

Systemes Embarqués II

Google Sheet

ISAT – EPHEC 2020-2021

Juan Alvarez et Olivier Grabenweger

2 Table des matières

1	Introduction.....	2
2	Mindmapping - Logigramme	2
	a) Mindmap.....	2
	b) Logigramme.....	3
3	Schéma de câblage	4
	a) Listes des composants :	4
4	Code source	5
	a) Code ESP8266 :.....	5
	b) Script Google	8
5	Conclusion	9
6	Annexes et bibliographie	9
	a) Annexes :	9
	b) Bibliographies :.....	9

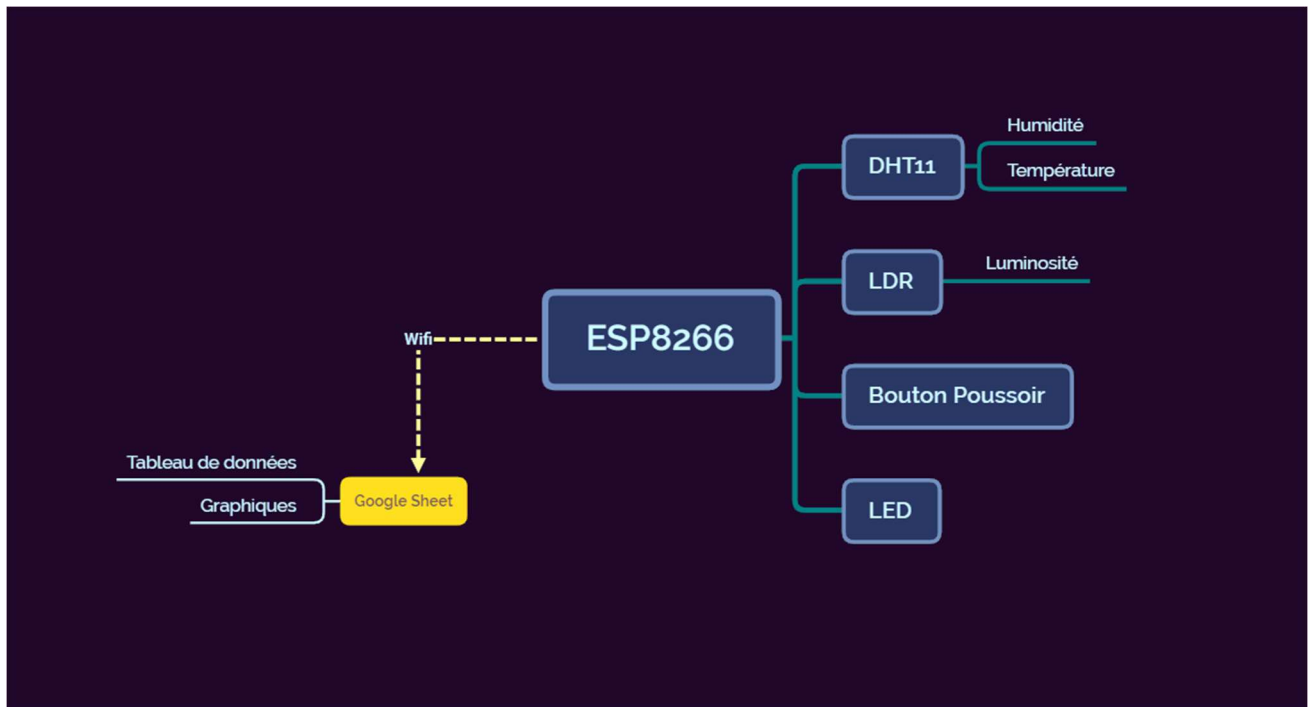
1 Introduction

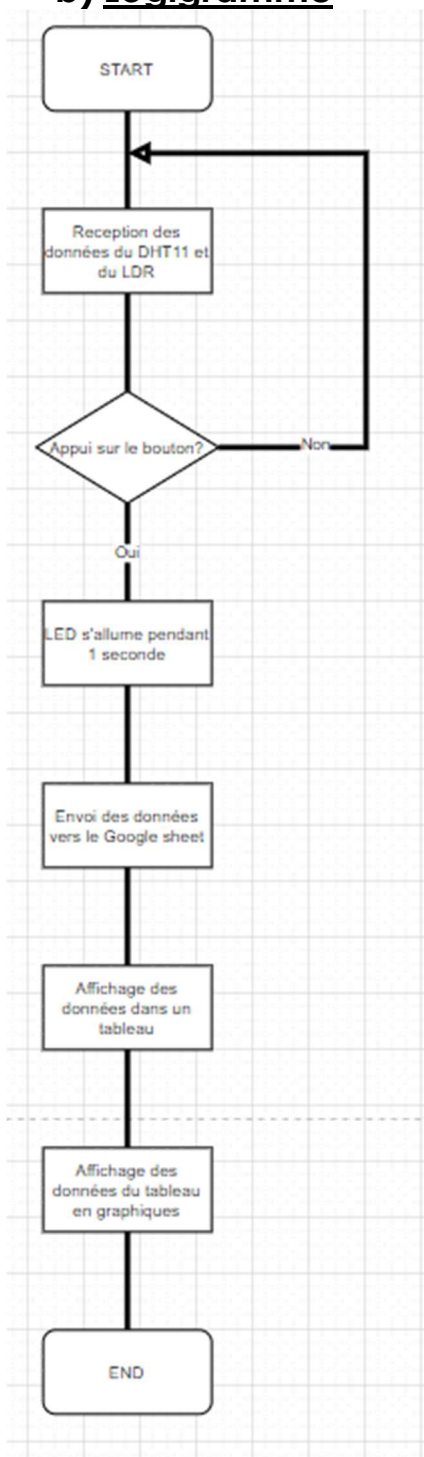
