TP N° 01:

Introduction

Le but de ce TP/Lab est de se familiariser avec les fonctionnalités de l'Arduino MKR , les capteurs et de l'IDE, ainsi que d'assembler notre premier setup. Nous avons fait cela en écrivant un programme Arduino fonctionnel simple en utilisant des composants externes avec l'Arduino MKR.

Partie théorique :

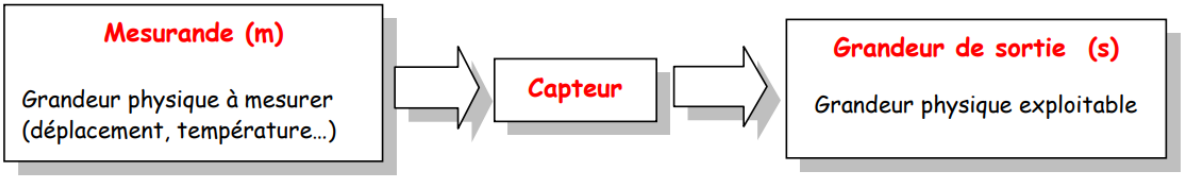
1. Familiarisation avec les capteurs :

a. Définition théorique de capteurs

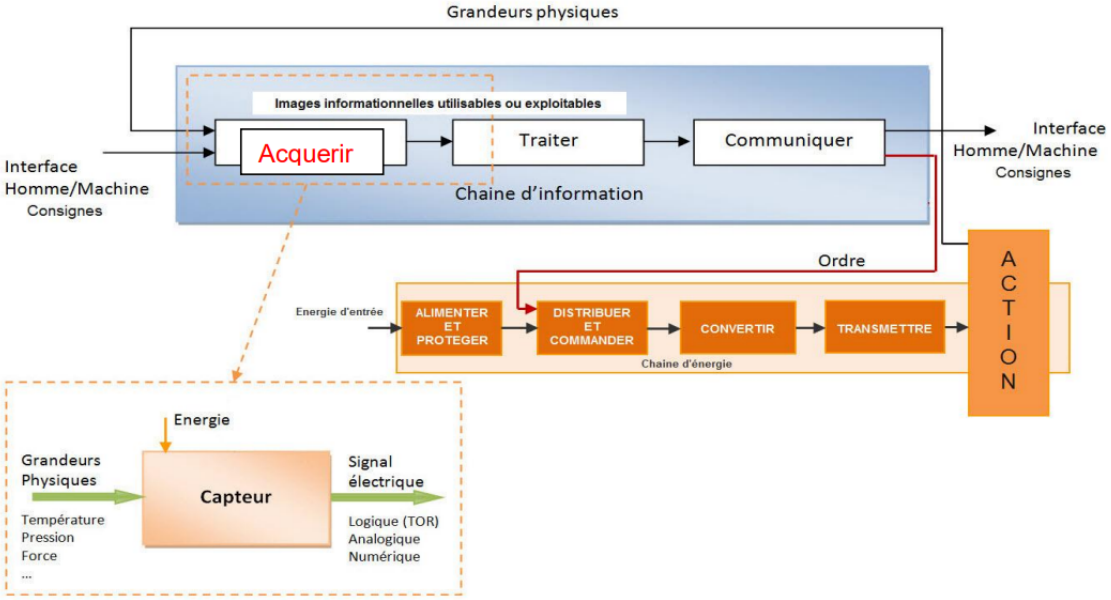
Les capteurs sont des composants matériels, qui une fois correctement connectés à votre carte Arduino, peuvent fournir des informations sur le monde extérieur. Ceux-ci ont comme rôle de saisir une grandeur physique et de la convertir en information numérique. Il est ainsi possible de capter la plupart des phénomènes physiques comme le vent, la pression, la température, la lumière, la torsion, la distance, etc.

C'est quoi un capteur ?

Un capteur est un constituant ou un organe capable d'acquérir une grandeur physique à mesurer, et de la transformer en une grandeur exploitable par une unité de traitement. Le capteur est caractérisé par sa fonction : **s = F(m)** où s' est la grandeur de sortie ou la réponse du capteur.



