

PREMIERS PAS SUR ESP 32

PAR THÉO-FÉLIX ADAM

DIRE QU'ON CODE EN **UTF 8** SUR HTML.

SOMMAIRE

Appareil.....	4
1) Caractéristiques.....	4
2) ESP 32.....	5
a) Caractéristiques techniques.....	5
b) Liste d'options :.....	5
c) Broches.....	6
d) Liste d'accessoires.....	7
3) GPS.....	9
a) Principe de fonctionnement.....	9
Équation de navigation.....	9
transformation de Lorentz.....	9
sae.....	10
1) projet.....	10
Programmation.....	11
1) Logiciels de programmation.....	11
2) Languages.....	11
3) Arduino IDE.....	12
a) Installation Arduino.....	12
b) Installation du support ESP32.....	12
4) Micro Python.....	13
a) Définitions.....	13
b) Installation via thonny ide.....	14
thonny ide.....	14
Installation de thonny ide.....	14
Installation du support ESP32.....	15
Installation des paquets.....	16
Programmer une led.....	16
5) Clignotement DEL.....	18
a) Clignotement simple.....	18
6) Détonateur.....	20
7) Voi télécommandée.....	22
a) Contrôles.....	22

APPAREIL

1) CARACTÉRISTIQUES

Selon l'étiquette de la pochette, le produit est RoHS, c'est-à-dire qu'il respecte la directive Restriction of Hazardous Substances. Ce qui signifie qu'il limite l'utilisation de 10 matières dangereuses.

Étiquette :

266-0558
11.44.9011 – 10
Hergestellt in China

Plaque :

WiFi ESP-WROOM-32
211-161007
FCCID : 2AG7Z-ESP32WROOM32

2) ESP 32

A) CARACTÉRISTIQUES TECHNIQUES

C'est un microcontrôleur SoC (système sur puce), intégrant la gestion du WiFi. Cette carte est idéale pour l'IdO (Internet des Objets), le domestique et le pont réseau (bridge). Elle intègre une antenne.

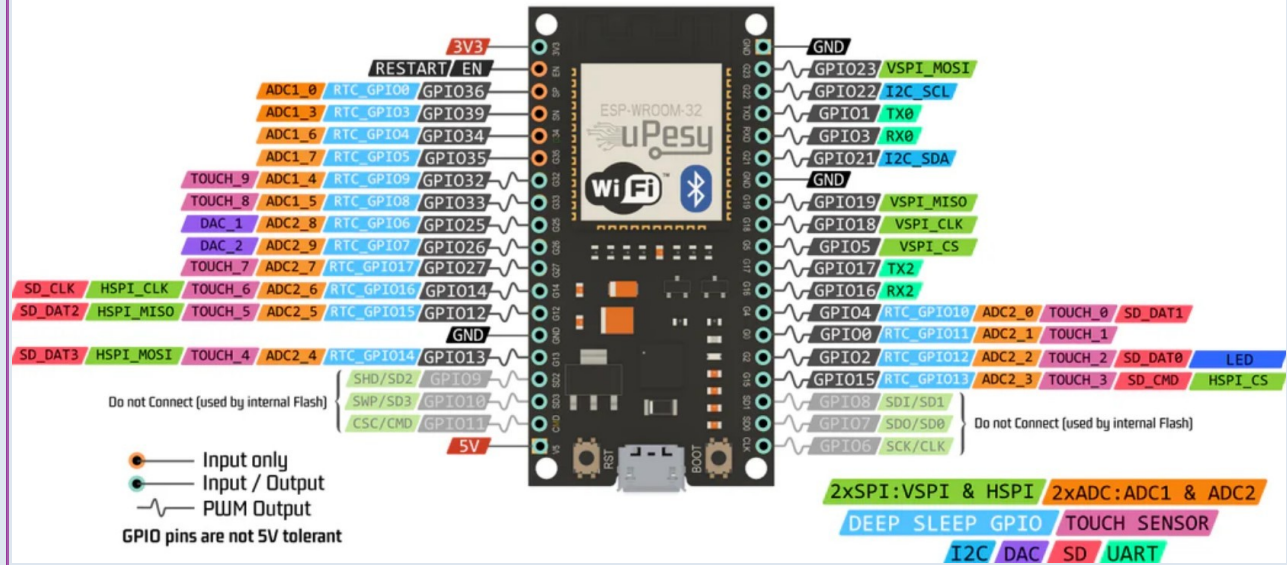
https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.youtube.com/watch%3Fv%3DHQmsdKdwQ_4&ved=2ahUKEwiK18uQr7iIAxXcAPsDHX-XBOUQtWJ6BAg5EAI&usg=AOvVaw05t68X3ooJrdq6KRRUhPBW

