

Home Automation Technologies Using Arduino

**Report Prepared in Complete Fulfillment of the
Project Course: CS F376 Design Project**

By

REUBEN MENEZES

2015A7PS0291U

Under The Supervision of

Dr SUJALA SHETTY

Assistant Professor, Computer Science

At



BITS Pilani, Dubai Campus

Dubai International Academic City, Dubai

U.A.E

Second Semester, 2017-18

Home Automation Technologies Using Arduino

**Report Prepared in Complete Fulfillment of the
Project Course: CS F376 Design Project**

By

REUBEN MENEZES

2015A7PS0291U

Under The Supervision of

Dr SUJALA SHETTY

Assistant Professor, Computer Science

At



BITS Pilani, Dubai Campus

Dubai International Academic City, Dubai

U.A.E

Second Semester, 2017-18

ABSTRACT

Course name: Design Project

Course number: CS F376

Duration: 23.01.2018 to 24.05.2018

Date of start: 23.01.2017

Date of submission: 22.05.17

Title of Report: Home Automation Technologies using Arduino

Name / ID Number: REUBEN MENEZES / 2015A7PS0291U

Discipline of Student: B.E (Hons) Computer Science Engineering

Name of Project Supervisor: Dr. Sujala Shetty

Keywords: Home Automation, IOT, Arduino, Sensors, Pubnub

Project Area: IOT, Home Automation,

Abstract: Aim is to build a prototype Home automation System, which would give you basic details such as the temperature of your room. It would also have a motion sensor so it could notify you if there are unexpected movement in your house. You could also control the lights in your room by clicking a button on your laptop. It also has a gas sensor which would notify you if there is a gas leak.

Signature of the Student

Date:

Signature of Supervisor

Date:

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to Prof. Dr. R.N. Saha, Director - BITS Pilani, Dubai Campus who has given me an opportunity to understand and apply engineering concepts in real-life problems through the Project.

I am very grateful to Prof. Dr. Sujala Shetty, Assistant Professor Computer Science Department, BITS PILANI Dubai Campus, my mentor and Project in-charge, for assisting me enthusiastically in the entire project. It is only with her guidance that the objectives of the project have been accomplished.

I am grateful to examiners Dr Santhosh Kumar and CS Department Faculty Members for their valuable suggestions.

Above all, I thank the Lord for giving me the strength to carry out this work to the best of my abilities.

Name: Reuben Menezes

ID: 2015A7PS0291U

Certificate

To Whomsoever it May Concern

This is to certify that the Semester Project Report titled , Home Automation using Arduino ,submitted by Reuben Menezes, ID: 2015A7PS0291U ,in complete fulfillment of the requirement of CSF376 ,Design Project embodies the work done by her under my supervision.

Signature of the Supervisor

Name: Dr Sujala Shetty

Designation: Assistant Professor

Date : 15/05/18

Table of contents

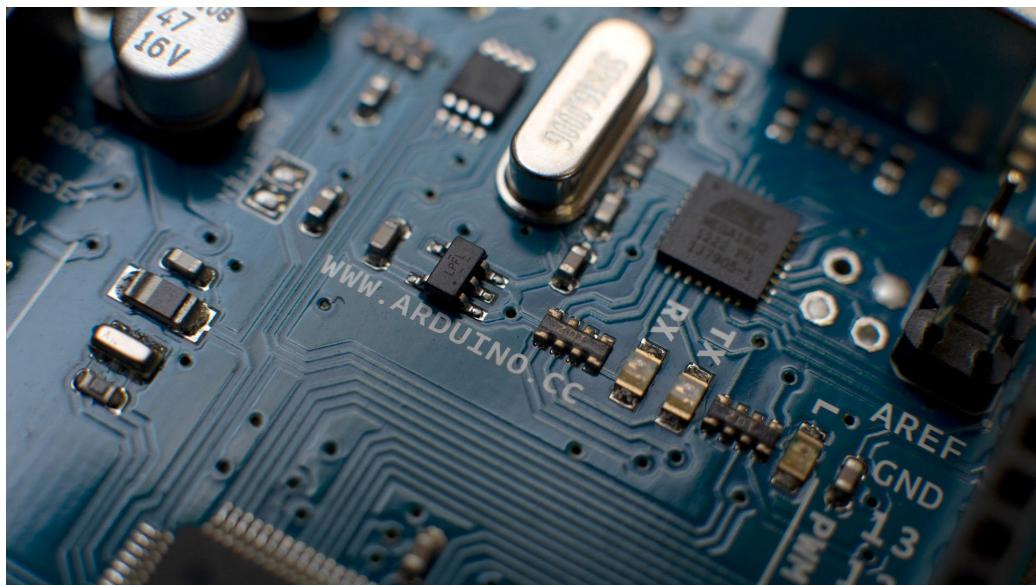
CHAPTER	TITLE	PAGE No.
1.	Acknowledgement	4
2.	Certificate from the Supervisor	5
3.	Abstract	5
4.	Table of Contents	6
5.	Objectives	7
6.	Arduino	8
7.	Gas Sensor	9
8.	PIR Sensor	11
9.	Temperature Sensor	13
10.	Pubnub	15
11.	Front –End : Dashboard	16
12.	User Interface : Dashboard	21
13.	Hands On Control Of Lights from the receiving Computer	23
14.	User Interface : Control Lights	24
15.	Back-End : Control Lights	25
16	Annex	27

OBJECTIVES

My Objective is to build a Home Automation Technology that can be controlled via your laptop at a tap of a mouse button. I'll be using IOT to display constant data which is compiled by the arduino and send it to the customized webpage that I have created. It will also be capable of physically turning on/off the lights at a tap of a mouse button. My project will use the following components :

- Arduino
- Gas Sensor
- PIR Sensor
- Breadboard
- LDR's
- Temperature Sensor
- LED's

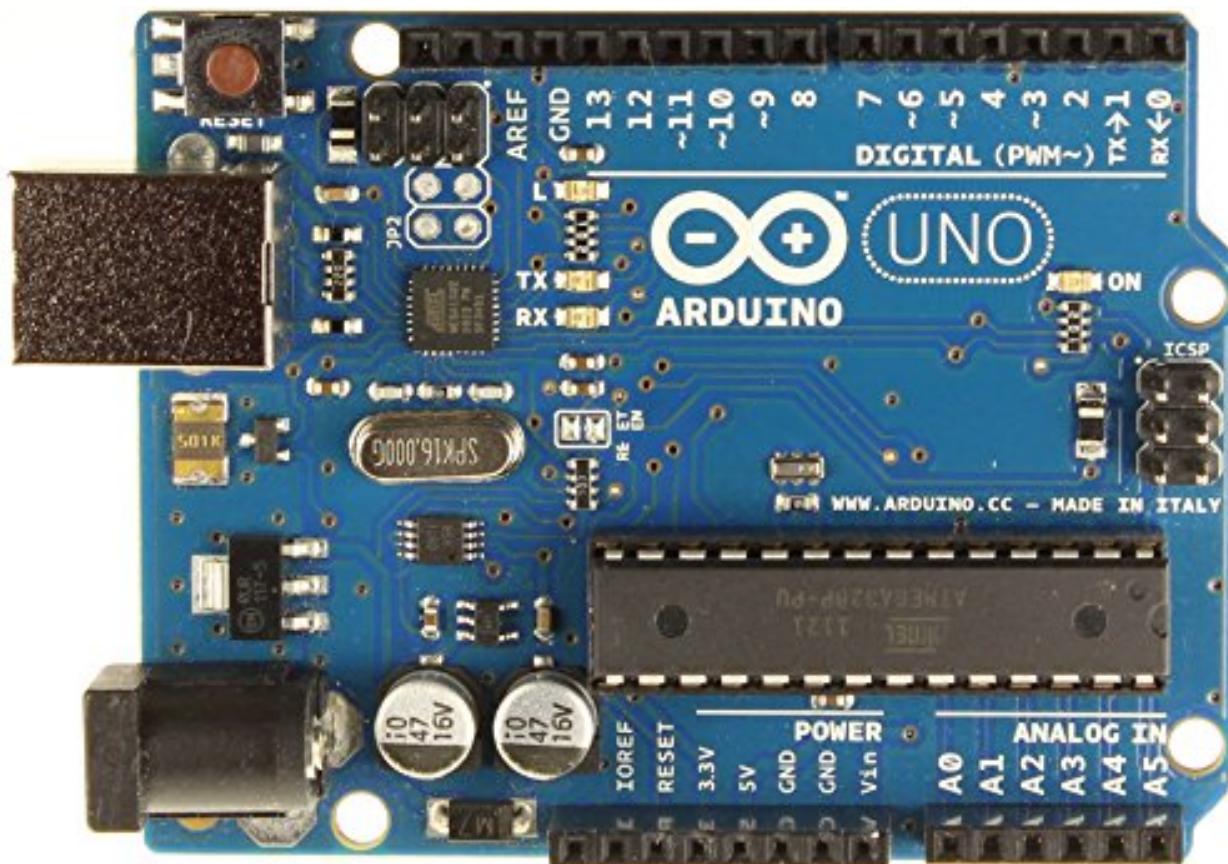
For the streaming of real time data I will be using API's from Pubnub which is global datastream network company based in San Franciso.



ARDUINO

Arduino is a Microcontroller. It's an electronics platform that's open source. It's cross platform and can be run on Linux ,Windows and Mac platforms .All it takes is to buy the hardware and download the software .To send instructions to the arduino we use Arduino Programming Language .The Arduino Programming Language is basically just C and C++ functions.

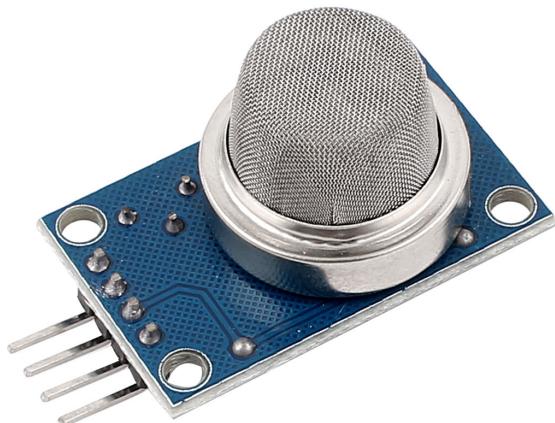
I choose using the arduino over raspberry pie because of the fact that I needed to use both digital and analog pins .There are different kind of arduino boards based on your requirements .For my project I choose to use the **Arduino UNO R3**.



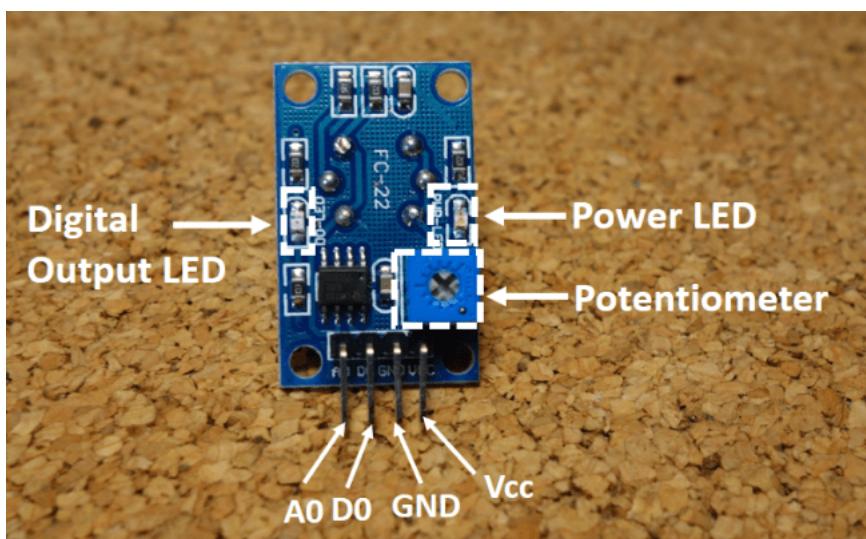
GAS SENSOR

The Gas Sensor (MQ2) module which I have used is useful in detecting gas leaks, be it Liquefied Petroleum Gas ,Carbon Monoxide, Propane or Smoke.

