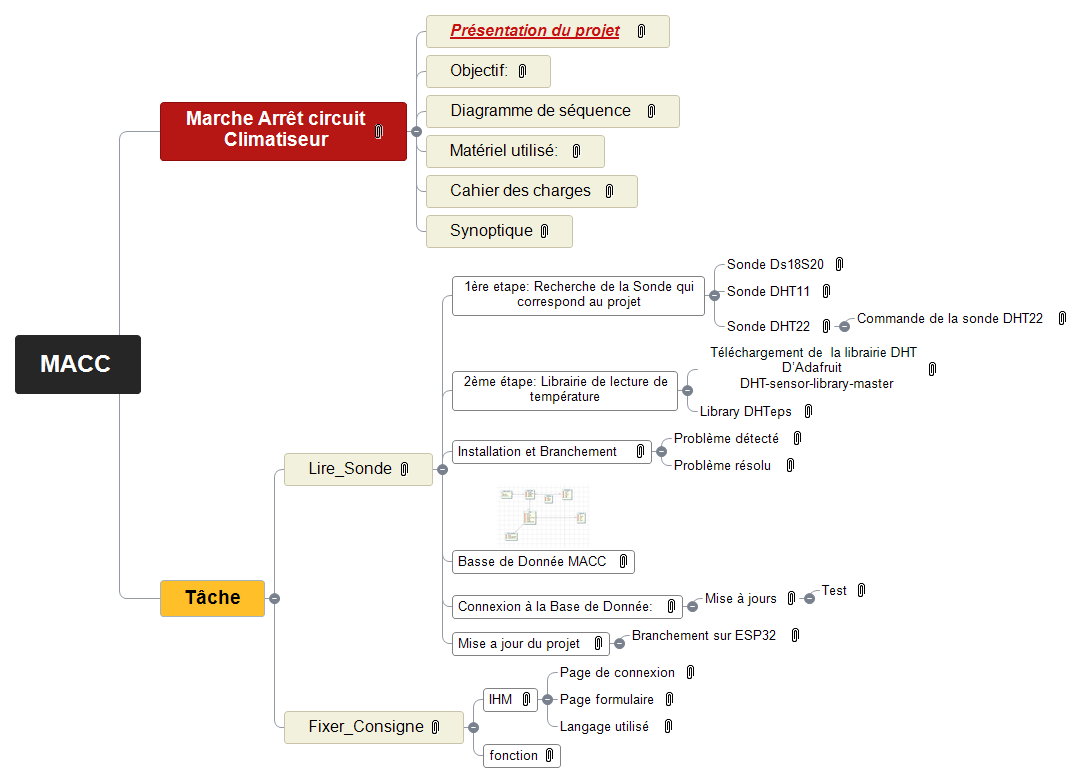
MACC



[Marche Arrêt circuit Climatiseur 4](#_Toc44000113)

[Présentation du projet 4](#_Toc44000114)

[Objectif: 5](#_Toc44000115)

[Diagramme de séquence 5](#_Toc44000116)

[Matériel utilisé: 5](#_Toc44000117)

[Cahier des charges 6](#_Toc44000118)

[Synoptique 6](#_Toc44000119)

[Tâche 7](#_Toc44000120)

[Lire\_Sonde 7](#_Toc44000121)

[1ère etape: Recherche de la Sonde qui correspond au projet 8](#_Toc44000122)

[2ème étape: Librairie de lecture de température 11](#_Toc44000123)

[Installation et Branchement 12](#_Toc44000124)

[Basse de Donnée MACC 15](#_Toc44000125)

[Connexion à la Base de Donnée: 15](#_Toc44000126)

[Mise a jour du projet 20](#_Toc44000127)

[Fixer\_Consigne 21](#_Toc44000128)

[IHM 21](#_Toc44000129)

[fonction 24](#_Toc44000130)

Outils utilés

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Version | Environnement | Rôle |
| IDE Arduino : | 1.8.4 | Système d’exploitation :  Windows 10 | espace de développement intégré (EDI) qui me permet d'écrire, de compiler et d'envoyer du code. |
| Notepad++ |  |  | Editeur de code source |
| Git/GitLab |  |  | Suivi des versions |
| EasyPHP DevServer |  |  |  |
| Putty |  |  |  |
| Microsoft office PowerPoint 2013 |  |  | Logiciel de création de diaporamas qui permettra de presenter le travail |
|  |  |  |  |

# Marche Arrêt circuit Climatiseur

N° Projet : CSG1

Nom du projet: Marche Arrêt Circuit Climatiseurs

Abréviation du nom projet: MACC

Nombre de personne dans le projet : 4 étudiants

* Etudiant1: Enregistrer IR, Fixer Consigne « horaire »
* Etudiant2: Lire Sonde, Fixer Consigne « température »
* Etudiant3: Rejouer IR, Visualiser température
* Etudiant4: Rejouer Ir

## Présentation du projet

Ce système à vocation à être utilisé dans un établissement (hôtels, bureaux, écoles…) qui dispose de nombreuses climatisations individuelles disséminées dans des pièces éparses.

L’objectif de ce système est de réduire les coûts énergétiques attribué à l’oublie de l’arrêt par le personnel des systèmes de climatisation.

Un allumage ou une extinction distant sous contrainte, horaire journalier, température ambiante, ou par opérateur direct permettra une diminution sensible du coût des factures énergétiques de l’établissement concerné.

Une commande par smartphone connecté permettra en étant dans la salle concernée de s’affranchir des télécommandes des constructeurs et une commande à distance par le superviseur pour fixer les consignes.

Je suis l'étudiante 2 (Lire Sonde, Fixer consigne « température »)

Dans cette partie du projet Lire Sonde, je dois lire la température et l’humidité de la pièce ou se situe le capteur de température.

Ensuite après avoir lu la température et l’humidité, je dois envoyer ces données dans la base de données 4 fois par heures, c’est-à-dire toutes les 15 minutes il y aura une nouvelle insertion dans la BDD.

Pour fixer consigne: partie superviseur, je dois créer une IHM (interface homme machine) qui permettra à l’utilisateur d’inscrire la température minimale à ne pas dépasser.

Mais avant de pouvoir inscrire ces données l’utilisateur devra s’identifier en mettant son login et son mot de passe.

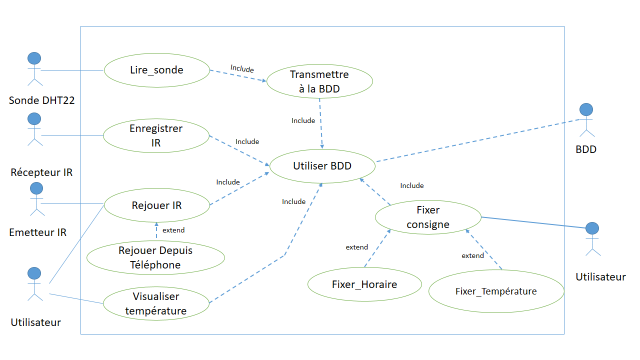
## Objectif:

-Réduction des coûts énergétiques

-gestion automatique des climatiseurs en cas d’oublie

-Gestion à distance

## Diagramme de séquence

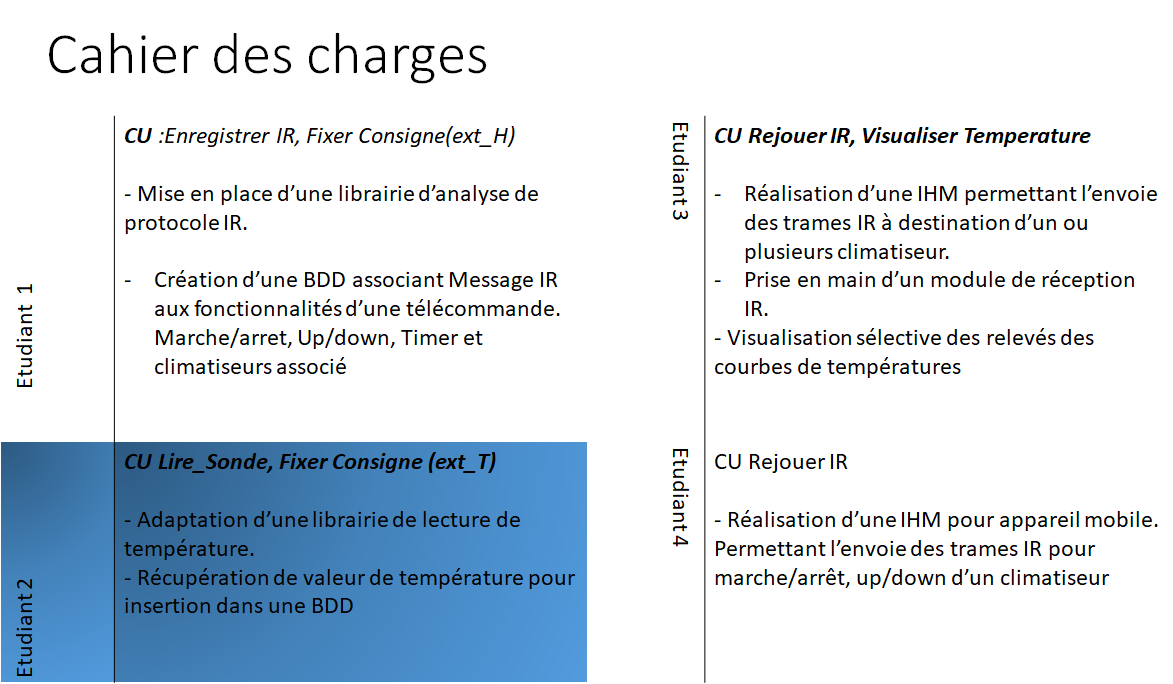


## Matériels utilisés:

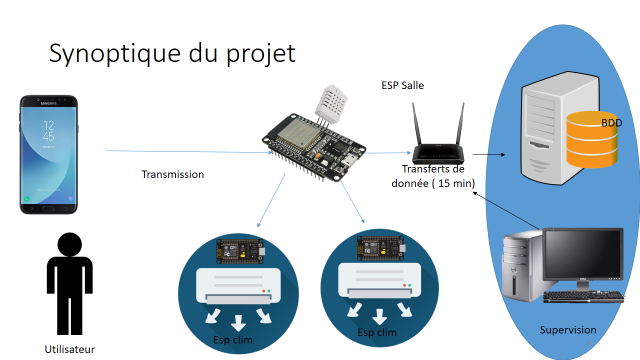
Pour réaliser ce projet nous avons utilisé :

|  |
| --- |
| 2 ESP8266 nodemcu 1.0 |
| 1 ESP32 |
| 1 Led infrarouge (résistance de 180 Ω) |
| récepteur ir |
| Sonde DHT22 (AM2302) |
| Ordinateur |
| Téléphone |
| 1 Raspberry pi (pour la base de données) |

## Cahier des charges



## Synoptique



# Tâche

Dans cette partie je vais vous présenter mon travail.

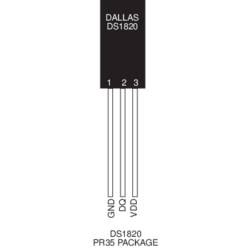
## Lire\_Sonde

Le principe de Lire\_Sonde, est de lire la température et l’humidité d'une pièce, et d'envoyer les données récolté toutes les 15 minutes dans la base de données.

### 1ère etape: Recherche de la Sonde qui correspond au projet

#### Sonde Ds18S20

Après des recherches à la maison sur les différentes sondes « capteur de température ».

J’ai trouvé le capteur de température DS18S20, Ainsi que la librairie 1-wire

C’est caractéristique :

L’interface 1-Wire nécessite seulement une broche de port pour communiquer.

Chaque composant possède un numéro de série unique sur 64 bits (18446 millions de milliards !) gravé en ROM.

Il ne nécessite aucun composant supplémentaire.

Peut être alimenté par la ligne de données. Plage d’alimentation de 3 à 5,5V.

Étendue de mesure de températures de –55°C à +125°C.

Précision de +/- 0,5 °C sur la plage –10°C à +85°C.

Thermomètre de résolution 9bits.

Temps de conversion maximal de 0,75 secondes.

Seuils d’alarme définissables par l’utilisateur et stockés en mémoire non volatile.

Commande de recherche de dépassement des alarmes qui identifie et adresse les composants dont la température excède les limites d’alarmes préprogrammées.

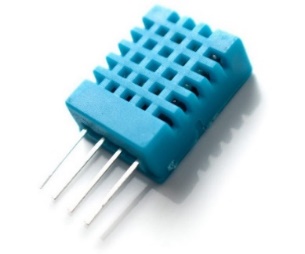
Applications telles que : contrôle thermostatique, systèmes industriels, thermomètre ou tout système sensible à la température.

Mais cette sonde est un capteur de température seulement il ne mesure pas l’humidité.

#### Sonde DHT11

|  |  |
| --- | --- |
|  | **DHT11** |
| Humidité (relative %) | 20 ~ 80% |
| Précision (humidité) | +/- 5% |
| Température | 0 ~ +50°C |
| Précision (température) | +/- 2°C |
| Fréquence mesure max | 1Hz (1 mesure par seconde) |
| Tension d'alimentation | 3 ~ 5 volts |
| Stabilité à long terme | +/- 1% par an |

**La sonde DHT11**

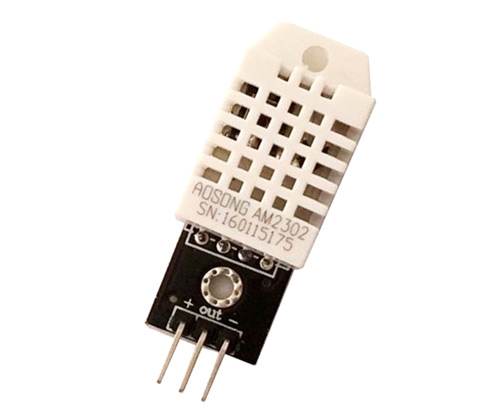


|  |
| --- |
| Le capteur DHT11 est lui capable de mesurer des températures de 0 à +50°C avec une précision de +/  2°C et des taux d'humidité relative de 20 à 80% avec une précision de +/  5%. Une mesure peut être réalisée toutes les secondes. |

#### Sonde DHT22 module

Cependant en voyant le petit frère du DHT11, le DHT22, j’ai fait une comparaison des caractéristiques :

|  |  |  |
| --- | --- | --- |
|  | **DHT22 / AM2302** | **DHT11** |
| Humidité (relative %) | 0 ~ 100 % | 20 ~ 80% |
| Précision (humidité) | +/- 2% (+/- 5% aux extrêmes) | +/- 5% |
| Température | -40 ~ +150°C | 0 ~ +50°C |
| Précision (température) | +/- 0.5°C | +/- 2°C |
| Fréquence mesure max | 2Hz (2 mesures par seconde) | 1Hz (1 mesure par seconde) |
| Tension d'alimentation | 3 ~ 5 volts | 3 ~ 5 volts |
| Stabilité à long terme | +/- 0.5% par an | +/- 1% par an |



Et après consultation du groupe j’ai décidé de prendre le DHT22.

