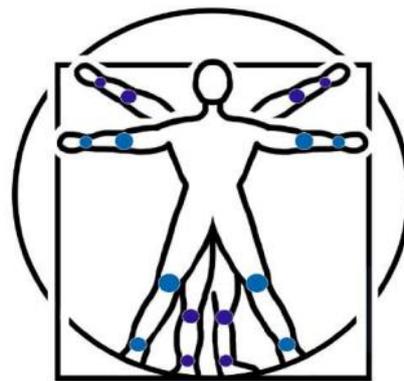




Projet Ingénieur n°15 :

« Conception et réalisation d'un équipement mesurant la position et la charge d'un individu en milieu industriel »



VRGOAPT

Notice d'utilisation

Client : Thierry BLANDET

Encadrante : Jelila LABED

Mentor : Pauline NOE

Equipe : Emilie FERREIRA, Farah GHERIR, Nathan LEBAS,
Nathan GRILLET-NIESS, Loïs GALLAUD

Sommaire

1) Présentation du projet	3
Description du projet et des acteurs	3
Objectifs du projet	3
2) Notice d'utilisation	3
A) Composition globale	3
<i>a- Modules inertiels</i>	4
<i>b- Modules de charge</i>	5
<i>c- Module de contrôle</i>	7
<i>d- Connectiques</i>	8
B) Utilisation	10
<i>a- Installation</i>	10
<i>b- Utilisation du dispositif</i>	10
<i>c- Première utilisation /téléversement du code (s'il y a une modification à faire)</i>	11
<i>d- Récupération des données</i>	16

1) PRÉSENTATION DU PROJET

a) Description du projet et des acteurs

Sujet : Conception et réalisation d'un équipement mesurant la position et la charge d'un individu en milieu industriel

Clients : BLANDET Thierry, ingénieur dans l'équipe IGG

Encadrante : LABED Jelila

Equipe : Emilie FERREIRA, Farah GHERIR, Nathan LEBAS, Nathan GRILLET-NIESS, Loïs GALLAUD

Mentor : Pauline NOE

b) Objectifs du projet

L'objectif principal de ce projet consiste à réaliser un périphérique permettant d'interagir avec une plateforme de réalité virtuelle dans un milieu industriel afin d'améliorer la qualité des gestes et l'ergonomie des postes de travail. Pour se faire, le client nous demande de concevoir un équipement facilement amovible permettant de visualiser les mouvements avec et sans charge et selon différentes contraintes : le poids du système, sa portabilité ou encore la précision du dispositif. Une fois réalisé, ce système sera testé en milieu industriel.

2) NOTICE D'UTILISATION:

A) Composition globale

Cet équipement sophistiqué est conçu pour mesurer avec précision le port de charge et la position de l'utilisateur, offrant ainsi une solution complète pour diverses applications nécessitant une surveillance et une analyse des mouvements. A noter que ce manuel d'utilisation permet également de refaire chacun des composants.

Son contenu est le suivant :

Notice d'utilisation

- 17 modules inertiels :



- 1 module de contrôle



- 2 modules de charge



- 14 câbles pour les connexions



- 11 attaches pour le montage



- 1 harnais pour le port.



a- Modules inertiels :

i) Composition

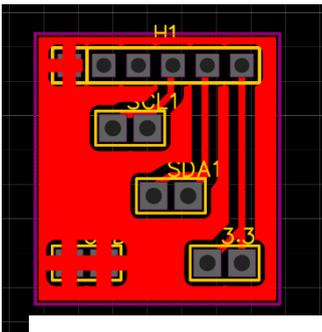
Le module inertiel est composé d'une carte électronique simple mais efficace, conçue spécifiquement pour connecter un capteur inertiel CMPS14 aux connectiques des boîtiers, expliqué plus en détail dans le rapport technique. Cette carte permet de relier deux connecteurs femelles au capteur, facilitant ainsi son intégration dans diverses configurations.



1: Capteur inertiel ouvert

Le CMPS14 est un capteur inertielle avancé qui intègre un accéléromètre 3 axes, un gyroscope 3 axes et un magnétomètre 3 axes. Cette centrale inertielle offre également une fusion des données, permettant une mesure précise et fiable des mouvements.

ii) Caractéristiques techniques

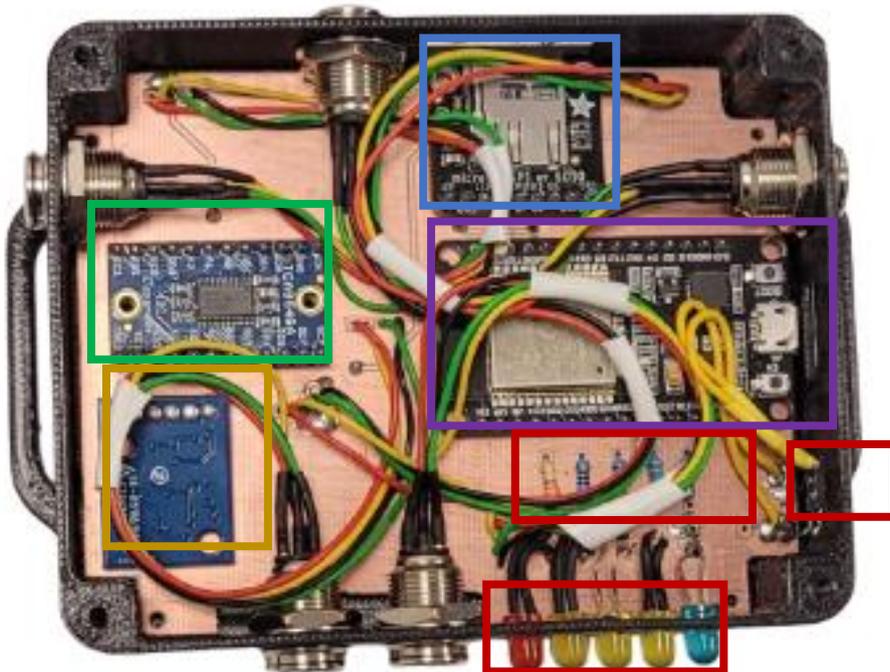


2: pcb module inertiel

Le module inertielle est équipé d'un PCB conçu sur EasyEDA pour garantir une efficacité optimale. Il est conçu pour relier deux connecteurs femelles à un capteur CMPS14. La ligne H1 de la carte comprend 6 trous de 1.1 mm, espacés de 2.54 mm, destinés au bus I2C (SDA, SCL) et à l'alimentation (3.3V, GND). Cette disposition assure une compatibilité avec les standards industriels pour une intégration simplifiée du capteur.

