

Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky

System domácej automatizácie

Semestrálny projekt

Meno: Marek Slaninka
Študijný program: MSUS
Predmet: Vnorené systémy
Ročník: 1. ING
Akademický rok: 2015/2016

Obsah

1. Zadanie	3
2. Ciele projektu	4
3. Opis hardvérového riešenia	4
3.1. Použité súčiastky.....	4
3.2. Zapojenie teplomerov	7
3.3. Zapojenie diaľkového ovládača	8
3.4. Schéma celkového zapojenia.....	8
4. Implementácia softvérového riešenia na Raspberry Pi	9
4.1. Operačný systém pre Raspberry Pi	9
4.2. Načítanie a uloženie teplôt	10
4.3. Prístup k údajom v DB zo siete	12
4.4. Otváranie brány zo siete	14
5. Implementácia mobilných aplikácií	14
5.1. Aplikácia myHome	14
5.2. Aplikácia Gate.....	16
6. Záver	17

1. Zadanie

Navrhňte a zrealizujte zariadenie, založené na Raspberry Pi, ktoré bude zaznamenávať teploty zo štyroch teplomerov, rozmiestnených vonku, v bojleri, na výstupnej rúre z kotla a vo vnútri domu. Teploty budú merané automaticky každú minútu a ukladané spolu s časom do lokálnej databázy. Na systéme bude spustený server, ktorý na vyžiadanie odošle teploty v zadanom rozsahu cez GET dotaz. Ďalej bude na zariadenie pomocou relé dosky pripojený ovládač od brány, ktorý bude ovládaný cez POST dotaz. Na komunikáciu so zariadením bude spravená iOS aplikácia, s autentifikáciou pomocou odtlačku prsta a s možnosťou zobrazovať namerané teploty a manuálnym spúšťaním bojlera.

Projekt musí obsahovať:

1. Ciele projektu
2. Opis hardvérového riešenia
3. Implementáciu
4. Pamäťové médium obsahujúce softvérové časti riešenia spolu so všetkou dokumentáciou

2. Ciele projektu

Cieľom projektu bolo zhotoviť systém, s využitím vnorených systémov a reálnym použitím v našej domácnosti. Pri kúrení kotlom na tuhé palivo s manuálnym ovládaním môže nastať situácia, pri ktorej sa kotol prehreje a nevydrží tlak vody, následkom čoho nastáva explózia kotla. K takémuto scenáru môže veľmi ľahko prísť, keď sa v kotly zakúri a zabudne sa kontrolovať. Pre ľahšiu kontrolu som využil teplomer pripojený ku Raspberry Pi, ktorý teplotu sníma a je možné k nej pristupovať po sieti. Keďže Raspberry zvládne toho oveľa viac, tak som naňho pripojil ďalšie dva teplomery, ktoré snímajú vonkajšiu teplotu a teplotu v bojleri.

Zároveň som využil aj polohu zariadenia v blízkosti brány a presunul som k nemu ovládač na otváranie brány spínaný pomocou relé. Predchádzajúce umiestnenie bolo umiestnené za niekoľko stenami a bol problém s dosahom.

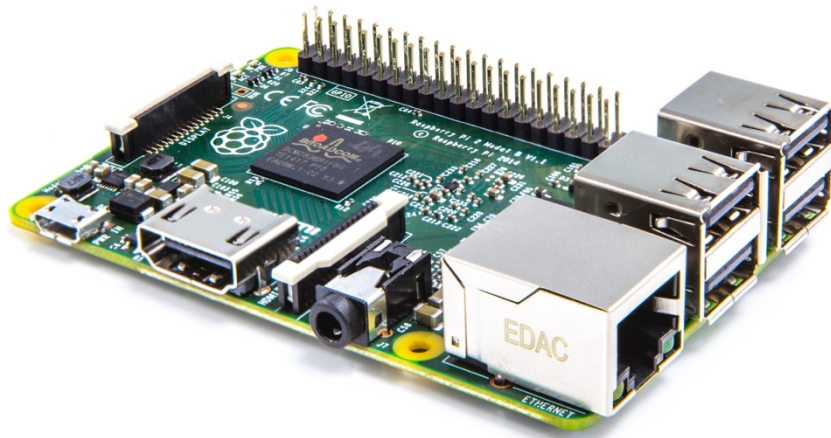
3. Opis hardvérového riešenia

3.1. Použité súčiastky

Raspberry Pi 2 model B: Malé jednodoskové zariadenie, založené na ARM procesore od spoločnosti Broadcom so štvormi jadrami taktovanými na 900MHz. Operačná pamäť má kapacitu 1GB. Veľkými výhodami pri tomto zariadení voči iným podobným zariadeniam je ľahká dostupnosť, vysoký počet doplnkových zariadení či škatuliek a rozsiahla komunita na internete. Ako Operačný systém je použitá linuxová distribúcia Raspbian, no je možné použiť aj novú verziu OS Windows IoT. Ja som sa rozhodol pre Raspbian, pre dostupnosť riešení a návodov, keďže Windows je len na začiatku vývoja pre IoT zariadenia.

