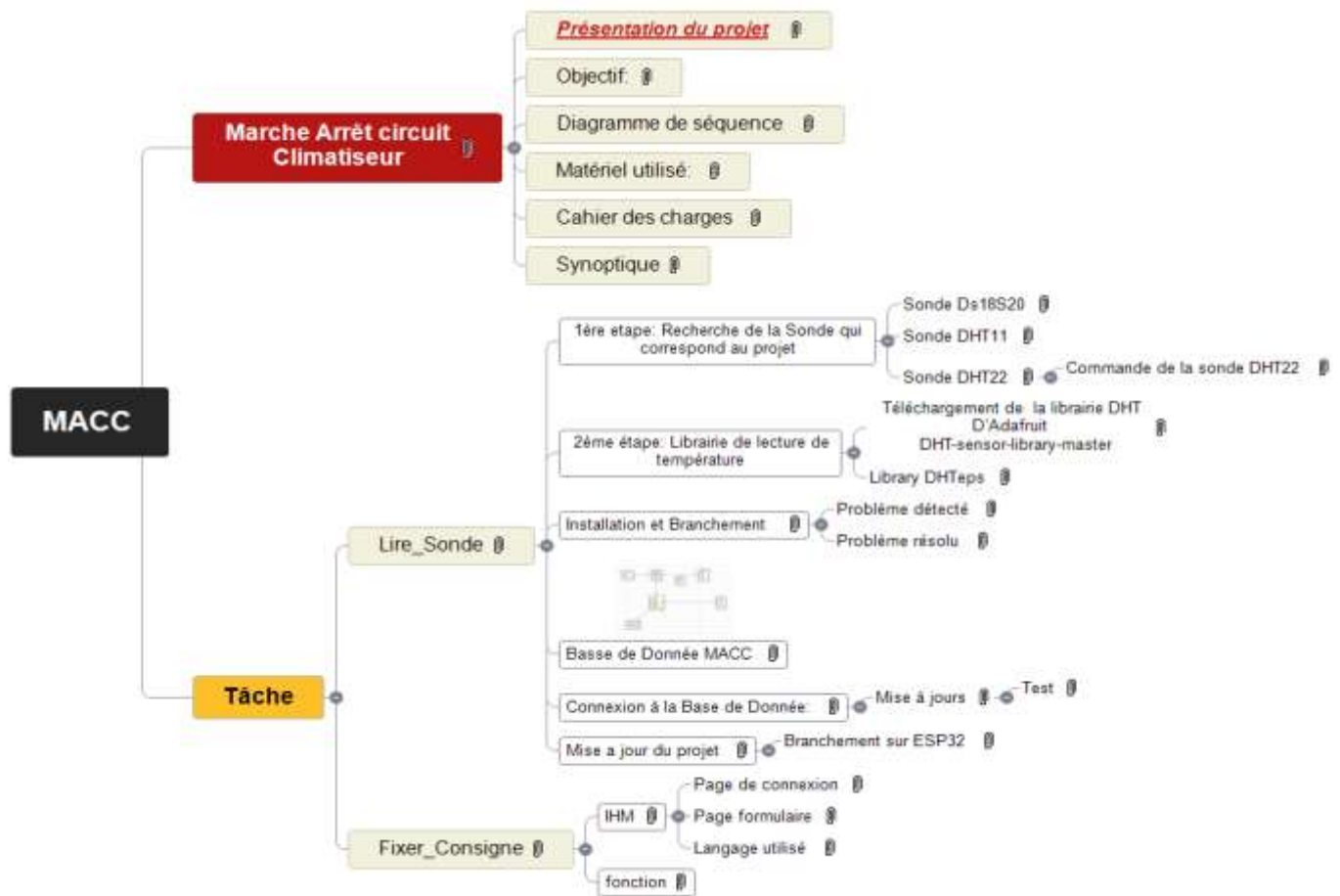


ETUDIANTE 2



1	TâcheS	3
1.1	Lire_Sonde.....	3
1.1.1	1ère etape: Recherche de la Sonde qui correspond au projet.....	3
1.1.2	Protocole de communication du capteur.....	5
1.1.3	2ème étape: Librairie de lecture de température.....	6
1.1.4	Installation et Branchement	7
1.1.5	Basse de Donnée MACC	9
1.1.6	Connexion à la Base de Donnée:	9
1.1.7	Mise a jour du projet	14
1.2	Fixer_Consigne	15
1.2.1	IHM	15
1.2.2	fonction	17
2	ANNEXE	18

