

ภาคผนวก

```
1 #include <WiFi.h> //ทำการ Import Liberry เข้ามาเพื่อใช้ในการเชื่อม WIFI
2 #include <TridentTD_LineNotify.h> //ทำการ Import Liberry ที่เชื่อมต่อหรือส่งการแจ้งเตือนผ่าน LINE Notify
3 #include <HTTPClient.h> //ทำการ Import Liberry ที่เป็นการส่งข้อมูลไปยัง GoogleSheet ผ่าน HTTP
4 #define WIFI_STA_NAME "Yam" //กำหนด ID Wifi
5 #define WIFI_STA_PASS "96991617" // กำหนด PASSWORD Wifi
6 #define LINE_TOKEN "AP4z04QmTijUN4IAW2LkStBriUVm270Ne9NZPpmZDW" // กำหนด TOKEN เพื่อเชื่อมกับแชท Line ที่จะทำการแจ้งเตือน
7
8 String urlSensor = "https://script.google.com/macros/s/AKfycbzczHun4oPuokXJdSBjQ4d9Mmf0iKiUjVdUnWUsLkHrREFCKwKEjzKS26tsNPuLw1R_g/exec";
9 //กำหนด Url เพื่อสำหรับการเก็บข้อมูลไปยัง Google Sheet ผ่าน Appsheet
10
11
12
13 #define BLYNK_TEMPLATE_ID "TMPL6wc4xf86B" //กำหนด Blynk ID
14 #define BLYNK_TEMPLATE_NAME "Tmp" //กำหนด Blank Name
15
16
17
18 #define BLYNK_FIRMWARE_VERSION "0.1.0"
19 #define BLYNK_PRINT Serial
20 #define APP_DEBUG
21
22
23 #include "DHT.h"
24 #define DHTPIN 4 //กำหนดให้ Sensor DHT22 อยู่ที่ ขา 4
25 #define DHTTYPE DHT22 //กำหนดว่า DHT เป็น DHT22
26
27 DHT dht(DHTPIN, DHTTYPE);
28 float h = dht.readHumidity();
29 float t = dht.readTemperature();
30 float f = dht.readTemperature(true);
31 int temperature ;//กำหนดตัวแปร temperature
32 int humidity ; // กำหนดตัวแปร humidity
33
34
35 #define gas 34 //กำหนด Sensor Gas อยู่ที่ ขา 34
36 int Smoke_pm; //กำหนดตัวแปร Smoke_pm
37 int ppm ;
38
39 #define LEDR 5 //กำหนด LED อยู่ที่ ขา 5
40 #define SW 35 //กำหนด Button อยู่ที่ ขา 35
41 #define BUZ 19 //กำหนด Buzzer อยู่ที่ ขา 34
42 int Button; // กำหนดตัวแปร Button
43
44 int statusButton; // กำหนดตัวแปร statusButton
45 int swState ; // กำหนดตัวแปร swState
46 #include "BlynkEdgent.h" //ทำการ Import Liberry Blynk
47
48 int i=0, j=0, k=0; //กำหนดตัวแปรต่างๆ
49
50 #define Pump 21 //กำหนด ปั๊มน้ำ อยู่ที่ ขา 18
51 int PumpWater;
```

ขั้นตอนที่ 1 : คือการประกาศ ตัวแปรต่างๆ และเป็นการ Import Library เข้ามาใช้

```

54 void setup()
55 {
56   wifi_connected(); //เป็นการ Function ในการเชื่อมต่อ WiFi
57   Serial.begin(115200); //กำหนดอัตรา 115,200 สำหรับการสื่อสารซีเรียล
58   BlynkEdgent.begin(); //ใช้ในการเริ่มต้นการใช้งาน Blynk Edgent library
59   dht.begin(); // ใช้ในการเริ่มต้นการใช้งาน dht library
60
61
62   pinMode(gas, INPUT); //กำหนด pinMode gas เป็น INPUT
63   pinMode(LED_R, OUTPUT); //กำหนด pinMode LED_R เป็น OUTPUT
64   pinMode(SW, INPUT); //กำหนด pinMode SW เป็น INPUT
65   pinMode(BUZ, OUTPUT); //กำหนด pinMode BUZ เป็น OUTPUT
66   pinMode(Pump, OUTPUT); //กำหนด pinMode Pump เป็น OUTPUT
67
68 }

```

ขั้นตอนที่ 2 : เป็นการ Set Up และกำหนดขา Input และ Output

```

70 void loop() {
71   h = dht.readHumidity(); // ให้ตัวแปร h อ่านค่า Humidity
72   t = dht.readTemperature(); // ให้ตัวแปร t อ่านค่า temperature
73   SensorTemp(); //เป็นการใช้ Function SensorTemp มาใช้
74   Smoke(); //เป็นการใช้ Function Smoke มาใช้
75
76   if(Button == 1){ // ถ้าค่า Button == 1 ให้ทำงานในเงื่อนไข
77     LINE.notify("Smoke = " + String(Smoke_pm)); //ทำการแจ้งเตือน ข้อความ และค่าควัน ไปยัง Line
78     LINE.notify("Humidity = " + String(h)); //ทำการแจ้งเตือน ข้อความ และค่า Humidity ไปยัง Line
79     LINE.notify("Temperature = " + String(t)); //ทำการแจ้งเตือน ข้อความ และค่า Temperature ไปยัง Line
80
81     if(Smoke_pm > 250){ //แล้วในเมื่อค่าควัน มีค่ามากกว่า 150 จะทำในเงื่อนไข
82       LINE.notifySticker(4, 264); //ทำการแจ้งเตือนสติ๊กเกอร์
83       while(Smoke_pm > 250){ //และในเมื่อควัน มีค่ามากกว่า 150 จะทำในเงื่อนไข while จนกว่า ค่าควัน จะลดลง
84         SensorTemp(); //เป็นการใช้ Function Sensortemp มาใช้
85         Smoke(); //เป็นการใช้ Function Smoke มาใช้
86         //digitalWrite(BUZ, LOW); //ให้ BUZ เป็นค่า LOW เพราะว่าค่า LOW คือเปิด และ HIGH คือปิด เป็นแค่เฉพาะกับ Buzzder
87         digitalWrite(LED_R, HIGH); //ให้ LED_R เป็นค่า HIGH
88         digitalWrite(Pump, HIGH); //ให้ Pump เป็นค่า HIGH
89