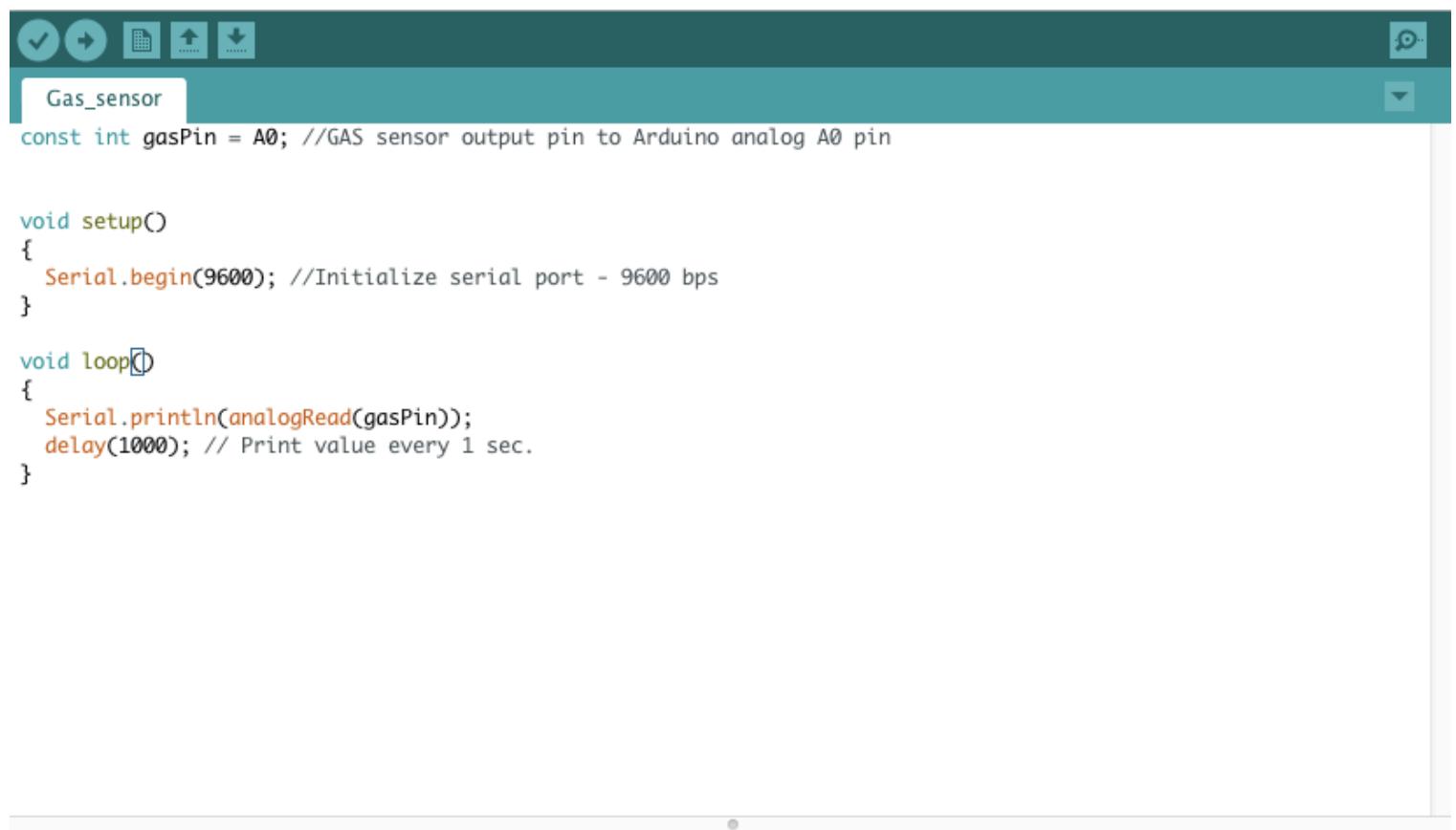
Due to it's high sensitivity and response time appropriate measures can be taken as soon as possible. The sensor integrates well with an Arduino but fails to integrate with a raspberry pi by the conventional methods.



The Gas Sensor works by increasing the output voltage when the concentration of gas increases.



The Following Code works for only interfacing gas sensor with the arduino after the gas sensor is connected to the analog input A0.



The screenshot shows the Arduino IDE interface. The title bar says "Gas_sensor". The code area contains the following Arduino sketch:

```
const int gasPin = A0; //GAS sensor output pin to Arduino analog A0 pin

void setup()
{
    Serial.begin(9600); //Initialize serial port - 9600 bps
}

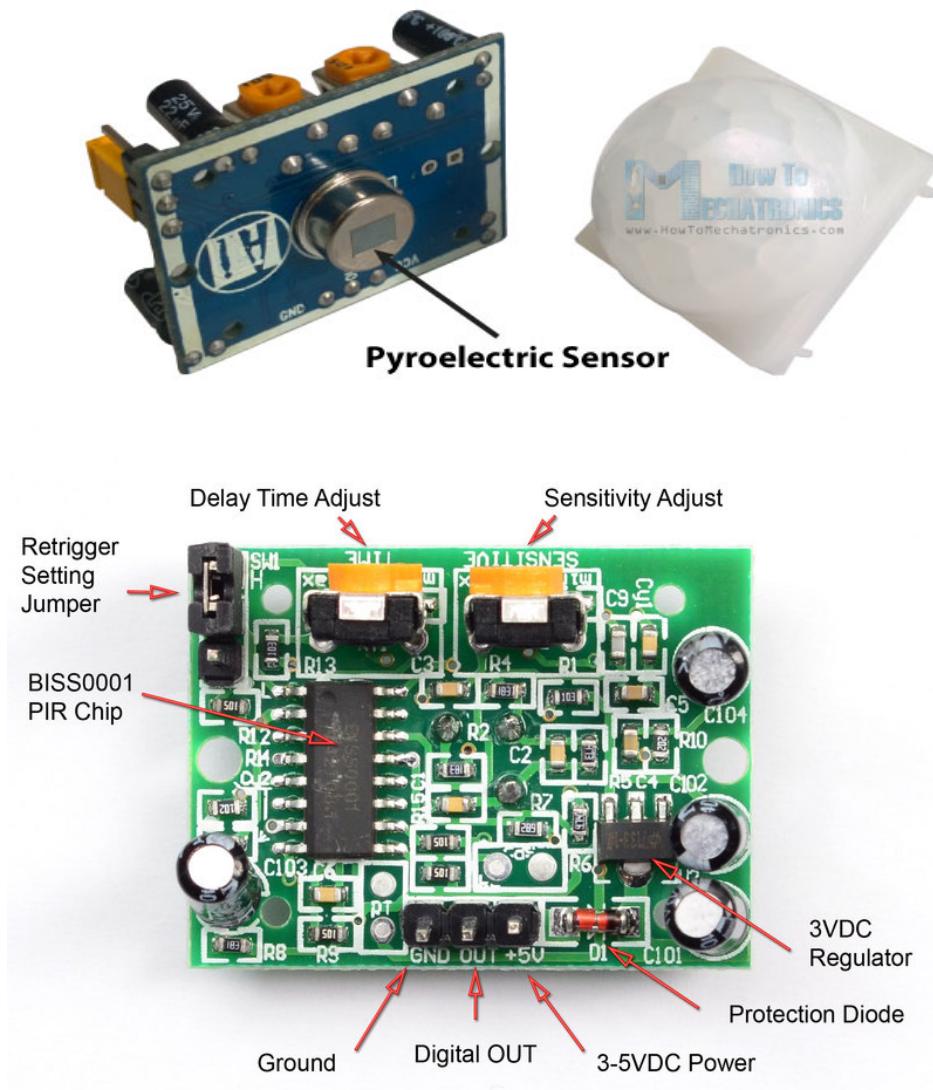
void loop()
{
    Serial.println(analogRead(gasPin));
    delay(1000); // Print value every 1 sec.
}
```

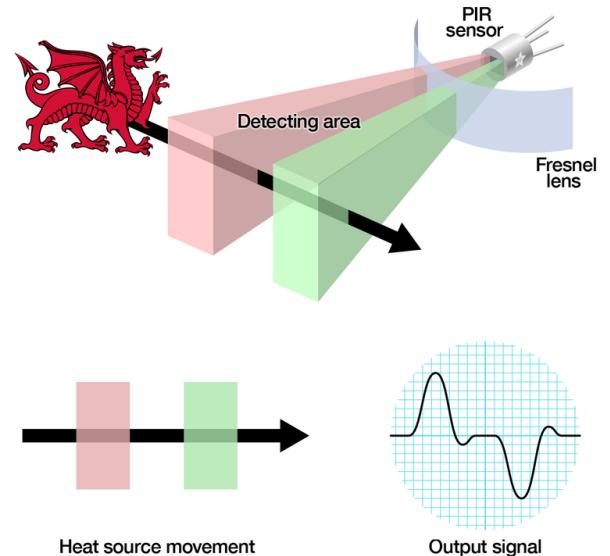
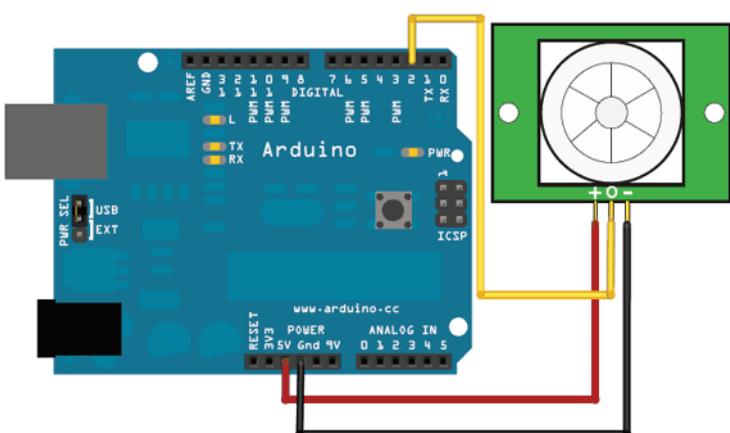
Output:the sensor gives gas concentration every sec.

PIR SENSOR

The Passive Infrared Sensor is used to detect the presence of a being in its field of view .Every person emits a specific amount of infrared rays and the pir sensor catches on that.When a person is in it's field of view the sensor changes the voltage in parallel to the change in the infrared signatures it's receiving, and this triggers a detection.

The PIR Sensors come in many kinds, the one I have used have a number of Fresnel Lenses which increases the range of detection to upto 30 ft and has a field of view to something less than 180 degrees.





