



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 A MONITOROVÁNÍ KLIMATU VE
SKUPINÁCH TERÁRIÍ

CONTROL AND MONITORING OF CLIMATE IN GROUPS OF TERRARIUMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Pavlišin

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Macho, Ph.D.

BRNO 2017

Diplomová práce

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

Student: Bc. Tomáš Pavlišin
Ročník: 2

ID: 154829
Akademický rok: 2016/17

NÁZEV TÉMATU:

Řízení a monitorování klimatu ve skupinách terárií

POKYNY PRO VYPRACOVÁNÍ:

1. Rozeberte možnosti realizace systému pro monitorování a řízení klimatu (teploty a vlhkosti) ve skupinách terárií. Zvolte nejvhodnější koncepci. Každá skupina bude mít svůj lokální řídicí systém, který bude komunikovat s centrálním počítačem. Je požadován sběr údajů o teplotě a vlhkosti v jednotlivých teráriích, jejich archivace v databázovém systému na centrálním počítači a zpřístupnění přes webové rozhraní.
2. Vyberte vhodné bezdrátové moduly a navrhnete komunikační protokol pro komunikaci mezi centrálním počítačem a lokálními řídicími systémy.
3. Navrhnete systém pro lokální řízení klimatu ve skupině terárií. Zvolte vhodná čidla teploty a vlhkosti. Nakreslete obvodové schéma zapojení a stanovte hodnoty jednotlivých součástek. Systém realizujte.
4. Vytvořte softwarové vybavení pro systém řízení klimatu.
5. Nainstalujte a nakonfigurujte potřebný software (web server, databázový server) na centrální počítač.
6. Ověřte funkčnost celého zřízení a vyhodnoťte dosažené výsledky (přesnost řízení teploty a vlhkosti, spolehlivost bezdrátové komunikace).

DOPORUČENÁ LITERATURA:

[1] FROHN, Manfred - OBERTHÜR, Wolfgang - SIEDLER, Hans-Jobst - WIEMER Manfred - ZASTROW, Peter. Elektronika - polovodičové součástky a základní zapojení. Praha: BEN 2006. 500 s. ISBN 80-7300-123-3.

Termín zadání: 6.2.2017

Termín odevzdání: 15.5.2017

Vedoucí práce: Ing. Tomáš Macho, Ph.D.

Konzultant:

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

UPOZORNĚNÍ:

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

Abstrakt

Cieľom tejto diplomovej práce je navrhnúť systém monitorovania a regulácie klimatu v skupinách terárií s využitím platformy Raspberry Pi a s následným priehľadným zobrazením cez webový server. Každá skupina terárií má vlastné regulačné zariadenie, ktoré bezdrôtovo komunikuje s riadiacim počítačom Raspberry Pi. Namerané hodnoty sa ukladajú do MySQL databáze v riadiacom počítači. Na webovej stránke sú graficky zobrazené namerané hodnoty.

Klíčová slova

Raspberry Pi, Atmega32, Arduino NANO, nRF24L01, DHT22, Dallas 18B20, DS3231, Apache, MySQL, PHP, Python, regulácia, monitorovanie, teplota, vlhkosť

Abstract

The aim of this master thesis is to propose a system for monitoring and regulating the climate in groups of terrariums using the Raspberry Pi platform and subsequent transparent display through the web server. Each group of terrariums has its own control device that wirelessly communicates with the Raspberry Pi control computer. The measured values are stored in the MySQL database on the control computer. The measured values are graphically displayed on the web page.

Keywords

Raspberry Pi, Atmega32, Arduino NANO, nRF24L01, DHT22, Dallas 18B20, DS3231, Apache, MySQL, PHP, Python, regulation, monitoring, temperature, humidity

Bibliografická citace:

PAVLIŠIN, T. *Řízení a monitorování klimatu ve skupinách terárií*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 56s. Vedoucí práce: Ing. Tomáš Macho, Ph.D.

Prohlášení

„Prehlasujem, že svoju záverečnú prácu na téma Řízení a monitorování klimatu ve skupinách terárií som vypracoval samostatne pod vedením vedúceho diplomové práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedené záverečné práce ďalej prehlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomí následkov porušenia ustanovení § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení druhej časti, hlavy VI. diel 4 Trestného zákonníku č. 40/2009 Sb.

V Brně dne **15. května 2017**

.....
podpis autora

Obsah

1	Úvod	10
2	ROZBOR ZAdania	11
2.1	Regulácia	11
2.2	Komunikácia	11
2.3	Vybavenie zariadení	12
3	klimatológia	13
3.1	Teplota vzduchu	13
3.2	Vlhkosť vzduchu	13
4	realizácia	14
5	hardware.....	15
5.1	Rešerše.....	15
5.1.1	Mikrokontrolér	15
5.1.2	Bezdrôtový modul	15
5.1.3	Modul reálneho času.....	15
5.1.4	Snímače teploty a vlhkosti.....	16
5.1.5	Nadradený počítač.....	16
5.2	Zvolený hardware.....	16
5.2.1	Mikrokontrolér –ATmega32	16
5.2.2	Modul pre bezdrôtovú komunikáciu – nRF24L01+.....	17
5.2.3	Snímač teploty – DS18B20	17
5.2.4	Snímač vlhkosti – DHT22.....	18
5.2.5	Modul reálneho času – DS3231	18
5.2.6	Výkonový člen	18
5.2.7	Nadradený počítač – Raspberry Pi B+	19
5.3	Použité typy komunikácie	19
5.3.1	Zbernica SPI	19
5.3.2	Zbernica I ² C.....	20
6	Návrh Komunikácie	21
6.1	Prenos správ	21
6.2	Správy (Payload).....	21
6.2.1	Žiadosť o dáta	22
6.2.2	Odosielanie dát	22
6.2.3	Nastavenie medzí	22
6.2.4	Aktualizácia času	23
6.2.5	Čas regulácie	23
7	Mikrokontrolér – návrh dps	25
8	Zapojenie.....	26

8.1	Raspberry Pi	26
8.2	Arduino NANO	27
9	Nastavenie Raspberry pi b+	28
9.1	Nastavenie statickej IP.....	30
9.2	Inštalácia softwaru.....	30
9.2.1	Apache.....	30
9.2.2	PHP.....	31
9.2.3	Databáza MySQL.....	32
9.2.4	Python.....	37
10	WEBOvé rozhranie.....	39
10.1	Riadiaca aplikácia.....	39
10.2	Voľba technológií pre riadiacu aplikáciu	39
10.3	Štruktúra riadiacej aplikácie.....	40
10.4	Bezpečnosť	42
10.4.1	Sprístupnenie do internetu.....	43
10.4.2	Autentizácia užívateľa	44
11	Riešenie problémov	46
12	Vyhodnotenie.....	47
12.1	Možnosť ďalšieho rozvoja riadiacej aplikácie.....	47
13	Záver	48
	Literatura	49
	Seznam příloh.....	51

Seznam obrázků

Obr. 1: Návrh realizácie systému	14
Obr. 2: MCU ATmega32	16
Obr. 3: Bezdrôtový modul nRF24L01+	17
Obr. 4: Teplotný snímač DS18B20	17
Obr. 5: Snímač teploty a vlhkosti DHT22	18
Obr. 6: Modul reálneho času DS3231	18
Obr. 7: Relé - výkonový člen.....	Chyba! Záložka není definována.
Obr. 8: Raspberry Pi B+	19
Obr. 9: Enhanced ShockBurst paket ako bežný paket.....	21
Obr. 10: Správa - Žiadosť o dáta	22
Obr. 11: Správa - Prenos dát	22
Obr. 12: Správa - Nastavenie medzí	22
Obr. 13: Správa - Slave OK1.....	23
Obr. 14: Správa - Aktualizácia času	23
Obr. 15: Správa - Slave OK2.....	23
Obr. 16: Správa - Čas regulácie	24
Obr. 17: Správa - Slave OK3.....	24
Obr. 18: DPS DHT22	25
Obr. 19: Schéma zapojenia DPS DHT22.....	25
Obr. 20: Pripojenie nRF24L01 k RPi GPIO	26
Obr. 21: Win32DiskImager	28
Obr. 22: raspi-config.....	29
Obr. 23: Apache server screen	31
Obr. 24: PHPinfo	32
Obr. 25: MySQL tabuľka databáz.....	34
Obr. 26: MySQL - monitor tabuľka	35
Obr. 27: MySQL - terrarium tabuľka	35
Obr. 28: MySQL - terrarium_groups tabuľka	36
Obr. 29: MySQL - time_periods tabuľka.....	36
Obr. 30: Login WWW	46

Seznam tabulek

Tabulka 1: Zoznam súčiastok DPS DHT22	25
---	----

1 ÚVOD

Meranie teploty a vlhkosti je celkom bežná vec. Pre terárijské zvieratá je toto meranie životne dôležité, pretože nevytvorením vhodných podmienok je to pre daný druh živočícha utrpenie. Spínanie topných telies empiricky pomocou časovačov, rosenie rozprašovačom bez presnejšej spätnej väzby a časová viazanosť ma človeka prestane po nejakej dobe baviť. Preto vznikla táto práca, ktorá pre každý druh živočícha vytvorí vhodné podmienky, prehľadne ich zobrazí prípadne dá možnosť meniť potrebné hodnoty kdekoľvek a kedykoľvek s možnosťou pripojenia na internet.

V dobe písania tohto dokumentu budem riadiť dve autonómne systémy, to znamená dve skupiny terárií, kde každá skupina ovláda dve terária. V prvej skupine žije druh *Epicrates cenchria cenchria* (Hroznýšovec duhový) pôvodom z pralesa. Tento druh potrebuje vyššiu vlhkosť okolo 80-95% a denné teploty okolo 28°C. V druhej skupine sa nachádza druh *Python Regius* (Krajta kráľovská) pôvodom zo savany, takže preferuje vyššie teploty (cca 30°C) a nižšiu vlhkosť. Komunikácia medzi nadradeným systémom a podriadeným systémom je tvorená bezdrôtovo, kvôli vzdialenosti týchto terárijských skupín. Taktiež je neskôr možnosť pripojiť do tohto systému ďalšie terária bez nutnosti ťahania kabeláže a bez hardwarového obmedzenia.

Cieľom tejto práce je rozobrať možnosti realizácie systému. Zvoliť vhodný hardware. Navrhnuť a vytvoriť merací, regulačný a zobrazovací systém, ktorý meria vlhkosť a teploty v teráriách, reguluje ich podľa pokynov a vhodne zobrazuje. Nainštalovať a nakonfigurovať potrebný software.

V tomto systéme budú terária v skupinách. Každá skupina bude mať vlastné regulačné zariadenie, ktoré dokáže regulovať až 5 terárií. Namerané hodnoty sa budú zaznamenávať do databázy a prehľadne zobrazovať cez webový server.

2 ROZBOR ZADANIA

Jedná sa o distribuovaný systém, ktorý obsahuje jeden centrálny počítač, s ktorým bezdrôtovo komunikujú lokálne zariadenia monitorujúce a riadiace skupiny terárií. Meria sa teplota a vlhkosť ovzdušia. Tieto hodnoty sa zaznamenávajú do databázy v centrálnom počítači a prehľadne zobrazujú cez webový server, ktorý je primerane zabezpečený.

2.1 Regulácia

Najvhodnejší typ regulácie pre takýto systém je takzvaná "On-Off" regulácia, ktorá na základe nameraných hodnôt, zadaných limit a hysterezie spína výkonové členy. Tento typ regulácie sa dá jednoducho aplikovať na rôzne regulované sústavy, rôznych veľkostí.

Pokiaľ by sa jednalo o systém, v ktorom sú regulované sústavy(terária) rovnaké bolo by najvhodnejšie zmerať prechodovú charakteristiku danej sústavy a navrhnúť PI, PID, fuzzy PI-PD regulátor a podobne.

2.2 Komunikácia

Typ komunikácie medzi centrálnym počítačom a lokálnymi riadiacimi zariadeniami môže byť jak deterministický tak nedeterministický.

Z nedeterministických(náhodných) by bol vhodný niektorý z typov Carrier Sense Multiple Access (CSMA). Napríklad typ 1-persistent CSMA, ktorý ak je na kanály voľno tak vysielá, ak nie, tak čaká než vysielanie skončí a po uvoľnení kanálu ihneď vysielá. Vhodné pre málo zaťažené siete, čo vlastne pri dvoch skupinách terárií tento systém je. Nevýhodou je, že keď chcú vysielat' dve zariadenia, tak vždy dôjde ku kolízii.

Z deterministických napríklad Time-slotting, kde je vytvorený časový plán a stanice sú synchronizované. Výhoda je, že nedochádza ku kolíziám ale je nutné vytvoriť časový plán. To taktiež v tak malom systéme nie je problém, ale pri pridávaní skupín sa návrh stáva zložitejší.

Ja som zvolil komunikáciu Master-Slave, kde centrálny počítač je typu Master a lokálne stanice sú typu Slave. Slave zariadenia si regulujú vlastnú skupinu a zálohujú si namerané dáta a nastavenia. Master si v danom intervale obchádza Slave zariadenia. Vyšle žiadosť o dáta a počká na odpoveď. Po prijatí dát začne komunikáciu s ďalším podriadeným zariadením.

2.3 Vybavenie zariadení

Pre realizáciu riadenia a monitorovania klímy v skupinách terárií je potrebný vhodný hardware. Nadriadený počítač (Master) má vyššie nároky na software, preto je nutné zvoliť zariadenie s operačným systémom, v ktorom bude možné daný software nakonfigurovať. Hardwarové nároky sú minimálne, pokiaľ sa jedná o vstupy/výstupy(periférie). Postačuje pripojenie na internet (Ethernet) a zbernica SPI pre bezdrôtový modul.

Podriadený počítač (Slave) naopak potrebuje dostatok vstupov/výstupov pre pripojenie snímačov a výstupných výkonových členov, komunikačné zbernice SPI a I²C pre bezdrôtový modul a modul real-time hodín a dostatočnú pamäť pre dáta a program. Bolo by vhodné aby bol systém navrhnutý tak, aby spotrebovával čo najmenej energie a mohol tak byť napájaný z batérií. To vytvára nárok jak na softwarovú časť kde by zariadenie malo byť v šetrnom móde čo najdlhšie, tak na návrh hardware, kde sa musí počítať so znížením pracovnej frekvencie aby bola možnosť napájať mikrokontrolér nižším napätím a tak šetriť dobu života batérie.

