



## Práctica: Configuración del ambiente e introducción al ESP32

### Objetivo:

- Familiarizarse con el IDE de Arduino
- Agregar y configurar las bibliotecas y recursos necesarios para futuras prácticas

### Introducción

Un microcontrolador es un circuito integrado capaz de realizar actividades previamente programadas y descargadas en su memoria. De manera general podemos comprender a los microcontroladores en 3 bloques funcionales, su unidad de procesamiento, su memoria y sus respectivos puertos de entrada o salida.

### Descripción:

Una de las mayores tendencias comerciales debido a la explosión del denominado “*Internet of Things*” fue sin duda el uso de microcontroladores, esto debido a factores como su bajo costo de producción además de un bajo consumo energético.

La presente práctica hace uso de uno de los microcontroladores más famosos de la empresa ESPRESSIF, el cual además de tener un bajo, a su vez integra en el mismo chip tecnología como Wi-Fi, Bluetooth e incluso un módulo de cámara integrado.

A continuación, algunos aspectos generales del microcontrolador que se usará

Tabla 1 Características generales del ESP-32

Características del ESP32-CAM	
Procesador	Procesador de 32 bits de 160 MHz
RAM	520 KB SRAM integrada, expandible 4 MB PSRAM
Puertos soportados	UART/SPI/I2C/PWM/ADC/DAC
Tecnologías soportadas	TFCards, OV2640 and OV7670 cameras

### Tabla de entradas y salidas:

A continuación, se muestra una imagen con las entradas y salidas del ESP32-CAM:

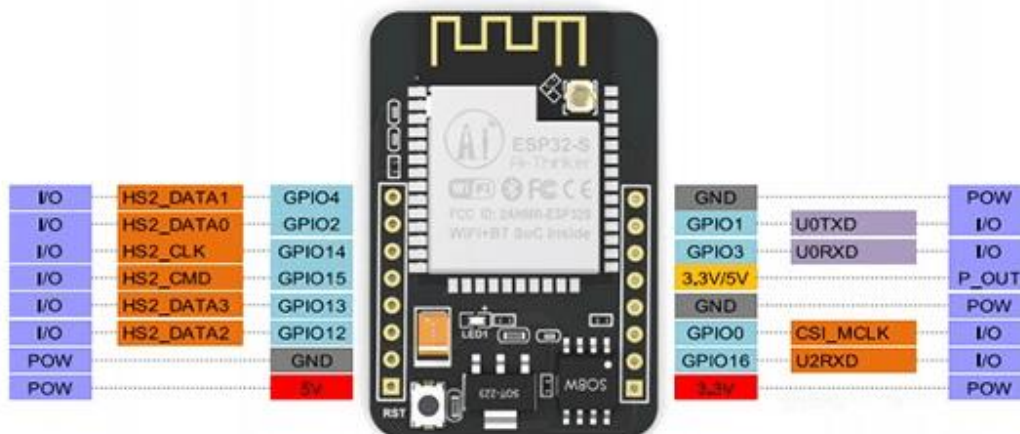


Figura 1. Características generales del ESP-32

Tabla 2 Descripción de los puertos del ESP-32

Puerto	Tipo	Descripción
<b>GPIO 4</b>	I/O	Datos 1 aunque también se puede emplear para controlar el LED que está integrado en el ESP32
<b>GPIO 2, 12 y 13</b>	I/O	Puertos de entrada y salida de datos
<b>GPIO 1 y 3</b>	TXD y RXD	Son los pines del puerto serial por lo que se usan para cargar el código en memoria
<b>GPIO 0</b>	MCLK	Además de ser de entrada y salida de datos sirve también para determinar los modos del microcontrolador
<b>GPIO 14 y 15</b>	CLK y CMD	Además de ser puertos de entrada y salida de datos, también pueden ser usados para realizar acciones a más bajo nivel

Además, podemos notar que también tenemos 3 pines GND además de 2 pines para alimentación uno de 3.3V y otro de 5V.

### Diagrama de conexiones:

Entre las muchas ventajas que nos ofrece el ESP32-CAM es un módulo de cámara\*, específicamente el modelo cam-OV2640, sin embargo, debido a lo compacto que es podemos observar que no integra una conexión directa a USB, por lo que se usará un programador FTDI para de esta forma descargar nuestros proyectos en memoria.

