Systèmes Embarqués II

Blynk

ISAT – EPHEC 2020-2021

Juan Alvarez et Olivier Grabenweger

2 Table des matières

1 Introduction

Cette fois-ci, nous allons utiliser la dashboard d'une application mobile de smartphone pour transmettre des données et interagir avec notre NodeMCU et ses divers capteurs et actionneurs. On va concevoir cette application à l'aide de Blynk, une plate-forme IoT (internet des objets).

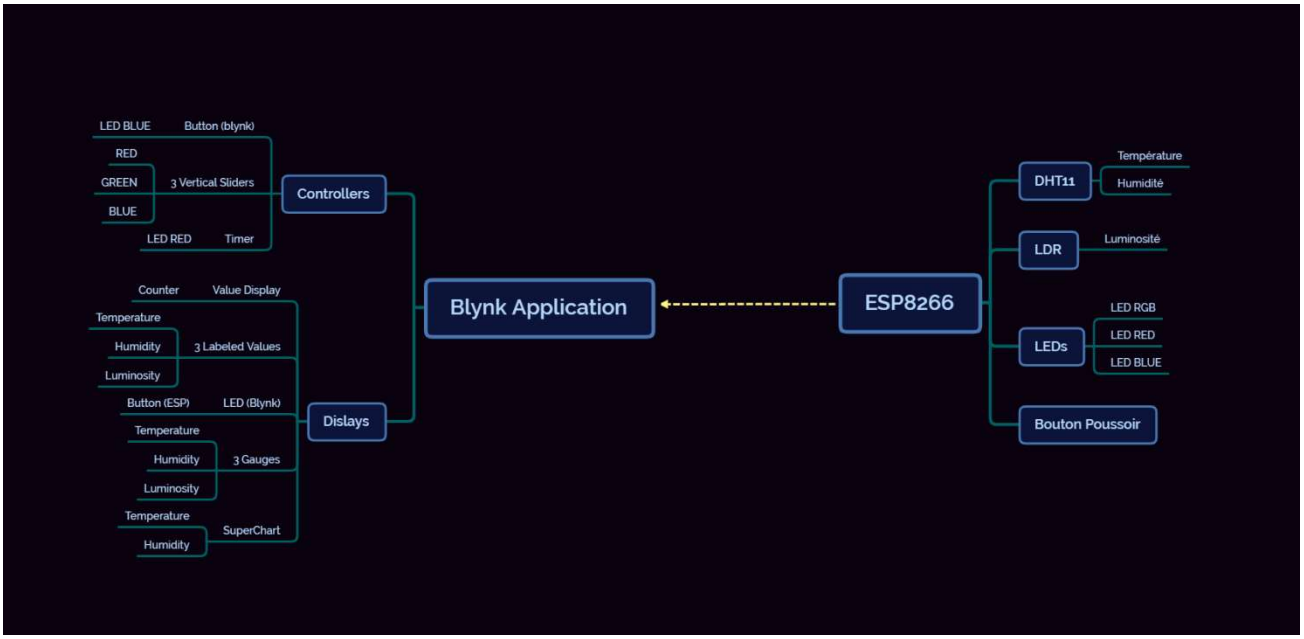
Grâce à un serveur cloud privé de l'EPHEC mis à notre disposition, nous avons accès un nombre presque illimité d'énergie, ce qui nous permettra de concevoir notre application de manière complète et sans restriction (chaque Widget coûte de l'énergie).

Donc le but sera d'avoir une application comportant au minimum :

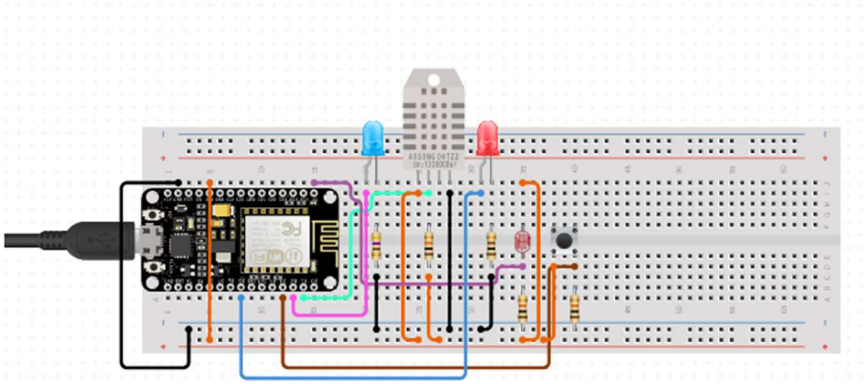
- Controllers :
  - 1 Button
  - 1 Vertical Slider
  - 1 Timer
- Displays :
  - 1 Value Display
  - 1 Labeled Value
  - 1 LED
  - 1 Gauge
  - 1 SuperChart

La façon d'utiliser ces widgets étant laissé libre, la façon dont l'application a été programmé sera défini dans ce rapport.