b- Module de contrôle :

i) Composition



3: Module de contrôle ouvert

Notice d'utilisation

- *ESP32 (violet) : Microcontrôleur avec 30 broches d'entrée/sortie. Seules 14 broches sont utilisées dans notre conception.*
- *LEDs et Bouton (rouge) : Contrôlés par des résistances de 33 Ohm, connectées aux broches du côté droit de l'ESP32.*
- *Multiplexeur TCA9548A (vert) : Permet le tri et la transmission des données des connectiques et du capteur. Utilise l'I2C pour diriger sélectivement les données vers l'ESP32.*
- *Module SD (bleu) : Enregistre les données sur une carte micro SD via le protocole SPI ou SDIO.*
- *Capteur inertielle (jaune).*

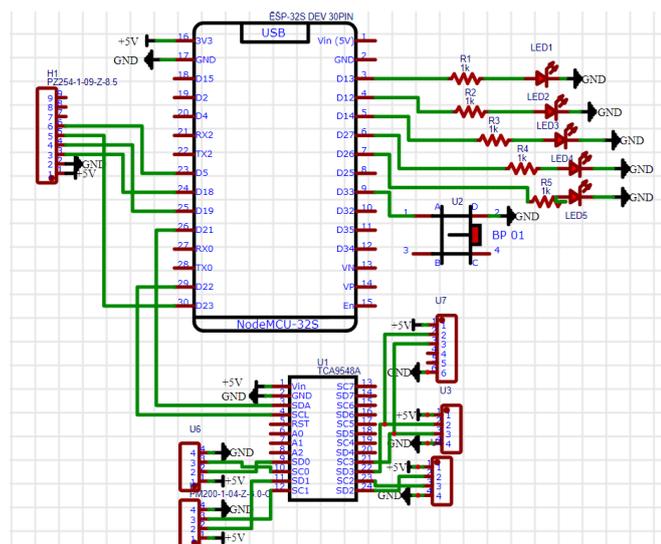
Cette configuration permet une utilisation efficace et coordonnée des composants pour la collecte, le traitement et le stockage des données.

ii) Caractéristiques techniques :

Le schéma du PCB, conçu sur EasyEDA(et de plus nette qualité dans le rapport technique), illustre les connexions des différents membres du corps humain aux capteurs et au système électronique. :

Chaque composant est soigneusement disposé pour garantir une surveillance précise de la position et une optimisation de la place en prenant en compte la taille des différents composants ainsi que des prises femelles qui se trouvent à l'intérieur de la boîte.

- *U1 : multiplexeur*
- *H1 lecteur de carte sd*
- *Bp 01 : bouton de pression*
- *U6 est associé au bras gauche.*
- *U5 est lié à la jambe gauche.*
- *U4 est relié à la jambe droite.*
- *U7 est lié au tronc*
- *U9 est lié au bras droit*



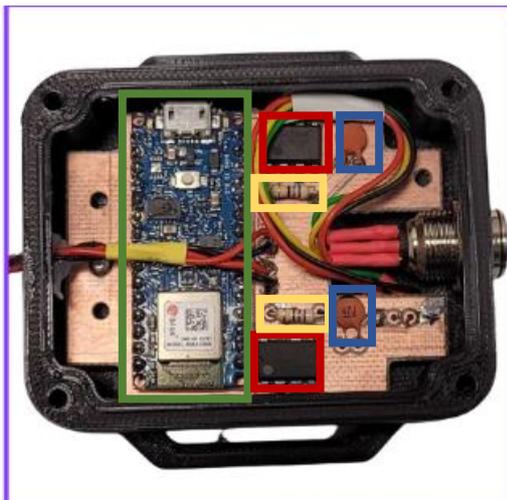
4: Pcb module de contrôle

c- Module de charge :

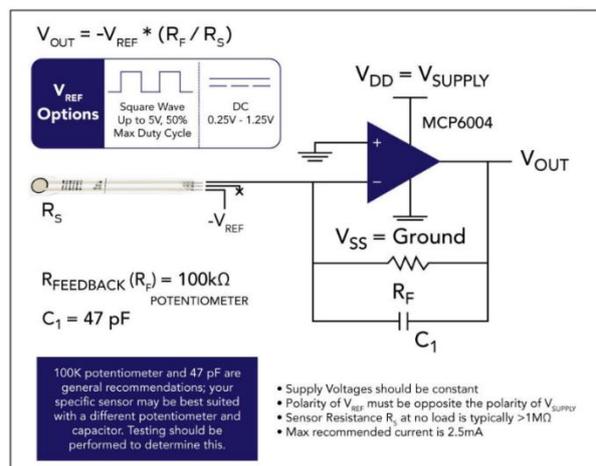
i) Composition :

Le module de charge est constitué des éléments suivants :

- Deux Résistances (jaune) : Utilisées pour contrôler et réguler le courant électrique lors du processus de charge.
- Deux Capacités (bleu) : Stockent et fournissent l'énergie électrique nécessaire pour alimenter le système de charge.
- Arduino Nano (vert) : Intégré avec un capteur de position pour surveiller et contrôler le processus de charge.
- Deux Amplificateurs Opérationnels (AOP) (rouge) : Assurent une amplification et un traitement précis du signal électrique dans le système de charge.



