



Cryptocurrency Price Ticker

status **active**

Cryptocurrency Price Ticker

Table of Contents

- [About](#)
- [Getting Started](#)
- [Circuit](#)
- [Usage](#)
- [List Of Components](#)
- [Built Using](#)
- [Authors](#)

About

This repo contains

- Backend
- Firmware
- Detailed instructions

for Cryptocurrency Price Ticker.

Getting Started

These instructions will get you a copy of the project up and running on your system.

Prerequisites

Things you need to install the FW.

- Arduino IDE

Installing

A step by step series that tell you how to get the Firmware and Backend running

ESP32 Configuration

You should have Arduino IDE Installed

1. Add ESP32 Board to your Arduino IDE
2. In your Arduino IDE, go to File> Preferences Installing ESP32 Add-on in Arduino IDE Windows, Mac OS X, Linux open preferences
3. Enter https://dl.espressif.com/dl/package_esp32_index.json into the "Additional Board Manager URLs" field then, click the "OK" button: Note: if you already have the ESP32 boards URL, you can separate the URLs with a comma(each board will go to new line) as follows:
https://dl.espressif.com/dl/package_esp32_index.json,\n
http://arduino.esp8266.com/stable/package_esp8266com_index.json
4. Open the Boards Manager. Go to Tools > Board > Boards Manager...
5. Search for ESP32 and press install button for the ESP32 by Espressif Systems":
6. That's it. It should be installed after a few seconds.
7. In your Arduino sketchbook directory, create tools directory if it doesn't exist yet.
8. Unpack the tool into tools directory(present in libs/ESP32FS-1.0.zip) (the path will look like <home_dir>/Arduino/tools/ESP32FS/tool/esp32fs.jar).
9. Close and re-open the Arduino IDE.
10. Now copy the contents of the libs folder to the libraries directory of your Arduino
 1. If you are using windows, the libraries directory will be Documents/Arduino/libraries

ESP32 Node FW Uploading

1. Select ESP32 Dev Module from Tools->Board->ESP32
2. Select the correct port from Tools->Port
3. Then open Firmware.ino file,
4. Select Tools > ESP32 Sketch Data Upload menu item. This should start uploading the files into ESP32 flash file system.
5. Now Upload the Code to your ESP32 Dev Module.
6. Your ESP32 is now ready to be used.

