

Česká zemědělská univerzita v Praze

Technická fakulta

Katedra elektrotechniky a automatizace



Bakalářská práce

Sledování a regulace fyzikálních veličin v izolovaném boxu

Vedoucí práce: prof. Ing. Jaromír Volf, DrSc.

Autor práce: Jan Stoklasa

© 2017 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Technická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jan Stoklasa

Informační a řídicí technika v agropotravinářském komplexu

Název práce

Sledování a regulace fyzikálních veličin v izolovaném boxu

Název anglicky

Monitoring and Control of Physical Values in Isolated Box

Cíle práce

Cílem práce je měření a regulace teploty a vlhkosti v izolovaném boxu pomocí jednodeskového počítače Raspberry Pi a předávání naměřených hodnot a stavů regulace pro další zpracování a prezentaci.

Metodika

Seznamte se s problematikou měření teploty a vlhkosti. Dále se seznamte s možnostmi regulace těchto veličin. V bakalářské práci se zaměřte na měření a regulace teploty a vlhkosti v izolovaném boxu pomocí jednodeskového počítače Raspberry Pi a předávání naměřených hodnot a stavů regulace pro další zpracování a prezentaci.

Doporučený rozsah práce
do 40 stran

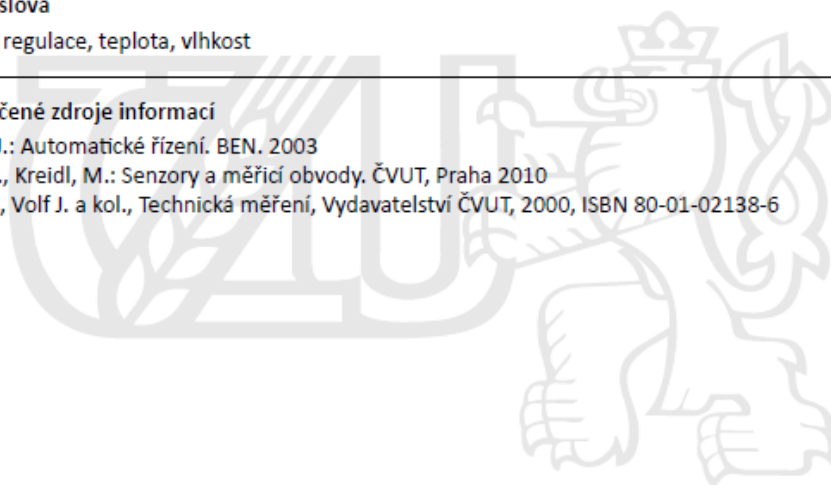
Klíčová slova
měření, regulace, teplota, vlhkost

Doporučené zdroje informací

Balátě, J.: Automatické řízení. BEN. 2003

Ďaďo, S., Kreidl, M.: Senzory a měřicí obvody. ČVUT, Praha 2010

Jencík J., Volf J. a kol., Technická měření, Vydavatelství ČVUT, 2000, ISBN 80-01-02138-6



Předběžný termín obhajoby
2016/17 LS – TF

Vedoucí práce
prof. Ing. Jaromír Volf, DrSc.

Garantující pracoviště
Katedra elektrotechniky a automatizace

Elektronicky schváleno dne 12. 1. 2016
prof. Ing. Jaromír Volf, DrSc.
Vedoucí katedry

Elektronicky schváleno dne 2. 3. 2016
prof. Ing. Vladimír Jurča, CSc.
Děkan

V Praze dne 26. 03. 2018

Čestné prohlášení

„Prohlašuji, že jsem diplomovou/bakalářskou práci na téma: Sledování a regulace fyzikálních veličin v izolovaném boxu vypracoval/a samostatně a použil/a jen pramenů, které cituji a uvádím v seznamu použitých zdrojů.

Jsem si vědom/a, že odevzdáním bakalářské práce souhlasím s jejím zveřejněním dle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů, ve znění pozdějších předpisů, a to i bez ohledu na výsledek její obhajoby.

Jsem si vědom/a, že moje bakalářská práce bude uložena v elektronické podobě v univerzitní databázi a bude veřejně přístupná k nahlédnutí.

Jsem si vědom/a že, na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, ve znění pozdějších předpisů, především ustanovení § 35 odst. 3 tohoto zákona, tj. o užití tohoto díla.“

V Praze dne:.....

.....

Poděkování

Tímto chci poděkovat panu prof. Ing. Jaromíru Volfovi, DrSc. za odborné vedení bakalářské práce, poskytování rad a věnovaný čas.

Sledování a regulace fyzikálních veličin v izolovaném boxu

ANOTACE:

Tato práce popisuje praktický příklad řešení měření a regulace teploty pomocí mikropočítače Raspberry Pi. Naměřené hodnoty jsou zaznamenávány do databáze a prezentovány pomocí grafů na webovém serveru. Regulace teploty je prováděna spínáním napájení přímotopu nebo ventilátoru.

V úvodní části práce jsou popisovány obecné principy. V druhé části je představeno použité programové a technické vybavení. V další části je popsána praktická implementace a údržba systému.

Práce si klade za cíl představit jednoduché a levné řešení měření a regulace teploty.

Klíčová slova:

Měření a regulace teploty a vlhkosti, Raspberry Pi, Dallas DS18B20, Sainsmart Relay Module

ABSTRACT:

This paper describes a practical example solutions for measurement and control of temperature using Raspberry Pi microcomputer. The measured values are recorded in a database and presented in graphs on a web server. Temperature control is performed by switching the heater power supply or fan.

In the introductory part of the paper describes the general principles. The second part presents the used software and hardware. The next section describes the practical implementation and system maintenance.

The work aims to present a simple and inexpensive solution to the measurement and control of temperature.

Keywords:

Measurement and control of temperature, Raspberry Pi, Dallas DS18B20, Sainsmart Relay Module

Obsah

1 Úvod	1
2 Cíle práce	2
3 Metodika	2
4 Teplota.....	3
4.1 Rozdělení dotykových teploměrů	4
4.2 Měření teploty.....	4
4.3 Regulace teploty.....	4
4.4 Způsoby regulace	5
4.4.1 Dvoustupňová regulace.....	5
4.4.2 PWM Regulace	6
4.5 Dallas DS18B20.....	7
5 Vlhkost	9
5.1 Měření vlhkosti.....	9
5.1.1 Psychrometr	9
5.1.2. Vlhkoměr s pevným elektrolytem	10
5.2 Čidlo DHT11.....	10
6 Technické a programové vybavení – Hardware a Software	11
6.1 Mikropočítač Raspberry Pi	11
6.1.1 Hardware.....	11
6.1.2. OS Raspbian.....	12
6.1.3 WiringPi	12
6.1.4 Apache.....	13
6.1.5 Dvoustavová regulace – software	13
6.1.6. RRDTool	15
6.2 Příkazy čidlu.....	17

6.3 Spínací prvky.....	18
6.3.1 Plošný spoj s optotriaky (vlastní výroba).....	18
6.3.2 Sainsmart Relay Module	19
7 Praktická část implementace	21
7.1 Verze 1.0.....	21
7.2 Verze 1.1.....	22
7.3 Verze 2.0.....	22
4.4 Verze 2.1.....	24
4.5 Verze 3.0.....	24
8 Údržba a provoz systému jako celku	25
8.1 Údržba paměti (SD karta)	25
8.2 Kontrola funkce	25
9 Závěr	26

Seznam tabulek

Tab. 1 - Převodní vztahy mezi teplotními stupnicemi.....	3
--	---

Seznam obrázků

Obr. 1 - Teplotní limity	5
Obr. 2 - PWM regulace	6
Obr. 3 - graf regulace PWM.....	7
Obr. 4 - Pouzdro DS18B20	8
Obr. 5 - Princip psychrometru	9
Obr. 6 - Připojení čidla.....	10
Obr. 7 - Blokové schéma Raspberry Pi	11
Obr. 8 - Vývojový diagram	14
Obr. 9 - Hodinová teplota.....	16
Obr. 10 - Denní teplota.....	16

Obr. 11 - Měsíční teplota.....	16
Obr. 12 - terminál putty, volání 18B20.....	18
Obr. 13 - schéma plošného spoje.....	18
Obr. 14 - SaintSmart relay module schéma	19
Obr. 15 - SaintSmart relay	19
Obr. 16 - Schéma připojení čidla DS18B20.....	21
Obr. 17 - Schéma zapojení párů	22
Obr. 18 - Provozní zapojení	23
Obr. 19 - zadávání teplot.....	24

1 Úvod

Úkolem práce bylo sestavit zařízení, které bude měřit a zaznamenávat teplotu v několika místech uzavřeného prostoru a vhodným způsobem teplotu regulovat. Zaznamenané teploty musí být uživatelsky přívětivým způsobem prezentovány. Toto zařízení mělo být založeno na mikropočítači Raspberry Pi. Měřicí a spínací prvky jsou voleny s ohledem na kompatibilitu s tímto mikropočítačem.