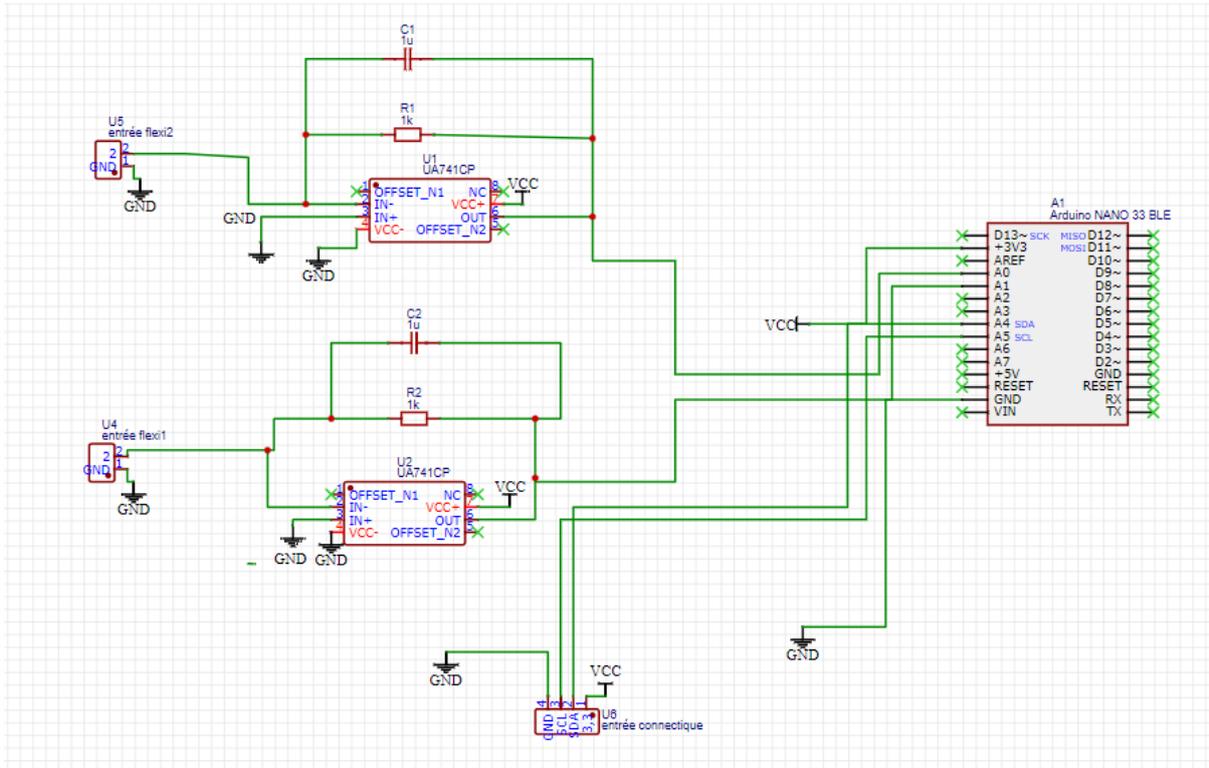
5: Module de charge ouvert



6 : Schéma de montage du module de charge

ii) Caractéristiques techniques

Le PCB constitue la structure centrale du module de charge. Sur ce circuit imprimé, les composants sont rigoureusement soudés conformément aux spécifications techniques du produit. Pour une référence visuelle plus détaillée de la disposition des composants, veuillez consulter le rapport technique fourni avec le module. Lors de l'installation du module, veuillez souder les connectiques selon les emplacements dédiés sur le PCB. Ces emplacements sont clairement identifiés pour assurer une connexion précise et sécurisée.



5: PCB du module de charge

d- Connectiques :

Pour établir les connexions, deux types de connectiques sont utilisés.



6: Sertissage prise femelle



7: Sertissage prise mâle

- Pour les connecteurs femelles intégrés à l'intérieur des modules, il suffit de souder des fils selon le code couleur indiqué sur la figure 6 : le jaune correspond au SDA, le vert au SCL, le rouge au 3.3V et le bleu à la masse.



8: Prise mâle soudure

- Pour les connecteurs mâles, il est nécessaire de souder les câbles avec les connectiques mâles. Dans les deux sens du câble, il est important de respecter la correspondance des couleurs : chaque couleur du câble mâle doit être soudée à la même couleur (voir figure 8) pour assurer une connexion correcte.

B) Utilisation :

a- Installation

Pour installer l'équipement, suivez ces étapes :

- **1. Fixation des Modules :**

Installez tous les modules en utilisant les attaches désignées (voir le tableau ci-contre pour les correspondances).

- **2. Connexion des Modules :**

Connectez tous les modules entre eux en utilisant les câbles. Le nom de la position est écrit sur chaque câble pour faciliter l'installation.

- **3. Raccordement du Module Central :**

Raccordez le module central avec tous les autres capteurs en respectant les connexions appropriées. En suivant ces instructions, vous pourrez installer rapidement et facilement l'équipement, prêt à être utilisé pour la surveillance et la mesure des mouvements.

	-	Avant-bras
	-	Biceps
	-	Main
	-	Cuisse
	-	Mollet
	-	Harnais taille
	-	Harnais épaule-module de contrôle
	-	Harnais épaule-avant du harnais
	-	Harnais dos-épaules



9: Légende de couleurs des attaches

b- Utilisation du Dispositif :

Pour utiliser l'appareil, branchez-le à une source d'alimentation. Les LED effectueront alors des allers-retours, indiquant l'initialisation du système. Une fois cette étape terminée, les LED clignoteront sans changer, signalant que le dispositif est prêt à être lancé. Le nombre de capteurs connectés à la boîte sera indiqué en binaire par les LED (de 17 à 1, au maximum).

Pour lancer l'enregistrement, maintenez le bouton enfoncé pendant 3 secondes. Une animation s'affichera et la LED bleue s'allumera pour indiquer que l'enregistrement est en cours. Si la carte SD n'est pas branchée, la LED d'erreur s'allumera en plus.

Pour arrêter l'enregistrement, maintenez à nouveau le bouton enfoncé pendant 3 secondes. Le dispositif reviendra à l'état de clignotement, affichant le nombre de capteurs branchés, prêt pour une nouvelle utilisation.

