

Mounting instructions Raduino32 v6.1

The Raduino32 board version 6.1 is designed to easily replace the original Raduino board with Arduino Nano from uBitx v6. It supports the use of an ESP32 microcontroller with the advantages in terms of memory size and other functions, preserving the 2.8 "TFT touch screen.

THE ESP32 MODULE, NOR THE ADS1115 IS NOT INCLUDED!

For the basic operation, it is not necessary to modify the main board of the uBitx, only if you want to use the additional functions, you have to carry out some modifications, which are otherwise simple..

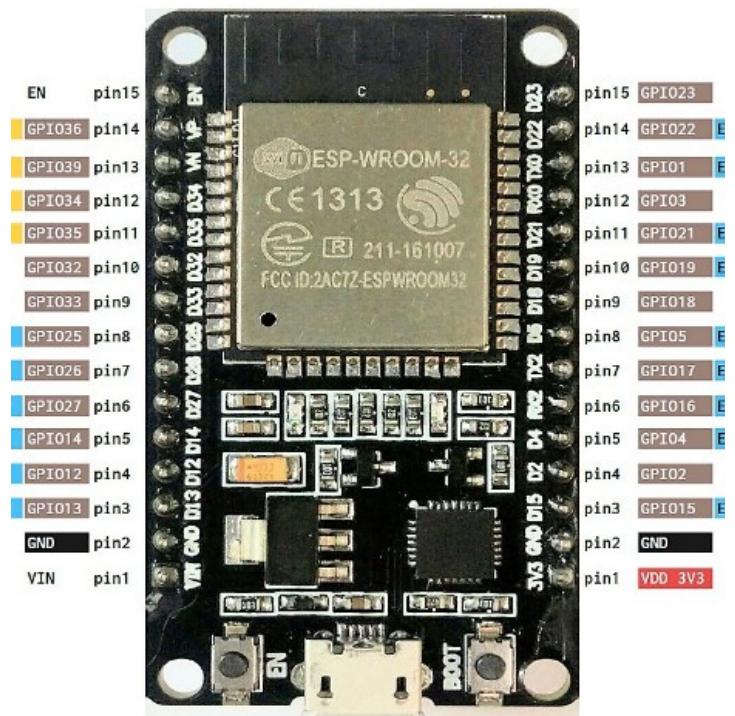
This version of the Raduino32 includes the following main components from the factory:

- IC Si5351, clock signal generator for oscillator (U1)
- Reg. LM7805, to power the circuits at 5v (U2)
- IC ACS712, Hall effect sensor for current measurement (U3)
- IC LM386, sensor for the S-meter (U4)
- P-channel MOSFET, for protection against polarity reversal (Q1)

- Connectors to connect the Raduino32 to the main board of the uBitx (H8, H9)
- Pins to insert the 2.8 "TFT touch screen (H1) socket.
- Pins to connect the Encoder (CN1) socket.
- Socket to insert the 2x15-pin ESP32 DoIt module (H3, H5)
- Sockets for inserting two ADS1115 ADC modules (H2, H4). Optional if you want to measure:
 - S-Meter signal
 - VTOT, ITOT, Supply voltage and current + 12v.
 - VForw, vRefl, SWR sensor signals.
- Socket to insert a MCP23017 (H10) module, a 16 I / O multiplexer controlled by I2C. Not used yet, it would allow to control other signals: antenna tuner, power amplifier, etc.
- Socket to insert a PCA9685 (H12) module, with 16 PWM channels. Only used experimentally to control servo motors of an antenna tuner.
- Socket for connecting a board to send and receive audio via TCP / UDP (H11)
- Pins for extension of audio signals (CN3).
- Sockets for serial ports 1 and 2 (H13, H14)

- Connector for 3 analog inputs (H7)
- SWR sensor signal connector (CN2)
- DS18B20 temperature probe connector (CN4)
- Connector for +12 volt input, + 12VI (CN5)
- Connector for +12 volt output, + 12VO (CN6)
- Pins to bypass the + 12V circuit (H15)

For basic operation it is necessary to add an ESP32 DoIt module or equivalent. The module must have 30 pins (2 x 15) and an arrangement according to the image:



MINIMUM ASSEMBLY

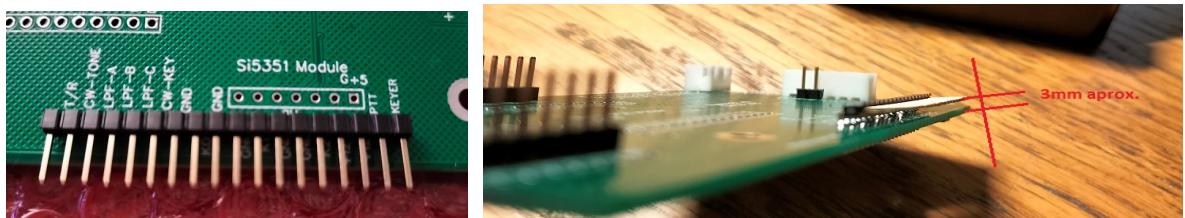
ATTENTION! SMD components are factory welded, be very careful when handling the board to avoid damaging them.

This section describes the assembly of the components necessary for operation with the basic functions.

The order of assembly is only indicative and based on my own experience.

WELDING

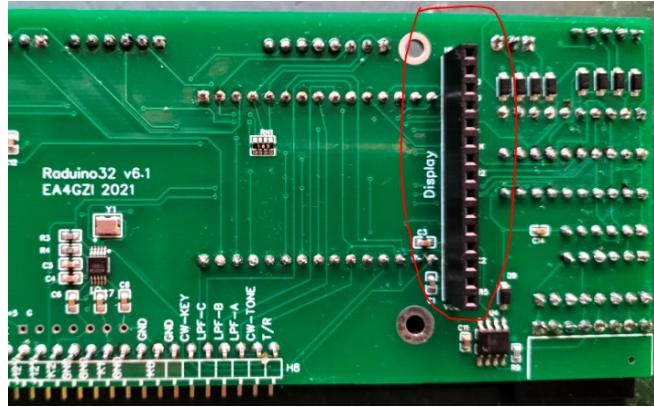
1. Solder the male strip H8, 18 pins and 90°, that connects the Raduino32 to the main board of the uBitx. **ATTENTION! This header pins is soldered on the BOTTOM side, even though it is drawn on the upper face on the PCB.**



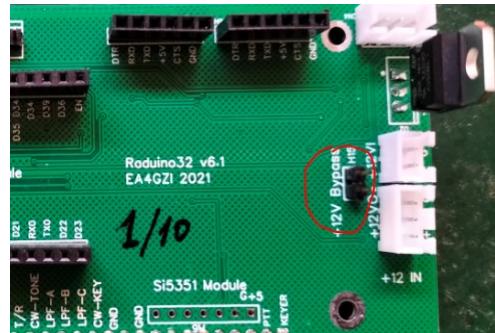
2. Solder the female socket H9, 8 pins and 90°, that connects the Raduino32 to the audio signals of the uBitx. **ATTENTION! This header pins is soldered on the BOTTOM side, even though it is drawn on the upper face on the PCB.**