Le but de ce TP est de parvenir à envoyer les données de nos capteurs vers un google Sheet afin de faire un tableau de données et des graphiques. Les données envoyées via notre ESP8266 par WiFi sont la température et l'humidité grâce au DHT11 et la luminosité grâce au LDR.

Pour cela, nous devons donc préparer et lier un google sheet et le script associé avec le code de l'ESP8266 qui transmettra les données de nos capteurs.

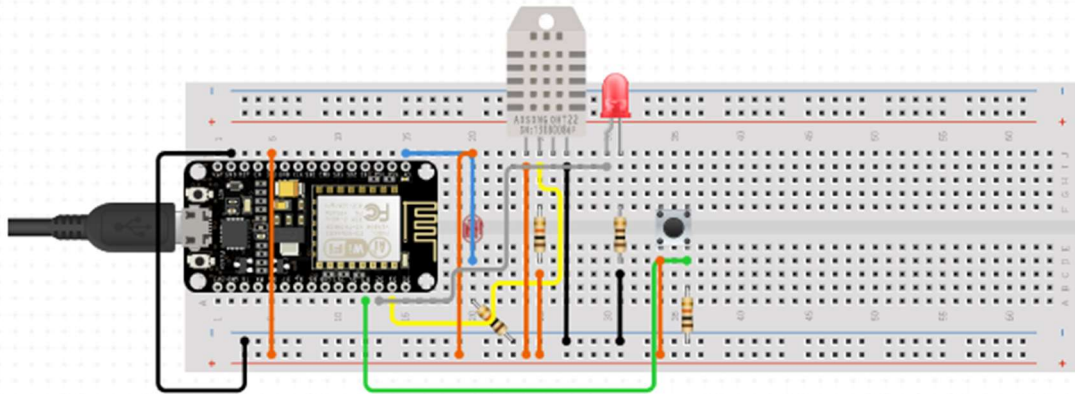
2 Mindmapping - Logigramme

a) Mindmap



b) Logigramme

3 Schéma de câblage



! Attention aux PINs qui ne reflètent pas celles utilisées dans le code !

a) Listes des composants :

