

# ORANGE PI ZERO

## MMDVM RF7021SE STM32F103C8T6

Este documento pretende describir la puesta en marcha de un sistema DMO completo, basado en la ORANGE PI ZERO y el MMDVM\_HS + MMDVM\_HOST.

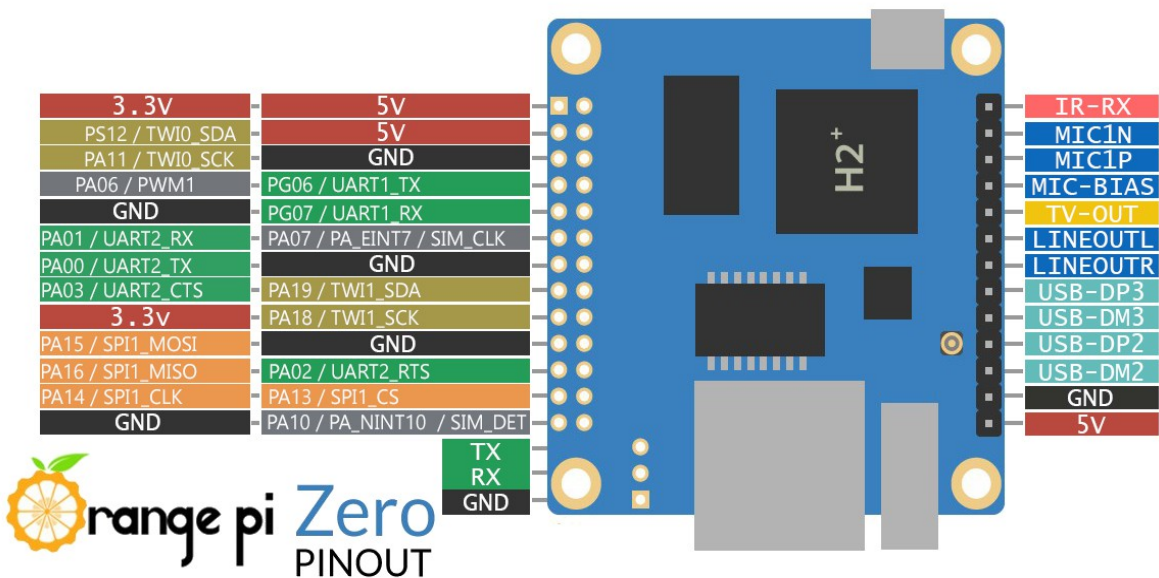
Página de referencia SOFTWARE

[https://github.com/juribeparada/MMDVM\\_HS](https://github.com/juribeparada/MMDVM_HS)

<https://github.com/g4klx/MMDVMHost>

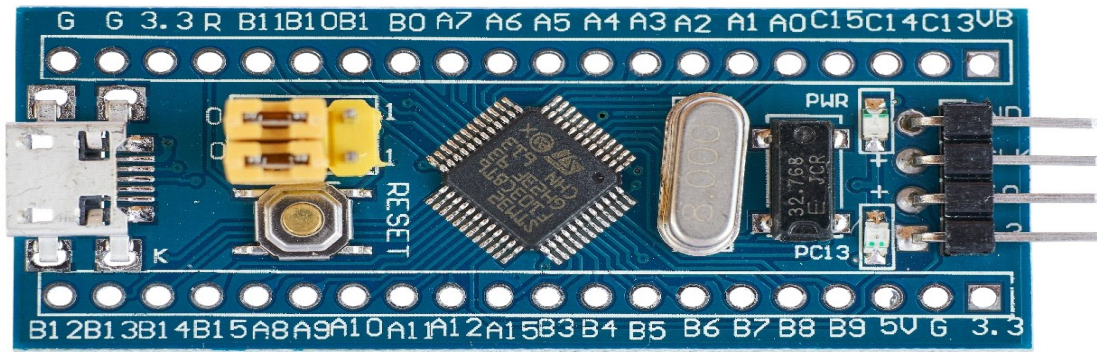
●Atención especial al fichero: BUILD.TXT

[https://github.com/juribeparada/MMDVM\\_HS/blob/master/BUILD.md](https://github.com/juribeparada/MMDVM_HS/blob/master/BUILD.md)