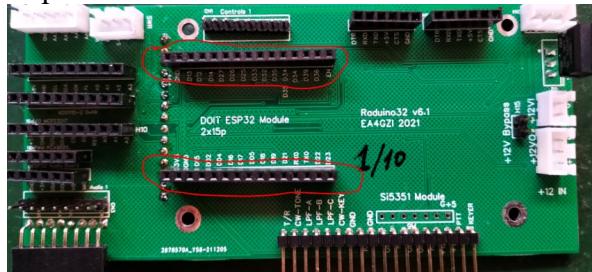
3. Solder the 15-pin female socket that is inserted into the TFT screen. **ATTENTION! This header pins is soldered on the TOP side.**



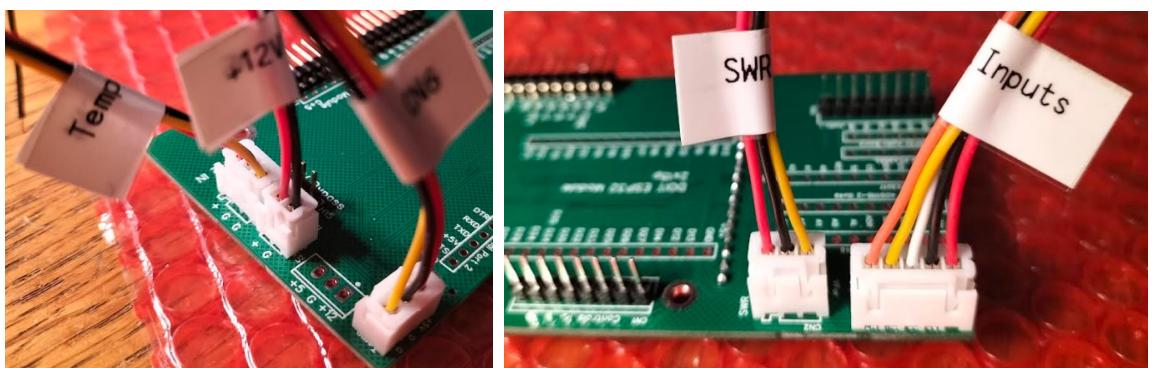
4. Before continuing, it is convenient to verify that the PCB can be correctly inserted in the uBitx main board and in the TFT.
5. Solder the 2-pin male header H15.



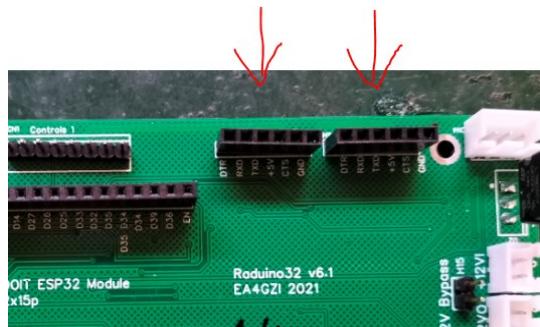
6. Solder the two 1 x 15-pin female sockets to insert the ESP32 module.



7. Solder the connectors H7, CN2, CN4, CN5 and CN6 as indicated in the image:



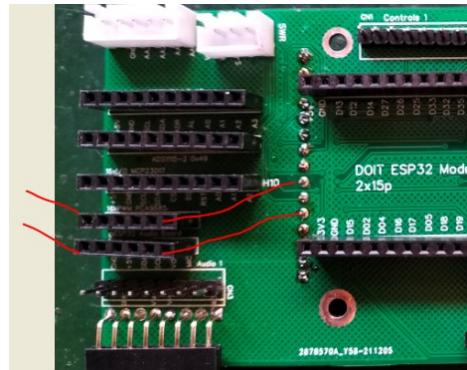
8. Solder the female 6-pin sockets H13 and H14



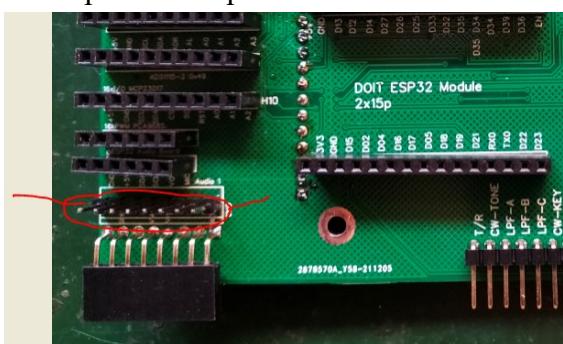
9. Solder the female 10-pin sockets H2 and H4



10. Solder the female 6-pin sockets H11 and H12.



11. Solder the 8-pin header pin CN3



12. Finally, solder the 7805 U2 regulator.



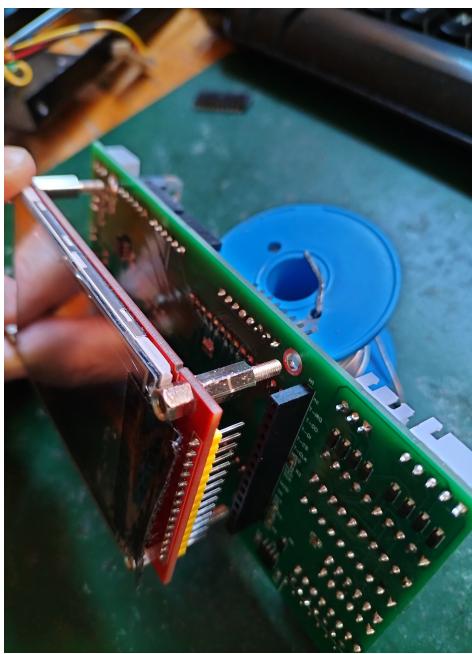
TFT SCREEN

1. Remove the front panel of the uBitx radio.
 - Release the connector that goes to the Encoder.
 - The nuts that hold the MIC, SPK and KEYER connectors.
 - The 4 screws that hold the TFT screen.
 - The 4 screws that hold the front panel, 2 on the top and 2 on the bottom.