Fonctionnement d'un capteur :



Classification des capteurs:

Les capteurs peuvent être classifiés selon des critères différents ; la grandeur physique observée, le type de l'information renvoyée , le temps de réponse , sa sensibilité , sa précision … etc.
Nous allons se concentrer sur le classement des capteurs selon les critères suivants :

I. La grandeur observée :

En fonction de la caractéristique électrique de la grandeur de sortie, on peut classer les capteurs en deux grandes familles : les capteurs passifs et les capteurs actifs.

Les capteurs passifs :

Dans la plupart des cas, les capteurs passifs ont besoin d'une énergie extérieure pour fonctionner (Température , Humidité …), ils sont souvent modélisés par une impédance. Une variation du phénomène physique étudié (mesuré) engendre une variation de l'impédance. Il faut leur appliquer une tension pour obtenir un signal de sortie. Le capteur se comporte en sortie comme un dipôle passif qui peut être résistif, capacitif ou inductif. En fonction de la grandeur mesurée, on utilise plusieurs effets pour réaliser la mesure.

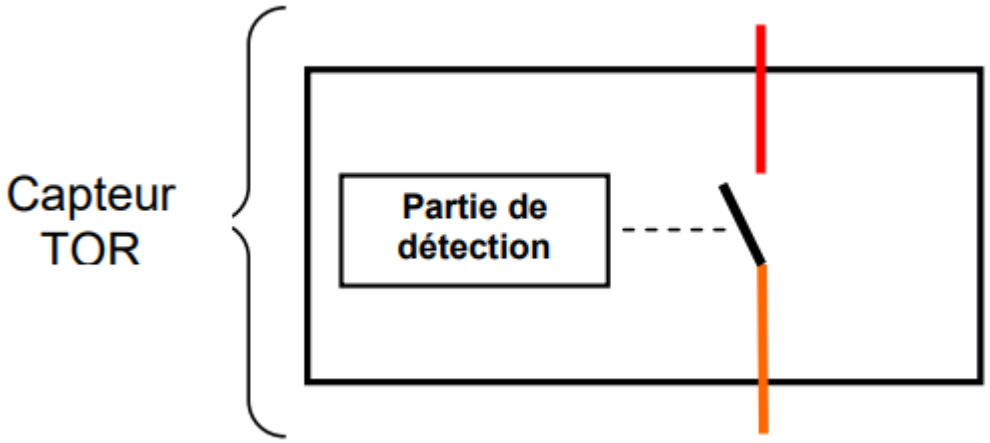
Les capteurs actifs :

Lorsque le phénomène physique qui est utilisé pour la détermination du mesurande effectue directement la transformation en grandeur électrique, on est en présence d'un capteur actif. C'est la loi physique elle-même qui relie le mesurande et grandeur électrique de sortie. La sortie du capteur est assimilée à un générateur. C'est un dipôle actif qui peut être du type courant, tension ou charge électrique Q en coulombs. Certains principes physiques peuvent être mis en jeu.

II. Le type de l'information renvoyée :

Les capteurs logiques ou TOR:

Ce type de capteur renvoie une information de type logique c'est-à-dire 0 ou 1. En d'autres termes, la sortie peut prendre uniquement deux états. Par exemple, un clavier est une matrice de capteurs logiques, car chaque touche est un interrupteur (ou un contact) qui peut être soit « ouvert » soit « fermé ».



Les capteurs analogiques :

Ce type de capteur renvoie une valeur proportionnelle à la grandeur physique mesurée. Par exemple, une photorésistance (capteur de lumière) est un capteur qui convertit la luminosité en valeur électrique. Il existe des capteurs analogiques qui renvoient une information linéaire et d'autres dont la réponse suit une courbe différente. Par exemple, un potentiomètre linéaire sera noté B (nomenclature japonaise) et A (nomenclature

européenne) tandis qu'un potentiomètre logarithmique sera noté A (nomenclature japonaise) et B (nomenclature européenne).