- 1 ESP8266
- 1 Bred Board
- 1 LED
- 1 LDR
- 1 DHT11
- 1 Bouton Poussoir
- 3 Résistances de 10k Ohm
- 1 Résistance de 330 Ohm

4 Code source

a) Code ESP8266 :

```

1  /* Bibliothèques */
2
3  #include <Arduino.h>
4  #include <ESP8266WiFi.h>
5  #include <WiFiClientSecure.h>
6  #include "DHT.h"
7
8  /* Variables */
9
10 WiFiClientSecure gsclient;
11 String GAS_ID = "AKfycbybjsotK0WczFlCYcgxVW9FIzJPjyyWERmYpc7iVP3F1HJZBrA"; //getactivespreadsheetID
12 const char* fingerprint = "46 B2 C3 44 9C 59 09 8B 01 B6 F8 BD 4C FB 00 74 91 2F EF F6"; //génération d'un fingerprint
13 const char* host = "script.google.com"; // configuration de l'hôte
14 const int httpsPort = 443; // configuration du port https
15
16 #ifndef STASSID //configuration nom du Wi-Fi et de son mot de passe
17 #define STASSID "bbox-Sophiel" // nom du Wi-Fi
18 #define STAPSK "20150509Sophi" // mot de passe du Wi-Fi
19 #endif
20 const char* ssid = STASSID;
21 const char* password = STAPSK;
22
23 const int BUTTON = D0; // Lier la Pin D0 à notre bouton
24 const int LEDred = D1; // Lier la Pin D1 à notre LED rouge
25 const int LDR = A0; // Lier la Pin A0 à notre capteur de Luminosité
26
27
28 float humy; // Variable pour l'humidité
29 float temp; // Variable pour la température
30 int lumi; // Variable pour la luminosité
31 int state_button = 0; // Variable pour l'état de notre bouton
32 int flag_button = 0; // flag du bouton
33 int count = 0; // compteur du nombre d'appui sur le bouton
34
35 #define DHTPIN D6 // Lier la Pin D6 à notre capteur DHT
36 #define DHTTYPE DHT11 // Préciser le type de DHT, ici le DHT 11
37 DHT dht(DHTPIN, DHTTYPE); // configuration de notre DHT en précisant via nos variables: La Pin et le type de DHT
38
39 /* Fonction du Bouton poussoir */
40
41 void button_Read() //cette fonction va servir à ce que pour un appui il y ai un envoi de donnée
42 {
43     state_button = digitalRead(BUTTON); // Lire l'état du bouton et l'attribuer à notre variable state_button
44     if (state_button == LOW) // comme le bouton est en pull-up on met le flag à 1 quand l'état du bouton est BAS
45     {
46         flag_button = 1;
47     }
48     if (flag_button == 1 && state_button == HIGH)
49     {
50         count += 1; //incrémenter le compteur
51         flag_button = 0;
52     }
53 }

```

```

54
55  /* Fonction d'envoi de données */
56
57  void sendData(int x, int y, int z) // prend comme paramètre les données que l'on souhaite envoyer, ici température, humidité et luminosité
58  {
59      Serial.print("connecting to ");
60      Serial.println(host);
61      gsclient.setInsecure();
62      if (!gsclient.connect("script.google.com", httpsPort)) // on vérifie le port et le fingerprint
63      {
64          Serial.println("connection failed");
65          return;
66      }
67      if (gsclient.verify(fingerprint, host)) // on vérifie le port et le fingerprint
68      {
69          Serial.println("certificate matches");
70      }
71      else
72      {
73          Serial.println("certificate doesn't match");
74      }
75      // x = température, y = humidité et z = luminosité
76      String string_x = String(x, DEC); // conversion de x en string
77      String string_y = String(y, DEC); // conversion de y en string
78      String string_z = String(z, DEC); // conversion de z en string
79
80      Serial.println(string_x);
81      Serial.println(string_y);
82      Serial.println(string_z);
83      //création de l'url qui enverra les données sur le google sheet
84      String url = "/macros/s/" + GAS_ID + "/exec?temperature=" + string_x + "&humidity=" + string_y + "&luminosity=" + string_z;
85      Serial.print("requesting URL: ");
86      Serial.println(url);
87
88
89      gsclient.print(String("GET ") + url + " HTTP/1.1\r\n" +
90          "Host: " + host + "\r\n" +
91          "User-Agent: BuildFailureDetectorESP8266\r\n" +
92          "Connection: close\r\n\r\n");
93
94      Serial.println("request sent");
95
96      while (gsclient.connected()) {
97          String line = gsclient.readStringUntil('\n');
98          if (line == "\r")
99          {
100              Serial.println("headers received");
101              break;
102          }
103      }
104      String line = gsclient.readStringUntil('\n');
105      Serial.println(line);
106      if (line.startsWith("{\"state\":\"success\"}"))
107      {
108          Serial.println("esp8266/Arduino CI successful!");
109      }

