

# BITX40 / $\mu$ BITX Transceiver Kits

Andy Stewart  
KB1OIQ  
March 15, 2018

Presented to the  
Nashoba Valley Amateur Radio Club  
Pepperell, MA

# BITX40 / $\mu$ BITX – What are they?

- QRP transceiver kits from India
  - by Ashhar Farhan (VU2ESE)
- Kit includes:
  - Main circuit board already assembled (including toroids)
  - Raduino (custom board)
    - Arduino Nano
    - Si5351 CMOS clock generator (up to 160 MHz)
  - 16x2 LCD display
  - Wiring harnesses, connectors, potentiometers, hardware, etc.
- User supplies the enclosure
- Shipped in 1 week from India via DHL (\$10 extra)

# BITX40 / $\mu$ BITX – How do they compare?

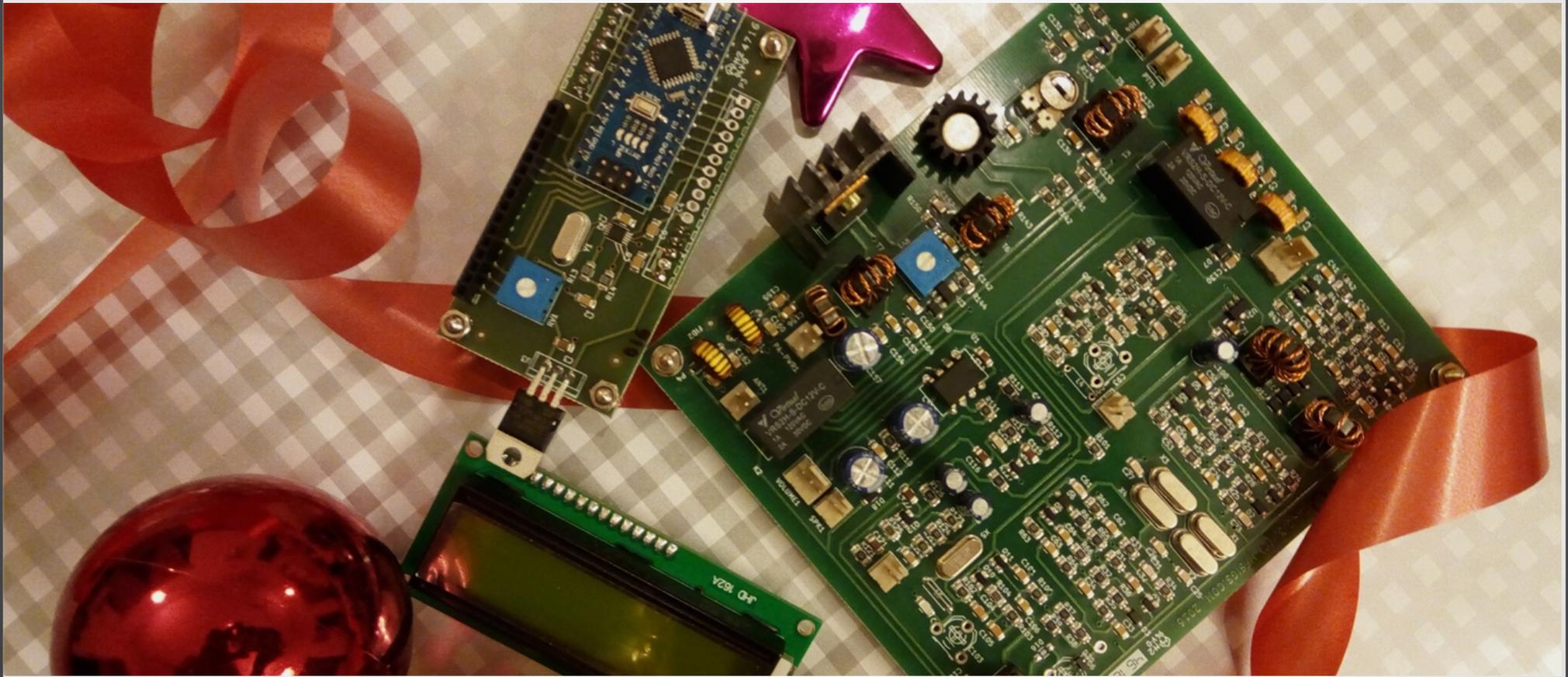
- BITX40

- Single band 40m
- 6.5-7.0 watts @13.8V
- One IRF510
- 16x2 LCD
- Si5351
- Potentiometer tuning
- PTT directly to board
- 12 MHz IF,  
Si5351 @5MHz
- Separate PA power rail
- Primitive MIC

- $\mu$ BITX

- Multi-band 80m-10m
- Up to 10 watts @13.8V
- Two IRF510s (push pull)
- 16x2 LCD
- Si5351
- Rotary encoder tuning
- PTT goes to Arduino
- 1st IF upconvert to 45MHz  
2<sup>nd</sup> IF downconvert to 12MHz
- Separate PA power rail
- Primitive MIC

