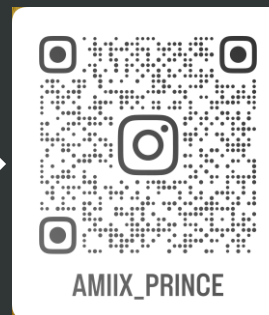




Annexe

Collection des
images & video
du Smart Detector

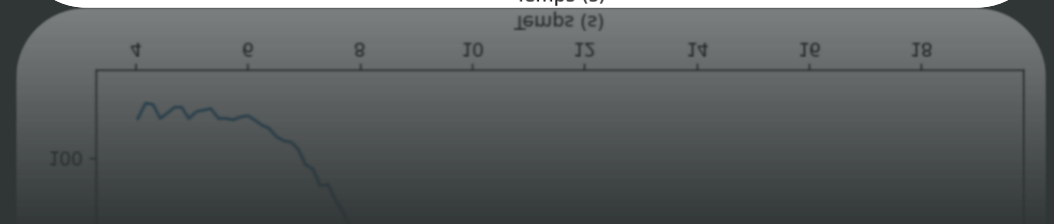
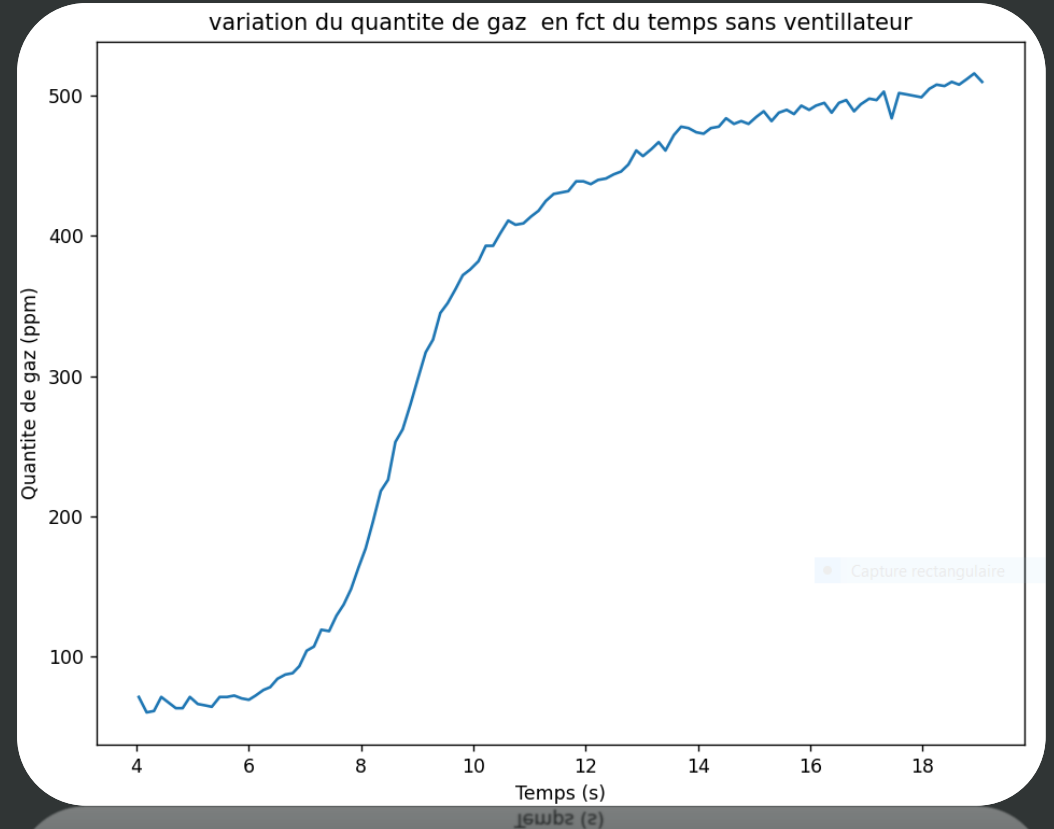