B) LISTE D'OPTIONS :

- CPU Xtensa 32 bits à 160 ou 240 MHz
- Mémoire de 520 KiO SRAM
- WiFi 802.11 b/g/n
- Bluetooth v4.2 BR/EDR et BLE v5.0 ou v5.1
- Segmentation 12 bits sur CAN jusqu'à 18 canaux
- 2 x 8 bits CNA
- 10 capteurs de touché GPIO
- 4 SPI
- 2 interfaces I²S
- 2 interfaces I²C
- 3 UART
- Contrôle hôte SD/SDIO/CE-ATA/MMC/eMMC
- Contrôleur esclave SDIO/SPI
- interface MAC Ethernet
- Bus CAN 2.0
- Contrôleur infrarouge distant (jusqu'à 8 canaux)
- Moteur PWM
- DEL PWM
- Capteur à effet Hall
- Pré-amplificateur analogique ultra-basse consommation
- fe

<https://fr.wikipedia.org/wiki/ESP32>

ESP32 Wroom DevKit Full Pinout



HW-395 V0.0.2

<https://fr.aliexpress.com/i/1005006356827958.html?gatewayAdapt=glo2fra>

C) BROCHES

D) LISTE D'ACCESSOIRES

- **Entrée mono**
 - Mécanique
 - ILS
 - Fin de course
 - Bouton poussoir
 - Non mécanique
 - Débitmètre
 - Hall
 - Opto-fourche
- **Entrée multiple**
 - Mécanique
 - Clavier matriciel
 - Non mécanique
- **Entrée impulsion**
 - Distance US
- **Entrée analogique mono**
 - Linéaire
 - ACS712-30A -20A -5A (non résistif)
 - LM35 (non résistif)
 - Résistance variable (résistif)
 - Non linéaire
 - Sharp distance (non résistif)
 - Capteur humidité sol (résistif)
 - CTN (résistif)
 - Photo R (résistif)
- **Entrée analogique multiple**
 - Non résistif
 - Résistif
 - Joystick
- **Sorties mono**
 - Direct
 - LED
 - LED IR
 - Laser
 - Puissance
 - Relais
 - Solid state R
 - MOSFET 5A
 - MOSFET 20A
- **Sorties multiples**
 - Direct :
 - Module de feux tricolores
 - I2C (module MD0035)
 - BPx5 (sur A0)
 - Puissance :
 - ULN 2003 (circuit intégré pour contrôler moteur pas à pas, relais, lampes...)

- Pont en H
 - L298
 - A4988
 - TB6600
- **Sorties MLI**
 - RGB
- **Sorties fréquence**
 - Piézo
- **Sorties servo**
 - Servomoteur
- **I2C**
 - Capteurs
 - ADS1115 16bits
 - Entrées
 -
 - Modules
 - « Multiprises » I2C
 - Afficheurs OLED
 - Digit
 - MultiLED
 - LCD
 - RTC DS3231
- **Divers**
 - DHT
 - DHT 11 sensor
 - Entrées
 - HX711
 - TTL
 - Modules
- **WS2812**
 - Rubans
 - Matrices
 - Modules
- **UART**
 - Bluetooth
- **SPI**
 - Carte SD SPI

3) GPS

A) PRINCIPE DE FONCTIONNEMENT

ÉQUATION DE NAVIGATION

Chaque signal satellite donne au récepteur l'équation suivante :

$$\rho_i = c \times (t_{r_i} - t_{e_i}) = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} + c\Delta t$$

avec :

ρ_i : la distance théorique

x_i, y_i, z_i : les coordonnées du satellite i

x, y, z : les coordonnées du récepteur

Δt : le décalage horaire de l'horloge interne du récepteur

t_{r_i} : le temps de réception du signal émis t_{e_i} par le satellite

En développant on obtient alors :

$$\begin{aligned} (\rho_i - c\Delta t)^2 &= (x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 \\ (x_i^2 + y_i^2 + z_i^2 - \rho_i^2) - 2 \times (x_i x + y_i y + z_i z - \rho_i c\Delta t) + (x^2 + y^2 + z^2 - (c\Delta t)^2) &= 0 \end{aligned}$$

On introduit alors $s_i = [\]^T$, et le pseudo-produit scalaire de Lorentz défini pour tout quadrivecteur u et v par . L'équation se réécrit alors :

TRANSFORMATION DE LORENTZ

SAE

1) PROJET

Réalisation d'un système de chronométrage pour le challenge robot de première année.