# BITX40 Circuit Boards



# BITX40 / $\mu$ BITX Features

- Inexpensive kits:
  - BITX40: \$69 includes DHL shipping from India
  - $\mu$ BITX: \$119 (also includes shipping)
- Schematics are freely available
- Arduino program (sketch) is Free Software GPL v3
- Very active(!) community on the internet
- Many software and hardware mods are available
  - Different sketches available for stock rig
  - Custom sketches to match custom mods also available

# Arduino Nano

- Open Source Hardware (Creative Commons)
- Microcontroller: ATmega328
- 5Vdc, 19mA, 16MHz
- Programmable using Arduino IDE and available libraries
- Arduino IDE runs on multiple operating systems
- 32KB FLASH memory
  - (2KB bootloader, 30KB available to user for program)
- 1KB EEPROM, 2KB SRAM
- Leftover pins for the user:
  - BITX40: 6 digital, 5 analog
  - $\mu$ BITX: 0 digital, 1 analog

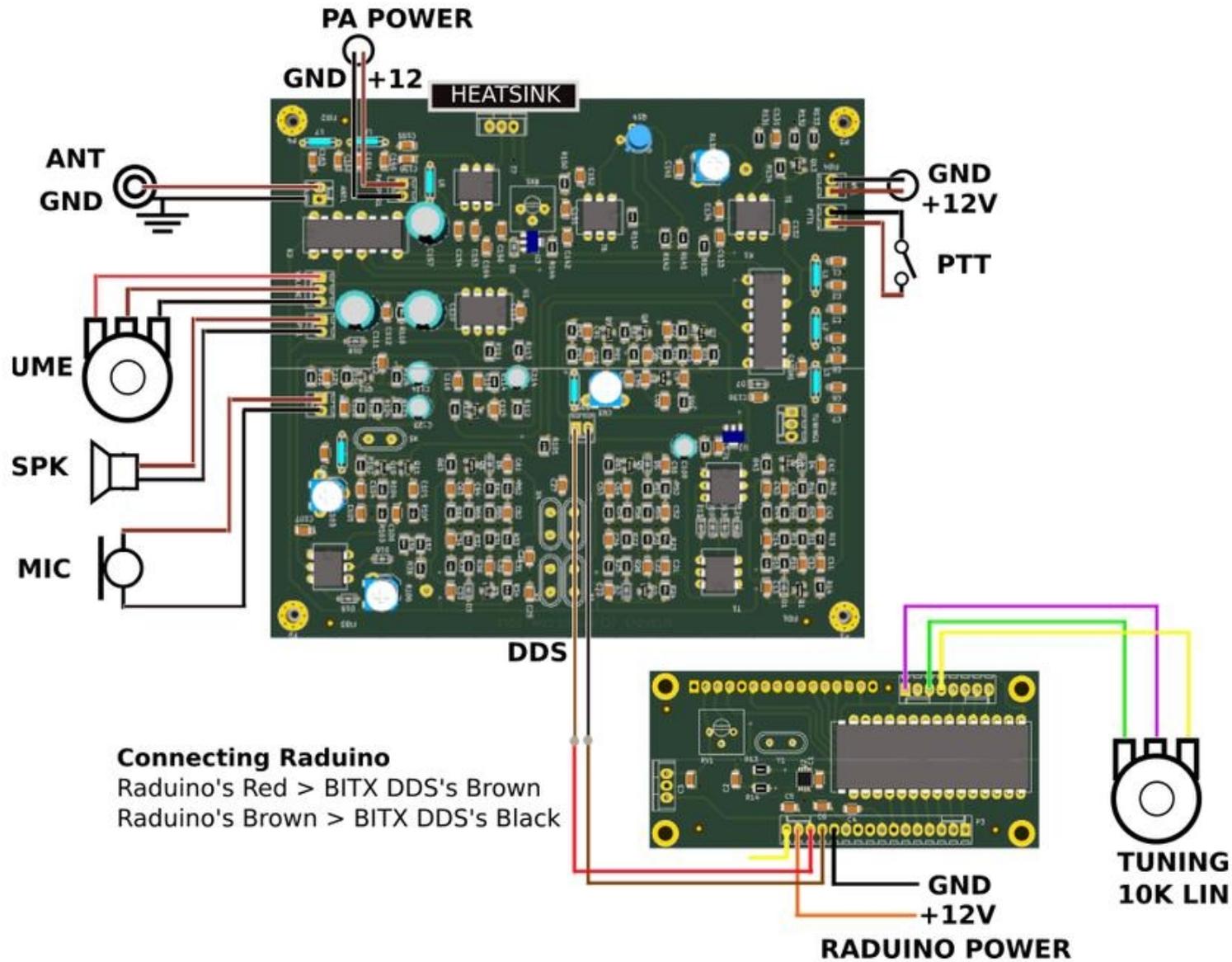
# Si5351

- CMOS clock generator
  - 3 clock outputs (only 1 used in BITX40)
- I2C device
  - 2 pin serial protocol, serial clock, serial data
- Generates frequencies up to 160MHz (0 ppm error)
- Glitchless frequency changes
- Ideally suited for replacing crystals, crystal oscillators, VCXOs (voltage controlled crystal oscillators)

