

## **Raduino 32**

Se trata en este proyecto de utilizar para el control de la radio Ubitx v6 un microcontrolador ESP32 en lugar de un Arduino Nano. El ESP32 es más rápido y, además, dispone de hasta 4Mb de RAM y de la posibilidad de utilizar archivos mediante SPIFFS.

Adicionalmente, permite utilizar conexión WiFi o BT.

He partido del firmware original de Ashhar Farhan, VU2ESE y partes del firmware CEC de KD8CEC, a los que agradezco su esfuerzo.

### **FUNCIONES**

Se han conservado las funciones básicas del diseño original, por ejemplo

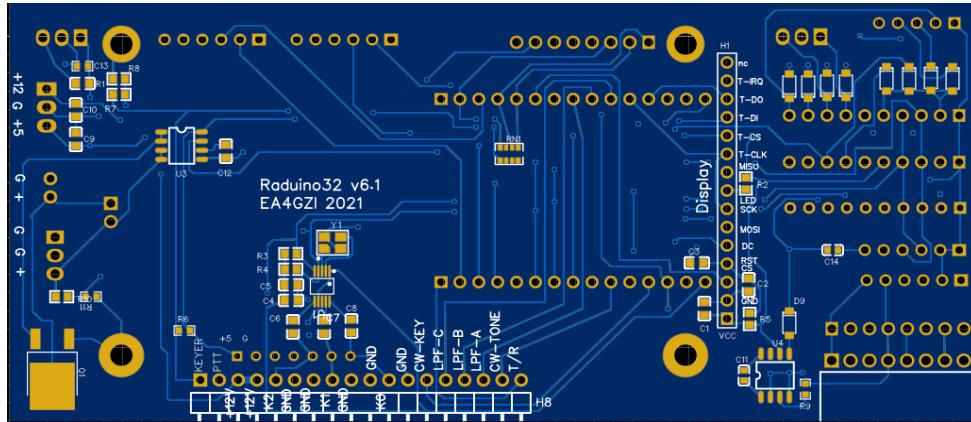
- Modos LSB,USB,CW.
- Doble VFO
- RIT
- Split
- Entrada directa de frecuencia.
- CAT control.
- 

y se han añadido algunas nuevas:

- La configuración reside en un sector de memoria EEPROM (en la v 1.1, 3/2/2021)
- Cambios en la interfaz de usuario.
- Múltiples pasos de sintonía, desde 1 hz hasta 100 Mhz.
- Función SCAN.
- Temperatura transistores de potencia y regulador LM7805
- Medidor de ROE (SWR)
- Control y configuración vía Web
- Calibración
- Parámetros usuario.
- Límites de bandas
- Servicios de red: MQTT, FTP server.
- RTC vía NTP
- Actualización remota del firmware vía web.
- Segundo puerto serie para depuración u otros usos, además del primero usado para el CAT.
- Idiomas Español e Inglés. Pueden añadirse fácilmente otros idiomas mediante un fichero.
- 100 memorias.

## Instrucciones Montaje Raduino32 v6.1

La placa Raduino32, versión 6.1 está pensada para sustituir de forma sencilla a la placa Raduino original con Arduino Nano del uBitx v6. Admite el uso de un microcontrolador ESP32 con las ventajas en cuanto a tamaño de memoria y otras funciones, conservando el display TFT táctil de 2.8".



### ¡NO SE INCLUYE EL MÓDULO ESP32, NI EL ADS1115!

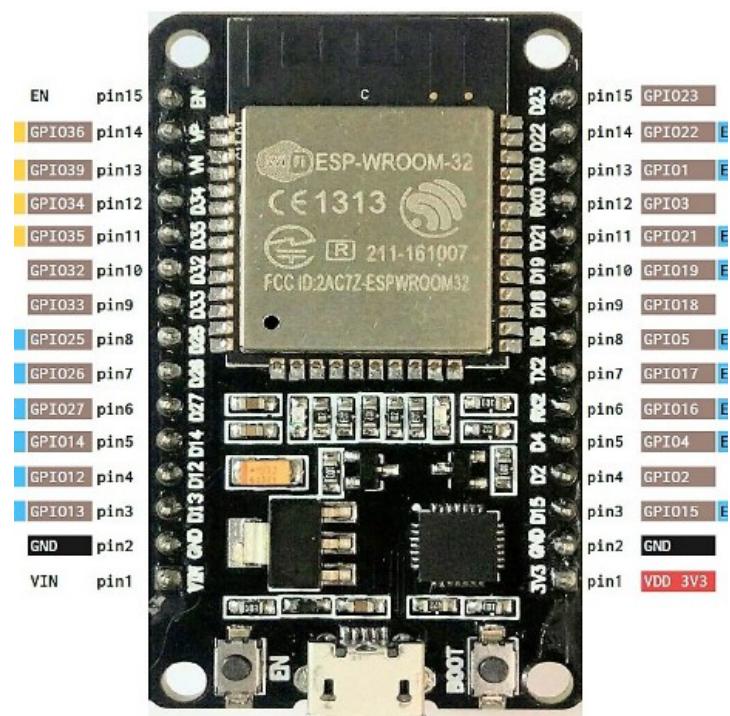
Para el funcionamiento básico no es necesario modificar la placa principal del uBitx, solamente en el caso en que se quieran utilizar las funciones adicionales hay que realizar unas modificaciones, por otra parte sencillas.

Esta versión del Raduino32 incluye de fábrica los siguientes componentes principales:

- IC Si5351, generador de señales de reloj para el oscilador (U1)
  - Reg. LM7805, para alimentar los circuitos a 5v (U2)
  - IC ACS712, sensor de efecto Hall para la medida de corriente (U3)
  - IC LM386, sensor para el S-meter (U4)
  - MOSFET P-channel, para protección contra inversión de la polaridad (Q1)
- 
- Conectores para conectar el Raduino32 a la placa principal del uBitx (H8,H9)
  - Pines para insertar la pantalla TFT táctil de 2.8" (H1)
  - Pines para conectar el Encoder (CN1)
  - Zócalo para insertar el módulo ESP32 DoIt de 2x15 pines (H3,H5)
  - Zócalos para insertar dos módulos ADC ADS115 (H2,H4). Opcional si se desea medir:
    - Señal S-Meter
    - VTOT, ITOT, Tensión y corriente de alimentación +12v.
    - VForw, vRefl, señales del sensor de ROE.
  - Zócalo para insertar un módulo MCP23017 (H10), un multiplexor de 16 I/O controlado por I2C. No usado aún, permitiría controlar otras señales: sintonizador de antena, amplificador de potencia, etc.
  - Zócalo para insertar un módulo PCA9685 (H12), de 16 canales PWM. Sólo usado experimentalmente para controlar servomotores de un sintonizador de antena.

