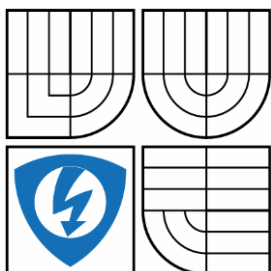


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A
KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

INTELIGENTNÍ ZAHRADA

INTELLIGENT GARDEN

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Jakub Dvořák

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Radek Štohl, Ph.D.

BRNO 2016

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Jakub Dvořák

ID: 134299

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Inteligentní zahrada

POKYNY PRO VYPRACOVÁNÍ:

1. Proveďte literární rešerši o řízení inteligentních rodinných domů.
2. Navrhněte projekt inteligentní zahrady, specifikujte potřebný HW.
3. Navrhněte a realizujte řídicí HW.
4. Navrhněte a realizujte řídicí SW a webový server na platformě Raspberry Pi.
5. Ověřte a vyhodnoťte své řešení na reálné zahradě.

DOPORUČENÁ LITERATURA:

MERZ, HANSEMAN, HÜBNER. Automatizované systémy budov: sdělovací systémy KNX/EIB, LON a BACNet. Praha. GRADA Publishing a.s. 2009. ISBN 978-80-247-2367-9.

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

Termín zadání: 8.2.2016

Termín odevzdání: 16.5.2016

Vedoucí práce: Ing. Radek Štohl, Ph.D.

Konzultant diplomové práce:

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

Hlavním cílem této práce je vytvořit za pomoci Raspberry Pi inteligentní zahrádku. První část je věnována pojmu inteligentní dům. V druhé části jsou popsány použité HW prvky. Další část je věnována SW vybavení mikropočítače Raspberry a mikrokontroleru Arduino. V poslední kapitole je popsán realizovaný model inteligentní zahrádky.

Klíčová slova

Raspberry, RPi, Arduino, Inteligentní zahrádka, Web server

Abstract

The main objective of this work is created using Raspberry Pi intelligent garden. The first part is devoted to the concept of intelligent building. The second part describes the used hardware components. Another section is devoted SW Raspberry microcomputer and microcontroller Arduino. The last chapter describes the implemented model of a intelligent garden.

Keywords

Raspberry, RPI, Arduino, Intelligent garden, Web server

Bibliografická citace:

DVOŘÁK, J. *Intelligentní zahrada*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 53s. Vedoucí diplomové práce byl Ing. Radek Štohl, Ph.D.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Inteligentní zahrada jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si 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.

V Brně dne: **16. května 2016**

.....
podpis autora

Poděkování

Děkuji Josefu Cabrnochovi a Lubošovi Ráčkovi za účinnou odbornou pomoc a další cenné rady, dále pak děkuji rodině za podporu při studiích a poděkování patří taktéž vedoucímu mé diplomové práce ing. Radku Štohlovi Ph.D.

V Brně dne: **16. května 2016**

.....
podpis autora

Obsah

1	Úvod.....	12
2	Inteligentní rodinné domy	13
3	Návrh Inteligentní zahrádky.....	15
4	Raspberry	16
4.1	Parametry Raspberry	18
4.2	GPIO piny Raspberry 2 Model B	19
4.3	OS Raspberry Pi.....	19
4.4	Instalace Raspbianu.....	20
4.5	Spuštění Raspbianu pomocí Putty.....	21
5	Ostatní HW.....	24
5.1	Arduino	24
5.2	Senzory.....	26
5.2.1	Senzor teploty Dallas DS18B20	27
5.2.2	Digitální senzor intenzity světla BHI1750FVI	27
5.2.3	Ultrazvukový senzor vzdálenosti HC – SR04	28
5.2.4	Čidlo vlhkosti půdy	28
5.3	Osvětlení, čerpadlo.....	29
5.3.1	LED osvětlení	29
5.3.2	Čerpadlo Barwig	29
5.3.3	Arduino relé 2 kanály	30
5.4	Zdroj.....	30
5.5	Webkamera	31
6	SW vybavení	32
6.1	Použité prostředky pro realizaci systému.....	32
6.2	Webový server	33
6.2.1	Apache 2.....	33
6.2.2	MySQL	33
6.2.3	PhpMyAdmin	34
6.2.4	Webmin	34
6.2.5	Nastavení routeru TP-link TL-WR940N.....	35
6.3	Databáze.....	36
6.4	Komunikace pomocí sériového portu	36
6.5	Sběr dat ze senzorů a zápis do MySQL.....	37

6.5.1	Program na výčet hodnot z čidel Uart.py	38
6.5.2	Čtení a odesílání hodnot z Arduina do RPi	39
6.5.3	Ukládání do databáze db.py.....	42
6.6	Aktivace osvětlení a čerpadla.....	42
6.7	Cron.....	43
6.8	Program pro obsluhu kamery	44
6.9	Webové rozhraní	44
6.9.1	Stránka Home	45
6.9.2	Stránka Historie	47
7	Realizace Modelu.....	49
8	Závěr	50

Seznam obrázků:

Obrázek 1: Modely Raspberry Pi [4]	17
Obrázek 2: Rozmístění GPIO pinů Raspberry Pi 2 Model B [5]	19
Obrázek 4: Win32DiskImage.....	21
Obrázek 6: Program Putty	22
Obrázek 8: Program raspi-config	23
Obrázek 9: Zapojení senzorů k Arduinu	27
Obrázek 10: Senzor teploty Dallas DS18B20 [9]	27
Obrázek 11: Světelný senzor BH1750FVI [11]	28
Obrázek 12: HC - SR04 [13].....	28
Obrázek 13: Senzor vlhkosti půdy [14].....	29
Obrázek 14: 12 V čerpadlo.....	30
Obrázek 15: Arduino relé 2 kanály [15].....	30
Obrázek 16: Konektory ATX zdroj [16]	31
Obrázek 17: Webkamera FaceCam 320 [17]	31
Obrázek 18: Schématické propojení komponent.....	32
Obrázek 19: Databáze MySQL	34
Obrázek 20: Program PhpMyAdmin.....	34
Obrázek 21: Program Webmin.....	35
Obrázek 22: Nastavení virtuálního serveru v routeru TP link.....	35
Obrázek 23: Databáze garden	36
Obrázek 24: Buffer seriové linky	37
Obrázek 25: Software pro sběr dat ze senzorů a ukládání do databáze.....	38
Obrázek 26: Práva video0	44
Obrázek 27: Grafické zobrazení aktuálních hodnot ze senzorů	46
Obrázek 28: Jednoduché ovládání osvětlení a čerpadla	46
Obrázek 29: Část webové stránky Historie	47
Obrázek 30: Model inteligentní zahrady	49

1 ÚVOD

Úkolem této diplomové práce je realizace inteligentní zahrádky. Tato práce je rozdělena celkem na 7 kapitol.

Kapitola nacházející se po úvodu se zabývá inteligentními rodinnými domy, kde je popsáno co spojení inteligentní dům vůbec znamená a jaké má inteligentní dům výhody.

V další kapitole je vytvořen slovní návrh pro následnou realizaci inteligentní zahrádky. Kromě navrženého řídicího hardwaru jsou zde vypsány i požadavky na inteligentní zahrádku.

Následující kapitola je věnována hlavnímu řídicímu prvku a to konkrétně počítači Raspberry Pi. Který slouží jako webový server.

V kapitole číslo šest jsou popsány ostatní hardware prvky jako je Arduino Uno, veškeré senzory použité při realizaci inteligentní zahrádky a dále pak osvětlení a čerpadlo, které slouží k zalívání.

V další kapitole jsou popsány veškeré software vybavení použité a naprogramované při realizaci zahrádky.

V poslední kapitole je popsán fyzický model inteligentní zahrádky.

2 INTELIGENTNÍ RODINNÉ DOMY

Z důvodu velkého nárůstu informačních technologií se začínají stávat tyto technologie běžnou součástí života. Co začalo počítačem, mobilem se dále rozvíjí v tzv. internet věcí (Internet of Things), což je síť tvořena fyzickými objekty s vnitřní elektronikou. Ta se skládá především ze senzorů, výpočetní jednotky a komunikačních prostředků. S rostoucí integrací vestavných zařízení umožňuje sledovat stále více aspektů našeho života. [1]

Internet věcí je základem domácí automatizace. Jedná se o propojení senzorů a reakčních prvků v domě, které umožňují automaticky reagovat na právě vzniknuté situace bez potřeby zásahu obyvatel domu. Jako příklad může být vytápění domácnosti. V každé místnosti se nachází zdroj tepla a všechny zdroje tepla jsou obvykle řízeny jedním termostatem na celý dům. To má za následek, že v místnosti, kde se nachází termostat je teplota námi nastavená, ale v ostatních místnostech je hodnota odlišná. Přidáním teplotních senzorů do každé místnosti a upravením regulace tak, aby bylo možné spínat samostatné zdroje tepla výrazně zlepšuje situaci a díky tomuto lze změřit, jakým způsobem ovlivňuje teplotu ve sledované místnosti topení z okolních místností.[1]

Hlavní výhody nasazení a také množství výhod je závislé na konkrétních prvcích, které jsou v domě nasazené. Nejde ani poskytnout úplný seznam výhod, jelikož domácí automatizace patří k jedné z nejrychleji se rozvíjejících aplikací vestavných zařízení dneška a oblasti využití se rychle rozšiřují. Integrací senzorů a akčních prvků do spotřebičů můžeme zvýšit pohodlí, které dům poskytuje svým uživatelům a stejně tak jejich pocit bezpečí. Běžná je situace, kdy se před dovolenou na poslední chvíli používá žehlička, a následně pak člověk po cestě přemýšlí, jestli náhodou nezůstala zapojená. Dalším hezkým příkladem je varná deska. I u ní se může stát podobná situace jako s žehličkou. Co třeba televize? Kolikrát se stane, že dítě ráno odchází z domu a nechá zapnutou televizi, která následně běží až do večera a neustále spotřebovává proud? Právě tyto případy, řeší možnost ovládání zásuvek případně širší možnosti vestavěné ve spotřebičích. V mobilní aplikaci jde lehce zkontrolovat, který spotřebič zůstal běžet a je možné ho vypnout. [1]

Osvětlení je jedním z hlavních spotřebitelů elektrické energie v domě. Tato situace se sice s příchodem LED osvětlení mění, ale stále může být dosaženo výrazných úspor. Běžné použití znamená zapínání světel pomocí vypínače a jejich provoz na plný výkon. Plný provoz zůstává i v situaci, kdy v místnosti nikdo nezůstal, ale poslední odcházející zapomněl zhasnout v dobré víře, že se za chvíli vrátí. Oproti tomu chytrý dům sleduje pohyb svých obyvatel, sleduje úroveň světla v jednotlivých místnostech a je schopný světla sepnout nebo vypnout dle potřeby. Při

využití světla, která jsou ztlumitelná a je možné i regulovat, jak intenzivně světlo svítí. Díky tomu je možné dosáhnout i mnohem příjemnější atmosféry. [1]

Zabezpečení vstupu pomocí inteligentních zámků výrazně usnadňuje život, protože kolikrát se stane, že při cestě do práce se člověk přistihne s myšlenkou, jestli náhodou nezapomněl zamknout dveře? Co když dítě zapomene klíče od domu a nemá se jak dostat dovnitř? Pokud přidáme do dveří elektromechanická kování, můžeme odkudkoliv zkontrolovat stav zámků a případně zamknout nebo odemknout dveře. [1]

Kontrola zavřených oken a dveří. Dveřní senzory umožňují detekovat, zda je otevřeno nebo zavřeno a tuto informaci poskytnout svému uživateli. Pro větší pocit bezpečí je možné na sklo integrovat senzory vibrací, která alarmují, pokud došlo k rozbití skla. [1]

Kamerový systém v domě dává přehled o aktuálním dění. V kombinaci s nahrávacím zařízením může například také pomoci s usvědčení pachatelů, ať už trestných činů, nebo domácích mazlíčků, kteří v nepřítomnosti „páníčka“, dělají zakázané věci. [1]

Integrací senzorů sledujících teplotu, vlhkost a kvalitu vzduchu můžeme udržovat všude v domě příjemné klima. Pokud se nikdo přes den v domě nenachází, může být automaticky snížena teplota, a před návratem zase opětovně zvýšena. Toto opatření vede nejen ke zvýšenému pohodlí, ale také ke snížení výdajů za energie. Sledováním škodlivin ve vzduchu (např. CO) lze odhalit požár nebo obsah jedovatých plynů, a ihned varovat uživatele. [1]

Ekonomické úspory inteligentnímu řízení a monitorování. Jak už bylo zmíněno u některé z předešlých výhod, je možné dosáhnout výrazných úspor za energie, protože se využívá pouze potřebná energie. Výrazně klesnou náklady na vytápění a na spotřebu elektrické energie. Zároveň je možné díky monitorovacímu systému sledovat spotřebu a omezit plýtvání.[1]

Systém inteligentních domů má kromě spousty výhod i jednu zásadní nevýhodu a to, že je velké nebezpečí hackerských útoků, protože většina systémů chytrých domácností už je běžně připojena k veřejné webové síti, proto se musí při výběru a realizaci dbát velký důraz na bezpečnost systému.

