Allstar, Mabel and PiTone combine to make a Yaesu Fusion repeater controller

Steve, N8AR and Larry, K8UT

THE PROBLEM

- The Yaesu Fusion DR-1X repeater has a very rudimentary controller in the analog FM mode.
  - CW ID, Tail and TOT timers.
  - No courtesy tones or voice announcements
  - No external control
  - A mediocre squelch system
- The DR-1X switches automatically from analog FM to digital C4FM modes. We only want to control it in the analog FM mode.
- The external interface signals that Yaesu provided for an external controller are poorly defined and not the desired signals.
- The repeater “locks up” if an external controller incorrectly switches modes.

THE PROBLEM (CONTINUED)

- ARCOM worked with YAESU to develop a board called the ADR to supervise proper switching of the DR-1X modes. It can be used with different brands of repeater controllers – ARCOM, CAT, ACC, LINK, SCOM, etc

- Repeater owners using the ADR were still experiencing lockups.

- Justin Reed, NV8Q wrote some macros to use with the SCOM 7330 to control the mode switching. This solution works well but is over $500.

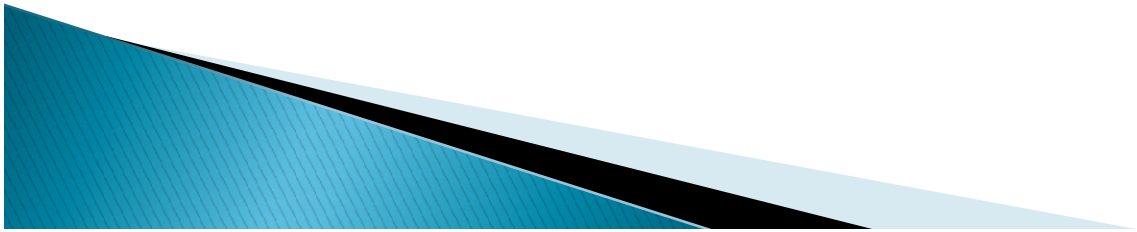
- All of these controllers use firmware based controllers and have no (or use antiquated) remote interfaces.

DESIRED FEATURES FOR OUR REPEATER CONTROLLER

- EASY TO PROGRAM AND CONTROL VIA THE INTERNET

- FEATURE RICH
  - SETTABLE TIMERS
  - CHANGEABLE COURTESY TONES
  - VOICE AND CW ID
  - SYNTHESIZED VOICE AND WAVE FILES
  - SCHEDULED MESSAGES– “THE TIME IS……”
  - TAIL MESSAGES

- GOOD SUPPORT AND LARGE USER BASE



THE SEARCH

- Initially, we considered writing a simple repeater controller program to run on a Raspberry Pi 3 (rPi3). We played with:
  - Code to generate a CW ID
  - Code to play a WAV file
- Became aware of the Open Repeater Project which is the development of a low cost, low power, but feature rich duplex Linux based amateur radio repeater controller using single board computers (SBCs) like the Raspberry Pi 2 and Beaglebone Black.
  - Not going anywhere fast
  - Would not support our need to switch modes on the DR-1X
- (Re)-Discovered the repeater controller capability of Allstar known as Asterisk/app_rpt

What is Asterisk/ app-rpt

- 1999 – Mark Spencer (Huntsville AL) started a Linux software company. They needed phone services but discovered available systems were very expensive. So they wrote their own PBX software called Asterisk.

- Asterisk is written to allow additional applications to extend its capability.

- The app_rpt repeater controller application was written by Jim Dixon, WB6NIL(SK) and Steve Rodgers, WA6ZFT in the mid 2000's to fulfill his need for a repeater controller and VOIP interface. Required a PC and a special interface card or USB Radio Interface Module (RIM)

- AllStar Asterisk/app_rpt is a complete HAM radio repeater controller and link system built on the Asterisk VOIP platform.

ENTER THE RASPBERRY PI

- 2014 – Doug Crompton WA3DSP began work on a version to run on a Raspberry Pi. Distros now available for the rPi 2 and 3 running Archlinux.

