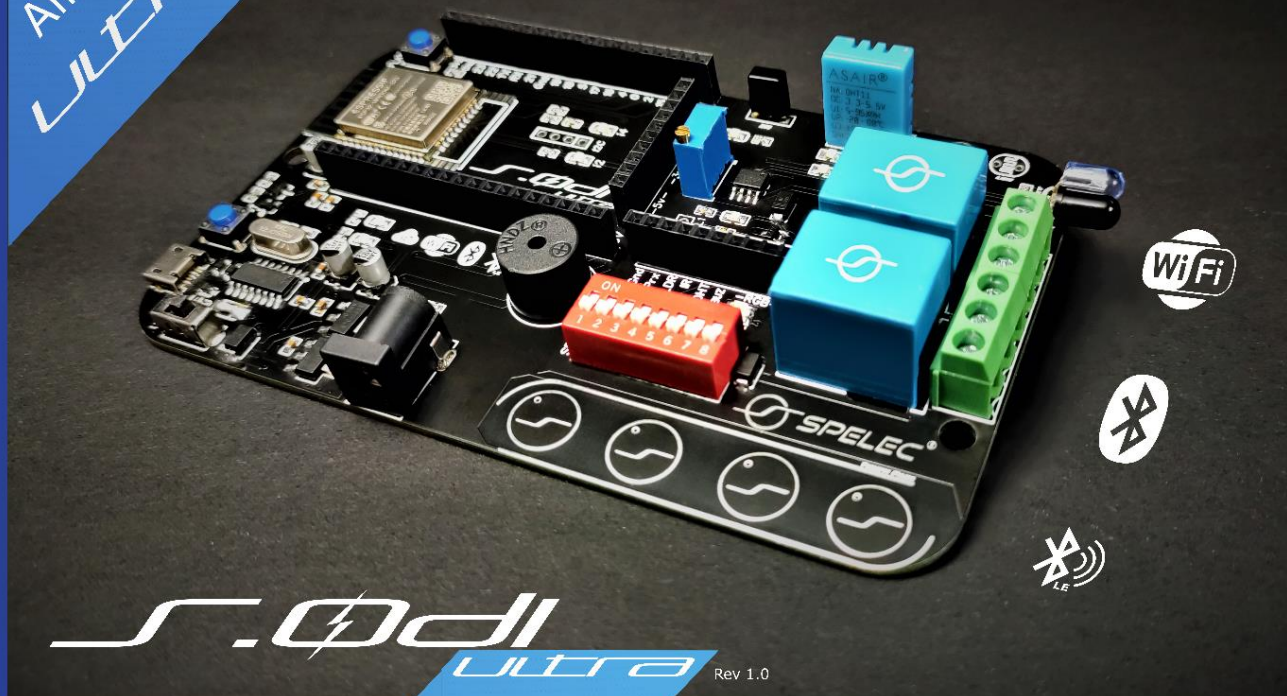


All New  
**S.ODI**  
Series

**SPELEC**  
TRANSFORMING THE WORLD



**S.ODI**  
**ULTRA**

Rev 1.0



ARDUINO IDE



PLATFORM IO



MICRO-PYTHON



OPEN-HARDWARE

Dual-Core Processor

Wi-Fi

Bluetooth 4.2, BLE

Infra-red communication

Ultra Low Power co-processor

Several Peripherals

Multiple sensors

All Gpio Pulsewidth modulated  
handles upto 25 sensors

Powered by Dual core40nm technology, S.ODI Ultra provides a robust,highly integratedplatform, which helps to develop effecient prototype & application development with its compact design,mulultiple sensor interfaces and high Performance cores

Visit us at:

[https://github.com/SPELECaautomations/S.ODI\\_ultra](https://github.com/SPELECaautomations/S.ODI_ultra)

 [www.spelec.in](http://www.spelec.in)

## Get Started

- Introduction
- What You Need
- Guides
- S.ODI Ultra v1.0 Dev Board
  - Overview
  - Functionality Overview
  - Functional Description
  - Board Architecture
  - Key Components
  - Setup Toolchain
  - CH340G Drivers
  - Download and Install Drivers
  - Arduino IDE
  - Installation step by step
  - Start a Project
  - Libraries & Frameworks
  - Examples
  - Micro Python
  - Esp-IDF
  - Libraries & Frameworks
  - Applications
  - Product Comparisons

## System. On-Board Development Interfaces

**TRANSFORM YOUR DREAMS TO REALITY**

You can construct your intelligent hardware system with very low to no effort using our development board S.ODI developed by SPELEC.

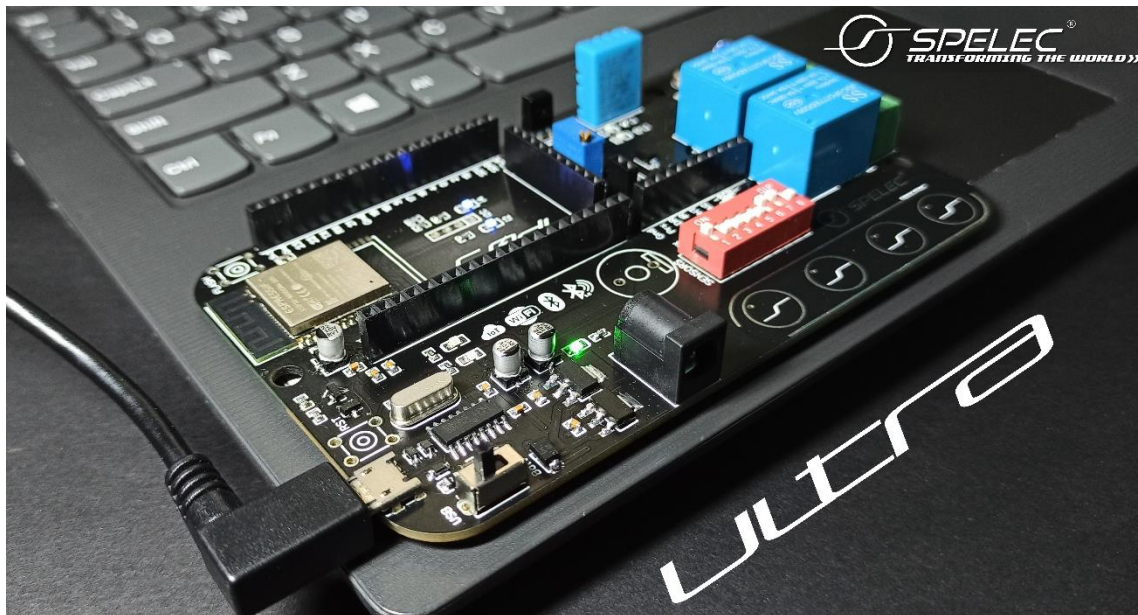
The S.ODI (System onboard development interface) is a Development board that has integrated sensors regarding Humidity, Temperature, flame, LDR and infrared it also has RGB (LED light) which works as an indicator. S.ODI also comes with onboard WI-FI, Bluetooth, BLE which are used to connect with various devices.

With the collection of onboard hardware components and the open-source firmware that we supply one can quickly learn how to get hardware connected online, and how to use Web API and/or smartphones to interact with it.