3 NÁVRH INTELIGENTNÍ ZAHŘÁDKY

Před zahájením návrhu je důležité stanovit si základní cíle, jejichž naplnění bude požadováno. V prvním kroku bylo přibližně rozhodnuto, co se od modelu inteligentní zahrádky očekává a dle toho byly vybrány HW komponenty.

Hlavní požadavky jsou:

- Nízké pořizovací náklady.
- Sběr dat ze vstupních senzorů (teplota, vlhkost, vlhkost půdy, osvětlení, hladina vody v nádrži), jejich zpracování a vyhodnocení.
- Dle nastavených parametrů a vyhodnocených podmínek spínat zalévání a osvětlení.
- Zobrazení okolních podmínek ve webovém prohlížeči na celosvětové webové síti.
- Možnost online sledování květin pomocí webkamery.

Vzhledem k hlavním požadavkům byly vybrány komponenty, které jsou cenou zcela na jiné úrovni než prvky, které jsou určeny výhradně k účelům inteligentního řízení domů (budov).

Jako hlavní řídicí prvek byl zvolen jednodeskový počítač Raspberry Pi 2 Model B, který bude sloužit jako webový server. K Raspberry Pi bude připojen pomocí UART vývojový kit Arduino UNO, který bude sbírat data ze vstupních senzorů a dále bude ovládat vodní čerpadlo a osvětlení. Arduino bude v určitých cyklech odesílat získaná data ze senzorů po UART do zařízení RPi, které je následně bude ukládat do databáze. Data budou moci být zobrazována pomocí webového rozhraní.

4 RASPBERRY

V této kapitole je představen výpočetní modul Raspberry Pi [2]. Kapitola se věnuje stručným popisem historie RPi, dále jsou vypsány jeho možnosti, také je zde uveden možný operační systém pro RPi.

Raspberry Pi je plnohodnotný, jednodeskový, miniaturní počítač, který byl vyvinutý britskou nadací Raspberry Pi Foundation s cílem podpory výuky informační techniky na středních školách. Odborníci navrhovali RPi s vizí malého programovatelného počítače se snadno dostupnými a ovladatelnými periferiemi, tak aby se žáci software naučili ne jenom vyvíjet, ale aby také pochopili hardware a dokázali poznat jakým způsobem je počítač postaven a jak funguje. Cíl byl tedy vytvořit malý, levný počítač, aby měl možnost každý žák pracovat na svém výpočetním modulu. Ve své snaze uspěli – Raspberry Pi model B, počítač s velmi malými rozměry (kreditní karta) určený pro běh optimalizovaných operačních systémů na bázi GNU/Linux se na trhu objevil 29. února 2012 a byl ve své době revoluční především poměrem ceny a výkonu. Za US\$35 bylo možné pořídit počítač osazený SoC Broadcom BCM2835 integrující procesor ARM11 taktovaný na frekvenci 700 MHz a grafický procesor schopný dekódovat video ve FullHD kvalitě v reálném čase. Na desce je k dispozici také operační paměť o velikosti 512 MB a různorodé periferie a sběrnice, které mají za úkol zpopularizovat výuku informatiky. O rok později, 4. března, Raspberry Pi Foundation uvedla na trh model A, oproti předchozímu modelu má poloviční velikost operační paměti, jeden USB port místo dvou a Ethernetové rozhraní nebylo integrováno. Touto redukcí hardwaru se snížila cena modelu na US\$25 a klesla energetická spotřeba modulu z 3,5 W na 2,5 W.

Mezitím se Raspberry Pi stalo ve světě populární. Splnilo původní záměr v atraktivní výuce informatiky a hlavně pak příznivci Raspberry Pi využívají tento miniaturní počítač pro všechno možné od domácího serveru, přes řídicí jednotku robota, quadrokoptéry či domovní automatizace až k metrologickým stanicím umístěným porůznu po světě, automatizovaným časosběrným fotoaparátům či multimediálním centrům. Raspberry Pi se dokonce svými rozměry, cenou, univerzálností, možnostmi vývoje a otevřeností linuxového prostředí uplatnilo v průmyslu (reakcí nadace Raspberry Pi Foundation je otevřená platforma Raspberry Pi Compute Module uvedená na trh v dubnu 2014 za US\$200, jež návrh a vývoj software a hardware usnadňuje). Další změnou je větší počet GPIO pinů oproti uživatelskému RPi, modul je celkově menší a má 4GB eMMC Flash paměť integrovanou přímo na čipu atd.

Dne 15. července 2014 vydala nadace revizi modelu B s označením model B+. Uživatelé nabízejí za cenu shodnou s modelem B dva USB porty navíc, patičku GPIO rozšířenou o čtrnáct pinů (z 26 na 40), nižší spotřebu a operační systém se nově nahrává z MicroSD karty oproti SD kartě

v předchozích dvou verzích. Počet prodaných kusů všech modelů Raspberry Pi ke konci února roku 2015 překonal číslo 4,5 miliónů.

Raspberry Model „B“ [3] se začal prodávat v únoru 2015. Jeho základem je SoC BCM2836 z rodiny ARM Cortex-A7 opět od firmy Broadcom, který obsahuje čtveřici procesorových jader s taktem 900 MHz, posílenou jednotkou SIMD a 1 GiB paměti RAM. Osazen je grafický procesor VideoCore IV, podporující OpenGL ES 2.0, 1080p30, MPEG-4. Stejně jako jeho předchůdci obsahuje slot pro microSD kartu, čtyři porty USB 2.0, ethernetový adaptér 10/100 Mbps s konektorem RJ-45 a vývody GPIO.

V Únoru 2016 byl uveden do prodeje model Raspberry Pi 3 [3], který je vybaven 64 bitovým CPU o taktu 1,2GHz, Procesor je 4-jádrový ARM Cortex-A53 a dle výrobce je o 50% rychlejší, než CPU v modelu Raspberry Pi2. Stejně jako Model 2 je vybaven 1GB operační pamětí. Nov2 jsou integrované WIFI a Bluetooth moduly. Rozmístění konektorů se oproti modelu RPi 2 Model B nemění.



Obrázek 1: Modely Raspberry Pi [4]

4.1 Parametry Raspberry

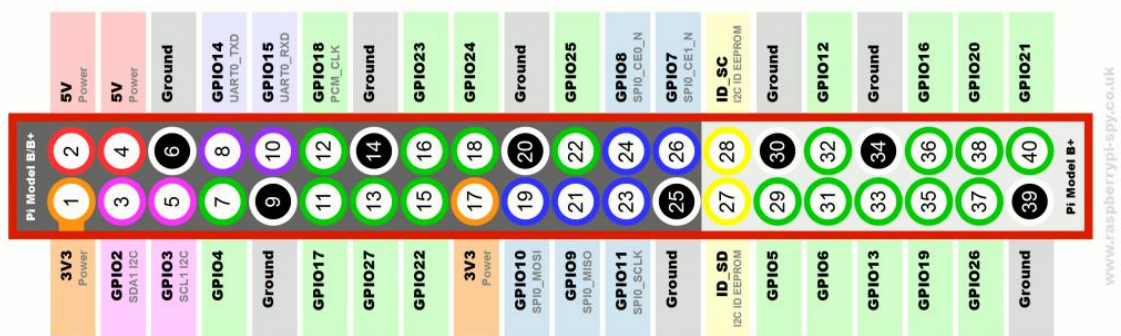
V tabulce (Tabulka 1) jsou uvedeny parametry všech modelů RPi, v této práci je použito Raspberry Pi 2 Model B.

	model A	Model B	Model B +	Model B2	Model B3
System on a chip	Broadcom BCM2835 (CPU + GPU + DSP + SDRAM) [5]			Broadcom BCM2836 (CPU + GPU + DSP + SDRAM + jeden USB port)	Broadcom BCM2837
CPU:	700 MHz ARM1176JZF-S jádro (ARM11 rodina)			900 MHz quad-core ARM Cortex-A7	1 GHz quad-core ARM Cortex-A53 64 Bit
GPU:	Broadcom VideoCore IV , OpenGL ES 2.0, 1080p 30 H.264 / MPEG-4 AVC high-profile dekódování				
Memory (SDRAM)	256 MB (sdílená s GPU)	256 nebo 512 MB (sdílená s GPU)	512 MB (sdílená s GPU)	1 GB (sdílené s GPU)	1GB LPDDR2 (900 MHz) (sdílené s GPU)
Porty USB 2.0:	1	2 (přes USB hub)	4 (přes USB hub)		
Video výstupy:	RCA Composite (PAL aNTSC), HDMI (verze: 1.3, 1.4)		RCA Composite (PAL a NTSC) pro 4-pin (TRRS) konektoru jack , HDMI(verze: 1.3 a 1.4)		RCA Composite (PAL a NTSC) pro 4-pin (TRRS) konektor jack ,HDMI
Audio výstup:	3,5 mm jack , HDMI				3,5 mm jack , HDMI
Paměťová média:	Konektor karty SD / MMC /SDIO		MicroSD		
Připojení k síti:	nedostatek	10/100 Ethernet (RJ45)			10/100Ethernet (RJ45) , Wi-Fi (2,4 802.11n), Bluetooth
GPIO:	28 x GPIO		40 x GPIO		
Napájení:	500 mA (2,5 V)	700 mA (3,5 W)	600 mA (3,0 W),	900 mA (3,1),	2.5A
Napájecí zdroj:	5 V s konektorem microUSB , případně přes konektor GPIO				
rozměry:	85.60 x 53,98 mm				85.60 x 53,98 mm
hmotnost:	31 g	40 g			
Podporované operační systémy:	Raspbian , Debian GNU / Linux ,Fedora , Arch Linux ,FreeBSD			Raspbian , Windows 10 IoT Jádro, Debian GNU / Linux, Ubuntu prostořeký jádro OpenELEC, MATE Ubuntu,Fedora , Arch Linux,FreeBSD	Raspbian , Windows 10 IoT Jádro OSMC_Pi2, Noobs, RISC OS Ubuntu MATE

Tabulka 1: Parametry modelů RPi

4.2 GPIO piny Raspberry 2 Model B

Raspberry Pi 2[3] mimo složitějších rozhraní také disponuje čtyřiceti programovatelnými GPIO (General-Purpose Input/Output) piny uspořádanými v patici P1. Rozložení pinů v patici je 2x20 se standardní roztečí 2,54 mm. GPIO pinů bez alternativní funkce je sedmáct, zbylé piny realizují SPI, I2C a UART rozhraní, PWM modulaci a napěťové výstupy +3,3 V, +5 V a GND. Pokud programátor nebude využívat alternativní funkce pinů, které popisuje tabulka níže, má celkem k dispozici dvacet šest GPIO pinů. Piny nejsou 5 V tolerantní a jakékoli napětí vyšší než 3,3 V může nenávratně poškodit celé zařízení.



Obrázek 2: Rozmístění GPIO pinů Raspberry Pi 2 Model B [5]

4.3 OS Raspberry Pi

Všechny modely RPi jsou určeny pro běh operačního systému, většinou na bázi GNU/Linuxu, a na většinu z nich je možné nainstalovat přinejmenším jeden ze systémů zde jmenovaných. Systémy se liší zejména v míře optimalizace pro danou architekturu procesoru. Níže jsou popsány operační systémy spolu s jejich hlavními klady a zápory (OS jsou nabízené distributorem Raspberry Pi a jsou ke stažení na oficiálních stránkách RPi).

NOOBS (New Out of Box Software) [2] – nejedná se o skutečný operační systém, ale o zaváděcí program zajišťující snadnou instalaci různých OS a jejich vyzkoušení. Je možné vybrat ze šesti OS a během každého spuštění Raspberry Pi si uživatel může vybrat jiný operační systém. OS nabízené aplikací NOOBS jsou následující:

- Raspbian
- Pidora
- OpenELEC
- RaspBMC

- RISC OS
- Arch Linux

Raspbian [2] – derivace Debianu, verze Wheezy. Raspbian je nejpoužívanější a doporučovaný nadací Raspberry Pi Foundation, z tohoto důvodu stojí za Raspbianem obrovské množství uživatelů, kteří již vyřešili na veřejných fórech mnoho problémů, na které může uživatel při používání tohoto OS narazit. Podobně – jako Debian – je prezentován jako stabilní systém vhodný pro dlouhodobý provoz. Tento operační systém byl vybrán k realizaci této práce.

