MMDVM-Nucleo rev. 1.0 board

- Designed for repeater and high power hotspot applications – connects to user supplied Raspberry Pi board and radio(s)
- Connects to STM32F446 MB1136 Nucleo Board
- Fifth generation analogue filter design that provides an extremely low BER compared to previous generations
- Two multi-turn pots for fine RX and TX adjustments
- Onboard LEDs to show status and modes (PTT, COS, Power, D-Star, DMR, P25, Fusion, NXDN and POCSAG)
- Onboard LED to show receive signal level clipping
- 8 pin JST Connector with pigtail wires
- Utilizes open source MMDVM software
- Supports NucleoTNC project

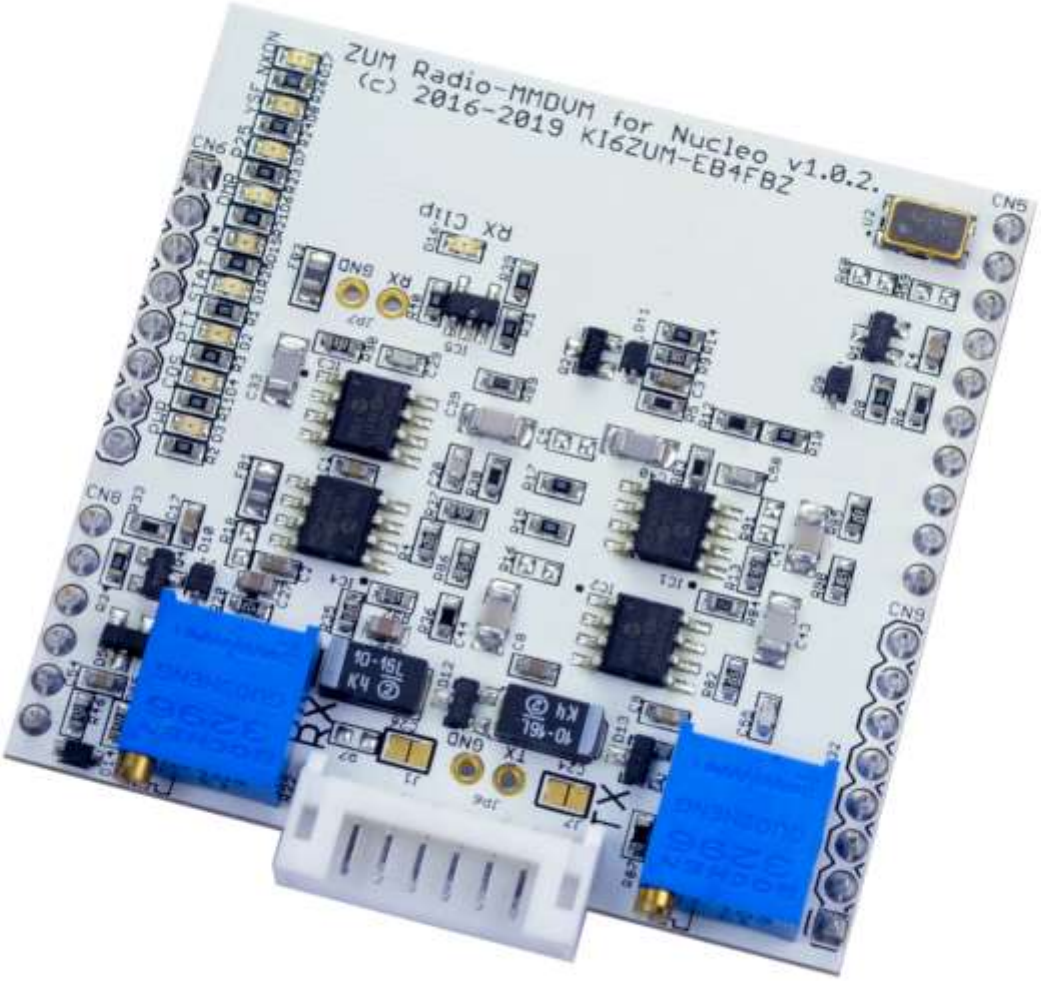
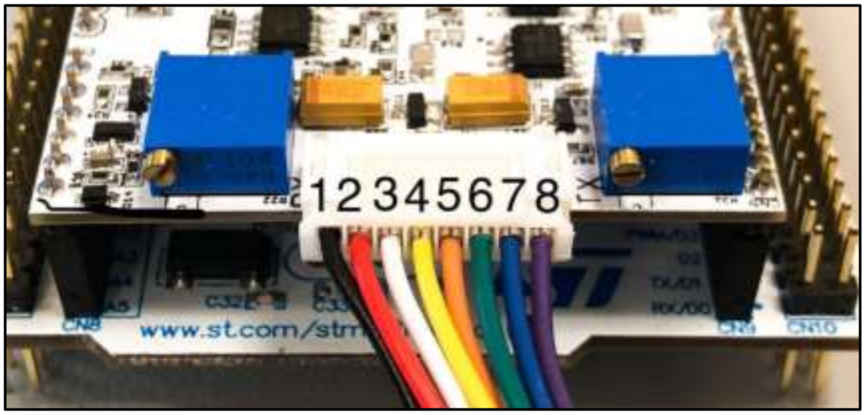

ZUM Radio-MMDVM for Nucleo v1.0.2.
(c) 2016-2019 KI6ZUM-EB4FBZ
RX Clip
GND
RX
CN5
CN6
CN8
CN9
RX
GND
TX
TX

Table of Contents:

Wiring connections

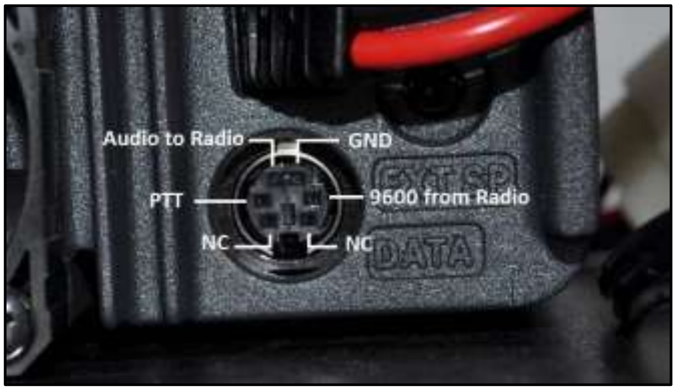
Here is a picture of the 8 pin header with cable showing wire colors:



Here is a table of the pin numbers, names and wire colors:

Pin number	Signal name	Description	Wire color
1	CTRL	Control (output)	Black
2	COS/STAT1	Carrier sense (input)	Red
3	RX audio	Receive audio from radio (input)	White
4	Ground	Signal ground	Yellow
5	Ground	Signal ground	Orange
6	TX audio	Transmit audio to radio (output)	Green
7	PTT	Push to talk (enable transmit) (input)	Blue
8	RSSI	Received signal strength indicator (input)	Purple

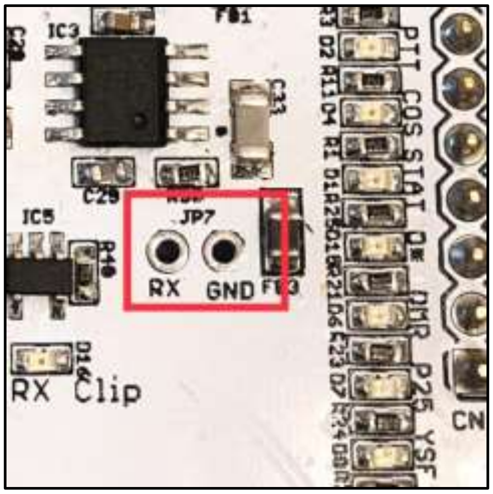
Here’s an example radio’s (FT-7900) pin connection via the mini DIN 6 connector on the rear of the radio:

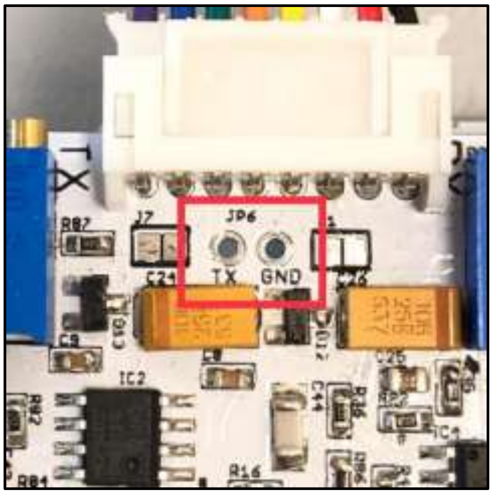


Pin number	Signal name	Wire color	Mini DIN 6 connector
3	RX audio	White	9600 from radio
4	Signal ground	Yellow	GND
6	TX audio	Green	Audio to radio
7	PTT	Blue	PTT

Adjusting signal levels (hardware)

Here is a picture of the test points on the PCB that can be used to look at the RX and TX signal levels on an oscilloscope. The signal on JP7 is the RX signal audio on the ADC input pin of the STM32 chip. JP6 has the TX signal audio on the pin going to the radio.

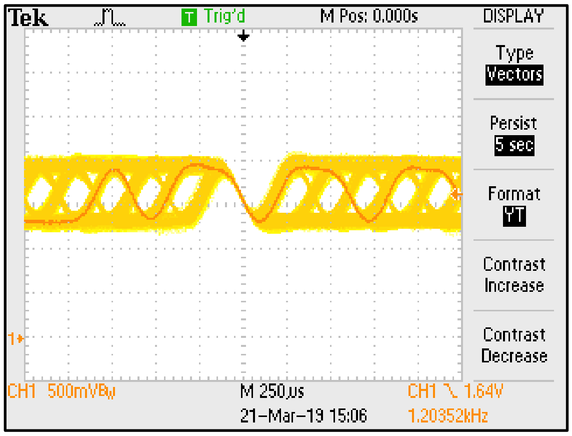




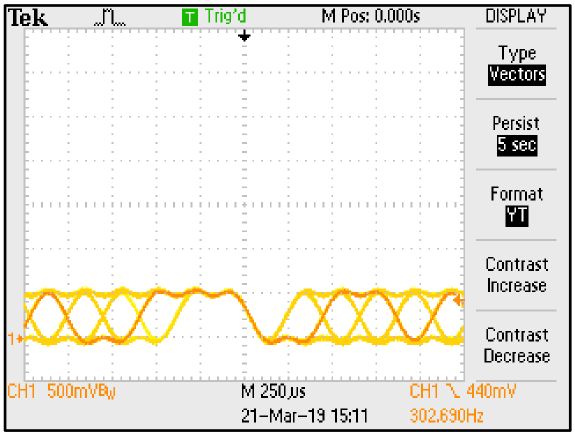
Here are some signal images captured on an Oscilloscope

RX-(JP7)

TX-(JP6)

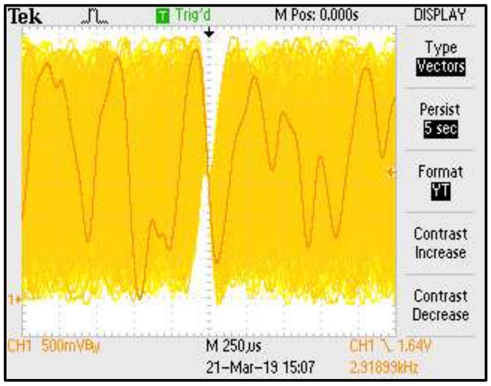


M Pos: 0.000s
Type Vectors
Persist 5 sec
Format YT
CH1 500mVBw
M 250µs
CH1 ↘ 1.64V
21-Mar-19 15:06
1.20352kHz



M Pos: 0.000s
Type Vectors
Persist 5 sec
Format YT
CH1 500mVBw
M 250µs
CH1 ↘ 440mV
21-Mar-19 15:11
302.690Hz

Noise without signal (JP7)



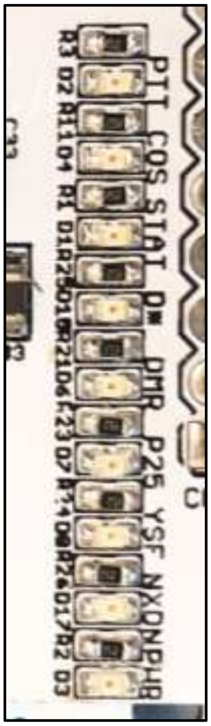
M Pos: 0.000s
Type Vectors
Persist 5 sec
Format YT
CH1 500mVBw
M 250µs
CH1 ↘ 1.64V
21-Mar-19 15:07
2.91899kHz

Here is a list of the jumpers on the board and their settings:

Jumper Description Table

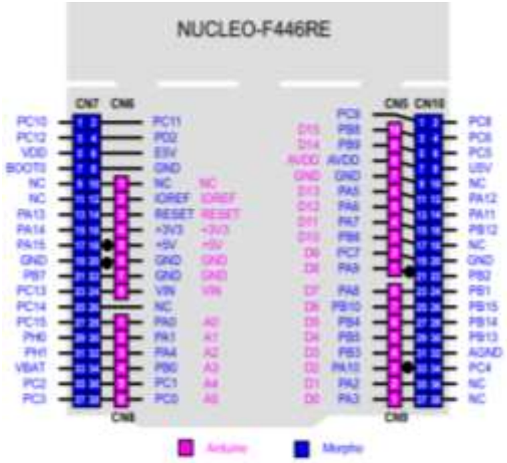
Bypass Pad	**ON**	**OFF**	**Description**
J1	X		Disable DC blocking capacitor on Radio RX audio
J1		X	Enable DC blocking capacitor on Radio RX audio
JP1	X		Bypass R7
JP1		X	Enable R7, this allows you to reduce the RX signal coming from the radio so it won't oversaturate and clip the signal.
J2	X		Bypass R32 (10K variable resistor)
J2		X	Enable R32 (10K variable resistor) for adjusting TX signal strength
J7	X		Disable DC blocking capacitor on Radio TX audio
J7		X	Enable DC blocking capacitor on Radio TX audio

LED Description Table

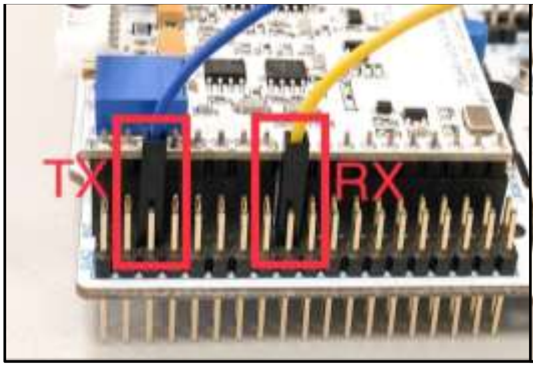
	LED NAME	Description
	PTT	Board is transmitting
	COS	Board is receiving
	STAT	Status
	NXDN	NXDN Enabled
	POCSAG	POCSAG Enabled
	D*	D-Star Enabled
	DMR	DMR Enabled
	P25	P25 Enabled
	YSF	YSF Enabled
	PWR	Board is receiving power

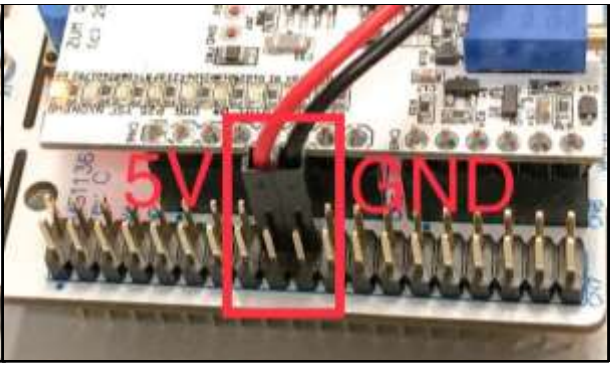
Adding an external Nextion display

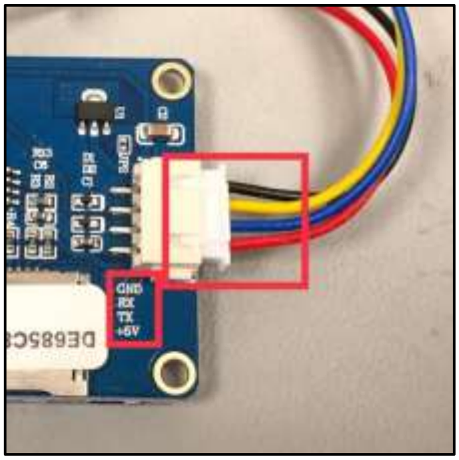
A NEXTION display can be used by connecting it to the Morpho headers on the Nucleo board.