Problematika měření a regulace teploty by se na první pohled mohla zdát jednoduchá, ale skrývá v sobě mnohá zákoutí a úskalí, která na první pohled nemusí být patrná, ale mohou zásadním způsobem ovlivnit průběh a výsledek celé práce.

Vlastní zařízení, použité měřicí a ovládací periferie jsou použity v maximální možné míře sériově vyráběné. Pro programování jsou využity standardní nástroje a nadstavbové aplikace operačního systému a programovací jazyk vyšší generace.

Centrálním prvkem zařízení je jednodeskový mikropočítač Raspberry Pi. Pro měření teploty jsou použity číslicové teploměry Dallas DS18B20 a vlhkosti DHT11. Pro spínání topení, resp. chlazení je použit Sainsmart 4-Channel Relay Module. Topení je realizováno přímotopem a pro chlazení je použit ventilátor. Měření teploty, vyhodnocování a spínání je napsáno v jazyce Python, data jsou zaznamenávána do databáze RRD. Prezentace a ovládání je pomocí webového serveru Apache.

Popisované zařízení na měření a regulaci teploty bylo sestaveno a prakticky odzkoušeno v boxu pro pěstování hlívy ústřičné. Zařízení sloužilo ke sledování teploty, vlhkosti a udržování teploty v určitém rozmezí. V první fázi technologického procesu bylo nutné požadované teploty docílit topením, následně po nastartování procesu v substrátu bylo nutné box chladit. Proces regulace teploty musel zvládnout také řešení nesystematických jevů v podobě nárazového chlazení otevíráním dveří při vstupu do boxu a manipulaci se substrátem.

2 Cíle práce

Cílem práce je měření a regulace teploty a vlhkosti v izolovaném boxu pomocí jednodeskového počítače Raspberry Pi a předávání naměřených hodnot a stavů regulace pro další zpracování a prezentaci.

3 Metodika

Seznamte se s problematikou měření teploty a vlhkosti. Dále se seznamte s možnostmi regulace těchto veličin. V bakalářské práci se zaměřte na měření a regulace teploty a vlhkosti v izolovaném boxu pomocí jednodeskového počítače Raspberry Pi a předávání naměřených hodnot a stavů regulace pro další zpracování a prezentaci

4 Teplota

Teplota je jednou ze základních termodynamických vlastností a je součástí soustavy prvků SI. Je zároveň jednou z nejdůležitějších stavových veličin, které ovlivňují stavy a procesy v přírodě. Pro správné měření teploty není zapotřebí pouze teploty správně číst a mít dostatečně přesný teploměr. Je zároveň nutné měřicí zařízení umístit a zabudovat tak, aby teplotu měřilo tak jak je požadováno.[]

Při měření teplot lze využívat různých fyzikálních principů, jako například teplotní roztažnost látek, změny elektrického odporu ve vodičů a polovodičů, vznik elektrického napětí a podobně. Pro dosažení stejných výsledků měření, byla stanovena jednotná teplotní stupnice, která nesměla být závislá na typu a principu měření. Taková stupnice je nazývána termodynamická teplotní stupnice, odvozená z účinnosti vratného tepelného stroje (Carnotův ideální stroj). Základní jednotkou termodynamické teploty T této stupnice je kelvin (K)

Definice: Jeden kelvin je 273,16-tý díl termodynamické teploty trojného bodu vody při tlaku 101,325 kPa.

Nejnižší teplota je udána jako absolutní nula, při níž dochází k zastavení veškerého pohybu. K této teplotě se lze pouze přiblížit, nedá se jí dosáhnout. Lze také označit jako nula termodynamické stupnice $T = 0 \text{ K}$, tj. $-273,15 \text{ } ^\circ\text{C}$.

Kromě termodynamické teploty T (K) vyjadřované v kelvinech, používá se ještě Celsiova stupnice vyjadřovaná v Celsiových stupních t ($^\circ\text{C}$), jejíž jednotkou je 1°C a je roven $t = T - 273,15$. Teplotní rozdíl 1 K je roven přesně 1°C . [1]

Tab. 1 - Převodní vztahy mezi teplotními stupnicemi

Teplota:	T (K)	t ($^\circ\text{C}$)	T_1 ($^\circ\text{R}$)	t_1 ($^\circ\text{F}$)
T (K) =	1	$t + 273,15$	$(5/9) \cdot T_1$	$(5/9)(t_1 + 459,67)$
t ($^\circ\text{C}$) =	$T - 273,15$	1	$(5/9)(T_1 - 491,67)$	$(5/9)(t_1 - 32)$
T_1 ($^\circ\text{R}$) =	$(9/5) \cdot T$	$(9/5) \cdot t + 491,67$	1	$t_1 + 459,67$
t_1 ($^\circ\text{F}$) =	$(9/5) \cdot T - 459,67$	$(9/5) \cdot t + 32$	$T_1 - 459,67$	1

Zdroj: [1]

4.1 Rozdělení dotykových teploměrů

Dotykové teploměry jsou takové, které jsou v neustálém kontaktu s měřeným objektem (prostředím). Dotykové teploměry lze rozdělit podle fyzikálního principu funkce na:

- a) dilatační - využívají teplotní roztažnosti tuhých látek, kapalin a plynů,
- b) parní - využívají teplotní závislosti tlaku syté páry teploměrové kapaliny,
- c) odporové - využívají teplotní závislosti elektrického odporu vodičů a polovodičů,
- d) polovodičové - využívají změny vlastností polovodičů s teplotou, velmi často je tato změna převáděna na změnu odporu čidla, proto jsou také často řazeny k odporovým.
- e) termoelektrické - využívají teplotní závislosti termoelektrického napětí termoelektrického článku,
- f) optovláknové - využívají změny vlastností optických vláken s teplotou[1]

4.2 Měření teploty

Měření teploty provádíme různými druhy teploměrů. Nejpoužívanější je kapalinový teploměr, kdy se do uzavřené nádoby umístí kapalina, která vlivem okolních teplotních podmínek mění svůj objem. Hodně užívaný je i digitální teploměr. Ten funguje na principu měření elektrického odporu vodičů a polovodičů, který je závislý právě na teplotě. Též závislosti prahového napětí PN přechodu na teplotě. Měřením teploty se zabývá termometrie.
[1]

4.3 Regulace teploty

Regulace v obecném slova smyslu znamená provádění řídicí či usměrňující činnosti vykonávané za účelem dosažení určitého cíle. Mým cílem je sestavit automatické zařízení, které bude pomocí přímotopu a ventilátoru regulovat teplotu v uzavřeném prostoru.

Zpracování regulace teploty lze pojmout několika způsoby. Je možné nakoupit poměrně drahé termostaty, které ale většinou umí spínat pouze jednu zásuvku (230V). Proto se dá mluvit jen o regulaci omezenou zdola (spínání ohřevu při poklesu teploty na určitou mez) nebo omezenou shora (spínání chlazení při překročení teplotní meze). Tyto termostaty nelze připojit k žádné PC periférii, a proto není možné řídit proces spínání ani zaznamenávat naměřená data a přenášet je do uživatelského prostředí.

Další možností je vyrobit periférii, na kterou se dá napojit libovolný počet čidel a regulačních prvků pomocí sériových portů (například RS232), a tuto periférii připojit k osobnímu počítači (PC). Do osobního počítače se nainstaluje vhodný systém, například Unixový - Fedora, FreeBSD apod, pomocí kterého lze poměrně jednoduchým způsobem tuto periférii ovládat a číst a zaznamenávat z ní data. Zásadním problémem tohoto řešení je velikost osobního počítače, který chceme umístit přímo do provozní místnosti nebo v bezprostřední blízkosti provozní místnosti, ve které měříme a regulujeme teplotu. Pokud by byl řídicí počítač umístěn jinde než v provozní místnosti, bylo by nutné instalovat několikametrové kabely pro spojení s čidly.

Řešení regulace pomocí mikropočítače RaspberryPi umožní číst a zaznamenávat velký počet teplotních čidel, spínat několika periférií a bude zachován požadavek na minimální prostorové nároky. Dalším podstatným rozdílem je cena, která je pro RaspberryPi výrazně nižší. S ohledem na velikost mikroPC ho lze umístit takřka kamkoli. Pokud by bylo nutné, není problém ho dát do izolovaného pouzdra ven. Odpovídá všem požadavkům, které na sledování a regulaci máme. Umí také vytvořit server (Apache), na kterém bude možné prezentovat zaznamenaná data a zadávat minimální a maximální teploty pro regulaci v izolovaném prostoru. Proto bylo asi nejvhodnější použít tento způsob.[3]

4.4 Způsoby regulace

Pro tento projekt připadali v úvahu dva způsoby regulace teploty.