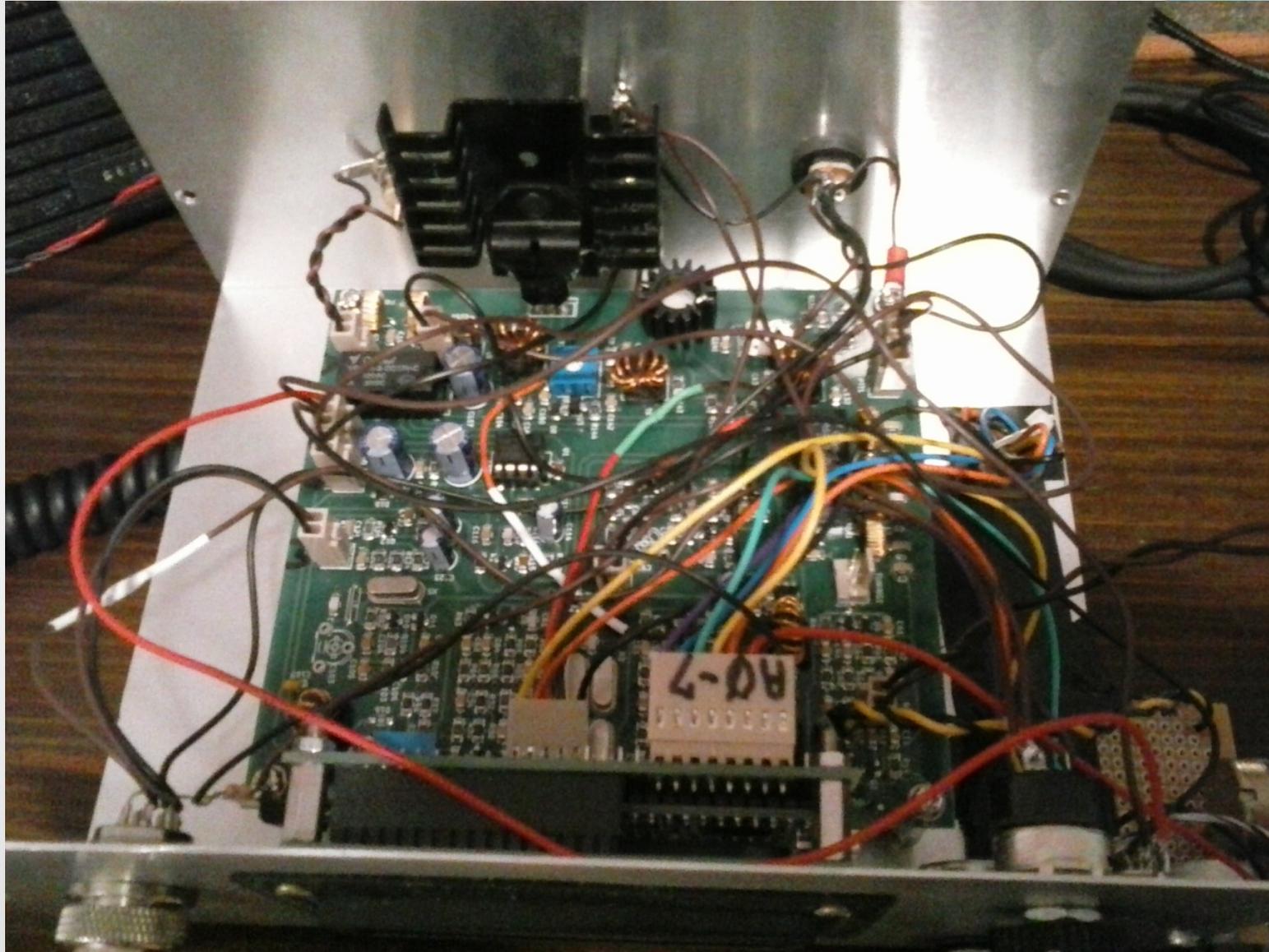
# Building the BITX40 / $\mu$ BITX

- Obtain a suitable enclosure
- Drill holes and cut opening for LCD display
- Other enclosure machining as needed
- Mount all of the parts in the enclosure
- Software is already loaded into the Arduino

# BITX40 Wiring Diagram



# BITX40 internals by KB1OIQ



# BITX40 Enclosure by KB1OIQ



- External speaker from NEAR-Fest
- ICOM mic
- Awesome knobs from elmer's junk box
- Metal enclosure
- Brushed metal look from green kitchen scrubbing pad
- Homebrew tie wrap bezel