Courbe linéaire

Courbes non linéaire

Pour un capteur, une courbe linéaire signifie que la progression de sa valeur suit une progression constante et régulière. Une courbe non linéaire appartient à une fonction plus complexe dont la valeur ne progresse pas également dans le temps.

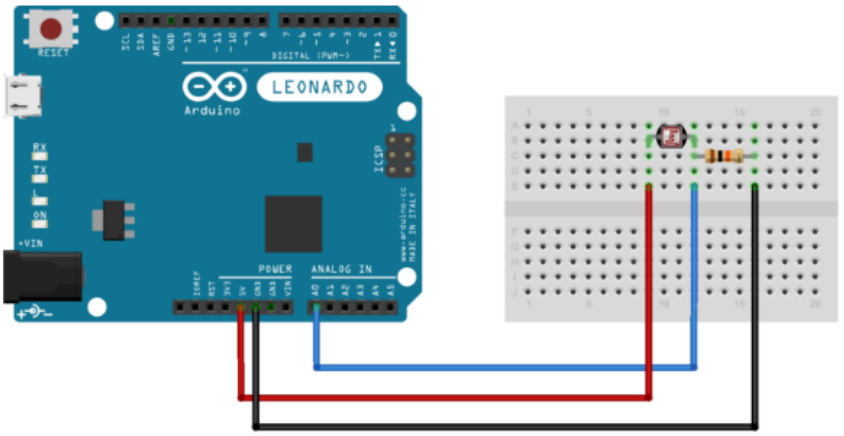
b. Décrire deux capteurs au choix.

1. Capteur LDR (photorésistance):

Le module LDR pour Light Dependent resistor ou en français photorésistance est un capteur passif qui modifie sa résistance selon le niveau de lumière visible. Sa résistance est plus faible en pleine lumière. Passant de 50 K ohm dans l'obscurité à 500 ohm en pleine lumière.

Pour pouvoir utiliser l'information fournie par la photorésistance il faudrait convertir la valeur de la résistance en une tension linéaire. Le montage le plus facile à faire est d'utiliser le principe du diviseur de tension, en ajoutant une résistance fixe en série ; puis utiliser la fonction **`analogRead()`** sur un pin analogique.

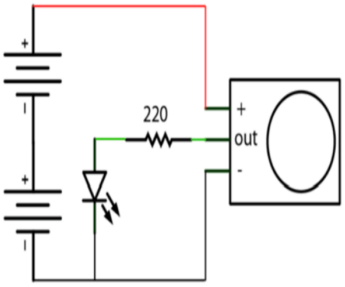
Montage :



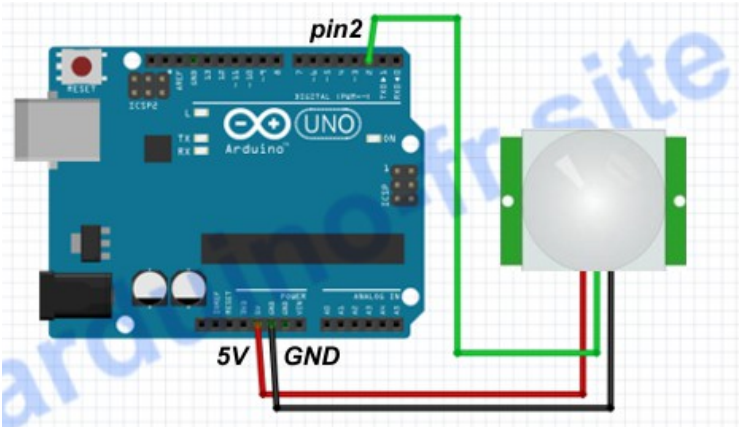
2. Détecteur de mouvement (capteur infrarouge passif):

Un capteur passif infrarouge (en anglais PIR ou bien HC-SR501) est probablement l'alarme anti-vol la plus courante. Tous les objets chauds émettent une lumière infrarouge invisible. Un capteur PIR réagit au changement de lumière infrarouge. Cette modification est en général provoquée par la chaleur d'un être humain qui se déplace.