Špecifikácia:

- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- 10/100 Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core



Dallas DS18B20: Senzor teploty používa na komunikáciu 1-Wire zbernicu. Umožňuje pripojiť až 150 zariadení s unikátnym sériovým číslom na jednu zbernicu. Teplomér som vybral pre jeho vysoký rozsah teplôt, ktoré použijem pri meraní vonkajšej teploty, keď bude klesať hlboko pod bod mrazu a zátrovenň pre meranie teplôt v bojleri a na kotli, kde dosahujú maximálne teploty vyše 70 °C. Je možné ho kúpiť aj vo vodotesnom prevedení na dlhšom trojžilovom alebo dvojžilovom kábli. Pri dvojžilovom zapojení slúži jeden vodič aj ako napájací aj ako Data a druhý ako GND. Ja som sa pre bezproblémovosť prevádzky, a pre zvýšenú odolnosť voči poveternostným vplyvom rozhodol pre vodotesné trojžilové prevedenie tohto teplomera. Na jednotlivé žily som prispájkoval piny typu samica pre jednoduchšie pripájanie a odpájanie od zapojenia.

- napájanie je 3,0V – 5,5V
- rozsah meranej teploty od -55°C do $+125^{\circ}\text{C}$ (-67°F do $+257^{\circ}\text{F}$)
- presnosť $0,5^{\circ}\text{C}$ v rozmedzí -10°C do $+85^{\circ}\text{C}$
- meranie teploty s rozlíšením 12 bit trvá 750ms



2 Channel Relay Board: Modul s dvomi relé zapojenými cez optočleny, ktoré slúžia ako ochrana ovládacieho zariadenia pred prerazeným prúdom. Modul má dva vstupy pre napájanie, dva piny pre GND a dva ovládacie piny. Je možné ho napájať jedným zdrojom napätia z Raspberry, ale riadiacu časť zvlášť a relé zvlášť. Toto slúži ako ďalšia ochrana pred nechceným prierazom napätia.

- Ovládacie napätie je 3,3V až 5V
- Maximálne spínacie napätie je 250V AC
- Maximálny spínaný prúd je 10A



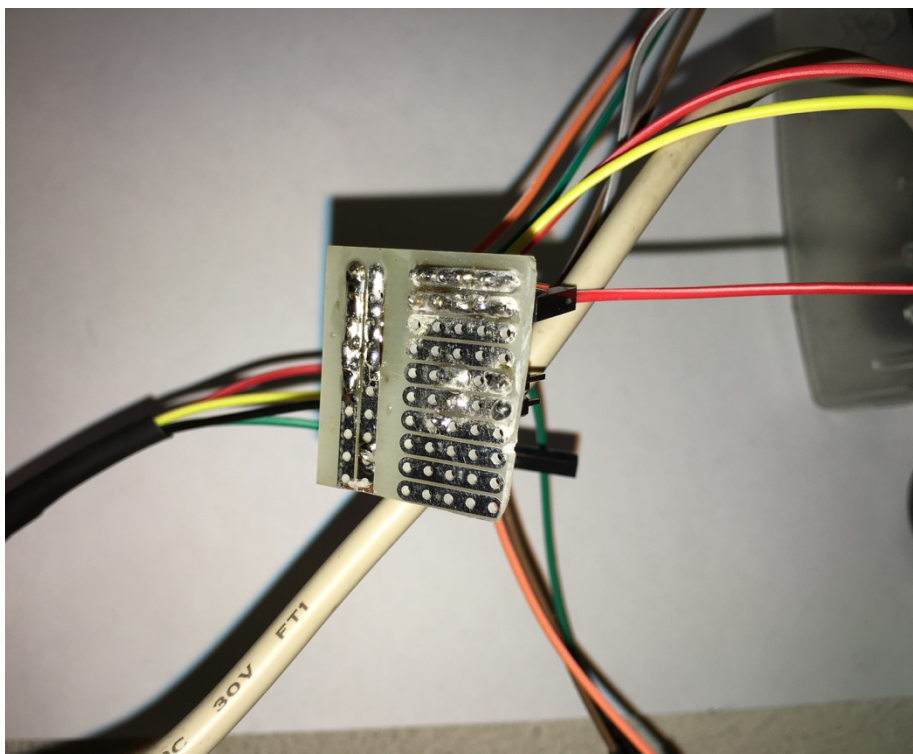
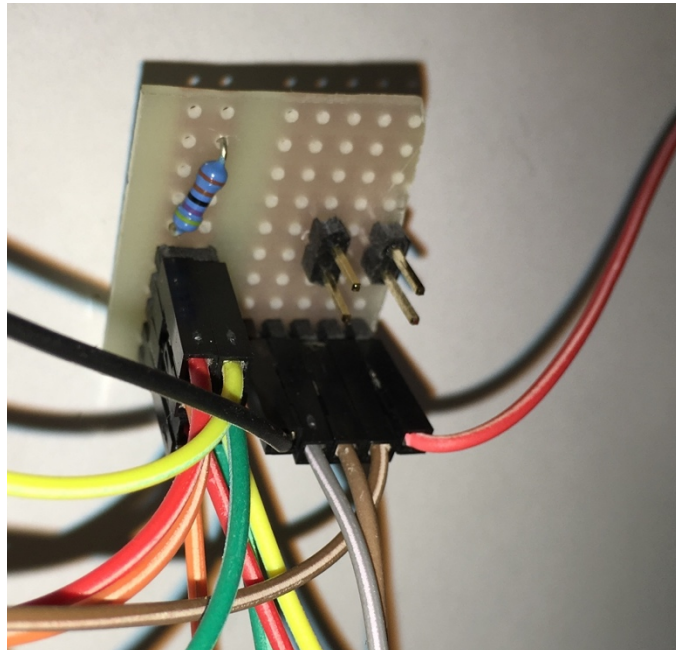
Doska plošných spojov: Pre jednoduchšie zapájanie, rozširovanie a výmenu súčiastok som využil predpripravenú dosku plošných spojov, na ktorú som prispájkoval piny typu samec a spojil ich päť za sebou v troch oddelených radoch. Jeden rad slúži ako GND, druhý ako napájanie a tretí spája dátové vodiče z teplomerov.