# Prepare for first power-on

- Connect 12Vdc (13.8Vdc is OK)
- Use a 2 amp fuse!
- Connect the antenna cable (or dummy load) to the BNC connector.
- Plug in an external speaker
  - **Don't use headphones!**
  - It is much too loud and variable! No AGC(!)
- Connect microphone
- Throw the switch! Hear the speaker pop...good sign!
- Sniff for smoke.....
- Watch the LCD display, with the original sketch:
  - It should say “Raduino”, then display the frequency

# First QSO

- Called my elmer: Steve (W1KBE)
- Transmitting 6.5 watts 40m LSB phone
- First record QSO: 0.215 miles / watt !
- It works!!!
- Fast forward many days/months:
  - Best DX so far: Croatia, about 4000 miles

# BITX40 Hacks Done by Others

- Implement USB for digital modes
- Bend the circuitry and software to allow for CW use
- Several options for the following:
  - AGC circuit
  - RF gain circuit
- PA rail is intentionally separate from 12V rail
  - Bigger IRF510 heat sink and 24V at PA
  - 17 – 20 watts output on LSB
- Replace tuning potentiometer with a rotary encoder

# KB1OIQ BITX40 Modifications

- Tuning algorithm modifications
- ICOM MIC connector
- RF Gain circuit
- Keypad
- Dual VFOs
- Save rig parameters in EEPROM
- Fine tuning potentiometer
- Added USB (for digital modes)
- Voice synthesis via Talkie library

# BITX40: Tuning Behavior

- Tuning via 10K ohm potentiometer between +5V and GND
- Wiper is connected to Arduino analog input
- Arduino ADC converts voltage to integer  $0 \leq N \leq 1023$
- To reduce tuning sensitivity, full range of the tuning knob is 50kHz. Minimum tuning is 100Hz (controlled in SW).
- HOWTO get to another 50kHz section of the band?
- Tuning knob extremes intentionally cause frequency to jump in large increments.
  - There is no warning that this will happen.
  - Many found this behavior undesirable.

# BITX40: Tuning Modification

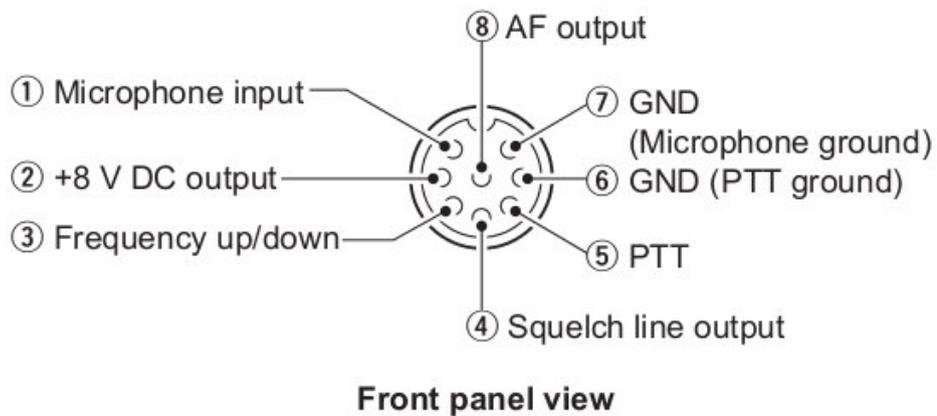
- Sketch changes:
  - Change sketch to tune only a 20kHz region
  - Remove “jump” feature at potentiometer edges
- Implement MIC modification (described next)
  - Use UP and DN buttons to jump to other 20 kHz regions
- Tuning granularity is still 100 Hz
  - Fixed later with Fine Tuning mod

# MIC modification

- Wire MIC to BITX40 main board
- Connect UP/DN to an unused Arduino analog pin
  - UP: GND, DN: thru 470 ohms, 4.7K pullup to +5V
- BITX40:
  - Potentiometer range: changed to 20kHz
  - To get to the next 20kHz region, use UP or DN button
  - Remedies the “galloping VFO” problem
- $\mu$ BITX: haven't done this mod yet, but shortly...
  - Not sure what to do with UP/DN buttons
  - Uses rotary encoder for tuning