2 Mindmapping



3 Schéma de câblage



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

a) Liste des composants

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

4 Code source

a) Code ESP8266 :

```
/* Bibliothèques */

#include <Arduino.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

/* Variables */

char auth[] = "8kVbd1Bo9CIy25-HdSLoAAJWb3cKdw5K"; //Le Auth Token reçu par mail de Blynk
char ssid[] = "bbox-Sophie1"; // Nom du WiFi
char pass[] = "20150509Sophi"; // Mot de passe du WiFi
#define DHTPIN D6   // Pin connectée au DHT
#define DHTTYPE DHT11  //Préciser le type de DHT: ici le DHT11
DHT dht(DHTPIN, DHTTYPE); // Fonction de paramètrage du DHT avec PIN et Type

BlynkTimer timer; //

//Attribution des PIN à nos différents capteurs/actionneurs
const int BUTTON = D0;
const int RED = D3;
const int GREEN = D4;
const int BLUE = D7;
const int LDR = A0;

long send_time;

int state_button = 0; // Etat du bouton
int flag_button = 0;  // Flag du bouton
int count = 0;  // Compteur

/* Fonction du Bouton */

void button_Read()
{
    state_button = digitalRead(BUTTON); // Lire l'état du bouton
    if (state_button == LOW)  // Le bouton est en pull*up donc si le bouton est LOW c'est qu'il
est activé
    {
        flag_button = 1;    // Le flag va servir à ne pas avoir de résonnance sur le bouton
        Blynk.virtualWrite(V8, 255);    // Envoi de donnée vers blynk pour allumer la LED sur
l'application en TAB4
    }
    if (flag_button == 1 && state_button == HIGH) // Si bouton relâché
    {
      count += 1;   // incrémentation compteur
      Serial.println(count);
      Blynk.virtualWrite(V9, count);    // Envoi de donnée vers blynk pour transmettre la valeur
du compteur au value display en TAB4
      Blynk.virtualWrite(V8, 0);    // Envoi de donnée vers blynk pour éteindre la LED sur
l'application en TAB4
      flag_button = 0;
    }
```

```
55 }
56
57 /* Fonction d'envoi de données */
58
59 void sendSensor()
60 {
61   if(send_time <= millis())
62   {
63     float h = dht.readHumidity(); // Lecture de la valeur de l'humidité sur le DHT
64     float t = dht.readTemperature(); // Lecture de la valeur de la température sur le DHT
65     int lumi = analogRead(LDR); // Lecture de la valeur de la luminosité sur le DHT
66     if (isnan(h) || isnan(t)) {   // gestion d'erreur si on ne reçois pas de données du DHT
67       Serial.println("Failed to read from DHT sensor!");
68       return;
69     }
70     // Please don't send more that 10 values per second.
71     Blynk.virtualWrite(V6, h);  // Envoi de l'humidité vers blynk    sur TAB1 et TAB2
72     Blynk.virtualWrite(V5, t);  // Envoi de la température vers blynk   sur TAB1 et TAB2
73     Blynk.virtualWrite(V7, lumi); // Envoi de la luminosité vers blynk    sur TAB1
74     send_time = millis() + 1000;
75   }
76 }
77
78 /* Fonction setup */
79
80 void setup()
81 {
82   Serial.begin(9600);  //initialisation du serial àune fréquence de 9600
83   pinMode(BUTTON, INPUT); // Définir le bouton en input
84   send_time = millis();
85
86   Blynk.begin(auth, ssid, pass, IPAddress(193,190,65,122), 8080);   //Initialisation de blynk
87 Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
88   dht.begin();    //Initialisation du DHT
89   delay(1000);
90 }
91
92 /* Fonction loop */
93
94 void loop()
95 {
96   Blynk.run();  // Lancer Blynk
97   timer.run();   // Lancer Timer
98   sendSensor();   // Fonction d'envoi de données
99   button_Read();    // Fonction du bouton
100 }
```

b) Explications du fonctionnement des interactions entre l'ESP et Blynk

TAB 1 : Il y a 3 Gauges et 3 Labeled value.