	Nextion display pins	Nucleo pins
	5V	CN7-18
	GND	CN7-20
	RX	CN10-21
	TX	CN10-33

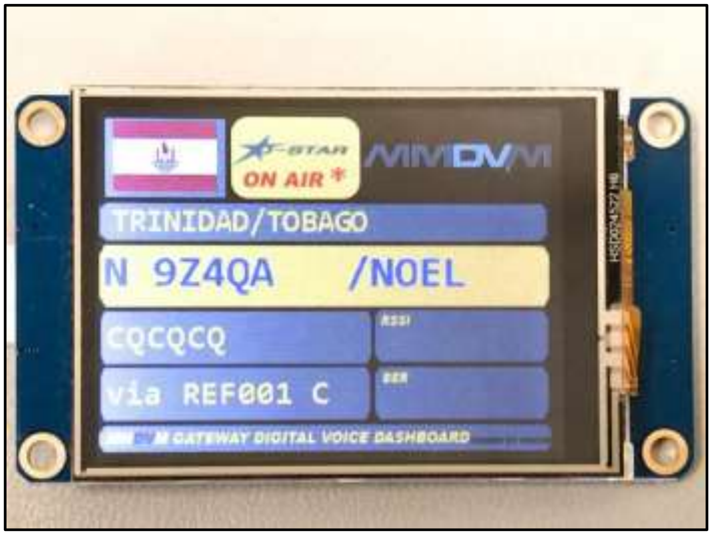
For connecting a NEXTION display, make sure that each colored wire matches the connections shown in the images below.



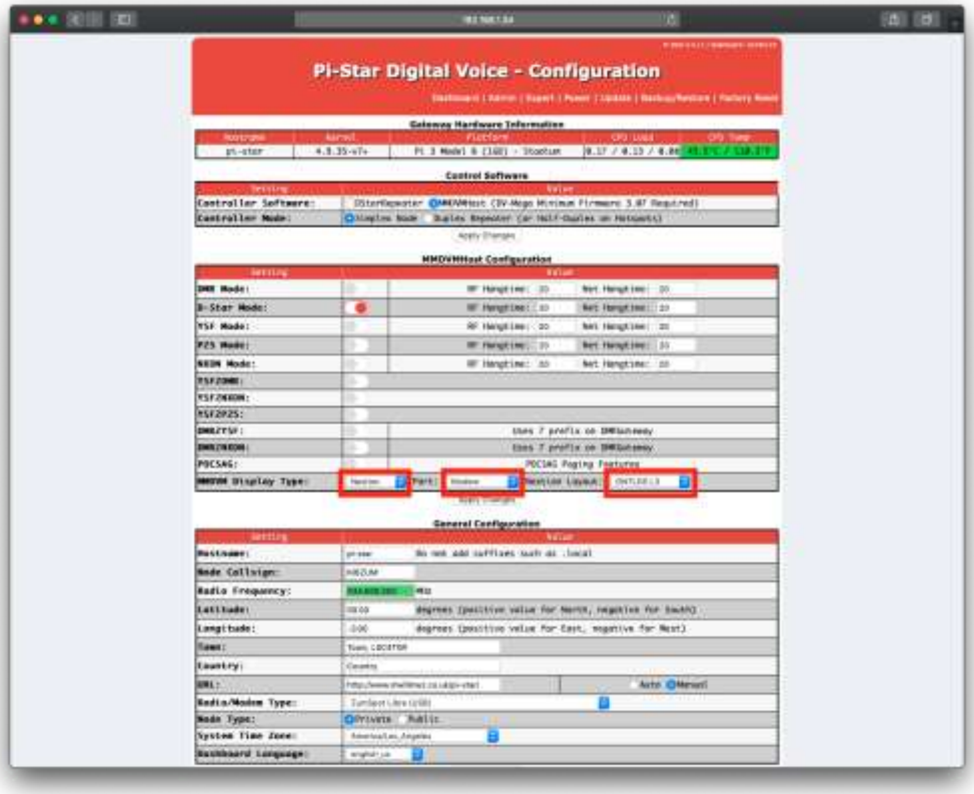




Here is an image of the display working.



To get the Nextion display working, make sure your settings are correct within Pi-Star. The *MMDVM Display Type:* should be set to "Nextion", the *Port:* should be set to "Modem" and the correct *Nextion Layout:* should be selected. In this example we have used the "ON7LDS L3" layout.



Pi-Star Digital Voice - Configuration

Dashboard | Admin | Expert | Power | Update | Backup/Restore | Factory Reset

Gateway Hardware Information

Hostname	Kernel	Platform	CPU Load	CPU Temp
pi-star	4.9.35-v7+	Pi 3 Model B (1GB) - [illegible]	0.17 / 0.13 / 0.0[illegible]	[illegible]

Control Software

Setting	Value
Controller Software:	○ DStarRepeater ● MMDVMHost (DV-Mega Minimum Firmware 3.07 Required)
Controller Mode:	● Simplex Node ○ Duplex Repeater (or Half-Duplex on Hotspots)

Apply Changes

MMDVMHost Configuration

Setting	Value	
DMR Mode:		RF Hangtime: 20 Net Hangtime: 20
D-Star Mode:	●	RF Hangtime: 20 Net Hangtime: 20
YSF Mode:		RF Hangtime: 20 Net Hangtime: 20
P25 Mode:		RF Hangtime: 20 Net Hangtime: 20
NXDN Mode:		RF Hangtime: 20 Net Hangtime: 20
YSF2DMR:		
YSF2NXDN:		
YSF2P25:		
DMR2YSF:		Uses 7 prefix on DMRGateway
DMR2NXDN:		Uses 7 prefix on DMRGateway
POCSAG:		POCSAG Paging Features
MMDVM Display Type:	Nextion	Port: Modem Nextion Layout: ON7LDS L3

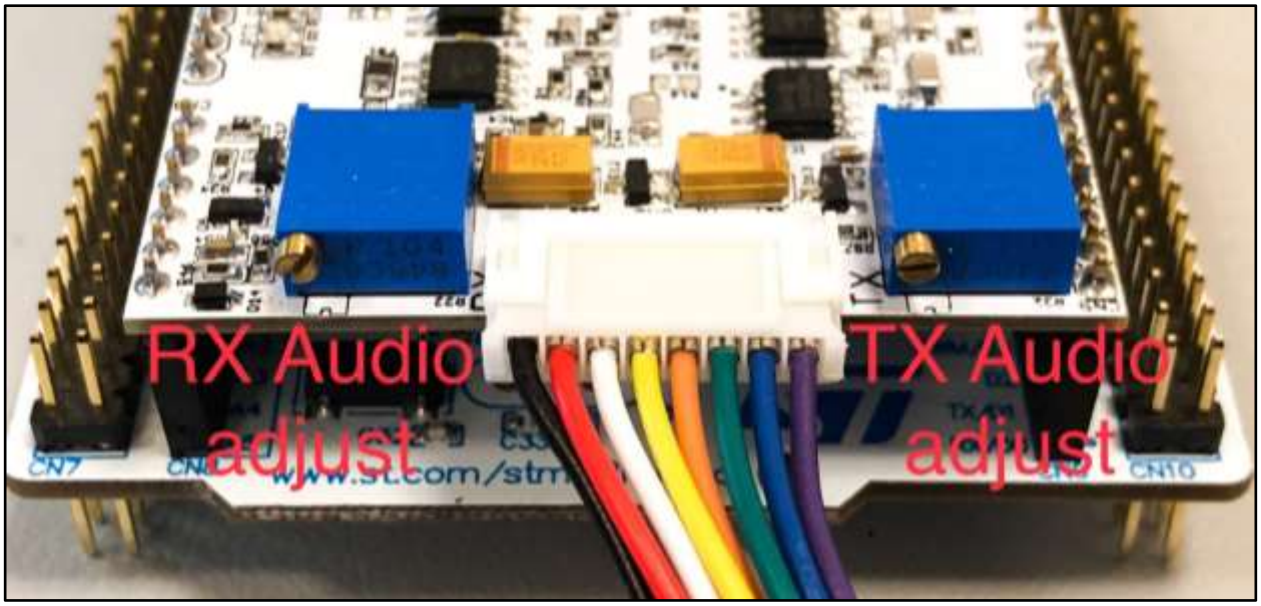
Apply Changes

General Configuration

Setting	Value
Hostname:	pi-star Do not add suffixes such as .local
Node Callsign:	[illegible]
Radio Frequency:	[illegible] MHz
Latitude:	[illegible] degrees (positive value for North, negative for South)
Longitude:	[illegible] degrees (positive value for East, negative for West)
Town:	Town, LOCATOR
Country:	Country
URL:	http://www.mw0mwz.co.uk/pi-star ○ Auto ● Manual
Radio/Modem Type:	ZumSpot Libre (USB)
Node Type:	● Private ○ Public
System Time Zone:	America/Los_Angeles
Dashboard Language:	english_us

Adjusting RX and TX signals

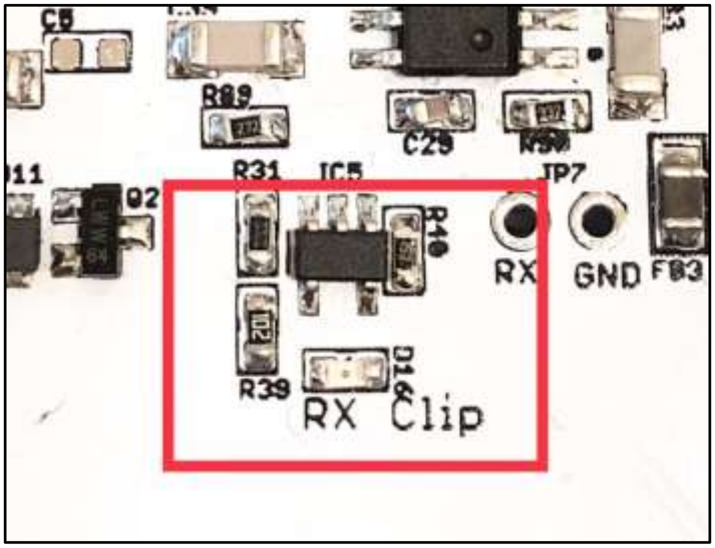
Here is a picture of the two trim pots used to adjust the signal levels of the RX and TX audio.



Turning the RX audio trim pot counter-clockwise will increase the signal level, while turning the trim pot clockwise will decrease the signal level.

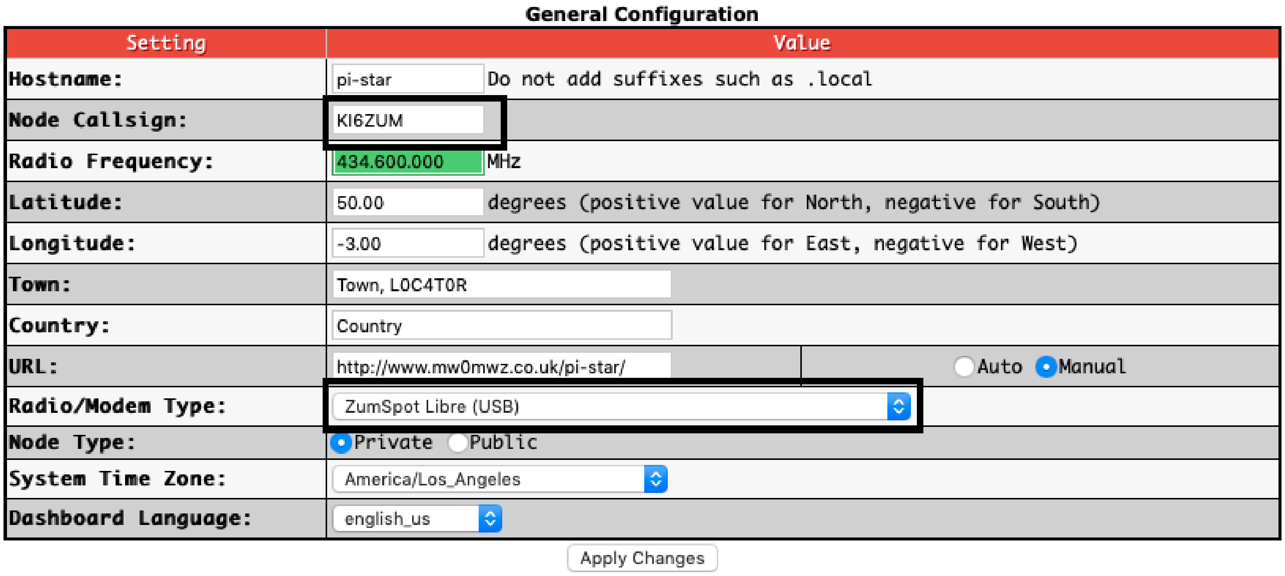
Turning the TX audio trim pot counter clockwise will decrease the signal level, while turning the trim pot clockwise will increase the signal level.

When there is no audio stream coming in, adjust the RX pot until the RX CLIP LED barely lights up.



Configuring Pi-Star (including adjusting signal levels via software)

From the “Configuration” menu, set the Radio/Modem type to “ZUMSpot Libre (USB)”, set your “Callsign” and then select Apply Changes.

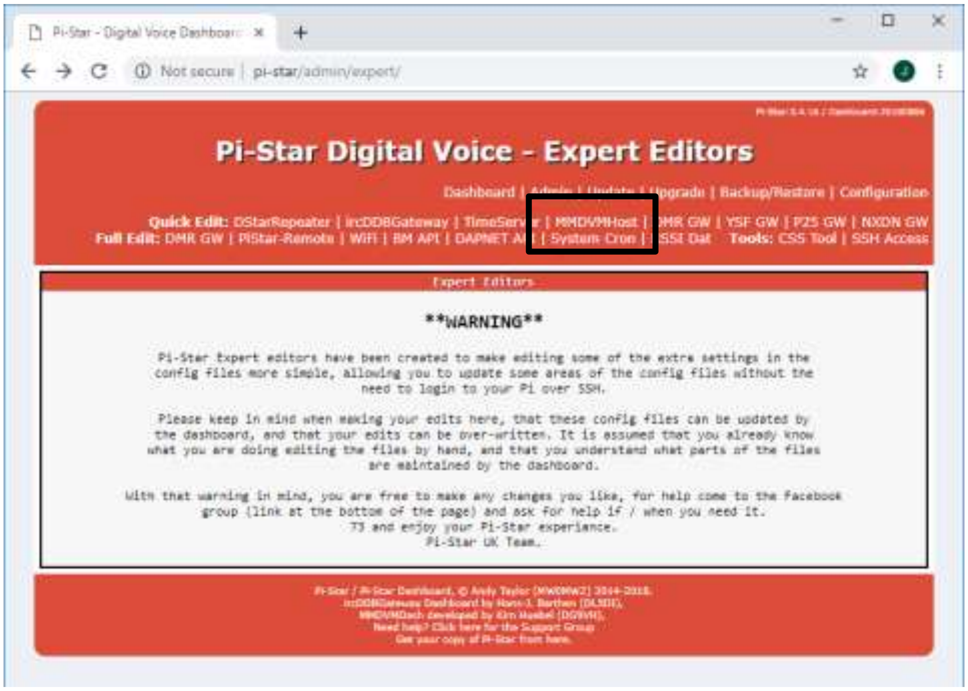


General Configuration

Setting	Value
Hostname:	pi-star Do not add suffixes such as .local
Node Callsign:	KI6ZUM
Radio Frequency:	434.600.000 MHz
Latitude:	50.00 degrees (positive value for North, negative for South)
Longitude:	-3.00 degrees (positive value for East, negative for West)
Town:	Town, L0C4T0R
Country:	Country
URL:	http://www.mw0mwz.co.uk/pi-star/ ○ Auto ● Manual
Radio/Modem Type:	ZumSpot Libre (USB)
Node Type:	● Private ○ Public
System Time Zone:	America/Los_Angeles
Dashboard Language:	english_us

Apply Changes

Next select the “expert” menu, and then select “MMDVMHost”