```

```

110     else
111     {
112         Serial.println("esp8266/Arduino CI has failed");
113     }
114     Serial.println("reply was:");
115     Serial.println("closing connection");
116     }
117
118 /* fonction setup */
119
120 void setup()
121 {
122     Serial.begin (9600);
123     pinMode(LEDred, OUTPUT);    // défini la LED en output
124     digitalWrite(LEDred, LOW);  // Eteind la LED
125     pinMode(BUTTON, INPUT);    // défini le bouton en input
126     dht.begin();               // Initialise le DHT
127     WiFi.mode(WIFI_STA);       // Précise le mode du Wifi
128     WiFi.begin(ssid, password); // Attribue le nom et le mot de passe du wifi
129     // Wait for connection
130     while (WiFi.status() != WL_CONNECTED) {    // Tant que la connection au wifi n'est pas établie, en imprime des points sur le terminal
131         delay(1000);
132         Serial.print(".");
133     }
134 }
135
136 /* fonction loop */
137
138 void loop()
139 {
140     button_Read();
141     float temp = dht.readTemperature(); //on lit la valeur de la température du DHT et on l'attribue à une variable
142     float humy = dht.readHumidity();    //on lit la valeur de l'humidité du DHT et on l'attribue à une variable
143     int lumi = analogRead(LDR);         //on lit la valeur de la luminosité du capteur LDR et on l'attribue à une variable
144
145     if (flag_button == 1) // On envoie nos données quand on appuie sur le bouton et pour le vérifier une LED s'allume brièvement
146     {
147         sendData(temp,humy,lumi);
148         Serial.println(temp);
149         Serial.println(humy);
150         Serial.println(lumi);
151         digitalWrite(LEDred, HIGH);
152         delay(100);
153         digitalWrite(LEDred, LOW);
154     }
155 }

```


b) Script Google

```

1 function doGet(e) {
2   Logger.log( JSON.stringify(e) ); // view parameters
3   var result = 'Ok'; // assume success
4   if (e.parameter == 'undefined') {
5     result = 'No Parameters';
6   }
7   else {
8     var sheet_id = '1NzX6m5DuVqKsICEiNpMvH5jggmSMWx3CYS1PRBigUyc'; // Sheet ID
9     var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet(); // get Active sheet //getSheetByName('nameOfSheet') for another sheet
10    var newRow = sheet.getLastRow() + 1;
11    var rowData = [];
12
13    rowData[0] = new Date(); // Timestamp in column A
14    for (var param in e.parameter) {
15      Logger.log('In for loop, param=' + param);
16      var value = stripQuotes(e.parameter[param]);
17      Logger.log(param + ':' + e.parameter[param]);
18      switch (param) {
19        case 'temperature': //Parameter
20          rowData[1] = value; //Value in column B
21          result = 'Write on Column B';
22          break;
23        case 'humidity': //Parameter
24          rowData[2] = value; //Value in column C
25          result += ', Write on column 3';
26          break;
27        case 'luminosity': //Parameter
28          rowData[3] = value; //Value in column D
29          result += ', Write on column 4';
30          break;
31        default:
32          result = "unsupported parameter";
33      }
34    }
35    Logger.log(JSON.stringify(rowData));
36    // Write new row below
37    var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
38    newRange.setValues([rowData]);
39  }
40  // Return result of operation
41  return ContentService.createTextOutput(result);
42 }
43 /**
44  * Remove leading and trailing single or double quotes
45  */
46 function stripQuotes( value ) {
47   return value.replace(/^[\'"]|[\']$/g, "");
48 }

