypertext Markup Language. HTML je jedným z prvých jazykov používaných pre programovanie internetových stránok, od svojho vzniku prešiel mnohými modifikáciami až do svojej dnešnej podoby.

HTML dokumenty možno vytvárať v akomkoľvek textovom editore. Jedinou podmienkou je možnosť uloženia dát v čistom texte, teda ASCII súbore. Na úpravu je vhodné použiť napríklad poznámkový blok pre Windows. Možno samozrejme tiež použiť nejaké WYSIWYG editory, napríklad Dreamweaver alebo Microsoft FrontPage.

HTML dokument ohraničený takzvanými značkami (tag) sa skladá z hlavičky ohraničenej značkami a tela ohraničeného značkami. Hlavička obsahuje informácie o dokumente, ktoré nie sú v prehliadači zobrazované (sú určené pre rôzne internetové vyhľadávače) a ďalej informácie o vkladaných súboroch obsahujúci kód JavaScriptu alebo kaskádové štýly. Ďalšou dôležitou informáciou obsiahnutou v hlavičke dokumentu je titulok stánky, ktorý je zobrazovaný nielen na hornej lište prehliadača, ale aj na lište operačného systému. V tele dokumente sú zahrnuté všetky značky, ktoré editujú vlastnú stránku. Ide napríklad o tabuľky, vrstvy, odseky, prípadne značky editujúce priamo daný text." [33]

2.7.3 Programovací jazyk Java script

"JavaScript je interpretovaný, multiplatformný programovací jazyk so základnými objektovo orientovanými schopnosťami. Univerzálny jadro jazyka bolo vložené do webových prehliadačov a rozšírené pridaním objektov reprezentujúci okno prehliadača a jeho obsah. Klientská verzia JavaScriptu umožňuje vložiť do webových stránok uskutočniteľný obsah. Stránky sa tak stávajú dynamickými - môžu obsahovať najrôznejšie programy, ktoré komunikujú s užívateľom, riadi prehliadač, či dynamicky vytvára obsah HTML. Pri práci skriptu nie je potrebné kontaktovať server, všetku prácu skriptu zabezpečuje sám prehliadač.

Jadro jazyka syntakticky pripomína jazky C ++ a Javu. Avšak syntaxou ich podobnosť končí. JavaScript je jazyk bez typovej kontroly, čo znamená, že premenné nemusí mať špecifikovaný typ. A navyše JavaScript je čisto interpretovaný jazyk, na rozdiel napríklad od kompilovaných C a C ++ a na rozdiel od Javy, ktorá je pred interpretáciou vybudovaný do bajtového kódu." [34]

Základné vlastnosti JavaScriptu môžeme teda zhrnúť do nasledujúcich prívlastkov [34]:

- interpretovaný - jazyk JavaScript sa nemusia kompilovať;
- objektový - využíva objektov prehliadača a zabudovaných objektov;
- závislý na prehliadači - funguje však vo väčšine používaných prehliadačov;
- case senzitívne - tzn., že záleží na veľkosti písma v zápise;
- syntaxou je blízky jazykom C, Java a im podobným.

2.7.4 Programovací jazyk Jquery

JQuery je JavaScriptová knižnica (sada funkcií), ktorá uľahčuje prácu s JavaScriptom. Kladie dôraz na jednoduchosť, čitateľnosť a rýchlosť. Je multiplatformová (funguje na viacerých operačných systémoch) a je dostupná zdarma. Čo všetko jQuery dokáže ? Lepšia by bola otázka, čo všetko jQuery nevie. Všetko, čo urobíte v javascripte urobíte pomocou jQuery asi s tretinou kódu a navyše ešte veľa vecí navyše. [35]

Hlavné vlastnosti JQuery [35]:

- manipulácia s DOM objektmi;
- udalosti (events);
- manipulácia s CSS;
- selektory;
- efekty (pomocou preddefinovaných funkcií);
- animácie - jednoduchá tvorba, veľmi efektný aj efektívny výsledok;
- Ajax - načítanie obsahu serveru bez nutnosti obnovenie stránky;
- utility - napr. informácie o prehliadači alebo funkcie each.

3 SOFTWARE

"Na to aby počítač mohol fungovať a vykonávať požadované operácie, musí mať okrem technických prostriedkov aj programové vybavenie – softvér. Každá činnosť počítača, od obyčajného sčítania až po spracovanie zložitých aritmetických a logických operácií, musí byť naprogramovaná pomocou inštrukcií, ktorým počítač rozumie a ktoré určujú, ako sa jednotlivé časti počítača majú správať. Tieto elementárne inštrukcie sú zlučované do celkov a vytvárajú programy.

Programom je operačný systém počítača, ale aj textový editor, tabuľkový kalkulátor, aplikácia na spracovanie účtovníctva, počítačová hra a pod." [36]

Programové vybavenie počítačov delíme do niekoľkých skupín [36]:

- Základný softvér – operačné systémy a ich nadstavby, ovládače.
- Kancelárske aplikácie – textové editory, tabuľkové kalkulátory, databázy, grafické editory, prezentačné programy a pod.
- Aplikačný softvér – programy, ktoré riešia konkrétne úlohy, napr. účtovníctvo, riadenie výroby, skladové hospodárstvo a pod."

V tejto práci je po softwarovej stránke popísaný spôsob pripojenia a programovania Raspberry Pi, tvorba riadiaceho programu zavlažovacieho systému písaného v jazyku python a taktiež rozbor grafického rozhrania vytvoreného pomocou webovej stránky.

Pri prvom stretnutí s PC som využíval vtedy bežný MS-DOS, následne som používal výhradne operačné systémy od Microsoftu. Keďže Raspberry Pi je založené na Linuxovej platforme, bolo pre mňa vcelku náročné pochopiť a zvyknúť si na nový spôsob ovládania tohto diametrálne odlišného konceptu operačného systému. O to horšie bolo, keď som sa rozhodol používať operačný systém bez grafického rozhrania.

Pred písaním tejto práce som mal možnosť programovať v rôznych jazykoch. Z tých nízkoúrovňových to bol napríklad asembler. Ďalej mám skúsenosti s jazykmi C/C++ alebo s programovaním v matlabe a rôznych PLC systémov. Programovanie v pythone mi prišlo po chvíli vcelku jednoduché. Python má obrovskú komunitu a každý problém, na ktorý som narazil už niekto predo mnou úspešne vyriešil.

Ďalšou veľkou prekážkou bolo programovanie grafického rozhrania, ktoré som sa rozhodol vytvoriť pomocou webovej stránky. Pre komunikovanie s Raspberry samozrejme nestačilo základné HTML ale bolo potrebné rozšíriť web stránku pomocou Javascriptu a Jquery.

Spomínané programovacie jazyky som sa teda musel naučiť tak nejako za pochodu, počas realizovania tejto práce. Preto budú možno niektoré časti vytvorených programov nie príliš jednoducho a efektívne napísané. Pri tvorení softwaru ako takého som sa snažil dbať na funkčnosť a robustnosť.

3.1 Nastavenia Raspberry Pi

Pred tvorbou samotného programu a vizualizácie bolo potrebné do Raspberry Pi nahrať operačný systém, pripraviť ho na prvé použitie a prepojiť ho s počítačom. Následne bolo možné vykonať základné nastavenia, povoliť potrebné periférie a pridať dôležité knižnice. V neposlednom rade sa museli prideliť rôzne prístupové práva, nastaviť statickú IP adresu pre jednoduchšiu komunikáciu a taktiež upraviť niekoľko konfiguračných súborov.

Celý postup sa pokúsim zhrnúť do niekoľkých podkapitol a taktiež vysvetliť prečo a ako som jednotlivé kroky vykonal a z akých zdrojov som čerpal informácie.

Pri väčšine nastavení som sa inšpiroval veľmi prehľadným a užitočným návodom, ktorý vytvorili kolegovia z robotiky na našom ústave UAMT do predmetu BPRP - Robotika a počítačové vidění z roku 2016 [37] a z roku 2017 [38].

3.1.1 Operačný systém

"Ako prvé som musel do Raspberry Pi nahrať operačný systém. Potreboval som k tomu micro SD kartu, čítačku kariet, počítač a samozrejme zdroj (napríklad nabíjačka od telefónu s micro USB konektorom). Z oficiálnej stránky Raspberry [39] som stiahlo súbor Raspbian Stretch Lite a po jeho rozbalení vznikol obraz operačného systému s príponou ".img".

Ďalej stiahneme aplikáciu Win32DiskImager [40], ktorú je potrebné rozbaliť a nainštalovať. Zasunieme micro SD kartu do čítačky kariet, spustíme aplikáciu, vyberieme súbor s obrazom Raspbianu, zvolíme správne písmeno jednotky a zapíšeme obraz na SD kartu. Po zapísaní obrazu sa na micro SD karte objavia súbory.

Pre nasledovné pripojenie sa k Raspberry pomocou SSH protokolu je nutné toto rozhranie povoliť, keďže je od 11/2016 defaultne zakázané. Pre aktiváciu SSH po boote raspberry je nutné na karte vytvoriť súbor ssh, bez prípony, všetky písmená malým. Najjednoduchšie to urobíme cez príkazový riadok pomocou príkazu "echo >E:\ssh".

SD kartu vyjmeme (bezpečne odobrať !.) a vložíme do Raspberry. Pripojíme napájanie Raspberry, mala by sa rozsvietiť oranžová LED, a poblikávať zelená - to značí že Raspberry bootuje. Počkáme až zelená LED dobliká a sme pripravení oživiť software. Napájanie raspberry v tomto kroku neodpojujeme! Ak dôjde k strate napájania v tomto kroku, bude potrebné znova vytvoriť súbor ssh na karte." [38]

3.1.2 Prvotné nastavenia

"Ako náhle bude Raspberry v pokoji, môžeme sa k nemu pripojiť pomocou eternetového kábla. Programom Angry IP scanner [41] je možné zistiť IP adresu, ktorú mu DHCP server pridelil. Skenovaný rozsah adries zistíme podľa aktuálne pridelenej IP adrese počítača v danej lokálnej sieti. Raspberry sa bude volať Raspberry Pi a podľa tohto mena ho spoznáme.

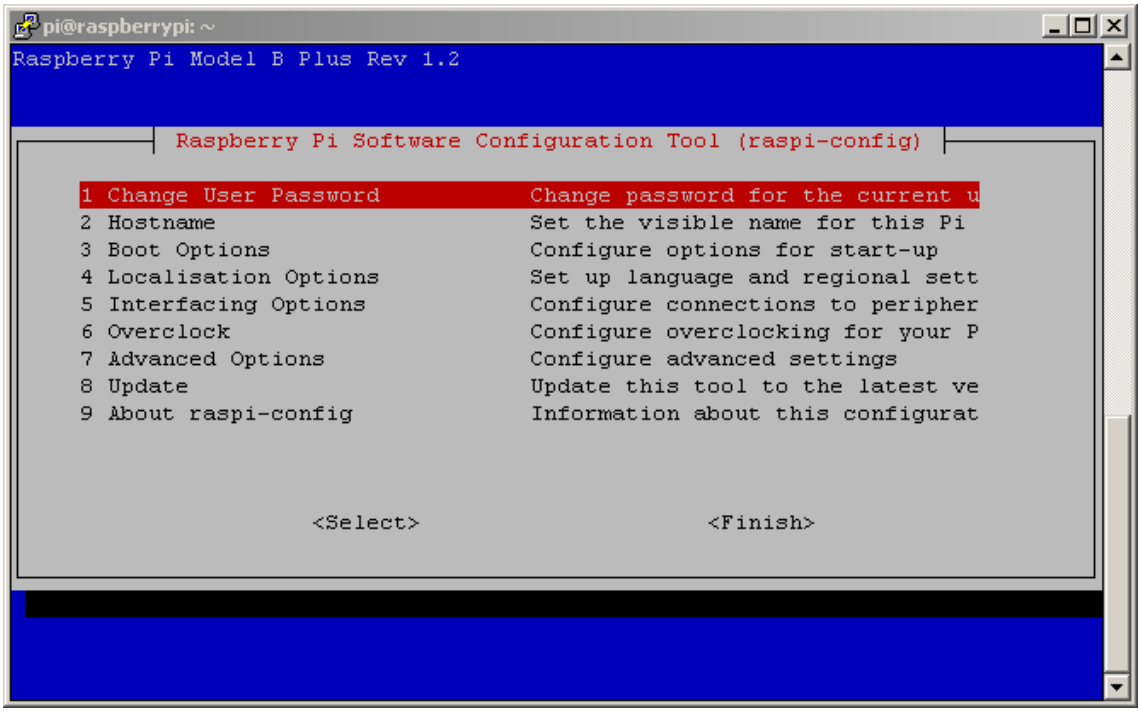
Pre pripojenie sa k Raspberry použijeme program Putty. Po spustení hneď na prvom dialógovom okne zadáme IP adresu získanú programom Angry IP scanner a

klikneme na Connect. Po úspešnom pripojení sa nás malinovo čierne okno pýta na meno *pi* a potom na heslo *raspberry*. Po prihlásení nás malina prívetivo uvíta.

Ako prvú vec, ktorú treba urobiť, než začneme ďalej pracovať je konfigurácia vlastného systému. Zadáme teda do príkazového riadka:

- sudo raspi-config

Zjaví sa okno:



Obr. 21 konfiguračné menu Raspberry [38]

A budeme postupovať podľa jednotlivých bodov zvrchu:

- V menu 1 Change User Password si zmeníme heslo.
- V menu 2 Hostname si nastavíme meno nášho Pi (nepovinné)
- V menu 3 Boot Options si nastavíme iba položku B2 Wait for network at Boot na hodnotu No. Tým zaistíme, že nám bude program bežať aj bez siete.
- V menu 4 Localisation Options nebudeme nastavovať češtinu / slovenčinu - výsledné chybové hlášky systému sú potom kryptické a ťažko sa hľadá závada.
- V menu 5 Interfacing options povolíme try položky: P2 SSH, P4 SPI, P5 I2C

To je všetko v tomto konfiguračnom programe. Zvolíme teda Finish a na dotaz reštartujeme." [38]

3.1.3 Knižnice a balíčky

"Raspbian je klonom Debianu, takže všetok softvér sa inštaluje ako balíčky cez aplikáciu apt-get. Najprv teda nainštalujeme všetky aktualizácie k operačnému systému, aby sme začínali s rovnakými verziami balíčkov:

- sudo apt-get update && sudo apt-get upgrade

Po aktualizácii OS je možné pristúpiť k inštalácii jednotlivých balíčkov. To je možné vykonať príkazom:

- sudo apt-get install <balíček>

Je možné inštalovať aj viac balíčkov naraz. Stačí ich oddeliť medzerou. V mojej práci budeme používať nasledujúce balíčky (a treba ich teda nainštalovať):

- **apache2** HTTP web server;
- **i2c-tools** pomôcky príkazového riadku pre I^2C zariadenia;
- **python-spidev** modul pre komunikáciu s SPI zariadeniami;
- **python-smbus** modul pre komunikáciu s I^2C zariadeniami;
- **python-pip** nástroj na inštaláciu a správu balíčkov;
- **git** verzovací systém.

Spomínané balíčky a nástroje som samozrejme inštaloval z rôznych dôvodov. V nasledujúcich podkapitolách uvediem dôvody a postupy inštalácií." [38]

3.1.3.1 Wiring Pi

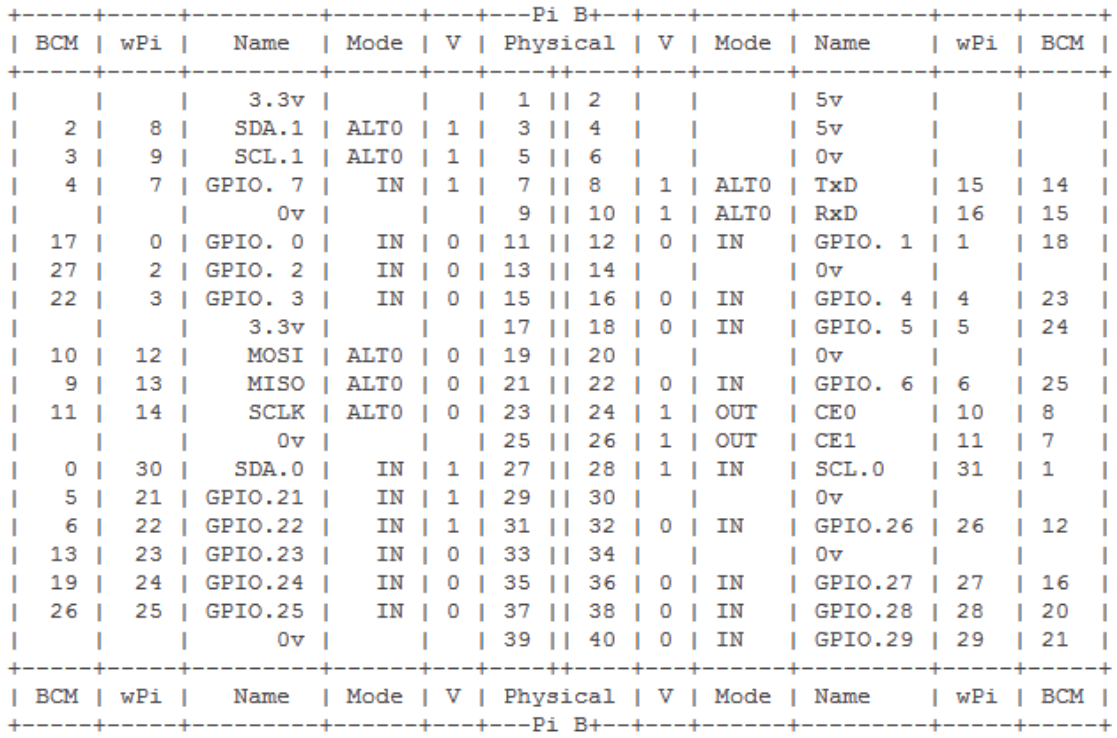
Najskôr som potreboval ovládať digitálne vstupy a výstupy, ktoré sú na Raspberry Pi vyvedené pomocou 40 pinového konektoru. K tomu som použil knižnicu WiringPi.