Si le module central est relié à un moniteur série, on peut lire les données directement.

c- Première utilisation- téléversement du code (s'il y a une modification à faire) :

1) Téléchargement des codes utilisés

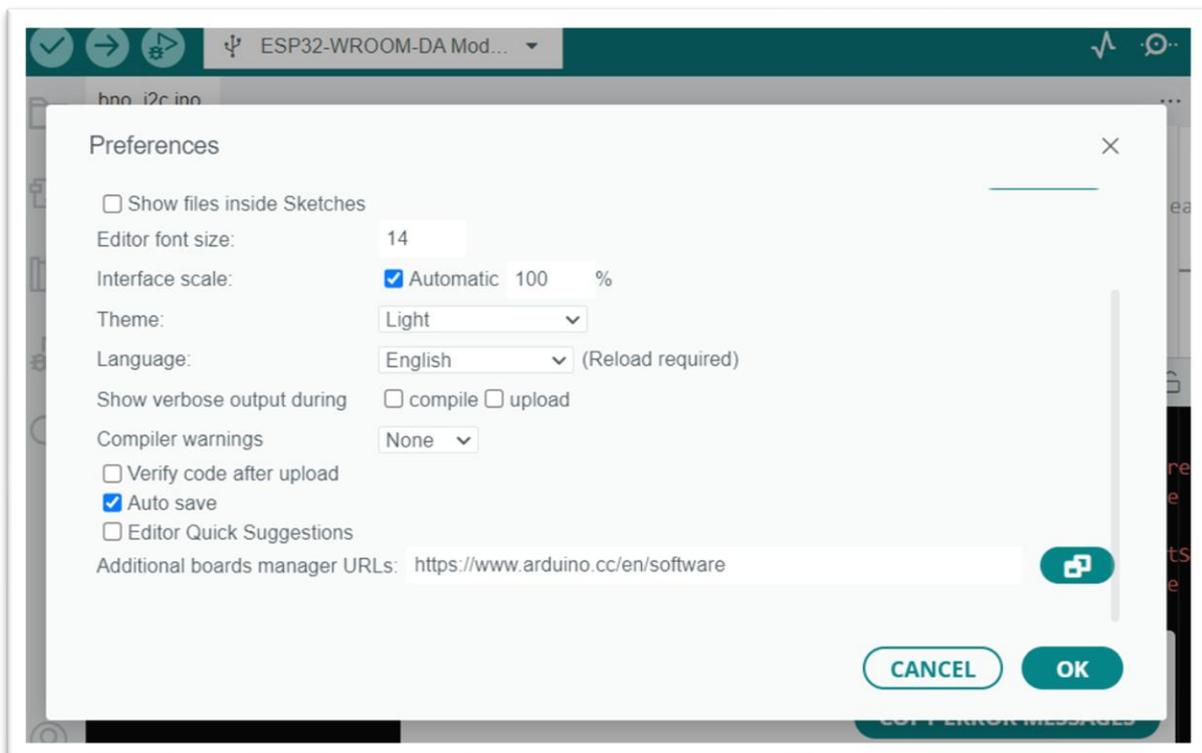
Afin de pouvoir téléverser le code Arduino sur l'ESP32, il faut en premier lieu télécharger le logiciel Arduino suivant le système d'exploitation.

Télécharger sur <https://www.arduino.cc/en/software>.

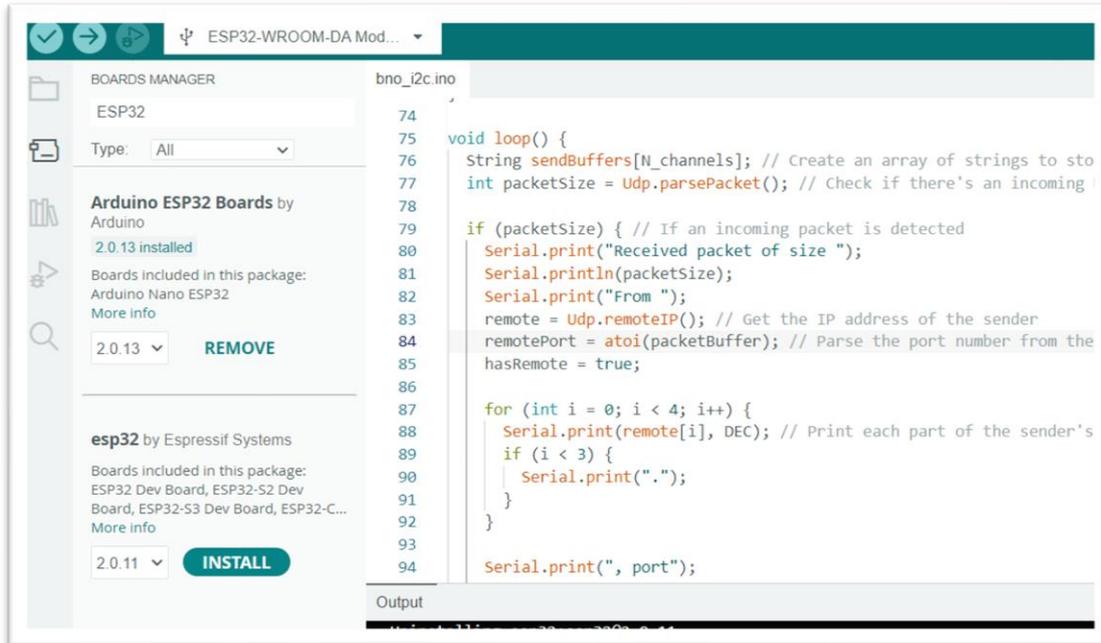
- Copier le code Arduino s'il existe après avoir lancé le logiciel.
- Brancher la carte via un port USB sur l'ordinateur.

La seconde étape consiste à rechercher la BOARD de telle sorte à ce que l'ESP32 soit visible sur le logiciel.

- Cliquer sur « File » puis « Preferences »
- Ajouter l'URL : https://dl.espressif.com/dl/package_esp32_index.json dans l'onglet « Settings » dans la partie « Additional boards manager URLs ».



