

# I2C raspberry <--> ESP32

## Installer le module smbus python3

```
sudo apt-get install python3-smbus
```

faire :

```
sudo pip list ( pour verifier l'installation )
```

## Programme en python sur RPI

[exemple010.py](#)

```
from smbus import SMBus

addr = 0x55 # L'adresse du bus I2C sur l'ESP32
bus = SMBus(1) # indique le repertoire /dev/ic2-1

numb = 1

print ("taper 1 pour ON ou 0 pour OFF")

while numb == 1:
    ledstate = input(">>>>")
    if ledstate == "1": # Allume la Led gpio 12 sur l'ESP32
        bus.write_byte(addr, 0x1) # Allume sur 1
    elif ledstate == "0": # Eteint la Led gpio 12 sur l'ESP32
        bus.write_byte(addr, 0x0) # Eteint sur 0
    else:
        numb = 0
```

## Programme reception I2C C-Arduino sur ESP32

[exemple012.ino](#)

```
/*Arduino Slave for Raspberry Pi Master
i2c_slave_ard.ino
Connects to Raspberry Pi via I2C

DroneBot Workshop 2019
https://dronebotworkshop.com
*/
```

```
// Include the Wire library for I2C
#include <Wire.h>

// LED on pin 12
const int ledPin = 12;

void setup() {
  // Configure l'adresse I2C de l'ESP32 SLAVE
  Wire.begin(0x55);

  // Reçoit les données du bus I2C
  Wire.onReceive(receiveEvent);

  // Active la LED 13 sur GPIO 12 et l 'eteint
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
}

void receiveEvent(int howMany) {
  while (Wire.available()) { // boucle tant que l on reçoit des
    caractères sur le bus I2C
      char c = Wire.read(); // reçoit les caracteres envoyés sur le bus
      digitalWrite(ledPin, c); // Allume ou eteint la LED 12
    }
}

void loop() {
  delay(100);
}
```

From:

<http://chanterie37.fr/fablab37110/> - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:

<http://chanterie37.fr/fablab37110/doku.php?id=start:raspberrypi2c>

Last update: **2025/03/02 18:34**

