

# Labo communication

# Arduino-PC

---

Projet S2ei

Hiver 2025



# Installsallation de VScode

---

Installer VScode sur votre ordinateur

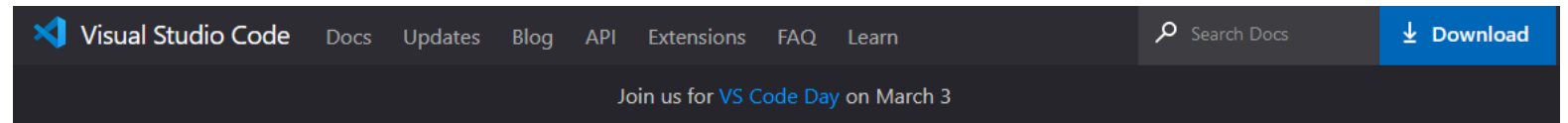
Ajouter l'extension PlatformIO

Ajouter le compilateur MinGW

# Installation de VScode


Télécharger VScode

<https://code.visualstudio.com/download>



## Download Visual Studio Code


Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows

Windows 7, 8, 10, 11

User Installer	64 bit	32 bit	ARM
System Installer	64 bit	32 bit	ARM
.zip	64 bit	32 bit	ARM



↓ .deb


Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE

.deb	64 bit	ARM	ARM 64
.rpm	64 bit	ARM	ARM 64
.tar.gz	64 bit	ARM	ARM 64

Snap Store



↓ Mac

macOS 10.11+

.zip Universal Intel Chip Apple Silicon

# Installation de PlatformIO IDE

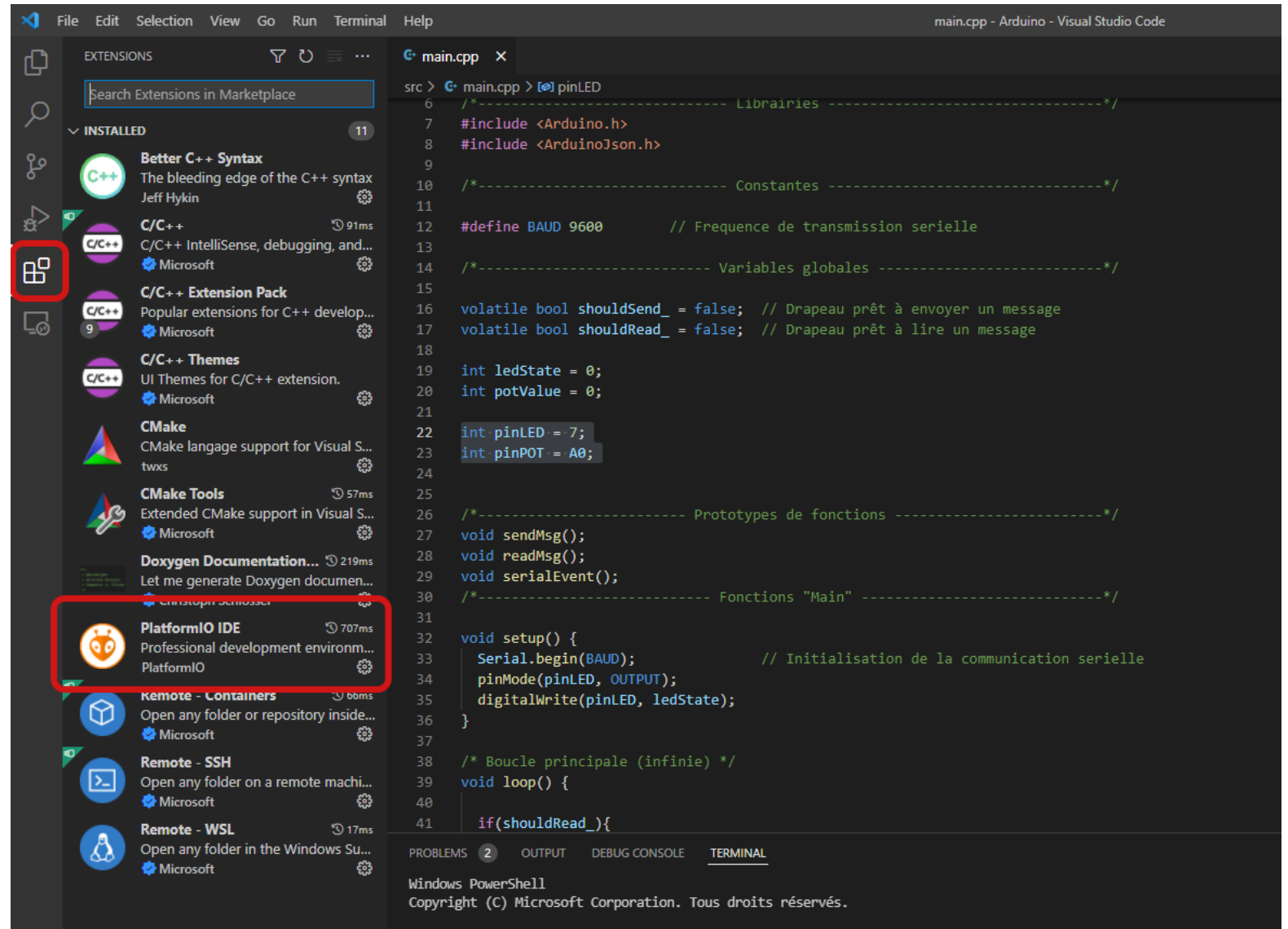
Dans le menu de gauche  
cliquer sur le bouton des  
extensions.

Ajouter PlatformIO IDE

ou

Télécharger

<https://platformio.org/>



# Téléchargement de MinGW

Télécharger MinGW

<https://sourceforge.net/projects/mingw/>




mingw : Les versions récents peuvent avoir des problèmes. Nous avons conçu le code avec la version 6.