- Cliquer sur « tools » -> « Boards » -> « Boards manager »
- Chercher et installer l' « esp32 by espressif systems » comme suit :



- Sélectionner la carte et téléverser le code. Si la carte n'est pas visible lire la partie II.

II) Problème carte non visible

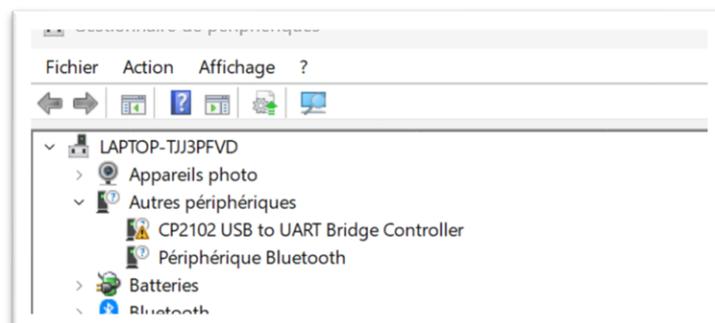
Si la carte n'est pas visible, une autre méthode peut être utilisée sur windows ou sur Ubuntu :

Sur **Ubuntu** (en modifiant le nom ici Nathan) :

- Ecrire sur le terminal les commandes suivantes : « sudo adduser nathan dialout » et « sudo chmod a+rw /dev/ttyUSB », La carte est désormais visible

Sur **Windows** :

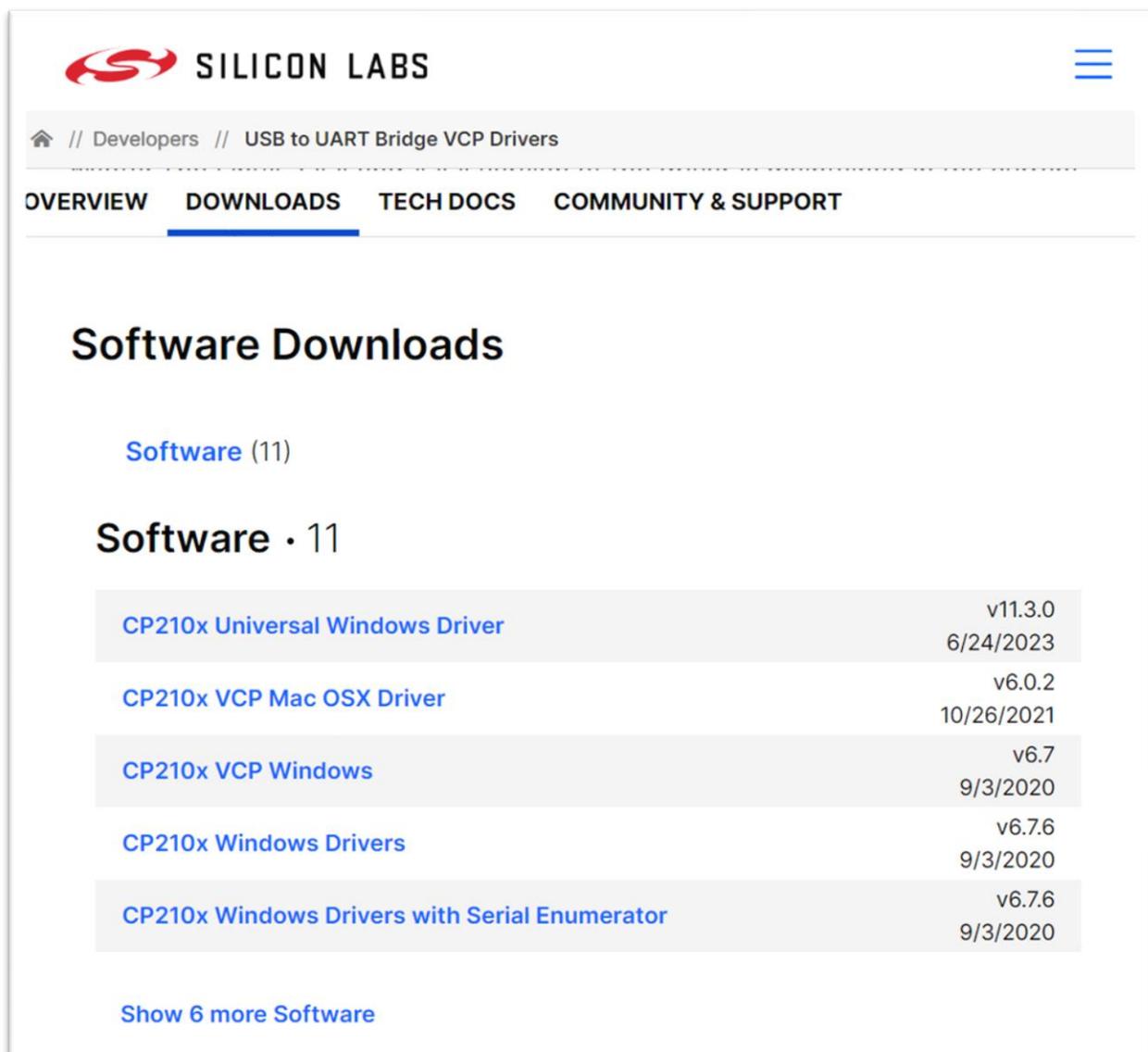
- Rechercher sur le gestionnaire de périphériques de l'ordinateur le port USB qui n'est pas connecté (CP2102 USB to UART Bridge Controller)



- En cliquant sur ce périphérique, une boîte de dialogue s'ouvre : cliquer sur « Mettre à jour les pilotes »
- Cliquer sur « Rechercher automatiquement les pilotes ».

Si aucun pilote n'est trouvé, il faut les télécharger sur internet.