https://www.instagram.com/amiix_prince/

```

1 import datetime
2 import matplotlib.pyplot as plt
3 valeurs_temps = ['18:29:39.06', '18:29:38.92', '18:29:38.79', '18:29:38.65',
4 '18:29:38.52',
5 '18:29:38.39', '18:29:38.25', '18:29:38.12', '18:29:37.98', '18:29:37.85',
6 '18:29:37.72', '18:29:37.58', '18:29:37.45', '18:29:37.31', '18:29:37.18',
7 '18:29:37.05', '18:29:36.90', '18:29:36.78', '18:29:36.64', '18:29:36.51',
8 '18:29:36.38', '18:29:36.25', '18:29:36.10', '18:29:35.98', '18:29:35.84',
9 '18:29:35.71', '18:29:35.58', '18:29:35.44', '18:29:35.31', '18:29:35.17',
10 '18:29:35.04', '18:29:34.90', '18:29:34.77', '18:29:34.64', '18:29:34.50',
11 '18:29:34.37', '18:29:34.23', '18:29:34.10', '18:29:33.97', '18:29:33.83',
12 '18:29:33.70', '18:29:33.57', '18:29:33.42', '18:29:33.30', '18:29:33.17',
13 '18:29:33.02', '18:29:32.90', '18:29:32.76', '18:29:32.63', '18:29:32.50',
14 '18:29:32.36', '18:29:32.22', '18:29:32.09', '18:29:31.96', '18:29:31.83',
15 '18:29:31.69', '18:29:31.56', '18:29:31.43', '18:29:31.29', '18:29:31.16',
16 '18:29:31.03', '18:29:30.89', '18:29:30.75', '18:29:30.62', '18:29:30.48',
17 '18:29:30.35', '18:29:30.22', '18:29:30.09', '18:29:29.94', '18:29:29.81',
18 '18:29:29.68', '18:29:29.54', '18:29:29.41', '18:29:29.28', '18:29:29.15',
19 '18:29:29.01', '18:29:28.88', '18:29:28.74', '18:29:28.61', '18:29:28.48',
20 '18:29:28.35', '18:29:28.21', '18:29:28.08', '18:29:27.95', '18:29:27.82',
21 '18:29:27.69', '18:29:27.56', '18:29:27.43', '18:29:27.29', '18:29:27.16',
22 '18:29:27.03', '18:29:26.90', '18:29:26.78', '18:29:26.65', '18:29:26.51',
23 '18:29:26.38', '18:29:26.26', '18:29:26.12', '18:29:26.00', '18:29:25.87',
24 '18:29:25.74', '18:29:25.61', '18:29:25.48', '18:29:25.34', '18:29:25.22',
25 '18:29:25.09', '18:29:24.95', '18:29:24.82', '18:29:24.70', '18:29:24.57',
26 '18:29:24.44', '18:29:24.31', '18:29:24.18', '18:29:24.04']
27 valeurs = [
28 510, 516, 512, 508, 510, 507, 508, 505, 499, 500, 501, 502, 484, 503, 497, 498,
29 494, 489, 497, 495, 488, 495, 493, 490, 493, 487, 490, 488, 482, 489, 485, 480,
30 482, 480, 484, 478, 477, 473, 474, 477, 478, 472, 461, 467, 462, 457, 461, 451,
31 446, 444, 441, 440, 437, 439, 439, 432, 431, 430, 425, 418, 414, 409, 408, 411,
32 402, 393, 393, 382, 376, 372, 362, 352, 345, 326, 317, 298, 280, 262, 253, 226,
33 218, 196, 177, 163, 148, 137, 129, 118, 119, 107, 104, 93, 88, 87, 84, 78, 76,