```

ขั้นตอนที่ 3 : เป็นตอนต้นของฟังก์ชันในการทำงานของระบบหลัก ให้ตัวแปร t มาอ่านค่า Humidity และ t มาอ่านค่า Temperature อีกทั้งให้ตัวแปร Smoke_pm อ่านค่า analog ของ Sensor Gas และสั่งให้ Blynk เริ่มทำงาน อีกทั้งยังเรียกใช้

อีก 2 Function

```

76 if(Button == 1){ // ถ้าค่า Button == 1 ให้ทำงานในเงื่อนไข
77     LINE.notify("Smoke = " + String(Smoke_pm)); //ทำการแจ้งเตือน ข้อความ และค่าควัน ไปยัง Line
78     LINE.notify("Humidity = " + String(h)); //ทำการแจ้งเตือน ข้อความ และค่า Humidity ไปยัง Line
79     LINE.notify("Temperature = " + String(t)); //ทำการแจ้งเตือน ข้อความ และค่า Temperatureไปยัง Line
80
81     if(Smoke_pm > 250){ //แล้วในเมื่อค่าควัน มีค่ามากกว่า 150 จะทำในเงื่อนไข
82         LINE.notifySticker(4, 264); //ทำการแจ้งเตือนสติ๊กเกอร์
83         while(Smoke_pm > 250){ //และในเมื่อควัน มีค่ามากกว่า 150 จะทำในเงื่อนไข While จนกว่า ค่าควัน จะลดลง
84             SenserTemp(); //เป็นการใช้ Function Sensortemp มาใช้
85             Smoke(); //เป็นการใช้ Function Smoke มาใช้
86             //digitalWrite(BUZ, LOW); //ให้ BUZ เป็นค่า LOW เพราะค่า LOW คือเปิด และ HIGH คือปิด เป็นแค่เฉพาะกับ Buzzder
87             digitalWrite(LED_R, HIGH); //ให้ LED_R เป็นค่า HIGH
88             digitalWrite(Pump, HIGH); //ให้ Pump เป็นค่า HIGH
89
90             Blynk.virtualWrite(V4, HIGH); //ให้ Datastream V4 เป็น HIGH
91             Blynk.virtualWrite(V3, " Warning!!!"); //ให้ Datastream V3 เป็น เป็นข้อความดังกล่าวที่ใช้กับ LCD ในBlynk
92             Blynk.virtualWrite(V5, " !!Alert!!!"); //ให้ Datastream V5 เป็น เป็นข้อความดังกล่าวที่ใช้กับ LCD ในBlynk
93             if(Smoke_pm > 550){ //แล้วในเมื่อค่าควัน มีค่ามากกว่า 200 จะทำในเงื่อนไข
94                 LINE.notify("Smoke = " + String(Smoke_pm)); //ทำการแจ้งเตือน ข้อความ และค่าควัน ไปยัง Line
95                 LINE.notify("Humidity = " + String(h)); //ทำการแจ้งเตือน ข้อความ และค่า Humidity ไปยัง Line
96                 LINE.notify("Temperature = " + String(t)); //ทำการแจ้งเตือน ข้อความ และค่า Temperatureไปยัง Line
97                 LINE.notifySticker(4, 274); //ทำการแจ้งเตือนสติ๊กเกอร์
98
99                 while(Smoke_pm > 550){ //และในเมื่อควัน มีค่ามากกว่า 200 จะทำในเงื่อนไข While จนกว่า ค่าควัน จะลดลง
100                     SenserTemp(); //เป็นการใช้ Function Sensortemp มาใช้
101                     Smoke(); //เป็นการใช้ Function Smoke มาใช้
102                     digitalWrite(BUZ, LOW); //ให้ BUZ เป็นค่า LOW
103                     digitalWrite(LED_R, HIGH); //ให้ LED_R เป็นค่า HIGH
104                     digitalWrite(Pump, HIGH); //ให้ Pump เป็นค่า HIGH
105
106                     Blynk.virtualWrite(V4, HIGH); //ให้ Datastream V4 เป็น HIGH
107                     Blynk.virtualWrite(V3, " Burning!!!"); //ให้ Datastream V3 เป็น เป็นข้อความดังกล่าวที่ใช้กับ LCD ในBlynk
108                     Blynk.virtualWrite(V5, " !!!Run!!!"); //ให้ Datastream V5 เป็น เป็นข้อความดังกล่าวที่ใช้กับ LCD ในBlynk
109                 }
110             }
111         }
112     }
113 }
114 }
115 }
116 digitalWrite(Pump, LOW); //ให้ Pump เป็นค่า LOW
117

```

ขั้นตอนที่ 4 : ในเงื่อนไขนี้ เราจะเช็คค่า Button จากแอปพลิเคชัน Blynk เมื่อ ค่า Button ตรงตามเงื่อนไข

จะทำการแจ้งเตือนค่าต่างๆ ไปยัง Line โดยถ้าค่าควัน มีค่าเกินกว่าเกณฑ์ที่กำหนด เราจะเข้าเงื่อนไข while อีกครั้ง