3 KLIMATOLÓGIA

Klimatológia je veda o podnebiach na Zemi, o podmienkach a príčinách ich utvárania a taktiež o pôsobení klimatu na objekty činnosti človeka, na samotného človeka i na rôzne prírodné deje a naopak. Úlohou klimatológie je študovať obecné zákonitosti klimatických javov, vytváranie zemského klimatu, jeho zmeny a kolísanie s cieľom využitia poznatkov pre predpovedanie klimatu. [1]

3.1 Teplota vzduchu

Meteorologický prvok udávajúci tepelný stav ovzdušia. To znamená schopnosť vzduchu prijímať alebo predávať tepelnú energiu. Pole teploty vzduchu je považované za jedno z najzložitejších – teplota sa v horizontálnom smere často mení skokovo (napríklad pri prechode atmosférickej fronty). Hlavnou príčinou premenlivosti teploty je nehomogénnosť zemského povrchu. [2]

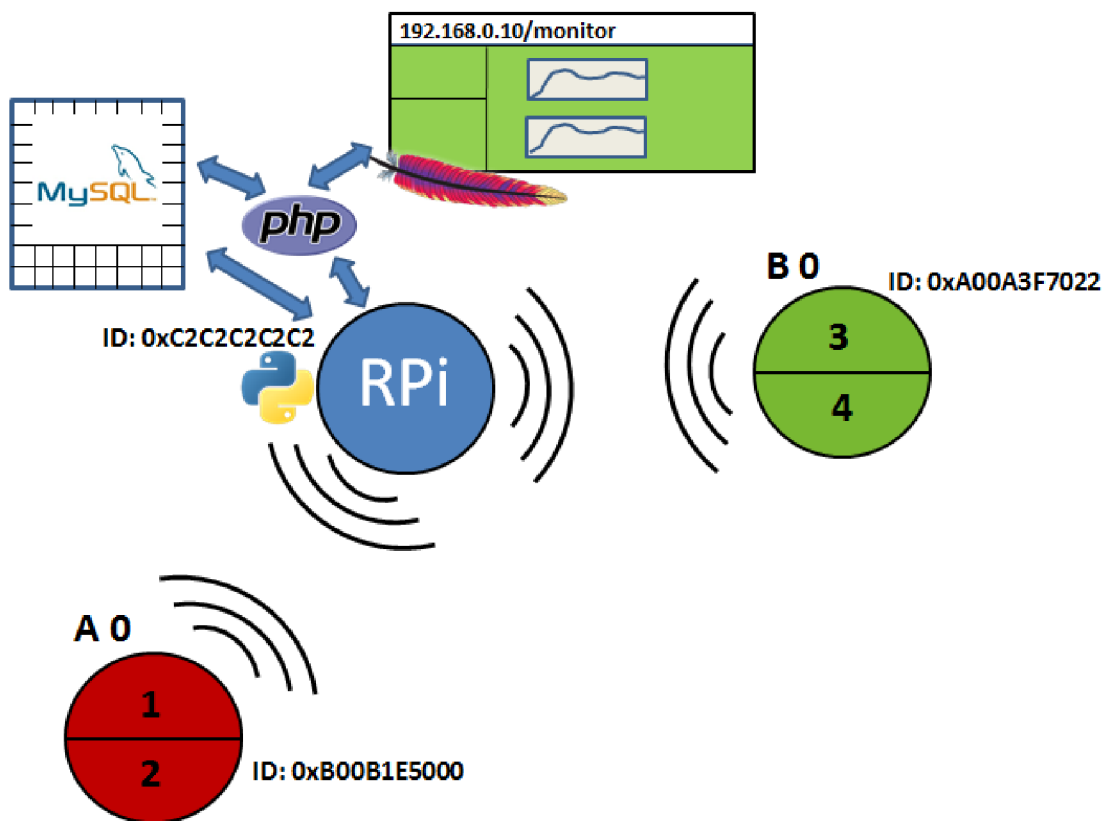
3.2 Vlhkosť vzduchu

Vzduch obsahuje vždy vo svojich spodných vrstvách vodné páry. Vodná para sa dostáva do atmosféry jednak vyparovaním vody z povrchu morí, riek, rybníkov a jazier, ale aj napríklad dýchaním rastlín. Všetky tieto javy pozorujeme u zemského povrchu, kde je vodných par v ovzduší najviac. Taktiež platí, že nad hladinami veľkých vodných plôch je vzduch vlhčí než nad súšou. Rýchlosť vyparovania vody závisí na jej teplote. Čím je teplota vyššia, tým viac molekúl vody ju opustí.

Voda sa však do svojho okolia nemôže vyparovať stále. Keď koncentrácia pár dosiahne určitú hranicu, je počet vyparených molekúl a molekúl skondenzovaných rovnaký. V tomto prípade hovoríme o mokrom vzduchu. Ak vzduch neobsahuje molekuly vody, jedná sa o suchý vzduch. Podľa množstva vodných pár určujeme vlhkosť vzduchu. [3]

4 REALIZÁCIA

Pre meranie skupín terárií som vytvoril nasledujúci návrh zobrazený na obrázku č.1



Obr. 1: Návrh realizácie systému [5]

Každá regulovaná skupina(A0, B0) môže obsahovať 1-5 terárií riadených jedným mikrokontrolérom. Mikrokontrolér je v tomto systéme ako podriadený člen – *slave*. Každý slave je vybavený senzormi pre meranie teploty a vlhkosti, bezdrôtovým modulom pre komunikáciu s nadradeným počítačom – *master*, modulom reálneho času a spínacími členmi, ktoré regulujú teplotu a vlhkosť. Zariadenia slave monitorujú teploty a vlhkosť v teráriách v určitom intervale. Porovnaním nameraných a žiadaných hodnôt spínajú výkonové členy. Na žiadosť od nadriadeného počítača, ktorý si postupne obchádza všetky skupiny, odosiela namerané hodnoty. Master tieto hodnoty ukladá do databázy a sprístupňuje ich webovému serveru Apache, na ktorom je vytvorená webová stránka. Táto stránka zobrazuje grafy teplôt a vlhkostí v určitom časovom rozsahu pre vybrané terárium, aktuálnu teplotu a vlhkosť tohto terária, polia pre nastavenie medzí, nočného režimu a iné.

5 HARDWARE

Na vytvorenie regulačného systému bol navrhnutý modul, ktorý pozostáva z mikrokontroléra, snímačov teploty a vlhkosti, real-time clock modulu, bezdrôtového modulu a výkonových prvkov.

Zobrazovaciu a záznamovú časť tvorí jednodoskový počítač Raspberry PI verzia B+.

5.1 Rešerše

Možnosti výberu vhodného hardwaru pre túto aplikáciu.

5.1.1 Mikrokontrolér

Na riešenie tejto aplikácie je možné použiť mnoho druhov vývojových kitov ako sú Arduino, LPC11C24, FRDM-KE06Z a mnoho ďalších. Táto varianta má výhodu v tom, že obsahuje vlastný programátor a nie je nutné navrhovať dosku plošných spojov. Ja som sa rozhodol vytvoriť vlastný návrh s použitím 8-bitového mikrokontroléra a pripojiť k nemu potrebné periférie. Takto totiž môžem vytvoriť vhodný mikrokontrolér pre dané zadanie.

Vhodnými kandidátmi sú mikrokontroléry PIC od firmy Microchip a ATmega od firmy Atmel. Kvôli vyššej náročnosti na počet vstupných a výstupných periférií som zvolil SMD puzdro TQFP-44. Z toho vyplývajú mikrokontroléry ATmega16, ATmega32, PIC18F45K80 a PIC18F46K80. Zo zmienovaných mikrokontrolérov som zvolil typ ATmega32 kvôli dostupnosti zdrojov a prípadnej jednoduchosti. [6][9][8][7][10]

5.1.2 Bezdrôtový modul

V kategórii RF modulov som hľadal jednoduchý a lacný modul s možnosťou obojstrannej komunikácie a dostupnosťou knižničných súborov pre implementáciu. Z týchto podmienok prišiel v úvahu jediný modul nRF24L01+.

Firma Microchip disponuje rôznymi variantmi RF modulov ako napríklad RN2483, MRF89XAM8A, MRF24J40MD a iné. Avšak cena týchto modulov je 2-5x vyššia než modul nRF24L01+. [13]**Chyba! Nenalezen zdroj odkazů.**

5.1.3 Modul reálneho času

DS1307 s DS3231 sú moduly reálneho času s dostupnými knižničnými súbormi a jednoduchou implementáciou. Pre komunikáciu s mikrokontrolérom využívajú zbernicu I²C. Rozdiel medzi týmito zariadeniami je presnosť sledovania času.

U modulu DS1307 môže teplota okolia ovplyvniť frekvenciu oscilátora,

ktorý riadi vnútorný počítač. To znamená, že sa hodiny môžu posunúť asi o 5 minút v mesiaci.

DS3231 obsahuje vnútorný oscilátor, ktorý neovplyvňujú vonkajšie faktory. Preto je výrazne presnejší. Od spoločnosti Microchip som našiel jediný modul reálneho času MCP7941X PICtail Plus Daughter Board (I²C). Tento modul sa pre túto aplikáciu nehodí, pretože je navrhnutý pre vývojové kity. Z týchto modulov som zvolil modul DS3231 z dôvodu presnosti merania času. [17]**Chyba! Nenalezen zdroj odkazů.Chyba! Nenalezen zdroj odkazů.**

5.1.4 Snímače teploty a vlhkosti

Pretože pre meranie teploty v teráriách nie je potrebná presnosť na desatiny stupňov, zameral som sa na digitálne snímače, ktoré sú jednoduchšie na implementáciu. Analógové snímače sú vhodné pre meranie teploty v inkubátoroch, kde sa využije ich presnosť. To ale nie je predmetom tejto práce. Ďalšou podmienkou je OneWire prenos dát namiesto komunikácie cez I²C z dôvodu vyššieho počtu snímačov v skupine terárií. Posledná podmienka je vode odolné puzdro, vzhľadom k vlhkosti v teráriách.

Podľa týchto podmienok som zvolil snímač na meranie teploty DS18B20 s vode odolným puzdrom. Pre meranie vlhkosti som zvolil snímač DHT22, ktorý je presnejší než snímač DHT11. Tieto snímače majú taktiež výhodu v dostupných knižničných súboroch. [16][11]

5.1.5 Nadradený počítač

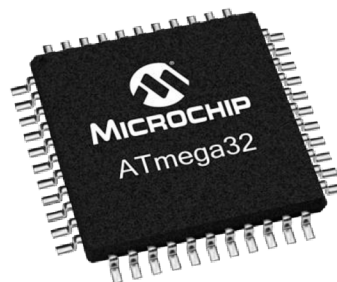
Ako nadradený počítač som zvolil Raspberry Pi variantu B+. Na trhu sa vyskytuje viacero podobných počítačov ako napríklad BeagleBone Black, Banana Pi, ODR0ID-C1+, NanoPC-T1 a iné. Pre Raspberry Pi som sa rozhodol kvôli cene, ktorá bola v danej dobe nízka a množstvu zdokumentovaných projektov. [19]

5.2 Zvolený hardware

Použitý hardware pre aplikáciu monitorovania a regulácie terárií.

5.2.1 Mikrokontrolér - ATmega32

ATmega32 je vysoko výkonný 8 bitový AVR mikrokontrolér s architektúrou RISC. Je vybavený 32kB programovej pamäte Flash, 2KB pamäte SRAM, 1024B pamäte EEPROM. Rýchlosť CPU je 16 MIPS. Periférie pre digitálnu komunikáciu: 1x UART, 1x SPI, 1x I²C. Časovače: 2x 8-bit, 1x 16-bit.

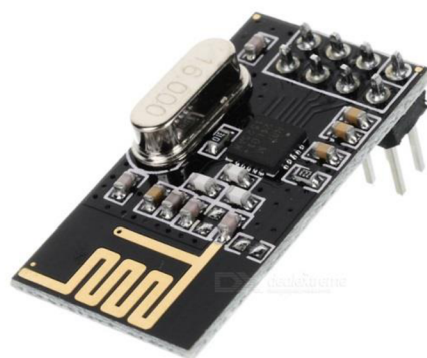


Obr. 2: MCU ATmega32[7]

Napájanie 2,7-5,5V. Tento čip obsahuje mnoho ďalších vlastností, ktoré sú popísané na stránke[10]. Tento mikrokontrolér som vybral z dôvodu vhodných periférií, vyváženej spotreby a hlavne dostatočnej pamäte a počtu pinov (SMD puzdro TQFP-44).

5.2.2 Modul pre bezdrôtovú komunikáciu - nRF24L01+

Bezdrôtový modul nRF24L01 je pripojený k Raspberry Pi a ku každej skupine terárií pomocou dátovej zbernice SPI. Modul pracuje v nelicencovanom pásme 2.4GHz s nastaviteľnou prenosovou rýchlosťou (250kbit/s, 1Mbit/s a 2Mbit/s). Každý modul obsahuje transmitter i receiver a jednotlivé moduly je možné adresovať. Adresovaním získame možnosť pridávať zariadenia do systému. Modul disponuje nízkym prúdovým odberom. Spoľahlivý prenos vo voľnom priestranstve je možný na vzdialenosť až 100m. V reálnych podmienkach budovy je vzdialenosť znížená na 30m.[14]



Obr. 3: Bezdrôtový modul nRF24L01+[14]

5.2.3 Snímač teploty - DS18B20

Tento digitálny teplotný snímač z kategórie OneWire využíva na komunikáciu jeden dátový vodič, ktorý je možné použiť aj na napájanie senzora. Má programovateľné rozlíšenie v rozsahu 9 - 12 bitov a možnosť pripojenia viacerých senzorov na jeden pin procesora. Teplotný rozsah snímača je -55°C - 125°C s napájaním od 3V do 5.5V.

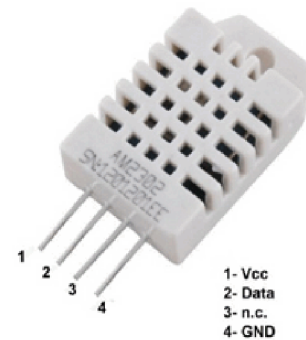
Tento snímač používam na meranie maximálnej teploty v teráriách (umiestnený na najteplejšom mieste). Vo svojej aplikácii používam vode odolnú variantu.[16]



Obr. 4: Teplotný snímač DS18B20[16]

5.2.4 Snímač vlhkosti – DHT22

Digitálny snímač DHT22 využíva na meranie vlhkosti kapacitný snímač a termistor na meranie teploty okolitého vzduchu. Namerané hodnoty je možné snímať z dátového pinu. Je jednoduchý na používanie ale je nutné dať pozor na časovanie zberu dát, ktoré je možné najskôr po dvoch sekundách. Každý senzor je nutné pripojiť k vlastnému pinu procesora. Rozsah meranej vlhkosti je 0-100% s presnosťou $\pm 2-5\%$ a teploty $-40-80^{\circ}\text{C}$ s presnosťou $<\pm 0,5^{\circ}\text{C}$. Možné napájať 3,3-6V.