OpenELEC [2] – XBMC Mediacenter distribuce je ořezaný OS vhodný pro použití Raspberry Pi jako multimediálního centra. K přehrávání video souborů v kompresi MPEG-2 nebo VC-1 je nutné zakoupit kodeky, které jsou vázány na sériové číslo Raspberry Pi, a proto nejsou přenositelné.

RISC OS [2] – GUI prostředí optimalizované pro běh při rozlišení 1080p kompilované pro architekturu ARM týmem lidí, kteří původně navrhli ARMové procesory. Nejedná se o derivaci linuxu.

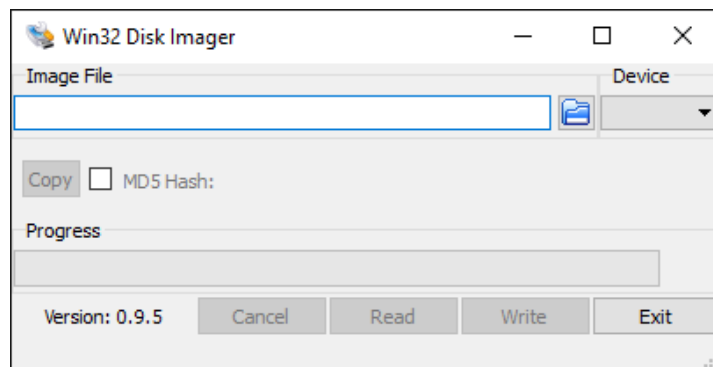
Arch Linux – Distribuce Arch Linux zkompileovaná pro ARM zařízení, je vhodná pro zkušenější uživatele, a těm umožňuje přizpůsobit si systém bez nadbytečných součástí. Rozsáhlý seznam operačních systémů přizpůsobených pro Raspberry Pi je k nahlédnutí na stánkách.

4.4 Instalace Raspbianu

Ke spuštění a instalaci systému na Raspberry Pi je potřeba toto:

- Raspberry Pi
- Paměťovou kartu SDHC (doporučená je minimálně 4 GB)
- Čtečka SDHC karet (nyní součástí většiny Notebooku)
- Napájecí zdroj s konektorem micro – USB, 5V, 700 mA. Jako zdroj může sloužit USB port u PC, avšak poskytuje nanejvýš 500 mA
- Připojení k internetu

Instalační image OS Raspbian je možné stáhnout v sekci Download na stránkách <http://www.raspberrypi.org/>. Tento instalační balík je nainstalován na SD kartu například pomocí programu Win32DiskImager (obrázek 4) , který je možné stáhnout na adrese <https://sourceforge.net/projects/win32diskimager/>.



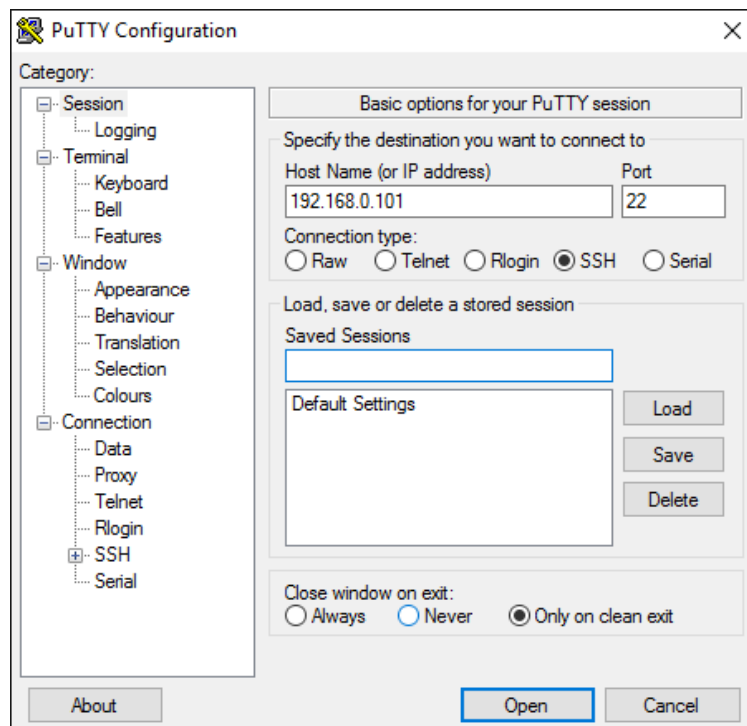
Obrázek 3: Win32DiskImage

4.5 Spuštění Raspbianu pomocí Putty

Po připojení Raspberry pomocí síťového kabelu do sítě je vytvořena lokální síť a je možné se pomocí protokolu SSH (Secure Shell) připojit k RPi. Protokol SSH lze realizovat například pomocí programu Putty (Obrázek 5). Do Host Name je vyplněna adresa zařízení, kterou lze zjistit v případě zapojení do místní sítě (routeru) např. pomocí příkazu v cmd `arp -a`, který nám ukáže tabulku připojených zařízení k routeru a jejich IP adresy. Lze také nastavit hodnotu IP adresy před instalací a to konkrétně když je zařízení připojeno přímo do laptopu, na SD kartě do souboru `cmdline.txt` a to:

```
ip= 192.168.0.100:255.255.255.0
```

SSH protokol využívá port 22. Spojení je provedeno stisknutím tlačítka Open.



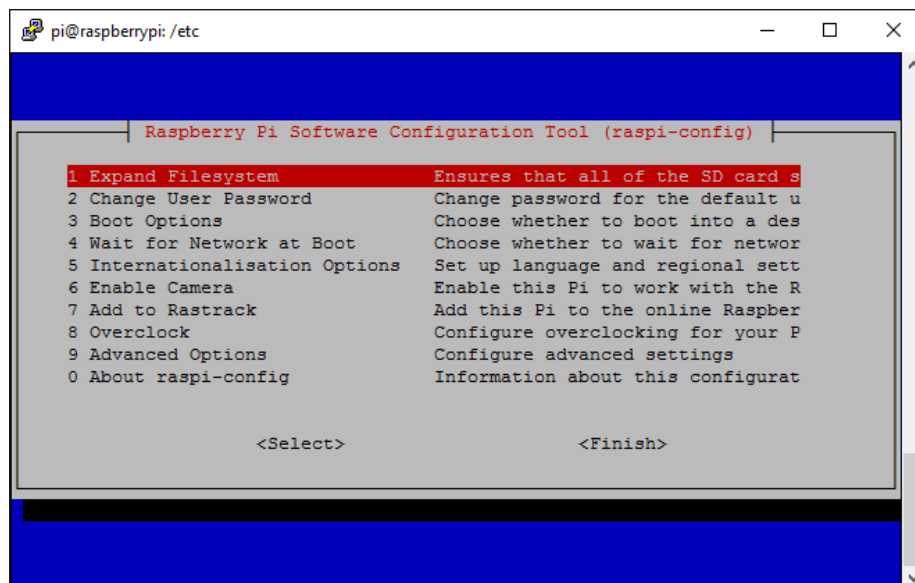
Obrázek 4: Program Putty

Dalším krokem je výzva k vyplnění přihlašovacích údajů, které jsou defaultně nastaveny takto:

Uživatelské jméno: pi
Heslo: raspberry

Pro možnost přistupování k zařízení pomocí veřejné adresy a tudíž umožnění přístupu odkudkoli z veřejné sítě byl požádán poskytovatel internetového připojení o přesměrování portů 22 a 8081. Kde port 22 je připojení k SSH a port 8081 je webový server Apache.

Po prvním spuštění se doporučuje spustit příkaz `sudo raspi-config` (Obrázek 8) který umožňuje základní nastavení systému. V případě, že je zařízení připojováno do veřejné sítě, jako to je v této práci, doporučuje se změnit uživatelské heslo a to v sekci „*Change User Password*“. V této práci bylo změněno na heslo: garden.

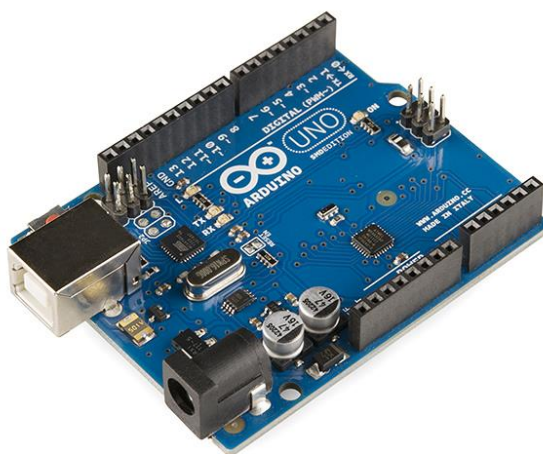


Obrázek 5: Program raspi-config

5 OSTATNÍ HW

5.1 Arduino

Arduino UNO [6] patří mezi volně dostupné platformy pro elektrické modelování založené na flexibilním a jednoduše použitelném softwaru a hardwaru. Platforma Arduino obsahuje modul s mikroprocesorem i vlastním vývojovým prostředím, které využívá vyšší programovací jazyk C++. Moduly jsou založené na mikroprocesorech od společnosti Atmel.



Obrázek Arduino Uno [7]

Arduino může být napájeno před USB připojení nebo s externím napájením, kdy je napájecí zdroj vybrán automaticky. Externí napájení může pocházet z AC-DC adaptéru nebo baterie. Platforma je napájena napětím 6-20 V. Pokud dodáváme méně než 7 V, může pin 5 dodávat méně než 5 V a deska se může stát nestabilní. Pokud dodáváme více jak 12 V, může regulátor napětí způsobit přehřátí a poškození desky. Proto doporučený rozsah je mezi 7 V a 12 V.

Napájecí piny pro napájení:

- Vin vstupní napětí na Arduino desce, pokud se použije externí zdroj napájení
- 5V napájecí napětí 5 V,
- 3V3 napájecí napětí 3,3V,
- maximální odběr proudu 50mA
- GND zem ,

- IOREF správně nakonfigurovaná deska může přečíst IOREF pin napětí a vybere si vhodný zdroj napájení nebo povolí napětí na výstupech pro práci s 5 V nebo 3,3 V.

Paměť

Ve vývojovém kitu Arduino Uno nalezneme mikroprocesor ATmega 328, který má paměť o velikosti 32 KB.

Vstupy a výstupy

- RX a TX používají se pro příjem (RX) a přenos (TX) TTL dat. Jsou připojeny k odpovídajícím pinům mikroprocesoru.
- IRQ vnější přerušení s označením 2 a 3, tyto piny mohou být konfigurovány ke spuštění, přerušení na nízkou hodnotu, na náběžnou nebo sestupnou hranu, nebo ke změně hodnoty.
- PWM (pulzně šířková modulace) piny: 3,5,6,9,10 a 11 poskytují 8 bitový PWM výstup s analogovou hodnotou. PWM se často používá pro řízení výkonu motorů, v pulzních zdrojích a někdy také v audio technice pro různé zvukové efekty.
- SPI (sériové periferní rozhraní): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Tyto piny podporují SPI komunikaci pomocí knihovny SPI.
- LED 13, je vestavěný LED připojení pro digitální pin 13, pokud se na pinu vyskytuje vysoká hodnota - dioda svítí.
- Arduino UNO má 6 analogových vstupů, označených jako A0 až A5, z nichž každý poskytuje 10 bitů rozlišení (1024 různých hodnot).

Komunikace

Arduino Uno má řadu zařízení pro komunikaci s počítačem. ATmega328 disponuje TTL (5V) sériovou komunikací, která je vyvedena na digitálních pinech 0 (RX) a 1 (TX). Integrovaný mikročip FTDI FT232RL zajišťuje propojení komunikačního rozhraní s rozhraním USB a ovladač FTDI, který je součástí vývojového prostředí Arduino, poskytuje virtuální sériový port pro software na počítači.