4.4.1 Dvoustupňová regulace

Dvoustupňová regulace (zapnuto/vypnuto) využívá pouze maximálního výkonu topného tělesa (chladicí jednotky). Jinými slovy existují pouze dva stavy, a to provoz, kdy těleso pracuje na 100% a vytápí (chladí) svůj okolní prostor, nebo klidový stav, kdy je těleso



Obr. 1 - Teplotní limity

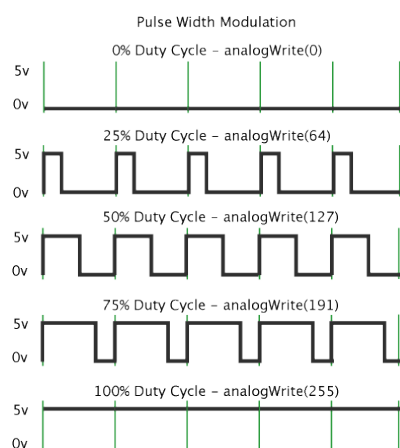
vypnuto (pracuje na 0 %). Jedná se o skokovou regulaci (On/Off).

Výhodou této regulace je poměrně jednoduchý regulační program, který pracuje pouze s dvěma (po úpravě programu třemi) hodnotami teplot – teplota minimální a maximální (ideální).

RaspberryPi je nastaveno tak, aby při vybočení teplot z požadovaného teplotního rozmezí (min, max) dokázalo sepnout potřebné zařízení (přímotop, ventilátor) a nechalo ho spuštěné tak dlouho, než dosáhne ideální hodnoty teploty. Na obrázku č. 1 je vidět teplotní rozsah, který se pohybuje mezi 21°C až 28°C. Při překročení jednoho z limitů bude sepnuto požadované zařízení pro změnu teploty a bude topit až do fáze, kdy se teplota překlopí přes ideální aby nebyla potřeba neustále spínat a vypínat topné těleso, ventilaci nebo klimatizaci, což je relativně energicky náročné.

4.4.2 PWM Regulace

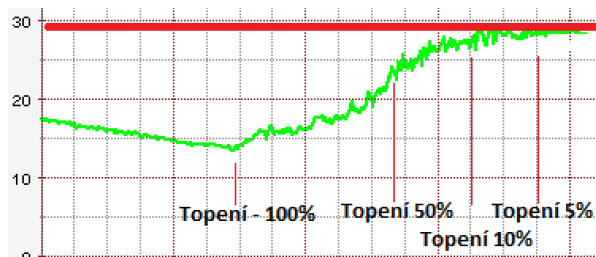
Regulace PWM (Pulse-width modulation) neboli pulzní šířková modulace je založena na procentuelní regulaci výkonu koncových prvků (přímotop, ventilátor). Například vezmeme časový úsek 10 sekund. Při výkonu 50% by 5 sekund byl na výstupu nastaveno 100 % výkonu a poté 5 sekund 0 % výkonu. Příklady PWM modulace je vidět na obrázku č. 2 (5V – logická jednička znamená 100% výkonu, 0V – logická nula znamená 0 % výkonu). Je zde vidět průběh logických úrovní závislých na čase.



Obr. 2 - PWM regulace [4]

Pro upřesnění funkce jsem vyrobil graf, na kterém je vidět regulace v praxi. Při poklesu teplot pod minimální teplotní mez (rozdíl aktuální a požadované teploty je velký) se sepe spínač na 100%. Teplota začne růst a procesor při překročení určité teploty vyhodnotí stav, aby nedošlo k „přetopení“ prostoru. Proto sníží výkon topení na 50% (tzn. začne střídavě zapínat a vypínat topení) a s rostoucí teplotou se dále výkon snižuje (50%→10%→5%). Tento

druh regulace je určitě výhodnější pro stabilitu teploty, ale je náročnější na programové vybavení.



Obr. 3 - graf regulace PWM

RaspberryPi má pro PWM připraven přímo jeden pin – GPIO18[4]

Hlavním úkolem práce bylo sestavit funkční zařízení, které bude měřit a regulovat teplotu bez ohledu na její průběh. Pro dosažení požadované teploty jsem zvolil jednodušší způsob regulace – dvoustavovou. Úpravou programového vybavení může být způsob regulace změněn na PWM.

4.5 Dallas DS18B20

DS18B20 je digitální teplotní čidlo schopné měřit teploty v rozsahu -55°C až 125°C s přesností 2°C . V rozsahu -10°C až 85°C by měla být přesnost $0,5^{\circ}\text{C}$. Je vyrábět nejčastěji v pouzdru TO-92 s třemi vývody – Vcc, Data, GND. Pracuje na sběrnici typu 1-Wire™, kterou využívá i RaspberryPi. Čidlo je schopné změřit teplotu za 750ms.

Všechna čidla 18B20 mají do své paměti ROM při výrobě zapsaný 64 bitové identifikační kód. Tento kód vždy začíná číslem 28, které bylo přiděleno společnosti Dallas. Celý kód vypadá kupříkladu 28-000003cee4ca. Díky tomuto ID je možné zapojit více čidel na jednu sběrnici a jednotlivé čidla volat. [5]

Čtení hodnoty čidla probíhá v několika fázích. Základem je zaslání jednotlivých příkazů.



Obr. 4 - Pouzdro DS18B20 [5]

5 Vlhkost

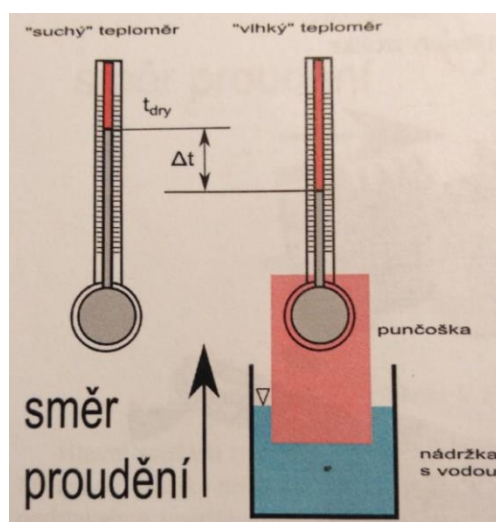
Vlhkost je základní vlastností vzduchu a udává množství vodní páry, která je v suchém vzduchu daného množství. Pro vyjádření vodních par ve vzduchu lze použít několik charakteristik – absolutní vlhkost vzduchu, tlak vodní páry, relativní vlhkost vzduchu, rosný bod apod.

Absolutní vlhkost vzduchu vypočítáme $\phi = \frac{m}{V} [kg/m^3]$, kde m je hmotnost vodní páry v daném objemu. Relativní vlhkost poté $\phi = \frac{p_v}{p_v''}$ kde p_v je parciální tlak vodní páry a p_v'' je parciální tlak vodní páry nasyceného vzduchu[2]

5.1 Měření vlhkosti

5.1.1 Psychrometr

Asi nejpřesnější měření vlhkosti lze dosáhnout psychrometrem. Je to velice jednoduchý princip měření relativní vlhkosti. Využívají se závislosti mezi relativní vlhkostí, teplotou a teplotou nasyceného vzduchu. Psychrometr se skládá ze dvou teploměrů nebo snímačů teploty.



Obr. 5 - Princip psychrometru [2]

Jeden z teploměrů, tzv. suchý, měří teplotu plynu, jehož vlhkost chceme měřit. Druhé tzv. vlhký, měří teplotu stejného plynu, ale při nasycení na 100% vlhkost. Maximální vlhkosti lze dosáhnout nasazením tenké „punčošky“ na vlhký teploměr tak, aby její konec byl v malé nádobě s vodou. Nádoba se při měření musí průběžně doplňovat. [2]

5.1.2. Vlhkoměr s pevným elektrolytem

Snímače vlhkosti tohoto typu využívají k měření změnu kapacity a elektrického odporu u oxidu hlinitého Al_2O_3 . Tento materiál vyniká pro své savé vlastnosti, a tedy velice dobře absorbuje vzdušnou vlhkost.

Snímač se skládá z vrstvy Al_2O_3 , která je uzavřena mezi dvěma elektrodami. Jedna elektroda je porézní, druhá pak neporézní. Porézní elektroda je vyrobena ze zlata.