### Get Started

This document is intended to help you set up the software development environment for the hardware based on the S.ODI Ultra Dev Board by SPELEC.

After that, a simple example will show you how to use S.ODI (System onboard Development Interfaces) and software's for menu configuration, then building, and flashing firmware onto an S.ODI board.



## Introduction

S.ODI is a system on board that integrates the following features:

- **Wi-Fi (2.4 GHz band)**
- **Bluetooth 4.2**
- **Bluetooth low Energy**
- **Infrared comm**
- **Dual high-performance cores**
- **Ultra-Low Power co-processor**
- **Several peripherals & Sensors**

Powered by 40 nm technology, S.ODI provides a robust, highly integrated platform, which helps meet the continuous demands for efficient power usage. It also comes with compact design, security, high performance and reliability.

SPELEC provides basic hardware and software resources to help application developers realize their ideas using the S.ODI series hardware. The software development framework by SPELEC is intended for the development of Internet-of-Things (IoT) applications with Wi-Fi, Bluetooth, power management and several other system features.

## What You Need

Hardware:

- An S.ODI Dev board
- USB cable - USB A to micro B

Computer running Windows, Linux, or macOS Software:

- Arduino IDE
- ESP-IDF
- Micro-Python



S.ODI is an ESP32-based development board produced by SPELEC. This board features multiple integrated sensors and interfaces.

It's another distinguishing feature includes the multiple Embedded Sensors, Interfaces and CH340G / CP2102 - advanced multi-interface USB bridges. This chip enables usage of UART for direct debugging of S.ODI through the USB interface. S.ODI makes development effortless, convenient, easy, and cost-effective. Most of the S.ODI I/O pins are broken out to the board's pin headers for easy access.

## Functionality Overview

### • Wi-Fi Key Features

802.11 b/g/n

- 802.11 n (2.4 GHz), up to 150 Mbps
- WMM
- TX/RX A-MPDU, RX A-MSDU
- Immediate Block ACK
- Defragmentation
- Automatic Beacon monitoring (hardware TSF)
- 4 × virtual Wi-Fi interface
- Antenna diversity





### • BT Key Features

- Enhanced Power Control
- +12 dBm transmitting power
- NZIF receiver with -97 dBm BLE sensitivity
- Compliant with Bluetooth v4.2 BR/EDR and BLE specification
- Class-1, class-2 and class-3 transmitter without external power amplifier
- Adaptive Frequency Hopping (AFH)
- Standard HCI based on SDIO/SPI/UART
- High-speed UART HCI, up to 4 Mbps
- Bluetooth 4.2 BR/EDR BLE dual mode controller
- Synchronous Connection-Oriented/Extended (SCO/eSCO)
- CVSD and SBC for audio codec
- Bluetooth Pico net and Scatter net
- Multi-connections in Classic BT and BLE
- Simultaneous advertising and scanning
- Ultra-Low Power Consumption



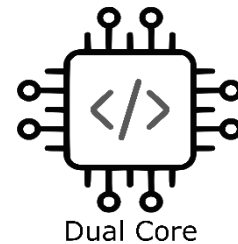
Robust Design

- **MCU and Advanced Features**

PROCESSOR SERIES	PROCESSOR CORES	EMBEDDED FLASH MEM.	PACKAGE SIZE
ESP32- D0WDQ6		16MB	

- **CPU and Memory**

- Xtensa® single-/dual-core
- 32-bit LX6 microprocessor(s), up to 600 MIPS
- 16 MB FLASH Memory
- 520 KB SRAM
- 16 KB SRAM in RTC
- QSPI supports multiple flash/SRAM chips

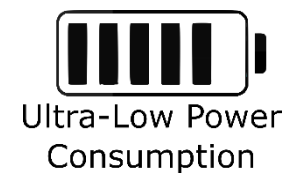


- **Clocks and Timers**

- Internal 240 MHz oscillator with calibration
- Internal RC oscillator with calibration
- External 2 MHz ~ 60 MHz crystal oscillator (40 MHz only for Wi-Fi/BT functionality)
- External 32 kHz crystal oscillator for RTC with calibration
- Two timer groups, including 2 × 64-bit timers and 1 × main watchdog in each group
- One RTC timer
- RTC watchdog

- **Advanced Peripheral Interfaces**

- 30 × programmable GPIOs
- 12-bit SAR ADC up to 18 channels
- 2 × 8-bit DAC
- 10 × touch sensors (4 On-board)
- 4 × Hi-SPI
- 2 × I<sup>2</sup>S
- 2 × I<sup>2</sup>C
- 3 × UART
- 1 host (SD/eMMC/SDIO)
- 1 slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- CAN BUS 2.0





- **Flash encryption**
  - 1024-bit OTP, up to 768-bit for customers
  - Cryptographic hardware acceleration
- **On-board Sensors and Interfaces**
  - Motor PWM
  - LED PWM up to 16 channels
  - Hall sensor
  - Temperature and Humidity Sensor
  - Proximity Sensor o Dual Channel Relay's
  - 4 Touch Sensor's
  - LDR
  - Buzzer
  - DHT Sensor
  - Infrared Remote
  - Thermal sensor
  - RGB LED



Multiple Sensors & interfaces

## Development

- [Arduino IDE \(C++\)](#)
- [Zephyr Project](#)

A scalable real-time operating system (RTOS) supporting multiple hardware architectures, optimized for resource constrained devices, and built with security in mind.
- [Mongoose OS](#)
- [Amazon AWS IoT](#)
  - [Internet button & Amazon AWS IoT](#)
  - [Secure remote device management](#)
- [Micro Python](#)
- [Lua Script](#)
- [Lua Node](#)

Programming with the Lua programming language directly or using a block-based programming language that translates blocks to Lua.
- [Espruino \(java script based firmware\)](#)

etc.....



## Functional Description

### USB-to-UART Bridge

A single chip USB-to-UART bridge provides up to 3 Mbps transfers rates.

### Boot Button

Download button - holding down the Boot button and pressing the EN button initiates the firmware download mode. Then user can download firmware through the serial port. EN Button

Reset button - pressing this button resets the system.