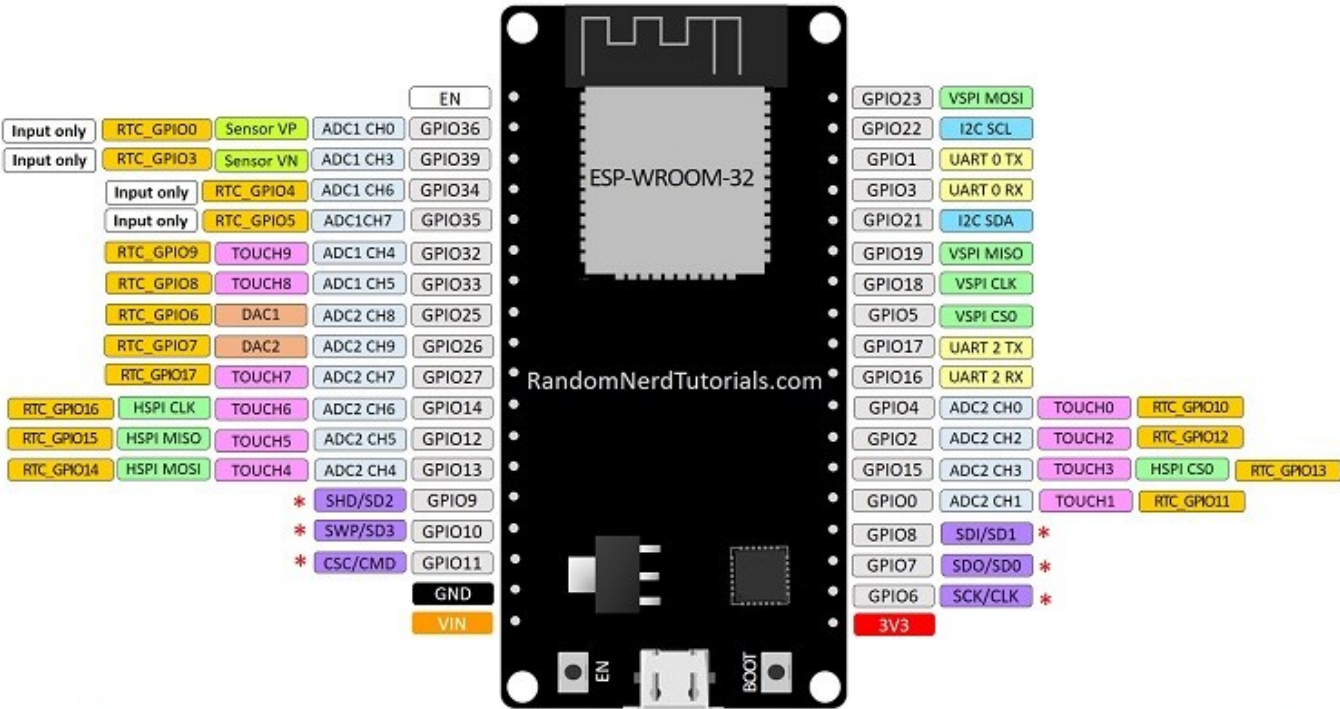
Circuit

ESP32 Dev Module Pinout

Follow the pinout diagram given below to connect different components to your ESP32 Dev Module board.

