

Club de Robotique et d'Electronique Programmable de Ploemeur

Atelier du 15 avril 2023

Nicolas Le Guerroué

16 avril 2023

Table des matières

	Page
Glossaire	2
Liste des figures	2
Préambule	3
0.1 Conventions	4
1 Les broches d'interruptions	6
1.1 Présentation	6
1.1.1 Évènements d'interruption	6
1.1.2 Chronogrammes d'interruption	7
1.1.3 Les broches disponibles	8
1.2 Exemple avec les capteur PIR	8
1.2.1 Utilisation	9
1.2.2 Code PIR sans interruption	9
1.2.3 Code PIR avec interruption	10
1.2.4 Code d'un bouton avec interruption	10
2 Les modules de communication	12
2.1 Objectifs	12
2.2 Les différents modules	12
2.3 Les modules Bluetooth	13
2.3.1 Configuration	13
2.3.2 Branchements	14
2.3.3 Code Arduino	14
2.4 Les modules radio	18
2.4.1 Branchements	19
2.5 Code Arduino	20
2.5.1 Code complet de l'émetteur	20
2.5.2 Code complet du récepteur	21
2.5.3 Dimensionnement de l'antenne	22
2.6 Les modules Xbee	23
3 Les réseaux	24
3.0.1 Les adresses MAC	24
3.0.2 Les adresse IP	24
3.0.3 Les adresses interdites	26
3.1 Choix des adresses IP	26
3.2 Quelques exemples de type de réseau	26
3.3 Récupération des adresse IP	27
4 Accès global au réseau	30
4.1 Matériel	31
4.2 Configuration de base	31

4.3	Adresse IP publique du routeur	32
4.4	Configuration du routeur	32
4.4.1	Accéder à la page d'administration	33
4.4.2	Une adresse statique	34
4.5	Faire une redirection de port	37
5	Configuration d'un nom de domaine personnalisé	42
5.1	Création du compte	42
5.2	Ajout d'un nom de domaine	44
5.3	Mise en place de l'ESP12	45
5.4	Questionnaire	46

Glossaire

ARP *Address Resolution Protocol* - Table qui gère la correspondance entre les adresses IP et MAC.

DHCP *Dynamic Host Configuration Protocol* - Serveur qui gère l'attribution des adresses IP sur le serveur.

DNS *Domain Name System* - Service qui fait le lien entre un nom de domaine et son adresse IP.

IP *Internet Protocol* - Adresse d'une machine sur un réseau.

MAC *Media Access Control* - Adresse physique (unique) d'un périphérique.

PIR *Passive-Infra-Red* - Capteur de présence.

SFTP *Secure File Transfert Protocol* - Protocole de communication sécurisé dédié au transfert de fichiers sur le réseau.

SSH *Secure Shell* - Protocole de communication sécurisé.

TCP *Transmission Control Protocol* - Protocole réseau qui permet à deux hôtes de communiquer.


UDP *User Datagram Protocol* - Protocole réseau qui permet de transmettre des données en mode broadcast.


Liste des figures

1.1	Un front montant	6
1.2	Un front descendant	7
1.3	Un front montant puis descendant	7
1.4	Un front descendant puis montant	7
1.5	Exemple avec mode RISING	8
1.6	Exemple avec mode FALLING	8
1.7	Exemple avec mode CHANGE	8
1.8	Un capteur PIR	8
1.9	Diagramme temporel du capteur	9
2.1	Un module Bluetooth	13
2.2	Branchements du module Bluetooth	14
2.3	Inclusion de la bibliothèque SoftwareSerial	15
2.4	Un module receveur et émetteur	18
2.5	Branchement de l'émetteur	19
2.6	Branchement du récepteur	19
2.7	Des signaux à différentes fréquences	22
2.8	Un module Xbee	23
3.1	Un réseau plus évolué	27
3.2	La commande ipconfig	28
3.3	La commande ifconfig	29
4.1	Appel du serveur ESP12	30
4.2	Un réseau local	31
4.3	L'adresse IP publique	32
4.4	Un réseau plus vaste	32
4.5	Portail de la Freebox	33
4.6	Confirmation de connexion	33
4.7	Page personnelle de la Freebox	34
4.8	Affichage des paramètres avancés	34
4.9	Ouverture des paramètres DHCP	35
4.10	Ouverture des paramètres DHCP	35
4.11	Ouverture du moniteur série	36
4.12	Configuration des bauds statiques	36


4.13	Ajout du bail statique	36
4.14	Ajout du bail statique	37
4.15	Confirmation du bail statique	37
4.16	Gestion des ports	37
4.17	Bouton d'ajout de redirection de port	38
4.18	Paramètres de redirection	38
4.19	Confirmation du port et de la redirection	39
4.20	Représentation des ports du routeur	40
4.21	Représentation des ports	41
5.1	Interface d'accueil de No-IP	42
5.2	Bouton Sign Up	43
5.3	Formulaire d'inscription	43
5.4	Page principale du compte No-IP	43
5.5	Nombre de nom de domaine configuré	44
5.6	Nombre de noms de domaine configurés	44
5.7	Ajout d'un nom de domaine	44
5.8	Saisie de l'adresse publique	45
5.9	branchement du capteur de température	46
5.10	Votre nom de domaine dans le navigateur Internet	46


Préambule


 Document réalisé en \LaTeX par Nicolas Le Guerroué pour le Club de Robotique et d'Électronique Programmable de Ploemeur (CREPP)


 Version du 16 avril 2023


 Taille de police : 11pt (carlito)

 N'hésitez pas à faire des retours sur le document, cela permettra de l'améliorer

 nicolasleguerroue@gmail.com

 <https://github.com/CREPP-PLOEMEUR>¹

 Permission vous est donnée de copier, distribuer et/ou modifier ce document sous quelque forme et de quelque manière que ce soit.

 **Dans la mesure du possible, évitez d'imprimer ce document si ce n'est pas nécessaire. Il est optimisé pour une visualisation sur un ordinateur et contient beaucoup d'images.**

Conventions










Les commandes à saisir sont dans des encadrés similaires :

```
sudo apt-get update
```

Code 1 - Exemple de commande

Parfois, ces encadrés contiendront des instructions qu'il faudra placer dans certains fichiers.

1. Click-droit et **Copier l'adresse du lien**

- Les fichiers sont indiqués par le repère  fichier
- Les dossiers sont indiqués par le repère  dossier
- Les logiciels sont indiqués par le repère  logiciel²
- Les adresses IP sont indiquées par le repère  Adresse IP
- Les adresses MAC sont indiquées par le repère  Adresse MAC
- Les liens sont indiqués par le repère  Lien
- Les broches génériques des composants sont indiquées par le repère  Broche et se scindent en deux parties :
 - Les broches d'entrée par  Broche
 - Les broches de sortie par  Broche

Versions

octobre 2021	Fusion des supports d'ateliers
novembre 2021	Ajout de l'atelier sur les servomoteurs
décembre 2021	Ajout de l'atelier sur les moteurs pas-à-pas
janvier 2022	Ajout de l'annexe pour l'installation des bibliothèques ESP8266
février 2022	Ajout de l'atelier pour le serveur Web ESP8266 NodeMCU
mars 2022	Ajout de l'atelier pour la partie Réseaux (Adressage) et de l'atelier Capteurs
mai 2022	Ajout de l'atelier sur les modules de communication
janvier 2023	Ajout de l'atelier sur les écrans
mars 2023	Ajout de l'atelier sur les redirections des ports

2. Sont également concernés les paquets Linux et les bibliothèques des langages

Section 1

Les broches d'interruptions

Présentation

Dans certains cas, il est souhaitable de récupérer la valeur d'une broche à tout moment du programme, même quand celui ci est occupé dans une tâche et même dans une fonction de temporisation ¹.

Pour remédier à ce problème, on peut utiliser les **broches d'interruption** qui permettent de récupérer la main sur l'ensemble du programme lorsque'un évènement survient sur une broche.

Concrètement, lorsque un évènement **e** survient sur la broche **b**, la fonction **f** est appelée, quelque soit l'état du programme principal.

Un évènement est une modification de l'état de la broche.

Évènements d'interruption

Il existe différents états pour les broches :

- RISING : front montant

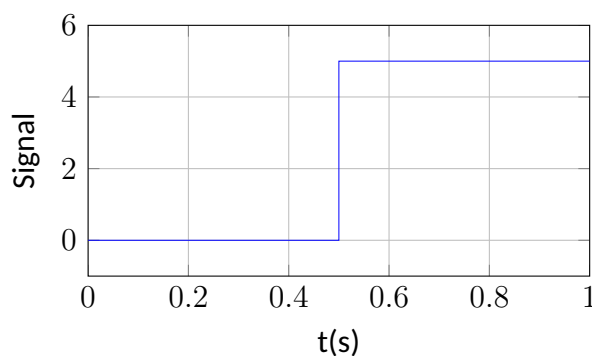


Figure 1.1 – Un front montant

1. Voir `delay()`, `delayMicroseconds()`

- FALLING : Front descendant

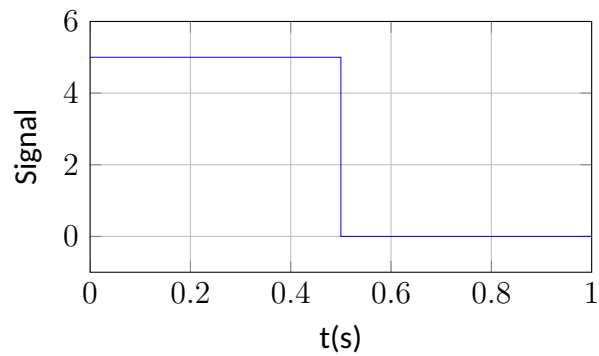


Figure 1.2 – Un front descendant

- CHANGE : Front montant puis descendant ou descendant puis montant

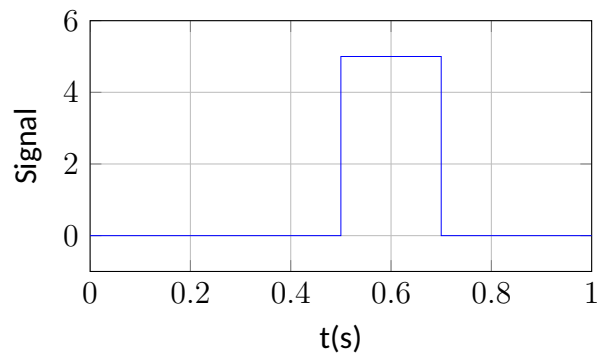


Figure 1.3 – Un front montant puis descendant

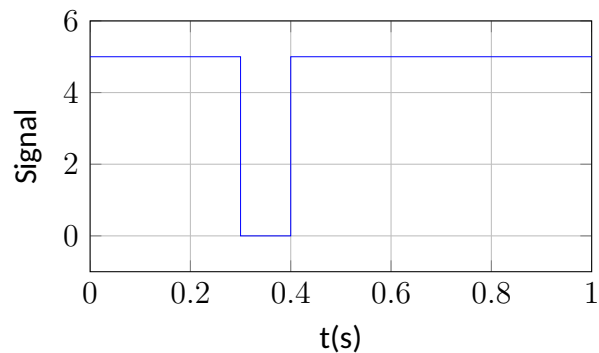


Figure 1.4 – Un front descendant puis montant

Chronogrammes d'interruption

Lorsqu'un évènement déterminé survient sur une broche, le code principal va se mettre en pause et la fonction spéciale sera appelée. Une fois cette fonction terminée, le code principale reprend son traitement là où il s'est arrêté.