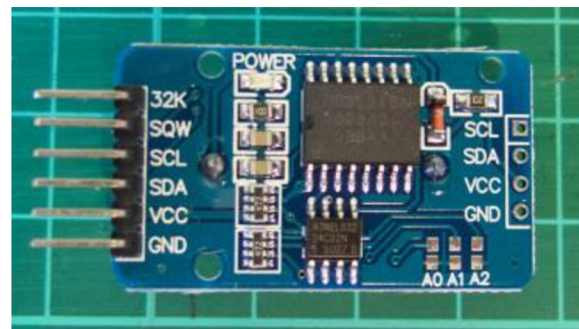


Obr. 5: Snímač teploty a vlhkosti DHT22[11]

Tento snímač využívam na meranie vlhkosti a minimálnej teploty v teráriu (umiestnený na najchladnejšom mieste). [11]

5.2.5 Modul reálneho času – DS3231

DS3231 je integrovaný obvod real-time hodín (RTC), ktorý je extrémne presný, vďaka teplotnej kompenzácii kryštálového oscilátora a kryštálu. Tým nepôsobí zmena okolitej teploty na funkčnosť obvodu. Tento obvod komunikuje pomocou I²C zbernice a má malú spotrebu.



Obr. 6: Modul reálneho času DS3231[17]

RTC obvod používam pre presný záznam času a dátumu merania. Vďaka tomuto záznamu nevznikajú duplicitné záznamy a zaznamenávam presný čas merania pre zápis hodnôt do grafov. [17]Chyba! Nenalezen zdroj odkazů.

5.2.6 Výkonový člen

Momentálne je ako výkonový člen zvolené relé. Je to z dôvodu práce s vysokým napätím, ktoré je potrebné na spínanie topných žiaroviek a čerpadla na vytápanie respektíve rosenie. Kvôli bezpečnosti je nutné zapojiť minimálne 2 relé sériovo za sebou pre prípad skratu a následného spojenia kontaktov. Táto výkonová časť sa nachádza mimo dosky plošných spojov mikrokontroléra. V pláne je návrh tejto časti z polovodičových súčiastok.

5.2.7 Nadradený počítač – Raspberry Pi B+

Na funkciu nadradeného systému som zvolil jednodoskový počítač Raspberry Pi 1 Model B+(RPI). Raspberry Pi sa vyrába v niekoľkých verziách. Prvá verzia (Raspberry Pi 1 Model B) bola vydaná v roku 2012. Nasledovala jednoduchšia a lacnejšia verzia Model A. V roku 2014 bol vyvinutý model B+ so zväčšenou pamäťou, pridanými USB



Obr. 7: Raspberry Pi B+[4]

portami a so zlepšeným dizajnom vo veľkosti kreditnej karty. Ďalšia

generácia Raspberry Pi 2 sa vyznačuje pridanou pamäťou RAM a tretia generácia Raspberry Pi 3 vyniká vlastnosťami WiFi, Bluetooth a USB boot. Existujú aj verzie v menšej veľkosti s redukovaný počtom vstupov/výstupov (I/O) ako Raspberry Pi Zero.

Hlavnou výhodou Raspberry Pi je, že oproti klasickému stolnému počítaču alebo notebooku obsahuje špecializované zbernice pre pripojenie hardwaru. Vďaka tomu je z RPI nástroj na riadenie a monitorovanie. Raspberry Pi obsahuje tri základné rozhrania:

- **GPIO – poskytuje vstupno/výstupné piny, špeciálne rozhranie UART, zbernice I2C, SPI. Vďaka GPIO môžeme k RPI pripojiť veľkú radu senzorov, expanderov zbernic a prevodníkov.**
- **CSI camera interface - slúži pre pripojenie špecializovanej kamery cez rozhranie CSI**
- **DSI display interface - slúži k pripojeniu externého LCD displeja[4]**

5.3 Použité typy komunikácie

V tomto projekte sú využité dva druhy komunikácie. Komunikácia pomocou zbernice SPI pre prenos dát s bezdrôtovým modulom nRF24L01+ a komunikácia pomocou zbernice I²C, ktorú používa modul reálneho času DS3231.

5.3.1 Zbernica SPI

Zbernica SPI je sériová externá zbernica slúžiaca na komunikáciu (vzdialené pripojenie) dvoch a viacerých komunikujúcich uzlov. Obvykle jeden z týchto uzlov pracuje v režime master a ostatné uzly v režime slave. Master obsahuje generátor hodinového signálu, ktorý sa rozvádza do všetkých ostatných uzlov zbernice, čo

umožňuje synchronný a obojsmerný prenos dát. Hodinový signál sa prenáša vodičom SCK. Okrem hodinového signálu obsahuje dvojicu vodičov pre obojsmerný prenos dát. Tieto vodiče sú označené ako MISO (master in, slave out) a MOSI (master out, slave in). Posledný vodič tejto zbernice je označený ako CS (chip select), ktorý sa používa pre výber aktívneho uzlu. [15]

5.3.2 Zbernica I²C

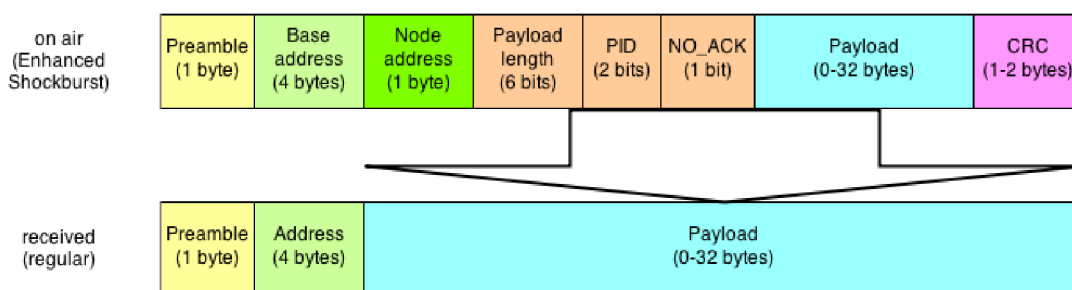
Táto zbernica využíva pre komunikáciu len dva vodiče, sériový hodinový (SCL) a sériový dátový (SDA). To znamená, že sa jedná o synchronizovaný prenos dát len jedným smerom. V danom okamžiku existuje len jedno vysielajúce zariadenie a ľubovoľný počet zariadení, ktoré prijímajú dáta. Každé zariadenie, pripojené k tejto zbernici je softwarovo adresovateľné jednou unikátnou adresou pre celý systém. V každej chvíli existuje jednoduchý vzťah master/slave. Obvod sa stáva ako master, ak zahajuje prenos dát po zbernici a generuje hodinový signál. V tomto čase sú ostatné zariadenia chápané ako slave. Master môže operovať ako vysielateľ či prijímač. I²C je zbernica typu multi-master so zabudovanou detekciou kolízií. [18]

6 NÁVRH KOMUNIKÁCIE

Návrh komunikácie zahrňuje popis prenášaného paketu a typy správ.

6.1 Prenos správ

Prenos správ funguje nasledovne. Dátový paket sa prenáša medzi zdrojom a cieľom s nastavenými TX(odosielanie) módom pre zdroj a RX(prijímanie) módom pre cieľ. Po odoslaní paketu sa zdroju automaticky nastaví RX mód a čaká na prijatie ACK(potvrdenie) paketu. Pri prijatí paketu cieľom, odošle potvrdzujúci paket zdroju. Ak zdroj neprijíme potvrdzovací paket, automaticky sa prepošle originálny dátový paket po prečkaní nastaveného oneskorenia.



Obr. 8: Enhanced ShockBurst paket ako bežný paket

Každá správa začína preambulou, ktorá slúži na synchronizáciu demodulátora. Nasleduje adresa cieľového zariadenia, fixná veľkosť prenášanej správy, PID – Packet ID, ktorý informuje či ide o nový paket alebo o preposlaný, NO_ACK – nastavenie automatického potvrdenia, prenášaná správa a vhodné CRC(Cyclic Redundancy Check) pre kontrolu prijatého paketu. CRC sa počíta z celkovej správy bez preambule a samotného CRC. **Chyba! Nenalezen zdroj odkazů.**

6.2 Správy (Payload)

Typy správ použitých na komunikáciu medzi nadriadeným počítačom a podriadenými zariadeniami. Veľkosť správy môže byť v rozsahu 0-32 bytov. Každé pole znázorňuje jeden byte.

Každá správa obsahuje Device ID, pomocou ktorého je možné pristupovať k slave zariadeniu. Device ID má veľkosť 2 Byty. Prvý byte označuje slave zariadenie, s ktorým chceme komunikovať. Pomocou druhého bytu môžeme pristupovať k zariadeniu ako k celku alebo k jednotlivým teráriám.

Tretí byte správy označuje typ prenášanej správy:

- 'Q' – žiadosť o dáta (reQuest)
- 'L' – nastavenie limít pre reguláciu(Limits)

- 'C' – aktualizácia času (Clock)
- 'T' – nastavenie doby regulácie (night Time)

6.2.1 Žiadosť o dáta

Týmto typom správy master žiada o namerané hodnoty podriadených zariadení. Správa obsahuje ID uzlu(A), voľbu prístupu(0) a označenie typu správy (Q).

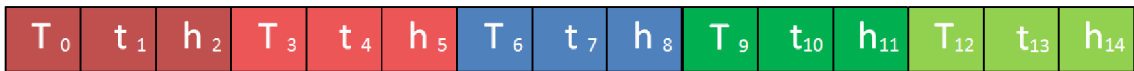


Obr. 9: Správa - Žiadosť o dáta

6.2.2 Odosielanie dát

Slave odpovedá správou, v ktorej sú namerané hodnoty radené podľa terárií.

V prvom byte je hodnota teploty zo snímača DS18B20, ktorý meria maximálnu teplotu(T). Nasledujú hodnoty zo snímača DHT22. Najprv hodnota minimálnej teploty(t) a potom vlhkosti(h). Takto sa za sebou pridávajú hodnoty z ostatných terárií v skupine.



Obr. 10: Správa - Prenos dát

6.2.3 Nastavenie medzí

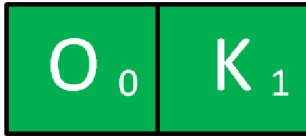
Touto správou master odosiela žiadané medze teplôt.

Správa obsahuje hodnoty medzí pre jednotlivé terária v skupine. Prvý byte obsahuje ID uzlu. Nasleduje ID terária, pre ktoré sa nastavujú limity a samotné hodnoty limít v tvare maximálna, minimálna teplota a vlhkosť.



Obr. 11: Správa - Nastavenie medzí

Slave potvrdzuje prijatie správy:



Obr. 12: Správa - Slave OK1

6.2.4 Aktualizácia času

Táto správa slúži na nastavenie aktuálneho času a dátumu.

Keďže Raspberry Pi je pripojené k internetu má možnosť aktualizovať čas jednotlivých slave zariadení. Po prijatí tejto správy si podriadené zariadenie nastaví čas v real-time clock module DS3231. Správa je odoslaná v danom čase.



Obr. 13: Správa - Aktualizácia času

S – sekundy

M – minúty

H – hodiny

W – deň v týždni (1-7) 1 = Nedeľa

d – deň v mesiaci

m – mesiac

y – rok

Slave potvrdzuje prijatie správy:



Obr. 14: Správa - Slave OK2

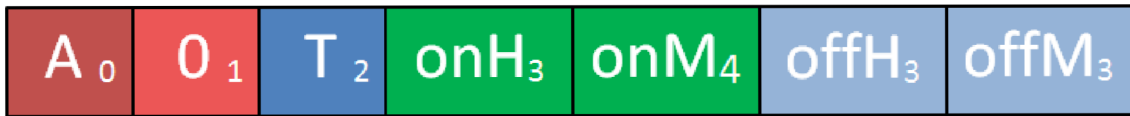
6.2.5 Čas regulácie

Funkciou tejto správy je nastavenie času od kedy do kedy je spustená regulácia v teráriách.

Táto funkcia je vhodná pre nastavenie časového okna, kde je požiadavka len na monitoring bez regulácie. Napríklad cez noc nechať poklesnúť teplotu na izbovú. Taktiež v priestoroch kde sa spí a iné.

Tieto hodnoty sa nastavujú pre celú skupinu, nie pre jednotlivé terária.

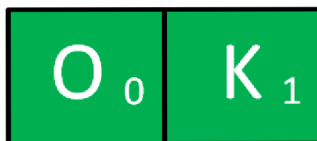
Master odosiela časy správou:



Obr. 15: Správa - Čas regulácie

kde onH, onM je čas zapnutia a offH, offM je čas vypnutia v tvare hodiny, minúty.

Slave potvrdzuje prijatie správy:



Obr. 16: Správa - Slave OK3

7 MIKROKONTROLÉR – NÁVRH DPS

Návrh dosky plošných spojov som navrhoval pomocou softwaru Eagle. Doska má rozmery 38,6x53,4mm a je navrhnutá na dvoch vrstvách. Pri návrhu bolo nutné pridať niektoré knižnice. Snahou bolo vytvoriť riadiace zariadenie s čo možno najmenšími rozmermi. V pláne je aj návrh vhodného krytu, ktorý bude vytvorený pomocou 3D tlačiarne.

Pre napájanie som použil miniUSB konektor, ktorý je bežne používaný a odolnejší než microUSB konektor. Taktiež sú k dispozícii zdiery pre možnosť napájania z batérií. Doska obsahuje mikrokontrolér ATmega32, napätový regulátor na 3,3V, 16MHz rezonátor, pasívne súčiastky, konektor pre pripojenie bezdrôtového modulu a dostatočné množstvo konektorov pre napájanie pripojených zariadení od firmy Würth. Pre vstupy typ wire to board a pre výstupy typ board to board pre možnosť pripojenia dosky s výkonovými členmi. Medzi výstupy je zaradený aj konektor pre prenos napájania (5V, 3,3V, GND).

