```
1 # Import libraries
 2 import csv
 3 import time
 4 import socket
 5 import RPi.GPIO as GPIO
 6 import requests
 7 import gspread
 8
 9 # Import python files
10 from switch import *
11 from temperature import *
12 from light import *
13
14 # Functions
15 def getip():
16
      try:
17
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
18
        s.connect(("8.8.8.8", 80))
19
        return s.getsockname()[0]
20
      except:
21
        return 0
22
23 def TelMess(text):
24
      try:
25
        base_url = 'https://api.telegram.org/bot5189477795:
   AAEVYv V0PWOicis7RtdYsNIQZFNOMHxIJk/'\
26
               'sendMessage?chat_id=-682759305&text={text}'.format(
27
          text=text)
28
        requests.get(base_url)
29
      except:
30
        return 0
31
32
33 # Mysql Parameters
34 ip = 'XXX.XXX.XXX.XXX'
35 port = '3306'
36 database = 'Bakalarka'
37
38 # Setup relay
39 led y = 6
40 led r = 13
41 heat foil bed = 26
42 heat_foil_liv = 19
43
44 # Setup switches
45 \text{ button}_1 = 27
46 button 2 = 23
```

```
47 button 3 = 17
48 button 4 = 18
49
50 # Setup DHT Sensors
51 temp_cycle_limit = 10
52
53 # Variables
54 \text{ cycle} = 0
55 \text{ run} = 1
56 cycle but 1 = 0
57 cycle but 2 = 0
58 \text{ cycle\_but3} = 0
59 cycle but4 = 0
60 relay = [False, False, False, False]
61 online = [2,2,2,2,2,2,2,2,2,2,2,2,2]
62
63 # Temp setting
64 \text{ set temp bed} = 20
65 set temp liv = 23
66 mer_dokument = 'regulace_v2.csv'
67 delta temp = 0.5
68 time zone = 3600 # +1 hour to time (Prague)
69 mess delay = 30 # 30 sec mess delay
70
71 # Switches
72 bedroom switch = Switch('switch', 'QYKPMdbKNydPTW5k', ip, database,
    port, "'loznice", 0, button_1)
73 living_room_switch_1 = Switch('switch', 'QYKPMdbKNydPTW5k', ip,
    database, port, "'obyvak1", 0, button 2)
74 living room switch 2 = Switch('switch', 'QYKPMdbKNydPTW5k', ip,
    database, port, "'obyvak2", 0, button 3)
75
76 #Lights
77 bedroom light = Light('light', '3yRdaB3r6by5', ip, database, port, "'obyvak"
    , 0, led r)
78 living room light = Light('light', '3yRdaB3r6by5', ip, database, port, "'loznice
    "", 0, led_y)
79
80 GPIO.setmode(GPIO.BCM)
81
82 # Setup switch to INPUT
83 GPIO.setup(button 1, GPIO.IN, pull up down=GPIO.PUD UP) # Button 1
    for Red led
84 GPIO.setup(button 2, GPIO.IN, pull up down=GPIO.PUD UP) # Button for
    Yellow led
85 GPIO.setup(button 3, GPIO.IN, pull up down=GPIO.PUD UP) # End loop
    button
```

```
86 GPIO.setup(button 4, GPIO.IN, pull up down=GPIO.PUD UP) # Button 1
     for Red led
 87
 88 # Setup Relay to 0 by default value
 89 GPIO.setup(heat foil bed, GPIO.IN) # Heating foil 1
 90 GPIO.setup(heat foil liv, GPIO.IN) # Heating foil 2
 91 GPIO.setup(led r, GPIO.IN) # Led yellow
 92 GPIO.setup(led y, GPIO.IN) # Led red
 93
 94 DhtSensor1 = DhtSensor('TempSensor', 'Ki#7](J&haH>QYx`', ip, database
     , port, "Obyvak DHT", 0)
 95 DhtSensor2 = DhtSensor('TempSensor', 'Kj#7](J&haH>QYx`', ip, database
     , port, "Loznice_DHT", 0)
 96
 97 # Setup Google sheet API
 98 try:
 99
       sa = gspread.service account(filename="../../odevzdání/
    service account google.json")
       sh = sa.open("Nastaveni")
100
       wks_sett = sh.worksheet("Default setting")
101
102
103
       set temp bed = float(wks sett.acell('B5').value)
104
       set temp liv = float(wks sett.acell('B4').value)
105
       delta temp = float(wks sett.acell('B3').value)
106
       mess delay = float(wks sett.acell('B6').value)
107 except:
108
       print("Google sheets not working")
109 # Connect user into database
110 trv:
111
       DhtSensor1.conn to database()
112
       DhtSensor2.conn to database()
113
       bedroom switch.conn to database()
114
       living room switch 1.conn to database()
115
       living room switch 2.conn to database()
116
       bedroom light.conn to database()
117
       living room light.conn to database()
118
       online[0] = 1
119 except:
120
       online[0] = 0
121
122 seconds = time.time()
123 local time = time.ctime(seconds + time zone) # 3600 timezone to Prague
124 next measurement = time.ctime(seconds + time zone + mess delay) # 30
     sec delay for measurement
125 print("Local time:", local time)
126
127 # print("Ip address: ",getip())
```

```
128 TelMess("Aplikace Smart-Home byla spuštěna")
129 TelMess("Ip adresa RPI je:" + str(getip()))
130
131 # Measurement to csv file herader
132 header = ['TIME', 'TEMP OB', 'TEMP LOZ', 'REL OB', 'REL LOZ']
