

IOM PARTIE 1: SENS CLIENT-SERVEUR

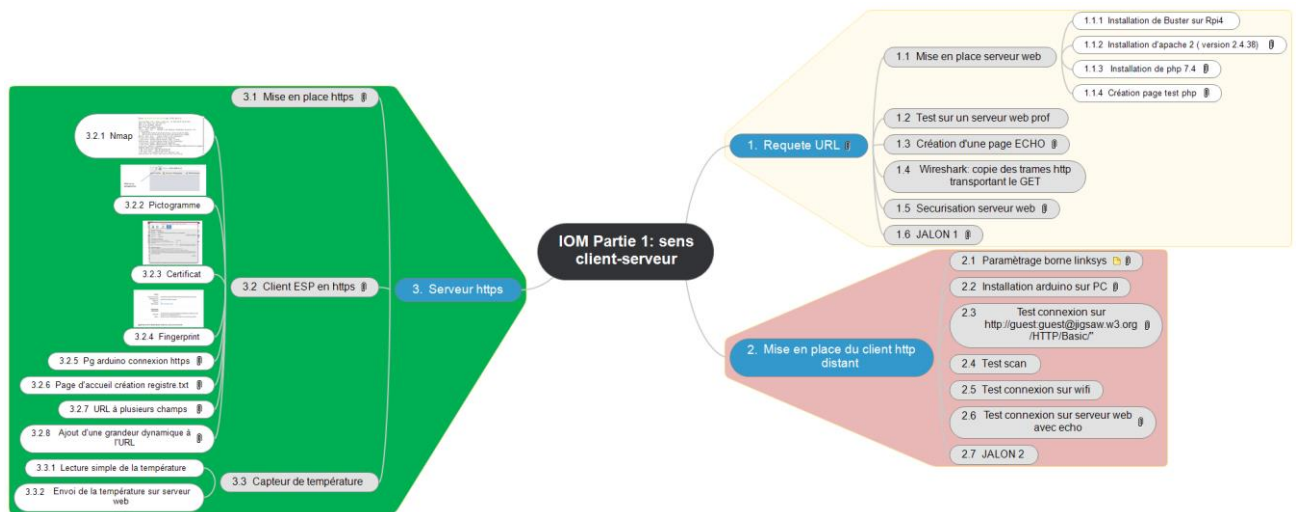


Figure 1 :carte mentale du projet IOM

1 REQUETE URL

Matériel:

Rpi4+ carte Sd+Alim+clavier+souris

2 PC fixes

1.1 MISE EN PLACE DES PC FIXES

Vous prendrez 2 PC par groupe (trinome) qui seront restaurés avant utilisation.
Les PC seront sous Debian 10, le login est **tp** le mot de passe **tp**

1.2 MISE EN PLACE DE LA BORNE LINKSYS

La borne linksys doit proposer un point d'accès wifi qui pourra permettre à un ESP de se connecter en wifi et aux 2 PC et au RPI de se connecter en filaire.

- Faites un Reset de la borne
- Connecter le pc à la borne (sur les ports switch de la borne)
- Mettez-en en place le nouveau mot de passe
- Nommer le ssid **binome_x** avec x votre numéro de binome.
- Pour le wifi, mettez la sécurité en mode **wpa2 personal**
- Pour le wifi, mettez le mot de passe **tpRT9025**
- Connectez la borne au portail captif (via le connecteur Ethernet de la borne)
- Sur le PC fixe, se connecter à <http://www.perdu.com> pour valider le portail captif

1.3 MISE EN PLACE SERVEUR WEB

1.3.1 INSTALLATION DE BUSTER SUR RPI4

Voir [TP_fiche3_mise_en_place_RPi4](#) mais prendre version **Buster du 4 avril 2022**
Utilisez le torrent

Aide : Pour Décompresser un fichier xz <code>xz -d monfichier.xz</code>
--

- Connectez votre rpi sur votre borne. L'écran HDMI du second ordinateur et le clavier et la souris seront utilisés de manière transitoire.
- Activez le service ssh
- Testez la connexion à internet. Vérifiez que vous n'avez plus besoin de vous identifier sur le portail captif.