Car le DHT22 est clairement beaucoup plus précis et stable que le DHT11.

Mais il est aussi deux fois plus cher.

Le choix se résume donc à une question de balance entre budget, précision et rapidité de mesure.

Ce capteur dans son écrin de plastique blanc intègre :

-Un capteur capteur de température basé sur une thermistance NTC (à coefficient thermique négatif).

-Un capteur d’humidité capacitif utilisant une capacité polymer

- un micro-contrôleur de 8 bits pour lire et communiquer de manière digitale les résultats des mesures sur un seul pin (DATA).

Il y a aussi une puce très basique à l'intérieur qui fait une conversion analogique-numérique et crache un signal numérique avec la température et l'humidité. Le signal numérique est assez facile à lire en utilisant n'importe quel microcontrôleur.

### Protocole de communication du capteur

Le capteur possède quatre pins et n'en utilise que 3 :

* **VCC**
* **DATA**
* NULL
* **GND**

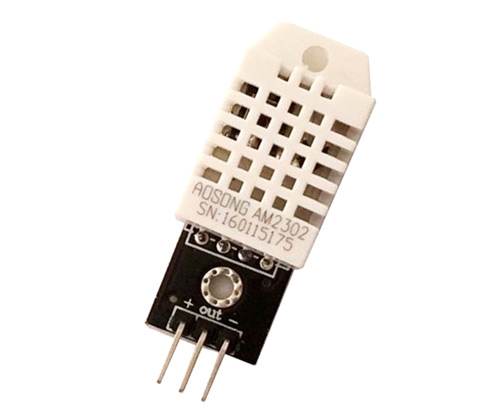
Le capteur communique avec son hôte via le seul pin data et les données sont envoyées "MSB first" (bit de poids fort) sous le format :

|  |
| --- |
| ***Data format:*** |
| 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. |
| If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data". |
| *Example: MCU has received 40 bits data from RHT03 as* |
| ***0000 0010 1000 1100 0000 0001 0101 1111 1110 1110*** |
| *16 bits RH data 16 bits T data check sum* |
| *Check sum=0000 0010+1000 1100+0000 0001+0101 1111=1110 1110* |
| *RH= (0000 0010 1000 1100)/10=65.2%RH* |
| *T=(0000 0001 0101 1111)/10=35.1* |
| *Lorsque le bit de température le plus élevé est 1, cela signifie que la température est inférieure à 0 degré Celsius.* |
| *Example:* |
| ***1****000 0000 0110 0101, T= minus 10.1 DONC -10 ,1°* |
| *16 bits T data* |
|  |

Le protocole de communication entre le capteur et l'hôte tel que décrit dans la datasheet se déroule comme ceci :

1. Le capteur est par defaut en mode "veille" et se reveil lorsque l'utilisateur tire la PIN data, qui doit possèder une resistance de "pull-up", à la masse pendant au moins 1ms.
2. Le capteur Tire la pin DATA à 0 durant 80us puis à 1 durant 80us ce qui signifie qu'il est pret à emetre les données.
3. Les 40 bits constituant la trame sont envoyés en commencant à chaque fois par tirer la pin DATA à 0 durant 50us puis en relachant la pin DATA à 1 durant 26 à 28 us si le bit vaut 0 ou 70 us si le bit a transmettre vaut 1.

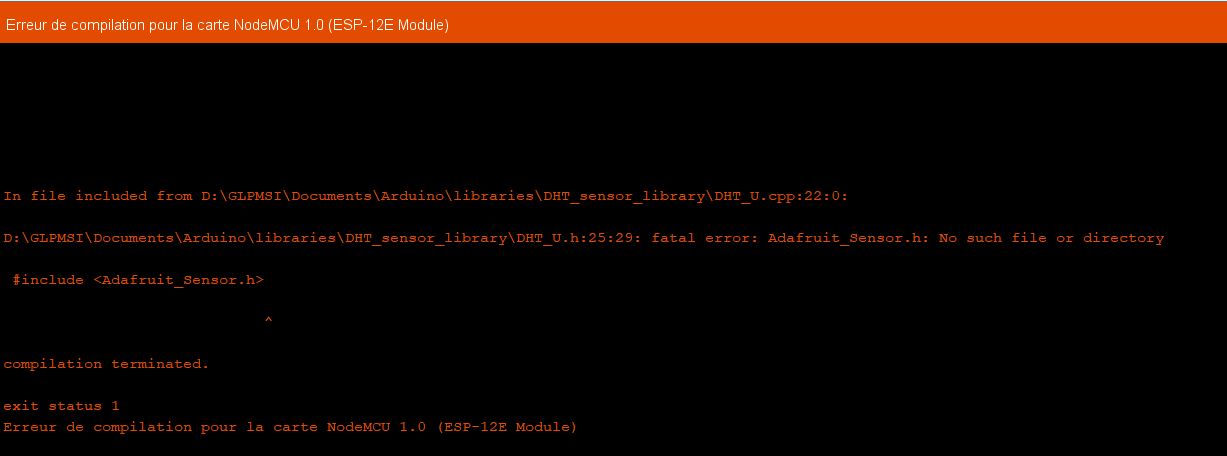
##### Commande de la sonde DHT22



### 2ème étape: Librairie de lecture de température

Je devais trouver une librairie qui me permettra de lire température et humidité

#### Téléchargement de la librairie DHT D’Adafruit DHT-sensor-library-master



Erreur la librairie n'est pas compatible avec l'AVR de l'esp

#### Library DHTeps

2ème librairie DHTEPS

J’ai effectué des tests

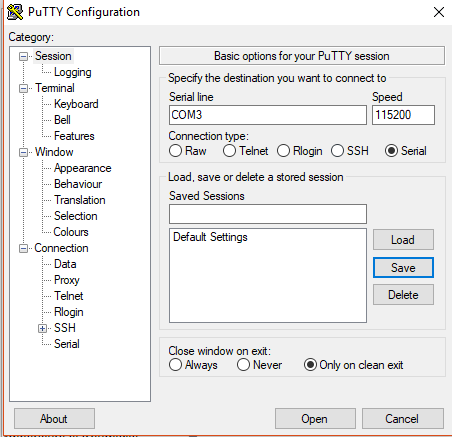
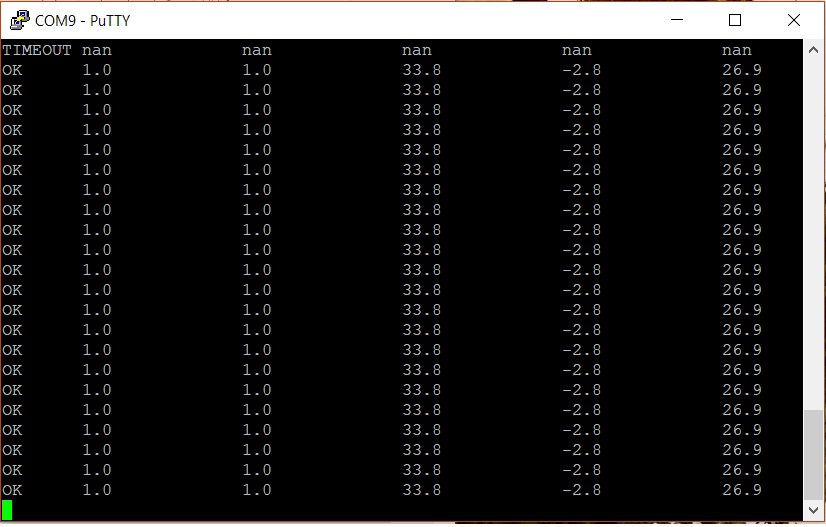
Je m’attendais à ce que les résultats soit compréhensible de tous.