Dial'kové ovládanie k pohonu brány: Poslednou použitou súčiastkou bolo dial'kové ovládanie značky Key



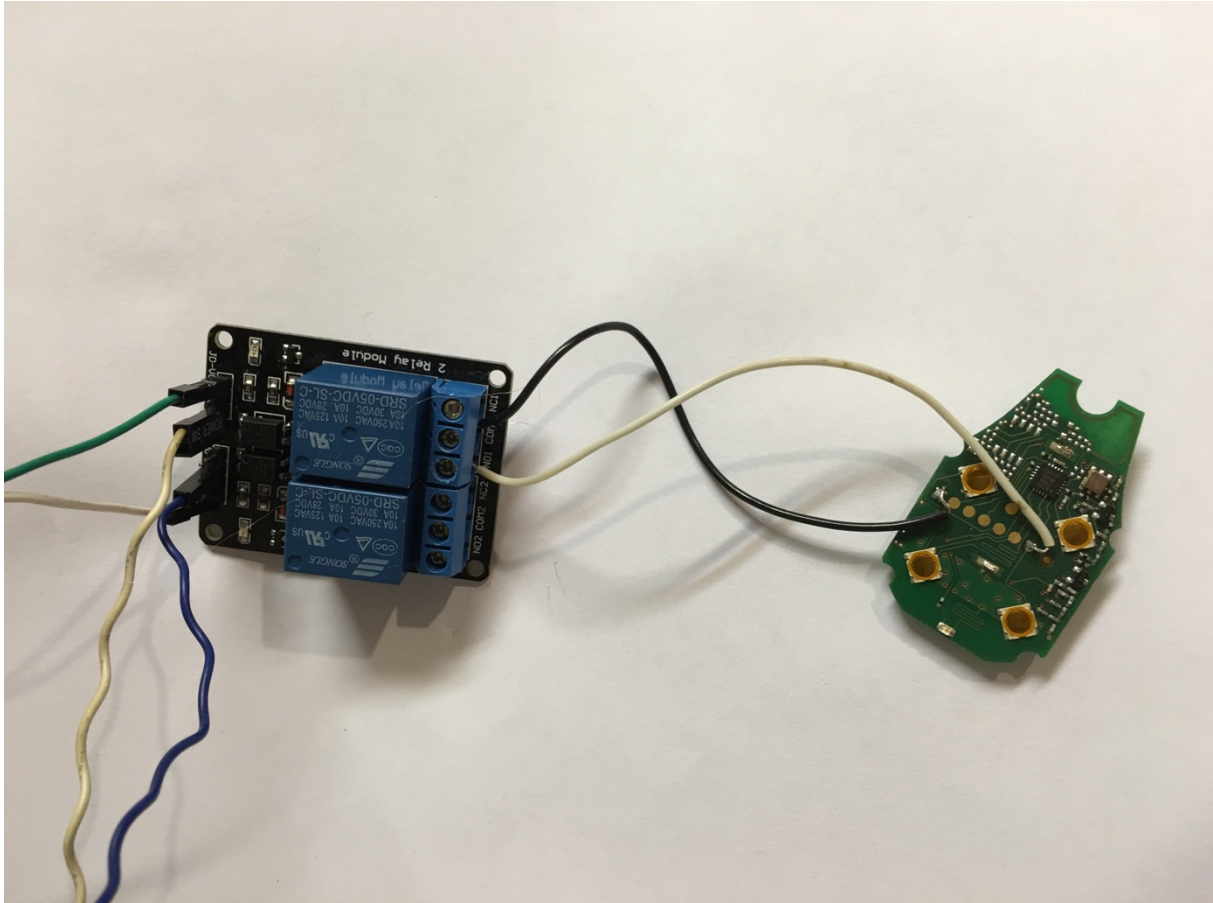
3.2. Zapojenie teplomerov

Na zapojenie viac zariadení na jeden pin som použil dosku plošných spojov, do ktorej som zapojil príslušné teplomery a dva rady pinov prislúchajúcich napájaniu a dátovému vodiču som prepojil rezistorom s odporom 4,7 k Ω . Kvôli vzdialenosti ku cieľovým umiesteniam som musel použiť predlžovacie káble. Na toto mi poslúžili nepoužité sieťové UTP káble, z ktorých som použil iba tri potrebné žily. Doska plošných spojov bola prepojená tromi káblami s Raspberry Pi. Pin 1 3V3 slúži ako napájací pin, pin s označením 6 GND slúži ako zem a pin s číslom 7 s označením GPIO4 slúži ako dátový pin.