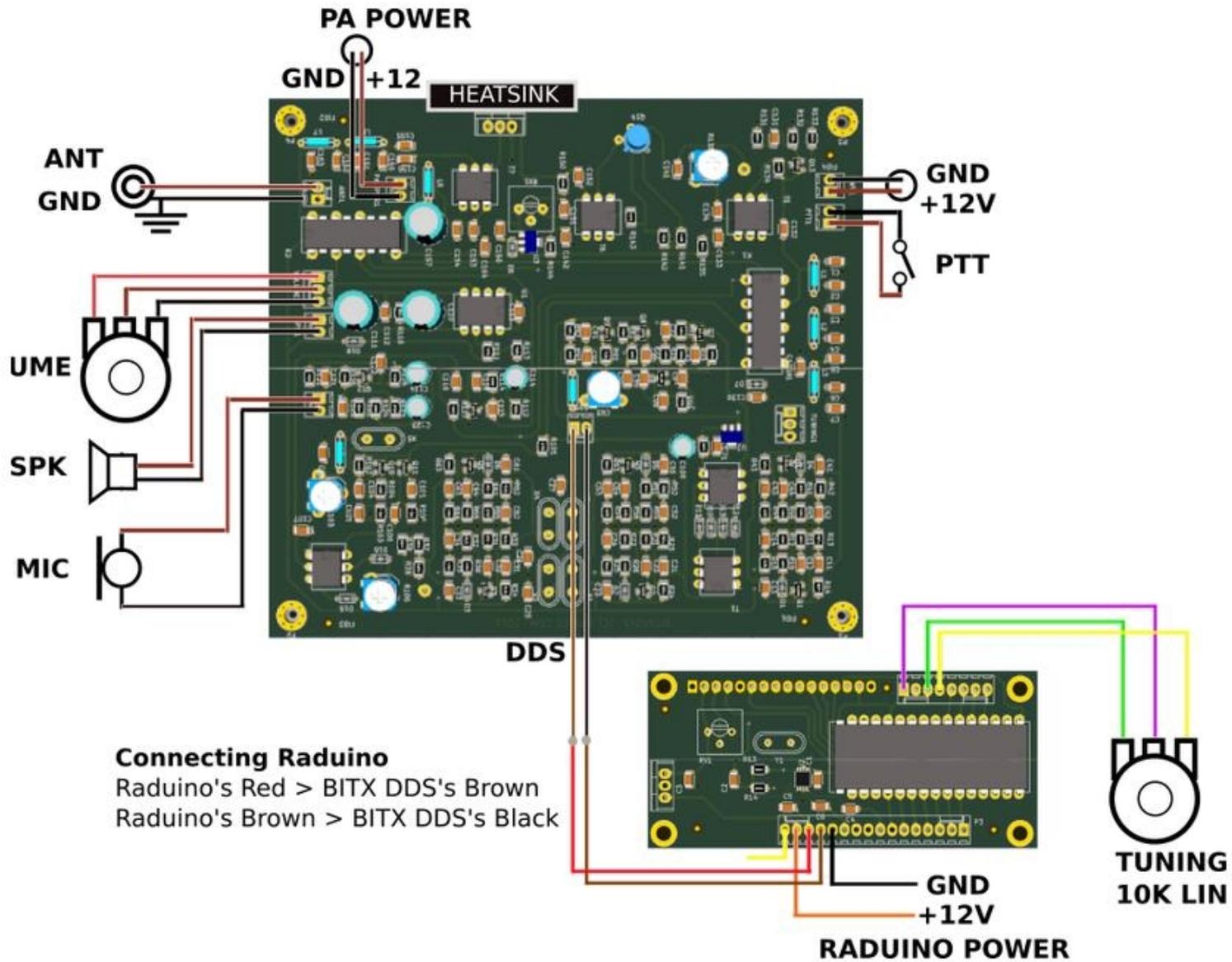
# MIC connector pinout

## Microphone connector



PIN No.	DESCRIPTION
①	Microphone input (Impedance: 600 $\Omega$ )
②	+8 V DC output (Maximum 10 mA)
③	Up: Ground Down: Ground through 470 $\Omega$
④	Grounded when the squelch opens.
⑤	PTT
⑥	PTT ground
⑦	Microphone ground
⑧	AF output (varies with the [AF] control.)