### Micro USB Port

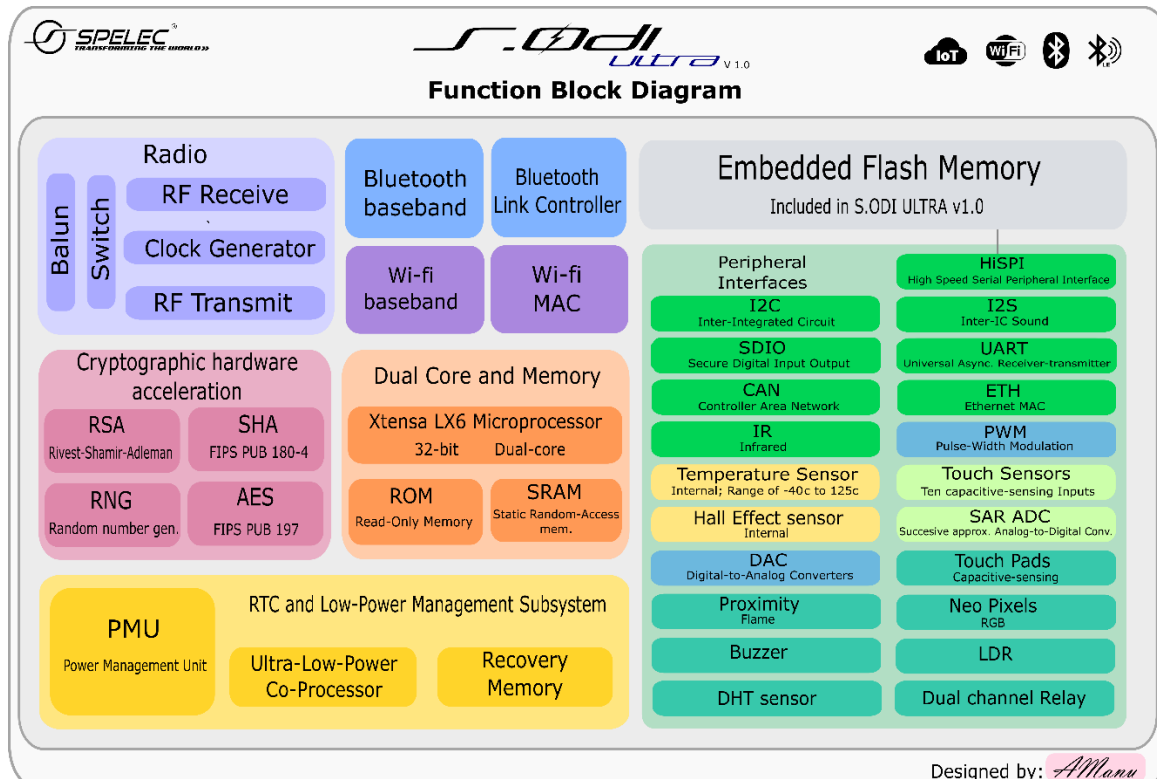
USB interface- It functions as the power supply for the board and the communication interface between PC and the ESP module. Power on LED -Turns on when the power supply is applied to the board.

### I/O Connector

Most of the pins on the ESP module are broken out to the female pin headers on the board. Users can program ESP32 to enable multiple functions such as **PWM, ADC, DAC, I2C, I2S, SPI**, etc.