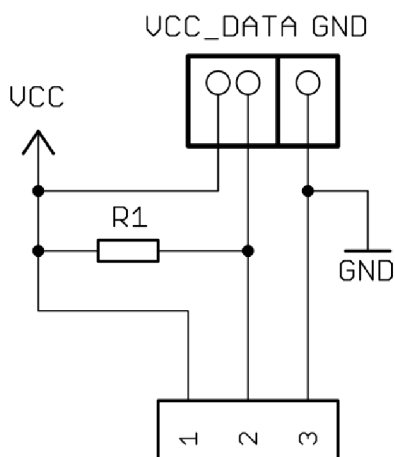
Na doske výstupu by sa mal nachádzať DC/DC menič, ktorý posluží ako galvanické oddelenie napájania a optočleny pre galvanické oddelenie dátových pinov od mikrokontroléra.

Spínacie relé sú mimo DPS kvôli jednoduchosti a možnosti meniť túto časť. Schéma zapojenia, predlohy dosky, osadzovací plán a zoznam súčiastok sú uvedené v [prílohe](#).

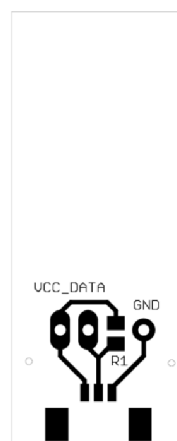
Vytvoril som návrh DPS aj pre snímač vlhkosti a teploty DHT22, kvôli možnosti vyrobiť kryt proti zásahu vodou pri manuálnom rosení.

Tabulka 1: Zoznam súčiastok DPS DHT22

Názov	Popis	
DHT22	1-wire TTL 5V	
R1	4k7	SMD 0805
Konektor	653103131822	WR-WTB 1,25mm



Obr. 18: Schéma zapojenia DPS DHT22



Obr. 17: DPS DHT22

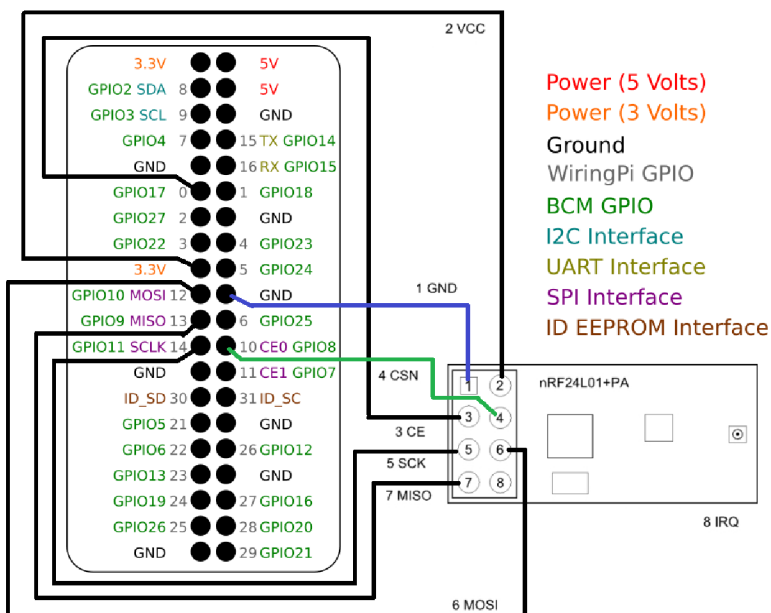
8 ZAPOJENIE

Pripojenie modulov k nadriadenému a podriadenému systému.

8.1 Raspberry Pi

Nadradený systém má na svoje GPIO rozhranie pripojený len bezdrôtový modul nRF24L01+, napájaný napätím 3,3V. Tento systém má oproti podriadenému systému bohatú softwarovú výbavu. Okrem iného obsahuje Apache server, MySQL databázu a knižnice pre komunikáciu s PHP5 a Python, ako aj tento software.

Pre pripojenie bezdrôtového modulu pripojíme SPI piny modulu s RPi. IRQ pin vynecháme. CE pripojíme na pin 17 (GPIO0), CSN pin 24 (GPIO17)



Obr. 19: Pripojenie nRF24L01 k RPi GPIO[14][4]

8.2 Arduino NANO

Mikrokontroléry Arduino NANO v tomto systéme pracujú ako podriadené (slave). To znamená, že sú ovládané nadradeným systémom (RPI). Ich úlohou je merať teploty a vlhkosť v teráriách, zaznamenávať ich po určitú dobu a vykonávať akcie pre splnenie podmienok.

Arduino ako samostatný systém pracuje nasledovne. Pri spustení si nahrá potrebné knižnice. Potom definuje konštanty a deklaruje globálne premenné. Nasleduje *void setup()* v ktorom nastaví a inicializuje potrebné komunikácie a piny pripojených zariadení. V ďalšom kroku si zistí počet OneWire zariadení, ich adresy a nastaví im rozlíšenie. Pri prvom spustení je nutné nastaviť čas pre RTC. Na to stačí zadať do pripravenej funkcie aktuálne hodnoty a nahráť sketch. Treba počítať s určitým časovým meškaním cca 12 sekúnd než sa sketch nahrá do dosky. Týmto končí časť nastavení a prechádza sa k hlavnému programu.

Hlavný program je založený na prerušeníach. Jedno prerušenie je od 16 bitového časovača TIMER 1 a druhé od modulu pre bezdrôtovú komunikáciu pripojeného na SPI zbernicu a externé prerušenie INTO.

Časovač je nastavený tak , aby dokázal vyvolať prerušenie každú sekundu. Vložením čítača do rutiny, dokážem nastaviť čas, kedy sa má rutina vykonať.

V hlavnej smyčke *while loop()* len čakám na prerušenie, ktoré vyvolá správa od nadriadeného počítača. Správa môže obsahovať dáta alebo žiadosť o dáta. Po prijatí správy sa začne vykonávať rutina prerušenia, kde sa vyhodnotí o aký typ správy šlo.

Ak šlo o žiadosť o dáta, tak sa do bufra načítajú namerané hodnoty z posledného merania a odošlú sa.

Ak sa jednalo o správu nastavenia limít alebo nočného režimu, tak sa načítajú aktuálne hodnoty z pamäte pre porovnanie. Ak sa hodnoty nezhodujú, tak sa do pamäte EEPROM zapíšu.

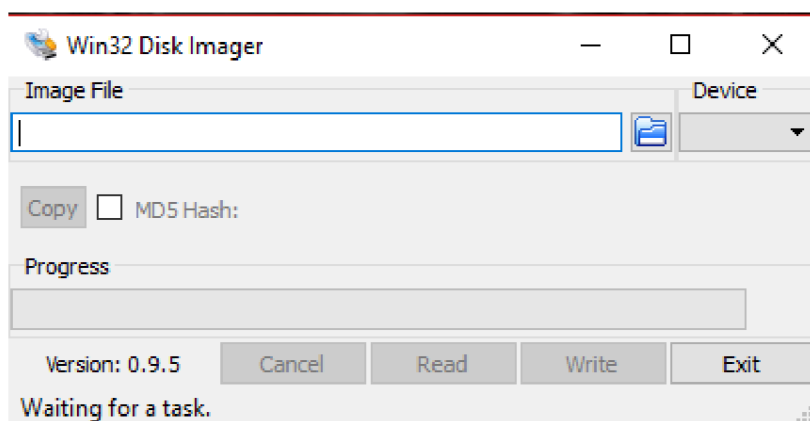
Ak šlo o správu pre nastavenie aktuálneho času, tak sa správa spracuje a v najbližšom prerušení sa vykoná daná funkcia.

V rutine prerušenia od časovača sa v danom intervale merajú hodnoty teplôt a vlhkosti. Tieto hodnoty sa ukladajú do logera. Následne sa hodnoty porovnávajú a rieši sa jednoduchá ON – OFF regulácia s ohľadom na zadanú hysteréziu a *nightTime*.

9 NASTAVENIE RASPBERRY PI B+

V mojom prípade som do Raspberry Pi (RPi) nainštaloval operačný systém Raspbian Jessie Lite. Zvolil som Lite verziu, ktorá nemá nainštalované grafické rozhranie. *Chyba! Nenalezen zdroj odkazů.*

Inštalácia operačného systému je jednoduchá. Stačí si stiahnuť súbor .img a pomocou programu Win32DiskImager vybrať správne zariadenie (microSD kartu class 10), zadať cestu k .img súboru a zapísať pomocou *Write*.



Obr. 20: Win32DiskImager

Pre komunikáciu s RPi používam SSH, (Secure Shell) ktorý umožňuje bezpečnú komunikáciu medzi dvoma počítačmi cez príkazový riadok. Novšie verzie tohto operačného systému majú prednastavené blokovanie ssh. Pridaním súboru *ssh.txt* do */boot* problém vyrieši. Pred samotným spustením musíme skontrolovať nastavenie routera.

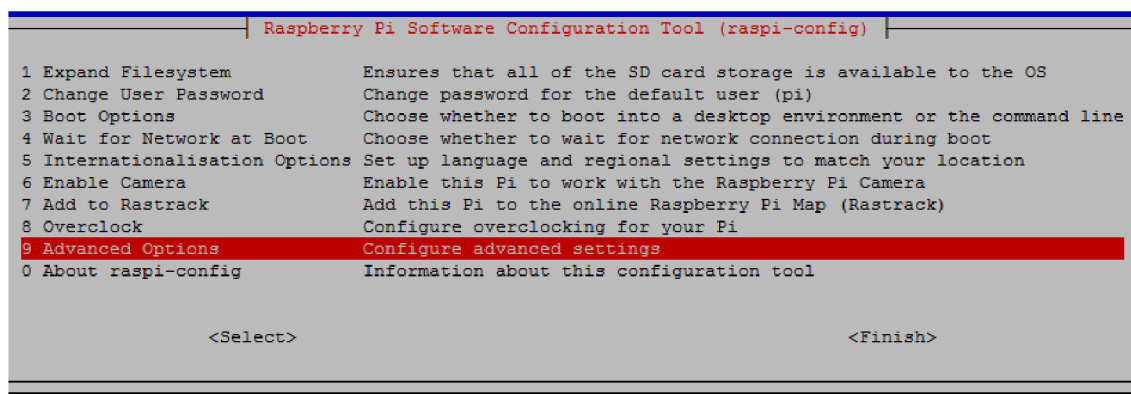
Po zapnutí RPi by sa mal v IP tabuľke zobrazit' Host Name -> raspberrypi s priradenou IP a MAC adresou. Vhodné je pomocou MAC adresy pridelit' RPi statickú IP adresu, ktorá sa vždy priradí danej MAC adrese. Túto možnosť ale nemá každý. V mojom prípade je to adresa 192.168.0.10, ktorú budem používať pre prístup k RPi.

Pre pripojenie z operačného systému Windows je potrebné stiahnuť si program. Po spustení programu zadáme do Host Name IP adresu RPi, port 22 a typ pripojenia nastavíme na SSH. V štandardnom nastavení je prihlasovacie meno nastavené na „pi“ a heslo na „raspberrypi“. Po úspešnom prihasení sa objaví príkazový riadok. Pri použití operačného systému Linux stačí do príkazového riadku napísať príkaz `ssh pi@192.168.0.10` (username@hostname), po vyzvaní zadať heslo. Pri prvom prihlásení je

vhodné spustiť update a upgrade systému a nastaviť základné nastavenia. Príkazom:

sudo apt-get update
sudo apt-get upgrade
sudo raspi-config

updatujeme systém,
upgradujeme systém,
zobrazí tabuľku základných nastavení.



Obr. 21: raspi-config

1. **Expand Filesystem**: zaistí, že celá veľkosť SD karty je k dispozícii na OS
2. **Change User Password**: zmena hesla prednastaveného užívateľa (pi)
3. **Enable Boot to Desktop-Scratch**: výber, či chceme spúšťať z prostredia desktopu, Scratch alebo z príkazového riadku
4. **International Options**: nastavenie jazyka a miestnych nastavení, pre našu pozíciu
5. **Enable Camera**: povolenie tohto Pi prácu s Raspberry Pi kamerou
6. **Add to Rastrack**: pridať toto Pi do on-line Pi Mapa Paspberry (Rastrack)
7. **Overlock**: konfigurácia pretaktovania pre naše Pi
8. **Advanced Options**: konfigurácia pokročilých nastavení
9. **About raspi-config**: informácie o tomto konfiguračnom nástroji

Tu je potrebné v Advanced options povoliť SPI komunikáciu a nastaviť Memory Split na rozloženie pamäte RAM. Toto nastavenie určuje koľko operačnej pamäte bude využívať hlavný procesor a koľko grafický. Vzhľadom k tomu, že grafický procesor nepotrebujem a celková dostupná pamäť nie je tak ohromujúca, tak zvolíme minimum pamäte pre GPU (16). Táto voľba zásadne ovplyvňuje výkon serveru. Ďalej je vhodné nastaviť správne časové pásmo, prípadne zmeniť heslo. Nastavenie hesla je v podstate zbytočné pretože si z bezpečnostných dôvodov vytvorím nového užívateľa a užívateľa pi vymažem. O tom však v časti o bezpečnosti.

9.1 Nastavenie statickej IP

Pokiaľ nie je možné nastaviť routru MAC adresu nášho Raspberry Pi, existuje ďalšia možnosť ako doň pristupovať cez nami zvolenú IP adresu.

Príkazom:

```
sudo nano /etc/dhcpd.conf
```

môžeme editovať konfiguračný súbor. Na konci súboru pridáme nasledujúce riadky.

Môžeme nastaviť statickú IP adresu na ethernetové(eth0) pripojenie, bezdrôtové pripojenie(wlan0) alebo oboje. Pre ethernet nastavíme:
interface eth0

```
static ip_address=192.168.0.10/24  
static routers=192.168.0.1  
static domain_name_servers=192.168.0.1
```

- interfaces – definuje, aký typ siete konfigurujeme
- static ip_address – IP adresa, ktorú chceme nastaviť nášmu RPi(nezabudnúť masku /24)
- static routers – IP adresa gateway (pravdepodobne IP adresa routru)
- static domain_name_servers – IP adresa našej DNS (pravdepodobne IP adresa routru). Je možné pridať ďalšie adresy, stačí oddeliť medzerov.

Ctrl +x, potvrdiť “Y” a Enter pre uloženie a ukončenie editácie. Pre bezdrôtovú sieť stačí namiesto “eth0” zadať “wlan0”. Pre aktiváciu zmien je nutné reštartovať RPi.

9.2 Inštalácia softwaru

Pre tento projekt je nutné nainštalovať potrebný software. Každý software je zadarmo a open-source (voľné zdroje). Pred inštaláciou je vhodné aktualizovať RPi príkazom *sudo apt-get update*.