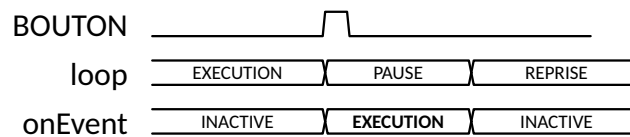


Figure 1.5 – Exemple avec mode RISING

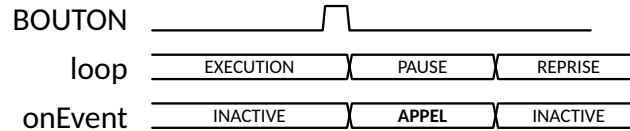


Figure 1.6 – Exemple avec mode FALLING

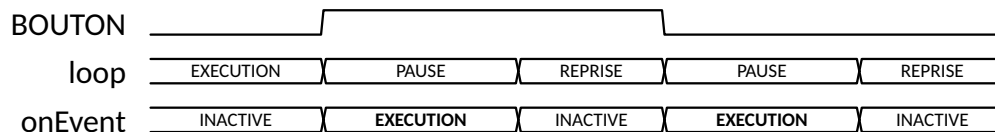


Figure 1.7 – Exemple avec mode CHANGE

Les broches disponibles

Pour les cartes Arduino Uno, il est possible d'utiliser les broches D2 et D3 en tant que broche d'interruption.

Pour les cartes Arduino Mega, il y a plus de broches, pour plus d'informations, se reporter au lien suivant :

[Documentation Arduino](#)

Exemple avec les capteur PIR

Par exemple, on souhaite réagir prioritairement à un capteur PIR quel que soit l'état du programme.

Les capteurs PIR² détectent les rayonnements infrarouges émis par un objet.

Puisque tout objet émet un rayonnement infrarouge, le capteur PIR est muni de deux cellules sensibles aux infrarouges qui vont détecter ces rayons infrarouges réfléchit ou émit par l'objet.



Figure 1.8 – Un capteur PIR

2. PIR: *Passive-Infra-Red* - Capteur de présence

Lorsqu'il n'y a pas de mouvement, le niveau d'infrarouge reçu est le même pour les deux cellules. Lors du passage d'un objet, l'émission de ces rayons va être modifiée sur une cellule puis sur l'autre ce qui va permettre de détecter le mouvement.

Le cache blanc, qui couvre et protège généralement le capteur, est une lentille de Fresnel avec plusieurs facettes qui permet de concentrer le rayonnement infrarouge sur les cellules.

Utilisation

Ces capteurs possèdent une broche de sortie qui est mise à l'état HAUT pendant une certaine durée³ lorsqu'il y a détection d'un mouvement.

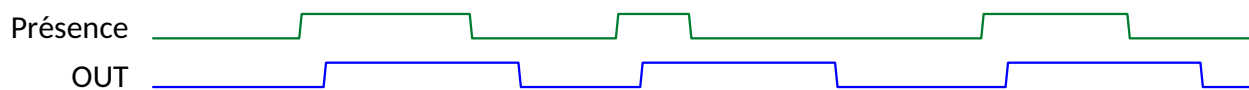


Figure 1.9 – Diagramme temporel du capteur

Nous allons présenter un code réalisé sans interruption et un autre qui utilise les interruptions.

Code PIR sans interruption

Voici un code d'exemple pour réagir dès qu'une présence est détectée.

```
#define LED 13      //Broche de la LED
#define OUT 2       //Broche du capteur PIR

void setup()
{
    pinMode(LED, OUTPUT); //LED en sortie
    pinMode(OUT, INPUT);  //Broche du capteur en entrée
    Serial.begin(9600);    //Vitesse de communication à 9600 bauds
} //End setup

void loop(){

    outValue = digitalRead(OUT);           //Lire l'état du capteur

    if (outValue == HIGH)                  //Détection d'un mouvement
    {
        Serial.println("Detection");
    } //End else
```

3. Cette durée est réglable avec le potentiomètre sur le capteur

```
}//End loop
```

Code 2 - Code sans interruption

Maintenant, supposons que la carte exécute un programme qui prend plusieurs secondes. Comment faire ?

On peut utiliser une interruption externe.

Code PIR avec interruption

```
#define LED 13    //Broche de la LED
#define OUT 2     //Broche du capteur PIR

void setup() {

    Serial.begin(9600); //Vitesse de communication à 9600 bit/s

    pinMode(LED, OUTPUT); //LED en sortie
    pinMode(OUT, INPUT);  //Broche du capteur en entrée

    attachInterrupt(digitalPinToInterrupt(OUT), presence, RISING); //Appel de
la fonction presence à chaque changement de front du bouton
    Serial.println("Init");
}

void loop() {

    delay(5000); //Pause du programme principal

}

void presence() {

    Serial.println("Presence !");
}
```

Code 3 - Code avec interruption

Code d'un bouton avec interruption

En dernier exemple, prenons le cas d'un bouton qui doit changer l'état d'une LED à n'importe quel moment du programme :



Le code complet est disponible en faisant un clic-droit puis **Copy link Address** sur le lien suivant : [Lien du code](#) .
Il ne vous reste plus qu'à coller le lien dans votre navigateur Internet.

```
int ledPin = 13;    //Led interne
int BOUTON = 2;    //Bouton relié à la broche 2 avec une résistance de charge

volatile int state = LOW; //Etat courant de la LED

void setup() {

    Serial.begin(9600); //Vitesse de communication à 9600 bit/s

    pinMode(ledPin, OUTPUT);           //Led mise en sortie
    pinMode(BOUTON, INPUT_PULLUP);     //Bouton mis en entrée

    attachInterrupt(digitalPinToInterrupt(BOUTON), onEvent, CHANGE); //Appel
de la fonction onEvent à chaque changement de front du bouton
    Serial.println("Init");
}

void loop() {

    delay(5000); //Pause du programme principal
}

void onEvent() {

    state = !state; //Inverse l'état de la LED

    if(state){
        Serial.println("ON");
    }else{
        Serial.println("OFF");
    }
    digitalWrite(ledPin, state); //Met à jour l'état de la LED
}
```

Code 4 -

Ici, quelque soit l'action effectuée dans la fonction loop, dès qu'un front montant est détecté sur la broche BOUTON (2), la fonction onEvent() sera exécutée et changera l'état de la LED à chaque front.

Section 2

Les modules de communication

Objectifs

Ce chapitre a pour but de faire un petit tour d'horizon des différents modules de communication et les technologies associées.

Les différents modules

Il existe une multitude de modules :

- Modules Infrarouge
- Modules Radio basses fréquence (433MHz)
- Modules Radio hautes fréquences (Wifi, Bluetooth) à 2.4 GHz

Les deux derniers modules se déclinent en une multitude de modules :

- Modules Lora
- Modules Xbee
- Modules HC-04 ou HC-05¹
- Modules Crius

Les modules Bluetooth

1. Les modules HC-05 sont configurables en mode maître ou esclave et les modules HC-06 en mode esclave uniquement.



Figure 2.1 – Un module Bluetooth

La plupart des modules Bluetooth communiquent en liaison série (broche RX et TX) et se configurent avec les commandes AT.

Configuration

Il s'agit d'un jeu d'instruction pour gérer les paramètres des modules comme l'identifiant, le nom, la vitesse de communication, etc.

Ces commandes étaient utilisées à l'origine pour les modem Hayes Smartmodem 300 et sont donc également appelées **commandes Hayes**.

Chaque commande est envoyée sous la forme d'une ligne de texte encodée en ASCII, débutant par le mot **AT** et se terminant par le caractère `␣` (code ASCII 13).

Le module retourne une réponse sous la forme d'une ou plusieurs lignes selon la commande envoyée, chaque ligne se terminant par les caractères `␣` (code ASCII 13) et `␠` (code ASCII 10).

Branchements

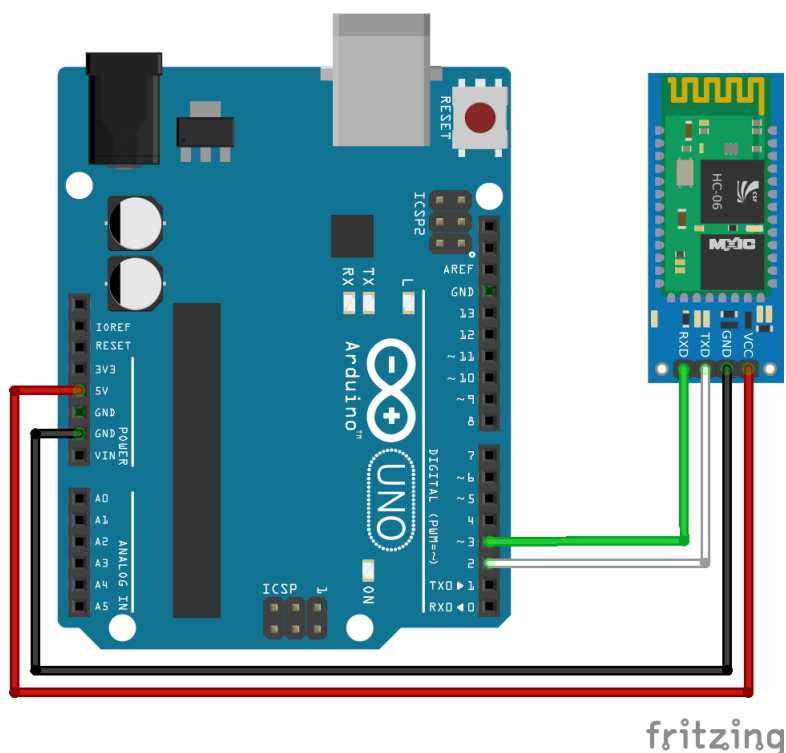


Figure 2.2 – Branchements du module Bluetooth

Pour l'exemple, nous nous baserons sur un module HC-06 ou HC-05.

il faut relier le « +5V » du module au 5 Volts de la carte Arduino et la masse du module à celle de la carte.

Ensuite, nous allons relier la broche TX du module à la broche 2 de la carte et la broche RX à la broche 3.

Code Arduino



Afin de lire les données du module, nous allons "émuler" une voie série, en l'occurrence les broches 3 et 2. C'est-à-dire que nous allons déclarer que ces broches recevront et enverront des données.

Pour cela, il faut utiliser la bibliothèque **SoftwareSerial**. Dans le logiciel Arduino, allez dans **croquis** puis **inclure une bibliothèque** et sélectionnez **SoftwareSerial**.