34 72, 69, 70, 72, 71, 71, 64, 65, 66, 71, 63, 63, 67, 71, 61, 60, 71]
35
36 valeurs_temps_secondes = []
37 for valeur_temps in valeurs_temps:
38     temps = datetime.datetime.strptime(valeur_temps, '%H:%M:%S.%f')
39     temps_secondes = temps.second + temps.microsecond / 1e6 - 20
40     valeurs_temps_secondes.append(temps_secondes)
41
42 valeurs_temps_secondes_inverse = valeurs_temps_secondes[::-1] # Inversion de la
43 liste temps
44 valeurs_inverse = valeurs[::-1] # Inversion de la liste temps
45
46 plt.plot(valeurs_temps_secondes_inverse, valeurs_inverse,)
47 plt.xlabel('Temps (s)')
48 plt.ylabel('Quantite de gaz (ppm)')
49 plt.title('variation du quantite de gaz en fct du temps avec ventilateur')
50 plt.show()

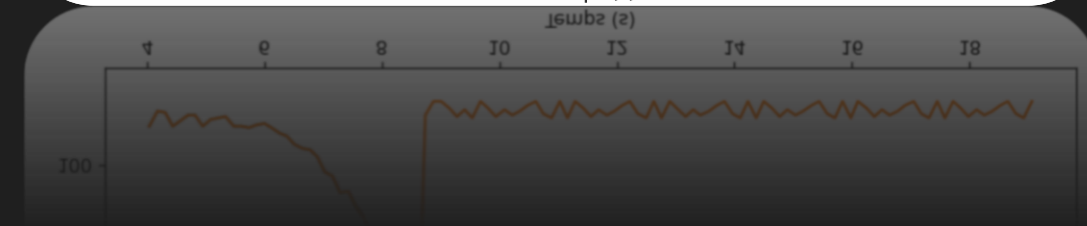
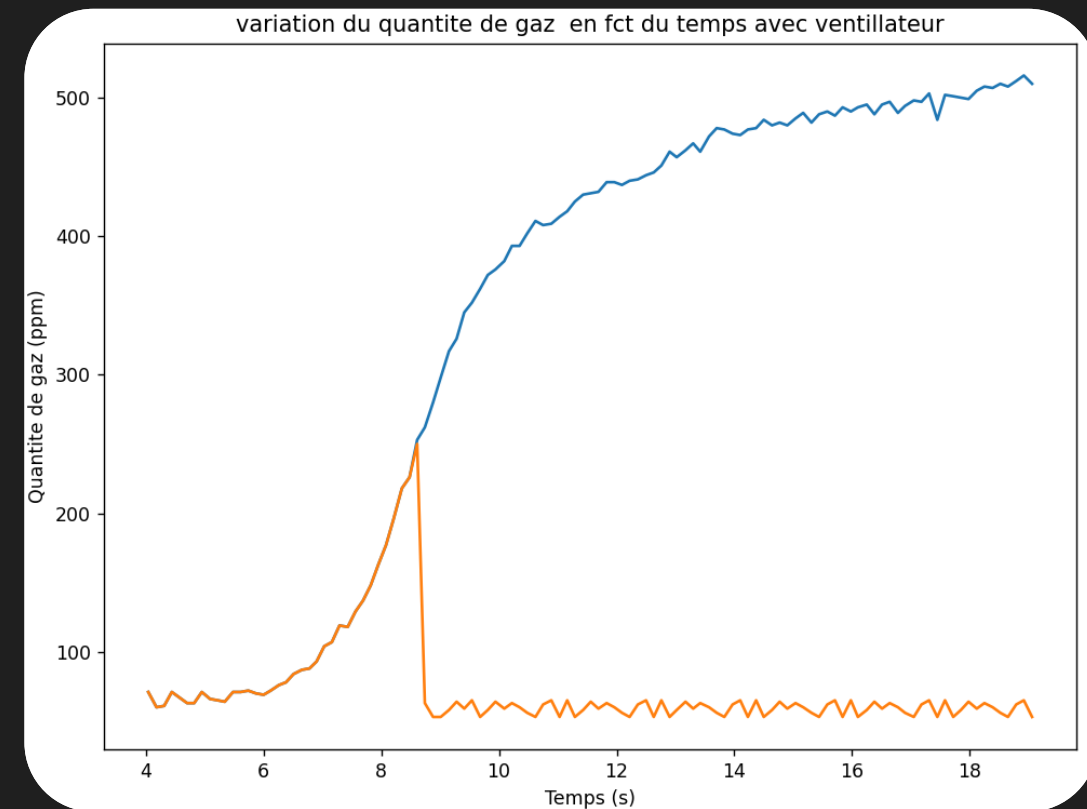
```



```

1 import datetime
2 import matplotlib.pyplot as plt
3 valeurs_temps = ['18:29:39.06', '18:29:38.92', '18:29:38.79', '18:29:38.65', '18:29:38.52',
4   '18:29:38.39', '18:29:38.25', '18:29:38.12', '18:29:37.98', '18:29:37.85',
5   '18:29:37.72', '18:29:37.58', '18:29:37.45', '18:29:37.31', '18:29:37.18',
6   '18:29:37.05', '18:29:36.90', '18:29:36.78', '18:29:36.64', '18:29:36.51',
7   '18:29:36.38', '18:29:36.25', '18:29:36.10', '18:29:35.98', '18:29:35.84',
8   '18:29:35.71', '18:29:35.58', '18:29:35.44', '18:29:35.31', '18:29:35.17',
9   '18:29:35.04', '18:29:34.90', '18:29:34.77', '18:29:34.64', '18:29:34.50',
10  '18:29:34.37', '18:29:34.23', '18:29:34.10', '18:29:33.97', '18:29:33.83',
11  '18:29:33.70', '18:29:33.57', '18:29:33.42', '18:29:33.30', '18:29:33.17',
12  '18:29:33.02', '18:29:32.90', '18:29:32.76', '18:29:32.63', '18:29:32.50',
13  '18:29:32.36', '18:29:32.22', '18:29:32.09', '18:29:31.96', '18:29:31.83',
14  '18:29:31.69', '18:29:31.56', '18:29:31.43', '18:29:31.29', '18:29:31.16',
15  '18:29:31.03', '18:29:30.89', '18:29:30.75', '18:29:30.62', '18:29:30.48',
16  '18:29:30.35', '18:29:30.22', '18:29:30.09', '18:29:29.94', '18:29:29.81',
17  '18:29:29.68', '18:29:29.54', '18:29:29.41', '18:29:29.28', '18:29:29.15',
18  '18:29:29.01', '18:29:28.88', '18:29:28.74', '18:29:28.61', '18:29:28.48',
19  '18:29:28.35', '18:29:28.21', '18:29:28.08', '18:29:27.95', '18:29:27.82',
20  '18:29:27.69', '18:29:27.56', '18:29:27.43', '18:29:27.29', '18:29:27.16',