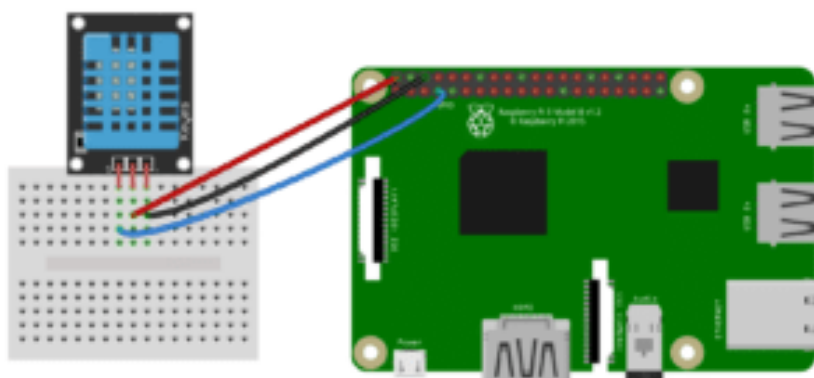
Otvory v této elektrodě umožňují vnikání vlhkosti do Al_2O_3 . Druhá elektroda je hliníková, neporézní. Vyhodnocuje se poté kapacita nebo elektrický odpor mezi elektrodami.

Protože je relativní vlhkost závislá i na teplotě, je obvyklé, že čidla měří zároveň i teplotu. Přesnost měření je přibližně 1 % - 5 %.[2]

5.2 Čidlo DHT11

Teploměr a vlhkoměr, který byl vybrán pro realizaci společně s mikroPC Raspberry Pi byl zvolen s ohledem na nízkou cenu a dle dostupných informací i přijatelnou přesnost pro toto měření. Výhodou byla také jednoduchost komunikace přes GPIO výstupy.

Teplotu měří v rozmezí 0 – 50 stupňů Celsia s přesností ± 1 °C a vlhkost v rozmezí 20 – 90 % s přesností ± 4 %.



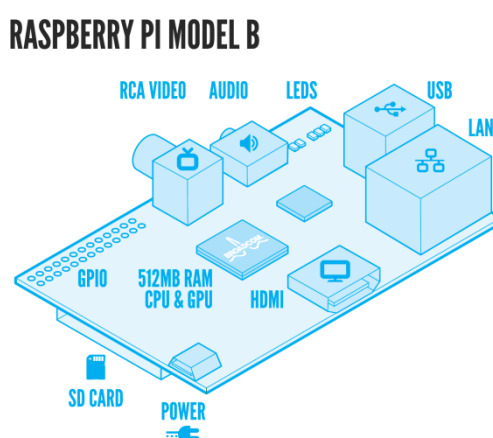
Obr. 6 - Připojení čidla [5]

Vzhledem k tomu, že dle prováděných měření při praktické části bylo zjištěno, že vlhkost je v boxu s hlívu ústřičnou neustále kolem 90% a více, nebylo potřeba dále vlhkost jakkoli regulovat či měřit.[5]

6 Technické a programové vybavení – Hardware a Software

6.1 Mikropočítač Raspberry Pi

Raspberry Pi je jednodeskový microPC se spoustou výhod nejen pro nováčky v oblasti systému Unix. Je vyvíjen britskou nadací Raspberry Pi Foundation pro rozšíření výuky informatiky na školách. První verze Raspberry Pi byla vydána v roce 2011. Měla však několik nedostatků, a proto 5. září 2012 vydána deska verze 2.0. Tento typ se vyrábí do dnes ve dvou provedeních – model A a model B. Verze A je ochuzena o síťovou kartu LAN a o jedno USB. Má nižší paměť RAM (256MB) a stojí asi o 20\$ méně.[6]



Obr. 7 - Blokové schéma Raspberry Pi [6]

6.1.1 Hardware

- Procesor - ARM1176JZF-S z rodiny ARM11 taktovaný na 700 Mhz
- Grafický procesor - VideoCore IV, podporující OpenGL ES 2.0, 1080p30, MPEG-4
- 512 MB RAM sdílených s grafickou kartou
- 2x USB port
- Obrazový výstup Composite RCA, HDMI, DSI
- Zvukový výstup přes 3,5 mm konektor, HDMI
- slot pro SD nebo MMC kartu
- Ethernetový adaptér 10/100 s konektorem RJ45
- 8xGPIO, UART, I²C, sběrnici SPI

K RaspberryPi je možné připojit USB WiFi přijímač, díky kterému se můžete nadálku připojit v Routeru mimo dosah drátů.[6]

6.1.2. OS Raspbian

Raspbian je linuxový operační systém vycházející z jádra starší verze OS Debian. Jedná se o upravenou verzi určenou přímo pro distribuci RaspberryPi.[6]

6.1.3 WiringPi

WiringPi je jednoduchá aplikace k ovládní programovatelných výstupů GPIO, kterou lze stáhnout na stránce <http://wiringpi.com/>. Díky aplikaci je možné pomocí jednoho příkazu v konzoli nastavit vstupní či výstupní parametry jednotlivých pinů GPIO. Těch je na RaspberryPi celkem 8.

Tento postup je pouze pro nainstalovaný systém Raspbian. Pro jiné systémy je třeba užít jiný postup uvedený <http://wiringpi.com/>.

Nainstalujeme systém GIT

```
apt-get install git-core
```

Pokud by RaspberryPi hlásilo nějaké chyby, zkuste zkontrolovat verzi

```
apt-get update
```

```
apt-get upgrade
```

Pokud je vše v pořádku, nainstalujte WiringPi pomocí příkazu

```
git clone git://git.drogon.net/wiringPi
```

Připravíme instalaci příkazy

```
cd wiringPi
```

```
git pull origin
```

Pro instalaci zadejte

```
cd wiringPi
```

```
./build
```

Nyní už jen stačí nastavit jednotlivé gpio piny do požadovaného módu, popřípadě výstupní hodnoty

```
gpio mode 0 out - První příkaz je zadání výstupního módu na pin GPIO 0
```

```
gpio write 0 1 - Zapiš na gpio 0 hodnotu 1
```

gpio write 0 0 - Zapiš na gpio 0 hodnotu 0

Těmito příkazy poté můžeme nastavovat kterýkoli pin gpio.[7]

6.1.4 Apache

Apache je softwarový webový server s otevřeným kódem pro platformy takřka všech operačních systémů. Jeho vývoj začal v roce 1993 National Center for Supercomputing Applications na Illinoiské univerzitě.

Instalace tohoto softwaru není složitá. Je třeba postupovat podle návodu, kterých je v dnešní době na internetu více než dostatek.[8]

6.1.5 Dvoustavová regulace – software

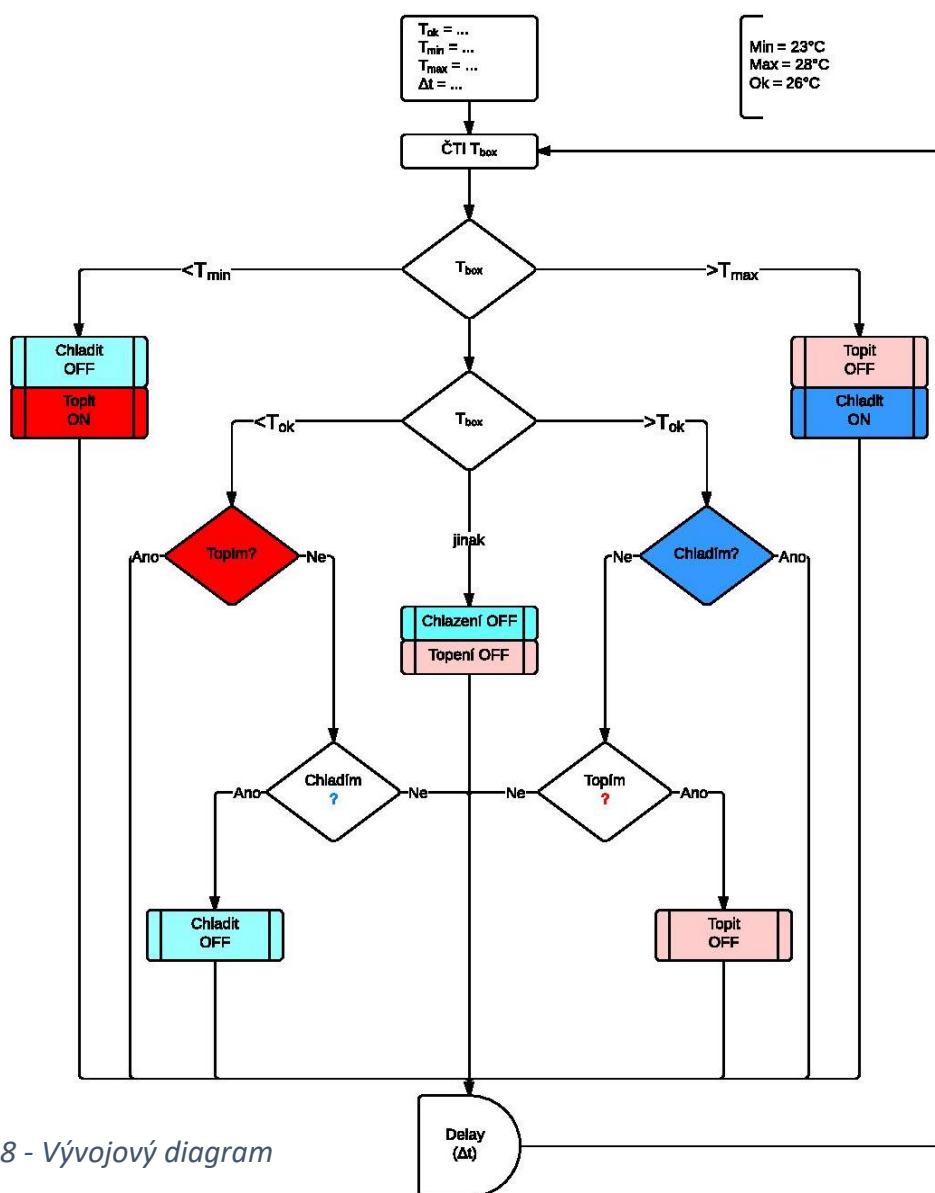
Pro správný chod Dvoustavová regulace bylo nutné vytvořit program, který by spolehlivě dokázal obsloužit jednotlivé stavy čidel umístěných v izolovaných boxech. Pro tento účel byl sestaven vývojový diagram (obr. č. 8).