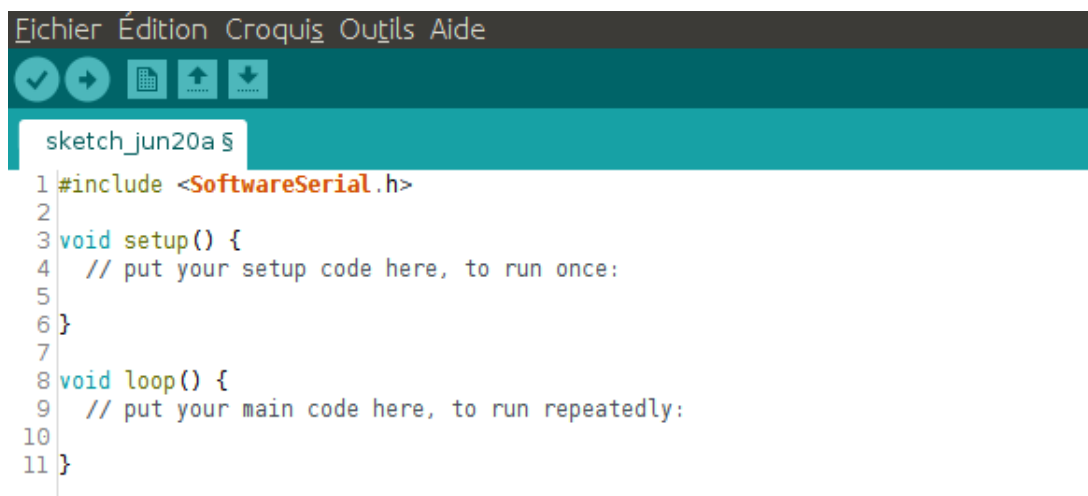


Figure 2.3 – Inclusion de la bibliothèque SoftwareSerial

Maintenant, nous allons définir les broches du module :

```
const int RX = 3; //RX du module
const int TX = 2;
```

Code 5 - Définitions des broches RX et TX

Après ceci, il faut déclarer un objet **SoftwareSerial** qui prendra en argument respectivement les broches TX et Rx, un peu comme lors de la déclaration d'un écran LCD (« LiquidCrystal lcd (RS,E,D4,D5,D6,D7); »).

On obtient donc :

```
#include <SoftwareSerial.h>

const int RX = 3; //RX du module
const int TX = 2;

SoftwareSerial device(RX,TX);
```

Code 6 - Définition de l'objet SoftwareSerial

Bien entendu, "device" peut être remplacé par ce que vous voulez.

Ensuite, on déclare que la communication carte-module peut débuter avec "device.begin(9600);" Où 115200 correspond à la vitesse de transmission en bauds (comme "Serial.begin(9600);");

```
SoftwareSerial device(RX,TX);
void setup() {
    device.begin(9600);
}
```

Code 7 - Vitesse de communication



Par la suite, en cas d'erreurs de transmission Bluetooth (pas de données...), il conviendra de vérifier le branchement des broches RX et TX (essayer de les intervertir) et d'éventuellement changer la vitesse de communication car certains modules communiquent à 115200 bauds !

Pour lire les données du module, ce sont les mêmes fonctions que pour le port série :
En effet :

Pour le port série :

- ▶ Serial.begin(115200);
- ▶ Serial.available();
- ▶ Serial.read();
- ▶ Serial.print();
- ▶ Serial.println();

Pour le module Bluetooth :

- ▶ device.begin(115200);
- ▶ device.available();
- ▶ device.read();
- ▶ device.print();
- ▶ device.println();

Tant que des données (caractères) sont disponibles, nous allons les assembler en une chaîne de caractère (concaténation).

Ensuite, avec un **if** , nous allons voir si cette chaîne en question correspond par exemple à "a".

Il faut donc définir un caractère x et une chaîne de caractère.

Donc dans le programme Arduino, avant la fonction **setup** , on rajoute :

```
#include <SoftwareSerial.h>

const int RX = 3; //RX du module
const int TX = 2;

char c;
String message;
```

Code 8 - Définition des structures du message

La boucle while va permettre de lire les données : Dans la boucle **loop** : on écrit :

```
void loop() {  
  
    while() {  
  
    }//Fin while  
  
}//Fin void loop
```

Code 9 - Boucle de lecture partielle

Maintenant que la boucle va attendre des données, il suffit de les lire et de les transformer en chaîne de caractère.

Pour cela :

- ▶ on lit le premier caractère c
- ▶ on définit que la chaîne message = message + c

On obtient :

```
void loop() {  
  
    while(device.available()>0) {  
  
        c = device.read();  
        message = message + c;  
    }//Fin while  
  
}//Fin void loop
```

Code 10 - Boucle de lecture complète

La structure conditionnelle est très simple :

Après la boucle « while », mettez :

```
void loop() {  
  
    while(device.available()>0) {  
  
        c = device.read();  
        message = message + c;  
    }//Fin while  
  
    if(message=="c") {  
  
        Serial.println("C");  
    }//Fin if message=="c"
```

```
}//Fin void loop
```

Code 11 - Structure conditionnelle

Les modules radio

Ces modules radio, émettant généralement sur la fréquences de 433 MHz, sont abordables et relativement faciles à mettre en oeuvre.

Comme toute communication radio, il faut donc un émetteur et un récepteur.

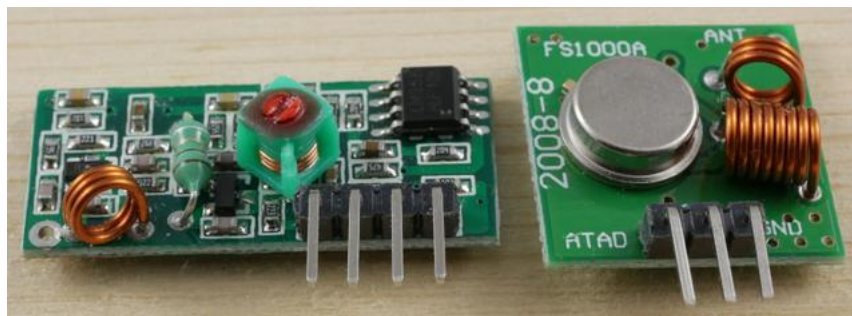


Figure 2.4 – Un module receveur et émetteur

Dans cette section, nous utiliserons un code qui permet d'envoyer et recevoir des données sans antenne adaptée. Cependant, pour une meilleure efficacité et la mise en oeuvre d'un projet plus élaboré,², il faudra se reporter à la section pour le dimensionnement d'une antenne (Section [2.5.2](#))

Branchements

2. c'est à dire un projet sortant du cadre de cet atelier sur les récepteur et émetteurs radio.

Module émetteur

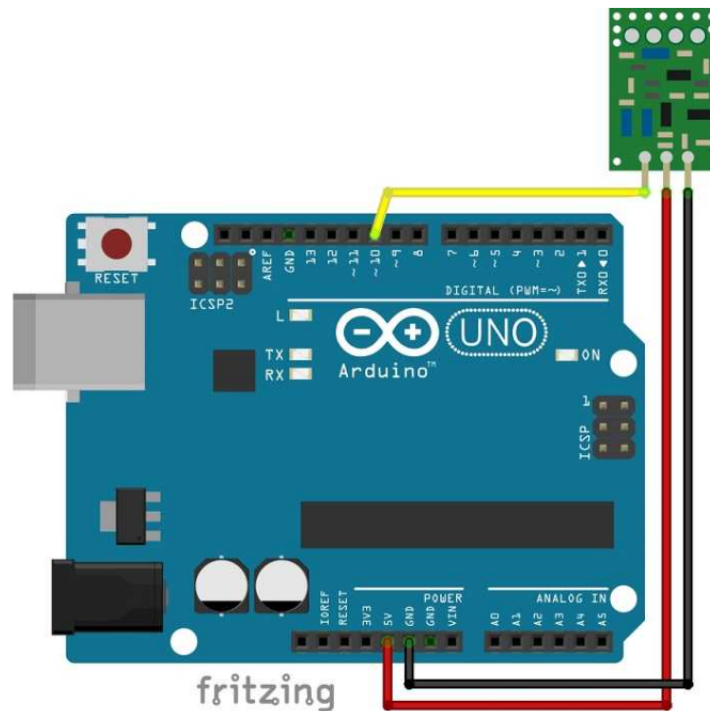


Figure 2.5 – Branchement de l'émetteur

- **VCC** sur le **+5V** de l'Arduino
- **GND** sur le **GND** de l'Arduino
- **Data** sur la broche **10** de l'Arduino

Module récepteur

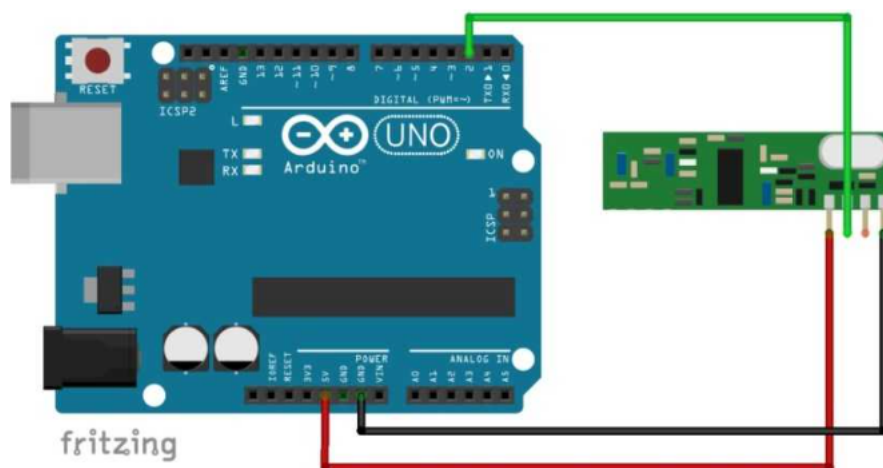









Figure 2.6 – Branchement du récepteur

-  **VCC** sur le  **+5V** de l'Arduino
-  **GND** sur le  **GND** de l'Arduino
-  **Data** sur la broche  **D2** de l'Arduino La broche D2 est utilisée en tant que broche d'interruption.