- Zócalo para conectar una placa para enviar y recibir audio a través de TCP/UDP (H11)
  - Pines para extensión de señales de audio (CN3).
  - Zócalos para puertos serie 1 y 2 (H13, H14)  
  - Conector para 3 entradas analógicas (H7)
  - Conector para la señales del sensor de ROE (CN2)
  - Conector para sondas de temperatura DS18B20 (CN4)
  - Conector para entrada de +12 voltios, +12VI (CN5)
  - Conector para salida de +12 voltios, +12VO (CN6)
  - Pines para hacer un bypass para el circuito de +12V (H15)

Para el funcionamiento básico es necesario añadir un módulo ESP32 DoIt o equivalente. El módulo debe tener 30 pines (2 x 15) y una disposición según la imagen:



## MONTAJE MÍNIMO

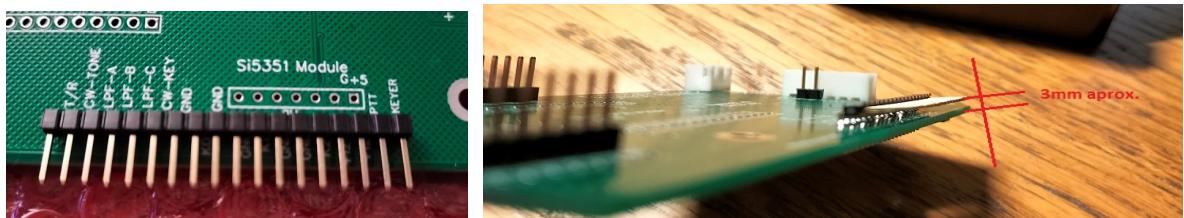
**¡ATENCIÓN! Los componentes SMD están soldados en fábrica, tenga mucho cuidado al manejar la placa para no dañarlos.**

En esta apartado se describe el montaje de los componente necesarios para el funcionamiento con las funciones básicas.

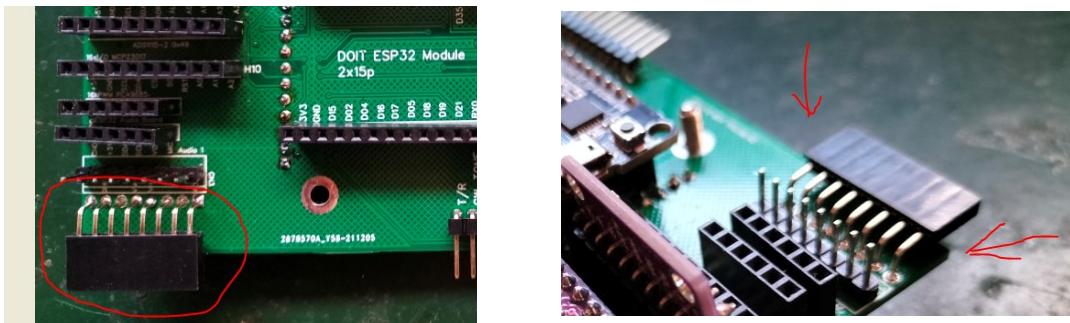
El orden de montaje es solamente indicativo y basado en mi propia experiencia.

## **SOLDADURAS**

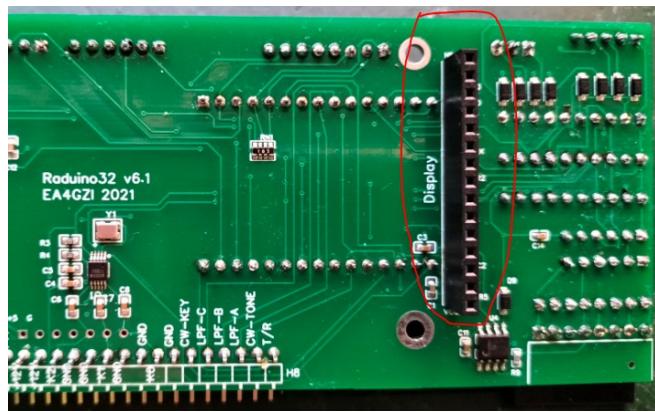
1. Soldar la tira macho H8, de 18 pines y 90°, que conecta el Raduino32 a la placa principal del uBitx. **¡ATENCIÓN! Esta tira de pines se suelda por la cara INFERIOR, a pesar de que en la PCB aparece dibujada por la cara superior.**



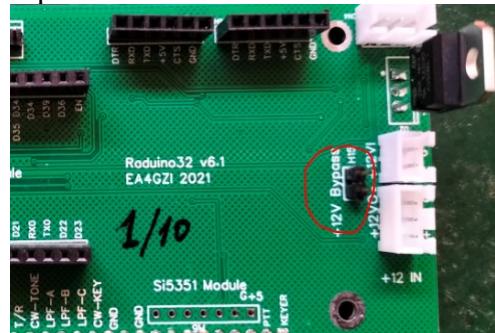
2. Soldar el zócalo hembra H9, de 8 pines y 90°, que conecta el Raduino32 a las señales de audio del uBitx. **¡ATENCIÓN! Esta tira de pines se suelda por la cara INFERIOR, a pesar de que en la PCB aparece dibujada por la cara superior.**



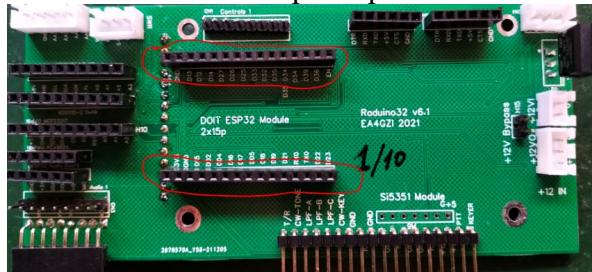
3. Soldar el zócalo hembra de 15 pines que se inserta en la pantalla TFT. ¡ATENCIÓN! Esta tira de pines se suelda por la cara **SUPERIOR**.



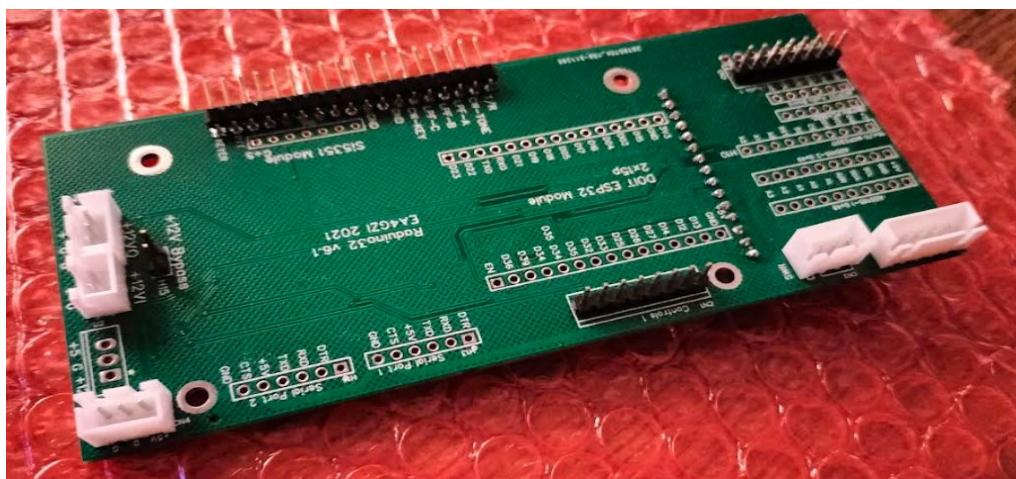
4. Antes de continuar es conveniente verificar que se puede insertar correctamente la PCB en la placa principal del uBitx y en la TFT.
  5. Soldar la tira macho de 2 pines H15.