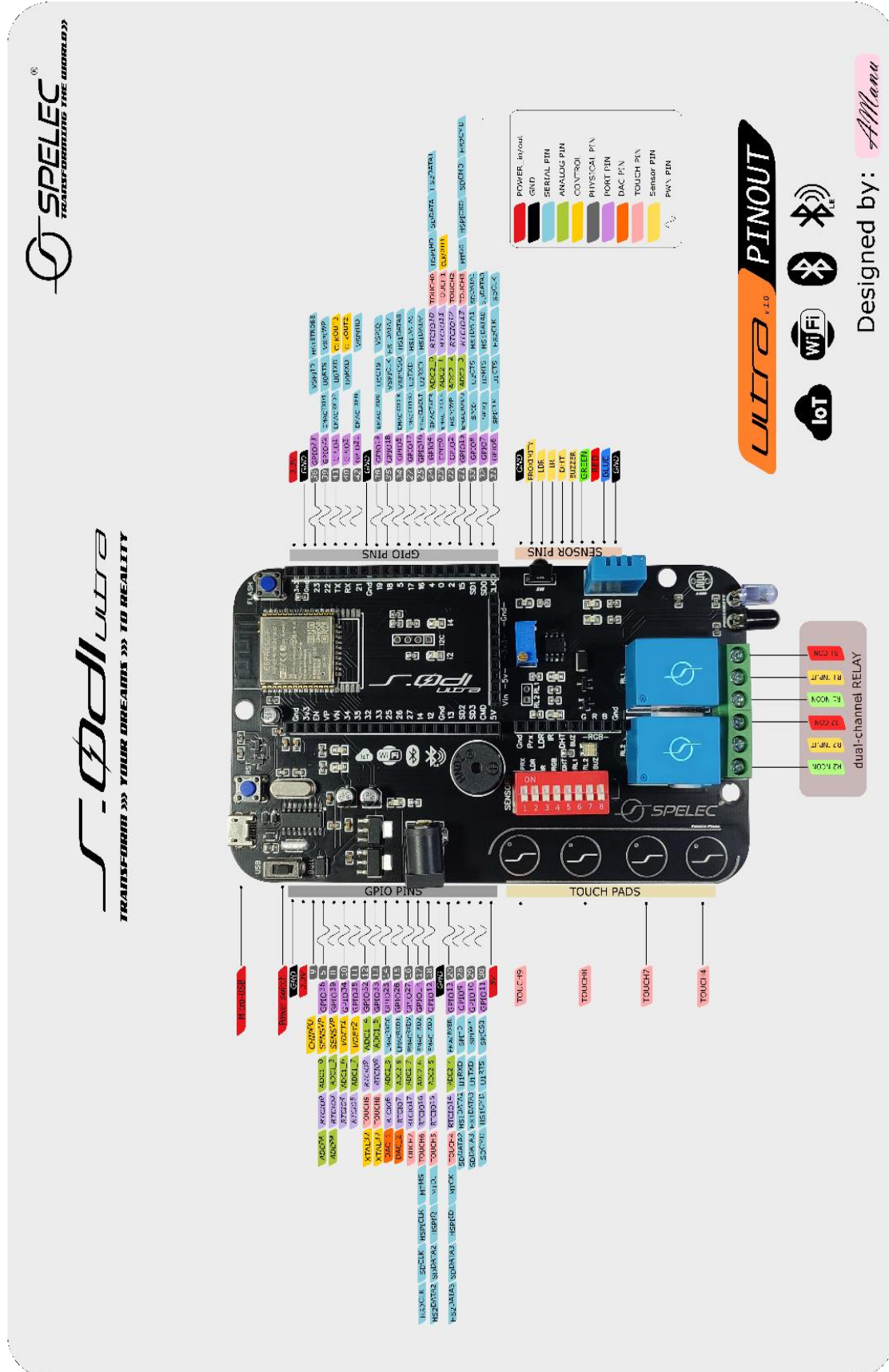
## Board Architecture

### Function Block Diagram of S.ODI ultra





## Pin Description





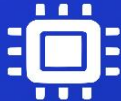
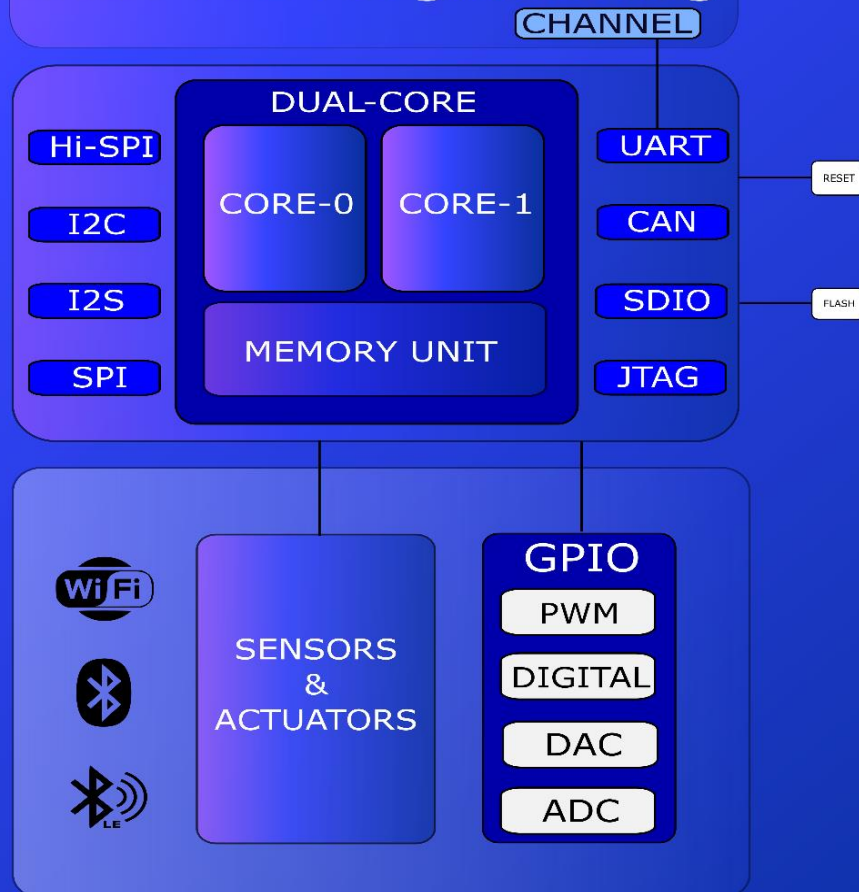
# Program Flow

Plug

Program

Play

## USB-TTL-Programming



GPIO	Input	Output	Multiplexed	Notes
0	pulled up	OK	ADC2_1, Touch 1, RTC_11	outputs PWM signal at boot RTC GPIO is routed to the RTC Low-Power Subsystem.
1	TX pin	OK		debug output at boot
2	OK	OK	ADC2_2, Touch 2, RTC_12	connected to the on-board LED 1 RTC Low-Power Subsystem.
3	OK	RX pin		HIGH at boot
4	OK	OK	ADC2_0, Touch 0, RTC_10	connected to the on-board LED 2 RTC Low-Power Subsystem.
5	OK	OK	VSPI CS0	outputs PWM signal at boot
6	X	X	SCK/CLK	connected to the integrated SPI flash
7	X	X	SDO/SD0	connected to the integrated SPI flash
8	X	X	SDI/SD1	connected to the integrated SPI flash
9	X	X	SHD/SD2	connected to the integrated SPI flash
10	X	X	SWP/SD3	connected to the integrated SPI flash
11	X	X	CSC/CMD	connected to the integrated SPI flash
12	OK	OK	HSPI MISO, RTC_15, ADC2_5	boot fail if pulled high. RTC Low-Power Subsystem.
13	OK	OK	Touchpad 4, HSPI, RTC_14, ADC2_4	Touch 4, RTC Low-Power Subsystem.
14	OK	OK	HSPI CLK, RTC_16, ADC2_6	outputs PWM signal at boot, SPI CLK RTC Low-Power Subsystem.
15	OK	OK	ADC, Touch 3, HSPI CS, RTC_13, ADC2_3	outputs PWM signal at boot RTC Low-Power Subsystem.
16	OK	OK	HSPI Data4	U2 Rx
17	OK	OK		U2 Tx
18	OK	OK		VSPI CLK
19	OK	OK		VSPI MISO
21	OK	OK		SDA of I2C
22	OK	OK		SCL OF I2C
23	OK	OK		VSPI MOSI

automations

25	OK	OK	RTC_6, ADC2_8, DAC_1	DAC(Digital to Analog Converter) RTC Low-Power Subsystem.
26	OK	OK	RTC_8, ADC2_9, DAC_1	DAC(Digital to Analog Converter) RTC Low-Power Subsystem.
27	OK	OK	TouchPad 7, RTC_17, ADC2_7	TouchPad 7 RTC Low-Power Subsystem.
32	OK	OK	TouchPad 9, RTC_9, ADC1_4	TouchPad 9 RTC Low-Power Subsystem.
33	OK	OK	TouchPad 8, RTC_8, ADC1_5	TouchPad 8 RTC Low-Power Subsystem.
34	OK		RTC_4, ADC1_6	Input only RTC Low-Power Subsystem.
35	OK		RTC_5, ADC1_7	Input only RTC Low-Power Subsystem.
36	OK		RTC_0, ADC1_0	Input only RTC Low-Power Subsystem.
37	OK		ADC1_1	ADC input channels
38	OK		ADC1_2	ADC input channels
39	OK		RTC_3	Input only RTC Low-Power Subsystem.
Prx		OK	Sensor pin	output only Proximity Sensor
LDR		OK	Sensor pin	output only LDR (Light Dependent Resistor) used in light varying sensor circuits.
IR	OK	OK		IR(Infrared) communication
DHT		OK	Sensor pin	output only DHT 11 sensor
BUZ	OK		Actuators pin	Buzzer for audio Signalling devices, alarms.
RGB	OK		RGB pin	Neo Pixel PWM and Digital supported.
And	OK			Power Grounding
R1	OK		Actuator pin	input only Relay1 Control Unit
R2	OK		Actuator pin	input only Relay2 Control Unit
USB			Power switch-USB power	USB Power.
Batt	DC Jack		DC-input	switch to battery mode.




## Key Components

Key Components	Description
USB-TTL CH340G	CH340 is a USB bus converter chip and it can realize USB converter to serial interface, USB converter to IrDA infrared or USB converter to printer interface. In serial interface mode, CH340 supplies common MODEM liaison signal, used to enlarge asynchronous serial interface of a computer or upgrade the common serial device to USB bus directly Defragmentation
12MHz Program OSC	To synchronize the programming serial data to be written into the processor
PWR SWITCH	To switch ON/OFF the MCU with USB power or external power source
DUAL CORE PROCESSOR	This is a 32bit dual core processor featuring 16 MB of Memory and 512KB RAM with 240MHz clock which runs up to 600 MIPS
Antenna FOR Wi-Fi, Bluetooth, BLE	The 2.4 GHz transmitter modulates the quadrature baseband signals to the 2.4 GHz RF signal and drives the antenna with a high-powered Complementary Metal Oxide Semiconductor (CMOS) power amplifier. It delivers up to +20.5 dBm of power for an 802.11b transmission and +18 dBm for an 802.11n transmission.
HISPD-SPI	ESP32 features three SPIs (SPI, HSPI and VSPI) in slave and master modes in 1-line full-duplex and 1/2/4-line half-duplex communication modes. Four modes of SPI transfer format, which depend on the polarity (CPOL) and the phase (CPHA) of the SPI clock up to 80 MHz to 64-byte FIFO. All SPIs can also be connected to the external flash/SRAM and LCD. Each SPI can be served by DMA controllers.
2x On-board PWM LED	These are 2 pulse width modulation enabled LED's connected to GPIO-2 and GPIO-4
DC external Power	To give external power from 9V to 5V
Buzzer	On-board buzzer for applications
Sensor Switch	To switch ON/OFF the sensors on board