Test de la library sur putty avec une vitesse de 115200 bauds (1 bit/s).

Qu’est-ce que putty ?

PuTTY est un émulateur de terminal doublé d'un client pour les protocoles SSH, Telnet, rlogin, et TCP brut. Il permet également des connexions directes par liaison série RS-232.

Résultat :

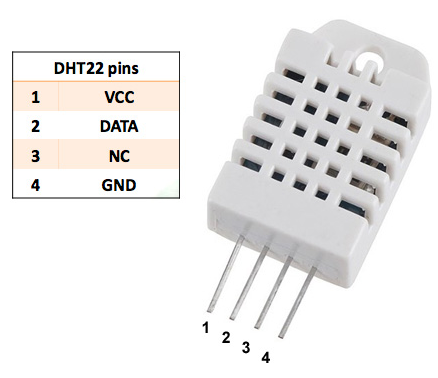
 

Le résultat attendu n’était pas celui que j’attendais.

### Installation et Branchement

Installation et branchement de l'esp8266 et de la sonde DHT22 :

|  |
| --- |
| Sur ce branchement vous pouvez voir qu’il n’y a pas de résistance, mais en réalité un condensateur de filtrage et une résistance de pull-up intégrés |

Branchement :

1( 3,3 sur le 3,3 ESP8266)

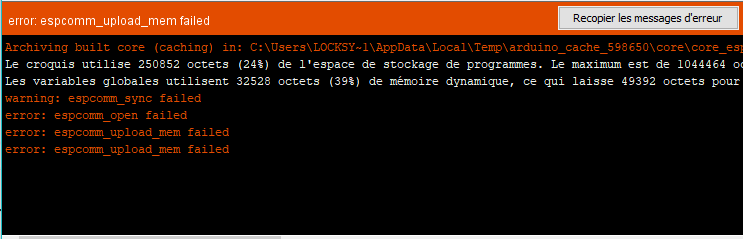
2 (Data sur GPIO4 (2) ESP8266)

3 (GND sur GND ESP8266)

#### Problème détecté

Après avoir effectué les tests sur l'IDE arduino (moniteur série) et putty j'ai eu des difficultés pour avoir les données du capteur de température et d’humidité DHT22

MSG:





#### Problème résolu

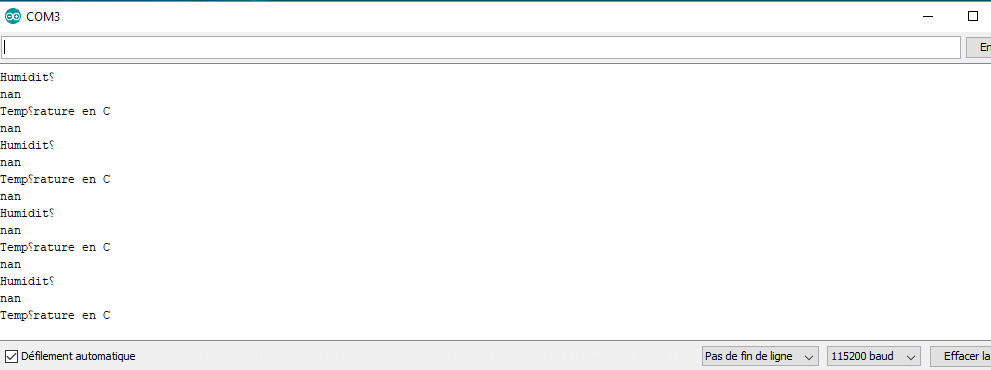
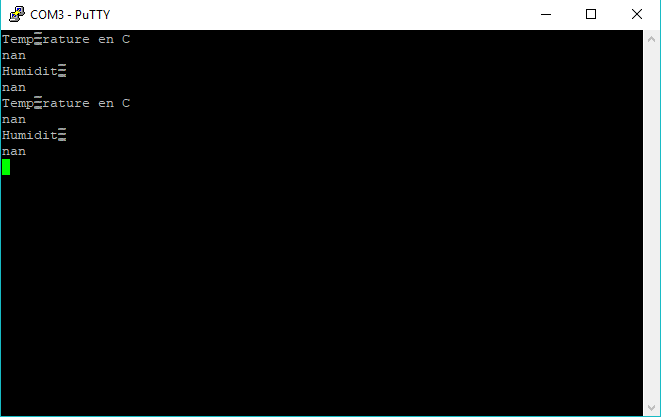
Pour m’éviter de perdre du temps j'ai décidé de codé moi-même le code sur le site tuniot for esp8266:

Avec la libraire: DHT SENSOR LIBRAIRY

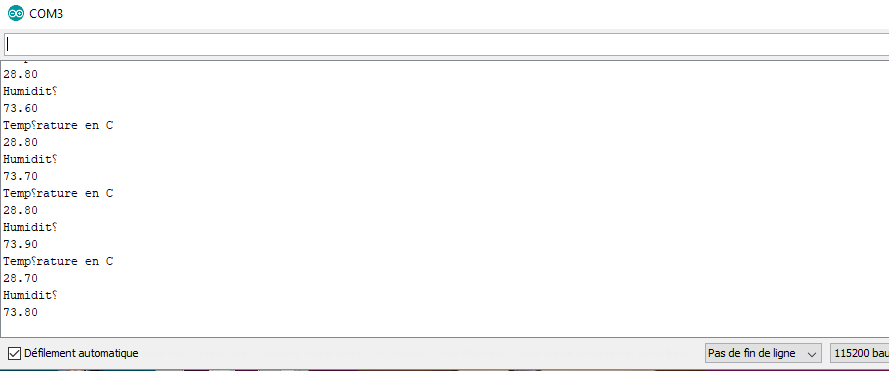
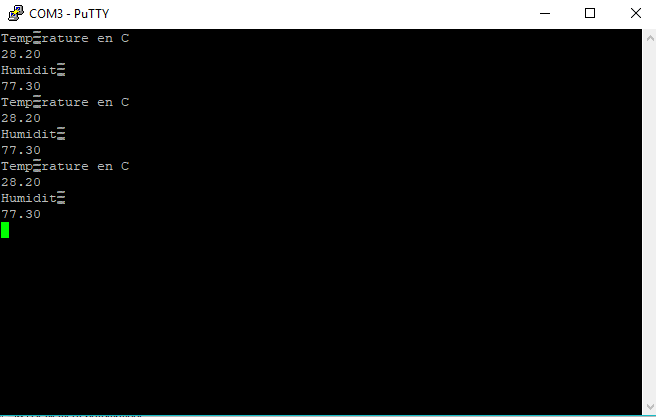
Et pour le code se présente comme ceci :

|  |  |
| --- | --- |
| *Intégrer la librairie DHT* | #include "DHT.h" |
|  |  |
| *Précision de la sonde et du pin utilisé* | DHT dht2(2,DHT22); |
|  | void setup() |
|  | { |
| *La vitesse en BAUDS (1bit/S)* | Serial.begin(115200); |
|  | } |
|  | void loop(){ |
|  | Temperature(); |
|  | } |
|  | void Temperature(){ |
|  | Serial.println("Temperature en C:"); |
| *Lecture de la température* | Serial.println((dht2.readTemperature())); |
|  | Serial.println("Temperature en F:"); |
| *Lecture de la température en fahrenheit* | Serial.println((dht2.readTemperature(true))); |
|  | Serial.println("Humidite:"); |
| *Lecture de l’humidité* | Serial.println((dht2.readHumidity( ))); |
| *Exécuter la boucle toute les 15 mins* | delay(900000); //15 min) |
| *Fin de la boucle* | } |