- 2. Unplug the Raduino module that contains the Arduino Nano.
 - Remove the 4 nuts that hold the TFT screen to the original Raduino. They are the nuts on the opposite side of the screen.
 - Remove the TFT screen

- Insert the screen into the new Raduino32

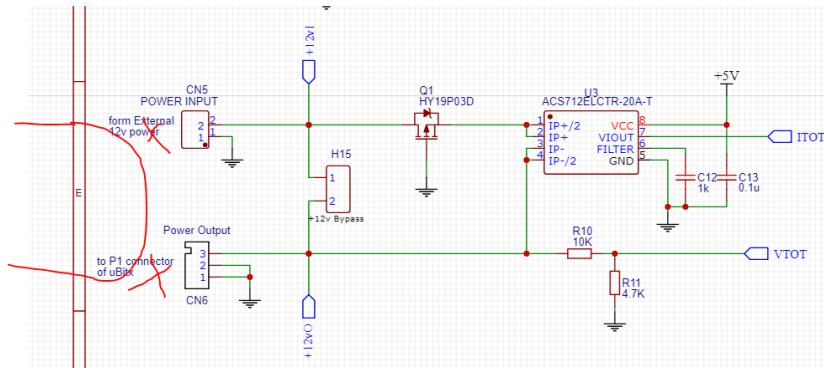


- Insert the ESP32 module into the Raduino32 board (Note: in this case the module has a modification to connect an external antenna. This modification is not essential and is explained later)



- Modification to connect + 12VI and + 12VO power

If you do not make this modification, the equipment will work but the supply voltage and current cannot be measured.



Carefully cut the outer left track and solder the red wires to the tracks on the PCB. The 3-pin cable on the right and the 2-pin cable on the left. The black wires can be soldered to the GND of the screw base. The yellow wire can be left unconnected or connected to GND.



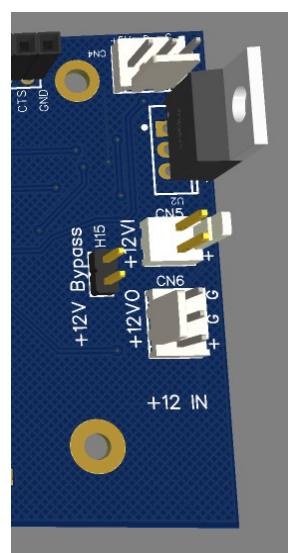
With this modification, the Voltage and Current measurement will work (if the ADS1115 modules have been equipped)

Insert the 2-pin and 3-pin connectors into their respective sockets.

Versión 6



Versión 6.1

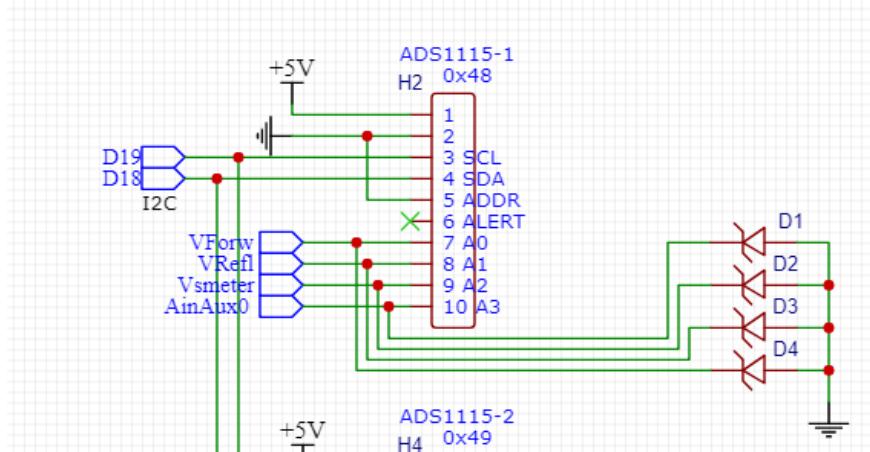


So far, the assembly for the operation of the basic functions. Hereinafter the assembly of the optional elements is shown.

ADS1115 MODULES

The Raduino32 can be equipped with 2 ADC modules of the type ADS1115. Each of these modules has 4 analog inputs. These modules communicate with the ESP32 microprocessor via I2C and each one has its specific address, set by the wiring. It is not modifiable by software.

Module #1 has the I2C address 0x48 (hex) and is inserted into socket H2.
As seen, the ADDR terminal is connected to GND.

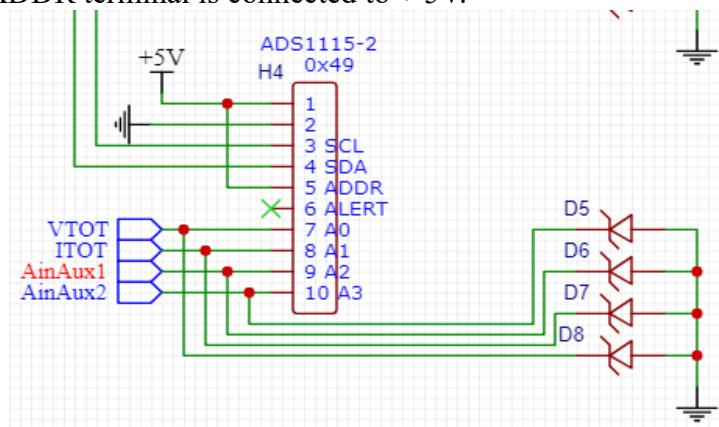


The 4 analog inputs are used to

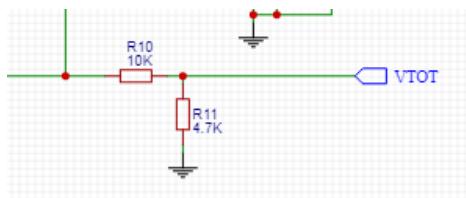
- A0: VFOW, SWR sensor, forward voltage.
- A1: VREFL, reflected wave voltage of the SWR sensor.
- A2: VSMETER, S-meter sensor voltage
- A3: AinAux0, free for other uses. Appears at connector H7.

The 4.7v Zener diodes serve to limit the voltage of the analog inputs. The ADS1115s are very sensitive to these surges.

Module #2 has the I2C address 0x49 (hex) and is inserted into socket H4.
As seen, the ADDR terminal is connected to + 5V.



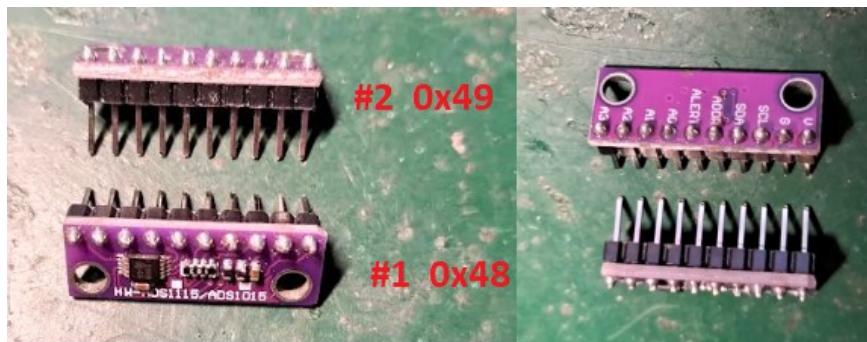
- A0: VTOT, supply voltage measurement + 12v.
This voltage is taken through a voltage divider to adjust it to the range supported by the ADS1115.