เพราะว่าเนื่องจาก การกดปุ่ม ให้ตั้งค่าเป็นแบบ Push คือถ้ากด ค่าจะเป็น 1 และ ถ้าปล่อยจะเป็น 0 และทำการเรียกใช้ Function Smoke เพื่อมาใช้ ใน ลูป while อีกครั้ง เพื่อทำ Set Sensor ทั้ง 3 Sensor ให้ทำงาน อีกทั้งยัง set ค่า ใน

แอปพลิเคชัน Blynk

```

117
118 }else if( swState == HIGH ){ // ถ้า SW == 1 ให้ทำงานในเงื่อนไข
119     LINE.notify("Smoke = " + String(Smoke_pm)); //ทำการแจ้งเตือน ข้อความ และค่าควัน ไปยัง Line
120     LINE.notify("Humidity = " + String(h)); //ทำการแจ้งเตือน ข้อความ และค่า Humidity ไปยัง Line
121     LINE.notify("Temperature = " + String(t)); //ทำการแจ้งเตือน ข้อความ และค่า Temperatureไปยัง Line
122     if(Smoke_pm > 450){ //แล้วในเมื่อค่าควัน มีค่ามากกว่า 150 จะทำในเงื่อนไข
123         LINE.notifySticker(4, 264); //ทำการแจ้งเตือนสติ๊กเกอร์
124         while(Smoke_pm > 450){ //และในเมื่อควัน มีค่ามากกว่า 150 จะทำในเงื่อนไข While จนกว่า ค่าควัน จะลดลง
125             SenserTemp(); //เป็นการใช้ Function Sensortemp มาใช้
126             Smoke(); //เป็นการใช้ Function Smoke มาใช้
127             digitalWrite(BUZ, LOW); //ให้ BUZ เป็นค่า LOW
128             digitalWrite(LED1, HIGH); //ให้LED1 เป็นค่า HIGH
129             digitalWrite(Pump, HIGH); //ให้ Pump เป็นค่า HIGH
130
131             Blynk.virtualWrite(V4, HIGH); //ให้ Datastream V4 เป็น HIGH
132             Blynk.virtualWrite(V3, "Warning!!!"); //ให้ Datastream V3 เป็น เป็นข้อความดังกล่าวที่ใช้กับ LCD ในBlynk
133             Blynk.virtualWrite(V5, "!!Alert!!"); //ให้ Datastream V5 เป็น เป็นข้อความดังกล่าวที่ใช้กับ LCD ในBlynk
134             if(Smoke_pm > 550){ //แล้วในเมื่อค่าควัน มีค่ามากกว่า 150 จะทำในเงื่อนไข
135                 LINE.notify("Smoke = " + String(Smoke_pm)); //ทำการแจ้งเตือน ข้อความ และค่าควัน ไปยัง Line
136                 LINE.notify("Humidity = " + String(h)); //ทำการแจ้งเตือน ข้อความ และค่า Humidity ไปยัง Line
137                 LINE.notify("Temperature = " + String(t)); //ทำการแจ้งเตือน ข้อความ และค่า Temperatureไปยัง Line
138                 LINE.notifySticker(4, 274); //ทำการแจ้งเตือนสติ๊กเกอร์
139
140                 while(Smoke_pm > 550){ //และในเมื่อควัน มีค่ามากกว่า 200 จะทำในเงื่อนไข While จนกว่า ค่าควัน จะลดลง
141                     SenserTemp(); //เป็นการใช้ Function Sensortemp มาใช้
142                     Smoke(); //เป็นการใช้ Function Smoke มาใช้
143                     digitalWrite(BUZ, LOW); //ให้ BUZ เป็นค่า LOW
144                     digitalWrite(LED1, HIGH); //ให้LED1 เป็นค่า HIGH
145                     digitalWrite(Pump, HIGH); //ให้ Pump เป็นค่า HIGH
146
147                     Blynk.virtualWrite(V4, HIGH); //ให้ Datastream V4 เป็น HIGH
148                     Blynk.virtualWrite(V3, "Burnning!!!"); //ให้ Datastream V3 เป็น เป็นข้อความดังกล่าวที่ใช้กับ LCD ในBlynk
149                     Blynk.virtualWrite(V5, "!!!Run!!!"); //ให้ Datastream V5 เป็น เป็นข้อความดังกล่าวที่ใช้กับ LCD ในBlynk
150                 }
151             }
152         }
153     }
154
155 }
156 digitalWrite(Pump, LOW); //ให้ Pump เป็นค่า LOW
157