Home / Browse / Development / Build Tools / MinGW - Minimalist GNU for Windows




## MinGW - Minimalist GNU for Windows

A native Windows port of the GNU Compiler Collection (GCC)  
Brought to you by: [cstrauss](#), [earnie](#), [gressett](#), [keithmarshall](#)

★★★★☆ 163 Reviews Downloads: 3,147,653 This Week Last Update: 2021-09-05

 **Download**  Get Updates  Share This

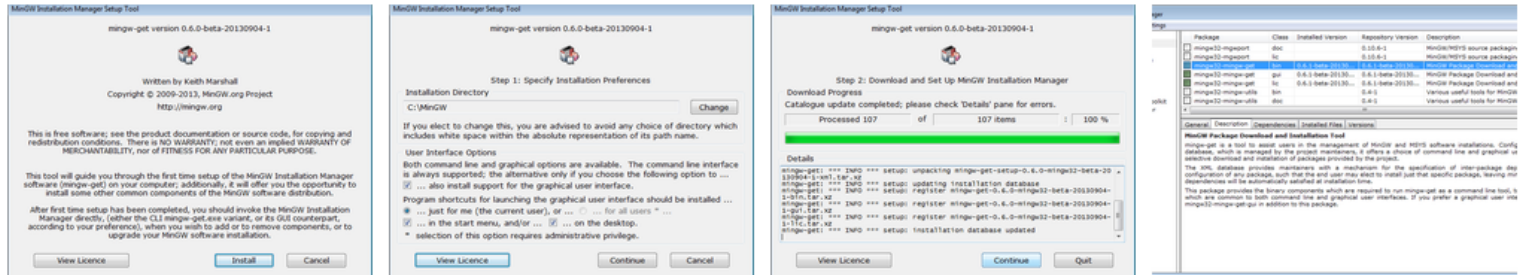
Windows

Summary Files Reviews Support News Wiki  Mailing Lists Tickets  Git 

This project is in the process of moving to [osdn.net/projects/mingw](https://osdn.net/projects/mingw), you can continue to follow us there.

MinGW: A native Windows port of the GNU Compiler Collection (GCC), with freely distributable import libraries and header files for building native Windows applications; includes extensions to the MSVC runtime to support C99 functionality. All of MinGW's software will execute on the 64bit Windows platforms.

### Project Samples



# Étapes d'installation

Cliquez Install.

Gardez les options par défaut et cliquez Continue.

Une fois les fichiers téléchargés, cliquez sur Continue. L'Installation Manager (IM) devrait démarrer. Ce qui suit se passe dans l'IM.

Assurez-vous qu'à gauche, Basic Setup est sélectionné.

Dans la partie du haut de la section de droite, assurez-vous que dans la colonne

Package au moins les packages suivants sont sélectionnés en faisant un clic-droit sur chacun et en sélectionnant Mark for installation :

- mingw-developer-toolkit
- mingw32-base
- mingw32-gcc-g++
- msys-base

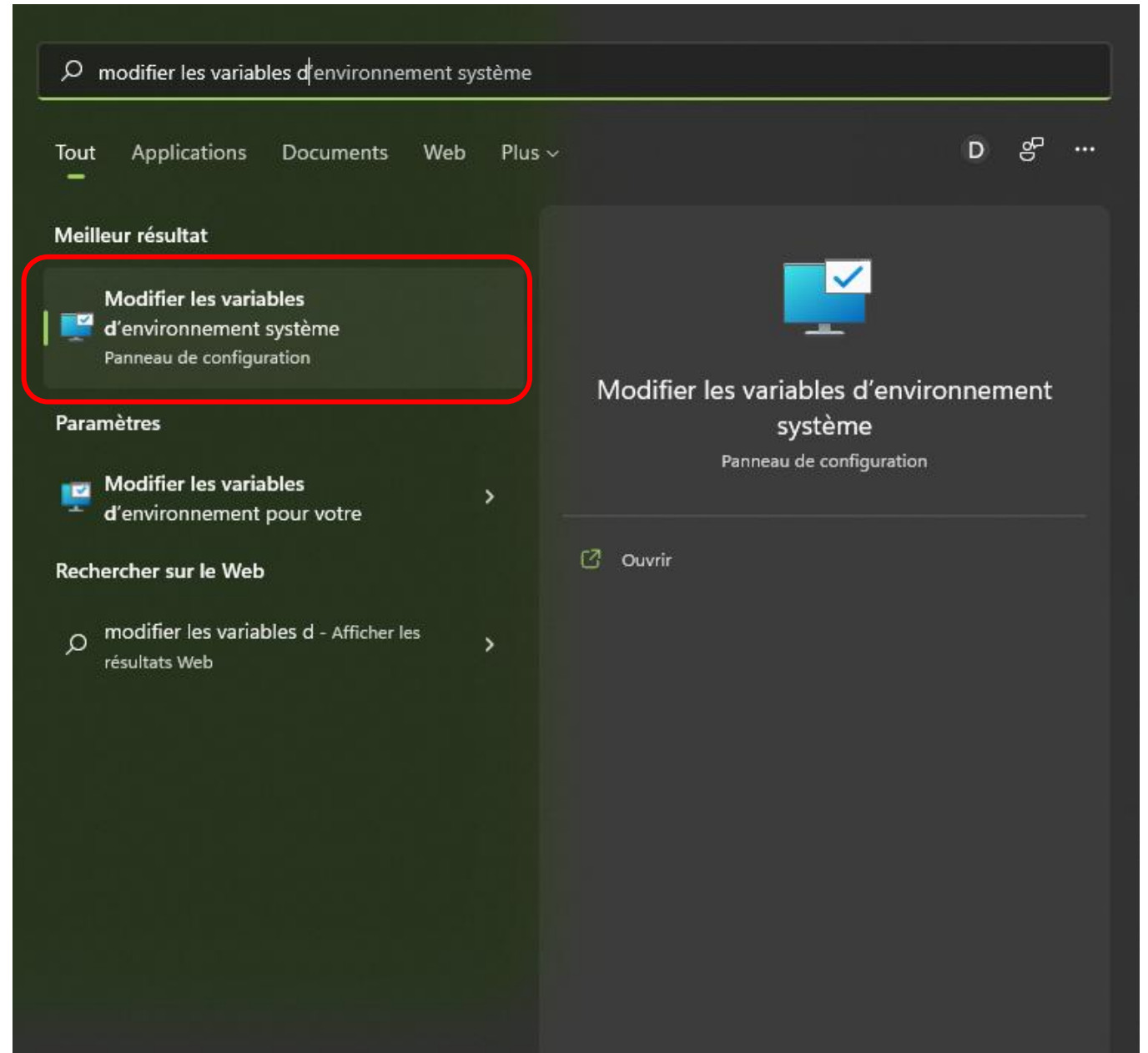
Sélectionnez Apply changes du menu Installation.