Proximity & Flame	IT is used to detect the presence of nearby objects without any physical contact.
IR Sensor	Used to transmit and receive infrared rays for IR remote application and wireless IR transmissions
4+6 Touch Sensor	These are capacitive touch sensors on board for T4, T7, T8, T9(13,27,33,32) and 6 touch sense enabled pins
LDR Sensor	Light Dependent resistor sensor is used to detect light
DHT-Sensor	Digital Humidity and Temperature Sensor is used to find the humidity and temperature of the surroundings
Dual-channel Relay	Dual relay unit for switching high voltage devices for IoT applications
Sensor threshold adjusters	To adjust the threshold of the digital sensors for various applications
Hall & CPU temp Sensor	Integrated with a Hall sensor based on a N carrier resistor. When the chip is in the magnetic field, the Hall sensor develops a small voltage laterally on the resistor, which can be directly measured by the ADC.
SD-card interface	External SD-card interfacing GPIO's for Memory Expansion and Data applications.
USB interface	Universal Serial Bus type-B for serial data programming and serial monitoring providing read and write applications for the processor.
GPIO's	<p>All the pins on the S.ODI are broken out to pin headers. You can program S.ODI to enable multiple functions, such as PWM, ADC,</p> <hr/> <p>DAC, I2C, I2S, SPI, etc</p>



## Setup Toolchain

		
<u>Windows</u>	<u>Linux</u>	<u>Mac OS</u>

### For Programmer CH340G Drivers

Download and Install S.ODI Programmer Drivers

The CH340 chip is used by several Arduino compatible boards to provide USB connectivity, you may need to install a driver, don't panic, it's much easier than it sounds all you need to do is just click on the link provided below and install it and follow the instructions provided below.

Get the drivers here: [S.ODI ultra-programmer drivers](#)

### Arduino IDE

Using Arduino IDE with the development repository

- [Instructions for Windows](#)
- [Instructions for Mac](#)
- [Instructions for Debian/Ubuntu Linux](#)

#### Prerequisites: Arduino IDE Installed

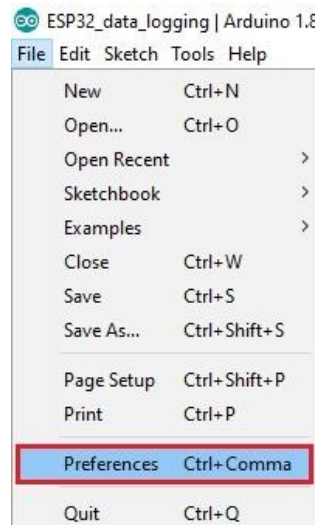
Before starting this installation procedure, make sure you have the latest version of Arduino IDE installed in your computer. If you don't, uninstall it and install it again. Otherwise, it may not work.

Having the latest Arduino IDE software installed from [arduino.cc/en/Main/Software](http://arduino.cc/en/Main/Software), continue with this tutorial.

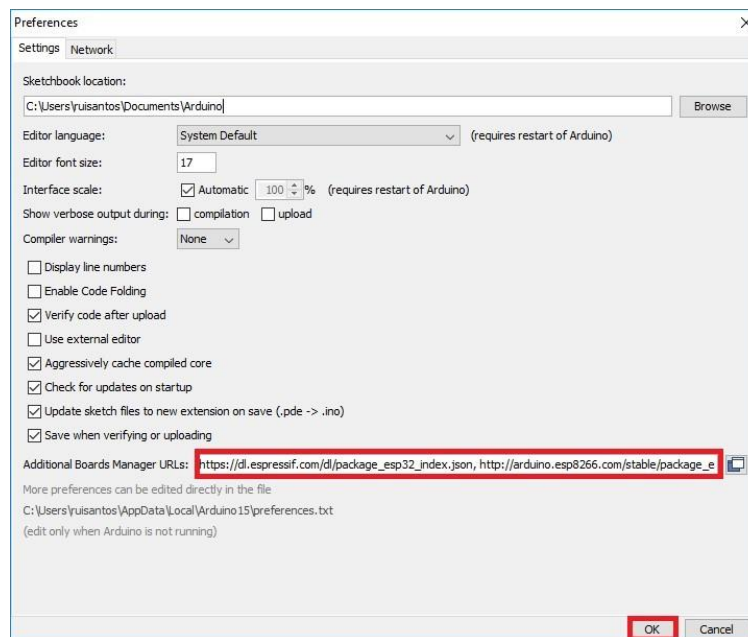
## Installing step by step S.ODI ESP32 Add-on in Arduino IDE

To install the S.ODI Dev board in your Arduino IDE, follow these next instructions:

1. In your Arduino IDE, go to File> Preferences



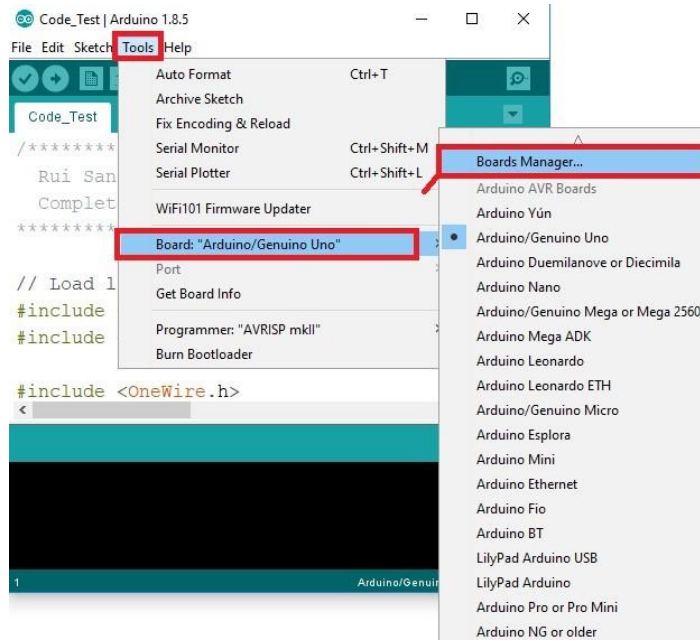
2. Enter [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) into the “Additional Board Manager URLs” field as shown in the figure below. Then, click the “OK” button:



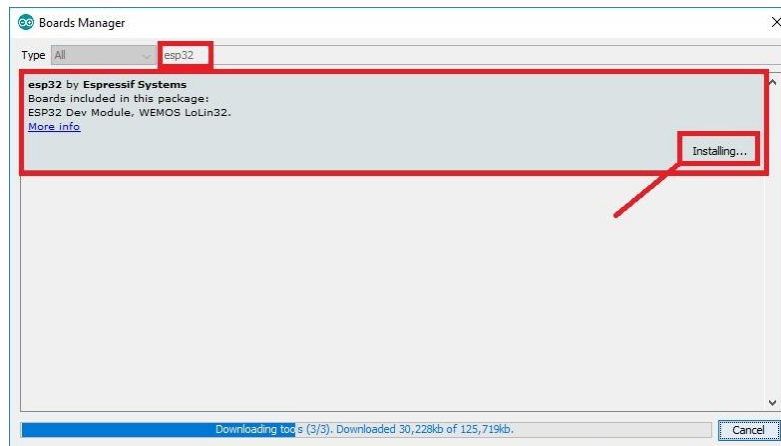
Note: if you already have the ESP8266 boards URL, you can separate the URLs with a comma as follows:

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

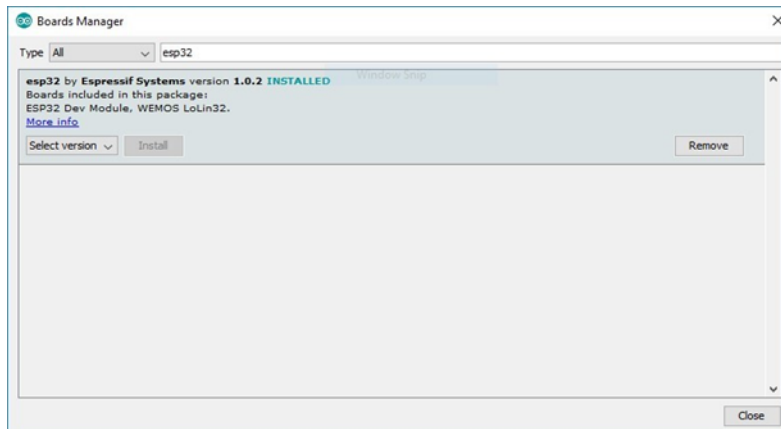
3. Open the Boards Manager. Go to Tools > Board > Boards Manager...



4. Search for ESP32 and press install button for the "ESP32 by Espressif Systems":



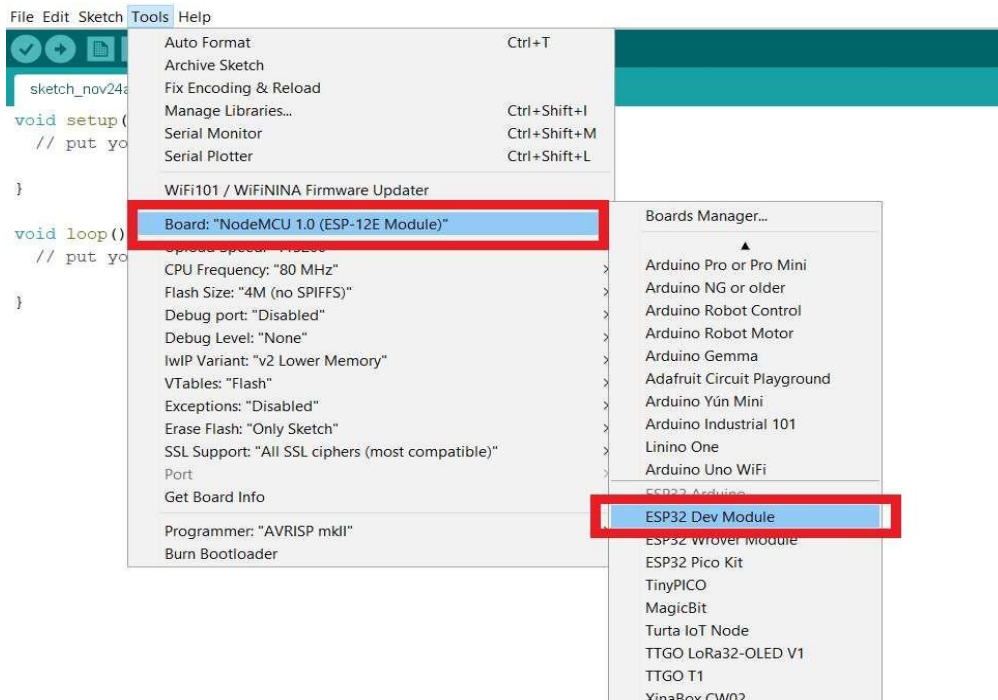
5. That's it. It should be installed after a few seconds.



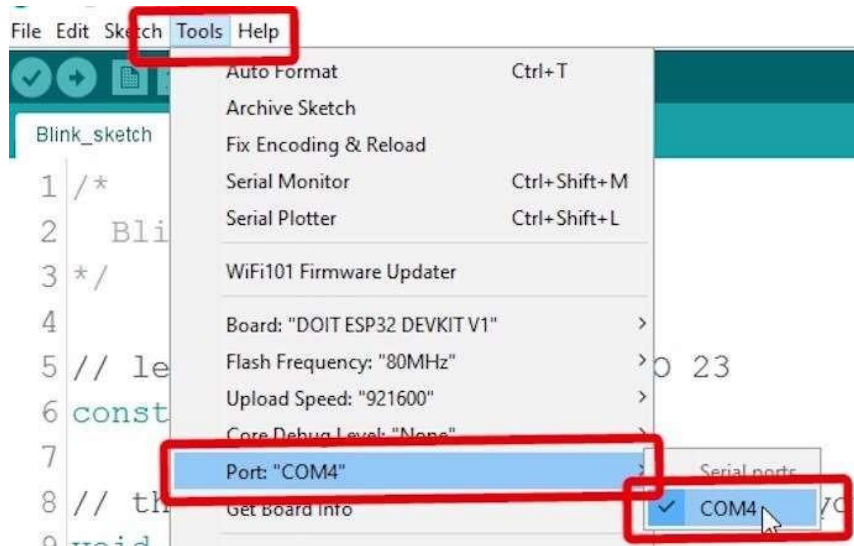
## Start a Project

Plug the S.ODI Dev board to your computer. With your Arduino IDE open, follow these steps:

1. Select your Board in Tools > Board menu (in my case it's the)

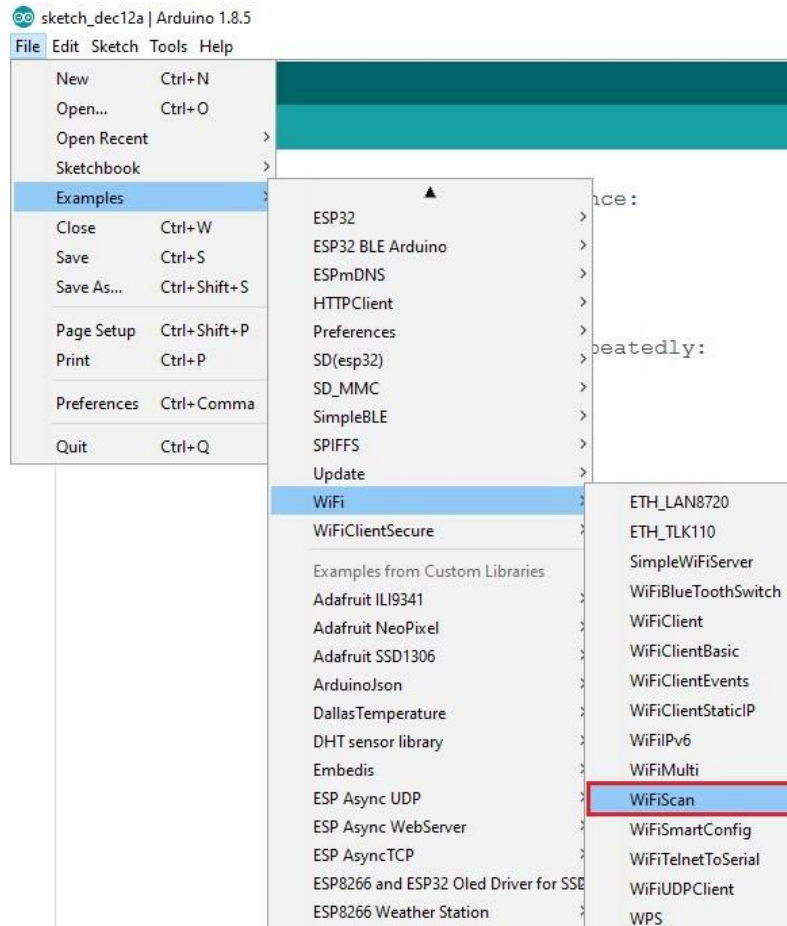


2. Select the Port (if you don't see the COM Port in your Arduino IDE, you need to install the [CH340G USB to UART - TTL DRIVER](#)):