```

ขั้นตอนที่ 5 : ในเงื่อนไขนี้ เราจะเช็คค่า Button ถ้าค่า ตรงตามเงื่อนไข จะทำการแจ้งเตือนค่าต่างๆ ไปยัง Line โดยถ้าค่าควัน มีค่าเกินกว่าเกณฑ์ที่กำหนด เราจะเข้าเงื่อนไข while อีกครั้งเพราะว่าเนื่องจาก การกดปุ่ม ให้ตั้งค่าเป็นแบบ Push คือถ้ากด ค่าจะเป็น 1 และ ถ้าปล่อยจะเป็น 0 และทำการเรียกใช้ Function Smoke เพื่อมาใช้ ใน ลูป while อีกครั้ง เพื่อทำ Set Sensor ทั้ง 3 Sensor ให้ทำงาน อีกทั้งยัง set ค่า ในแอปพลิเคชัน Blynk

```

157
158 }if(Smoke_pm > 650){ //แล้วในเมื่อค่าควัน มีค่ามากกว่า 150 จะทำในเงื่อนไข
159
160     digitalWrite(BUZ, LOW); //ให้ BUZ เป็นค่า LOW
161     digitalWrite(LED1, HIGH); //ให้ LED1 เป็นค่า HIGH
162     digitalWrite(Pump, HIGH); //ให้ Pump เป็นค่า HIGH
163     int i = 0;
164
165     if( i % 20 == 0){
166         LINE.notify("Smoke = " + String(Smoke_pm)); //ทำการแจ้งเตือน ข้อความ และค่าควัน ไปยัง Line
167         LINE.notify("Humidity = " + String(h)); //ทำการแจ้งเตือน ข้อความ และค่า Humidity ไปยัง Line
168         LINE.notify("Temperature = " + String(t)); //ทำการแจ้งเตือน ข้อความ และค่า Temperature ไปยัง Line
169     }
170     i++;
171     Blynk.virtualWrite(V4, HIGH); //ให้ Datastream V4 เป็น HIGH
172     Blynk.virtualWrite(V3, " Warning!!!"); //ให้ Datastream V3 เป็น เป็นข้อความดังกล่าวที่ใช่กับ LCD ในBlynk
173     Blynk.virtualWrite(V5, " !!Alert!!"); //ให้ Datastream V5 เป็น เป็นข้อความดังกล่าวที่ใช่กับ LCD ในBlynk
174     LINE.notifySticker(4, 264); //ทำการแจ้งเตือนสติ๊กเกอร์
175 }
176
177 }if(Smoke_pm > 750 ){ //แล้วในเมื่อค่าควัน มีค่ามากกว่า 200 จะทำในเงื่อนไข
178
179     LINE.notifySticker(4, 274); //ทำการแจ้งเตือนสติ๊กเกอร์
180     digitalWrite(BUZ, LOW); //ให้ BUZ เป็นค่า LOW
181     digitalWrite(LED1, HIGH); //ให้ LED1 เป็นค่า HIGH
182     digitalWrite(Pump, HIGH); //ให้ Pump เป็นค่า HIGH
183
184     Blynk.virtualWrite(V4, HIGH); //ให้ Datastream V4 เป็น HIGH
185     Blynk.virtualWrite(V3, " Burning!!!"); //ให้ Datastream V3 เป็น เป็นข้อความดังกล่าวที่ใช่กับ LCD ในBlynk
186     Blynk.virtualWrite(V5, " !!!Run!!!"); //ให้ Datastream V5 เป็น เป็นข้อความดังกล่าวที่ใช่กับ LCD ในBlynk

```

ขั้นตอนที่ 6 : การทำงานในส่วนนี้ การทำงาน โดยถ้าค่าควัน มีค่าเกินกว่าเกณฑ์ที่กำหนด เราจะเข้าสู่การทำงานในเงื่อนไข โดยทำการ Set Sensor ทั้ง 3 Sensor ให้ทำงาน และจะทำการแจ้งเตือนค่าต่างๆ ไปยัง Line อีกทั้ง ยังทำการ Set ค่าใน แอปพลิเคชัน Blynk

```

187     digitalWrite(Pump, LOW); //ให้ Pump เป็นค่า HIGH
188     digitalWrite(LED1, LOW); //ให้ Pump เป็นค่า HIGH
189     digitalWrite(BUZ, LOW); //ให้ BUZ เป็นค่า LOW
190
191     Blynk.virtualWrite(V4, LOW); //ให้ Datastream V4 เป็น LOW
192     Blynk.virtualWrite(V3, "    Peaceful"); //ให้ Datastream V3 เป็น เป็นข้อความดังกล่าวที่ใส่กับ LCD ในBlynk
193     Blynk.virtualWrite(V5, " "); //ให้ Datastream V5 เป็น เป็นค่าว่าง
194
195     BlynkEdgent.run();
196
197
198     SendButtonTOSheet(); //ดึง Function SendButtonTOSheet มาใช้
199     SendTempTOSheet(); //ดึง Function SendTempTOSheet มาใช้
200     SendGasTOSheet(); //ดึง Function SendGasTOSheet มาใช้
201     SendPumpWaterTOSheet(); //ดึง Function SendPumpWaterTOSheet มาใช้
202     delay(2000);
203
204 }
205

```

ขั้นตอนที่ 7: การทำงานนี้คือการที่ ถ้าค่าควันที่รับมาจาก Sensor Gas อยู่ในค่าปกติ จะทำการกำหนด จะไม่เกิดการ
ทำงานใดๆ และทำการ set ค่าใน แอปพลิเคชัน Blynk อีกทั้งเรายังมีการ เรียกใช้ Function ทั้ง 4 มาใช้

```

206 void SenserTemp(){ //เป็น Function ของ Sensor DHT
207
208
209     if (isnan(h) || isnan(t)) { //ถ้าค่าไม่มีค่า
210
211         Serial.println("Failed to read from DHT sensor!"); //จะแสดงข้อความดังกล่าว
212         return; //ทำงาน Return
213
214     }
215
216     Serial.print("Humidity: "); //จะแสดงข้อความดังกล่าว
217     Serial.print(h); //จะแสดงค่า h
218     Serial.print(" %\t");
219     Serial.print("\nTemperature: "); //จะแสดงข้อความดังกล่าว
220     Serial.print(t); //จะแสดงค่า t
221     Serial.println(" *C "); //จะแสดงข้อความดังกล่าว
222     Blynk.virtualWrite(V1, h); //ให้ Datastream V1 มาค่าเท่ากับ ค่าของตัวแปร h
223     Blynk.virtualWrite(V2, t); //ให้ Datastream V2 มาค่าเท่ากับ ค่าของตัวแปร t
224     delay(1000); //ให้ มีDelay 1 วิ
225
226 }
227