Postup inštalácie a overenie funkčnosti bol použitý zasa zo stránky robotiky [37].

- git clone git://git.drogon.net/wiringPi
- cd wiringPi
- ./build

Pre overenie správnej funkčnosti použijeme príkaz, ktorý nám zobrazí očíslovaný a popísaný konektor:

- gpio readall



BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
		3.3v			1 || 2			5v		
2	8	SDA.1	ALT0	1	3 || 4			5v		
3	9	SCL.1	ALT0	1	5 || 6			0v		
4	7	GPIO. 7	IN	1	7 || 8	1	ALT0	TxD	15	14
		0v			9 || 10	1	ALT0	RxD	16	15
17	0	GPIO. 0	IN	0	11 || 12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13 || 14			0v		
22	3	GPIO. 3	IN	0	15 || 16	0	IN	GPIO. 4	4	23
		3.3v			17 || 18	0	IN	GPIO. 5	5	24
10	12	MOSI	ALT0	0	19 || 20			0v		
9	13	MISO	ALT0	0	21 || 22	0	IN	GPIO. 6	6	25
11	14	SCLK	ALT0	0	23 || 24	1	OUT	CE0	10	8
		0v			25 || 26	1	OUT	CE1	11	7
0	30	SDA.0	IN	1	27 || 28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29 || 30			0v		
6	22	GPIO.22	IN	1	31 || 32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33 || 34			0v		
19	24	GPIO.24	IN	0	35 || 36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37 || 38	0	IN	GPIO.28	28	20
		0v			39 || 40	0	IN	GPIO.29	29	21

Obr. 22 Raspberry Pi B+ konektor - WiringPi

3.1.3.2 Apache

Vizualizáciu som sa rozhodol spraviť pomocou webovej stránky, preto bolo potrebné nainštalovať webový server, pomocou ktorého bude neskôr vytvorená vizualizácia sprístupnená pre klientov na lokálnej sieti, časom možno aj cez internet.

Podľa stránky Raspberry Pi [42] sú najčastejšie používané webové servy Apache a NGINX, ktoré majú veľmi podobné vlastnosti.

Apache je najrozšírenejší softvér pre webové servery. Apache, vyvinutý a udržiavaný Apache Software Foundation, je open source softvér dostupný zdarma. Beží na 67% všetkých webových serverov na svete. Je rýchly, spoľahlivý a bezpečný. [43]

Po nainštalovaní balíčku podľa návodu na oficiálnej stránke Raspberr Pi [44] je možné skontrolovať funkčnosť serveru zadaním lokálnej adresy do prehliadača. Mal by nás privítať nápis "It works!".

- sudo apt-get install apache2

Nakoniec je potrebné ešte pomocou príkazu:

- sudo chown -R pi /var/www

upraviť prístupové práva, aby sme neskôr mohli do priečinku s adresou /var/www/html nakopírovať našu webovú stránku.

3.1.3.3 Tornado

Ako bolo už spomenuté, pre vizualizáciu bola použitá webová stránka, s ktorou bolo potrebné komunikovať. Najjednoduchšou možnosťou komunikácie by bolo manuálne alebo automatické znovu načítavanie stránky, ďalej je možné použitie metódy zvanej "short pooling", kedy sa v pravidelných intervaloch dotazujeme serveru, či nie sú k dispozícii nové dáta. Trochu sofistikovanejší prístup poskytuje modifikovaný variant s označením "long pooling", kedy sa čaká pokiaľ neprídu nové dáta a potom sa spojenie ukončí a znovu nadviaže. Ďalej je možné využiť metódu zvanú " Server-sent events", ktorá ako názov napovedá nadviaže dlhodobú komunikáciu a server nepravidelne posiela nové dáta na základe udalostí. Poslednou najzložitejšou ale pre moje účely najvhodnejšou možnosťou bolo použiť metódu nazvanú "WebSockets". Na začiatku sa nadviaže dlhodobé obojsmerné komunikačné spojenie, cez ktoré môže posielať dáta aj klient aj server. [45]

Implementácia tohto druhu komunikácie nie je triviálna, preto som sa rozhodol použiť niektorý z dostupných webových rámcov. Na výber bolo z viacerých variant ako napríklad Flask, Tornado a podobne. Po niekoľkých týždňoch hľadania a porovnávania, testovania a zoznamovania sa s princípom fungovania tohto spôsobu komunikácie sa mi podarilo nájsť funkčný návod na komunikáciu webovej stránky s python skriptom. [46]

Preto som využil niektoré knižnice a funkcie webového rámca Tornado. Nainštaloval som ho pomocou príkazu:

- pip install tornado

Tornado je webová rámec Pythonu a asynchrónna sieťová knižnica, pôvodne vyvinutá v systéme FriendFeed. Používaním neblokujúceho sieťového I/O môže Tornado škálovať až desiatky tisíc otvorených pripojení, čo je ideálne pre "long pooling", WebSockety a ďalšie aplikácie, ktoré vyžadujú dlhodobé pripojenie ku každému používateľovi [47].

3.1.4 Konfigurácia I^2C modulov

Počas navrhovania zavlažovacieho systému som sa rozhodol využiť dve zariadenia pripojené pomocou I^2C rozhrania. Konkrétne modul hodín reálneho času DS3231 a senzor teploty a vlhkosti vzduchu DHT12.

Pred samotným testovaním I^2C modulov je nutné toto rozhranie v Raspberry povoliť. Pokiaľ potrebujeme zariadenia len otestovať, je to možné urobiť jednoducho pomocou konfiguračného menu, ktoré som už raz použil:

- sudo raspi-config

V prípade, že chceme, aby bolo I^2C rozhranie povolené ihneď po štarte zariadenia je potrebné upraviť ďalší konfiguračný súbor. Pomocou príkazu:

- sudo nano /etc/modules

Otvoríme konfiguračný súbor a na koniec súboru pridáme nasledovné riadky:

- i2c-bcm2708
- i2c-dev

Riadok s povolením driveru bcm2708 je potrebné pridať kvôli DHT12 modulu.

3.1.4.1 Konfigurácia RTC modulu

Konfigurácia a otestovanie RTC modulu s označením DS3231 bolo vcelku bezproblémové. Postupoval som podľa tohto návodu [48].

Pomocou príkazu:

- sudo nano /boot/config.txt

som doplnil v konfiguračnom súbore tento riadok:

- dtoverlay=i2c-rtc,ds3231

Následne som upravil druhý konfiguračný súbor pomocou príkazu:

- sudo nano /lib/udev/hwclock-set

V tomto súbore bolo potrebné zakomentovať nasledovné riadky:

- #if [-e /run/systemd/system] ; then
- # exit 0
- #fi

Po reštarte systému bolo možné otestovať pomocou nasledovného príkazu správnu funkčnosť hodín reálneho času:

- sudo hwclock -r

V prípade, že nám systém odpovie zmysluplným časom, ktorý bude pravdepodobne neaktuálny, môžeme dátum a čas v module nastaviť pomocou ďalšieho príkazu:

- sudo hwclock -s

Je potrebné si uvedomiť, že pred použitím príkazu na prestavenie času v RTC module je potrebné mať správne nastavený dátum a čas v Raspberry. V prípade, že sme správne

pripojený k internetu, stačí overiť že máme správne nastavenú časovú zónu. Ak nemáme prístup k internetu je potrebné nastaviť dátum, čas a časové pásmo ručne pomocou už spomenutého príkazu:

- sudo raspi-config

3.1.4.2 Konfigurácia DHT12 modulu

Konfigurácia a otestovanie DHT12 modulu bolo časovo oveľa náročnejšie. Postupoval som podľa tohto príspevku na Raspberry fóre [49]. Na začiatok bolo potrebné stiahnuť pomocou nasledovného príkazu potrebnú knižnicu s názvom smbus a pomocný nástroj na prácu s I^2C zariadeniami, takzvaný i2c-tools:

- sudo apt-get install -y python-smbus i2c-tools

V spomínanom príspevku bol uvedený aj vzorový kód na otestovanie tohto modulu. Pomocou nástroju i2c-tools bolo potrebné najskôr zistiť adresu použitého I^2C zariadenia. Pomocou nižšie uvedeného príkazu sa nám vykreslí tabuľka práve pripojených zariadení:

- i2cdetect -y 1

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- 5c -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Obr. 23 Tabuľka pripojených I^2C zariadení

Z tabuľky je možné vidieť, že pripojený DHT12 modul má adresu 5c, tento údaj som použil v testovacom skripte a pokúsil sa prečítať aktuálnu hodnotu vlhkosti a teploty vzduchu.

V tomto okamihu nastal nečakaný, zdanlivo nepodstatný problém. Zariadenie síce komunikovalo, ale nie správne. Namiesto zmysluplných hodnôt som neustále dostával hodnoty 255 (0xFF). Keďže toto zariadenie malo posielať 4 bajty údajov a piaty bajt bol kontrolný súčet, bolo mi jasné, že niekde nastala chyba. Skúšal som nájsť pomocou internetu riešenie, ale bohužiaľ neúspešne. Všetky rady a nájdené príspevky mi nepomohli úspešne naviazať komunikáciu s týmto modulom. Preto som sa rozhodol, že na ďalší deň zájdem za vedúcim práce a poprípade sa pokúsim modul pripojiť k osciloskopu.

Keďže mi tento problém neustále vŕtal hlavou, napadlo ma, že by samotný modul mohol byť taktiež pokazený. Túto možnosť som ale vylúčil otestovaním modulu

pomocou arduina, ktoré som mal k dispozícii. Po pripojení som na prvý krát dostal zmysluplné hodnoty vlhkosti a teploty vzduchu.

Ešte v ten deň sa mi podarilo zapožičať si od jedného študenta USB signálny analyzátor. Pomocou neho som bol schopný jednoznačne identifikovať problém pri komunikácii. Po analýze priebehov získaných pri komunikácii modulu s arduinom a Raspberry som došiel k nasledovnému záveru. Môj modul nepodporoval príkaz s opakujúcim sa štartom (repeated start). Keďže som teraz vedel, kde konkrétne je problém, podarilo sa mi nájsť príspevok na stránke github [50], kde bol tento istý problém vyriešený. Podľa odporučenia som si stiahol staršiu verziu problematického overlay (prekryvný modul) súboru. Pomocou programu WinSCP, ktorý je spomenutý v kapitole použitých programov, som nakopíroval súbor do domovského adresáru, odkiaľ ho prekopírujem na požadované miesto použitím príkazu:

- sudo cp i2c1-bcm2708.dtbo /boot/overlays

Následne je potrebné ešte upraviť ďalší konfiguračný súbor príkazom:

- sudo nano /boot/config.txt

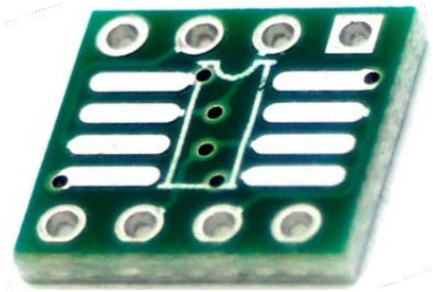
a pridať potrebný riadok, ktorý zaistí načítanie starej verzie driveru

- dtoverlay=i2c1-bcm2708

Po takmer celom dni neúspešných pokusov a dokončení hore uvedeného postupu začal modul posielať zmysluplné hodnoty.

3.1.5 Konfigurácia SPI modulu

Počas realizácie zavlažovacieho systému som sa rozhodol využiť taktiež jedno zariadene pripojené pomocou SPI rozhrania. Konkrétne A/D prevodník typu MCP3202 s puzdrom SO8. Aby bolo možné tento mikročip pripojiť k Raspberry, zakúpil som si pomocnú dosku, ktorú je možné vložiť do nepájivého kontaktného poľa.



Obr. 24 Adaptér z puzdra SO8 na DIP8 [51]

Konfigurácia a otestovanie prevodníku prebiehalo bezproblémovo. Postupoval som podľa tohto webového príspevku [52].

Pred samotným testovaním SPI prevodníku je nutné toto rozhranie v Raspberry povoliť. Pokiaľ potrebujeme zariadenia len otestovať, je to možné urobiť jednoducho pomocou konfiguračného menu, ktoré som taktiež už použil:

- sudo raspi-config

V prípade, že chceme, aby bolo SPI rozhranie povolené ihneď po štarte zariadenia je potrebné upraviť ďalší konfiguračný súbor. Pomocou príkazu:

- sudo nano /boot/config.txt

otvoríme konfiguračný súbor a na koniec súboru pridáme nasledovný riadok:

- dtparam=spi=on

Po povolení SPI rozhrania nasleduje reštart a overenie funkčnosti pomocou príkazu:

- lsmod | grep spi_

Ak sa nám následne zobrazí text "spi_bcm2708" alebo "spi_bcm2835" rozhranie je povolené a pripravené na komunikáciu.

Následne bolo potrebné stiahnuť pomocou príkazu potrebnú python knižnicu s názvom spidev:

- sudo apt-get install python-spidev

a taktiež stiahnuť a preložiť ďalšiu knižnicu pomocou príkazov:

- wget https://github.com/Gadgetoid/py-spidev/archive/master.zip
- unzip master.zip
- rm master.zip
- cd py-spidev-master
- sudo python setup.py install
- cd ..

Nakoniec nainštalujeme posledný potrebný balíček s menom GPIO Zero:

- sudo apt install python-gpiozero

Konečne sme pripravený SPI prevodník pripojiť a otestovať pomocou jednoduchého python skriptu vytvoreného na základe dokumentácie ku knižnici GPIO Zero [53]:

- from gpiozero import MCP3202
- adc1 = MCP3202(0)
- print adc1.value

V prípade že skript vypíše hodnotu medzi 0 a 1, funguje všetko v poriadku.

3.1.6 Nastavenie statickej IP adresy

Aby som nemusel po každom reštartovaní Raspberry hľadať jeho IP adresu pridelenú DHCP serverom, rozhodol som sa nastaviť mu statickú IP adresu. Úpravou konfiguračného súboru:

- sudo nano /etc/dhcpcd.conf

interface eth0
static ip_address=192.168.1.250/24
static routers=192.168.1.250
static domain_name_servers=192.168.1.250 8.8.8.8 fd51:42f8:caae:d92e::1

3.1.7 Konfigurácia vývojového prostredia

Pre vývoj aplikácie som sa rozhodol použiť vývojové prostredie Pycharm, ktoré umožňuje vývoj aplikácií v pythone a taktiež aj webových stránok prepojených s java skriptom. Základné vlastnosti prostredia sú popísané v kapitole *2.6.5*.

V programe Pycharm som nastavil parametre SSH komunikácie a povolil automatickú synchronizáciu, pomocou ktorej sa automaticky načítajú knižnice nainštalované v Raspberry a taktiež sa synchronizujú súbory projektu.

3.1.8 Automatické spustenie programu

Pre správne fungovanie neskôr vytvoreného zavlažovacieho systému je potrebné, aby sa riadiaci program spustil automaticky po zapnutí alebo reštarte systému. Postupoval som podľa dokumentácie na oficiálnej webovej stránke Raspberry Pi [54].

Pomocou príkazu:

- sudo nano /etc/rc.local

pridáme do konfiguračného súboru nasledovný riadok:

- su pi -c 'python home/pi/zavlaha/main.py >home/pi/zavlaha/zavlaha.log 2>&1' &

Pretože chcem skript spúšťať ako používateľ "pi", príkaz je potrebné vykonať ako správca. Nasleduje cesta a názov k spúšťanému skriptu a presmerovanie výpisu konzole a chybových hlásení do rovnakého súboru so zadanou cestou a názvom.

3.1.9 Záloha

Pomocou spomínanej aplikácie Win32DiskImager som na začiatku nahral operačný systém na micro SD kartu. Po všetkých vyššie spomenutých úkonoch je praktické si aktuálny obraz karty zálohovať. Po vložení karty do čítačky spustíme program, zadáme cestu a názov súboru pre uloženie obrazu systému uloženého na karte, vyberieme správne písmeno jednotky a klikneme na tlačidlo Read. Pomocou takto vytvorenej kópii systému sa v prípade potreby vieme vrátiť k už nakonfigurovanému systému a nemusím všetky počiatočné nastavenia absolvovať znovu.

3.2 Riadiaci program

Na úplnom začiatku som sa musel rozhodnúť, aký programovací jazyk využijem pre tvorbu riadiaceho programu. Ponúkali sa dve najrozšírenejšie možnosti a to C/C++ alebo python. Keďže som musel komunikovať s webovou stránkou a chcel som sa naučiť nový programovací jazyk rozhodol som sa programovať v pythone.

Aktuálne sú dostupné dve verzie python 2 a 3. Staršia verzia je masívne rozšírená a väčšina návodov a príspevkov na fórach sú písané práve v tejto verzii. Novšia verzia je síce plne stabilná a taktiež hojne používaná ale je potrebné si ju dodatočne stiahnuť, preto som sa rozhodol programovať vo verzii 2.7.14.