- Rechercher sur un moteur de recherche : <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>
- Télécharger le pilote CP210x Universal Windows Driver depuis le site silicon labs et dezipper. :



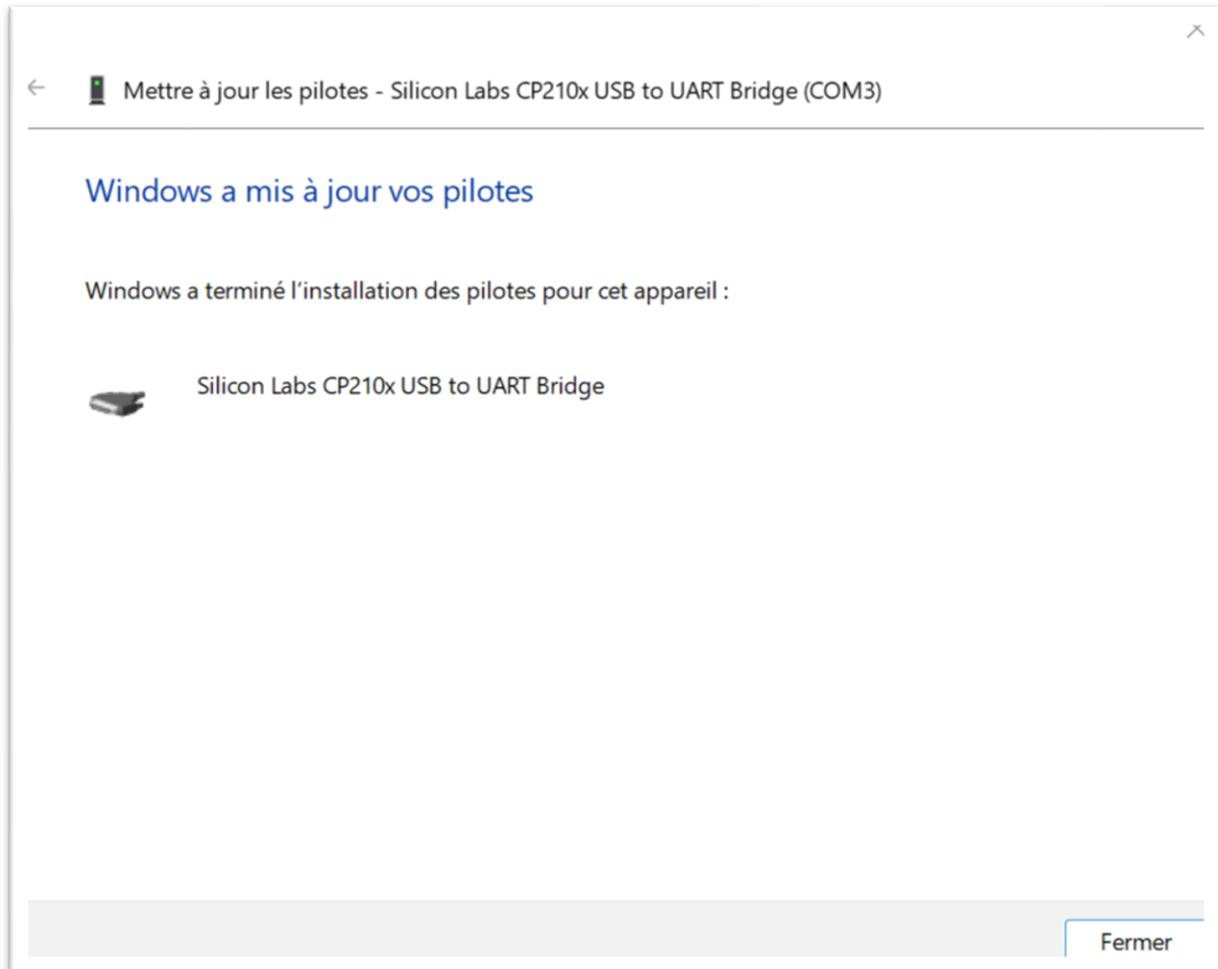
The screenshot shows the Silicon Labs website's 'Software Downloads' page. The breadcrumb trail is 'Developers // USB to UART Bridge VCP Drivers'. The navigation menu includes 'OVERVIEW', 'DOWNLOADS', 'TECH DOCS', and 'COMMUNITY & SUPPORT'. The main heading is 'Software Downloads' with a sub-heading 'Software (11)'. Below this, there is a list of software items:

Software Name	Version	Date
CP210x Universal Windows Driver	v11.3.0	6/24/2023
CP210x VCP Mac OSX Driver	v6.0.2	10/26/2021
CP210x VCP Windows	v6.7	9/3/2020
CP210x Windows Drivers	v6.7.6	9/3/2020
CP210x Windows Drivers with Serial Enumerator	v6.7.6	9/3/2020

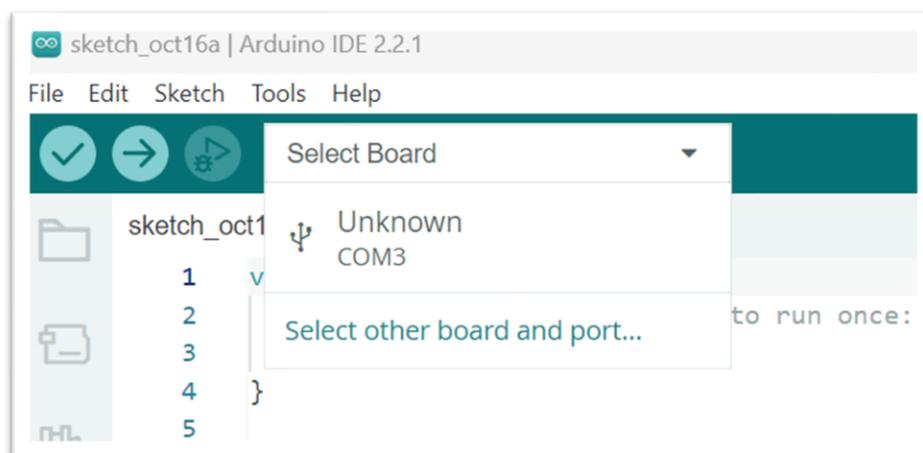
At the bottom of the list, there is a link: [Show 6 more Software](#)

- Revenir sur le gestionnaire de périphériques et sélectionner « Parcourir mon poste de travail pour rechercher des pilotes » puis le dossier téléchargé dezipper.