Plusieurs robots devraient participer à une course. Notre robot serait en charge d'estimer le temps des autres robots.

PROGRAMMATION

1) LOGICIELS DE PROGRAMMATION

- Arduino
- Visual Studio Code (a.k.a VS Code)
- Thony ??
- ESP-IDF de chez Expressif ??
- FreeRTOS (portable et open source) ??
-

2) LANGUAGES

- MicroPython
- C

3) ARDUINO IDE

A) INSTALLATION ARDUINO

Installer l'IDE Arduino.

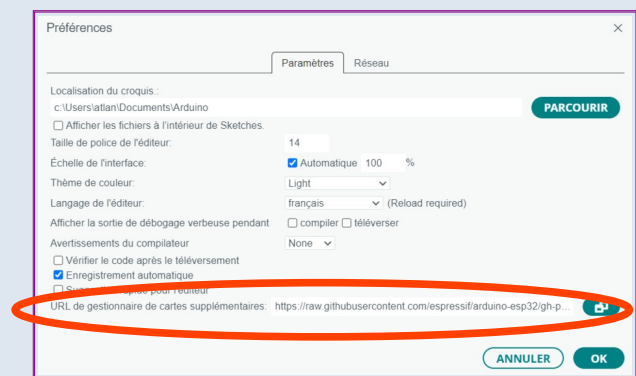
B) INSTALLATION DU SUPPORT ESP32

Pour programmer le ESP32 sous Arduino, il faut et copier :

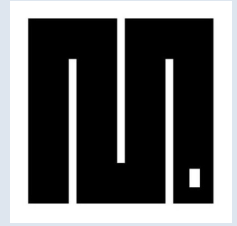
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Fichiers ⇒ Préférences :

dz :



4) MICRO PYTHON



Site d'aide : <https://micropython.fr/>

A) DÉFINITIONS

C'est le portage du langage Python sur des microcontrôleurs.

Pour l'ESP32, on peut utiliser WebREPL, servant à communiquer via websocket¹ depuis un navigateur avec Micro Python. Il est possible de mettre en place un serveur websocket pour avoir une application « Brython » dans le navigateur.

Cf : <https://docs.micropython.org/en/latest/esp32/tutorial/intro.html>

Le MicroPython permet une grande rapidité de téléversement, contrairement à Arduino.

1 Websocket : Protocole réseau pour créer une communication entre un navigateur web TCP et un full duplex.

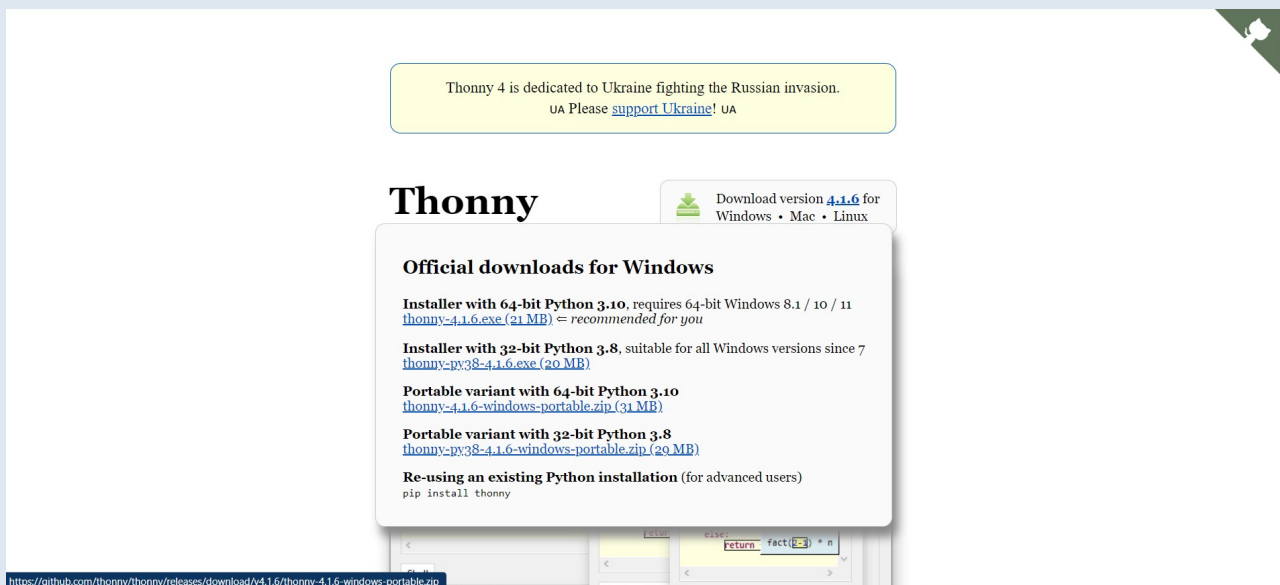
B) INSTALLATION VIA THONNY IDE

THONNY IDE

Thonny IDE est un IDE libre de droit pour utiliser du Python facilement.