Tento vývojový diagram je vytvořen tak, aby reagoval na jakoukoli možnou situaci uvnitř izolovaného boxu. Jedná se zejména o prudké změny teplot, kdy dojde například k otevření dveří a podobně.

Program byl napsán v jazyce Python a je přiložen na CD. Skládá se zejména z podmínek if/else, které porovnávají teploty importované z teplotních čidel DS18B20 a teploty zadané v uživatelském prostředí.

Použité symboly ve vývojovém diagramu

- Tok – Ideální teplota
- T_{min} – Spodní hranice teploty
- T_{max} – Horní hranice teploty
- T_{box} – Aktuální naměřená teplota
- Δt – Prodleva mezi měřeními
- ČTI T_{box} – Přečte aktuální teplotu čidla
- Topit ON/OFF – Zapnutí, vypnutí topení
- Chladit ON/OFF – Zapnutí, vypnutí chlazení
- Topím?/Chladím? – Zjištění aktuálního stavu topení, chlazení



Obr. 8 - Vývojový diagram

6.1.6. RRDTool

RRDtool (zkratka pro round-robin database tool) je nástroj, který se zaměřuje na zpracování a ukládání časově závislých dat, např. teplota. Tato data jsou uložena v databázi, která má konstantní velikost. To je zajištěno ukládáním dat po předem stanoveného časového rámce a v určitých časových intervalech. Např. data budou sledována v 24hodinovém časovém okně a každých 5 minut bude zaznamenána aktuální hodnota teploty, tzn., bude uloženo 288 hodnot teplot ($24 \times (60/5) = 288$).

RRDTool je softwarový nástroj, který obsahuje příkazy pro založení databáze, zápis dat, výpis uložených hodnot a tvorbu grafů z uložených hodnot. Užití jednotlivých příkazů je popsáno níže.

Založení databáze:

```
rrdtool create teplota-01.rrd --start N --step 60 \  
DS:temp:GAUGE:120:U:U \  
RRA:AVERAGE:0.5:1:60 \  
RRA:AVERAGE:0.5:1:1440 \  
RRA:AVERAGE:0.5:60:168 \  
RRA:AVERAGE:0.5:60:720 \  
RRA:AVERAGE:0.5:1440:365
```

Vytvářím soubor teplota-01.rrd, do kterého jsou každou minutu zapisovány hodnoty aktuální teploty a ukládány do časových oken: hodina, den, týden, měsíc a rok. Počátek ukládání dat je stanoven parametrem `-start N` na hodnotu „now“ (nyní). Nebude-li do 120 sekund zadána hodnota, bude zadána hodnota „unknown“ (neznámá-nezadaná).

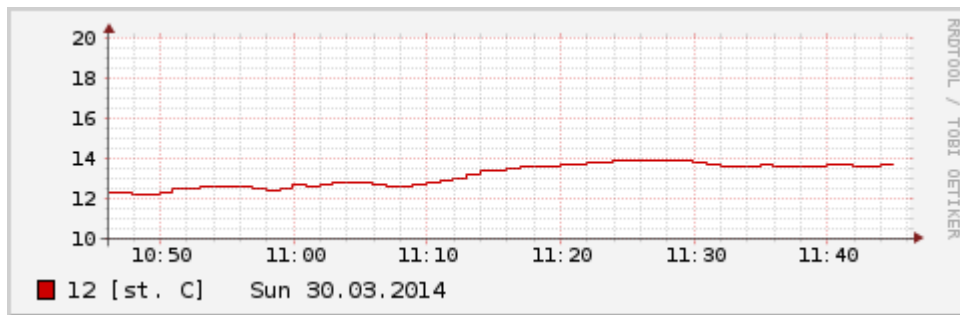
Zápis dat do databáze:

```
rrdtool update teplota-01.rrd N 23.7
```

Tvorba grafu:

```
rrdtool graph teplota-hodinova_cidlo-01.png --start -1h \  
DEF:temp=teplota-01.rrd:temp:AVERAGE \  
LINE:RED:"12 [st.C] Sun 30.03.2014"
```


Do souboru teplota-hodinova_cidlo-12.png je vygenerován graf teplot za poslední hodinu.



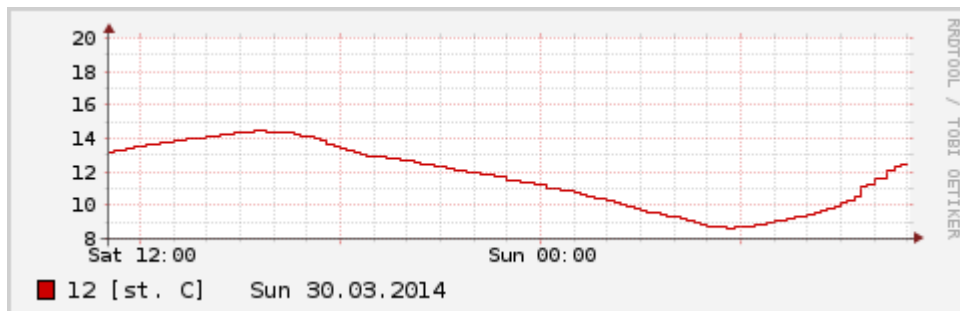
Obr. 9 - Hodinová teplota

Obdobným způsobem lze vygenerovat z uložených dat graf za posledních 24 hodin:

```
rrdtool graph teplota-denni_cidlo-12.png --start -1d \
```

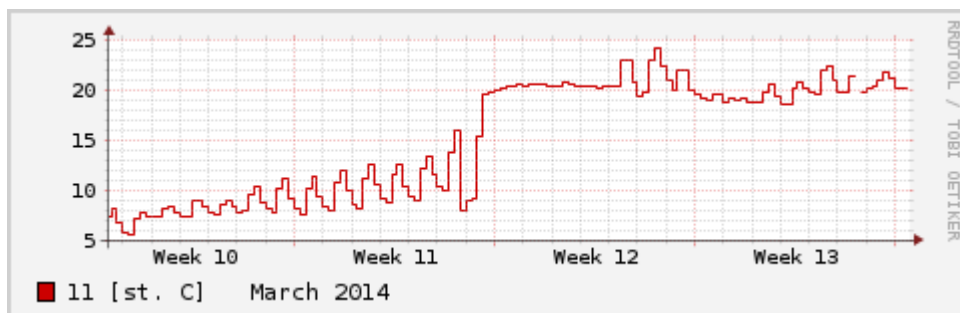
```
DEF:temp=teplota-01.rrd:temp:AVERAGE \
```

```
LINE:RED:"12 [st.C] Sun 30.03.2014"
```



Obr. 10 - Denní teplota

Na Obr. 8 je uveden graf měsíčního přehledu naměřených hodnot teplot čidla 11:



Obr. 11 - Měsíční teplota

V pravidelných intervalech jsou spouštěny skripty pro tvorbu grafů, které jsou použity v prezentaci naměřených hodnot na webovém serveru. Na grafu je znázorněný průběh sledování teploty v izolovaném boxu s hlívou ústřičnou.