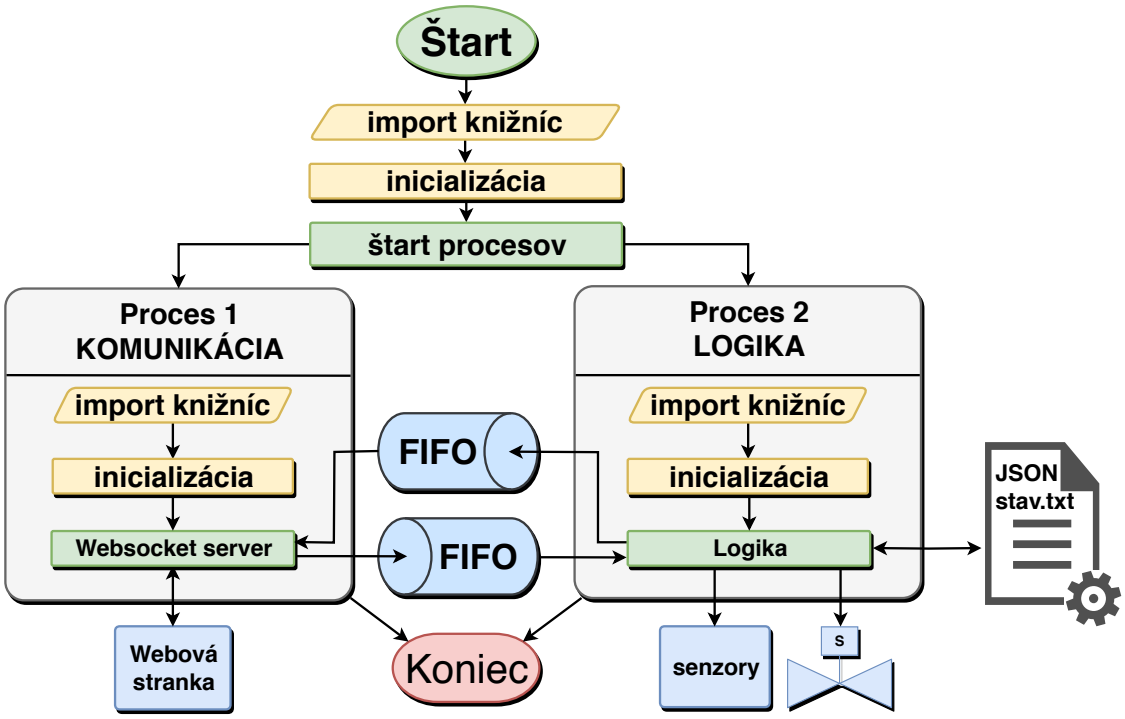
3.2.1 Popis riadiaceho programu

Riadiaci program je rozdelený na hlavný proces a dva nezávislé pod procesy. Prvý pod proces zabezpečuje komunikáciu s webovou stránkou a druhý má za úlohu riadenie zavlažovacieho systému.

Kód riadiaceho programu je vcelku rozsiahli (cca 500 riadkov), preto ho nebudem opisovať príliš detailne a pokúsim sa objasniť najdôležitejšie logické časti.

3.2.1.1 Náčrt riadiaceho programu

Pre rýchle a jednoduché pochopenie princípu fungovania riadiaceho programu som vytvoril a opísal náčrt, ktorý sa nachádza na nasledujúcom obrázku.



Obr. 25 Náčrt riadiaceho programu

Po spustení programu sa importujú potrebné knižnice, deklarujú sa pomocné premenné a definujú sa dva pod procesy - komunikačný a riadiaci. Následne sa inicializujú dva multiprocesné FIFO zásobníky, spustia sa oba pod procesy a do konzole sa vypíše meno programu a aktuálny dátum s časom.

Pod procesy síce bežia nezávisle, ale v prípade že sa v jednom z nich vyskytne neošetrená výnimka, hlavný proces ukončí aj druhý pod proces a následne ukončí celý program. Výpis z konzole a chybové hlásenie sa uloží do logu.

Pod procesy komunikujú medzi sebou pomocou dvoch FIFO zásobníkov a využívajú na to slovníky, v ktorých sú pod kľúčmi uložené príkazy a dáta.

3.2.1.2 Proces komunikácie

Komunikačný proces taktiež najprv načíta potrebné knižnice, deklaruje pomocné premenné, nachystá si prázdne pole pre neskoršie ukladanie aktuálne pripojených klientov a definuje sa trieda "WSHandler" obsluhujúca WebSocket server slúžiaci na komunikáciu s vizualizáciou.

Trieda "WSHandler" obsahuje nasledovné metódy:

- open;
- on_message;
- on_close;
- check_origin.

Metóda "open" pridá novo pripojeného klienta do zoznamu klientov a priradí mu unikátne ID, ktoré sa neskôr použije pre odoslanie inicializačného príkazu.

Druhá metóda s názvom "on_message" sa pokúsi dekódovať z prijatého JSON reťazca slovník s dátami. V prípade, že pri dekódovaní nastane chyba, prijatý reťazec sa za účelom ladenia odošle späť klientovi, aby bolo možné problematický reťazec identifikovať. Ak sa správu podarí dekódovať, zisťuje sa či existuje kľúč s príkazom. V prípade, že neexistuje, prijatý reťazec sa zasa odošle klientovi. Ak kľúč s príkazom existuje a jedná sa o inicializačný príkaz, do slovníku sa pridá kľúč a hodnota klientovho ID. V opačnom prípade sa ID klienta nemusí pridávať, pretože požiadavka je určená pre všetkých klientov. Nakoniec sa slovník s dekódovaným JSON reťazcom, v ktorom môže ale nemusí byť pridané ID vloží do FIFO zásobníku, odkiaľ ho neskôr spracuje riadiaci program.

Metóda "on_close" odstráni daného klienta po jeho odpojení zo zoznamu klientov.

Posledná metóda "check_origin" slúži ako bezpečnostná ochrana proti skriptovacím útokom medzi stránkami prehliadačov. Keďže túto metódu nepoužívam, ako radí dokumentácia [55], stačí ju prepísať aby vždy vracala hodnotu "True". Celá aplikácia funguje na domácej lokálnej sieti, preto nepovažujem za nutnosť využiť túto bezpečnostnú ochranu. V prípade ovládania zavlažovacieho systému pomocou internetu by bolo vhodné túto metódu implementovať podľa spomínanej dokumentácie.

Následne sa vytvorí pomocou definovanej triedy "WSHandler" inštancia webovej aplikácie.

Funkcia "posli_prikaz()" kontroluje obsah FIFO zásobníku, v ktorom sa nachádzajú príkazy od riadiaceho pod procesu. V prípade, že sa jedná o inicializačný príkaz, pošle ho len klientovi s príslušným ID a naopak u ostatných príkazoch pošle správu všetkým pripojeným klientom. Funkcia "posli_prikaz()" sa vykonáva cyklicky každých 100ms. Je volaná ako vedľajšia rutina, tak aby neblokovala činnosť hlavnej slučky, ktorá obsluhuje komunikáciu s webovou stránkou.

Predposlednou časťou kódu procesu komunikácie je funkcia main, v ktorej sa nachádza definícia WebSocket serveru s potrebnými parametrami ako je napríklad adresa, port, pred pripravená inštancia webovej aplikácie a podobne. Nasleduje spustenie hlavnej nekonečnej slučky a taktiež vedľajšej synchrónnej rutiny s volaním funkcie "posli_prikaz()".

Poslednou časťou tohto procesu je korektné zastavenie komunikačného serveru, v prípade že by nastala neošetrená výnimka

3.2.1.3 Proces logiky

Proces logiky obdobne najprv načíta potrebné knižnice, vytvorí pomocné premenné a nastaví režim použitých GPIO pinov na vstupný alebo výstupný. Vstupným pinom sa programovo priradí pulldown odpor, aby sa v prípade nepripojených periférií nenačítavali náhodné hodnoty.

V main funkcii sa najprv program pokúsi načítať svoj predošlý stav z textového súbor "stav.txt", v ktorom je zakódovaný pomocou JSON formátu. Tento formát je dobre čitateľný a v prípade potreby je možné jednotlivé kľúče a ich hodnoty meniť ľubovoľným textovým editorom. Ak sa nepodarí stav načítať, program sa pokúsi načítať stav z inicializačného súboru "init.txt" a ak sa mu to nepodarí, program sa ukončí.

Nasleduje nekonečná slučka, v ktorej sa cyklicky opakujú tieto logické celky:

- aktualizácia vstupov;
- vyhodnotenie príkazov;
- riadiaca logika;
- zápis stavu do súboru.

Aktualizácia analógových vstupov a následné predanie nových hodnôt grafickému rozhraniu prebieha v dvoch intervaloch. Raz za 60 sekúnd sa zapne napájanie zrážkomeru a vlhkomeru pôdy, načítajú sa aktuálne hodnoty a napájanie sa zasa vypne. Minútový interval je zvolený zámerne, aby nedochádzalo k rýchlemu opotrebovaniu sond. Hodnoty sa načítavajú pomocou knižnice gpiozero a triedy MCP3202. Dvojkanálový ADC prevodníku MCP3202 je pripojený pomocou SPI zbernice. Každých 10 sekúnd sa pomocou knižnice smbus načítajú hodnoty teploty a vlhkosti vzduchu pomocou DHT12 senzoru pripojeného I^2C zbernicou.

V sekcii rozpoznávania príkazov sa vyhodnocujú podporované príkazy od webovej stránky, ktoré sa načítajú z FIFO zásobníku. V prípade, že sa vyhodnotí inicializačný príkaz, pošle sa celý stav systému danému klientovi. V ostatných prípadoch sa spracujú jednoduché špecifické príkazy ako napríklad požiadavka na zmenu prepínača, hodnoty prahu alebo času. Po spracovaní požiadavky a zmene stavu sa informujú všetci aktuálne pripojený klienti. Štruktúra stavovej premennej sa nachádza v prílohe *C* a zoznam podporovaných príkazov sa nachádza v prílohe *D*.

Riadiaca logika je rozčlenená na dva logické celky. Automatické riadenie je popísané v kapitole *3.3.1.1 Popis automatického režimu zavlažovania* a manuálne riadenie jednotlivých okruhov je popísané v kapitole *3.3.1.2 Popis manuálneho režimu zavlažovania*.

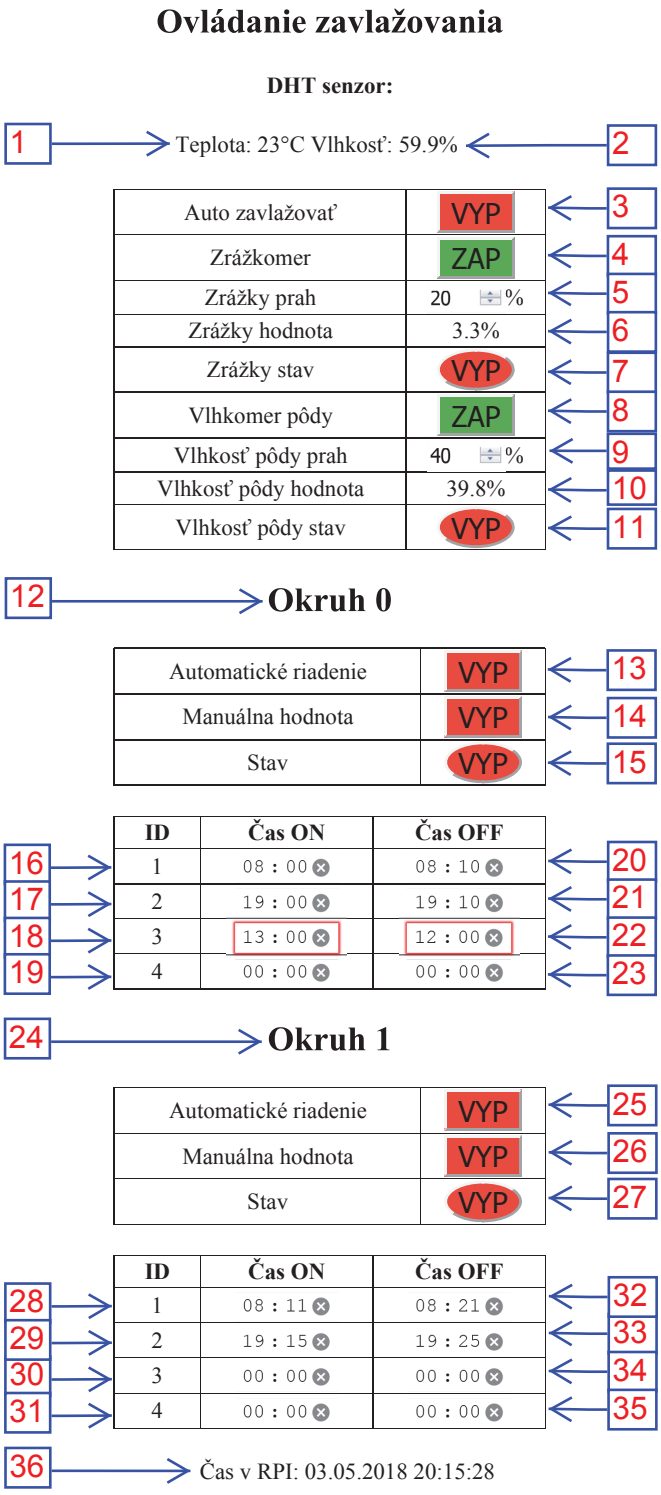
Poslednou časťou riadiaceho programu je zápis stavu systému v čitateľnom JSON formáte do textového súboru s názvom "stav.txt" a 100ms čakanie, aby sa zbytočne nepreťažoval procesor mikrokontroléru.

3.3 Vizualizácia

Vizualizáciu som sa rozhodol robiť pomocou webovej stránky. Tým som dosiahol maximálnej kompatibility s rôznymi zariadeniami ako sú počítače, notebooky, telefóny tablety a podobne. Stačí mať pripojenie do miestnej siete a podporovaný webový prehliadač. V opačnom prípade by som musel vyvíjať niekoľko rôznych aplikácii pre každý operačný systém zvlášť (Windows, MAC, Linux, Android, IOS...).

3.3.1 Popis ovládania vizualizácie

Na nasledujúcom obrázku sa nachádza konečná verzia vizualizácie zavlažovacieho programu. V tejto podkapitole popíšem význam jednotlivých indikačných a ovládacích prvkov.



Obr. 26 Webová stránka - vizualizácia

V prípade, že by webový server fungoval a riadiaci program zavlažovacieho systému by nebol spustený, zobrazí sa okno s upozornením "Hostiteľ nedostupný !".

Ak funguje všetko ako má, načíta sa nám vizualizácia, ktorú je možné vidieť na Obr. 26.

Očíslované ovládacie a indikačné prvky majú nasledovný význam:

1. indikátor teploty vzduchu zo senzoru DHT (0-100°C);
2. indikátor vlhkosti vzduchu zo senzoru DHT(0-100%);
3. prepínač automatického zavlažovacieho režimu (ZAP/VYP);
4. prepínač zrážkomeru (ZAP/VYP);
5. nastavenie prahu zrážkomera (0-100%);
6. indikátor aktuálneho množstva zrážok (0-100%);
7. indikátor prekročenia prahu zrážok (prší/neprší);
8. prepínač vlhkomeru (ZAP/VYP);
9. nastavenie prahu vlhkosti pôdy (0-100%);
10. indikátor aktuálnej hodnoty vlhkosti pôdy (0-100%);
11. indikátor prekročenia prahu vlhkosti pôdy (vlhko/sucho);
12. číslo okruhu;
13. prepínač automatického riadenia okruhu číslo 0 (aut/man);
14. prepínač manuálnej hodnoty okruhu číslo 0 (ZAP/VYP);
15. stav zavlažovania okruhu číslo 0 (ZAP/VYP);
16. - 19. nastavenie časov pre automatické zapnutie zalievania okruhu 0;
20. - 23. nastavenie časov pre automatické vypnutie zalievania okruhu 0;
24. číslo okruhu;
25. prepínač automatického riadenia okruhu číslo 1 (automaticky/manuálne);
26. prepínač manuálnej hodnoty okruhu číslo 1 (ZAP/VYP);
27. stav zavlažovania okruhu číslo 1 (ZAP/VYP);
28. - 31. nastavenie časov pre automatické zapnutie zalievania okruhu 0;
32. - 35. nastavenie časov pre automatické vypnutie zalievania okruhu 0;
36. aktuálny dátum a čas v Raspberry.

3.3.1.1 Popis automatického režimu zavlažovania

V prípade, že chceme niektorý z okruhov zavlažovať pomocou automatického režimu musí byť prepínač 3 v polohe "ZAP" (zelene podfarbený). Ak chceme počas zavlažovania detegovať zrážky alebo vlhkosť pôdy, je potrebné mať prepínače 4 a 8 taktiež v polohe "ZAP" a pomocou ovládacích prvkov 5 a 9 nastavenú hodnotu príslušného prahu. Nasledovne je nutné mať prepínače automatického riadenia

príslušného okruhu 13 a 25 v polohe "ZAP". Posledným krokom je nestavenie času zalievania. Príslušnou dvojicou ovládacích prvkov (16 + 20 apod.) sa nastaví začiatok a koniec zavlažovacieho cyklu. V prípade že sú nastavené nezmyselné hodnoty, prehliadač Mozzila na to upozorní červeným zarámovaním nesprávnych hodnôt (viď. ovládacie prvky 18 a 22). V tomto prípade je možné ponechať chybne zadané hodnoty ale zavlažovací systém sa samozrejme automaticky nespustí.

3.3.1.2 Popis manuálneho režimu zavlažovania

Ak potrebujeme jeden z okruhov polievať manuálne a druhý automaticky, stačí príslušný prepínač režimu automatického riadenia daného okruhu 13 alebo 25 nastaviť do polohy "VYP" (červeno podfarbený) a následne je možné zvolený okruhu pomocou prepínaču manuálnej hodnoty 14 alebo 26 zapnúť a vypnúť, pričom druhý okruh bude naďalej ovládaný automatickým režimom zavlažovacieho systému.