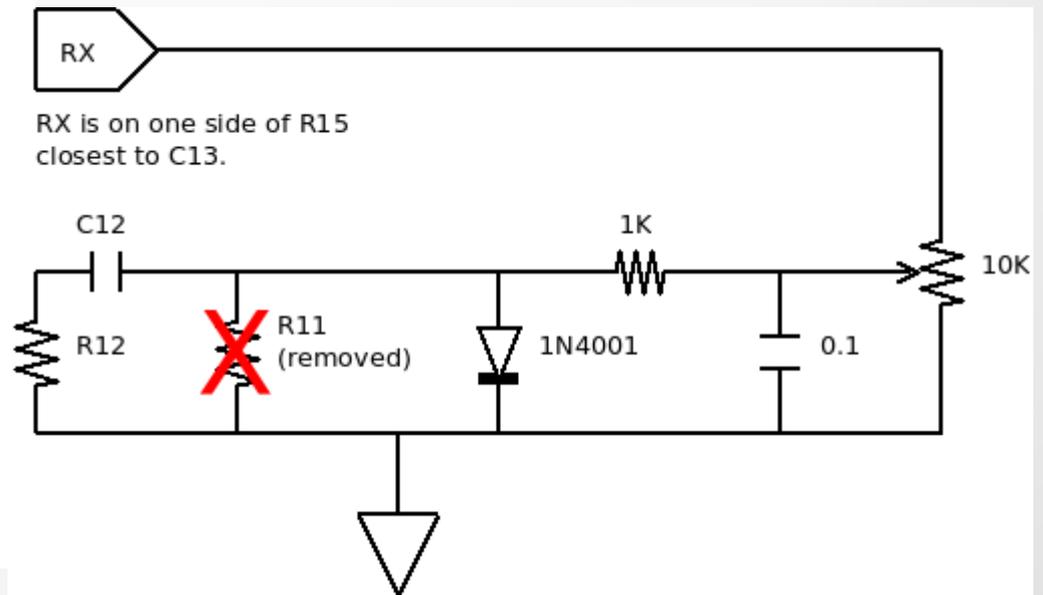
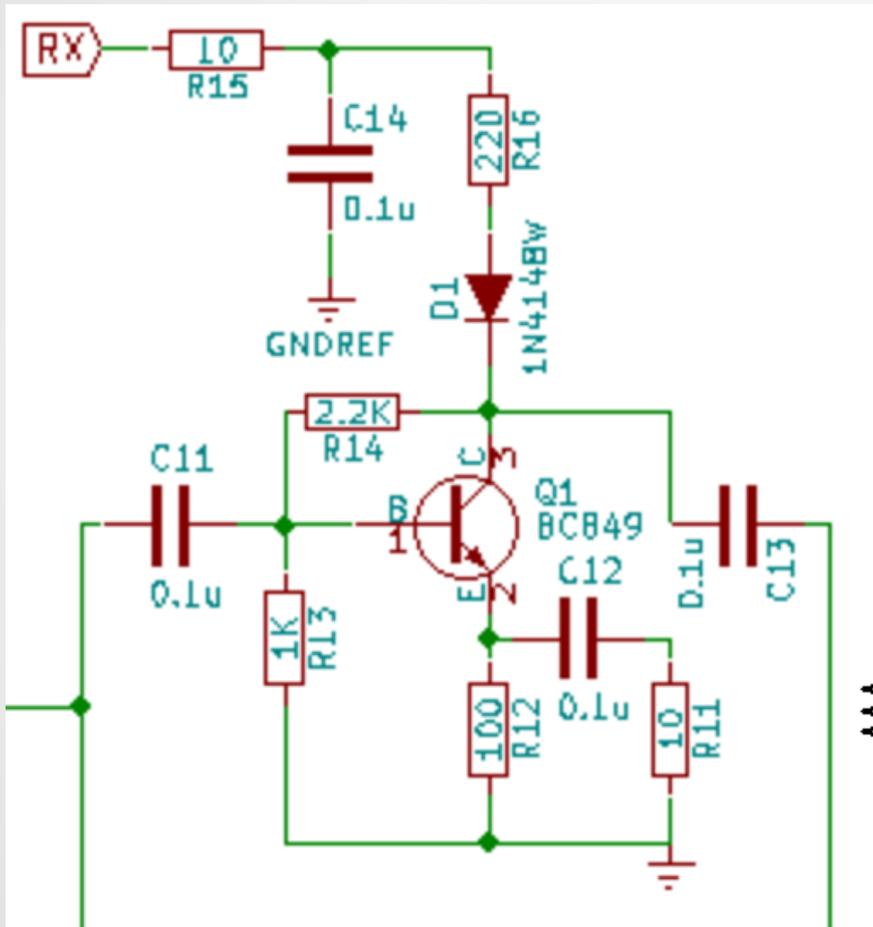
# Wiring Diagram



## BITX40: RF Gain mod

- This receiver has **A LOT** of gain
  - S9+ signal overloads the receiver and distorts the audio
- Elmer recommended a circuit straight from EMRFD
  - Experimental Methods in RF Design, by ARRL
  - Mod based on: Revised 1<sup>st</sup> Edition  
Page 6.16, Figure 6.41
- I chose to remove R11 and add circuit to R11 pads
- No sketch mod is needed, this is just an analog circuit

# Mod: RF Gain



## BITX40 / $\mu$ BITX: Add a Keypad

- Resistive keypad
- Single output, different voltage for each key
- Connect it to an Arduino analog input
- Modify sketch to decode keypad
- Requires: +5V, GND, and keypad output
  - Used a stereo headphone jack
    - Warning: don't insert/remove plug when powered!
    - I'm glad I use fuses!
  - Looking at different connector type for  $\mu$ BITX
- Connect “signal” to unused Arduino analog pin

# Decoding the Keypad

- `byte key = decode_keypad(analogRead(KEYPAD));`
- `int decode_keypad(int tmp) {`
- `int key = 999; // error value, 0 means no key pressed, or keypad not present`
- 
- `if (((tmp >= 0) && (tmp <= 20)) || ((tmp > 520) && (tmp <= 1023))) { key = 0; } else {`
- `if ((tmp > 465) && (tmp < 495)) { key = 1; } else`
- `if ((tmp > 420) && (tmp < 450)) { key = 2; } else`
- `if ((tmp > 380) && (tmp < 410)) { key = 3; } else`
- `<snip>`
- `if ((tmp > 65) && (tmp < 95)) { key = 11;} else`
- `if ((tmp > 25) && (tmp < 55)) { key = 12;} else key = 999;`
- `}`
- `return key;`
- `}`