Principe de fonctionnement:
Une fois le capteur PIR brancher il lui faut 30 secondes pour s'adapter à son milieu. Une fois cela effectuer le moindre mouvement d'une source de chaleur devant lui sera détecter est un signal sera envoyer à l'Arduino, sous la forme d'une impulsion.

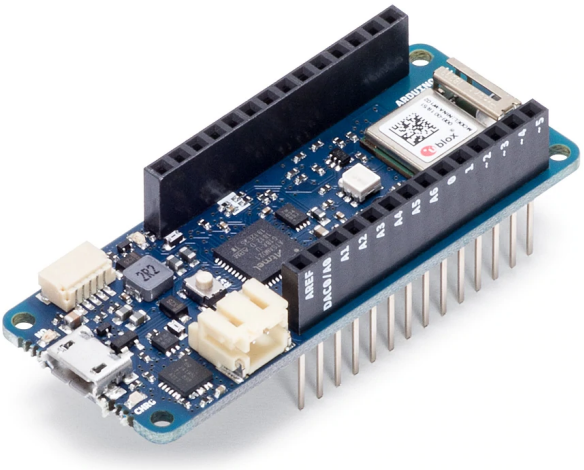


Montage :



2. Familiarisation avec Arduino

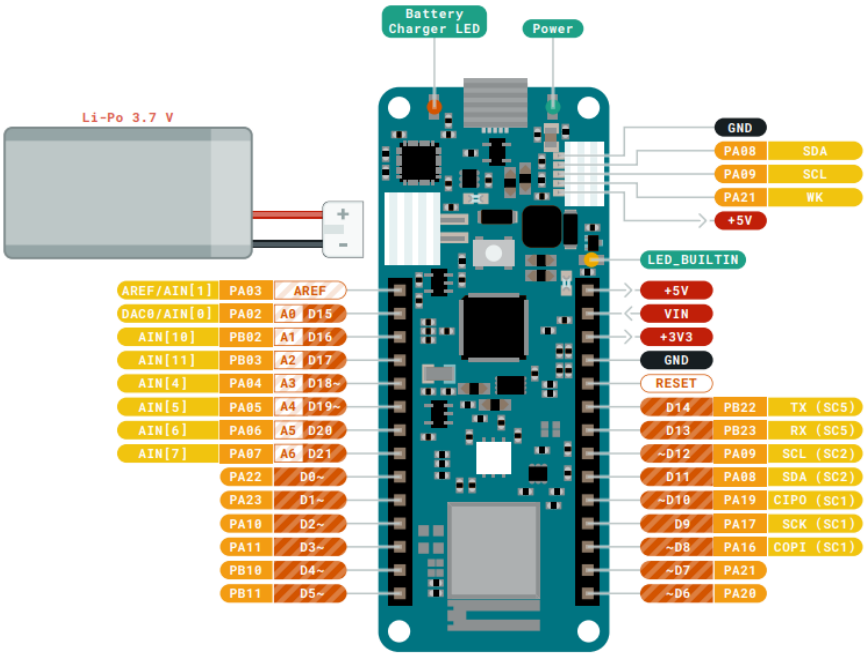
Arduino est un circuit imprimé qualifié de libre et open-source . Seuls le nom et le logo sont réservés. Sur Arduino se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques. En gros, on y branche des capteurs, le programme de l'Arduino traite les informations données par le capteur et déclenche des actions (comme allumer ou éteindre, augmenter, diminuer...). Il peut être utilisé pour effectuer des tâches très diverses comme la charge de batteries, la domotique (le contrôle des appareils domestiques (éclairage, chauffage...), le pilotage d'un robot, etc. Arduino peut être utilisé pour construire des objets interactifs indépendants (prototypage rapide), ou bien peut être connecté à un ordinateur pour communiquer avec ses logiciels.



Description de la carte Arduino MKR WIFI 1010

a. **Aspect hardware :**

Arduino MKR WiFi 1010 est la première carte à intégrer la connectivité WiFi et Bluetooth.
Forme : Il mesure 61,4 mm x 25 mm et ne pèse que 32 grammes.
Puissance: Il peut fonctionner avec 5V où 3.3V
Processeur: La carte est livrée avec un processeur ARM SAMD21 Cortex-M0 + 32 bits de faible puissance.
Mémoire: La carte a 256 Ko de mémoire flash et 32 Ko de SRAM.