```

ขั้นตอนที่ 8 : เป็น Function ในการทำงานของ Sensor DHT22 อย่างแรก เราจะทำการเช็คก่อนว่า ตัวแปร t และตัวแปร h มีการรับค่าหรือไม่ ถ้าไม่ จะทำการแสดงข้อความดังกล่าว ให้ผู้ใช้ได้ทราบ อีกทั้งยังแสดงค่าไปยัง Data Stream V1, V2 และให้ Delay ในการทำงาน 1 วิ

```

229 void Smoke(){ //เป็น Function ของ Sensor Gas
230
231     Smoke_pm = analogRead(gas); //ให้ ตัวแปร Smoke_pm อ่านค่าแบบ analog จาก gas
232     Blynk.virtualWrite(V9,Smoke_pm); //ให้ Datastream V1 มาค่าเท่ากับ ค่าของตัวแปร Smoke_pm
233     Serial.print("\nSmoke_Pm = "); //จะแสดงข้อความดังกล่าว
234     Serial.println(Smoke_pm); // พร้อมกับค่า Smoke_pm
235
236
237
238 }

```

ขั้นตอนที่ 9 : เป็น Function การทำงานของ Sensor Gas กำหนด ตัวแปร Smoke_pm เพื่อมาใช้ในการอ่านค่า analog และทำการแสดงค่าไปยัง Data Stream V9 ในแอปพลิเคชัน Blynk ต่อจากนั้นในขณะเดียวกันเราจะ print ค่า Smoke ใน

Serial Monitor

```

237 void wifi_connected(){ //เป็น Function ในการเชื่อม WiFi
238
239     WiFi.mode(WIFI_STA); //
240     WiFi.begin(WIFI_STA_NAME, WIFI_STA_PASS); // เริ่มต้นการเชื่อมต่อ Wi-Fi
241     Serial.println(LINE.getVersion()); //เรียกฟังก์ชัน getVersion() ของไลบรารี LINE
242     Serial.println("WiFi connected"); //แสดงข้อความดังกล่าว
243     Serial.print("IP address: "); //แสดงข้อความดังกล่าว
244
245     while (WiFi.status() != WL_CONNECTED) { // รอการเชื่อมต่อ Wi-Fi
246
247         delay(500); // ให้ Delay 0.5 วิ
248         Serial.print("."); //แสดงข้อความดังกล่าว
249         LINE.setToken(LINE_TOKEN); //ทำการSetToken ที่เรากำหนดไว้
250
251     }
252 }

```

ขั้นตอนที่ 10 : เป็น Function การทำงาน ของการเชื่อมต่อ Wi - Fi

```

254 BLYNK_WRITE(V0){ //เป็น Method ที่เอาไว้รับค่าจากการกดปุ่ม จาก แอปพลิเคชัน Blynk
255
256     Button = param.asInt(); //ให้ตัวแปร Button เก็บค่า param จาก Datastream V0
257
258 }

```

ขั้นตอนที่ 11 : เป็นการ Method ที่เอาไว้รับค่าจาก แอปพลิเคชัน Blynk ที่ Data Stream V0

ให้ตัวแปร Button มารับค่าจากการกด


```

259 void SendTempTOSheet(){ // เป็น Function SendTempTOSheet
260 temperature = t; //ให้ตัวแปร temperature เท่ากับค่า t
261 humidity = h; //ให้ตัวแปร humidity เท่ากับค่า h
262
263
264 j++; //ให้ค่า j+1
265 if(j==60){ //ถ้า j == 60ให้ทำในเงื่อนไข
266     String urlSensors = urlSensor + "?temp=" + temperature + "&humidity=" + humidity; //ให้ตัวแปรhttpCodetemp และตัวแปร humidity ส่งค่าร้องข้อมูล GET
267
268     HTTPClient http; //ใช้เพื่อส่งค่าร้องข้อมูล HTTP ไปยัง URL ของเว็บเซิร์ฟเวอร์
269     http.begin(urlSensors.c_str()); //ใช้สำหรับเริ่มต้นค่าร้องข้อมูล HTTP
270     http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS); //ใช้ในการกำหนดว่าค่าร้องข้อมูล HTTP ควรทำการติดตามการเปลี่ยนเส้นทาง (redirects) อย่างไร เช็คอัปเดตในมิด
271     int httpCodetemp = http.GET(); //ให้ตัวแปรhttpCodetemp ส่งค่าร้องข้อมูล GET
272
273
274     if (httpCodetemp == 200 || httpCodetemp == 201) { //ถ้าhttpCodetemp มีค่า 200 หรือ 201 แสดงว่าส่งค่าเข้ามิดได้
275         Serial.println("Temp ok"); //ให้แสดงขึ้นmonitor ว่า Temp ok
276     } else { //แต่ถ้าไม่ใช่
277         Serial.println("TempFail: " + String(httpCodetemp)); //ในกรณีที่ TempFail: ที่ค่าhttpCodetempเท่าไร
278     }
279     j=0; //เมื่อเข้าเงื่อนไขแล้ว ก่อนออกให้เช็คค่า j เป็น 0
280 }
281 }