1.3.2 INSTALLATION D'APACHE 2 (VERSION 2.4.38)

voir Fiche 4 mise_en_place_serveur_web_Apache
Mettez en place Apache 2 sur le RPI

Aide :

Fichier de configuration dans `/etc/apache2/apache2.conf`

Fichier de configuration dans `/etc/php/7.4/apache2/php.ini`

Pour qu'Apache démarre en même temps que debian

=> `systemctl enable apache`

Pour redémarrer Apache

=> `systemctl restart apache2`

1.3.3 INSTALLATION DE PHP 7.4

Installez php sur le Rpi

```
sudo apt install php php-mbstring
```

1.3.4 CREATION PAGE TEST PHP

Mettez dans `var/www/html` le fichier ***phpinfo.php*** qui permet de renvoyer l'ensemble des infos sur le serveur.

Voici les lignes qui composent ce fichier.

```
<?php
phpinfo();
?>
```

1.4 TEST SUR UN SERVEUR WEB

Depuis un PC fixe, connectez-vous à la page ***phpinfo.php*** du serveur.

Notez les versions d'Apache et de php

1.5 CREATION D'UNE PAGE ECHO

On souhaite qu'un PC distant nous transmette des informations via l'URL comme dans l'URL suivante (192.168.1.x représente l'adresse IP de notre serveur web) :

```
http://192.168.1.x/echo.php?string=1
```

Cette URL signifie que vous passez à la page `echo.php`, la variable « string » à 1
La réponse de serveur à cette requête du client sera :

```
You said 1
```

La création de la page echo.php utilisera la méthode GET

Mettez dans var/www/html le fichier *echo.php*

L'expression `$_GET["string"]` sera utilisée pour récupérer le contenu de la variable "string" et le renvoyer à la requête http via la syntaxe.

You said: `<?php echo $_GET["string"] ?>`

Testez que votre URL permet de renvoyer la valeur de variable passée dans l'URL.

1.6 WIRESHARK: COPIE DES TRAMES HTTP TRANSPORTANT LE GET

Mettez en place Wireshark sur le PC debian 10 en vous basant sur la fiche ressource 3 : TP_fiche10_installation_Wireshark.pdf

1.7 INFORMATION DONNEES PAR UN SERVEUR WEB

Vérifier qu'en mettant un mauvais fichier dans l'URL, le serveur nous délivre des infos sur ses caractéristiques

1.8 JALON 1

- Donner les Versions de:
 - Debian=> `cat /etc/debian-version`
 - Apache=> `apt-cache policy apache2`
 - Php=> `php -v`
- Copie écran de la réponse page test php
- Copie écran page html réponse echo
- Copie des trames acquises par wireshark des trames concernant la requête GET (utiliser les filtres et enregistrer uniquement les données pertinentes).
- Copie d'écran retour serveur web lors d'une requête client erronée.

2 MISE EN PLACE DU CLIENT HTTP DISTANT

2.1 INSTALLATION ARDUINO SUR PC

On désire installer arduino version 1.8

```
sudo apt-get update
```

Téléchargez la version 64 bits (à adapter) :

```
cd Téléchargements
wget -c http://downloads.arduino.cc/arduino-1.8.15-linux64.tar.xz
```

Décompressez l'archive et l'installez à l'endroit voulu :

```
tar -xJf arduino-1.8.15-linux64.tar.xz
sudo mv arduino-1.8.15 /usr/share/arduino
```

Téléchargez l'icône arduino et l'installez à l'endroit voulu :

```
wget http://entropie.org/3615/download/arduino.png
sudo mv arduino.png /usr/share/arduino/
```

Pour intégrer le programme au menu de Xfce, créez un fichier `arduino.desktop` dans le répertoire `/usr/share/applications` :

```
cd /usr/share/applications
sudo nano arduino.desktop
```

```
[Desktop Entry]
Type=Application
Name=Arduino
Comment=Un environnement de développement pour Arduino
GenericName=Create physical computing projects
Exec=/usr/share/arduino/arduino
Icon=/usr/share/arduino/arduino.png
Terminal=false
Categories=Development;Engineering;Electronics;
MimeType=text/x-arduino
Keywords=embedded electronics;electronics;avr;microcontroller;
```

Pour que le port série soit accessible et éviter de devoir utiliser Arduino en mode administrateur :

```
sudo usermod -a -G tty $USER ($USER à adapter)
sudo usermod -a -G dialout $USER ($USER à adapter)
```

Redémarrer.