21  '18:29:27.03', '18:29:26.90', '18:29:26.78', '18:29:26.65', '18:29:26.51',
22  '18:29:26.38', '18:29:26.26', '18:29:26.12', '18:29:26.00', '18:29:25.87',
23  '18:29:25.74', '18:29:25.61', '18:29:25.48', '18:29:25.34', '18:29:25.22',
24  '18:29:25.09', '18:29:24.95', '18:29:24.82', '18:29:24.70', '18:29:24.57',
25  '18:29:24.44', '18:29:24.31', '18:29:24.18', '18:29:24.04']
26 valeurs = [
27   510, 516, 512, 508, 510, 507, 508, 505, 499, 500, 501, 502, 484, 503, 497, 498,
28   494, 489, 497, 495, 488, 495, 493, 490, 493, 487, 490, 488, 482, 489, 485, 480,
29   482, 480, 484, 478, 477, 473, 474, 477, 478, 472, 461, 467, 462, 457, 461, 451,
30   446, 444, 441, 440, 437, 439, 439, 432, 431, 430, 425, 418, 414, 409, 408, 411,
31   402, 393, 393, 382, 376, 372, 362, 352, 345, 326, 317, 298, 280, 262, 253, 226,
32   218, 196, 177, 163, 148, 137, 129, 118, 119, 107, 104, 93, 88, 87, 84, 78, 76,
33   72, 69, 70, 72, 71, 71, 64, 65, 66, 71, 63, 63, 67, 71, 61, 60, 71]
34
35 valeur = [ 53, 65, 62, 53, 56, 60, 63, 59, 64, 58, 53, 65, 53, 65, 62, 53, 56, 60, 63, 59, 64, 58, 53, 65
36   , 53, 65, 62, 53, 56, 60, 63, 59, 64, 58, 53, 65, 53, 65, 62, 53, 56, 60, 63, 59, 64, 58, 53, 65,
37   53, 65, 62, 53, 56, 60, 63, 59, 64, 58, 53, 65, 53, 65, 62, 53, 56, 60, 63, 59, 64, 58, 53, 65
38   , 59, 64, 58, 53, 53, 63, 250, 226,
39   218, 196, 177, 163, 148, 137, 129, 118, 119, 107, 104, 93, 88, 87, 84, 78, 76,
40   72, 69, 70, 72, 71, 71, 64, 65, 66, 71, 63, 63, 67, 71, 61, 60, 71]
41
42 valeurs_temps_secondes = []
43 for valeur_temps in valeurs_temps:
44     temps = datetime.datetime.strptime(valeur_temps, '%H:%M:%S.%f')
45     temps_secondes = temps.second + temps.microsecond / 1e6 - 20
46     valeurs_temps_secondes.append(temps_secondes)
47
48 valeurs_temps_secondes_inverse = valeurs_temps_secondes[::-1] # Inversion de la liste temps
49 valeurs_inverse = valeurs[::-1] # Inversion de la liste temps
50 valeur_inverse = valeur[::-1]
51
52 plt.plot(valeurs_temps_secondes_inverse, valeurs_inverse,)
53 plt.plot(valeurs_temps_secondes_inverse, valeur_inverse)
54 plt.xlabel('Temps (s)')
55 plt.ylabel('Quantite de gaz (ppm)')
56 plt.title('variation du quantite de gaz en fct du temps avec ventilateur')
57 plt.show()

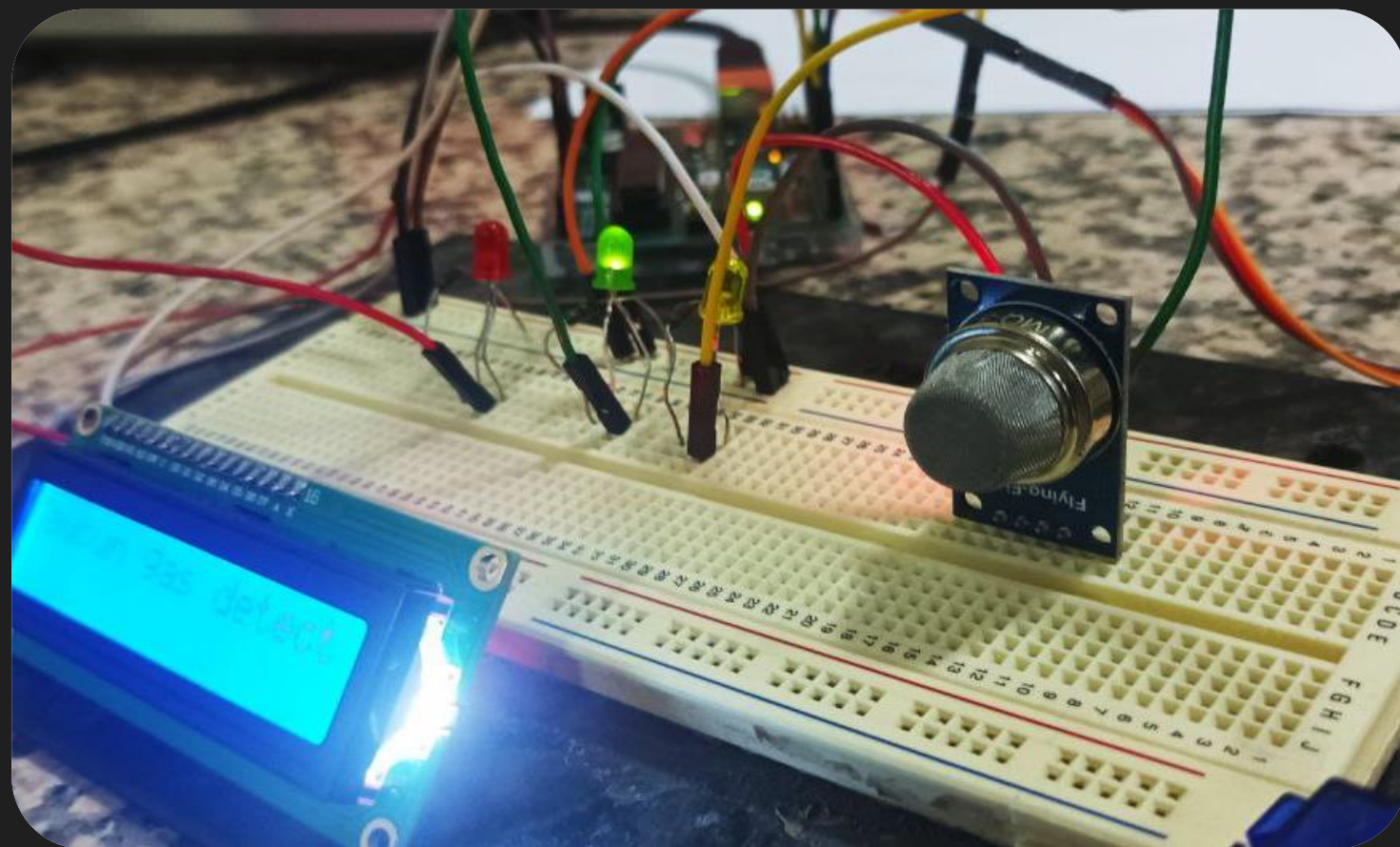
```




```

1 #include <MQUnifiedSensor.h>
2 #include <MQ5.h>
2 #include <LiquidCrystal_I2C.h>
3
