

|  |
| --- |
|  |
| Projet final |
|  |
| **Industrie 4.0 & Systèmes embarqués II** |

**GAYAP CYRILLE, Voufo CHRIST**

03 janvier 2023

Projet final

Industrie 4.0 & Systèmes embarqués II

Table des matières

[INTRODUCTION 2](#_Toc123846928)

[CAHIER DE CHARGES (CDC) 2](#_Toc123846929)

[Schéma blocs : 2](#_Toc123846930)

[Carte mentale (mindmapping) : 2](#_Toc123846931)

[Description fonctionnelle : 3](#_Toc123846932)

[Planning de Gantt : 4](#_Toc123846933)

[PARTIE : SYSTEMES EMBARQUES II 5](#_Toc123846934)

[Schéma électronique complet : 5](#_Toc123846935)

[Pcb : 5](#_Toc123846936)

[Routage Pcb : 5](#_Toc123846937)

[Visualisation finale : 6](#_Toc123846938)

[Partie Hardware : 6](#_Toc123846939)

[Code source : 8](#_Toc123846940)

[Analyse du code : 8](#_Toc123846941)

[Conclusion partielle : 8](#_Toc123846942)

[PARTIE : INDUSTRIE 4.0 8](#_Toc123846943)

[Structure des dashboards : 8](#_Toc123846944)

[Gestion des données : 8](#_Toc123846945)

[Gestion des serveurs : 8](#_Toc123846946)

[Conclusion partielle : 8](#_Toc123846947)

[CONCLUSION GENERALE 9](#_Toc123846948)

[Lien vers GitHub : 9](#_Toc123846949)

[BIBLIOGRAPHIE/WEBOGRAPHIE 9](#_Toc123846950)

# INTRODUCTION

Dans le cadre de notre formation de Bachelier en Automatisation plus précisément pendant ce 1er quadrimestre, nous devions réaliser un travail sous forme d’un projet par binôme.

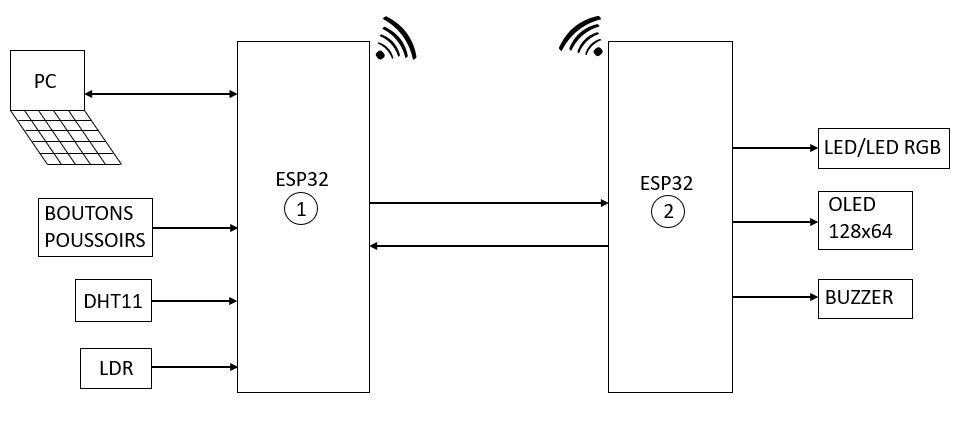
Pour ce projet, nous avons décidés de mettre sur pieds un petit automatisme tout en mettant en pratique les connaissances vues en cours aux UEs d’Industrie 4.0 & Systèmes embarqués II.

# CAHIER DE CHARGES (CDC)

Notre système est basé sur le concept de contrôle de température, humidité et luminosité dans un local. Ce mini projet contient des fonctionnalités limitées mais essentielles. Il est utile pour gérer l’ajout, la mise à jour et le stockage des données.

