STU FIIT
SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
ÚSTAV POČÍTAČOVÉHO INŽINIERSTVA A
APLIKOVANEJ INFORMATIKY

Automatické sledovanie senzorov s grafickým webovým rozhraním

Semestrálny projekt

Meno: Bc. Martin Dekan
Študijný program: Softvérové inžinierstvo
Ročník: 2.
Predmet: Vnorené systémy
Akademický rok: 2016/2017

STU FIIT
SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
ÚSTAV POČÍTAČOVÉHO INŽINIERSTVA A
APLIKOVANEJ INFORMATIKY

ZADANIE SEMESTRÁLNEHO PROJEKTU

Predmet: **VNORENÉ SYSTÉMY**
Riešiteľ: **Bc. Martin Dekan**
Školský rok: **2016/2017**
Názov projektu: **Automatické sledovanie senzorov s grafickým webovým rozhraním**

Zadanie:
Navrhnite a zrealizujte systém automatického sledovania stavu vybraných senzorov. Vnorený systém bude odčítavať hodnoty zo senzorov a údaje odosielať na webový server vo formáte JSON. Senzory budú k vnorenému systému pripojené pomocou bezdrôtovej siete ZigBee. Údaje zobrazte pomocou grafického webového rozhrania vo forme grafov, ktoré budú dostupné prostredníctvom lokálnej siete. V prípade vysokej vlhkosti alebo teploty systém vyšle hlásenie. Systém bude sledovať aj otvorenie okna.

Projekt musí obsahovať:

1. Analýzu problematiky
2. Opis postupu riešenia
3. Výsledky riešenia a ich zhodnotenie
4. Zoznam použitej literatúry
5. Technickú dokumentáciu

Termín odovzdania: Posledné cvičenie z predmetu v letnom semestri 2016/2017

V Bratislave, 15.3.2017 doc. Ing. Tibor Krajčovič, PhD.

1. Analýza

Táto kapitola je venovaná analýze problémovej oblasti, zariadeniam. V nasledovaných podkapitolách sú opísané podmienky nasadenia, použité hardvérové vybavenie.

1.1 Problémová oblasť

Problémovou oblasťou je kuchyňa. V kuchyni je vyššia vlhkosť a teplota než v iných častiach domu. Táto skutočnosť môže spôsobovať zvýšený výskyt plesní. Taktiež zvýšená vlhkosť má za následok znehodnocovanie nábytku. Preto je nutné v kuchyni použiť odvetrávanie buď použitím klimatizácie alebo vetraním. Počas varenia sa relatívna vlhkosť vzduchu v miestnosti postupne zvyšuje a preto si osoby nachádzajúce sa v miestnosti môžu zvyknúť na zvýšenú vlhkosť, ktorá sa neskôr zráža najmä v oblasti okien.

1.2 Riadiaca jednotka

Ako riadiaca jednotka bol zvolený minipočítač Raspberry pi 3. Tento minipočítač svojim výkonom postačuje na menej náročné úlohy ako je napríklad odčítanie údajov z hlavne senzorovej jednotky (gateway).

Raspberry pi 3 bol zvolený najmä kvôli dostupnosti mnohých operačných systémov, veľkej používateľskej základni a množstvu návodov.
Raspberry pi 3 má približnú cenu 40€. V prípade tejto práce môže byť nahradené aj iným zariadením s podporou OS linux.

1.3 Senzory

Na sledovanie vlhkosti, otvorenia okien a teploty bol použitý Xiaomi Mijia Smart Home Kit v približnej hodnote 75€. Tento kit obsahuje:

- vstupnú bránu (gateway)
- senzor vlhkosti a teploty
- senzor otvorenia okna založený na magnetickom princípe
- pohybový senzor
- bezdrôtový prepínač
- zásuvku na sledovanie spotreby – univerzálna zásuvka bez uzemnenia

1.4 Ďalší hardvér

Ďalší potrebný hardvér pre nastavenie zariadení je mobilný telefón alebo tablet s operačným systémom Android v minimálnej verzií 5.0 alebo iOS minimálnej verzií 8.0.

1.5 Softvérové vybavenie

Ako operačný systém pre riadiacu jednotku bol zvolený operačný systém Lubuntu, ktorý má odhľahčené grafické prostredie LXDE. Na riadiacej jednotke bol nainštalovaný program Home Assistant vo virtualizovanom prostredí s doinštalovanými komponentmi

HomeAssistant Xiaomi Hub Component by Rave[1] alebo homeassistant-aqara[2]. Na mobilnom zariadení bola použitá aplikácia Mi Home.

