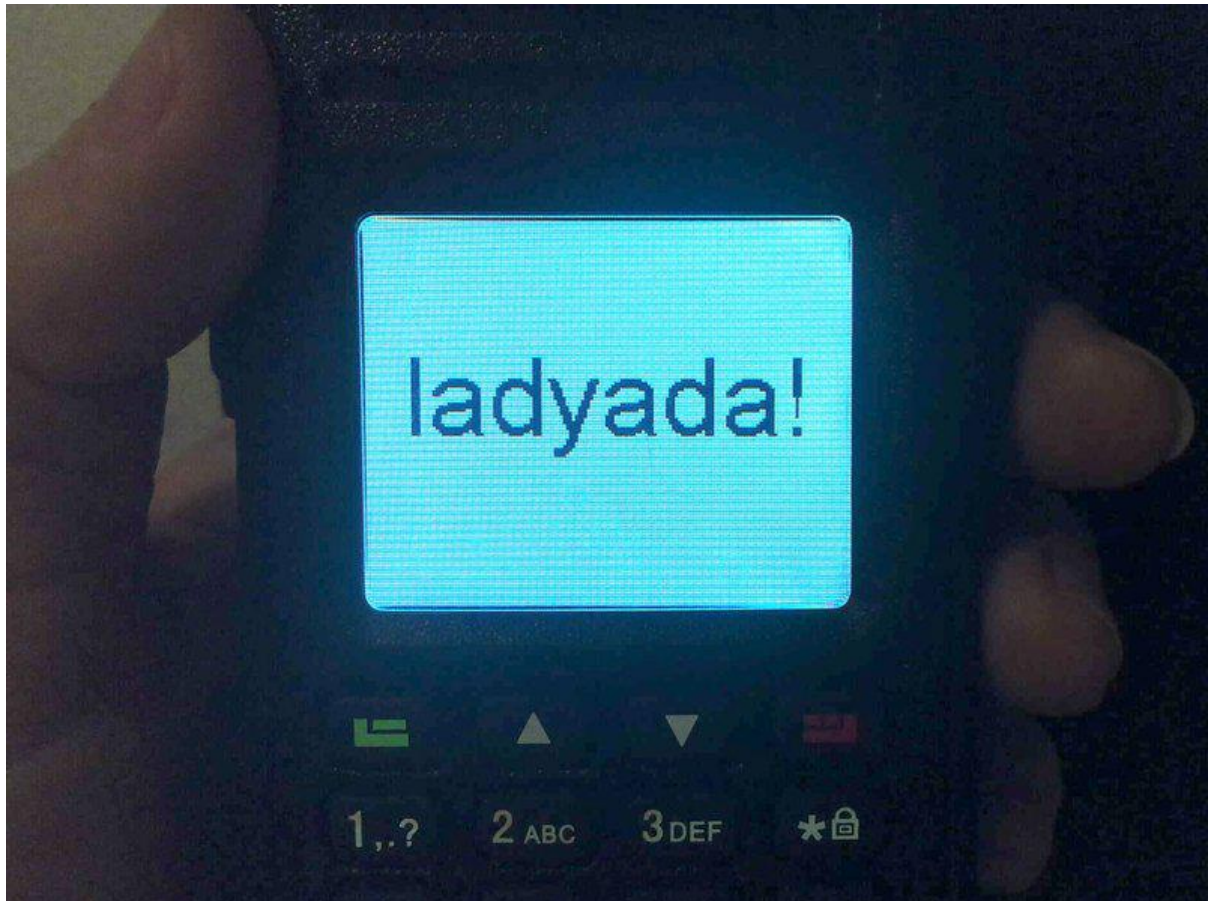




You and Your Tytera MD-380 DMR

Created by lady ada



<https://learn.adafruit.com/tytera-md-380-dmr>

Last updated on 2021-11-15 06:41:34 PM EST

Table of Contents

Overview	3
Get your DMR ID	3
Updating MD-380 Firmware	6
• DFU Driver	6
• Download MD380 Tools	7
• Downloading & Patching	8
• Uploading the Firmware	11
• Put the Radio into DFU Mode	13
FlashDB	14
• Install PyUSB	14
• Install a pyusb backend in cygwin	15
• Swapping Drivers	16
• Finally!	19
• Removing the libusbK driver	20
Finding a DMR Repeater	20
• Finding your local DMR Repeater	20
• Talk Groups	20
Programming	23
• Install TYT software	23
• Read in Radio Setup	26
• Fresh codeplug!	27
• Program in DMR ID	28
• Program in Contacts	29
• Entering Channels	30
• Set Name and Frequency	31
• Write Data	34
Custom Bootscreen	35
• Make a Bitmap	35
• Save Bitmap	36
• Open bitmap in GIMP	36
• Add checksum	37
• Edit Makefile	38

Overview

Hey everyone! It's me, ladyada! I recently got my Amateur Radio License ('HAM ticket') and I wanted to start out with a radio that's both inexpensive, easy to use, and fun to hack! After seeing some videos from CCC about the MD-380, I thought this radio is the one for me to start with.

Not only does this radio do FM analog on the 400MHz band, but it can also do DMR - digital encoded radio! This gives you the ability to use an Internet-backed voice network with no static and huge world-wide repeater reach.

I'm just getting started on my HAM-venture but I thought I'd take some notes on the hacking and programming I've done! Some of it, like updating the Firmware and flashing User DB, is optional - but it's also the most fun!

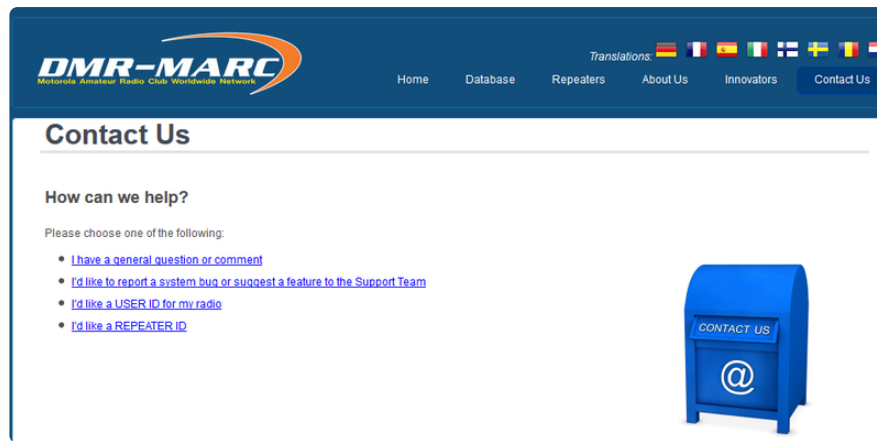
Tons of credit for all this stuff to Friendly Neighbor Travis & the MD380-Tools team at <https://github.com/travisgoodspeed/md380tools>, I'm just documenting how I used it under windows!

Get your DMR ID

OK so it's probably a good idea to apply for your DMR ID soonish since it may take a few days. Even though you have a call sign, that's not what is used for message routing on the DMR network. Instead, a unique identifier, granted by a central authority, lets you identify who is calling and who you want to call. It's a little like a phone number!

Visit <http://dmr-marc.net/> (<https://adafru.it/nqD>) to apply for your ID. You'll likely want one for each of your DMR radios

Click on Contact Us



Then select I'd like a USER ID for my radio

You'll get this lovely warning:

Requests for UPDATES to both Users and Repeaters

Please contact the ID Team: idteam@dmr-marc.net

***** DO NOT USE REGISTRATION FORMS BELOW TO MAKE CORRECTIONS, SEE THE LINE ABOVE *****

REPEATER registration requires a valid frequency coordination from the recognized Spectrum Management or Repeater Council.

By proceeding to the **USER** or **REPEATER** registration, you hereby agree to the above terms and conditions

Older versions of IE do not work well, please use a modern browser.

****** Registration process takes AT LEAST 24 HOURS, re-applying will not speed up your request ******

****** ALL @YAHOO MAIL WILL NOT BE DELIVERED, BLAME YAHOO. ******

Click on User Registration and enter in your amateur radio call sign

DMR USER Registration - Callsign Validation

Welcome to DMR User registration, before we proceed, we will validate your callsign.

Enter Callsign

Fill out the rest of the form

ADD DMR USER

Country

United States

Callsign

AC2SN

First Name

Limor

Surname

Fried

Nickname

ladyada

City

New York

State/Prov

(3136) New York

Radio Type

DMR

E-Mail Address

@

Comment

can't wait to try out DMR!

Register

When complete, click Register and you'll get a notice that registration completed

DMR USER Registration Results

Thank you, You will be contacted by email and provided with the ID you requested shortly.

****NOTE:** Please check the database for your ID before contacting the ID Team if you do not receive an email within 48hrs. We get blocked by mail servers as SPAM a lot because our emails are template driven and always the same. Also check your own SPAM folders.

DMR User Registration Request

DMR ID	Callsign	Name	Nickname	City	State/Prov	Country	Remarks
PENDING	AC2SN	Limor	Ladyada	New York	New York	United States	DMR

A few days later (please be patient with the people running the registration, they're volunteers!) you'll get an email with your DMR ID

```

Hello AC2SN,

Here is your requested User ID

ID: 3136923
Callsign: AC2SN
Name: Limor
City: New York
State: New York
Country: United States
Email: 
Remarks: DMR

You have been issued an International ID which works on the DMR networks and CCS7 networks. It will work
immediately on the DMR network. Allow several days for the CCS7 database to sync up and display your callsign.
If you have difficulties with CCS7, contact europa@opendmr.net

Thank You
DMR Registration System

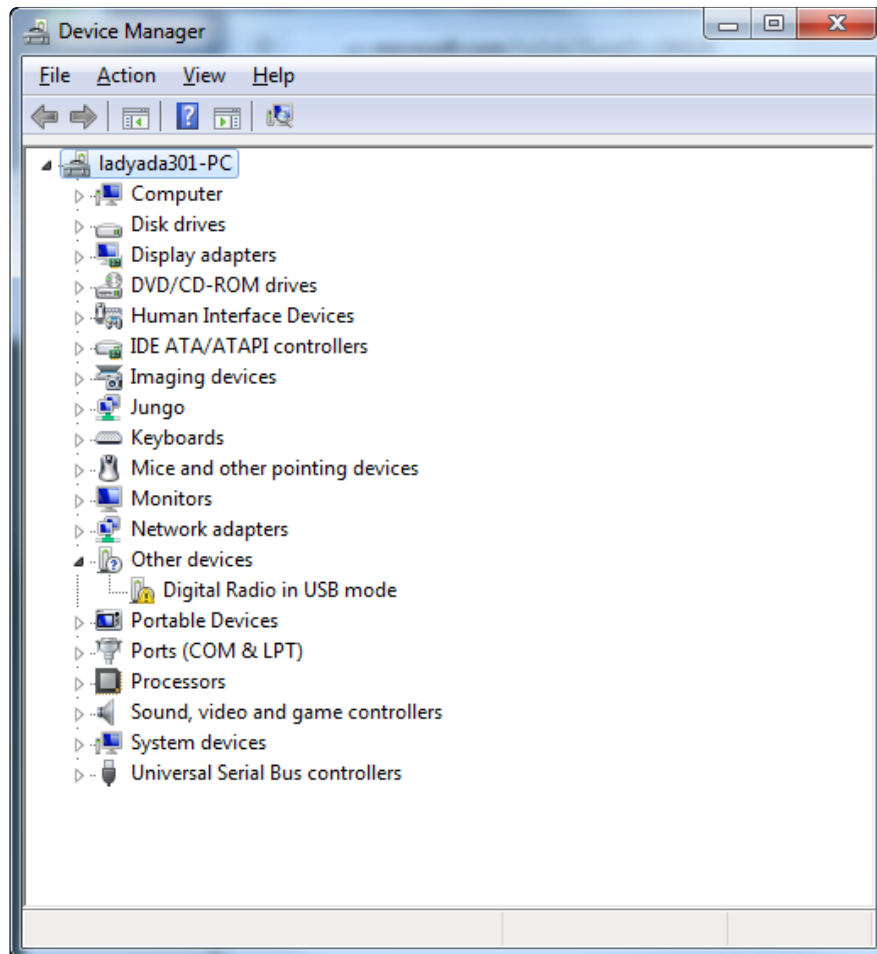
```

Note that your name, call sign and ID number are not private - they are visible in the DMR user database registry!