Pi-Star - Digital Voice Dashboard
Not secure | pi-star/admin/expert/
Pi-Star Digital Voice - Expert Editors
Dashboard | Admin | Update | Upgrade | Backup/Restore | Configuration
Quick Edit: DStarRepeater | ircDDBGateway | TimeServer | MMDVMHost | DMR GW | YSF GW | P25 GW | NXDN GW
Full Edit: DMR GW | PiStar-Remote | WiFi | BM API | DAPNET API | System Cron | RSSI Dat Tools: CSS Tool | SSH Access
Expert Editors

WARNING

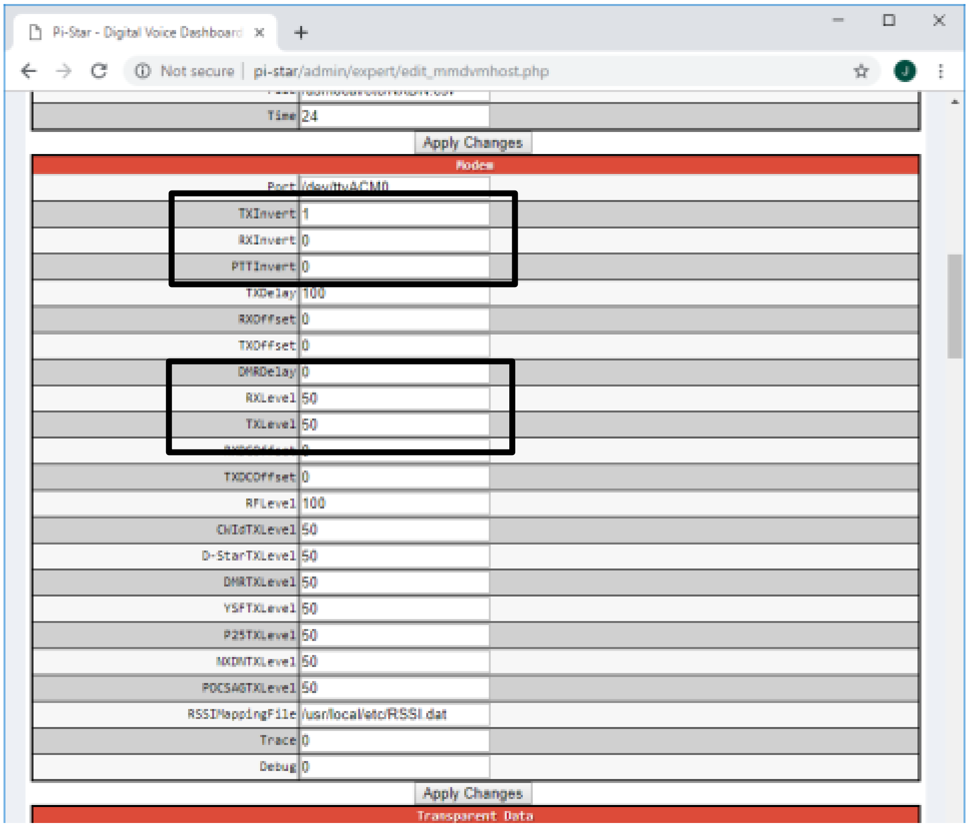
Pi-Star Expert editors have been created to make editing some of the extra settings in the config files more simple, allowing you to update some areas of the config files without the need to login to your Pi over SSH.

Please keep in mind when making your edits here, that these config files can be updated by the dashboard, and that your edits can be over-written. It is assumed that you already know what you are doing editing the files by hand, and that you understand what parts of the files are maintained by the dashboard.

With that warning in mind, you are free to make any changes you like, for help come to the Facebook group (link at the bottom of the page) and ask for help if / when you need it.
73 and enjoy your Pi-Star experiance.
Pi-Star UK Team.

Pi-Star / Pi-Star Dashboard, © Andy Taylor (MW0MWZ) 2014-2018.
ircDDBGateway Dashboard by Hans-J. Barthen (DL5DI),
MMDVMDash developed by Kim Huebel (DG9VH),
Need help? Click here for the Support Group
Get your copy of Pi-Star from here.

Scroll down to the "Modem" section to access the invert and level settings then select "Apply Changes"



Pi-Star - Digital Voice Dashboard
Not secure | pi-star/admin/expert/edit_mmdvmhost.php

Time	24

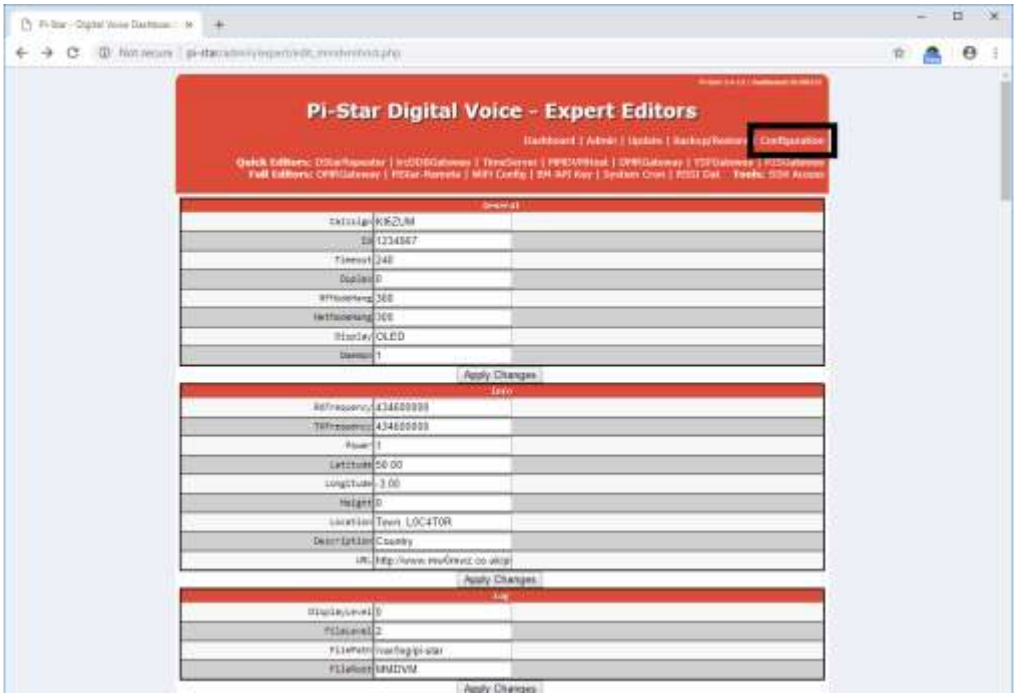
Apply Changes

Modem	
Port	/dev/ttyACM0
TXInvert	1
RXInvert	0
PTTInvert	0
TXDelay	100
RXOffset	0
TXOffset	0
DMRDelay	0
RXLevel	50
TXLevel	50
RXDCOffset	0
TXDCOffset	0
RFLevel	100
CWIdTXLevel	50
D-StarTXLevel	50
DMRTXLevel	50
YSFTXLevel	50
P25TXLevel	50
NXDNTXLevel	50
POCSAGTXLevel	50
RSSIMappingFile	/usr/local/etc/RSSI.dat
Trace	0
Debug	0

Apply Changes

Transparent Data

Scroll back up to the top of the page and select "Configuration".



Pi-Star - Digital Voice Dashboard
Not secure | pi-star/admin/expert/edit_mmdvmhost.php
Pi-Star Digital Voice - Expert Editors
Dashboard | Admin | Update | Backup/Restore | Configuration
Quick Editors: DStarRepeater | ircDDBGateway | TimeServer | MMDVMHost | DMRGateway | YSFGateway | P25Gateway
Full Editors: DMRGateway | RSStar-Remote | WiFi Config | BM API Key | System Cron | RSSI Dat Tools: SSH Access

General	
Callsign	KI6ZUM
Id	1234567
Timeout	240
Duplex	0
RFModeHang	300
NetModeHang	300
Display	OLED
Daemon	1

Apply Changes

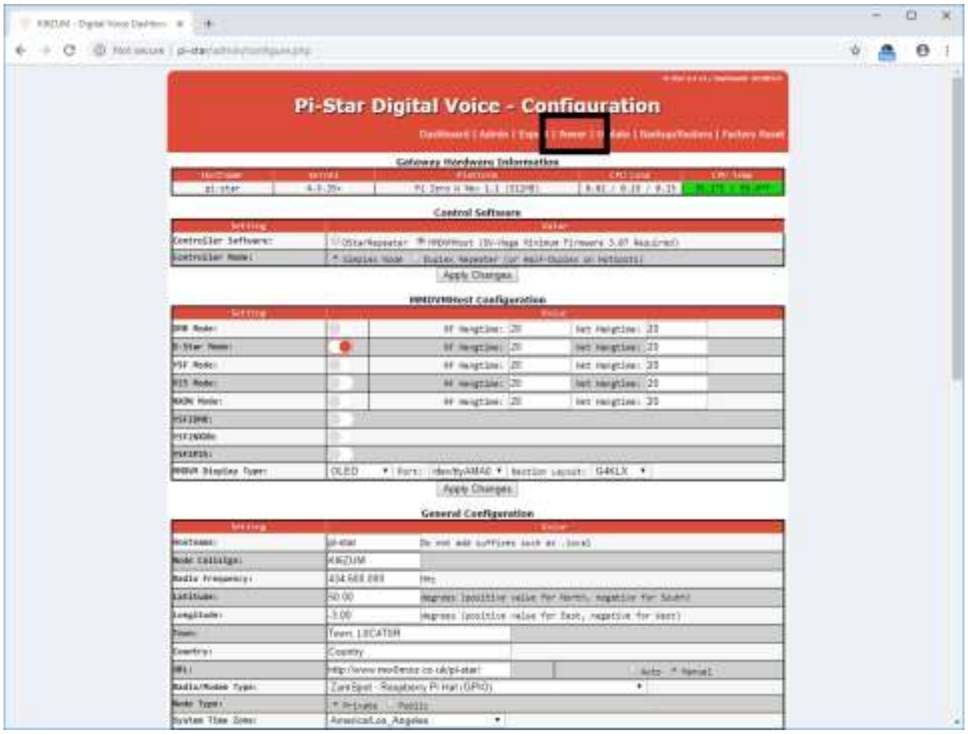
Info	
RXFrequency	434600000
TXFrequency	434600000
Power	1
Latitude	50.00
Longitude	-3.00
Height	0
Location	Town, L0C4T0R
Description	Country
URL	http://www.mw0mwz.co.uk/p

Apply Changes

Log	
DisplayLevel	0
FileLevel	2
FilePath	/var/log/pi-star
FileRoot	MMDVM

Apply Changes

Next, select “Power”



KI6ZUM - Digital Voice Dashboard

Not secure | pi-star/admin/configure.php

Pi-Star Digital Voice - Configuration

Dashboard | Admin | Expert | Power | Update | Backup/Restore | Factory Reset

Gateway Hardware Information

Hostname	Kernel	Platform	CPU Load	CPU Temp
pi-star	4.9.35+	Pi Zero W Rev 1.1 (512MB)	0.01 / 0.10 / 0.15	[illegible]

Control Software

Setting	Value
Controller Software:	○ DStarRepeater ● MMDVMHost (DV-Mega Minimum Firmware 3.07 Required)
Controller Mode:	● Simplex Node ○ Duplex Repeater (or Half-Duplex on Hotspots)

Apply Changes

MMDVMHost Configuration

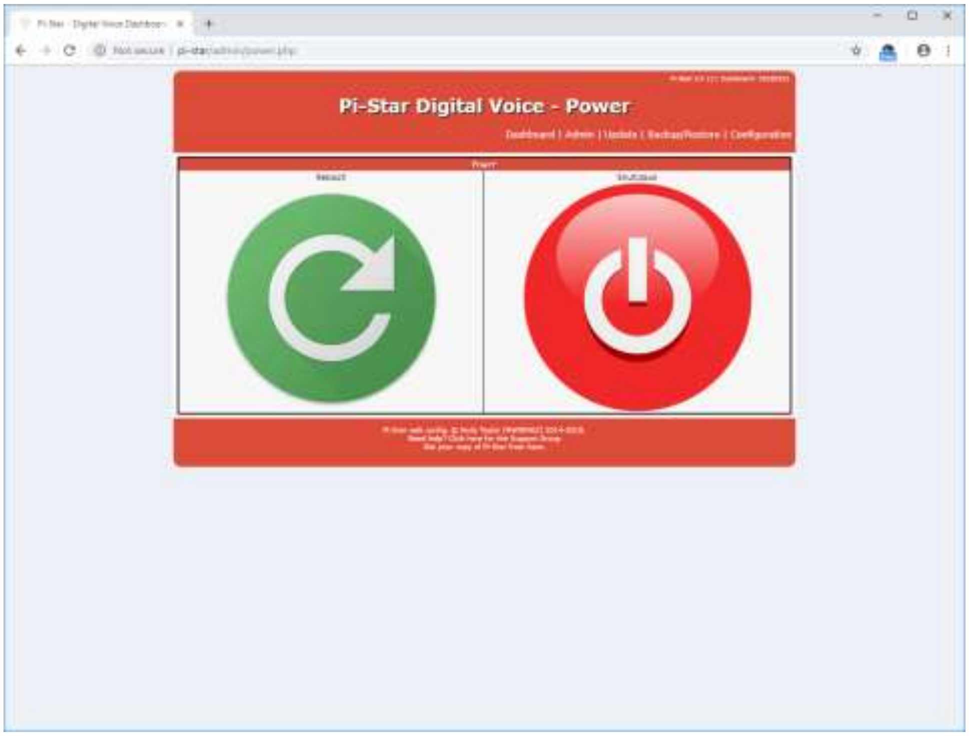
Setting		Value	
DMR Mode:		RF Hangtime: 20	Net Hangtime: 20
D-Star Mode:	●	RF Hangtime: 20	Net Hangtime: 20
YSF Mode:		RF Hangtime: 20	Net Hangtime: 20
P25 Mode:		RF Hangtime: 20	Net Hangtime: 20
NXDN Mode:		RF Hangtime: 20	Net Hangtime: 20
YSF2DMR:			
YSF2NXDN:			
YSF2P25:			
MMDVM Display Type:	OLED	Port: /dev/ttyAMA0	Nextion Layout: G4KLX

Apply Changes

General Configuration

Setting	Value
Hostname:	pi-star Do not add suffixes such as .local
Node Callsign:	KI6ZUM
Radio Frequency:	434.660.000 MHz
Latitude:	50.00 degrees (positive value for North, negative for South)
Longitude:	-3.00 degrees (positive value for East, negative for West)
Town:	Town, L0CATOR
Country:	Country
URL:	http://www.mw0mwz.co.uk/pi-star/ ○ Auto ● Manual
Radio/Modem Type:	ZumSpot - Raspberry Pi Hat (GPIO)
Node Type:	● Private ○ Public
System Time Zone:	America/Los_Angeles

Next, select “Reboot”



Pi-Star - Digital Voice Dashboard

Not secure | pi-star/admin/power.php

Pi-Star Digital Voice - Power

Dashboard | Admin | Update | Backup/Restore | Configuration

Power

Reboot	Shutdown

Pi-Star web config, © Andy Taylor (MW0MWZ) 2014-2018.
Need help? Click here for the Support Group
Get your copy of Pi-Star from here.