Broches E/S : L'Arduino MKR WiFi 1010 offre:

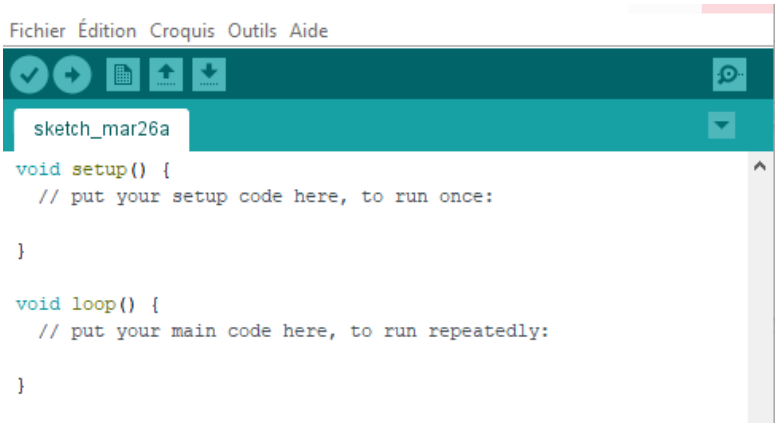
- 22 broches d'E / S numériques, broches 0 - 21
- 7 broches d'entrée analogique (ADC 8/10/12 bits), broches A0 / A6.
- 1 broche de sortie analogique (DAC 10 bits), la broche identifiée par DAC0 / A0.

Le courant CC par broche d'E / S est de 7 mA.
Les broches 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 18, 19 sont des broches PWM.

b. Structure d'un programme Arduino :

Comment utiliser une carte arduino ?

1. Télécharger la dernière version d'Arduino version 1.8.8 sur: https://www.arduino.cc/en/main/software
2. Lancer l'exécutable.
3. Connecter la carte ARDUINO à l'ordinateur à l'aide du câble USB Connecter la carte à l'ordinateur en utilisant le câble USB. Une lumière verte ou rouge doit s'allumer sur la carte, intitulée «ON».
4. Installer le driver USB de la carte Arduino (COM3).
5. Lancer l'IDE Arduino et choisir votre carte.
6. Écrire le sketch (programme) qui va s'exécuter sur la carte Arduino (en C/C++).



Les programmes Arduino peuvent être divisés en trois parties principales : Structure, Valeurs (variables et constantes), et Fonctions.Commençons par la structure. La structure du logiciel se compose de deux fonctions principales :

- La fonction `setup()`
- La fonction `loop()`

La fonction **`setup()`** est appelée lorsqu'un sketch démarre. On l'utilise pour initialiser les variables, les modes des broches, commencer à utiliser les bibliothèques, etc. La fonction setup ne sera exécutée qu'une seule fois, après chaque mise sous tension ou réinitialisation de la carte Arduino.

Après avoir créé une fonction **`setup()`**, qui initialise et définit les valeurs initiales, la fonction **`loop()`** fait précisément ce que son nom suggère, et boucle consécutivement, permettant à votre programme de changer et de réagir. Utilisez-la pour contrôler activement la carte Arduino.

Partie pratique :

a. Branchement d'une LED :

Matériel utilisé :

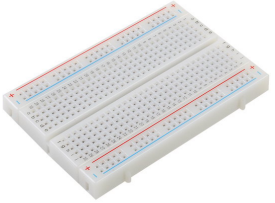
LED

LED (Diode électro-luminescente) : lorsque vous observez une LED, on note que l'un des connecteurs est plus long que l'autre. Le plus long (anode) sera connecté à la borne positive du circuit, alors que le plus court (cathode) sera connecté à la borne négative aussi appelée "ground" (GND) ou "masse". Elle sert principalement pour la signalisation.