- Ce message apparaît et le pilote est mis à jour.



- Sur Arduino, la carte est désormais visible on peut donc la sélectionner comme suit :



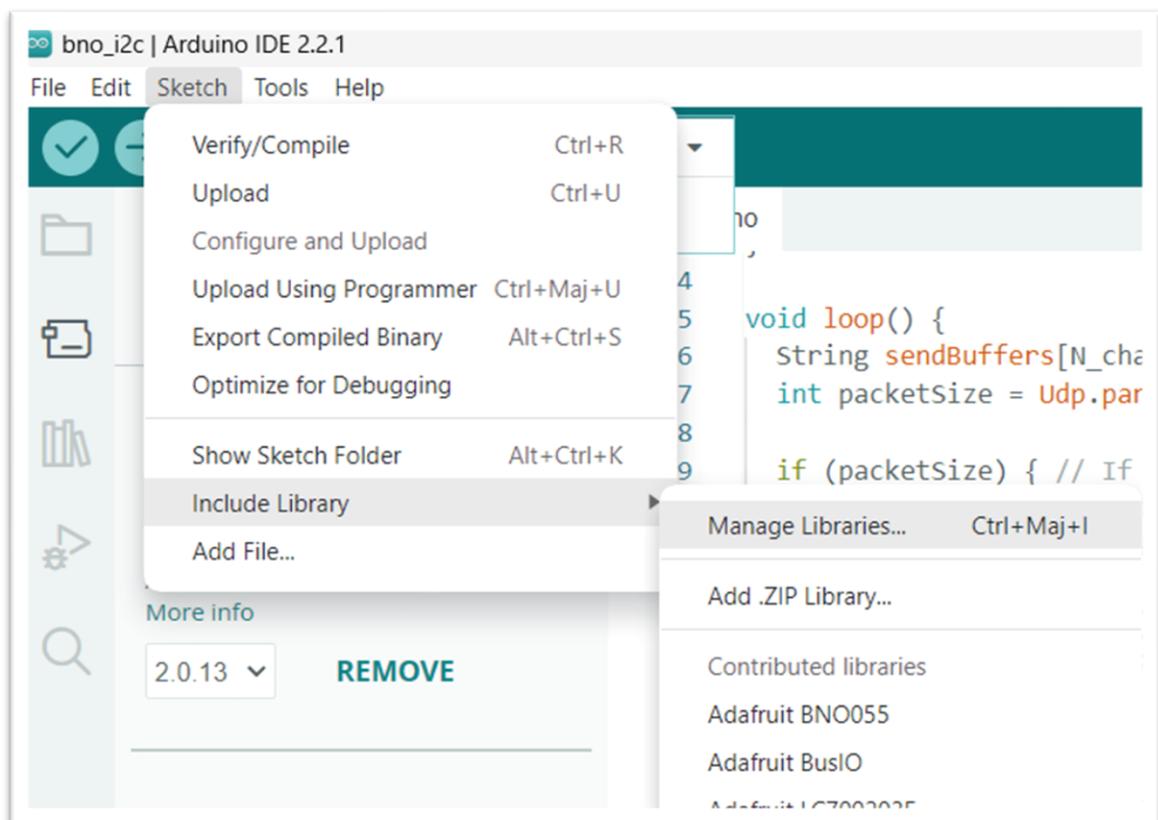
III) Sauvegarde des bibliothèques

Lors de la première utilisation de la carte, il est nécessaire de télécharger des bibliothèques Arduino qui sont utilisées dans le programme.

Pour cela, il faut télécharger des différents dossiers contenant les bibliothèques utiles.

Pour chaque bibliothèque, l'import se fait de manière suivante :

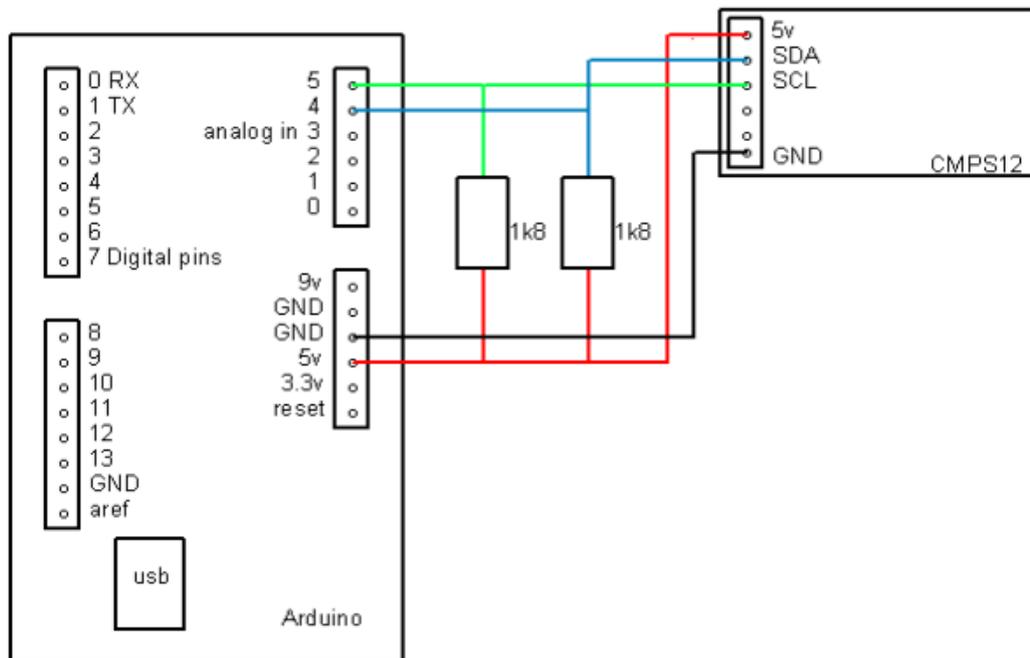
- Sur la barre d'outil, cliquer sur « Sketch », sur « Include Library » puis « Manage Libraries »



- Sélectionner le dossier choisi puis installer la bibliothèque
- Une fois toutes les bibliothèques installées, le code peut être téléversé.

d- Récupération des données :

- Disclaimer: Cette notice permet le débogage et la programmation d'un seul capteur. Il ne fonctionne pas directement avec plusieurs capteurs simultanément sur un bus I2C.
Branchement type:



Le montage ci-dessus permet de récupérer les données depuis une carte arduino.