133 with open(mer_dokument, 'w', encoding='UTF8', newline=") as f:
134
       writer = csv.writer(f)
135
       writer.writerow(header)
136
137 # Lists
138
139 Light list = [living room light,bedroom light]
140 Switch list = [bedroom switch, living room switch 1, living room switch 2]
141
142 # Super loop
143 while run == 1:
144
       cycle = cycle + 1
145
146
       # Button repair
147
       if cycle but1 > 50000:
148
         cycle but1 = 50000
       if cycle but2 > 50000:
149
150
          cycle but2 = 50000
       if cycle but3 > 50000:
151
152
          cycle but3 = 50000
153
       if cycle but4 > 50000:
154
          cycle but4 = 50000
155
156
       # Control switches
       if GPIO.input(bedroom_switch.gpio_port) == 0:
157
158
          cycle but1 = cycle but1 + 1
159
       else:
160
          cycle but 1 = 0
161
162
       if GPIO.input(living room switch 1.gpio port) == 0:
163
          cycle but2 = cycle but2 + 1
164
       else:
165
          cycle but 2 = 0
166
167
       if GPIO.input(living room switch 2.gpio port) == 0:
168
          cycle_but3 = cycle_but3 + 1
169
       else:
170
          cycle_but3 = 0
171
172
       if GPIO.input(button 4) == 0:
173
          cycle but4 = cycle but4 + 1
174
       else:
```

```
175
          cycle but4 = 0
176
177
       if cycle but1 > 20000:
178
          if bedroom switch.actual state != 1:
179
            bedroom switch.actual state = 1
180
            online[1] = bedroom switch.update database state()
181
       else:
182
          if bedroom switch.actual state != 0:
            bedroom switch.actual state = 0
183
184
            online[2] = bedroom switch.update database state()
185
186
       if cycle but2 > 20000:
187
          if living room switch 1.actual state != 1:
            living room switch 1.actual state = 1
188
189
            online[3] = living room switch 1.update database state()
190
       else:
191
          if living room switch 1.actual state != 0:
192
            living room switch 1.actual state = 0
            online[4] = living room switch 1.update database state()
193
194
195
       if cycle but3 > 20000:
          if living room switch 2.actual state != 1:
196
197
            living room switch 2.actual state = 1
198
            online[5] = living room switch 2.update database state()
199
       else:
200
          if living room switch 2.actual state != 0:
201
            living room switch 2.actual state = 0
202
            online[6] = living_room_switch_2.update_database_state()
203
204
       if cycle but4 > 20000: # Program END
205
          run = 0
206
       # Light set on/off
       if bedroom switch.actual state == 1:
207
208
          if not relay[0]:
209
            relay[0] = True
210
            bedroom light.actual state = 1
211
            online[7] = bedroom light.update datab state()
212
       else:
213
          if relay[0]:
214
            relay[0] = False
215
            bedroom light.actual state = 0
216
            online[8] = bedroom light.update datab state()
217
218
       if living room switch 1.actual state != living room switch 2.actual state
219
          if not relay[1]:
220
            living room light.actual state = 1
```

```
221
            online[9] = living room light.update datab state()
222
            relay[1] = True
223
       else:
224
          if relay[1]:
225
            living room light.actual state = 0
226
            online[10] = living room light.update datab state()
227
            relay[1] = False
228
229
       if local time >= next measurement:
230
231
          # new time for measurement
232
          next measurement = time.ctime(seconds + time zone + mess delay)
233
          print("Another measurement will be in: ", str(next measurement))
234
235
          # Measure temperature
236
          DhtSensor1.mess temperature()
237
          DhtSensor2.mess temperature()
238
          # Living room switch
239
          if DhtSensor2.temperature > set temp liv + delta temp:
240
            GPIO.setup(heat foil liv, GPIO.IN)
241
            relav[3] = False
242
          elif DhtSensor2.temperature < set temp liv - delta temp:
243
            GPIO.setup(heat foil liv, GPIO.OUT)
244
            relay[3] = True
245
          # Bedroom switch
246
          if DhtSensor1.temperature > set temp bed + delta temp:
247
            GPIO.setup(heat foil bed, GPIO.IN)
248
            relay[2] = False
249
          elif DhtSensor1.temperature < set temp bed - delta temp:
250
            GPIO.setup(heat foil bed, GPIO.OUT)
            relay[2] = True
251
252
253
          TelMess("Teplota v ložnici: " + str(DhtSensor1.temperature) + "°C \n
     Teplota v obýváku: "+
254
              str(DhtSensor2.temperature) + "°C" + "\nNastavení teplot pro
     relé: \nLožnice: " +
255
              str(set temp bed) + "°C \nObývák: " + str(set temp liv) + "°C\n
     Delta: " + str(delta temp) +
              "°C \nStav osvětlení: \nLožnice: " + str(relay[0]) + "\nObývák: "
256
     + str(relay[1]) +
257
               "\nStav topných folií: \nLožnice: " + str(relay[2]) + "\nObývák: "
     + str(relay[3]))
258
259
          online[11] = DhtSensor1.add temp database()
260
          online[12] = DhtSensor2.add temp database()
261
262
          print(online)
```

```
263
264
          data = [local_time, DhtSensor2.temperature, DhtSensor1.temperature,
     relay[2], relay[3]]
265
          with open(mer_dokument, 'a', encoding='UTF8', newline=") as f:
266
            writer = csv.writer(f)
267
            writer.writerow(data)
268
269
          # Check database connection
          if 0 in online:
270
271
            print("Databáze není připojena")
272
            try:
273
               DhtSensor1.conn to database()
274
               DhtSensor2.conn to_database()
275
               bedroom switch.conn to database()
276
               living room switch 1.conn to database()
277
               living room switch 2.conn to database()
               bedroom light.conn to database()
278
279
               living room light.conn to database()
280
               print("Databáze opět připojena")
281
            except:
282
               print("Nepodařilo se databázi připojit")
283
284
          else:
285
            print("Databáze je připojena")
286
287
          # Update data from and to google sheets
288
          try:
289
            # update from
290
            set temp bed = float(wks sett.acell('B5').value)
            set temp liv = float(wks sett.acell('B4').value)
291
            delta temp = float(wks sett.acell('B3').value)
292
293
            mess delay = float(wks sett.acell('B6').value)
294
295
            # update to
296
            wks sett.update('F2', '{teplota} °C'.format(teplota = DhtSensor1.
297
     temperature))
298
            wks sett.update('F3', '{teplota} °C'.format(teplota=DhtSensor2.
     temperature))
299
            wks_sett.update('F4', '{stav}'.format(stav=relay[0]))
300
            wks sett.update('F5', '{stav}'.format(stav=relay[1]))
301
            wks sett.update('H2', '{stav}'.format(stav=relav[3]))
            wks_sett.update('H3', '{stav}'.format(stav=relay[2]))
302
303
            wks sett.update('H5', '{time}'.format(time=local time))
304
            wks sett.update('H6', '{IP}'.format(IP=getip()))
305
          except:
306
            print("Google sheets not working")
```

```
File - C:\Users\fuxa9\Desktop\Smart-Home\odevzdani\programs\main.py
307
308
        if cycle == 50000: # End of Cycle
309
          seconds = time.time()
310
          local_time = time.ctime(seconds + time_zone)
311
          cycle = 0
312
313 #End of program
314 GPIO.cleanup()
315 TelMess("Aplikace Smart-Home byla ukončena")
316 try:
317
        DhtSensor1.close_database()
        DhtSensor2.close_database()
318
319
        bedroom switch.close database()
320
        living_room_switch_1.close_database()
321
        living room switch 2.close database()
322
        living room light.close database()
323
        bedroom_light.close_database()
324 except:
        print("Nebylo správně ukončeno")
325
```

```
1 from datab con import *
 2 import RPi.GPIO as GPIO
 3
 4 class Light(database):
      def __init__(self, user, password, host, name_of_database, port, location,
    actual_state, gpio_port):
 6
         super().__init__(user, password, host, name_of_database, port)
 7
         self.location = location
 8
         self.actual_state = actual_state
 9
         self.gpio port = gpio port
10
11
      def update_datab_state(self):
12
         if self.actual_state == 1:
13
           GPIO.setup(self.gpio_port, GPIO.OUT)
14
         else:
15
           GPIO.setup(self.gpio_port, GPIO.IN)
16
17
         sql = "UPDATE `Bakalarka`.`lights` SET state = {state} WHERE
    location={light_name}".format(state=self.actual_state,
18
                                                          light_name=self.