- N4IRS, Steve Zingman recently released a distro running Raspbian.

- So we have a powerful repeater controller running on a rPi but how to implement the task of switching the DR-1X to the external controller using app_rpt? Modify app_rpt?

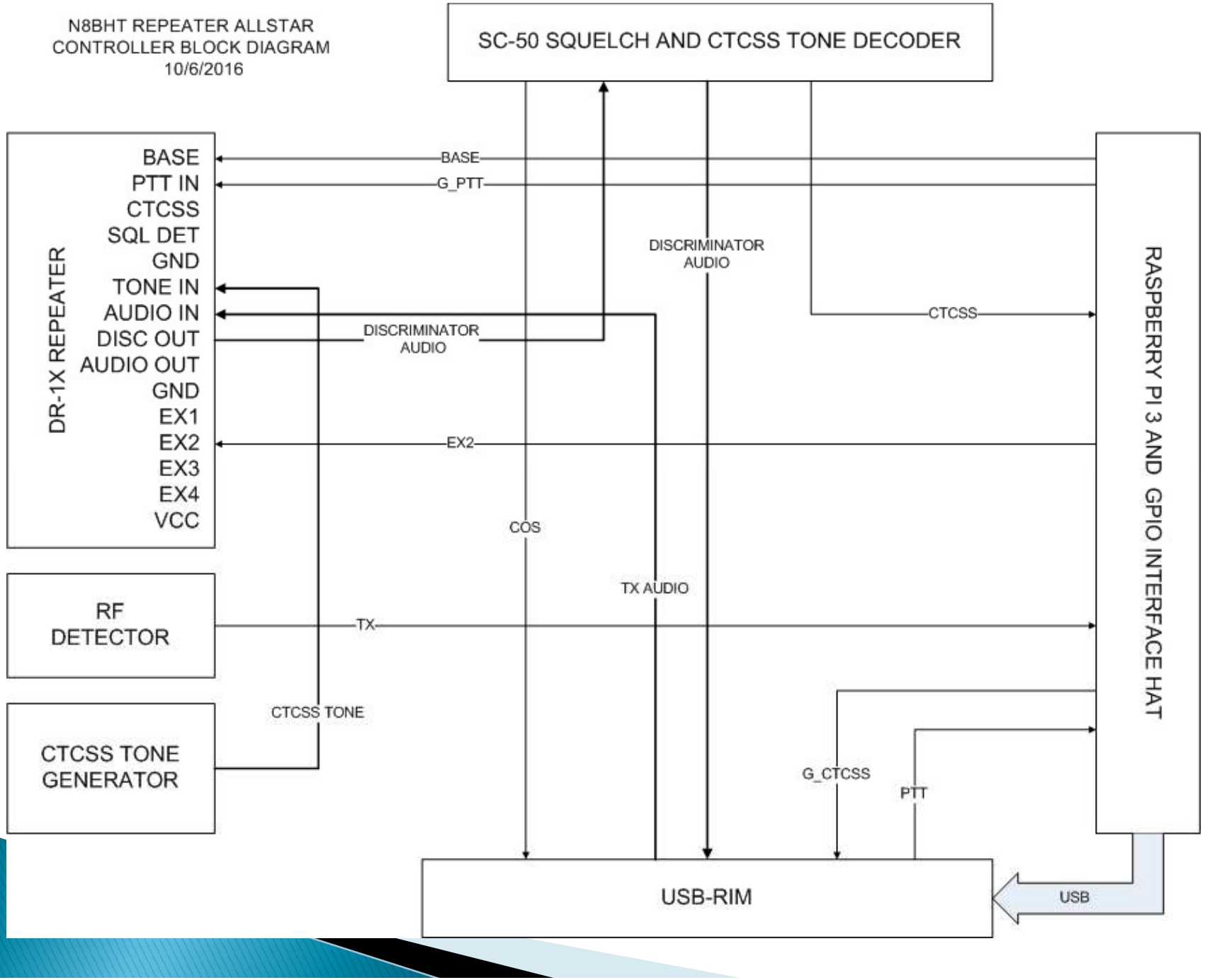
- app_rpt is open source so the code could be modified and recompiled to perform this task. We soon discovered that app_rpt is more than 20,000 lines of code. Not a SMOP as far as we were concerned!