Arduino est maintenant installé dans application/programmation

2.2 PARAMETRAGE D'ARDUINO POUR L'ESP8266

Voir fiche ressource 4 : `Tp_fiche8_mise_en_place_ESP_8266_lolin.pdf`

- Dans le menu « fichier » => « préférences », vérifier que l'URL de gestion des cartes est bien http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Dans le menu « outils » => « type de carte » => « gestionnaire de cartes »=>rechercher les cartes ESP=> installer le paquet esp8266
- Connecter l'ESP8266 sur le port USB, ouvrir l'IDE Arduino.
- Dans le menu « Outils » => « port », choisir le port de connexion.

- Dans le menu « outils » => « type de carte » => choisir « NodeMCU 1.0 (ESP-12E module) »
- Faire une Capture d'écran du menu Préférences onglet Paramètres d'Arduino

2.3 TEST SCAN

- Lancer le fichier wifiscan.ino que vous trouverez dans Exemple=> ESP8266Wifi=>wifiScan
- Téléversez le programme dans l'Esp et ouvrez le terminal série afin de voir les hotspots wifi.
- Copier les données reçues sur le port série lorsque wifiClientBasic.ino est exécuté sur l'ESP par exemple dans un fichier appelé Wi-Fi.txt

2.4 TEST CONNEXION SUR WIFI

- Lancez le fichier wifiClientBasic.ino que vous trouverez dans Exemple=> ESP8266Wifi=>wifiClientBasic
- Modifiez :
 - Le nom du SSID
 - Le mot de passe de votre wifi
 - Supprimez le contenu du loop et enlevez les variables inutiles
 - le débit du port série à 115200 kbits/s
- Téléversez le programme dans l'Esp et ouvrez le terminal série afin de noter l'adresse IP attribuée par votre wifi.
- Faire l'acquisition en mode texte des données reçues sur le port série lorsque wifiClientBasic.ino est exécuté sur l'ESP

2.5 TEST CONNEXION SUR SERVEUR WEB AVEC ECHO

- Lancez le fichier BasicHttpClient.ino que vous trouverez dans Exemple=> ESP8266HttpClient=> BasicHTTPClient.ino

Ce programme se connecte à votre wifi et envoie une requête vers :

<http://jigsaw.w3.org/HTTP/connection.html>

- Modifiez :
 - Le nom du SSID
 - Le mot de passe de votre wifi
 - La requête http afin qu'elle se connecte à la page web echo.php de votre serveur pour envoyer string=1 (voir création d'une page ECHO au chapitre 1.5)
- Téléversez le programme dans l'Esp et ouvrez le terminal série afin de noter

- la réponse renvoyée par votre serveur web lorsqu'il reçoit votre requête.
- Faire l'acquisition en mode texte des données reçues sur le port série lorsque BasicHttpClient.ino est exécuté sur l'ESP

2.6 JALON 2

Ce jalon (qui sera transmis en pdf) est la compilation des preuves suivantes

- Capture d'écran du menu préférence d'Arduino
- Texte des données reçues sur le port série lorsque wifiScan.ino est exécuté sur l'ESP
- Texte des données reçues sur le port série lorsque wifiClientBasic.ino est exécuté sur l'ESP
- Texte des données reçues sur le port série lorsque BasicHttpClient.ino est exécuté sur l'ESP.

3 SERVEUR HTTPS

3.1 MISE EN PLACE HTTPS

- Mettre APACHE EN HTTPS :
- Suivre les instructions données sur le lien ci-dessous pour mettre en place le https

<https://techexpert.tips/fr/apache-fr/activer-https-sur-apache/>

Aide : Se mettre en sudo

- Depuis un PC fixe, connectez-vous en https avec la requête `https://192.168.1.121/echo.php?string=1`, vérifiez que vous recevez bien la réponse attendue.
- Faire une copie d'écran de l'URL en https et de sa réponse.
- Faire une capture Wireshark qui montre que l'URL passe en clair malgré le fait d'être en https.