Une donnée d'un capteur sera affichée sur une Gauge et un Value Display. C'est donc le cas pour :

- La température en haut à gauche du TAB

- L'humidité en haut à droite du TAB

- La luminosité en bas au centre du TAB

Pour afficher ces données sur ces Widgets nous avons dû attribuer, dans notre code, une pin virtuelle à chaque valeur de nos capteurs et préciser sur Blynk la pin choisie.

On a aussi précisé la valeur min et max des valeurs lues pour que les données soient plus visibles.

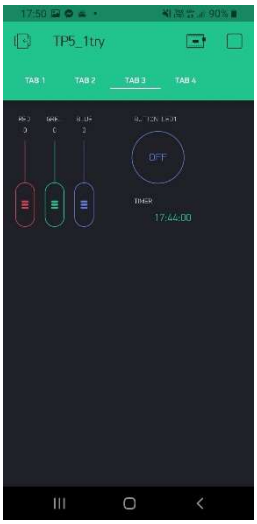
Données	PIN	Intervalle
Température	V5	0-50
Humidité	V6	0-100
Luminosité	V7	0-1023

Les fréquences de lecture des widgets cités jusqu'ici ont été paramétrés sur « PUSH », c'est-à-dire chaque fois qu'une donnée est envoyée à partir de l'ESP, elle sera automatiquement sauvegardée sur le serveur.

TAB 2 : Il y a un SuperChart

Le superChart affiche les données de température et d'humidité en fonction du temps. On peut modifier l'axe des abscisses pour que les données du DHT soient affichées en temps réel ou sur 15 minutes, 30 minutes et jusqu'à un intervalle de 13h.

En paramètres, on doit juste préciser le titre des données et la pin d'où elles sont prises, ici V5 et V6.



TAB 3 : Il y a trois vertical slider, un button et un timer

Les trois slider servent à régler l'intensité des différentes couleurs de la LED RGB : rouge, vert et bleu.

Le bouton sert à allumer le LED BLUE sur notre ESP.

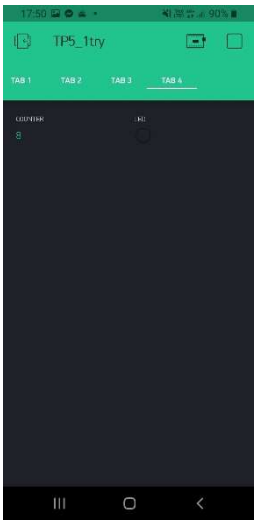
Et le Timer va allumer la LED RED sur l'ESP à l'heure qui a été paramétrée (17h44'00'' sur l'image)

Sur les sliders, on doit préciser la pin de l'ESP sur laquelle est connectée une des couleurs de la RGB, sa valeur maximum et sa valeur minimum (0-255).

Pour le bouton c'est le même principe, il faut donc juste indiquer la pin qui est connectée sur l'ESP à la LED BLUE mais pas oublier de vérifier que le bouton est en mode « PUSH » et non « Switch ».

Le Timer a besoin de trois indications : la pin à laquelle est connectée la LED RED, l'heure à laquelle la LED doit s'allumer et l'heure à laquelle elle doit s'éteindre.

Widgets	PIN	Actionneur
Slider vertical RED	GPIO 0	RGB (Rouge)
Slider vertical BLUE	GPIO 2	RGB (Vert)
Slider vertical GREEN	GPIO 13	RGB (Bleue)
Button	GPIO 4	LED BLUE
Timer	GPIO 5	LED RED



TAB 4 : Un Value display et une LED

Le Value display affiche le compteur du nombre de fois qu'on a appuyé sur le bouton connecté à l'ESP (valeur maximum choisie = 1023).

La LED s'allume quand on appuie sur ce même bouton.