```

```

283 void SendGasTOSheet(){ // เป็น Function SendGasTOSheet
284 ppm = Smoke_pm; //ให้ตัวแปร ppm เท่ากับค่า Smoke_pm
285 k++; //ให้k+1
286 if(k==60){ //ถ้าค่าk=60แสดงว่าวนครบรอบรอบ หรือเข้าที่แล้ว ให้เข้าเงื่อนไข
287     String urlSensors = urlSensor + "?ppm=" + ppm; //ประกาศตัวแปรurlSensors เพื่อรับค่าppm ไม่เกินที่urlSensor
288
289     HTTPClient http; //ใช้เพื่อส่งค่าร้องข้อมูล HTTP ไปยัง URL ของเว็บเซิร์ฟเวอร์
290     http.begin(urlSensors.c_str()); //ใช้สำหรับเริ่มต้นค่าร้องข้อมูล HTTP
291     http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS); //ใช้ในการกำหนดว่าค่าร้องข้อมูล HTTP ควรทำการติดตามการเปลี่ยนเส้นทาง (redirects) อย่างไร เช็คอัปเดตในมิด
292     int httpCodeGas = http.GET(); //ให้ตัวแปรhttpCodeGas ส่งค่าร้องข้อมูล GET
293
294
295     if (httpCodeGas == 200 || httpCodeGas == 201) { //ถ้าhttpCodeGas มีค่า 200 หรือ 201 แสดงว่าส่งค่าเข้ามิดได้
296         Serial.println("Ok Gas"); //ให้แสดงขึ้นmonitor ว่า Ok Gas
297     } else { //แต่ถ้าไม่ใช่
298         Serial.println("Fail: " + String(httpCodeGas)); //ในกรณีที่ Fail: ที่ค่าhttpCodeGasเท่าไร
299     }
300     k=0; //เมื่อเข้าเงื่อนไขแล้ว ก่อนออกให้เช็คค่า k เป็น 0
301 }
302 }
303 }

```

```

308 void SendButtonTOSheet(){ // เป็น Function SendButtonTOSheet
309 statusButton = SW; // ให้ตัวแปร statusButton เท่ากับค่า SW
310 k++; //ให้k+1
311 if(k==60){ //ถ้าค่าk=60แสดงว่าวนครบรอบรอบ หรือเข้าที่แล้ว ให้เข้าเงื่อนไข
312     String urlSensors = urlSensor + "?Button=" + statusButton; //ประกาศตัวแปรurlSensors เพื่อรับค่าButton ไม่เกินที่urlSensor
313
314     HTTPClient http; //ใช้เพื่อส่งค่าร้องข้อมูล HTTP ไปยัง URL ของเว็บเซิร์ฟเวอร์
315     http.begin(urlSensors.c_str()); //ใช้สำหรับเริ่มต้นค่าร้องข้อมูล HTTP
316     http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS); //ใช้ในการกำหนดว่าค่าร้องข้อมูล HTTP ควรทำการติดตามการเปลี่ยนเส้นทาง (redirects) อย่างไร เช็คอัปเดตในมิด
317     int httpCodeBut = http.GET(); //ให้ตัวแปรhttpCodeBut ส่งค่าร้องข้อมูล GET
318
319
320     if (httpCodeBut == 200 || httpCodeBut == 201) { //ถ้าhttpCodeBut มีค่า 200 หรือ 201 แสดงว่าส่งค่าเข้ามิดได้
321         Serial.println("Ok But"); //ให้แสดงขึ้นmonitor ว่า Ok But
322     } else { //แต่ถ้าไม่ใช่
323         Serial.println("Fail: " + String(httpCodeBut)); //ในกรณีที่ Fail: ที่ค่าhttpCodeButเท่าไร
324     }
325     k=0; //เมื่อเข้าเงื่อนไขแล้ว ก่อนออกให้เช็คค่า k เป็น 0
326 }
327 }

```

```

327 void SendPumpWaterTOSheet(){ // เป็น Function SendPumpWaterTOSheet
328 PumpWater = Pump; //ให้ตัวแปร PumpWater เท่ากับค่า Pump
329 i++; //ให้ i เพิ่มขึ้นค่า เมื่อมาถึงส่วนนี้
330 if(i==60){ //ถ้าค่าi=60แสดงว่าวนครบรอบรอบ หรือเข้าที่แล้ว ให้เข้าเงื่อนไข
331     String urlSensors = urlSensor + "?pump=" + PumpWater; //ประกาศตัวแปรurlSensors เพื่อรับค่าPumpWater ไม่เกินที่urlSensor
332
333     HTTPClient http; //ใช้เพื่อส่งค่าร้องข้อมูล HTTP ไปยัง URL ของเว็บเซิร์ฟเวอร์
334     http.begin(urlSensors.c_str()); //ใช้สำหรับเริ่มต้นค่าร้องข้อมูล HTTP
335     http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS); //ใช้ในการกำหนดว่าค่าร้องข้อมูล HTTP ควรทำการติดตามการเปลี่ยนเส้นทาง (redirects) อย่างไร เช็คอัปเดตในมิด
336     int httpCodepump = http.GET(); //ให้ตัวแปรhttpCodepump ส่งค่าร้องข้อมูล GET
337
338     if (httpCodepump == 200 || httpCodepump == 201) { //ถ้าhttpCodepump มีค่า 200 หรือ 201 แสดงว่าส่งค่าเข้ามิดได้
339         Serial.println("Ok But"); //ให้แสดงขึ้นmonitor ว่า Ok But
340     } else { //แต่ถ้าไม่ใช่
341         Serial.println("Fail: " + String(httpCodepump)); //ในกรณีที่ Fail: ที่ค่าhttpCodepumpเท่าไร
342     }
343     i=0; //เมื่อเข้าเงื่อนไขแล้ว ก่อนออกให้เช็คค่า i เป็น 0
344 }
345 }
346 }

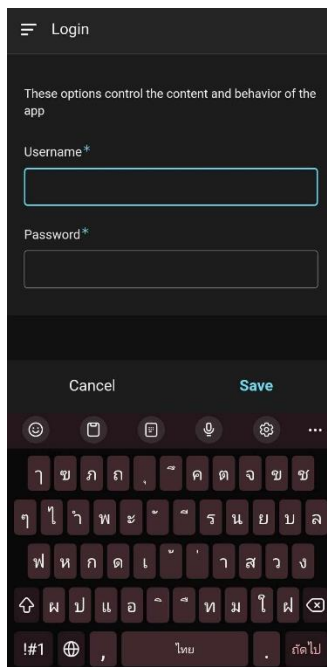
```