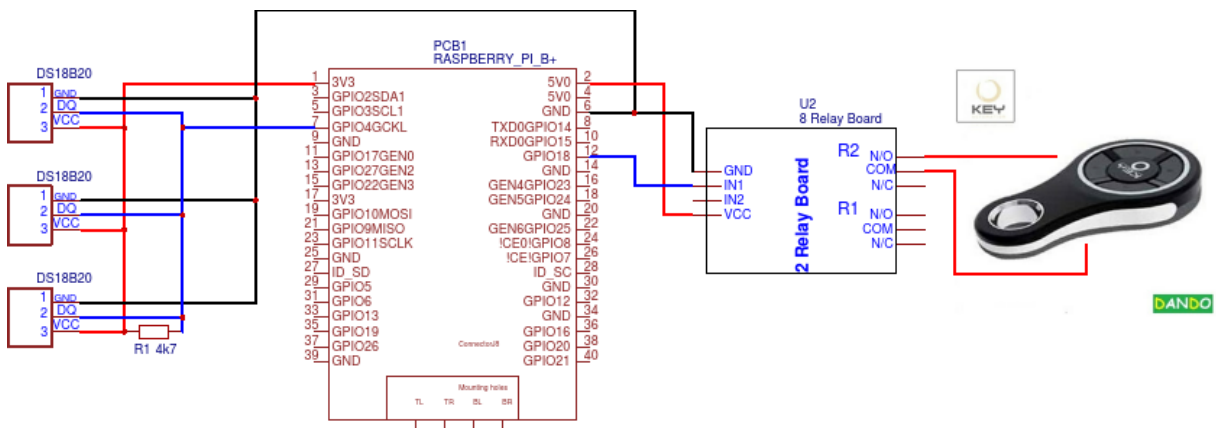


3.3. Zapojenie diaľkového ovládača

Na napájanie dosky s relé je použitý pin číslo 5 s označením 5V0. Ako zem je použitý rovnaký pin ako pre teplomery číslo 6 s označením GND. Ako riadiaci pin ktorým ovládam relé je použitý pin číslo 12 s označením GPIO18.



3.4. Schéma celkového zapojenia



4. Implementácia softvérového riešenia na Raspberry Pi

4.1. Operačný systém pre Raspberry Pi

Ako prvé bolo potrebné nainštalovať operačný systém Raspbian, ktorý je postavený na distribúcii Debian a je optimalizovaný pre hardvér zariadenia Raspberry Pi. Výber padol práve na tento systém, pre jeho aktívny vývoj, dobrú optimalizáciu, podporu komunity a ľahkou inštaláciou z vyše 35000 predpripravených softvérových balíčkov.