9.2.1 Apache

Apache pôsobí ako náš webový server. Jeho hlavnou úlohou je nielen spracovávanie požiadavkou, ktoré užívatelia odosielajú prostredníctvom svojich webových prehliadačov, ale taktiež zobrazovanie výsledkov

pripravených pomocou kódu umiestneného vo vyžiadaných súboroch. Apache je veľmi výkonný nástroj a môže splniť prakticky všetky prania ktoré správca webu môže mať. Základné funkcie:

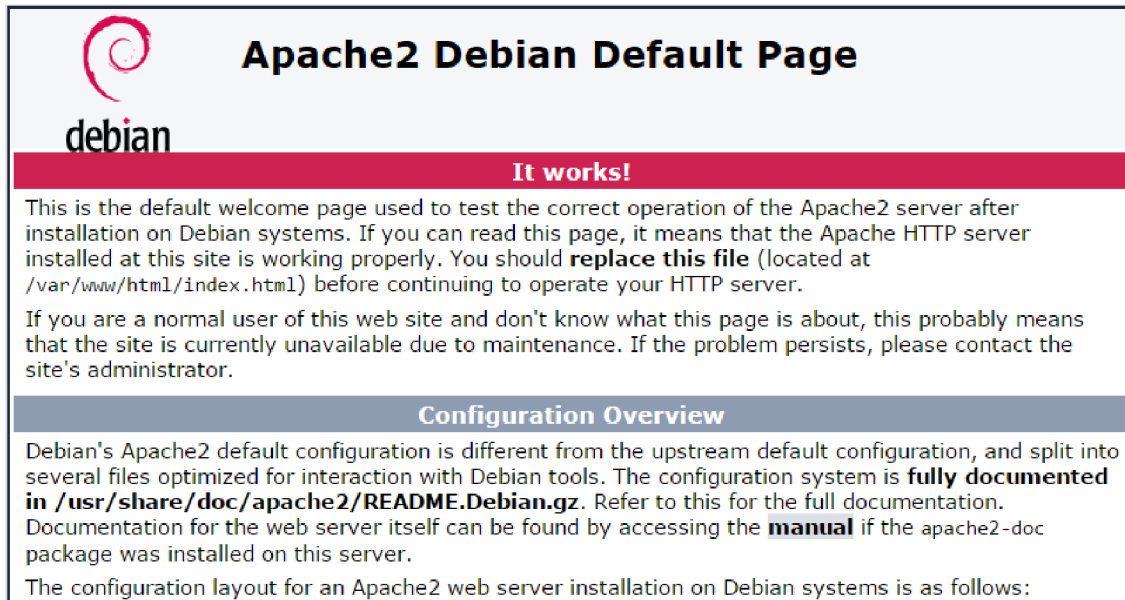
- Stránky chránené pomocou hesiel rôznych užívateľov,
- prispôbené stránky oznamujúcich chyby,
- nastavenie rôznych IP adries mapovaných na rovnaký webový server,
- zobrazenie kódu v rôznych úrovniach HTML, tak i schopnosť určiť, na ktorej úrovni môže prehliadač prijať ponúkaný obsah,
- prezívania adries URL alebo prepisovanie bez pevného limitu,
- atd.

[5]

Samotná inštalácia sa spustí príkazom

```
sudo apt-get install apache2
```

Test funkčnosti webového serveru overíme zadaním *localhost* (prípadne 127.0.0.1) alebo IP adresy RPi 192.168.0.10 do webového prehliadača. Pokiaľ je všetko v poriadku tak sa objaví prednastavená stránka apache s konfiguračným prehľadom a textom „It works!“. Cesta k tomuto súboru je */var/www/html/index.html*. V priečinku *html* je možné vytvárať ďalšie priečinky a súbory, ktoré je možné spúšťať v internetovom prehliadači doplnením cesty k súboru napríklad 192.168.10/monitor.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at */var/www/html/index.html*) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in */usr/share/doc/apache2/README.Debian.gz***. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the *apache2-doc* package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

Obr. 22: Apache server screen

9.2.2 PHP

PHP je jazyk pre skriptovanie na strane serveru. Umožňuje dynamizáciu webového serveru. PHP je skratkou „PHP: Hypertext Preprocesor“. Jeho flexibilita a vcelku rýchle osvojenie programovania z neho robí jeden

z najobľúbenejších skriptovacích jazykov. PHP sprostredkováva komunikáciu medzi serverom a databázou a klientom. Vo svojom projekte používam verziu PHP5.

[5]

Inštalácia PHP a knižníc pre komunikáciu s apache sa spustí príkazom `sudo apt-get install php5 libapache-mod-php5`

Funkčnosť nainštalovaného php môžeme otestovať napríklad vytvorením nového priečinka v html .

`cd /var/www/html`

pre premiestnenie sa do html priečinka

`mkdir ./test`

vytvorenie priečinka v aktuálnom priečinku

`sudo nano index.php`

vytvorenie PHP súboru

Do tohto súboru napíšeme php kód

```
<?php phpinfo(); ?>
```

pomocou ktorého zistíme všetky informácie o nainštalovanom PHP pridaním `/test` za IP adresu vo webovom prehliadači.

PHP Version 5.6.19-0+deb8u1	
System	Linux raspberrypi 4.1.21+ #872 Wed Apr 6 17:27:13 BST 2016 armv6l
Build Date	Apr 2 2016 19:47:28
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-readline.ini

Obr. 23: PHPinfo

Pri vzniku nejakých problémov s PHP alebo pri úmysle pridaní zakázaných funkcií ako napríklad zobrazovanie chýb je nutné editovať súbor `php.ini`. Tento súbor nájdeme v `/etc/php5/apache2/`

9.2.3 Databáza MySQL

Pre ukladanie nameraných hodnôt, časov meraní a ďalších potrebných dát využívam databázový systém MySQL.

MySQL umožňuje technológiám PHP a Apache spolupracovať na sprístupnení a zobrazení dát vo formáte čitateľnom v internetových prehliadačoch. Je to server, ktorý spracováva dopyty (otázky) v SQL jazyku (Structured Query Language) navrhnutý pre spracovanie veľkého množstva veľmi zložitých dopytov. Ide o relačný databázový systém, ktorý umožňuje spojovanie veľa rôznych tabuliek, vďaka čomu ponúka maximálnu efektivitu a rýchlosť. Zoznam niekoľkých základných a zároveň najobľúbenejších funkcií:

- Početné typy stĺpcov pokrývajúcich prakticky všetky dátové typy,
- skupiny funkcií pre matematické výpočty a triedenie,
- príkazy, ktoré umožňujú jednoduchý prístup správcov k informáciám o databázach,
- možnosť spúšťania na nezávislej hositeľskej platforme,
- systém overovania užívateľov na základe hesla,
- možnosť využitia väčšieho počtu procesorov prostredníctvom vlákien jadra,
- možnosť definovania 32 indexov pre každú tabuľku,
- a iné.

[5]

Okrem samostatnej inštalácie MySQL je potrebné nainštalovať knižnice pre komunikáciu s php5 a programovacím jazykom Python.

Takže:

<code>sudo apt-get install mysql-server</code>	pre inštaláciu MySQL serveru
<code>sudo apt-get install php5-mysql</code>	pre komunikáciu s php5
<code>sudo apt-get install python-mysqldb</code>	pre komunikáciu s Python.

Pri prvom spustení MySQL sa prihlásime ako *root* príkazom

`mysql -u root -p`

následne zadáme heslo pre užívateľa *root*.

Pre ukončenie MySQL zadáme príkaz *quit*.

Najprv vytvoríme nového užívateľa zadaním:

```
CREATE USER 'tempHUM'@'localhost' IDENTIFIED BY 'passwd';
```

vytvoríme užívateľa "tempHUM" s heslom "passwd".

Pomocou:

<code>CREATE DATABASE tempHUM;</code>	vytvoríme databázu tempHUM,
<code>SHOW DATABASES;</code>	

```
mysql> SHOW DATABASES;
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema |
| tempHUM           |
+-----+
```

Obr. 24: MySQL tabuľka databáz

USE tempHUM; pre používanie zvolenej databáze

Z bezpečnostných dôvodov je prednastavený prístup do MySQL server len lokálne pomocou *localhost* (127.0.0.1).

Pre povolenie prístupu všetkých adries je potrebné editovať súbor *my.cnf*, kde prepíšeme 127.0.0.1 na 0.0.0.0. Tento súbor sa nachádza v */etc/mysql/*.

Pre nastavenie zmien je potrebné reštartovať databázu.

sudo service mysql restart

Tento prístup zatiaľ môže používať len užívateľ root. Preto pridáme všetky práva vytvorenému užívateľovi pre našu databázu

```
GRANT ALL PRIVILEGES ON tempHUM.* TO 'tempHUM'@'%' IDENTIFIED BY 'passwd';
```

flush privileges; priradí privilégiá.

Pre priradenie privilégií len jedenej IP adrese sa znak „%“ nahradí danou IP adresou.

Teraz potrebujeme vytvoriť tabuľky. Prvá tabuľka „*monitor*“ uchováva namerané hodnoty a časy.

```
CREATE TABLE monitor(
id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
terrarium_id INT UNSIGNED NOT NULL,
tH_time DATETIME,
tH_temperatureDS TINYINT,
tH_temperatureDHT TINYINT,
tH_humidity TINYINT(3) UNSIGNED);
```

- *terrarium_id* je ID terária, uloženého v tabuľke „*terrarium*“
- *tH_time* je uchováva čas a dátum merania
- *tH_temperatureDS* ukladá teplotu z čidla DS18B20 merajúceho maximálnu teplotu

- tH_temperatureDHT ukladá teplotu z čidla DHT22 merajúceho minimálnu teplotu
- tH_humidity ukladá hodnoty vlhkostí z čidla DHT22
-

```
mysql> describe monitor;
```

Field	Type	Null	Key	Default	Extra
id	int(10) unsigned	NO	PRI	NULL	auto_increment
terrarium_id	int(10) unsigned	NO		NULL	
tH_time	datetime	YES		NULL	
tH_temperatureDS	tinyint(4)	YES		NULL	
tH_temperatureDHT	tinyint(4)	YES		NULL	
tH_humidity	tinyint(3) unsigned	YES		NULL	

Obr. 25: MySQL - monitor tabuľka

Tabuľka *terrarium* ukladá názvy terárií a ich limity a hard limity (limity, ktoré nie je možné prekročiť).

```
CREATE TABLE terrarium(
  id TINYINT(3) UNSIGNED NOT NULL DEFAULT 0 AUTO_INCREMENT
  PRIMARY KEY,
  name VARCHAR(15) DEFAULT NULL,
  group_id INT UNSIGNED NOT NULL DEFAULT NULL,
  minTemperature TINYINT NOT NULL DEFAULT 20,
  maxTemperature TINYINT NOT NULL DEFAULT 20,
  minHumidity TINYINT NOT NULL DEFAULT 0,
  minTemperature_hard TINYINT NOT NULL DEFAULT 20,
  maxTemperature_hard TINYINT NOT NULL DEFAULT 35,
  minHumidity_hard TINYINT NOT NULL DEFAULT 0);
```

- name je názov terária
- group_id obsahuje ID skupiny
- minTemperature ukladá hodnotu limity pre minimálnu teplotu
- maxTemperature ukladá hodnotu limity pre maximálnu teplotu
- minHumidity ukladá hodnotu limity pre vlhkosť

premenne s *_hard* ukladajú predurčené tvrdé limity pre webové rozhranie pre bezpečnosť

```
mysql> describe terrarium;
```

Field	Type	Null	Key	Default	Extra
id	tinyint(3) unsigned	NO	PRI	0	
name	varchar(15)	YES		NULL	
group_id	int(10) unsigned	NO		NULL	
minTemperature	tinyint(4)	NO		20	
maxTemperature	tinyint(4)	NO		20	
minHumidity	tinyint(4)	NO		0	
minTemperature_hard	tinyint(4)	NO		20	
maxTemperature_hard	tinyint(4)	NO		35	
minHumidity_hard	tinyint(4)	NO		0	

Obr. 26: MySQL - terrarium tabuľka

Tabuľka *terrarium_groups* ukladá názvy terárijných skupín a ich ID.

```
CREATE TABLE terrarium_groups(  
    id INT UNSIGNED NOT NULL DEFAULT NULL AUTO_INCREMENT PRIMARY  
    KEY,  
    group_id VARCHAR NOT NULL DEFAULT NULL);
```

- id ID skupiny
- group_id obsahuje mená pre skupiny

```
mysql> describe terrarium_groups;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type                | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id    | int(10) unsigned   | NO   | PRI | NULL    | auto_increment |  
| name  | varchar(255)       | NO   |     | NULL    |                 |  
+-----+-----+-----+-----+-----+-----+
```

Obr. 27: MySQL - terrarium_groups tabuľka

Tabuľka *time_periods* ukladá hodnoty limitných časov pre nočný režim terárijných skupín a ich ID.

```
CREATE TABLE time_periods(  
    id INT UNSIGNED NOT NULL DEFAULT NULL AUTO_INCREMENT PRIMARY  
    KEY,  
    group_id INT UNSIGNED NOT NULL DEFAULT NULL,  
    time_on TIME NOT NULL DEFAULT NULL,  
    time_off TIME NOT NULL DEFAULT NULL);
```

```
mysql> describe time_periods;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type                | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| id         | int(10) unsigned   | NO   | PRI | NULL    | auto_increment |  
| group_id  | int(10) unsigned   | NO   |     | NULL    |                 |  
| time_on   | time                | NO   |     | NULL    |                 |  
| time_off  | time                | NO   |     | NULL    |                 |  
+-----+-----+-----+-----+-----+-----+
```

Obr. 28: MySQL - time_periods tabuľka

- group_id obsahuje ID skupín
- time_on obsahuje čas v formáte time pre zapnutie regulácie
- time_off obsahuje čas v formáte time pre vypnutie regulácie

Pre výber hodnôt z tabuliek sa používa príkaz:

SELECT (čo - * všetko) *FROM* (názov tabuľky) *WHERE* (id = - napríklad podľa daného ID);

Pre zápis hodnôt do tabuliek sa používa príkaz:

INSERT INTO (názov tabuľky) (výpis premenných do ktorých chceme zapisovať) *VALUES* (dáta, dáta,);

9.2.4 Python

Python je dynamický interpretovaný jazyk, niekedy býva zaradovaný medzi skriptovacie jazyky. Tento programovací jazyk bol navrhnutý tak, aby umožňoval tvorbu rozsiahlych, plnohodnotných aplikácií. Je to hybridný jazyk. To znamená, že umožňuje pri písaní programov používať nielen objektovo orientované paradigma, ale aj procedurálne a obmedzene aj funkcionálne.

Chyba! Nenalezen zdroj odkazů.