INSTALLATION DE THONNY IDE

Télécharger le fichier d'installation correspondant à votre configuration sur le site <https://thonny.org/>.

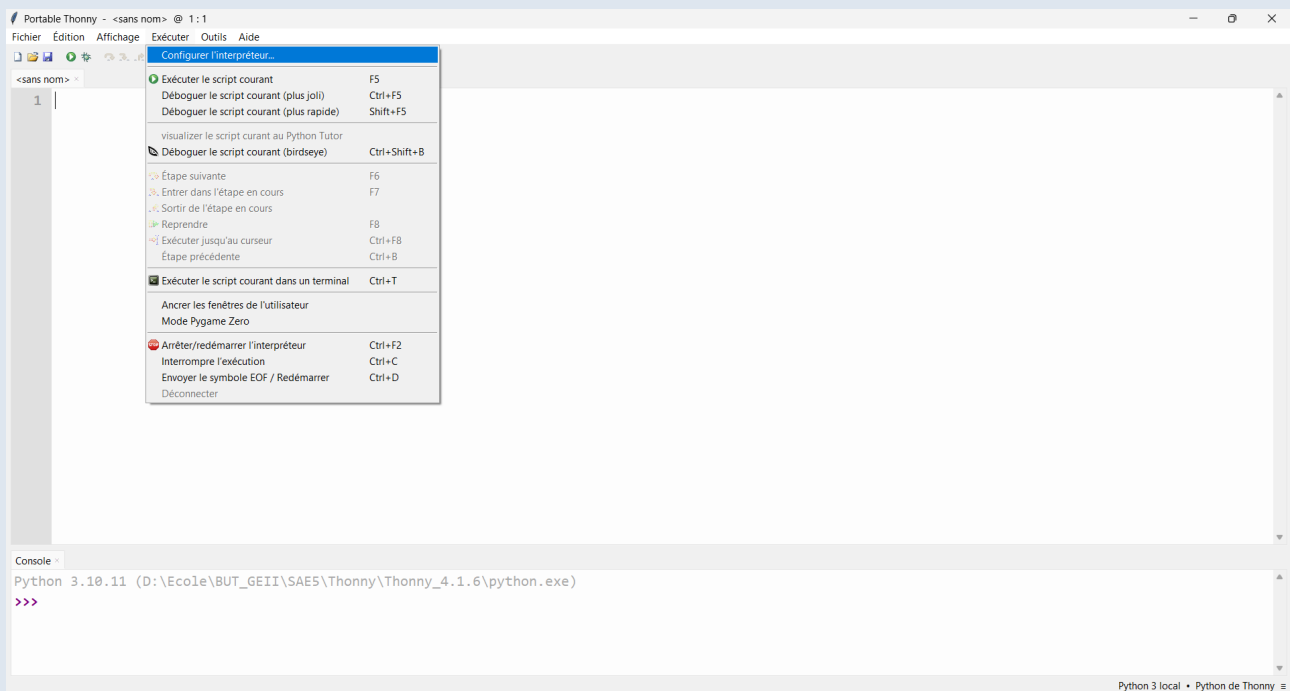


Si vous souhaitez utiliser cet IDE sur votre propre ordinateur, vous pouvez télécharger la version de base. Mais si vous souhaitez travailler sur plusieurs ordinateurs, alors il est recommandé de prendre la version portable du logiciel. Cela permet d'utiliser une clé USB pour transporter le logiciel, aussi pratique si vous ne possédez pas de droit administrateur sur votre machine.