Dans la fenêtre qui apparaît, nommée Schedule of Pending Actions, cliquez sur Apply.

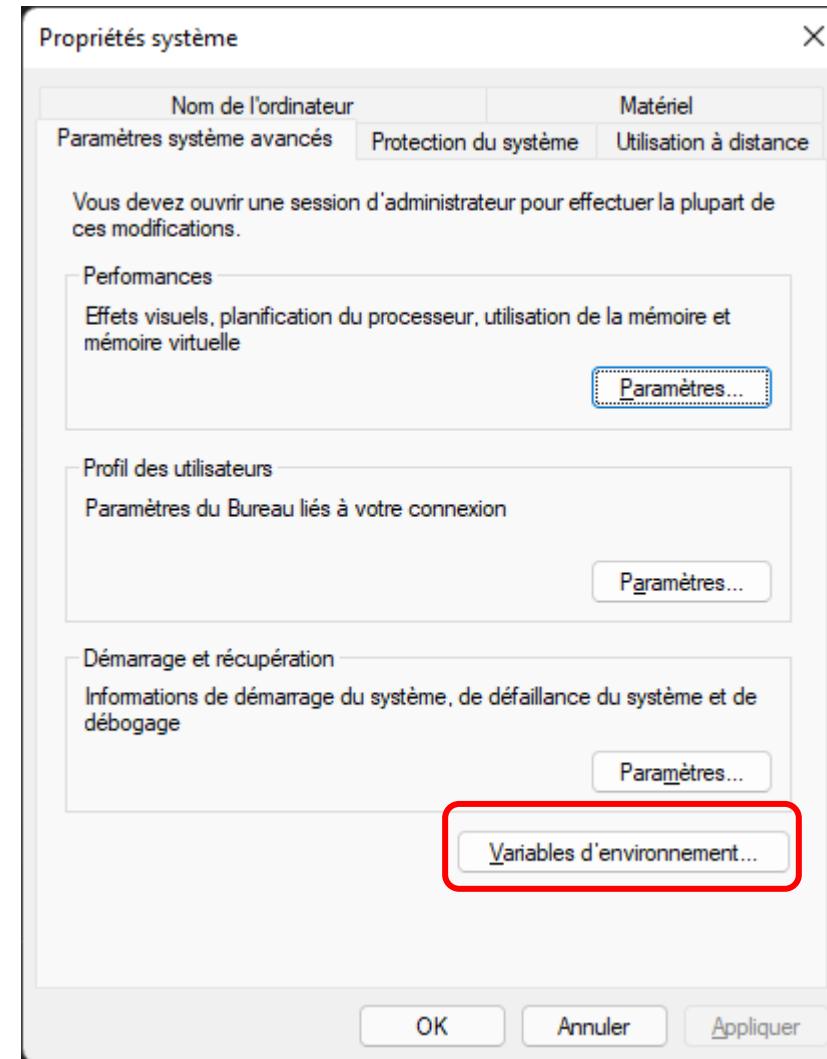
Soyez patient pendant le téléchargement et l'installation. Ensuite, cliquez sur Close et vous pouvez quitter l'IM.

# Configuration des variables d'environnement

Écrire « Modifier les variables d'environnement système » dans le menu démarrer ou l'outil de recherche de Windows.