ขั้นตอนที่ 12 : เป็นการทำงานของการทำงานส่งข้อมูล ไปเก็บไว้ใน Google Sheet

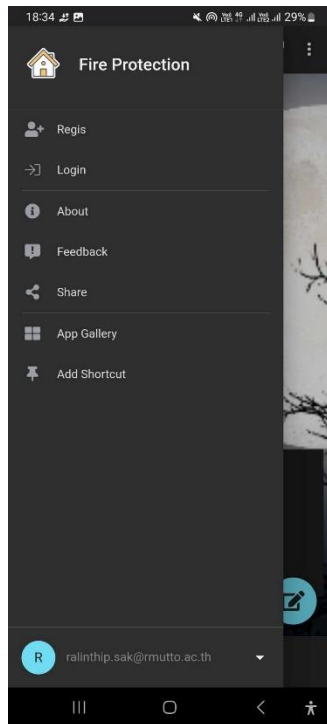
APP SHEET



เริ่มต้นระบบก็จะทำการเข้าหน้าโหลดเพื่อไปยังหน้า Log in

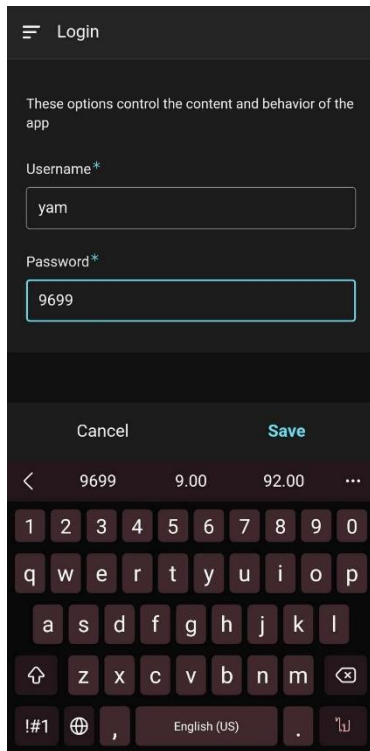


เมื่อเข้ามาจะเป็นหน้าล็อกอิน



หากยังไม่ได้ Login จะมีฟังก์ชันการทำงาน2อย่าง คือ Regis และ Login

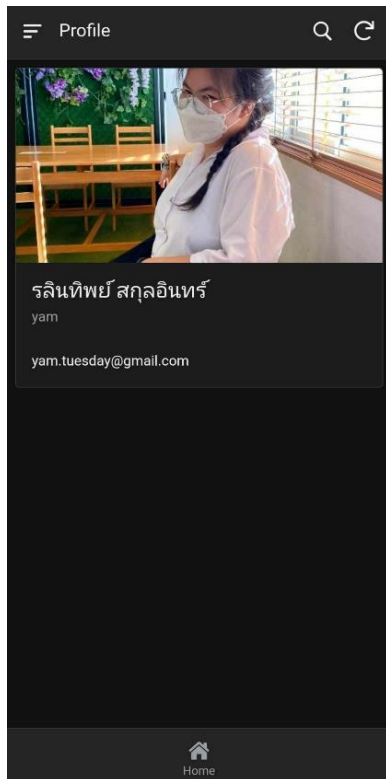
กรณีที่ผู้ใช้อย่างไม่เคยสมัครเข้าใช้งาน
สามารถทำการกดRegis แล้วจะดึงผู้ใช้เข้ามา
ในหน้าจอของ Regis เพื่อใส่ข้อมูลทำการสมัคร



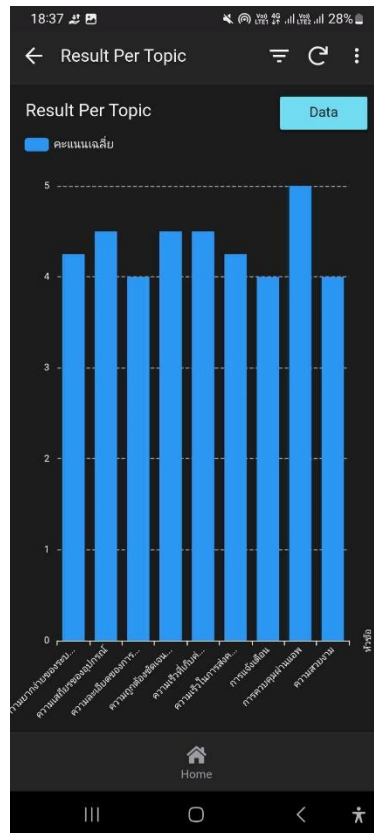
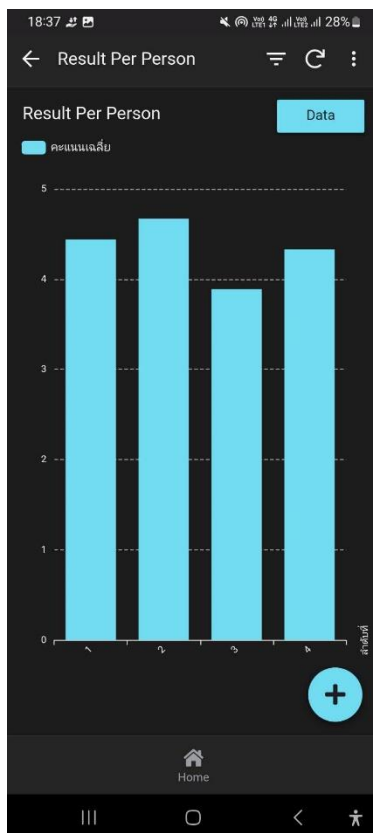
ดังรูปนี้จะเป็นการเข้ารหัสโดยผู้ที่มีสถานะเป็น Admin