**DIAGRAMAS**

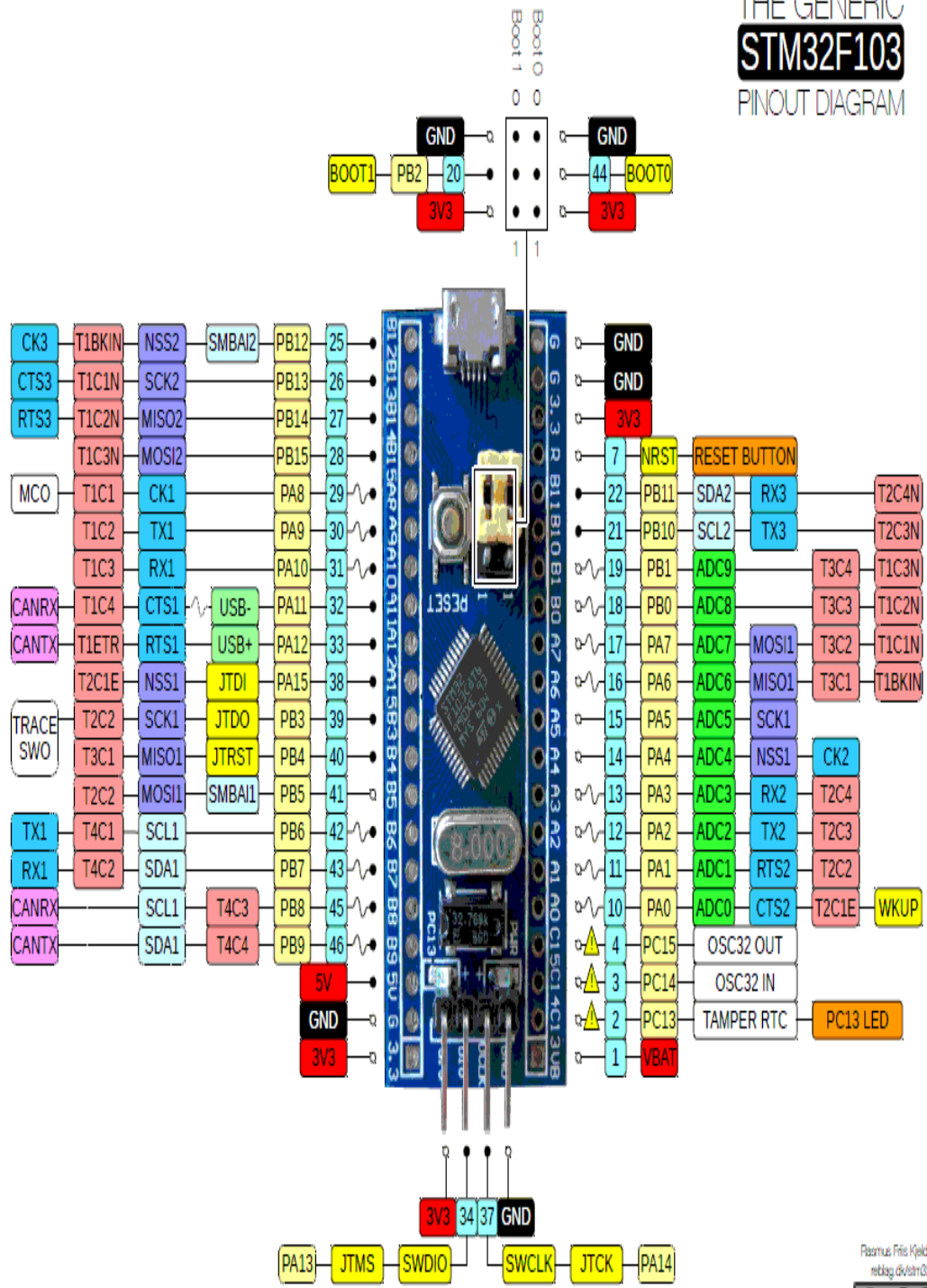
En el citado gráfico se muestra las conexiones del STM.



# THE GENERIC STM32F103 PINOUT DIAGRAM

## LEGEND

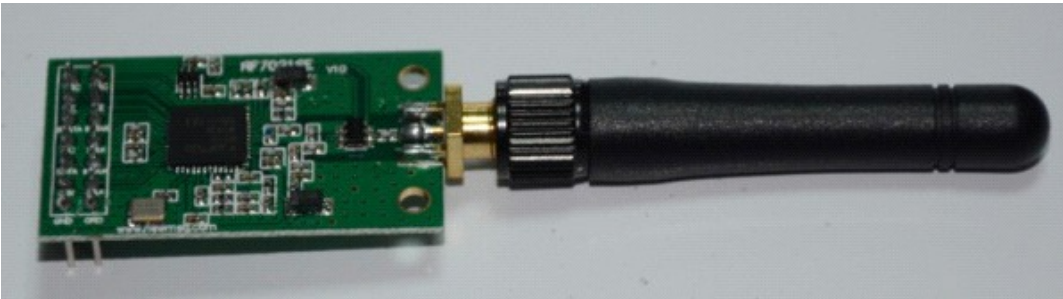
POWER
GROUND
PHYSICAL PIN
PIN NAME
CONTROL
ANALOG
TIMER & CHANNEL
USART
SPI
I2C
CAN BUS
USB
MISC
BOARD HARDWARE
● 5V tolerant
○ Not 5V tolerant
~ PWM pin
— Alternate function
⚠ PC13, PC14, PC15: Sink max 3mA, source 0mA, max 2mhz, max 30pF
Absolute MAX 150mA total source/sink for entire CPU
Max ±20mA per pin, ±8mA recommended