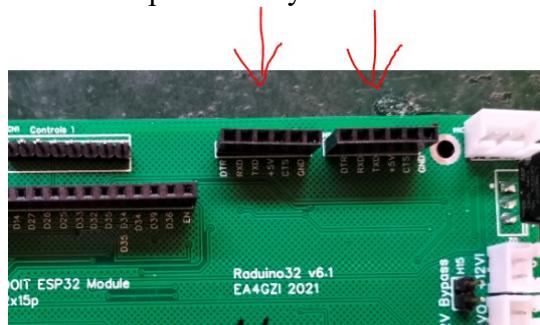
6. Soldar los dos zócalos hembras de 1 x 15 pines para insertar el módulo ESP32.



7. Soldar los conectores H7, CN2, CN4, CN5 y CN6 según se indica en la imagen:



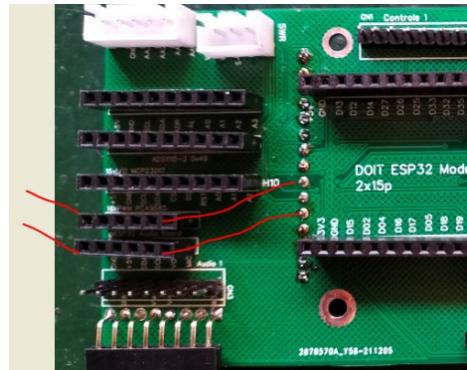
8. Soldar los zócalos hembras de 6 pines H13 y H14



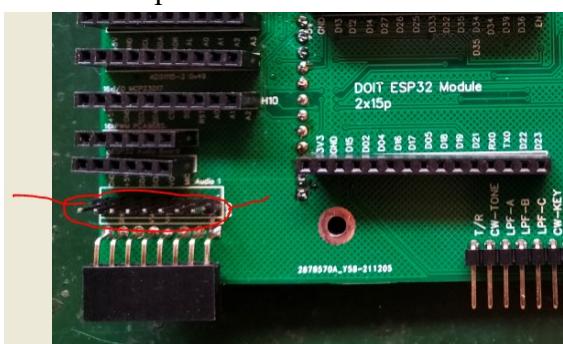
9. Soldar los zócalos hembras de 10 pines H2 y H4



10. Soldar los zócalos hembras de 6 pines H11 y H12.



11. Soldar la tira de 8 pines CN3



12. Finalmente, soldar el regulador 7805 U2.

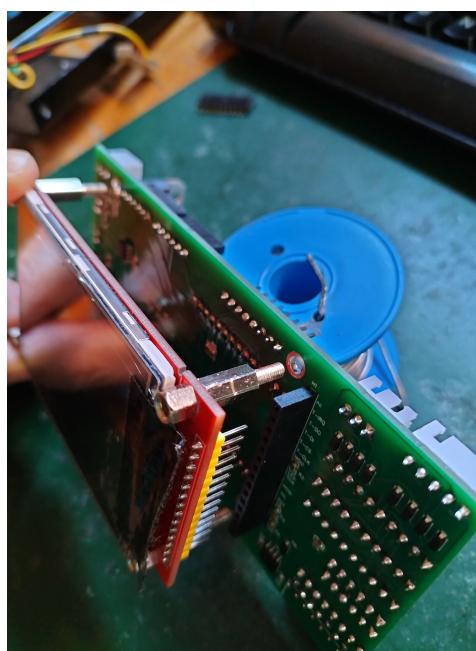


## PANTALLA TFT

1. Desmontar el panel frontal de la radio uBitx.
  - Soltar el conector que va al Encoder.
  - Las tuercas que sujetan los conectores de MIC, SPK y KEYER.
  - Los 4 tornillos que sujetan la pantalla TFT.
  - Los 4 tornillos que sujetan el panel frontal, 2 arriba y 2 abajo.
2. Desenchufar el módulo Raduino que contiene el Arduino Nano.
  - Quitar las 4 tuercas que sujetan la pantalla TFT al Raduino original. Son las tuercas que están al lado opuesto de la pantalla.
  - Extraer la pantalla TFT



- Inserta la pantalla en el nuevo Raduino32

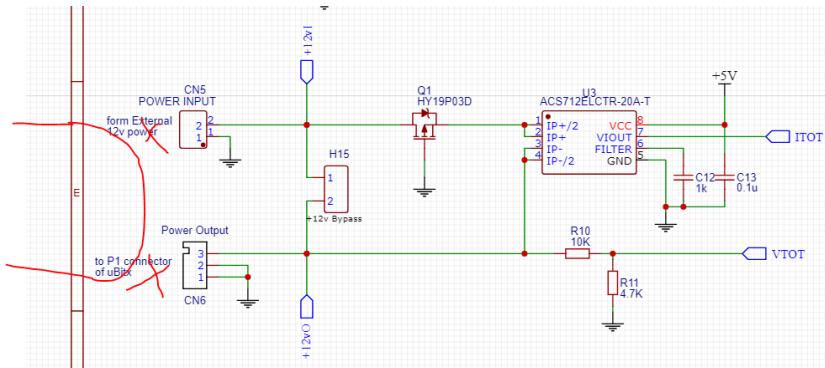


- Inserta el módulo ESP32 en la placa del Raduino32 (Nota: en este caso el módulo tiene una modificación para conectar una antena exterior. Esta modificación no es imprescindible y se explica más adelante)



- Modificación para conectar la alimentación +12VI y +12VO

Si no realiza esta modificación, el equipo funcionará pero no podrán medirse la tensión ni la corriente de alimentación.



Cortar con cuidado la pista exterior izquierda y soldar los cables rojos a lñas pistas de la PCB. El cable de 3 pinos a la derecha y el cable de 2 pinos a la izquierda. Los cables negros pueden soldarse a la GND de la base del tornillo. El cable amarillo puede dejarse sin conexión o conectarlo a GND.



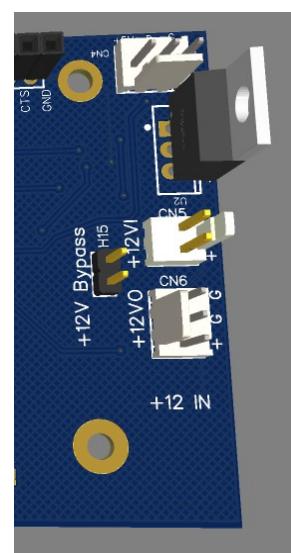
Con esta modificación, funcionará la medida de Tensión y Corriente (si se han equipado los módulos ADS1115)

Insertar los conectores de 2 y 3 pinos en sus respectivos zócalos.

Versión 6



Versión 6.1

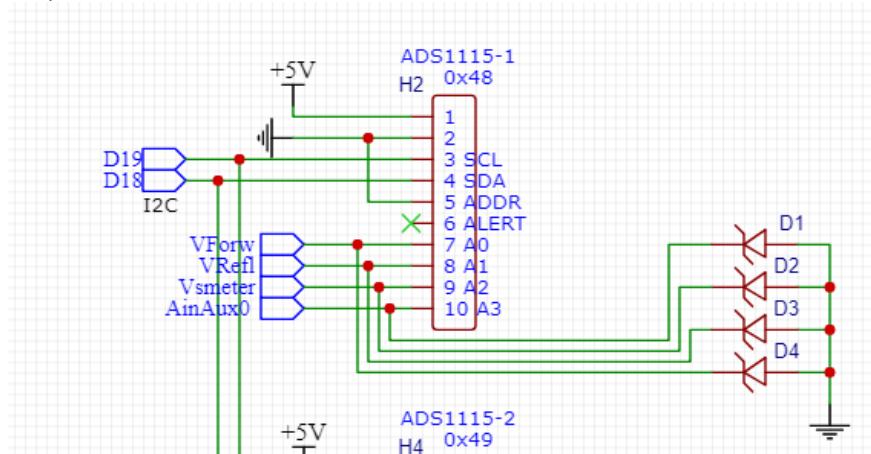


Hasta aquí, el montaje para el funcionamiento de las funciones básicas. En adelante se muestra el montaje de los elementos opcionales.