3.2 CLIENT ESP EN HTTPS

Il existe 3 méthodes différentes pour des Requêtes https :

- Fingerprint => c'est le hash d'un certificat (validité 1 an)
- Client.setInsecure()=> méthode qui permet de ne pas valider le certificat

- Mettre les hash des certificats dans la mémoire SPIFF de l'ESP (durée 20ans)

Nous allons utiliser le Fingerprint

Aide : 2 sites pour bien comprendre

<https://maakbaas.com/esp8266-iot-framework/logs/https-requests/>

<https://arduino-esp8266.readthedocs.io/en/2.4.0/esp8266wifi/client-secure-examples.html>

3.2.1 PICTOGRAMME

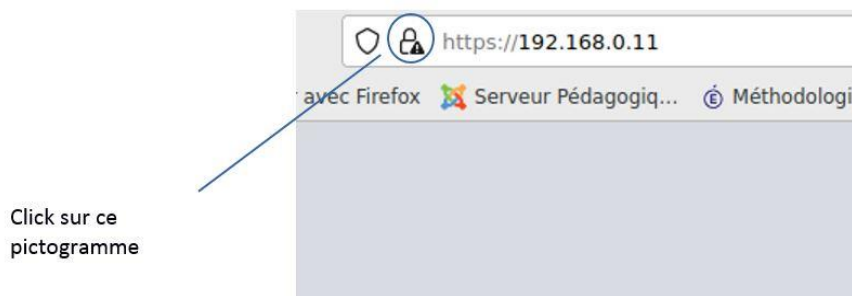


Figure 2 : pictogramme d'une page web https

Pour avoir des informations sur le certificat de ce serveur web, il faut cliquer sur le pictogramme du cadenas

3.2.2 CERTIFICAT

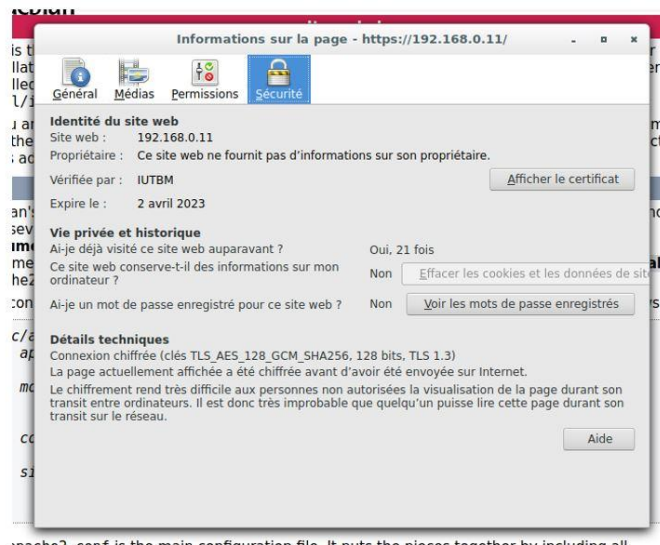


Figure 3 : informations d'une page web https

Pour afficher le certificat de sécurité, il faut valider les informations de la page et valider l'affichage du certificat.

3.2.3 FINGERPRINT

Divers	
Numéro de série	3C:9B:30:0C:4A:CD:09:FD:96:60:9E:D9:F0:E8:EE:E1:D6:42:51:61
Algorithme de signature	SHA-256 with RSA Encryption
Version	3
Télécharger	PEM (cert) PEM (chain)
Empreintes numériques	
SHA-256	22:35:83:5B:12:10:AA:ED:3B:0B:88:2E:4B:68:81:52:58:BA:E1:C2:83:46:9D:42:42:17:D9:BA:5E:A4:61:F2
SHA-1	96:9F:03:C2:F8:7F:E6:BF:DB:D0:7D:BD:52:1A:3D:1A:5F:A5:F9:69

96:9F:03:C2:F8:7F:E6:BF:DB:D0:7D:BD:52:1A:3D:1A:5F:A5:F9:69

Figure 4 :Empreinte numérique d'une page web https

Le fingerprint est donc le hash de votre certificat numérique de sécurité. Dans l'exemple ci-dessus, c'est l'empreinte SHA-1

Relever cette empreinte.

3.2.4 PG ARDUINO CONNEXION HTTPS

- Lancez le fichier BasicHTTPSClient.ino que vous trouverez dans Exemple=> ESP8266HttpClient=> BasicHTTPSClient.ino
- Modifier :
 - Le SSID
 - Le mot de passe
 - le fingerprint
 - La requête https afin qu'elle se connecte à la page web echo.php de votre serveur pour envoyer string=12

Ce programme se connecte à votre wifi et envoie une requête en https vers votre serveur.

- Faire une copie d'écran de l'URL en https et de sa réponse.

3.2.5 PAGE D'ACCUEIL CREATION REGISTRE.TXT

Vous allez modifier votre fichier echo.php (renommer echo2.php) afin que le fichier stocke dans un fichier texte appelé exemple.txt, l'ensemble des valeurs envoyées dans l'URL.