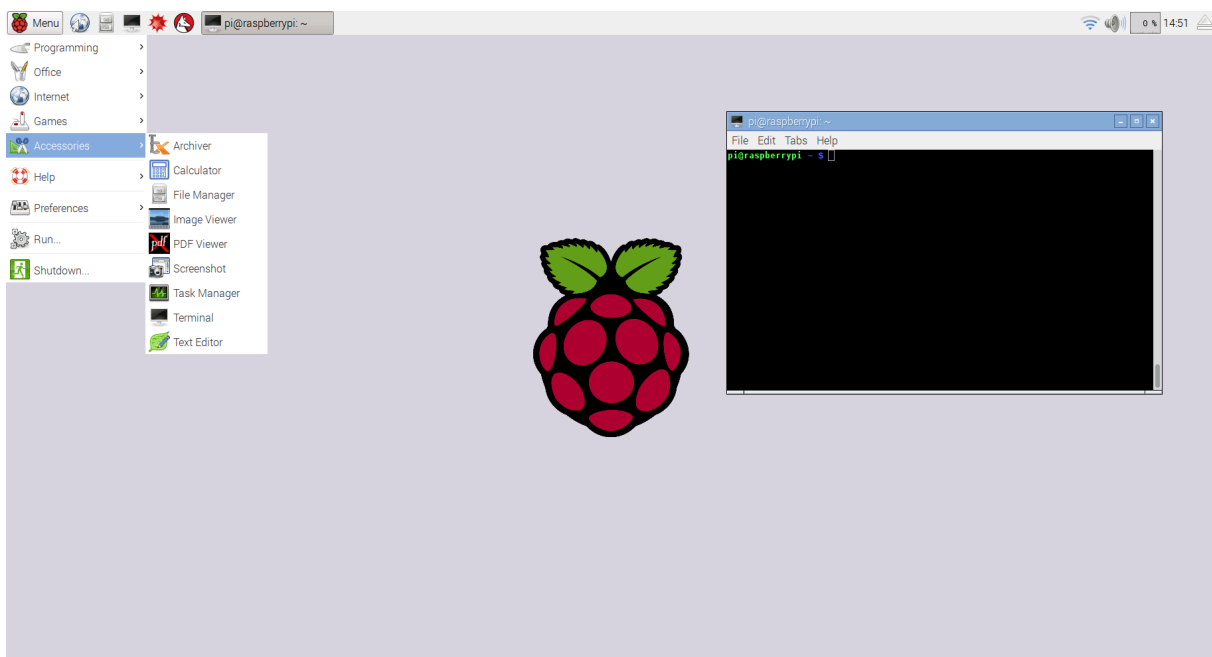
Na inštaláciu je predpripravený ZIP balíček, ktorý obsahuje inštalátor operačného systému. Po jeho stiahnutí a rozbalení na čistú microSD kartu, ju stačí zasunúť do zariadenia, pripojiť monitor cez hdmi kábel a pripojiť napájanie.

Inštalátor ponúkne viacero systémov na inštaláciu, no ja som vybral spomínaný Raspbian.

Po inštalácii nasleduje prvé prihlásenie. Prednastavné prihlasovacie údaje sú pi a heslo raspberry. Po prvom príkaze

`startx`

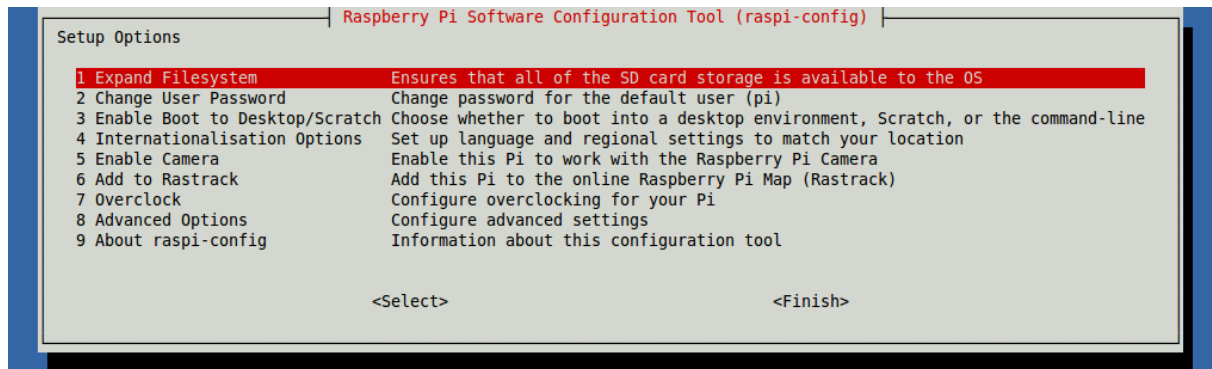
sa naštartuje vizuálne prostredie operačného systému.



Pre vzdialený prístup do systému som v nastaveniach pomocou príkazu

`sudo raspi-config`

zapol ssh (secure shell) čo nám umožňuje pripojenie cez príkazový riadok z iného počítača na sieť.



Načítanie teploty

Pre overenie funkčnosti pripojených teplomerov stačí zadať do príkazového riadku príkaz

```
ls -l /sys/bus/w1/devices/
```

ktorý vypíše všetky propojené zariadenia cez 1wire zbernicu aj s ich identifikačným číslom. V našom prípade :

```
pi@raspberrypi:~ $ ls -l /sys/bus/w1/devices/
total 0
lrwxrwxrwx 1 root root 0 May 14 10:32 28-021564b0d3ff -> ../../../../devices/w1_bus_master1/28-021564b0d3ff
lrwxrwxrwx 1 root root 0 May 14 10:32 28-021564cfccff -> ../../../../devices/w1_bus_master1/28-021564cfccff
lrwxrwxrwx 1 root root 0 May 14 10:32 28-021564f2ecff -> ../../../../devices/w1_bus_master1/28-021564f2ecff
```