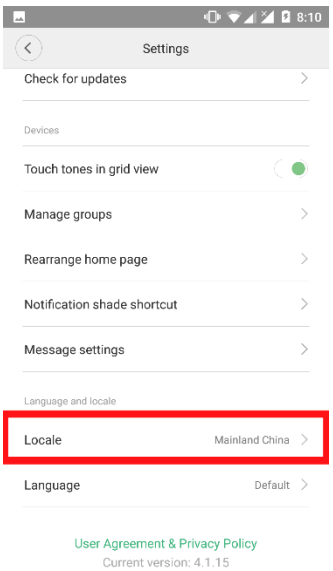
2 Opis riešenia

Táto kapitola opisuje riešenie a technické rozhodnutia vykonané pri tvorbe projektu.

2.1 Spojazdnenie senzorov

Návod na aktualizáciu aplikácie Mi Home na najnovšiu možnú:

1. Inštalácia aplikácie Mi Home
2. V aplikácie Mi Home nastaviť locale na Mainland China
3. Aktualizovať aplikáciu cez možnosť Check for updates



Obrázok 1: Nastavenie locale - je nutné vybrať možnosť Mainland China inak nebude možné pripojiť hlavné zariadenie, ani aktualizovať aplikáciu

Pripojenie hlavnej jednotky na wifi sieť:

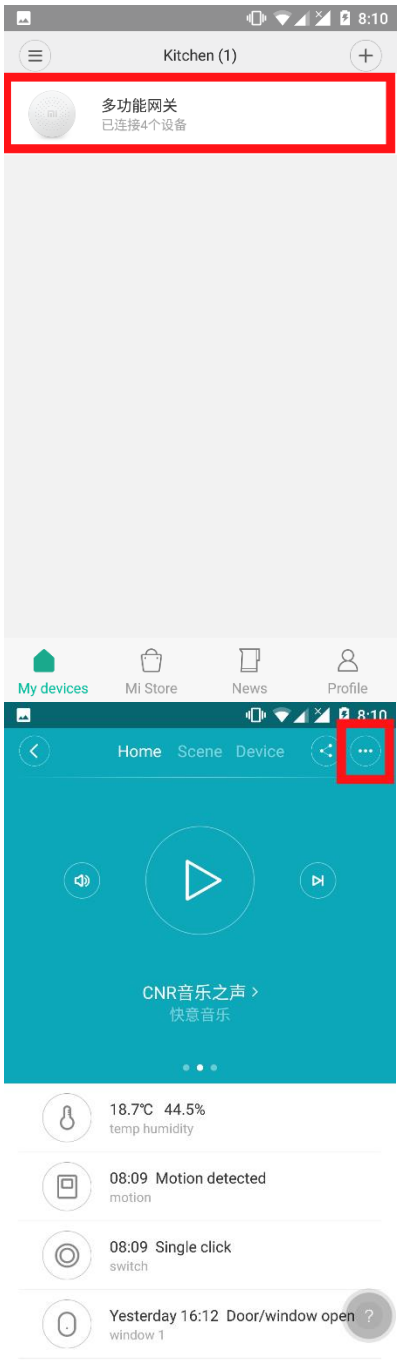
1. Zapojenie hlavnej jednotky do elektrickej siete
2. Registrácia v aplikácií Mi Home
3. Pripojenie na wifi sieť vytvorenú hlavnou jednotkou (gateway)
4. Zadanie údajov wifi siete kam bude hlavná jednotka integrovaná

Párovanie senzorov s hlavnou jednotkou

1. Na hlavnom zariadení stlačiť 3 krát za sebou tlačidlo
2. Po zaznení hlášky stlačiť tlačidlo na senzore
3. Skontrolovať v aplikácie počet pripojených zariadení na hlavnej jednotke
4. A aktualizovať firmvér všetkých senzorov

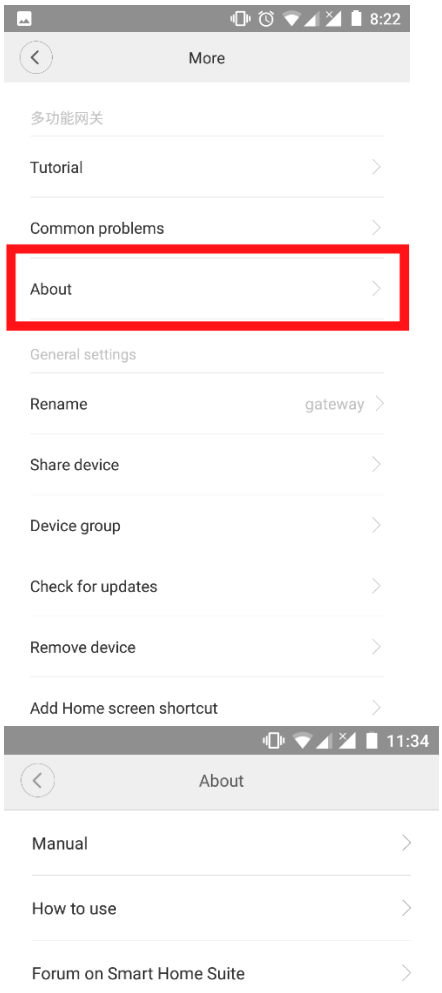
Pozn. párovanie senzorov nie je nutné ak sú zakúpené ako kit.