Reamus Frits Kjelksen  
reblog.dk/stm32  
CC BY SA  
V1.0

La sección de RF se basa en el circuito: [ADF7021](#)

<http://www.analog.com/media/en/technical-documentation/data-sheets/ADF7021.pdf>

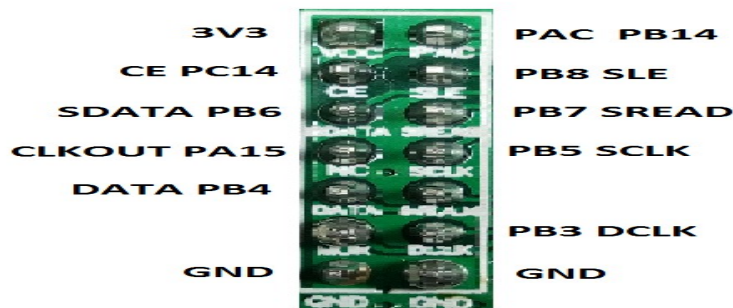


Como hemos indicado anteriormente, es necesario sustituir el TXCO por un valor de los contemplados en el fichero Config.h

- Frequency range 80Mhz - 650Mhz band.
- Transfer rate support 0.05kbps-32.8 kbps.
- Programmable output power from -16dBm to +13 dBm, stepping 0.3dBm.
- Power consumption : Transmit mode (10dBm) 29.2mA; receive mode 26.4mA.
- 2.2-3.6V supply voltage.

## SEÑALES UTILIZADAS

La comunicación entre el STM y el RR7021SE son las siguientes (módulo principal):



Las líneas TxRxDATA (PB4/PA4) y TxRxCLK (PB3/PA5) es conveniente añadir en serie una Resistencia de 10 Ohm.

Otras señales son:

### SERIAL

TXD – PA9 – SERIAL HOST / RXD – PA10 – SERIAL HOST

### NEXTION LCD

DISP\_TXD – PA2 NEXTION LCD / DISP\_RXD – PA3 NEXTION LCD

### LEDS

COS\_LED – PB15 / PTT\_LED – PB14 / P25\_LED – PB0 / YSF\_LED – PB1

DMR\_LED – PB13 / DSTAR\_LED – PB12

### STATUS

PIN\_LED – PC13 STATUS / PIN\_DEB – PB9 DEBUGGING

## MATERIAL NECESARIO

### **ORANGE PI ZERO – 512 mbyte**

<https://es.aliexpress.com/item/New-Orange-Pi-Zero-H2-Quad-Core-Open-source-512MB-development-board-beyond-Raspberry-Pi/32761500374.html?spm=a2g0s.9042311.0.0.yGlzRd>

### **RF7021SE ADF7021**

<https://es.aliexpress.com/item/Rf7021se-adf7021-wireless-module-wireless-module/1331907442.html?spm=2114.13010608.0.0.RZlWoz>

### **POWER MODULE**

<https://es.aliexpress.com/item/15924-Free-shipping-DC-DC-Step-Down-Converter-Module-LM2596-DC-4-0-40-to-1/32354635261.html?spm=a2g0s.9042311.0.0.TlxS1h>