Tento software by mal byť nainštalovaný, pokiaľ nie je stačí zadať:

```
sudo apt-get install python-dev
```

V pythne som vytvoril šesť skriptov (spustiteľný kód). Skript *connect.py* obsahuje funkciu, ktorá obsahuje prihlasovacie údaje do MySQL databáze a vytvorenie tejto komunikácie.

Skript *DB.py* obsahuje štyri funkcie, ktoré sú volané ostatnými skriptami. Sú to funkcie:

```
getValuesFROMterrarium(), getValuesFROM(), setActualValuesTOMonitor()
a getCurrentDateTime().
```

Prvá funkcia je využívaná skriptom *limits.py*. Tento skript načíta hodnoty z databázy pre dané terárium a odošle ich pomocou bezdrôtovej komunikácie podriadeným zariadeniam. Vytvára prenášanú správu tvorenú limitami teplôt a vlhkosti a id terária. Tento skript je spúšťaný z webového rozhrania pri zadaní nastavenia limit pomocou php príkazu `exec()`, ktorý prenáša aj id terária pre ktoré chceme limity nastaviť. Skript vytvorenú správu odošle, počká si na potvrdenie a ukončí sa.

Druhú funkciu využíva skript *night.py*, ktorý si načíta časové hodnoty nočného režimu z databázy v použiteľnom formáte (konverzia z dátového typu *time* na *char*) a prepošle ich danej skupine terárií. Tento skript je taktiež spúšťaný z webového rozhrania, kde sa tieto časy zadávajú.

Tretia funkcia je využívaná skriptom *sendCRON.py*, ktorý taktiež vytvára bezdrôtovú komunikáciu a odosiela požiadavku pre podriadené systémy aby odoslali namerané záznamy hodnôt. Táto funkcia zapisuje prijaté hodnoty do databázy. Skript *sendCRON.py* je spúšťaný v určitom intervale. Na začiatku načíta argument z ktorého zistí ID skupiny terárií, ktorej pošle žiadosť o namerané dáta. Po prijatí dát si vytiahne z databázy posledný záznam a porovná ho aktuálnymi hodnotami. Pokiaľ sa žiadne hodnoty nelíšia, tak sa do databázy nové hodnoty nezapisujú. Pri zistení rozdielu niektorej z hodnôt sa u posledného záznamu v databáze posunie čas merania o 2 minúty a zapíše späť do databázy. Následne sa zapíšu aj aktuálne hodnoty.

Porovnávanie nameraných hodnôt s posledným záznamom má výhodu v tom, že sa v databáze nekumulujú záznamy s rovnakými hodnotami. Nevýhodou je, že pokiaľ sa niektorá z nameraných hodnôt dlhšiu dobu nemení, tak graf nemá čo zobrazovať.

Tento skript je automaticky spúšťaný každé 2 minúty pomocou softwaru *crontab*. *Crontab* spustíme príkazom `sudo crontab -e` a zadáme

```
*/2 * * * * /usr/bin/python /home/constructor_rpi/NRF24L01/REQUEST.p >>  
/tmp/listener.log.
```

Existuje generátor pre crontab software, kde stačí zadať kedy sa ma daný skript spúšťať a cestu ku skriptu, prípadne možnosť nastavenia kam odosielať výstup.

Poslednú funkciu využíva skript *clock.py*, ktorý načíta aktuálny čas a odošle ho do podriadeného zariadenia pre aktualizáciu času. Po prijatí slave počká na prerušenie v ktorom nastaví aktuálny čas. Funkcia generuje aktuálny čas a dátum s možnosťou vrátiť buď čas aj dátum alebo len čas.

10 WEBOVÉ ROZHRAŇIE

Popis webového rozhrania.

10.1 Riadiaca aplikácia

Centrálnym blokom navrhnutého systému je riadiaca aplikácia, ktorej účelom je predovšetkým interakcia systému z užívateľom. Aplikácia zobrazuje aktuálny stav systému a umožňuje užívateľovi vykonávať úpravy riadiacich parametrov.

10.2 Voľba technológií pre riadiacu aplikáciu

Tvorbu takejto aplikácie je možné pojať celou radou rôznych spôsobov, napríklad by bolo možné vytvoriť sadu skriptov spúšťaných z príkazového riadku. Pre väčší užívateľský komfort by bolo možné vytvoriť i aplikáciu s grafickým užívateľským rozhraním (Graphical User Interface - GUI).

Vzhľadom k tomu, že medzi požiadavky na riadenie systému patrí aj možnosť vzdialeného prístupu v reálnom čase je nutné pri návrhu riadiacej aplikácie počítať s implementáciou sieťových technológií pre prenos a jeho zabezpečenie. Z tohto hľadiska sa javí ako výhodné využiť k realizácii riadenia už existujúce technológie navrhnuté pre prevádzku siete. Pre svoju vysokú rozšíriteľnosť a štandardizovanosť sa potom priamo ponúka využitie webových technológií.

Konkrétne to znamená užívateľské rozhranie vytvorené v jazykoch HTML, CSS a JavaScript – uvedené technológie sú v súčasnej dobe podporované prakticky každým osobným počítačom a chytrým mobilným telefónom bez ohľadu na výrobcu, sú teda univerzálne použiteľné.

Takto vytvorené rozhranie sa do koncového zariadenia(klienta) prenáša zo serveru protokolom HTTP. Jeho podporu na serveru je možné zaistiť nasadením ľubovoľného HTTP serverového software. Zvolil som voľne dostupný server Apache HTTPd. Alternatívne je možné použiť aj iný webserver, z voľne dostupných napríklad Nginx alebo Lighttpd. Z hľadiska riešeného projektu sa líši nanajvýš spôsobom konfigurácie – poskytované funkcie vyplývajú zo štandardu protokolu HTTP.

Rozhranie zobrazuje informácie uložené v databázy na serveru. Pre prácu s týmito dátami a ich zapracovanie je nevyhnutná technológia pre programovanie na strane serveru. Zvolený server Apache HTTPd umožňuje využitie širokej škály programovacích skriptovacích jazykov. Pre implementáciu riadiacej aplikácie som sa rozhodol použiť jazyk PHP.

Databázový server som zvolil MySQL.

Vlastná implementácia riadiacej aplikácie sa riadi princípmi objektovo orientovaného programovania(OOP), čím je i relatívne jednoduché dosiahnuť znovupoužiteľnosti kódu v aplikácii(napr. pre komunikáciu s databázou). Návrh aplikácie využíva i architektúru MVC(Model-View-Controller), je však nutné uviesť, že prevedené riešenie nesplňuje princípy MVC dôsledne, pretože s ohľadom na jednoduchosť aplikácie by to nebolo účelné.

10.3 Štruktúra riadiacej aplikácie

Adresár aplikácie obsahuje niekoľko súborov a podadresárov lib/.

V adresári lib/ sa nachádzajú súbory s definíciami objektových tried – každá trieda je uložená v samostatnom súbore, ktorého názov je zložený z názvu triedy a prípony .php.

Časť role MVC controlleru zastáva trieda TerraMonitor. Objekt triedy TerraMonitor vo svojich vlastnostiach drží spojenie s databázovým serverom a inštancie ďalších objektov načítaných z databázy. K načítaniu z databázy slúžia metódy triedy TerraMonitor.

Triedy TerrariumGroup, Terrarium a TimePeriod plnia rolu MVC modelov k dátam uloženým v databázových tabuľkách terrarium_groups, terrarium a time_periods. Inštancie niektorej z týchto tried potom obsahuje informácie uložené práve na jednom riadku príslušnej databázovej tabuľky. Všetky tri triedy sú rozšírením triedy DbTable, ktorá univerzálnym spôsobom implementuje uloženie hodnôt v inštancii triedy databáze.

Aplikácia sa spúšťa prostredníctvom súboru index.php. Ten vytvára inštanciu triedy TerraMonitor a spracováva vstupné parametre, následne volá tvorbu užívateľského rozhrania v závislosti na zvolenom typu objektu, ktorý má byť zobrazený. Z hľadiska MVC teda ide o súčasť Controlleru, obsahuje však taktiež definíciu užívateľského rozhrania v časti, ktorá je spoločná obom existujúcim pohľadom. Spoločnou časťou sa rozumie základná štruktúra HTML dokumentu vrátane definície CSS štýlov.

Rolu MVC View plnia súbory view-terrarium.php a view-terrariumGroup.php. Tieto súbory nie sú otvárané priamo užívateľom, sú načítané zo súboru index.php na základe vstupného parametru displayId predávaného v URL.

Súbor/pohľad view-terrariumGroup.php je použitý v prípade, že užívateľ v rozbaľovacej ponuke zvolí niektorú zo skupín terárií. Pohľad umožňuje len nastavenie doby svietenia, pre väčší užívateľský komfort potom zobrazuje odkazy na všetky terária v danej skupine.

Súbor/pohľad view-terrarium.php je použitý v prípade, že užívateľ v rozbaľovacej ponuke zvolí konkrétne terárium. Podobne, ako v predchádzajúcom pohľade, zobrazuje odkaz na skupinu terárií, do ktorej patrí, a ovládacie prvky pre nastavenie medzných hodnôt teploty a vlhkosti. Navyše je vypísaný aktuálny stav zvoleného terária (tzn. Namerané teploty DS a DHT a zistené vlhkosť).

Hodnoty teploty a vlhkosti namerané za posledných 24 hodín sú zobrazené v grafoch. Časový interval pre grafy je možné zmeniť príslušnými vstupnými poliami. Grafy sú vykresľované prostredníctvom JavaScriptovej knižnice Google Charts. Tá sa načíta online, pre správnu funkciu je teda nutné, aby klientske zariadenie malo k dispozícii funkčné pripojenie k internetu, a to i v prípade, že užívateľ k aplikácii pristupuje len v rámci miestnej siete.

Formulár pre nastavenie medzných hodnôt u jednotlivých polí zobrazuje aj "hard-limity". Uvedené hodnoty aplikácia načíta z databázy, kde sú uložené pre každé terárium, a používa ich validáciu nových nastavení – aplikácia neumožní nastaviť vyššiu (resp. nižšiu) medznú hodnotu, než povoľuje príslušný hard-limit. Jedná sa o opatrenie proti chybám užívateľa z nepozornosti, ktoré by pre chované zvieratá mohlo mať fatálne následky. Aplikácia neumožňuje hard-limity priamo meniť, ich úprava je možná len priamym prístupom do databázy.

Stisk potvrdzujúceho tlačítka vo formulároch pre nastavenie predá hodnoty vo formulári metódou POST kódu v súboroch setLimits.php (pre nastavenie konkrétneho terária) alebo setTimeperiod.php (pre nastavenie časového intervalu osvetlenia skupiny terárií). Obe súbory prijaté hodnoty spracujú, pri splnení kontrolných podmienok uloží do databázy a zaistí aplikáciu zmenou spustením externých skriptov v jazyku Python (limits.py, resp. night.py), ktoré predajú aktuálne nastavenie v databázy riadiacim obvodom pre jednotlivé terária. Následne je webový prehliadač presmerovaný späť na index.php.

Prípadné chybové hlásenia sú zobrazené v užívateľskom prostredí vytvorenom súborom index.php, kam sa preniesú mechanizmom PHP Sessions (tzn. Chybové hlásenie je pred presmerovaním uložené na serveru pod náhodne vygenerovaným identifikátorom, ktorý je ďalšej stránke predaný prostredníctvom Cookies alebo ako parameter v URL).

Adresár aplikácie konečne obsahuje taktiež súbor monitor.css, ktorý obsahuje špecifikáciu vzhľadu užívateľského rozhrania v jazyku CSS, a podadresárom img/s grafickými súbormi odkazovanými z CSS štýlu.

10.4 Bezpečnosť

Keďže naše zariadenie odhalíme svetu internetu, musíme ho zabezpečiť pred jeho hrozbami, ako sú rôzne automatické skripty ktoré používajú slovníky, ktoré pozostávajú z bežných užívateľských mien a hesiel ako root: 1234 atd. Záznamy o žiadostiach k prístupu do RPi sa ukladajú do `/var/log/auth.log`. Ak necháme bežať RPi pripojené k internetu pár dní, uvidíme niekoľko pokusov o prístup do systému. Dôležitá vec je držať svoje RPi aktualizované, zvoliť silné heslo, zakázať vzdialený prístup užívateľa root...

Okrem klasického updatu a upgradu je vhodné aktualizovať aj firmware. Táto aktualizácia sa spustí príkazom

```
sudo rpi-update
```

ak tento balík nie je nainštalovaný spustíme

```
sudo apt-get install rpi-update
```

Týmto máme aktualizovaný software a firmware nášho RPi.

Následne je vhodné užívateľa „pi“ zmeniť na niečo náročnejšie na uhádnutie a taktiež prideliť silné heslo. Pri prihlásení sa k RPi zistíme pomocou príkazu `groups` ku ktorým zdrojom máme prístup. Príklad výstupu:

```
pi adm dialout cdrom sudo video plugdev users netdev input
```

Pre vytvorenie nového užívateľa potrebujeme zadať náš list skupín v nasledujúcom formáte:

```
sudo useradd -m -G adm,dialout,cdrom,sudo,video,plugdev,users,netdev,input USERNAME
```

Následne priradíme heslo

```
sudo passwd USERNAME
```

a reštartujeme RPi

```
sudo reboot
```

Po reštarte sa prihlásime pod novým užívateľom. Teraz máme všetky možnosti ako užívateľ pi. Aby nebolo možné sa ako užívateľ pi prihlasovať tak tento účet zrušíme.

```
sudo deluser --remove-all-files pi
```

Po veľkom množstve textu sa na konci vypíše 'Done'. Teraz užívateľ pi a všetky jeho súbory sú vymazané zo systému. Chyba! Nenalezen zdroj odkazů.

Následne môžeme zrušiť vzdialený prístup užívateľovi root. To spravíme pomocou sshd konfiguračného súboru

```
sudo nano -etc-ssh-sshd_config
```

v ktorom zmeníme v riadku `PermitRootLogin` na `no`. V tomto súbore môžeme zmeniť prednastavený SSH port(22) na niečo iné (napr. 2020). To obyčajne zníži počet pokusov, ale stále ich bude celkom dost'.

K predošlým zabezpečeniam som pridal ochranu softwarom `fail2ban`.

```
sudo apt-get install fail2ban    inštalácia fail2ban
```

Tento software je možné použiť na akúkoľvek službu, ktorá vytvára logovacie súbory ako Apache, FTP a iné. Konfiguráciu rôznych služieb nájdeme v `/etc/fail2ban/jail.conf`. Prednastavená konfigurácia monitoruje len SSH a blokuje (ban) podozrivé IP adresy po šiestich neúspešných pokusoch na 600 sekúnd. Zmeniť nastavenia je možné v `/etc/fail2ban/jail.local`. Napríklad zablokovať IP adresy už po dvoch neúspešných pokusoch permanentne.