- A1: ITOT, current intensity measured by the ACS712 sensor.
- A2: AinAux1, free for other uses. Appears at connector H7.
- A3: AinAux2, free for other uses. Appears at connector H7.

Preparation of the ADS1115 modules

In order to correctly insert the modules in their sockets, it is necessary to solder the pins in a different way in each one.



As seen in the image, in the #1 the pins are soldered on the BOTTOM side, the one that has no components. In module #2, they are welded on the TOP side, the component side.



Make sure that the pins corresponding to V and GND are on the correct side, towards the outside of the Raduino32 PCB.

S-METER / VTOT/ITOT

To use the **S-meter function** you need to equip **module #1** of ADSS115.

To use the **supply voltage and current measurement**, it is necessary to equip **module # 2**.

SOFTWARE

The software is under constant development and there may be new features or minor changes to existing features.

I use the UECIDE editor (<https://uecide.org/>), for development, which is more powerful and faster than the Arduino IDE. It can also be compiled with the Arduino IDE.

I have started from the code of KD8CEC, although I have modified many parts of it with the purpose of simplifying it, eliminating the unused parts. Some functions I have used as they were, others I have modified and in other cases I have used the original ones from VU2ESE.

The result is a hodgepodge of their own code and theirs. If someone finds it interesting I will be satisfied.

The code, also the ***ubitx32_20.bin*** file, can be found at <https://github.com/ramonlh/ubitx32>

For programming the ESP32 module directly with the .bin file, *Flash Download Tools* from Espressif can be used: <https://www.espressif.com/en/support/download/other-tools>

UECIDE INSTALLATION for ubitx32_20

DOWNLOAD

Download Uecide from <https://uecide.org/download> (latest version 0.11.10)
If it is the first time, download the “full” version. Install with default options.

JAVA

If Java Runtime is not installed, go to <https://java.com/es/download/> and install it.

[Descargar Java para Windows](#)

Recomendado Version 8 Update 321 (Tamaño de archivo: 2.15 MB)

Fecha de publicación: 18 de enero de 2022

⚠ Actualización importante de la licencia de Oracle Java

La licencia de Oracle Java ha cambiado para las versiones publicadas a partir del 16 de abril de 2019.

El nuevo acuerdo de licencia de Oracle Technology Network para Oracle Java SE es sustancialmente diferente a las licencias de Oracle Java anteriores. La nueva licencia permite ciertos usos, como el uso personal y de desarrollo, sin coste alguno (aunque podría haber otros usos autorizados en licencias de Oracle Java anteriores que ya no estén disponibles). Revise las condiciones con atención antes de descargar y utilizar este producto. Puede consultar las preguntas frecuentes [aquí](#).

La licencia comercial y el soporte están disponibles con una [suscripción de Java SE](#) de bajo coste.

Oracle también ofrece la última versión de OpenJDK con la [licencia pública general](#) de código abierto en jdl.java.net.

Aceptar e iniciar descarga gratuita

Al descargar Java, confirma que ha leído y acepta las condiciones del [acuerdo de licencia de Oracle Technology Network](#) para Oracle Java SE

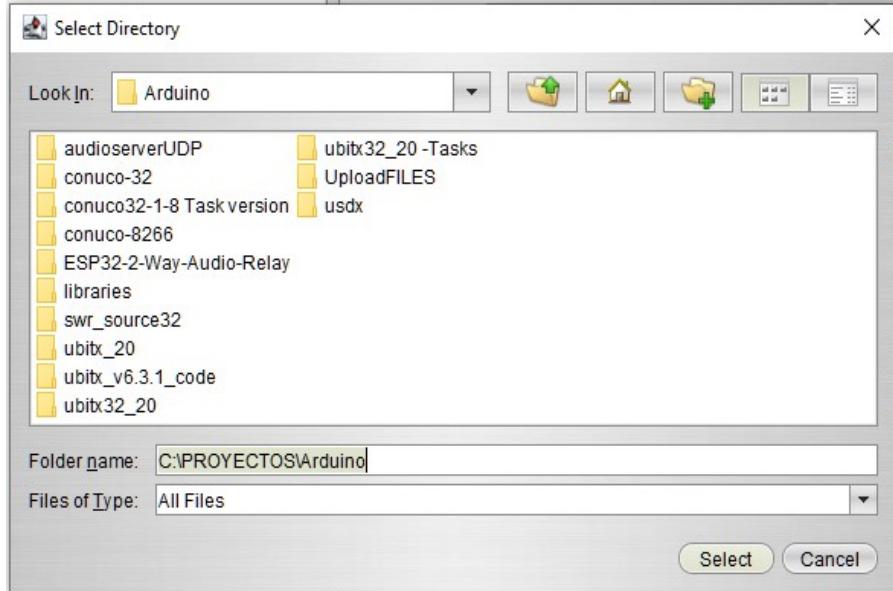
Once the Java Runtime is installed, run Uecide again.

INSTALL ESP32 BOARD

Go to Tools->Plugin manager->Boards

Search “ESP32”, install board esp32doit-devkit-v1

Go to **File->Preferences->Locations**: put “SketchBook location” to the folder where the Arduino projects are, for example:



INSTALL LIBRARIES

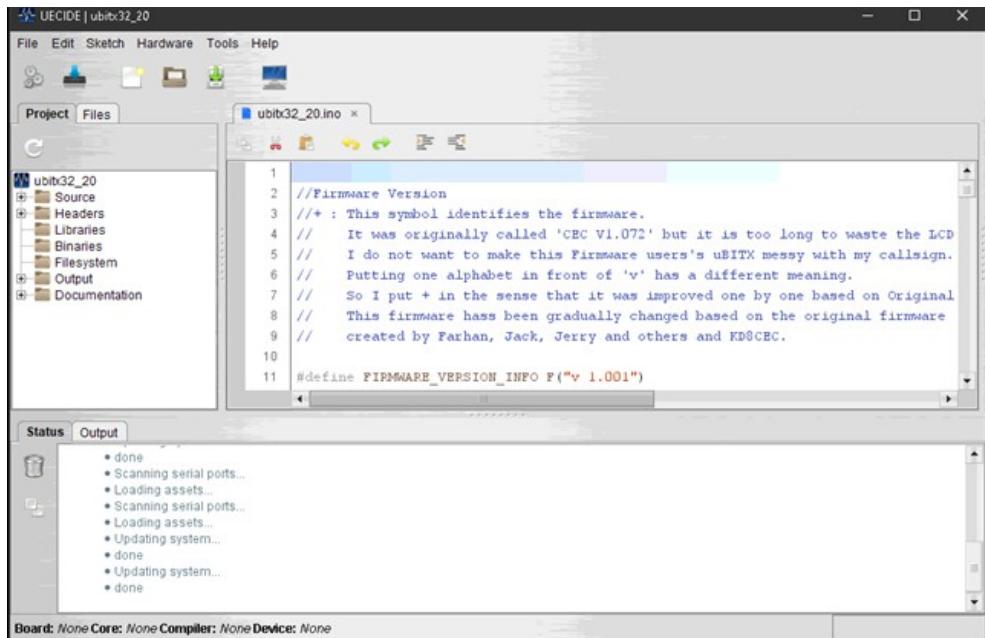
Download the libraries from <https://github.com/ramonlh/ubitx32/tree/master/libraries> in a folder of our choice.