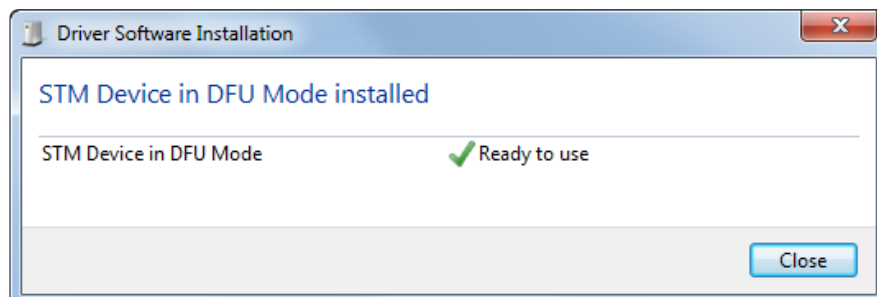
Updating MD-380 Firmware

DFU Driver

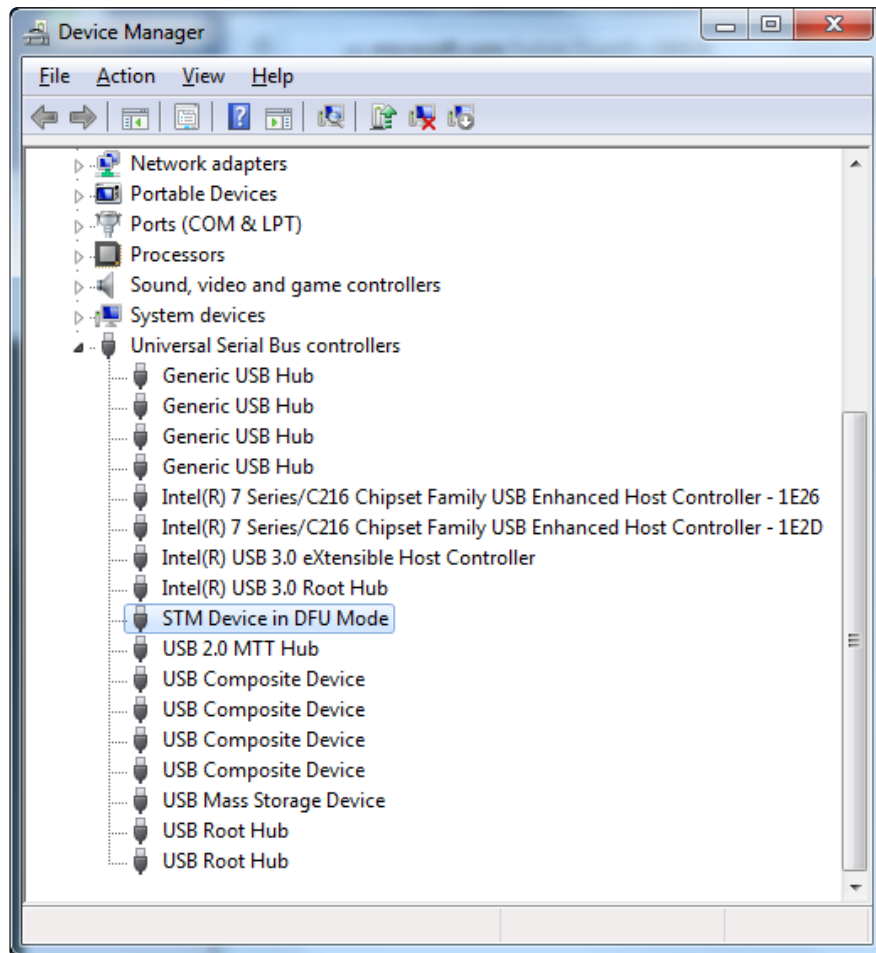
The good news is that the DFU driver is automatically found by Windows update. When you first plug in the radio, your device manager will say it's a Digital Radio in USB mode



And Windows update will search for the driver



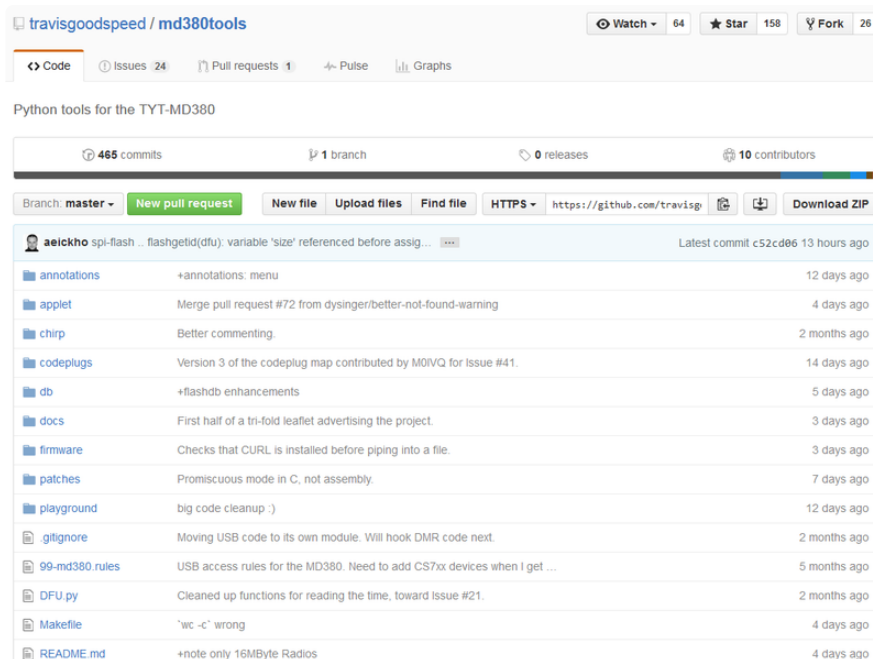
Once complete, the radio will come up as an STM DFU device



Download MD380 Tools

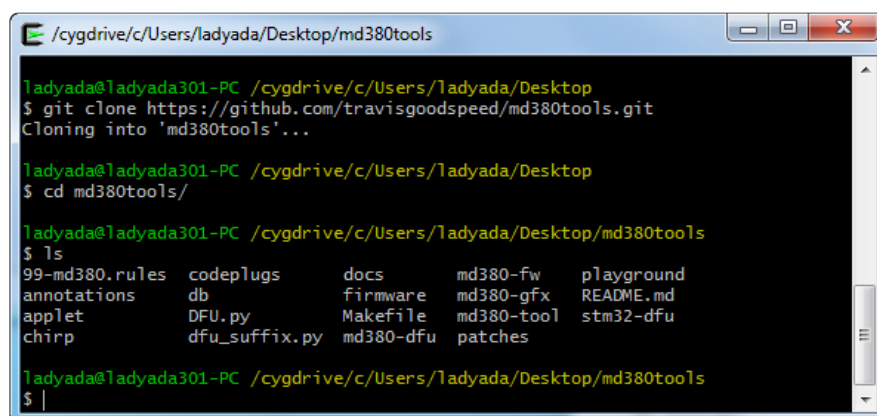
You'll need a bunch of tools and utilities to compile and upload the MD380 hacked firmware. For mac and linux, you probably have this already. [You can visit our Windows tools tutorial for information on how to install git and other command line tools \(https://adafru.it/nqE\)](https://adafru.it/nqE)

Once you have things like git and cygwin installed, [download the latest md380-tools codebase from Github \(https://adafru.it/nqF\)](https://adafru.it/nqF)



I used git within cygwin to

git clone https://github.com/travisgoodspeed/md380tools.git



and then cd into the new directory and make sure the list of files is similar to the above

Downloading & Patching

You can run the tool for downloading and patching the firmware by running

make clean dist

(which will clean up the current setup and run the distribution procedure)

When you run it you should get something like this:

```

Cygdrive/c/Users/ladyada/Desktop/md380tools
ladyada@ladyada301-PC: /cygdrive/c/Users/ladyada/Desktop/md380tools
$ make clean dist
cd patches/2.032 && make clean
make[1]: Entering directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/patches/2.032'
rm -f *.bin *.img
make[1]: Leaving directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/patches/2.032'
cd applet && make clean
make[1]: Entering directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/applet'
make -C lib clean
make[2]: Entering directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/applet/lib'
rm -f misc.o stm32f4xx_dma.o stm32f4xx_rc.o stm32f4xx_adc.o stm32f4xx_exti.o stm32f4xx_rng.o stm32f4xx
xx_flash.o stm32f4xx_rtc.o stm32f4xx_crc.o stm32f4xx_fsmc.o stm32f4xx_sdio.o stm32f4xx_cryp_aes.o stm32
m32f4xx_spi.o stm32f4xx_cryp.o stm32f4xx_hash.o stm32f4xx_syscfg.o stm32f4xx_cryp_des.o stm32f4xx_hash
X_tim.o stm32f4xx_cryp_tdes.o stm32f4xx_hash_shal.o stm32f4xx_usart.o stm32f4xx_dac.o stm32f4xx_i2c.o s
o stm32f4xx_dbgmcu.o stm32f4xx_iwdg.o stm32f4xx_dcmi.o stm32f4xx_pwr.o libstm32f4.a
make[2]: Leaving directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/applet/lib'
rm -f *.img *.bin *.elf *.hex *.o *.sym src/*~ src/version.h
make[1]: Leaving directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/applet'
rm -f *.pyc
cd firmware && make all
make[1]: Entering directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/firmware'
make[1]: Leaving directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/firmware'
cd patches/2.032 && make all
make[1]: Entering directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/patches/2.032'
cd ../.. /firmware && make D002.032.bin
make[2]: Entering directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/firmware'
make[2]: 'D002.032.bin' is up to date.
make[2]: Leaving directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/firmware'
../md380-fw --unwrap ../Firmware/D002.032.bin unwrapped.img
DEBUG: reading "../Firmware/D002.032.bin"
INFO: base address 0x00c000
INFO: length 0xf2c00
DEBUG: writing "unwrapped.img"
./patch.py
Creating patches from unwrapped.img.
Patching range from 0809c714 to 080d0f80 to FF.
Patching word at 0800c020 to 080fa969
Patching word at 0800c004 to 0809cf01
Patching hword at 0809cf00 to 4840
Patching hword at 0809cf02 to 2100
Patching hword at 0809cf04 to 3901
Patching hword at 0809cf06 to 4508
Patching hword at 0809cf08 to d100
Patching hword at 0809cf0a to 483c
Patching hword at 0809cf0c to 4700
Patching word at 0809cffc to 080fa969
Patching wide string at 080d14d8 to 'MD380Tools Ver.'.
Patching string at 080f9e4c to 'Travis Goodspeed KK4VCZ'.

```

[illegible]

```
/cygdrive/c/Users/ladyada/Desktop/md380tools
/bin/ld.exe: warning: lib\libstm32f4.a(stm32f4xx_rcc.o) uses 2-byte wchar_t yet the output is to use 4-
use of wchar_t values across objects may fail
c:/program files (x86)/gnu tools arm embedded/4.8 2014q1/bin/./lib/gcc/arm-none-eabi/4.8.3/./.././../
/bin/ld.exe: warning: lib\libstm32f4.a(stm32f4xx_gpio.o) uses 2-byte wchar_t yet the output is to use 4
use of wchar_t values across objects may fail
arm-none-eabi-objcopy -O binary main.elf main.img
arm-none-eabi-objdump -t main.elf >main.img.sym
cd ../patches/2.032 && make all
make[2]: Entering directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/patches/2.032'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/patches/2.032'
cp ../patches/2.032/patched.img ./experiment.img
# use an fresh experiment.img
cp ../patches/2.032/patched.img ./experiment.img
python2 merge.py experiment.img main.img 0x0809D000
Merging an applet.
Loading symbols from main.img.sym
Hooking a menu call.
Merging main.img into experiment.img at 0809D000
../md380-fw --wrap experiment.img experiment.bin
DEBUG: reading "experiment.img"
INFO: base address 0x800C000
INFO: length 0xF2C00
DEBUG: writing "experiment.bin"
make[1]: Leaving directory '/cygdrive/c/Users/ladyada/Desktop/md380tools/applet'
rm -rf md380tools-'date +%Y-%m-%d'
mkdir -p md380tools-'date +%Y-%m-%d'/python
cp applet/experiment.bin md380tools-'date +%Y-%m-%d'/firmware-'date +%Y-%m-%d'.bin
cd md380tools-'date +%Y-%m-%d' && unzip ../firmware/D002.032.zip
Archive: ../Firmware/D002.032.zip
  creating: Firmware 2.32/
  inflating: Firmware 2.32/MD-380-D2.32(AD).bin
  inflating: Firmware 2.32/MSVCRT.DLL
  inflating: Firmware 2.32/Operation.doc.pdf
  inflating: Firmware 2.32/STDFU.dll
  inflating: Firmware 2.32/STTubeDevice30.dll
  inflating: Firmware 2.32/Upgrade.exe
  inflating: Firmware 2.32/mfc42.dll
  inflating: Firmware 2.32/setting.ini
  inflating: Firmware 2.32/setup.log
  inflating: Firmware 2.32/uninstall.exe
mv md380tools-'date +%Y-%m-%d'/Firmware 2.32 md380tools-'date +%Y-%m-%d'/windows
rm md380tools-'date +%Y-%m-%d'/windows/MD-380-D2.32(AD).bin md380tools-'date +%Y-%m-%d'/windows/O
df
cp DFU.py 99-md380.rules md380-dfu md380-tool md380tools-'date +%Y-%m-%d'/python/

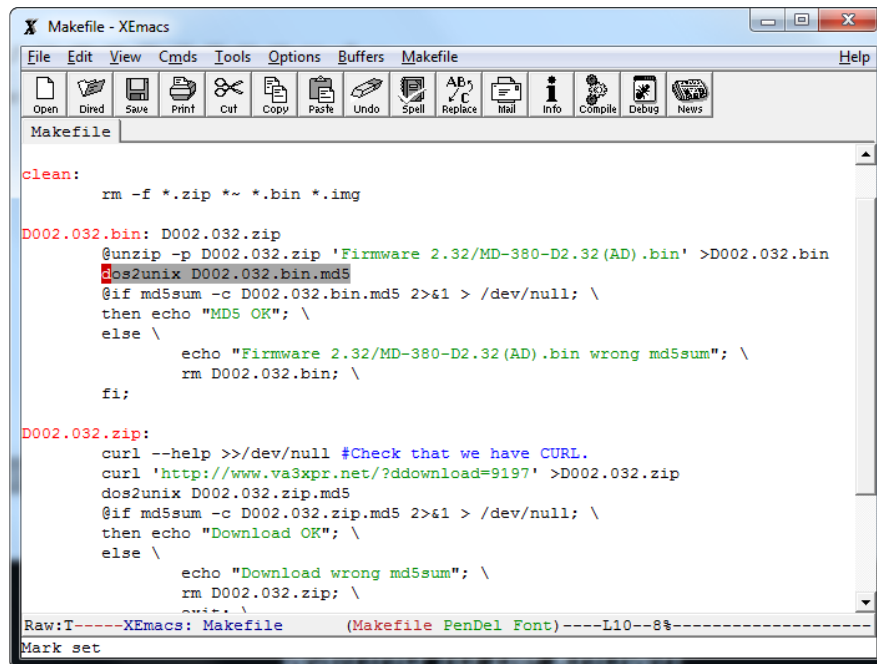
ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Desktop/md380tools
$
```

If you are running cygwin and get an md5sum error like this

```
/cygdrive/c/Users/ladyada/Dropbox (Personal)/RF/test/md380tools
rm -f *.img *.bin *.elf *.hex *.o ~* *.sym src/~* src/version.h
make[1]: Leaving directory '/cygdrive/c/Users/ladyada/Dropbox (Personal)/RF/test/md380tools/applet'
rm -f ~* *.pyc
cd firmware && make all
make[1]: Entering directory '/cygdrive/c/Users/ladyada/Dropbox (Personal)/RF/test/md380tools/firmware'
curl --help >>/dev/null #Check that we have CURL.
curl 'http://www.va3xpr.net/?download=9197' ->D002.032.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total   Spent    Left   Speed
100 2319k  100 2319k    0     0 3164k      0  --:--:-- --:--:-- --:--:-- 3164k
md5sum: 'D002.032.zip'$\r': No such file or directory
md5sum: WARNING: 1 listed file could not be read
Download wrong md5sum
Makefile:9: recipe for target 'D002.032.bin' failed
make[1]: *** [D002.032.bin] Error 9
make[1]: Leaving directory '/cygdrive/c/Users/ladyada/Dropbox (Personal)/RF/test/md380tools/firmware'
Makefile:15: recipe for target 'firmwares' failed
make: *** [firmwares] Error 2

ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox (Personal)/RF/test/md380tools
$
```

Then you can edit the Makefile in firmwares and add `dos2unix D002.032.bin.md5` and `dos2unix D002.032.zip.md5` before the md5sum commands to clean up the dos line endings.