Nadproudová ochrana

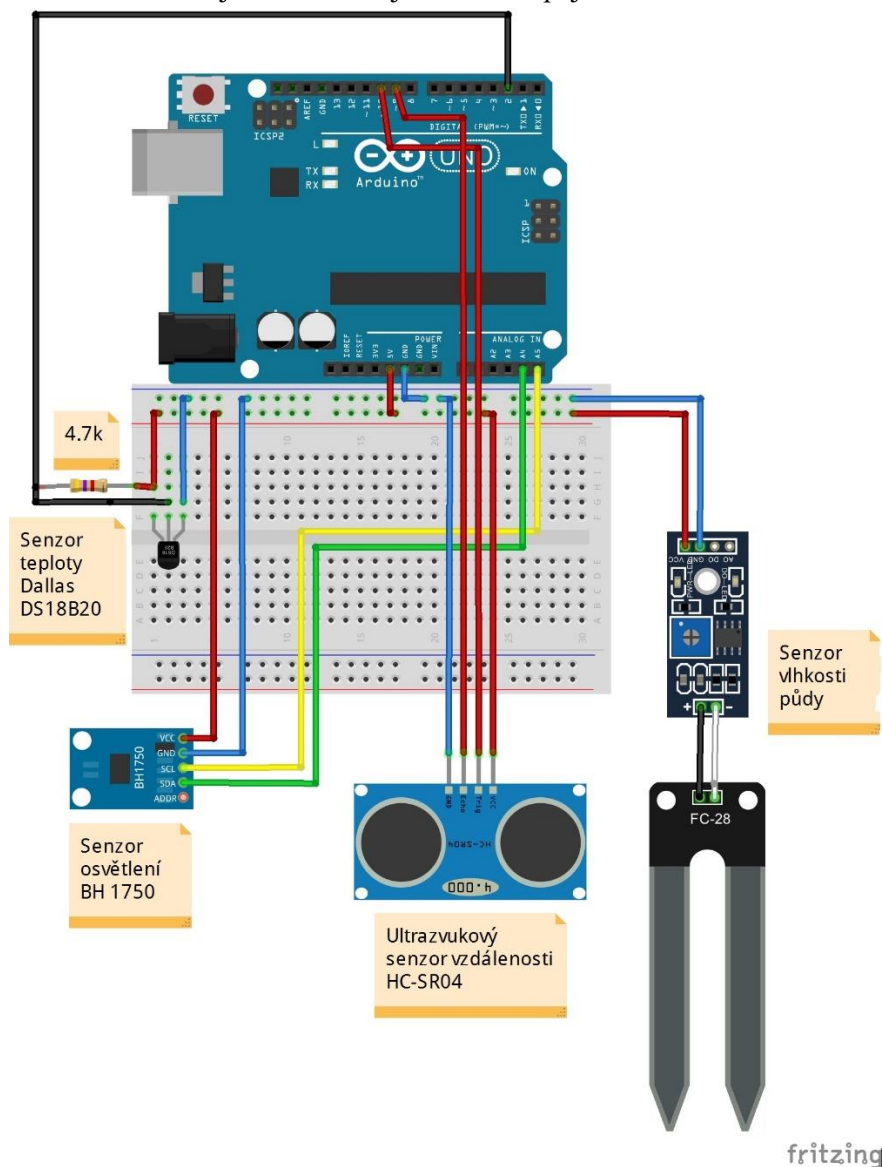
Arduino Uno má vratnou pojistku, která chrání počítač, při připojení desky přes USB. Ačkoliv většina počítačů poskytuje svojí vlastní vnitřní ochranu, pojistka poskytuje další vrstvu ochrany. Pokud je více než 500 mA aplikováno do USB portu, pojistka automaticky ukončí spojení.

Programování

Modul Arduino Uno je programovatelný pomocí vývojového prostředí Arduino, které nalezneme na stránkách výrobce (www.arduino.cc). Použitý mikroprocesor Atmega328 má již předpřipravený podprogram pro zavedení startovacího kódu, který umožňuje nahrát zdrojový kód do mikroprocesoru bez nutnosti použití externího zařízení pro programování. Podle protokolu STK500 probíhá komunikace při nahrávání zdrojového kódu. Podprogram pro zavedení startovacího kódu je možné přemístit a programovat tak procesor Atmega168 pomocí rozhraní ISCP.

5.2 Senzory

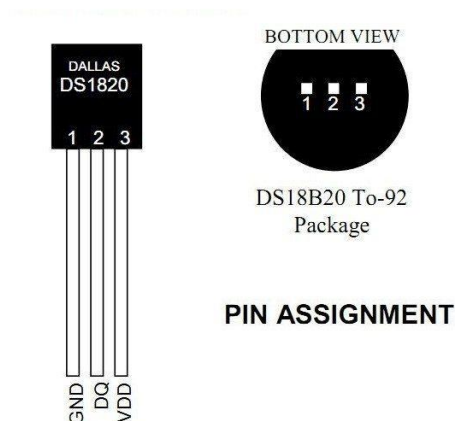
Veškeré senzory obsluhuje Arduino. Na Obrázek 6: Zapojení senzorů k Arduino. Obrázek 6: Zapojení senzorů k Arduino je znázorněno jednotlivé zapojení senzorů.



5.2.1 Senzor teploty Dallas DS18B20

Digitální čidlo teploty Dallas DS18B20 [8] poskytuje 9-bitové až 12-bitové měření teploty. DS18B20 komunikuje přes sběrnici 1-Wire které ze své podstaty vyžaduje pouze jednu datovou linku (a zem) pro komunikaci s mikroprocesorem. Má provozní teplotní rozsah -55°C až $+125^{\circ}\text{C}$ a s přesností $\pm 0,5^{\circ}\text{C}$ v rozsahu -10°C až $+85^{\circ}\text{C}$.

Čidlo obsahuje dva oscilátory, z nichž jeden je s nízkým teplotním koeficientem a druhý naopak s vysokým teplotním koeficientem. Jakmile přijme po sběrnici čidlo příkaz k zahájení měření teploty, spustí se čítání obou těchto oscilátorů. Čítání těchto pulzů trvá dobu úměrnou rozlišení čidla a pro 12 bitové rozlišení je to 750 ms. Po ukončení tohoto měření je v čítacím registru 12 bitová hodnota odpovídající aktuální teplotě. Ta je poté uložena ve vnitřních registrech obvodu a je připravena na vyčtení. Datový vývod je připojen do Arduina k pinu č.2.

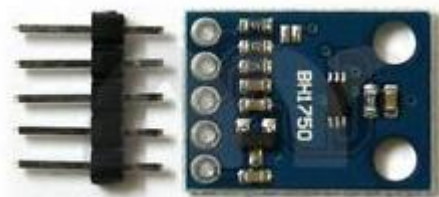


Obrázek 7: Senzor teploty Dallas DS18B20 [9]

5.2.2 Digitální senzor intenzity světla BHI1750FVI

Digitální světelný senzor BHI1750FVI [10] komunikující přes I2C sběrnici, na Arduino jsou to piny A4 (SCL) a A5 (SDA). Adresu senzoru lze nastavit hardwarově pomocí vstupního pinu ADDR na jednu ze dvou variant. Výstupem je intenzita osvětlení v jednotkách lx. Rozsah měření je od 1 lx do 65 535 lx. Rozlišení je nastavitelné od 0,5 lx do 4 lx. Pracovní napětí je od 2,4 V do 3,6 V. Problémem tohoto senzoru je malý rozsah pro měření intenzity slunečního svitu, kde horní hranice je navíc snížena vlivem přesnosti měření na pouhých 54 612 lx ($= 65\,535/1,2$). Intenzita osvětlení přímého slunce je větší než 100 000 lx. Tento senzor je také možné použít

pouze pro učení doby přímého slunečního svitu. Je nutné však experimentálně nalézt vhodnou mezní hranici.



Obrázek 8: Světelný senzor BH1750FVI [11]

5.2.3 Ultrazvukový senzor vzdálenosti HC – SR04

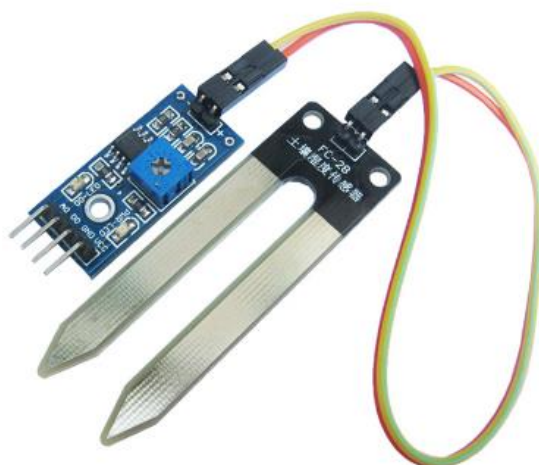
HC-SR04 [12] je kompaktní modul, který nepotřebuje ke své funkci žádné další součástky. Obsahuje dva ultrazvukové měniče (vysílač, přijímač) a integrovanou elektroniku. Umožňuje měřit vzdálenosti od 2 cm do 4 m. Má celkem 4 piny, napájení, zem, vstup a výstup. Měření probíhá následovně, na vstupní pin Trig se přivede pulz o šířce 10 μ s, čímž modul vyšle signál o frekvenci 40 kHz. Po návratu signálu se na výstupním pinu Echo objeví pulz, jehož šířka udává dobu letu vyslaného signálu. Vypočet vzdálenosti je poté vypočítán softwarově za pomoci známé hodnoty rychlosti zvuku při teplotě 25°C, což je 346,3 m/s což je 0,03463 mm/s a protože ultrazvuk urazí cestu dvakrát, je tato hodnota ještě vydělena 2.



Obrázek 9: HC - SR04 [13]

5.2.4 Čidlo vlhkosti půdy

Čidlo vlhkosti má dva výstupy, jeden označený "A0" a jeden označený "D0". "A0", je sériový signál 0-5 voltů, který je na vstupu arduina převeden na digitální hodnotu 0 – 1024. Při úplné suchosti půdy, v které se nachází snímač je na výstupu 5 V, když je snímač zcela mokrá, tak je na výstupu nulové napětí. Sonda je tvořena dvěma jazýčky, mezi kterými působí půda jako proměnný odpor. Čím více vody, tím je vodivost mezi jazýčky lepší.



Obrázek 10: Senzor vlhkosti půdy [14]

5.3 Osvětlení, čerpadlo

Jak osvětlení, tak i čerpadlo jsou napájeny 12 V a jsou spouštěny pomocí relé, které jsou ovládány pomocí digitálních 5 V výstupů arduina.

5.3.1 LED osvětlení

Jako osvětlení je použit UV LED pásek 60 SMD LED 5050 s příkonem 14,4 W/m a vlnovou délkou 405 nm. S touto vlnovou délkou se vydávané záření nejvíce blíží UVA záření. Asi 99 % UV záření, které dopadne na zemský povrch, je ze spektrální oblasti UV A. Diody jsou spínány digitálním výstupem arduina č. 8.

5.3.2 Čerpadlo Barwig

K zalévání slouží 12 V čerpadlo Barwig, s výkonem 12 l/min při 0,6 barech. Čerpadlo je spínáno pomocí digitálního výstupu arduina č. 7.



Obrázek 11: 12 V čerpadlo

5.3.3 Arduino relé 2 kanály

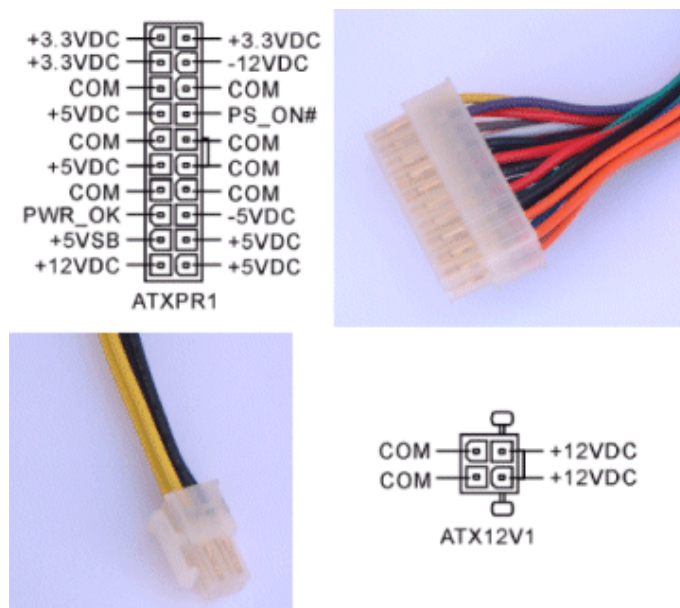
Pro spínání 12 V pro čerpadlo a osvětlení bylo použito 2-kanálové relé spínané 5 V s požadovaným buzením 15 – 20 mA. Výstup relé je až 250 V AC 10 A.



Obrázek 12: Arduino relé 2 kanály [15]

5.4 Zdroj

Nedílnou součástí systému je jeho napájení. Pro tento model byl použit počítačový ATX12V zdroj EC 400 W, model 300XXv2. Byly použity 5 V a 12 V vývody z konektorů Main Power a +12 Power (Dle přítomnosti tohoto pinu se pozná, zda je zdroj ATX nebo ATX12V). Pro zprovoznění zdroje bylo třeba spojit na konektroou Main Power zelený pin "PS_ON#" s pinem GND. Úpravou zdroje byl získán Stabilizovaný zdroj napětí 12 a 5 V.