Go to **File->Preferences->Locations**, click “**Add Location**”, select the folder where we have downloaded the libraries.

Unzip all the libraries, each one in its own folder.

After adding the libraries it is necessary to restart Ucide.

OPEN PROJECT “ubitx32_20”



Select Board: **Hardware->Board->ESP32->Doit.....**

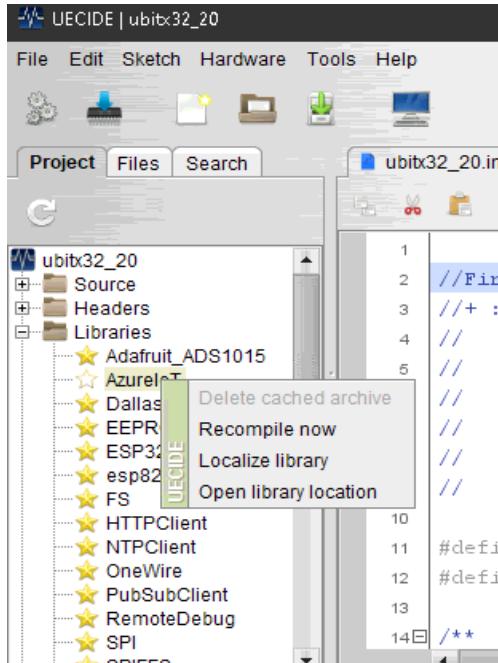
Select Serial Port: **Hardware->Devices->COMXX**

Compile ubitx32_20

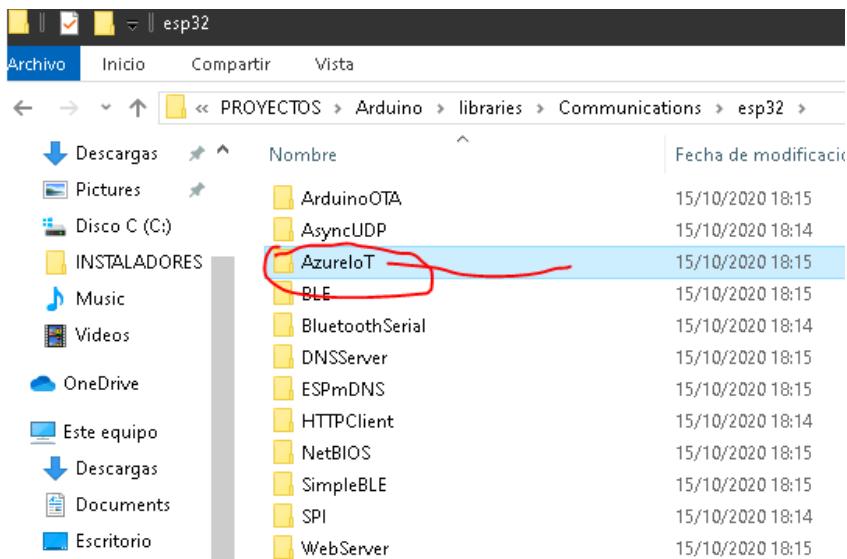
Open project: **File->Open...ubitx32_20**

For some reason, the first time the **AzureIoT** library is added to the project and an error occurs. You need to remove it from the project:

Click on the right button and “Open library location”.



A file explorer window will open:



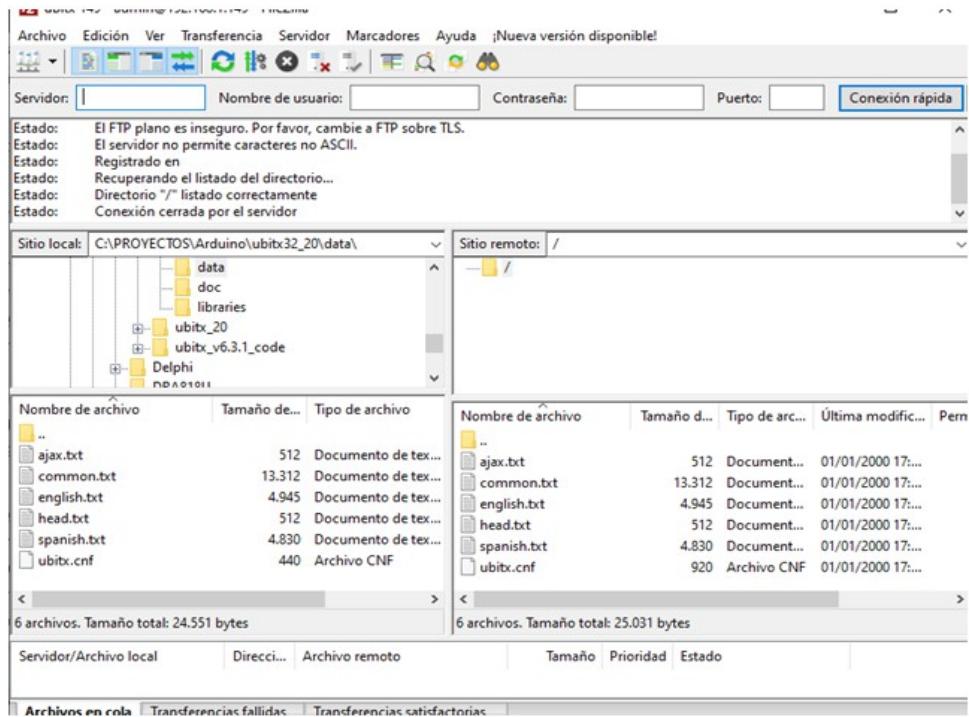
Delete **AzureIoT** folder.

Close and reopen Uecide. The library will no longer be there and will compile correctly.

FTP ACCESS

Access via FTP allows you to manage system files, make backup copies or add new language files.

At the moment, the FTP access password is “admin/admin”. The FTP server can be disabled from the web page.