4 int redled = 2;
5 int greenled = 3;
6 int yellowled = 4;
7 int sensor = A0;
8 int sensorThresh = 500;
9 int sensormoyenn = 250;
10 LiquidCrystal_I2C lcd_1(0x27, 16, 2);
11
12 void setup() {
13     lcd_1.begin();
14     lcd_1.print("  welcome  ");
15     lcd_1.setCursor(0, 1);
16     lcd_1.print(" to smart detector");
17     pinMode(redled, OUTPUT);
18     pinMode(greenled, OUTPUT);
19     pinMode(yellowled, OUTPUT);
20     pinMode(sensor, INPUT);
21     Serial.begin(9600);
22 }
23
24 void loop() {
25     int analogValue = analogRead(sensor);
26     Serial.println(analogValue);
27     if (analogValue > sensorThresh) {
28         lcd_1.clear();
29         lcd_1.print("Dangereux!!!");
30         lcd_1.setCursor(0, 1);
31         lcd_1.print(analogValue);
32         digitalWrite(redled, HIGH);
33         digitalWrite(greenled, LOW);
34         digitalWrite(yellowled, LOW);
35     }
36     if (analogValue > sensormoyenn and analogValue < sensorThresh) {
37         lcd_1.clear();
38         lcd_1.print("l'environnement!!");
39         lcd_1.setCursor(0, 1);
40         lcd_1.print(analogValue);
41         digitalWrite(yellowled, HIGH);
42         digitalWrite(greenled, LOW);
43         digitalWrite(redled, LOW);
44     }
45     else {
46         lcd_1.clear();
47         lcd_1.print("aucun gas detecte");
48         lcd_1.setCursor(0, 1);
49         lcd_1.print(analogValue);
50         digitalWrite(yellowled, LOW);
51         digitalWrite(greenled, HIGH);
52         digitalWrite(redled, LOW);
53     }
54 }
55 }

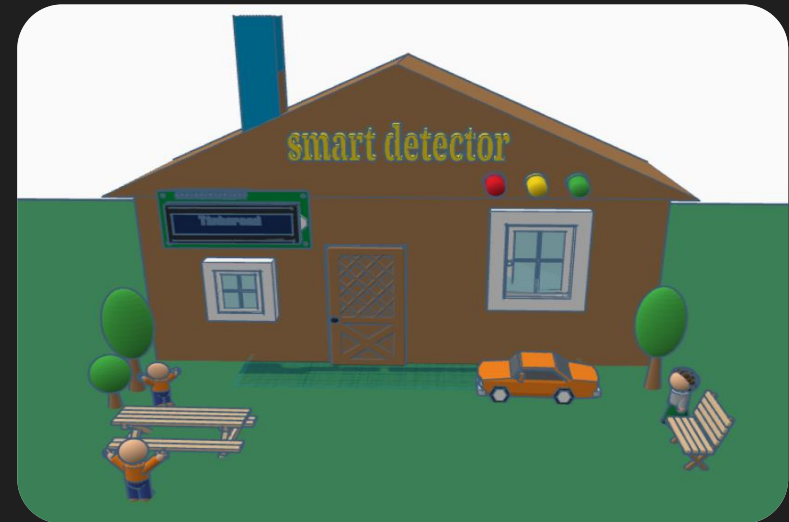
```



```

1 #include <MQUnifiedSensor.h>
2 #include <MQ5.h>
3 #include <LiquidCrystal_I2C.h>
4 #include <Servo.h>
5 #include <SoftwareSerial.h>
6
7 SoftwareSerial sim(10, 11);
8 String number = "+212675105556"; //-> change with your number
9 int redled = 2;
10 int greenled = 3;
11 int yellowled = 4;
12 int sensor = A0;
13 int sensorThresh = 500;
14 int sensormoyenn = 250;
15 int fan = 11;
16 int pos = 0;
17 Servo servo_9;
18 LiquidCrystal_I2C lcd_1(0x27, 16, 2);
19
20 void setup() {
21   lcd_1.begin();
22   lcd_1.print("   welcome   ");
23   lcd_1.setCursor(0, 1);
24   lcd_1.print(" to smart detector");
25   _buffer.reserve(50);
26   sim.begin(9600);
27   delay(1);
28   servo_9.attach(9, 500, 2500);
29   pinMode(redled, OUTPUT);
30   pinMode(greenled, OUTPUT);