MABEL IS CONCEIVED

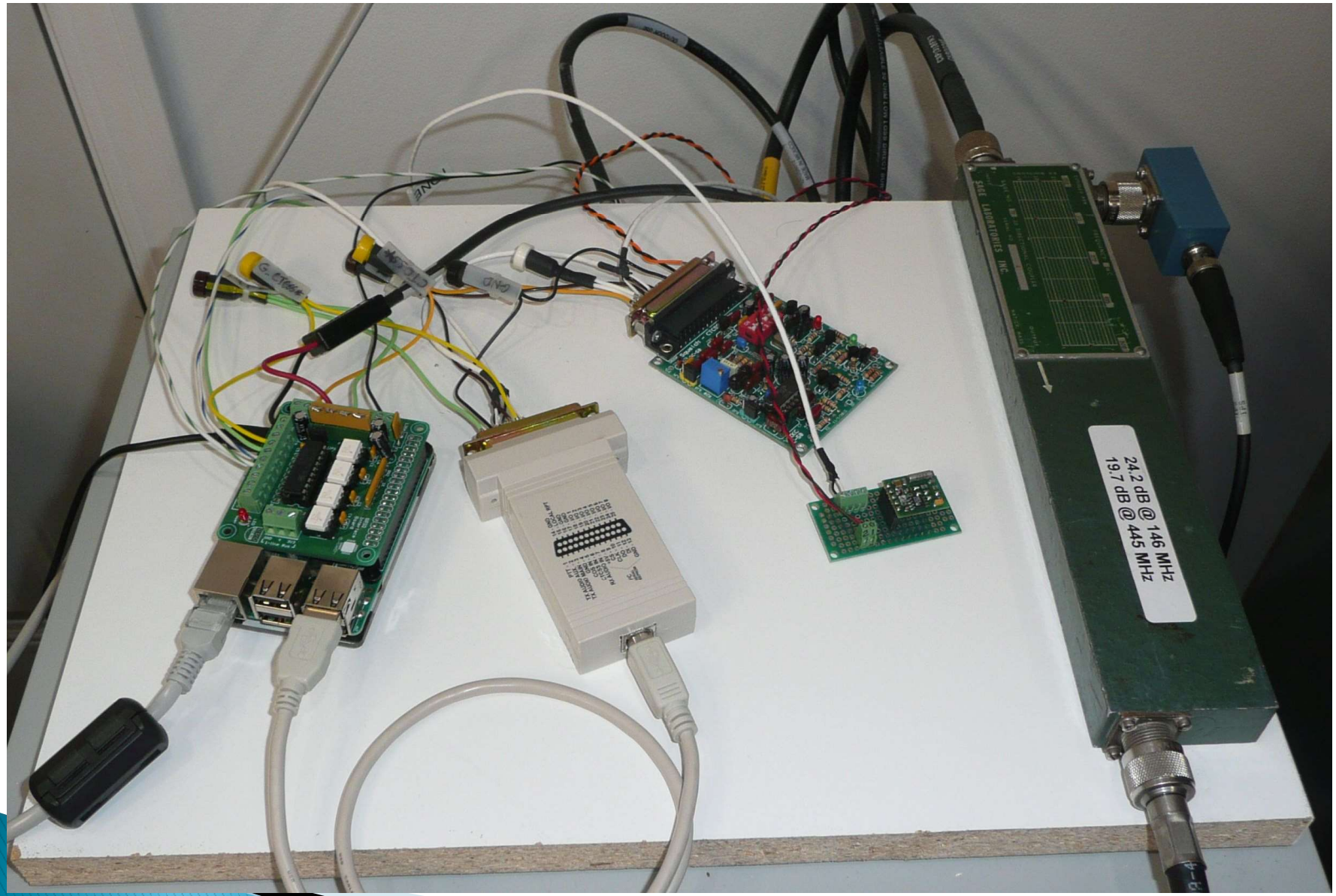
- We conjectured that we could route some of the signals used by app_rpt through another software program before it was sent to app_rpt. We would call this concept MABEL.