6.2 Příkazy čidlu

0CCh

Procesor tento příkaz použije po resetu sběrnice. Jedná se o usnadnění komunikace, kdy procesor naváže sekvenci požadavků níže uvedených. Příkaz nemůže být použit zároveň se čtením dat, protože by mohlo dojít ke kolizi.

044h

Tento příkaz spustí převod teplot z teplotních snímačů do dvoubajtové paměti RAM. Po vykonání tohoto příkazu se čidlo vrátí do nízko odběrového režimu.

0BEh

Umožní přečtení paměti obvodu. Předun začíná nejnižším bajtem (nultým) přes celou paměť až ke konci, kde je uložen CRC kód. Při problémech přenosu je možné ukončit čtení pomocí příkazu reset.

Příklad sekvence pro měření teploty (jedno čidlo na sběrnici)

1. Reset sběrnice.
2. Příkaz 0CCh, Příkaz 044h – zahájíme měření
3. Časový limit 0,75 sekund potřebný pro načtení dat
4. Příkaz 0CCh, Příkaz 0BEh - obvod je připravený vyslat obsah svojí paměti na sběrnici.
5. Čtení bajtu ze sběrnice, bajt obsahuje spodních 8 bitů naměřených hodnot.
6. Čtení zbylých bajtů, synchronizace, CRC atd.

Je nutné zdůraznit, že reálné čtení pomocí sběrnice na RaspberryPi s OS Debian (Raspbian) je mnohem jednodušší. Poslední verze tohoto operačního systému je totiž na komunikaci s tímto čidlem připravena, a proto je nutné pouze aktivovat ovladače pro čtení teploty z čidla příkazy jako správce systému:

```
Sudo bash
modprobe w1-gpio
modprobe w1-therm
```

Poté už jen stačí čidla zavolat a spustit sekvenci čtení jejich hodnot. (viz obrázek 9)

```

pi@raspberrypi /sys/bus/wl/devices $ ls
28-000004c06738 28-000004cd5d4f wl_bus_master1
28-000004c38473 28-00000527bbdf
pi@raspberrypi /sys/bus/wl/devices $ cd 28-00000527bbdf
pi@raspberrypi /sys/bus/wl/devices/28-00000527bbdf $ ls
driver id name power subsystem uevent wl_slave
pi@raspberrypi /sys/bus/wl/devices/28-00000527bbdf $ cat wl_slave
c8 00 4b 46 7f ff 08 10 02 : crc=02 YES
c8 00 4b 46 7f ff 08 10 02 t=12500
pi@raspberrypi /sys/bus/wl/devices/28-00000527bbdf $

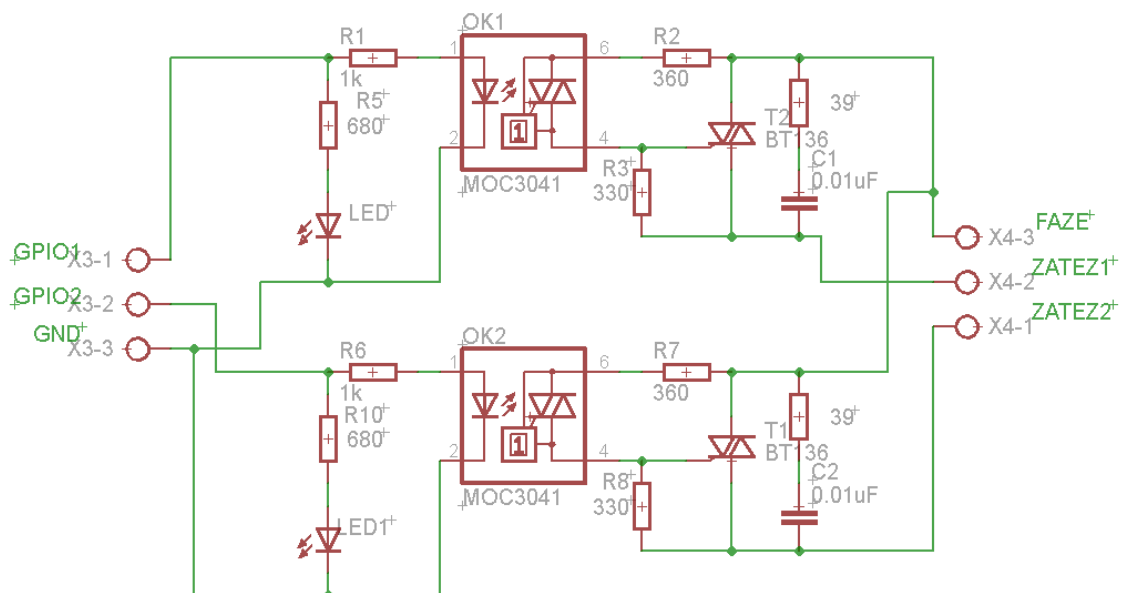
```

Obr. 12 - terminál putty, volání 18B20

6.3 Spínací prvky

6.3.1 Plošný spoj s optotriak (vlastní výroba)

MicroPC disponuje výstupním napětím na pinech GPIO 3,3V. Pomocí tohoto napětí je třeba zajistit sepnutí zařízení pro ohřev a chlazení (přímotop a ventilátor), které jsou napájeny 230V, 10A. Pro vyšší bezpečnost byl zvolen spínací prvek optotriak, který díky své konstrukci dokáže oddělit slaboproudou a silnoproudou část plošného spoje.



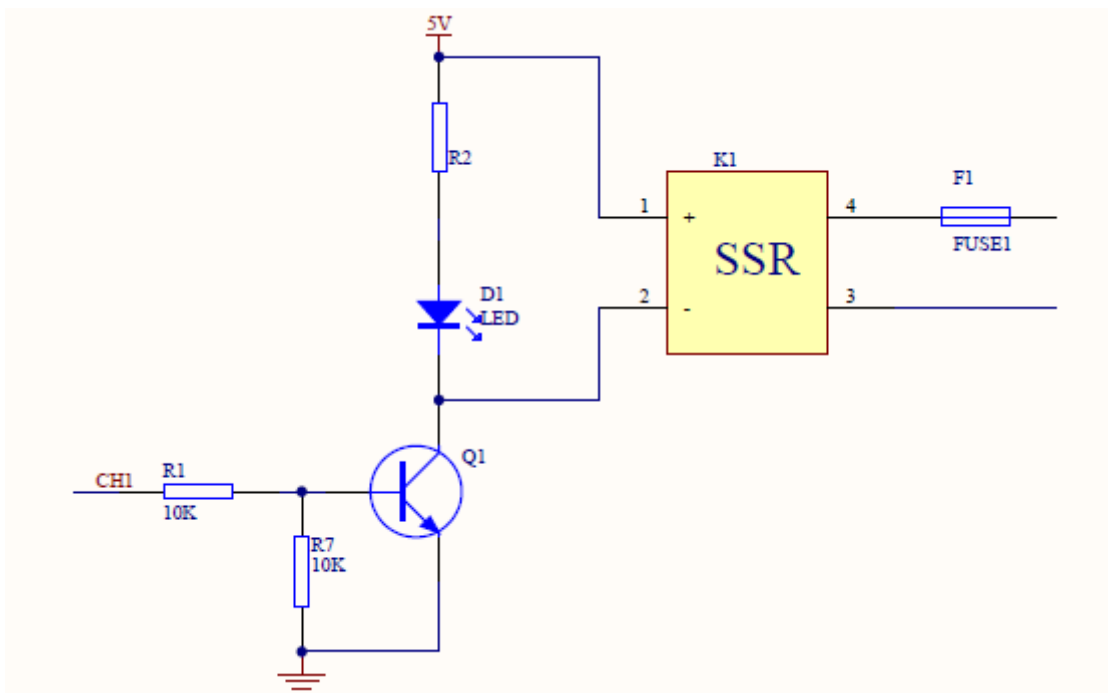
Obr. 13 - schéma plošného spoje

Malým proudem z GPIO rozsvěcujeme jednak kontrolní diodu LED a jednak diodu umístěnou v optotriaku MOC3041. Ta při aktivním stavu sepne optotriak, Slaboproudá a silnoproudá část je od sebe galvanicky oddělena. Proto nemůže dojít k poškození obvodu řídicího počítače. Triak umístěný v součástce MOC3041 poté propustí proud na řídicím vstupu (gate) výkonového triaku (v našem případě BT139 pro 16A), který sepne topící zařízení.