A continuación, se muestra el esquema de cómo se conectaría el ESP32 con el programador FTDI:

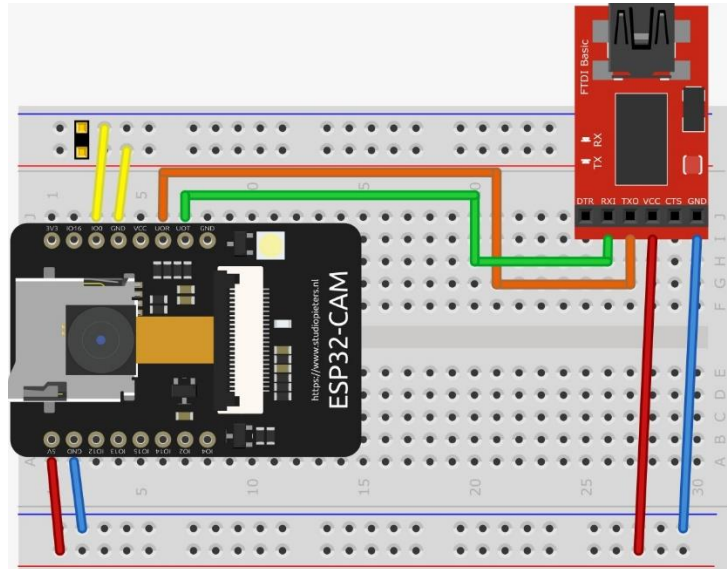


Figura 2. Esquema de conexión del ESP32 y el programador FTDI

Observamos que tenemos usados los puertos correspondientes GND y alimentación de 5V (Azul y rojo), a su vez observamos que se encuentran conectados el GPIO1 y GPIO3 (Naranja y verde) que serán los puertos para descargar nuestros futuros programas, por otro lado, debido a que vamos a grabar nuestro programa en memoria necesitamos conectar GPIO 0 a GND en este caso se empleó directamente el que está al lado del pin 0 (Amarillo).

### Configuración del ambiente en Arduino IDE:

Una vez conocido el hardware, es necesario configurar el ambiente de desarrollo que emplearemos para programar, compilar y descargar nuestros programas.

En este caso para poder compilar y descargar el código a nuestro microcontrolador se empleará Arduino IDE, en caso de no tenerlo descárgalo de la página oficial de Arduino;  
<https://www.arduino.cc/>