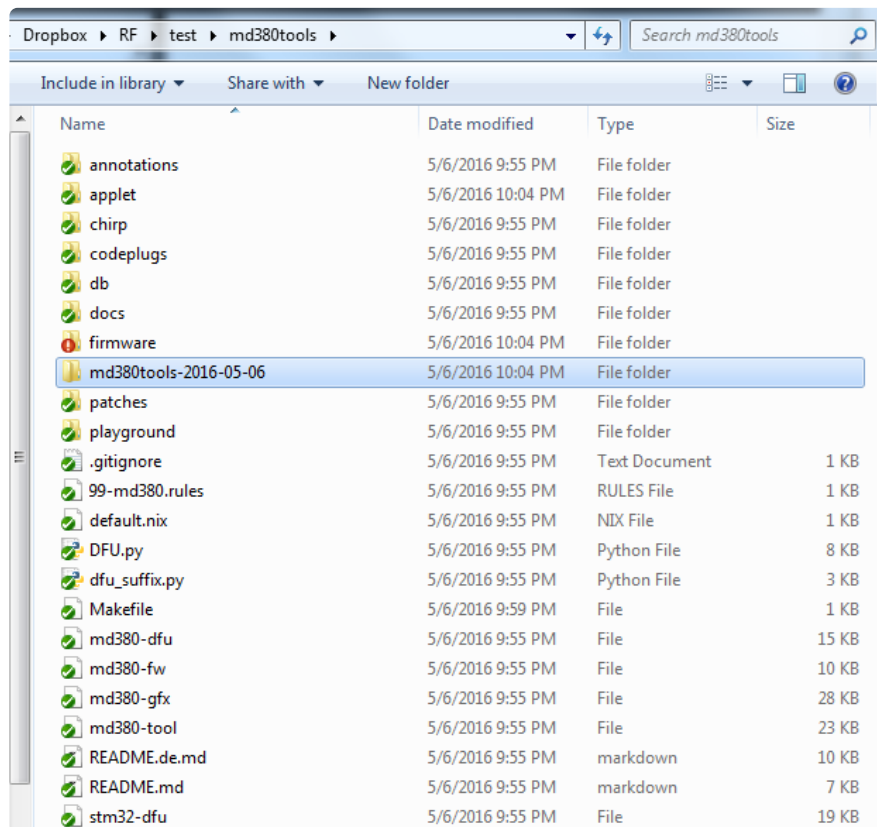
```
clean:
    rm -f *.zip *~ *.bin *.img

D002.032.bin: D002.032.zip
    @unzip -p D002.032.zip 'Firmware 2.32/MD-380-D2.32 (AD).bin' >D002.032.bin
    dos2unix D002.032.bin.md5
    @if md5sum -c D002.032.bin.md5 2>&1 > /dev/null; \
    then echo "MD5 OK"; \
    else \
        echo "Firmware 2.32/MD-380-D2.32 (AD).bin wrong md5sum"; \
        rm D002.032.bin; \
    fi;

D002.032.zip:
    curl --help >>/dev/null #Check that we have CURL.
    curl 'http://www.va3xpr.net/?ddownload=9197' >D002.032.zip
    dos2unix D002.032.zip.md5
    @if md5sum -c D002.032.zip.md5 2>&1 > /dev/null; \
    then echo "Download OK"; \
    else \
        echo "Download wrong md5sum"; \
        rm D002.032.zip; \
    fi;
```

Uploading the Firmware

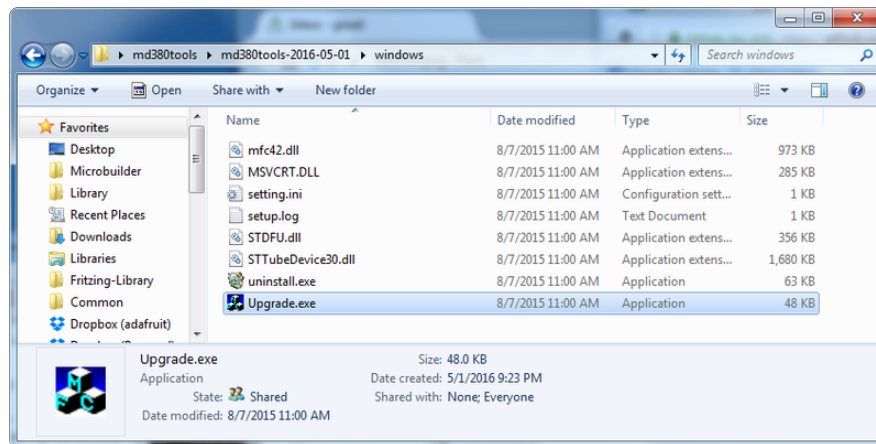
OK now you're ready to upload the firmware. Check for a newly created folder md380tools-yyyy-mm-dd with the current date



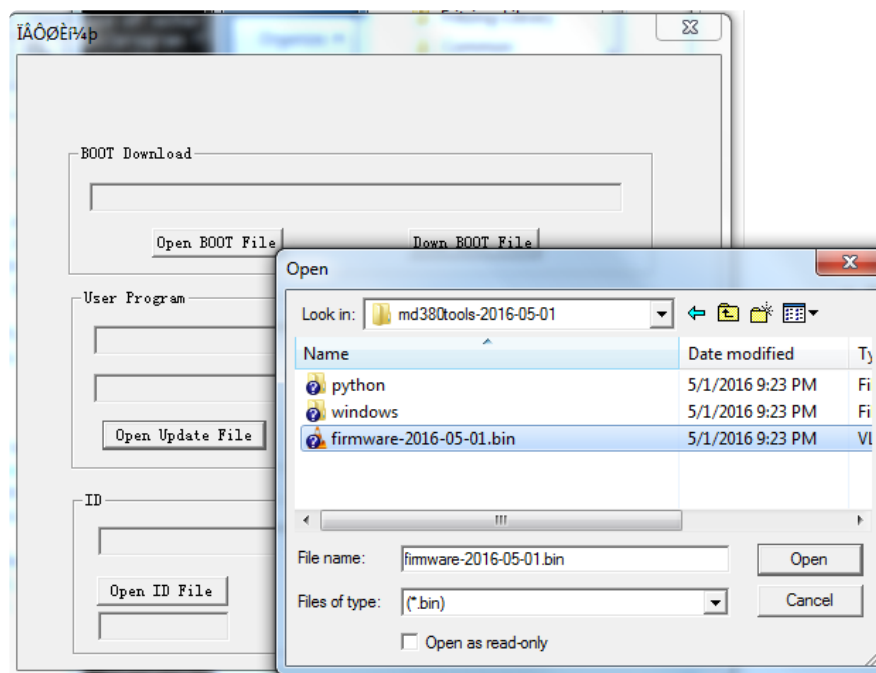
Inside you'll find your freshly patched firmware

Name	Date modified	Type	Size
python	5/1/2016 9:23 PM	File folder	
windows	5/1/2016 9:23 PM	File folder	
firmware-2016-05-01.bin	5/1/2016 9:23 PM	VLC media file (.bin)	972 KB

And inside the windows directory you'll even get the upgrade.exe tool from Tytera



Run this program - it's in half Mandarin half English but that's OK. Click Open Update File and select that firmware from the upper directory

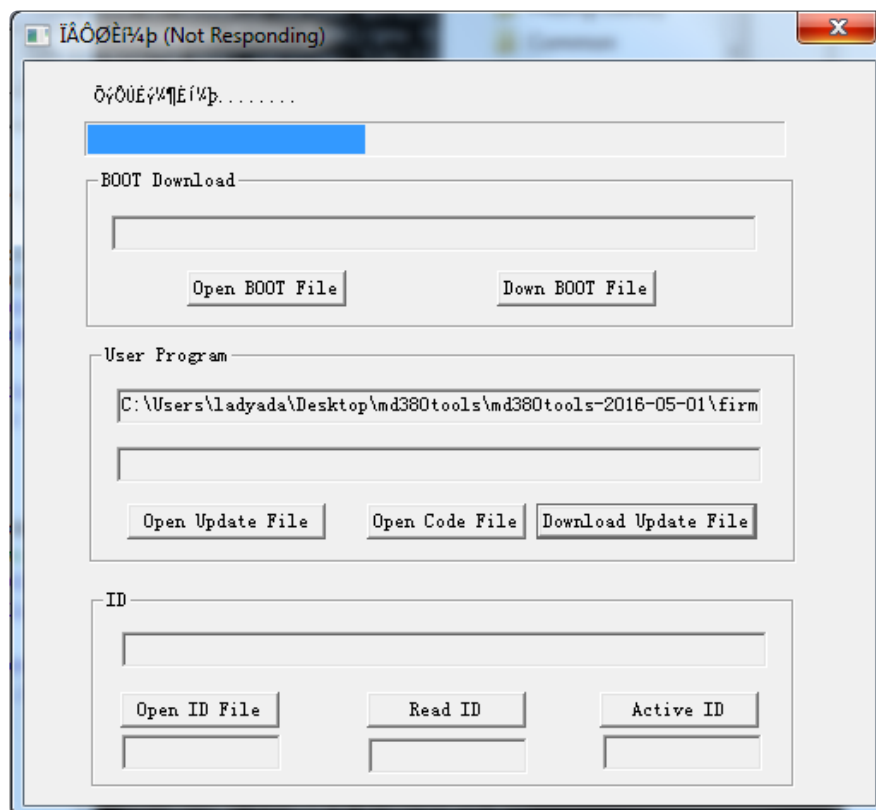


Put the Radio into DFU Mode

OK right before you actually download the firmware into your radio you'll need to put it into DFU mode - so that it accepts new firmware (not the same mode as for updating the settings!)

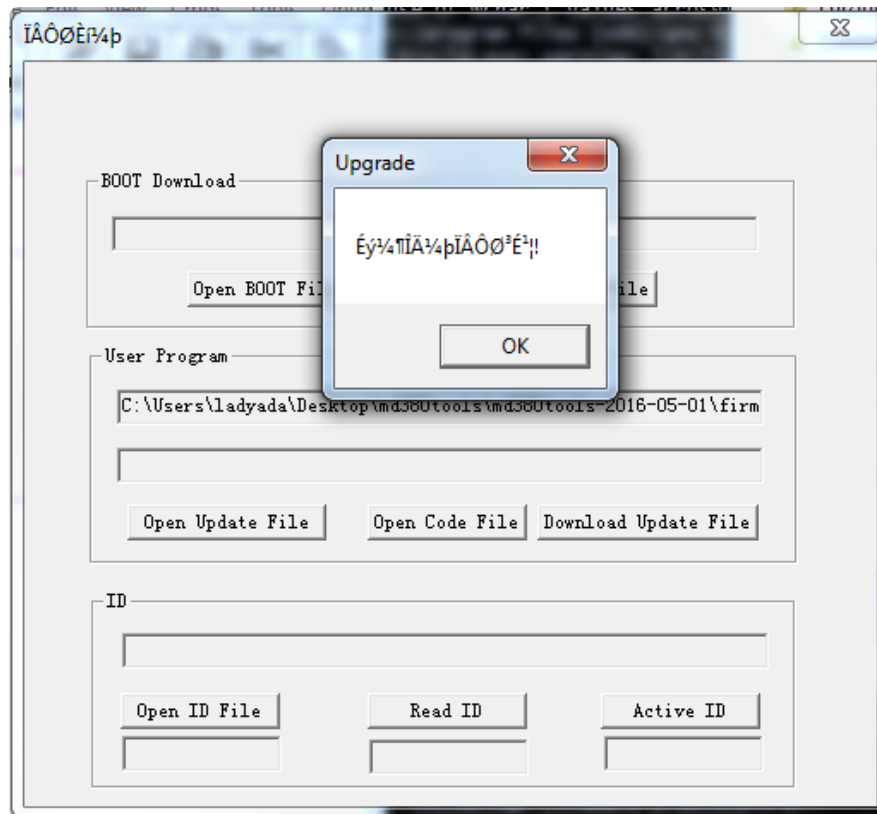
1. Turn the radio off (using the volume knob)
2. Connect the USB programming cable to your computer
3. Peel off the headphone/mic insert & plug in the other side of the cable (it can only go one way)
4. Hold down the PTT and top button
5. Turn on the radio while the buttons are held
6. You will see the LED blink red/green

Then click Download Update File



Give it a few minutes, it will look like it 'hangs' but its working, it is just a little longer than you think it'll take!