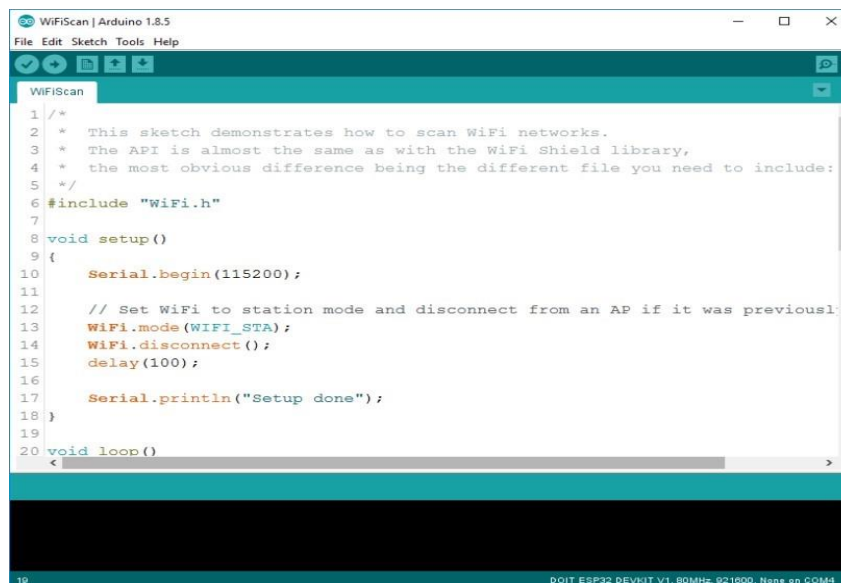


automations

3. Open the following example under File > Examples > WIFI (ESP32) > WIFI Scan



4. A new sketch opens in your Arduino IDE:

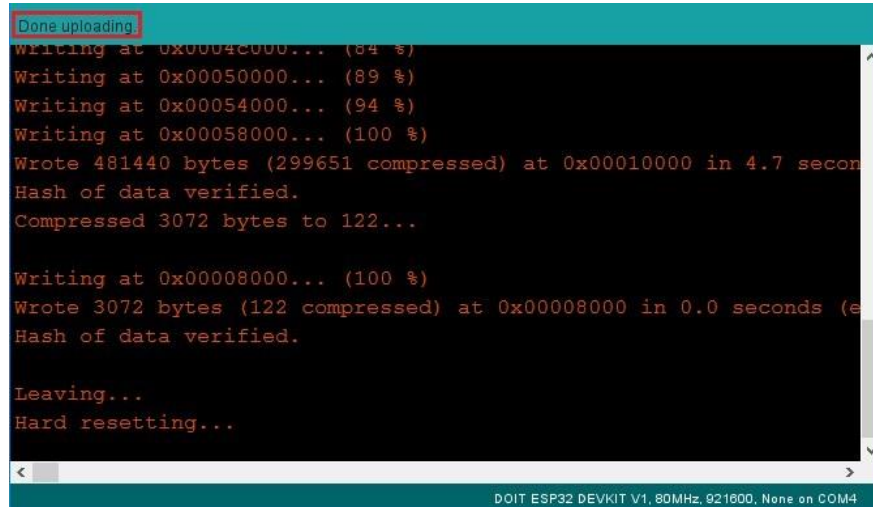


automations

5. Press the Upload button in the Arduino IDE. Wait a few seconds while the code compiles and uploads to your board.



6. If everything went as expected, you should see a "Done uploading." message.

A screenshot of the Arduino IDE Serial Monitor window. The text is as follows:

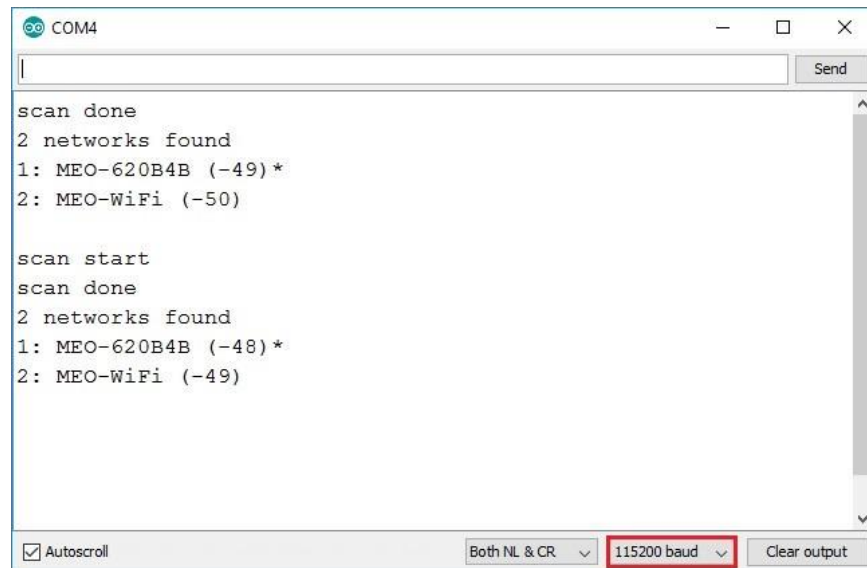
```
Done uploading.  
Writing at 0x0004c000... (84 %)  
Writing at 0x00050000... (89 %)  
Writing at 0x00054000... (94 %)  
Writing at 0x00058000... (100 %)  
Wrote 481440 bytes (299651 compressed) at 0x00010000 in 4.7 seconds  
Hash of data verified.  
Compressed 3072 bytes to 122...  
  
Writing at 0x00008000... (100 %)  
Wrote 3072 bytes (122 compressed) at 0x00008000 in 0.0 seconds (effective 115200 bps)  
Hash of data verified.  
  
Leaving...  
Hard resetting...
```

The status bar at the bottom indicates 'DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on COM4'. The 'Done uploading.' message is highlighted with a red box.