The following Code prints “Motion Detected “ if there is a change in infrared rays in the field of view.

```

Upload Using Programmer
sensor_final
// choose the pin for the LED
inputPin = 2; // choose the input pin (for PIR sensor)
t ledpin = 10;
pirState = LOW; // we start, assuming no motion detected
val = 0; // variable for reading the pin status
t pinSpeaker = 10; //Set up a speaker on a PWM pin (digital 9, 10, or 11)

setup() {
  pinMode(inputPin, INPUT); // declare sensor as input
  pinMode(ledpin, OUTPUT); // declare sensor as input
  Serial.begin(9600);
}

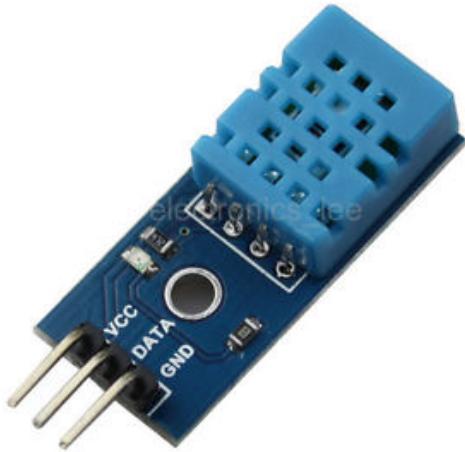
loop(){
  l = digitalRead(inputPin); // read input value
  (val == HIGH) { // check if the input is HIGH
    Serial.println("Motion detected");
    digitalWrite(ledpin, HIGH);
    delay(150);
  }

  else if (val == LOW) {
    // we have just turned on
    Serial.println("No motion detected");
    //digitalWrite(ledpin,LOW);
    // We only want to print on the output change, not state
    delay(150);
}

```

TEMPERATURE SENSOR

DHT11 - Temperature and Humidity Sensor

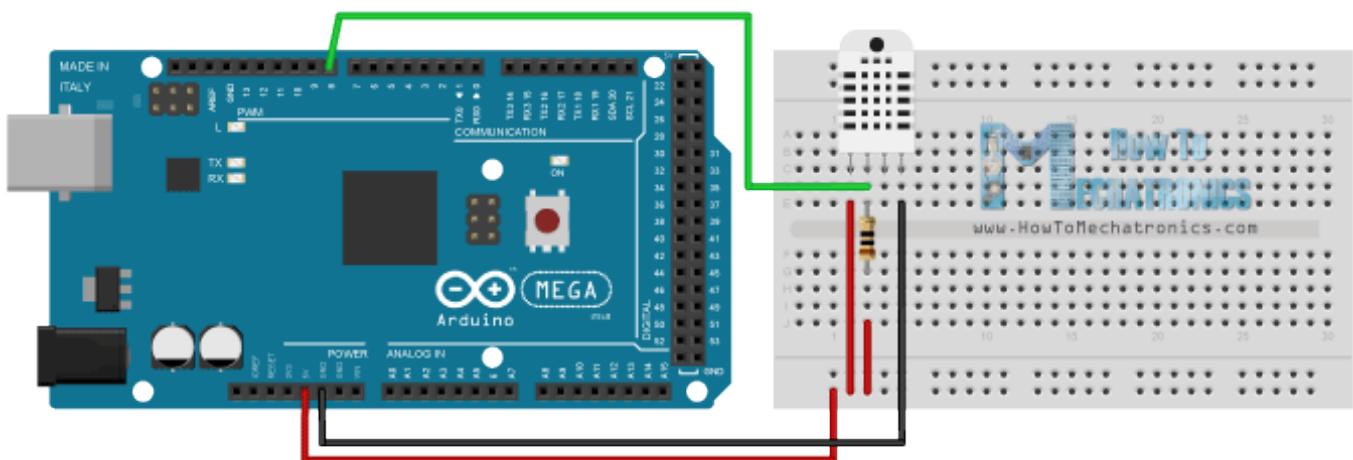


The DHT11 Temperature and humidity sensor can measure a range of 0-50 Degrees Celsius with +2 degrees accuracy.

Operating Voltage is 3V-5V

Interfacing with the Arduino :

We need to use a 5k to 10k Ohms resistor to enable the communication between the arduino and sensor board.



Source Code to interface with the Arduino :

```
DHT_Unified_Sensor_Final

uint32_t delayMS;

void setup() {
  Serial.begin(9600);
  // Initialize device.
  dht.begin();
  Serial.println("DHTxx Unified Sensor Example");
  // Print temperature sensor details.
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  Serial.println("-----");
  Serial.println("Temperature");
  Serial.print ("Sensor: "); Serial.println(sensor.name);
  Serial.print ("Driver Ver: "); Serial.println(sensor.version);
  Serial.print ("Unique ID: "); Serial.println(sensor.sensor_id);
  Serial.print ("Max Value: "); Serial.print(sensor.max_value); Serial.println(" *C");
  Serial.print ("Min Value: "); Serial.print(sensor.min_value); Serial.println(" *C");
  Serial.print ("Resolution: "); Serial.print(sensor.resolution); Serial.println(" *C");
  Serial.println("-----");
  // Print humidity sensor details.
  dht.humidity().getSensor(&sensor);
  Serial.println("-----");
  Serial.println("Humidity");
  Serial.print ("Sensor: "); Serial.println(sensor.name);
  Serial.print ("Driver Ver: "); Serial.println(sensor.version);
  Serial.print ("Unique ID: "); Serial.println(sensor.sensor_id);
  Serial.print ("Max Value: "); Serial.print(sensor.max_value); Serial.println("%");
  Serial.print ("Min Value: "); Serial.print(sensor.min_value); Serial.println("%");
  Serial.print ("Resolution: "); Serial.print(sensor.resolution); Serial.println("%");
  Serial.println("-----");
  // Set delay between sensor readings based on sensor details.
  delayMS = sensor.min_delay / 1000;
}


```

```
DHT_Unified_Sensor_Final

Serial.print ("Unique ID: "); Serial.println(sensor.sensor_id);
Serial.print ("Max Value: "); Serial.print(sensor.max_value); Serial.println("%");
Serial.print ("Min Value: "); Serial.print(sensor.min_value); Serial.println("%");
Serial.print ("Resolution: "); Serial.print(sensor.resolution); Serial.println("%");
Serial.println("-----");
// Set delay between sensor readings based on sensor details.
delayMS = sensor.min_delay / 1000;
}

void loop() {
  // Delay between measurements.
  delay(delayMS);
  // Get temperature event and print its value.
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  if (!isnan(event.temperature)) {
    Serial.println("Error reading temperature!");
  }
  else {
    Serial.print("Temperature: ");
    Serial.print(event.temperature);
    Serial.println(" *C");
  }
  // Get humidity event and print its value.
  dht.humidity().getEvent(&event);
  if (!isnan(event.relative_humidity)) {
    Serial.println("Error reading humidity!");
  }
  else {
    Serial.print("Humidity: ");
    Serial.print(event.relative_humidity);
    Serial.println("%");
  }
}


```

PUBNUB

Role Of Pubnub

I used Pubnub to transfer the real time data from the arduino via the internet using the Subscribe and Publish Key given to me by pubnub. So as long as the arduino is connected to the internet and the receiving system is connected to the internet, the system will work at full capacity.

Therefore you can access the information of your room and even control the lights from anywhere in the world.

View and Debug all PubNub Events 

SUBSCRIBE UNSUBSCRIBE SSL PAM 

connection status
CONNECTED - Press "Send" Button

channel my_channel	publish key demo
origin pubsub.pubnub.com	subscribe key demo
auth key <input type="checkbox"/> user-defined auth value	cipher key <input type="checkbox"/> user-defined encryption key
filter	

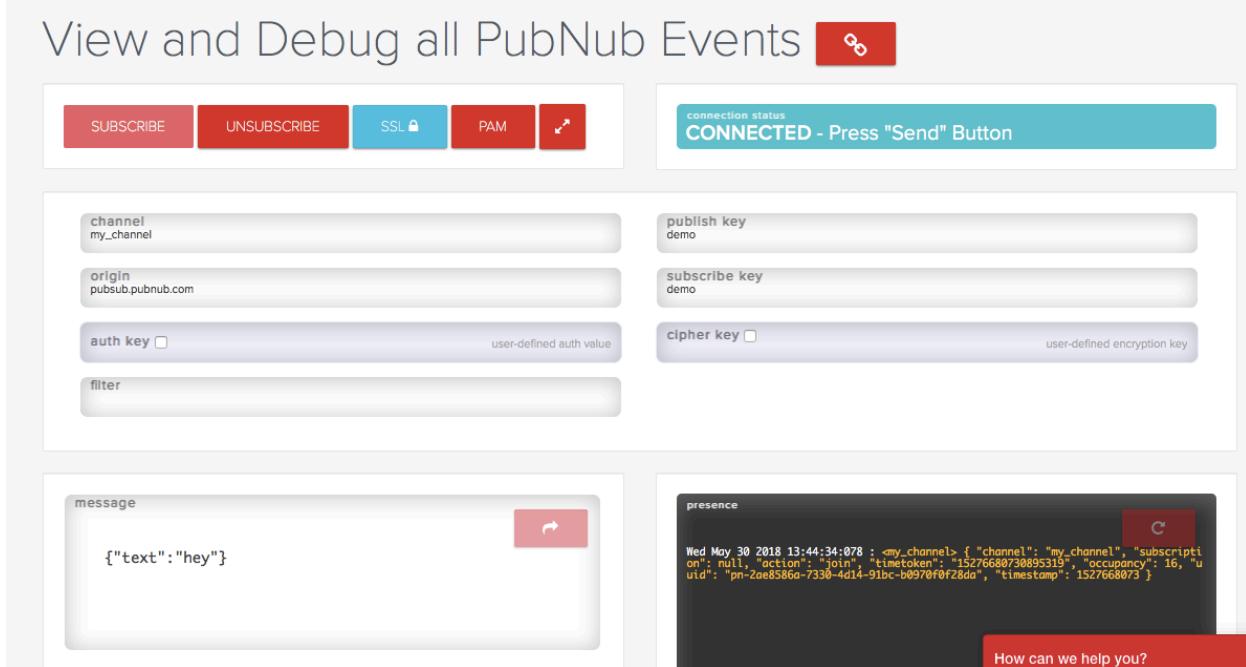
message 

```
{"text": "hey"}
```

presence 

```
Wed May 30 2018 13:44:34:078 : <my_channel> { "channel": "my_channel", "subscription": null, "action": "join", "timetoken": "15276680730895319", "occupancy": 16, "uid": "pn-2ae6586a-733b-4d14-91dc-b0970f0f28da", "timestamp": 1527668073 }
```

How can we help you?

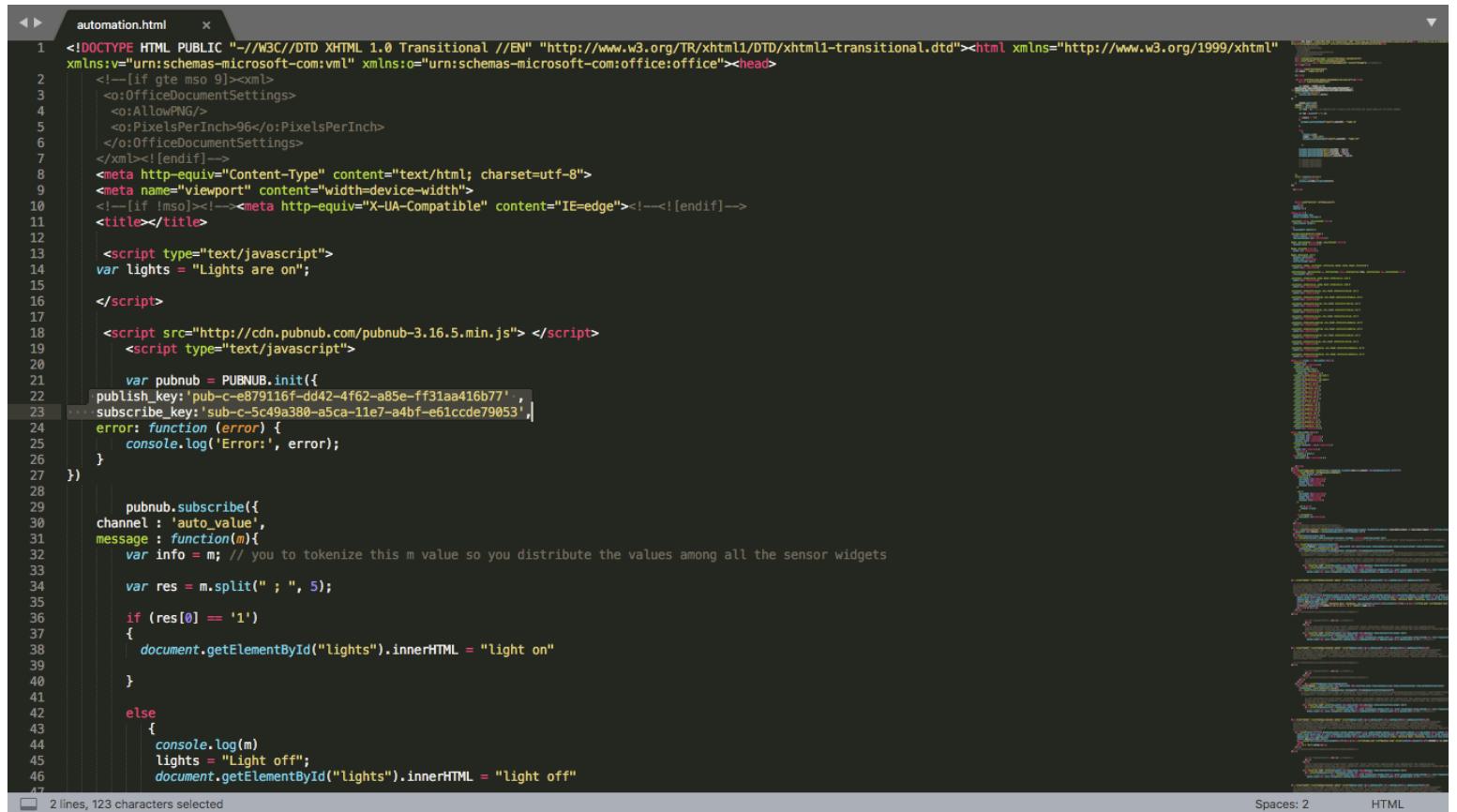


FRONT END

Dashboard

The Dashboard created for the project, which is part of the front end, is created only by using JavaScript. It contains information about the