Obrázek 13: Konektory ATX zdroj [16]

5.5 Webkamera

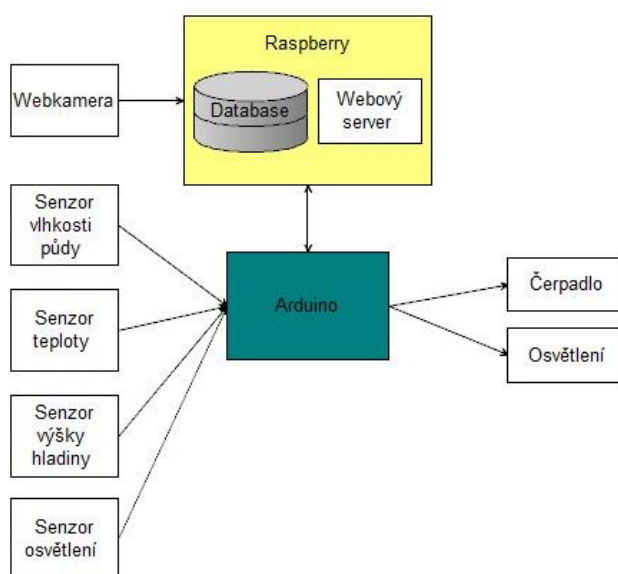
Aby byla zajištěna možnost náhledu na zahrádku skrz web byla použita webkamera Genius FaceCam 320 na USB. Byla vyzkoušena i kamera pro RPi NOiR, avšak nastal problém s flexi kabelem, proto byla vyměněna.



Obrázek 14: Webkamera FaceCam 320 [17]

6 SW VYBAVENÍ

Raspberry Pi je hlavní výpočetní zařízení celé inteligentní zahrádky. Toto zařízení se stará o komunikaci s Mikrokontrolerem Arduino, od kterého získává data, která RPi ukládá do MySQL databáze, jež jsou interpretovány na webu. Dále je k RPi připojena webkamera, pomocí které je po požadavku z prohlížeče uživatelem spuštěna funkce photo.py vytvářející aktuální fotografii. Fotografie je zobrazena na webu. Schématické zapojení všech prvků je znázorněno na Obrázku 18.



Obrázek 15: Schématické propojení komponent

Tato kapitola se zabývá programovým vybavením jednotlivých modulů. V první části jsou popsány prostředky použité pro návrh a realizaci systému. V další části je popsána komunikace mezi RPi a Arduinem. Dále následuje popis programu pro samotné Arduino a poté pro Raspberry Pi. V poslední kapitole je popsán webový server spolu s MySQL databází a webovým rozhraním HTML.

6.1 Použité prostředky pro realizaci systému

Pro návrh a realizaci celého řídicího systému pro inteligentní zahrádku byly použity následující programy a vybavení.

Kód programu pro Arduino byl vytvořen v programovacím jazyce C s využitím vývojového prostředí Arduino IDE.

Jako programovací jazyk pro Raspberry Pi byl zvolen jazyk Python. Tvorba programů probíhala v textovém editoru PSPad. Dále pak jako webový server byl zvolen Apache 2. Programové vybavení webového rozhraní v jazycích HTML, PHP, JavaScript a vše bylo taktéž tvořeno v editoru PSPad. Jako databáze pro ukládání dat byla zvolena MySQL.

6.2 Webový server

Veškerá data získaná ze senzorů jsou zobrazována skrz webové rozhraní a to jak na lokální síti, tak i veřejné. Z tohoto důvodu byl použit webový server Apache 2, dále pro ukládání dat byla použita databáze MySQL a pro lepší zprávu databáze byl nainstalován program PhpMyAdmin. Pro správu systému skrze webové rozhraní byl nainstalován program Webmin. Zařízení je připojené k routeru TP Link WR 940-N a bylo třeba nastavit virtuální server tak, aby bylo zařízení přístupné z veřejné sítě.

6.2.1 Apache 2

Webový server apache se nachází v repozitáři Raspbianu a proto je instalace spuštěna jednoduše pomocí apt-get. Jako první je vhodné aktualizovat repozitář pomocí příkazu:

```
sudo apt-get update
```

Apache je poté nainstalován pomocí příkazu:

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

Po instalaci je nutné restartovat systém pomocí příkazu:

```
sudo reboot
```

Po instalaci je možné nainstalovat podporu skriptovacích jazyků PHP a Python. Potřebné balíky se taktéž nacházejí v archivu a nainstalují se pomocí příkazů:

```
sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt  
sudo apt-get install python-dev
```

Ověření funkčnosti nainstalovaného serveru je možné zadáním url adresy localhost do webového prohlížeče. Zobrazí se Apache2 Debian, výchozí stránka sloužící právě k testování správné funkčnosti. Vlastní html stránky jsou poté vkládány do adresáře `/var/www/html/`. Apache server interpretuje všechny soubory v tomto adresáři.

6.2.2 MySQL

Databáze je postavena na databázovém systému MySQL nyní vyvíjeném firmou Oracle. MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí

jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

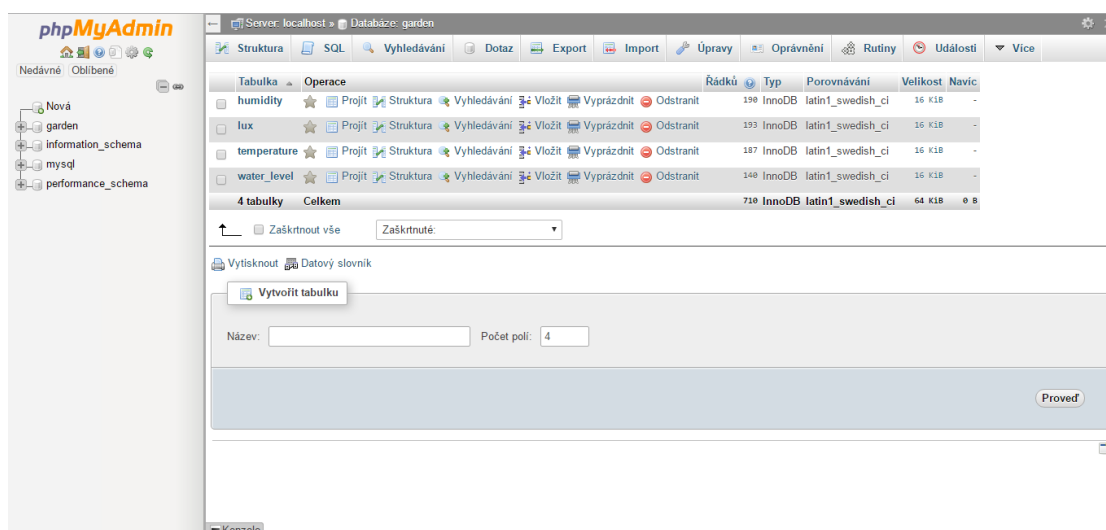
garden humidity id : int(11) value : double time : datetime	garden lux id : int(11) value : double time : datetime	garden water_level id : int(11) value : double time : datetime	garden temperature id : int(11) value : double time : text
---	--	--	--

Obrázek 16: Databáze MySQL

Ukládání dat do databáze je prováděno za pomoci funkcí programu db.py. Vyčítání hodnot z databáze a následné zobrazení na webovém rozhraní probíhá přímo v php scriptu webové stránky.

6.2.3 PhpMyAdmin

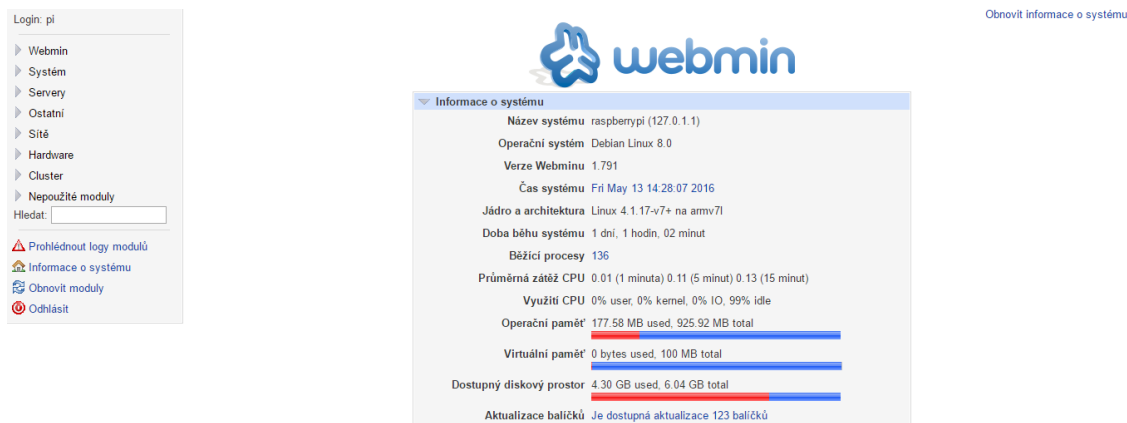
Díky tomuto programu, který je nahrán na straně serveru je možné upravovat databáze, mazat, vytvářet nové tabulky nebo zpravovat klíče. Do tohoto programu je případě této práce přistupováno z lokální sítě díky adrese <http://192.168.0.101/pma/>.



Obrázek 17: Program PhpMyAdmin

6.2.4 Webmin

Program Webmin je určený ke kompletní správě systému přes webové rozhraní. Používá rozšíření, která jsou nadefinována pro správu jednotlivých aplikací. Při správě serveru Apache 2 se velice osvědčil.

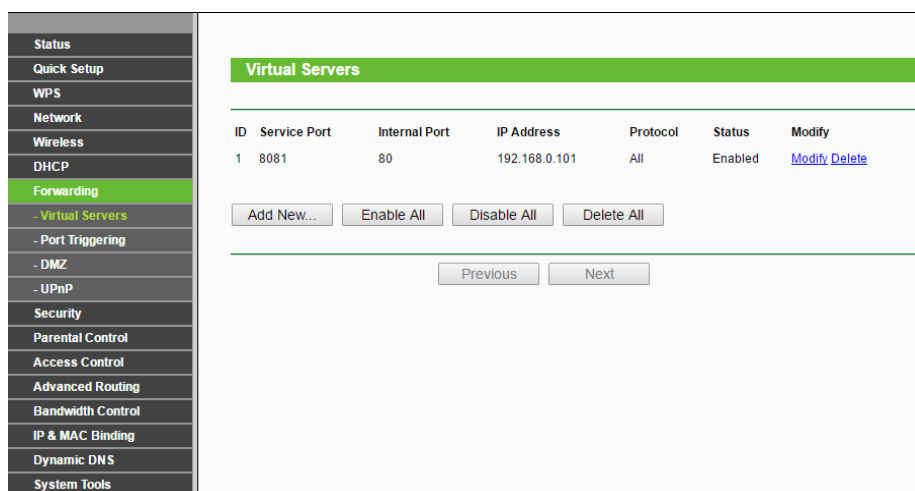


Obrázek 18: Program Webmin

Tento program je možné spouštět na lokální síti, adresa je “ip zařízení RPi”:10000. V případě této práce je to 192.168.0.101:10000. Pro to aby bylo možné k tomuto program přistupovat pomocí veřejné adresy by bylo třeba přesměrovat port 10000, což musí umožnit poskytovatel internetového připojení.

6.2.5 Nastavení routeru TP-link TL-WR940N

Nastavení virtuálního serveru umožňuje přístup z veřejné sítě k webovému serveru. Virtuální server je definován jako servisní port a všechny požadavky z internetu k tomuto portu budou přesměrovány na počítač (v tomto případě RPi) určený IP adresou serveru. V tomto případě bude požadavek z veřejné sítě pod adresou 84.21.104.214:8081 (8081 číslo portu) přesměrován na adresu 192.168.0.101. Interní port je číslo portu interního zařízení, na kterém aplikace běží. V tomto případě RPi.

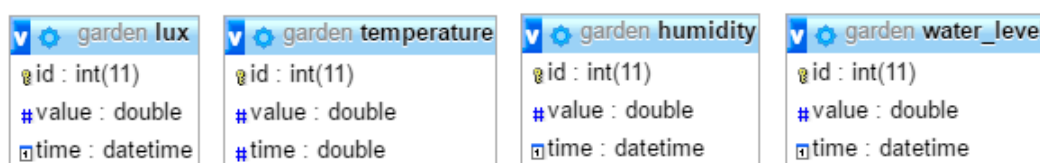


Obrázek 19: Nastavení virtuálního serveru v routeru TP link

6.3 Databáze

Databáze garden byla vytvořena pomocí programu PhpMyAdmin, skládá se ze 4 tabulek, pro ukládání hodnot z čidel a to konkrétně lux (osvětlení), temperature (teplota), humidity (vlhkost půdy) a water_level (výška hladiny v nádrži).