## MÓDULOS ADS1115

El Raduino32 puede estar equipado con 2 módulos ADC del tipo ADS1115. Cada uno de estos módulos dispone de 4 entradas analógicas. Estos módulos se comunican con el microprocesador ESP32 vía I2C y cada uno tiene su dirección específica, fijada por el cableado. No es modificable por software.

**El módulo #1** tiene la dirección I2C 0x48 (hex) y se inserta en el zócalo H2.  
Como se ve, el terminal ADDR está conectado a GND.

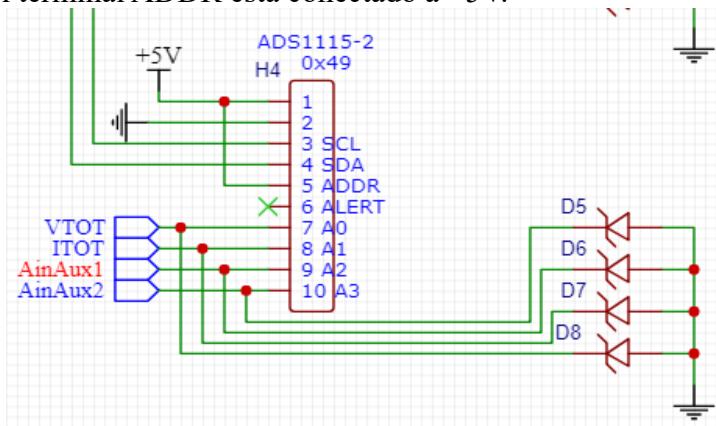


Las 4 entradas analógicas se usan para

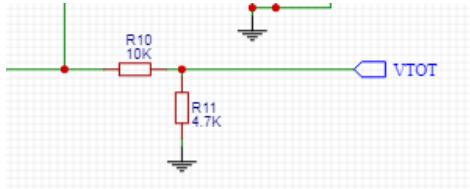
- A0: VFOW, tensión onda directa del sensor ROE.
- A1: VREFL, tensión onda reflejada del sensor ROE.
- A2: VSMETER, tensión del sensor del S-meter
- A3: AinAux0, libre para otros usos. Aparece en el conector H7.

Los diodos Zener de 4.7v sirven para limitar la tensión de las entradas analógicas. Los ADS1115 son muy sensibles a estas sobretensiones.

**El módulo #2** tiene la dirección I2C 0x49 (hex) y se inserta en el zócalo H4.  
Como se ve, el terminal ADDR está conectado a +5V.



- A0: VTOT, medida de tensión de alimentación +12v.  
Esta tensión se toma a través de un divisor de tensión para ajustarla al rango que admite el ADS1115;



- A1: ITOT, intensidad de corriente medida por el sensor ACS712.
- A2: AinAux1, libre para otros usos. Aparece en el conector H7.
- A3: AinAux2, libre para otros usos. Aparece en el conector H7.

## Preparación de los módulos ADS1115

Para poder insertar correctamente los módulos en sus zócalos es necesario soldar los pines de diferente manera en cada uno.



Como se ve en la imagen, en el # los pines se sueldan por la parte INFERIOR, la que no tiene componentes. En el módulo #2, se sueldan por el lado SUPERIOR, el lado de los componentes.



Hay que asegurar que los pines correspondientes a V y GND quedan en el lado correcto, hacia el exterior de la PCB del Raduino32.

## S-METER / VTOT/ITOT

Para usar la función S-meter se necesita equipar el módulo #1 de ADSS115.  
Para usar la medida de Tensión e Intensidad de alimentación, se precisa equipar el módulo #2.

## SOFTWARE

El software está en constante desarrollo y puede haber nuevas funciones o cambios menores en las ya existentes.

Utilizo para el desarrollo el editor UECIDE (<https://uecide.org/>), más potente y rápido que el IDE de Arduino. No he probado a compilarlo con Arduino IDE.

He partido del código de KD8CEC, aunque he modificado muchas partes del mismo con el propósito de simplificarlo, eliminando las partes no usadas. Algunas funciones las he utilizado tal como estaban, otras las he modificado y en otros casos he utilizado las originales de VU2ESE.

El resultado es una mezcolanza de código propio y de unos y de otros. Si a alguien le resulta interesante me daré por satisfecho.

El código, también el fichero ***ubitx32\_20.bin***, puede encontrarse en  
<https://github.com/ramonlh/ubitx32>

Para la programación del módulo ESP32 directamente con el fichero **.bin** puede usarse *Flash Download Tools de Espressif* : <https://www.espressif.com/en/support/download/other-tools>

### **INSTALACIÓN DE UECIDE para ubitx32\_20**

## DESCARGA

Descargar Uecide desde <https://uecide.org/download> (última versión 0.11.10)

Si es la primera vez, descargar la versión “full”.

Instalar con las opciones por defecto.

## JAVA

Si no está instalado Java Runtime, ir a <https://java.com/es/download/> e instalarlo.

[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 [jdk.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

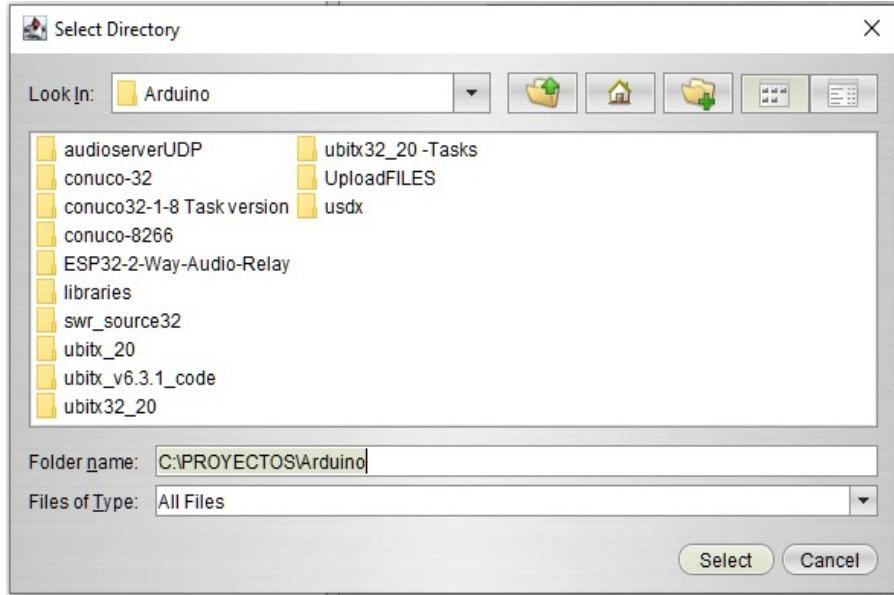
Una vez instalado Java Runtime, ejecutar de nuevo Uecide.

## INSTALAR PLACA

Ir a Tools->Plugin manager->Boards

Buscar “**ESP32**”, instalar la placa **esp32doit-devkit-v1**

Ir a File->Preferences->Locations: poner “SketchBook location” a la carpeta donde están los proyectos Arduino, por ejemplo;



## INSTALAR LIBRERÍAS

Descargar las librerías desde <https://github.com/ramonlh/ubitx32/tree/master/libraries> en una **carpeta** a nuestro elección.