- Temperature of the room
- Intruder Alert [Motion Detector]
- Gas [Warns if there is a gas leak]
- Network [Informs if the system is connected to the internet]



```
automation.html
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional //EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html xmlns="http://www.w3.org/1999/xhtml"
2 xmlns:v="urn:schemas-microsoft-com:vml" xmlns:o="urn:schemas-microsoft-com:office:office"><head>
3 <!--[if gte mso 9]><xmlelement vml="0">
4 <o:OfficeDocumentSettings>
5 <o:AllowPNG>
6 <o:PixelsPerInch>96</o:PixelsPerInch>
7 </o:OfficeDocumentSettings>
8 </xmlelement><!--endif-->
9 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
10 <meta name="viewport" content="width=device-width">
11 <!--[if !msie]><!--><meta http-equiv="X-UA-Compatible" content="IE=edge"><!--><!--[endif]-->
12 <title></title>
13 <script type="text/javascript">
14 var lights = "Lights are on";
15
16 </script>
17
18 <script src="http://cdn.pubnub.com/pubnub-3.16.5.min.js"> </script>
19 <script type="text/javascript">
20
21     var pubnub = PUBNUB.init({
22         publish_key:'pub-c-e879116f-dd42-4f62-a85e-ff31aa416b77' ,
23         subscribe_key:'sub-c-5c49a380-a5ca-11e7-a4bf-e61ccde79053' ,
24         error: function (error) {
25             console.log('Error:', error);
26         }
27     });
28
29     pubnub.subscribe({
30         channel : 'auto_value',
31         message : function(m){
32             var info = m; // you to tokenize this m value so you distribute the values among all the sensor widgets
33
34             var res = m.split(" ; ", 5);
35
36             if (res[0] == '1')
37             {
38                 document.getElementById("lights").innerHTML = "light on"
39             }
40
41             else
42             {
43                 console.log(m)
44                 lights = "Light off";
45                 document.getElementById("lights").innerHTML = "light off"
46             }
47         }
48     });
49
50     pubnub.publish({
51         channel : 'auto_value',
52         message : function(m){
53             var info = m; // you to tokenize this m value so you distribute the values among all the sensor widgets
54
55             var res = m.split(" ; ", 5);
56
57             if (res[0] == '1')
58             {
59                 document.getElementById("lights").innerHTML = "light on"
60             }
61
62             else
63             {
64                 console.log(m)
65                 lights = "Light off";
66                 document.getElementById("lights").innerHTML = "light off"
67             }
68         }
69     });
70
71     pubnub.setOptions({
72         log : false
73     });
74
75     pubnub.setLog(true);
76
77     pubnub.setLog(true);
78
79     pubnub.setLog(true);
80
81     pubnub.setLog(true);
82
83     pubnub.setLog(true);
84
85     pubnub.setLog(true);
86
87     pubnub.setLog(true);
88
89     pubnub.setLog(true);
90
91     pubnub.setLog(true);
92
93     pubnub.setLog(true);
94
95     pubnub.setLog(true);
96
97     pubnub.setLog(true);
98
99     pubnub.setLog(true);
100
101     pubnub.setLog(true);
102
103     pubnub.setLog(true);
104
105     pubnub.setLog(true);
106
107     pubnub.setLog(true);
108
109     pubnub.setLog(true);
110
111     pubnub.setLog(true);
112
113     pubnub.setLog(true);
114
115     pubnub.setLog(true);
116
117     pubnub.setLog(true);
118
119     pubnub.setLog(true);
120
121     pubnub.setLog(true);
122
123     pubnub.setLog(true);
124
125     pubnub.setLog(true);
126
127     pubnub.setLog(true);
128
129     pubnub.setLog(true);
130
131     pubnub.setLog(true);
132
133     pubnub.setLog(true);
134
135     pubnub.setLog(true);
136
137     pubnub.setLog(true);
138
139     pubnub.setLog(true);
140
141     pubnub.setLog(true);
142
143     pubnub.setLog(true);
144
145     pubnub.setLog(true);
146
147     pubnub.setLog(true);
148
149     pubnub.setLog(true);
150
151     pubnub.setLog(true);
152
153     pubnub.setLog(true);
154
155     pubnub.setLog(true);
156
157     pubnub.setLog(true);
158
159     pubnub.setLog(true);
160
161     pubnub.setLog(true);
162
163     pubnub.setLog(true);
164
165     pubnub.setLog(true);
166
167     pubnub.setLog(true);
168
169     pubnub.setLog(true);
170
171     pubnub.setLog(true);
172
173     pubnub.setLog(true);
174
175     pubnub.setLog(true);
176
177     pubnub.setLog(true);
178
179     pubnub.setLog(true);
180
181     pubnub.setLog(true);
182
183     pubnub.setLog(true);
184
185     pubnub.setLog(true);
186
187     pubnub.setLog(true);
188
189     pubnub.setLog(true);
190
191     pubnub.setLog(true);
192
193     pubnub.setLog(true);
194
195     pubnub.setLog(true);
196
197     pubnub.setLog(true);
198
199     pubnub.setLog(true);
200
201     pubnub.setLog(true);
202
203     pubnub.setLog(true);
204
205     pubnub.setLog(true);
206
207     pubnub.setLog(true);
208
209     pubnub.setLog(true);
210
211     pubnub.setLog(true);
212
213     pubnub.setLog(true);
214
215     pubnub.setLog(true);
216
217     pubnub.setLog(true);
218
219     pubnub.setLog(true);
220
221     pubnub.setLog(true);
222
223     pubnub.setLog(true);
224
225     pubnub.setLog(true);
226
227     pubnub.setLog(true);
228
229     pubnub.setLog(true);
230
231     pubnub.setLog(true);
232
233     pubnub.setLog(true);
234
235     pubnub.setLog(true);
236
237     pubnub.setLog(true);
238
239     pubnub.setLog(true);
240
241     pubnub.setLog(true);
242
243     pubnub.setLog(true);
244
245     pubnub.setLog(true);
246
247     pubnub.setLog(true);
248
249     pubnub.setLog(true);
250
251     pubnub.setLog(true);
252
253     pubnub.setLog(true);
254
255     pubnub.setLog(true);
256
257     pubnub.setLog(true);
258
259     pubnub.setLog(true);
260
261     pubnub.setLog(true);
262
263     pubnub.setLog(true);
264
265     pubnub.setLog(true);
266
267     pubnub.setLog(true);
268
269     pubnub.setLog(true);
270
271     pubnub.setLog(true);
272
273     pubnub.setLog(true);
274
275     pubnub.setLog(true);
276
277     pubnub.setLog(true);
278
279     pubnub.setLog(true);
280
281     pubnub.setLog(true);
282
283     pubnub.setLog(true);
284
285     pubnub.setLog(true);
286
287     pubnub.setLog(true);
288
289     pubnub.setLog(true);
290
291     pubnub.setLog(true);
292
293     pubnub.setLog(true);
294
295     pubnub.setLog(true);
296
297     pubnub.setLog(true);
298
299     pubnub.setLog(true);
300
301     pubnub.setLog(true);
302
303     pubnub.setLog(true);
304
305     pubnub.setLog(true);
306
307     pubnub.setLog(true);
308
309     pubnub.setLog(true);
310
311     pubnub.setLog(true);
312
313     pubnub.setLog(true);
314
315     pubnub.setLog(true);
316
317     pubnub.setLog(true);
318
319     pubnub.setLog(true);
320
321     pubnub.setLog(true);
322
323     pubnub.setLog(true);
324
325     pubnub.setLog(true);
326
327     pubnub.setLog(true);
328
329     pubnub.setLog(true);
330
331     pubnub.setLog(true);
332
333     pubnub.setLog(true);
334
335     pubnub.setLog(true);
336
337     pubnub.setLog(true);
338
339     pubnub.setLog(true);
340
341     pubnub.setLog(true);
342
343     pubnub.setLog(true);
344
345     pubnub.setLog(true);
346
347     pubnub.setLog(true);
348
349     pubnub.setLog(true);
350
351     pubnub.setLog(true);
352
353     pubnub.setLog(true);
354
355     pubnub.setLog(true);
356
357     pubnub.setLog(true);
358
359     pubnub.setLog(true);
360
361     pubnub.setLog(true);
362
363     pubnub.setLog(true);
364
365     pubnub.setLog(true);
366
367     pubnub.setLog(true);
368
369     pubnub.setLog(true);
370
371     pubnub.setLog(true);
372
373     pubnub.setLog(true);
374
375     pubnub.setLog(true);
376
377     pubnub.setLog(true);
378
379     pubnub.setLog(true);
380
381     pubnub.setLog(true);
382
383     pubnub.setLog(true);
384
385     pubnub.setLog(true);
386
387     pubnub.setLog(true);
388
389     pubnub.setLog(true);
390
391     pubnub.setLog(true);
392
393     pubnub.setLog(true);
394
395     pubnub.setLog(true);
396
397     pubnub.setLog(true);
398
399     pubnub.setLog(true);
400
401     pubnub.setLog(true);
402
403     pubnub.setLog(true);
404
405     pubnub.setLog(true);
406
407     pubnub.setLog(true);
408
409     pubnub.setLog(true);
410
411     pubnub.setLog(true);
412
413     pubnub.setLog(true);
414
415     pubnub.setLog(true);
416
417     pubnub.setLog(true);
418
419     pubnub.setLog(true);
420
421     pubnub.setLog(true);
422
423     pubnub.setLog(true);
424
425     pubnub.setLog(true);
426
427     pubnub.setLog(true);
428
429     pubnub.setLog(true);
430
431     pubnub.setLog(true);
432
433     pubnub.setLog(true);
434
435     pubnub.setLog(true);
436
437     pubnub.setLog(true);
438
439     pubnub.setLog(true);
440
441     pubnub.setLog(true);
442
443     pubnub.setLog(true);
444
445     pubnub.setLog(true);
446
447     pubnub.setLog(true);
448
449     pubnub.setLog(true);
450
451     pubnub.setLog(true);
452
453     pubnub.setLog(true);
454
455     pubnub.setLog(true);
456
457     pubnub.setLog(true);
458
459     pubnub.setLog(true);
460
461     pubnub.setLog(true);
462
463     pubnub.setLog(true);
464
465     pubnub.setLog(true);
466
467     pubnub.setLog(true);
468
469     pubnub.setLog(true);
470
471     pubnub.setLog(true);
472
473     pubnub.setLog(true);
474
475     pubnub.setLog(true);
476
477     pubnub.setLog(true);
478
479     pubnub.setLog(true);
480
481     pubnub.setLog(true);
482
483     pubnub.setLog(true);
484
485     pubnub.setLog(true);
486
487     pubnub.setLog(true);
488
489     pubnub.setLog(true);
490
491     pubnub.setLog(true);
492
493     pubnub.setLog(true);
494
495     pubnub.setLog(true);
496
497     pubnub.setLog(true);
498
499     pubnub.setLog(true);
500
501     pubnub.setLog(true);
502
503     pubnub.setLog(true);
504
505     pubnub.setLog(true);
506
507     pubnub.setLog(true);
508
509     pubnub.setLog(true);
510
511     pubnub.setLog(true);
512
513     pubnub.setLog(true);
514
515     pubnub.setLog(true);
516
517     pubnub.setLog(true);
518
519     pubnub.setLog(true);
520
521     pubnub.setLog(true);
522
523     pubnub.setLog(true);
524
525     pubnub.setLog(true);
526
527     pubnub.setLog(true);
528
529     pubnub.setLog(true);
530
531     pubnub.setLog(true);
532
533     pubnub.setLog(true);
534
535     pubnub.setLog(true);
536
537     pubnub.setLog(true);
538
539     pubnub.setLog(true);
540
541     pubnub.setLog(true);
542
543     pubnub.setLog(true);
544
545     pubnub.setLog(true);
546
547     pubnub.setLog(true);
548
549     pubnub.setLog(true);
550
551     pubnub.setLog(true);
552
553     pubnub.setLog(true);
554
555     pubnub.setLog(true);
556
557     pubnub.setLog(true);
558
559     pubnub.setLog(true);
560
561     pubnub.setLog(true);
562
563     pubnub.setLog(true);
564
565     pubnub.setLog(true);
566
567     pubnub.setLog(true);
568
569     pubnub.setLog(true);
570
571     pubnub.setLog(true);
572
573     pubnub.setLog(true);
574
575     pubnub.setLog(true);
576
577     pubnub.setLog(true);
578
579     pubnub.setLog(true);
580
581     pubnub.setLog(true);
582
583     pubnub.setLog(true);
584
585     pubnub.setLog(true);
586
587     pubnub.setLog(true);
588
589     pubnub.setLog(true);
590
591     pubnub.setLog(true);
592
593     pubnub.setLog(true);
594
595     pubnub.setLog(true);
596
597     pubnub.setLog(true);
598
599     pubnub.setLog(true);
599 2 lines, 123 characters selected
855 856 Spaces: 2
857 858 HTML
```