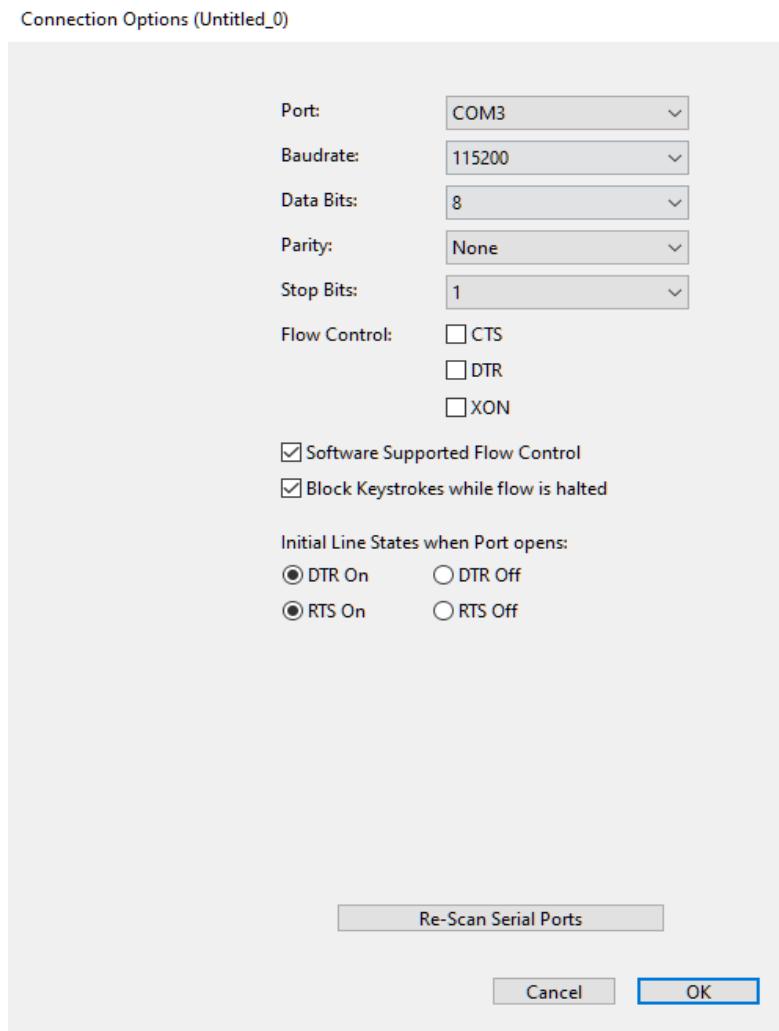


USING THE SERIAL PORT 2

Serial port 2 can be used for debugging and basic Raduino32 configuration.

I use the terminal program **CoolTerm** which can be found here: <http://freeware.the-meiers.org>.
Thanks from here to Roger.

Select the COMxx serial port and configure the terminal with the following parameters:



Click Connect

Turn on the uBitx radio or press Reset on the ESP32 board. We will see the following information when starting the radio (it may vary depending on the options or the version):

```
===== Init =====
Serial 1 started
Serial 2 started
Vers.:v 1.201
TFT started
SPIFFS 0
/spanish.txt 6298
/english.txt 6301
/head.txt 512
/dash.txt 12576
/ajax.txt 512
```

```

/ubitx.cnf 1520
/ubitx.mem 4000
/german.txt 6223
/common.txt 14698
Ports OK
readconfEEPROM:1520
Read readconf():1520
I2C started
Oscillators started
Pin Tone OK
DS18B20 probes
    Sondas Temp.:3
    Modo: Power
    404615637050000233
    40125803805000021
    4022323737050000012
    Started
    Values read
ADS1115 started
ATU started
WiFiMode: AP+STA
AP mode:
    Channel:10
    AP MAC:3C:71:BF:FE:65:19
    IP:192.168.4.1
STA mode:
    Static IP:YES 192.168.1.149.
    Gateway:192.168.1.1.
    EEMask:255.255.255.0.
    EEDns:8.8.8.8.
    EEDns2:8.8.4.4.
    Connecting: conuco4/password..... OK
    STA IP: 192.168.1.149
    STA MAC: 3C:71:BF:FE:65:18
    Subnet Mask: 255.255.255.0
    Gateway IP: 192.168.1.1
    DNS: 8.8.8.8
IP services
    FTP server started, port 21 admin/admin
    HTML server started
    Web server started, port 8088
    TCP server started, port 8084
    UDP-S service started, port 8085
    UDP-F service started, port 8087
    Debug server disabled
hostmyip:icanhazip.com
checkInternet host:www.google.com:80/= 200
    Check Internet: OK
Internet services
    initTime
    MyIP server: icanhazip.com OK 81.32.101.202
msg: /dweet/for/ubitxserver-EA4GZI?ip=81.32.101.202
    Dweet sending... OK
    Dweet resp: {"this":"succeeded","by":"dweeting","the":"dweet","with":
{"thing":"ubitxserver-EA4GZI","created":"2022-01-28T18:27:50.832Z","content":
{"ip":"81.32.101.202"},"transaction":"e2040676-9f20-4b50-a05c-8c34f8fa17c4"}}
TPA2016 started
radio power:80
END SETUP
=====
Type 'h' to help

```

In another section the boot information will be described.

You can interact with the Raduino32 through this serial port with a series of commands. To see them write “h” and press ENTER.

```
-----
a          (auto WiFi: yes/no)
f          (check files)
F          (reset factory)
h          (help)
p,SSIDpass (set passSTA)
P,SSID-APpass (set pass AP)
r          (reset)
s,SSIDname (set SSID STA)
S,SSID-AP  (set SSID AP)
t          (format SPIFFS)
u          (show status)
w,n       (wifimode:0,1,2,3=OFF,STA,AP,STA+AP)
```

Commands:

u → Enter	Show current parameters

	WiFi Mode: AP+STA IP: 192.168.1.149 Port: 8088 mask: 255.255.255.0 GW: 192.168.1.1 ssid: conuco4 pass: xxxxxxxx ssidAP: UBITX passAP: 12341234 autoWiFi: NO Conn: OK
f → Enter	Check if the necessary files exist
	/spanish.txt 6298 /english.txt 6301 /head.txt 512 /dash.txt 12576 /ajax.txt 512 /ubitx.cnf 1520 /ubitx.mem 4000 /german.txt 6223 /common.txt 14698 files OK
F → Enter	Factory reset.
H → Enter	Show available commands
r → Enter	Reset
t → Enter	Format SPFFS file system
p,SSID → Enter	Defines the SSID to connect to
s,PASS → Enter	Define the SSID password
S,SSID → Enter	Define the name of the SSID in AP mode
P,PASS → Enter	Defines the password for AP mode
w,x → Enter	Define Wi-Fi mode. 0:OFF, 1:STA, 2:AP, 3:AP+STA
a → Enter	Changes the state of the Auto WiFi value. If it is YES, and the WiFi mode is equal to 1 or 3, the Raduino32 will try to connect to the network defined by SSIDname/SSIDpass.