Le programme stockant les données dans un fichier txt et le fichier txt seront stockés dans le mm répertoire pour des questions de droits et de sécurité.

Aide : Basez-vous sur le lien ci-dessous

<https://www.pierre-giraud.com/php-mysql-apprendre-coder-cours/creer-ecrire-fichier/>

- Effectuez plusieurs requête https afin qu'elle se connecte à la page web echo2.php de votre serveur pour envoyer différentes valeurs « string »
- Vérifiez que votre fichier exemple.txt contient plusieurs valeurs.

3.2.6 URL A PLUSIEURS CHAMPS

On souhaite envoyer plusieurs champs dans une URL respectant la signification suivante : temperature/numeroRequete

comme https://192.168.1.121/echo_parametres.php?temp=11&numero=4

- Modifiez votre programme echo2.php en echo_parametres.php qui stockera les 2 paramètres transmis

Aide :

<https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/912799-ecoutez-la-requete-de-vos-utilisateurs-grace-aux-url>

- Effectuez plusieurs requête https afin qu'elle se connecte à la page web echo_parametres.php de votre serveur pour envoyer les différents champs »
- Vérifiez que votre fichier exemple.txt contient plusieurs champs.

Pour aller plus loin sur la sécurité :

<https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/913099-administrez-des-formulaires-de-facon-securisee>

3.2.7 AJOUT D'UNE GRANDEUR DYNAMIQUE A L'URL

Lors de l'envoi d'une requête, il est important d'identifier le µc qui à l'origine de cet envoi.

En même temps que la donnée, on va envoyer l'adresse mac du µc.
L'envoi de cette donnée dynamique permet de différencier les requêtes de différents clients.

- Vous ajouterez les éléments ci-dessous dans votre programme arduino BasicHttpClient.ino que vous renommerez BasicHttpClient_champs_multiples.ino
- Vous modifierez votre programme php pour qu'il prenne en compte 3 champs (temperature/numero de requete/ adresse mac)

```
#include <Arduino.h>

.....

String MAC_Address = " ";

.....

void setup() {

.....

}

void loop() {

....

    MAC_Address=WiFi.macAddress();

    Serial.print("ESP Board MAC Address: "+MAC_Address);

    ....

    if (https.begin(*client, "https://votreURL="+MAC_Address)) { // HTTPS

.....

    }
```

3.3 URL A CHAMPS VARIABLES

Dans l'IOT, en plus d'identifier l'émetteur du message, il est important de connaître le numéro de la requête. En effet, cela permet de savoir si les messages envoyés sont correctement reçus.

Vous allez modifier votre programme que vous nommerez *envoi_3_parametres.ino* de manière à envoyer 3 champs dans votre URL à savoir :

- L'adresse Mac lue par une méthode : `WiFi.macAddress();`
- La température (qui sera constante dans ce chapitre)
- Une requête qui sera incrémentée après chaque envoi.

Le serveur enregistrera ses 3 champs dans un fichier texte de la manière suivante :

```
mac:BC:DD:C2:0A:A7:22 temperature:12 requete n°:1
```

```
mac:BC:DD:C2:0A:A7:22 temperature:12 requete n°:2
```

```
mac:BC:DD:C2:0A:A7:22 temperature:12 requete n°:3
```

Vous modifierez votre programme afin que la chaîne de caractère de l'URL soit la concaténation de ces 3 champs.

Aide :

La concaténation se fait par un +=> chaîne1=chaîne2+chaîne3

Le numéro de requête est une variable de type int qui devra être casté en String pour être associée à l'URL => `String(num_req)`

3.4 JALONS 3

1. Copie d'écran de l'URL en https et de sa réponse.
2. Copie du fichier exemple.txt contenant plusieurs valeurs du string initial.
3. Copie du fichier exemple.txt contenant température et adresse mac
4. Copie du fichier exemple.txt contenant température ,adresse mac et numéro de requête.
5. Copie du programme Arduino, permettant d'envoyer l'URL à plusieurs Champs

3.5 CAPTEUR DE TEMPERATURE

3.5.1 BIBLIOTHEQUES DE CAPTEURS DE TEMPERATURE DS18B20

Vous allez utiliser un capteur One wire qui possède 3 pattes (masse-alim-data). Il devra être associé à une résistance.