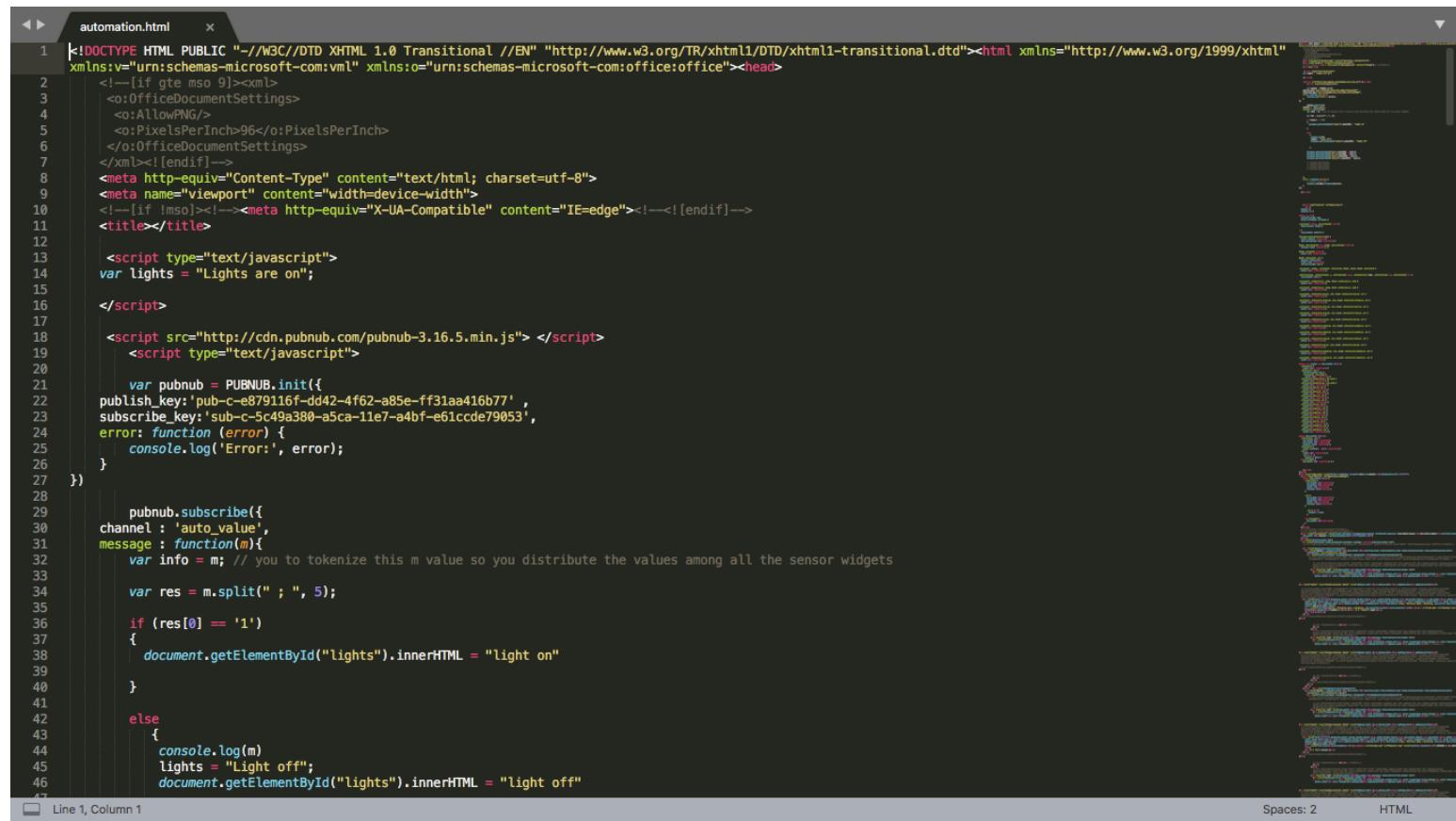
The subscribe key: 'sub-c-5c49a380-a5ca-11e7-a4bf-e61ccde79053',

The Publish Key: 'pub-c-e879116f-dd42-4f62-a85e-ff31aa416b77',

Are provided by pubnub itself and is used to isolate and connect the correct home automation device to the respective dashboard.

By simply changing the keys we can have multiple devices and multiple dashboards for different rooms.

You can find the complete code in the annex and also in the screenshot below.



The screenshot shows a code editor window with the file 'automation.html' open. The code is written in HTML and JavaScript. It includes meta tags for content type and viewport, a title, and a script that initializes PubNub with publish and subscribe keys. The main logic is in a script block that subscribes to a channel named 'auto_value'. It tokenizes the received message 'm' into an array 'res' and checks if the first element is '1'. If true, it sets the innerHTML of an element with id 'lights' to 'light on'. If false, it logs the message to the console, sets 'lights' to 'Light off', and updates the element's innerHTML. The code editor interface shows line numbers from 1 to 47, a status bar at the bottom indicating 'Line 1, Column 1', 'Spaces: 2', and 'HTML', and a sidebar on the right containing a tree view of files and a search bar.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional //EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:v="urn:schemas-microsoft-com:vml" xmlns:o="urn:schemas-microsoft-com:office:office"><head>
  <!--[if gte mso 9]><xml>
    <o:OfficeDocumentSettings>
      <o:AllowPNG/>
      <o:PixelsPerInch>96</o:PixelsPerInch>
    </o:OfficeDocumentSettings>
  </xml><!--[endif]-->
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width">
  <!--[if !msie]><!--><meta http-equiv="X-UA-Compatible" content="IE=edge"><!--><!--[endif]-->
  <title></title>
  <script type="text/javascript">
    var lights = "Lights are on";
  </script>
  <script src="http://cdn.pubnub.com/pubnub-3.16.5.min.js"> </script>
  <script type="text/javascript">
    var pubnub = PUBNUB.init({
      publish_key: 'pub-c-e879116f-dd42-4f62-a85e-ff31aa416b77',
      subscribe_key: 'sub-c-5c49a380-a5ca-11e7-a4bf-e61ccde79053',
      error: function (error) {
        console.log('Error:', error);
      }
    });

    pubnub.subscribe({
      channel : 'auto_value',
      message : function(m){
        var info = m; // you to tokenize this m value so you distribute the values among all the sensor widgets
        var res = m.split(" ", 5);

        if (res[0] == '1'){
          document.getElementById("lights").innerHTML = "light on"
        }
        else{
          console.log(m)
          lights = "Light off";
          document.getElementById("lights").innerHTML = "light off"
        }
      }
    });
  </script>
</head>
<body>
  <div id="lights">Lights are on</div>
</body>
</html>
```

```
48
49     }
50
51     document.getElementById("gas").innerHTML = res[3];
52     document.getElementById("pir").innerHTML = res[2];
53     document.getElementById("temp").innerHTML = res[0];
54     document.getElementById("network").innerHTML = res[4];
55
56     // console.log(res[1]);
57     // console.log(res[2]);
58     // console.log(res[3]);
59     // console.log(res[4]);
60
61
62 },
63 error : function (error) {
64     // Handle error here
65     console.log(JSON.stringify(error));
66 }
67 });
68
69 </script>
70
71
72
73
74
75
76 <style type="text/css" id="media-query">
77   body {
78     margin: 0;
79     padding: 0; }
80
81 table, tr, td {
82   vertical-align: top;
83   border-collapse: collapse; }
84
85 .ie-browser table, .mso-container table {
86   table-layout: fixed; }
87
88 * {
89   line-height: inherit; }
90
91 a[x-apple-data-detectors=true] {
92   color: inherit !important;
93   text-decoration: none !important; }
94
```

Line 1, Column 1

Spaces: 2

HTML

```
94 [owa] .img-container div, [owa] .img-container button {
95   display: block !important; }
96
97 [owa] .fullwidth button {
98   width: 100% !important; }
99
100 [owa] .block-grid .col {
101   display: table-cell;
102   float: none !important;
103   vertical-align: top; }
104
105 .ie-browser .num12, .ie-browser .block-grid, [owa] .num12, [owa] .block-grid {
106   width: 500px !important; }
107
108 .ExternalClass, .ExternalClass p, .ExternalClass span, .ExternalClass font, .ExternalClass td, .ExternalClass div {
109   line-height: 100%; }
110
111 .ie-browser .mixed-two-up .num4, [owa] .mixed-two-up .num4 {
112   width: 164px !important; }
113
114 .ie-browser .mixed-two-up .num8, [owa] .mixed-two-up .num8 {
115   width: 328px !important; }
116
117 .ie-browser .block-grid.two-up .col, [owa] .block-grid.two-up .col {
118   width: 250px !important; }
119
120 .ie-browser .block-grid.three-up .col, [owa] .block-grid.three-up .col {
121   width: 166px !important; }
122
123 .ie-browser .block-grid.four-up .col, [owa] .block-grid.four-up .col {
124   width: 125px !important; }
125
126 .ie-browser .block-grid.five-up .col, [owa] .block-grid.five-up .col {
127   width: 100px !important; }
128
129 .ie-browser .block-grid.six-up .col, [owa] .block-grid.six-up .col {
130   width: 83px !important; }
131
132 .ie-browser .block-grid.seven-up .col, [owa] .block-grid.seven-up .col {
133   width: 71px !important; }
134
135 .ie-browser .block-grid.eight-up .col, [owa] .block-grid.eight-up .col {
136   width: 62px !important; }
137
138 .ie-browser .block-grid.nine-up .col, [owa] .block-grid.nine-up .col {
139   width: 55px !important; }
140
```

Line 1, Column 1

Spaces: 2

HTML