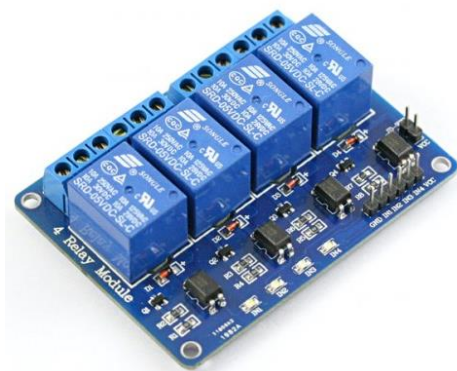
U tohoto zařízení však není zajištěna ochrana před úrazem elektrickým proudem, a proto bylo zvoleno sériově vyráběné zařízení SaintSmart relay module, u kterého je s ohledem na provedení vyšší stupeň bezpečnosti.

6.3.2 Sainsmart Relay Module

Relay modul je spínací prvek vyráběný společností Sainsmart. Je postaven na principu spínání síťových napětí a proudů (230V, 5A) pomocí malých proudů z RaspberryPi (3,3V, 10mA). Jedná se o sériově vyráběný prvek s elektronickým relé. [9]



Obr. 14 - SaintSmart relay module schéma [9]



Obr. 15 - SaintSmart relay [9]

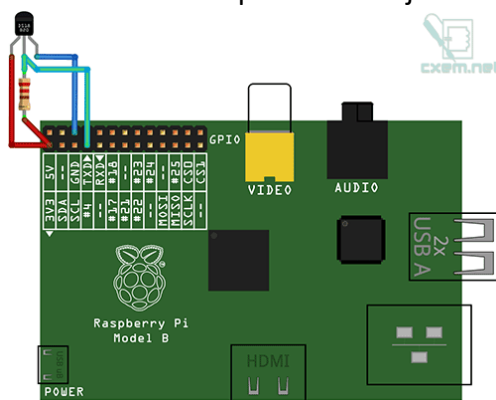
Princip spínání je podobný jako u doma vyrobené desky, jen na místo optotriaků je zde použit tranzistor a relé. V praxi je však modul mnohem užitečnější, je přímo přizpůsoben pro fungování s RaspberryPi. [9]

7 Praktická část implementace

7.1 Verze 1.0

Po zakoupení RaspberryPi a čidel Dallas 18B20 jsem začal testovat funkčnost, zaznamenávání údajů a případné problémy v komunikaci. Výhodou těchto čidel je hlášení o správnosti přenosu (kontrolní součet CRC). Při demonstračním zapojení na nepájivém poli nevznikal žádný problém a čidla bez ohledu na počet měřila a fungovala přesně, jak měla. Proto jsem vytvořil jednoduchý skript, který měl za úkol jednou za 20 sekund požádat čidlo "x" o hodnoty a zkontrolovat jeho CRC. V případě špatného CRC hodnotu nezapisovat (program naleznete v příloze). Postupně jsem přidával další čidla, vše proběhlo bez problému. Občasné chyby byly jen vypadlé drátky od čidel k RaspberryPi.

Po simulaci měření v laboratorních podmínkách jsem se rozhodl uvést RaspberryPi do



Obr. 16 - Schéma připojení čidla DS18B20 [5]

provozu. Pro první verzi měření bylo nutné monitorovat teplotu na 3 místech (3 různé boxy), přičemž RaspberryPi bylo umístěno na jedno z těchto míst. K ostatním bylo nutné natáhnout kabel, o délce cca 20m-40m dlouhý. Byl zvolen standardní čtyřpárový kabel, přičemž používané byly pouze dva páry. Fungovalo pouze jedno čidlo, které bylo připojeno přímo k výstupům. Ostatní dvě čidla měla úspěšnost přenosu cca 30%, a proto jsem začal hledat chybu. První co mě napadlo bylo nedostatečné napájení (3V3) z RaspberryPi. Proto jsem vzal stabilizovaný zdroj napájení 3,3V a pokusil se ho připojit k čidlům. Procento úspěšnosti vzrostlo asi na 50%, ale to samozřejmě stále nestačilo. Chyba nakonec byla ve špatném zapojení po párech (viz obr.17) a jejich zapojení zemí.

Úspěšnost přenosu poté vzrostla na 99% a hodnoty se zaznamenávaly správně do souboru .txt ve formátu <datum> <čas> <místo čidla> <teplota v °C> .

. (Připojení čidel DS18B20 - forum.cxem.net)

=> (2014-02-11 17:54:32 box-1 T 1 22.875 C)

7.2 Verze 1.1

Po dvoutýdenním provozu bylo objeveno několik nedostatků. Hlavní byl ve fyzickém připojení čidel k RaspberryPi. Drátky napojené pouze do nepájivého spojovacího pole při jakékoli manipulaci vypadly, a proto nebylo možné navázat spojení s čidly. Druhý problém byl také provoz ve velké vlhkosti, který ani čidlům ani RaspberryPi nesvědčil. Proto jsem objednal 10 čidel v kovovém pouzdru s metrovým kabelem. RaspberryPi jsem zavřel do pouzdra s minimálním přístupem vzduchu pro minimalizování přístupu vlhkosti.

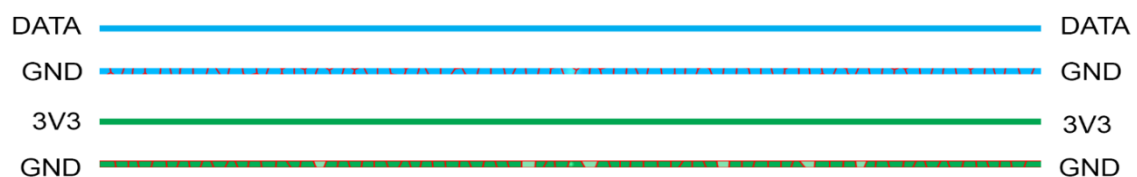
Po obdržení čidel jsem je postupně připojoval k RaspberryPi, abych zjistil jejich 12-ti místný unikátní kód (např.: 28-000004cd5d4f). Poté jsem začal řešit konektory, kterými bych zajistil pohodlné připojení k RaspberryPi bez problému špatného připojení vodičů. První myšlenka byla zakoupení 3.5 mm jack, který má 3 oddělené části. Vyřešil by se tím i problém s polaritou. Problém by však mohl vzniknout při zapojování jednotlivých čidel. Při fyzickém připojení totiž hrozí zkratování jednotlivých částí. Proto byla tato možnost vyloučena. Jako nejvhodnější se proto ukázaly klasické připojovací konektory PCB. Ty jsem přiletoval na předpřipravené pájivé pole.

7.3 Verze 2.0

Po vyřešení všech menších problémů s měřením bylo nutné vytvořit program na regulaci teploty. Před samotným programováním bylo nutné sestavit vývojový diagram, podle kterého budu postupovat (viz obrázek 4).

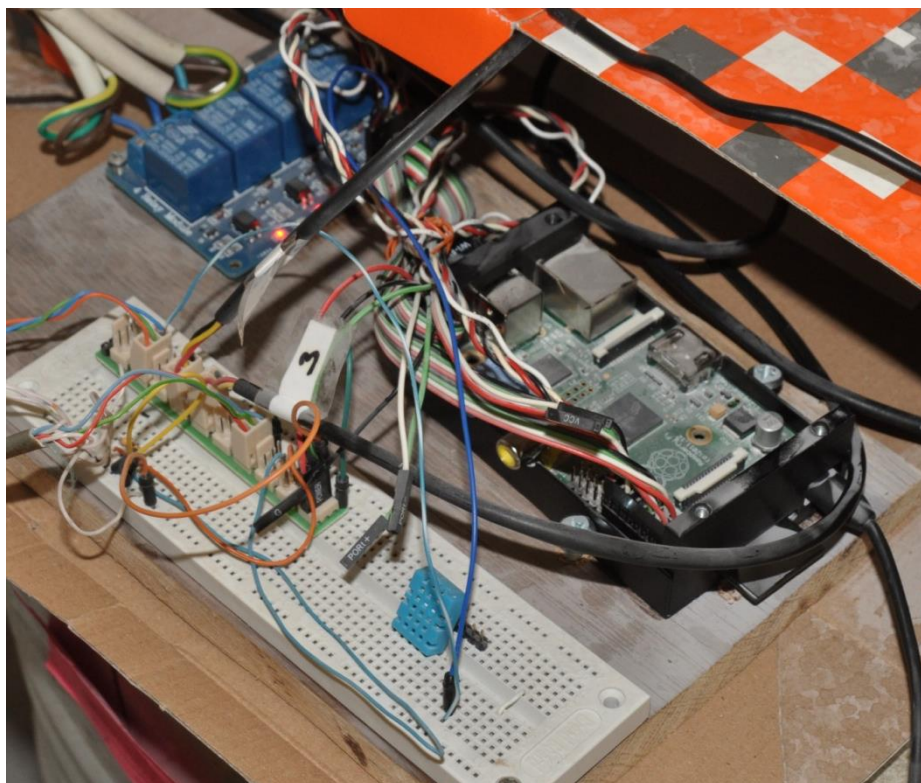
Pro ujasnění si projdeme krok po kroku vývojový diagram.