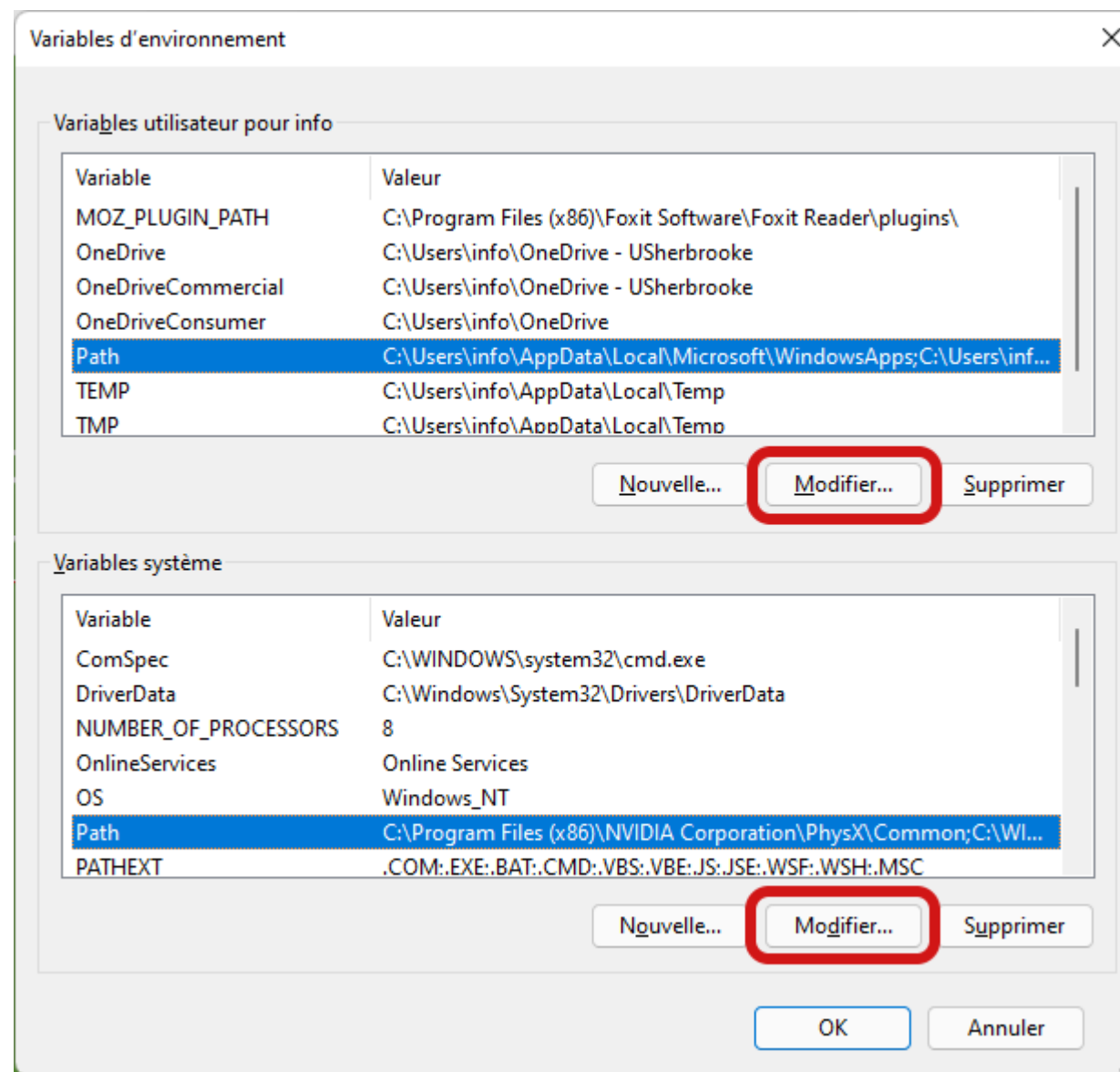
# Configuration des variables d'environnement