```

automation.html x
185 @media (max-width: 520px) {
186   .block-grid, .col {
187     min-width: 320px !important;
188     max-width: 100% !important;
189     display: block !important;
190   }
191   .block-grid {
192     width: calc(100% - 40px) !important;
193   }
194   .col {
195     width: 100% !important;
196     .col > div {
197       margin: 0 auto;
198     }
199     img.fullwidth {
200       max-width: 100% !important;
201     }
202   }
203 
```

Line 1, Column 1 Spaces: 2 HTML

```

automation.html x
306   </div>
307   </div>
308   <!--[if (mso)|(IE)]--></td><td align="center" width="250" style=" width:250px; padding-right: 0px; padding-left: 0px; padding-top:5px; padding-bottom:5px; border-top: 0px solid transparent; border-left: 0px solid transparent; border-bottom: 0px solid transparent; border-right: 0px solid transparent;" valign="top"<!--[endif]-->
309   <div class="col num6" style="max-width: 320px;min-width: 250px;display: table-cell;vertical-align: top;">
310     <div style="background-color: transparent; width: 100% !important;">
311       <!--[if !(mso)&(IE)]--><!--><div style="border-top: 0px solid transparent; border-left: 0px solid transparent; border-bottom: 0px solid transparent; border-right: 0px solid transparent; padding-top:5px; padding-bottom:5px; padding-right: 0px; padding-left: 0px;"><!--[endif]-->
312     <div align="center" class="button-container center" style="padding-right: 10px; padding-left: 10px; padding-top:10px; padding-bottom:10px;">
313       <!--[if mso]><table width="100%" cellpadding="0" cellspacing="0" border="0" style="border-spacing: 0; border-collapse: collapse; mso-table-lspace:0pt; mso-table-rspace:0pt;"><tr><td style="padding-right: 10px; padding-left: 10px; padding-top:10px; padding-bottom:10px;" align="center"><v:roundrect xmlns:v="urn:schemas-microsoft-com:vml" xmlns:w="urn:schemas-microsoft-com:office:word" href="#" style="height:92px; v-text-anchor:middle; width:112px;" arcsize="5%" strokecolor="#3AAEE8" fillcolor="#3AAEE8"><w:anchorlock/><center style="color:#ffffff; font-family:Arial, 'Helvetica Neue', Helvetica, sans-serif; font-size:16px;"><!--[endif]-->
314         <div style="color: #ffffff; background-color: orange; border-radius: 4px; -webkit-border-radius: 4px; -moz-border-radius: 4px; max-width: 130px; width: 52px; width: auto; border-top: 0px solid transparent; border-right: 0px solid transparent; border-bottom: 0px solid transparent; border-left: 0px solid transparent; padding-top: 30px; padding-right: 30px; padding-bottom: 30px; padding-left: 30px; font-family: Arial, 'Helvetica Neue', Helvetica, sans-serif; text-align: center; mso-border-alt: none;">
315           <span style="font-size:18px; line-height:32px;"> </span>  TEMPERATURE<br/> <br/>
316         <div id = "temp"> 0 </div>
317       </div>
318       <!--[if mso]></center></v:roundrect></td></tr></table><!--[endif]-->
319     </div>
320   <!--[if !(mso)|(IE)]--><!--><div><!--[endif]-->
321   <!--[if !(mso)|(IE)]--><!--><div><!--[endif]-->
322   <!--[if !(mso)|(IE)]--><!--><div><!--[endif]-->
323   <!--[if !(mso)|(IE)]--><!--><div><!--[endif]-->
324   <!--[if !(mso)|(IE)]--><!--><div><!--[endif]-->
325   <!--[if !(mso)|(IE)]--><!--><div><!--[endif]-->
326   <!--[if !(mso)|(IE)]--><!--><div><!--[endif]-->
327   <!--[if !(mso)|(IE)]--></td></tr></table></td></tr></table><!--[endif]-->
328   </div>
329   </div>
330   </div>
331   <div style="background-color: transparent;">
332     <div style="Margin: 0 auto;min-width: 320px;max-width: 500px;overflow-wrap: break-word;word-wrap: break-word;word-break: break-word;background-color: transparent;" class="block-grid two-up">
333       <div style="border-collapse: collapse;display: table; width: 100%;background-color:transparent;">
334         <!--[if (mso)|(IE)]--><table width="100%" cellpadding="0" cellspacing="0" border="0" style="background-color:transparent;" align="center"><tr><td style="background-color:transparent;" align="center"><table cellpadding="0" cellspacing="0" border="0" style="width: 500px;"><tr class="layout-full-width" style="background-color:transparent;"><!--[endif]-->
335         <!--[if (mso)|(IE)]--><td align="center" width="250" style=" width:250px; padding-right: 0px; padding-left: 0px; padding-top:5px; padding-bottom:5px; border-top: 0px solid transparent; border-left: 0px solid transparent; border-bottom: 0px solid transparent; border-right: 0px solid transparent;" valign="top"><!--[endif]-->
336         <div class="col num6" style="max-width: 320px;min-width: 250px;display: table-cell;vertical-align: top;">
337           <div style="background-color: transparent; width: 100% !important;">
338             <!--[if !(mso)|(IE)]--><!--><div><!--[endif]-->
339           </div>
340         </div>
341       </div>
342     </div>
343   </div>
344 
```

Line 1, Column 1 Spaces: 2 HTML

```

automation.html
264     solid transparent;" valign="top">><![endif]>-->
265 <div class="col num6" style="max-width: 320px;min-width: 250px;display: table-cell;vertical-align: top;">
266     <!--[if !mso&(IE)]><!--><div style="border-top: 0px solid transparent; border-left: 0px solid transparent; border-bottom: 0px solid transparent; border-right: 0px solid transparent; padding-top:5px; padding-bottom:5px; padding-right: 0px; padding-left: 0px;"><![endif]>-->
267
268
269
270 <div align="center" class="button-container center" style="padding-right: 10px; padding-left: 10px; padding-top:10px; padding-bottom:10px;">
271     <!--[if mso]><table width="100%" cellpadding="0" cellspacing="0" border="0" style="border-spacing: 0; border-collapse: collapse; mso-table-lspace:0pt; mso-table-rspace:0pt;"><tr><td style="padding-right: 10px; padding-left: 10px; padding-top:10px; padding-bottom:10px;" align="center"><v:roundrect
272         xmlns:v="urn:schemas-microsoft-com:vml" xmlns:w="urn:schemas-microsoft-com:office:word" href="" style="height:92px; v-text-anchor:middle; width:112px;" arcsize="5%" strokeweight="#3AAEE0" fillcolor="#3AAEE0"><w:anchorlock/><center style="color:#ffffff; font-family:Arial, 'Helvetica Neue', Helvetica, sans-serif; font-size:16px;"><![endif]>-->
273 </td></tr></table><![endif]>-->
274 </div>
275
276
277     <!--[if (!mso)&(!IE)]><!--></div><!--><![endif]>-->
278     </div>
279 </div>
280     <!--[if (mso)|(IE)]></td></tr></table></td></tr></table><![endif]>-->
281 </div>
282 </div>
283 </div>
284     <div style="background-color:transparent;">
285         <div style="Margin: 0 auto;min-width: 320px;max-width: 500px;overflow-wrap: break-word;word-wrap: break-word;word-break: break-word;background-color: transparent;" class="block-grid two-up">
286             <div style="border-collapse: collapse;display: table;width: 100%;background-color:transparent;">
287                 <!--[if (mso)|(IE)]><table width="100%" cellpadding="0" cellspacing="0" border="0"><tr><td style="background-color:transparent;" align="center"><table
288                     cellpadding="0" cellspacing="0" border="0" style="width: 500px;"><tr class="layout-full-width" style="background-color:transparent;"><td>
289                         <!--[if (mso)|(IE)]><td align="center" width="250" style="width:250px; padding-right: 0px; padding-left: 0px; padding-top:5px; padding-bottom:5px; border-top: 0px solid transparent; border-left: 0px solid transparent; border-bottom: 0px solid transparent; border-right: 0px solid transparent;">
290                         <div class="col num6" style="max-width: 320px;min-width: 250px;display: table-cell;vertical-align: top;">
291                             <div style="background-color: transparent; width: 100% !important;">
292                                 <!--[if (!mso)&(!IE)]><!--><div style="border-top: 0px solid transparent; border-left: 0px solid transparent; border-bottom: 0px solid transparent; border-right: 0px solid transparent; padding-top:5px; padding-bottom:5px; padding-right: 0px; padding-left: 0px;"><![endif]>-->
293
294
295 <div align="center" class="button-container center" style="padding-right: 10px; padding-left: 10px; padding-top:10px; padding-bottom:10px;">
296     <!--[if mso]><table width="100%" cellpadding="0" cellspacing="0" border="0" style="border-spacing: 0; border-collapse: collapse; mso-table-lspace:0pt; mso-table-rspace:0pt;"><tr><td style="padding-right: 10px; padding-left: 10px; padding-top:10px; padding-bottom:10px;" align="center"><v:roundrect
297         xmlns:v="urn:schemas-microsoft-com:vml" xmlns:w="urn:schemas-microsoft-com:office:word" href="" style="height:92px; v-text-anchor:middle; width:112px;" arcsize="5%" strokeweight="#3AAEE0" fillcolor="#3AAEE0"><w:anchorlock/><center style="color:#ffffff; font-family:Arial, 'Helvetica Neue', Helvetica, sans-serif; font-size:16px;"><![endif]>-->
```

Line 1, Column 1

Spaces: 2

HTML