Résistance

Une résistance est un composant électronique ou électrique dont la principale caractéristique est d'opposer une plus ou moins grande résistance (mesurée en ohms) à la circulation du courant électrique

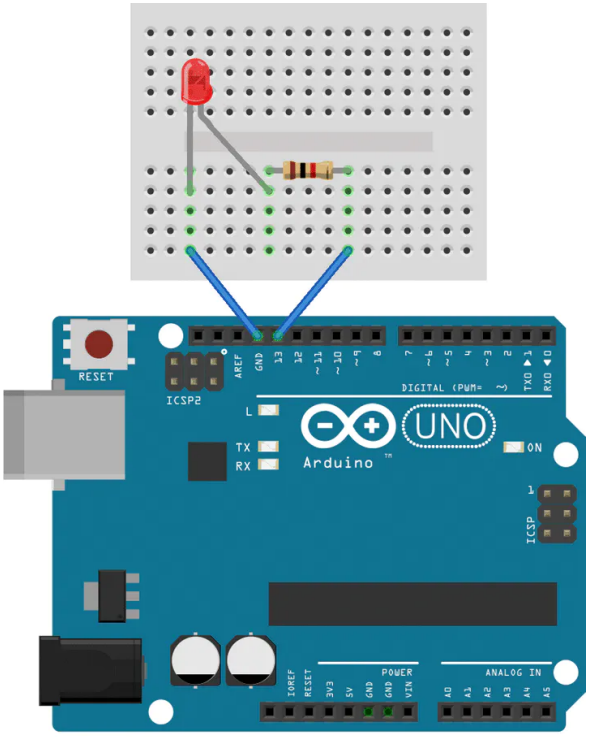
Breadboard

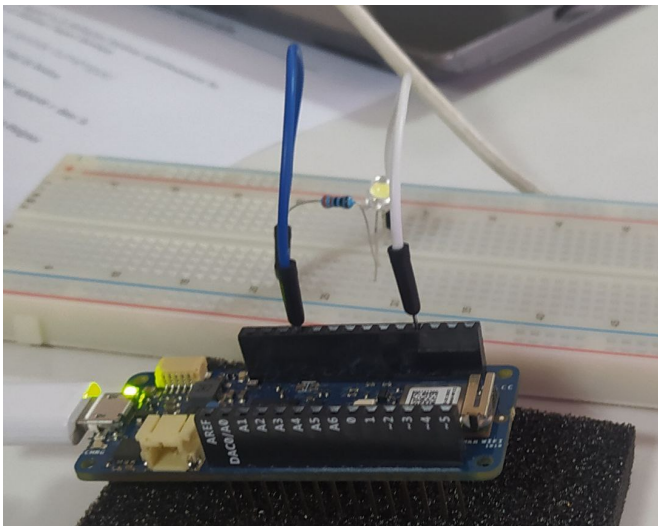


Une plaque d'essai permet de réaliser des montages électroniques sans soudure générale, les plaques d'essais sont de forme rectangulaire. Il y a plusieurs rangées de trous: certaines rangées sont verticales tandis que d'autres sont horizontales. Elle s'utilise avec des

≪straps≫ qui sont des fils de cuivre isolés, de longueur et couleur variables.

Montage:





Programme:

1/ En utilisant une LED interne (LED BUILTIN):

```
// la fonction setup s'exécute une fois lorsque on appuie sur reset ou
// allumer la carte
void setup() {
 // initialise le digital pin LED_BUILTIN (led interne de l'arduino)
 // comme une OUTPUT
  pinMode(LED_BUILTIN, OUTPUT);
}

// La fonction loop s'exécute tant que l'arduino est connecté (toujours)
void loop() {
// allumer la LED on mettant le voltage à la valeur HIGH (5V)
  digitalWrite(LED_BUILTIN, HIGH);
// attendre pour une seconde
  delay(1000);
// eteindre la LED en mettant le voltage à la valeur LOW (0V)
  digitalWrite(LED_BUILTIN, LOW);
// attendre pour une seconde
  delay(1000);
}
```

2/ En utilisant une LED externe :