Ku nameranej teplote dokážeme pristúpiť pomocou príkazu

```
cat /sys/bus/w1/devices/28-021564b0d3ff/w1_slave
```

ktorý nám prečíta hodnoty z teplomera. V druhom riadku výstupu sa nachádza načítaná teplota vynásobená 1000. Po jej predelení dostaneme aktuálnu teplotu nameranú teplomerom v stupňoch Celzia. V našom prípade $43562/1000 = 43,562$ °C.

```
pi@raspberrypi:/sys/bus/w1/devices/28-021564b0d3ff $ cat /sys/bus/w1/devices/28-021564b0d3ff/w1_slave
b9 02 4b 46 7f ff 0c 10 b5 : crc=b5 YES
b9 02 4b 46 7f ff 0c 10 b5 t=43562
```

4.2. Načítanie a uloženie teplôt

Pre pravidelne opakované načítavanie a ukladanie všetkých teplôt a aktuálneho času som potreboval vytvoriť skript, ktorý bude spúšťaný pomocou cronu.

Ako prvé som nainštaloval mysql databázu pomocou príkazu

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

a doplnkovú knižnicu pre programovací jazyk Python pomocou

```
sudo apt-get install python-mysqldb
```

Po úspešnej inštalácii bolo treba prihlásiť sa do mysql

```
sudo mysql -u root -p
```

Ďalším krokom bolo vytvorenie našej databázy príkazom a začat ju používať:

```
CREATE DATABASE temp_database;
```

```
USE temp_database;
```

Na zaznamenanie troch teplôt a aktuálneho času a ako predprípravu pre neskoršie dodanie ssenzora vlhkosti bolo treba vytvoriť tabuľku s päť stĺpcami

```
CREATE TABLE tempLog(datetime DATETIME NOT NULL, tempKotol FLOAT(5,2) NOT NULL, , tempBojler FLOAT(5,2) NOT NULL, tempVonku FLOAT(5,2) NOT NULL, humidity FLOAT);
```

A výsledkom bola tabuľka:

Field	Type	Null	Key	Default	Extra
datetime	datetime	NO		NULL	
tempKotol	float(5,2)	NO		NULL	
tempBojler	float(5,2)	NO		NULL	
tempVonku	float(5,2)	NO		NULL	
humidity	float	YES		NULL	

Načítavanie teplôt funguje pomocou python skriptu spúšťaného unixovým plánovačom Crontab.

Ako prvé skontroluje koľko zariadení je pripojených cez 1Wire zbernicu.

```
allTemp = glob.glob(base_dir + '28*')
nTemp = len(allTemp)
```

Z každého pripojeného zariadenia zistíme jeho jedinečný identifikátor, ktorý zakomponujeme do cesty k teplote a prečítame teplotu, predelíme číslom 1000 a vložíme na koniec pola s teplotami. Po úspešnom načítaní teplôt do pola, zistíme presný čas a uložíme ho do premennej. Ako posledný krok zostrojíme insert dopyt na mysql databázu a vykonáme ho nad pripojenou databázou:

```
db = MySQLdb.connect(host="localhost", user="****", passwd="****",
db="temp_database")
cur = db.cursor()
sql = ("""INSERT INTO tempLog
(datetime,tempBojler,tempVonku,tempKotol) VALUES
(%s,%s,%s,%s)""", (datetimeWrite,pole[0],pole[1],pole[2]))
cur.execute(*sql)
db.commit()
```

Po uložení sa odpojí databáza a program sa ukončí.

Na naplánovanie programu v Crontabe použijeme príkaz

```
crontab -e
```

a na koniec otvoreného súboru zapíšeme riadok

```
*/5 * * * * /home/pi/tempLog/thermoWithLog.py
```

ktorý hovorí systému aby spustil program na ceste /home/pi/tempLog/thermoWithLog.py každých 5 minút. Pri tomto stave ale ešte nieje program spustiteľný crontabom, preto je nutné na prvý riadok program pridať text:

```
#!/usr/bin/env python
```

a zmeniť zapisovacie práva na náš program pomocou

```
sudo chmod +x readTempSQL.py
```

4.3. Prístup k údajom v DB zo siete

Na prístup k nameraným teplotám uloženým v databáze potrebujeme nainštalovať webserver a vytvoriť php súbor, ktorý načítá údaje a vráti v tvare JSON.

LIGHTTPD server som vybral pre jeho nenáročnosť na hardvér a jednoduchosť inštalácie. Pre plnú funkčnosť servera zadáme tieto príkazy:
Inštalácia lighttpd

```
sudo apt-get -y install lighttpd
```

Inštalácia PHP

```
sudo apt-get -y install php5-common php5-cgi php5
```

Inštalácia FastCGI pre prácu s php

```
sudo lighty-enable-mod fastcgi-php
```

Reštartovanie lighttpd server pre znovu načítanie všetkých modulov

```
sudo service lighttpd force-reload
```

Úspešnosť inštalácie otestujeme pomocou zadania ip adresy zariadenia do prehliadača a mala by sa nám zobraziť stránka s nadpisom Placeholder page.

tempDbJSON.php súbor slúži pre načítanie záznamov podľa GET parametrov a vrátenie v tvare JSON.