Code Arduino

Code complet de l'émetteur



Le code complet est disponible en faisant un clic-droit puis **Copy link Adress** sur le lien suivant :  **Lien du code** .
Il ne vous reste plus qu'à coller le lien dans votre navigateur Internet.

```
//Inclusion de la bibliothèque RCSSwitch
#include "RCSSwitch.h"

RCSSwitch deviceSender = RCSSwitch(); //Création de l'objet

#define SENDER_PIN 10 //Broche d'envoi
#define FREQUENCY 433 //Fréquence en MHz

#define MAX_MESSAGE_SIZE 8 //Taille maximale du message en bit - 1 octets

void setup() {

    deviceSender.enableTransmit(SENDER_PIN);
    deviceSender.setPulseLength(FREQUENCY); //Détermine la fréquence
    deviceSender.setRepeatTransmit(10); //Nombre maximal d'envois

    pinMode(13, OUTPUT); //Pour faire clignoter la LED interne toutes les
secondes

}

void loop() {

    deviceSender.send(0xFF, MAX_MESSAGE_SIZE); //Envoi
    //0xFF en hexadécimal = 1111 1111 en binaire = 255 en binaire
    delay(970);

    //Clignotement
    digitalWrite(13, HIGH);
    delay(30);
}
```

```
digitalWrite(13, LOW);  
  
}
```

Code 12 - Code de l'émetteur

Code complet du récepteur



Le code complet est disponible en faisant un clic-droit puis **Copy link Address** sur le lien suivant : [Lien du code](#) .

Il ne vous reste plus qu'à coller le lien dans votre navigateur Internet.

```
//Inclusion de la bibliothèque RCSSwitch  
#include "RCSSwitch.h"  
  
RCSSwitch deviceReceiver = RCSSwitch(); //Création de l'objet  
  
#define RECEIVER_PIN 0 //Broche de réception [Broche d'interruption -> D2]  
  
void setup() {  
  
    deviceReceiver.enableReceive(RECEIVER_PIN);  
    Serial.begin(115200); //Communication série  
    Serial.println("Start...");  
  
}  
  
void loop() {  
  
    //Si des données sont disponibles...  
    if(deviceReceiver.available()>0)  
    {  
        uint8_t data = deviceReceiver.getReceivedValue(); //Récupère les données  
        Serial.println("Value = "+String(data));  
        deviceReceiver.resetAvailable(); //Vide le buffer  
        delay(100);  
    }  
  
}
```

Code 13 - Code du récepteur

Dimensionnement de l'antenne

Une antenne permet de recevoir et d'envoyer les données. En dimensionnant correctement une antenne, nous pouvons améliorer les performances. La plupart du temps, on utilise une antenne **quart d'onde** ou **demi-onde**.

Les relations fondamentales

Un signal est déterminé par sa fréquence, c'est à dire le nombre de motifs élémentaires en 1 seconde. La fréquence est exprimée en Hertz (Hz). Ainsi, 1 Hz représente un motif élémentaire en 1 seconde.

La figure représente l'évolution de la tension de 2 signaux.

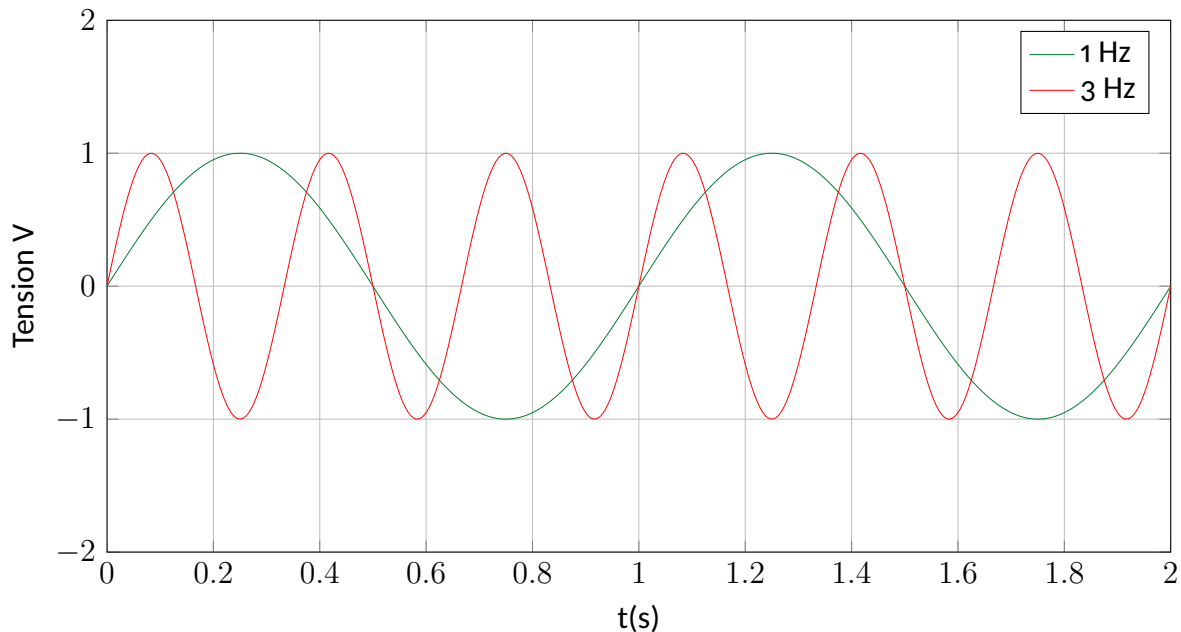


Figure 2.7 – Des signaux à différentes fréquences

La période d'un signal est le temps que met le signal pour passer par tous les points du signal élémentaire. La période, exprimée en seconde, vaut :

$$T = \frac{1}{f}$$

Enfin, penchons nous sur la longueur d'onde.

La longueur d'onde est la distance séparant deux maxima consécutifs de l'amplitude d'un signal. Cette grandeur, exprimée en mètre, dépend de la vitesse de l'onde en question. Dans le cas des ondes électromagnétiques, cette vitesse correspond à celle de la lumière, c'est à dire $3.0 \cdot 10^8 \text{ m} \cdot \text{s}^{-1}$

La longueur d'onde est souvent notée λ (lambda).

Nous avons donc :

$$\lambda = \frac{c}{f}$$

avec λ la longueur d'onde en mètre, c la vitesse de l'onde en $\text{m} \cdot \text{s}^{-1}$ et f la fréquence de l'onde en Hz

Dans le cas de nos modules à 433 MHz, nous avons :

$$\lambda = \frac{3 \cdot 10^8}{433 \cdot 10^6} = 0.69 \text{ m}$$

L'onde envoyée par l'émetteur possède donc une longueur d'onde valant 69 cm. Une adaption correcte des antennes pour ces applications est de prendre une antenne droite³ donc la longueur vaut le quart de la longueur d'onde : c'est-à-dire 17 cm.

Les modules Xbee

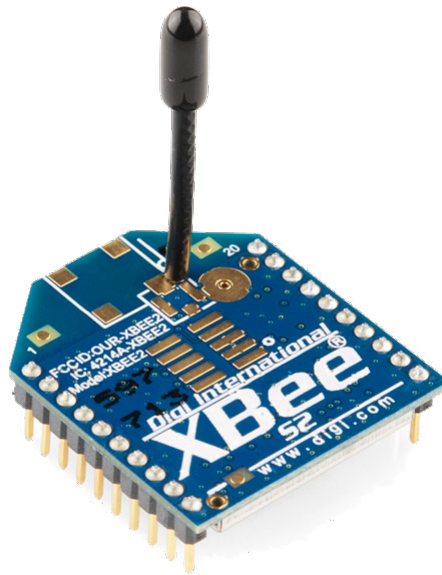


Figure 2.8 – Un module Xbee

Ce module communique via une liaison série.

L'un des inconvénient de ce module est l'espacement des broches de 2mm et non de 2.54 mm. Il faut donc utiliser un shield adapté.

La fréquence de communication est cette fois ci de 2.4GHz.

3. En prenant un câble rigide.

Section 3

Les réseaux

Un réseau est un ensemble de machines¹ reliées entre elles. Dans notre cas, la liaison sur le réseau se fera en Ethernet ou WiFi.

Pour communiquer entre elles, il faut définir des adresses IP² exactement comme une adresse postale pour transmettre un courrier.

Les adresses MAC

Une adresse MAC³ est une adresse unique qui désigne la machine⁴. Elle est de la forme MAC XX:XX:XX:XX:XX:XX (six octets usuellement exprimés et hexadécimal) et reste invariante dans le temps.

Contrairement aux adresses IP, les adresses MAC ne sont normalement pas attribuées explicitement par configuration ; elles sont au contraire attribuées à la production de la carte réseau par son fabricant. La table ARP⁵ d'une machine sert à associer des adresses IP à des adresses MAC.

Les adresse IP

Une adresse IP est une adresse donnée à une machine qui rejoint le réseau.

A quoi ressemble une adresse IP ?

- Version 4 (Ipv4) : Les adresses sont codées sur 32 bits

192.168.0.1 = BIN 1100000.10101000.0000000.00000001 (En Binaire)

1. Serveurs, PC, Tablettes, téléphones, imprimantes


2. IP: *Internet Protocol* - Adresse d'une machine sur un réseau

3. MAC: *Media Access Control* - Adresse physique (unique) d'un périphérique

4. Plus précisément l'adresse de la carte réseau de la machine

5. ARP: *Address Resolution Protocol* - Table qui gère la correspondance entre les adresses IP et MAC

- version 6 (Ipv6) : les adresses sont codées sur 128 bits

. L'adresse de version 4 (IPv4) est encore actuellement la plus utilisée.

Dans un réseau, chaque machine possède une adresse IP fixée par l'administrateur du réseau. Il est interdit de donner la même adresse à 2 machines différentes sous peine de dysfonctionnement.

Une adresse IPv4 est une suite de 32 bits (4 octets) notée en général a.b.c.d avec a, b, c, et d des entiers « décimal » compris entre 0 et 255. Chaque valeur a, b, c ou d représente dans ce cas une suite de 8 bits.

Exemple 1. Une machine qui a comme adresse IP 134.214.80.12 (En décimal) :

- a vaut 134 soit (1000 0110) en binaire.
- b vaut 214 soit (1101 0110) en binaire.
- c vaut 80 soit (0101 0000)
- d vaut 12 soit (0000 1100).

En binaire, l'adresse IP s'écrit  et puisque le codage se fait sur 8 bits les valeurs seront obligatoirement comprises entre 0 et 255.

Le net-id et le host-id

Au sein d'un même réseau IP, toutes les adresses IP commencent par la même suite de bits. L'adresse IP d'une machine va en conséquence être composée de 2 parties :

- Net-id : La partie fixe de l'adresse IP
- Host-id : La partie réservée aux machines qui viennent sur le réseau

Masque de réseau IP

Le masque du réseau permet de connaître le nombre de bits du net-id. On appelle N ce nombre. Il s'agit d'une suite de 32 bits composée en binaire de N bits à 1 suivis de 32-N bits à 0. C'est le masque du réseau qui définit la taille d'un réseau IP : c'est-à-dire la plage d'adresses assignables aux machines du réseau.

Ainsi, pour connaître le net-id, on va faire un ET (& ou .) logique entre le masque et l'adresse IP.

Exemple 2. Prenons l'adresse IP  et le masque .

Que vaut le net-id ?