## SW Mod: VFOs

- VFOs
  - Just another memory location holding a frequency
  - SW mod plus a keypad button to activate the function
- Two VFOs: vfoA and vfoB
- Set active VFO – toggle between vfoA and vfoB
- Swap VFOs – trade contents between vfoA and vfoB
- Make VFOs equal (both equal to active VFO)
- Note to self: carry mode with VFO, not just frequency

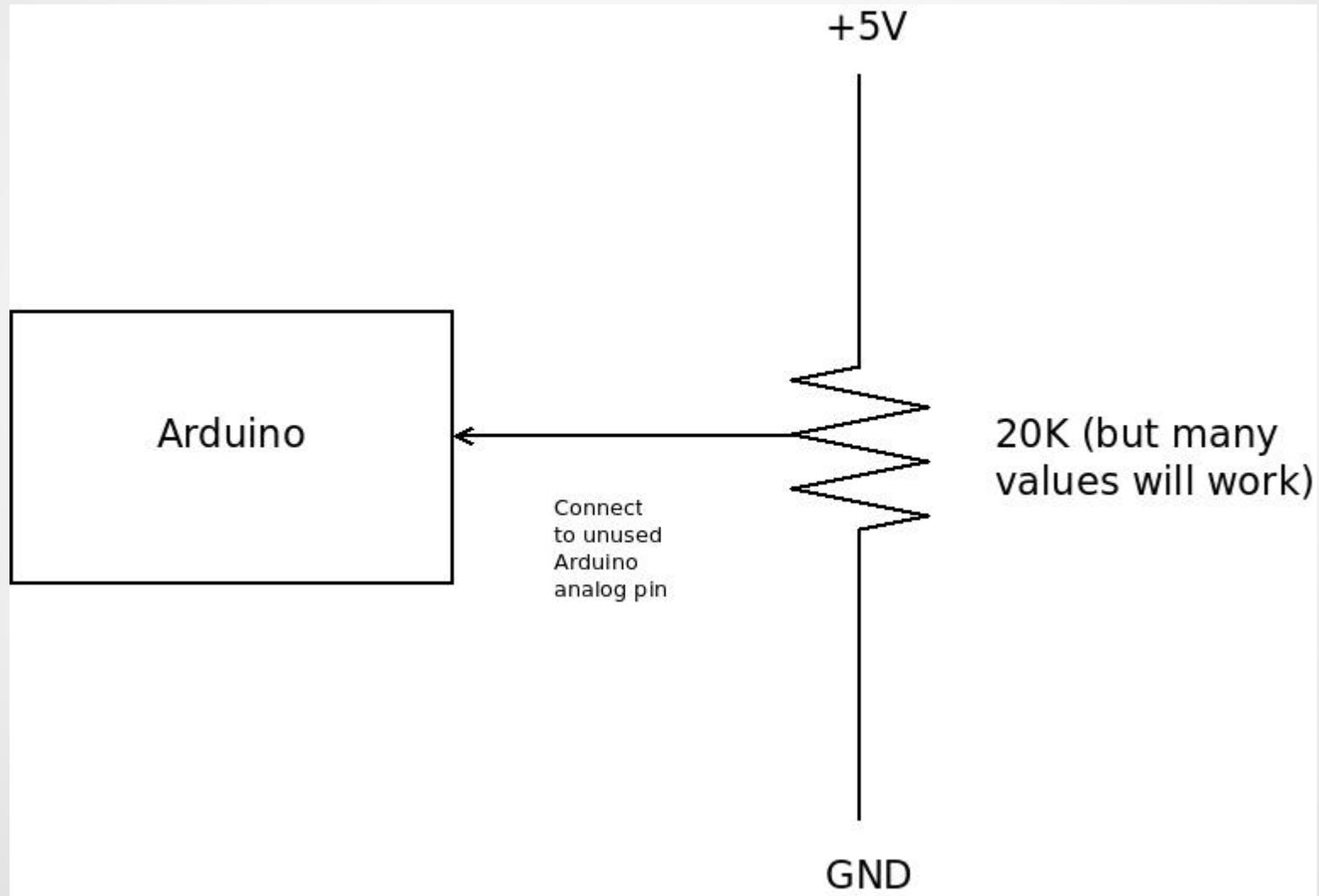
## SW Mod: Save/restore rig parameters

- EEPROM is accessible inside of the Arduino
- Use keypad button to save rig parameters
- On power up, rig parameters are restored
- What is saved:
  - Active VFO (A or B)
  - vfoA and vfoB frequency range (add mode, too)
  - Calibration values
- May add more to this in the future.