Blokované IP adresy zobrazíme príkazom

```
sudo iptables -L -n --line
```

Viac informácií na [Chyba! Nenalezen zdroj odkazů.](#)

10.4.1 Sprístupnenie do internetu

Pokiaľ máme na svojom domovskom routeru statickú IP adresu tak je prístup k RPi ako aj k webovej stránke jednoduchý. Avšak ten kto nemá, musí použiť niečo iné. Ja som zvolil možnosť použitia dynamickej DNS (DDNS).

Dynamické DNS je označenie systému, ktoré umožňuje v reálnom čase aktualizovať záznamy uložené o internetovej doméne na DNS serveru. Najčastejšie sa jedná o prepojenie názvu počítača s dynamicky pridelenou IP adresou, čo umožňuje používať pre spojenie s počítačom stabilné DNS meno namiesto neustále sa meniacej IP adresy. Takže vlastne meniaci sa IP adresa routeru má stále rovnaké DNS meno, pomocou ktorého sa dostanem do vlastnej siete z vonku.

Je mnoho stránok poskytujúcich túto možnosť. Ja som zvolil verziu zdarma pomocou *noip.com*. Keďže je to zdarma, tak je nutné po určitej dobe buď zmeniť DNS meno alebo ho zrušiť a vytvoriť nové.

Najprv si musíme vytvoriť účet. Stačí na *noip.com* cez "Sing in" a "register" zadať e-mail, užívateľské meno a heslo a potvrdiť "Create My Free Account".

Po prihlásení stačí zadať "Hostname" a vybrať free doménu. K prístupu k webovej stránke z internetu je nutné v nastavení routeru prideliť smerovanie (forwarding) na port 80 pre IP adresu RPi. Pre povolenie SSH je to port 22. Takto máme vytvorený prístup do vlastnej siete cez DDNS.

Nasleduje nastavenie RPi. V domovskom adresár(*cd*):

```

mkdir noip                vytvoríme priečinok noip.
cd ./noip                 presun do priečinku
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
                             stiahneme software
tar vzxvf noip-duc-linux.tar.gz rozbalíme
cd noip-2.1.9-1           presunieme sa....
sudo make                 a nainštalujeme
sudo make install        po príkaze install príde požiadavka na
                             zadanie prihlasovacích údajov a na nastavenie intervalu pre aktualizáciu IP
                             adresy. Zadaný čas je v minútach, minimálna doba je 5 minút.
Pre overenie funkčnosti:
sudo /usr/local/bin/noip2
sudo noip2 -S

```

Teraz máme prístup k RPi a webovej stránke z internetu. Stačí zadať 'hostname'.domain'/monitor.

10.4.2 Autentizácia užívateľa

Pri vzdialenom prístupe je nevyhnutné vykonávať autentifikáciu užívateľa. Bez nej by náhodný návštevník (čo by mohol byť napr. aj automatizovaný spamový robot) meniť limitné hodnoty pre terária a chovaným zvieratám spôsobiť ujmu. Na dosiahnutie tohto cieľa pripadá do úvahy hneď niekoľko možností.

Ako najprimitívnejšie metódu možno použiť umiestnenie aplikácie na adrese, ktorú nemožno odhadnúť, napr.

http://hostname/monitor_7xsbWDfTTJ/index.php.

Používanie takejto URL je však užívateľsky nepohodlné a jej použitie v akomkoľvek zariadení, ktoré nie je pod plnou kontrolou používateľa aplikácie, by znamenalo faktickú kompromitáciu aplikácie. Ide o typický príklad "Security by Obscurity" a pre potreby riadiace aplikácie je tento postup úplne nedostačujúci.

Autentizáciu je možné implementovať priamo vo webovej aplikácii, ktorá prostredníctvom webového formulára prijme užívateľské meno a heslo, porovná ho so záznamami vo svojej databáze a v prípade zhody užívateľovi sprístupní tie funkcie, ku ktorým je oprávnený používať. Tento postup je vhodný najmä pre zložitejšie aplikácie, ktoré rozlišujú rôzne úrovne oprávnenia užívateľov alebo ich rôzne roly. Implementácia vlastného prihlasovania potom aplikácii umožňuje veľmi detailné riadenie prístupu.

Ďalšou možnosťou je využiť podporu autentizácie v protokole HTTP. V prípade, že webový prehliadač pristupuje k obsahu, ktorý je v konfigurácii

webservera označený ako chránený, odošle webserver prehliadači HTTP stavový kód 401 Authorization required, na základe čoho webový prehliadač zobrazí výzvu užívateľovi k prihláseniu (výzvu zobrazuje priamo prehliadač prostredníctvom vlastného vstavaného mechanizmu). Až po zadaní správnych prihlasovacích údajov webserver overí požiadavku a vráti požadovaný obsah (kód stavu 200 OK), prípadne chybové hlásenie (napr. 404 Not Found).

Posledný z uvedených možností je veľmi jednoduché nasadiť - jedná sa o obyčajnú úpravu konfigurácie webservera, navyše neprihláseným používateľom znemožňuje akýkoľvek prístup k aplikácii. To znamená, že za predpokladu, že webserver neobsahuje bezpečnostné chyby, nemôžu neprihlásenie používateľa hľadať prípadné bezpečnostné nedostatky aplikácie samotnej. Pretože riadiace aplikácie nevyžaduje jemné riadenie prístupu, javí sa využitie HTTP autentizácia ako najvhodnejšie pre daný účel

Pre zabezpečenie stránky použijeme konfiguračný súbor serveru .htaccess a .htpasswd pre vytvorenie a zašifrovanie hesla.

.htaccess je textový súbor, ktorý slúži k tomu aby si autor stránok mohol sám upraviť niektoré vlastnosti serveru bez toho, aby o to žiadal správcu.

Najprv tento súbor vytvoríme

```
sudo nano /var/www/html/monitor/.htaccess
```

Do súboru napíšeme

```
AuthType Basic
```

```
AuthName "Restricted area. Please login."
```

```
AuthUserFile /var/www/html/monitor/.htpasswd
```

```
Require valid-user
```

Týmto nastavíme typ autorizácie na bežný, správu, oznamujúcu oblasť do ktorej je nutné sa autorizovať, cestu k súboru s našimi vytvorenými užívateľmi a ich heslami a povolenie pre prihlásenie sa všetkých užívateľov v tomto súbore.

Súbor .htpasswd vytvoríme v priečinku monitor. Tento súbor vytvorí užívateľa s heslom, na ktoré použije šifrovanie bcrypt pomocou príkazu

```
htpasswd -nbB myName myPassword
```

Užívateľ pre tento projekt je na ukážku user s heslom user

```
htpasswd -nbB user user
```

Po zadaní do prehliadača IP adresy RPI s cestou k našej stránke uvidíme

Prosím, přihlaste se

Doména http://192.168.0.10 vyžaduje zadání uživatelského jména a hesla.

Spojení s tímto webem není soukromé.

Přihlašovací jméno:

Heslo:

Obr. 29: Login WWW

Po zadání mena a hesla (user: user) sa dostaneme k našej stránke
Chyba! Nenalezen zdroj odkazů.

11 RIEŠENIE PROBLÉMOV

V priebehu vypracovávaní tejto diplomovej práce vzniklo niekoľko problémov.

Prvým problémom bol časový sklz spôsobený výrobou dosiek plošných spojov, ktorá zabrala mesiac a následná komplementácia súčiastok. Na začiatku keď som mal kompletnú dosku musel som riešiť akúkoľvek komunikáciu s riadiacim počítačom. Bohužiaľ mnou používaný programátor nekomunikoval s navrhnutým zariadením. Po dlhej snahe nájsť chybu som sa tento problém snažil vyriešiť pomocou iného programátora. Avšak ani to nepomohlo.

Potom som použil Arduino NANO ako ISP programátor. Ďalším problémom sa stala snaha rozchodiť navrhnutý plošný spoj s bezdrôtovou komunikáciou. Toto sa ukázalo ako najzávažnejší problém, pretože na navrhnutom a vyrobenom zariadení sa nepodarilo vytvoriť bezdrôtovú komunikáciu medzi centrálnym počítačom a vytvoreným zariadením. Kvôli tomu som musel siahnuť po vývojovom kíte Arduino NANO, ktoré som pôvodne nechcel použiť. Tu sa taktiež vyskytla chyba, kde pri programovaní padla komunikácia a pravdepodobne na vývojovom kíte odišiel programátor. Čo viedlo k ďalšiemu zdržaniu v pokračovaní práce.

12 VYHODNOTENIE

Mnou navrhnutý systém je v zásade funkčný, pretože meria a reguluje skupiny terárií. Taktiež šetrí miesto v databáze, do ktorej nezapisuje duplicitné záznamy.

Webová stránka navrhnutá ako “user friendly”, kde užívateľ nemusí niekoľkokrát preklikávať výber terárií.

Bezdrôtová komunikácia je funkčná ale nie bezchybná. V určitom okamžiku nastane u zariadenia blok a prestane prijímať správy. Tento blok sa väčšinou zruší po prerušení od časovača, avšak niekedy trvá dlhšie. Na otestovanie komunikácie je potreba viac času, než som mal k dispozícii.

Keďže meriam teploty na celé čísla, a nie na desatinné tak namerané hodnoty sa rovnajú skutočným. Otestoval som viacero snímačov a namerané výsledky sa zhodovali u teplôt. Pri meraní vlhkosti vzduchu sa namerané hodnoty mierne líšili. Z vyskytnutých problémov popísaných vyššie a teda nedostatku času som nestihol dôkladnejšie premerať závislosti snímačov.

12.1 Možnosť ďalšieho rozvoja riadiacej aplikácie

Pre lepšie zabezpečenie komunikácie, najmä pre ochranu hesla pred odpočúvaním, možno použiť protokol HTTPS. K tomu je potrebné získať certifikáty (buď podpísané uznávanou autoritou, alebo self-signed) a upraviť konfiguráciu webservera. Aplikácia samotná nevyžaduje úpravy.

Väčšieho komfortu používania možno dosiahnuť použitím knižníc hotových ovládacích prvkov, ako sú napr. JQuery UI, jQuery Mobile alebo SAP UI5, prípadne frameworkov typu Bootstrap. Užívateľské rozhranie by tak bolo možné sprehľadniť napríklad použitím záložiek.

Možnosti aplikácie možno rozšíriť pridaním nových funkcií (napr. Správa terárií, ktorá by umožnila pridať nové terária a ich skupiny, čo je aktivita, ktorú v súčasnom riešení možno vykonať len priamym prístupom do databázy), rozšírením existujúcich funkcií (napr. umožniť pridanie viac intervalov svietenie pre skupiny terárií) alebo pridaním nových prvkov (napr. výstup z webkamery sledujúci určitej terárium).

13 ZÁVER

Cieľom tejto diplomovej práce bolo navrhnúť systém pre monitorovanie a regulovanie klimatických podmienok v skupinách terárií. Každá skupina zostavená z 1-5 terárií má vlastný riadiaci systém, ktorý bezdrôtovo komunikuje s nadradeným systémom. Prenesené namerané hodnoty sú archivované v databáze a vhodne zobrazené cez webové rozhranie.

Navrhol som systém komunikácie a typy správ, podľa požiadaviek vybral vhodné komponenty, navrhol schémy zapojenia a dosky plošných spojov. Následne som vyrobil dosky plošných spojov, kde som sa snažil o minimalizáciu. Vyrobená doska má dve vrstvy a vyvedené zdieľky pre možnosť pridania napájania z batérií. Naprogramoval som bezdrôtovú komunikáciu. Slave zariadenie meria a reguluje teplotu a vlhkosť. Namerané hodnoty sa ukladajú do databáze a následne sa prehľadne zobrazujú na webovom serveri. Taktiež je možné nastavovať limity a čas regulácie z webovej stránky, to je však možné len v dovolených hraniciach. Ďalej som vhodne zabezpečil prístup do nadradeného počítača.

Následne by bolo vhodné navrhnúť náhradu spínania relé polovodičovými súčiastkami pre spínanie/rozopínanie pulzom napätia.

Literatura

- [1] Meteorologie: Klimatologie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-15]. Dostupné z: <https://cs.wikipedia.org/wiki/Meteorologie>
- [2] Teplota vzduchu. *Meteocentrum* [online]. [cit. 2017-05-15]. Dostupné z: <http://www.meteocentrum.cz/zajimavosti/encyklopedie/teplota-vzduchu>
- [3] Vlhkost vzduchu. *Artemis* [online]. [cit. 2017-05-15]. Dostupné z: <http://artemis.osu.cz/Gemet/meteo2/vlhkost.htm>
- [4] *Raspberrypi* [online]. [cit. 2017-05-15]. Dostupné z: http://www.geeetech.com/wiki/index.php/Raspberry_Pi_Model_B%2B
- [5] *PHP5, MySQL, Apache: Vytváříme webové aplikace*. Computer Press, 2006, s. 813. ISBN 80-251-1073-7.
- [6] *Arduino* [online]. [cit. 2017-05-15]. Dostupné z: <http://arduino.cz>
- [7] MegaAVR Microcontrollers. *Microchip* [online]. [cit. 2017-05-15]. Dostupné z: <http://www.atmel.com/products/microcontrollers/avr/megaavr.aspx>
- [8] LPC11C24. *AllDatasheet* [online]. [cit. 2017-05-15]. Dostupné z: <http://www.alldatasheet.com/view.jsp?Searchword=Lpc11c24>
- [9] Freedom KE06 User's Guide. *Freescale* [online]. 2014 [cit. 2017-05-15]. Dostupné z: http://cache.freescale.com/files/microcontrollers/doc/user_guide/FRDMKE06UG.pdf
- [10] PIC18F45K80. *Microchip* [online]. [cit. 2017-05-15]. Dostupné z: <http://www.microchip.com/wwwproducts/en/PIC18F45K80>
- [11] DHT22 Datasheet PDF – Sensor Module. *DataSheetCafe* [online]. [cit. 2017-05-15]. Dostupné z: <http://www.datasheetcafe.com/dht22-datasheet-pdf/>
- [12] NRF24L01+ sniffer. *Embedded Innovation* [online]. [cit. 2017-05-15]. Dostupné z: <http://yveaux.blogspot.cz/2014/07/nrf24l01-sniffer-part-1.html>