Résultat sur le moniteur série et sur putty :

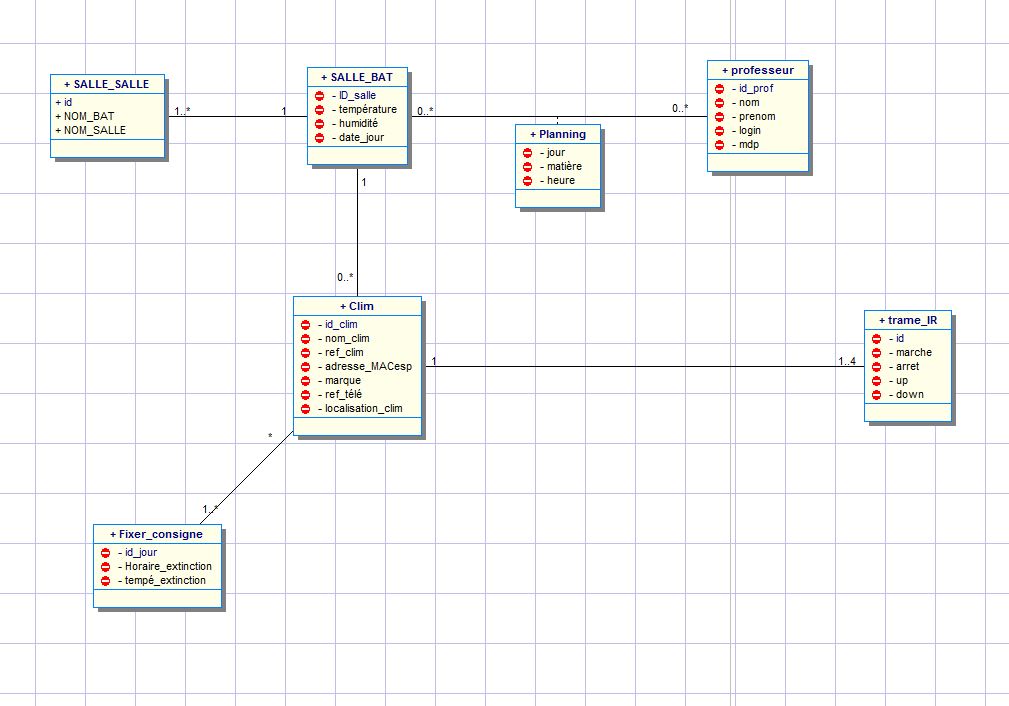
 

Pour résoudre le problème pour "nan" j'ai dû débrancher et attendre 10 A 20 Sec et puis brancher :

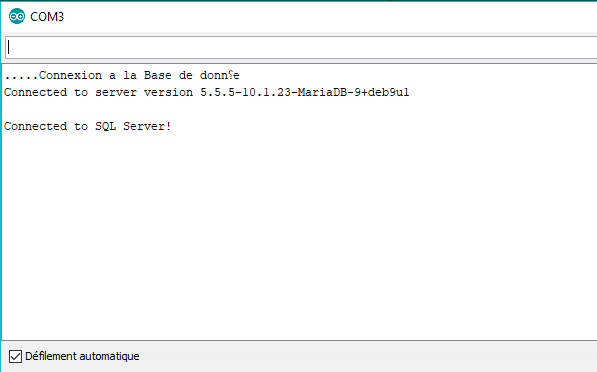
 

Pour la partir insertion des données dans la base de données, j’utilise comme table la table salle\_bat

### Basse de Donnée MACC

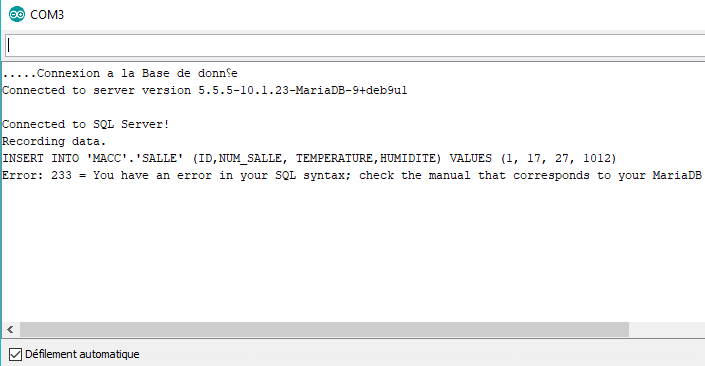


### Connexion à la Base de Donnée:



|  |
| --- |
| Code : |
|  |
| char ssidWIFI[] = "Jenni"; // Network Name |
| char passwordWIFI[] = "25111998"; // Network Password |
| char nomHOST[] = "MACC IR"; |
| WiFiClient client; |
| MySQL\_Connection conn((Client \*)&client); |
|  |
| char INSERT\_SQL[] = "INSERT INTO 'MACC'.'SALLE' (ID,NUM\_SALLE, TEMPERATURE,HUMIDITE) VALUES (1, 17, 27, %d)"; |
| char query[128]; |
|  |
| IPAddress server\_addr(93,121,229,118); // MySQL server IP |
| char user[] = "pi"; // MySQL user |
| char password[] = "Simconolat"; // MySQL password |
|  |
| void setup() { |
|  |
| Serial.begin(9600); |
|  |
| pinMode(sensorPin1, INPUT); |
| //pinMode(sensorPin2, INPUT); |
|  |
| WiFi.mode(WIFI\_STA); |
| WiFi.begin(ssidWIFI, passwordWIFI); |
| WiFi.hostname(nomHOST); |
|  |
| while (WiFi.status() != WL\_CONNECTED) { |
| delay(500); |
| Serial.print("."); |
| } |
| Serial.println("Connexion a la Base de donnée"); |
|  |
| while (conn.connect(server\_addr, 3306, user, password) != true) { |
| delay(200); |
| Serial.print ( "." ); |
| } |
| Serial.println(""); |
| Serial.println("Connected to SQL Server!"); |
|  |
| } |
| void loop() { |
|  |
| int soil\_hum = 1024 - analogRead(sensorPin1); |
| //float t = dht.readTemperature(); |
|  |
| //Serial.println(t); |
| delay(1000000); //10 sec |
|  |
| sprintf(query, INSERT\_SQL, soil\_hum); |
| Serial.println("Recording data."); |
| Serial.println(query); |
|  |
| MySQL\_Cursor \*cur\_mem = new MySQL\_Cursor(&conn); |
|  |
| cur\_mem->execute(query); |
|  |
| delete cur\_mem; |
|  |
| } |

Insertion a la base:



Voir l’annexe

#### Mise à jours

|  |
| --- |
| #include "DHT.h" |
| #include <esp\_event.h> |
| #include <esp\_event\_loop.h> |
| #include <MySQL\_Connection.h> |
| #include <MySQL\_Cursor.h> |
| #include <WiFi.h> |
|  |
| //\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| // ici on utilise la broche IO2 de ESP32 pour lire les données |
| #define DHTPIN 2 |
| // notre capteur est DHT22 type |
| #define DHTTYPE DHT22 |
| // crée une instance de |
| DHT dht(2, DHT22); |
| //\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |
| const char\* ssidWIFI = "DESKTOP-GLPMSI";// nom du réseau // connexion wifi |
| const char\* pWIFI = "97197170"; // Mot de passe du réseau |
| const char\* nomHOST = "MACC IR"; |
|  |
| IPAddress server\_addr(192,168,13,91); // IP of the MySQL \*server\* here |
| char user[] = "pi"; // MySQL user login username |
| char password[] = "Simconolat"; // MySQL user login password |
|  |
| WiFiClient client; |
| MySQL\_Connection conn((Client \*)&client); |
|  |
| //sample query |
| char INSERT\_SQL[] = " USE `MACC`; INSERT INTO `MACC`.`SALLE\_BAT` (`ID\_SALLE`,`NOM\_BAT`, `TEMPERATURE`, `HUMIDITE`) VALUES ( '14','BTV', '%f', '%f')"; |
| char query[256]; |
| char temperature[10]; |
| char humidity[10]; |
|  |
| void setup() { |
| Serial.begin(115200); |
| Serial.println ("Hello je suis le capteur DHT22!" ); |
| // appel à commencer à démarrer le capteur |
| dht.begin (); |
|  |
| // Connect to WiFi network |
| Serial.println(); |
| Serial.println(); |
| Serial.print("Connecting to "); |
| Serial.println(ssidWIFI); |
|  |
| pinMode(2, INPUT); |
| //pinMode(sensorPin2, INPUT); |
|  |
| WiFi.mode(WIFI\_STA); |
| WiFi.begin(ssidWIFI, pWIFI); |
| WiFi.setHostname(nomHOST); |
|  |
| while (WiFi.status() != WL\_CONNECTED) { |
| delay(500); |
| Serial.println("."); |
| } |
| Serial.println(""); |
| Serial.println("WiFi connected"); |
| Serial.println("DB - Connecting..."); |
|  |
| //Attente de connexion a la BDD |
| while(conn.connect(server\_addr, 3306, user, password)!= true) { |
| delay(200); |
| Serial.println ( "." ); |
| } |
| } |
| void loop() { |
|  |
| // utilise les fonctions fournies par la bibliothèque. |
| float h = dht.readHumidity (); |
| // Lit la température comme Celsius (la valeur par défaut) |
| float t = dht.readTemperature (); |
| // Vérifie si les lectures ont échoué et quitte tôt (pour réessayer). |
| if (isnan (h) || isnan (t)) { |
| Serial.println ( "Impossible de lire le capteur DHT!" ); |
| delay(1000); |
| return ; |
| } |
| // afficher le résultat dans Terminal |
| Serial.print ( "Humidite:" ); |
| Serial.print (h); |
| Serial.print ( "% \ " ); |
| Serial.print ( "Temperature:" ); |
| Serial.print (t); |
| Serial.println ( " \*C" ); |
| delay(1000); |
|  |
| // create the DHT22 query |
| sprintf(query, INSERT\_SQL, t, h); |
|  |
| Serial.println("Recording data."); |
|  |
| // Initiate the query class instance |
| MySQL\_Cursor \*cur\_mem = new MySQL\_Cursor(&conn); |
| // Execute the query |
| cur\_mem->execute(query); |
| delete cur\_mem; |
| Serial.println("DHT22 data recorded"); |
|  |
| } |

Ce code me permet de me connecter au réseaux, a la base de données, lire les temperature et l’humidité et de faire l’insertion dans la base de données

##### Test

Compte-rendu détaillé du test

Etape1 : préparation

-Brancher esp32 à la sonde DHT22( VCC= 3,3v, DATA=pin2, GND=GND)

-Lancer ARDUINO IDE avec le code

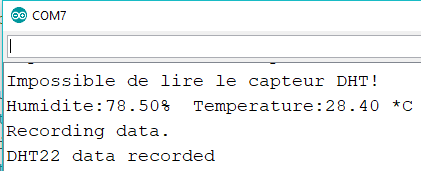
-Brancher l’esp a l’ordinateur

-Téléversser le code vers esp

Test OK

ETAPE 2 vérification de la lecture des données

-ouvrir le moniteur serie d’arduino IDE



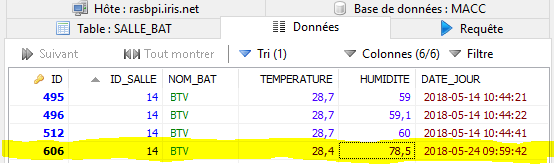
Test OK

Etape 3 vérification de l’insertion des données dans la BDD

-lancer HeidiSQL

-taper Nom ou IP de l’hôte, utilisateur, mot de passe, port

-Constater l’insertion des données



Test OK

### Mise a jour du projet

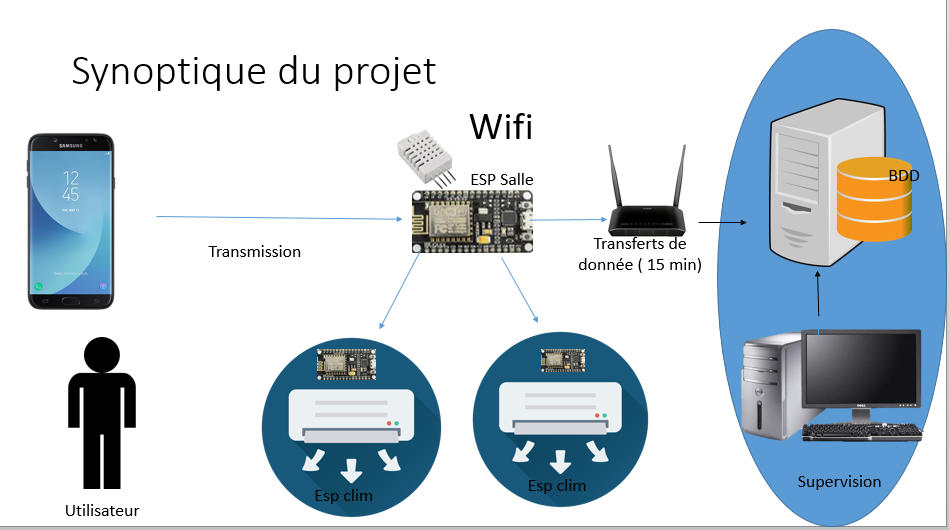
La synoptique tu projet a évolué, je travaillais sur l'esp8266, maintenant je travaille sur le esp32,

Qui a pour nom ESP\_SALLE

Quelle est les différences entre le 8266 et le 32 ?

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

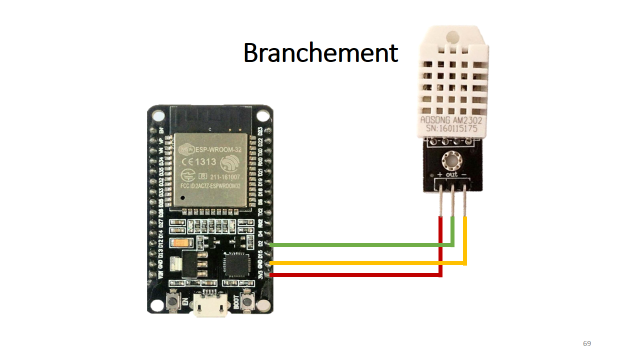
Avant la mise à jour mon capteur était sur chaque esp8266, mais pour une question de\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



Mon capteur DTH22 est placé sur l'ESP32 (ESPSALLE), donc il captera la température et l'humidité de la salle.

#### Branchement sur ESP32

|  |
| --- |
| VCC sur 3,3 v  DATA sur pin 2  GND SUR GND |



## Fixer\_Consigne

Objectif: Permettre à un utilisateur qui est inscrit dans la base de données, de fixer des consignes de température et d’horaire

### IHM

L'IHM est constitué d'une page d'authentification avec un login et un mot de passe

Ensuite si l'authentification est correcte l'utilisateur à un accès à un formulaire d'inscription pour entrer les valeurs de température, horaire d'allumage et d'extinction.