# Fine Tuning Mod

- Normal tuning granularity is 100 Hz
  - Fine tuning granularity is 1 Hz
  - Range -255 Hz → +256 Hz
  - Improves intelligibility of SSB signal
- Very simple circuit (see next slide)
- Requires sketch modification to read value and adjust frequency

# Mod: Fine Tuning Circuit



## BITX40: USB Mode

- Desire to do digital modes with BITX40
- SW:
  - Keep track of USB/LSB mode
  - Modify `set_frequency()`
  - BFO frequency constant around 12 MHz
  - VFO frequency set to 5 MHz or 19 MHz
- HW:
  - Remove C91 and C92 (or USB power is only 1W)
  - Build a custom cable between RigBlaster Plug 'n Play and mic jack

# Voice Synthesis

- Might this be useful to visually impaired hams?
  - 3 people have expressed interest
- BITX40
  - Used software Talkie library
  - Very limited vocabulary, proof of concept
  - Triggered by keypad
- $\mu$ BITX
  - using Emic2 text-to-speech hardware module
  - MUCH better voice quality and vocabulary
- Audio mixer board (or two speakers)
  - To mix voice sounds with “radio sounds” in one speaker

# BITX40: Modifications Considered

- AGC circuit
  - Audio: haven't found a good circuit
  - RF: ?
- Split mode
  - Need to send PTT to Arduino, then drive PTT on board
  - Nope, but PTT in uBITX is different...
- Simple rig control via hamlib
  - Nope, not enough code space

# Questions about BITX40?

- Answer questions
- Moving on to  $\mu$ BITX

# KB10IQ's $\mu$ BITX Prototype



# KB1OIQ's $\mu$ BITX Prototype

- 12"x12" copper clad circuit board
- Rear panel jacks
  - Antenna, 12V, speaker, key/paddle
- Emic2
- On/off/volume, rotary encoder
- Keypad
- White box: amplified speaker (temporary)
- Someday, this will be in a proper enclosure
  - Need to see what has to go in there...

## μBITX mods by KB1OIQ (so far)

- Major surgery on the sketch
  - Refactored everything into C++ classes
    - Saved most of Farhan's original sketch
  - Memory is a scarce resource! Use it wisely!
    - put all text strings in PROGMEM
    - byte (1 byte) vs. int (2 bytes) vs. long (4 bytes)
    - don't copy/paste code, use classes!
  - Reimplemented the menu system
- Converted LCD to I2C (recovered 6 digital pins!)
- Added Emic2, separate audio output (used 2 digital pins)
- Added keypad (used remaining analog pin)

# μBITX: What works today for KB1OIQ?

- Everything printed to LCD is also spoken by Emic2
- Menus rewritten with C++ class library (MenuSystem)
  - CW Menu: key, paddle, sideTone, keyer speed, etc.
  - Tuning Menu: lsb/usb/cw, RIT(nyi), select VFO A/B
  - Voice Menu: choose voice, talking speed, volume
- Keypad:
  - help, menu mode, direct frequency input
  - modes, VFOs (saved in EEPROM), speak on demand
- CW: key and paddle (left or right handed)
- Won't transmit out of band, CW key disabled in LSB/USB
- Fine grain tuning by 1Hz by pressing rotary encoder button

# KB1OIQ $\mu$ BITX: Future Tasks

- Need to do:
  - RF gain
  - Anti-pop for speaker during RX  $\rightarrow$  TX or TX  $\rightarrow$  RX transition
  - AGC
  - MIC connector and UP/DN buttons
  - Remove RIT from the sketch (fine tuning fixes this)
  - Add Split mode between VFOs
  - Suitable enclosure
  - Audio mixer or 2 separate speakers?
  - Consider putting CAT code back into the sketch
- Very much a work in progress! Lots of FUN!

## Summary (next to last slide)

- BITX40 /  $\mu$ BITX are FUN kits !
- HW and SW are freely documented and easily available
- Active user community on the internet (groups.io and Facebook)
- Hacking the HW and/or SW are encouraged!
- Cost: very reasonable
- KB1OIQ mods are available on Sourceforge:
  - Search for KB1OIQ ham-radio-projects
  - <https://sourceforge.net/projects/kb1oiq-ham-radio-projects/files>
  - Look in the bitx40 and ubitx subdirectories
  - Photos, videos, detailed docs for mods, source code

## Last slide

- I hope you enjoyed this talk – thank you for attending!
- Email: [kb1oiq@arrl.net](mailto:kb1oiq@arrl.net)

# References

- Main website: <http://www.hfsigs.com>
- Discussion group: <https://groups.io/g/BITX20>
- Useful Hacks:
  - <http://bitxhacks.blogspot.com>
  - <http://ubitx.net/>
- Designer: Ashhar Farhan ([VU2ESE](#)), thank you!!
- Arduino Nano:  
<https://www.arduino.cc/en/Main/arduinoBoardNano>
- SI5351B:  
<https://cdn-shop.adafruit.com/datasheets/Si5351.pdf>