Manuálně zadáme 3 hodnoty teplot - $T_{ideální}$, $T_{maximální}$, $T_{minimální}$



Obr. 17 - Schéma zapojení párů

RaspberryPi požádá o teplotu z adresovaného čidla 18B20



Obr. 18 – Zkušební zapojení

Dostáváme se k prvnímu měření, řekněme, že v boxu je teplota 15°C. Program vyhodnotí, že $T_{\text{aktuální}}$ je menší než $T_{\text{minimální}}$ a pokračuje do levé větve diagramu.

Registry se nastaví na hodnoty topení - zapnout, chlazení - vypnout. Reálně se na výstupu předem nastaveného GPIO pinu nastaví 0 nebo 1. To posléze bude spínat 230V pro topení nebo klimatizaci.

Teď se koukněme kousek dál ->

RaspberryPi požádá o teplotu z adresovaného čidla 18B20

Měření stanoví teplotu na 27°C, program vyhodnotí, že $T_{\text{aktuální}}$ není ani nižší ani vyšší, a jde dál. Dostane se k porovnání, teplota je vyšší než T_{ok} -> dále se ptáme, chladím? -> NE, Topím? -> ANO -> vypneme topení a neděláme nic, dokud teplota opět nedosáhne hraniční hodnoty.

V procesoru je "topení on/off" realizováno výstupní 1/0 na jednom z GPIO pinů. Pro demonstraci jsem udělal testovací verzi s LED diodami, a jedním čidlem. Čidlo jsem střídavě zahříval a chladil podle potřeby a zkoušel jednotlivé fáze.

4.4 Verze 2.1

V dalším kroku bylo nutné připravit spínací mechanismus, který by pomocí 5V dokázal spínat 230V, 10A

Podle schématu jsem zhotovil návrh plošného spoje (viz obr. 13), který jsem zhotovil osvitovou cestou. Na první zapojení obvod bohužel krom rozsvícených signalizačních diod obvod nedělal vůbec nic. Vyzkoušel jsem proto postupně jednotlivé části obvodu - spínal jsem samostatně jen optotriak MOC3041, tak triak BT-139. Samostatně oba chodily bez problému, ale společně se mi je zapojit nepovedlo.

Začal jsem proto hledat náhradní řešení. Na internetu jsem objevil spínací prvky Sain Smart, které se přesně hodí k tomuto spínání. Jeho funkce je popsána výše. Stačilo tento modul připojit k RaspberryPi a ověřit jeho funkci. Díky optotriakům nebyl problém připojit modul přímo k mikroPC, který za běžných podmínek nemůže napájet externí moduly napřímo z důvodu maximálního odběru.

Pro simulaci jsem použil pouze větrák a objímku s žárovkou. Vše na první zapojení fungovalo. Přemístil jsem proto modul do provozní místnosti. Zvažoval jsem, jestli od modulu do izolovaných boxů natáhnu kabel na 230V a nebo celý modul dám do boxu a k RaspberryPi připojím jen pomocí čtyřpárového kabelu, stejně jako čidla Dallas 18B20.

4.5 Verze 3.0

Nyní už bylo nutné jen zvládnout regulaci teploty v uživatelském prostředí. Nechal jsem zařízení běžet několik týdnů, aby nasbíralo data pro grafy do databáze RRDtool, kterou jsem popsal výše. Databáze poté generuje za určitý časový úsek obrázky PNG (portable network graphics). Na tento obrázek grafu se odkazují v souboru index.html (základní stránka webového rozhraní). Zároveň bylo nutné naprogramovat zadávání teploty do systému. Zvolil



Obr. 19 - zadávání teplot

jsem programování v jazyce Python, který mi přijde asi nejvhodnější pro užívání na RaspberryPi. Po odladění již vše fungovalo, jak má a nebylo třeba dalších zásahů

8 Údržba a provoz systému jako celku

Celý výše popsaný systém je s ohledem na spínání spotřebičů s napájením 230V nutné udržovat v bezvadném provozuschopném stavu a vyvarovat se případným problémům z funkčností. Proto je potřeba vyhodnocovat jednotlivé menší i větší problémy a případně jim předcházet.

8.1 Údržba paměti (SD karta)

Do paměti RaspberryPi (SD karta) se denně zapisují hodnoty z jednotlivých čidel. Počet čidel nemusí být konstantní. Je možné přidávat jednotlivá čidla a měřit teplotní hodnoty na více místech. Data je dobré periodicky zálohovat na jiné paměťové médium s ohledem na spolehlivost SD karty a obsah dat jednou za týden. V tomto případě data stahují pomocí software WinSCP, kterým jednoduše překopíruji požadované soubory do adresáře záloh osobního počítače. S ohledem na volbu ukládání naměřených dat do RRD databáze není nutné provádět kontrolu obsazenosti paměti častěji než jednou za měsíc. Garantovaný počet zápisů na jednotlivé segmenty SD karty je udáván 100 000. Při periodě zápisu jedné hodnoty teploty za minutu je teoretická životnost segmentu karty cca půl roku. Při zápisu hodnot ze 4 čidel by byla teoretická životnost segmentu karty cca 1,5 měsíce. V provozu zapisují data za čtyř čidel na SD kartu již více než 4 měsíce bez jakékoliv známky problémů nebo poškození karty.

8.2 Kontrola funkce

Kontrolu funkčnosti systému provádíme denně zhlédnutím grafů na webovém serveru. Po zobrazení stránky s grafy a regulací na serveru Apache je možné pozorovat případné výpadky měření jednotlivých čidel a přesahy teplot mimo stanovené meze. Změna teploty a jeho regulace směrem k ideální hodnotě je patrná z průběhu naměřených teplot. Nejčastější chyba je ztráta spojení s routerem pomocí WiFi rozhraní. Ta se řeší restartem mikropočítače bez dalších problémů.

9 Závěr

Projekt byl řešen v průběhu půl roku. Práce s mikroPC RaspberryPi se více než osvědčila, a proto toho řešení mohu doporučit. Do budoucna by bylo vhodné zapojit regulaci teploty PWM, kterou by bylo možné dosáhnout lepších finálních výsledků a stability teplot.

Regulace vlhkosti nebyla řešena prakticky z důvodu neustále vysokých hodnot, které byly způsobeny pasterizací substrátu pro pěstování hlívy ústříčné.

- [1] JENČÍK, Josef a Jaromír VOLF. *Technická měření*. Praha: Vydavatelství ČVUT, 2000. ISBN 80-01-02138-6.
- [2] NOVÁK, Martin. *Technická měření*. Praha: Nakladatelství ČVUT, 2018. ISBN 978-80-01-06388-0.
- [3] *Wikipedie - otevřená encyklopedie* [online]. 2018 [cit. 2018-03-28]. Dostupné z: <https://cs.wikipedia.org/wiki/Teplota>
- [4] *Arduino: PWM* [online]. 2018 [cit. 2018-03-28]. Dostupné z: <http://arduino.cc/en/Tutorial/PWM>
- [5] *Circuit Basics: Raspberry Pi DS18B20* [online]. 2018 [cit. 2018-03-28]. Dostupné z: <http://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/>
- [6] *Webový server s Raspberry Pi: Část 1: Hardware* [online]. 2018 [cit. 2018-03-28]. Dostupné z: <http://vojtech.myslivec.net/webovy-server-s-raspberry-pi-cast-1-hardware>
- [7] *Wiring Pi: GPIO Interface library for the Raspberry Pi* [online]. 2018 [cit. 2018-03-28]. Dostupné z: <http://wiringpi.com/>
- [8] *Wikipedie - otevřená encyklopedie: Apache HTTP Server* [online]. 2018 [cit. 2018-03-28]. Dostupné z: https://cs.wikipedia.org/wiki/Apache_HTTP_Server
- [9] *Sain Smart: 4-Channel 5V Relay Module* [online]. Nedatováno [cit. 2018-03-28]. Dostupné z: <https://www.sainsmart.com/products/4-channel-5v-relay-module>