3.3.1.3 Upozornenia

Vizualizácia má naprogramovaných niekoľko upozornení, ktoré sa zobrazujú pomocou vyskakovacieho okna.

Prvé upozornenie s textom "Viacero okruhov je spustených naraz" sa zobrazí v prípade, že sú súčasne zavlažované oba okruhy. V takomto prípade bude tlak vody v oboch okruhoch nízky a pravdepodobne sa dostatočne nezavlaží ani jedna časť trávniku.

Druhé upozornenie s textom "Prší, nebude sa zalievať" sa zobrazí ak platia všetky nasledovné podmienky súčastne:

- je zapnutý automatický zavlažovací režim (prepínač 3 v polohe ZAP);
- zrážkomer je zapnutý (prepínač 8 v polohe ZAP);
- prší (indikátor 10 - aktuálne množstvo zrážok > prah zrážkomera - 9;
- aspoň jeden z okruhov je v automatickom režime (prepínač 13/25 v polohe ZAP);
- jeden z okruhov sa má zavlažovať (čas zapnutia ≤ aktuálny čas ≤ čas vypnutia).

Posledné upozornenie s textom "Je vlhko, nebude sa zalievať" sa zobrazí ak platia všetky nasledovné podmienky súčastne:

- je zapnutý automatický zavlažovací režim (prepínač 3 v polohe ZAP);
- vlhkomer je zapnutý (prepínač 4 v polohe ZAP);
- je vlhká pôda (indikátor 6 - aktuálne množstvo zrážok > prah zrážkomera - 5);
- aspoň jeden z okruhov je v automatickom režime(prepínač 13/25 v polohe ZAP);
- jeden z okruhov sa má zavlažovať (čas zapnutia ≤ aktuálny čas ≤ čas vypnutia);

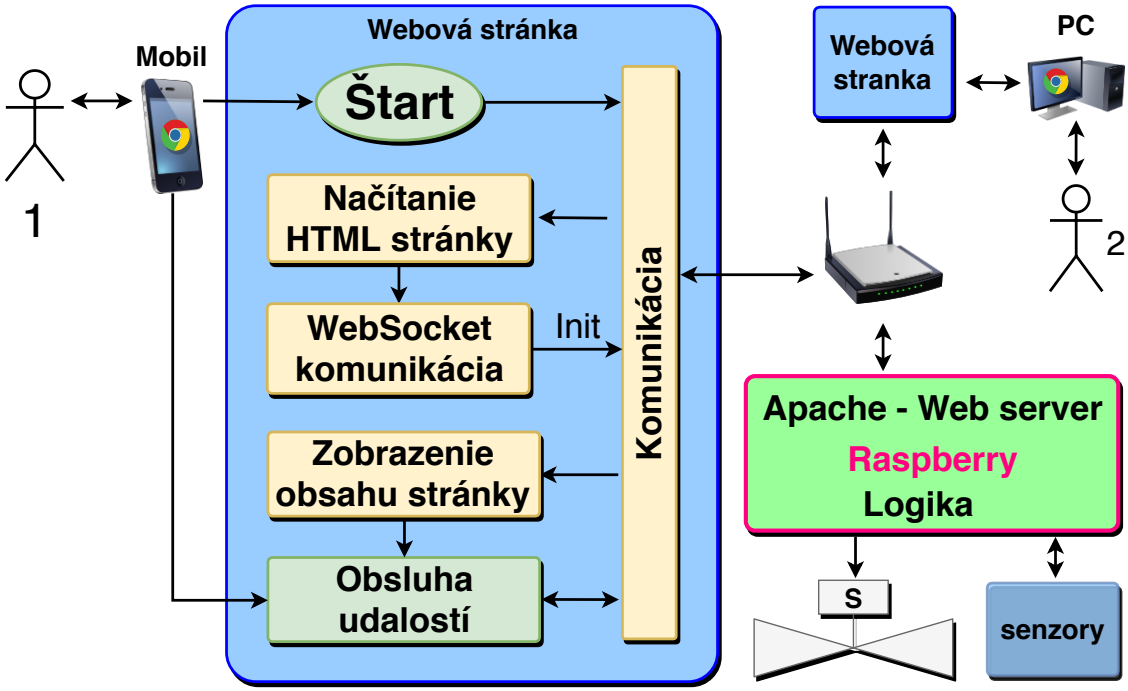
3.3.2 Popis programu vizualizácie

Základom vizualizácie je jednoduchá HTML stránka, ktorá má za úlohu popis a rozmiestnenie indikačných a ovládacích prvkov. O komunikáciu s riadiacim programom a funkcionalitu jednotlivých tlačidiel a indikátorov sa stará Javascript a Jquery.

Z jednoduchého "Hello world" návodu [46]som postupne vybudoval kompletné grafické rozhranie pre ovládanie zavlažovacieho systému. Pri tvorbe webovej stránky som čerpal zo stránky "w3schools", ktorá obsahuje dokumentáciu a nespočetné množstvo praktických ukážok využitia HTML [56] a Javascriptu [57].

3.3.2.1 Náčrt programu vizualizácie

Pre rýchle a jednoduché pochopenie princípu fungovania a komunikovania webovej stránky s riadiacim programom som vytvoril a opísal náčrt, ktorý sa nachádza na nasledujúcom obrázku.



Obr. 27 Náčrt programu vizualizácie

Náčrt som rozdelil na štyri logické celky:

- načítanie HTML stránky;
- nadviazanie komunikácie;
- inicializácia obsahu;
- detekcia zmien.

Po zadaní statickej lokálnej adresy Raspberry Pi, na ktorej je spustený webový server sa prehliadač pokúsi načítať webovú stránku s vizualizáciou riadiaceho programu závlahy.

Po úplnom načítaní HTML stránky sa pomocou skriptu nadviaže komunikácia s WebSocket serverom a vyšle sa požiadavka na inicializáciu. Po prijatí sa dáta v JSON formáte dekódujú a všetkým indikačným a ovládacím prvkom sa nastavia aktuálne hodnoty stavu riadiaceho programu. Následne sa takto prichystaná stránka zobrazí užívateľovi a je pripravená na detekciu zmien.

Vyvolať zmenu stavu vizualizácie môžu tieto tri udalosti:

- aktuálny užívateľ webovej stránky;
- iný užívateľ webovej stránky;
- riadiaci program.

V prípade že si aktuálny užívateľ načíta webovú stránku, môže pomocou nej posielať požiadavky na zmenu prepínačov automatického a manuálneho ovládania závlahy, okruhov a podobne, prahových hodnôt snímaču zrážok a vlhkosti pôdy alebo času zalievania.

Uvedené požiadavky môže zároveň posielať aj iný užívateľ, ktorý je v tom istom čase pripojený. Po spracovaní každého príkazu sa zmeny premietnu všetkých aktuálne pripojeným klientom.

Posledným a najčastejším zdrojom udalostí meniacich vizualizáciu je samotný riadiaci program zavlažovacieho systému, ktorý v pravidelných intervaloch meria teplotu a vlhkosť vzduchu, aktuálne množstvo zrážok a vlhkosť pôdy a taktiež kontroluje časy zavlažovania. Pri zmene ktoréhokoľvek z týchto parametrov, pošle správu s novými informáciami všetkým pripojeným klientom.

3.3.2.2 HTML

HTML časť stránky má za úlohu správne rozmiestniť indikačné a ovládacie prvky. Kód sa delí na dve hlavné časti - hlavičku a telo.

V hlavičke stránky sa nastaví titulok na "Závlaha", typ dekódovania špeciálnych znakov sa špecifikuje na "UTF-8", nasleduje nastavenie centrovania textu a definícia štýlu zobrazovania tabuliek. Ku koncu sa načíta lokálna kópia súboru "jquery.js", aby bolo možné ovládať zavlažovací systém aj bez nutnosti pripojenia k internetu a ako posledné sa v hlavičke vykoná rozsiahli skript, ktorý bude popísaný v nasledujúcej kapitole.

V tele dokumentu je definovaných päť oddielov:

1. komunikácia;
2. stav;
3. správa;
4. terminál;
5. refresh.

V komunikačnej časti sa nastavujú statické parametre Websocket serveru ako je IP adresa serveru, port a taktiež uri. Táto sekcia je pre užívateľa skrytá.

Oddiel s pomenovaním "stav" je najrozsiahlejší a obsahuje definície typov a vlastností všetkých viditeľných nápisov, tlačidiel a indikátorov. Prepínače a textové indikátory hodnôt majú svoje unikátne ID, pomocou ktorého sa dajú identifikovať, vďaka čomu je možné v skripte načítavať a meniť stav týchto prvkov. Na Obr. 26 sa nachádza prehľad a stručný popis týchto prvkov.

V sekcii "správa" sa nachádza pozostatok z pôvodného "Hello world" návodu. Tento oddiel je taktiež pre užívateľa skrytý. Pomocou textového vstupu a tlačidla pre odoslanie som bol schopný počas vývoja stránky a riadiaceho programu testovať funkčnosť novo implementovaných komunikačných príkazov.

Terminál slúži ako takzvaný "log", zaznamenávajú sa doň ho všetky prijaté a odoslané príkazy, pomocné informácie a hlásenia. Počas vývoja stránky som ho využíval pre ladiace účely. Vo finálnej verzii je skrytý.

Poslednou časťou tela webovej stránky je refresh tlačidlo s nápisom "Obnoviť stránku", ktoré sa zobrazí v prípade, že hostiteľ nie je aktuálne dostupný.

3.3.2.3 Skript

Na začiatku skriptu je definovaná logovacia funkcia "log = function(data)", ktorá slúžila na ladenie programu. Akýkoľvek text sa pridal na začiatok tohto textového indikátoru.

Nasleduje funkcia "$(document).ready(function () {...});", v ktorej sa nachádza podstatná časť kódu. Ako vyplýva z definície, vizualizácia čaká na úplné načítanie webovej stránky a následne začne pracovať.

Obsah skriptu som rozčlenil do nasledovných logických celkov:

1. skrytie obsahu stránky;
2. detekcia prehliadača;
3. identifikácia HTML objektov;
4. WebSocket komunikácia;
5. onclick a onchange metóda.

3.3.2.3.1 Skrytie obsahu stránky

Po načítaní celého obsahu stránky sa jej obsah skryje. V prípade, že sa nepodarí pripojiť k WebSocket serveru a následne inicializovať obsah stránky, zobrazí sa varovné hlásenie "Hostiteľ nedostupný!" a odkryje sa iba tlačidlo na obnovenie stránky.

3.3.2.3.2 Detekcia prehliadača

Každý programátor webových stránok vie, že kompatibilita s Internet Explorerom je takmer vždy problém. Niektoré veci tam fungujú podobne, ďalšie úplne inak a niektoré vôbec. Preto som sa rozhodol, že prehliadač Edge a Internet Explorer nebude vizualizácia zavlažovacieho systému podporovať.

Pre detekciu a následné varovanie som sa inšpiroval príspevkom na stránke stackoverflow [58]. Využil som metódu zvanú "duck typing", ktorá je spoľahlivejšia ako detekcia pomocou takzvaného "User agent" reťazca, ktorý obsahuje informácie o aktuálnom webovom prehliadači. "Duck typing" metóda je presnejšia, pretože je založená na detekcii vlastností špeciálnych objektov.

3.3.2.3.3 Identifikácia HTML objektov

V skripte nasleduje definícia niekoľkých pomocných premenných a priradenie príslušných HTML objektov k rozsiahlej stavovej premennej "stav", ktorá má rovnakú štruktúru ako premenná riadiaceho programu v python kóde. Na identifikáciu HTML objektov som použil metódu document.getElementById("ID")

3.3.2.3.4 WebSocket komunikácia

Ako prvé, sa vytvorí nová inštancia Websocket komunikácie, pre ktorú je potrebné zadať IP adresu serveru, port a uri. Z týchto dát sa poskladá výsledná adresa serveru "ws://192.168.1.250:8888/ws".

Následne sa definujú tri asynchrónne metódy (udalosti):

- ws.onmessage=function(evt);
- ws.onclose=function(evt);
- ws.onopen=function (evt).

Onmessage metóda má za úlohu z prijatých dát dekódovať správu z JSON formátu a v prípade chyby, prijaté dáta vypíše do logu. Ak sa podarí dáta bezchybne dekódovať, začne sa správa lúštiť. Štruktúra premennej stav sa nachádza v prílohe C a zoznam všetkých príkazov sa nachádza v prílohe D.

Prvým dôležitým parametrom správy je príkaz s pomenovaním "cmd". V prípade že to je príkaz "init", načítajú sa hodnoty všetkých zobrazovacích a ovládacích prvkov webovej stránky, nastavia sa príslušné farby týchto prvkov a v prípade že nenastane žiadna chyba zobrazia sa tieto hodnoty užívateľovi. Ak príde iný podporovaný príkaz, identifikuje sa pomocou unikátneho mena a vykoná sa príslušná časť kódu, ktorá je zvyčajne tvorená zmenou hodnoty (metóda .value=hodnota) a farby daného objektu (metóda .style.background=farba). Po rozpoznaní a vykonaní známeho

príkazu sa zapíše do logu pomocný informačný text. V prípade, že sa prijme nepodporovaný príkaz, obsah celej správy sa zapíše do logu.

Onclose metóda sa volá v prípade, že sa nepodarí nadviazať alebo ak skončí komunikácia s Websocket serverom. V tomto prípade sa užívateľovi zobrazí varovné hlásenie "Hostiteľ nedostupný !!!" a schovajú sa všetky oddiely webovej stránky až na obnovovacie tlačidlo.

Onopen metóda je veľmi jednoduchá. Po nadviazaní komunikácie s Websocket serverom sa pošle požiadavka na odoslanie aktuálneho stavu riadiaceho programu zavlažovacieho systému - príkaz inicializácie "init". Po spracovaní požiadavku sa prijaté dáta automaticky spracujú už popísanou metódou onmessage.

3.3.2.3.5 Onclick a onchange metóda

Koniec skriptu tvorí niekoľko funkcií, ktoré využívajú taktiež asynchrónne onclick a onchange metódy (udalosti).

Po kliknutí na niektorý z prepínačov sa pošle správa s požiadavkou na zmenu hodnoty tohto prvku, ktorá je následne spracovaná riadiacim systémom závlahy. Po spracovaní príkazu sa pošle správa s novou hodnotou prepínača a pomocou už popísanej metódy onmessage sa zmení text a farba daného prepínača.

V prípade, že sa zadá nová hodnota prahu zrážkomera, vlhkomera alebo času zalievania pošle sa obdobne požiadavka riadiacemu systému a následne sa spracuje.

4 REALIZÁCIA A OVERENIE FUNKČNOSTI

V úvode tejto kapitoly sa zameriam na popis postupu návrhu dosky plošných spojov, popíšem použité typy modulov, senzorov a súčiastok. Následne podrobnejšie rozoberiem navrhnutú schému zapojenia a výsledné rozmiestnenie súčiastok na doske. V poslednej časti sa budem venovať overeniu funkčnosti navrhnutého systému.

4.1 Doska plošných spojov

Na pripojenie vstupno-výstupných zariadení bolo potrebné navrhnúť, vyrobiť, oživiť a otestovať dosku plošných spojov. Pretože mám s návrhom DPS veľmi málo skúseností, bola to pre mňa jedna z ťažších častí tejto práce. Potrebnú dosku plošných spojov som sa rozhodol navrhnúť v programe Eagle z niekoľkých dôvodov. Spomínaný program je vcelku jednoduchý, často používaný a dostupný v neplatenej verzii. Neplatená verzia má obmedzený počet schematických listov (2), počet vrstiev DPS (2) a plochu navrhovanej DPS na 80 cm^2. Napriek týmto obmedzeniam je to pre návrh dosky aj tak dostatočné. [59]

4.1.1 Postup návrhu DPS

Prvý krokom pri návrhu dosky bol obvod pre ovládania dvojice výstupných relé, pomocou ktorých sa budú spínať solenoidy zavlažovacích elektro ventilov. Ďalším krokom bolo pripojenie periférií pomocou dostupných zberníc. Pre správne fungovanie zavlažovacieho systému je potrebné merať hodnotu vlhkosti pôdy, množstvo aktuálnych zrážok a samozrejme mať presný čas. Ako doplnkový senzor som sa rozhodol použiť senzor teploty a vlhkosti vzduchu.

4.1.1.1 Ovládanie relé

Pri návrhu obvodu spínaného relé som sa inšpiroval elektronickým zapojením uvedeným na webovej stránke [60]. Obvod je veľmi jednoduchý, obsahuje bežne dostupný NPN tranzistor s označením BC337. Jeho báza je pripojená cez 1k ohmový odpor, ktorým sa nastaví vhodný pracovný bod tranzistora, na príslušný výstupný GPIO pin z Raspberry Pi. Napájacie napätie +5V je cez cievku relé pripojené na kolektor tranzistoru. Po zopnutí príslušného GPIO výstupu sa tranzistor otvorí a uzemní tak cievku relé, ktorou začne tiecť prúd. Pretekajúci prúd prepne výstup relé, ktorý pripojí solenoid zavlažovacieho elektromagnetického ventilu k napájaciemu napätiu a ventil sa otvorí. Keďže pomocou tranzistoru spíname indukčnú záťaž, pri následnom vypnutí príslušného GPIO výstupu sa budú tvoriť napäťové špičky, ktoré potlačíme pridaním ochrannej diódy.