1 TACHES

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 où 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.

1.1 LIRE_SONDE

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

1.1.1 1ERE ETAPE: RECHERCHE DE LA SONDE QUI CORRESPOND AU PROJET

1.1.1.1 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 $+125^{\circ}\text{C}$.

Précision de $\pm 0,5^{\circ}\text{C}$ sur la plage -10°C à $+85^{\circ}\text{C}$.

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.

1.1.1.2 SONDE DHT11

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.



	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

1.1.1.3 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 de température basé sur une thermistance NTC (à coefficient thermique négatif).
- Un capteur d'humidité capacitif utilisant une capacité polymère
- un microcontrôleur 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 donne un signal numérique avec la température et l'humidité. Le signal numérique est assez facile à lire en utilisant n'importe quel microcontrôleur.

1.1.2 PROTOCOLE DE COMMUNICATION DU CAPTEUR

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

- **VCC**
- **DATA**
- **NC**
- **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 intégral RH data + 8bit décimal RH data + 8bit intégral T data + 8bit décimal T data + 8bit check sum.

Si la transmission de données est correcte, la somme de contrôle doit être la dernière 8bit de 8bit intégral RH data + 8bit décimal RH data + 8bit intégral T data + 8bit décimal T data".

Exemple: MCU a reçu 40 bits de données du DHT22 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.

Exemple:

1000 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 défaut en mode "veille" et se réveille lorsque l'utilisateur tire la PIN data, qui doit posséder une résistance de "pull-up", à la masse pendant au moins 1ms.
2. Le capteur Tire le pin DATA à 0 durant 80us puis à 1 durant 80us ce qui signifie qu'il est prêt à émettre les données.
3. Les 40 bits constituant la trame sont envoyés en commençant à chaque fois par tirer le pin DATA à 0 durant 50us puis en relâchant le pin DATA à 1 durant 26 à 28 us si le bit vaut 0 ou 70 us si le bit à transmettre vaut 1.



1.1.4 INSTALLATION ET BRANCHEMENT

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



DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



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)

1.1.4.1 PROBLEME DETECTE

Après avoir effectué les tests sur l'IDE arduino (moniteur série) et putty j'ai eu des difficultés à téléverser le code sur l'ESP8266 et avoir les données du capteur de température et d'humidité DHT22

MSG:

```
error: espcomm_upload_mem failed
Archiving built core (caching) in: C:\Users\LOCKSY~1\AppData\Local\Temp\arduino_cache_598650\core\core_es
Le croquis utilise 250852 octets (24%) de l'espace de stockage de programmes. Le maximum est de 1044464 octets
Les variables globales utilisent 32528 octets (39%) de mémoire dynamique, ce qui laisse 49392 octets pour
warning: espcomm_sync failed
error: espcomm_open failed
error: espcomm_upload_mem failed
error: espcomm_upload_mem failed
error: espcomm_upload_mem failed
error: espcomm_upload_mem failed
error: espcomm_upload_mem failed
```

Pour régler le problème j'ai dû changer à plusieurs reprises d'ESP8266, téléverser sur un autre ordinateur avec la nouvelle version arduino IDE, mais j'avais toujours le même résultat.