- MABELwould look for a valid analog FM signal (i.e. CTCSS) and switch the DR-1X mode from DIGITAL to FM-FM. Once the switch was complete, it would forward the signals on to app_rpt switching back to digital after loss of the FM signal.

- In additon to the rPi, we need some other hardware pieces
  - Detect PL (CTCSS) and Squelch (COS). We use the SC-50.
  - A USB RIM because that's what app_rpt requires for audio and the CTCSS, COS and PTT signals. We now use the RA-35.
  - Interface electronics to the rPi. Was Lo-Tech hat, now our design.
  - A method to detect that the repeater is transmitting. We use an RF detector instead of cutting into the repeater internal wiring carrying the PTT signal.


N8BHT REPEATER ALLSTAR
CONTROLLER BLOCK DIAGRAM
10/6/2016
SC-50 SQUELCH AND CTCSS TONE DECODER
DR-1X REPEATER
BASE
PTT IN
CTCSS
SQL DET
GND
TONE IN
AUDIO IN
DISC OUT
AUDIO OUT
GND
EX1
EX2
EX3
EX4
VCC
RF DETECTOR
CTCSS TONE GENERATOR
RASPBERRY PI 3 AND GPIO INTERFACE HAT
USB-RIM
BASE
G_PTT
DISCRIMINATOR AUDIO
CTCSS
DISCRIMINATOR AUDIO
EX2
COS
TX AUDIO
TX
CTCSS TONE
G_CTCSS
PTT
USB

FIRST PROTOTYPE CONTROLLER



ALLSTAR FOR rPi DOES NOT GENERATE TX PL

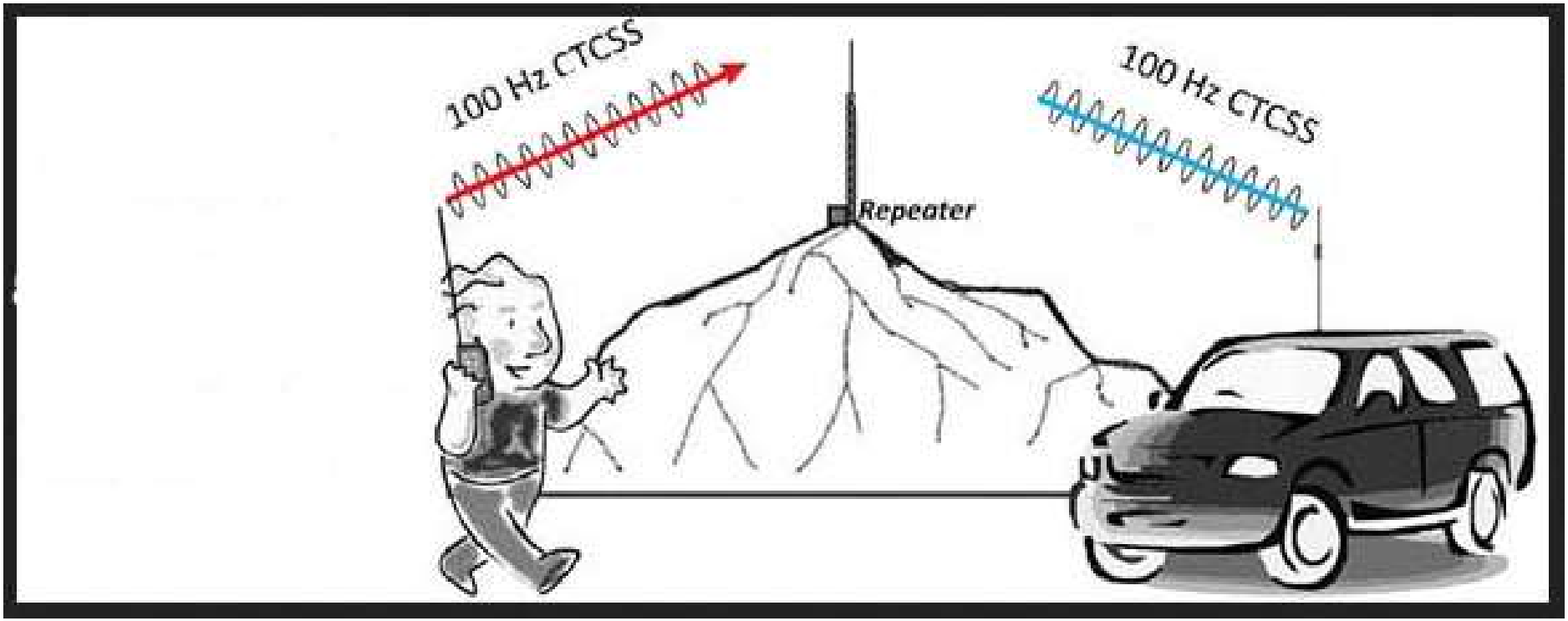
- Allstar for PC has a choice of two radio channel drivers
  - USB Radio – DSP does all decoding (PL, squelch and DTMF) and generates TX PL
  - SimpleUSB – Only does DTMF decoding.
- SimpleUSB is used with the rPi distros because DSP loads the rPi which affects audio quality. But SimpleUSB does not generate TX PL.
- We need to transmit a PL tone so that non-Fusion radios (i.e. analog FM) do not open their squelch on C4FM digital signals.
- We wondered if we could use the rPi and some custom electronics to generate TX PL. PiTone is born