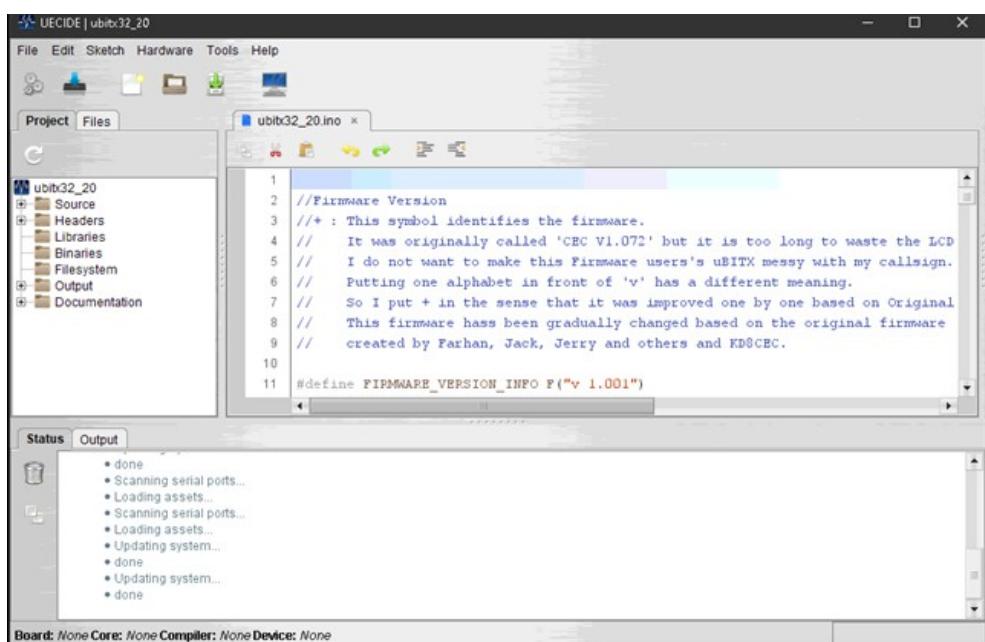
Ir a File->Preferences->Locations, click “**Add Location**”, seleccionar la carpeta en la que hemos descargado las librerías.

	Adafruit_ADS1015.zip
	DallasTemperature.zip
	ESP32PWM.zip
	NTPClient.rar
	OneWire.rar
	PubSubClient.rar
	RemoteDebug.rar
	RemoteDebug.zip
	TFT_eSPI.zip
	WebServer.zip
	esp8266FTPServer.zip

Descomprimir todas las librerías, cada una en su carpeta.

**Después de añadir las librerías es necesario reiniciar Uecide.**

## ABRIR PROYECTO “ubitx32\_20”.



Seleccionar la placa: **Hardware->Board->ESP32->Doit.....**

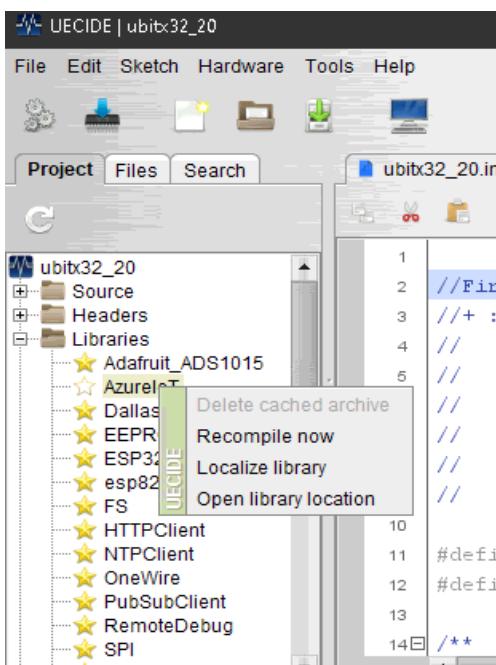
Seleccionar puerto serie: **Hardware->Devices->COMXX**

## Compile ubitx32\_20

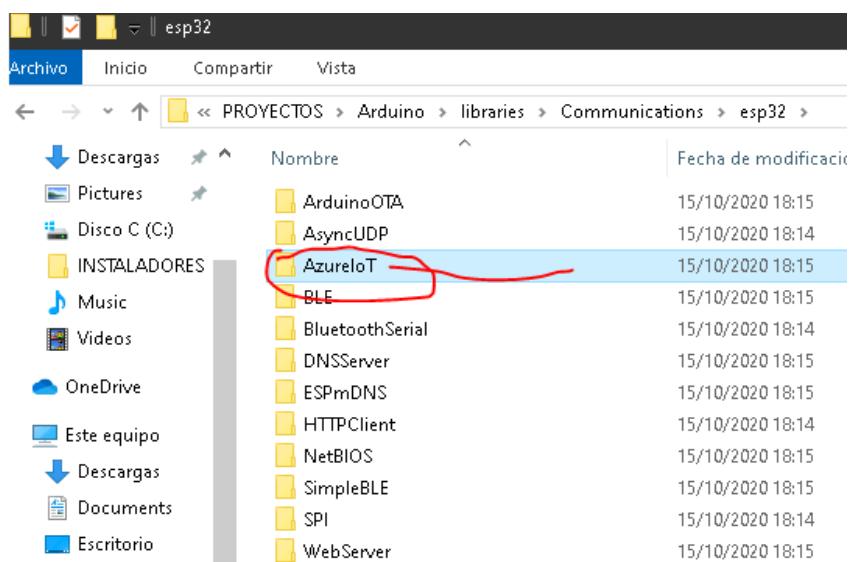
Abrir proyecto: **File->Open...ubitx32\_20**

Por alguna razón, la primera vez se añade al proyecto la librería **AzureIoT** y se produce un error. Es necesario eliminarla del proyecto:

Click sobre el botón derecho y “Open library location”.



Se abrirá una ventana del explorador de archivos:



Borrar la carpeta **AzureIoT**.

**Cerrar y volver a abrir Uecide.** La librería ya no estará y compilará correctamente.

## MANEJO

La pantalla principal tiene este aspecto

### Botones

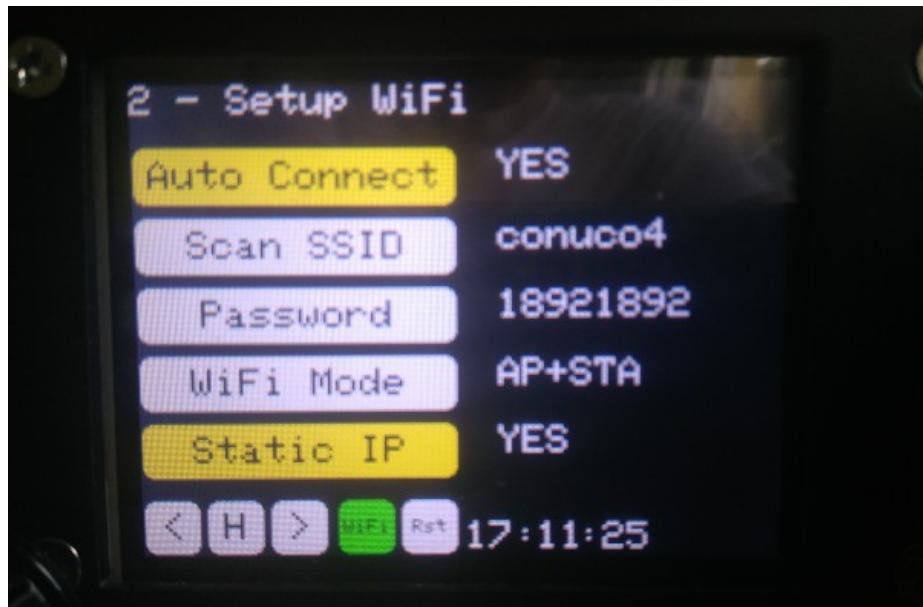


## Configuración

Pulsando el botón XXXX se avanza por las páginas de configuración

Página “1 – Setup”

En esta página se pueden editar Idioma, Distintivo, Latitud y Longitud pulsando en el botón correspondiente.



