

ESP32 ARDUINO

L'**ESP32** est un SOC développé par la société Espressif dédié à l'internet des objets (**IoT**) et plus particulièrement les communications sans fil Wifi et Bluetooth pour un coût réduit. Il a été décliné en version module l'**ESP-WROOM-32** qui a lui même été intégré par différents fabricants (Essentiellement Chinois) sur des cartes de développement.

Il présente plusieurs intérêts :

- Prix faible ~5€ pour le module ~8€ pour une petite carte de développement
- Intègre du Wifi 802.11 b/g/n/e/i (WPA)/WPA2/WPA2-Enterprise/Wi-Fi Protected Setup (WPS)
- Intègre Bluetooth **4.2** - BLE Bluetooth low Energy
- Compatible avec l'environnement de développement **ARDUINO**
- Intègre un microcontrôleur 32 bits performants et de nombreux périphériques (ADC **12bit**, DAC, 3xUART, PWM, I2C, SPI, etc ...)
- S'alimente directement en USB
- etc.



Caractéristiques

CPU

Xtensa® single-/dual-core 32-bit LX6 microprocessor(s), up to 600 DMIPS (200 DMIPS for single-core microprocessor)

- 448 kB ROM
- 520 kB SRAM
- 16 kB SRAM in RTC
- QSPI flash/SRAM, up to 4 x 16 MB
- Power supply: 2.3V to 3.6V

Hardware

- Périphériques : SD card, UART, SPI, SDIO, I2C, LED PWM, Motor

PWM, I2S, IR

- GPIO : capacitive touch sensor, ADC, DAC, LNA preamplifier
- Capteurs intégrés : Hall sensor, temperature sensor
- Alimentation : 2.7 ~ 3.6V
- Courant max : 500 mA
- Température de fonctionnement : -40°C ~ +85°C

WIFI

- Wi-Fi Protocols : 802.11 b/g/n/e/i (802.11n up to 150 Mbps)
- Wi-Fi mode : Station/SoftAP/SoftAP+Station/P2P
- Wi-Fi Security : WPA/WPA2/WPA2-Enterprise/WPS
- Network protocols : IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT

OS

- FreeRTOS

Brochage

PINOUT



ESP-WROOM-32



Installation de l'environnement de développement

Procédure d'installation de l'environnement ARDUINO

- Installer l'environnement ARDUINO : <https://www.arduino.cc/en/Main/Software>
- Télécharger l'add-on pour l'ESP32 : <https://github.com/espressif/arduino-esp32>
- Deziper l'add-on dans le répertoire : Mes Documents\Arduino\hardware\espressif\esp32



* Executer get.exe qui se trouve dans le répertoire Mes Documents\Arduino\hardware\espressif\esp32\tools * L'executable get.exe va télécharger et intégrer l'add-on dans l'environnement ARDUINO



L'executable get.exe va télécharger 3 fichiers ZIP qui seront téléchargés dans le répertoire dist (esptool-4dab24e-windows.zip, mkspiffs-0.2.1-windows.zip et xtensa-esp32-elf-win32-1.22.0-75-gbaf03c2-5.2.0.zip)

* Démarrer l'environnement ARDUINO et vérifier que l'ESP32 apparaît bien dans les types de carte :



* Brancher votre carte ESP32, vérifier le port COM qui lui a été attribué :



* Créer un nouveau sketch, sélectionner la carte: ESP32 Dev Module et le port qui lui a été associé



* Vérifier la compilation et le transfert

1er sketch de test - clignotement LED



Ma carte intègre une LED sur la sortie IO2

```
void setup() {
  // put your setup code here, to run once:
  pinMode(2, OUTPUT);          // set pin to output
}

void loop() {
  // put your main code here, to run repeatedly:

  digitalWrite(2, HIGH);      // sets the LED on
  delay(200);                 // waits for a second
  digitalWrite(2, LOW);       // sets the LED off
  delay(800);
}
```

Sketch exemples



De nombreux exemples sont fournis dans le répertoire :
Documents\Arduino\hardware\espressif\esp32\libraries

Sketch transmission wifi

Le sketch ci dessous va transmettre une trame toutes les secondes vers un serveur UDP. Remplacer par **your-ssid** par le nom de votre réseau WIFI, **your-password** par le mot de passe de votre réseau WIFI et 192.168.1.10 par l'adresse IP de votre PC.

```
/*
```

```
* This sketch sends random data over UDP on a ESP32 device
*
*/
#include <WiFi.h>
#include <WiFiUdp.h>

// WiFi network name and password:
const char * networkName = "**your-ssid**";
const char * networkPswd = "**your-password**";

//IP address to send UDP data to:
// either use the ip address of the server or
// a network broadcast address
const char * udpAddress = "**192.168.1.10**";
const int udpPort = 2205;

//Are we currently connected?
boolean connected = false;

//The udp library class
WiFiUDP udp;

void setup(){
  // Initilize hardware serial:
  Serial.begin(115200);
  //Connect to the WiFi network
  connectToWiFi(networkName, networkPswd);
}

void loop(){
  //only send data when connected
  if(connected){
    //Send a packet
    udp.beginPacket(udpAddress,udpPort);
    udp.printf("Seconds since boot: %u", millis()/1000);
    udp.endPacket();
  }
  //Wait for 1 second
  delay(1000);
}

void connectToWiFi(const char * ssid, const char * pwd){
  Serial.println("Connecting to WiFi network: " + String(ssid));

  // delete old config
  WiFi.disconnect(true);
  //register event handler
  WiFi.onEvent(WiFiEvent);
  //Initiate connection
```

```
WiFi.begin(ssid, pwd);

Serial.println("Waiting for WIFI connection...");
}

//wifi event handler
void WiFiEvent(WiFiEvent_t event){
  switch(event) {
    case SYSTEM_EVENT_STA_GOT_IP:
      //When connected set
      Serial.print("WiFi connected! IP address: ");
      Serial.println(WiFi.localIP());
      //initializes the UDP state
      //This initializes the transfer buffer
      udp.begin(WiFi.localIP(),udpPort);
      connected = true;
      break;
    case SYSTEM_EVENT_STA_DISCONNECTED:
      Serial.println("WiFi lost connection");
      connected = false;
      break;
  }
}
```

L'ESP32 va transmettre toutes les secondes une trame vers le port 3333 de votre PC. Pour visualiser les trames envoyées, il suffit d'utiliser un serveur UDP comme [HERCULE](#)

Démarrer HERCULE et dans l'onglet UDP mettre 3333 dans le port à utiliser puis cliquer sur "Listen"



Liens utiles

- [Site Espressif sur le ESP32](#)
- [Datasheet ESP32](#)
- [Datsheet carte ESP WROOM 32](#)
- [Projets DIY - Excellent site avec pas mal d'infos en français](#)

Schema de la carte de developpement ESP32 DEVKIT V1 www.doit.am/schematicsforesp32.pdf

From:

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

Permanent link:

<https://www.fablab37110.chanterie37.fr/doku.php?id=start:arduino:esp32:wikiexterne&rev=1609429795>

Last update: **2023/01/27 16:08**