1.1.4.2 PROBLEME RESOLU

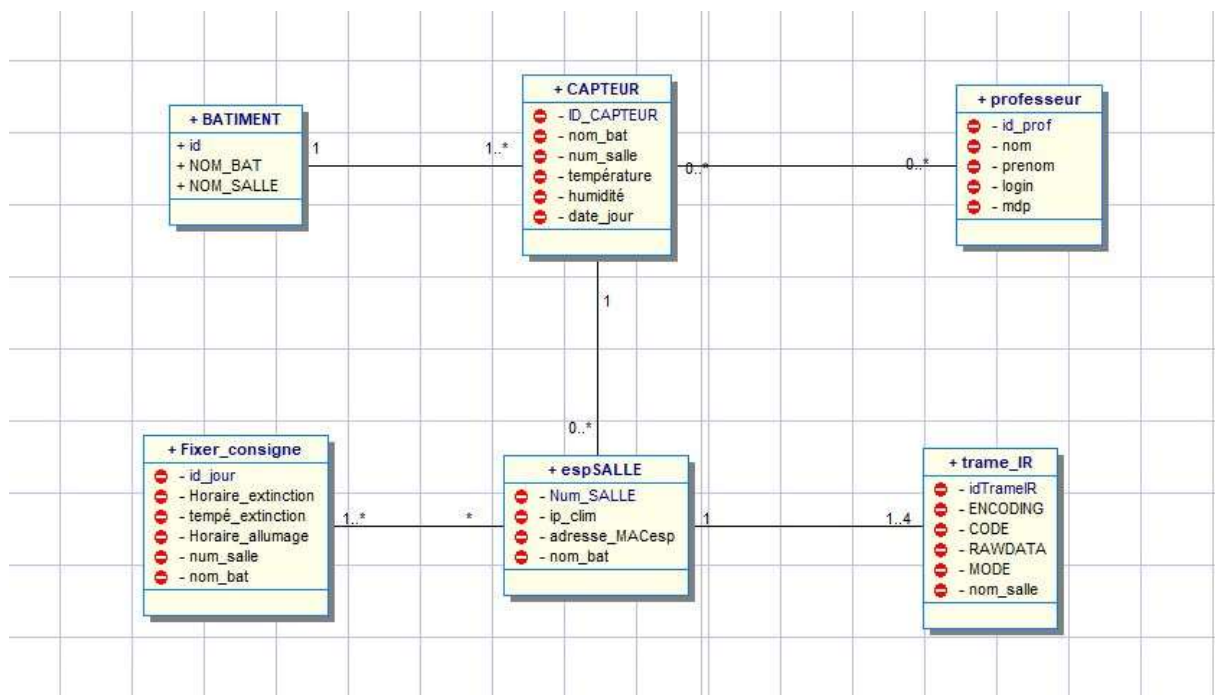
Pour m'éviter de perdre plus de temps j'ai décidé de coder moi-même le code sur le site [tuniot for esp8266](https://tuniot.fr/esp8266/):

Avec la librairie: DHT SENSOR LIBRARY.

Le code se présente comme ceci :

Pour la partir insertion des données dans la base de données, j'utilise comme table la table salle_bat

1.1.5 BASSE DE DONNEE MACC



1.1.6 CONNEXION A LA BASE DE DONNEE:

```
COM3
.....Connexion a la Base de donn e
Connected to server version 5.5.5-10.1.23-MariaDB-9+deb9ul
Connected to SQL Server!
```

Test de connexion à la base de données réussi.

```

Code :
char ssidWIFI[] = "Jenni";           // Nom du réseau
char passwordWIFI[] = "25111998";     // mot de passe du réseau
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:

```
COM3
.....Connexion a la Base de donnee
Connected to server version 5.5.5-10.1.23-MariaDB-9+deb9ul

Connected to SQL Server!
Recording data.
INSERT INTO `MACC`.`SALLE` (ID,SALLE, TEMPERATURE,HUMIDITE) VALUES (1, 17, 27, 1012)
Error: 233 = You have an error in your SQL syntax; check the manual that corresponds to your MariaDB
```

Problème d'insertion des données dans BDD (erreur de syntaxe)

J'ai modifié :