Obvod som rozšíril o vetvu pripojenú paralelne k cievke relé, v ktorej sa nachádza ochranný odpor a LED dióda signalizujúca zopnutie príslušného relé. Pomocou ohmového zákonu som vypočítal potrebnú veľkosť ochranného odporu, tak aby diódou netiekol príliš veľký prúd. Napájacie napätie je 5V a podľa katalógového

listu uvedeného na stránkach predajcu [61] pri nominálnom prúde tečúcom diódou 20mA vznikne na dióde napäťový úbytok 2,2V. Z týchto údajov sme schopný vypočítať hodnotu ochranného odporu:

$$R = \frac{U_N - U_D}{I} = \frac{5-2,2}{0,02} = 140\ \Omega \qquad \textbf{(1)}$$

Kde R je hodnotu ochranného odporu, U_N je napájacie napätie, U_D je napäťový úbytok na dióde a I je nominálny prúd diódou. Keďže sa odpory vyrábajú len v určitých hodnotách, radách je potrebné hodnotu odporu zaokrúhliť k najbližšej vyššej hodnote, čo je R=150 Ω. Minimálny výkon odporu sa určí výpočtom:

$$P_{Zmin} = I^2.R = 0,02^2.150 = 0,06W \qquad \textbf{(2)}$$

Kde P_{Zmin} je minimálny výkon odporu, I je nominálny prúd diódou a R je hodnotu ochranného odporu. Kvôli jednoduchšiemu návrhu dosky som sa rozhodol pre SMD odpory a keďže budem mať na doske dostatok miesta, môžem si dovoliť použiť puzdro o veľkosti 1206. Podľa katalógového listu predajcu [62] má odpor s týmto puzdrom maximálny stratový výkon P_{Zmax}= 0,25W čo je asi štvornásobne viac ako potrebná minimálna hodnota stratového výkonu. Toleranciu odporu som zvolil 1%.

Keďže bude zdroj +5V napätia použitý z dosky Raspberry Pi, je potrebné overiť jeho celkovú maximálnu hodnotu. Pomocou jednoduchej simulácie v programe Multisim, ktorej schéma zapojenia je na *Obr. 47*, som určil maximálnu hodnotu odoberaného prúdu I_{MAX}= 56,72mA. Táto hodnota prúdu nie je natoľko veľká, aby negatívne ovplyvnila chod Raspberry Pi. Maximálny prúd kolektorom jedného tranzistora I_{CMAX}=28,36mA je taktiež rádovo menší ako maximálna hodnota prúdu udávaná v katalógovom liste výrobcu [63]. Maximálna hodnota prúdu odoberaná z GPIO výstupu udávaná výrobcom [64] je 16 mA na pin a 50mA celkovo zo všetkých GPIO výstupov. Táto hodnota nebola taktiež prekročená, pretože prúd bázou tranzistoru zistený simuláciou je len približne I_B= 2,5mA.

4.1.1.2 Meranie zrážok a vlhkosti pôdy

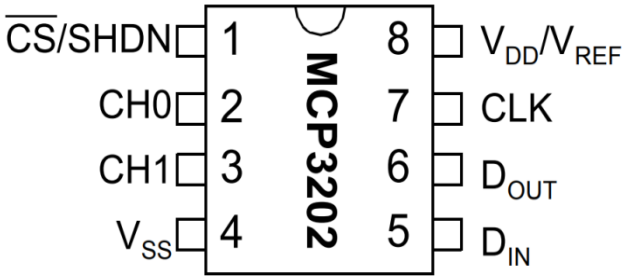
Ďalšou úlohou bolo získanie analógovej hodnoty z dvoch snímačov a to konkrétne snímaču vlhkosti pôdy a snímaču množstva zrážok. Keďže Raspberry Pi nemá hardvérové vybavenie na meranie analógových vstupných signálov bolo potrebné použiť pomocný obvod. Keďže použité moduly na meranie množstva zrážok a vlhkosti pôdy sú veľmi lacné a nepresné, rozlíšením a presnosťou sa moc nebudeme zaoberať. Dôležité je aby sme boli schopný merať dva vstupy minimálne raz za sekundu. Na výber bolo z niekoľko rôzne zložitých, veľkých a presných možností:

- RC článok (jednoduché pripojenie, malé rozmery, nepresná hodnota);
- modul s A/D prevodníkom (zložité pripojenie, veľké rozmery, presná hodnota);
- čip s A/D prevodníkom (jednoduché pripojenie, malé rozmery, presná hodnota);
- arduino (zložité pripojenie a komunikácia, veľké rozmery, presná hodnota).

Po konzultácii s vedúcim práce sme usúdili, že najlepšia možnosť bude použiť integrovaný čip. Bol mi doporučený mikročip s označením MCP3202, ktorý spĺňa všetky potrebné požiadavky.

Mikročip MCP3202 má nasledovné parametre [65]:

- 2 kanály
- 12-bit rozlíšenie
- SPI rozhranie
- 2.7V - 5.5V napájanie
- 50 000 vzorkou za sekundu pri VDD = 2.7V
- nízku spotrebu
- SOIC(SO8) puzdro

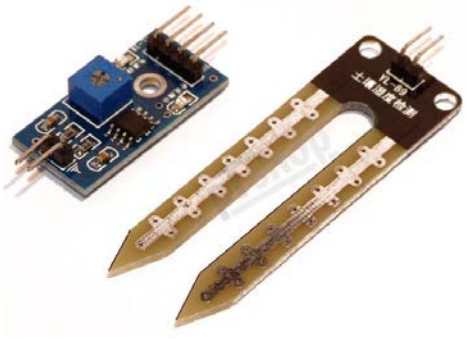


Obr. 28 Puzdro s označením vývodov MCP3202 [65]

Tabuľka 2 Popis vývodov MCP3202 [65]

VDD/VREF	Power, reference Input
CH0	Channel 0 Analog Input
CH1	Channel 1 Analog Input
CLK	Serial Clock
DIN	Serial Data In
DOUT	Serial Data Out
CS/SHDN	Chip Select/Shutdown Input

K spomínanému mikročipu s označením MCP3202 som následne mohol pripojiť senzor zrážok a vlhkosti pôdy. Oba moduly používajú ako základ komparátor s označením LM393 a majú analógový a taktiež digitálny výstup. U digitálneho výstupu je možné nastaviť prah. V práci som sa rozhodol použiť len analógový výstup, pretože v prípade potreby zmeny prahu by bolo nutné vždy prestaviť príslušný potenciometer.



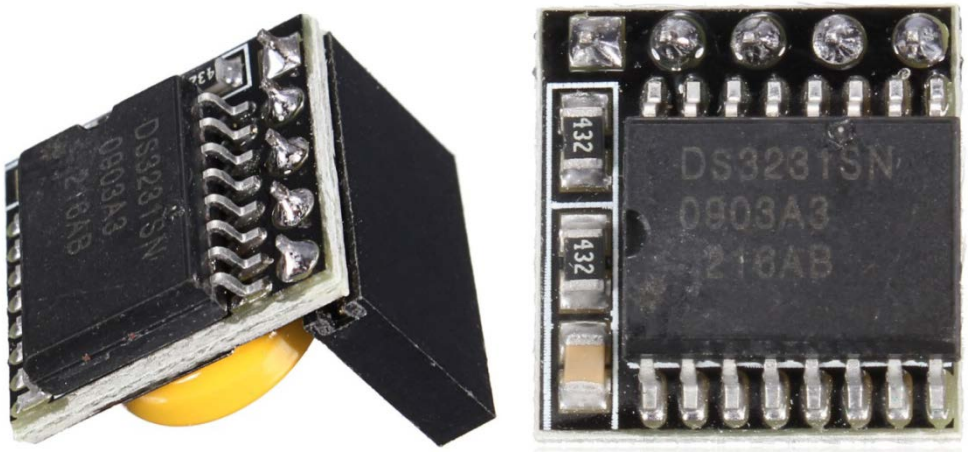
Obr. 29 Senzor vlhkosti pôdy [66]

Obr. 30 Senzor zrážok [67]

4.1.1.3 Modul reálneho času

Pri výbere RTC modulu bolo potrebné skontrolovať niekoľko parametrov. Moduly je možné pripojiť pomocou SPI / I^2C zbernice, majú rôzne napájacie napätia 3,3 - 12V, rôznu presnosť generovaného času, rozmery a počet pripojovacích pinov.

Z dostupných RTC modulov som sa rozhodol použiť modul s označením DS3231. Tento modul som zakúpil vo veľmi kompaktnom prevedení. Pripája sa pomocou I^2C zbernice, čo je výhodné vzhľadom na to, že k Raspberry Pi je možné pripojiť maximálne dve SPI slave zariadenia. Napájacie napätie je 3,3V a pripája sa pomocou 5 pinového konektoru. Presnosť RTC modulu je pre naše účely úplne dostatočná, pretože pri zavlažovacom systém nie je presnosť RTC modulu príliš podstatná a je nám jedno či sa zavlažovanie pustí o +- sekundu skôr alebo neskôr.



Obr. 31 Modul reálneho času DS3231 [68]

4.1.1.4 Meranie vlhkosti a teploty vzduchu

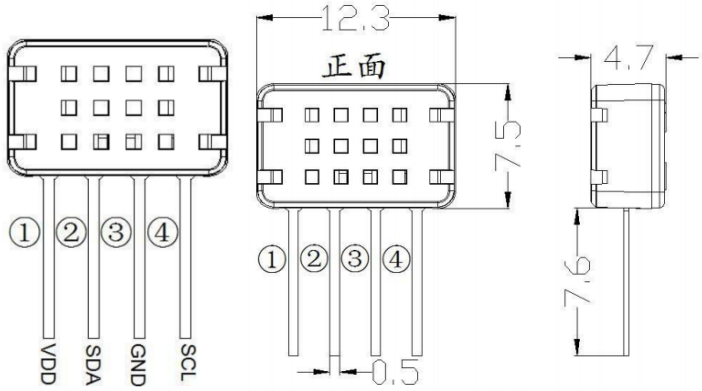
Posledný modul pripojený k zavlažovaciemu systému bude senzor slúžiaci na meranie vlhkosti a teploty vzduchu. Pre menej náročné aplikácie sa často používajú senzor z

rady označenej DHTxx. Na výber je z nasledujúcich možností : DHT11, DHT12, DHT 21, DHT 22, DHT33, DHT 44.

Kvôli jednoduchej komunikácii som sa rozhodol pre modul DHT12. Tento modul komunikuje pomocou I^2C rozhrania, má lepšiu presnosť ako modul DHT11 a zároveň je lacný. Napájacie napätie je 2.7V-5.5V. Ostatné parametre modulu DHT12:

Tabuľka 3 Parametre modulu DHT12 [69]

	Teplota	Vlhkosť
Rozsah	-20°C až 60°C	20 až 95 % RV
Presnosť	± 0,5 ℃	±5 % RV
Rozlíšenie	0.1 ℃	0.1 % RV
Opakovateľnosť	±0.2 ℃	±0.3 % RV



Obr. 32 Rozmery a popis vývodov modulu DHT12 [69]

4.1.2 Schéma zapojenia

Dominantnú časť schémy tvorí 40 pinový konektor, ktorý sa nachádza v strednej časti schémy a bude prepojený s GPIO konektorom na Raspberry Pi.

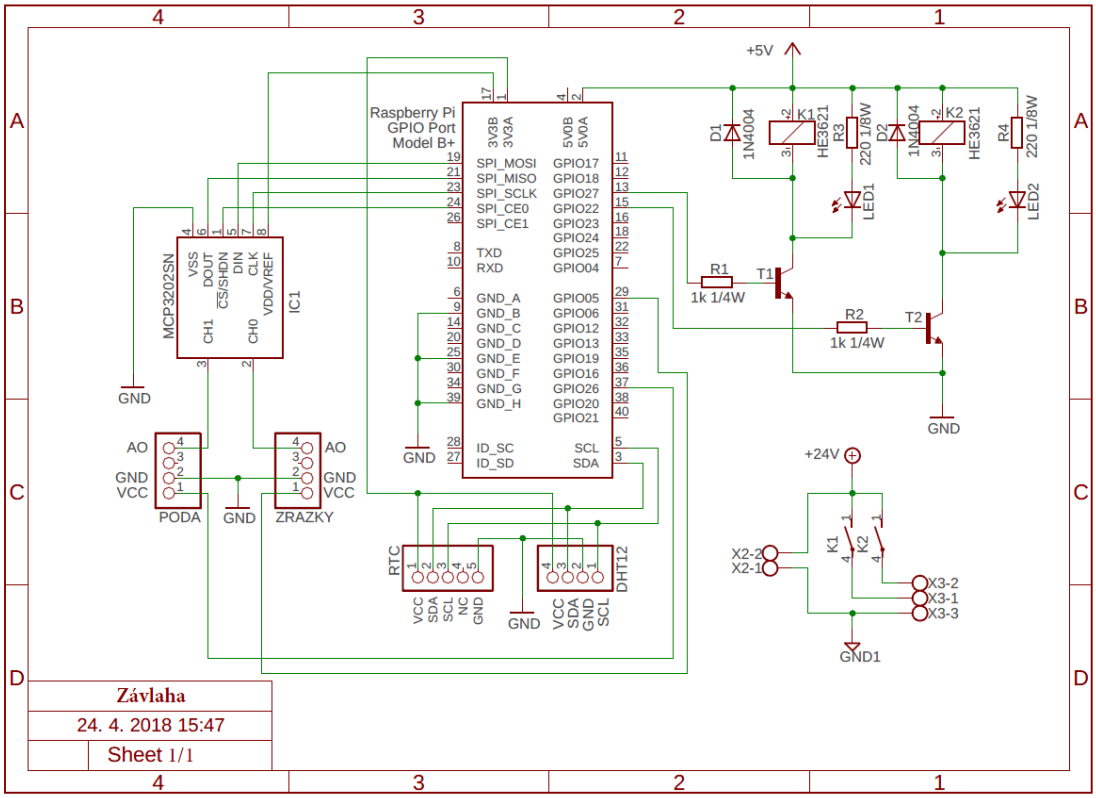
V pravej hornej časti sa nachádza obvod ovládajúci dvojicu relé pomocou GPIO výstupov a zapojenie signalizujúce zopnutie príslušného relé. Oba obvody sú napájané z +5V výstupu Raspberry Pi.

V pravej spodnej časti schematického zapojenia sa nachádza dvojica konektorov, ktoré slúžia na pripojenie vstupného napájacieho napätia pre solenoidy (X2) a pripojenie solenoidov samotných (X3). Solenoidy majú spoločnú zem a pomocou relé môžu byť jednotlivo alebo súčasne pripojené k napájaciemu napätiu.

V strednej spodnej časti sa nachádzajú moduly pripojené pomocou I^2C zbernice a to konkrétne RTC a DHT12.

V ľavej časti schémy je časť zapojenia zabezpečujúca meranie analógových hodnôt. Mikročip MCP3202 je pripojený pomocou SPI zbernice k Raspberry Pi a modulom na meranie zrážok a pôdnej vlhkosti. Keďže sú tieto moduly napájané jednosmerným napätím, pri trvalom napájaní týchto modulov by dochádzalo k rýchlej

degradácii ich senzorových plôch. Z uvedeného dôvodu sú moduly napájané pomocou dvoch GPIO výstupov aby bolo možné moduly zapínať na krátku dobu počas merania.



Obr. 33 Schéma zapojenia dosky plošných spojov

4.1.3 Rozmiestnenie súčiastok

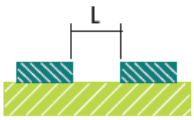
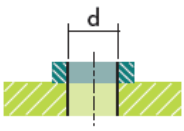
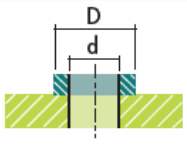
Základným kameňom tejto dosky je 40 pinový konektor, ktorý sa nasunie na rovnako veľký GPIO konektor umiestnený na Raspberry Pi B+. Rozmery dosky sú prispôsobené tak, aby nedošlo ku konfliktu so vstupno-výstupnými perifériami rozmiestnenými na základnej doske Raspberry Pi B+. Čísla GPIO výstupov boli zvolené vzhľadom k polohe umiestnenia súčiastok na doske tak, aby bolo prepojenie čo najjednoduchšie.

Doska je navrhnutá ako jednovrstvová so spojmi na spodnej strane dosky, kde sú taktiež umiestnené SMD súčiastky. Vyrobená doska je ale dvojvrstvová, keďže potrebujem zo strany drôtových súčiastok (vrchnej strany dosky) naletovať 40 pinový konektor.

Pri návrhu som použil niekoľko vstavaných knižníc pre základné súčiastky ako tranzistor, dióda, SMD odpory, relé, svorkovnice a taktiež pre mikročip MCP3202 s puzdrom SO8. Ďalej bolo potrebné stiahnuť alebo vytvoriť knižnice pre ostatné moduly. Konektor na prepojenie s Raspberry Pi som našiel na eagle stránke [70]. Ostatné konektory a značky pre DHT12, RTC a analógové moduly som vytvoril upravením vstavaných knižníc.