On success you'll get this message, which means it worked



You may have to turn off/on the radio to launch the new firmware

FlashDB

Installing the Users CSV and otherwise talking to the radio is done with md380-tools which is a python script that sends and receives USB data from the radio. It does take a little squidding to get working nicely on Windows under cygwin but lucky for you I had every possible issue come up!

Try running make flashdb to download the latest user and repeater database and uploading it to your radio

Install PyUSB

If your python install complains No module named usb.core you'll need to install [pyusb v1.0.0a2](https://adafru.it/nF1) (<https://adafru.it/nF1>)

```
/cygdrive/c/Users/ladyada/Dropbox/RF/md380tools

ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox/RF/md380tools
$ make flashdb
cd db && make clean update
make[1]: Entering directory '/cygdrive/c/Users/ladyada/Dropbox/RF/md380tools/db'
rm -f users.csv repeaters.csv users.json repeaters.json
curl 'http://www.dmr-marc.net/cgi-bin/trbo-database/datadump.cgi?table=users&format=csv&header=0' | sed 's,<br/>,' >users.csv
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 2517k    0 2517k    0     0  1991k      0  --:--:--  0:00:01 --:--:-- 2016k
make[1]: Leaving directory '/cygdrive/c/Users/ladyada/Dropbox/RF/md380tools/db'
cat db/users.csv | cut -d',' -f1-3,5-6 | sed 's/,\\s+/,/g' > data.csv
wc -c < data.csv > data
cat data.csv >> data
./md380-tool spiflashwrite data 0x100000
Traceback (most recent call last):
  File "./md380-tool", line 12, in <module>
    import time, sys, struct, usb.core
ImportError: No module named usb.core
Makefile:19: recipe for target 'flashdb' failed
make: *** [flashdb] Error 1

ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox/RF/md380tools
```

Download it into your working directory, uncompress and run python setup.py install in the cygwin terminal to install it

```
/cygdrive/c/Users/ladyada/Dropbox/RF/md380tools

ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox/RF/pyusb-1.0.0a2
$ python setup.py install
running install
running build
running build_py
creating build
creating build/lib
creating build/lib/usb
copying usb/control.py -> build/lib/usb
copying usb/core.py -> build/lib/usb
copying usb/legacy.py -> build/lib/usb
copying usb/util.py -> build/lib/usb
copying usb/_debug.py -> build/lib/usb
copying usb/_interop.py -> build/lib/usb
copying usb/_init_.py -> build/lib/usb
creating build/lib/usb/backend
copying usb/backend/libusb01.py -> build/lib/usb/backend
copying usb/backend/libusb10.py -> build/lib/usb/backend
copying usb/backend/openusb.py -> build/lib/usb/backend
copying usb/backend/_init_.py -> build/lib/usb/backend
running install_lib
creating /usr/lib/python2.7/site-packages/usb
creating /usr/lib/python2.7/site-packages/usb/backend
```

Install a pyusb backend in cygwin

If you run make flashdb you'll note that it may say No backend available - that's because you need to [install libusb 1.0.20 support in cygwin](https://adafru.it/nF2) (<https://adafru.it/nF2>).

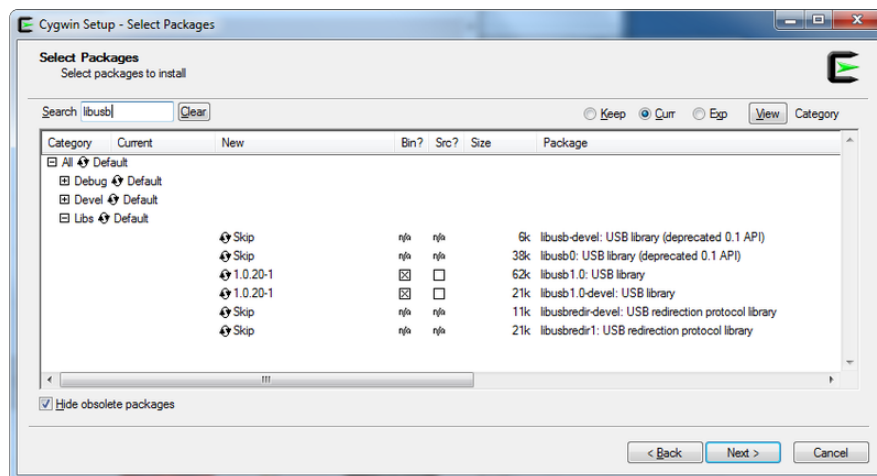
```

/cygdrive/c/Users/ladyada/Dropbox/RF/md380tools
ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox/RF/md380tools
$ make flashdb
cd db && make clean update
make[1]: Entering directory '/cygdrive/c/Users/ladyada/Dropbox/RF/md380tools/db'
rm -f users.csv repeaters.csv users.json repeaters.json
curl 'http://www.dmr-marc.net/cgi-bin/trbo-database/datadump.cgi?table=users&format=csv&header=0' | sed 's,<br/>,,>' >users.csv
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 2517k    0 2517k    0    0 2446k    0 --:--:--  0:00:01 --:--:-- 2446k
make[1]: Leaving directory '/cygdrive/c/Users/ladyada/Dropbox/RF/md380tools/db'
cat db/users.csv | cut -d',' -f1-3,5-6 | sed 's/,\\s+/,/g' > data.csv
wc -c < data.csv > data
cat data.csv >> data
./md380-tool spiFlashwrite data 0x100000
No backend available
Makefile:19: recipe for target 'flashdb' failed
make: *** [flashdb] Error 1

ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox/RF/md380tools
$ |

```

So go back to cygwin and do so



Swapping Drivers

OK finally you'll get a complaint that

[Errno 88] Operation not supported or unimplemented on this platform

```

/cygdrive/c/Users/ladyada/Dropbox/RF/master_md380tools
ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox/RF/master_md380tools
$ ./md380-tool spiFlashwrite data 0x100000
[Errno 88] Operation not supported or unimplemented on this platform

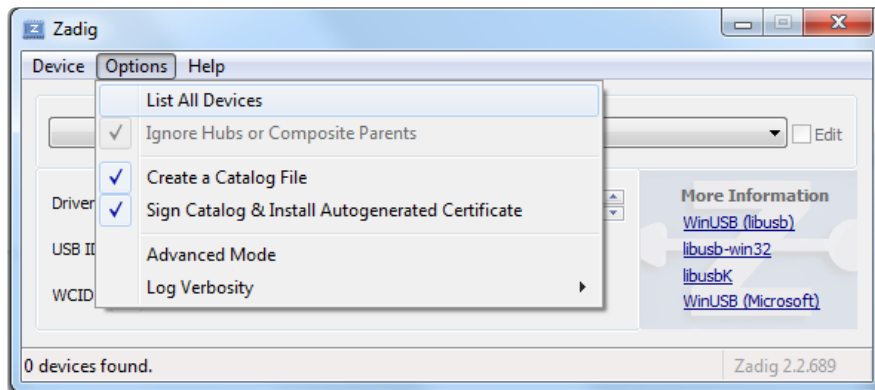
ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox/RF/master_md380tools
$ |

```

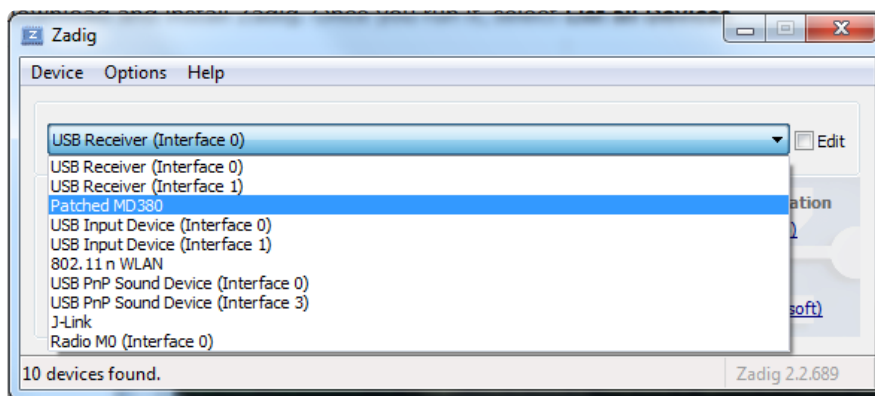
This is actually good, it means libusb loaded fine, but it can't talk to the radio. That's because the STM32 DFU driver has full control of the USB interface. What you need to do is replace that driver with a libUSB driver that we can poke at. Windows is funny that way.

The easiest way to do that is to [run Zadig](https://adafruit.it/eal) (<https://adafruit.it/eal>), which is a tool that will let you flip from the STM driver (the 'official' driver) to a libusb-compatible driver that lets you poke at the device directly.

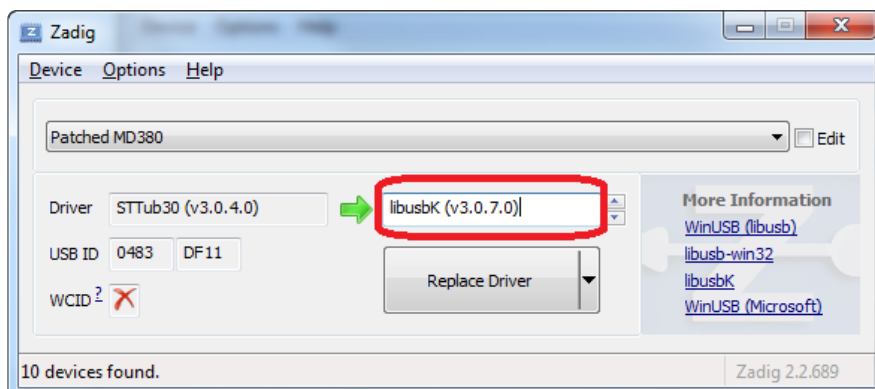
Download and install Zadig. Once you run it, select List all Devices



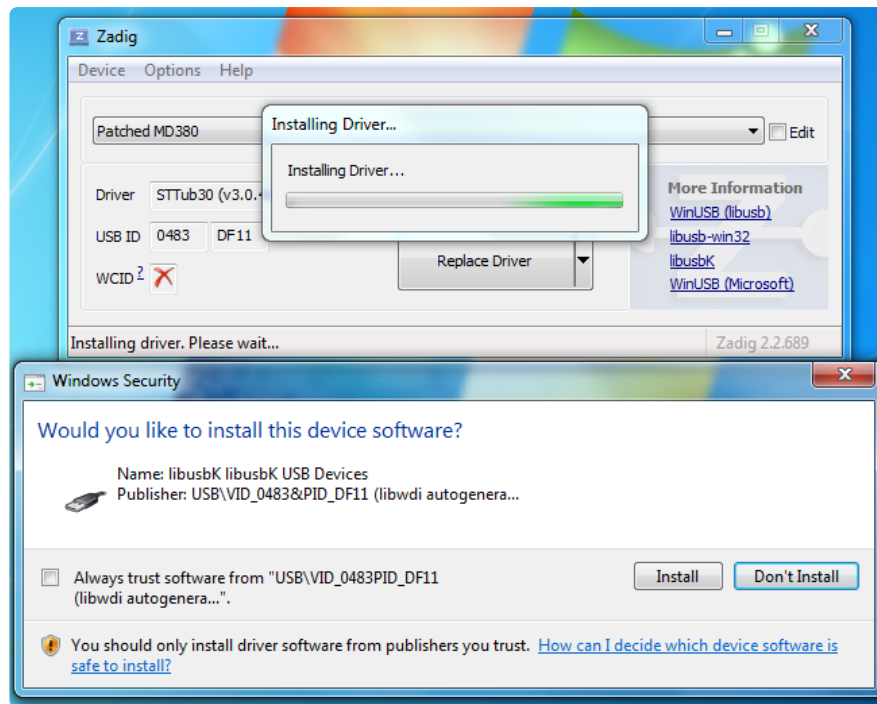
Then from the collection of devices, find Patched MD380 and select that



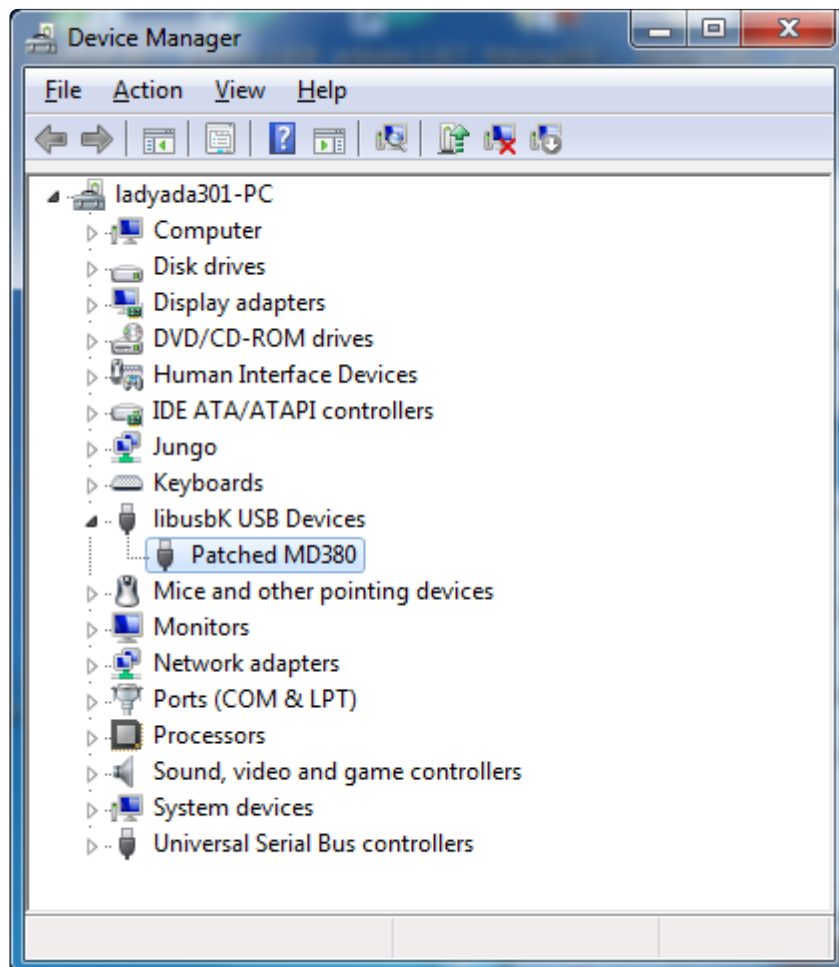
On the left, you'll see the current driver, STTub30 (the ST driver) and on the right hand side, change it to libusbK



Then click Replace Driver and replace it



Now in the device driver, you'll see the driver is under libusbK!