Building MMDVM firmware

On Windows 10

- Install bash using these instructions: bash-windows-10
- Once bash is installed, install GCC for ARM. Open the command prompt and type in the following instructions.
    - *bash*
    - *sudo apt-get install gcc*
    - *sudo apt-get install make*
    - *sudo apt-get remove gcc-arm-none-eabi*
    - *sudo apt-get install gcc-arm-none-eabi gdb-arm-none-eabi libstdc++-arm-none-eabi-newlib libnewlib-arm-none-eabi*
- Make sure git is installed. If it isn't then use this command: *sudo apt-get install git*
- Get the latest source code from GitHub:
    - *git clone https://github.com/g4klx/MMDVM*
    - *cd MMDVM*
    - *git submodule init*
    - *git submodule update*
- Edit Config.h. Uncomment the line: *#define MODE_LEDS*
    - If you want to connect a Nextion display, then also uncomment the line: #define SERIAL_REPEATER
    - To start build run: *make nucleo*
    - Binaries will be under the *bin/* folder

On Ubuntu

- Follow the same instructions as **Windows 10** but skip the part about installing bash

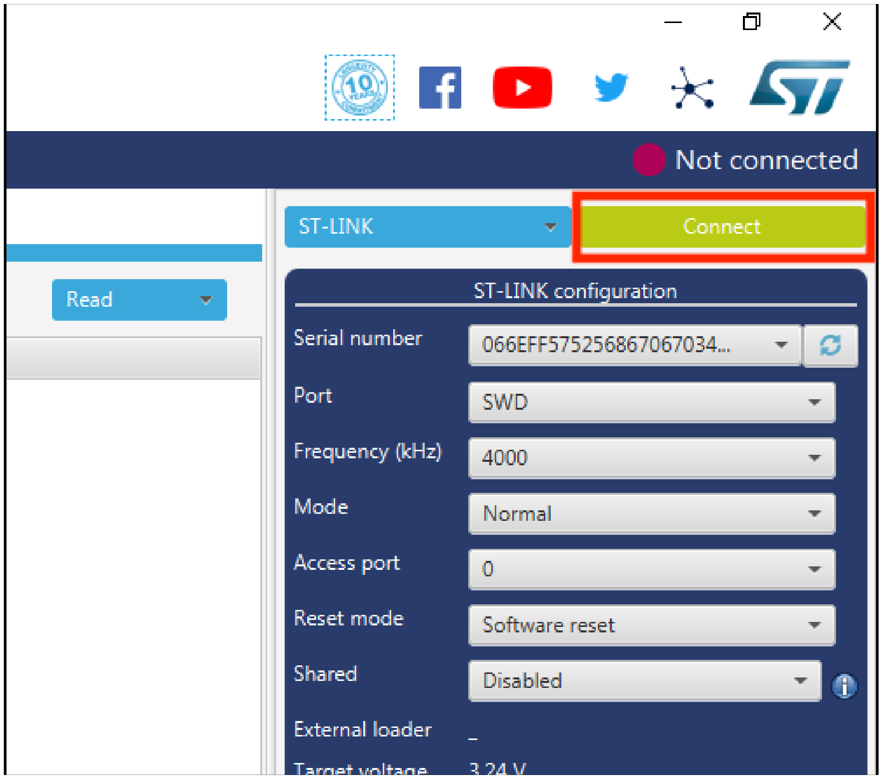
On macOS

- First install Homebrew:
    - Open the Terminal
    - Paste this in and press Enter: */usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"*
- Once Homebrew has been installed, run the following command: *brew install libusb autogen automake wget pkg-config cmake openocd*
- Install the ARM GCC toolchain:
    - Run the following command: *brew tap ArmMbed/homebrew-formulae*
    - Run the following command: *brew install arm-none-eabi-gcc*
- Get the latest source code from GitHub:
    - *git clone https://github.com/g4klx/MMDVM*
    - *cd MMDVM*
    - *git submodule init*
    - *git submodule update*

- Edit Config.h. Uncomment the line: *#define MODE_LEDS*
    - If you want to connect a Nextion display, then also uncomment the line: #define SERIAL_REPEATER
- To start build run: *make nucleo*
- Binaries will be under the *bin/* folder

Flashing Nucleo F446

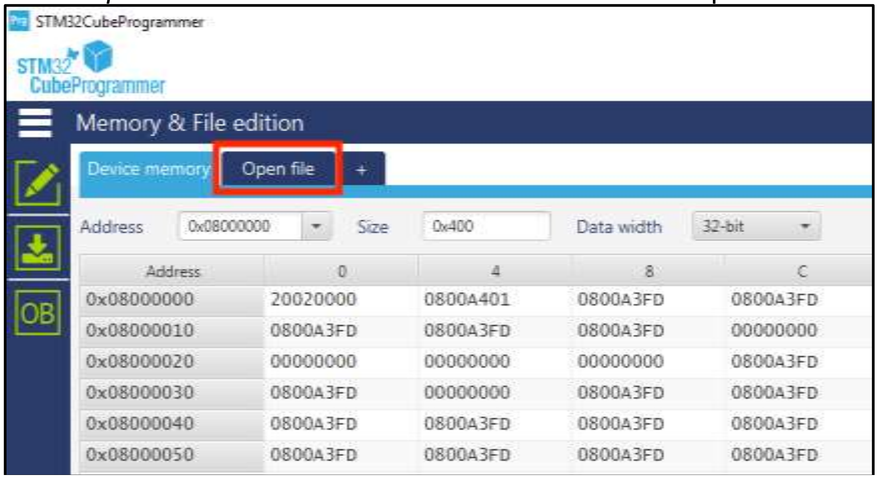
- Download the STM32CubeProgrammer from this link: STM32Cube
    - The programmer is available for Windows, Linux, and macOS
- Install the STM32CubeProgrammer
- Plug your Nucleo board to your computer
- Open the STM32CubeProgrammer on your computer
- Click “Connect” to connect to the Nucleo board



ST-LINK configuration

Serial number	066EFF575256867067034...
Port	SWD
Frequency (kHz)	4000
Mode	Normal
Access port	0
Reset mode	Software reset
Shared	Disabled
External loader	_
Target voltage	3.24 V

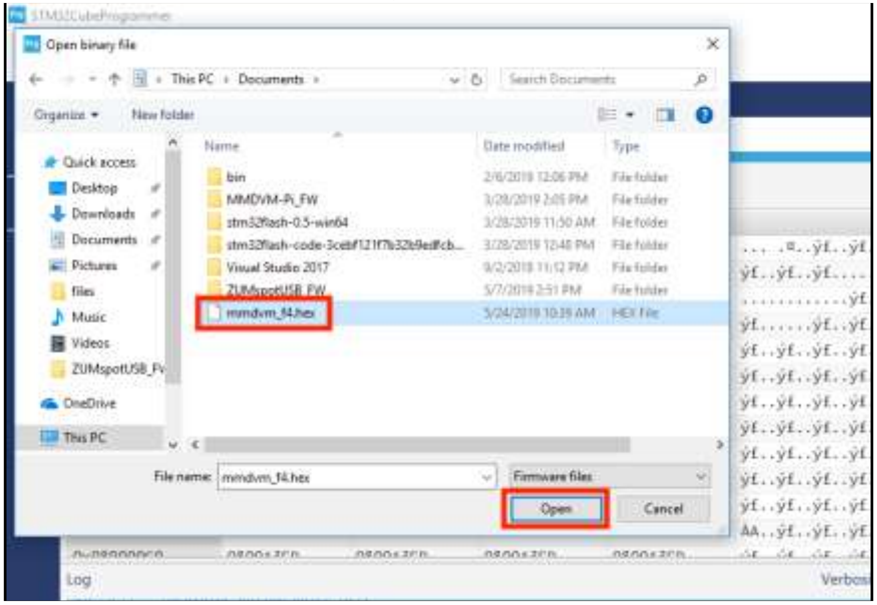
- Go to *Open File* and look for the folder where the compiled HEX file is in



Address 0x08000000 Size 0x400 Data width 32-bit

Address	0	4	8	C
0x08000000	20020000	0800A401	0800A3FD	0800A3FD
0x08000010	0800A3FD	0800A3FD	0800A3FD	00000000
0x08000020	00000000	00000000	00000000	0800A3FD
0x08000030	0800A3FD	00000000	0800A3FD	0800A3FD
0x08000040	0800A3FD	0800A3FD	0800A3FD	0800A3FD
0x08000050	0800A3FD	0800A3FD	0800A3FD	0800A3FD

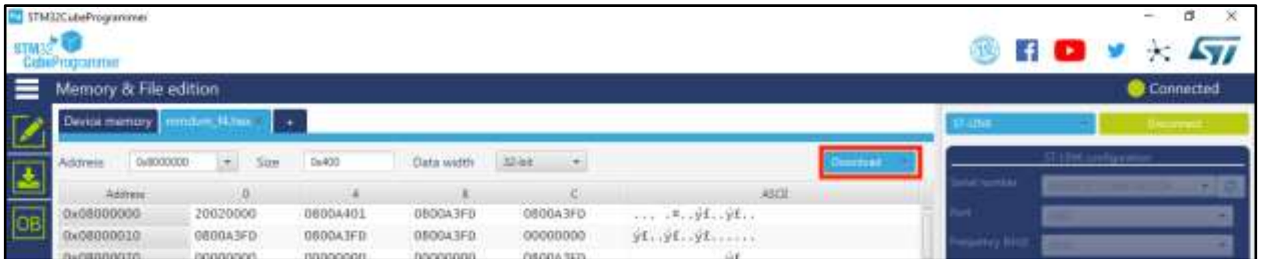
- Select the HEX file and click Open



Name	Date modified	Type
bin	2/6/2019 12:06 PM	File folder
MMDVM-Pi_FW	3/28/2019 2:05 PM	File folder
stm32flash-0.5-win64	3/28/2019 11:50 AM	File folder
stm32flash-code-3cebf121f7b32b9edfcb...	3/28/2019 12:48 PM	File folder
Visual Studio 2017	9/2/2018 11:12 PM	File folder
ZUMspotUSB_FW	5/7/2019 2:51 PM	File folder
mmdvm_f4.hex	5/24/2019 10:39 AM	HEX File

File name: mmdvm_f4.hex Firmware files

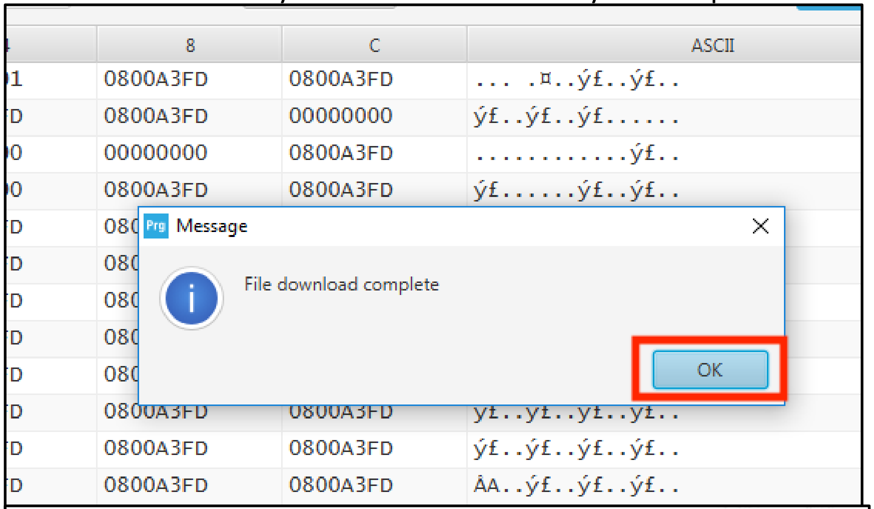
- Click on "Download"



Address 0x8000000 Size 0x400 Data width 32-bit

Address	0	4	8	C	ASCII
0x08000000	20020000	0800A401	0800A3FD	0800A3FD	... .¤..ý£..ý£..
0x08000010	0800A3FD	0800A3FD	0800A3FD	00000000	ý£..ý£..ý£......

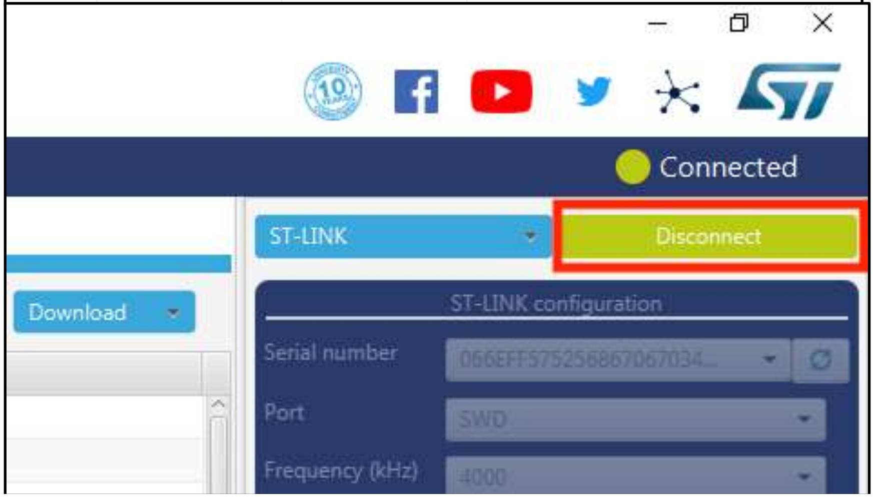
- Once the flashing is done. Click on “OK” in the popup window and then disconnect from the Nucleo board so you can remove it from your computer.



	8	C	ASCII
1	0800A3FD	0800A3FD	... .¤..ý£..ý£..
D	0800A3FD	00000000	ý£..ý£..ý£......
0	00000000	0800A3FD	...........ý£..
0	0800A3FD	0800A3FD	ý£......ý£..ý£..
D	080		
D	080		
D	080		
D	080		
D	080		
D	0800A3FD	0800A3FD	ý£..ý£..ý£..ý£..
D	0800A3FD	0800A3FD	ý£..ý£..ý£..ý£..
D	0800A3FD	0800A3FD	ÀA..ý£..ý£..ý£..

Message

File download complete



Connected

ST-LINK configuration

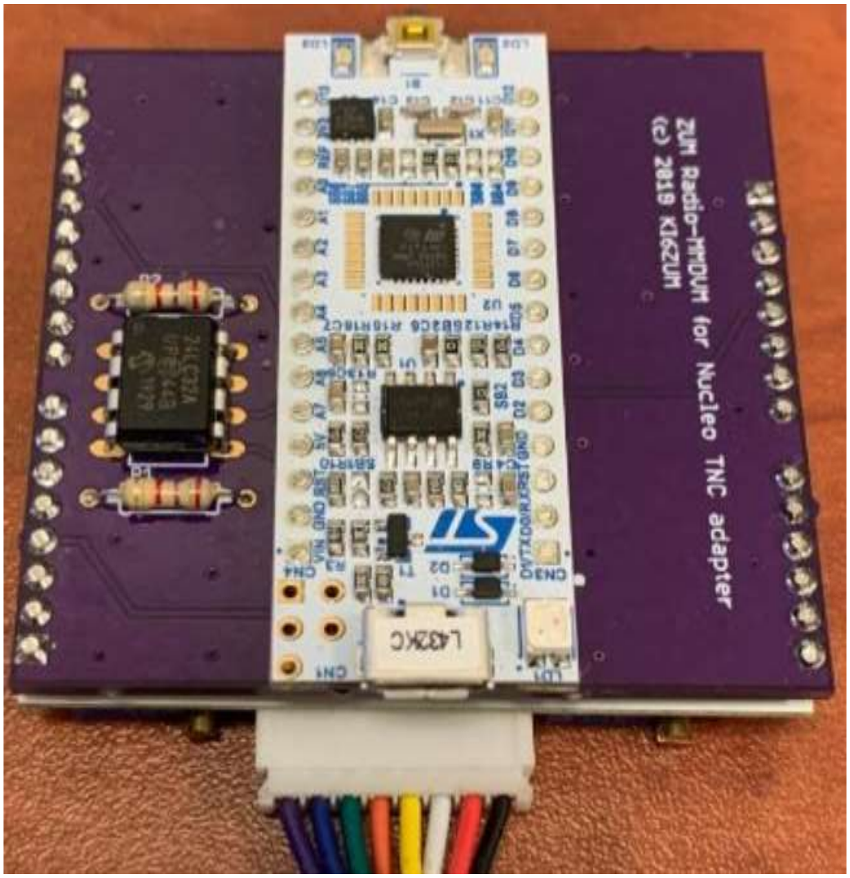
Serial number 066EFF575256867067034...