Návod pre zapnutie developer módu je tento krát obrázkový z dôvodu zlého prekladu aplikácie.

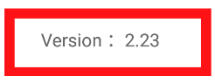
1. 

2.

3.

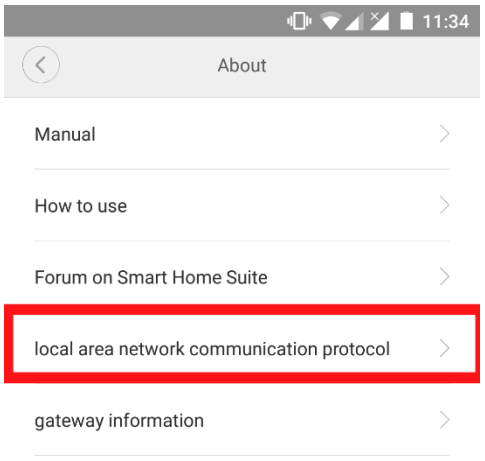


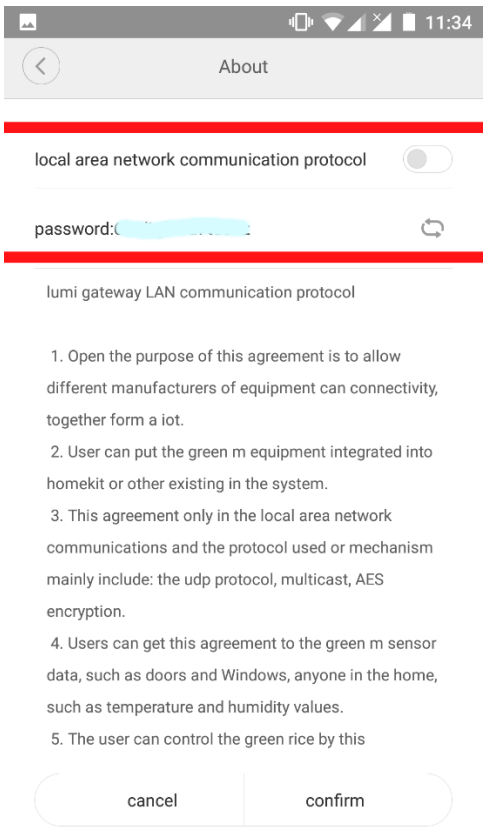
4.



Obrázok 2: je potrebné asi 10 krát stlačiť číslo verzie

5.





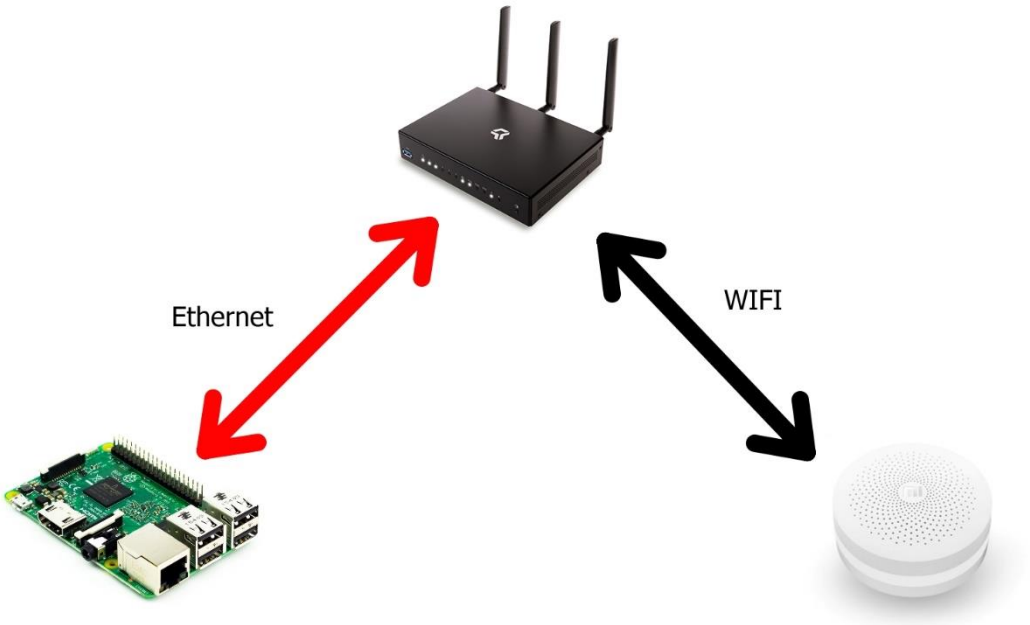
6.