Finally!

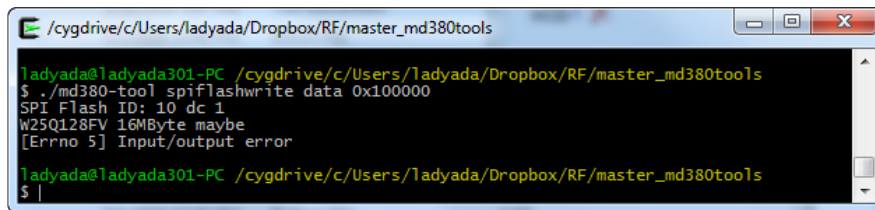
Run make flashdb again, this time you will have success!

```
/cygdrive/c/Users/ladyada/Dropbox/RF/master_md380tools
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
             Dload  Upload    Total      Spent    Left     Speed
100 560k    0 560k    0 0    1197k      0  --:--:--  --:--:--  --:--:-- 1197k
curl 'http://www.dmr-marc.net/cgi-bin/trbo-database/datadump.cgi?table=users&format=json&header=0' >users.json
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
             Dload  Upload    Total      Spent    Left     Speed
100 5951k    0 5951k    0 0    2847k      0  --:--:--  0:00:02  --:--:-- 2847k
curl 'http://www.dmr-marc.net/cgi-bin/trbo-database/datadump.cgi?table=repeaters&format=json&header=0' >repeaters.json
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
             Dload  Upload    Total      Spent    Left     Speed
100 950k    0 950k    0 0    1647k      0  --:--:--  --:--:--  --:--:-- 1647k
make[1]: Leaving directory '/cygdrive/c/Users/ladyada/Dropbox/RF/master_md380tools/db'
cat db/users.csv | cut -d',' -f1-3,5-6 | sed 's/,/s+/g' > data.csv
wc -c < data.csv > data
cat data.csv >> data
./md380-tool spiFlashwrite data 0x100000
SPI Flash ID: 10 dc 1
W25Q128FV 16MByte maybe
erase 100000
erase 101000
erase 102000
erase 103000
erase 104000
erase 105000
erase 106000
erase 107000
erase 108000
erase 109000
erase 10a000
erase 10b000
erase 10c000
erase 10d000
erase 10e000
erase 10f000
erase 110000
erase 111000
erase 112000
erase 113000
erase 114000
erase 115000
erase 116000
erase 117000
erase 118000
erase 119000
erase 11a000
erase 11b000
erase 11c000
```

```
/cygdrive/c/Users/ladyada/Dropbox/RF/master_md380tools
1024
1033
(OK, 0, dfuDNLOAD_IDLE, 0)
1707 1705 2aa400 1024
1024
1033
(OK, 0, dfuDNLOAD_IDLE, 0)
1707 1706 2aa800 1024
1024
1033
(OK, 0, dfuDNLOAD_IDLE, 0)
1707 2aac00 729
729
738
(OK, 0, dfuDNLOAD_IDLE, 0)
reboot radio now
ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox/RF/master_md380tools
$ |
```

If you get **[Errno 5] Input/output error**

Try restarting the radio (turn off/on) also check that you didn't accidentally install the libusb32 driver, you have to use libusbK!



```
ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox/RF/master_md380tools
$ ./md380-tool spiFlashwrite data 0x100000
SPI Flash ID: 10 dc 1
W25Q128FV 16MByte maybe
[Errno 5] Input/output error
ladyada@ladyada301-PC /cygdrive/c/Users/ladyada/Dropbox/RF/master_md380tools
$
```

Removing the libusbK driver

Simply uninstall + delete the driver in the device manager, restart the radio and you'll get back to the STM DFU driver automatically

Finding a DMR Repeater

Finding your local DMR Repeater

OK so now you have your radio up and running, and you want to [contact/talk/QSO](https://adafru.it/nFO) (<https://adafru.it/nFO>) with other friendly neighbors!

While you can talk directly, the power of the DMR network is that you can reach much farther with a repeater. Unlike most(?) analog repeaters, where you can only chat with other people who are accessing the same repeater, digital repeaters can connect thru the Internet to each other to create a huge network. For example, I can talk to someone in Boston from New York (normally not something a repeater could reach) by connecting to my local NYC repeater which then tunnels data to and from a Boston repeater.

So the most important thing is to find a repeater or two in your location that you can use to contact and program that into your radio.

Talk Groups

Since there's a huge network of hundreds of DMR repeater/routers and thousands of users, all essentially on one net, there's a desire to split off the groups so that when you send a message it goes to the smallest group necessary to still reach who you want. I think of it like a "chat room" channel

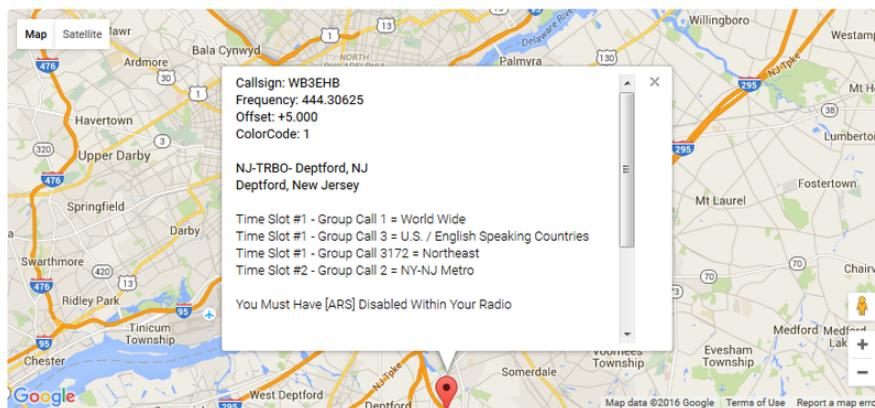
For example, if you want to talk to people in NYC only, you would talk on the NYC-local talk group. If you want to talk to people in New York state, join the NY talk group.

Tri-state? yep. New England? sure! USA? North America...all the way up to worldwide. And of course each city and state and region has their own talk groups as well.

If you have a person in particular you want to chat with, you'll need to figure out the minimal talk group you share in common with the repeaters you have available

Ideally, you would not have to use a "North America" talk group if you can avoid it since every repeater in the US+Canada would have to transmit your message.

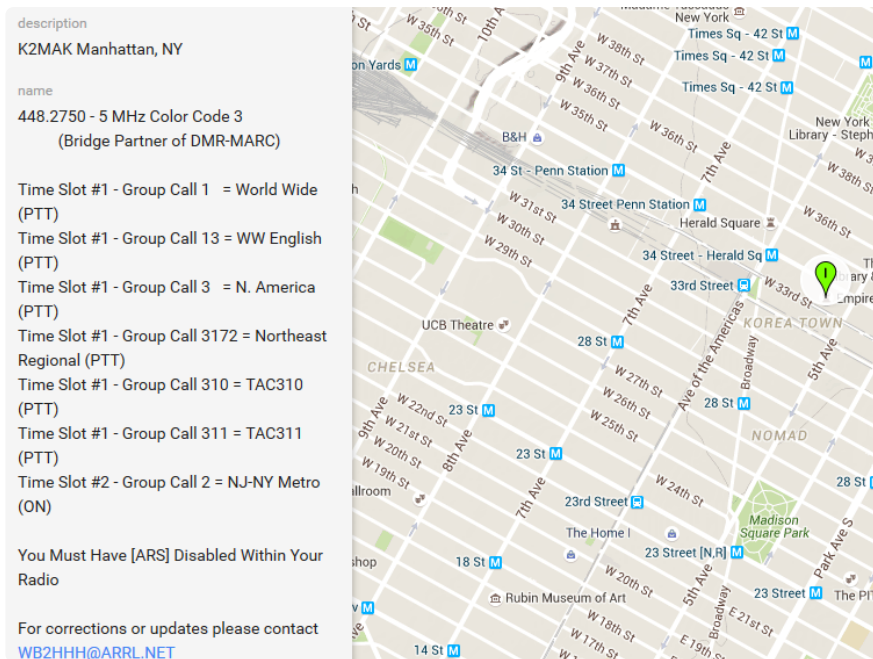
For example - Lets say I want to talk to my friendly neighbor in Philadelphia. Here's the repeater he can use:



This repeater WB3EHB carries 4 TalkGroups:

- Group 1 (World wide)
- Group 3 (US + English)
- Group 3172 (North East)
- Group 2 (NY-NJ Metro)

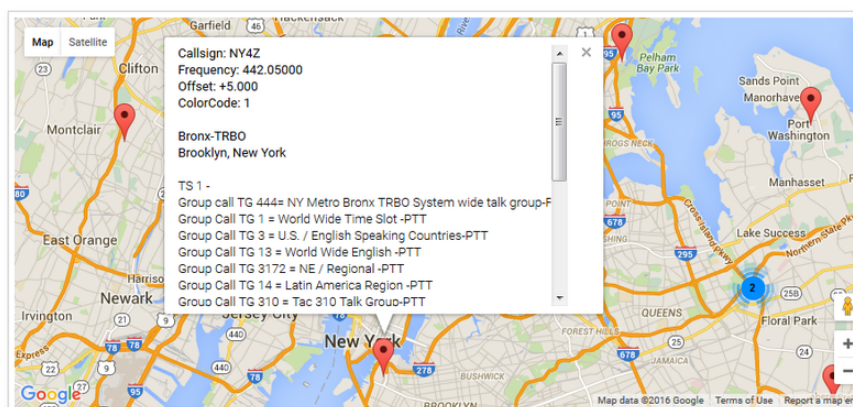
Lets look at the repeater I have in my backyard, K2MAK



This repeater carries even more talk groups:

- Group 1 (World wide)
- Group 13 (World wide English)
- Group 3 (US + English / North America)
- Group 3172 (North East)
- Group 2 (NY-NJ Metro)
- Two TAC groups (wont be covered here)

There's also a repeater near me in Brooklyn, NY4Z



This repeater carries similar, but not identical talk groups to the NYC one in midtown:

- Group 1 (World wide)
- Group 3 (US + English / North America)
- Group 13 (World wide English)

- Group 14 (Latin America)
- Group 3172 (North East)
- Group 4444 (Bronx TRBO only)
- Two TAC groups (wont be covered here)

The difference between this repeater and the midtown one are that this one carries #4444 and #13 but does not carry #2.

Since my friendly neighbor is in Philly and the 'smallest group' we could both share is TG2 (NY-NJ Metro) I am best off going with the K2MAK repeater if I want to talk to him since its the closest repeater that shares that talk group

Next up we'll program that into our radio!