# Configuration des variables d'environnement

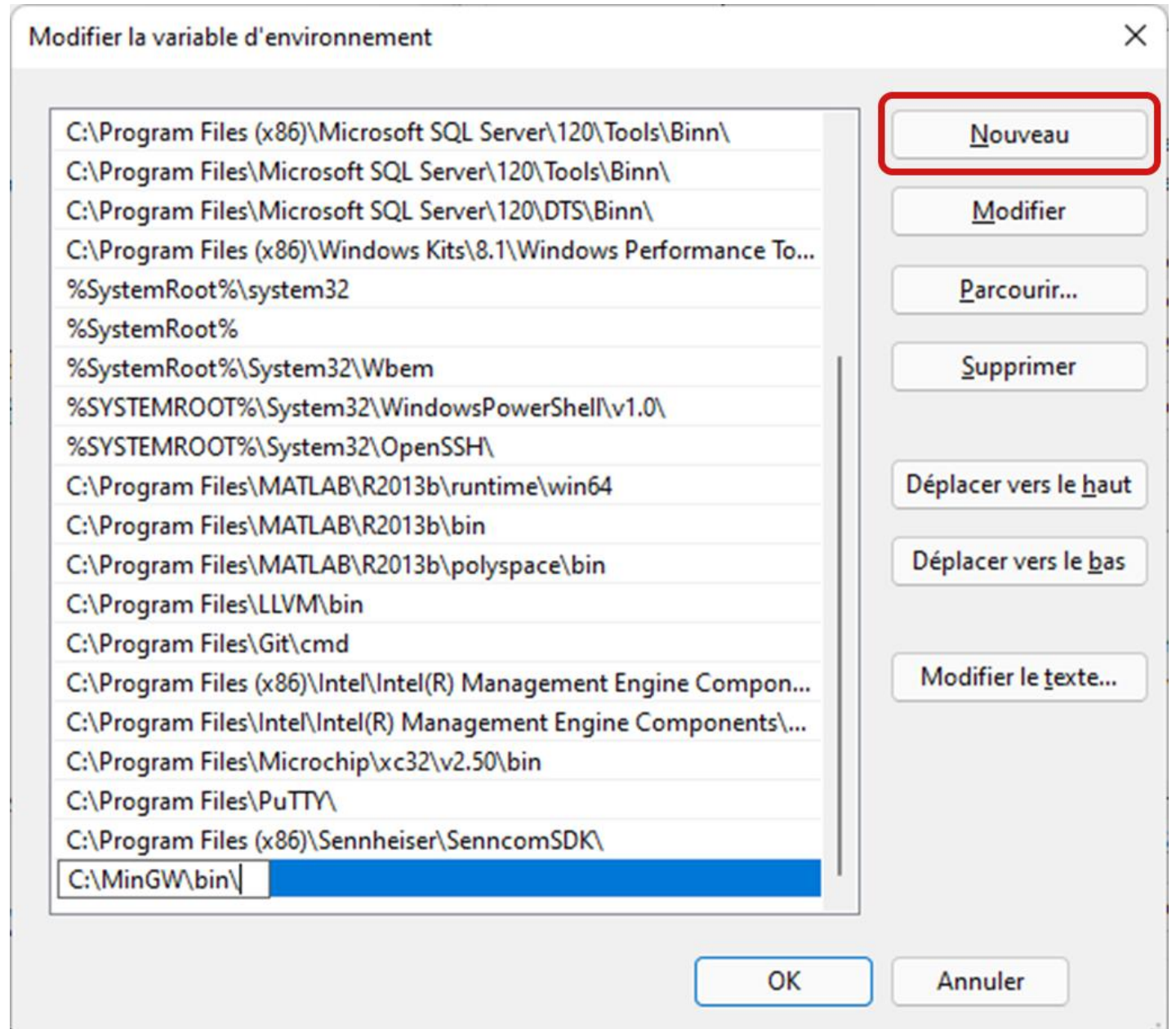
Modifier les 2 paramètres Path



# Configuration des variables d'environnement

Ajouter

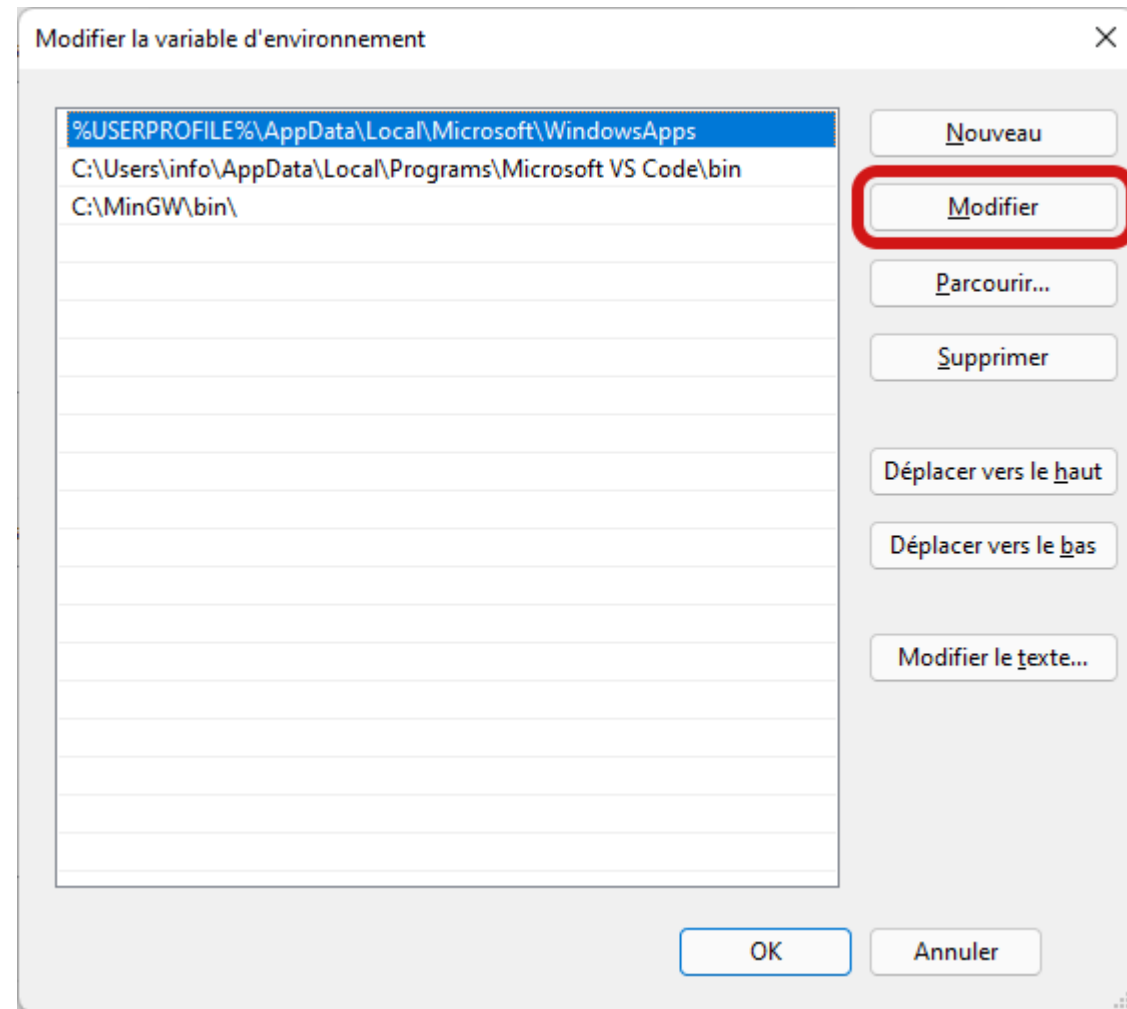
C:\MinGW\bin



# Configuration des variables d'environnement

Ajouter

C:\MinGW\bin



# Fonctionnement

---

Montage physique

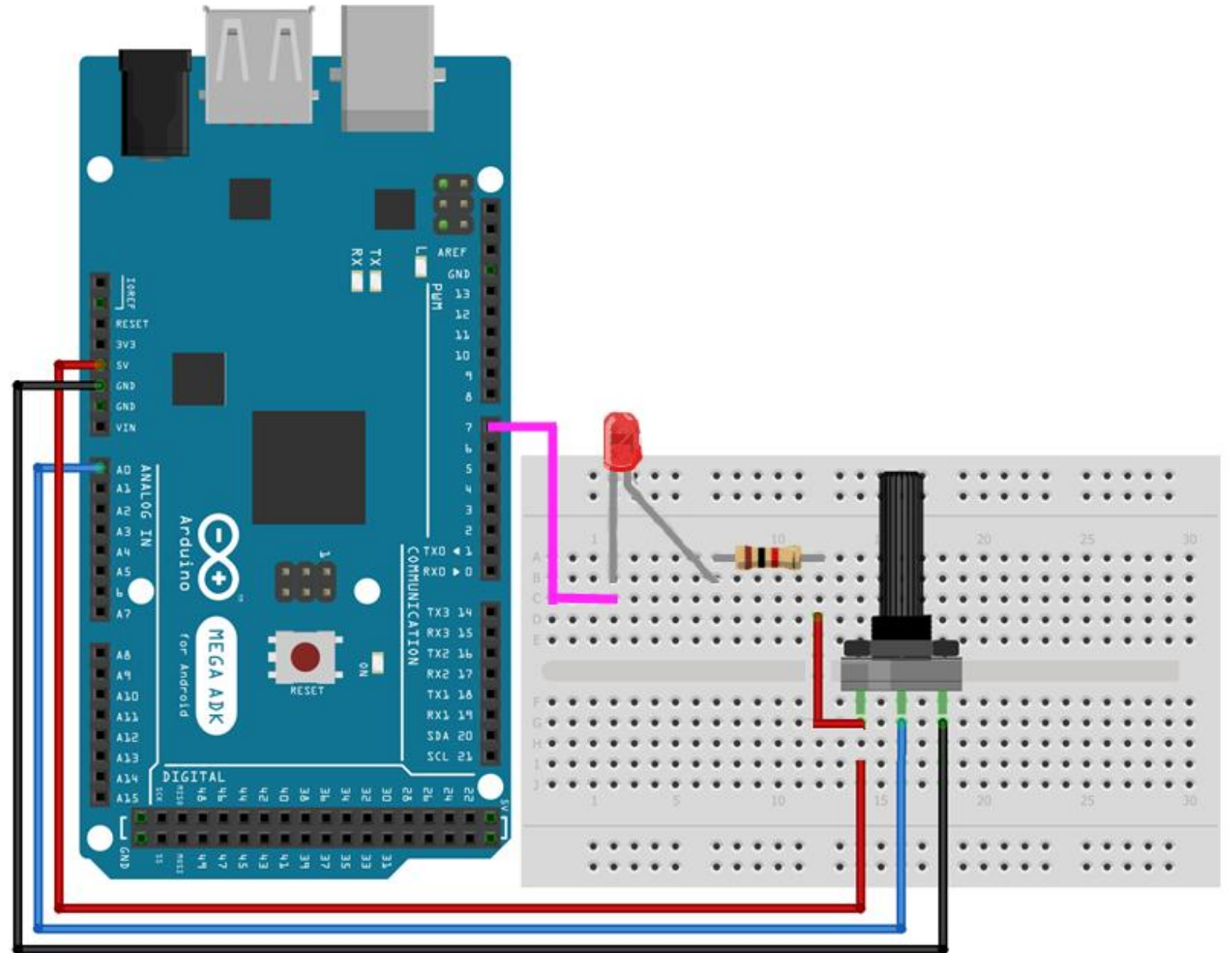
Explication du code

Modification du code

# Branchements sur le Arduino

pinLED = 7

pinPOT = A0



# Fonctionnement des codes

---

## Ordinateur (Maître)

1. Envoie des données (état DEL)
2. Réception des données (temps et pot)
3. Affichage des données (temps et pot)
4. Pause de 500 ms
5. Retour à l'étape 1