Obrázok 3: povoliť local area network communication protocol a skopírovať heslo

Aplikácia nám umožňuje nastaviť aj automatizáciu. Z dôvodu zlého prekladu sú všetky funkcie sú v čínštine a nastavenie automatizácie cez aplikáciu je bez znalosti čínštiny takmer nemožné.
Aplikácia nám umožňuje ovládať svetlo v hlavnej jednotke, vytváranie grafov a logy akcií (event management). Aplikácia obsahuje také množstvo funkcionality, že by bolo možné projekt vypracovať čisto pomocou nej. Problémom je preklad.

2.2 Použitie aplikácie home assistant

V mojom prípade je na routri pripojené cez wifi pripojená hlavná jednotka a cez ethernet pripojené Raspberry pi 3. Obe zariadenia sú v samostatnej VLAN sieti.



Návod na inštaláciu aplikácie home assistant (platí pre OS lubuntu a ostatné systémy založené na OS debian):

1. Aktualizácia systému
   a. `$ sudo apt-get update`
   b. `$ sudo apt-get upgrade -y`
2. Inštalácia pythonu - sudo apt-get install python3 python3-venv python3-pip
3. Vytvorenie systémového účtu (tento krok nie je nutný, ale zvyšuje bezpečnosť)
   a. $ sudo useradd -rm homeassistant
4. Vytvorenie priečinku aplikácie a zmena vlastníka priečinku
   a. $ cd /srv
   b. $ sudo mkdir homeassistant
   c. $ sudo chown homeassistant:homeassistant homeassistant
5. Vytvorenie a zmena virtuálneho prostredia
   a. $ sudo su -s /bin/bash homeassistant
   b. $ cd /srv/homeassistant
   c. $ python3 -m venv .
   d. $ source bin/activate aktiváciu virtuálneho prostredia sa vykonávajú všetky príkazy pod vyššie vytvoreným používateľom
6. Inštaláciu aplikácie homeassistant
   a. $ pip3 install homeassistant
   b. $ pip3 install pycrypto
7. Spustenie aplikácie
   a. $ hass
8. Aplikácia je dostupná na adrese localhost:8123 alebo na lokálnej ip adrese (získať ju je možné napríklad pomocou $ ifconfig -a)

Inštalácia komponentov pre použitie senzorov xiaomi:

1. Stiahnuť obsah repozitára https://github.com/lazcad/homeassistant
2. Vložiť komponenty do /srv/homeassistant/homeassistant_venv/lib/python3.4/site-packages/homeassistant/components
3. Upraviť súbor Configuration.yaml

```
xiaomi:
 gateways:
   - sid: vyplniť len ak
je v lokálnej sieti viac
hlavných jednotiek
     key: „kľúč získaný
z aplikácie mi home“
```

Po týchto nastaveniach už aplikácia zobrazuje stavy všetkých senzorov.

2.3 Aplikácia na varovanie pri prílišne vysokej vlhkosti

Pre varovanie bola vytvorená aplikácia, ktorá sleduje údaje v aplikácií home assistant a z jeho REST api získava aktuálne údaje vo formáte JSON. Rozhranie je dostupné na adrese http://ipHomeAssistant:8123/api/ kde ipHomeAssistant je adresa nášho riadiaceho zariadenia. Na adrese http://ipHomeAssistant:8123/api/bootstrap získavame všetky stavy a údaje zo systému.

Získavanie údajov z home assistanta prebieha približne každých 60 sekúnd. Ak sa zistí, že je vysoká vlhkosť alebo teplota a zároveň nie je otvorené okno alebo stlačený spínač na senzore tak sa ozve varovný signál.