7. Open the Arduino IDE Serial Monitor at a baud rate of 115200:



8. Press the S.ODI on-board Enable button and you should see the networks available near your S.ODI :

A screenshot of the Arduino IDE Serial Monitor window. The text is as follows:

```
COM4  
scan done  
2 networks found  
1: MEO-620B4B (-49)*  
2: MEO-WiFi (-50)  
  
scan start  
scan done  
2 networks found  
1: MEO-620B4B (-48)*  
2: MEO-WiFi (-49)
```

The '115200 baud' dropdown menu is highlighted with a red box. The status bar at the bottom shows 'Autoscroll' checked, 'Both NL & CR' selected, '115200 baud' selected, and 'Clear output' button.



## Libraries & Frameworks

Arduino-esp32 includes libraries for Arduino compatibility along with some object wrappers around hardware specific devices. Examples are included in the examples folder under each library folder. The ESP32 includes additional examples which need no special drivers.

<https://github.com/espressif/arduino-esp32/tree/master/libraries>

**Arduino OTA:** Over the Air firmware update daemon.

Use espota.py to upload to the device.

**Async UDP:** Asynchronous task driven UDP datagram client/server

**Azure IoT:** Library to interact with Microsoft Azure IoT services

**ESPmDNS:** mDNS service advertising

**FFat:** FAT indexed filesystem on SPI flash

**FS:** Filesystem virtualization framework

**HTTP Client:** A simple HTTP client, compatible with WiFiClientSecure

**HTTP Update:** Download a firmware update from HTTP and apply it using Update

**SPI:** Arduino compatible Serial Peripheral Interface driver (master only)

**SPIFFS:** SPI Flash Filesystem

(see [spiffs-plugin](#) to upload to device)

**Ticker:** A timer to call functions on an interval  
Update Sketch Update using ESP32 OTA  
Functionality.

**BLE:** Bluetooth Low Energy v4.2 client/server framework

**Bluetooth Serial:** Serial to Bluetooth redirection server

**DNS Server:** A basic UDP DNS daemon (includes captive portal demo)

**EEPROM:** Arduino compatibility for EEPROM (using flash)

**NetBIOS:** NetBIOS name advertiser

**SD input:** Secure Digital card filesystem using SPI access

**SD\_MMC:** Secure Digital card filesystem using 4-lane access

**Simple BLE:** Minimal BLE advertiser

**Webserver:** A simple HTTP daemon  
Wi-Fi Arduino compatible Wi-Fi driver (includes Ethernet driver)

**Wi-Fi Client Secure:** Arduino compatible Wi-Fi client object using embedded encryption

**Wire:** Arduino compatible I2C driver (master

### ESP32 Additional examples

- Analog Out
- Camera
- Chip ID
- Deep Sleep
- ESP Now
- Free RTOS
- GPIO
- Hall Sensor
- I2S
- Reset Reason
- RMT
- Time
- Timer
- Touch

## ESP-IDF installation guide

Related Documents

Go to this link to get support: [ESP-IDF installation](#)

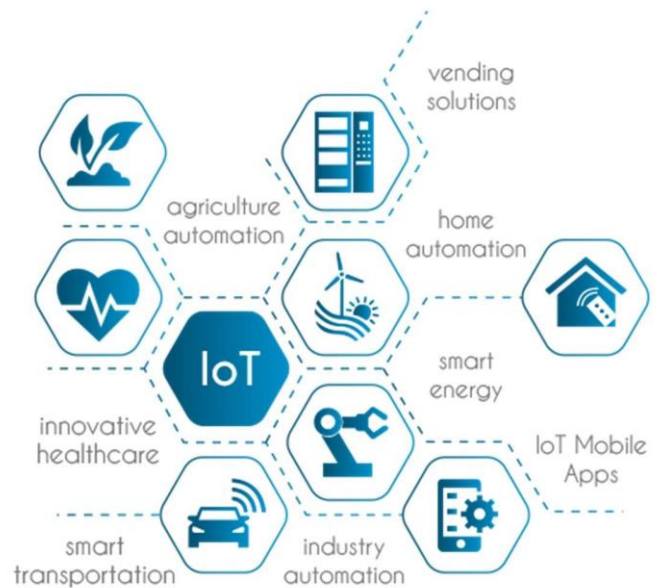
## Micro Python Installation Guide

Related Documents

Go to this link to get support: [Micro Python Installation](#)

## Applications

- **Generic Low-power IoT Sensor Hub**
- **Generic Low-power IoT Data Loggers**
- **Cameras for Video Streaming**
- **Over-the-top (OTT) Devices**
- **Speech Recognition**
- **Image Recognition**
- **Mesh Network**
- **Home Automation**
  - Light control
  - Smart plugs
  - Smart door locks
- **Smart Building**
  - Smart lighting Energy monitoring
  - Smart cities
  - Smart Grids
- **Industrial Automation**
  - Industrial wireless control
  - Industrial robotics
- **Smart Agriculture**
- **Wi-Fi-enabled Toys**
  - Smart greenhouses
  - Remote control toys
  - Smart irrigation
  - Proximity sensing toys
  - Agriculture robotics
  - Educational toys
- **Audio Applications**



- **Wearable Electronics**
  - Internet music players
  - Smart watches
  - Live streaming devices
  - Smart bracelets
  - Internet radio players
  - Audio headsets
- **Service robots**
- **Health Care Applications**
  - Cleaning Robots
  - Health monitoring
  - Autonomous cars
  - Baby monitors
  - Automated Toll Gates

## Product Comparisons

Based on trending boards available

Specs & Features	Arduino	Node-MCU	STM32 <sub>(F103)</sub>	S.ODI ultra
processor cores	Single	Single	Single	Dual
Architecture	AVR RISC (8-Bit)	ESP8266(LX106) (32-Bit)	Arm Cortex-M3 (32-Bit)	Xtensa LX6 (32-Bit)
CPU frequency	16MHz	80MHz	72MHz	240MHz
Speed (1000000 operations)	7.11sec	3.82sec		0.39sec
Wi-Fi	No	Yes	No	Yes
Bluetooth	No	No	No	Yes (v4.2)
BLE	No	No	No	Yes
Flash	32KB	4MB	64KB	16MB
EEPROM	1KB	4 to 4096 Bytes	NA	4KB
SRAM	2KB	128KB	20KB	520KB
GPIO	14 (10-Bit)	17 (10-Bit)	32 (12-Bit)	38 (12-Bit)
Busses	SPI, I2C, UART	SPI, I2C, UART, I2S	SPI, 2-I2C, 3-UART, CAN, USB	High-Speed SPI, 2-I2C, 2-I2S, 3-UART, SDIO, CAN

automations

<b>Proximity Sensor</b>	No	No	No	<b>Yes</b>
<b>LDR Sensor</b>	No	No	No	<b>Yes</b>
<b>IR Comm</b>	No	No	No	<b>Yes</b>
<b>DHT Sensor</b>	No	No	No	<b>Yes</b>
<b>Buzzer</b>	No	No	No	<b>Yes</b>
<b>Neo Pixel (RGB)</b>	No	No	No	<b>Yes</b>
<b>Relays (R1 &amp; R2)</b>	No	No	No	<b>Yes</b>
<b>PWM</b>	6	10	20	<b>32</b>
<b>Input voltage</b>	5v-9v	3.3v - 5v	2v – 3.6v	<b>5v - 25v</b>