Port SWD

Frequency (kHz) 4000

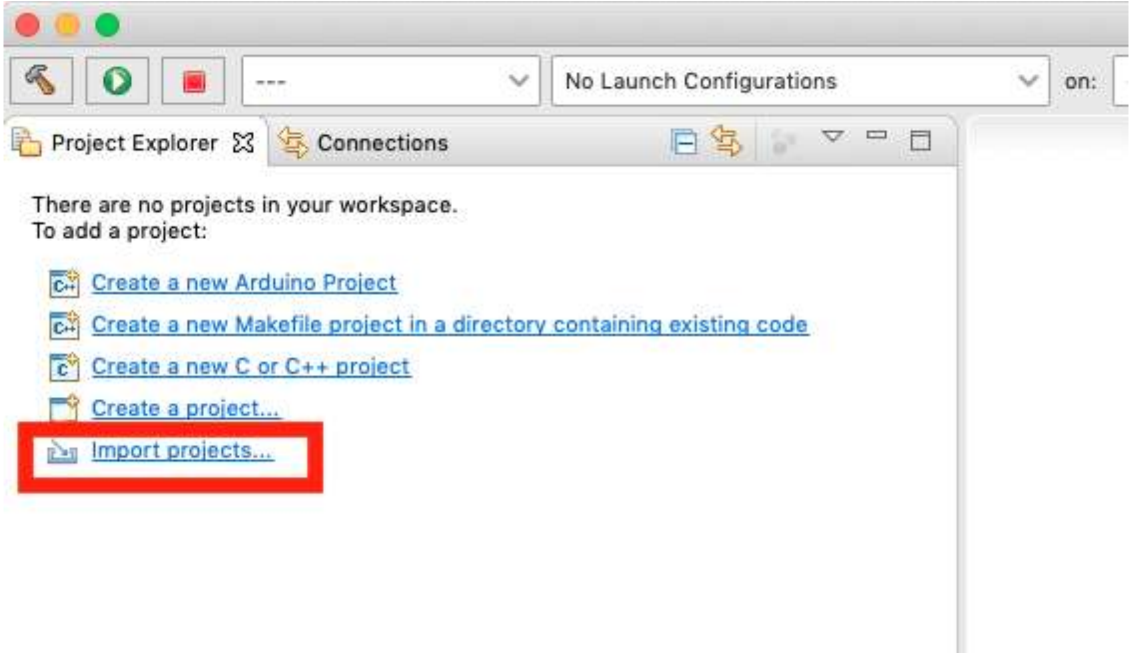
MMDVM-Nucleo with NucleoTNC firmware

NucleoTNC is an open source TNC project (https://github.com/mobilinkd/NucleoTNC) which runs on an STM32L432KC Nucleo32 board. It functions as a1200 baud KISS TNC over a USB serial port. With a simple adapter PCB the MMDVM-Nucleo board can be paired with the STM32L432 Nucleo board.



Building the NucleoTNC firmware

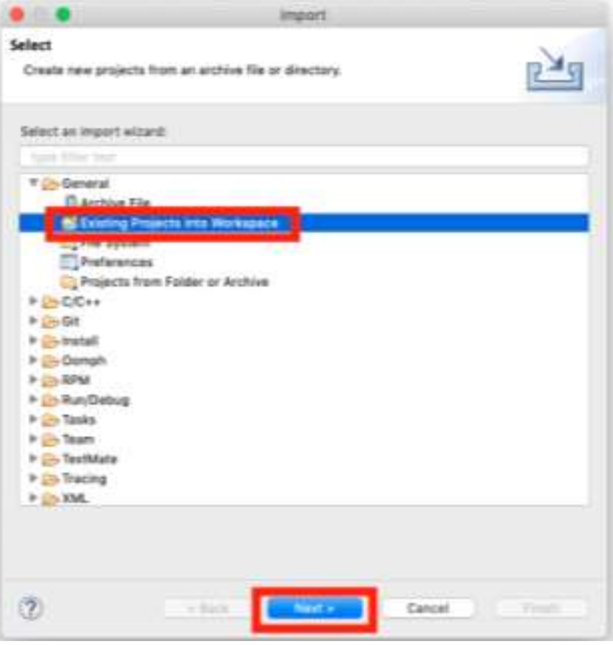
- Download CDT Eclipse with GNU MCU Eclipse plugin from here:
  https://github.com/gnu-mcu-eclipse/org.eclipse.epp.packages/releases/
  - If on macOS, follow the install steps listed on that page
- Get the latest NucleoTNC source code from GitHub
  - *git clone https://github.com/mobilinkd/NucleoTNC.git*
- Open Eclipse and select *Import projects…*



There are no projects in your workspace.
To add a project:

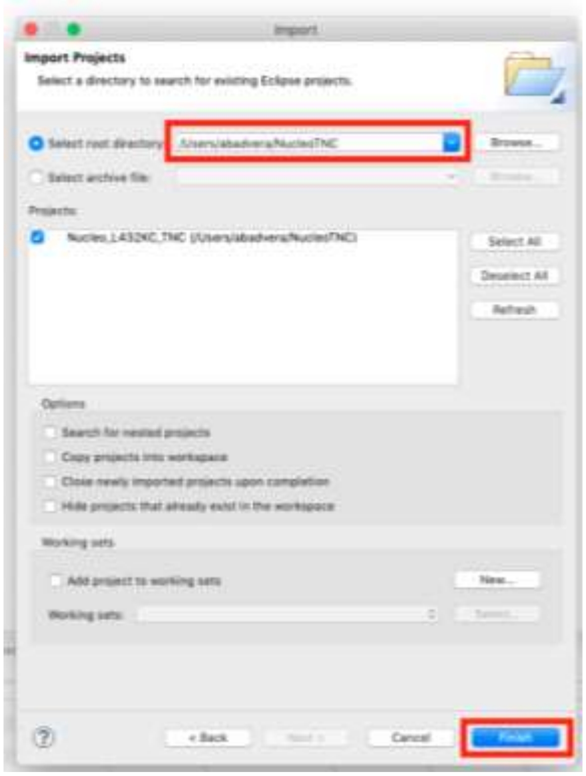
No Launch Configurations
on:
Project Explorer
Connections
Create a new Arduino Project
Create a new Makefile project in a directory containing existing code
Create a new C or C++ project
Create a project...
Import projects...

- Select *Existing Projects into Workspace* and click Next



Create new projects from an archive file or directory.

- Navigate to the directory where you cloned the NucleoTNC project. Once selected press Finish



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: /Users/abadvera/NucleoTNC

○ Select archive file:

☑ Nucleo_L432KC_TNC (/Users/abadvera/NucleoTNC)

□ Search for nested projects

□ Copy projects into workspace

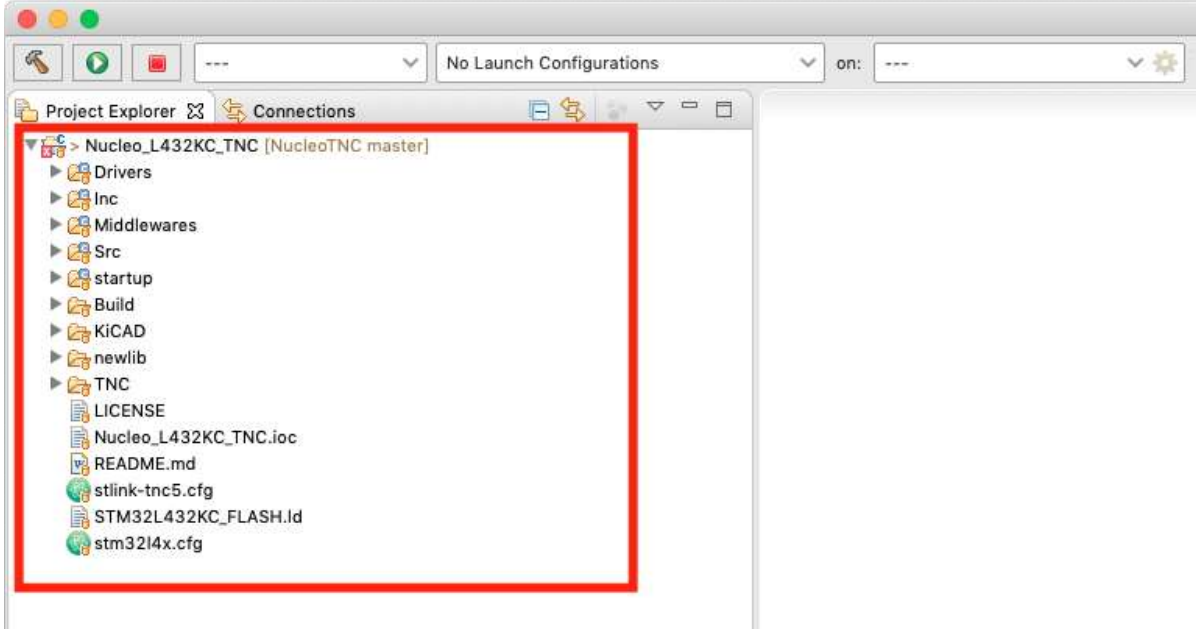
□ Close newly imported projects upon completion

□ Hide projects that already exist in the workspace

□ Add project to working sets

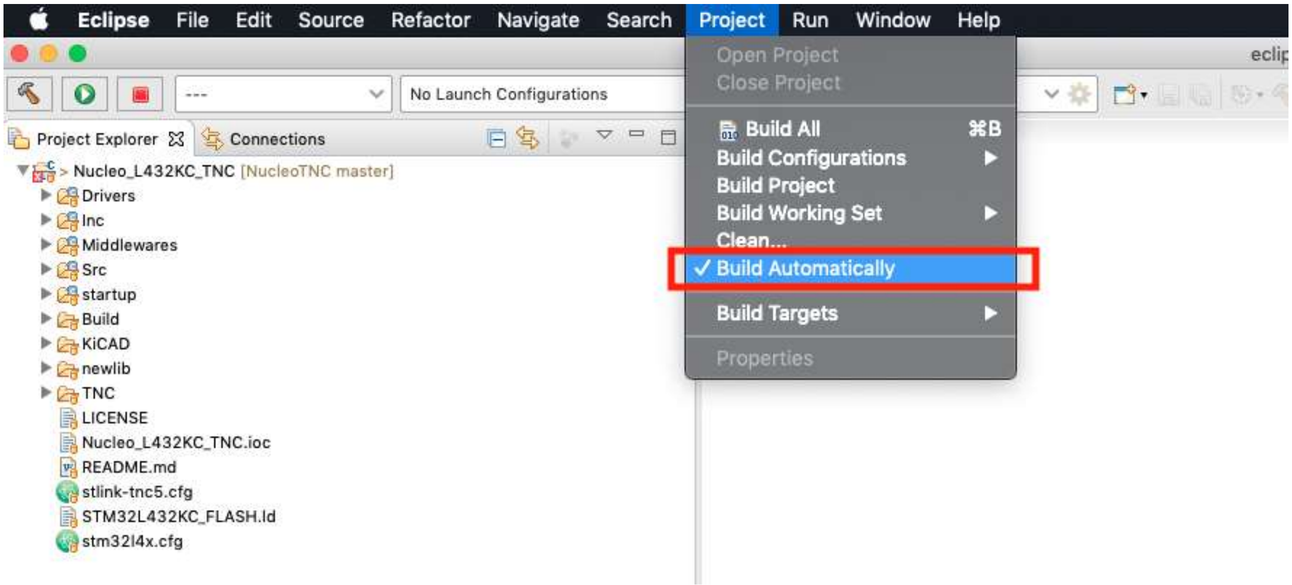
Working sets:

- You should now see the project name *Nucleo_L432KC_TNC***.** If you select the drop-down arrow you will see the folders and files that belong to the project.

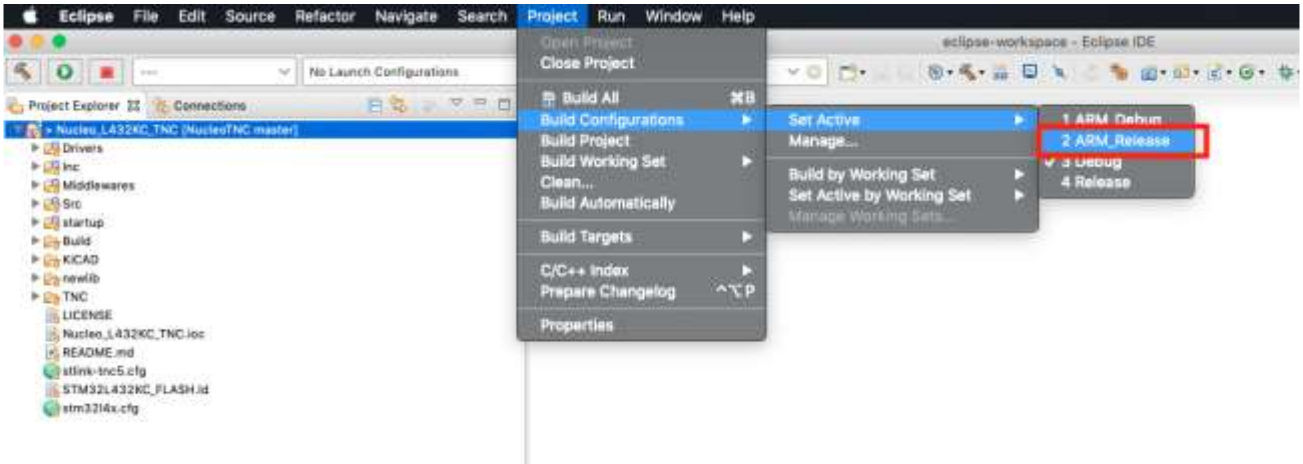


```
▼ > Nucleo_L432KC_TNC [NucleoTNC master]
  ▶ Drivers
  ▶ Inc
  ▶ Middlewares
  ▶ Src
  ▶ startup
  ▶ Build
  ▶ KiCAD
  ▶ newlib
  ▶ TNC
    LICENSE
    Nucleo_L432KC_TNC.ioc
    README.md
    stlink-tnc5.cfg
    STM32L432KC_FLASH.ld
    stm32l4x.cfg
```

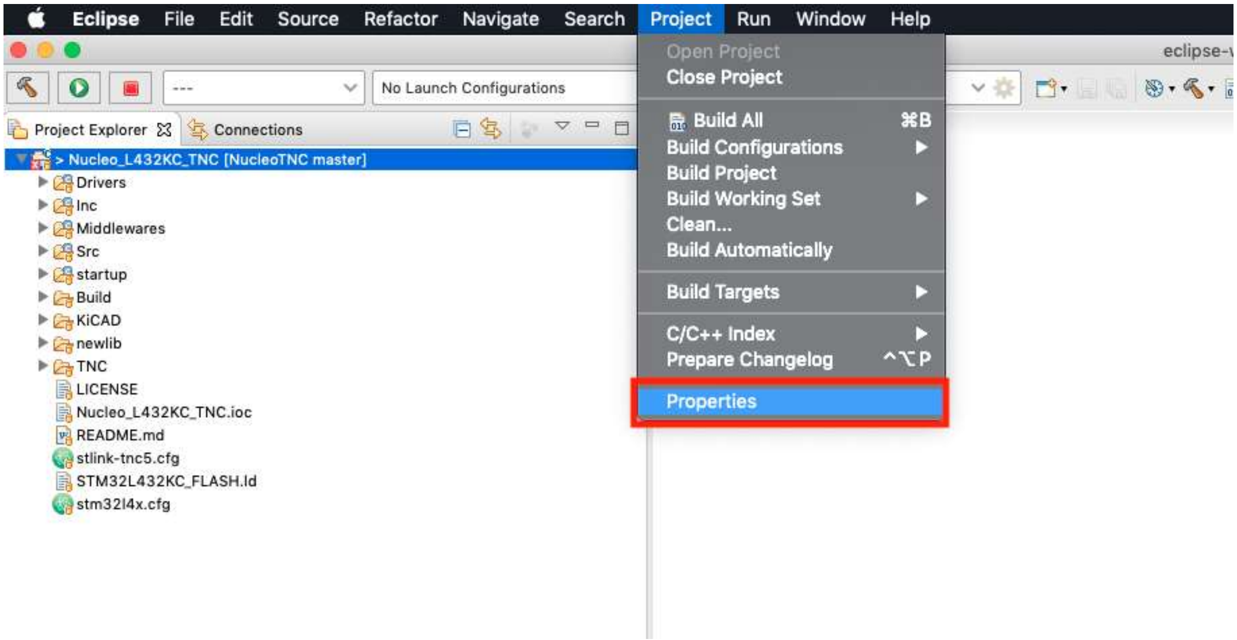
- With the project selected, click on *Project* and make sure is no check mark next to *Build Automatically.* If there is then unselect it so it doesn't constantly try and build while making modifications to the project settings.