Note : Le choix de 1000 ms est pour permettre à la DEL de clignoter à 1 Hz. Vous devrez changer ce paramètre.

## Arduino (Esclave)

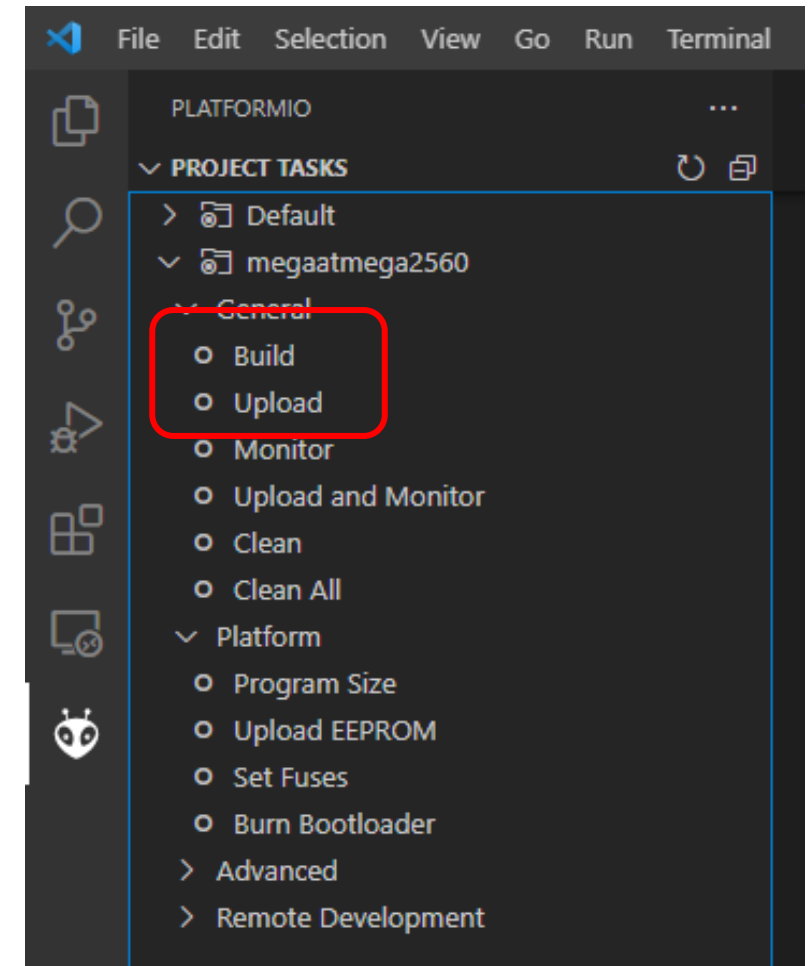
1. Réception des données (DEL)
2. Changement de l'état de la DEL
3. Lecture du potentiomètre
4. Envoie des données (temps et pot)
5. Pause de 10 ms
6. Retour à l'étape 1.

Note : Le choix de 10 ms est pour ne pas surcharger le processeur. Vous pouvez changer ce paramètre pour être compatible avec le pc.

# Code pour application Arduino

Vous devez utiliser le logiciel VScode avec l'extension PlatformIO.

1. Ouvrir le code de départ dans le dossier Arduino.
2. Compiler
3. Télécharger sur l'Arduino



# Code pour application PC

---

Vous devez utiliser le logiciel VScode avec le compilateur MinGW.