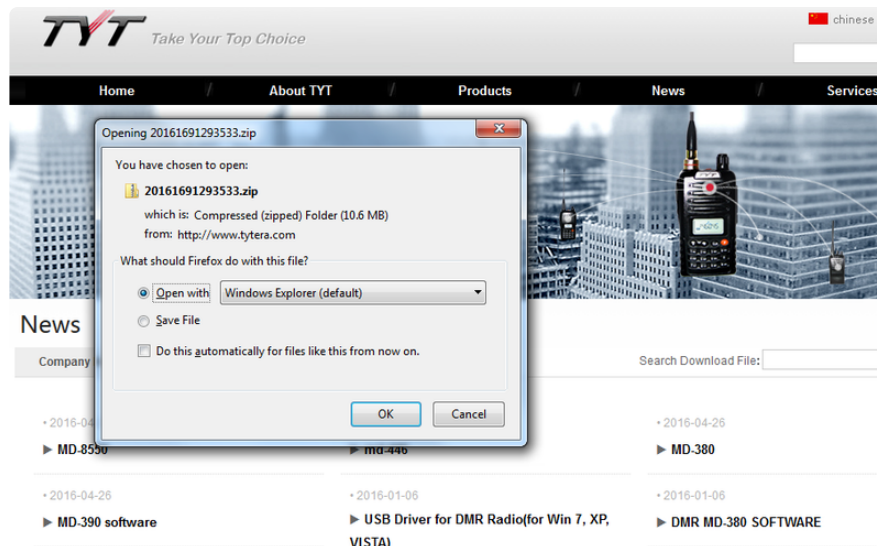
Programming

OK so unlike many simpler & analog radios, you can't just type in the frequencies you want to use into the keypad and off you go. Instead, you must run communication software that will set the repeaters you will be using. You can assign various repeaters to up to 16 different channels. You can also program in analog frequencies (both simplex & duplex)

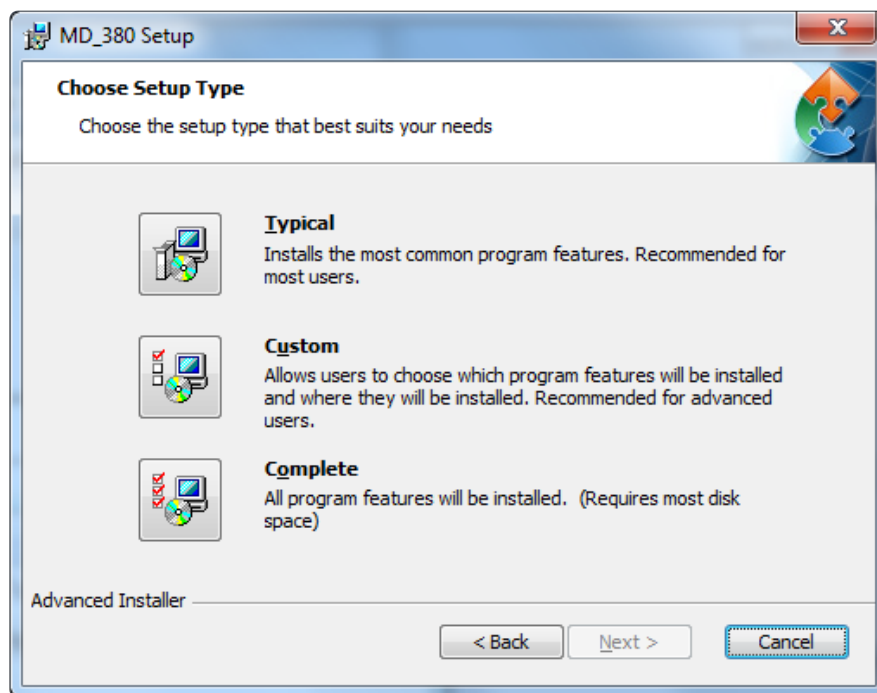
The good news is once you get your software file lined up, you can constantly tweak and re-program your radio until its just the way you like it.

Install TYT software

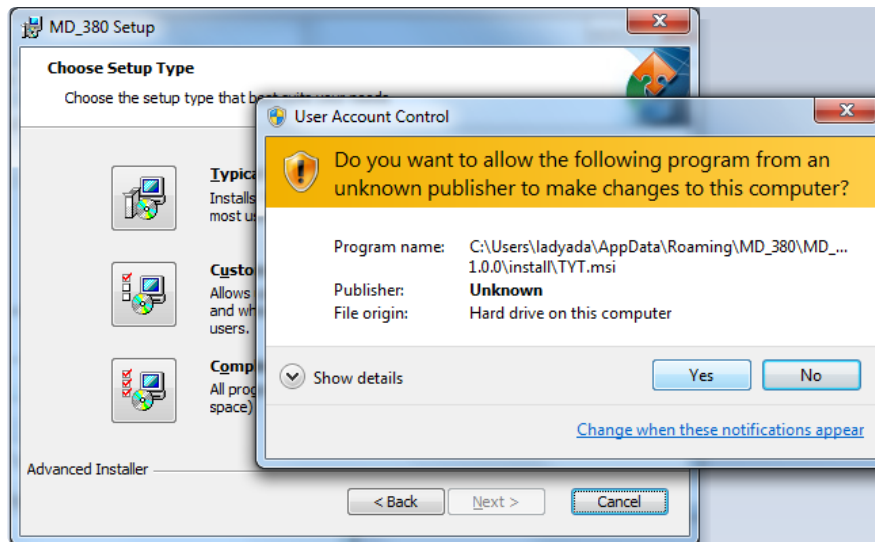
You'll use the official TYT interface software to update the settings, download and install from <http://www.tyt888.com/>



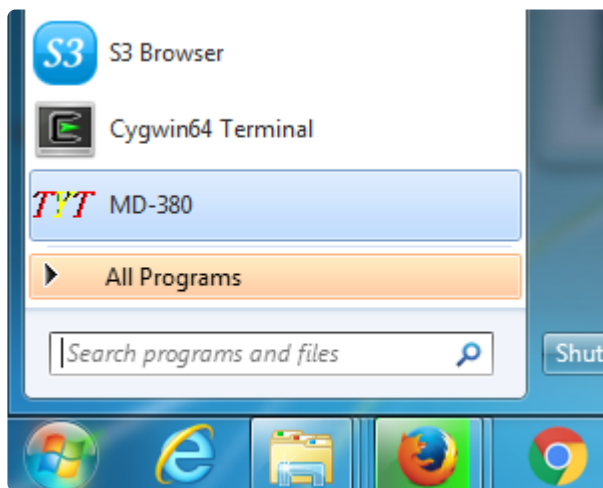
Uncompress and install



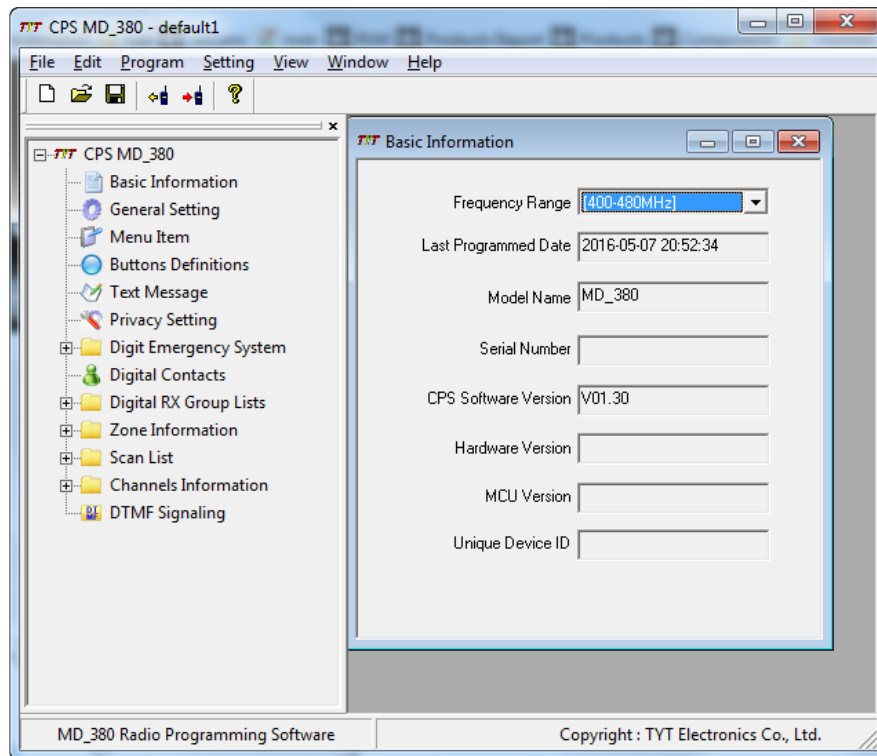
It's not signed, you may have to turn off program signing on your computer!



Once installed, run...

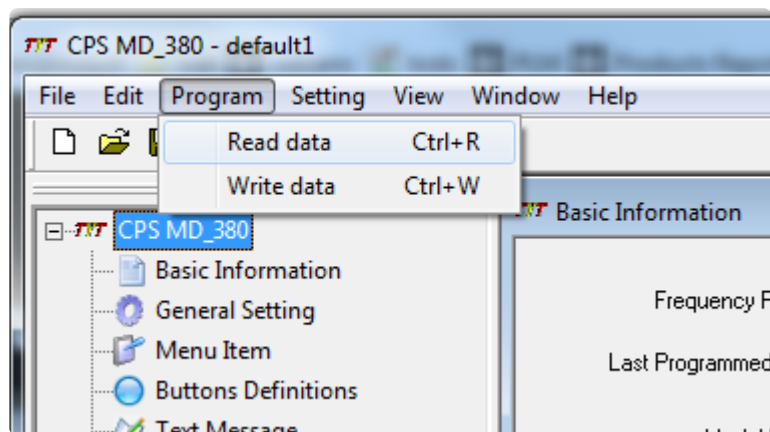


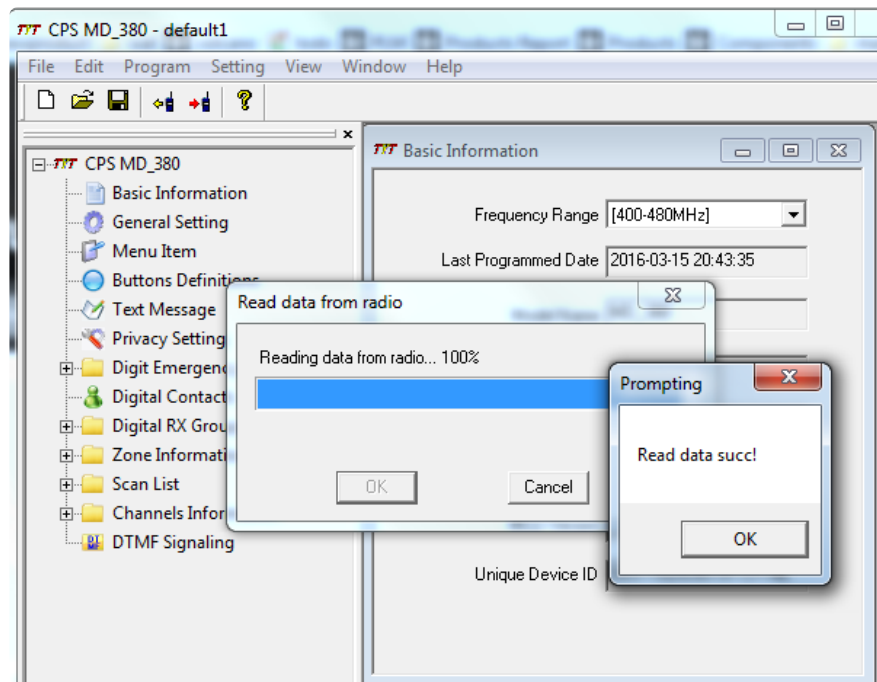
You'll get a 'default' setup like so:



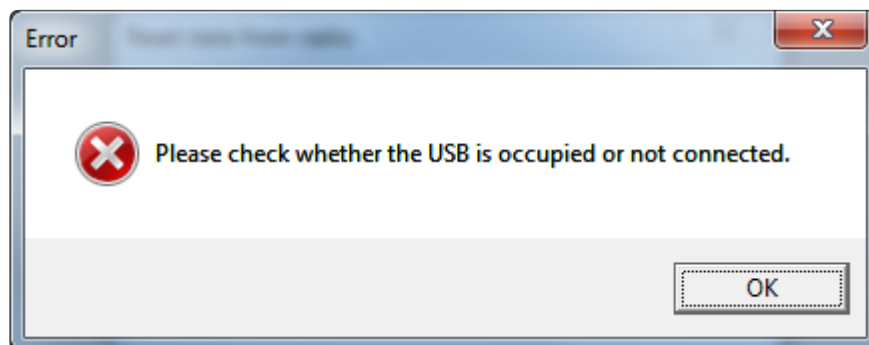
Read in Radio Setup

Turn off the radio, plug your radio into the programming cable, then turn on. Make sure it shows up as a USB device. Now select Read Data





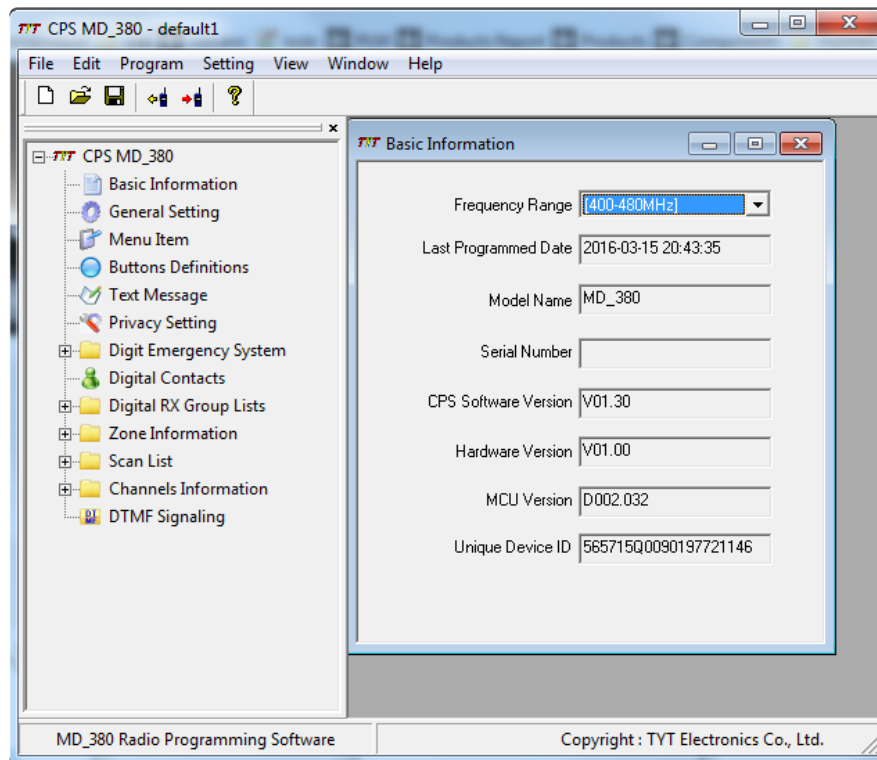
If you get something like:



Check that you have the STM DFU driver, not the libusbK driver installed

Fresh codeplug!