### **STM32F103C8T6 STM32**

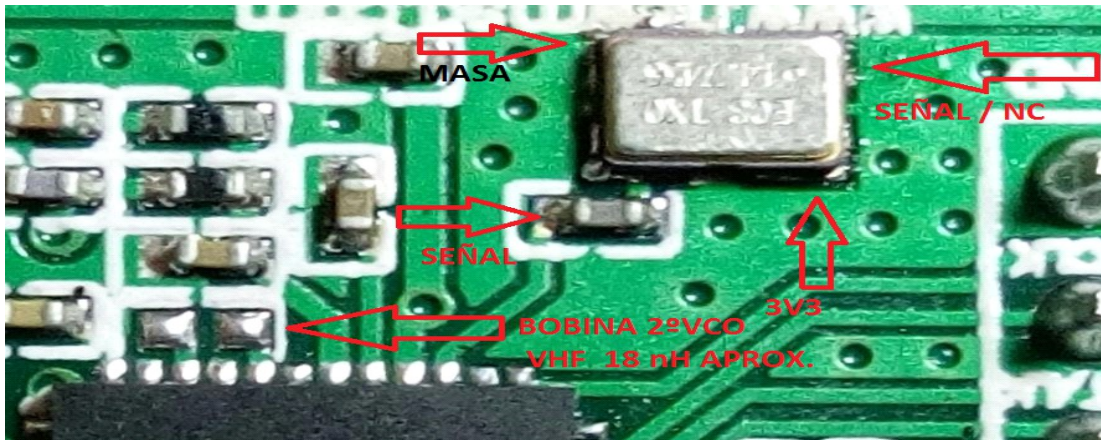
<https://es.aliexpress.com/item/1pcs-STM32F103C8T6-ARM-STM32-Minimum-System-Development-Board-Module-For-arduino/32583160323.html?spm=2114.13010608.0.0.RZlWoz>

**Es necesario sustituir el TCXO por uno de los siguientes por un valor de 14.7456 o 12.2888 Mhz**

**N.º Ref. Mouser: 520-TXO-3225-14.74T / Osciladores TCXO 14.7456MHz  
3.3V HCMOS 2.5ppm**

**N.º Ref. Mouser: 520-TXO-3225-12.288T / Osciladores TCXO 12.288MHz  
3.3V HCMOS 2.5ppm**

**Tamaño 3.2 mm x 2.5 mm**



Marca de montaje, punto o chaflanado es el terminal NC (no conectado).



Vamos a instalar las herramientas necesarias:

Abrir ventana terminal:

```
cd ~
```

- Instalación de las herramientas de programación

Install the necessary software tools:

```
sudo apt-get update
sudo apt-get install gcc-arm-none-eabi gdb-arm-none-eabi
libstdc++-arm-none-eabi-newlib libnewlib-arm-none-eabi
```

- Instalación código fuente y librerías

Download the sources:

```
cd ~
git clone https://github.com/juribeparada/MMDVM_HS
cd MMDVM_HS/
git clone https://github.com/juribeparada/STM32F10X\_Lib
```

- Editar Config.h

Edit Config.h:

```
nano Config.h
#define ADF7021_CARRIER_BOARD

#define ENABLE_ADF7021

#define ADF7021_14_7456

#define STM32_USART1_HOST

#define ENABLE_SCAN_MODE
```

- Construir el firmware

```
make clean
```

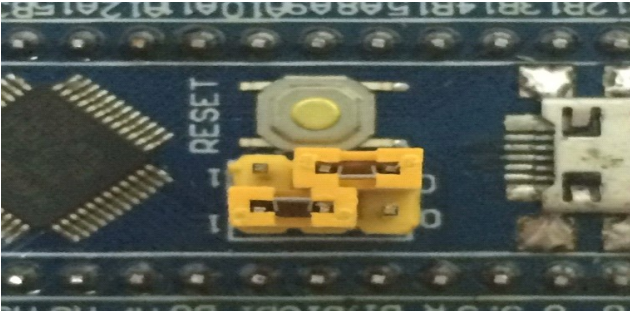


**make**

**Mover BOOT0 a 1, presionar RESET y ejecutar la orden** (Move BOOT0 jumper to 1, next press and release RESET and execute)

**sudo make serial devser=/dev/ttyS1**

**Suponemos que lo tenemos asignado al ttyS1 (ls /dev/ - podemos ver los dispositivos)**



```
pi@raspberrypi: ~/MMDVM_HS
Archivo  Editar  Pestañas  Ayuda
DMRDMOTX.o  LICENCE  SerialRB.cpp
pi@raspberrypi:~/MMDVM_HS $ sudo make serial devser=/dev/ttyUSB0
./STM32F10X_Lib/utils/rpi32/stm32flash -v -w bin/outp.bin -g 0x0 /dev/ttyUSB0
stm32flash Arduino_STM32_0.9

http://github.com/rogerclarkmelbourne/arduino_stm32

Using Parser : Raw BINARY
Interface serial_posix: 57600 8E1
Version      : 0x22
Option 1    : 0x00
Option 2    : 0x00
Device ID   : 0x0410 (Medium-density)
- RAM       : 20KiB (512b reserved by bootloader)
- Flash     : 128KiB (sector size: 4x1024)
- Option RAM : 16b
- System RAM : 2KiB
Write to memory
Erasing memory
Wrote and verified address 0x0800b1f4 (100.00%) Done.
Starting execution at address 0x08000000... done.
pi@raspberrypi:~/MMDVM_HS $
```