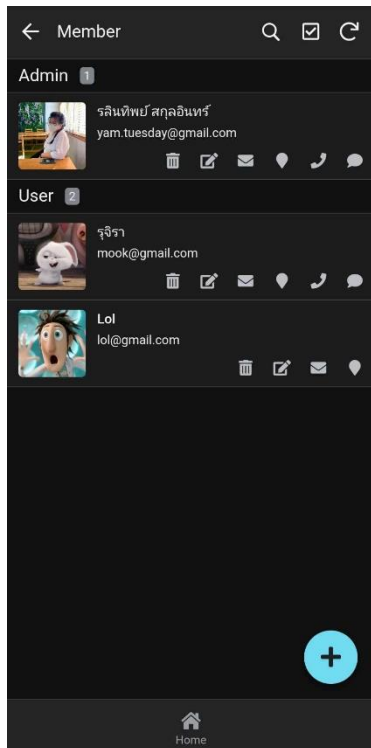
หลังจากนั้นจะเข้าสู่หน้า Home ในหน้านี้จะเห็นเป็น ตัวเลือก 5 ตัวเลือก เพราะว่าผู้ที่มีสถานะเป็น Admin



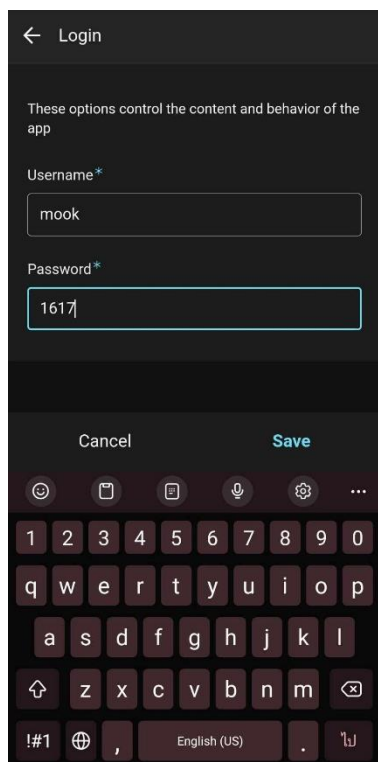
ในส่วนของหน้านี้จะแสดงเป็นโปรไฟล์ของผู้ที่เข้าใช้งาน



ผู้ใช้งานที่มีสถานะเป็น Admin จะสามารถ
เข้ามาเช็คในส่วนของหน้า Result Per Person
และ Result Per Topic ได้



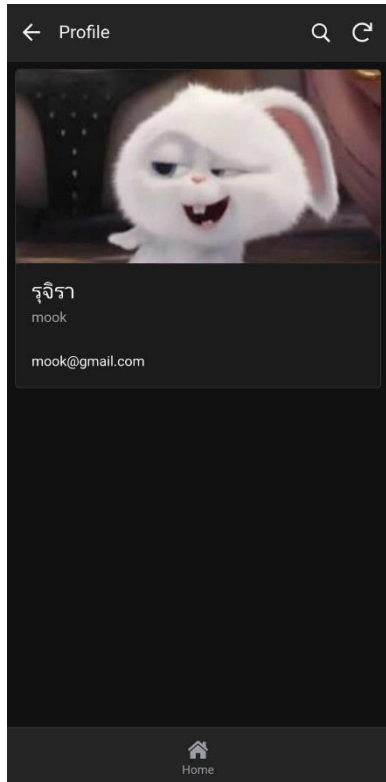
ในส่วนของหน้านี้จะมีแค่ผู้ที่มีสถานะเป็น Admin ที่จะสามารถเข้ามาตรวจเช็ค User อื่นๆได้และยังสามารถแก้ไขข้อมูล ลบข้อมูลของ User และยังสามารถสร้าง User เพิ่มขึ้นมาได้



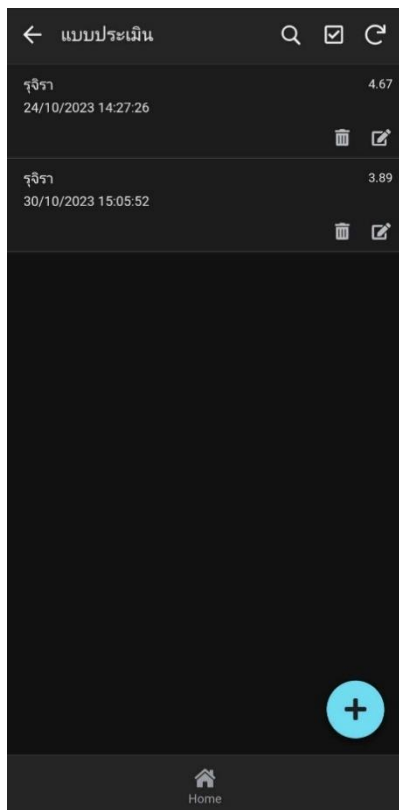
ต่อมาในส่วนของหน้านี้ชื่อผู้ใช้นี้จะมีสถานะเป็นแค่ Member



ในหน้านี้ผู้ใช้ที่มีสถานะเป็น Member จะเห็นแค่ 2 ตัวเลือกซึ่ง Member จะทำได้แค่การเปลี่ยนแปลงข้อมูลของตนและดูแบบประเมินได้แค่ของตนเองเท่านั้น



ในส่วนของหน้านี้จะแสดงเป็นโปรไฟล์ของผู้ที่เข้าใช้งาน



ในส่วนของหน้านี้จะเป็นหน้าแบบประเมินที่ผู้ใช้จะเห็นของตนเองว่า
ทำไปกี่ครั้ง



ในส่วนของหน้านี้จะเป็นแบบประเมินที่ผู้ใช้สามารถเข้ามาตรวจสอบหรือเช็ค
รายละเอียดได้แต่จะไม่สามารถแก้ไขข้อมูลต่างๆได้