Ukážky kódu

```
public static JSONObject readJsonFromUrl(String url) throws IOException, JSONException {
    InputStream is = new URL(url).openStream();
    try {
        BufferedReader rd = new BufferedReader(new InputStreamReader(is, Charset.forName("UTF-8")));
        String jsonText = rd.lines().collect(Collectors.joining());
        JSONObject json = new JSONObject(jsonText);
        return json;
    } finally {
        is.close();
    }
}

public static void play(String filename) {
    try {
        JFXPanel panel = new JFXPanel();
        Media hit = new Media(new File(filename).toURI().toString());
        MediaPlayer mediaPlayer = new MediaPlayer(hit);
        mediaPlayer.play();
    } catch (Exception exc) {
        exc.printStackTrace(System.out);
    }
}
```

Obrázok 4: získavanie jsonov a prehrávanie zvuku

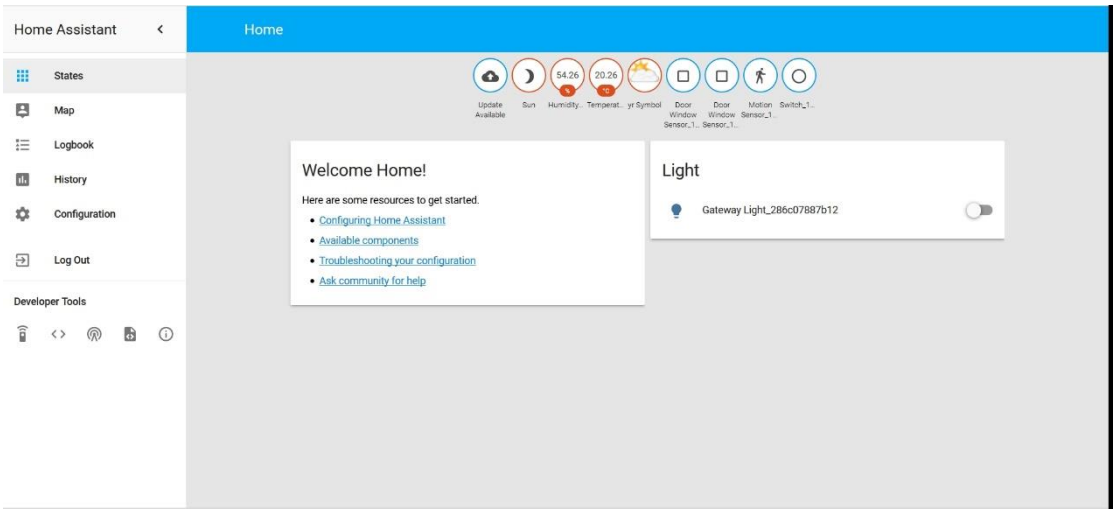
```
}*/
if (shouldUpdate && cooldown == 0) {
    System.out.println("update");
    cooldown = 1440;
}
else {
    cooldown--;
}
Thread.sleep(60000);
shouldUpdate = false;
```

Obrázok 5: Varovanie na update home assistanta alebo komponentov a uspávanie procesu na 60 sekúnd (údaje sa čítajú raz za 60 sekúnd)

```
if ((temperature > 25 || humidity > 65) && !windowOpened && !buttonPressed) {
            play("D:\\TEMP\\foghorn.mp3");
            System.out.println("temp or humidity");
        }
```