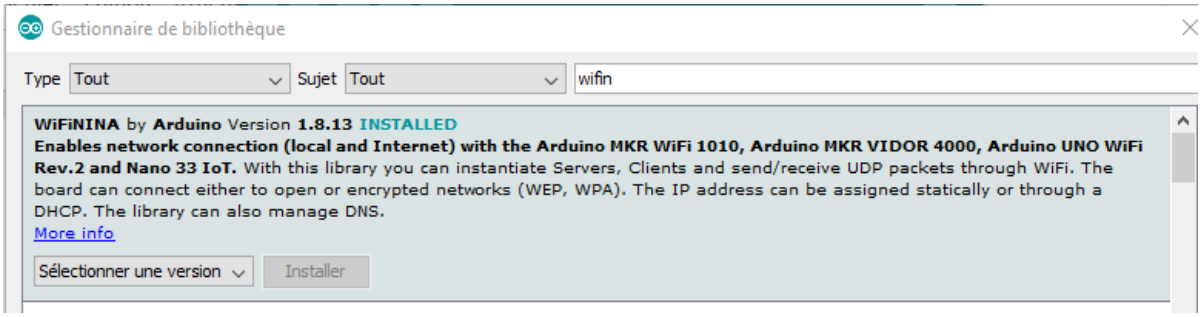
```
//numéro du pin dans la carte
int led = 6;
void setup() {
 // initialise le digital pin 6 comme une OUTPUT
  pinMode(led, OUTPUT);
}

// La fonction loop s'exécute tant que l'arduino est connecté (toujours)
void loop() {
// allumer la LED on mettant le voltage à la valeur HIGH (5V)
  digitalWrite(led, HIGH);
// attendre pour 5 secondes
// en modifiant la valeur du delay on remarque des blink à des délais
// différents
  delay(5000);
// eteindre la LED en mettant le voltage à la valeur LOW (0V)
  digitalWrite(led, LOW);
// attendre pour 5 secondes
  delay(5000);
}
```

b. Scan des réseaux sans fils WIFI:

Librairies utilisées : **<WiFiNINA.h>**

Fonction utilisé pour scanner les réseaux : **WiFi.scanNetworks();**

Arduino - WiFiScanNetworks



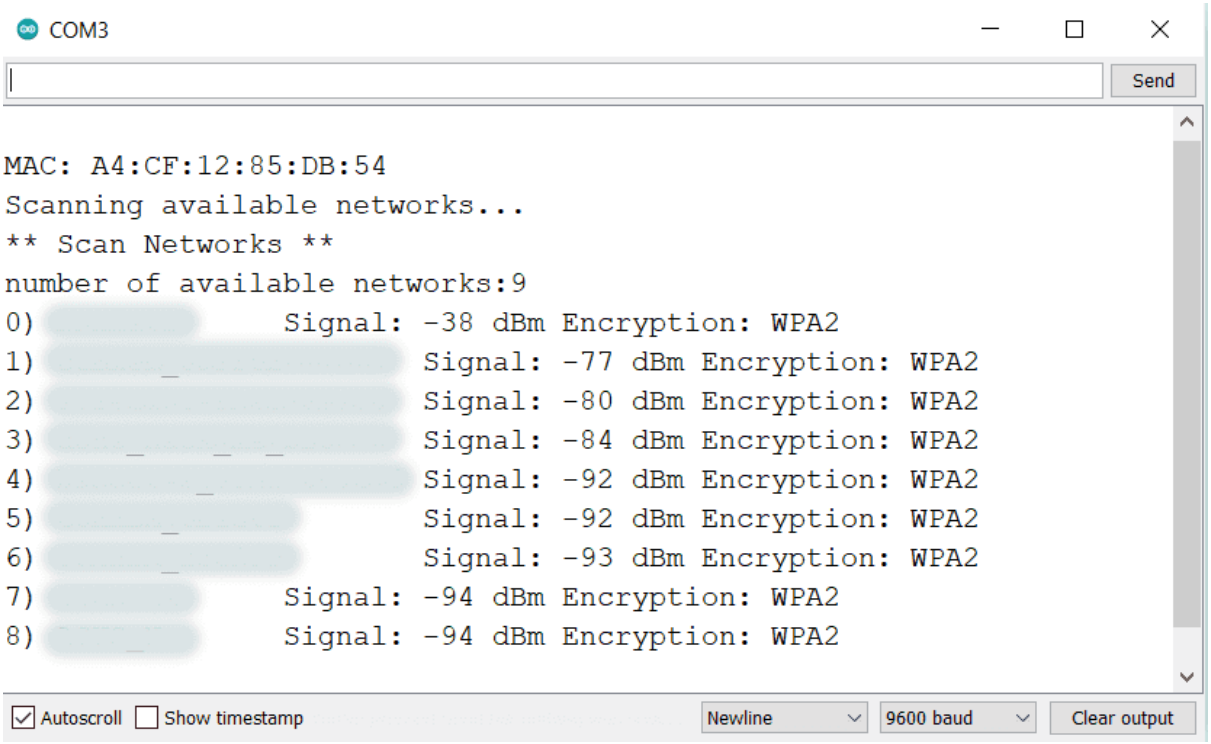
Programme:

Ce programme recherche un réseau 802.11b/g avec l'une des cartes qui supportent la bibliothèque **WIFI NINA** (dans ce cas MKR WIFI 1010). Le moniteur série du logiciel Arduino (IDE) imprimera des informations sur la carte et les réseaux qu'elle peut voir. Il ne se connectera pas à un réseau.

```
#include <SPI.h>
#include <WiFiNINA.h>

void setup() {
  //Initialization du port série
  Serial.begin(9600);
  //attendre que le port série se connecte. Nécessaire pour le port
  // USB natif uniquement
```

```
  while (!Serial) { ; }
  // vérifier l'existence de la librairie WIFI
  if (WiFi.status() == WL_NO_MODULE) {
    Serial.println("Communication with WiFi module failed!");
    while (true);
  }
  String fv = WiFi.firmwareVersion();
  if (fv < WIFI_FIRMWARE_LATEST_VERSION) {
    Serial.println("Please upgrade the firmware");
  }
  // affiche l'adresse MAC de la carte réseau
  byte mac[6];
  WiFi.macAddress(mac);
  Serial.print("MAC: ");
  printMacAddress(mac);
}
void loop() {
  // scan les réseaux existants
  Serial.println("Scanning available networks...");
  listNetworks();
  delay(10000);
}
void listNetworks() {
  // scan les réseaux à proximité
  Serial.println("** Scan Networks **");
  int numSsid = WiFi.scanNetworks();
  if (numSsid == -1) {
    Serial.println("Couldn't get a wifi connection");
    while (true);
  }
  // affiche la liste des réseaux détectés
  Serial.print("number of available networks:");
  Serial.println(numSsid);
  // affiche le ssid et les informations relatifs au réseaux détectés
  for (int thisNet = 0; thisNet < numSsid; thisNet++) {
    Serial.print(thisNet);
    Serial.print(") ");
    Serial.print(WiFi.SSID(thisNet));
    Serial.print("\tSignal: ");
    Serial.print(WiFi.RSSI(thisNet));
    Serial.print(" dBm");
    Serial.print("\tEncryption: ");
    printEncryptionType(WiFi.encryptionType(thisNet));
  }}
//Fonctions affichage Encryption & MAC
//...
```

Affichage série:


COM3
Send
MAC: A4:CF:12:85:DB:54
Scanning available networks...
** Scan Networks **
number of available networks:9
0) Signal: -38 dBm Encryption: WPA2
1) Signal: -77 dBm Encryption: WPA2
2) Signal: -80 dBm Encryption: WPA2
3) Signal: -84 dBm Encryption: WPA2
4) Signal: -92 dBm Encryption: WPA2
5) Signal: -92 dBm Encryption: WPA2
6) Signal: -93 dBm Encryption: WPA2
7) Signal: -94 dBm Encryption: WPA2
8) Signal: -94 dBm Encryption: WPA2
Autoscroll
Show timestamp
Newline
9600 baud
Clear output