Flash Download Tools fo ESP32/Esp8266

- Download tools from: <https://www.espressif.com/en/support/download/other-tools>

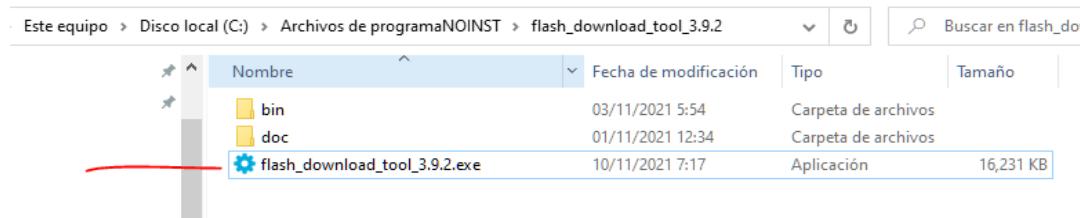
The screenshot shows the Espressif Support page. The navigation bar includes links for Products, Solutions, Support (which is highlighted), Ecosystem, Company, Join Us, and Contact Us. There are also links for Sales Questions, Technical Inquiries, Circuit Schematic & PCB Design Review, Get Samples, Become Our Supplier, and Comments & Suggestions. A search bar at the top right allows searching for keywords.

The main content area displays a table titled "Flash Download Tools" with three results. The columns are Title, Platform, Version, Release Date, and Download. The first result is circled in red. Below the table is another section titled "Certification and Test" with two entries, also circled in red. At the bottom right of the content area is a "Download selected" button.

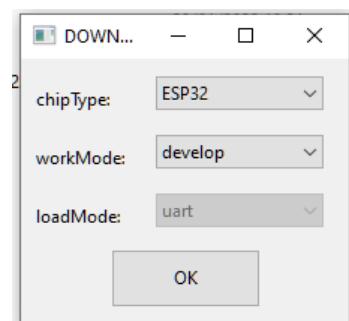
Title	Platform	Version	Release Date	Download
+ Flash Download Tools	Windows PC	V3.9.2	2021.11.10	

Title	Platform	Version	Release Date	Download
+ ESP RF Test Tool and Test Guide	ZIP	V2.8	2021.11.10	
+ ESP8266 & ESP32 WFA Certification and Test Guide	Windows PC	V1.1	2020.08.05	

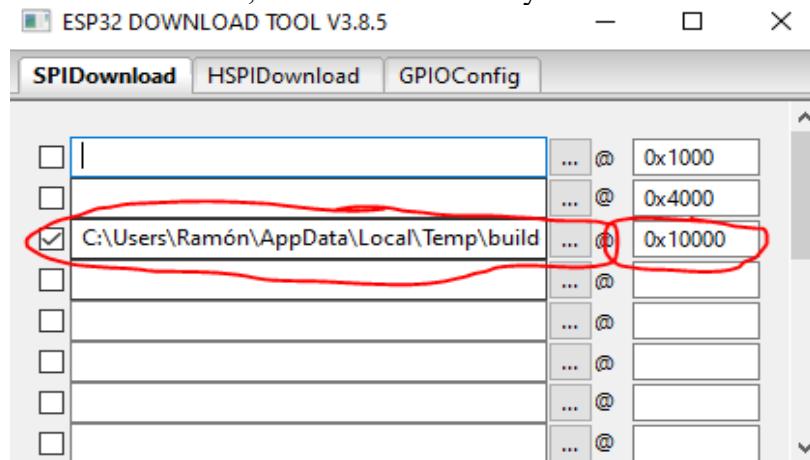
- Unzip folder (no needed installation)
- Run: *flash_download_tool_3.9.2*



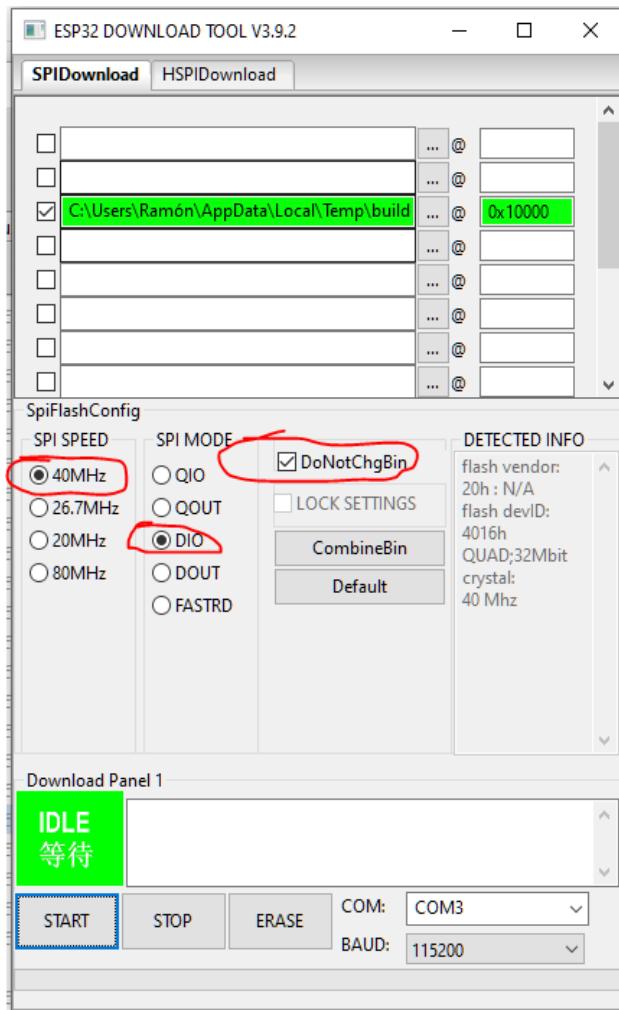
- Be patient, it takes a minute.
- Select “ESP32” and “Developper Mode”.



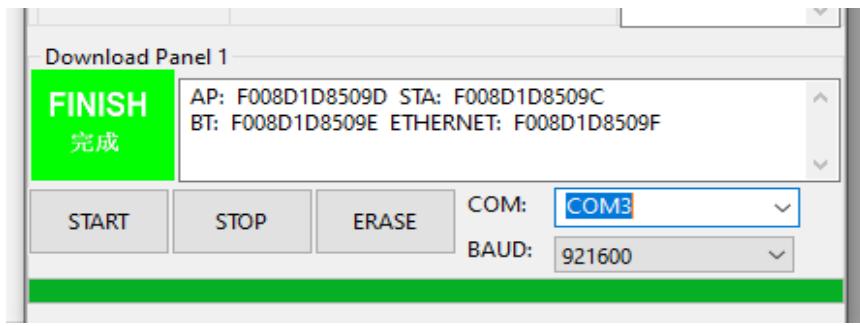
- Find file “**conuco32-1-8.bin**”, check CheckButton y escribir “**0x10000**”.