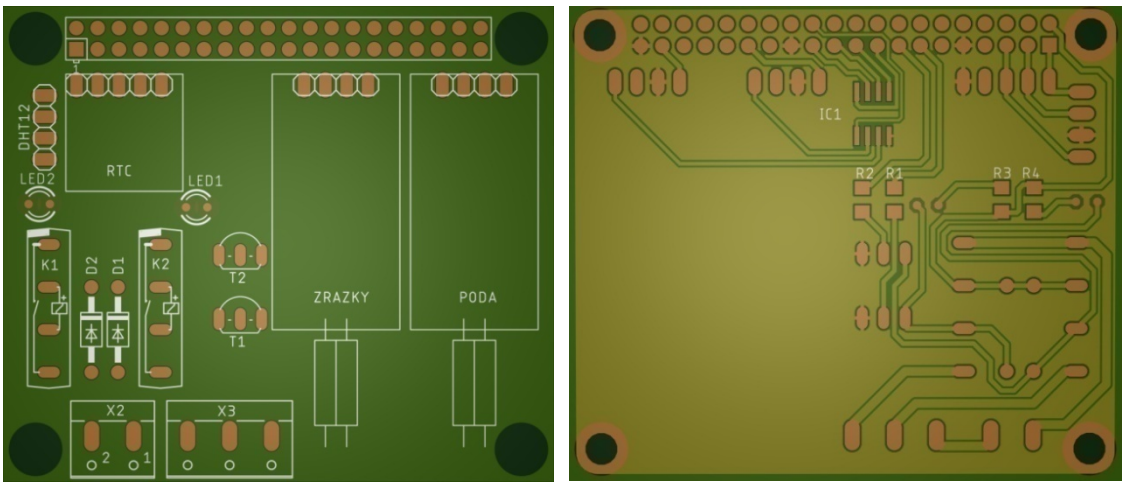
Pri vytváraní vodivých prepojení som sa snažil dodržiavať základné pravidlá návrhu DPS, ako napríklad použitie dostatočne veľkej šírky spojov, minimálne priblíženie výkonovej a logickej časti obvodu, logické usporiadanie komponentov, vhodná pozícia konektorov a podobne.

Podľa zadaných parametrov špecifikovaných firmou vyrábajúcu DPS som nastavil kontrolu týchto parametrov pomocou nástroju zvaného "DRC" (kontrola návrhových pravidiel) a dosku navrhoval so zreteľom na pravidlá nastavené na nasledujúce hodnoty:

tloušťka desky/Cu		tloušťka spoje (mm)		izolační dist.(mm)		vrták (mm)		mezikruží D - d (mm)	
								 0,3 < d ⩽ 1	
L=(mm)	l=(um)	min.	dop. ⩾	min.	dop. ⩾	min.	dop. ⩾	min.	dop. ⩾
0,5 - 1,5	35	0,2	0,25	0,2	0,25	0,2	0,6	0,5	0,5

Obr. 34 Návrhové pravidlá pre dosku plošných spojov [71]

V hornej časti dosky sa nachádza 40 pinový konektor, V ľavej hornej časti dosky sú konektory pre pripojenie RTC a DHT12 modulov. V pravej hornej časti dosky sú konektory pre pripojenie modulov na snímanie analógových hodnôt, ktoré zaberajú značnú časť dosky. V ľavej strednej časti je umiestnený obvod pre ovládanie dvojice relé a signálne led diódy. V ľavej spodnej časti sa nachádza dvojica konektorov pre pripojenie solenoidov a ich externého napájania.

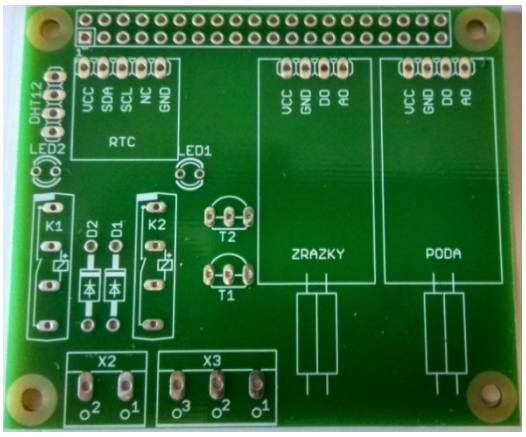


Obr. 35 Rozmiestnenie súčiastok dosky plošných spojov

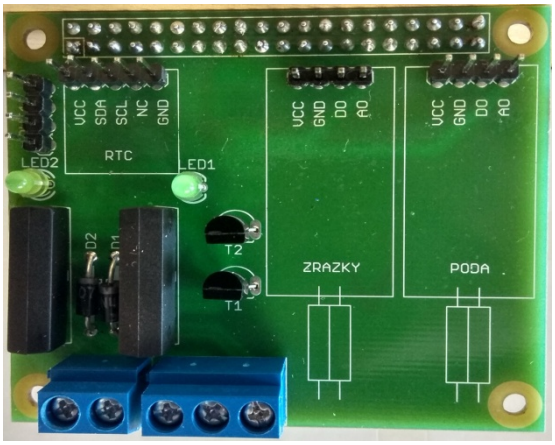
V *Tabuľka 4-E Zoznam použitých súčiastok* sa nachádza zoznam použitých súčiastok, ktorý bol vygenerovaný pomocou programu Eagle.

4.1.4 Osadenie dosky

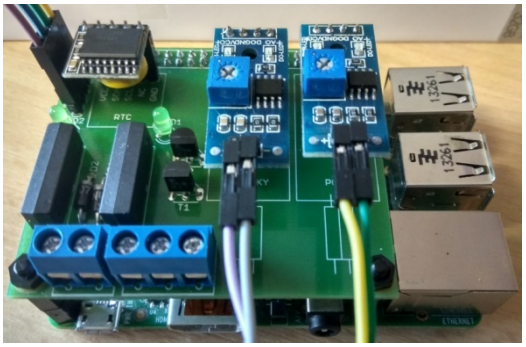
Po navrhnutí a vyrobení dosky bolo potrebné dosku osadiť a otestovať. Na nasledovných fotografiách môžeme vidieť holú dosku z výroby, následne osadenú súčiastkami a nakoniec aj s pripojenými modulmi a senzormi pripravenú na otestovanie.



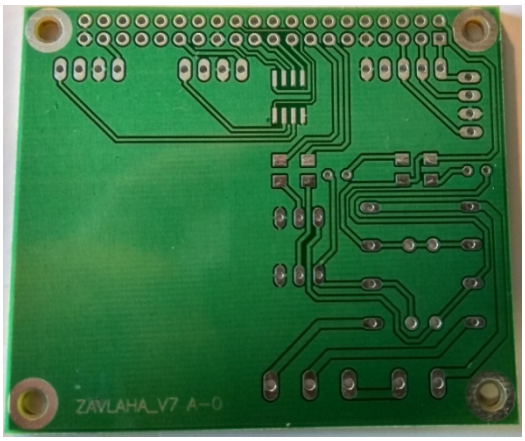
Obr. 36 Doska z výroby - vrchná časť



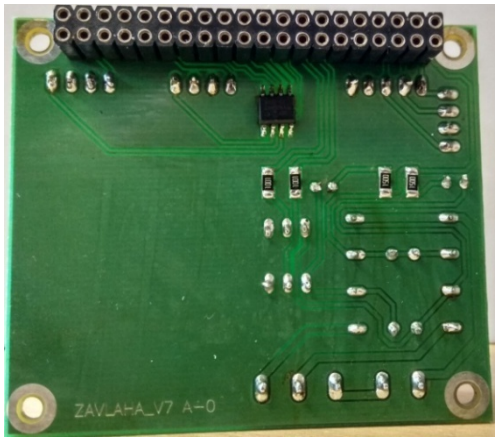
Obr. 37 Osadená doska - vrchná časť



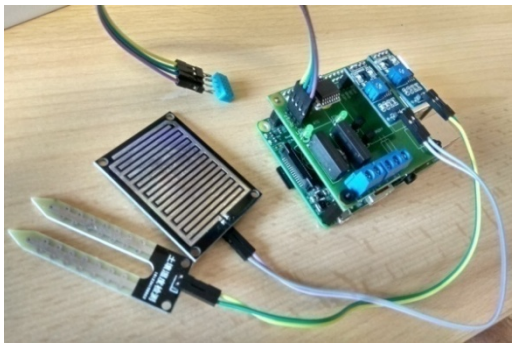
Obr. 38 Doska s modulmi



Obr. 39 Doska z výroby - spodná časť



Obr. 40 Osadená doska - spodná časť



Obr. 41 Doska s modulmi a senzormi

4.1.5 Opravy chýb dosky

Ako to tak pri tvorbe väčšiny prototypov býva, ani táto doska sa nezaobišla bez drobných návrhových chýb. Nájdené chyby stručne popíšem a keďže sa nejedná o závažné chyby a zapojenie je plne funkčné, nie je potrebné dávať dosku plošných spojov znovu do výroby.

Pri osadzovaní dosky som zistil, že chýba označenie poradia vývodov pre senzor DHT12 a svorkovníc pre solenoidy.

Taktiež plôšky u niektorých súčiastok neboli dostatočné veľké, čo malo za následok zložitejšie letovanie.

4.1.6 Napájanie a spotreba

V prípade výpadku napájacieho napätia sa elektromagnetické ventily vďaka ich konštrukčnému prevedeniu automaticky uzatvoria. Po následnom zapnutí systému sa načíta predošlí stav, ktorý je uložený v súbore stav.txt. Stavový súbor sa automaticky aktualizuje pri akejkoľvek zmene. Systém je vybavený modulom reálneho času s vlastnou batériou, preto nie je potrebné po každej strate napájania znovu nastavovať čas a dátum.

V prípade použitia elektromagnetických ventilov so solenoidmi napájanými jednosmerným napätím z batérie a pomocnej 5V batérie pre napájanie Raspberry, by bolo možné mnou navrhnutý systém používať aj v miestach bez možnosti pripojenia k elektrickej sieti. Keďže bol systém navrhovaný na napájanie zo siete, pomocná batéria by musela mať elektronický obvod, ktorý by zabezpečil stabilné napätie pre správny chod Raspberry.

Orientačná spotreba navrhnutého systému je približne [72]:

$$P_N = \mathrm{U_R}.\mathrm{I_R} = 5.0{,}24 \doteq 1{,}2\ \mathrm{VA} \qquad \textbf{(3)}$$

Kde P_N je výkon systému počas nečinnosti, teda ak sa nezavlažuje ani jeden okruh a beží len riadiaci program, U_R je napájacie napätie Raspberry a I_R je približný prúd odoberaný Raspberry počas behu riadiaceho programu [64].

$$P_T = \mathrm{U_R}.(\mathrm{I_R}+\mathrm{I_T}) + \mathrm{U_V}.\mathrm{I_{VP}} = 5.(0{,}24 + 28{,}36.10^{-3}) + 24.0{,}21 \doteq 6{,}38\ \mathrm{VA} \qquad \textbf{(4)}$$

Kde P_T je typický výkon systému počas zalievania jedného okruhu po odznení prechodových dejov, U_R je napájacie napätie Raspberry, I_R je približný prúd odoberaný Raspberry počas behu riadiaceho programu [64], I_T je prúd tečúci emitorom jedného otvoreného tranzistoru, U_V je napájacie napätie ventilov a I_{VP} je prídržný prúd tečúci jedným zavlažovacím ventilom.

$$P_M = \mathrm{U_R}.(\mathrm{I_R} + 2.\mathrm{I_T}) + \mathrm{U_V}.2.\mathrm{I_{VP}} = 5.(0{,}24 + 2.28{,}36.10^{-3}) + 0{,}37.2.24 \doteq 19{,}2\ \mathrm{VA} \quad \textbf{(5)}$$

Kde P_M je maximálny výkon systému počas spustenia zalievania dvoch okruhov súčasne, U_R je napájacie napätie Raspberry, I_R je približný prúd odoberaný Raspberry počas behu riadiaceho programu, I_T je prúd tečúci emitorom jedného otvoreného tranzistoru, U_V je napájacie napätie ventilov a I_{VP} je maximálny prechodný prúd tečúci jedným zavlažovacím ventilom.

4.2 Overenie funkčnosti

Pre overenie funkčnosti bolo potrebné zavlažovací systém upevniť na stenu, spraviť nové zásuvky pre napájanie Raspberry Pi a elektromagnetických ventilov, vhodne rozmiestniť a pripojiť senzory, prepojiť zavlažovací systém s domácou sieťou, vymeniť mechanické ventile za elektromagnetické a pripojiť ich spolu s napájaním k riadiacemu systému.

Boli použité elektromagnetické ventile typu Hunter solenoid PGV 1" [72]. Solenoidy ventilov sú napájané 24V striedavým napätím generovaným zvončekovým transformátorom. Napájanie Raspberry Pi je zabezpečené klasickou 5V/1A nabíjačkou na telefón s micro USB konektorom.

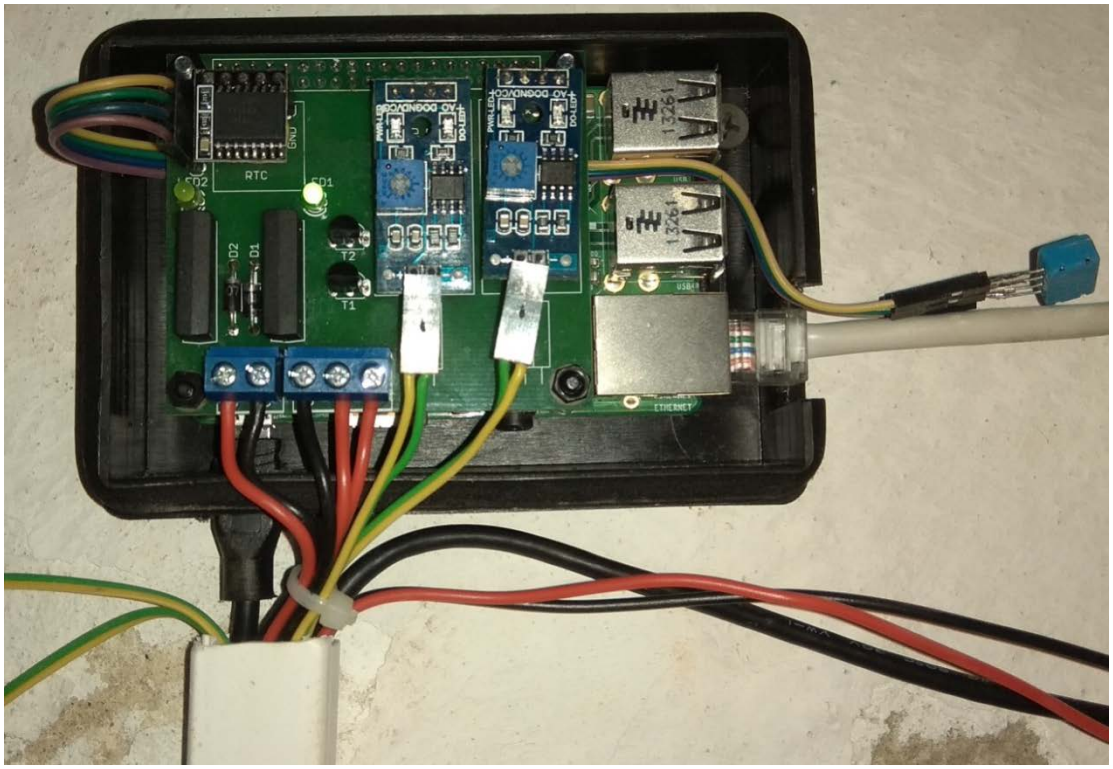
Po niekoľkých mesiacoch strávených:

- vyčerpávajúcim programovaním;
- čítaním enormného množstva katalógových listov, návodov a príspevkov;
- úspešnými a neúspešnými pokusmi;
- hľadaním vhodných modulov, senzorov a súčiastok;
- navrhovaním a osadzovaním dosky plošných spojov;
- pripravovaním elektroinštalácie;
- všetkým ostatnými zdanlivo zanedbateľnými činnosťami.

bolo načase zavlažovací systém otestovať. Po otvorení webového prehliadača a zadaní statickej lokálnej adresy ma napočudovanie privítalo moje grafické rozhranie a všetko fungovalo bezproblémovo.

Nasledovalo niekoľko hodín usilovného testovania celého systému a zhotovovanie foto dokumentácie. Počas testovania som nenarazil na žiadne nepredvídané chovanie alebo nestabilitu programu. Fungovali všetky senzory, manuálne aj automatické ovládanie jednotlivých okruhov a systém nemal taktiež problém s trojicou súčasne pripojených klientov pomocou wifi a mobilných telefónov.

Na druhý deň som oboznámil užívateľov (rodinu) s ovládaním závlahového systému a spoločne sme nastavili automatické časy zavlažovania.



Obr. 42 Zavlažovací systém

Obr. 43 Elektromagnetické ventile Toro

Obr. 44 Detail na elektromagnetický ventil

Obr. 45 Zvončekový transformátor [73]

5 ZÁVER

Cieľom diplomovej práce bolo navrhnúť a vytvoriť plne funkčný elektronický systém pre riadenie závlahového systému vybavený potrebnými senzormi, akčnými členmi a riadiacim programom vrátane vizualizácie.

V úvode práce som sa venoval rozboru zadania a dôvodom, ktoré ma videli k vypracovaní tejto práce. Taktiež boli stanovené ciele práce.

Teoretický rozbor obsahuje základné informácie o zavlažovacích systémoch a podobnejšie sa zaoberá elektronickými komponentmi týchto systémov. Ďalšia časť sa venuje návrhu konceptu elektronického zavlažovacieho systému a výberu vhodného mikrokontroléru na riadenie elektronickej závlahy. V závere teoretického rozboru sú vysvetlené princípy požitých zberníc a stručne charakterizované programy a programovacie jazyky, ktoré boli pri tvorbe práce využité.

Jadrom práce je vyvinutý software a hardware. Po softwarovej stránke je popísaný spôsob pripojenia a programovania Raspberry Pi, tvorba riadiaceho programu zavlažovacieho systému písaného v jazyku python a taktiež rozbor grafického rozhrania vytvoreného pomocou webovej stránky.

Záver práce je zameraný na popis postupu návrhu dosky plošných spojov, popisuje použité typy modulov, senzorov a súčiastok. Následne podrobnejšie rozoberá navrhnutú schému zapojenia a výsledné rozmiestnenie súčiastok na doske. Posledná časť práce je venovaná overeniu funkčnosti navrhnutého systému.