Une fois que ces informations son inscrit ils ont envoyé dans la base de donnée MACC.

Sur cette IHM il y a la possibilité de se déconnecter.

#### Page de connexion

Etape 1 : Préparation

Allumer le pc

Lancer EasyPhp

Ouvrir un navigateur

Tapez (http:// <http://127.0.0.1:8080/IHM/>)

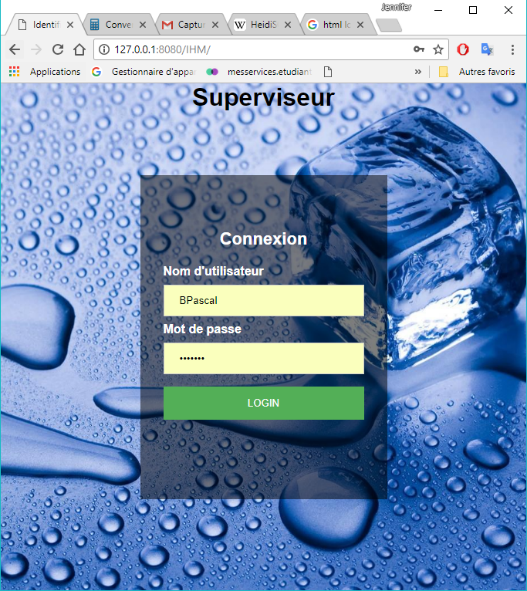
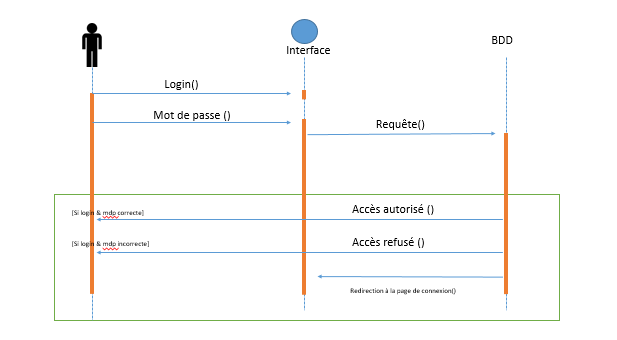
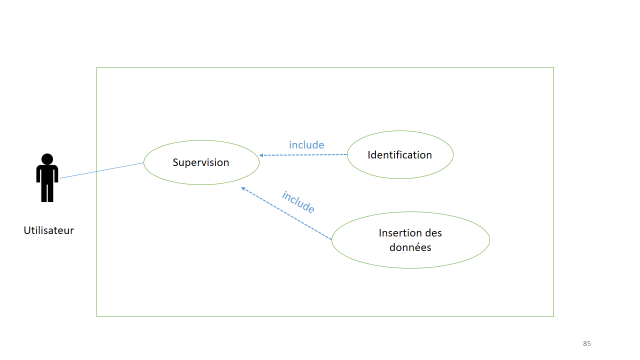


Diagramme de séquence :

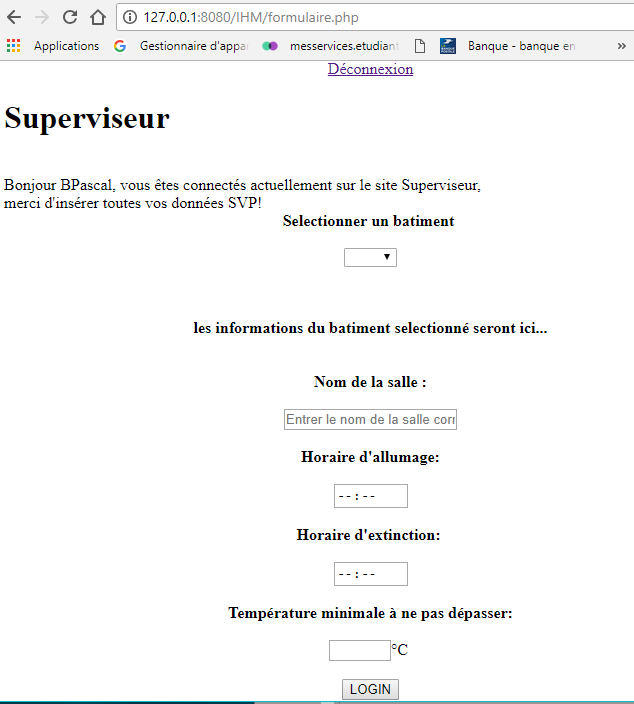


Cas d'utilisation diagramme



#### Page formulaire

Accès autriser l’utilisateur peut insert les données :



#### Langage utilisé



### fonction

J’ai créé une fonction qui \*\*\*\*\*

* (type) changerTemperature(id\_boutton, temperature\_actuelle){

if (id\_boutton == augmenter){

temperature\_actuelle = temperature\_actuelle + 1;

return temperature\_actuelle; //ou la fonction pour envoyer la nouvelle température}

if (id\_boutton == diminuer){

if(temperature\_actuelle == temperature\_min){

afficher('erreur'); }

if(temperature\_actuelle > temperature\_min){

temperature\_actuelle = temperature\_actuelle - 1;

return temperature\_actuelle; //ou la fonction pour envoyer la nouvelle température }}

# ANNEXE

#include <DHT.h>

#include <MySQL\_Connection.h>

#include <MySQL\_Cursor.h>

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#define sensorPin1 0

//#define sensorPin2 D2

//#define typeDHT DHT11

//DHT dht(sensorPin1, typeDHT);

char ssidWIFI[] = "Jenni"; // Network Name

char passwordWIFI[] = "25111998"; // Network Password

char nomHOST[] = "MACC IR";

WiFiClient client;

MySQL\_Connection conn((Client \*)&client);

char INSERT\_SQL[] = "INSERT INTO 'MACC'.'SALLE' (ID,NUM\_SALLE, TEMPERATURE,HUMIDITE) VALUES (1, 17, 27, %d)";

char query[128];

IPAddress server\_addr(93,121,229,118); // MySQL server IP

char user[] = "pi"; // MySQL user

char password[] = "Simconolat"; // MySQL password

void setup() {

Serial.begin(9600);

pinMode(sensorPin1, INPUT);

//pinMode(sensorPin2, INPUT);

WiFi.mode(WIFI\_STA);

WiFi.begin(ssidWIFI, passwordWIFI);

WiFi.hostname(nomHOST);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print("."); }

Serial.println("Connexion a la Base de donnée");

while (conn.connect(server\_addr, 3306, user, password) != true) {

delay(200);

Serial.print ( "." ); }

Serial.println("");

Serial.println("Connected to SQL Server!"); }

void loop() {

int soil\_hum = 1024 - analogRead(sensorPin1);

delay(1000000); //10 sec

sprintf(query, INSERT\_SQL, soil\_hum);

//sprintf(query, INSERT\_SQL, soil\_hum, t);

Serial.println("Recording data.");

Serial.println(query);

MySQL\_Cursor \*cur\_mem = new MySQL\_Cursor(&conn);

cur\_mem->execute(query);

delete cur\_mem;

}

Index.php

<html>

<head>

<title> Identification supervision </title>

<meta charset="utf-8">

<!-- importer le fichier de style -->

<link rel="stylesheet" href="CSS.css" type="text/css" />

</head>

<body>

<center><h1>Superviseur</h1></center>

<div id="container">

<!-- zone de connexion -->

<form action="verification.php" method="POST">

<center> <h1>Connexion</h1></center>

<label><b>Nom d'utilisateur</b></label>

<input type="text" placeholder="Entrer le nom d'utilisateur" name="username" required>