- [13] Microchip Wireless – Leader in Low Power Embedded Wireless Solutions. *Microchip* [online]. [cit. 2017-05-15]. Dostupné z: <http://www.microchip.com/design-centers/wireless-connectivity>
- [14] NRF24L01+. *Sparkfun* [online]. [cit. 2017-05-15]. Dostupné z: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus_Preliminary_Product_Specification_v1_0.pdf
- [15] TIŠNOVSKÝ, Pavel. Externí sériové sběrnice SPI a I2C. In: Root.cz - informace nejen ze světa Linuxu [online]. 30.12.2008 [cit. 2017-05-15]. Dostupné z URL: <https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/>
- [16] DS18B20- Dallas Semiconductor. *AllDatasheet* [online]. [cit. 2017-05-15]. Dostupné z: <http://www.datasheetcafe.com/dht22-datasheet-pdf/>
- [17] DS3231- Dallas Semiconductor. *AllDatasheet* [online]. [cit. 2017-05-15]. Dostupné z: <http://www.datasheetcafe.com/dht22-datasheet-pdf/>
- [18] I2C Real-Time Clock/Calendar. *Microchip* [online]. [cit. 2017-05-15]. Dostupné z: <http://www.microchip.com/design-centers/real-time-clock-calendar>
- [19] 10 Best Raspberry Pi and Pi 2 Alternatives. *Beebom* [online]. [cit. 2017-01-04]. Dostupné z: <http://beebom.com/raspberry-pi-and-pi-2-alternatives/>

Seznam příloh

A Navrhnuté DPS s MCU ATmega32 - 1

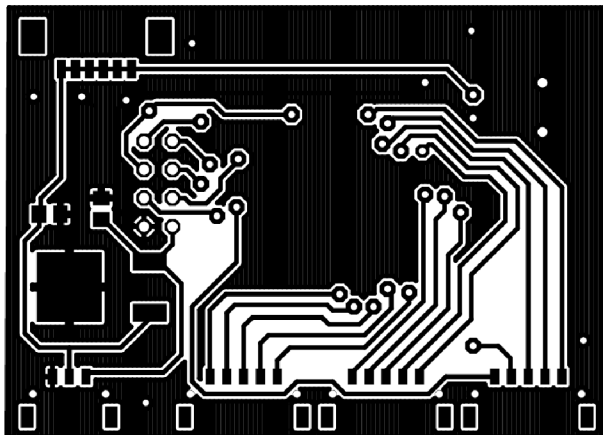
B Navrhnuté DPS s MCU ATmega32 - 2

C Schéma zapojenia

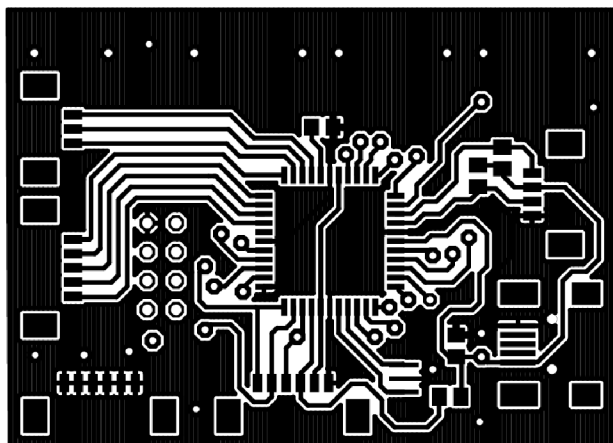
D Zoznam súčiastok

E Web stránka

A Navrhnuté DPS s MCU ATmega32 - 1

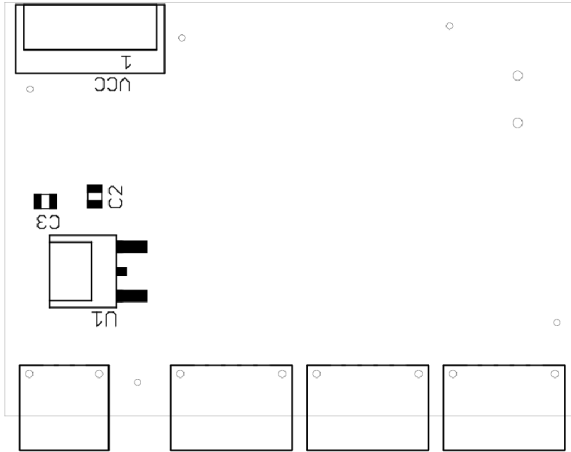


Obr. 30: DPS spodní layout

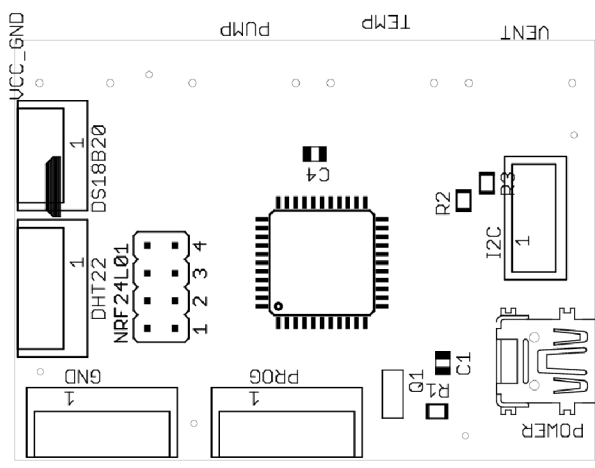


Obr. 31: DPS horní layout

B Navrhnuté DPS s MCU ATmega32 - 2

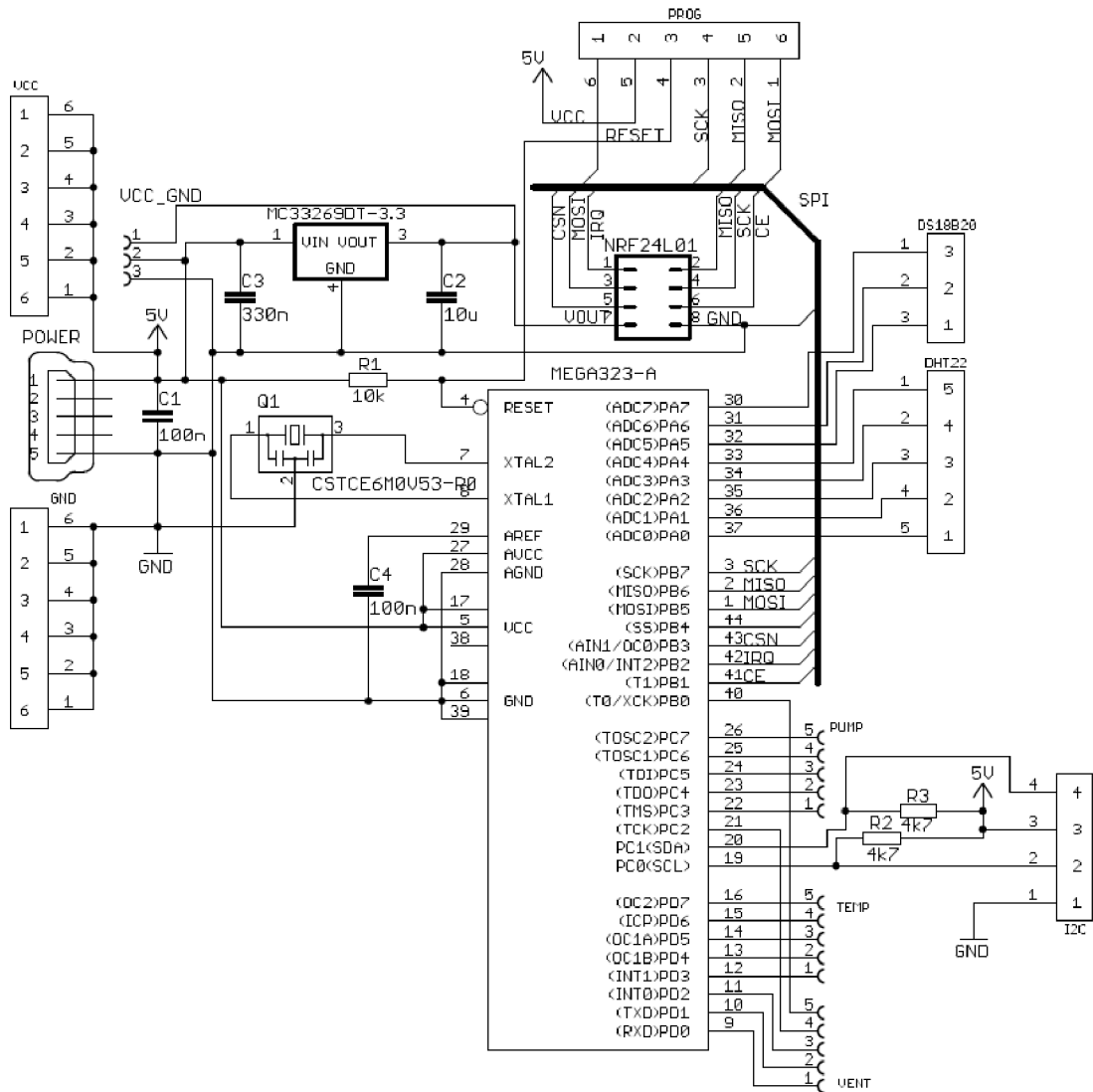


Obr. 32: DPS - osadenie spodná strana



Obr. 33: DPS - osadenie horná strana

C Schéma zapojenia



Obr. 34: Schéma zapojenia

D Zoznam súčiastok

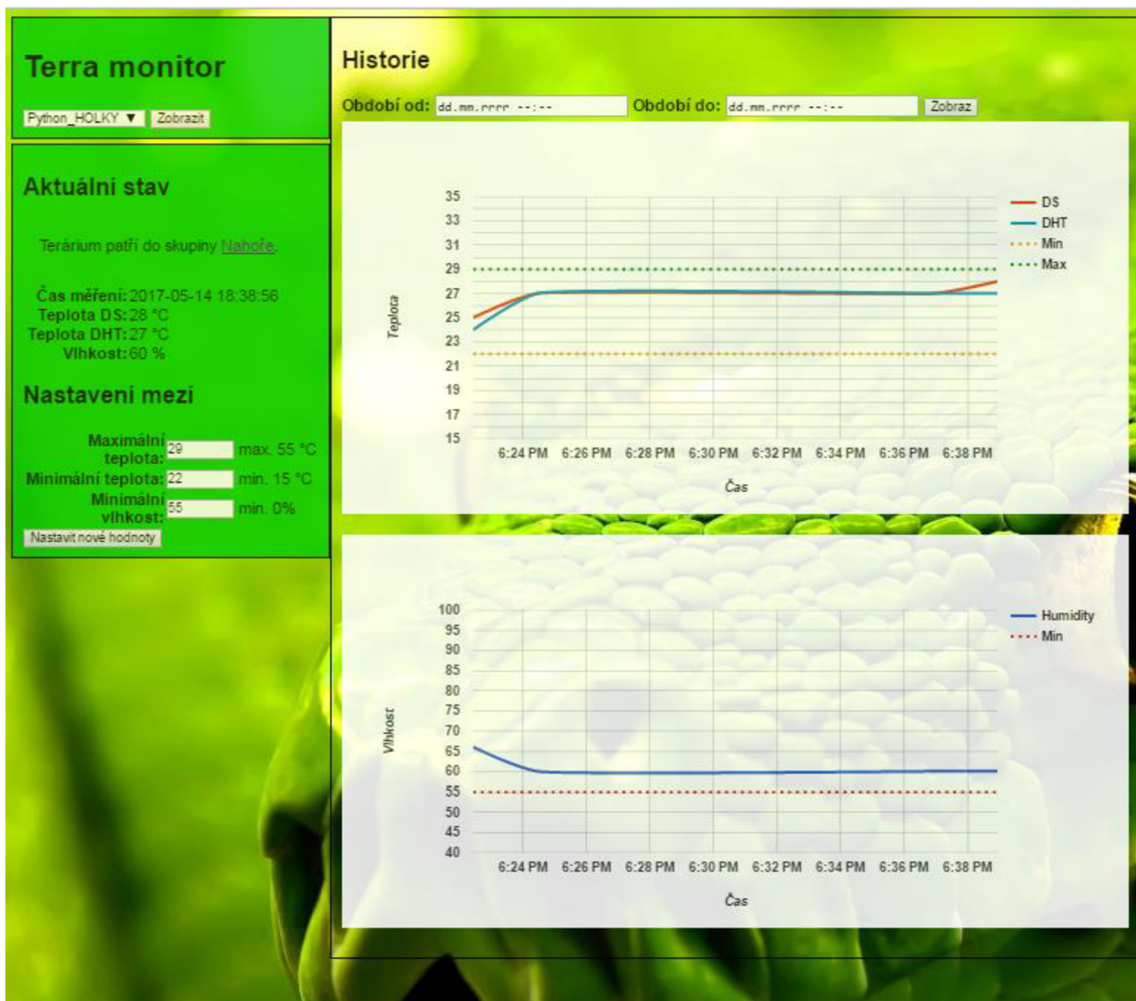
Tab. 2: Zoznam súčiastok

Počet	Názov	Púzdro	Referencie	Poppis
1	Atmega32	TQFP-44	Atmel	8bit mikrokontrolér
1	MC33269DT-3.3	DPAK ON	ON Semi	Regulátor 5/3,3V
1	CSTCE16M0V53-R0	CSTCRM6	MURATA	Rezonátor 16MHz
2	100nF	SMD 0805	C1, C4	Keramický kondenzátor
1	10uF/16V	CTS AVX typA	C2	Tantálový kondenzátor
1	330nF/35V	CTS YHC typA	C3	Tantálový kondenzátor
1	10k	SMD 0805	R1	Čipový rezistor
2	4k7	SMD 0805	R2, R3	Čipový rezistor
1	Mini USB	Typ AB USB 2.0	MULTICOMP	Napájanie 5V
1	3pin	WR-WTB 1.25mm	Würth	Konektor DS18B20 senzory
1	4pin	WR-WTB 1.25mm	Würth	Konektor DS3231 RTC
1	5pin	WR-WTB 1.25mm	Würth	Konektor DHT22 senzory
3	6pin	WR-WTB 1.25mm	Würth	VCC, GND, Programovanie
1	3pin	WR-LECO 1.50mm	Würth	5V, 3,3V, GND
3	6pin	WR-LECO 1.50mm	Würth	Regulácia OUT piny
1	4x2	2,54mm	nRF24L01+	Dutinková lišta

E Web stránka



Obr. 35: Web stránka - skupiny



Obr. 36: Web stránka - grafy, terária