Všetky stanovené ciele sa mi podarilo splniť. Navrhnutý systém je plne funkčný a momentálne nasadený v testovacej prevádzke. Počas testovania závlahového systému sa naskytlo niekoľko možných vylepšení ako napríklad sprehľadnenie vizualizácie, doplnenie rozlíšenia dní v týždni pre zavlažovacie cykly alebo možnosť ovládať závlahu aj cez internet.

Literatúra

[1] Ovládacie jednotky a čidlá. In: ProRain [online]. ©2018 [cit. 2018-01-04] Dostupné z: http://www.prorain.sk/zavlaha/ovladacie-jednotky-a-cidla/

[2] Interiérové ovládacie jednotky Rain Bird. In: ProRain [online]. ©2018 [cit. 2018-01-04] Dostupné z: http://www.prorain.sk/zavlaha/ovladacie-jednotky-a-cidla/interierove/

[3] Riadiaca jednotka TORO DDC4 - 4 sekcie - interná. In: KD Garden [online]. ©2016 [cit. 2018-05-11]. Dostupné z: http://www.kdgarden.sk/riadiaca-jednotka-toro-ddc4-4-sekcie-interna-products-1144-16.aspx?productcathegory=203

[4] Riadiaca jednotka TORO DDCWP 2 sekcie. In: KD Garden [online]. ©2016 [cit. 2018-05-11]. Dostupné z: http://www.kdgarden.sk/riadiaca-jednotka-toro-ddcwp-2-sekcie-products-1523-16.aspx?productcathegory=206

[5] Riadiaca jednotka HUNTER XCORE 201 I E - 2 sekcie - interná. In: KD Garden [online]. ©2016 [cit. 2018-05-11]. Dostupné z: http://www.kdgarden.sk/riadiaca-jednotka-hunter-xcore-201-i-e-2-sekcie-interna-products-2491-16.aspx?productcathegory=184

[6] Riadiaca jednotka HUNTER HYDRAWISE HC 601i-E 6 sekcií. In: KD Garden [online]. ©2016 [cit. 2018-05-11]. Dostupné z: http://www.kdgarden.sk/riadiaca-jednotka-hunter-hydrawise-hc-601ie-6-sekcii-interna-rocna-licencia-enthusiasm-zadarmo-products-2680-16.aspx?productcathegory=557

[7] Elektromagnetické ventily. In: ProRain [online]. ©2018 [cit. 2018-01-04] Dostupné z: http://www.prorain.sk/zavlaha/elektromagneticke-ventily/

[8] Daždový senzor k závlahovému systému. In: ProRain [online]. ©2018 [cit. 2018-01-04] Dostupné z: http://www.prorain.sk/zavlaha/ovladacie-jednotky-a-cidla/cidla/

[9] Senzor pôdnej vlhkosti SMRT-Y. In: ProRain [online]. ©2018 [cit. 2018-01-04] Dostupné z: http://www.prorain.sk/zavlaha/ovladacie-jednotky-a-cidla/cidla/senzor-podnej-vlhkosti-rain-bird-smrt-y/

[10] Raspberry Pi 1 Model B+ 512MB RAM. In: RPiShop.cz [online].[cit. 2018-05-01] Dostupné z: http://rpishop.cz/raspberry-pi-pocitace/74-raspberry-pi.html

[11] Pin Numbering - Raspberry Pi Model B+. In: Pi4J [online]. ©2012-2018 [cit. 2018-05-01] Dostupné z URL: http://pi4j.com/pins/model-b-plus.html

[12] Zbernica. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2006, aktualizované 2016-02-28 [cit. 2018-05-10]. Dostupné z: https://sk.wikipedia.org/wiki/Zbernica

[13] Synchrónna komunikácia SPI. In: Kiwiki [online]. 21.3.2013 [cit. 2018-05-10]. Dostupné z: www.kiwiki.info/index.php/Synchrónna_komunikácia_SPI

[14] Synchrónna komunikácia I2C. In: Kiwiki [online]. 21.3.2013 [cit. 2018-05-10]. Dostupné z: www.kiwiki.info/index.php/Synchrónna_komunikácia_I2C

[15] SPI Block Guide V03.06. In: OpenCores [online]. 21.1.2000 ©Motorola, Inc., 2001, aktualizované 4.2.2003 [cit. 2018-05-10]. Dostupné z: https://opencores.org/usercontent/doc/1499360489

[16] I2C BUS. John Heritage [online]. London: John Heritage, ©2017, March 10, 2014 [cit. 2018-05-10]. Dostupné z: http://www.johnheritage.co.uk/i2c-bus/

[17] Case study, implementation and examples of I2C. Communication Protocols Assignments [online]. Abhiraj Nambiar, 2015, 3 May 2015 [cit. 2018-05-10]. Dostupné z: http://cpassignments.blogspot.cz/2015/05/case-study-implementation-and-examples.html

[18] Win32 Disk Imager download. In: SourceForge.net [online]. Slashdot Media, ©2018 [cit. 2018-05-11]. Dostupné z: https://sourceforge.net/projects/win32diskimager/

[19] About. Angry IP Scanner [online]. Anton Keks [cit. 2018-05-10]. Dostupné z: http://angryip.org/about/

[20] Screenshots. Angry IP Scanner [online]. Anton Keks [cit. 2018-05-10]. Dostupné z: http://angryip.org/screenshots/

[21] PuTTY - populárny klient protokolov SSH, Telnet, Rlogin [online]. Ján Fečík, ©2018 [cit. 2018-05-10]. Dostupné z: https://www.putty.sk/

[22] Čo je WinSCP. WinSCP [online]. ©2000-2018, 2014-09-17 [cit. 2018-05-10]. Dostupné z: https://winscp.net/eng/docs/lang:sk

[23] User Interfaces. WinSCP [online]. ©2000-2018, 2014-09-17 [cit. 2018-05-10]. Dostupné z: https://winscp.net/eng/docs/interfaces

[24] PyCharm: Python IDE for Professional Developers. In: JetBrains [online]. 2018 [cit. 2018-05-11]. Dostupné z: https://www.jetbrains.com/pycharm/

[25] Informace o produktu EAGLE. In: Eagle Online [online]. ELCAD, ©1994-2014, aktualizované 16.7.2014 [cit. 2018-05-11]. Dostupné z: http://www.eagle.cz/info.htm

[26] 3D Gerber Viewer. In: MayhewLabs.com [online]. Mayhew Labs, ©2015 [cit. 2018-05-11]. Dostupné z: http://mayhewlabs.com/3dpcb

[27] 3D Gerber Viewer: Fio example. In: MayhewLabs.com [online]. Mayhew Labs, ©2015 [cit. 2018-05-11]. Dostupné z: http://mayhewlabs.com/webGerber/?demo=Fio

[28] ObjGen: Live JSON Generator. In: ObjGen.com [online]. Everett Ward Systems, ©2018 [cit. 2018-05-11]. Dostupné z: http://www.objgen.com/json

[29] ObjGen Help: Live JSON Generator. In: ObjGen.com [online]. Everett Ward Systems, ©2018 [cit. 2018-05-11]. Dostupné z: http://www.objgen.com/help/json

[30] About us. In: Draw.io [online]. JGraph [cit. 2018-05-11]. Dostupné z: https://about.draw.io/about-us/

[31] Examples. In: Draw.io [online]. JGraph [cit. 2018-05-11]. Dostupné z: https://about.draw.io/features/examples/

[32] BLAHO, Andrej. Programovanie v Pythone: 1. Úvod — Dokumentácia. In: Python.input.sk [online]. Georg Brandl and the Sphinx team, ©2007-2018, 2017, aktualizované 9.5.2018 [cit. 2018-05-11]. Dostupné z: http://python.input.sk/01.html#jazyk-python

[33] Charakteristika HTML. In: Garth.cz [online]. [cit. 2018-05-11]. Dostupné z: https://www.citacepro.com/dokument/aeh6NOmNgV8acolL

[34] Charakteristika JavaScriptu. In: Garth.cz [online]. [cit. 2018-05-11]. Dostupné z: http://www.garth.cz/uvod-do-javascriptu/charakteristika-javascriptu/

[35] JQuery návod: Vše okolo jQuery. In: JADRNÝ, Tomáš. JQuery-návod.cz [online]. TEMPLATED, ©2010 [cit. 2018-05-11]. Dostupné z: http://jquery-navod.cz/kategorie-ostatni-clanky/1-uvodni-clanek

[36] Počítačové spracovanie informácií: Programové vybavenie počítačov. In: Moodle.uniag.sk: Slovenská poľnohospodárska univerzita v Nitre [online]. Moodle Pty Ltd: © 2018 [cit. 2018-05-11]. Dostupné z: http://moodle.uniag.sk/mod/resource/view.php?id=6941

[37] Ing. František Burian Ph.D. Raspberry Pi - ssh, HelloWorld. In: BPRP - Robotika a počítačové vidění [online]. Weby Google, ©2018 [cit. 2018-05-11]. Dostupné z: https://sites.google.com/a/vutbr.cz/bprp/cviceni/2016/1

[38] Ing. František Burian Ph.D. Instalace RaspberryPi, příprava prostředí pro programování. In: BPRP - Robotika a počítačové vidění [online]. Weby Google, ©2018 [cit. 2018-05-11]. Dostupné z: https://sites.google.com/a/vutbr.cz/bprp/cviceni/2017/02

[39] Raspbian. In: Raspberry Pi [online]. Raspberry Pi Foundation [cit. 2018-05-11]. Dostupné z: https://www.raspberrypi.org/downloads/raspbian/

[40] Win32 Disk Imager download. In: SourceForge.net [online]. Slashdot Media, ©2018 [cit. 2018-05-11]. Dostupné z: https://sourceforge.net/projects/win32diskimager/files/latest/download

[41] Angry IP Scanner: Download for Windows, Mac or Linux. In: KEKS, Anton (angryziber). Angry IP [online]. [cit. 2018-05-11]. Dostupné z: http://angryip.org/download/#windows

[42] Setting up a web server on a Raspberry Pi. In: Raspberry Pi [online]. Raspberry Pi Foundation [cit. 2018-05-11]. Dostupné z: https://www.raspberrypi.org/documentation/remote-access/web-server/README.md

[43] What is: Apache. In: WPBeginner: Beginner's Guide for WordPress [online]. WPBeginner, ©2009-2018 [cit. 2018-05-11]. Dostupné z: http://www.wpbeginner.com/glossary/apache/

[44] Setting up an Apache Web Server on a Raspberry Pi. In: Raspberry Pi [online]. Raspberry Pi Foundation [cit. 2018-05-11]. Dostupné z: https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md

[45] A comparison between WebSockets, server-sent events, and polling. In: Aquil.io [online]. ©2018 [cit. 2018-05-11]. Dostupné z: https://aquil.io/articles/a-comparison-between-websockets-server-sent-events-and-polling

[46] Websocket Tutorial. In: Mbed [online]. Arm Limited, ©1995-2018 [cit. 2018-05-11]. Dostupné z: https://os.mbed.com/cookbook/Websockets-Server

[47] Tornado Web Server. In: Tornado [online]. The Tornado Authors, ©2009-2018 [cit. 2018-05-11]. Dostupné z: http://www.tornadoweb.org/en/stable/

[48] Install DS3231 Real Time Clock - Latest Info. In: Raspberry Pi [online]. Raspberry Pi Foundation, 26.9.2016 [cit. 2018-05-11]. Dostupné z: https://www.raspberrypi.org/forums/viewtopic.php?t=161133

[49] Python example and library for DHT12. In: Raspberry Pi [online]. Raspberry Pi Foundation, 15.1.2017 [cit. 2018-05-11]. Dostupné z: https://www.raspberrypi.org/forums/viewtopic.php?t=171601

[50] Smbus read_i2c_block_data Fails on Latest Kernel. In: GitHub [online]. 2018 [cit. 2018-05-11]. Dostupné z: https://github.com/raspberrypi/firmware/issues/828

[51] SOP8 SO8 SOIC8 TSSOP8 MSOP8 to DIP8 Adapter PCB Conveter Board. In: Inovatek Electronics Ltd [online]. [cit. 2018-05-11]. Dostupné z: http://www.flytron.com/connectorscables/293-sop8-so8-soic8-tssop8-msop8-to-dip8-adapter-pcb-conveter-board.html

[52] Enable SPI Interface on the Raspberry Pi. In: HAWKINS, Matt. Raspberrypi-spy.co.uk [online]. 2017, 24.8.2014 [cit. 2018-05-11]. Dostupné z: https://www.raspberrypi-spy.co.uk/2014/08/enabling-the-spi-interface-on-the-raspberry-pi/

[53] NUTTALL, Ben. API - SPI Devices: MCP3002. In: GPIO Zero [online]. Raspberry Pi Foundation, 2017 [cit. 2018-05-11]. Dostupné z: http://gpiozero.readthedocs.io/en/stable/api_spi.html#analog-to-digital-converters-adc

[54] RC.LOCAL. In: Raspberry Pi [online]. RaspberryPiFoundation [cit. 2018-05-11]. Dostupné z: https://www.raspberrypi.org/documentation/linux/usage/rc-local.md

[55] Source code for tornado.websocket. In: The Tornado Authors [online]. 2018 [cit. 2018-05-11]. Dostupné z: http://www.tornadoweb.org/en/stable/_modules/tornado/websocket.html

[56] HTML5 Tutorial. In: W3Schools [online]. Refsnes Data, ©1999-2018 [cit. 2018-05-11]. Dostupné z: https://www.w3schools.com/html/default.asp

[57] JavaScript Tutorial. In: W3Schools [online]. Refsnes Data, ©1999-2018 [cit. 2018-05-11]. Dostupné z: https://www.w3schools.com/js/default.asp

[58] How to detect Safari, Chrome, IE, Firefox and Opera browser?. In: Stack Overflow: Stack Exchange Inc [online]. ©2018, 20.7.2017 [cit. 2018-05-11]. Dostupné z: https://stackoverflow.com/questions/9847580/how-to-detect-safari-chrome-ie-firefox-and-opera-browser

[59] Download free version of EAGLE. In: Autodesk Inc [online]. ©2018 [cit. 2018-05-11]. Dostupné z: https://www.autodesk.com/products/eagle/free-download

[60] KEVIN. Raspberry Pi – Driving a Relay using GPIO. In: SusaNET: Just another WordPress site [online]. ©2018, 26.6.2012 [cit. 2018-05-11]. Dostupné z: https://www.susa.net/wordpress/2012/06/raspberry-pi-relay-using-gpio/

[61] DATA SHEET: BL-B2141. In: Bright Led Electronics Corp [online]. ©2018 [cit. 2018-05-11]. Dostupné z: https://www.gme.cz/data/attachments/dsh.511-051.1.pdf

[62] DATA SHEET: GENERAL PURPOSE CHIP RESISTORS RC1206. In: YAGEO Corporation [online]. 02.7.2009 [cit. 2018-05-11]. Dostupné z: https://www.gme.cz/data/attachments/dsh.900-330.1.pdf

[63] BC337…BC338: NPN Silicon Epitaxial Planar Transistor. In: GM electronic, spol. s.r.o. [online]. 2018 [cit. 2018-05-11]. Dostupné z: https://www.gme.cz/data/attachments/dsh.210-018.1.pdf

[64] FAQS: POWER. In: Raspberry Pi [online]. Raspberry Pi Foundation [cit. 2018-05-11]. Dostupné z: https://www.raspberrypi.org/help/faqs/#topPower

[65] MCP3202: 2.7V Dual Channel 12-Bit A/D Converter with SPI® Serial Interface. In: Microchip Technology Inc [online]. ©1999 [cit. 2018-05-11]. Dostupné z: https://www.gme.cz/data/attachments/dsh.931-011.1.pdf

[66] Modul senzoru vlhkosti zeminy. In: PTshop [online]. [cit. 2018-05-11]. Dostupné z: https://www.ptshop.cz/Modul-senzoru-vlhkosti-zeminy-d47.htm

[67] Modul detekce deště/sněhu. In: PTshop [online]. [cit. 2018-05-11]. Dostupné z: https://www.ptshop.cz/Modul-detekce-deste-snehu-d177.htm?tab=description

[68] DS3231 PRECISION RTC MEMORY MODULE FOR ARDUINO. In: Alex NLD [online]. ©2018 [cit. 2018-05-11]. Dostupné z: https://alexnld.com/product/ds3231-precision-rtc-memory-module-for-arduino/

[69] Digital temperature: DHT12 Product. In: Guangzhou Aosong Electronics Co., Ltd. [online]. ©2003-2018 [cit. 2018-05-11]. Dostupné z: http://www.robototehnika.ru/file/DHT12.pdf

[70] How do I use a 'template' from a library?. In: Autodesk [online]. ©Autodesk Inc 2018 [cit. 2018-05-10]. Dostupné z: https://knowledge.autodesk.com/support/eagle/troubleshooting/caas/discussion/t5/EAGLE-Forum/How-do-I-use-a-template-from-a-library-/td-p/6862687.html

[71] Návrhová pravidla pro plošné spoje v závislosti na tloušťce materiálu. In: AJ Technology [online]. [cit. 2018-05-10]. Dostupné z: http://www.ajtechnology.cz/tabulka.pdf

[72] PGV VALVE. In: Hunter Industries Incorporated [online]. ©2013 [cit. 2018-05-10]. Dostupné z: https://www.hunterindustries.com/sites/default/files/BR_PGV_em.pdf

[73] Kanlux 23260 KTF-8-24 Zvončekový transformátor. In: Svietidlá Starlux [online]. ©2014-2016 [cit. 2018-05-10]. Dostupné z: https://www.elampy.sk/26226-kanlux-23260-ktf-8-24-zvoncekovy-transformator