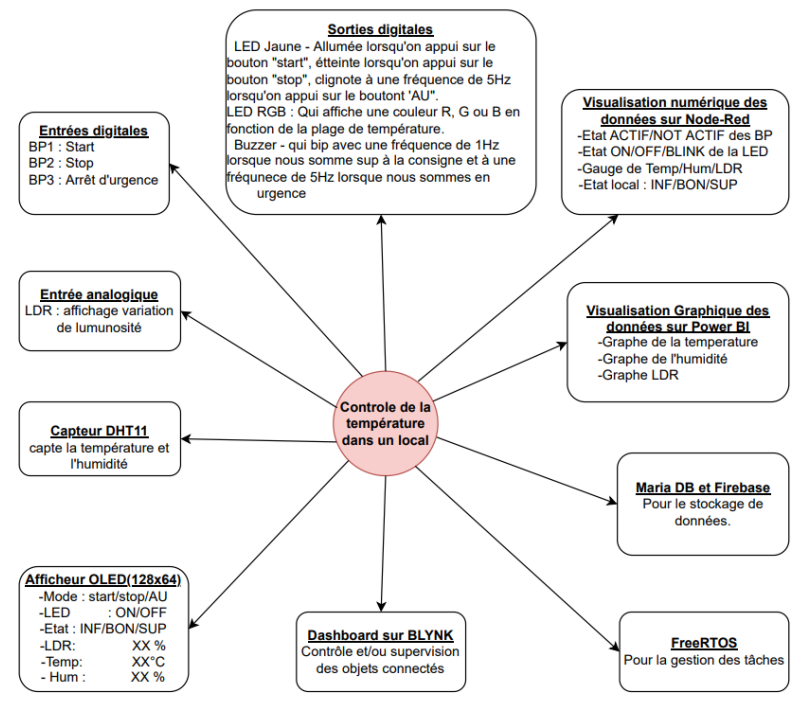
## Schéma blocs :

Ci-dessous le schéma blocs(Hardware) réalisé ;



Nous allons donc donner quelques détails des principaux éléments figurant dans ce projet.

## Carte mentale (mindmapping) :



## Description fonctionnelle :

Pour cela, nous aurons besoin de différents outils permettant de réaliser ce projet tels que :

* Un bouton BP1 pour le ‘‘START’’ qui permet de démarrer le cycle
* Un bouton BP2 pour le ‘‘STOP’’ qui permet d’arrêter le cycle
* Un bouton BP3 pour le ‘‘AU’’ pour l’arrêt d’urgence
* Une LED jaune qui s’allume en continue lorsque nous sommes en mode start, qui s’éteint lorsqu’on arrête le système et clignote à une fréquence de 5Hz lorsqu’on appuie sur l’arrêt d’urgence.
* Une RGB qui sera RED lorsque la température actuelle est supérieure à la consigne, GREEN lorsque la température actuelle est égale ou dans l’intervalle des valeurs souhaitées dans la consigne et BLUE lorsque la température actuelle est inférieure à la consigne.
* Un Buzzer qui va siffler à une fréquence de 1Hz lorsque la température actuelle est supérieure à la consigne et à une fréquence de 5Hz lorsqu’on appui sur BP3 « AU »
* Un capteur DHT11 qui nous permettra d’avoir les valeurs de température, humidité de notre local à temps réel.
* Un OLED 128x64 qui va afficher les différents éléments qui se trouvent dans le Mindmapping de la partie OLED.
* Un LDR pour obtenir la variation de luminosité dans le local
* Un Dashboard (Node-Red) sur lequel on pourra visualiser les jauges de la (LDR, Température et Humidité), l’état de la LED, RGB et des BPs
* Des zones de texte qui permettent d’afficher chacune l’état de chaque bouton, de la LED le nombre de places disponible dans le parking
* Un microcontrôleur ESP32.
* Les graphiques(LDR, température, humidité) seront visualisés sur Microsoft Power BI.
* FreeRTOS pour la gestion des tâches…lorsque la température actuelle est supérieure à la normale/consigne, la LED RGB est en rouge pendant un certain temps puis s’éteint et la LED jaune commence à clignoter jusqu’à ce que la température devienne inférieure ou égale à la consigne/normale.