Ako prvé skontroluje, či je nastavený GET parameter state na preddefinovaný string. Toto slúži pre zvýšenie bezpečnosti. Ak sa parameter zhoduje s preddefinovaným,

php skript pokračuje a zistí či má zadané parametre startDate a endDate. Ak sú tak vytvorí sql dopyt z parametrov

```
$query = "SELECT * FROM tempLog WHERE datetime >= '$startDate'  
AND datetime <= '$endDate'";
```

a ak nie sú nastavené tak vytvorí dopyt

```
$query = "SELECT * FROM tempLog";
```

V ďalšom kroku načíta údaje potrebné na pripojenie ku databáze z konfiguračného súboru, ktorý je umiestnený mimo priečinka z php skriptom pre zvýšenú bezpečnosť.

Pomocou načítaných údajov vytvorí spojenie z databázou a vykoná pripravený dotaz.

```
$host = $config['database']['host'];  
$user = $config['database']['user'];  
$pass = $config['database']['password'];  
$database = $config['database']['database'];  
$con = mysqli_connect($host,$user,$pass);  
@mysqli_select_db($con, $database) or die("Unable to select  
database");
```

```
$result = mysqli_query($con,$query);
```

Vytiahnuté údaje vo while cykle sparsuje a vloží ich do pola. Toto pole potom vráti ako odpoveď vo formáte JSON.

```
while ($row = mysqli_fetch_assoc($result))  
{  
    $dateAndTemps = array();  
    $datetime = $row["datetime"];  
    $tempKotol = $row["tempKotol"];  
    $tempBojler = $row["tempBojler"];  
    $tempVonku = $row["tempVonku"];  
  
    $dateAndTemps["Date"] = $datetime;  
    $dateAndTemps["tempKotol"] = $tempKotol;  
    $dateAndTemps["tempBojler"] = $tempBojler;  
    $dateAndTemps["tempVonku"] = $tempVonku;  
  
    array_push($tempValues, $dateAndTemps);  
}  
echo json_encode($tempValues);
```

4.4. Otváranie brány zo siete

Na otváranie brány slúži ďalší php skript `openGate.php`, ktorý po zavolaní zistí či má nastavené dva POST parametre `state` a `heslo`. Ak sú nastavené tak skontroluje či `state` parameter má správny string a porovná parameter `heslo` z heslom načítaným zo súboru umiestneného mimo priečinka dostupného zo siete. Ak je všetko v poriadku tak pomocou príkazu

```
exec('sudo python opengate.py', $out);
```

spustí python skript `opengate.py` a výstup zapíše do premennej `out`. Túto premennú vráti v JSON tvare, aby malo zariadenie odpoveď, či sa podarilo spustiť skript.