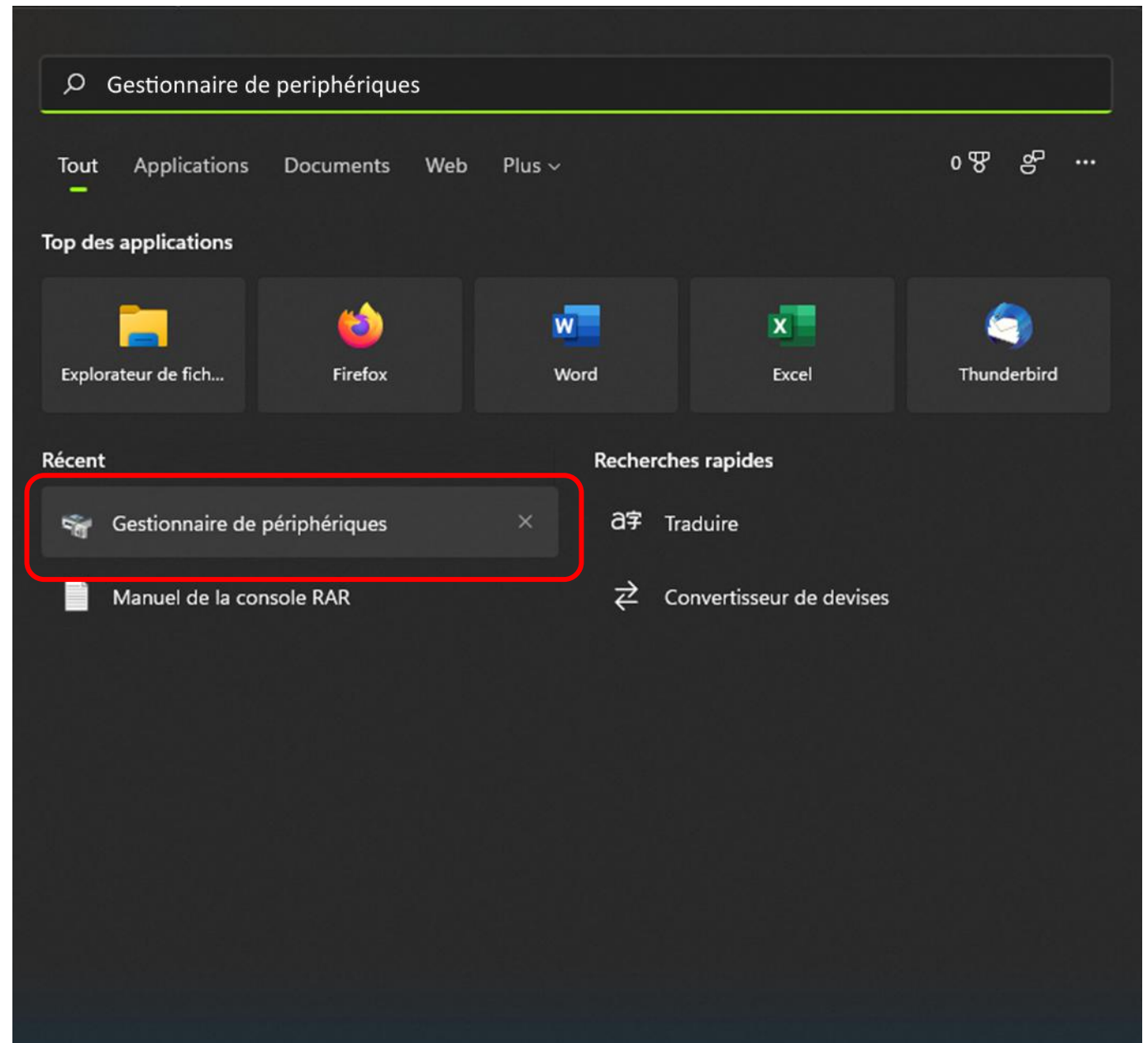
1. Ouvrir le code de départ dans le dossier PC.
2. Compiler
  - a) Ouvrir un nouveau terminal dans le menu Terminal
  - b) Écrire la commande :  
`g++ .\main.cpp .\include\serial\SerialPort.cpp`
3. Exécuter
  - a) Écrire la commande : `.\a.exe`
  - b) Écrire le numéro du port USB (voir diapo suivantes)

```
General\2-Labo Communication\Code_Arduino_PC\Code_Arduino_PC\PC> g++ .\main.cpp .\include\serial\SerialPort.cpp
General\2-Labo Communication\Code_Arduino_PC\Code_Arduino_PC\PC> .\a.exe
```

