

Instalación y configuración MMDVM_HS para Modo Duplex

```
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
```

```
cd ~
```

```
cd /home/pi
```

```
git clone https://github.com/juribeparada/MMDVM_HS
```

```
cd MMDVM_HS/
```

```
git clone https://github.com/juribeparada/STM32F10X_Lib
```

Editamos el Config.h

Sudo nano Config.h

Nos Saldrá algo así como esto:

Las lineas con # en primer lugar, son las que están activas.

```
// Select one board (STM32F103 based boards)
```

```
// 1) ZUMspot RPi or ZUMspot USB:
```

```
// #define ZUMSPOT_ADF7021
```

```
// 2) Libre Kit board or any homebrew hotspot with modified RF7021SE and Blue Pill  
STM32F103:
```

```
#define LIBRE_KIT_ADF7021
```

```
// 3) MMDVM_HS_Hat revisions 1.1, 1.2 and 1.4 (DB9MAT & DF2ET)
```

```
// #define MMDVM_HS_HAT_REV12
```

```
// 4) Nano hotSPOT (BI7JTA)
```

```
// #define NANO_HOTSPOT
```

```
// Enable ADF7021 support:
```

```
#define ENABLE_ADF7021
```

```
// Enable full duplex support with dual ADF7021 (valid for homebrew hotspots only):
```

```
// #define DUPLEX
```

```
// TCXO of the ADF7021
```

```

// For 14.7456 MHz:

#define ADF7021_14_7456

// For 12.2880 MHz:

// #define ADF7021_12_2880

// Host communication selection:

#define STM32_USART1_HOST

// #define STM32_USB_HOST

// Enable mode detection:

#define ENABLE_SCAN_MODE

// Send RSSI value:

#define SEND_RSSI_DATA

// Enable Nextion LCD serial port repeater on USART2 (ZUMspot Libre Kit and ZUMspot RPi):

// #define SERIAL_REPEATER

// Enable Nextion LCD serial port repeater on USART1 (Do not use with STM32_USART1_HOST
enabled):

// #define SERIAL_REPEATER_USART1

// Enable P25 Wide modulation:

// #define ENABLE_P25_WIDE

// Disable mode LEDs blink during scan mode:

#define QUIET_MODE_LEDS

Modificaremos las siguientes líneas:

Activaremos la línea DUPLEX (quitando las // que tiene delante #define DUPLEX

// Enable full duplex support with dual ADF7021 (valid for homebrew hotspots only):

#define DUPLEX

Activaremos modo STM32_USB_HOST (quitando las // que tiene delante) y desactivaremos el
modo STM32_USART1_HOST (poniendo // delante de #define STM32_USART1_HOST)

// Host communication selection:

// #define STM32_USART1_HOST

```

```
#define STM32_USB_HOST
```

Con estos cambios sería suficiente. Ahora si queremos podemos activar la pantalla Nextion y el modo RSSI (si no está activado, en este caso si lo está) en las siguientes líneas:

```
// Send RSSI value:
```

```
#define SEND_RSSI_DATA
```

```
// Enable Nextion LCD serial port repeater on USART2 (ZUMspot Libre Kit and ZUMspot RPi):
```

```
#define SERIAL_REPEATER
```

```
// Enable Nextion LCD serial port repeater on USART1 (Do not use with STM32_USART1_HOST enabled):
```

```
#define SERIAL_REPEATER_USART1
```

Una vez modificado todo guardamos el documento.

Ahora solo nos falta compilar con el comando `sudo make bl`.

si todo ha ido bien y no nos ha dado ningún error procederemos a cargar el firmware al STM.

Cogemos el conversor USB-TTL y lo conectaremos a los pines SERIAL correspondiente de la placa, (recuerda RX-TX cruzados).

Nos situamos en el carpeta MMDVM_HS, `cd /home/pi/MMDVM_HS/` cambiaremos de posición el jumper más alejado del RESET,

le pondremos de posición 0 a posición 1, seguidamente copiaremos el comando `sudo make serial-bl devser=/dev/ttyUSB0` en la consola(sin pulsar ENTER AUN), después pulsaremos el botón RESET y lo más rápido posible pulsamos ENTER, entonces empezará a programar el STM.

Cuando termine y si todo ha ido bien te dirá que está al 100%, cambiaremos el jumper movido a su posición original y ya tendremos programado el STM.

Si queremos actualizar el firmware no es necesario hacer todo este proceso de nuevo. Nos colocamos en la carpeta `cd /home/pi/MMDVM_HS/`, lanzamos el comando `sudo git pull`, si hay actualizaciones nos lo dirá. Procedemos a limpiar la carpeta con la comando `sudo make clean`, editamos el fichero `Config.h` por si necesitamos cambiar algo, `sudo nano Config.h`. Seguidamente compilaremos con el comando `sudo make bl`, si todo ha ido bien procederemos a cargar el firmware como hemos citado arriba.

EA4GAX - Sergio.