31   pinMode(yellowled, OUTPUT);
32   pinMode(fan, OUTPUT);
33   pinMode(sensor, INPUT);
34   Serial.begin(9600);
35 }
36
37 void loop() {
38   int analogValue = analogRead(sensor);
39   Serial.println(analogValue);
40   if (analogValue > sensorThresh) {
41     lcd_1.clear();
42     lcd_1.print("Dangereux!!!");
43     lcd_1.setCursor(0, 1);
44     lcd_1.print(analogValue);
45     digitalWrite(redled, HIGH);
46     digitalWrite(greenled, LOW);
47     digitalWrite(yellowled, LOW);
48     digitalWrite(fan, HIGH);
49     SendMessage();
50     callNumber();
51
52     for (pos = 0; pos <= 90; pos += 1) {
53       // tell servo to go to position in variable 'pos'
54       servo_9.write(pos);
55       // wait 15 ms for servo to reach the position
56       delay(1); // Wait for 15 millisecond(s)
57     }
58   }
59   if (analogValue > sensormoyenn and analogValue < sensorThresh) {
60     lcd_1.clear();
61     lcd_1.print("l'environnement!!");
62     lcd_1.setCursor(0, 1);
63
64
65     lcd_1.print(analogValue);
66     digitalWrite(yellowled, HIGH);
67     digitalWrite(greenled, LOW);
68     digitalWrite(redled, LOW);
69     digitalWrite(fan, HIGH);
70     SendMessage();
71     callNumber();
72
73     for (pos = 90; pos >= 0; pos -= 1) {
74       // tell servo to go to position in variable 'pos'
75       servo_9.write(pos);
76       // wait 15 ms for servo to reach the position
77       delay(1); // Wait for 1 millisecond(s)
78     }
79   }
80   }
81   else {
82
83     lcd_1.clear();
84     lcd_1.print("aucun gas detecte");
85     lcd_1.setCursor(0, 1);
86     lcd_1.print(analogValue);
87     digitalWrite(yellowled, LOW);
88     digitalWrite(greenled, HIGH);
89     digitalWrite(redled, LOW);
90     digitalWrite(fan, LOW);
91   }
92 }
93
94 void SendMessage()
95 {
96   //Serial.println ("Sending Message");
97   sim.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode
98   delay(200);
99   //Serial.println ("Set SMS Number");
100  sim.println("AT+CMGS=\"\" + number + "\"\r"); //Mobile phone number to send message
101  delay(200);
102  String SMS = "Gas Detect";
103  sim.println(SMS);
104  delay(100);
105  sim.println((char)26); // ASCII code of CTRL+Z
106  delay(200);
107 }
108
109 void callNumber() {
110   sim.print (F("ATD"));
111   sim.print (number);
112   sim.print (F(";r\n"));
113 }
114
115
116
117
118
119
120
121
122
123
124
125
126
127