Opengate.py súbor obsahuje príkazy na nastavenie pinu GPIO na hodnotu HIGH na dobu 1 sekundy.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18,GPIO.OUT)
print "otvara"
GPIO.output(18,GPIO.HIGH)
time.sleep(1)
GPIO.output(18,GPIO.LOW)
```

5. Implementácia mobilných aplikácií

Na prístup k zaznamenaným teplotám, ich vypísanie a zobrazenie v grafe, ako aj na otváranie brány som vytvoril aplikáciu pre mobilný operačný systém iOS.

5.1. Aplikácia *myHome*

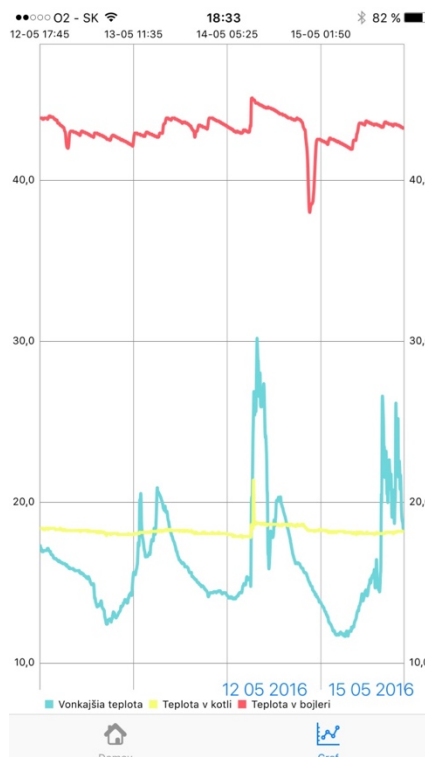
Aplikácia *myHome* slúži na načítavanie zaznamenaných teplôt z databázy uloženej na RPi, a ich uloženie do lokálnej databázy typu Core Data. Posledný uložený záznam zobrazí na prvej obrazovke aplikácie a na druhej vykreslí graf z nameraných teplôt v rozsahu zvolenom používateľom.

Načítanie dát zo servera prebieha nasledovne: Najskôr si vytiahnem posledný pridaný záznam z Core Data databázy a zistím jeho `cash`. Podľa toho času vytvorím parametre `startDate` a `endDate` pre GET dotaz. Parameter `endDate` nastavujem ako aktuálny čas, aby mi server vrátil všetky záznamy ktoré ešte nemám, alebo ak nemám žiadny tak `startDate` a `endDate` zostanú prázdne a dostanem ako odpoveď všetky záznamy.

Po prijatí záznamov vo formáte JSON ich sparsujem, uložím do Core Data databázy a hodnoty z posledného záznamu zobrazím na obrazovke.



Na druhej obrazovke je čiarový graf zobrazujúci namerané hodnoty v rozsahu posledných 24 hodín. Na komunikáciu medzi grafom a databázou slúži object typu NSFetchedResultsController, ktorý pri každej zmene údajov v databáze upozorní objekt grafu na ich zmenu a povie mu aj ktoré dáta sa zmenili, vymazali alebo pridali. Slúži aj na nastavenie predikátu, ktorým určujeme ktoré dáta chceme a ako ich chceme zoradené. Toto umožňuje meniť časový rozsah záznamov. Tento rozhas si dokáže používateľ zmeniť pomocou dvoch tlačidiel na spodnej strane obrazovky.



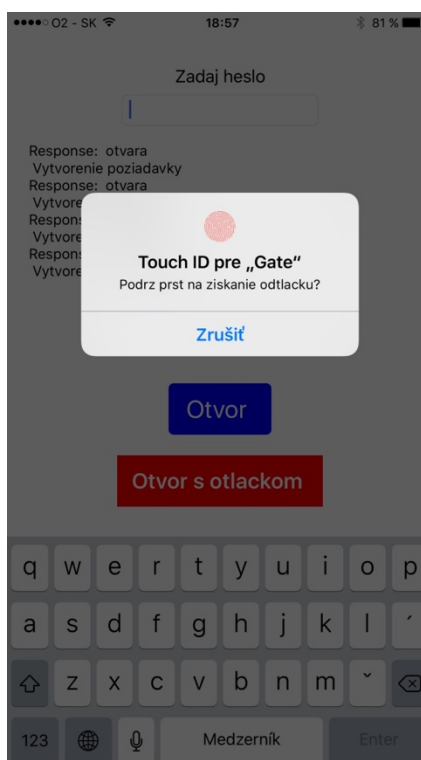
5.2. Aplikácia Gate

Táto aplikácia slúži na komunikáciu so serverom za účelom otvárania brány.

Obsahuje iba jednu obrazovku, s miestom na vpísanie hesla, miestom na zobrazovanie aktuálneho stavu a odpovede zo servera.

Používateľ má dve možnosti ako otvoriť bránu. Prvou je napísanie správneho hesla, toto heslo sa pošle na server a po overení sa začne otvárať brána. Ak heslo bolo nesprávne server vráti odpoveď Nesprávne heslo.

Druhou možnosťou je kliknúť na tlačidlo Otvor s otláčkom, ktoré vyzve používateľa na overenie identity pomocou odtlačku prsta. Ak sa odtlačok zhoduje z otláčkom uloženým v telefóne, aplikácia sama vyplní na pozadí heslo do POST parametra a odošle dotaz na server.



6. Záver

Nápad na tento projekt vznikol ešte pred tým, ako som začal chodiť na predmet Vnorené systémy. A práve tento predmet bol dobrým dôvodom ku zrealizovaniu tohto projektu, ktorého výsledky používam doma dennodenne a zjednodušili mi každodenné bežné činnosti, pri ktorých som musel otvárať bránu a vždy ísť pohľadať diaľkové ovládanie, alebo počas obdobia kúrenia mi tento projekt zjednodušil kontrolu teploty na kotly.

Ďalšie využitie je kontrola vonkajšej teploty a jej zaznamenávanie.

Pôvodný návrh projektu som splnil, no pri implementácii mi napadlo viacero rozšírení a vylepšení, ktoré by som chcel do budúcnosti urobiť. Ako prvá a hlavná úloha bude zvýšiť bezpečnosť komunikácie medzi telefónom a serverom.

Druhé zlepšenie je implementovať push notifikácie, ktoré budú chodiť na telefón v prípade, ak sa teplota na kotly zvýši nad určitú hranicu. Po prvej notifikácii bude server sledovať vývoj teploty a ak teplota nezačne stagnovať alebo klesať, pošle ďalšie notifikácie, aby zaujal používateľovu pozornosť. Týmto sa ďalej zvýši bezpečnosť a komfort pri kúrení kotlom na tuhé palivo.

Tretie zlepšenie bude sledovanie stavu brány pomocou magnetického snímača. Tým budem vždy vedieť, či sa brána naozaj otvorila, alebo naozaj zavrela až do koncovej polohy.

Posledným vylepšením bude doplniť snímače potrebné na presnejšie získavanie údajov o poveternostných podmienkach a zostrojiť meteostanicu aj s predpoveďou počasia na základe vlhkosti vzduchu a barometrického tlaku