Uážka kódu 1: podmienka pre varovaný signál

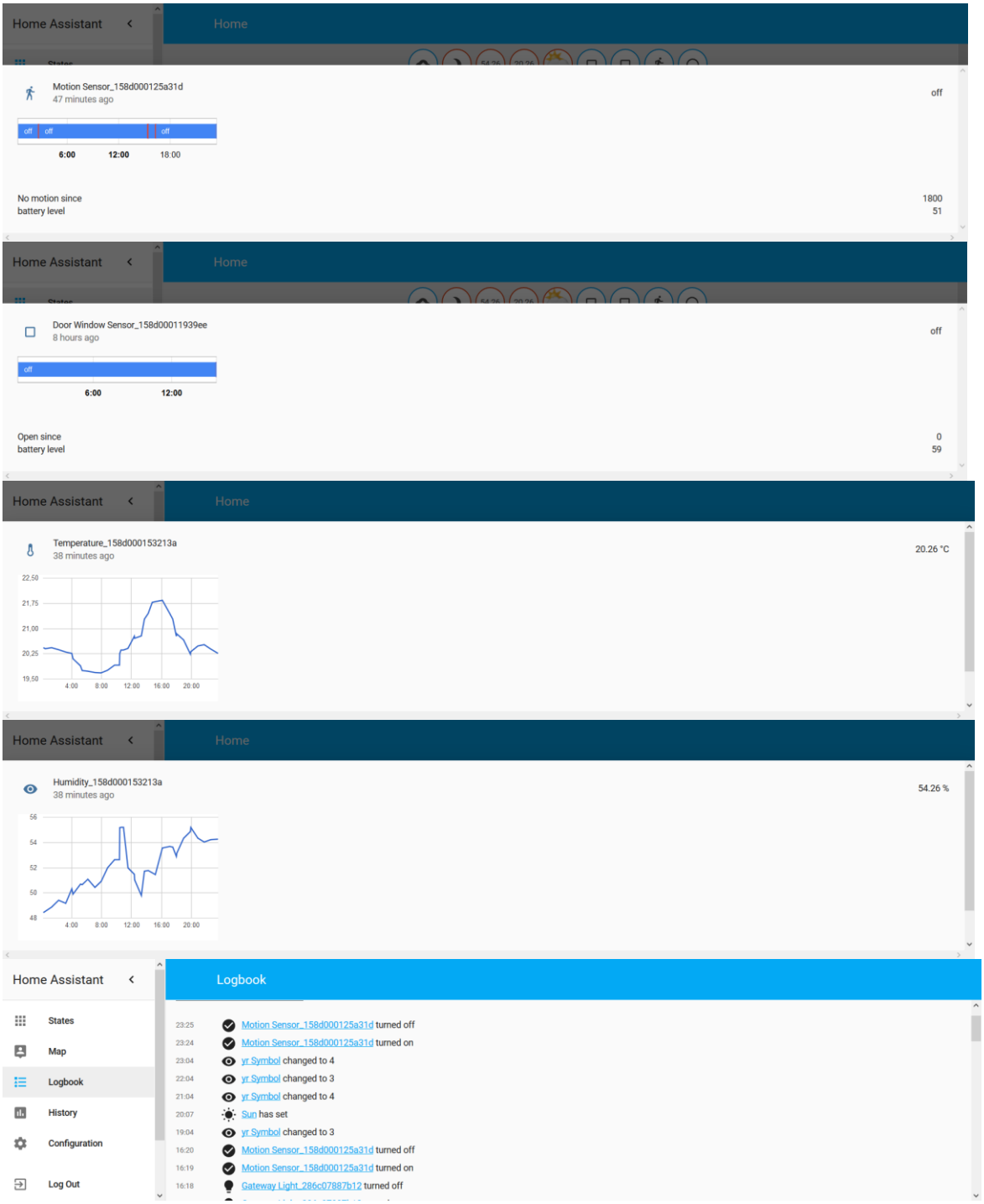
Ukážka z aplikácie home assistant:



Obrázok 6: je možné ovládať svetlo, sledovať hodnoty a stavy senzorov

Ukážka údajov z api:

```
states:
  0:
    attributes:
      Open since:           0
      battery_level:        59
      device_class:         "opening"
      friendly_name:        "Door Window Sensor_158d00011939ee"
    entity_id:              "binary_sensor.door_window_sensor_158d00011939ee"
    last_changed:           "2017-05-03T13:26:22.170278+00:00"
    last_updated:           "2017-05-03T13:26:22.170278+00:00"
    state:                  "off"
  1:
    attributes:
      auto:                 true
      entity_id:
        0:                  "light.gateway_light_286c07887b12"
      friendly_name:        "all lights"
      hidden:               true
      order:                0
    entity_id:              "group.all_lights"
    last_changed:           "2017-05-03T14:18:58.659726+00:00"
    last_updated:           "2017-05-03T14:18:58.659726+00:00"
    state:                  "off"
  2:
    attributes:
      friendly_name:        "Gateway Light_286c07887b12"
      supported_features:   17
    entity_id:              "light.gateway_light_286c07887b12"
    last_changed:           "2017-05-03T14:18:58.634885+00:00"
    last_updated:           "2017-05-03T14:18:58.634885+00:00"
    state:                  "off"
  3:
    attributes:
      battery_level:        57
      friendly_name:        "Temperature_158d000153213a"
      unit_of_measurement:  "°C"
    entity_id:              "sensor.temperature_158d000153213a"
    last_changed:           "2017-05-03T21:34:24.790421+00:00"
    last_updated:           "2017-05-03T21:34:24.790421+00:00"
    state:                  "20.26"
  4:
    attributes:
      friendly_name:        "Update Available"
      release_notes:        "https://home-assistant.io/blog/2017/04/22/ikea-tradfri-spotify/#release-0432---april-27"
    entity_id:              "updater.updater"
    last_changed:           "2017-05-03T09:55:49.735940+00:00"
    last_updated:           "2017-05-03T09:55:49.735940+00:00"
    state:                  "0.43.2"
  5:
    attributes:
      attribution:          "Weather forecast from yr.no, delivered by the Norwegian Meteorological Institute and the NRK."
      entity_picture:       "//api.met.no/weatherapi/weathericon/1.1/?symbol=3;content_type=image/png"
      friendly_name:        "yr Symbol"
    entity_id:              "sensor.yr_symbol"
    last_changed:           "2017-05-03T22:04:23.464283+00:00"
    last_updated:           "2017-05-03T22:04:23.464283+00:00"
    state:                  "3"
  6:
    attributes:
      azimuth:              348.67
      elevation:            -24.93
      friendly_name:        "Sun"
      next_rising:          "2017-05-04T03:27:10+00:00"
      next_setting:         "2017-05-04T18:09:22+00:00"
    entity_id:              "sun.sun"
    last_changed:           "2017-05-03T18:07:59.463750+00:00"
    last_updated:           "2017-05-03T22:05:30.607022+00:00"
    state:                  "below_horizon"
  7:
    attributes:
      Open since:           0
      battery_level:        51
      device_class:         "opening"
      friendly_name:        "Door Window Sensor_158d000153b078"
    entity_id:              "binary_sensor.door_window_sensor_158d000153b078"
    last_changed:           "2017-05-03T13:26:49.973911+00:00"
    last_updated:           "2017-05-03T13:26:49.973911+00:00"
    state:                  "off"
  8:
    attributes:
      battery_level:        48
      friendly_name:        "Switch_158d0000155e226"
    entity_id:              "binary_sensor.switch_158d0000155e226"
    last_changed:           "2017-05-03T13:21:42.981133+00:00"
    last_updated:           "2017-05-03T13:21:42.981133+00:00"
    state:                  "off"
  9:
    attributes:
      No motion since:      "1800"
      battery_level:        51
      device_class:         "motion"
      friendly_name:        "Motion Sensor_158d000125a31d"
    entity_id:              "binary_sensor.motion_sensor_158d000125a31d"
    last_changed:           "2017-05-03T21:25:51.866510+00:00"
    last_updated:           "2017-05-03T21:54:35.823674+00:00"
    state:                  "off"
  10:
    attributes:
      battery_level:        57
      friendly_name:        "Humidity_158d000153213a"
      unit_of_measurement:  "%"
    entity_id:              "sensor.humidity_158d000153213a"
    last_changed:           "2017-05-03T21:34:24.793060+00:00"
    last_updated:           "2017-05-03T21:34:24.793060+00:00"
    state:                  "54.26"
```



Obrázok 7: Na obrázkoch je možné vidieť tvorbu grafov, logy akcií (eventov) a stavy batérií jednotlivých senzorov

Záver

Výsledkom je vytvorený ekosystém, ktorý umožňuje monitorovať v reálnom čase (aplikáciami mi home a homeassistant). Zároveň je vytvorená aplikácia na varovanie pred vysokými teplotami a vlhkosťou a je umožnená tvorba grafov (Mi home aj homeassistant) pre neskoršiu analýzu (všetky údaje sú ukladané v SQLite databáze).

Týmto projektom sa ukázal veľký potenciál lacnejších senzorov z Číny. Cena týchto kitov je do 80€ a umožňujú širokú škálu funkcionalít od bezpečnosti až po domácu automatizáciu.
V porovnaní s drahými kitmi dostupnými na európskom trhu sú tieto senzory horšie len po softvérovej stránke, ktorú je možné kompletne nahradiť softvérovým riešením homeassistant a komponentmi od nezávislých vývojárov.

Použité zdroje

[1] https://github.com/lazcad/homeassistant
[2] https://github.com/fooxy/homeassistant-aqara
[3] https://home-assistant.io/docs/installation/raspberry-pi/
[4] https://github.com/fooxy/homeassistant-aqara/wiki/Enable-dev-mode
[5] bbs.xiaomi.cn/t-13198850
[6] https://github.com/louisZL/lumi-gateway-local-api
[7] https://itunes.apple.com/us/app/mi-home-xiaomi-for-your-smarthome/id957323480?mt=8
[8] https://play.google.com/store/apps/details?id=com.xiaomi.smarthome&hl=sk

Technická dokumentácia

Na vytvorenie je potrebný hardvér a softvér opísaný v predchádzajúcich kapitolách.

Raspberry pi 3 môže byť nahradené aj lacnejším zariadením ako je napríklad Raspberry Pi Zero, keďže hardvérové nároky na chod aplikácií sú veľmi nízke.

Je vyžadovaná operačná pamäť o veľkosti minimálne 1GB a aspoň 1Ghz procesor.
Všetky inštalácie je možné vykonať na riadiacej jednotke, alebo po pripojení cez ssh.