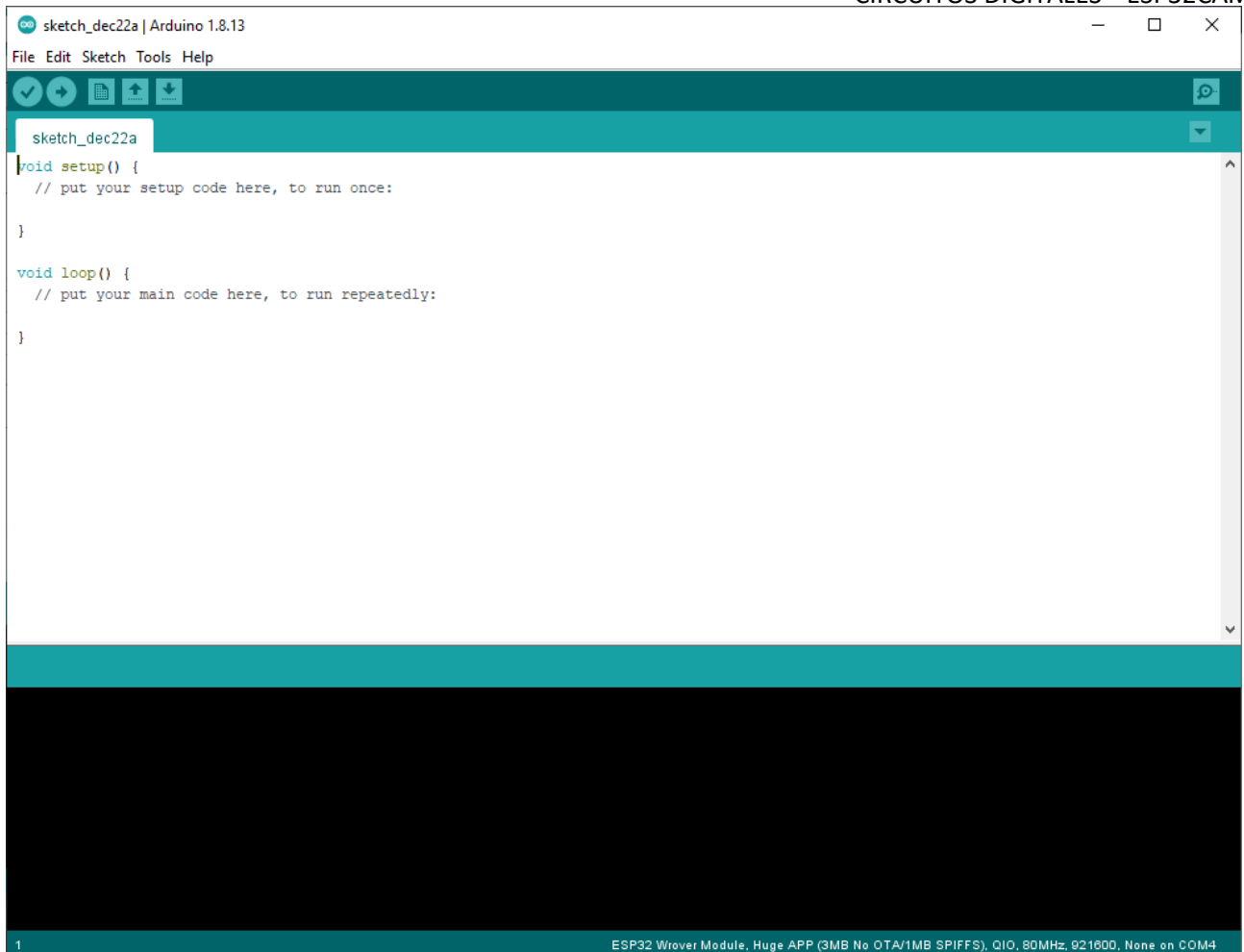


Figura 2. Aspecto general del IDE de arduino

Una vez instalado el IDE, lo primero que debemos hacer es descargar las bibliotecas de ESP32 necesarias para trabajar, para ello en el apartado de **Arduino > Preferences**, agregar el siguiente URL [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) en el apartado de “**Additional Board Manager URLs**”

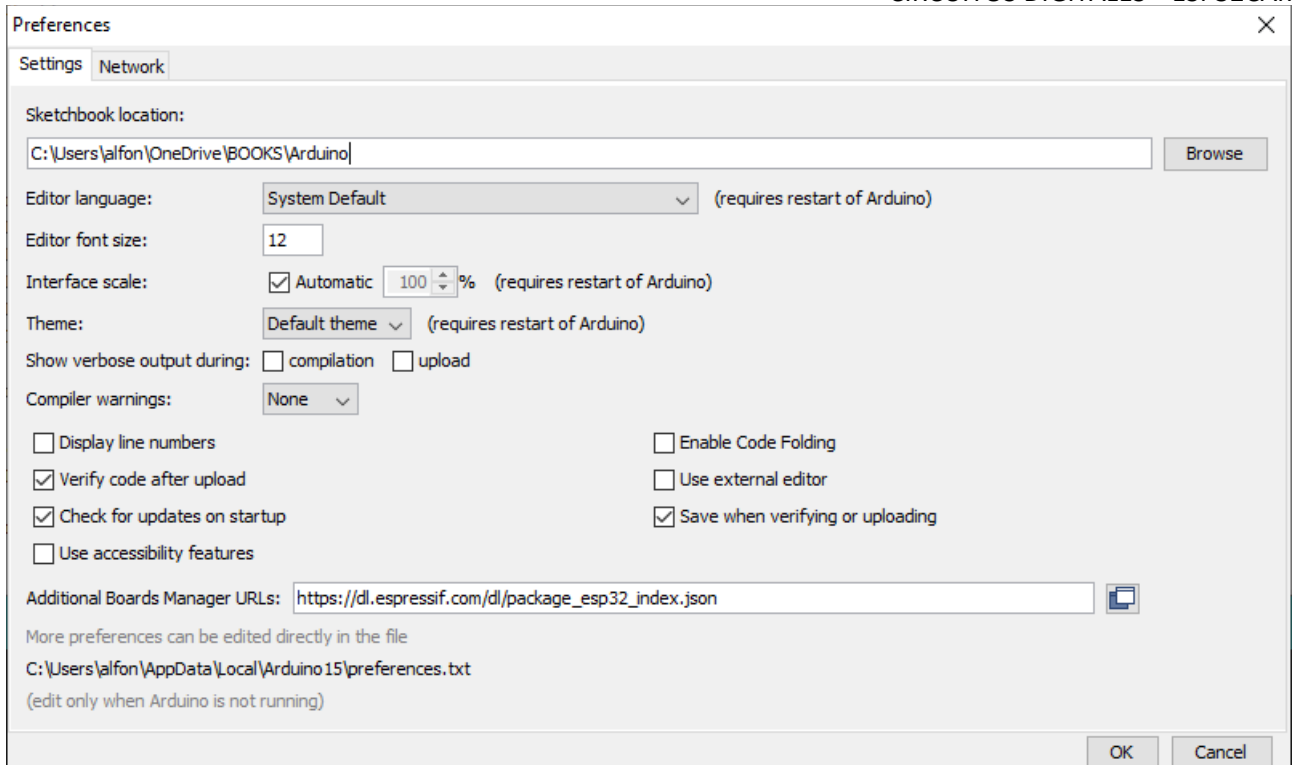


Figura 3. Apartado donde se agregan las bibliotecas externas en el IDE de arduino