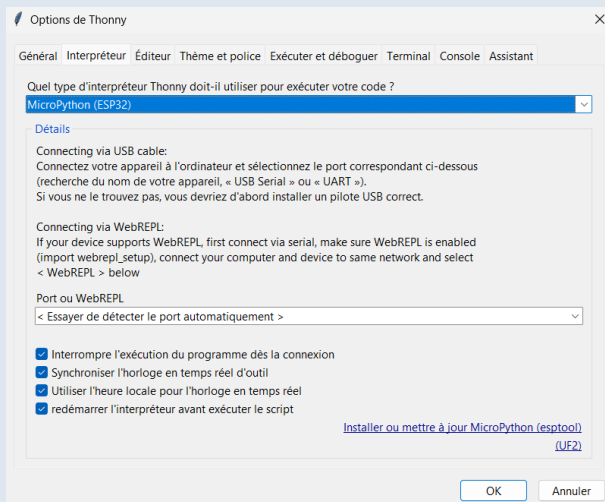
La version utilisée de Thonny est la 3.8 – 4.1.6.

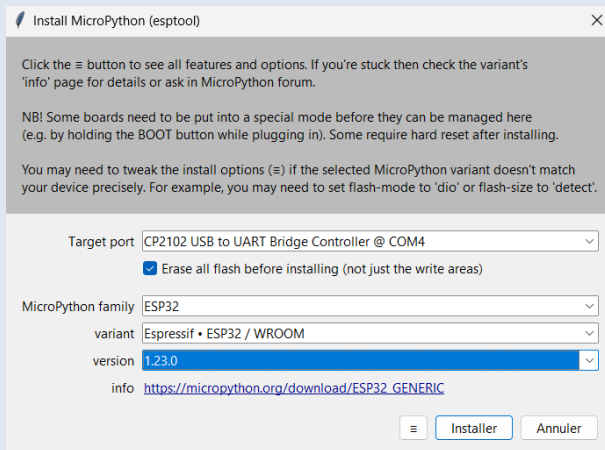
INSTALLATION DU SUPPORT ESP32

Pour installer MicroPyhton, il faut aller dans **Exécuter ⇒ Configurer l'interpréteur**.

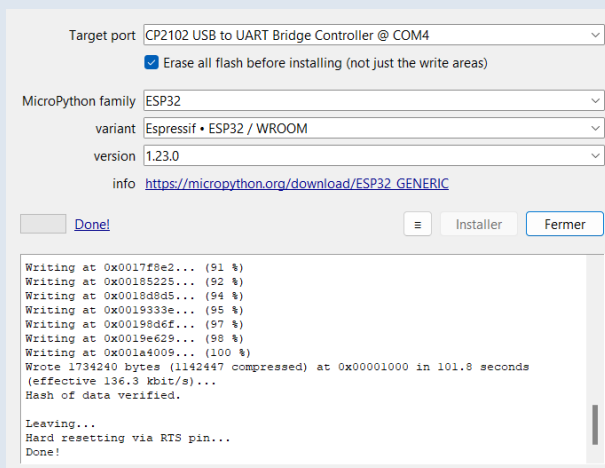


Brancher votre carte ESP 32. Choisir l'interpréteur MicroPyhton (ESP32) et cliquer sur « Installer ou mettre à jour MicroPython (esptool) ».





Si votre carte est bien branchée, elle sera déclarée dans la case « Target port ». Il faut ensuite que vous choisissiez la famille, la variante et la version du firmware. Puis installer.



INSTALLATION DES PAQUETS

La programmation en Python nécessite d'avoir plusieurs bibliothèques à disposition.

PROGRAMMER UNE LED

Utiliser le programme suivant :

```
from machine import Pin
```

```
LED = Pin(2, Pin.OUT)
```

```
LED.value(0)
```

Cf : <https://fr.macerobotics.com/apprendre-le-micropython-avec-la-carte-raspberry-pi-pico/>

Pour la faire clignoter, il faut inclure le temps :

```
from machine import Pin
import time
```

```
LED = Pin(2, Pin.OUT)
```

```
while True:
    LED.value(1)
    time.sleep(0.4)
    LED.value(0)
    time.sleep(0.4)
```

Le « time.sleep » donne le temps de l'action en secondes. Ici 400 ms.

Il est également possible d'utiliser la fonction « toggle » (à bascule) de la bibliothèque machine.

5) CLIGNOTEMENT DEL

A) CLIGNOTEMENT SIMPLE

Notre ESP 32 est composé de deux DEL.

6) DÉTONATEUR

Un jeu éducatif ^pir

Application sur téléphone

7) VOI TÉLÉCOMMANDÉE

A) CONTRÔLES

La voiture peut être contrôlée par le téléphone, soit en touchant sur la carte où doit se diriger le véhicule, soit avec avancer, reculer, gauche et droite.