## Planning de Gantt :

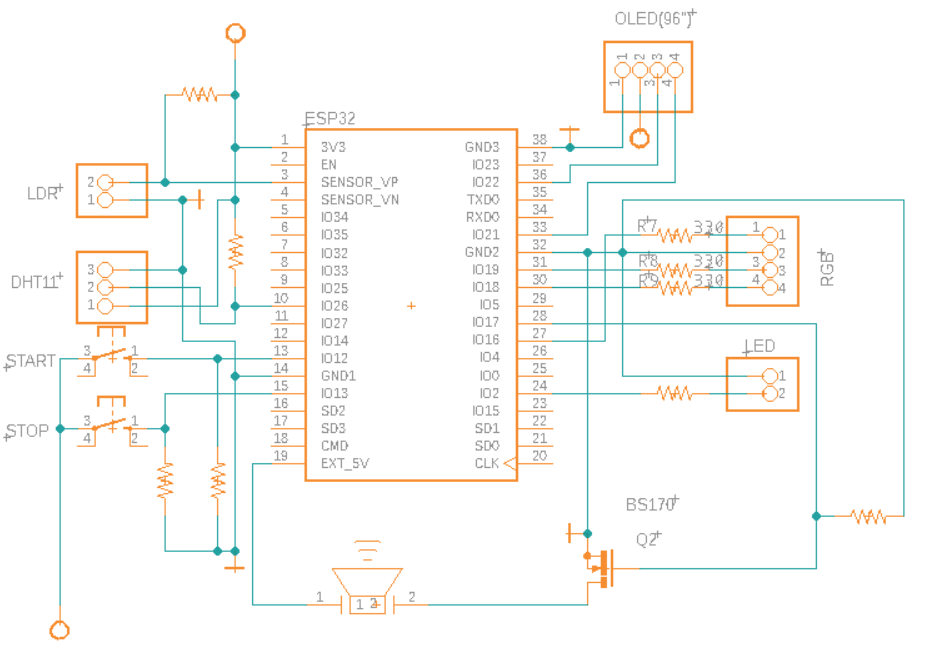
Une image contenant table

Description générée automatiquement

# PARTIE : SYSTEMES EMBARQUES II

## Schéma électronique complet :

Notre circuit électronique ci-dessous a été réalisé à l’aide de Fusion 360 ;



## Pcb :

Ci-dessous un aperçu de notre routage et visualisation avant impression.

### Routage Pcb :

Une image contenant texte, équipement électronique, circuit

Description générée automatiquement

### Visualisation finale :

Une image contenant texte, équipement électronique, circuit

Description générée automatiquement

## Partie Hardware :

**Module NodeMCU ESP32**

C’est un module basé sur un ESP32 cadencé à 240MHz et exécutant le firmware open source NodeMCU. Cette carte se programme via l’IDE Platform IO, Arduino et est compatible avec les scripts LUA.

Ce microcontrôleur dispose d’interfaces Wi-Fi et Bluetooth idéales pour les objets connectés. Des connecteurs latéraux mâles et femelles permettent d’enficher le module sur une plaque de montage rapide.

L’interface sans fil Wi-Fi permet la création de point d’accès sans fil, l’hébergement d’un serveur, la connexion à internet et le partage des données par exemple. Le module se programme directement à partir de l’IDE Platform IO et nécessite un cordon microUSB. Son implantation le rend compatible avec les plaques de connexions rapides.

Remarque : L’utilisation de cette carte est réservée à un public averti.

**Caractéristiques :**

* Alimentation :

- 5Vcc via microUSB

- 3,3Vcc via broches Vin

* Microcontrôleur : ESP32
* Microprocesseur : Tensilica LX6 Dual-Core
* Courant de fonctionnement : min. 500mA
* Gamme de fréquence d’horloge : 80MHz/240MHz
* Fréquence Wi-Fi : 2,4 GHz – 2,5GHz
* Mémoire SRAM : 512kB
* Mémoire Flash externe : 4MB
* E/S disponibles : 34
* Interfaces : I2C, SPI, I2S, CAN, UART
* Protocoles Wi-Fi : 802.11 b/g/n (802.11n jusqu’à 150 Mbps)
* Bluetooth : Classique / BLE
* Antenne intégrée
* Température de service : -40 à 125 °C
* Dimensions : 56 x 28 x 13 mm

**OLED**

C’est un composant électronique qui permet de produire de la lumière. La technologie OLED est utilisée pour l’affichage et elle possède des avantages intéressants par rapport à la technologie dominante des afficheurs à cristaux liquides (LCD). En effet, la propriété électroluminescente de l’OLED ne nécessite pas l’introduction d’un rétroéclairage ce qui confère à l’écran des niveaux de gris plus profonds et une épaisseur moindre avec un temps de réponse < 0,1 ms.

Le principal défaut des OLED est leur durée e vie (environ 14 000 heures), notamment pour les OLED bleues.

**DHT11**

Le Détecteur d’Humidité et Température est un capteur numérique de de température et d’humidité. Il utilise un capteur d’humidité capacitif et une thermistance pour mesurer l’air ambiant et renvoie en sortie un signal numérique. Facile à utiliser mais nécessite un certain timing pour saisir les données. Le véritable problème est que les données ne peuvent pas être obtenues qu’une fois max. toutes les 2 secondes.

**LDR**

Il s’agit ici d’un composant électronique ayant une résistivité variable en fonction de la lumière incidente : moins il est éclairé, plus sa resistivité augmente. Possédant plusieurs avantages à savoir : facilité de mise en œuvre, faible coût et sensibilité élevée, il a aussi des inconvénients tels que : instabilité à longue(vieillissement dû aux échauffements) et un temps de réponse élevé (0,1 µs à 100 ms).

**BUZZER**

C

**RGB**

D

## Code source :

## Analyse du code :

## Conclusion partielle :

PARTIE : INDUSTRIE 4.0

## Structure des dashboards :

## Gestion des données :

## Gestion des serveurs :

## Conclusion partielle :

# CONCLUSION GENERALE

## Lien vers GitHub :

# BIBLIOGRAPHIE/WEBOGRAPHIE