Každá tabulka se skládá ze 3 hodnot. Z ID což je primární klíč, který je automatický inkrementován a slouží pro číselné označení každého záznamu. Do dalšího sloupce tabulky jsou ukládány hodnoty přečtené z čidel jako datový typ double a poslední sloupec obsahuje přesný čas kdy byly data vloženy.



garden lux	garden temperature	garden humidity	garden water_level
id : int(11)	id : int(11)	id : int(11)	id : int(11)
value : double	value : double	value : double	value : double
time : datetime	time : double	time : datetime	time : datetime

Obrázek 20: Databáze garden

6.4 Komunikace pomocí sériového portu

Komunikace mezi Arduinem a Raspberry probíhá tak, že RPi posílá instrukce Arduino a to následně vykonává zadanou činnost a v případě požadavku na hodnoty z čidel odesílá zpět požadovaná data.

Existuje mnoho způsobů jak propojit Raspberry Pi a Arduino. V tomto projektu byly zařízení spojeny pomocí sériového portu, který je emulovaný přes USB. K tomuto spojení je třeba pouze USB kabel s konektory USB 2.0 A a USB B.

Pro správnou komunikaci musí být nastaven stejný baudrate, v případě této diplomové práce bylo použito 9600 b/s.

Raspberry Pi – komunikace pomocí sériového portu

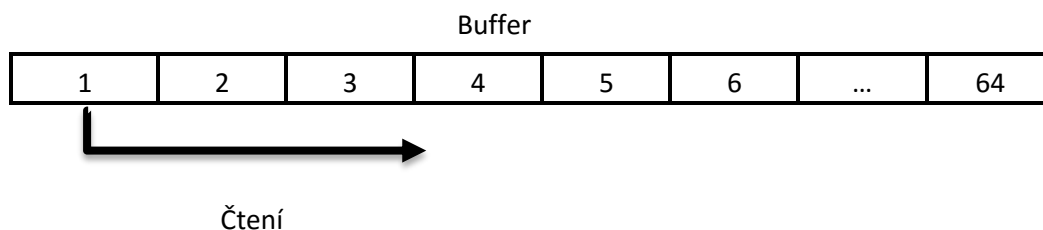
Pro snadnější práci se seriovým portem byl nainstalován do RPi modul PySerial [18].

```
get http://pypi.python.org/packages/source/p/pyserial/pyserial-2.6.tar.gz
tar -zxf pyserial-2.6.tar.gz
cd pyserial-2.6/
python setup.py build && sudo python setup.py install
```

V případě RPi je seriová komunikace v pythonu nastavena pomocí příkazu:

```
ser = serial.Serial('/dev/ttyACM0', 9600) #Nastavení komunikace po UART
```

Ke odesílání dat na sériový port je použita funkce `ser.write()`, kdy je odesílán znak, popřípadě sada znaků uvedených v závorce. Každý znak má hodnotu 1 bytu, celkově lze odeslat 64 bytu. K přijímání je použita funkce `ser.readline()`, která čte obsah celého bufferu, do kterého jsou ukládány data přicházející po sériové lince. V případě použití příkazu `ser.read()` je vždy přečten první znak, který je po přečtení výjmut z bufferu, a ostatní data se posouvají. Je velmi důležité před začátkem komunikace vyprázdnit buffer, aby nedošlo ke špatnému přenosu dat v případě, že jsou v bufferu nějaká data. K tomu slouží příkaz `ser.flushOutput()`. Stejně jako na začátku je dobré i po skončení komunikace buffer vyprázdnit.



Obrázek 21: Buffer sériové linky

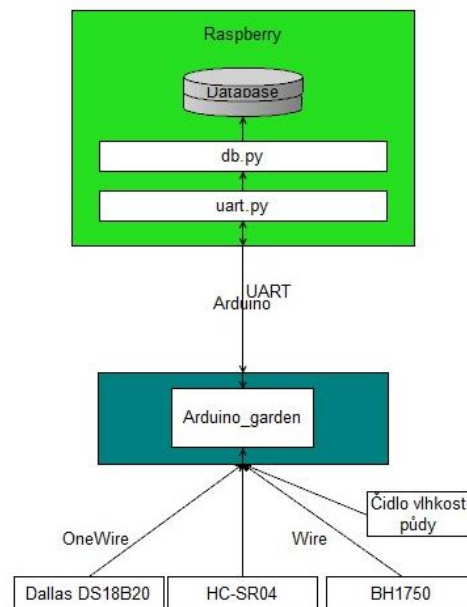
Arduino – komunikace pomocí sériového portu

V případě Arduina se nastavuje pomocí funkce `Serial.begin()`. Tato funkce se používá k zahájení sériové komunikace. Do závorek je psán parametr rychlosti této komunikace.

Pro odesílání dat slouží příkaz `Serial.print()`, tento příkaz odesílá po sériové lince znaky, které jsou převedeny na ASCII hodnoty. Příkaz `Serial.available()` říká, zda je v bufferu nějaká hodnota, konkrétně kolik je v bufferu znaků. Pro čtení dat přichozích dat z bufferu je použit příkaz `Serial.read()`. Stejně jako u RPi je také u Arduina důležité mít vyprázdněný buffer, což je prováděno příkazem `Serial.flush()`.

6.5 Sběr dat ze senzorů a zápis do MySQL

Pro sběr dat je využíván program v pythonu `uart.py`, který posílá po sériové lince dotazy do Arduina, které zasílá zpět data ze senzorů. Data jsou dále ukládána do MySQL databáze. Program `uart.py` je spouštěn pomocí Cronu spouštěn každých 15 min.



Obrázek 22: Software pro sběr dat ze senzorů a ukládání do databáze

6.5.1 Program na výčet hodnot z čidel Uart.py

Program `uart.py` slouží ke komunikaci po sériové lince s Arduinem. Konkrétně žádá Arduino o hodnoty ze senzorů.

V první části programu je nastavena sériová komunikace dle příkazu výše a v dalším kroku je sériový port připraven pro komunikaci:

```

time.sleep(0.2) # čekání než se seriová linka inicializuje
ser.flushOutput() # vyprázdnění bufferu
ser.flushInput()
ser.nonblocking() # přepnutí seriové linky do neblokujícího modu

```

Program poté posílá požadavky na jednotlivá data, která následně přijímá a ukládá do databáze pomocí příkazu v pořadí:

1. Požadavek na hodnotu osvětlení
2. Požadavek na hodnotu teploty
3. Požadavek na hodnotu výšky hladiny
4. Požadavek na hodnotu vlhkosti půdy

Ukázka požadavku na osvětlení a zapsání hodnoty do databáze:

```

ser.write('1') #Zapsání hodnoty 1 na port sériové linky
while(ser.inWaiting()==0): #Dotaz zda jsou v bufferu již data
    time.sleep(0.1) #Pokud ne, čeká 0.1 sec
licht = ser.readline() #Čtení řetězce ze sériové linky
db.add_lux(licht) #Zapsání hodnoty do databáze

```

Pro ukládání dat do databáze je vytvořen program db.py, který ukládá hodnoty ze senzorů do MySQL databáze.

6.5.2 Čtení a odesílání hodnot z Arduina do RPi

V Arduinu je nahrán program Arduino_Garden, jehož hlavní náplní je výčet dat ze senzorů, jejich odesílání po sériové lince do RPi a dle hodnot vyčtených spouštět čerpadlo nebo osvětlení.

Hlavní částí program jsou funkce `void setup()` a `void loop()`. Funkce `void setup()` je přípravná a provádí se jen jednou na začátku programu. Funkce `void setup()` je volána až po deklaraci všech proměnných na začátku programu. Tato funkce se používá k nastavení pinů Arduina na vstup nebo výstup, nastavení parametrů sériové komunikace, nastavení komunikace s čidly. Po funkci `void setup()` následuje funkce `void loop()`. Tělo této funkce obsahuje programový kód.

6.5.2.1 Hlavní funkce void loop()

```
void loop() {  
    lightLux = light();  
    temperatur = temp();  
    distance = tank();  
    humidityl = humidity();  
    controllight();  
    if (Serial.available()) {  
        uart();  
    }  
}
```

Tato funkce je prováděna v nekonečné smyčce, volají se v ní neustále funkce, které přepisují v proměnných neustále hodnoty, tak aby byly uloženy co nejaktuálnější hodnoty a ty mohli být po dotazu RPi odeslány. Dále je zde podmínka, v které je ověřováno, zde se nenacházejí v bufferu sériové linky data. Pokud se nacházejí, je vykonána funkce `uart()`.

int lichtLux()

```
int light() {  
    uint16_t lux = lightMeter.readLightLevel();  
    return ((int)lux);  
}
```

Tato funkce čte data z čidla BH1750. Pro čtení z tohoto čidla byla použita knihovna BH1750.h. Která pomocí funkce `lightMeter.readLightLevel()` vrací hodnotu

osvětlení v binární hodnotě. Pro lepší použití je hodnota přetypována na integer. Dále je použita knihovna wire.h pro komunikaci po sběrnici I2C.

float temp()

```
float temp() {  
    sensors.requestTemperatures();  
    return (float)(sensors.getTempCByIndex(0));  
}
```

Funkce temp() získává data ze senzoru Dallas DS18B20. Komunikace se senzorem je realizována pomocí sběrnice OneWire. Jsou použity knihovny OneWire.h a dále pak knihovna pro čtení hodnot z čidla DallasTemperature.h. Příkaz sensors.requestTemperatures() žádá senzor o zaslání teploty a příkaz sensors.getTempCByIndex(0) odečítá hodnotu teploty. Tímto příkazem lze vyčítat i v případě, že je na sběrnici připojeno více zařízení a vyčítá senzor uvedený v závorce. Protože je v této práci použito jen jedno čidlo pro měření teploty, je hodnota 0.

int tank()

```
int tank() {  
    digitalWrite(SR04TrigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(SR04TrigPin, LOW);  
  
    long pingTime = pulseIn(SR04EchoPin, HIGH);  
    int distance_cm = pingTime * 0.017315;  
  
    return (tankHeight - distance_cm);  
}
```

Tato funkce vrací hodnotu výšky vodní hladiny v nádrži. Funkce nejprve aktivuje a poté deaktivuje TrigPin. Hlavní funkcí je tu funkce pulseIn(), která vrací hodnotu času za kterou se vrátí vyslaný signal. Poté je tato hodnota vypočítána na vzdálenost a to tak, že rychlost zvuku je při zanedbání okolních vlivů (teplota, vlhkost, atd.) $346,3 \frac{m}{s} = 0,0346,3 \text{ cm}/\mu s$, a protože ultrazvuk urazí dráhu dvakrát, je třeba ještě tuto hodnotu vydělit dvěma, což je 0,017315. Tím byla získána hodnota vzdálenosti hladiny od snímače. Pro získání přesné výšky hladiny je nutné znát výšku nádrže. V tomto případě 15 cm. Rozdílem výšky nádrže a vzdálenosti čidla od hladiny je získána přesná výška hladiny.

int humidity()

```
int humidity() {  
    int sensorValue = analogRead(A0);  
    return (sensorValue / 10.23);  
}
```

```
}
```

Funkce `humidity()` vrací hodnotu vlhkosti půdy v procentech. Senzor je připojen na analogový vstup A0 a dle vlhkosti je přiváděno na vstup napětí v rozmezí 0 - 5 V, které je pomocí A/D převodníku převedeno v Arduinu na číselnou hodnotu 0 – 1023. Získané hodnoty lze rozdělit přibližně takto:

- 0 – 400 suchá půda
- 400 – 700 vlhká půda
- 700 – 1024 mokrá půda

Hodnota je ještě vydělena číslem 10,23, aby byla získána hodnota v procentech.

void uart()

```
void uart() {  
    operation = Serial.read(); //čtení posílaného znaku z bufferu sériové  
linky  
    switch (operation) { //podmínka pro switch  
        case '1': //podmínka pro znak 1  
            Serial.println((int)lightLux); //zapsání hodnoty lightLux na sériový  
port  
            Serial.flush (); //vyprázdnění sériového portu  
            break;  
        case '2':  
            Serial.println(temperatur);  
            Serial.flush ();  
            break;  
        case '3':  
            Serial.println(distance);  
            Serial.flush ();  
            break;  
        case '4':  
            Serial.println(humidity1);  
            Serial.flush ();  
            break;  
        case '5':  
            Serial.flush ();  
            break;  
        case '6': //podmínka pro zapnutí osvětlení  
            LEDlight(1);  
            break;  
        case '7': //podmínka pro vypnutí osvětlení  
            LEDlight(0);  
            break;  
        case '8':  
            spray(water)//podmínka pro spuštění čerpadla na dobu uvedenou v závorce  
            break;  
    }  
}
```

Tato funkce je volána v případě, že se v bufferu sériového portu objeví data. Je přečten znak pomocí `Serial.read()` a dále je dle hodnoty vykonána příslušná operace.