ESP32 DEVKIT V1 – DOIT

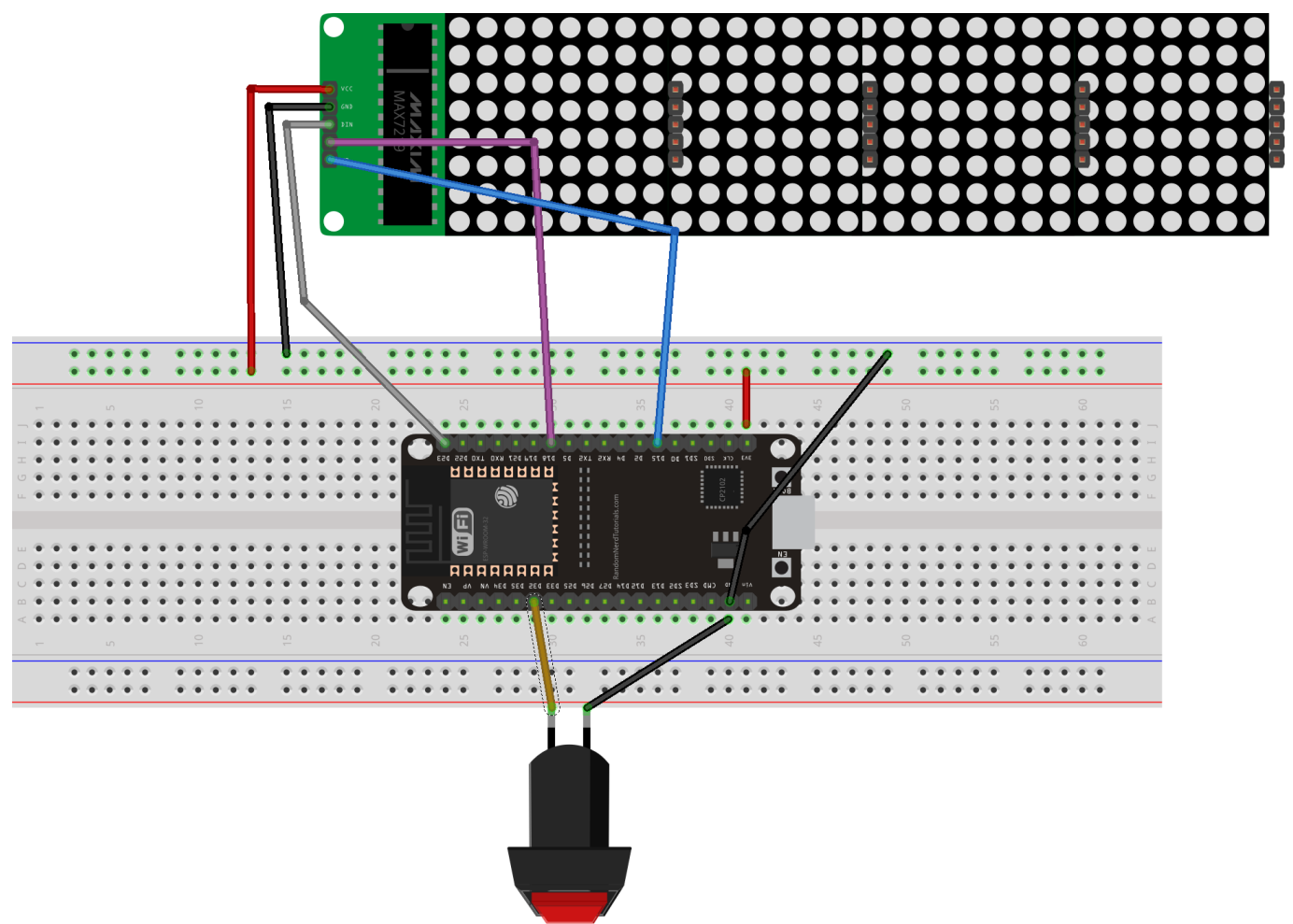
version with 36 GPIOs



* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

Complete Circuit Diagram

Here's the complete circuit diagram of the system.



fritzing

Other Components

Other components pin connection details

MAX7219 Dot Matrix Display

MAX7219 Dot Matrix Display Connected with ESP32

MAX7219 Pins	ESP32
3.3V	3.3V
CLK	D18
DIN	D23
CS	D15
GND	GND

Display On/Off Button

Display On/Off Button Connected with ESP32

MAX7219 Pins ESP32

PIN1 D32

PIN2 GND

Usage

1. Upload the code to your ESP32.
2. Power on your ESP32, it will present you with an AP named **CPT-abc** (while **CPT** can be changed in the portal and **abc** is a unique id for each esp32)
3. Default captive portal password **123456789AP** which can be changed in captive portal.
4. Connect to the ESP32 access point and open the web-browser and navigate to the link **http://esp32.local/_ac**. This link will work on most of the operating systems but if your operating system is not allowing to open it, you may want to check the captive portal IP Address from the serial monitor and can use that IP address inplace of the above mentioned URL.
5. The default access IP Address is **http://192.168.4.1/_ac**
6. You will be presented with a main dashboard as shown below(based on your device)

Cryptocurrency Price Ticker		Connect to WiFi	Saved WiFi Networks	Disconnect	Reset...	Settings	HOME
Established connection		hotspot2					
Mode		AP_STA(3)					
IP		192.168.115.49					
GW		192.168.115.181					
Subnet mask		255.255.255.0					
SoftAP IP		192.168.4.1					
AP MAC		24:0A:C4:AF:D0:C1					
STA MAC		24:0A:C4:AF:D0:C0					
Channel		6					
dBm		-40					
Chip ID		49360					
CPU Freq.		240MHz					
Flash size		4194304					
Free memory		244288					

7. Once connected to a WiFi network, you can again access the captive portal using same URL or the IP Address from the Serial monitor.
8. The data is published to the MQTT Topic **CPT/{hostname}** while the hostname is the one which you can define in Settings page of the captive portal.
9. You can open settings page with following default credentials
 1. User: **AP Name (CPT)**
 2. Password: **admin**
10. You can set a crypto currency in the setting page in 'Crypto Name' Section.

List of Components

Following components are used to make this project

1. ESP32 Dev Module ○ https://www.amazon.com/ESP32-WROOM-32-Development-ESP-32S-Bluetooth-Arduino/dp/B084KWNMM4/ref=sr_1_3?dchild=1&keywords=esp32+devkit&qid=1635342032&sr=8-3
2. LED Matrix 4x (8x8 Matrix) ○ https://www.amazon.com/HiLetgo-MAX7219-Arduino-Microcontroller-Display/dp/B07FFV537V/ref=sr_1_12?dchild=1&keywords=led+matrix&qid=1635341741&sr=8-12
3. On/OFF Switch ○ https://www.amazon.com/URWOOW-Latching-Button-Household-Appliances/dp/B08F7ZR4VW/ref=sr_1_10_sspa?dchild=1&keywords=on+off+button&qid=1610176872&sr=8-10-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzREIYREIPQU1lQkFXJmVuY3J5cHRIZElkPUeWOTk2NzUwMVIWTKJPQ1ozWDdSVyZlbnNyeXB0ZWRBZEIkPUeWNTg4Nzg1MUhTWTTNNSEcyRIZLViZ3aWRnZXROYW1lPjNwX210ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=

Built Using

- [Arduino](#) - Embedded Framework and IDE - For Sensor Node Design

Authors

- [@Nauman3S](#) - Development and Deployment