## Página “2 – Setup WiFi”

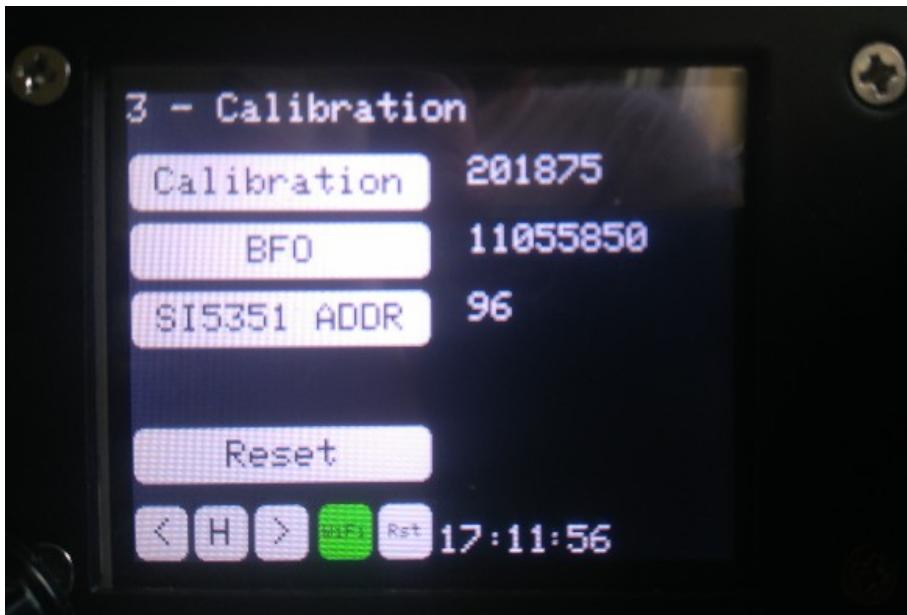
Opciones:

- “**Auto Connect**”: si está activado, el equipo intenta conectarse a la red definida cuando arranca.
- “**Scan SSID**”: explora las redes WiFi para seleccionar a la que nos queremos conectar.
- “**Password**”: Contraseña de la red WiFi a la que nos queremos conectar.
- “**WiFi Mode**”: el ESP32 puede funcionar de tres modos. Modo STA, como estación para conectarse a un AP. Modo AP, como punto de acceso para conectarnos desde otro equipo y modo AP+STA como ambos a la vez.

El modo AP podremos conectarnos con el PC u otro equipo y el ESP32 nos proporcionará una dirección del tipo 192.168.4.xx. El SSID será UBITX y la dirección IP de la radio será 192.168.4.1.

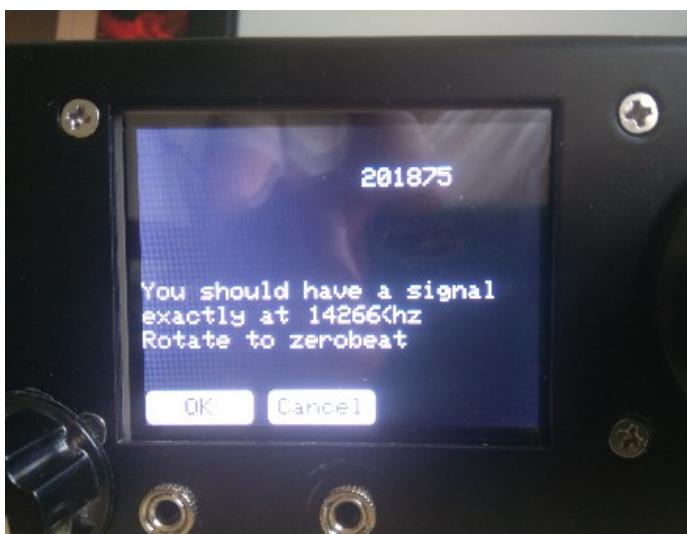
- “**Static IP**”: si está activado la dirección IP del equipo en modo STA será fija (no DHCP) y será 192.168.1.149. (Pendiente poder cambiar).

### Página “3 - Calibration”



En esta página se tiene acceso a los procedimientos de calibración inicial de la radio. Dirigirse al manual del equipo original ya que el procedimiento es el mismo.

Después de pulsar el botón “Calibration” aparece esta pantalla



Girando el mando del dial buscaremos el valor adecuado y lo aceptaremos pulsando el mismo mando o pulsando el botón “OK”.

**Este procedimiento es igual en todos aquellos casos en que hay que seleccionar un valor.**

También puede encontrarse mucha ayuda en:

Grupo bitx20: [groups.io/g/bitx20](https://groups.io/g/bitx20)

Página <https://ubitx.net/>

Página “4 – CW setting”



## Acceso al equipo mediante navegador

Si está activada la IP estática, la dirección será <http://192.168.1.149:88/>

### El panel principal

Panel	Configuración	Sistema
-------	---------------	---------

uBitx		
VFO	A	
Mode	USB	
CW	OFF	
RIT	OFF	
Split	OFF	
Freq. A	14266100	
Freq. B	7131000	
Freq. actual	14266100	
Band 160 m	1989600	1999900
Band 80 m	3769900	3799000
Band 60 m	5366200	5366900

### Pantalla “Configuración”

Panel	Configuración	Sistema		
Dispositivo	Bandas	Red	Serv. Red	Local I/O

CALLSIGN	EA4GZI	
Calibration/usbCarrier	201875	11055850
Latitud/ Longitud	112345.000000	123344.000000
Idioma	Español ▼	
CW Speed	60	
cwDelayTime	60	
cwKeyType	iambica ▼	
SI5351BX_ADDR	96	
Reset periodico (horas)	1 ▼	
Guardar		

## Pantalla “Bandas”

Desde esta pantalla se pueden redefinir los límites de cada banda si es necesario.