```

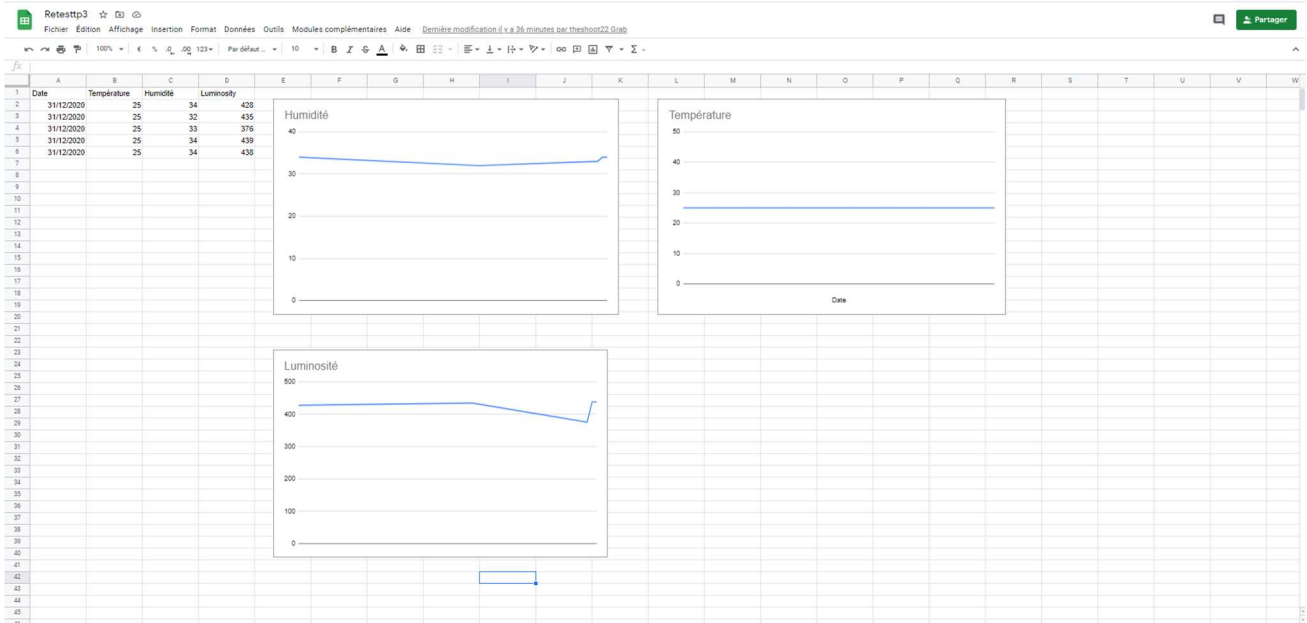
5 Conclusion

Finalement, nous découvrons grâce à ce TP combien il est simple et pratique d'utiliser un outil tel que google Sheet afin de répertorier et traiter de simples données. En faisant des recherches sur le sujet, on a remarqué qu'il y a beaucoup plus de gens qui utilisent cet outil que nous le pensions.

Ça nous a aussi donné envie de réutiliser nos connaissances acquises ici dans le TP sur nodered.

6 Annexes et bibliographie

a) Annexes :



b) Bibliographies :

- <https://script.google.com/>
- <https://docs.google.com/spreadsheets/>
- <https://www.circuito.io/> : schéma de câblage
- <http://draw.io/> : organigramme
- Xmind : Mindmap