Pour rappel :

- ▶ 0.0 = 0
- ▶ 0.1 = 0
- ▶ 1.0 = 0
- ▶ 1.1 = 1

Solution 1. L'adresse IP 192.168.1.0 vaut BIN 11000000.10101000.00000000.00000000 et
 L'adresse MASK 255.225.255.0 vaut BIN 11111111.11111111.11111111.00000000

01100000.10101000.00000000.00000000
 ET 11111111.11111111.11111111.00000000
 = 01100000.10101000.00000000.00000000 = BIN 01100000.10101000.00000000.00000000

L'adresse IP 192.168.0.0 obtenue représente donc l'adresse du réseau .

Les autres bits à 0 sont donc réservés aux périphériques du réseau. Étant sur 8 bits, il y a donc $2^8 - 2$ adresses disponibles sur ce réseau.

Les adresses interdites

Il est interdit d'attribuer à une machine d'un réseau l'adresse du réseau et l'adresse de broadcast (diffusion)

L'adresse de diffusion

Cette adresse permet à une machine d'envoyer une info à toutes les machines d'un réseau. Cette adresse est celle obtenue en mettant tous les bits de l'host-id à 1.

Dans notre cas c'est l'adresse IP 192.168.0.255

L'adresse de réseau

Cette adresse est celle obtenue après avoir fait le ET entre l'adresse IP et le masque de sous-réseau. Dans notre cas c'est l'adresse IP 192.168.0.0

Choix des adresses IP

Sur un réseau local, les adresses sont choisies par un serveur DHCP ⁶ .

Les adresses peuvent être :

- Statiques : Un appareil sur le réseau possède la même adresse après connexion puis déconnexion.
- Dynamiques : l'adresse IP varie dans le temps au bout d'une période d'expiration après une déconnexion (bail d'une journée par exemple)

A de rares exceptions, vous n'êtes pas censé attribuer une adresse IP vous-même à une machine.

Quelques exemples de type de réseau

Un réseau IP peut avoir une taille très variable :

6. DHCP: Dynamic Host Configuration Protocol - Serveur qui gère l'attribution des adresses IP sur le serveur

- Une entreprise moyenne aura un réseau comportant une centaine de machines.
- Un campus universitaire aura un réseau comportant de quelques milliers à quelques dizaines de milliers de machines.
- Le réseau d'une multinationale (un grand fournisseur d'accès par exemple) peut comporter des millions de postes.

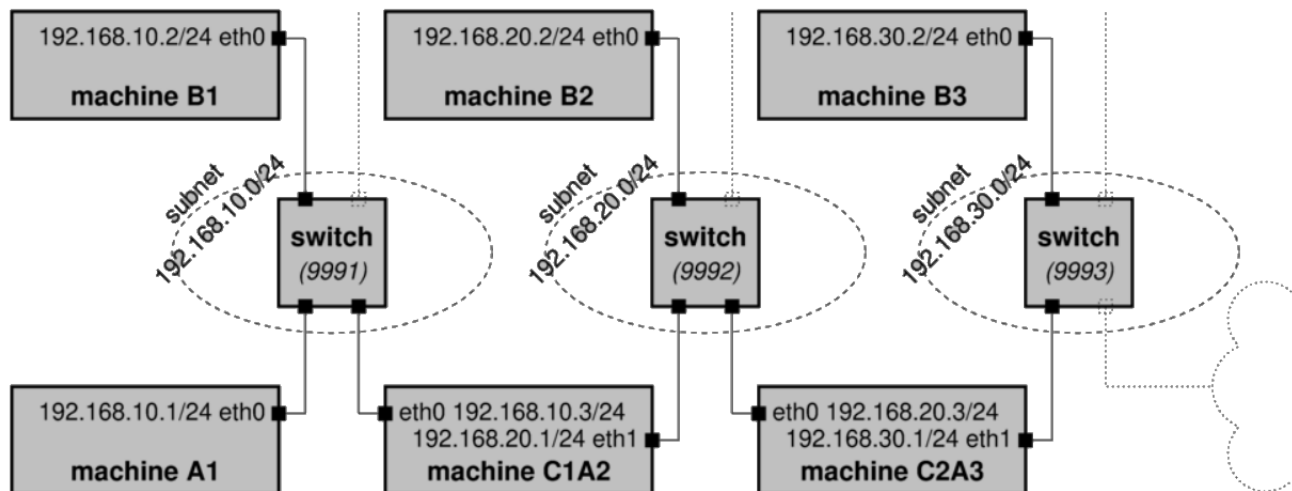


Figure 3.1 – Un réseau plus évolué

Il existe 2 types de réseau :

- Les réseaux publics Internet où chaque équipement connecté doit posséder une adresse unique et enregistrée au niveau mondial.
- Les réseaux privés, dans ce cas le choix des adresses est libre et ne doivent être uniques que dans ce réseau.

Si un réseau privé doit être inter-connecté avec le réseau Internet, il faudra alors utiliser des adresses privées qui ne puissent correspondre à des adresses publiques utilisées sur Internet. Des plages d'adresses réservées à usage privé existent et elles ne sont donc pas acheminées par les routeurs Internet, ce qui supprime tout risque de conflit (cf. document annexe).

Récupération des adresse IP

Voici 2 commandes pour récupérer l'adresse IP et le masque de sous-réseau.

Pour Windows

Sous un terminal Windows ⁷

7. Saisir **cmd** dans le gestionnaire de programme

```
ipconfig
```

Code 14 - Récupération des informations IP sous Windows

```
C:\Users\Admin>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet :

    Suffixe DNS propre à la connexion. . . :
    Adresse IPv6. . . . . : 2a01:e0a:227:a340:a4f7:9a72:aa0a:347f
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:227:a340:2d8f:1bca:f420:b777
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:227:a340:3d02:1319:b97a:cdbb
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:227:a340:4448:9d1:fe45:3916
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:227:a340:907f:4342:42d7:8451
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:227:a340:99ff:f749:3636:af2e
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:227:a340:ac70:9311:f199:596e
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:227:a340:ddc1:2e12:5b54:a4ed
    Adresse IPv6 de liaison locale. . . : fe80::a4f7:9a72:aa0a:347f%7
    Adresse IPv4. . . . . : 192.168.1.41
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : fe80::72fc:8fff:fe47:689c%7
                                   192.168.1.254
```

Figure 3.2 – La commande ipconfig

Pour Linux

Sous un terminal Linux, on peut utiliser la commande `ifconfig`⁸

```
ifconfig
```