```

automation.html
351
352     <!--[if (!mso)&(!IE)]><!--></div><!--><![endif]>-->
353     </div>
354 </div>
355     <!--[if (mso)|(IE)]></td align="center" width="250" style="width:250px; padding-right: 0px; padding-left: 0px; padding-top:5px; padding-bottom:5px; border-top: 0px solid transparent; border-left: 0px solid transparent; border-bottom: 0px solid transparent; border-right: 0px solid transparent;">
356     <div class="col num6" style="max-width: 320px;min-width: 250px;display: table-cell;vertical-align: top;">
357         <div style="background-color: transparent; width: 100% !important;">
358             <!--[if (!mso)&(!IE)]><!--><div style="border-top: 0px solid transparent; border-left: 0px solid transparent; border-bottom: 0px solid transparent; border-right: 0px solid transparent; padding-top:5px; padding-bottom:5px; padding-right: 0px; padding-left: 0px;"><![endif]>-->
359
360
361
362 <div align="center" class="button-container center" style="padding-right: 10px; padding-left: 10px; padding-top:10px; padding-bottom:10px;">
363     <!--[if mso]><table width="100%" cellpadding="0" cellspacing="0" border="0" style="border-spacing: 0; border-collapse: collapse; mso-table-lspace:0pt; mso-table-rspace:0pt;"><tr><td style="padding-right: 10px; padding-left: 10px; padding-top:10px; padding-bottom:10px;" align="center"><v:roundrect
364         xmlns:v="urn:schemas-microsoft-com:vml" xmlns:w="urn:schemas-microsoft-com:office:word" href="" style="height:92px; v-text-anchor:middle; width:112px;" arcsize="5%" strokeweight="#3AAEE0" fillcolor="#3AAEE0"><w:anchorlock/><center style="color:#ffffff; font-family:Arial, 'Helvetica Neue', Helvetica, sans-serif; font-size:16px;"><![endif]>-->
365         <div style="color: #ffffff; background-color: green; border-radius: 4px; -webkit-border-radius: 4px; -moz-border-radius: 4px; max-width: 112px; width: 52px; width: auto; border-top: 0px solid transparent; border-right: 0px solid transparent; border-bottom: 0px solid transparent; border-left: 0px solid transparent; padding-top: 30px; padding-right: 30px; padding-bottom: 30px; padding-left: 30px; font-family: Arial, 'Helvetica Neue', Helvetica, sans-serif; font-size:16px; line-height:32px;"></div>  NETWORK <br> <br>
366         <div id = "network"> connect </div>
367
368
369     </div>
370     <!--[if mso]></center></v:roundrect></td></tr></table><![endif]>-->
371 </div>
372
373
374     <!--[if (!mso)&(!IE)]><!--></div><!--><![endif]>-->
375     </div>
376     </div>
377     <!--[if (mso)|(IE)]></td></tr></table></td></tr></table><![endif]>-->
378 </div>
379 </div>
380 </div>
381 </td>
382 </tr>
383 </tbody>
384 </table>
385 <!--[if (mso)|(IE)]></div><![endif]>-->
386
387 </body></html>

```

Line 1, Column 1

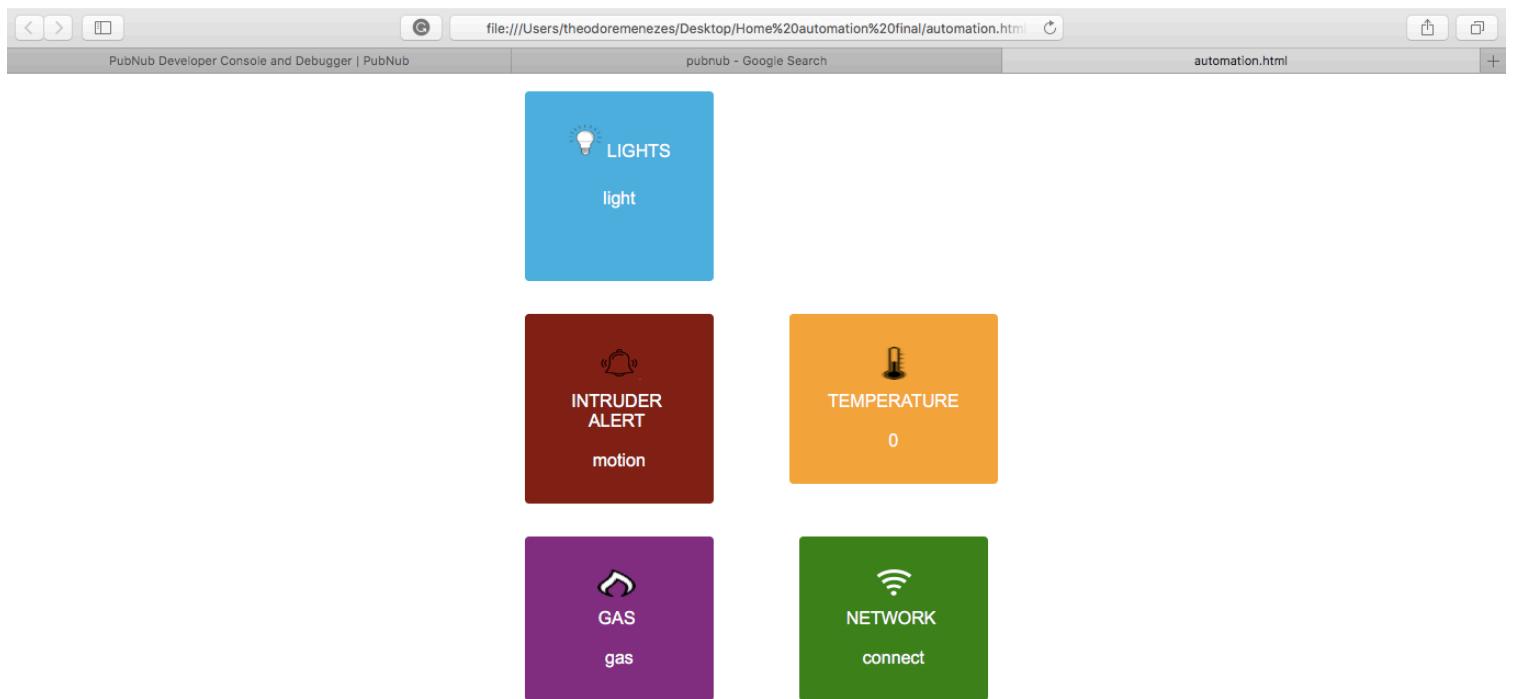
Spaces: 2

HTML

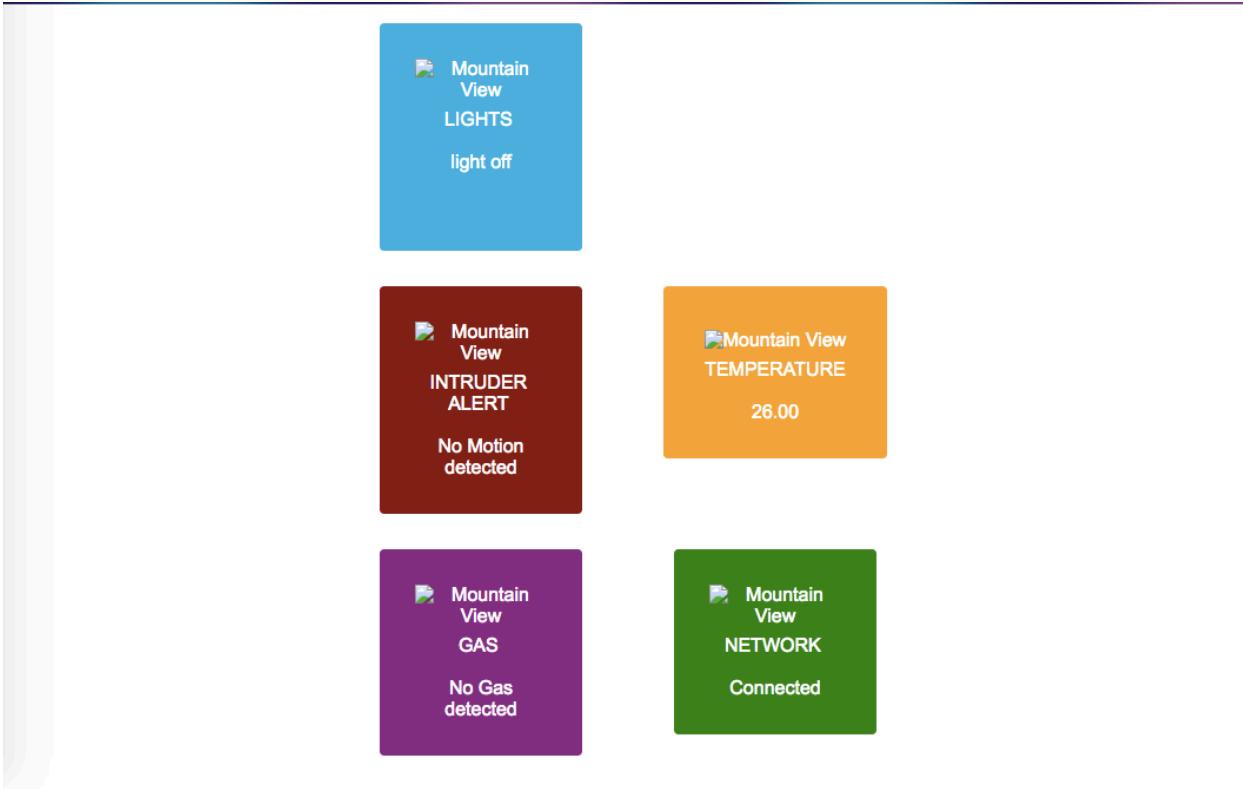
Dashboard Front-End

User View

Below is the Front End of the Dashboard the final user can see and will interact with.



Dashboard when not connected to the Internet[Above]



Dashboard Connected to the Internet

Hands On Control Of Lights from the receiving Computer

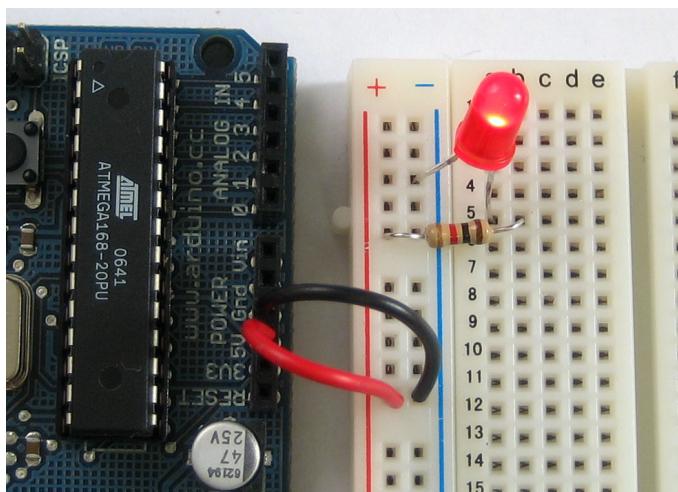
I built a model where you could switch on/off the lights in your room from anywhere around the world or even when you are in the room and do not want to physically get up to switch it off/on.

Working:

I used an LED to illustrate the concept but this could be built to work with real lights, the lights will require an independent power source and a relay so that it could be connected to the arduino.

Process:

Webpage coded in javascript ,incorporated the subscribe key and publish key from pubnub .Every button has a backend which switches on/off the lights.

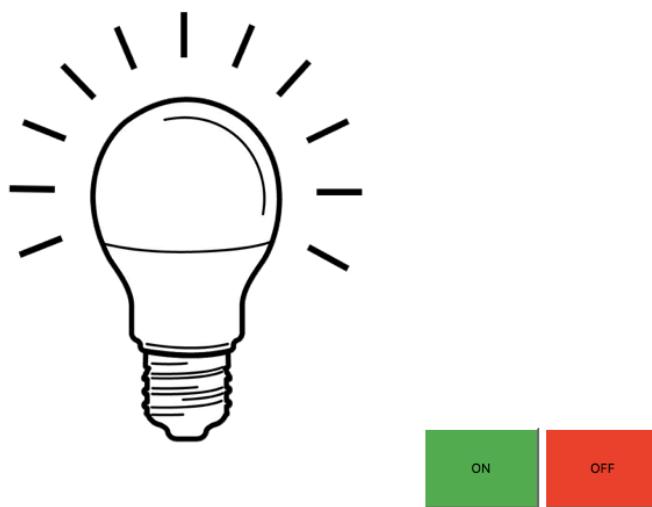


User Interface

Control Lights

turning lights on and off

THANK YOU TECHNOLOGY

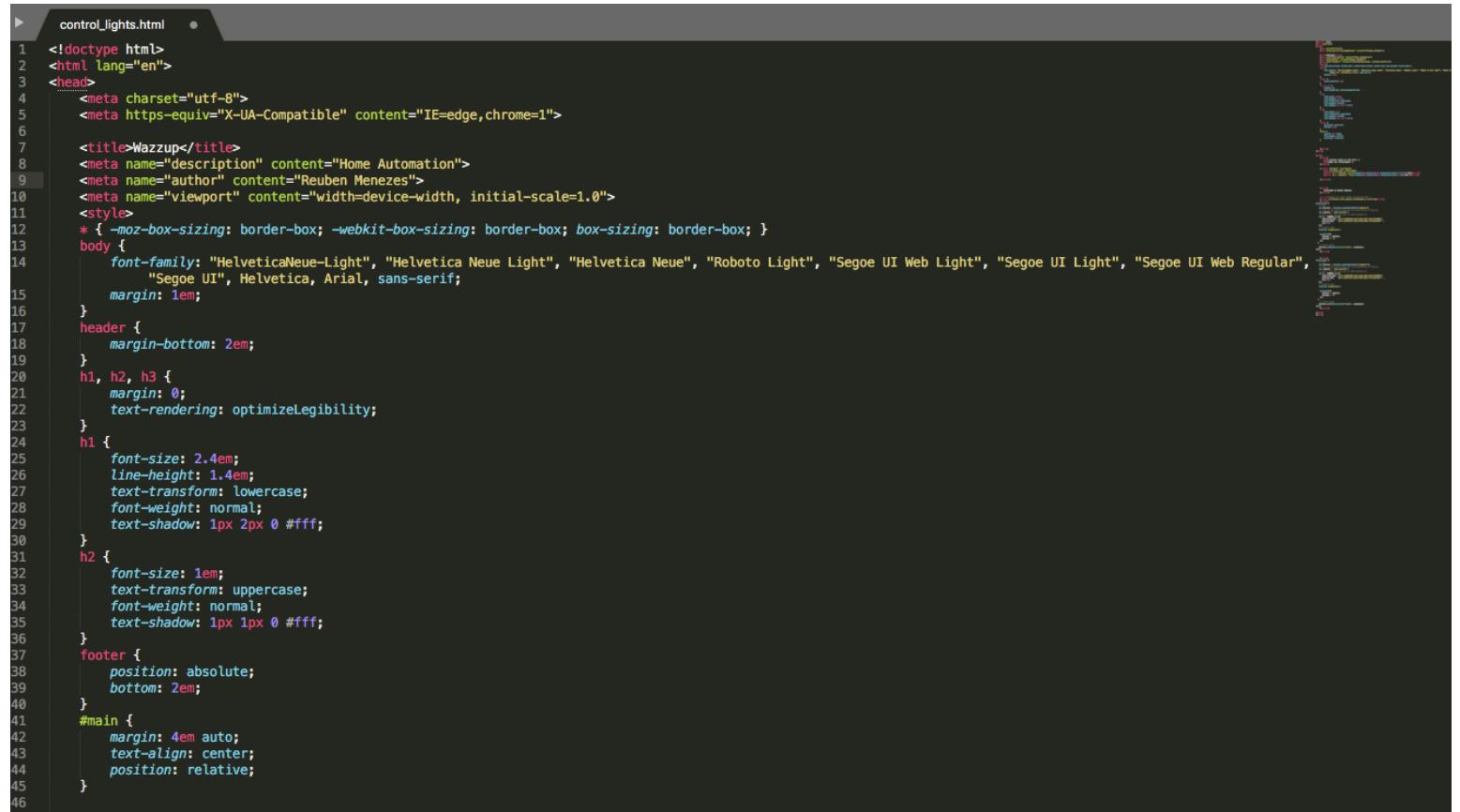


Developed by Reuben Menezes

By clicking on the On / Off buttons you can control the lights in your room.

Back-End

Control Lights



A screenshot of a code editor window titled "control_lights.html". The code is written in HTML and CSS. The CSS includes meta tags for charset, viewport, and various styles for the body, header, h1, h2, h3, and footer elements. The font-family for the body is set to a sans-serif font stack. The header has a margin-bottom of 2em. H1, H2, and H3 have margins of 0 and text-rendering: optimizeLegibility. H1 has a font-size of 2.4em and text-shadow: 1px 2px 0 #fff. H2 has a font-size of 1em, text-transform: uppercase, and text-shadow: 1px 1px 0 #fff. The footer is positioned absolutely at the bottom. The main container has a margin of 4em auto, text-align: center, and position: relative.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta https-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Wazzup</title>
<meta name="description" content="Home Automation">
<meta name="author" content="Reuben Menezes">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* { -moz-box-sizing: border-box; -webkit-box-sizing: border-box; box-sizing: border-box; }
body {
    font-family: "HelveticaNeue-Light", "Helvetica Neue Light", "Helvetica Neue", "Roboto Light", "Segoe UI Web Light", "Segoe UI Light", "Segoe UI Web Regular",
    "Segoe UI", Helvetica, Arial, sans-serif;
    margin: 1em;
}
header {
    margin-bottom: 2em;
}
h1, h2, h3 {
    margin: 0;
    text-rendering: optimizeLegibility;
}
h1 {
    font-size: 2.4em;
    line-height: 1.4em;
    text-transform: lowercase;
    font-weight: normal;
    text-shadow: 1px 2px 0 #fff;
}
h2 {
    font-size: 1em;
    text-transform: uppercase;
    font-weight: normal;
    text-shadow: 1px 1px 0 #fff;
}
footer {
    position: absolute;
    bottom: 2em;
}
#main {
    margin: 4em auto;
    text-align: center;
    position: relative;
}
```

```

control_lights.html ●
49   </style>
50 </head>
51
52 <body>
53   <header>
54     <h1> Turning lights on and off</h1>
55     <h2>Thank you technology</h2>
56   </header>
57
58   <section id="main" role="main">
59     
60     <button id = "lighton" style="height:70px; width:100px; background-color: #4CAF50;">ON</button>
61     <button id = "lightoff" style="height:70px; width:100px; background-color: red;">OFF </button>
62
63   </section>
64
65
66   <footer>
67     Developed by Reuben Menezes
68   </footer>
69
70
71   <!-- including the latest PubNub JavaScript SDK -->
72   <script src="https://cdn.pubnub.com/pubnub-3.7.10.js"></script>
73   <script>
74     (function() {
75       // DOM
76       var button1 = document.getElementById("lighton");
77       // This is the channel name you are subscribing in Main.py
78       var channel = 'auto_control';
79       // Init - Get your own keys at admin.pubnub.com
80       var p = PUBNUB.init({
81         subscribe_key: 'sub-c-5c49a380-a5ca-11e7-a4bf-e61ccde79053',
82         publish_key: 'pub-c-e879116f-dd42-4f62-a85e-ff31aa416b77' ,
83         ssl:true
84       });
85       // Sending data
86       function senddata() {
87
88         p.publish({
89           channel : channel,
90           message : 'a'
91         });
92       }
93       // Click event
94       button1.addEventListener('click', senddata);
95     })();
96   </script>
97
98
99   <script>
100  (function() {
101    // DOM
102    var button1 = document.getElementById("lightoff");
103    // This is the channel name you are subscribing in Main.py
104    var channel = 'auto_control';
105    // Init - Get your own keys at admin.pubnub.com
106    var p = PUBNUB.init({
107      subscribe_key: 'sub-c-5c49a380-a5ca-11e7-a4bf-e61ccde79053',
108      publish_key: 'pub-c-e879116f-dd42-4f62-a85e-ff31aa416b77' ,
109      ssl:true
110    });
111    // Sending data
112    function senddata() {
113
114      p.publish({
115        channel : channel,
116        message : 'b'
117      });
118    }
119    // Click event
120    button1.addEventListener('click', senddata);
121  })();
122  </script>
123
124 </body>
125 </html>