6.5.3 Ukládání do databáze db.py

Tento skript slouží k ukládání hodnot přijatých od Arduina do MySQL databáze. Základním prvkem je knihovna sqlalchemy, která se stará o ovládání databáze. Spojení s databází je prováděno příkazem:

```
engine = create_engine('mysql://root:garden@localhost/garden')
```

Pro ukládání dat do database slouží funkce add_temp(value), add_hum(value), add_lux(value) a add_wlevel(value). Definice funkce add_temp(value):

```
def add_temp(value):  
    res = temp.insert().values(value=value, time=datetime.datetime.now())  
    return engine.execute(res).inserted_primary_key[0]
```

6.6 Aktivace osvětlení a čerpadla

Osvětlení

Osvětlení je automaticky spouštěno pomocí cronu každý den na 12 hodin, v případě, že je intenzita osvětlení vyšší než 90 Lux, je toto vyhodnoceno programem v Aruinu a osvětlení vypnuto, v případě že klesne pod 70 Lux je opět zapnuto. K tomuto slouží funkce void controllight().

```
void controllight(){  
    if(lightLux < 70 && dayNight == true && lightOn == false){  
        LEDlight(1);  
    }  
    if(lightLux > 90 && dayNight == true && lightOn == true){  
        LEDlight(0);  
    }  
}
```

Hodnotu okolního osvětlení nebude rušit nainstalované osvětlení, protože se jedná o UV LED, které vyzařují UV A záření s vlnovou délkou 405 nm, které má svítivost v řádech LUX.

Lze jej taktéž zapínat nebo vypínat pomocí tlačítka ve webovém rozhraní. Spouštění osvětlení je pomocí php skriptu, který volá program on.py a po zapnutí ukládá do textového souboru 1, aby bylo patrné, zda je osvětlení sepnuté, či vypnuté.

```
<?php  
exec('python /home/pi/on.py');  
$soubor = fopen("soubor.txt", "r");  
$text = fwrite(0);  
fclose($soubor);  
header('Location: index.php');  
exit;  
?>
```

Program `on.py` zasílá informaci na sériový port o tom, že má Arduino sepnout osvětlení, jehož spínání je připojeno k pinu 8. Stejně tak fungují script `off.php` a program `off.py`, jen s rozdílem, že osvětlení vypínají.

Spouštění čerpadla

Pomocí `cronu` a programu `spray.py` je každou hodinu zasílán do Arduina požadavek, zda je dostatečně vlhká půda. Byla určena hranice 40, což je hranice mezi suchou a vlhkou půdou. V případě, že je půda suchá, je spuštěno čerpadlo na donou dobu (v tomto případě 2 sec). V programu Arduino obstarává tuto kontrolu funkce `void spray(int sec)`.

```
void spray(int sec) {  
  if(humidity<40){  
    digitalWrite(PumpPin, LOW);  
    delay(sec);  
    digitalWrite(PumpPin, HIGH);  
  }  
}
```

Čerpadlo je možné spouštět i manuálně z webového rozhraní, avšak stále je hlídáno, jestli není půda vlhká, v případě že ano, zalití se neprovede.

6.7 Cron

Cron je softwarový spouštěč příkazů, resp. procesů. V operačním systému se tedy jedná o plánovače úloh.

Zapisování hodnot z grafů do databáze

Tento cron je používán pro automatické spouštění odečítání dat z Arduina a ukládání do MySQL databáze. Příkaz automatického spouštění je:

```
*/15 * * * * sudo python /home/pi/uart.py
```

Data jsou odečítána každých 15 minut. Cron je možné taktéž spravovat pomocí programu `webmin`, jež je popsán v kapitole 6.2.4.

Kontrola vlhkosti půdy

Každou hodinu, probíhá kontrola vlhkosti půdy a v případě, že je půda suchá je spuštěno čerpadlo a půda zalita. Příkaz pro tuto událost má tento tvar:

```
10 * * * * sudo python /home/pi/spray.py
```

Zapnutí / Vypnutí osvětlení

Každý den v 8:00 je zapnuto osvětlení a v 20:00 je osvětlení vypnuto, v této době je vypínáno dle hodnoty okolního osvětlení. Příkazy pro tyto události mají tvar:

```
* 8 * * * sudo php /var/www/html/on.php
* 20 * * * sudo php /var/www/html/off.php
```

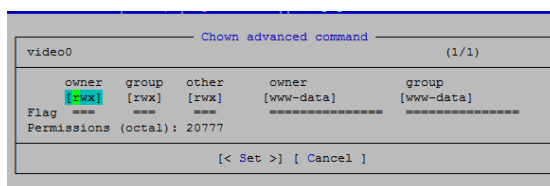
6.8 Program pro obsluhu kamery

Webcamera slouží k pořizování fotek na základě požadavku skrz webové rozhraní. Webcamera vyfotí snímek, který je následně zobrazen. K pořizování fotografie slouží program photo.py. Hlavní součástí tohoto programu je knihovna pygame.

```
import pygame, sys
from pygame.locals import *
import pygame.camera
pygame.init()
pygame.camera.init()
cam = pygame.camera.Camera("/dev/video0", (352, 288))
cam.start()
image= cam.get_image()
pygame.image.save(image, 'img.jpg')
cam.stop()
```

#inicializace kamery
#načtení kamery
#spuštění kamery
#vytvoření fotografie
#uložení do img.jpg
#vypnutí kamery

Aby bylo možné ukládat fotografii do souboru ve webovém prostoru webového serveru Apache, je třeba změnit práva souboru /dev/video0 na owner: **www-data** a group: **www-data**.



Obrázek 23: Práva video0

6.9 Webové rozhraní

Pro zobrazování dat z automatické zahrádky a jednoduché manuální ovládání bylo zvoleno webové rozhraní, které je vytvořeno v jazyce php. Design webové stránky byl realizován s důrazem na jednoduchost s použitím standartních funkcí a nástrojů. Celý web je rozdělen do 2 sekcí: Home a Historie

Jednotlivé PHP soubory a skripty lze nahrávat pomocí Putty, konkrétně příkazu sudo nano „Název souboru“. Tento způsob je však velmi zdlouhavý a pro zobrazení zdrojových kódů velmi nepřehledný, z tohoto důvodu je mnohem pohodlnější nahrávat soubory například pomocí programu FileZilla, což je program zajišťující serverové služby pro SFTP (SSH file transfere protocol) přenos. Jedná se o velice populární OpenSource program vytvořený pro platformu Windows.

6.9.1 Stránka Home

Tato stránka slouží k náhledu aktuálních hodnot ze senzorů, dále pak k jednoduchému manuálnímu ovládání osvětlení a čerpadla, také je možnost aktuálního náhledu pomocí webkamery. Na stránce je stále zobrazován pomocí jednoduchého příkazu v JavaScriptu čas a to v levém sidebaru. Na konci této stránky jsou také základní informace o tomto projektu. Celý zdrojový kód index.php je uveden v příloze na cd.

Ukazatel aktuálních hodnot

Na stránce Home se nachází v první části ukazatele hodnot ze senzoru, které byly do databáze uloženy jako poslední. Hodnoty jsou zobrazovány pomocí skriptu Visualization: Gauge, který je zdarma dostupný na <https://developers.google.com/>.

```
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
    google.charts.load('current', {'packages':['gauge']});
    google.charts.setOnLoadCallback(drawChart);
    function drawChart() {
```

Data jsou získávána pomocí php skriptu.

```
var data2 = google.visualization.arrayToDataTable([
    ['Teplota', 'Value'],
    ['Osvětlení',
<?php
$sql = "SELECT value FROM lux ORDER BY `lux`.`id` DESC LIMIT 1";
$result = mysqli_query($spojeni, $sql);
if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo $row["value"];
    }
} else {
    echo "0";
}
?>
],1]);
```

Dále jsou nadefinovány v JavaScriptu rozměry a vlastnosti zobrazovaného ukazatele.

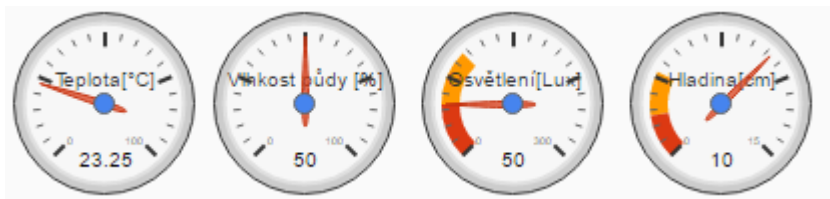
```
var options2 = {
width: 100, height: 120,
redFrom: 0, redTo: 50,
yellowFrom: 50, yellowTo: 100, max: 300,
minorTicks: 5
};
```

Poté je uložen celý ukazatel do proměnné a následně zobrazen pomocí html příkazu div.

```
var chart2 = new
google.visualization.Gauge(document.getElementById('chart2_div'));
```

```
<div id="chart2_div" class="grafy"></div>
```

Výsledkem je grafické zobrazení hodnot z čidel.



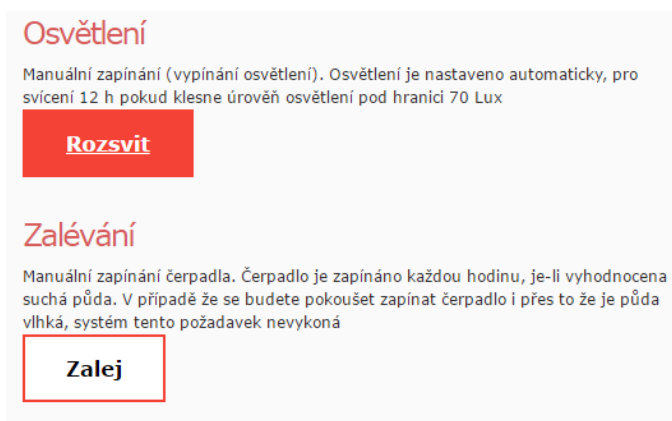
Obrázek 24: Grafické zobrazení aktuálních hodnot ze senzorů

Přístup k webovému rozhraní je v lokální síti na adrese <http://192.168.0.101> nebo ve veřejné síti na adrese <http://84.21.104.214:8081>.

Manuální ovládání osvětlení a čerpadla

Pomocí dvou tlačítek, lze jednoduše ovládat LED osvětlení a čerpadlo. V případě, že je LED osvětlení vypnuto, je aktivováno tlačítko „Rozsvit“ v opačném případě je aktivní tlačítko „Zhasni“. Tlačítko „Rozsvit“ aktivuje program on.py který je popsán v kapitole 6.6. v opačném případě aktivuje program off.py.

V případě aktivace tlačítka „Zalej“ je aktivováno čerpadlo, ale pouze v případě, že je půda suchá.



Obrázek 25: Jednoduché ovládání osvětlení a čerpadla

Webkamera

Aktuální snímek z webkamery lze pořídit kliknutím na fotografii. Fotografie je vytvořena pomocí programu, který je popsán v kapitole 6.7.

6.9.2 Stránka Historie

Na této stránce jsou zobrazována kompletní data uložená v databázi garden do grafů. Data jsou zobrazována pomocí skriptu Line Chart, který je zdarma dostupný na <https://developers.google.com/>.

Data jsou získávána obdobným způsobem jako v případě čtení pro grafické zobrazování na stránce Home, s rozdílem, že jsou čteny všechny řádky ze sloupců, kde jsou uloženy hodnoty ze snímačů a čas kdy byla hodnota změřena. Celý zdrojový kód history.php je vložen v příloze na cd.