Panel	Configuración	Sistema
Dispositivo	<b>Bandas</b>	Red
	Serv. Red	Local I/O

Banda	desde	hasta khz
160 m	1810	2000
80 m	3500	3800
60 m	5351	5367
40 m	7000	7300
30 m	10100	10150
20 m	14000	14350
17 m	18068	18168
15 m	21000	21450
12 m	24890	24999
10 m	28000	29700

## Pantalla “Red”

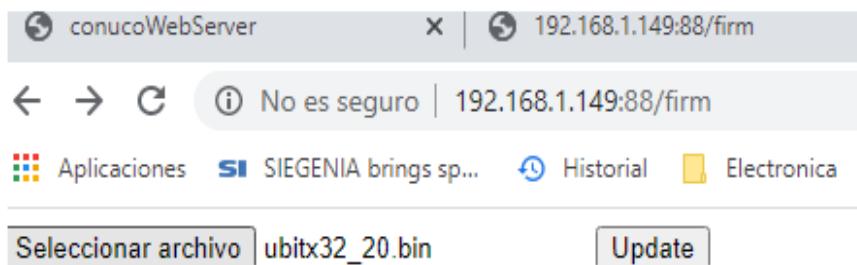
Acceso a los parámetros completos de la conexión WiFi

Panel	Configuración	Sistema
Dispositivo	<b>Bandas</b>	<b>Red</b>
	Serv. Red	Local I/O

Auto WiFi	<input checked="" type="checkbox"/>
Modo	AP+STA ▾
STA mode SSID	conuco4 <input type="button" value="Scan"/>
STA mode Password	[REDACTED]
AP mode SSID	UBITX
AP mode Password	12341234
Canal	3 ▾
MAC	
IP estática	<input checked="" type="checkbox"/>
Dir. IP	192.168.1.149
IP	192 168 1 149
Mask	255 255 255 0
Gateway	192 168 1 1
DNS	8 8 8 8
IP pública	185.178.80.2

## Pantalla “Actualizar”

Desde aquí puede actualizarse el firmware si se dispone del fichero “**UBIT32\_20.BIN**” una vez compilado.



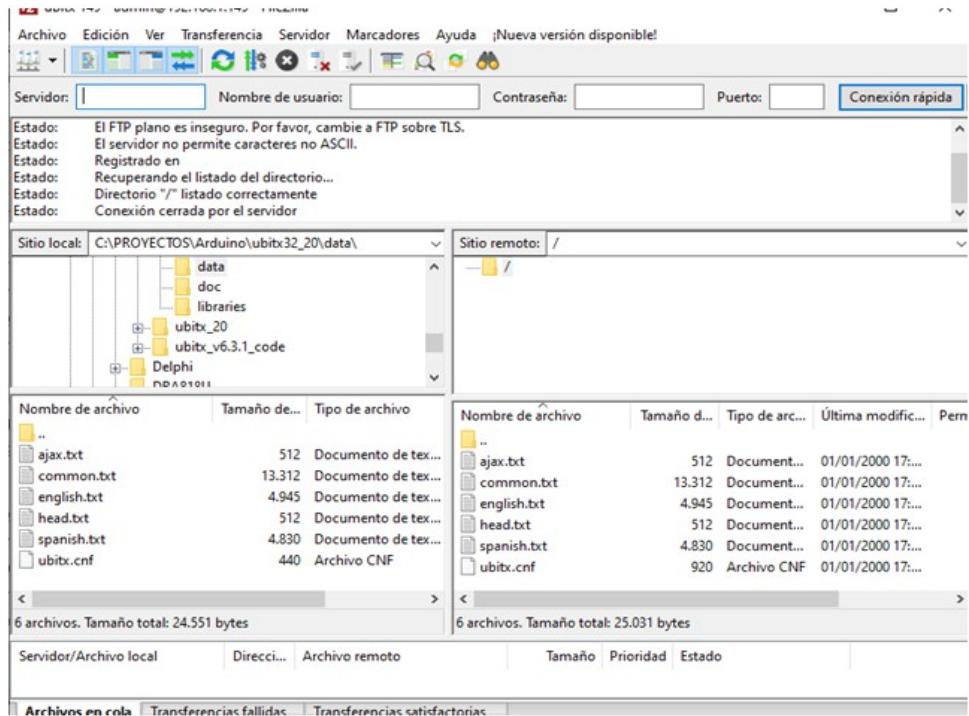
Si todo funciona como debe, se actualizará el firmware con el nuevo fichero y el equipo se reiniciará.

**Hay otras pantallas que no se describen porque están aún en desarrollo o no están probadas.**

## ACCESO FTP

El acceso mediante FTP permite gestionar los ficheros del sistema, hacer copia de seguridad o añadir nuevos ficheros de idioma.

Por el momento, la contraseña de acceso FTP es “admin/admin”. Puede desactivarse el servidor FTP des la página web.

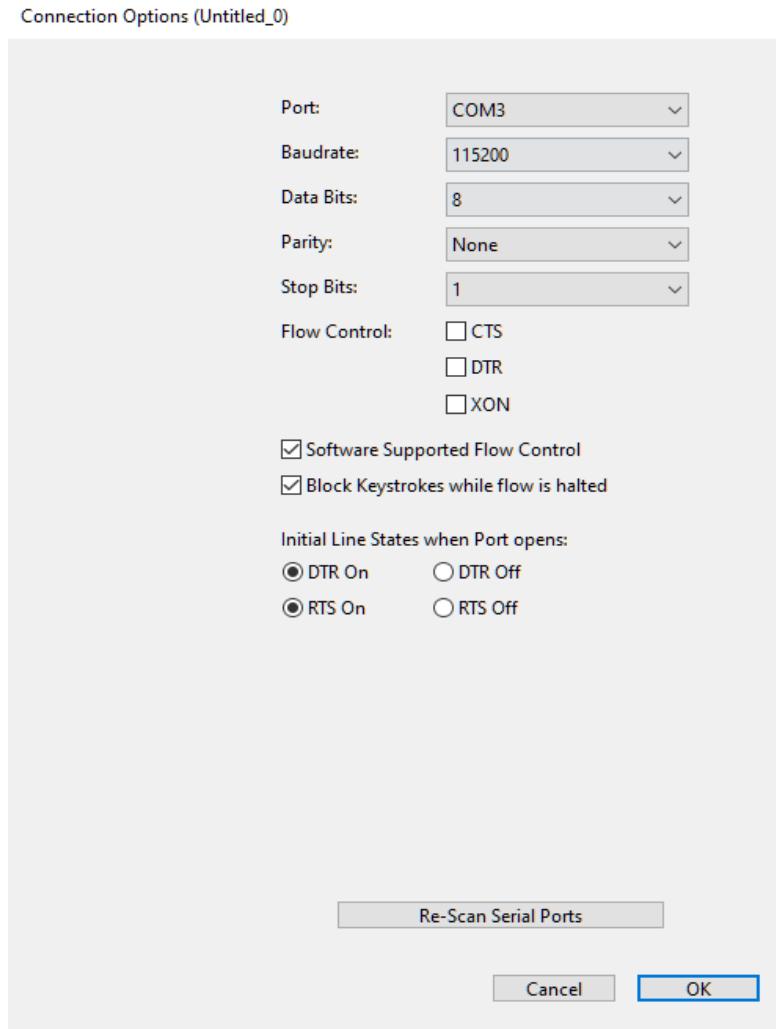


## USO DEL PUERTO SERIE 2

El puerto serie 2 puede usarse para depuración y para efectuar una configuración básica del Raduino32.

Yo utilizo el programa terminal **CoolTerm** que puede conseguirse aquí: <http://freeware.the-meiers.org>. Gracias desde aquí a Roger.

Seleccionar el puerto serie COMxx y configurar el terminal con los siguiente parámetros:



### Pulsar Connect

Encender la radio uBitx o pulsar Reset en la placa ESP32. Veremos la siguiente información al arrancar la radio (puede variar según las opciones o la versión):

```
===== 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

```

---

En otro apartado se describirá la información de arranque.