```

Annex

Arduino Code :

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN      3      // Pin which is connected to the DHT sensor.

#define DHTTYPE      DHT11  // DHT 11

#define LDRpin A1 // analog pin where we connected the LDR and the resistor

//gas sensor
int smokeA0 = A0;
// Your threshold value
int sensorThres = 1100;

//pir sensor
int inputPin = 2;          // choose the input pin (for PIR sensor)
int pirState = LOW;        // we start, assuming no motion detected
int val = 0;

//ldr
int LDRValue = 0; // result of reading the analog pin
int pinOut = 4; //digital pin where the relay / led is connected
```

```
DHT_Unified dht(DHTPIN, DHTTYPE);
```

```
uint32_t delayMS;
```

```
String temp = "";
```

```
String ldr = "";
```

```
String gas = "";
```

```
String pir = "";
```

```
void setup() {
```

```
    // put your setup code here, to run once:
```

```
    //ldr
```

```
    while(!Serial)
```

```
    {}
```

```
    pinMode(pinOut, OUTPUT);
```

```
    digitalWrite(pinOut, LOW);
```

```
    //temp sensor
```

```
    sensor_t sensor;
```

```
    dht.temperature().getSensor(&sensor);
```

```
    dht.begin();
```

```

delayMS = sensor.min_delay / 1000;

//gas sensor
pinMode(smokeA0, INPUT);

//pir sensor
pinMode(inputPin, INPUT); // declare sensor as input
Serial.begin(9600);

}

void loop() {
    // put your main code here, to run repeatedly:

    //ldr
    LDRValue = analogRead(LDRpin); // read the value from the LDR

    if(LDRValue>50) //there is light then print 1 (change the value of 50 as required)
        ldr = "1 ; ";
    else
        ldr = "0 ; ";
    //Serial.println(1); //write to the serial monitor which also be read by python
    //Serial.println(0);
}

```

```

//temp sensor

sensors_event_t event;

dht.temperature().getEvent(&event);

Serial.print(event.temperature);

Serial.print(" ; ");




//pir sensor

val = digitalRead(inputPin); // read input value

if (val == HIGH) {           // check if the input is HIGH

    pir = "Motion detected ; ";

    //Serial.print("Motion detected ; ");

}

else if (val == LOW) {

    // we have just turned on

    pir = "No Motion detected ; ";

    //Serial.print("No motion detected ; ");

}






//gas sensor

int analogSensor = analogRead(smokeA0);

if (analogSensor > sensorThres)

```

```

{
    gas = "Gas detected ; ";
    //Serial.print("Gas detected ; ");
}
else
{
    gas = "No Gas detected ; ";
    //Serial.print("No Gas detected ; ");
}

String finalVal = "";
finalVal.concat(ldr);
finalVal.concat(temp);
finalVal.concat(pir);
finalVal.concat(gas);
finalVal.concat(" ");
Serial.println(finalVal);

delay(1000);

//led on and off
if(Serial.available())
{
    Serial.print("gonna print");
    char ch = Serial.read(); //get the value from python code
}

```

```

Serial.println(ch);

if(ch==97)
{
    digitalWrite(pinOut, HIGH); //switch on light if 'a' is sent by python code

}

else if (ch ==98)
    digitalWrite(pinOut, LOW); //switch off light if 'b' is sent by python code

}
}

```

Python Code For efficient real time transfer of data from the arduino to the webpage through the pubnub platform:

```

import sys

import os

from time import sleep

from pubnub.callbacks import SubscribeCallback

from pubnub.pubnub import PubNub, SubscribeListener

from pubnub.enums import PNStatusCategory

from pubnub.pnconfiguration import PNConfiguration

from pubnub.pubnub import PubNub

from pubnub.enums import PNOperationType, PNStatusCategory

import serial

```

```
import urllib2

try:
    urllib2.urlopen('http://google.com', timeout=1)
    internet = " Connected"
else:
    internet = " Not Connected"

except urllib2.URLError as err:
```

ser = serial.Serial('/dev/cu.usbmodem1411', 9600) #change this line to ser = serial.Serial('COM8', 9600) if you are using windows. Change the port number/ COM according to arduino ide

```
pnconfig = PNConfiguration()
pnconfig.subscribe_key = 'sub-c-5c49a380-a5ca-11e7-a4bf-e61ccde79053'
pnconfig.publish_key = 'pub-c-e879116f-dd42-4f62-a85e-ff31aa416b77'
pubnub = PubNub(pnconfig)
channel = 'auto_value'
```

```
pnconfig1 = PNConfiguration()
pubnub1 = PubNub(pnconfig)
```

```
#Publishes data to pubnub and also subscribe to the channel auto_control
class MySubscribeCallback1(SubscribeCallback):
```

```
def status(self, pubnub1, status):
    pass

def presence(self, pubnub1, presence):
    pass # handle incoming presence data

def message(self, pubnub1, message):
    print('will now write something')
    print(message.message)

    if (message.message == 'a'):
        ser.write('a')
    elif (message.message == 'b'):
        ser.write('b')
    print(message.message)

pubnub1.add_listener(MySubscribeCallback1())
pubnub1.subscribe().channels('auto_control').execute()

#Subscribe to the channel auto_value
class MySubscribeCallback(SubscribeCallback):
```

```

def status(self, pubnub, status):
    pass

def presence(self, pubnub, presence):
    pass # handle incoming presence data

def message(self, pubnub, message):
    print(message.message)

pubnub.add_listener(MySubscribeCallback())
pubnub.subscribe().channels(channel).execute()

#read data from arduino serial and publish it to pubnub

def reading():
    while(True):
        try:
            print('reading...')
            sleep(2)
            data = ser.readline() #read from the serial and save in a variable
            print(data)
            data = data.rstrip() #remove the \n from the string
            # if(data =='1' or data=='0'): #change accordingly, how you add all the
sensor data

```

```
pubnub.publish().channel(channel).message(data + internet).sync()  
except KeyboardInterrupt:  
    ser.close()
```

```
if __name__ == '__main__':
```

```
    reading()
```

Home Automation using Arduino

ORIGINALITY REPORT



PRIMARY SOURCES

- 1 Submitted to BITS, Pilani-Dubai
Student Paper 3%

Exclude quotes On

Exclude matches < 3%

Exclude bibliography On