<label><b>Mot de passe</b></label>

<input type="password" placeholder="Entrer le mot de passe" name="password" required> <input type="submit" id='submit' value='LOGIN' >

<?php

if(isset($\_GET['erreur'])){

$err = $\_GET['erreur'];

if($err==1 || $err==2)

echo "<p style='color:red'>Utilisateur ou mot de passe incorrect</p>";

}

?>

</form>

</div>

</body>

</html>

Verification.php

<?php

session\_start();

if(isset($\_POST['username']) && isset($\_POST['password'])) {

// connexion à la base de données

$db\_user = 'pi';

$db\_password = 'Simconolat';

$db\_name = 'MACC';

// $db\_host = '192.168.13.91';

//$db\_host = '93.121.180.47';

$db\_host = '192.168.137.127'

$db = mysqli\_connect($db\_host, $db\_user, $db\_password,$db\_name)

or die('could not connect to database');

// on applique les deux fonctions mysqli\_real\_escape\_string et htmlspecialchars

// pour éliminer toute attaque de type injection SQL et XSS

$username = mysqli\_real\_escape\_string($db,htmlspecialchars($\_POST['username']));

$password = mysqli\_real\_escape\_string($db,htmlspecialchars($\_POST['password']));

if($username !== "" && $password !== "")

{

$requete = "SELECT count(\*) FROM PROFESSEUR where

LOGIN = '".$username."' and MDP = '".$password."' ";

$exec\_requete = mysqli\_query($db,$requete);

$reponse = mysqli\_fetch\_array($exec\_requete);

$count = $reponse['count(\*)'];

if($count!=0) // nom d'utilisateur et mot de passe correctes

{

$\_SESSION['username'] = $username;

header('Location: formulaire.php');

}

else {

header('Location: index.php?erreur=1'); // utilisateur ou mot de passe incorrect

} }

else {

header('Location: index.php?erreur=2'); // utilisateur ou mot de passe vide

}}

else{

header('Location: index.php');}

mysqli\_close($db); // fermer la connexion

?>

CSS.css

|  |
| --- |
| /\* Set a style for all buttons \*/  input[type=submit] {  background-color: #53af57;  color: white;  padding: 14px 20px;  margin: 8px 0;  border: none;  cursor: pointer;  width: 100%;  }  input[type=submit]:hover {  background-color: white;  color: #53af57;  border: 1px solid #53af57;  } |

body{

margin: 0;

padding: 0;

background: url(glaçon.jpg);

background-size: cover;

background-position: center;

font-family: sans-serif;

}

#container{

width:400px;

margin:0 auto;

margin-top:10%;

}

/\* Bordered form \*/

form {

width: 320px;

height: 420px;

background: rgba(0, 0, 0, 0.5);

color: #fff;

top: 50%;

left: 50%;

position: absolute;

transform: translate(-50%,-50%);

box-sizing: border-box;

padding: 70px 30px;

}

#container h1{

margin: 0;

padding: 0 0 20px;

text-align: center;

font-size: 22px;

}

/\* Full-width inputs \*/

input[type=text], input[type=password] {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

box-sizing: border-box;

Formulaire.php

<html>

<head>

<meta charset="utf-8">

<script>

/\*lorsqu'un utilisateur sélectionne une personne dans la liste

déroulante ci-dessus, une fonction appelée "showUser ()" est exécutée.

La fonction est déclenchée par l'événement onchange

\*/

function showUser(str) {

if (str == "") {

document.getElementById("txtHint").innerHTML = "";

return;

} else {

if (window.XMLHttpRequest) {

// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp = new XMLHttpRequest();

} else {

// code for IE6, IE5

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("txtHint").innerHTML = this.responseText;

}

};

xmlhttp.open("GET","getuser.php?q="+str,true);

xmlhttp.send();

}}

</script>

</head>

<body background="green">

<center>

<a href='index.php?deconnexion=true'><span>Déconnexion</span></a>

</center>

<div id="container">

<h1>Superviseur </h1>

<!-- zone de connexion -->

<?php

session\_start();

if(isset($\_POST['deconnexion'])) {

if($\_POST['deconnexion']==true) {

session\_unset();

header("location:login.php");

} }

else if($\_SESSION['username'] !== ""){

$user = $\_SESSION['username'];

// afficher un message

echo "<br>Bonjour $user, vous êtes connectés actuellement sur le site Superviseur,<br> merci d'insérer toutes vos données SVP!"; }

?> <br>

<center>

<form method="post" action="getuser.php">

<label><b>Selectionner un batiment</b></label> <br><br>

<select name="sal" onchange="showUser(this.value)">

<option option>

<option >BatV</option>

</select><br> <br>

<br>

</form>

<form method="post" action="get.php">

<div id="txtHint"><b>les informations du batiment selectionné seront ici...</b></div>

<br><br>

<label><b> Nom de la salle :</b></label><br> </br>

<input type="text" placeholder="Entrer le nom de la salle correctement..." name="nom\_salle"><br></br>

<label><b>Horaire d'allumage:</b></label><br> </br>

<input type="time" name="allumage"><br></br>

<label><b>Horaire d'extinction: </b></label><br> </br>

<input type="time" name="extinction"><br> <br>

<label><b>Température minimale à ne pas dépasser:</b></label><br> </br>

<input type="number" min="20" max="30" name="temp" >°C

<br/> <br>

<input type="submit" id='submit' value='Valider' >

</form></center> </div>

</body>

</html>

Getuser.php

<!DOCTYPE html>

<html>

<head>

<style>

table {

width: 100%;

border-collapse: collapse;

}

table, td, th {

border: 1px solid black;

padding: 5px;

}

th {text-align: left;}

</style>

</head>

<body>

<?php

//$con = mysqli\_connect('192.168.13.91','pi','Simconolat','MACC');

$con = mysqli\_connect('93.121.180.47','pi','Simconolat','MACC');

if (!$con) {

die('Could not connect: ' . mysqli\_error($con));}

mysqli\_select\_db($con,"");

$sql="SELECT \* FROM SALLE\_SALLE ";

$result = mysqli\_query($con,$sql);

echo "

</option>

<table>

<tr>

<th>NOM\_SALLE</th>

</tr>";

while($row = mysqli\_fetch\_array($result))

echo "<td>" . $row['NOM\_SALLE'] . "</td>";

echo "</tr>";}

echo "</table>";

mysqli\_close($con);

?>

</body>

</html>

Get.php

<?php

//on recupère les valeurs du formulaire

$allumage=$\_POST['allumage'];

$extinction=$\_POST['extinction'];

$temp=$\_POST['temp'];

$nom\_salle=$\_POST['nom\_salle'];

// connexion à la base de données

$db\_user = 'pi';

$db\_password = 'Simconolat';

$db\_name = 'MACC';

// $db\_host = '192.168.13.91';

$db\_host = '93.121.180.47';

$db = mysqli\_connect($db\_host, $db\_user, $db\_password,$db\_name)

or die("Connection failed: " . mysqli\_connect\_error());

$sql = "INSERT INTO FIXER\_CONSIGNE (`HORAIRE\_ALLUMAGE`, `HORAIRE\_EXTINC`, `TEMP\_FIXE`, `SALLE`)

VALUES ('".$allumage."','".$extinction."','".$temp."','".$nom\_salle."')";

if (mysqli\_query($db, $sql)) {

echo "New record created successfully";

} else {

echo "Error: " . $sql . "<br>" . mysqli\_error($db);

}

header('Refresh:0;url=fin.html');

mysqli\_close($db);

?>