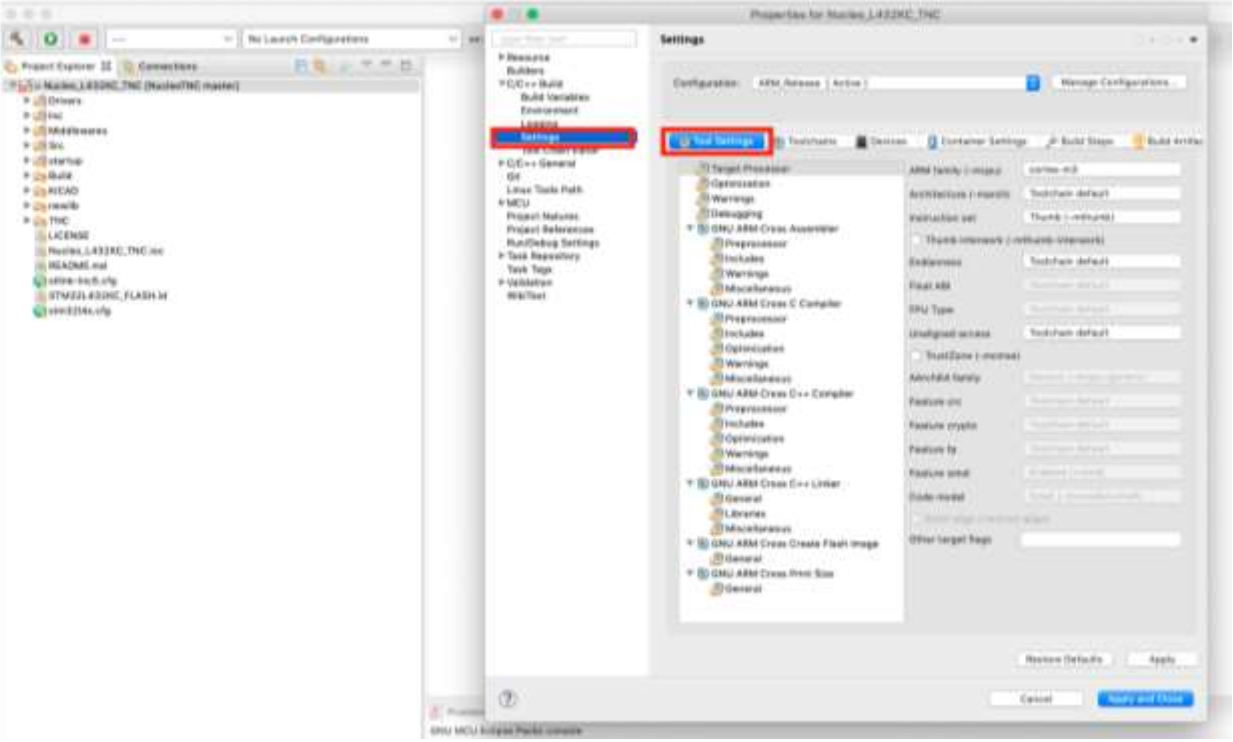
- With project selected, click on *Project* and go to *Build Configurations* → *Set Active* → *ARM_Release*. This will set this configuration as the active configuration to build.



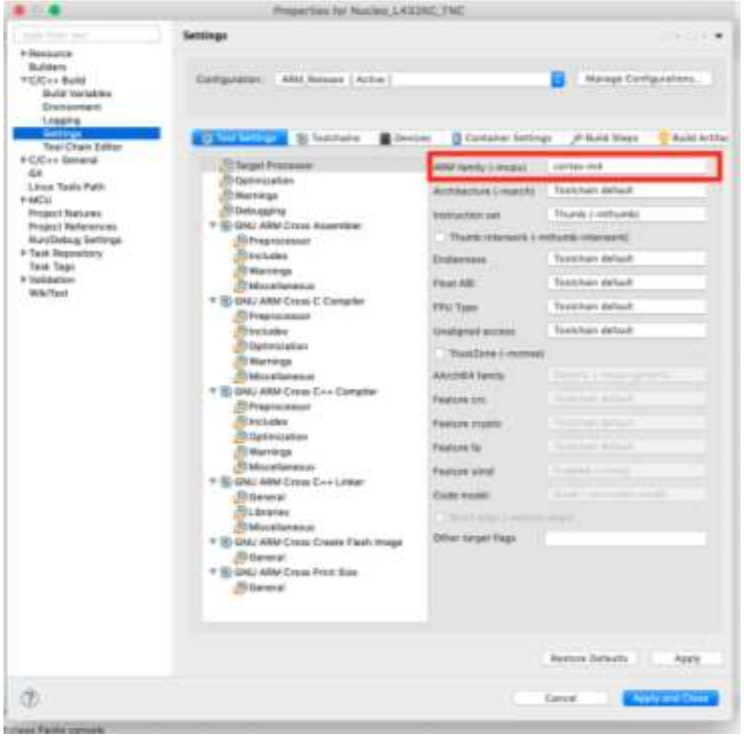
- With the project selected, go to *Project → Properties*



- On the properties window, Go to *C/C++ Build → Settings,* select the *Tool Settings* tab and make sure the *Target Processor* section is selected.



- Change the *ARM family (-mcpu)* section to *cortex-m4*

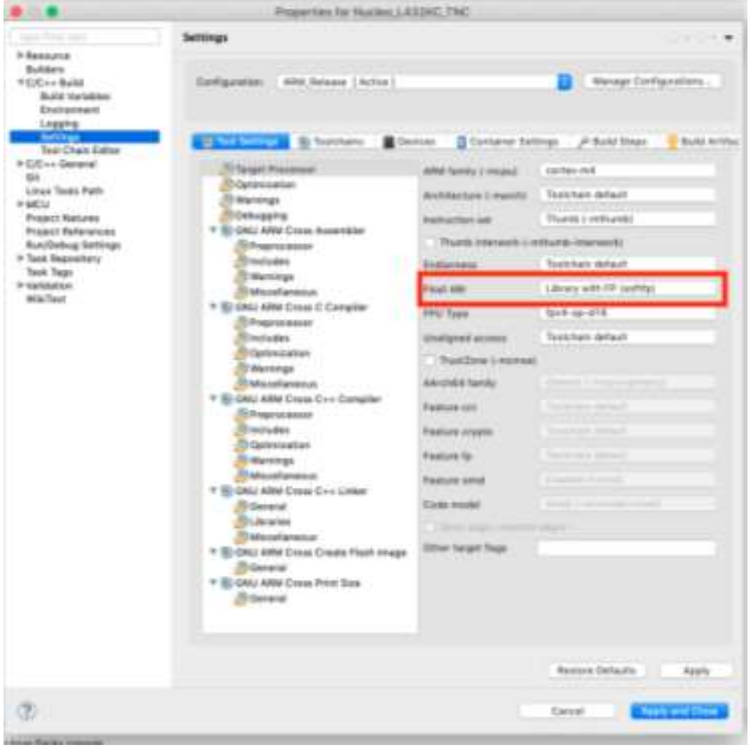


Properties for Nucleo_L432KC_TNC

Configuration: ARM_Release [Active]

ARM family (-mcpu): cortex-m4
Architecture (-march): Toolchain default
Instruction set: Thumb (-mthumb)
□ Thumb interwork (-mthumb-interwork)
Endianness: Toolchain default
Float ABI: Toolchain default
FPU Type: Toolchain default
Unaligned access: Toolchain default
□ TrustZone (-mcmse)
AArch64 family
Feature crc
Feature crypto
Feature fp
Feature simd
Code model
Other target flags:

- Change the *Float ABI* section to *Library with FP (softfp)*

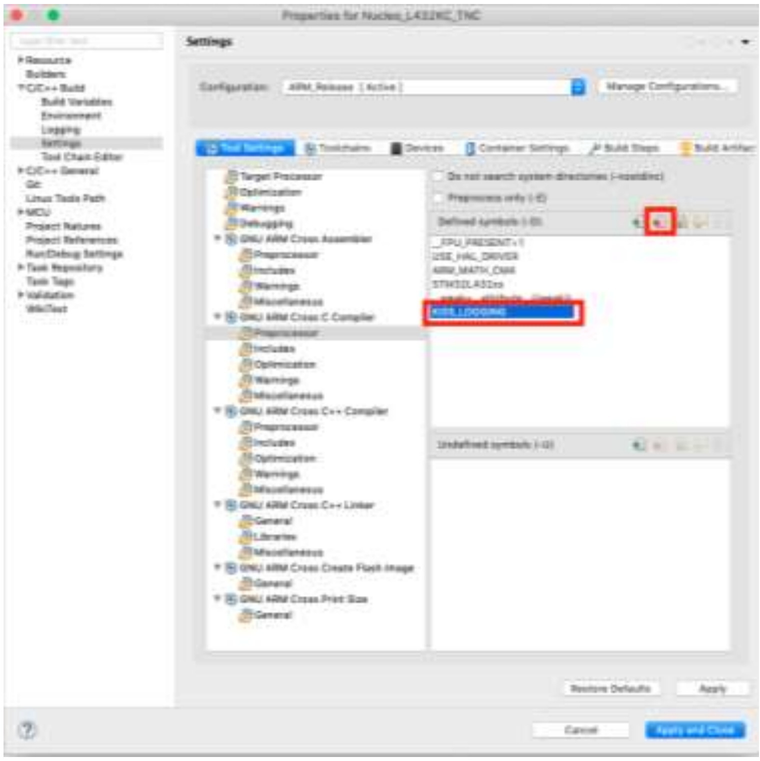


Properties for Nucleo_L432KC_TNC

Configuration: ARM_Release [Active]

ARM family (-mcpu): cortex-m4
Architecture (-march): Toolchain default
Instruction set: Thumb (-mthumb)
□ Thumb interwork (-mthumb-interwork)
Endianness: Toolchain default
Float ABI: Library with FP (softfp)
FPU Type: fpv4-sp-d16
Unaligned access: Toolchain default
□ TrustZone (-mcmse)
AArch64 family
Feature crc
Feature crypto
Feature fp
Feature simd
Code model
Other target flags:

- Click on *Preprocessor* under *GNU ARM Cross C Compiler*. Delete the *KISS_LOGGING* symbol under *Defined symbols (-D).*



Configuration: ARM_Release [Active]

□ Do not search system directories (-nostdinc)

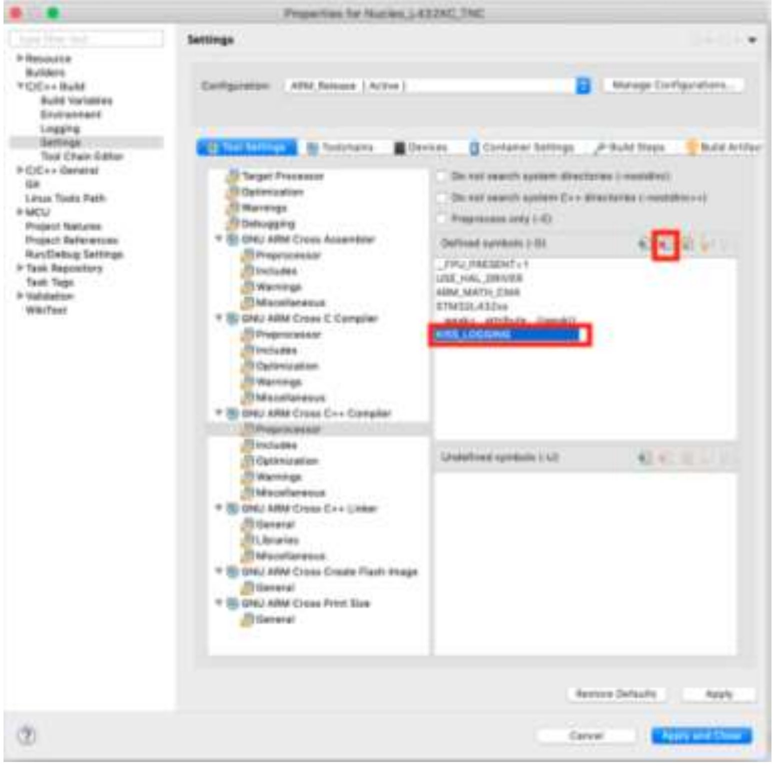
□ Preprocess only (-E)

Defined symbols (-D)

```
__FPU_PRESENT=1
USE_HAL_DRIVER
ARM_MATH_CM4
STM32L432xx
[illegible]
KISS_LOGGING
```

Undefined symbols (-U)

- Do the same thing for the *GNU ARM Cross C++ Compiler*



Configuration: ARM_Release [Active]

□ Do not search system directories (-nostdinc)

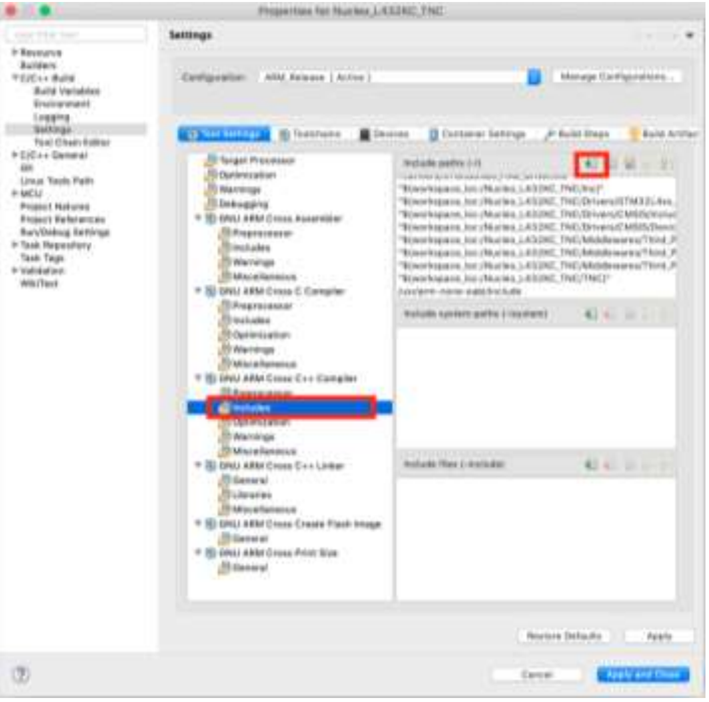
□ Do not search system C++ directories (-nostdinc++)

□ Preprocess only (-E)

Defined symbols (-D)

```
__FPU_PRESENT=1
USE_HAL_DRIVER
ARM_MATH_CM4
STM32L432xx
[illegible]
KISS_LOGGING
```

Undefined symbols (-U)

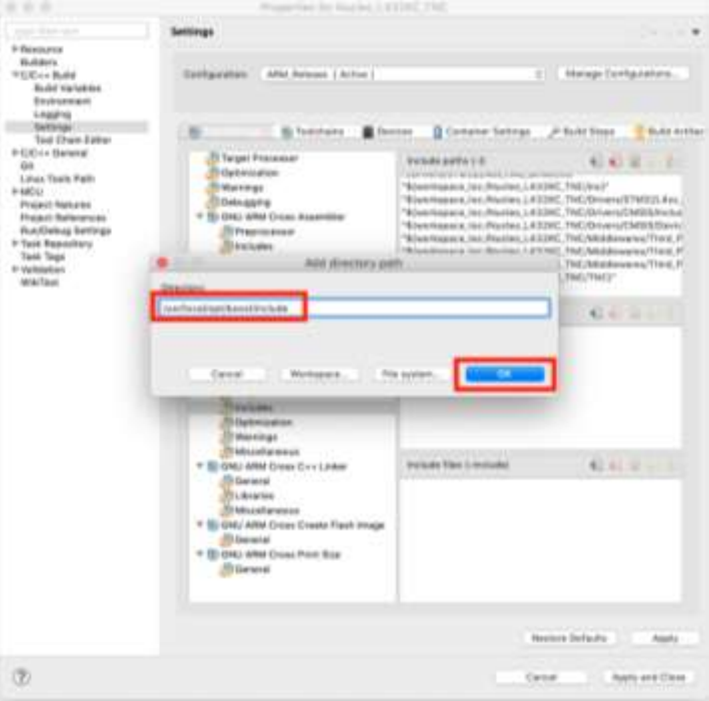


Configuration: ARM Release [Active]

Include paths (-I)

```
"${workspace_loc:/Nucleo_L432KC_TNC/Inc}"
"${workspace_loc:/Nucleo_L432KC_TNC/Drivers/STM32L4xx_
"${workspace_loc:/Nucleo_L432KC_TNC/Drivers/CMSIS/Inclu
"${workspace_loc:/Nucleo_L432KC_TNC/Drivers/CMSIS/Devic
"${workspace_loc:/Nucleo_L432KC_TNC/Middlewares/Third_P
"${workspace_loc:/Nucleo_L432KC_TNC/Middlewares/Third_P
"${workspace_loc:/Nucleo_L432KC_TNC/Middlewares/Third_P
"${workspace_loc:/Nucleo_L432KC_TNC/TNC}"
/usr/arm-none-eabi/include
```

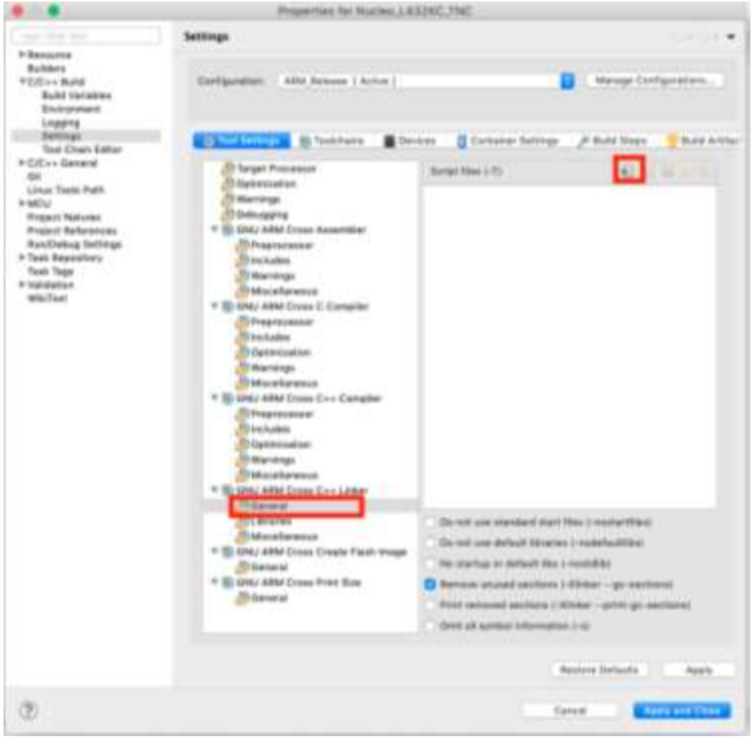
- Click on *Includes* under the *GNU ARM Cross C++ Compiler* section. Add the path to the boost library on your machine.