Les résistances sont actuellement facultatives pour la plupart des cartes Arduino ou Esp32 possédant des résistances internes, dites de "pull up".

- Référez-vous à la documentation (pinout) de vos microcontrôleurs pour savoir quels sont les pins compatibles I2C de ces derniers.
- Utiliser le code exemple "arduino_cmeps14_12_i2c" fourni par le constructeur:

https://www.robot-electronics.co.uk/files/arduino_cmeps14_12_i2c.ino

Pour une carte Arduino il n'y a pas de modifications à apporter au code.

Pour les cartes ESP32 il faut penser à utiliser la class TwoWire de Wire.h comme dans l'exemple ci-dessous:

```
#include <Wire.h>

#define CMPS12_ADDRESS 0xC6 // Address of CMPS12 shifted right one bit for arduino wire library
#define ANGLE_8 1 // Register to read 8bit angle from

#define I2C_SDA 22
#define I2C_SCL 21

unsigned char high_byte, low_byte, angle8;
char pitch, roll;
unsigned int angle16;

TwoWire I2CBME = TwoWire(0);

void setup()
{
  Serial.begin(115200); // Start serial port
  I2CBME.begin(I2C_SDA,I2C_SCL);
}

void loop()
{
  I2CBME.beginTransmission(CMPS12_ADDRESS>>1); //starts communication with CMPS12
  I2CBME.write(0x01); //Sends the register we wish to start reading from
  I2CBME.endTransmission();
```

- Les pins 22 et 21 sont les pins par défaut de beaucoup de cartes ESP32 mais il faut penser à changer ces valeurs pour les cartes où ce n'est pas le cas.

CMPS12_ADDRESS dépend du nombre de fois que clignote la led de votre CMPS14 ou 12. Pour savoir quelle adresse utiliser référez vous à la documentation du fournisseur:

<https://www.robot-electronics.co.uk/files/cmeps14.pdf>

I2CBME est une variable, libre à vous d'en changer le nom. Il faut remplacer toutes les utilisations de Wire par cette variable.

Ex: Wire.beginTransmission → I2CBME.beginTransmission

- Si votre capteur fonctionne correctement vous devriez voir apparaître dans le moniteur série de votre IDE:

```
roll: 76    pitch: 12    angle full: 345.9    angle 8: 246
roll: 77    pitch: 12    angle full: 348.2    angle 8: 247
roll: 72    pitch: 12    angle full: 342.1    angle 8: 243
roll: 70    pitch: 11    angle full: 340.0    angle 8: 241
roll: 73    pitch: 12    angle full: 345.0    angle 8: 245
```

- Si tel n'est pas le cas, lancez le programme "WireScanMPU" en effectuant les mêmes modifications en fonction de votre carte pour vérifier que la connexion entre votre capteur et votre carte est bien effectuée et par la même occasion pour obtenir l'adresse shifté de 1 bit vers la droite de votre capteur (division par 2).

- Si la connection est correctement établie et que votre branchement correspond vous devriez obtenir ceci:

```
-----  
Scanning for I2C devices ...  
I2C device found at address 0x60  
Scanning for I2C devices ...  
I2C device found at address 0x60
```

Ici 0x60 est la division par 2 de 0xC0, soit l'adresse en sortie d'usine de votre capteur.

Changement d'adresse:

Une fois assuré du bon fonctionnement de votre capteur, vous pouvez utiliser le code "ChangeAddr2" fourni.

```
1  #include <Wire.h>  
2  
3  #define OLD_ADDRESS 0xC0  
4  #define NEW_ADDRESS 0xC6  
5  
6  #define I2C_SDA 22  
7  #define I2C_SCL 21  
8  
9  TwoWire I2CBME = TwoWire(0);  
10  
11 void setup() {  
12     I2CBME.begin(I2C_SDA,I2C_SCL);  
13     Serial.begin(115200);  
14 }  
15  
16 void loop() {  
17  
18     Serial.print("Start...");  
19  
20     I2CBME.beginTransmission(OLD_ADDRESS>>1);  
21     I2CBME.write(byte(0x00)); // Code send In order to make the sensor listening to a Command  
22     I2CBME.write(byte(0xA0)); // Fist Command (See the documentation about CMPS14)  
23     I2CBME.endTransmission();  
24     delay(20);
```

- Ce code permet de changer l'adresse de votre capteur parmi les 8 disponible sur la série des CMPS.

OLD_ADDRESS est l'adresse actuelle de votre capteur.

NEW_ADDRESS est la future adresse de votre capteur.

SDA et SCL sont les pins de votre microcontrôleur.

beginTransmission: permet de démarrer une communication avec le capteur. On shift l'adresse du capteur de 1 vers la droite >>1 pour qu'elle ne fasse plus que 7 bits. En effet les capteurs fonctionnent avec des trames de 8 bits mais l'adresse reconnue par la librairie Wire.h n'en fait que 7. Il faut donc penser à faire ce shift pour communiquer avec ce dernier.

- Il faut ensuite envoyer une séquence de byte au microcontrôleur qui doivent être compris comme des commandes. Pour se faire il faut envoyer un premier byte 0x00 permettant

d'écrire au registre 0 puis envoyer le byte de la séquence avant de terminer la transmission pour l'envoi de la commande suivante.

Les 20 ms de pause entre chaque commande sont importantes.

Se référer à la documentation du CMPS14.

Normalement, après exécution du code et après avoir débranché/rebranché votre capteur le nombre de fois que la led rouge clignote devrait avoir changé.