Code 15 - Récupération des informations IP sous Linux

8. Si non installée sous Linux, saisir `sudo apt-get install net-tools`

```
nico@nico-ThinkPad-L580:~$ ifconfig
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether e8:6a:64:7d:be:44 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xe1200000-e1220000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 2228 bytes 337363 (337.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2228 bytes 337363 (337.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp5s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.49 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 2a01:e0a:227:a340:2080:e8d3:7ee3:2b7e prefixlen 64 scopeid 0x0<global>
    inet6 fe80::c48e:e202:5ade:c5fc prefixlen 64 scopeid 0x20<link>
    inet6 2a01:e0a:227:a340:d757:63b3:a442:734e prefixlen 64 scopeid 0x0<global>
    ether fc:77:74:1e:76:ed txqueuelen 1000 (Ethernet)
    RX packets 42721 bytes 38163746 (38.1 MB)
    RX errors 0 dropped 6 overruns 0 frame 0
    TX packets 24604 bytes 6842107 (6.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 3.3 – La commande ifconfig

Section 4

Accès global au réseau

L'objectif de ce chapitre est de présenter les différentes étapes pour accéder à un serveur depuis n'importe quel lieu avec une connexion Internet.

Dans le cadre de ce chapitre, nous souhaitons rendre accessible le serveur ESP12 mis en place précédemment. L'objectif est de récupérer les données du capteur de température DHT11/22 (Figure 4.1) et de les afficher sur une page Internet, accessible depuis n'importe où, à condition d'être connecté à Internet.

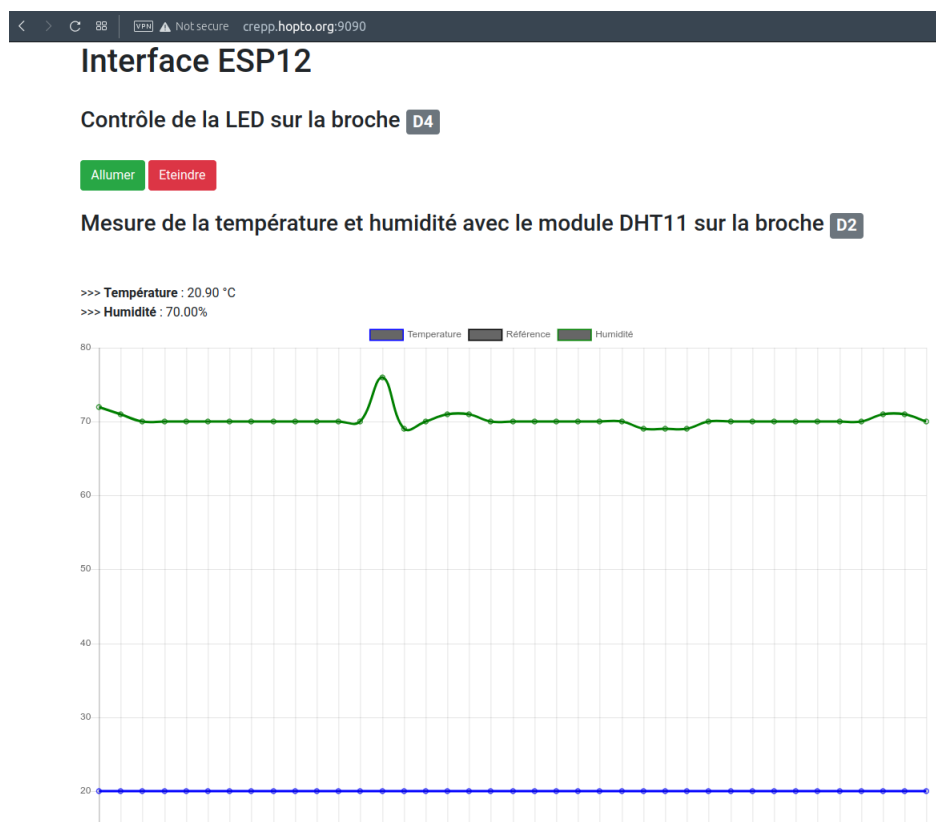


Figure 4.1 – Appel du serveur ESP12

Nous pouvons constater que le graphique est disponible à l'adresse crepp.hopto.org sur le port

9090¹ et que cette adresse pointe vers le serveur ESP12 mis en place chez l'auteur de ce document.

Vous saurez donc, à l'issue de ce tutoriel, configurer un routeur pour rendre accessible un serveur, en l'occurrence votre ESP12.

Matériel

- Un routeur (Box internet) avec les droits d'administration
- Un ESP12 avec un serveur actif configuré (Cf [Mise en place d'un serveur Web](#))

Configuration de base

Considérons un réseau privé avec l'adresse du réseau valant 192.168.0.0. Le masque de sous-réseau valant 255.255.255.0, il est donc possible de mettre $2^8 - 2 = 254$ périphériques sur ce réseau local².

Sur les figures suivantes, toutes les adresses sont fictives mais cohérentes en terme de longueur de champ.

De plus, des périphériques sont ajoutés (PC, smartphone) mais chez vous, vous aurez des adresses différentes et des périphériques différents.

Actuellement, notre serveur ESP12 est sur notre réseau local (figure 4.2), c'est à dire qu'il ne peut communiquer qu'avec les machines sur le réseau local, à savoir l'ordinateur avec l'adresse IP [IP 192.168.0.110](#) et le smartphone avec l'adresse IP [192.168.0.17](#).

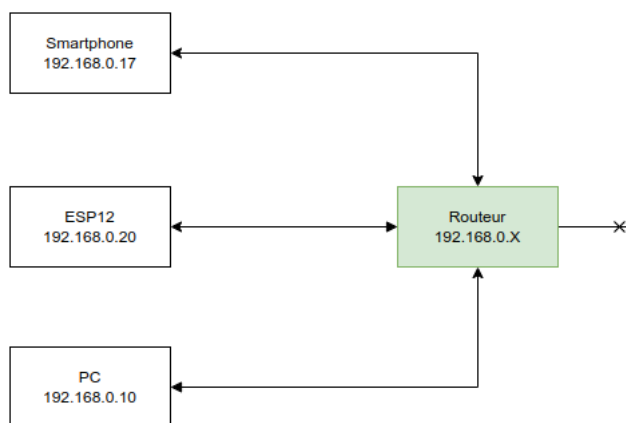


Figure 4.2 – Un réseau local

1. Nous reviendrons sur ce point plus tard dans le document.
2. Chez vous, il est très probable que vous ayez un réseau de ce type (classe C)

Adresse IP publique du routeur

Cependant, le routeur (Freebox, Livebox...) est relié à internet et possède donc une adresse IP **publique** car le routeur est vu comme un périphérique sur un réseau plus vaste : Internet.

Pour récupérer l'adresse publique, il est possible d'aller sur le site <http://www.mon-ip.com>. Une page similaire se lance avec l'adresse IP en question :



Figure 4.3 – L'adresse IP publique

Dans cet exemple, nous avons une adresse IPV6, il se pourrait que vous ayez une adresse IPV4 avec 4 champs au lieu des 8 ici.



Retenez bien cette adresse pour la suite

Avec l'adresse IP publique du routeur, un nouveau réseau apparaît :

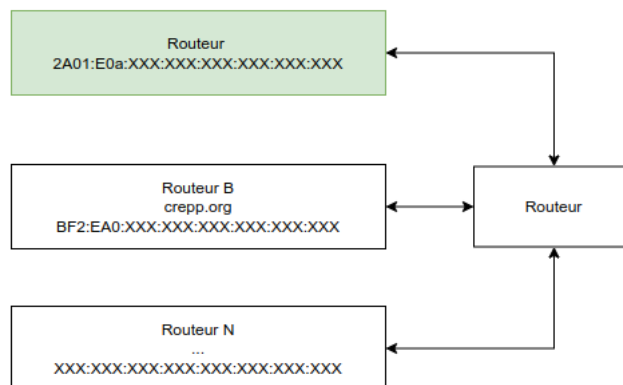


Figure 4.4 – Un réseau plus vaste

C'est par l'intermédiaire de cette adresse que votre réseau local a accès à Internet pour aller sur les différents sites (crepp.org....)

Configuration du routeur

Nous allons paramétrer notre routeur afin de faire ce que'on appelle une redirection de port. Pour cela, nous allons accéder à la page d'administration de notre routeur.

Accéder à la page d'administration

Voici l'adresse à saisir dans votre navigateur en fonction de votre Fournisseur Internet :

Fournisseur	Adresse à saisir
Free	mafreebox.freebox.fr
SFR	http://192.168.1.1
Orange	http://192.168.1.1

Table 4.1 – Adresse des routeur selon les fournisseurs

Ce tutoriel se base sur un routeur **Freebox** mais les notions sont similaires pour les autres routeurs. Après avoir saisi l'adresse de la Freebox dans mon navigateur Internet, l'interface suivante se lance :



Figure 4.5 – Portail de la Freebox

Il faut cliquer 2 fois sur **Paramètres de la Freebox** puis un message similaire apparaît :

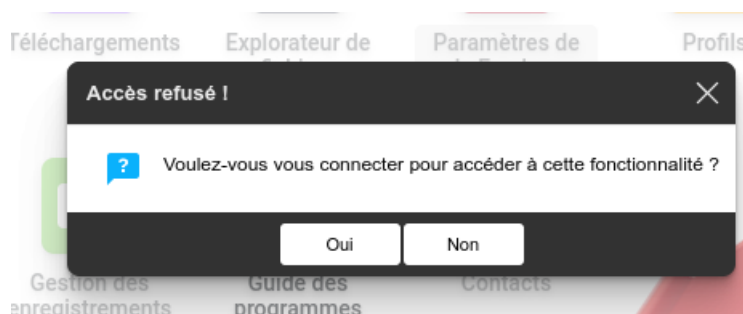



Figure 4.6 – Confirmation de connexion

Faites  puis il vous faudra renseigner le mot de passe de votre routeur, généralement écrit au dos du routeur. Une fois le mot de passe saisi, la page suivante apparaît :

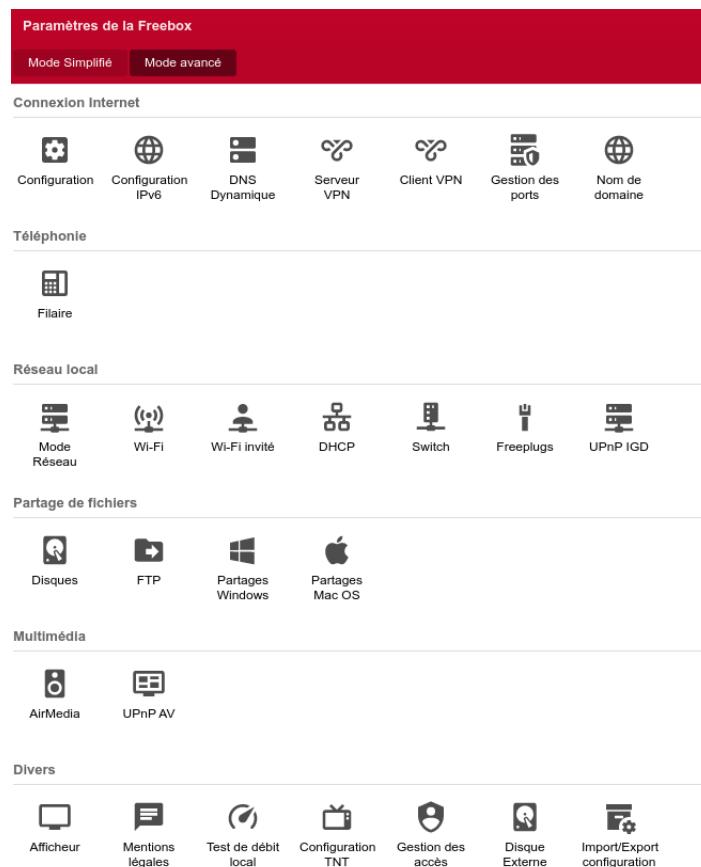


Figure 4.7 – Page personnelle de la Freebox

Veuillez vérifier que vous êtes en **mode avancé** ³

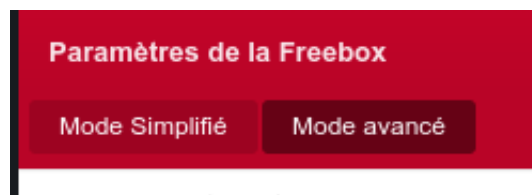


Figure 4.8 – Affichage des paramètres avancés

Une adresse statique

Pour la suite, nous devons impérativement mettre l'adresse IP de l'ESP12 (celle sur notre réseau local, à savoir dans notre exemple) en **statique**.

Question 1. Pourquoi donc ?

>>> 1. Tout simplement car, par la suite, lorsque nous souhaiterons accéder au serveur sur l'ESP12, si l'adresse IP de l'ESP12 change, nous n'y aurons plus accès !

3. En haut à gauche de la page

Configuration DHCP

Pour mettre une adresse IP statique, il faut faire un tour du côté du serveur DHCP⁴. C'est ce dernier qui va gérer l'attribution des adresses IP sur le réseau local.



Figure 4.9 – Ouverture des paramètres DHCP

La fenêtre suivante se lance :

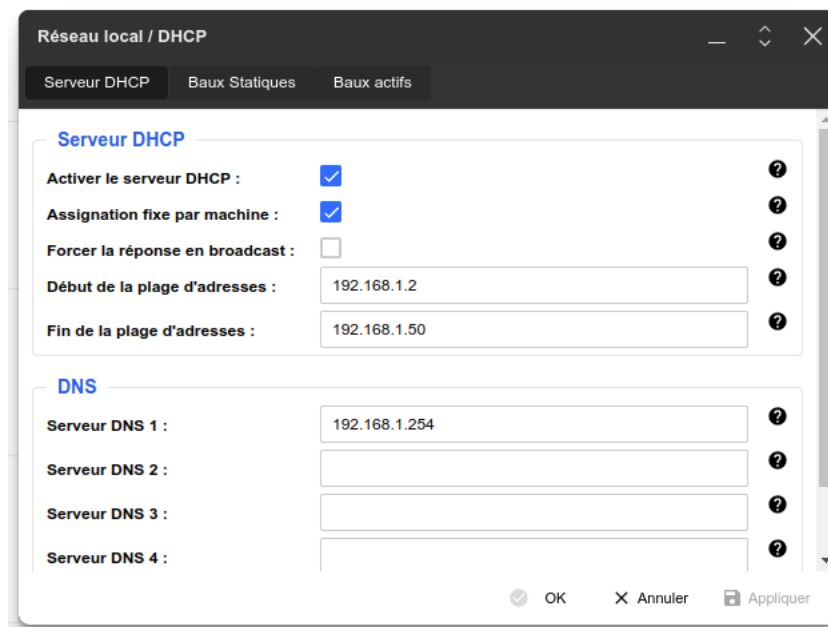


Figure 4.10 – Ouverture des paramètres DHCP

Ce qui nous intéresse, c'est la plage des adresses allouables par le serveur DHCP. Dans notre cas, nous voyons que les adresses des périphériques sur le réseau va de l'adresse `192.168.1.2` à `192.168.1.50`. Nous souhaitons mettre notre ESP12 sur l'adresse `192.168.1.20`, cette adresse est dans la plage ! Il est possible d'étendre la plage jusqu'à l'adresse `192.168.1.253`

Récupération de l'adresse MAC de l'ESP

Pour rendre une adresse IP statique de manière efficace, il faut connaître l'adresse MAC du périphérique, à savoir notre ESP12. Pour cela, nous allons télécharger le code suivant dans l'ESP12 :

```
#include <ESP8266WiFi.h>

void setup()
{
```

4. DHCP: *Dynamic Host Configuration Protocol* - Serveur qui gère l'attribution des adresses IP sur le serveur