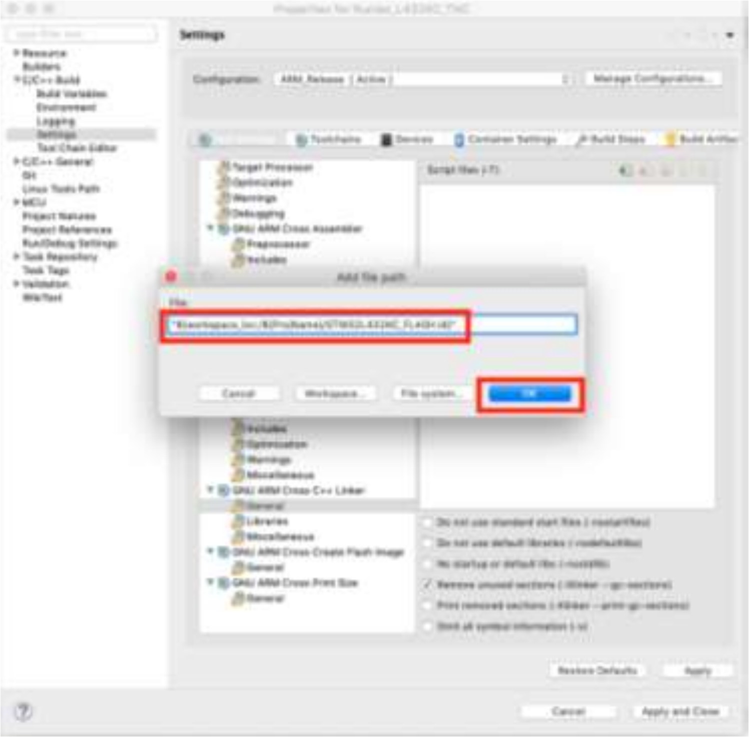


Directory: /usr/local/opt/boost/include

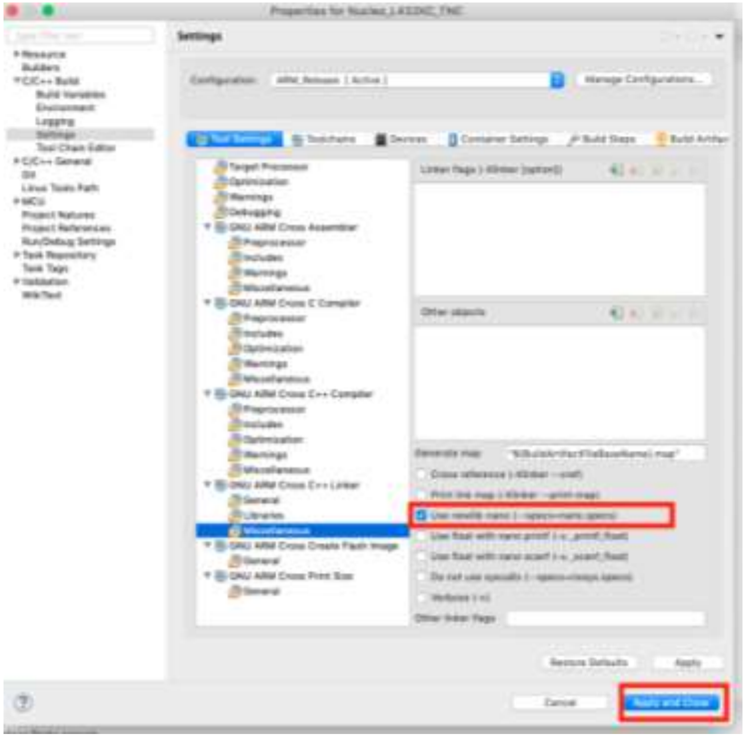
- To install boost:
  - Windows (with Bash installed) and Linux
  - sudo apt-get install libboost-all-dev
- macOS (using homebrew)
  - brew install boost

- Select *General* under *GNU ARM Cross C++ Linker*. Click on the plus button to add the linker script file: "${workspace_loc:/${ProjName}/STM32L432KC_FLASH.ld}"





- Select *Miscellaneous* under *GNU ARM Cross C++ Linker*. Make sure the check box next to the item *Use newlib-nano (--specs=nano.specs)* is checked. Then click on Apply and Close



Generate map: "${BuildArtifactFileBaseName}.map"

□ Cross reference (-Xlinker --cref)

□ Print link map (-Xlinker --print-map)

☑ Use newlib-nano (--specs=nano.specs)

□ Use float with nano printf (-u _printf_float)

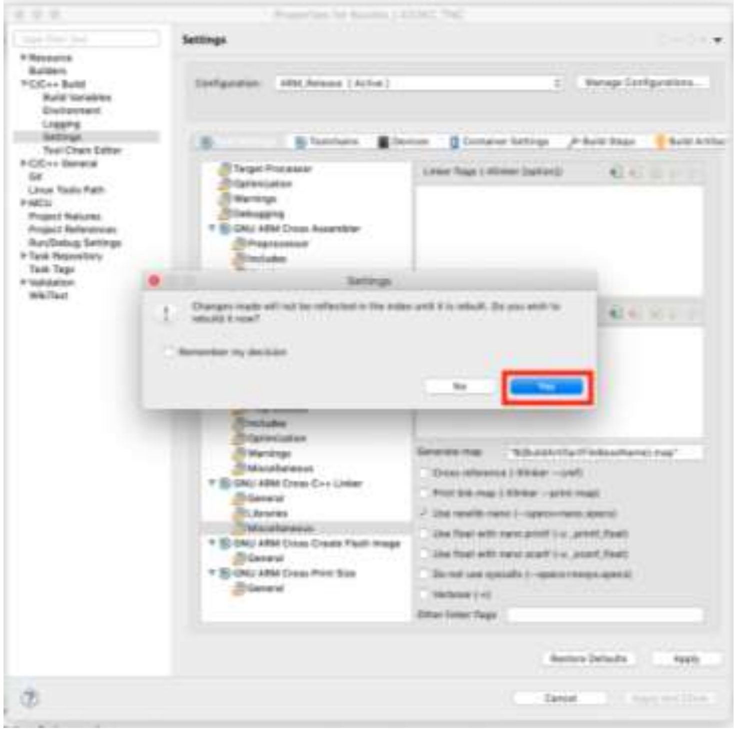
□ Use float with nano scanf (-u _scanf_float)

□ Do not use syscalls (--specs=nosys.specs)

□ Verbose (-v)

Other linker flags

- A pop up window will show. Click Yes

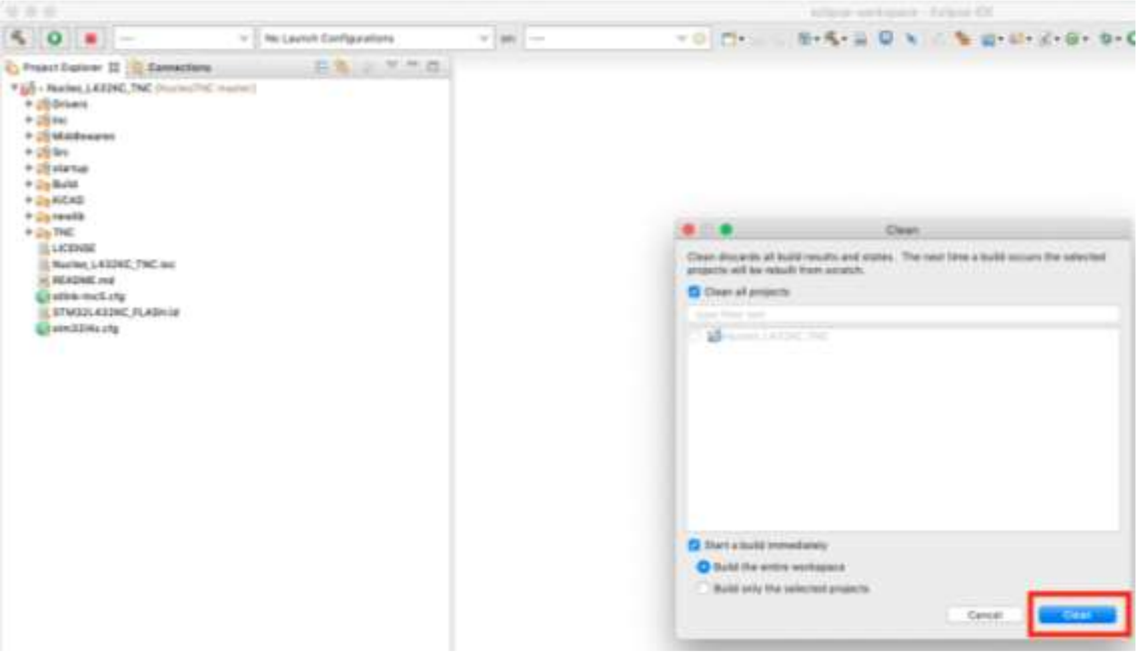


Settings

Changes made will not be reflected in the index until it is rebuilt. Do you wish to rebuild it now?

□ Remember my decision

- With project selected, click on *Project* and click on *Clean.* Click on the Clean button so that the project directory is cleaned before starting a build.



Clean discards all build results and states. The next time a build occurs the selected projects will be rebuilt from scratch.

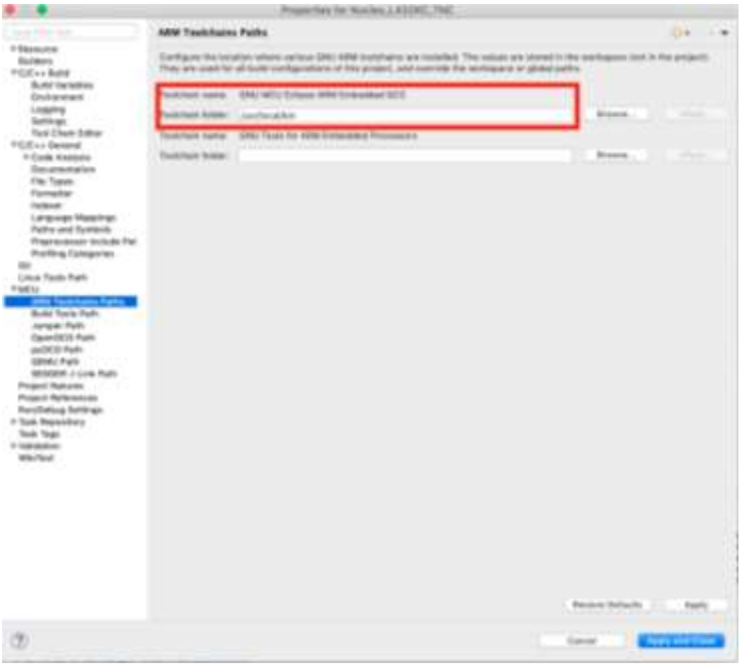
☑ Clean all projects

☑ Start a build immediately

● Build the entire workspace

○ Build only the selected projects

- If you run into this issue: *Error: Program "arm-none-eabi-gcc" not found in PATH*. Then try this:
    - With the project selected, go to Project -> Properties
    - Go to MCU -> ARM Toolchain Paths
    - Add the directory where you have arm-none-eabi-gcc installed. For me it's under /usr/local/bin
    - Then click on **Apply and Close**
    - Now when you Build you won't see that error.



ARM Toolchains Paths

Configure the location where various GNU ARM toolchains are installed. The values are stored in the workspace (not in the project). They are used for all build configurations of this project, and override the workspace or global paths.

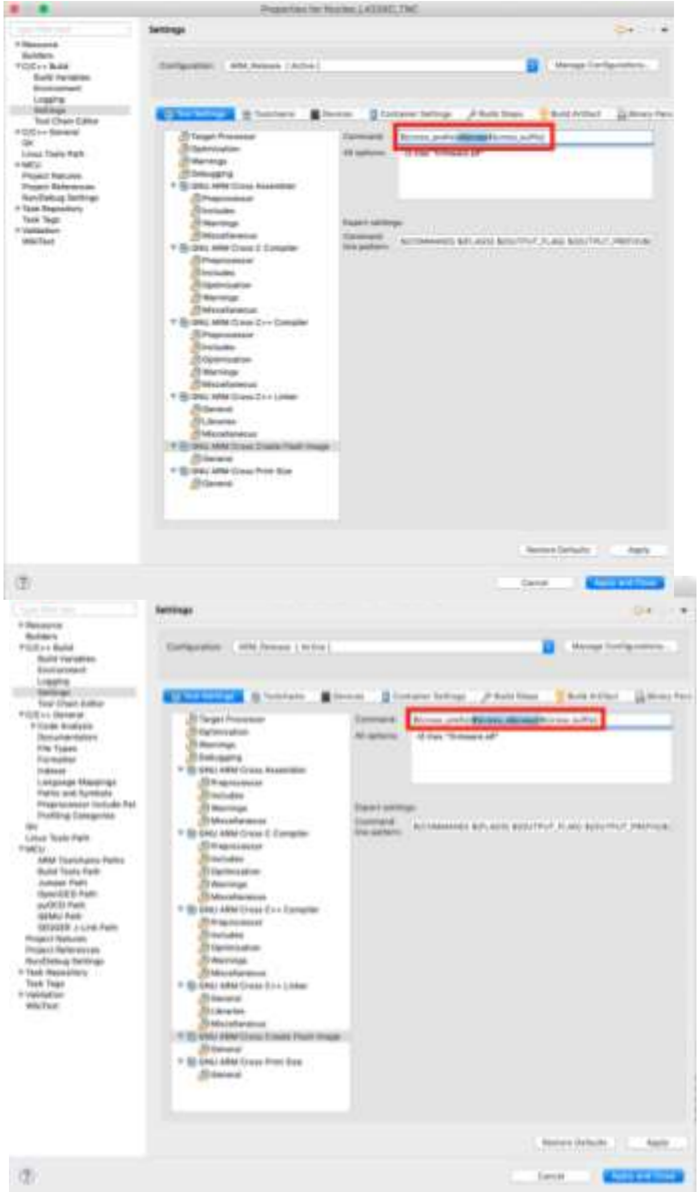
Toolchain name: GNU MCU Eclipse ARM Embedded GCC