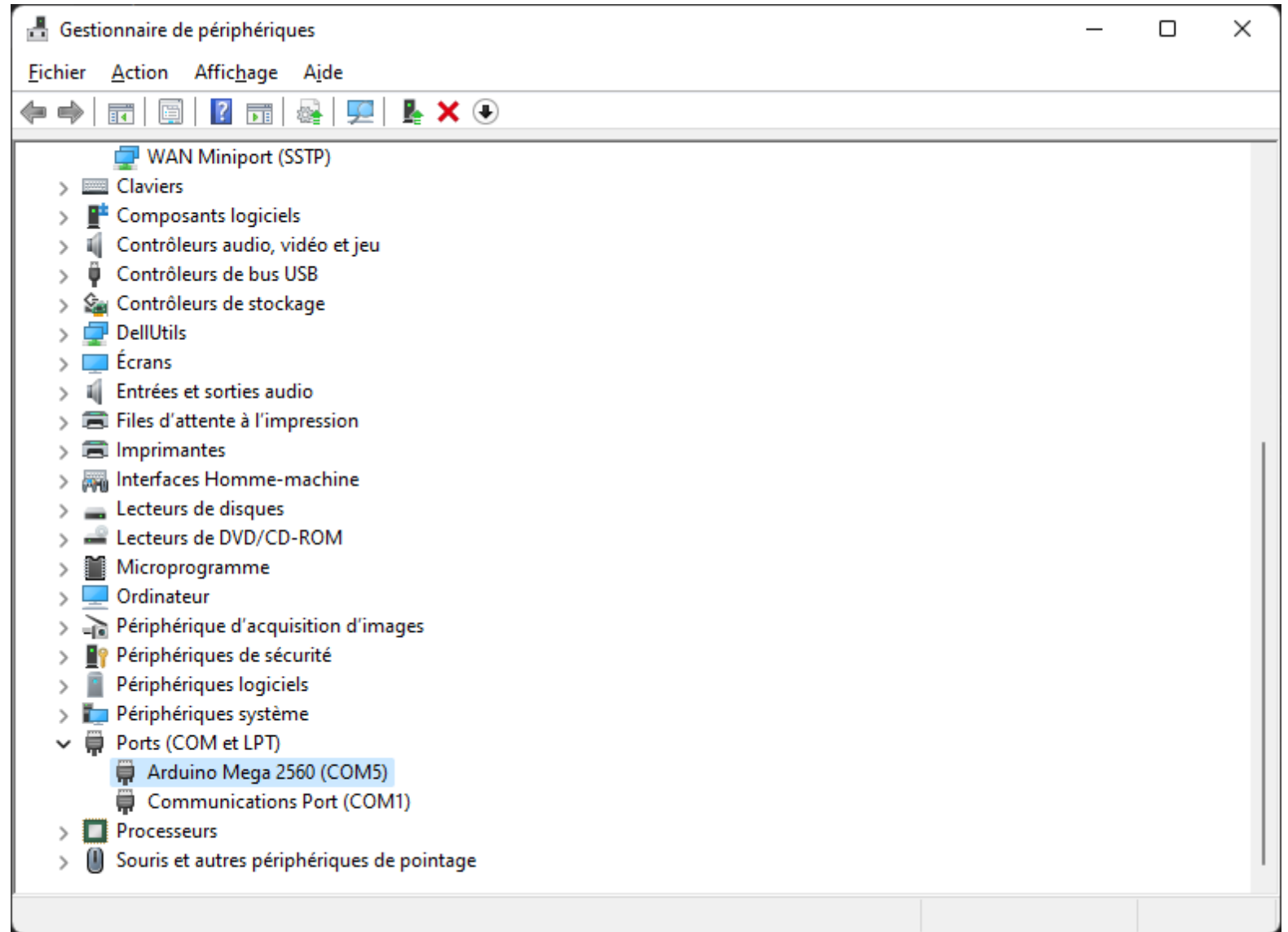


# Trouver le port USB du Arduino

Écrire « Gestionnaire de périphériques » dans le menu démarrer ou l'outil de recherche de Windows.



# Trouver le port USB du Arduino



# Modification des codes

---

Ajout d'entrées et de sorties, ainsi que les codes *json* associés

- Entrées : capteurs et boutons
- Sorties : DEL, affichage, moteurs...

Prétraitement des données sur le Arduino

- Moyenne des données
- Formatage des données analogiques
- Transformation en caractères ascii pour faciliter le débogage

Ajustement des temps de pause de 1000 ms sur le PC

- Selon vos besoins et de la capacité du processeur

# Communication avec code json avec VScode

Les *json* sont des chaines de caractères permettant de les retrouver dans les données transmises.

Les codes sont aussi pratiques pour afficher et reconnaître facilement les données.

```
// Structure de donnees JSON pour envoie et reception
int led_state = 1;
int pot_value = 0;
json j_msg_send, j_msg_rcv;

// Boucle infinie pour la communication bidirectionnelle Arduino-PC
while(1) {
    j_msg_send["led"] = led_state;    // Création du message à envoyer

    if(!SendToSerial(arduino, j_msg_send)) {    //Envoie au Arduino
        cerr << "Erreur lors de l'envoi du message. " << endl;
    }

    // Reception message Arduino
    j_msg_rcv.clear(); // effacer le message precedent
    if(!RcvFromSerial(arduino, raw_msg)) {
        cerr << "Erreur lors de la reception du message. " << endl;
    }

    // Impression du message de l'Arduino, si valide
    if(raw_msg.size()>0) {
        j_msg_rcv = json::parse(raw_msg);    // Transfert du message en json
        pot_value = j_msg_rcv["analog"];    // Transfert dans la variable pot_value
        cout << "Message de l'Arduino: " << j_msg_rcv << endl;
    }

    led_state = !led_state;    //Changement de l'etat led

    // Bloquer le fil pour environ 1 sec
    Sleep(1000); // 1000ms
}
```

# Communication avec code json avec Arduino

Vous pouvez transmettre plus d'items dans la communication en ajout des *json* supplémentaires dans chacun des codes (PC et Arduino).

```
/* Envoit du message Json sur le port serial */
void sendMsg() {
    StaticJsonDocument<500> doc;
    // Elements du message
    doc["time"] = millis();
    doc["analog"] = potValue;

    // Serialisation
    serializeJson(doc, Serial);

    // Envoit
    Serial.println();
    shouldSend_ = false;
}

void readMsg(){
    // Lecture du message Json
    StaticJsonDocument<500> doc;
    JsonVariant parse_msg;

    // Lecture sur le port Serial
    DeserializationError error = deserializeJson(doc, Serial);
    shouldRead_ = false;

    // Si erreur dans le message
    if (error) {
        Serial.print("deserialize() failed: ");
        Serial.println(error.c_str());
        return;
    }

    // Analyse des éléments du message message
    parse_msg = doc["led"];
    if (!parse_msg.isNull()) {
        // mettre la led a la valeur doc["led"]
        digitalWrite(pinLED, doc["led"].as<bool>());
    }
}
```

# Attention

## Bugs possibles

Vous devez vous assurer de ne pas avoir d'autre communication USB ouverte avec le Arduino.

- Ex : Serial monitor ouvert dans VScode.

Mettre les mêmes code Json dans l'ordi et le Arduino.

# Pin out Arduino Méga

<https://diyi0t.com/arduino-mega-tutorial/>

Ceci pourra vous aider à faire votre allocation de broches et faire vos branchements.