```
Serial.begin(115200); //Vitesse de communication à 115200 bauds
}

void loop()
{
  Serial.print("ESP12 MAC Address: ");
  Serial.println(WiFi.macAddress());
  delay(1000);
}
```

Code 16 - Récupération de l'adresse MAC de l'ESP

Une fois le programme téléversé, veuillez ouvrir le moniteur série (Figure 4.11), ce dernier affichera toutes les secondes l'adresse MAC.

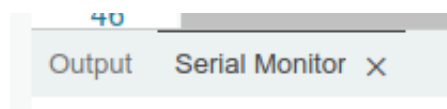


Figure 4.11 – Ouverture du moniteur série

Dans mon cas, l'adresse vaut **MAC C4 :5B :BE :64 :5E :98**

Ensuite, ouvrez de nouveau l'onglet **DHCP** dans les paramètres de votre routeur et aller dans la section **Bauds statiques**.

Un bail, en réseau, est une durée pour laquelle l'adresse IP ne change pas dans le temps. Avant, il était fréquent de trouver une période de bail valant 86480, c'est à dire le nombre de secondes dans une journée. cela voulait dire que l'adresse IP d'un périphérique changeait tous les jours. Notre objectif est de mettre une durée infinie.

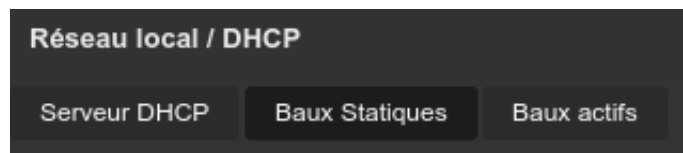


Figure 4.12 – Configuration des baux statiques

Puis cliquez sur **Ajouter un bail statique**

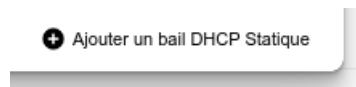


Figure 4.13 – Ajout du bail statique

Il ne reste plus qu'à renseigner l'adresse MAC précédemment trouvée puis l'adresse IP désirée. Il est possible de mettre un commentaire en supplément :

Figure 4.14 – Ajout du bail statique

En cliquant sur **Sauvegarder**, la fenêtre suivante résume la confirmation du bail statique.

Réseau local / DHCP		Commentaire
Serveur DHCP	Baux Statiques	Baux actifs
ESP-645E98 C4:5B:BE:64:5E:98 192.168.1.20		Serveur ESP12

Figure 4.15 – Confirmation du bail statique



Dorénavant, à chaque fois que notre ESP12 va se connecter au réseau local, il aura toujours la même adresse IP !

Nous pouvons désormais faire la redirection de port.

Faire une redirection de port

Depuis la page principale du routeur, nous allons nous rendre dans la section **Gestion des ports** :

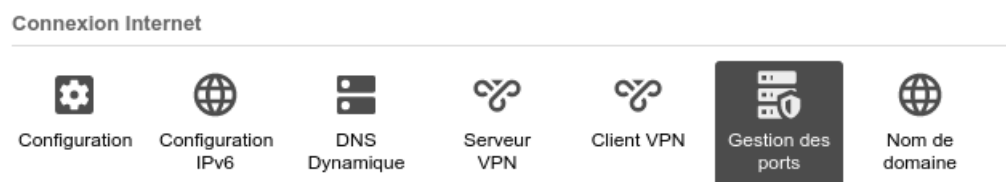


Figure 4.16 – Gestion des ports

Puis, en bas à droite de la nouvelle fenêtre, **Ajouter une redirection** :

Rafraîchir + Ajouter une redirection

Figure 4.17 – Bouton d'ajout de redirection de port

Une interface se lance :

Figure 4.18 – Paramètres de redirection

Voici les éléments à renseigner :

- **IP Destination**
L'IP statique de votre ESP12
- **Redirection active**
Veuillez cocher la case
- **IP sources**
Toutes
- **Protocole**
Nous choisirons le protocole TCP⁵ à la place du protocole UDP⁶. La grande différence réside dans le fait que pour le TCP, l'intégrité des données est analysée pour voir si des paquets ont été perdus, contrairement à l'UDP qui ne tient pas compte des pertes de paquets sur le réseau.
- **Port de début**
Nous utiliserons le port 9090⁷
- **Port de fin**
Nous utiliserons le port 9090 également. Il sera mis à la même valeur que le port de début.
- **Port de destination**
Il s'agit du port d'écoute de l'ESP12. Dans tous les codes pour l'ESP12 en mode serveur, nous avons utilisé une macro appelée **PORT** valant 80. Par défaut, notre serveur écoutait sur le port 80. Ici,

5. TCP: *Transmission Control Protocol* - Protocole réseau qui permet à deux hôtes de communiquer

6. UDP: *User Datagram Protocol* - Protocole réseau qui permet de transmettre des données en mode broadcast

7. Valeur arbitraire mais comprise entre 6000 et 10000, la plage des ports peu utilisés.

nous avons la possibilité de changer le port.

En terme de sécurité, il vaut mieux éviter d'utiliser le port 80 car c'est le port par défaut. Nous mettrons le port **9090** à la place. La section suivante explique plus en détail les ports.

La ligne suivante apparaît pour résumer la redirection :

Active	Protocole: tcp WAN : 9090 LAN: 9090 Commentaire:	Toutes	ESP-645E8
--------	---	--------	---------------------------

Figure 4.19 – Confirmation du port et de la redirection

Un peu plus de détails sur les ports

Un port peut être vu comme une porte d'entrée d'un réseau à travers un périphérique réseau (ordinateur, routeur....). Par exemple, un ordinateur relié à un réseau local peut utiliser différentes fonctionnalités :

- Un serveur de fichier
- Aller sur Internet
- Se connecter à un serveur de mail

Tous ces services utilisent des ports prédéfinis pour faire transiter les informations. Par exemple, le port 80 est le port pour aller sur Internet, le port 443 est également un port pour Internet mais utilisant le protocole HTTPS.

Notre routeur, bien qu'étant relié physiquement à un seul câble, peut être scindé en une multitude de canaux **virtuels** appelé ports, ayant des numéros compris entre 1 et 65535.

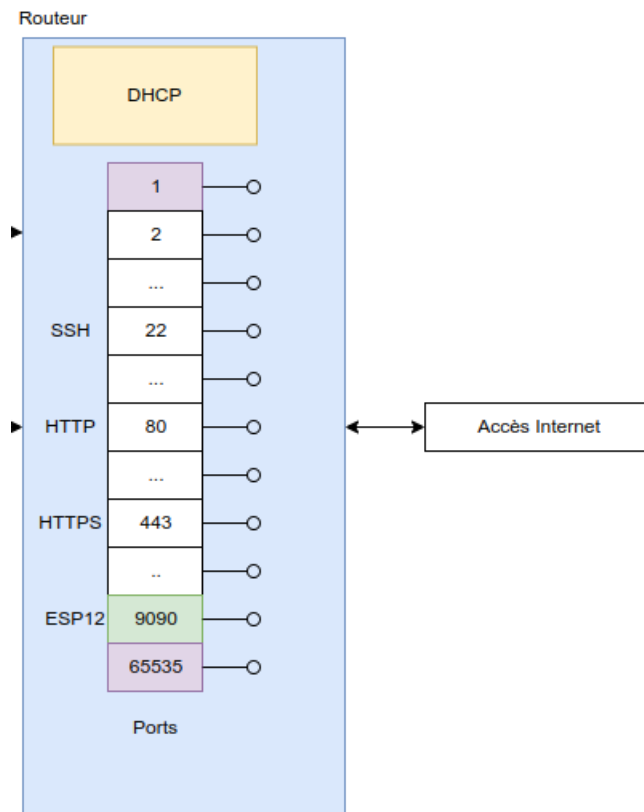


Figure 4.20 – Représentation des ports du routeur

Parmi ces numéros, certains ports sont déjà utilisés par des services dédiés (SSH⁸, SFTP⁹ ...), nous ne pouvons donc pas utiliser tous les numéros. Cependant, les ports entre 6000 et 10000 sont rarement utilisés et peuvent être utilisés à des fins de redirection.

Enfin, notre ESP12 possède également des ports et nous définissons le port d'écoute avec la ligne suivante dans nos codes de serveur pour ESP12 :