Posteriormente, ir al apartado de **Tools > Board > Boards Manager...** , escribir o buscar **ESP32** e instalar la biblioteca resultante:

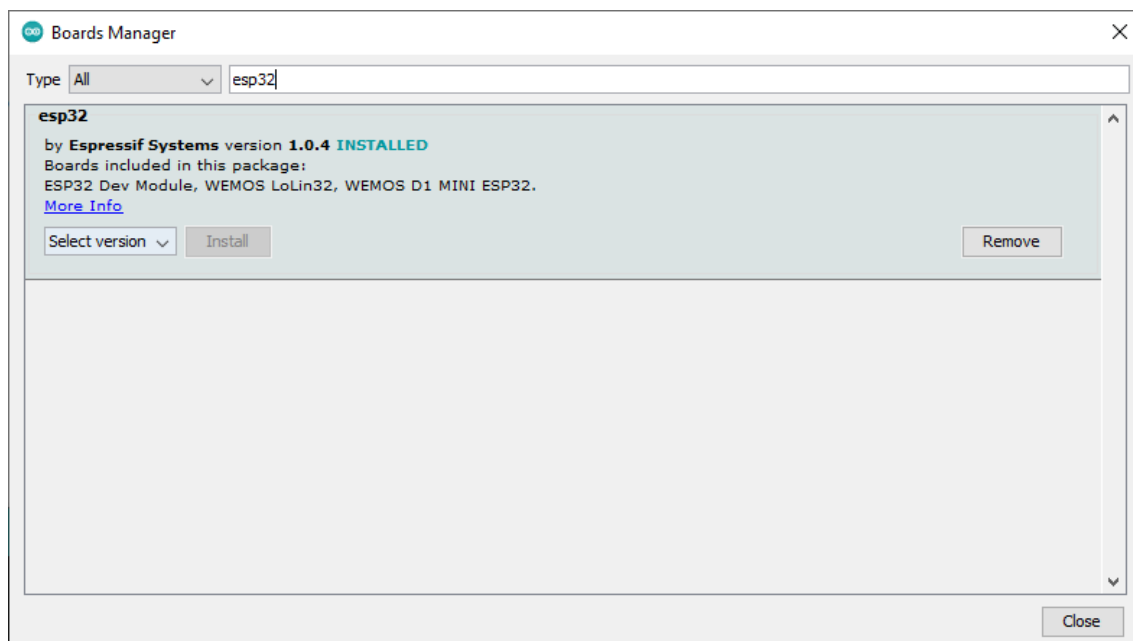


Figura 4. Apartado de descarga de la biblioteca principal del ESP32



Una vez instalada escoger la tarjeta a la que vamos a descargar nuestro programa, específicamente es la denominada **“ESP32 Wrover Module”**