    location)
19
        try:
20
           self.insert to database(sql)
21
           return 1
22
         except:
23
           return 0
```

```
File - C:\Users\fuxa9\Desktop\Smart-Home\odevzdani\programs\switch.py
  1 from datab con import *
 2
 3
 4 class Switch(database):
       def __init__(self, user, password, host, name_of_database, port, location,
    actual_state, gpio_port):
 6
          super().__init__(user, password, host, name_of_database, port)
 7
          self.location = location
 8
          self.actual_state = actual_state
 9
          self.gpio port = gpio port
10
11
       def get_actual_state(self):
12
          sql = "SELECT * FROM akalarka.switches WHERE location={
    name_switch}".format(name_switch=self.location)
13
          online state switch = self.select from database(sql)
14
          self.actual_state = int(online_state_switch[0][2])
15
          print(self.actual_state)
16
17
       def update_database_state(self):
18
          sql = "UPDATE `Bakalarka`.`switches` SET state = {state} WHERE
    name_switch={switch_name}".format(state=self.actual_state,
19
                                                            switch name=self.
    location)
20
          try:
21
            self.insert_to_database(sql)
22
            return 1
23
          except:
24
            return 0
```

```
import mysql.connector
 2
 3 class database:
 4
 5
      def init (self, user, password, host, name of database, port):
 6
         self.user = user
 7
         self.password = password
 8
         self.host = host
 9
         self.name of database = name of database
10
         self.port = port
11
12
      def conn_to_database(self):
13
         print("Attempting to connect a user: " + self.user)
14
         try:
15
           global my database
           my database = mysql.connector.connect(
16
17
              host=str(self.host),
18
              user=str(self.user),
19
              password=str(self.password),
20
             database = str(self.name_of_database),
21
              port = str(self.port)
22
23
           print("Connect to database was successful")
24
         except mysql.connector.Error as e:
25
           print(e)
26
27
      def insert to database(self,sql):
28
         cursor = my_database.cursor()
29
         cursor.execute(sql)
30
         my database.commit()
31
32
      def select from database(self,sql):
33
         try:
34
           cursor = my database.cursor()
35
           cursor.execute(sql)
36
           return cursor.fetchall()
37
         except:
38
           return 10
39
40
      def close database(self):
41
        try:
42
           user = self.user
43
           my_database.close()
44
           print("Connections is closed for " + user)
45
           return 1
46
         except:
47
           print("Connections isn't closed for " + user)
```

```
1 from datab con import *
 2
 3 import board
 4 import adafruit_dht
 5
 6 dhtDevice_obyvak = adafruit_dht.DHT22(board.D20, use_pulseio=False)
 7 dhtDevice loznice = adafruit dht.DHT22(board.D21, use pulseio=False)
 8
 9
10 class DhtSensor(database):
11
12
      def __init__(self, user, password, host, name_of_database, port, sensorID
    , temperature):
13
        super(). init (user, password, host, name of database, port)
14
        self.sensorID = sensorID
15
         self.temperature = temperature
16
17
      def mess temperature(self):
18
        temp = 0
19
        divide = 0
20
        for i in range(10):
21
           if self.sensorID == "Obyvak_DHT":
22
             try:
23
                temp += round(dhtDevice obyvak.temperature, 3)
24
                divide += 1
25
             except:
26
                temp = 0
27
                divide = 1
28
29
           if self.sensorID == "Loznice_DHT":
30
31
                temp += round(dhtDevice loznice.temperature, 3)
32
                divide += 1
33
             except:
34
                temp = 0
35
                divide = 1
36
        if divide != 0:
37
           self.temperature = round(temp / divide, 3)
38
        else:
39
           self.temperature = round(temp, 3)
40
41
      def add temp database(self):
         sql = "INSERT INTO `Bakalarka`.`temp_in` ( `sensor_id`, `temp`)
42
    VALUES ('{}',{})".format(self.sensorID,
43
                                                               self.temperature
    )
44
        try:
```