Aplikácia je dostupná na adrese: ipRaspberry:8123
REST api je dostupné na adrese: ipRaspberry:8123/api/
Všetky údaje z aplikácie je možné získať pomocou: ipRaspberry:8123/api/bootstrap

Zdrojové kódy:

```
6   package crawler;
7
8   import java.io.BufferedReader;
9   import java.io.File;
10  import java.io.IOException;
11  import java.io.InputStream;
12  import java.io.InputStreamReader;
13  import java.net.URL;
14  import java.nio.charset.Charset;
15  import java.util.stream.Collectors;
16  import javafx.embed.swing.JFXPanel;
17  import javafx.scene.media.Media;
18  import javafx.scene.media.MediaPlayer;
19  import org.json.*;
20
21  /**
22   *
23   * @author Mato
24   */
25  public class SiteGetter {
26
27      boolean notStopped = true;
28      boolean windowOpened = false;
29      boolean buttonPressed = false;
30      boolean motionDetected = false;
31      boolean gatewayLight = false;
32      boolean allLights = false;
33      boolean shouldUpdate = false;
34      float humidity = 0;
35      float temperature = 15;
36      int cooldown = 0;
37
38      void getParams() throws IOException, InterruptedException {
39          while (notStopped) {
40              JSONObject json = readJsonFromUrl("http://192.168.100.36:8123/api/bootstrap");
41              JSONArray states = json.getJSONArray("states");
42              for (int i = 0; i < 11; i++) {
43                  JSONObject state = (JSONObject) states.get(i);
44                  switch (state.get("entity_id").toString()) {
45                      case "binary_sensor.switch_158d000155e226":
                            if (!"off".equals(state.get("state"))) {
47                              buttonPressed = true;
48                          } else {
49                              buttonPressed = false;
```

```
50                  }
51                  break;
52              case "binary_sensor.motion_sensor_158d000125a31d":
                    if (!"off".equals(state.get("state"))) {
54                      motionDetected = true;
55                  } else {
56                      motionDetected = false;
57                  }
58                  break;
59              case "sensor.humidity_158d000153213a":
60                  humidity = Float.parseFloat(state.get("state").toString());
61                  break;
62              case "sensor.temperature_158d000153213a":
63                  temperature = Float.parseFloat(state.get("state").toString());
64                  break;
65              case "binary_sensor.door_window_sensor_158d000153b078":
                    if (!"off".equals(state.get("state"))) {
67                      windowOpened = true;
68                  } else {
69                      windowOpened = false;
70                  }
71                  break;
72              case "updater.updater":
73                  shouldUpdate = true;
74                  break;
75              case "light.gateway_light_286c07887b12":
                    if (!"off".equals(state.get("state"))) {
77                      gatewayLight = true;
78                  } else {
79                      gatewayLight = false;
80                  }
81                  break;
82              case "group.all_lights":
                    if (!"off".equals(state.get("state"))) {
84                      allLights = true;
85                  } else {
86                      allLights = false;
87                  }
88                  break;
89              case "binary_sensor.door_window_sensor_158d00011939ee":
                    if (!"off".equals(state.get("state"))) {
91                      windowOpened = true;
92                  } else {
```

```
                        windowOpened = false;
                    }
                    break;
                default:
                    break;
            }
        }
        if ((temperature > 25 || humidity > 65) && !windowOpened && !buttonPressed) {
            play("D:\\TEMP\\foghorn.mp3");
            System.out.println("temp or humidity");
        }
        if (shouldUpdate && cooldown == 0) {
            System.out.println("update");
            cooldown = 1440;
        }
        else {
            cooldown--;
        }
        Thread.sleep(60000);
        shouldUpdate = false;
    }
}

public static JSONObject readJsonFromUrl(String url) throws IOException, JSONException {
    InputStream is = new URL(url).openStream();
    try {
        BufferedReader rd = new BufferedReader(new InputStreamReader(is, Charset.forName("UTF-8")));
        String jsonText = rd.lines().collect(Collectors.joining());
        JSONObject json = new JSONObject(jsonText);
        return json;
    } finally {
        is.close();
    }
}

public static void play(String filename) {
    try {
        JFXPanel panel = new JFXPanel();
        Media hit = new Media(new File(filename).toURI().toString());
        MediaPlayer mediaPlayer = new MediaPlayer(hit);
        mediaPlayer.play();
    } catch (Exception exc) {
        exc.printStackTrace(System.out);
```

```
package crawler;

import java.io.IOException;

public class Crawler {

    public static void main(String[] args) throws IOException, InterruptedException {
        SiteGetter getter = new SiteGetter();
        getter.getParams();
    }

}
```