```
char INSERT_SQL[] = " USE `MACC`; INSERT INTO `MACC`.`SALLE_BAT` (`ID_SALLE`,`NOM_BAT`, `TEMPERATURE`,
`HUMIDITE`) VALUES ( '14','BTV', '%f', '%f')";
```

1.1.6.1 MISE A 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 ( "% \n " );
    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 la temperature et l'humidité et de faire l'insertion dans la base de données.

1.1.6.1.1 TEST

Compte-rendu détaillé du test

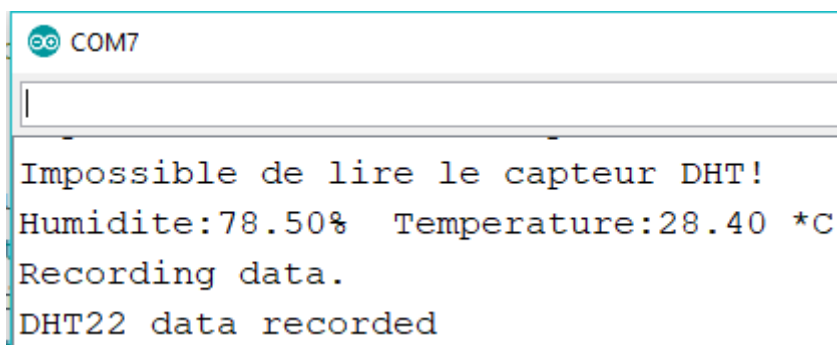
Etape1 : préparation

- 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
- Téléverser le code vers esp

Test OK

ETAPE 2 vérification de la lecture des données

- ouvrir le moniteur serie d'arduino IDE



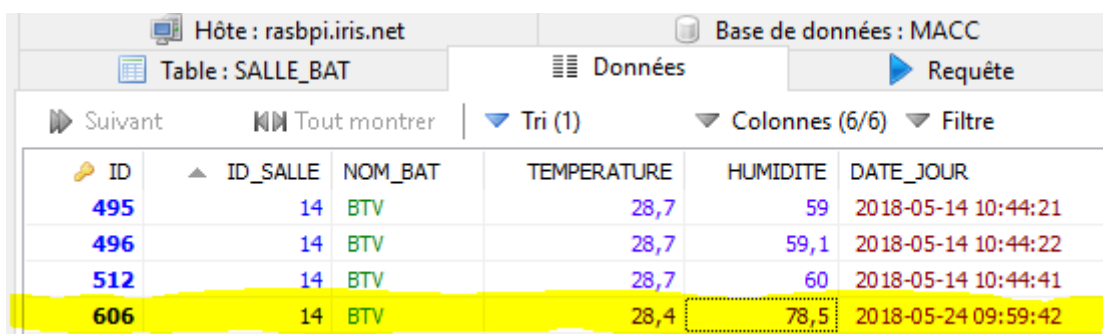
COM7