Obrázek 26: Část webové stránky Historie

Na konci stránky je tlačítko „Smazat data“ které smaže všechny data v databázi. K mazání dat slouží skript delete.php.

```
<?php
$spojeni=mysqli_connect("localhost", "root", "garden", "garden"); //
připojení k databázi
if (mysqli_connect_errno())
{
    echo "Nelze se připojit k MySQL: " . mysqli_connect_error(); // výpis v
případě chyby
}
$sql = "TRUNCATE water_level"; // vložení
příkazu do proměné
$result = mysqli_query($spojeni, $sql); //vykonání
příkazu

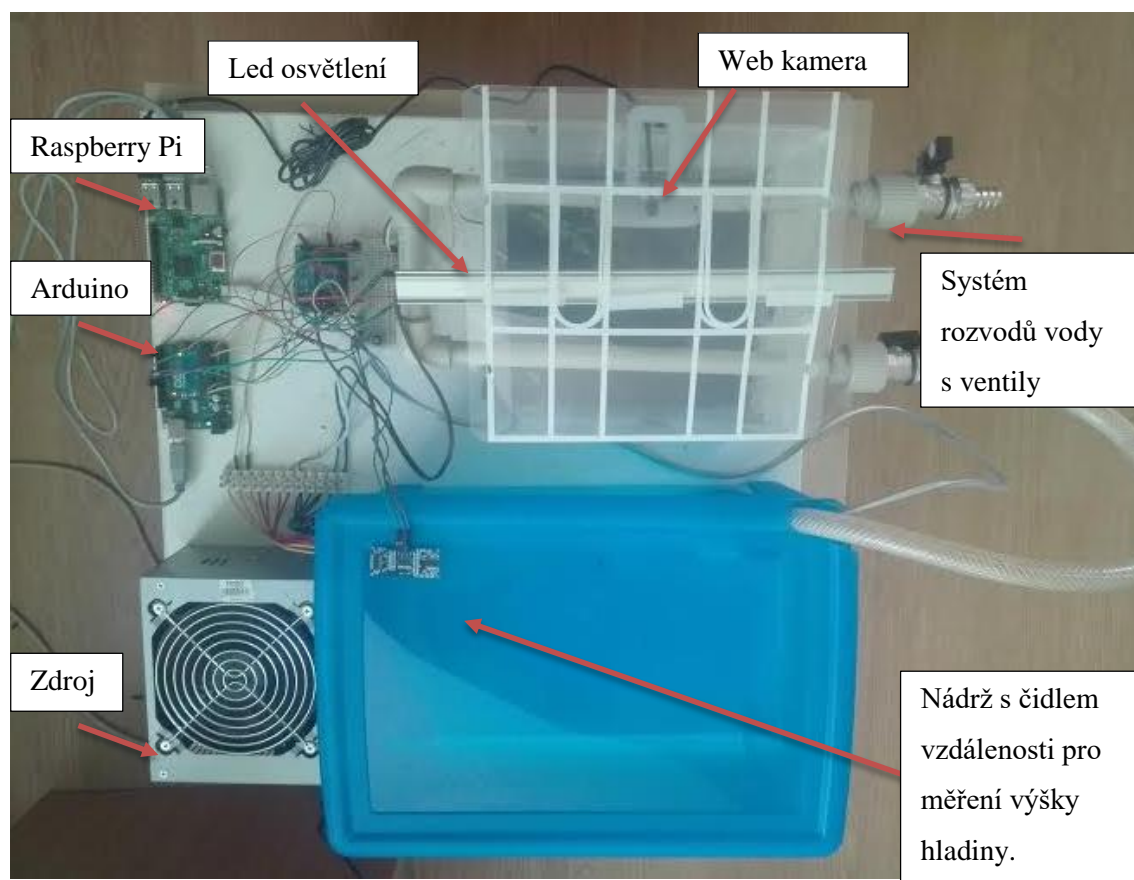
$sql = "TRUNCATE temperature";
$result = mysqli_query($spojeni, $sql);

$sql = "TRUNCATE humidity";
$result = mysqli_query($spojeni, $sql);
```

```
$sql = "TRUNCATE lux";  
$result = mysqli_query($spojeni, $sql);  
  
if ($result){  
header('Location: history.php');  
}else{  
echo "Nastala chyba";  
}  
?>
```

7 REALIZACE MODELU

Realizace celého projektu byla vytvořena jako model, který je umístěn na lamino desku. Vše je propojeno pomocí drátků a kontaktního pole. Součástí modelu je nádrž, jež je použita z bedny na potraviny, dále pak systém rozvodů, který byl vytvořen z plastových vodovodních rozvodů, a jsou na systému umístěny dva ventily. První z nich uzavírá vodu do zalévacího potrubí a druhý slouží k odvzdušnění systému, v případě že je systém zavzdušněn a nelze zalívat. Jako skleník byl použit skleníček z kampaně obchodního řetězce Albert Albertova zahrádka.



Obrázek 27: Model inteligentní zahrady

8 ZÁVĚR

Hlavním cílem této práce bylo vytvoření inteligentní zahrádky za pomoci běžně dostupných a cenově dostupných prvků.

V prvním kroku bylo seznámeno co je to pojem inteligentní rodinný dům, jaké výhody to lidem přináší.

Další částí práce bylo navrženo, jaký hardware bude použit pro vypracování inteligentní zahrádky. Jako hlavní prvek byl vybrán výpočetní modul Raspberry Pi 2 model B. Veškeré jeho vlastnosti jsou popsány v kapitole číslo 4, která je kompletně věnována celé rodině počítačů Raspberry.

V kapitole číslo pět bylo seznámeno se všemi ostatními hardware komponenty, které tvoří celý systém inteligentní zahrádky.

Kapitola číslo šest je kompletně věnována softwarovému vybavení. Jsou zde popsány veškeré použité programy a to komunikace mezi Raspberry Pi a Arduinem, dále pak způsob výčtu a ukládání dat ze senzorů do databáze. Je zde také popsán program pro jednoduché automatické a manuální řízení osvětlení a čerpadla sloužícího pro zalévání. Na konci této kapitoly je popsáno webové rozhraní, které slouží k prezentaci dat a jednoduchému ovládání osvětlení a čerpadla.

Poslední kapitola je věnována fyzickému modelu inteligentní zahrádky včetně popisu.

Zadání bylo splněno, avšak nebylo otestováno na reálné zahradě, pouze na modelu. Návrhy na zlepšení by mohly být tyto: propojení Raspberry Pi a Arduina pomocí wifi modulu, tak aby mohla být umístěna čidla do větší vzdálenosti od serveru a tedy ven do reálné zahrady. Dále by bylo vhodné doplnit systém o možnost nastavování parametrů zalívání a to z důvodu možnosti nastavování jiné délky osvětlení, nebo udržování hodnoty půdy dle nastavených hodnot. Jako další prvek by mohlo být přidáno automatické sbírání snímku a z nich dále pak vytvořené časosběrné video prezentované ve webovém rozhraní. Vylepšení je určitě spousta. Avšak základ je nyní dán a stačí pouze doplňovat funkce.

Literatura

- [1] BC. GROLIG, Lukáš. Implementace vlastního komplexního řídicího systému pro řízení rodinného domu. Zlín, 2015. UTB. Vedoucí práce Ing. Tomáš Sysala, Ph.D.
- [2] BC. HÜBNER, Pavel. Systém sběru dat s Raspberry Pi pro domovní automatizaci. Praha, 2015. České vysoké učení technické v Praze. Vedoucí práce Doc. Ing. Jan Fischer, CSc.
- [3] Přispěvatelé Wikipedie, Raspberry Pi [online], Wikipedie: Otevřená encyklopedie, c2016, Datum poslední revize 25. 04. 2016, 09:52 UTC, [citováno 15. 05. 2016] <https://cs.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=13590115>
- [4] NEZNÁMÝ. www.youtube.com [online]. [cit. 15.5.2016]. Dostupný na WWW: <https://i.ytimg.com/vi/s4bWA4g0qV0/maxresdefault.jpg>
- [5] NEZNÁMÝ. www.raspberrypi-spy.co.uk [online]. [cit. 15.5.2016]. Dostupný na WWW: <http://www.raspberrypi-spy.co.uk/wp-content/uploads/2012/06/Raspberry-Pi-GPIO-Layout-Model-B-Plus-rotated-2700x900-1024x341.png>
- [6] SOBOTKOVÁ, Š. Detektor dětského pláče s digitálním signálovým procesorem. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav biomedicínského inženýrství, 2013. 78s., 14 s. příloh. Diplomová práce. Vedoucí práce: Ing Jiří Sekora
- [7] NEZNÁMÝ. [wikimedia](http://wikimedia.org) [online]. [cit. 15.5.2016]. Dostupný na WWW: https://upload.wikimedia.org/wikipedia/commons/3/38/Arduino_Uno_-_R3.jpg
- [8] JANKŮ, Petr. Telemetrie a zaměřování polohy modelu pomocí GPS. Praha, 2008. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE. Vedoucí práce Ing. Ondřej Špinka.
- [9] NEZNÁMÝ. [Instructables](http://instructables.com) [online]. [cit. 15.5.2016]. Dostupný na WWW: <http://cdn.instructables.com/F3S/BPMK/IB228TLV/F3SBPMKIB228TLV.MEDIUM.jpg>
- [10] Lukáš Kraicinger: Automatická meteorologická stanice, bakalářská práce, Brno, FIT VUT v Brně, 2015
- [11] NEZNÁMÝ. [Santy](http://www.santy.cz) [online]. [cit. 15.5.2016]. Dostupný na WWW: http://www.santy.cz/data/product/181_817.jpg
- [12] KRAUS, V. Dvounohý kráčivý robot. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 67 s. Vedoucí bakalářské práce doc. Ing. Pavel Šteffan, Ph.D..

- [13] NEZNÁMÝ. Instrutables [online]. [cit. 15.5.2016]. Dostupný na WWW: <http://cdn.instructables.com/F90/JYNW/H7UR7RCS/F90JYNWH7UR7RCS.MEDIUM.jpg>
- [14] NEZNÁMÝ. Russia-nato [online]. [cit. 15.5.2016]. Dostupný na WWW: http://img-fotki.yandex.ru/get/64624/28883869.210d5/0_75770a_37eb9cb6_orig
- [15] NEZNÁMÝ. DXCDN [online]. [cit. 15.5.2016]. Dostupný na WWW: http://img.dxcn.com/productimages/sku_152072_1.jpg
- [16] EAGLE. Svět hardware [online]. [cit. 15.5.2016]. Dostupný na WWW: <http://www.svethardware.cz/specifikace-zdroju-atx12v/9807-2>
- [17] NEZNÁMÝ. Genius net [online]. [cit. 15.5.2016]. Dostupný na WWW: <http://www.geniusnet.com/Genius/wSite/public/Data/f1331701488968.jpg>
- [18] buben. Raspi. Raspi.cz. [online]. [cit. 2016-05-15]. Dostupné z: <http://www.raspi.cz/2013/02/propojujeme-raspberry-pi-arduino/>

Zkratky:

SDHC - Formát paměti jako SD, ale pro kapacity nad 2 GB. (Secure digital High capacity)

HW - Technické vybavení počítače. (HardWare)

UART - Univerzální asynchronní přijímač a vysílač (Universal Asynchronous Receiver Transmitter)

RPi – Raspberry Pi

GPIO - Všeobecný vstup/výstup (General purpose Input/Output)

PWM - Pulsně šířková modulace. Řízení výkonu poměrem dob připojení/odpojení zátěže ke zdroji. (Pulse Width Modulation)

GND – uzemnění (GrouND)

SPI - Sériové periferní rozhraní. Typ sběrnice (a protokolu) pro komunikaci s čipovými kartami.(Serial Peripheral Interface)

I2C - multi-masterová počítačová sériová sběrnice (Inter-Integrated Circuit)

OS - Operační systém (operační systém)

USB - Univerzální sériová sběrnice (Universal Serial Bus)

ARM - Zdokonalený počítač typu RISC (Advanced RISC Machine)

AC - Střídavý proud (Alternating Current)

DC - Stejnoseměrný proud (Direct Current)

RX – příjem, opak Tx (Receive)

TX – vysílání, opak Rx (

TTL - Tranzistorově tranzistorová logika. Technologie integrovaných obvodů. (Transistor Transistor logic)

SS – Vybrán slave (Slave select)

MOSI - Master odesílá, slave přijímá (Master Out, Slave In)

MISO – Master přijímá, slave odesílá (Master In, Slave Out)

HTML - Značkový jazyk pro hypertext. (HyperText Markup Language)

PHP - Bohatý skriptovací jazyk na straně serveru zvláště určený pro vývoj webových aplikací. (Hypertext Preprocessor)

IP - Protokol Internetu (Internet Protocol)

Seznam příloh

Příloha A. CD

A. Obsah CD

Zdrojové kódy php

- index.php
- history.php
- delete.php
- off.php
- on.php
- spray.php

Zdrojové kódy python:

- db.py
- off.py
- on.py
- photo.py
- spray.py
- uart.py

Zdrojový kód databáze MySQL

Zdrojový kód pro Arduino Arduino_garden.ino