Edición de los parámetros de funcionamiento:

**Desde la ventana de terminal:**

**cd ~**

**cd MMDVM\_HS/**

**nano Config.h**

**#define ENABLE\_ADF7021**

**#define BIDIR\_DATA\_PIN**

**#define ADF7021\_CARRIER\_BOARD – Placa por defecto con la que trabajamos**

**#define ADF7021\_14\_7456 – Valor TCXO**

**#define STM32\_USART1\_HOST – Para usar mediante SERIAL TTL ADAPTER, AMA0 (PINES GPIO RPi).**

**#define STM32\_USB\_HOST – Para usar mediante el USB, ACM0**

**#define ENABLE\_SCAN\_MODE – Activa el rastreo de modos**

**#define SEND\_RSSI\_DATA - Permite el envío de la señal RSSI**

**#define SERIAL\_REPEATER – Salida de datos para pantalla NEXTION**

**make clean – borra lo compilado anteriormente**

**make – genera fichero HEX**

**make bl – genera fichero HEX con BOOTLOADER**

**make zumspot-pi – serial conectado al USART1**

**make serial devser=/dev/ttyXXX – serial conectado a USB HOST**

**make serial-BL devser=/dev/ttyXXX – igual con carga del BOOTLOADER**

**make dfu devser=/dev/ttyACM0 – carga desde ACM0, requiere BOOTLOADER**

## CONEXIONADO PINES DE LA PLACA

La placa dispone de una serie de pines de conexión, que son los siguiente:

- Conector DC POWER (DC POWER JACK 2.5mm, + CENTRAL), POSITIVO CENTRAL

- USART:

- **ttyS0** - consola, sin uso

- **ttyS1** - Conexión al STM - MMDVM.INI

- **ttyS2** - Conexión para Nextion - MMDVM.INI

- CONECTOR POWER 5V – permite seleccionar desconectar la tensión de 5V de alimentación, queda asilada la parte del STM y RF.

- CONECTOR I2C GPIO – CONECTOR PANTALLA GPIO -, para conectar pantalla I2C directamente al GPIO

- PANTALLA Nextion puede trabajar mediante el ttyS2 o conectada a los pines directamente del STM, activando en el fichero Config.h mediante la línea de comando