Now you have a 'blank slate' to work with, which is the default 'codeplug'



If you by chance need to 'start over' with a blank original setup, this is the default codeplug file:

fresh380rdt.zip

<https://adafruit.it/D1z>

Program in DMR ID

Remember that ID number you got earlier? Now's the time to use it. Type it into the General Settings window under Radio ID

Program in Contacts

As we determined, K2MAK is a good place to start so let's program in this repeater that covers all of Manhattan:

description
K2MAK Manhattan, NY

name
448.2750 - 5 MHz Color Code 3
(Bridge Partner of DMR-MARC)

Time Slot #1 - Group Call 1 = World Wide (PTT)
 Time Slot #1 - Group Call 13 = WW English (PTT)
 Time Slot #1 - Group Call 3 = N. America (PTT)
 Time Slot #1 - Group Call 3172 = Northeast Regional (PTT)
 Time Slot #1 - Group Call 310 = TAC310 (PTT)
 Time Slot #1 - Group Call 311 = TAC311 (PTT)
 Time Slot #2 - Group Call 2 = NJ-NY Metro (ON)

You Must Have [ARS] Disabled Within Your Radio

For corrections or updates please contact
WB2HHH@ARRL.NET

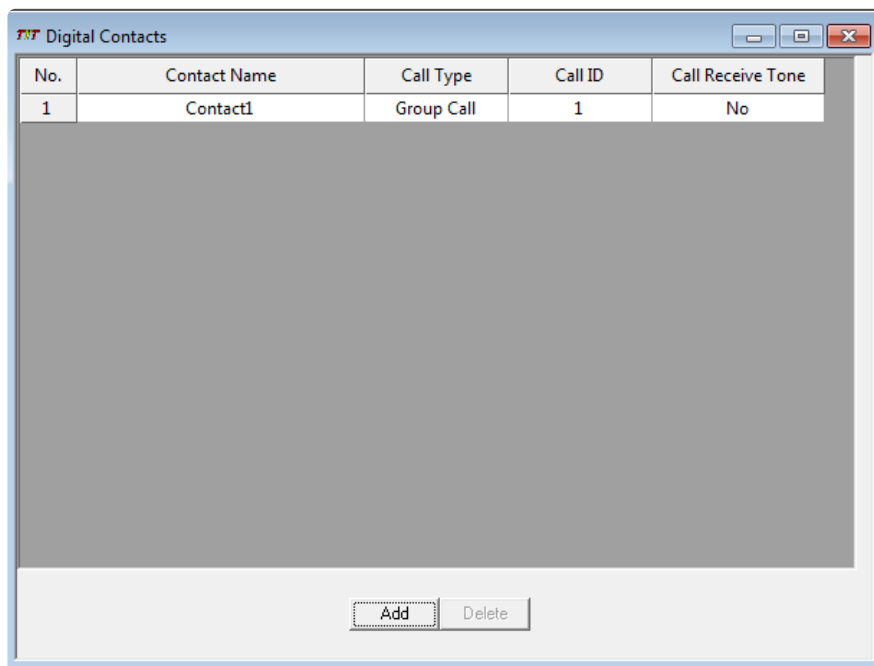
This repeater carries

- Group 1 (World wide)

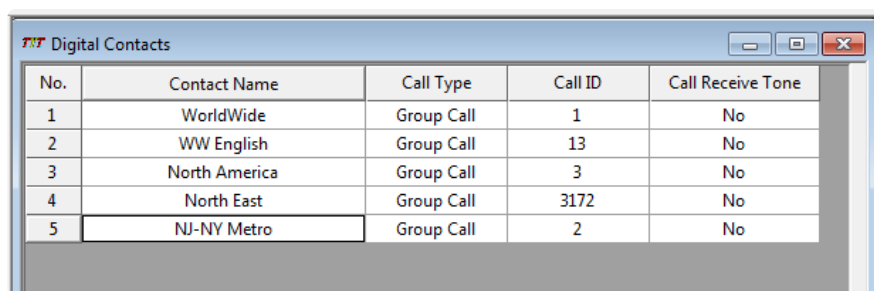
- Group 13 (World wide English)
- Group 3 (US + English / North America)
- Group 3172 (North East)
- Group 2 (NY-NJ Metro)
- Two TAC groups (wont be covered here)

We'll enter all of these in as optional ways for us to communicate with others.

Go to the Digital Contacts window, to start it will have only "Contact1" which is group call ID #1



Click Add at the bottom to add a bunch of empty slots, then fill each slot with the Talk group and TG ID #



Entering Channels

The blank/new codeplug has 17 Channel slots. Note that these aren't necessarily 'hardcoded' to the channel selector, you can change zones to change which channels

are used. In fact you can have dozens of Channels, split between zones. For now, though, we'll only keep the first Zone and set up one channel per talk group.

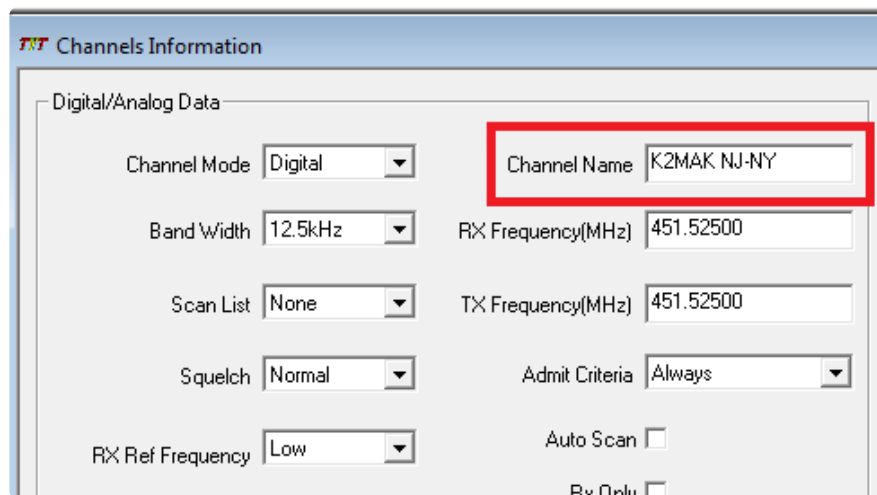
In general:

- Each channel is for a specific repeater and talk group
- You can have a different repeater/TG per channel
- You can change out all of the channels when you switch zones. So, if you are often in both NY and Texas, you can have a fully different set of repeaters and channels you switch between!

However, we'll start with just once channel for talk group 2, on the K2MAK repeater.

Set Name and Frequency

Let start by defining this to be the K2MAK NJNY Region channel:



The screenshot shows a 'Channels Information' window with the following settings:

Field	Value
Channel Mode	Digital
Channel Name	K2MAK NJ-NY
Band Width	12.5kHz
Scan List	None
Squelch	Normal
RX Frequency(MHz)	451.52500
TX Frequency(MHz)	451.52500
Admit Criteria	Always
RX Ref Frequency	Low
Auto Scan	<input type="checkbox"/>
Rx Only	<input type="checkbox"/>

The midtown Manhattan, NY repeater that is part of the Bronx-TRBO network. The Repeater Transmitting frequency is 448.2750 MHz and the Repeater Receiving frequency is $448.2750 - 5 = 443.2750$ MHz

That means we should set our channel RX to 448.2750 and the channel TX to 443.2750

Digital/Analog Data

Channel Mode: Digital

Channel Name: K2MAK NJ-NY

Band Width: 12.5kHz

Scan List: None

Squelch: Normal

Admit Criteria: Always

Auto Scan: ☐

RX Frequency(MHz): 448.27500

TX Frequency(MHz): 443.27500

You're not done! You next have to program in the Color Code for the repeater as well as the time slot. You'll get this info from the repeater as well:

description

K2MAK Manhattan, NY

name

448.2750 - 5 MHz Color Code 3
(Bridge Partner of DMR-MARC)

Time Slot #1 - Group Call 1 = World Wide (PTT)

Time Slot #1 - Group Call 13 = WW English (PTT)

Time Slot #1 - Group Call 3 = N. America (PTT)

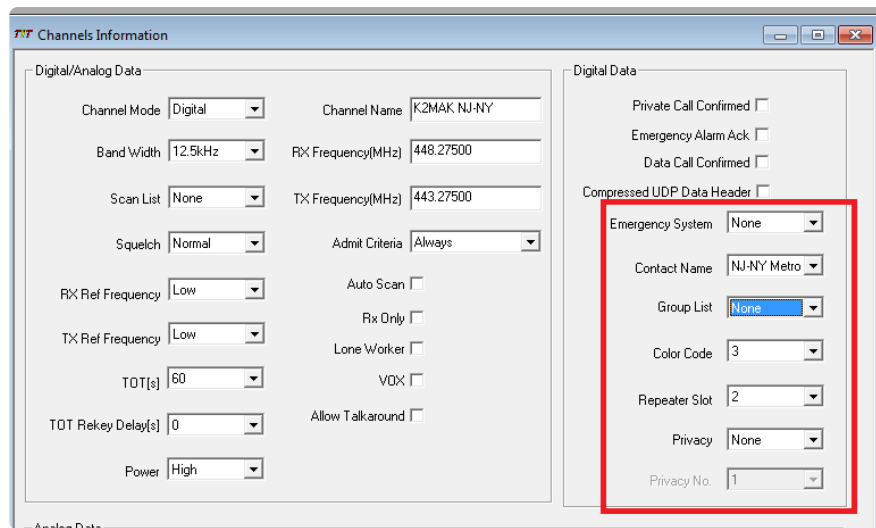
Time Slot #1 - Group Call 3172 = Northeast Regional (PTT)

Time Slot #1 - Group Call 310 = TAC310 (PTT)

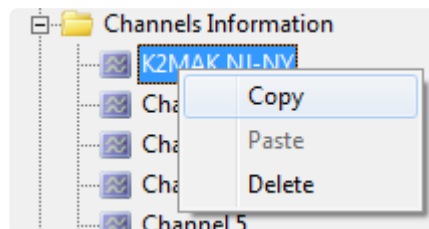
Time Slot #1 - Group Call 311 = TAC311 (PTT)

Time Slot #2 - Group Call 2 = NJ-NY Metro (ON)

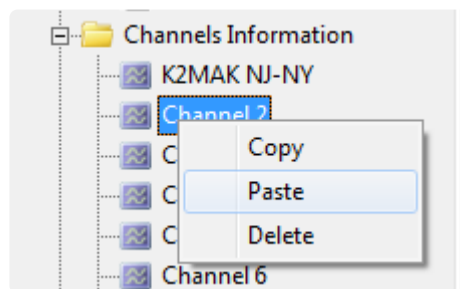
This goes on the right hand side, in particular make sure you have the right Color Code and Repeater Slot to match the Contact Name. The remaining you can set to None



Once you have the frequencies for your repeater set up you can speed things up by copying one channel:



and pasting it onto another. Note it will delete what was in the channel you're pasting into



This time, I'll program in the NE Regional, which is on Timeslot #1 so don't forget to change it!

Channel Mode

Channel Name

Band Width

RX Frequency(MHz)

Scan List

TX Frequency(MHz)

Squelch

Admit Criteria

RX Ref Frequency

Auto Scan

TX Ref Frequency

Rx Only

TOT[s]

VOX

TOT Rekey Delay[s]

Allow Talkaround

Power

Channel Mode

Channel Name

Band Width

RX Frequency(MHz)

Scan List

TX Frequency(MHz)

Squelch

Admit Criteria

RX Ref Frequency

Auto Scan

TX Ref Frequency

Rx Only

TOT[s]

VOX

TOT Rekey Delay[s]

Allow Talkaround

Power

Private Call Confirmed

Emergency Alarm Ack

Data Call Confirmed

Compressed UDP Data Header

Emergency System

Contact Name

Group List

Color Code

Repeater Slot

Privacy

Privacy No.

Private Call Confirmed

Emergency Alarm Ack

Data Call Confirmed

Compressed UDP Data Header

Emergency System

Contact Name

Group List

Color Code

Repeater Slot

Privacy

Privacy No.

I'll do the rest, ending with Worldwide. I set this to RX only so I don't accidentally TX on it!

Channel Mode

Channel Name

Band Width

RX Frequency(MHz)

Scan List

TX Frequency(MHz)

Squelch

Admit Criteria

RX Ref Frequency

Auto Scan

TX Ref Frequency

Rx Only

TOT[s]

VOX

TOT Rekey Delay[s]

Allow Talkaround

Power

Channel Mode

Channel Name

Band Width

RX Frequency(MHz)

Scan List

TX Frequency(MHz)

Squelch

Admit Criteria

RX Ref Frequency

Auto Scan

TX Ref Frequency

Rx Only

TOT[s]

VOX

TOT Rekey Delay[s]

Allow Talkaround

Power

Private Call Confirmed

Emergency Alarm Ack

Data Call Confirmed

Compressed UDP Data Header

Emergency System

Contact Name

Group List

Color Code

Repeater Slot

Privacy

Privacy No.