- Check options:



- Click “**START**”
 - Wait “**FINISH**”



- **Reset uBitx.**

FIRST START-UP

The configuration is stored in files in the memory of the ESP32 due to the fact that the EEPROM does not have enough size. These files must be uploaded only once to the board. I explain how to do it:

Install the file uploader tool from here: <https://randomnerdtutorials.com/install-esp32-filesystem-uploader-arduino-ide/>

In the folder where the “ubitx32_20” project is saved there is a subfolder called “data”

Nombre	Fecha de modificación	Tipo
ajax.txt	23/08/2021 18:53	Documento de te...
common.txt	13/10/2021 20:27	Documento de te...
english.txt	06/11/2021 22:02	Documento de te...
german.txt	10/10/2021 21:25	Documento de te...
head.txt	23/08/2021 18:53	Documento de te...
spanish.txt	11/10/2021 20:17	Documento de te...
ubitx.cnf	23/08/2021 18:53	Archivo CNF
ubitx.mem	23/08/2021 18:53	Archivo MEM

Open Arduino IDE

Load the sketch “ubitx32_20”

Select COM portxx

Go to Tools → ESP32 Sketch Data Upload

The files in the “data” folder will be uploaded to the ESP32.

Once the files are uploaded, turn off the uBitx radio. Press and hold the tuning button and turn on the radio to make a sete to the factory values. On the screen you should see:

Init...
Version: 1.203
Reset Factory...

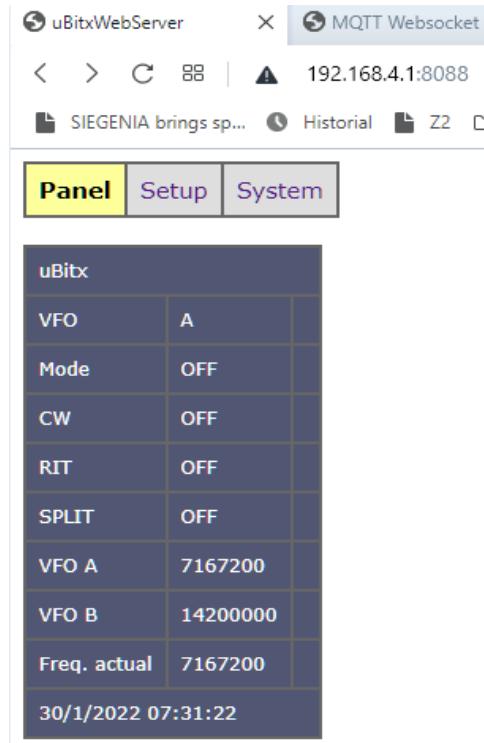
From this moment the radio is already usable.

SETUP Wi-Fi

By default, Raduino32 is activated in AP mode. You need to search your PC/Tablet/Phone for the SSID named “UBITX”.

Connect to this WiFi network, the password is “12341234”. Once connected, open a browser and go to the address <http://192.168.4.1:8088>

You should see a screen like this:



Go to option **Setup → Net**

Net	
Auto Conn.	<input checked="" type="checkbox"/>
Mode	AP+STA
STA mode SSID	yourssid
STA mode Password	yourpass
AP mode SSID	UBITX
AP mode Password	12341234
Channel	3
MAC	000000000000
Static IP	<input checked="" type="checkbox"/>
IP address	192.168.1.149
IP	192 168 1 149
Mask	255 255 255 0
Gateway	192 168 1 1
DNS	8 8 8 8
Public IP	81.32.101.202

Guardar

Click the **Scan** link to search for available networks and select your SSID:

nAP	15
conuco4	-27 dBm
vodafone9918	-82 dBm
MOVISTAR_8453	-83 dBm
CONUCO_150	-84 dBm
vodafoneBA1618	-85 dBm
	-85 dBm
MOVISTAR_26B2	-88 dBm
MOVISTAR_FCAD	-88 dBm
amparo	-88 dBm
	-90 dBm
	-90 dBm
ExtendedWiFi	-91 dBm
DIGIFIBRA-DR2F	-93 dBm
MOVISTAR_F130	-93 dBm
MOVISTAR_5FC8	-96 dBm

You will return to the previous screen and complete the fields and click **Save**:

Panel	Setup	System																																																					
Device	Bands	Memories	CW	S-meter	Net	N																																																	
<table border="1"><tr><td>Auto Conn.</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Mode</td><td>AP+STA</td></tr><tr><td>STA mode SSID</td><td>conuco4</td></tr><tr><td>STA mode Password</td><td>xxxxxxxx</td></tr><tr><td>AP mode SSID</td><td>UBITX</td></tr><tr><td>AP mode Password</td><td>12341234</td></tr><tr><td>Channel</td><td>3</td></tr><tr><td>MAC</td><td>000000000000</td></tr><tr><td>Static IP</td><td><input checked="" type="checkbox"/></td></tr><tr><td>IP address</td><td>192.168.1.149</td></tr><tr><td>IP</td><td>192</td><td>168</td><td>1</td><td>149</td></tr><tr><td>Mask</td><td>255</td><td>255</td><td>255</td><td>0</td></tr><tr><td>Gateway</td><td>192</td><td>168</td><td>1</td><td>1</td></tr><tr><td>DNS</td><td>8</td><td>8</td><td>8</td><td>8</td></tr><tr><td>Public IP</td><td>81.32.101.202</td></tr><tr><td>Guardar</td><td colspan="6"></td></tr></table>							Auto Conn.	<input checked="" type="checkbox"/>	Mode	AP+STA	STA mode SSID	conuco4	STA mode Password	xxxxxxxx	AP mode SSID	UBITX	AP mode Password	12341234	Channel	3	MAC	000000000000	Static IP	<input checked="" type="checkbox"/>	IP address	192.168.1.149	IP	192	168	1	149	Mask	255	255	255	0	Gateway	192	168	1	1	DNS	8	8	8	8	Public IP	81.32.101.202	Guardar						
Auto Conn.	<input checked="" type="checkbox"/>																																																						
Mode	AP+STA																																																						
STA mode SSID	conuco4																																																						
STA mode Password	xxxxxxxx																																																						
AP mode SSID	UBITX																																																						
AP mode Password	12341234																																																						
Channel	3																																																						
MAC	000000000000																																																						
Static IP	<input checked="" type="checkbox"/>																																																						
IP address	192.168.1.149																																																						
IP	192	168	1	149																																																			
Mask	255	255	255	0																																																			
Gateway	192	168	1	1																																																			
DNS	8	8	8	8																																																			
Public IP	81.32.101.202																																																						
Guardar																																																							

Turn the uBitx radio off and on. If everything has worked well, the Raduino32 should connect to the selected SSID as a client, not as an AP. Disconnect the PC from the UBITX network.

You should now be able to access Raduino32 again at <http://192.168.1.149:8088>

uBitx	
VFO	A
Mode	OFF
CW	OFF
RIT	OFF
SPLIT	OFF
VFO A	7167200
VFO B	14200000
Freq. actual	7167200

30/1/2022 07:40:55

You can now navigate through the different options to configure the radio to your liking.