**#define SERIAL\_REPEATER – Salida de datos para pantalla NEXTION**

**Prestar atención al conexionado de la pantalla, es el siguiente:**

PLACA	GND	5V	RXD	TXD
NEXTION	NEGRO	ROJO	TXD	RXD

## CONFIGURACION PANTALLA I2C @M0IEI/EA1CRF

### UPDATE + UPGRADE

```
cd ~
```

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

### ADD I2C LCD

```
cd /etc
```

```
sudo nano modules
```

```
i2c-bcm2708
```

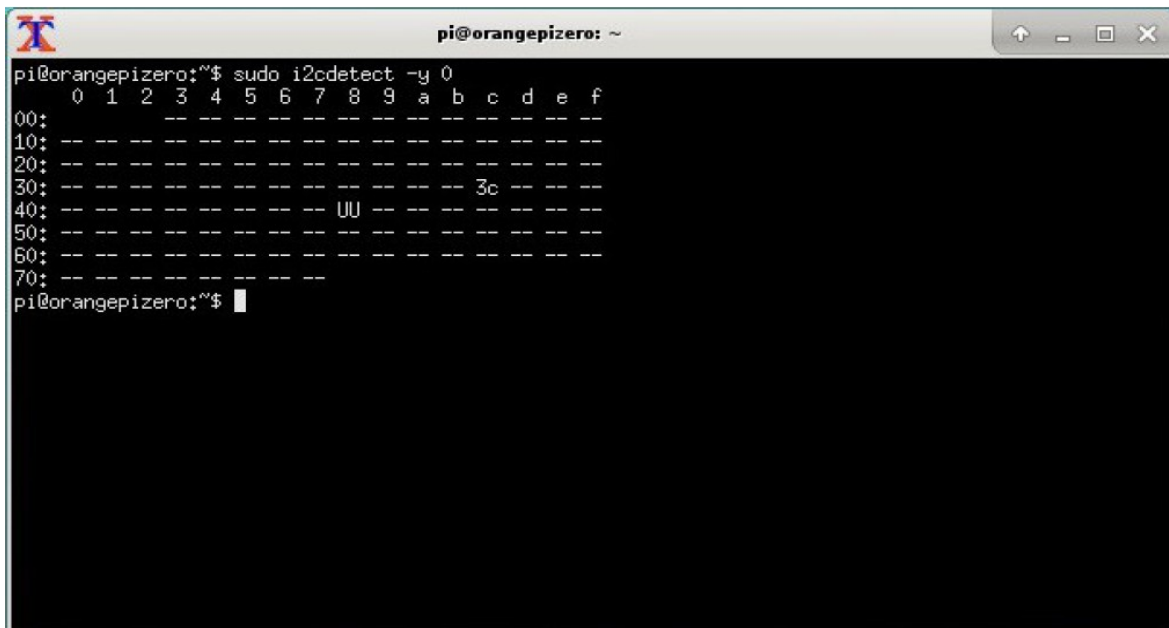
```
i2c-dev
```

### RESET

```
sudo apt-get install build-essential git-core libi2c-dev i2c-tools lm-sensors
```

### DETECT

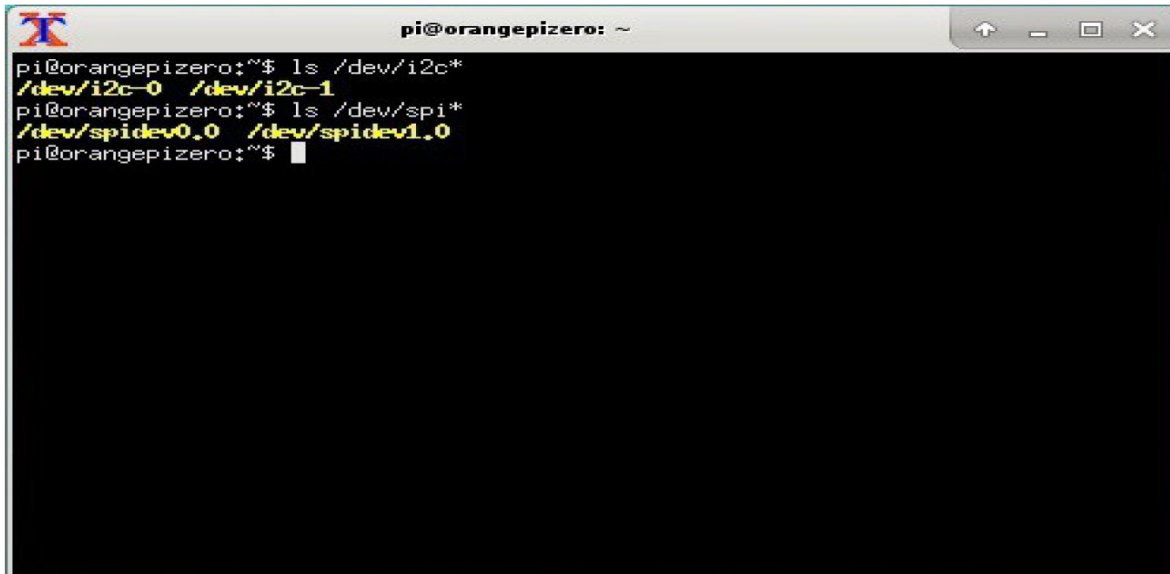
```
sudo i2cdetect -y 0
```



```
pi@orangezero: ~  
pi@orangezero:~$ sudo i2cdetect -y 0  
 0 1 2 3 4 5 6 7 8 9 a b c d e f  
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
30: -- -- -- -- -- -- -- -- -- -- 3c -- -- -- --  
40: -- -- -- -- -- -- UU -- -- -- -- -- -- -- --  
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  
pi@orangezero:~$
```

ls /dev/i2c\*

ls /dev/spi\*



```
pi@orangezero: ~  
pi@orangezero:~$ ls /dev/i2c*  
/dev/i2c-0 /dev/i2c-1  
pi@orangezero:~$ ls /dev/spi*  
/dev/spidev0.0 /dev/spidev1.0  
pi@orangezero:~$
```