```
Impossible de lire le capteur DHT!  
Humidite:78.50%  Temperature:28.40 *C  
Recording data.  
DHT22 data recorded
```

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



Hôte : raspbpi.iris.net Base de données : MACC

Table : SALLES_BAT Données Requête

Suivant Tout montrer Tri (1) Colonnes (6/6) Filtre

ID	ID_SALLE	NOM_BAT	TEMPERATURE	HUMIDITE	DATE_JOUR
495	14	BTV	28,7	59	2018-05-14 10:44:21
496	14	BTV	28,7	59,1	2018-05-14 10:44:22
512	14	BTV	28,7	60	2018-05-14 10:44:41
606	14	BTV	28,4	78,5	2018-05-24 09:59:42

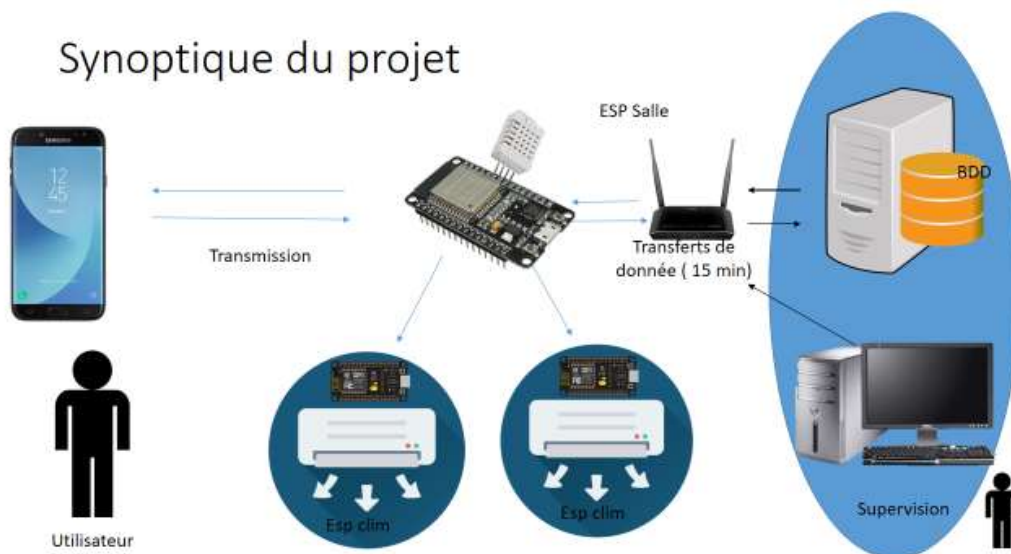
Test OK

1.1.7 MISE A JOUR DU PROJET

La synoptique du projet a évolué, je travaillais sur l'esp8266, maintenant je travaille sur le esp32, qui a pour nom ESP_SALLE.

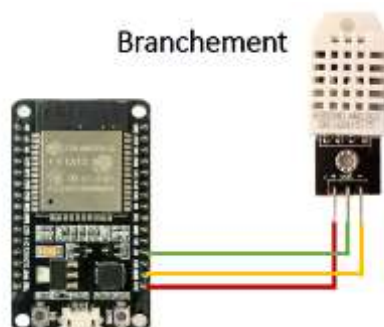
Avant la mise à jour mon capteur était sur chaque esp8266, mais pour une question de sécurité, parce que connecter sur le routeur n'importe pouvait contrôler le climatiseur étant connecté sur le mm réseau.

Et pour faciliter l'utilisation des climatiseurs parce qu'auparavant il fallait se connecter à chaque esp clim pour contrôler le climatiseur en question et se déconnecter pour contrôler l'autre



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

1.1.7.1 BRANCHEMENT SUR ESP32



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

Pour passer de l'esp8266 à l'esp32

J'ai dû intégrer des Library spécifique à l'esp32