```
#define PORT 80 //Valeur à changer dans le code
```

Code 17 - Récupération de l'adresse MAC de l'ESP



Lors de l'ouverture de notre serveur au monde extérieur, il ne faudra pas oublier de changer ce numéro

Notre réseau devient donc comme ceci :

8. SSH: *Secure Shell* - Protocole de communication sécurisé

9. SFTP: *Secure File Transfer Protocol* - Protocole de communication sécurisé dédié au transfert de fichiers sur le réseau

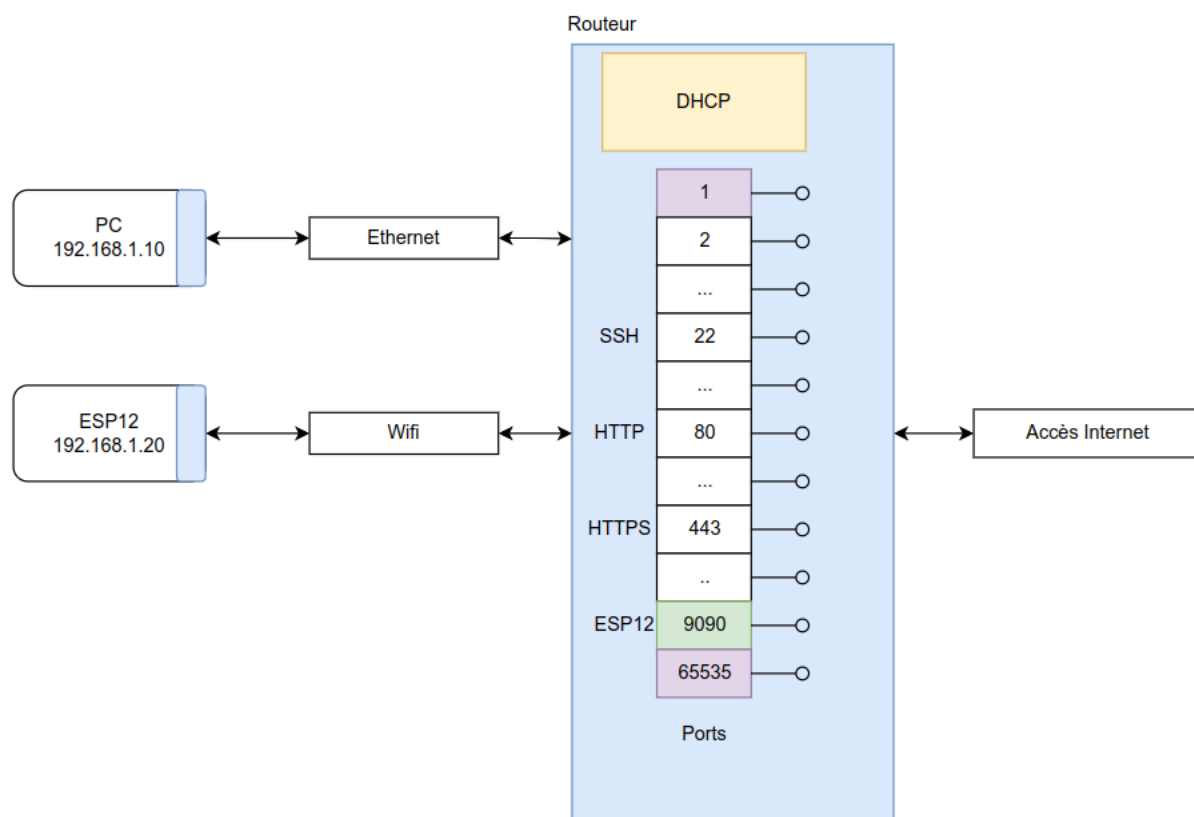


Figure 4.21 – Représentation des ports

Section 5

Configuration d'un nom de domaine personnalisé

Cette section a pour but de mettre en place un nom de domaine personnalisé afin d'accéder à son serveur en écrivant un nom de domaine à la place d'une adresse IP.

Question 2. *Quel est l'intérêt ?*

>>> **2.** *Il est plus facile de retenir **crepp.org** plutôt que de saisir une adresse IP comme suivant :*

IP 76.152.14.77

Les serveurs DNS¹ font donc le lien entre les adresses IP et les nom de domaine.



Nous utiliserons le service **No-IP** pour obtenir un nom de domaine gratuit.

Création du compte

Veuillez vous rendre à l'adresse suivante : no-ip.com.

Vous tombez sur une interface similaire :

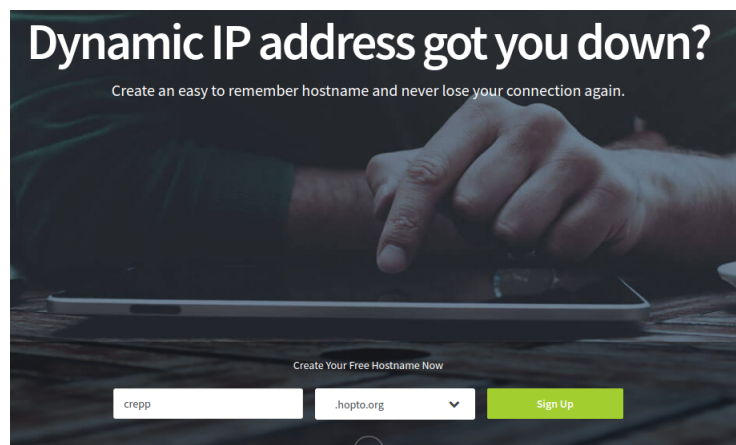


Figure 5.1 – Interface d'accueil de No-IP

1. DNS: *Domain Name System* - Service qui fait le lien entre un nom de domaine et son adresse IP

Vous cliquez ensuite sur  **Sign Up** :

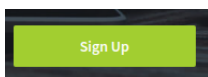


Figure 5.2 – Bouton Sign Up

Un formulaire doit être renseigné avec une adresse mail et un mot de passe pour le future compte **No-IP** .

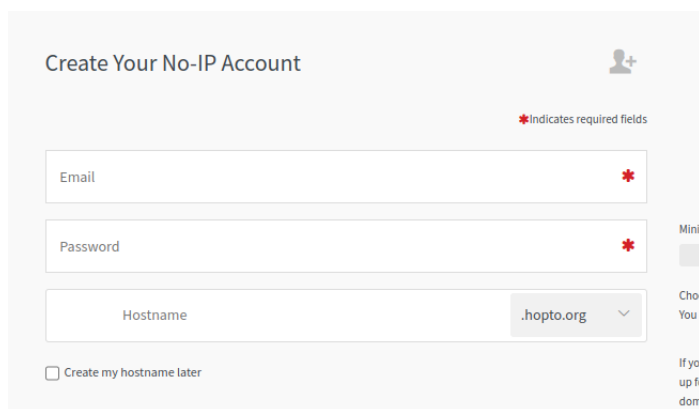


Figure 5.3 – Formulaire d'inscription

Veuillez cocher la case **Create my hostname later** puis validez le formulaire. Vous arrivez sur une interface similaire :

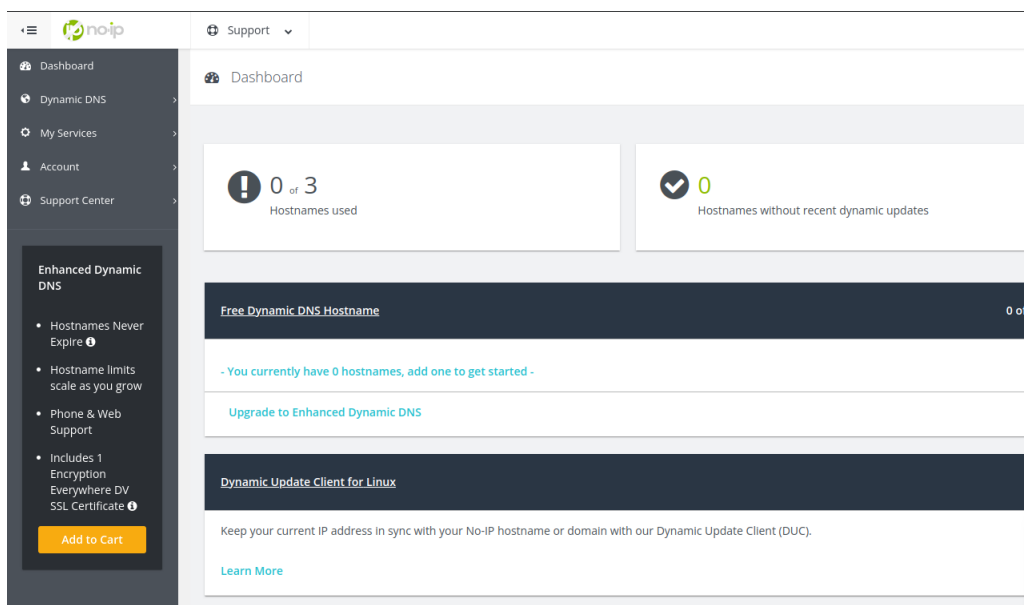


Figure 5.4 – Page principale du compte No-IP

Cette page recense les principales informations, notamment le nombre de nom de domaine utilisé avec votre compte.

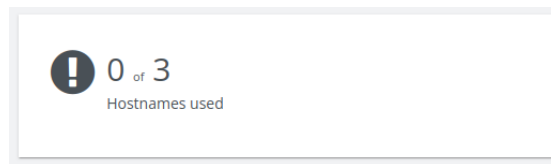


Figure 5.5 – Nombre de nom de domaine configuré

Il est possible de créer 3 noms de domaine avec votre compte

Ajout d'un nom de domaine

Nous allons ajouter un nom de domaine pour notre serveur.

Dans le cadre de ce tutoriel, j'ai choisi le nom de domaine **crepp.hopto.org**.
Veuillez cliquer sur le panneau indiquant le nombre de nom de domaine utilisés :

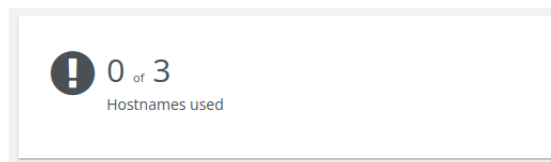


Figure 5.6 – Nombre de noms de domaine configurés

Puis sur le bouton **Create hostname** :

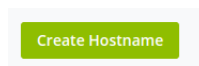


Figure 5.7 – Ajout d'un nom de domaine



Vous pouvez choisir à ce moment-là un nom de domaine, c'est à dire votre adresse que vous saisissez pour vous connecter à votre serveur ESP12

Il vous faudra renseigner l'adresse IP publique que vous avez récupéré précédemment et indiquer le nom de domaine choisi, qui doit être unique.

Create a Hostname

Hostname ⓘ

Domain ⓘ

Record Type

IPv4 Address ⓘ

Wildcard ⓘ

MX Records

crepp

hopto.org

☒ DNS Host (A) ⓘ

☐ AAAA (IPv6) ⓘ

☐ DNS Alias (CNAME) ⓘ

☐ Web Redirect ⓘ

Manage your Round Robin, TXT, SRV and DKIM records.

91.168.158.141

Upgrade to Enhanced

to enable wildcard hostnames.

+ Add MX Records

Cancel

Create Hostname

Figure 5.8 – Saisie de l'adresse publique

Si vous avez une adresse publique IPV6, veuillez cocher la case **AAAA (IPV6)**

Mise en place de l'ESP12

Nous allons mettre en place le code pour l'ESP12.

Pour cela, nous allons réutiliser le code du serveur pour afficher la température et l'humidité et effectuer le branchement suivant :

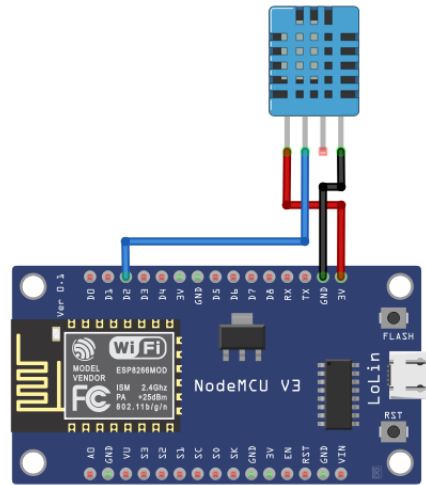


Figure 5.9 – branchement du capteur de température



Le code complet est disponible en faisant un clic-droit puis **Copy link Address** sur le lien suivant : [Lien du code](#) .
Il ne vous reste plus qu'à coller le lien dans votre navigateur Internet.

Il faut cependant changer le port d'écoute du serveur !

La ligne suivante dans le code de l'ESP12 :

```
#define PORT 80 //Port par défaut
```

Code 18 - Ancien port

Va devenir :

```
#define PORT 9090 //Nouveau port
```

Code 19 - Nouveau port

Après avoir compilé le code, ouvrez votre navigateur et saisissez le nom de domaine que vous avez choisi ainsi que le port.



Figure 5.10 – Votre nom de domaine dans le navigateur Internet

Questionnaire

Quiz

1. Le masque de sous-réseau permet de :

Masquer le réseau
pour les utilisateurs

Déterminer l'adresse
IP du routeur

De créer un nouveau
sous-réseau parallèle

2. L'adresse IP de l'ESP12 doit être :

statique

dynamique

3. Votre routeur est une passerelle possible pour accéder au monde extérieur (Internet) :

Vrai

Faux

4. Le serveur qui gère la correspondance entre les adresses IP et les noms de domaine est le serveur :

HTTP

DHCP

DNS

5. Le serveur qui gère l'attribution des adresses IP est le serveur :

DHCP

Apache

HTTP