## INSTALL

cd

git clone [https://github.com/hallard/ArduiPi\\_OLED](https://github.com/hallard/ArduiPi_OLED)

cd ArduiPi\_OLED

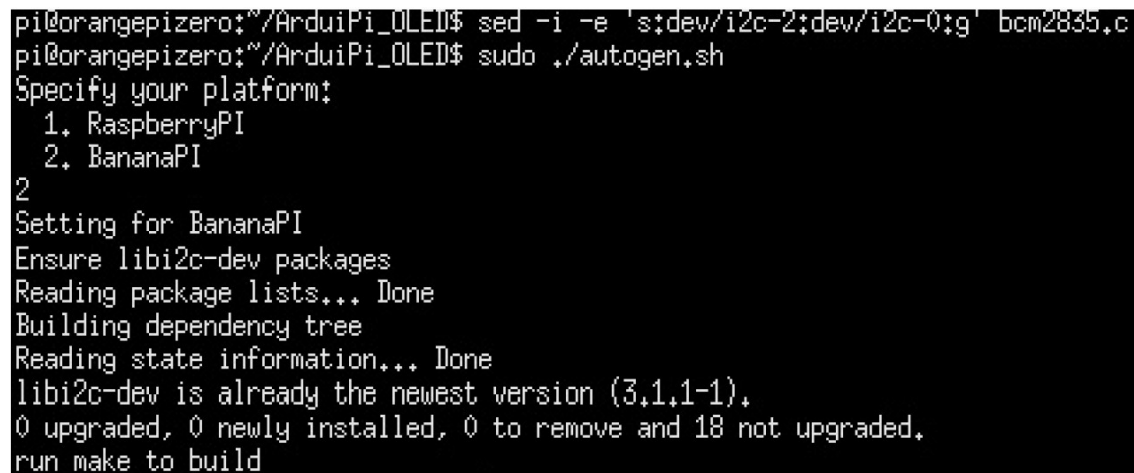
sed -i -e 's:dev/i2c-2:dev/i2c-0:g' bcm2835.c

sudo ./autogen.sh

choice "2 BananaPi"