- `#include <esp_event.h>` // Spécifique à l'ESP
- `#include <esp_event_loop.h>` // Spécifique à l'ESP

Conclusion la partie Lire_Sonde fonctionne bien.

1.2 FIXER_CONSIGNE

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

1.2.1 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 a un accès à un formulaire d'inscription pour entrer les valeurs de température, horaire d'allumage et d'extinction.

Une fois que ces informations sont inscrites, ils sont envoyés dans la base de données MACC.

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

1.2.1.1 PAGE DE CONNEXION

Etape 1 : Préparation

Allumer le pc

Lancer EasyPhp

Ouvrir un navigateur

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

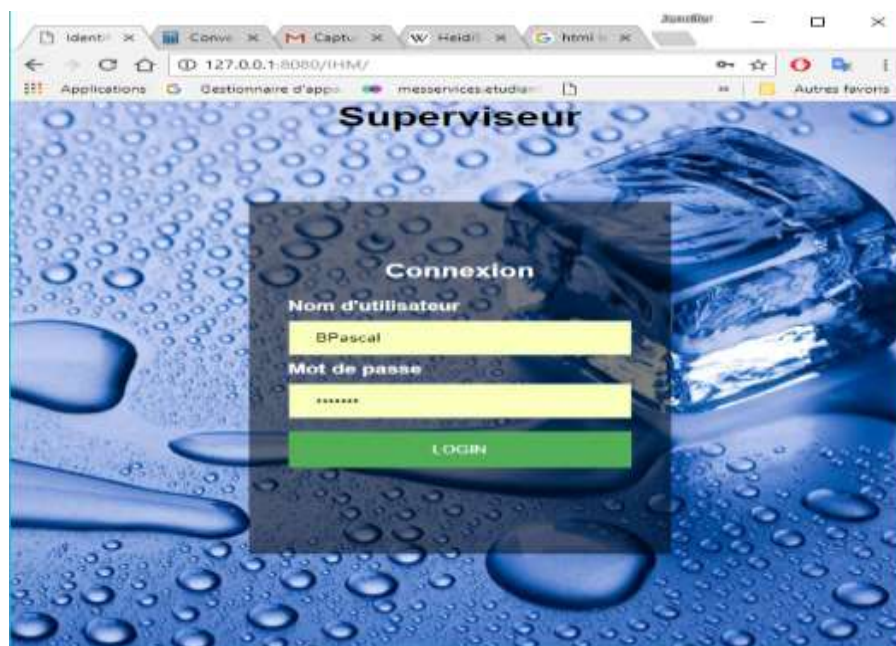
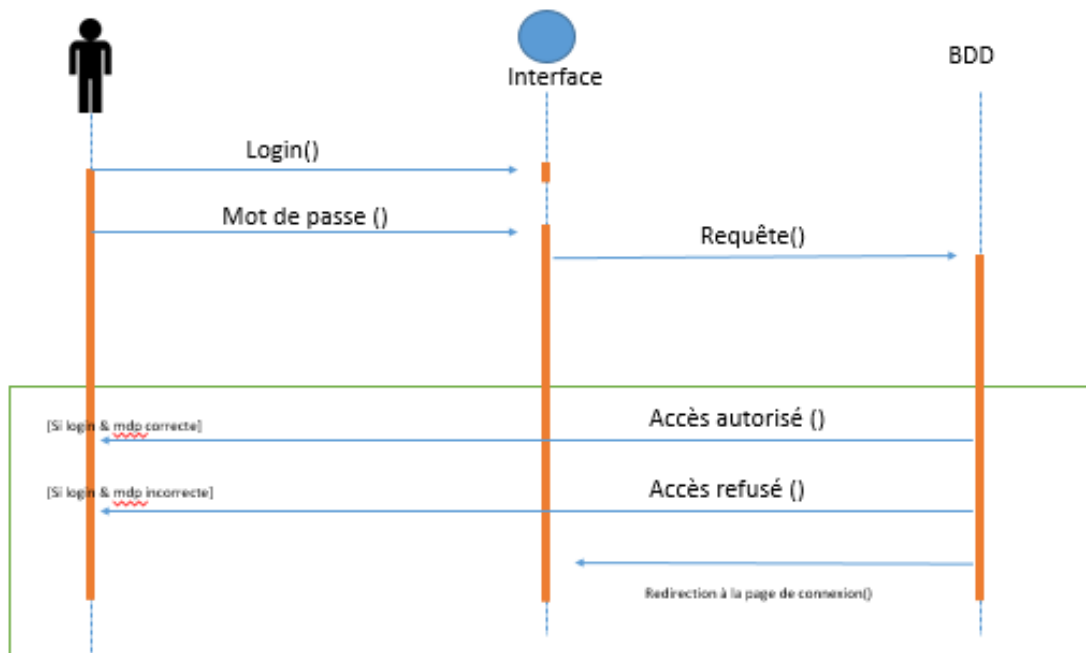
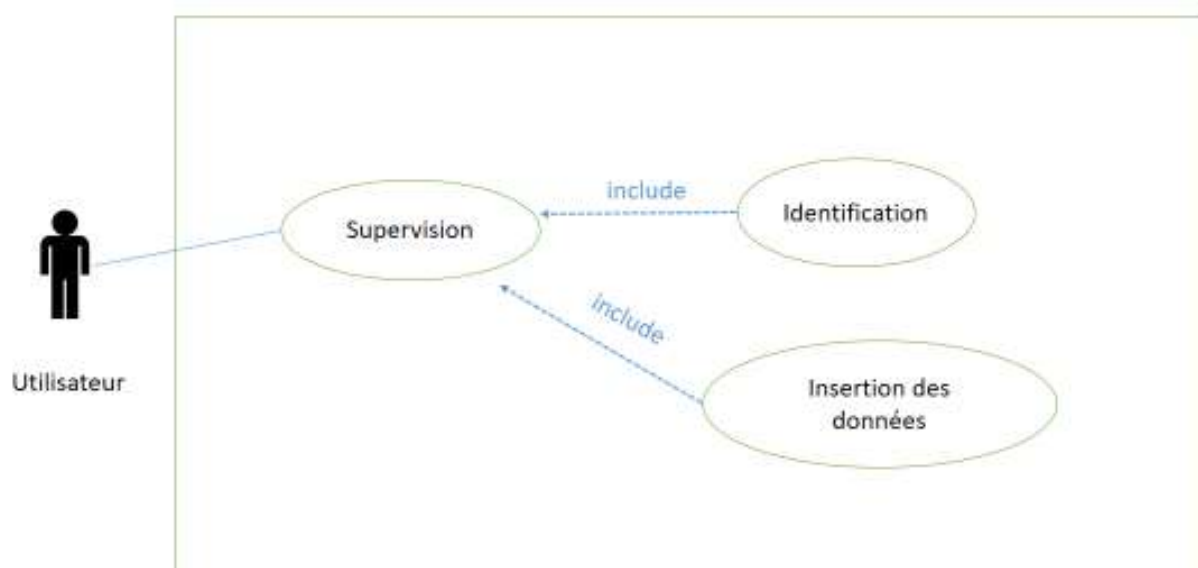


Diagramme de séquence :



Cas d'utilisation diagramme



Le but du superviseur est d'insérer des valeurs qui iront dans la bdd pour fixer consigne, donc si l'utilisateur se connecte et qui a accès à la page, c'est bien pour insérer des valeurs.

1.2.1.2 PAGE FORMULAIRE

Accès autorisé l'utilisateur peut insérer les données :

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8080/IHM/formulaire.php'. The browser's address bar also shows 'Applications', 'Gestionnaire d'applications', 'messervices.etudiant', and 'Banque - banque'. A 'Déconnexion' link is visible in the top right corner. The main content of the page is titled 'Superviseur' and includes a welcome message: 'Bonjour BPascal, vous êtes connectés actuellement sur le site Superviseur, merci d'insérer toutes vos données SVP!'. Below this, there is a section titled 'Selectionner un batiment' with a dropdown menu. A message below the dropdown states 'les informations du batiment selectionné seront ici...'. The form then contains several input fields: 'Nom de la salle :' with a text input field containing 'Entrer le nom de la salle con'; 'Horaire d'allumage:' with a time selection field showing '11:00'; 'Horaire d'extinction:' with a time selection field showing '18:00'; and 'Température minimale à ne pas dépasser:' with a temperature input field showing '22°C'.

Langage utilisé pour la conception de l'IHM : PHP, HTM et CSS

1.2.2 FONCTION

J'ai créé une fonction qui récupère la dernière insertion de fixer consigne et qui fonctionne comme ceci

La température fixée est de 22 degré

Si température est supérieur a 22° on peut augmenter ou baisser la température

Sinon impossible de de faire descendre la température

2 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
?>
```

```

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;

/* 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;
}

```

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);
?>
```