Private Call Confirmed

Emergency Alarm Ack

Data Call Confirmed

Compressed UDP Data Header

Emergency System

Contact Name

Group List

Color Code

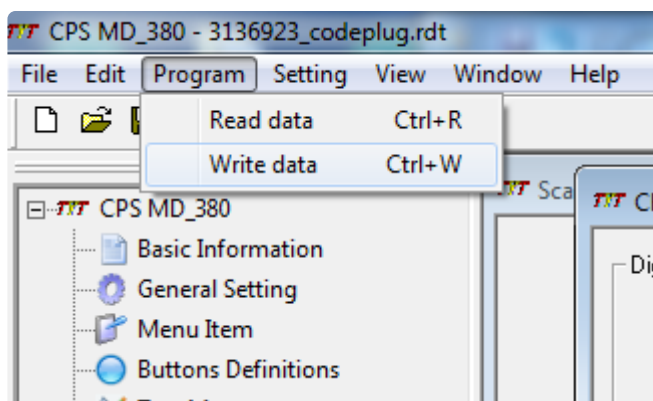
Repeater Slot

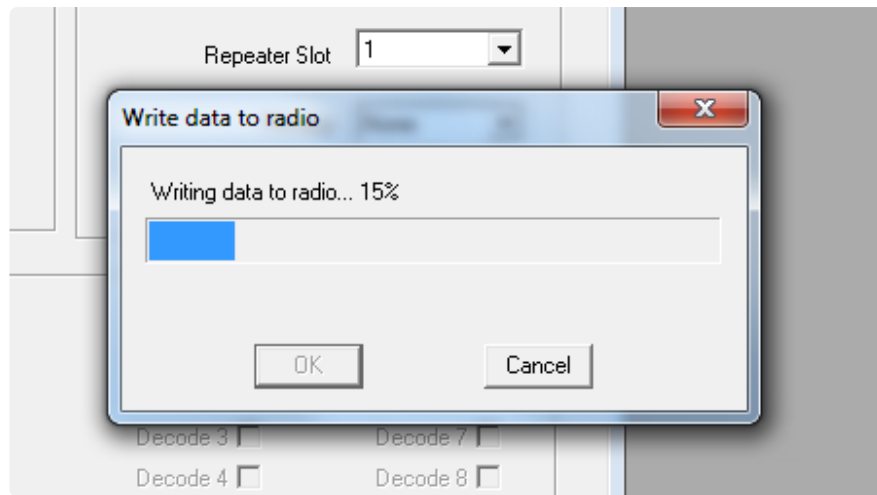
Privacy

Privacy No.

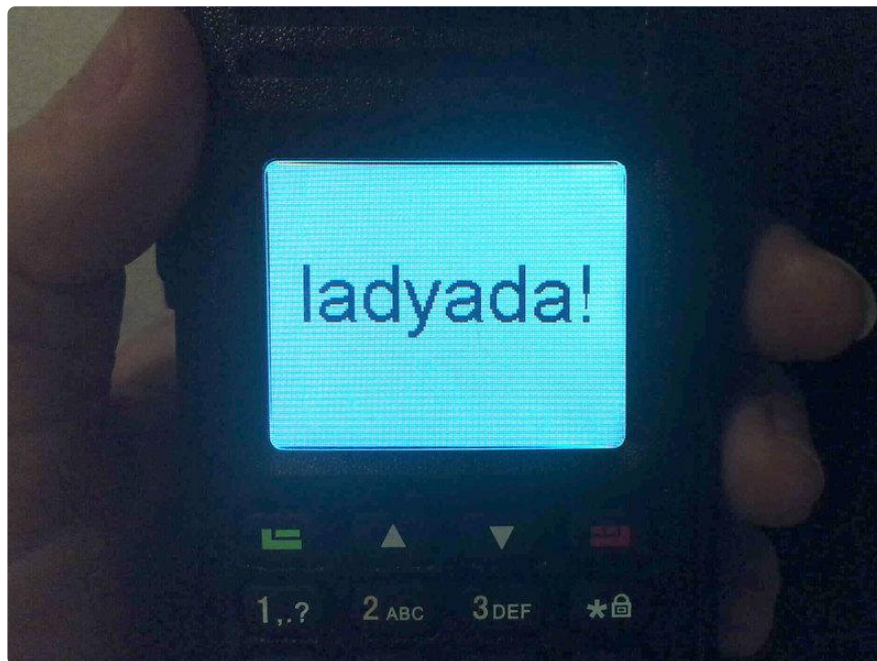
Write Data

That's it! You're ready to upload the data to your radio. Don't forget you can always go back and mess with your codeplug and then re-upload it if you make a mistake or want to have more repeaters or contacts





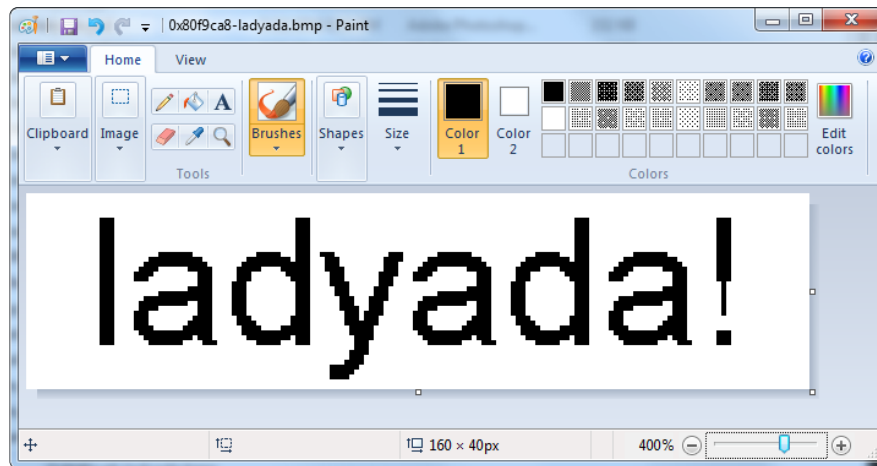
Custom Bootscreen



You can even make a custom bootscreen. Handy for looking 'leet and also for helping you identify your radio.

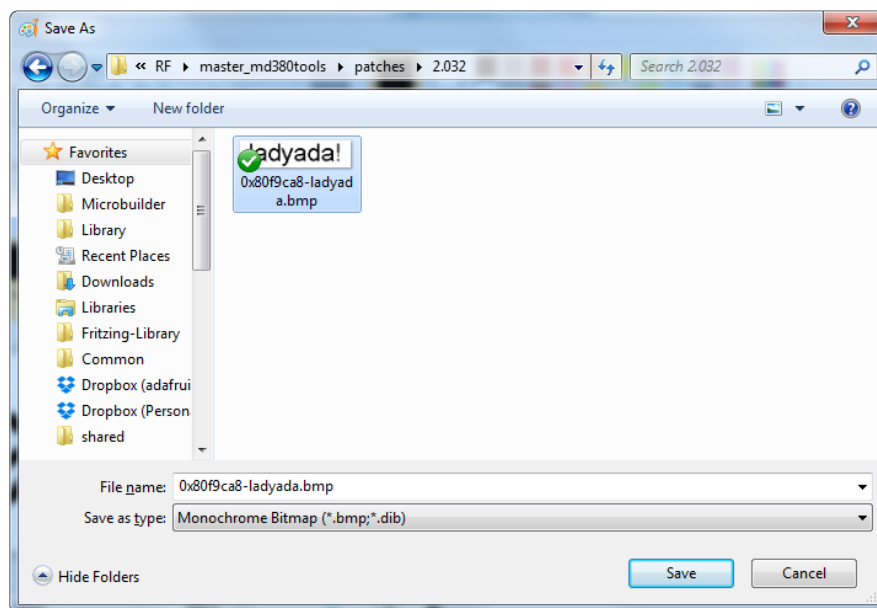
Make a Bitmap

Start by making a 160 x 40 bitmap image of what you want displayed on the screen. I'm going classy monochrome but you can have multiple colors as well.



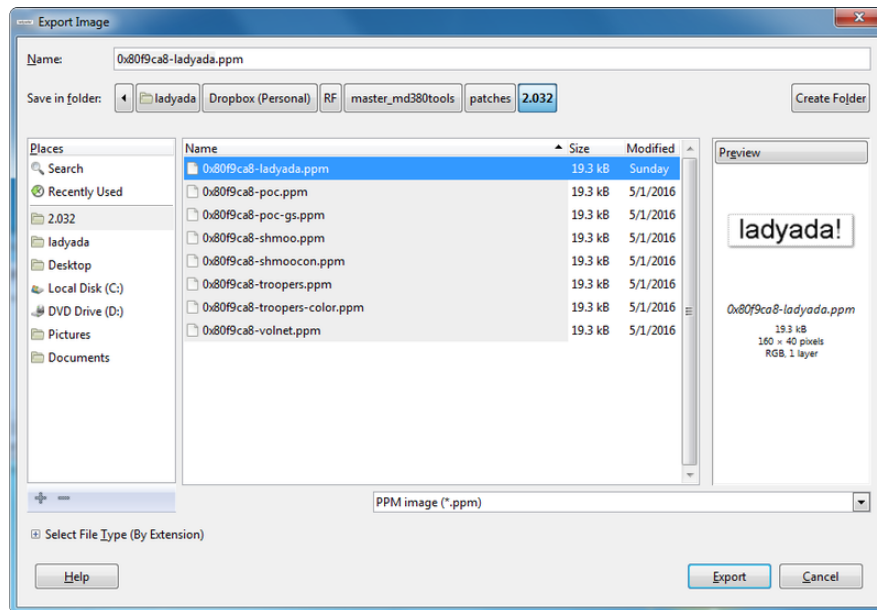
Save Bitmap

Save the bitmap to the md380tools\patches\2.032 directory and name it 0x80f9ca8-bitmapname.bmp where bitmapname is whatever you like.



Open bitmap in GIMP

Yeah we have to generate a ppm file and GIMP can do it, but GIMP is not happy about making bitmaps so its actually easier I think, to make the bitmap in mspaint and then open in GIMP. Then export as... and save as 0x80f9ca8-bitmapname.ppm

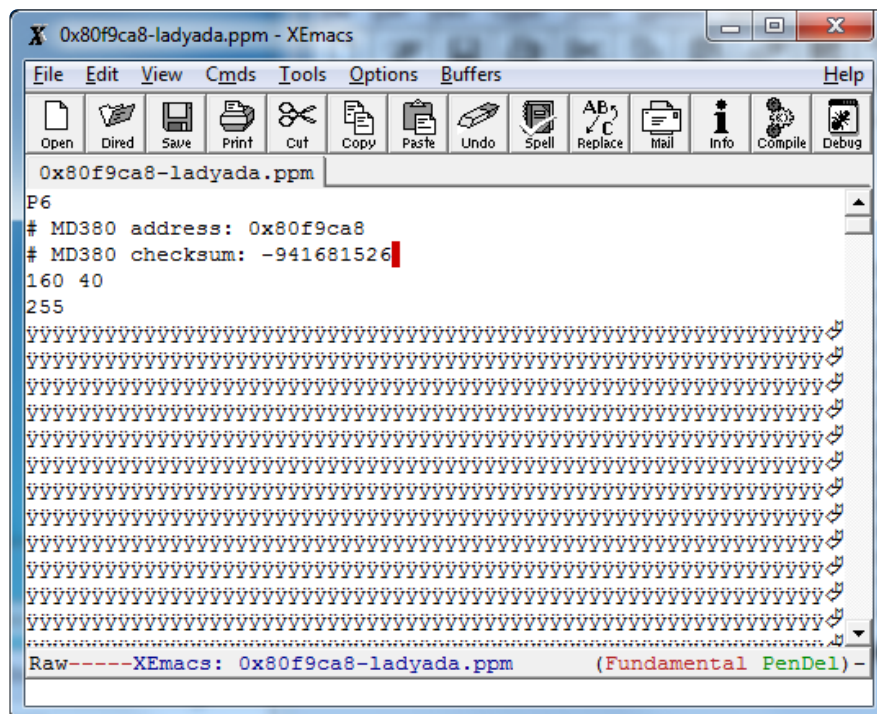


Add checksum

You're not done quite yet! Edit the ppm file with your favorite text editor and add

```
# MD380 address: 0x80f9ca8
# MD380 checksum: -941681526
```

after the first line (which will say P6). Basically it should look like this:

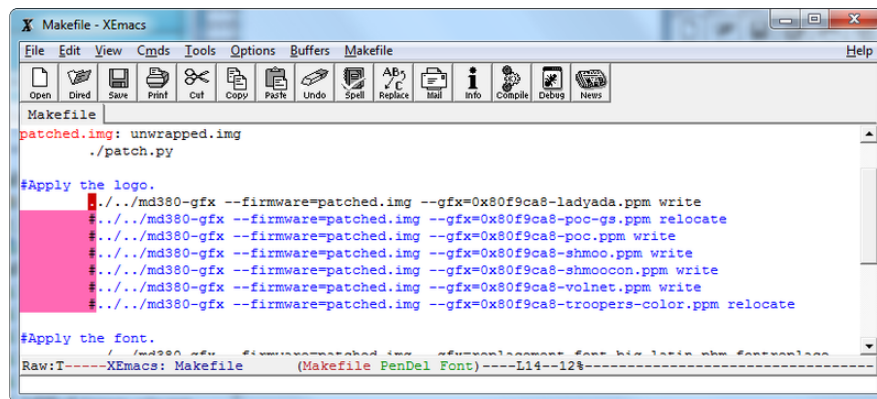


Edit Makefile

Edit the Makefile in md380tools\patches\2.032 and find the Apply the logo section, add a new line

```
../../md380-gfx --firmware=patched.img --gfx=0x80f9ca8-bitmapname.ppm  
write
```

and comment out the other lines



Whew that's it! Run make clean dist to upload your nice new firmware! You'll see your bitmap is detected and 'printed out' so you know it was chosen