Zoznam symbolov, veličín a skratiek

"	Palec, značka dĺžky
A	Ampér, značka prúdu
AC	Alternating current, striedavý prúd
ACK	Acknowledge, potvrdenie
AD/ADC	Analog to digital converter, analógovo digitálny prevodník
ARM	Advanced RISC Machine, architektúra procesorov
ASCII	American Standard Code for Information Interchange, kódovací systém
AVC	Advanced Video Coding, štandard kódovania obrazu
CAN	Controller Area Network, zbernica
CPHA	Clock phase, fáza hodinového signálu
CPOL	Clock polarity, polarita hodinového signálu
CSI	Camera Serial Interface, konektor kamerového rozhrania
CSS	Cascading Style Sheets, kaskádové štýly
CSV	Comma-separated values, hodnoty oddelené čiarkou
DA/DAC	Digital to nalog converter, digitálno analógový prevodník
DC	Direct current, jednosmerný prúd
DIP8	Typ púzdra
DOM	Document Object Model, objektový model dokumentu
DPS	Doska plošných spojov
DRC	Design Rule Checke, kontrola návrhových pravidiel
EEPROM	Electrically Erasable Programmable Read-Only Memory, druh pamäte
FIFO	First in First out, druh zásobníku
FTP	File Transfer Protocol, protokol prenosu súborov
GPIB	General Purpose Interface Bus, zbernica
GPIO	General-purpose input/output, všeobecne použiteľný vstup/výstup
GPU	Graphics processing unit, grafický procesor
GSM	Global System for Mobile Communications, Globálny systém mobilných komunikácií
H.264	Formát kódovania obrazu
HDMI	High-Definition Multimedia Interface, konektor
HTML	HyperText Markup Language, Hypertextový značkový jazyk
CH	Channel, kanál
I2C	Inter-Integrated Circuit, dvojžilová obojsmerná zbernica
I2S	Inter-IC Sound, audio konektor
ID	identifier, identifikačné číslo
IDE	Integrated Development Environment, integrované vývojové prostredie

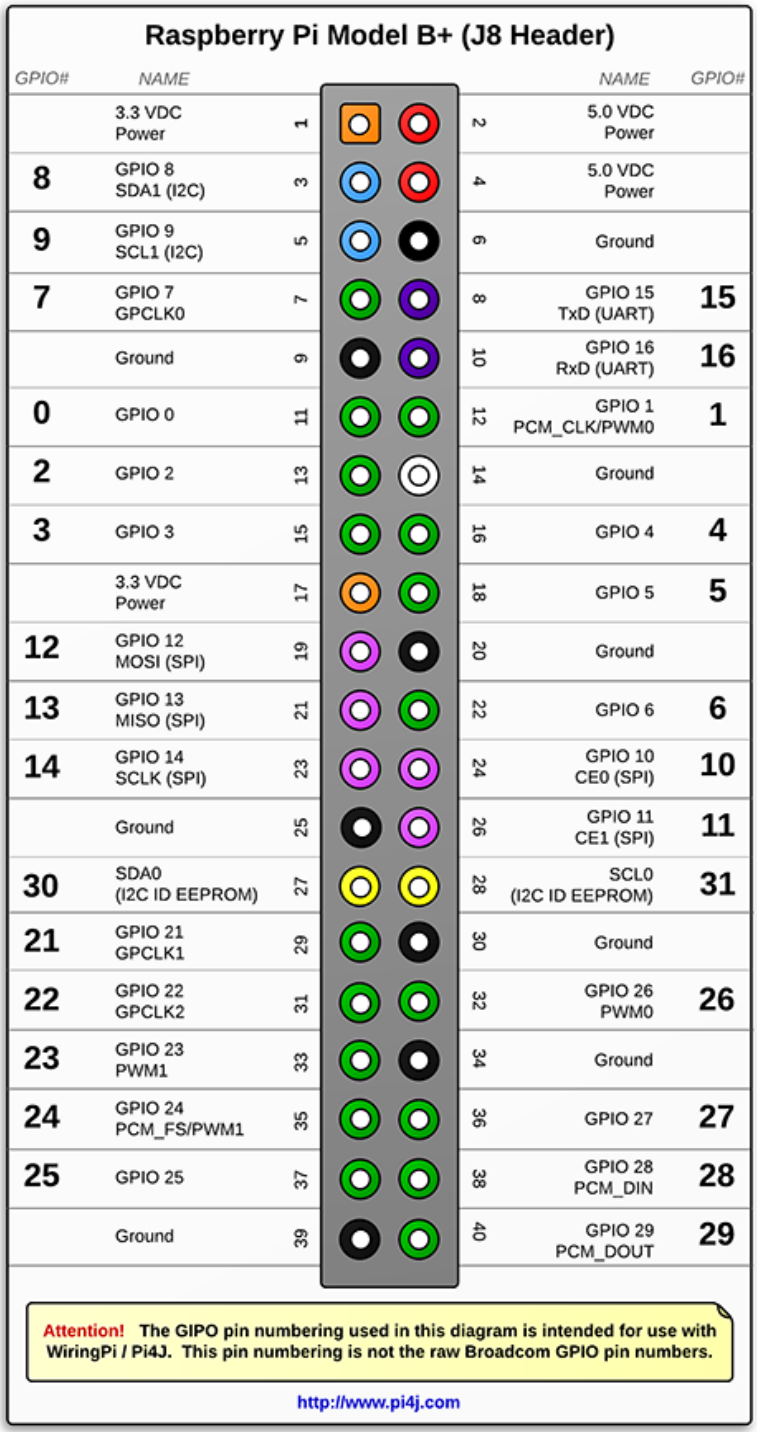
IEEE	Institute of Electrical and Electronics Engineers
IO	Input/output, vstupno-výstupný
iOS	iPhone OS, operačný systém
IP	Internet Protocol, komunikačný protokol
IP-Port	Špeciálny typ textového súboru
JSON	JavaScript Object Notation, dátový formát
LAN	Local area network, lokálna počítačová sieť
LCD	Liquid crystal display, displej s kvapalnými kryštálmi
LED	Light emitting diode, luminiscenčná dióda
MAC	Macintosh, druh PC
MicroSDHC	Micro Secure Digital High Capacity, formát pamäťovej karty
MIPIM	Konektor kamerového rozhrania
MISO/SDO/DO/SO	Master In, Slave Out
MMC	MultiMediaCard, pamäťová karta
MOSI/SDI/DI/SI	Master Out, Slave In
MPEG-2	Formát kódovania obrazu
MPEG-4	Formát kódovania obrazu
MS-DOS	Microsoft Disk Operating System, operačný systém
NetBIOS	Network Basic Input/Output System, softwarové rozhraní
OpenGL	Open Graphics Library, priemyselný štandard
OS	Operating system, operačný systém
PC	Personal Computer, osobný počítač
PCB	Printed circuit board, doska plošných spojov
PGV	Professional Grade Valves, typ ventilu
PHP	Hypertext Preprocessor, skriptovací jazyk
PLC	Programmable Logic Controller, programovatelný logický automat
RAM	Random Access Memory, pamäť s priamym prístupom
RS-232	Recommended Standard, štandard sériovej linky
RS-485	Recommended Standard, štandard sériovej linky
RTC	Real-time clock, hodiny reálneho času
RV	Relatívna vlhkosť
SCLK/SCK/CLK/SCL	Serial Clock, sériové hodiny
SCP	Secure Copy Protocol, zabezpečený protokol kopírovania
SCSI	Small Computer System Interface, zbernica
SD	Secure Digital, pamäťová akrta
SDA	Serial Data, sériové dáta
SDRAM	Synchronous Dynamic Random Access Memory, typ pamäte
SFTP	SSH File Transfer Protocol, SSH protokol prenosu súborov
SMBus	System Management Bus, zbernica
SMD	Surface mount devices, typ púzdra
SMS	Short Message Service, krátka textová správa

SO8	Small Outline, typ púzdra
SOIC	Small Outline Integrated Circuit, púzdro
SPI	Serial Peripheral Interface, sériové periferné rozhranie
SQL	Structured Query Language, strukturovaný dotazovací jazyk
SS/nCS/CS/CSB/STE	Slave Select
SSH	Secure Shell, sieťový protokol
SSL	Secure Sockets Layer, šifrovací protokol
TCP	Transmission Control Protocol, protokol riadenia prenosu
TLS	Transport Layer Security, šifrovací protokol
TRRS	Tip ring ring sleeve, konektor (jack)
TXT	prípona textového súboru
UAMT	Ústav automatizace a měřící techniky
UML	Unified Modeling Language, grafický jazyk
UNIX	Operačný systém
USART	Universal Synchronous / Asynchronous Receiver and Transmitter, sériové rozhranie
USB	Universal Serial Bus, univerzálna sériová zbernica
UTF-8	8-bit Unicode Transformation Format, bezstratové kódovanie
V	Volt, značka napätia
VCS	Version control systems, verzovací systém
VDD/VREF	Power/reference Input, napájanie/referencia
VYP	Vypnuté
W	Wat, značka výkonu
WiFi	Wireless Fidelity, označenie bezdrôtových sietí
WYSIWYG	What You See Is What You Get, čo vidíš, to dostaneš
XML	eXtensible Markup Language, rozšíriteľný značkovací jazyk
ZAP	Zapnuté
Zlib	Knižnica kompresie dát
Ω	Ohm, značka odporu

Zoznam príloh

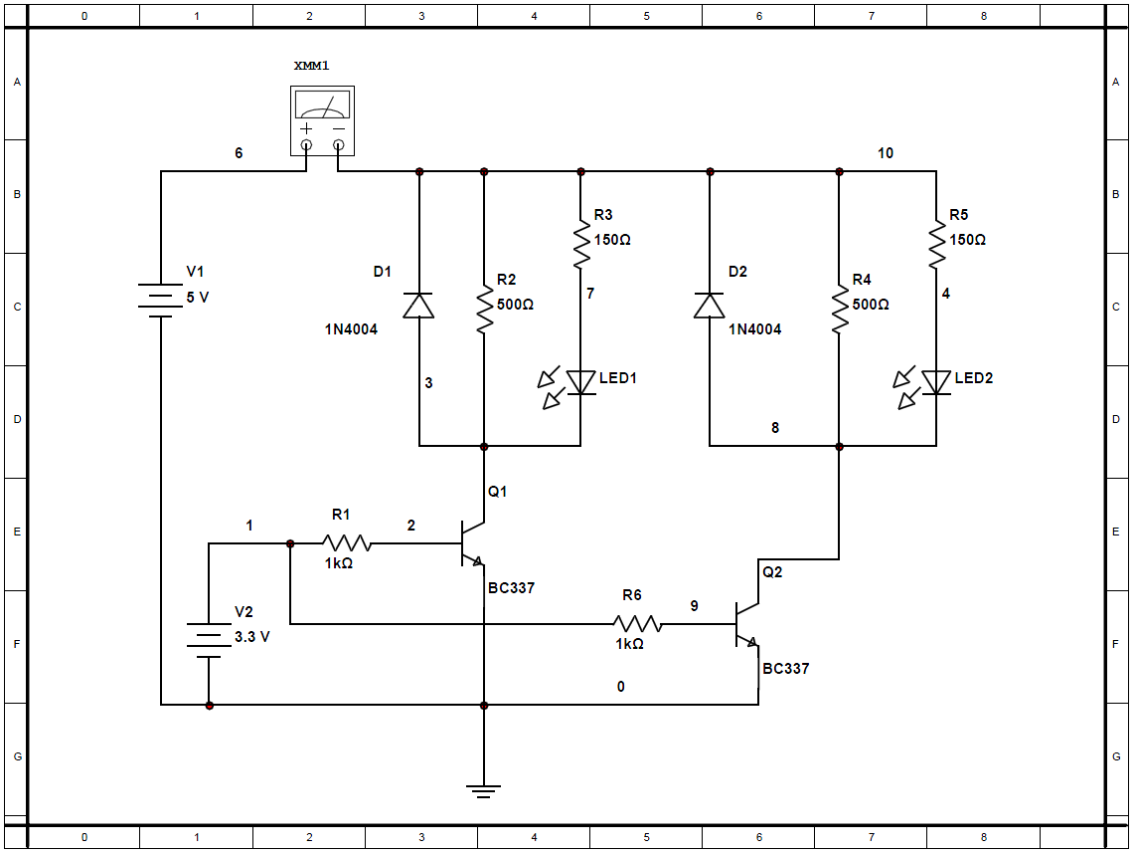
Prílohy

Príloha A: Popis 40 - pinového GPIO konektoru [8]



Obr. 46-A Popis GPIO konektoru [11]

Príloha B: Schéma zapojenia simulácie v Multisime



Obr. 47-B Schéma zapojenia simulácie v Multisime

Príloha C: Štruktúra stavovej premennej v JSON formáte


```
{
  "okruh": [
    {
      "stav": false,
      "cas": [
        {
          "on": "00:00",
          "off": "00:00"
        },
        {
          "on": "00:00",
          "off": "00:00"
        },
        {
          "on": "00:00",
          "off": "00:00"
        },
        {
          "on": "00:00",
          "off": "00:00"
        },
        {
          "on": "00:00",
          "off": "00:00"
        }
      ],
      "aut_rezim": true,
      "manualne": false
    },
    {
      "stav": false,
      "cas": [
        {
          "on": "00:00",
          "off": "00:00"
        },
        {
          "on": "00:00",
          "off": "00:00"
        },
        {
          "on": "00:00",
          "off": "00:00"
        },
        {
          "on": "00:00",
          "off": "00:00"
        }
      ],
      "aut_rezim": true,
      "manualne": false
    }
  ],
  "zaliev_aut_rezim": true,
  "cas": "20.04.2018 05:18:12",
  "cmd": "init",
  "prietok": {
    "stav": false,
    "hyst": 15,
    "bypass": false
  },
  "vlhkost": {
    "stav": false,
    "prah": 50,
    "hodnota": 2.0,
    "bypass": false
  },
  "zrazky": {
    "stav": false,
    "prah": 50,
    "hodnota": 2.1,
    "bypass": false
  },
  "dht": {
    "teplota": 26.5,
    "vlhkost": 35.8
  },
  "id": "91497622665589535"
}
```

Príloha D: Zoznam podporovaných komunikačných príkazov

{ "cmd": "zaliev_aut_rezim", "zaliev_aut_rezim": false }
{ "cmd": "okruh", "okruh": 0, "aut_rezim": false }
{ "cmd": "okruh", "okruh": 0, "cas": 0, "on": "06:10" }
{ "cmd": "okruh", "okruh": 0, "cas": 0, "off": "06:15" }
{ "cmd": "zrazky_bypass", "zrazky_bypass": false }
{ "cmd": "prietok_bypass", "prietok_bypass": false }
{ "cmd": "zrazky_stav", "zrazky_stav": true }
{ "cmd": "prietok_stav", "prietok_stav": true }
{ "cmd": "vlhkost_stav", "vlhkost_stav": true }
{ "cmd": "okruh", "okruh": 1, "stav": false }
{ "cmd": "prietok_hyst", "prietok_hyst": 15 }
{ "cmd": "init","stav":{},"id":123 }
{ "cmd": "okruh", "okruh": 1, "manualne": true }
{ "cmd": "warning", "msg": "xxxx" }
{ "cmd": "zrazky_prah", "zrazky_prah": 100 }
{ "cmd": "vlhkost_prah", "vlhkost_prah": 100 }
{ "cmd": "zrazky_hodnota", "zrazky_hodnota": 100 }
{ "cmd": "vlhkost_hodnota", "vlhkost_hodnota": 100 }
{ "cmd": "dht_teplota", "dht_teplota": 50 }
{ "cmd": "dht_vlhkost", "dht_vlhkost": 100 }

Príloha E: Zoznam použitých súčiastok

Tabuľka 4-E Zoznam použitých súčiastok

Part	Value	Device	Package	Description
D1	1N4004	1N4004	DO41-10	DIODE
D2	1N4004	1N4004	DO41-10	DIODE
DHT12		PINHD-1X4	1X04	PIN HEADER
IC1	MCP3202SN	MCP3202SN	SO-08	2.7V 2CH 12-Bit ADC SPI
K1	HE3621	HE3621	HE3621	RELAY
K2	HE3621	HE3621	HE3621	RELAY
LED1		LED3MM	LED3MM	LED
LED2		LED3MM	LED3MM	LED
PODA		ANALOG M	PIN_HEAD_1X4	PIN HEADER
R1	1k 1/4W	RR1206	R1206	RESISTOR
R2	1k 1/4W	RR1206	R1206	RESISTOR
R3	220 1/8W	RR1206	R1206	RESISTOR
R4	220 1/8W	RR1206	R1206	RESISTOR
RTC		RTCRTC	RTC	PIN HEADER
T1		BC337-25	TO92-CBE	NPN Transistror
T2		BC337-25	TO92-CBE	NPN Transistror
X1	RPI B+	RPI B+	RPI B+	RPi GPIO connector B+
X2		W237-102	W237-102	WAGO SCREW CLAMP
X3		W237-103	W237-103	WAGO SCREW CLAMP
ZRAZKY		ANALOG M	PIN_HEAD_1X4	PIN HEADER

Príloha F: CD

Obsah CD:

- PDF verzia práce "DP Marek Farba 2018.pdf".
- Staršia verzia I^2C driveru "i2c1-bcm2708.dtbo".
- Zdrojový kód webovej stránky - vizualizácie "index.html".
- Súbor potrebný pre inicializáciu riadiaceho programu "init.txt".
- Kópia Jquery knižnice "jquery.js".
- Zdrojový kód riadiaceho programu "main.py".
- Stavový súbor riadiaceho programu "stav.txt".
- Doska plošných spojov - doska "zavlaha_v7.brd".
- Doska plošných spojov - schéma "zavlaha_v7.sch".