Pour interfacer avec le capteur de température DS18B20, vous devez installer la Bibliothèque One Wire de Paul Stoffregen et la Bibliothèque de température de Dallas. Suivez les étapes suivantes pour installer ces bibliothèques.

1. Ouvrez Arduino et accédez à **Croquis > Inclure la bibliothèque > Gérer les bibliothèques**. Le gestionnaire de bibliothèque devrait s'ouvrir.
2. Tapez "**onewire**" dans le champ de recherche et installez la bibliothèque OneWire de JIM Studt et Paul Stoffregen

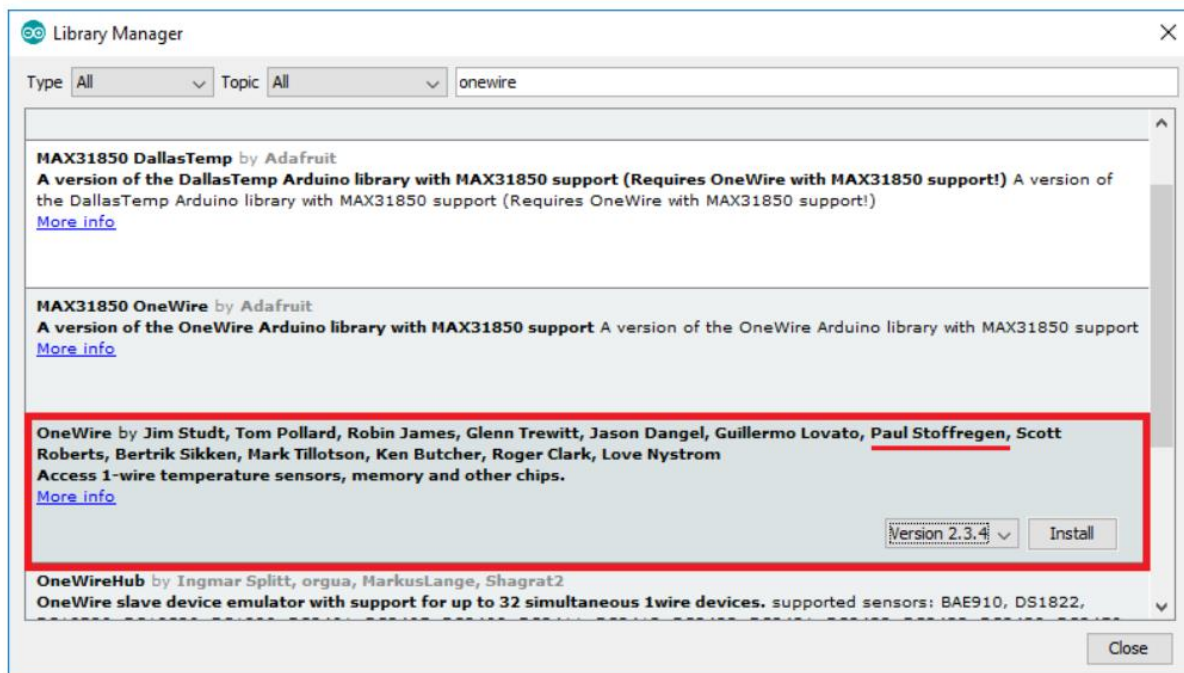


Figure 5 :ajout d’une bibliothèque OneWire dans Arduino

3. Ensuite, recherchez “Dallas” et installez la bibliothèque DallasTemperature de Miles Burton

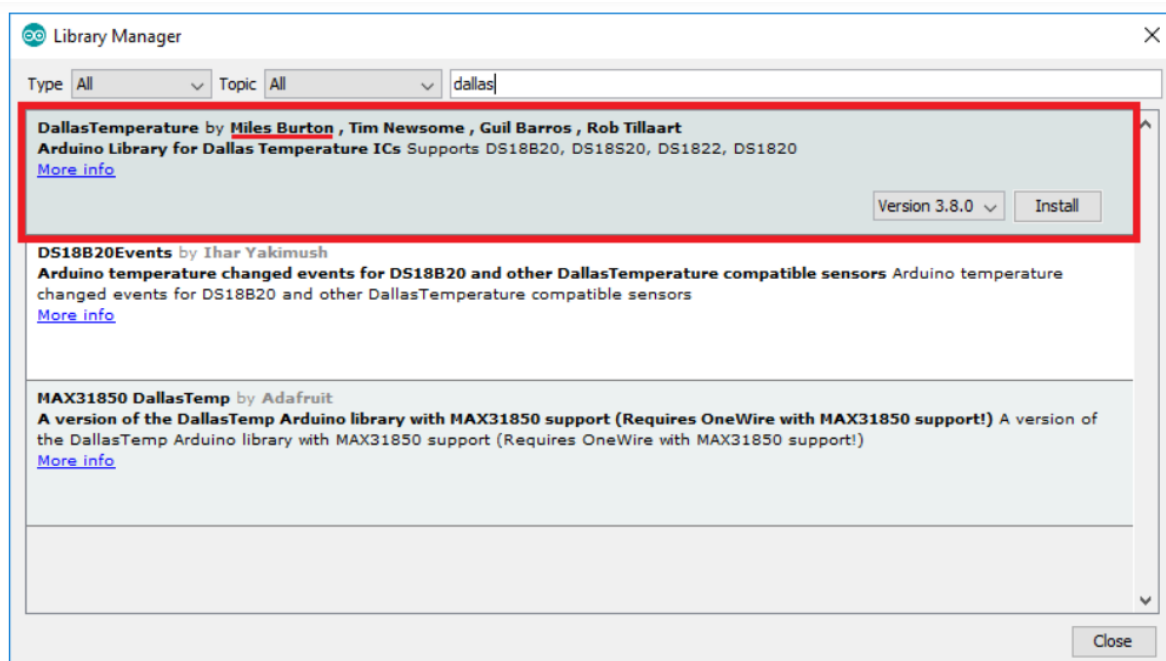


Figure 6 :ajout d’une bibliothèque Dallas dans Arduino

Après avoir installé les bibliothèques, redémarrez votre IDE Arduino.

3.5.2 SCHEMA DE CABLAGE

Câblez le DS18B20 à l'ESP8266 comme indiqué dans le schéma suivant avec la broche de données DS18B20 connectée à GPIO 4 (D2). La résistance est de 4,7kΩ.

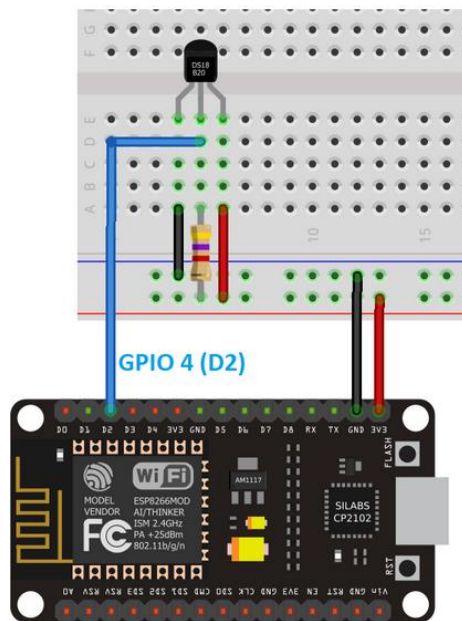


Figure 7 : câblage de l' ESP8266 et du capteur de température DS18B20