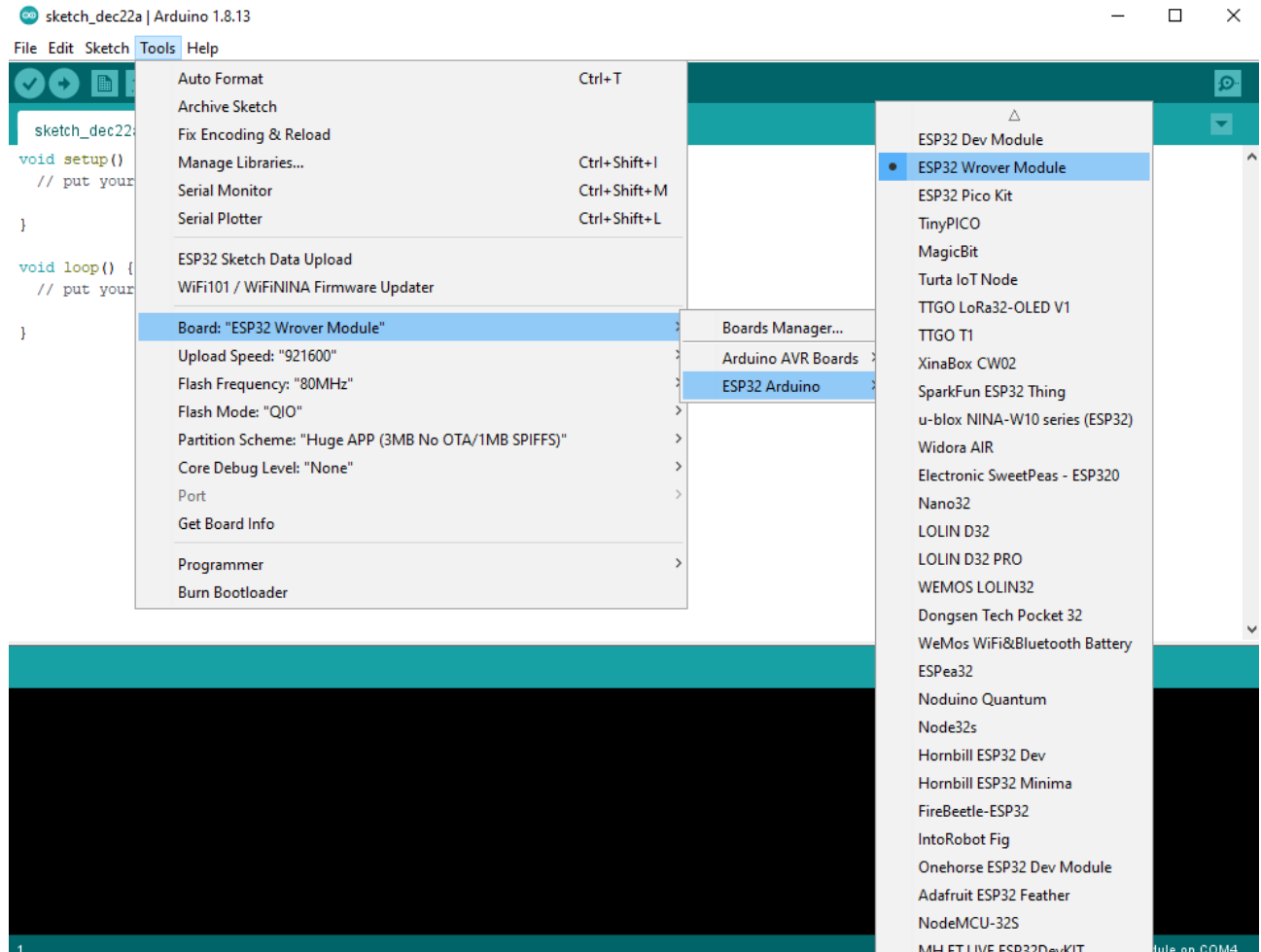


Figura 5. Selección del modelo específico de nuestro ESP-32

Por último, a continuación, se muestra la compilación y descarga de un programa en nuestro microcontrolador, podemos observar que en color rojo se nos pide que hagamos un reset mediante el botón que tiene nuestro micro, esto se debe a que estamos descargando el programa y es necesario fijarlo en memoria



```
ESP32CAM_Car | Arduino 1.8.13
File Edit Sketch Tools Help

ESP32CAM_Car app_httpd.cpp camera_index.h

#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! Make sure that you have either selected ESP32 Wrover Module,
//           or another board which has PSRAM enabled
//
// Adafruit ESP32 Feather

// Select camera model
// #define CAMERA_MODEL_WROVER_KIT
// #define CAMERA_MODEL_M5STACK_PSRAM
#define CAMERA_MODEL_AI_THINKER

const char* ssid = "murryFly"; //Enter SSID WIFI Name
const char* password = "oQEeCponFG"; //Enter WIFI Password

#if defined(CAMERA_MODEL_WROVER_KIT)
#define PWDN_GPIO_NUM    -1
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM    21
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27

#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36

Done uploading.
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 1536.0 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

1 ESP32 Wrover Module
```

Figura 6. Visualización general de cómo se vería la interfaz una vez compilado y descargado el código en nuestro ESP

## Notas y reglas de funcionamiento

Una vez descargado el programa, antes de oprimir el botón de reset es necesario desconectar el GPIO0 del GND (Amarillo), en caso de no hacerlo en el monitor serial de nuestro IDE se mostraría un mensaje de “programa por descargar”.

## Recursos extras:

Para más información acerca del ESP32, puede consultarse la página oficial de ESPRESSIF:  
<https://www.espressif.com/en/products/socs/esp32/overview>