sudo make

## COMPILE MMDVM

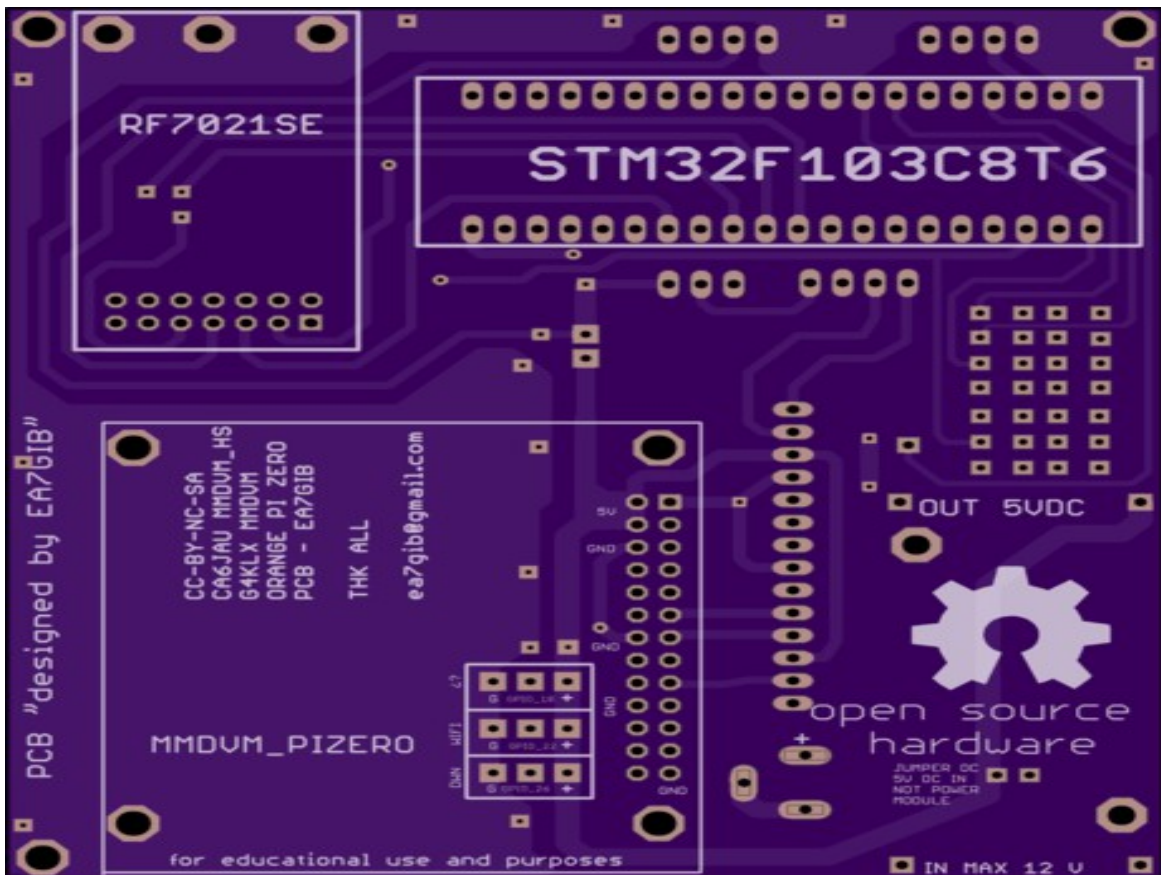
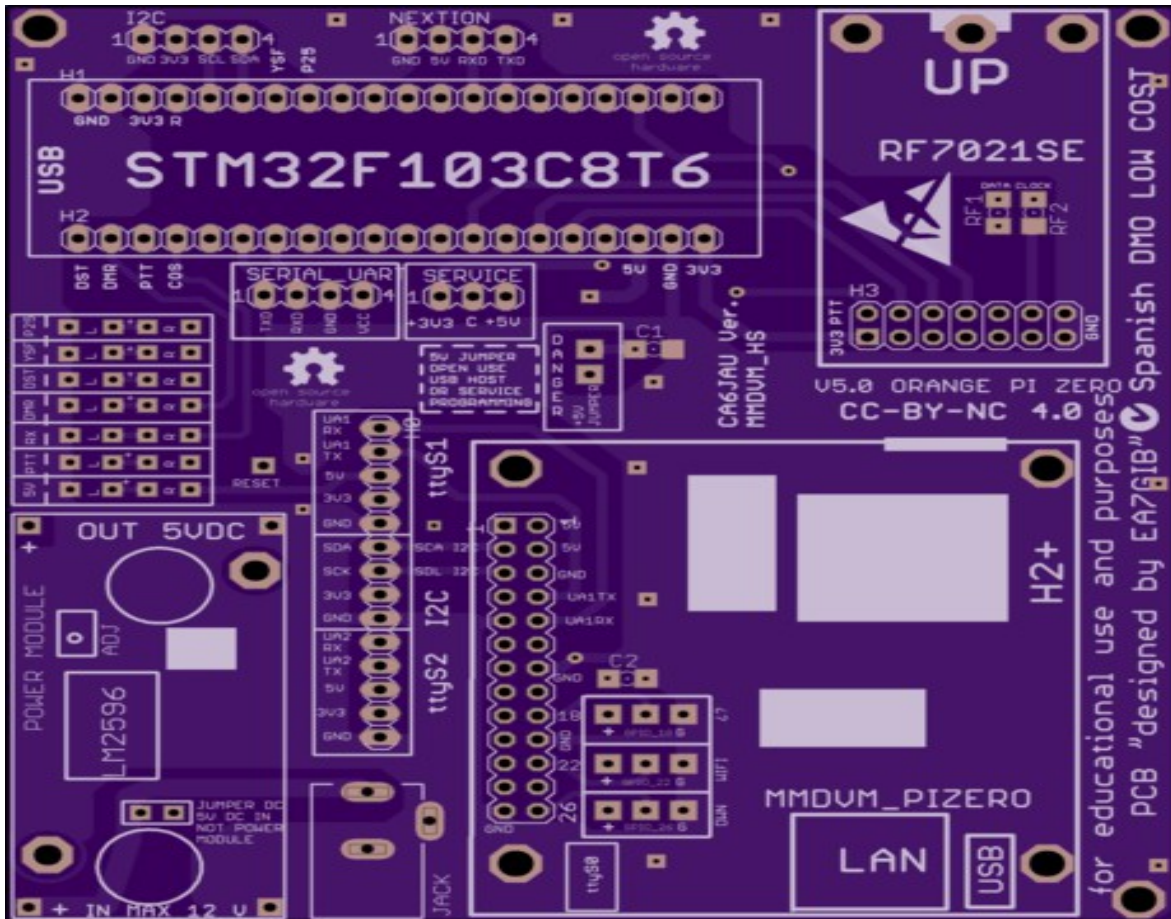


```
pi@orangezero:~/ArduiPi_OLED$ sed -i -e 's:dev/i2c-2:dev/i2c-0:g' bcm2835.c  
pi@orangezero:~/ArduiPi_OLED$ sudo ./autogen.sh  
Specify your platform:  
 1. RaspberryPI  
 2. BananaPI  
2  
Setting for BananaPI  
Ensure libi2c-dev packages  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
libi2c-dev is already the newest version (3.1.1-1).  
0 upgraded, 0 newly installed, 0 to remove and 18 not upgraded.  
run make to build
```

```
cd /home/pi/Applications/MMDVMHost/
```

```
sudo make clean
```

```
sudo make -f Makefile.Pi.OLED
```



ORANGE Pi Zero – MMDVM\_HS+MMDMV\_HOST – [ea7gib@gmail.com](mailto:ea7gib@gmail.com) V1.0