Widgets	PIN
Value display	V9
LED	V8

5 Conclusion

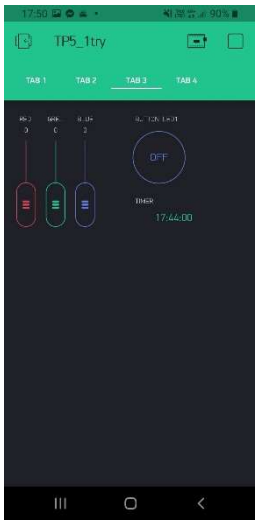
Encore une fois nous avons été impressionnés par la facilité d'utilisation de l'outil. D'ailleurs, nous avons à nouveau décidé d'utiliser le sujet de notre TP dans celui sur NodeRed.
Mais il est vrai qu'en tant que dashboard, Blynk est plus pratique et plus simple à utiliser que tous les autres.

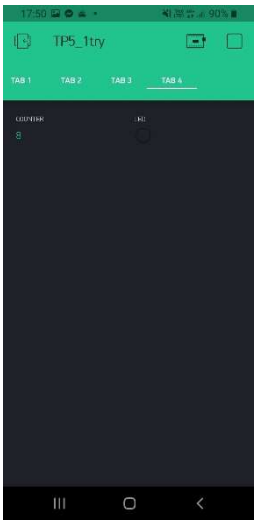
Avec la bonne bibliothèque, il n'est pas nécessaire d'écrire des lignes de code pour qu'un « controller » sur Blynk interagisse avec un actionneur connecté à l'ESP. Cette simplification implique que le code rédigé dans ce TP est relativement court. Ce qui présente un avantage certain quant 'on doit développer des applications dont le code, le fonctionnement, est complexe.

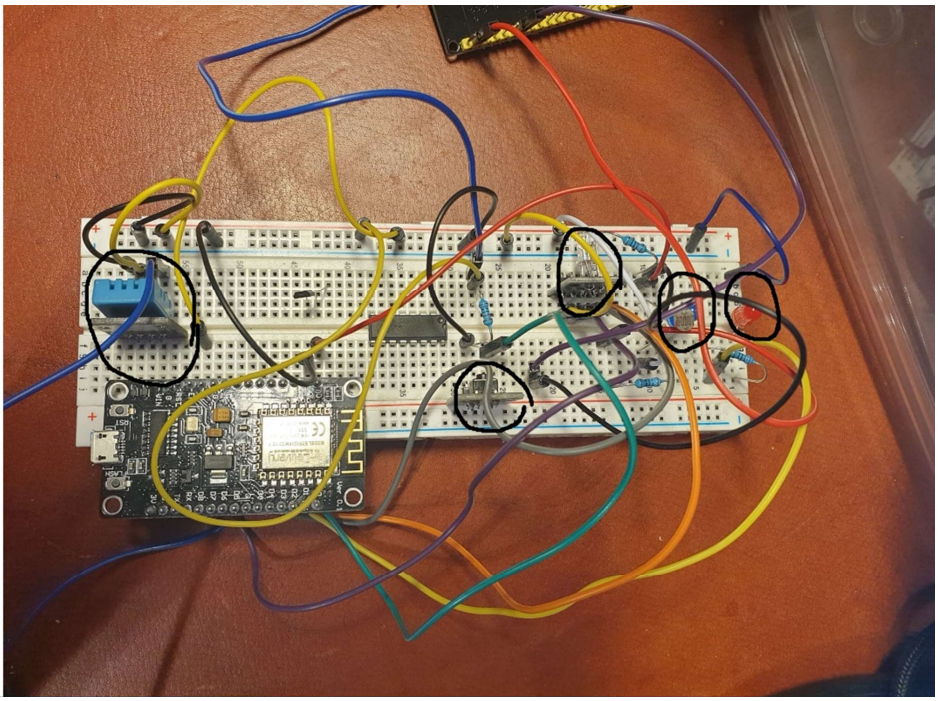
Notons toutefois qu'heureusement qu'on a eu accès, grâce à un serveur privé de l'EPHEC, à un nombre illimité d'énergie car la version de base ne laisse que très peu de flexibilité dans la construction de l'application.

6 Annexes, bibliographie et illustrations

a) Annexes :







b) Bibliographies

- https://examples.blynk.cc/?board=ESP8266&shield=ESP8266%20WiFi&example=GettingStarted%2FBlynkBlink
- https://blynk.io/en/getting-started
- http://docs.blynk.cc/
- https://www.circuito.io/ : schéma de câblage
- http://draw.io/ : organigramme
- Xmind : Mindmap