3.5.3 LECTURE SIMPLE DE LA TEMPERATURE

Voici le programme qui permet de lire une température sur ce capteur.

```
#include <OneWire.h>
#include <DallasTemperature.h>

const int oneWireBus = 4;
// Mise en place d'une instance oneWire pour communiquer avec des capteurs
OneWire
OneWire oneWire(oneWireBus);
// Mise en place d'une instance en lien avec le capteur Dallas
DallasTemperature sensors(&oneWire);
float temp; // variable Temperature

void setup() {
  sensors.begin();
```

```
Serial.begin(115200);
Serial.println();

}
void loop() {

  sensors.requestTemperatures();
  temp = sensors.getTempCByIndex(0); // Temperature in Celsius degrees
  Serial.printf("Message: %.2f \n", temp);
}
```

Téléversez ce programme et vérifiez que vous récupérez la température sur votre port série.

3.5.4 ENVOI DE LA TEMPERATURE SUR SERVEUR WEB

Il faut associer votre programme de lecture de température au programme « URL à champs variables » afin d'envoyer la température lue dans l'URL avec le numéro de requête qui augmente.

Vérifiez dans votre fichier exemple.txt que vous stockez bien l'ensemble des requêtes envoyées.

3.5.5 JALON 4

Copie de votre fichier exemple.txt