```



Tout

Non payé (0)

En cours (0)

Expédié (0)

Terminé

Commandes supprimées

Commande

No de commande, produit ou vendeur

Tout

Terminé

Commande passée le 23 nov. 2022

Numéro de commande: 8158298673178191

Détails de la commande

MQ-5 Module

MQ-2 MQ-3 MQ-4 MQ-5 MQ-6 MQ-7 MQ-8 MQ-9 MQ-135 Detection Sm...

MQ-5

US \$0.73 x1

Total:US \$3.61

Ajouter au panier

Supprimer

Terminé

Commande passée le 23 nov. 2022

Numéro de commande: 8158298673198191

Détails de la commande

1pcs keystudio 9G MINI SG90 90 degrees Servo Motor Blue with PH2.5...

US \$2.99 x1

Total:US \$4.45

Ajouter au panier

Supprimer

Terminé

Commande passée le 23 nov. 2022

Numéro de commande: 8158298673218191

Détails de la commande

New SIM800L GPRS GSM Module w/ PCB Antenna SIM Board Quad ban...

Glue stick antenna

US \$3.68 x1

Total:US \$6.44

Ajouter au panier

Supprimer

Terminé

Commande passée le 23 nov. 2022

Numéro de commande: 8158298673238191

Détails de la commande

LCD1602+I2C Module Blue / Yellow Green Screen 16x2 Character LCD ...

LCD1602 Blue I2C

US \$1.95 x1

Total:US \$3.71

Ajouter au panier

Supprimer

TOTALE

182,1 DH