Puede interactuarse con el Raduino32 por este puerto serie con una serie de comandos. Para verlos escribir “*h*” y pulsar *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)

---

## Comandos:

<b>u → Enter</b>	Muestra los parámetros actuales
	----- 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
<b>f → Enter</b>	Verifica si existen los ficheros necesarios.
	/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
<b>F → Enter</b>	Reset de fábrica.
<b>H → Enter</b>	Muestra comandos disponibles
<b>r → Enter</b>	Reset
<b>t → Enter</b>	Formatea sistema de archivos SPFFS
<b>p,SSID → Enter</b>	Define la SSID a la que se debe conectar
<b>s,PASS → Enter</b>	Define la contraseña de la SSID anterior
<b>S,SSID → Enter</b>	Define el nombre de la SSID en modo AP
<b>P,PASS → Enter</b>	Define la contraseña para el modo AP
<b>w,x → Enter</b>	Define el modo Wifi. 0:OFF, 1:STA, 2:AP, 3:AP+STA
<b>a → Enter</b>	Cambia el estado del valor Auto WiFi. Si es YES, y el modo WiFi es igual 1 ó 3, el Raduino32 intentará conectarse a la red definida por SSIDname/SSIDpass.

## Flash Download Tools for ESP32/Esp8266

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

Found 3 results

### Flash Download Tools

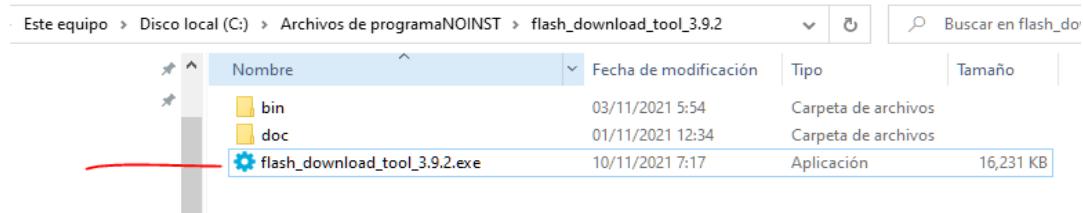
Title	Platform	Version	Release Date	Download
+ Flash Download Tools	Windows PC	V3.9.2	2021.11.10	<a href="#">Download</a>

### Certification and Test

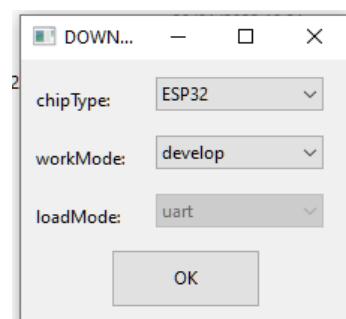
Title	Platform	Version	Release Date	Download
+ ESP RF Test Tool and Test Guide	ZIP	V2.8	2021.11.10	<a href="#">Download</a>
+ ESP8266 & ESP32 WFA Certification and Test Guide	Windows PC	V1.1	2020.08.05	<a href="#">Download</a>

[Download selected](#)

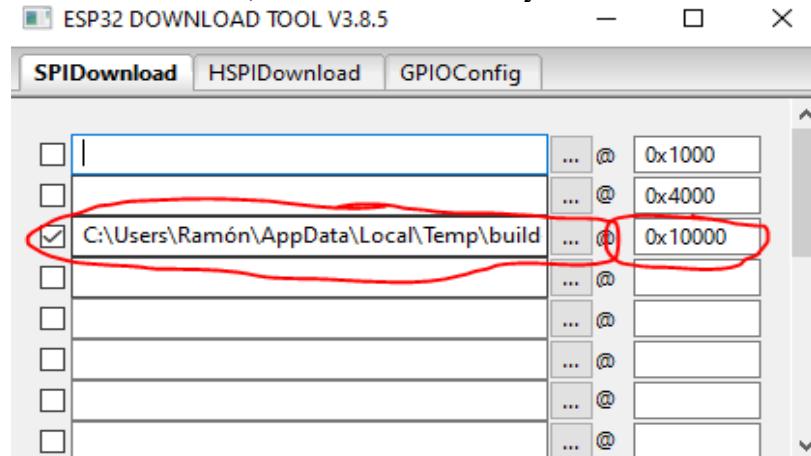
- Unzip folder (no needed installation)
- Run: `flash_download_tool_3.9.2`



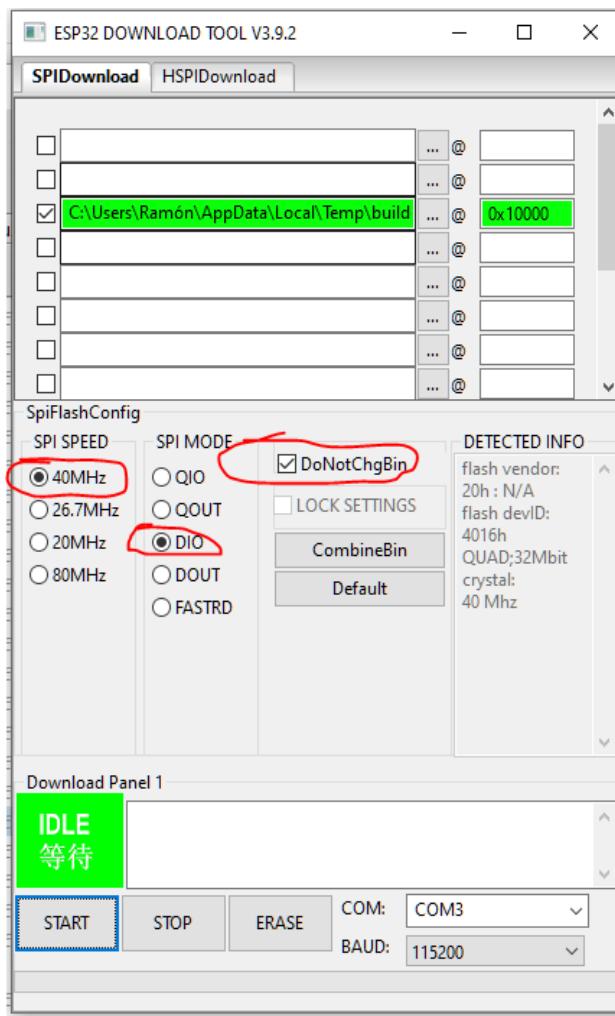
- Be patient, it takes a minute.
- Select “ESP32” and “Developer Mode”.



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

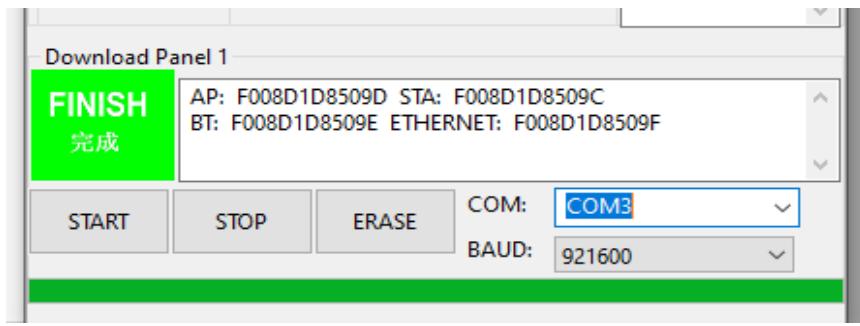


- Check options:



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





- **Reset uBitx.**