Toolchain folder: /usr/local/bin

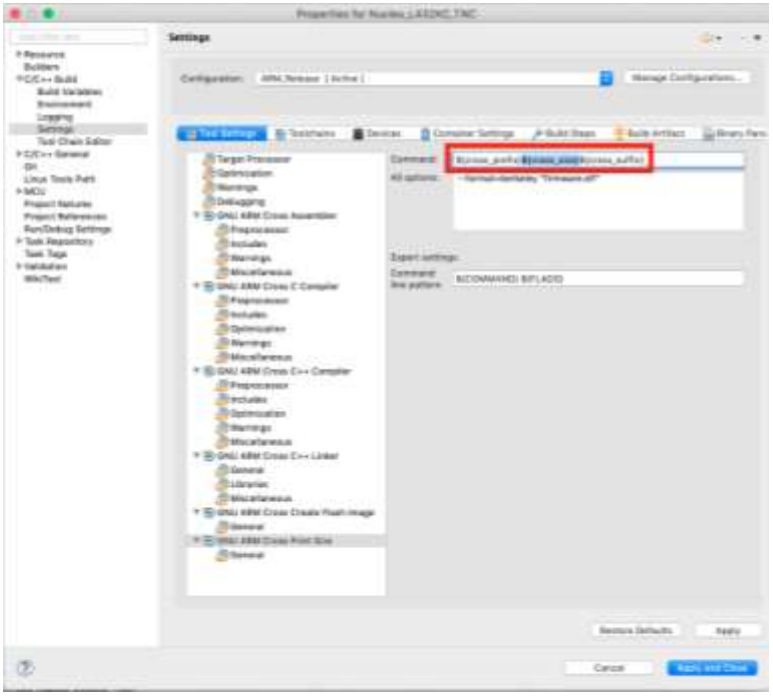
Toolchain name: GNU Tools for ARM Embedded Processors

Toolchain folder:

- During the Build steps if you run into this issue:
  arm-none-eabi- --format=berkeley firmware.elf Cannot run program "arm-none-eabi-": Unknown reason
- Do the following:
  - With the project selected, go to Project -> Properties
  - Go to *C/C++ Build -> Settings -> Tool Settings -> GNU ARM Cross Create Flash Image*
  - In the command field change the text ${cross_prefix}${objcopy}${cross_suffix}
  - To this: ${cross_prefix}objcopy${cross_suffix}
  - Go to *C/C++ Build -> Settings -> Tool Settings -> GNU ARM Cross Print Size*



- In the command field change the text ${cross_prefix}${cross_size}${cross_suffix}
    - To this: ${cross_prefix}size${cross_suffix}

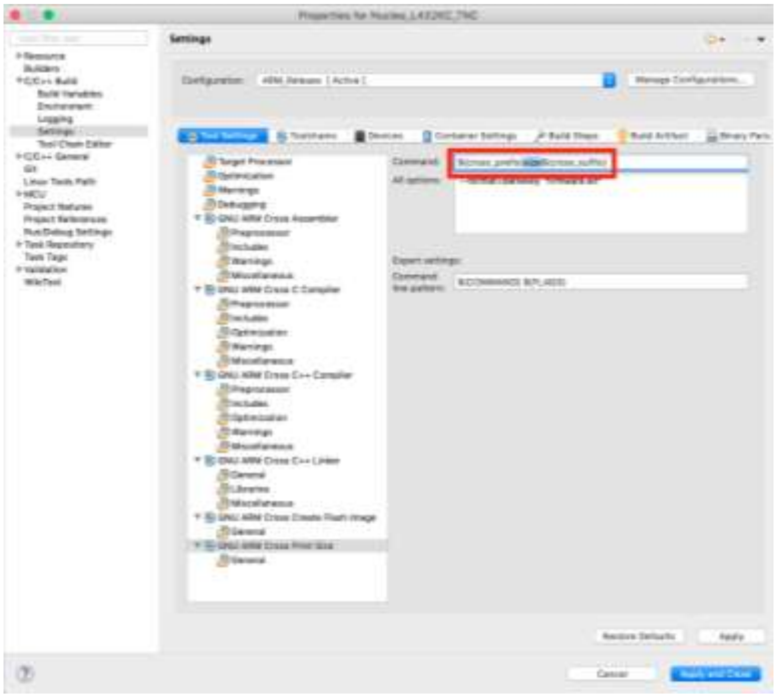


Command: ${cross_prefix}${cross_size}${cross_suffix}

All options: --format=berkeley "firmware.elf"

Expert settings:

Command line pattern: ${COMMAND} ${FLAGS}



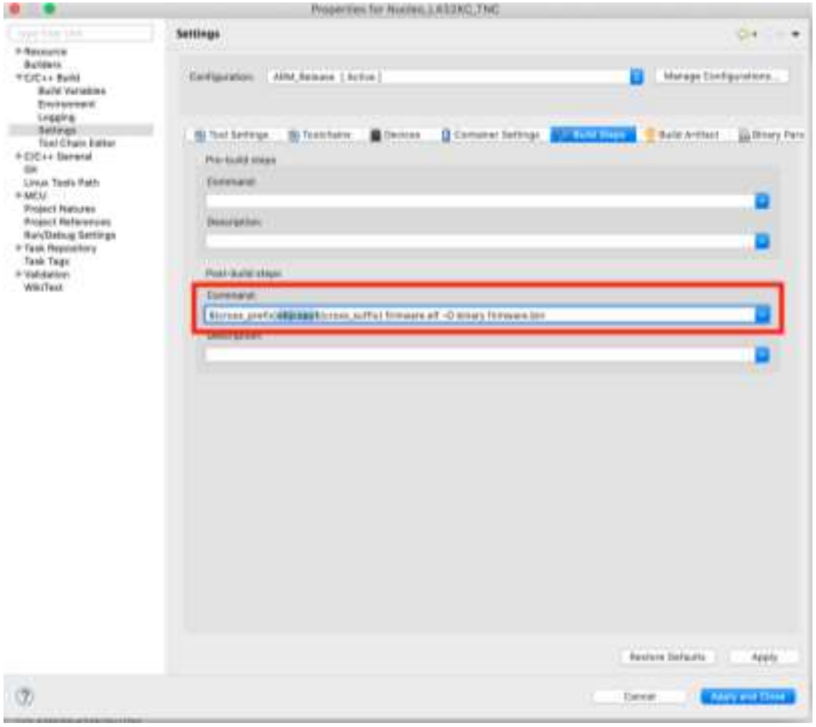
Command: ${cross_prefix}size${cross_suffix}

All options: --format=berkeley "firmware.elf"

Expert settings:

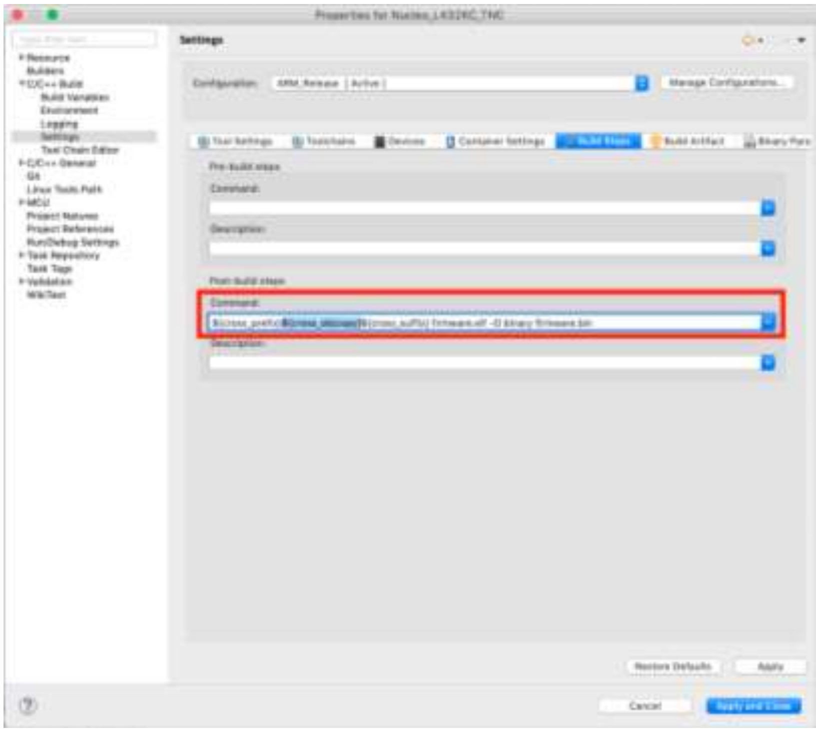
Command line pattern: ${COMMAND} ${FLAGS}

- Go to *C/C++ Build -> Settings -> Build Steps*
- Under the section *Post-build steps.* Change the text in the command field ${cross_prefix}${cross_objcopy}${cross_suffix} firmware.elf -O binary firmware.bin
- To this: ${cross_prefix}objcopy${cross_suffix} firmware.elf -O binary firmware.bin



Configuration: ARM_Release [Active]

Command: ${cross_prefix}objcopy${cross_suffix} firmware.elf -O binary firmware.bin



Configuration: ARM_Release [Active]

Command: ${cross_prefix}${cross_objcopy}${cross_suffix} firmware.elf -O binary firmware.bin

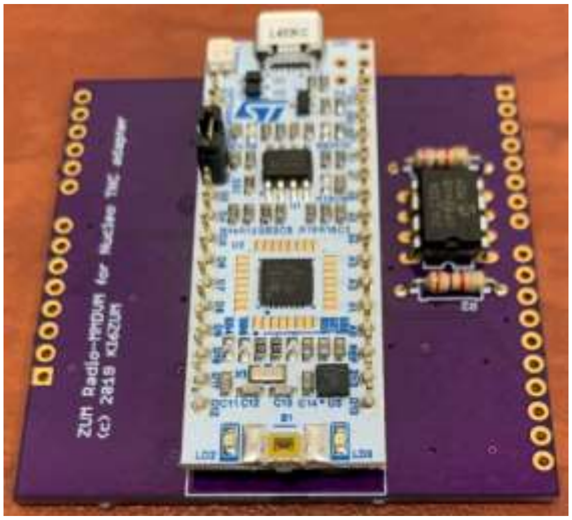
- Then click on **Apply and Close**
- Now when you Build you won’t see that error

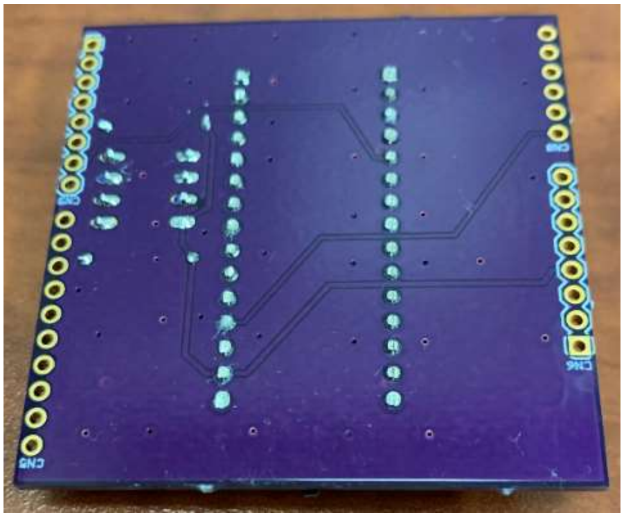
Assembling the NucleoTNC

These are the parts needed in order to use the MMDVM-Nucleo board with the NucleoTNC firmware.

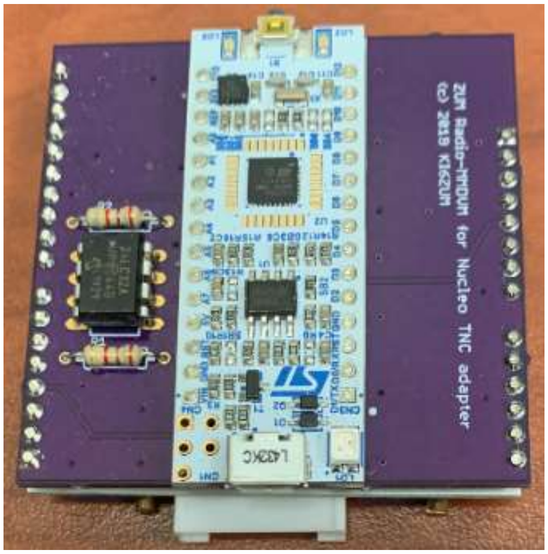
QTY	Part Name	Manufacturer	Part number	Link
1	MMDVM-Nucleo	ZUMRadio	MMDVM-Nucleo 1.0.2	HRO
1	NUCLEO-L432KC	ST Micro	NUCLEO-L432KC	Digi-Key
2	2.7K Resistors 1/4W	Stackpole	CF14JT2K70	Digi-Key
1	I2C EEPROM	Microchip	24LC32A-I/P	Digi-Key
1	Interface Board	OSH PARK	MMDVM_nucleo_TNC_0	OSH PARK

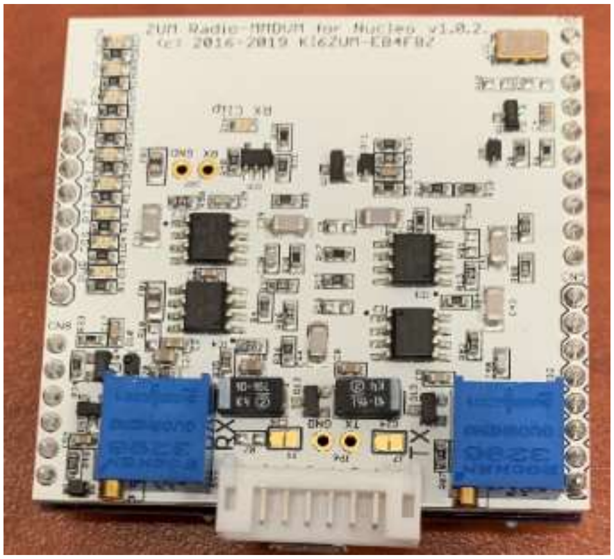
The first two images below show the Nucleo-L432 board as well as the EEPROM and resistors mounted on the top of the board and soldered on the bottom of the board. These components need to be soldered first since the MMDVM-Nucleo board will cover the pads. Make sure to trim the leads as short as possible to prevent the possibility of them shorting against the bottom of the MMDVM-Nucleo board.





These two images below show the MMDVM-Nucleo board mounted and soldered to the adapter board.





Flashing the NucleoTNC firmware

These steps are only for use with the NUCLEO-L432KC board.

- To flash the NucleoTNC firmware you must first either build the firmware following the steps in the section “Building the NucleoTNC firmware” or you can also download it from: https://github.com/mobilinkd/NucleoTNC/releases
- Once you have the BIN file, copy the firmware.bin file to the root of the Nucleo removable storage device. You should see the multi-color LED near the USB connector on the Nucleo flash while it is programming.
- Once you have accomplished that, you are ready to configure and use the TNC.

Configuring the NucleoTNC firmware

To configure the NucleoTNC firmware, follow the instructions in the “TNC Configuration” section of this NucleoTNC how-to guide:
https://github.com/mobilinkd/NucleoTNC/blob/master/Build/NucleoTNC.ipynb

Connecting the MMDVM-Nucleo board to radios

To connect the MMDVM-Nucleo board to a radio, follow the “Wiring connections” and “Adjusting signal levels” sections above.

Using mmdvmcal to calibrate the board

This is a web page with a good tutorial on how to use the mmdvmcal software tool and a spectrum analyzer to adjustment the signal levels for DMR:
https://www.f5uii.net/en/installation-calibration-adjustment-tunning-mmdvm-mmdvmhost-raspberry-motorola-gm360/5/

Another document describing the spectrum adjustment process:
http://www.swedmr.se/wp-content/uploads/2017/08/Justering-av-repeater-med-MMDVM.pdf

Support

Great video from W1MSG showing getting started with Pi-Star:
https://www.youtube.com/watch?v=B5G4gYDdJeQ

MMDVM Yahoo group:
https://groups.yahoo.com/neo/groups/mmdvm/conversations/messages

Pi-Star support forum:
https://forum.pistar.uk/

Pi-Star Facebook support group:
https://www.facebook.com/groups/pistar/

Pi-Star Wiki:
http://wiki.pistar.uk

A web page describing the setup of an MMDVM repeater:
https://sadigitalradio.com/digital-radio-how-tos/make-mmdvm-digital-repeater/