N8BHT's Assigned Tone: 110.9Hz

TABLE OF COMMON PL TONES (in Hz)

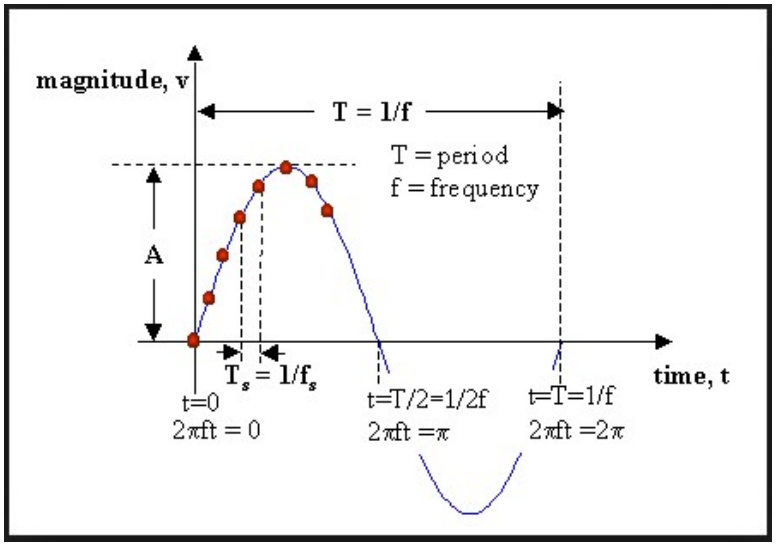
67.0	94.8	131.8	171.3	203.5
69.3	97.4	136.5	173.8	206.5
71.9	100.0	141.3	177.3	210.7
74.4	103.5	146.2	179.9	218.1
77.0	107.2	151.4	183.5	225.7
	110.9	156.7	186.2	229.1
82.5	114.8	159.8	189.9	233.6
85.4	118.8	162.2	192.8	241.8
88.5	123.0	165.5	196.6	250.3
91.5	127.3	167.9	199.5	254.1

You Need Two Tone Generators
One to Get In; Another to Get Out



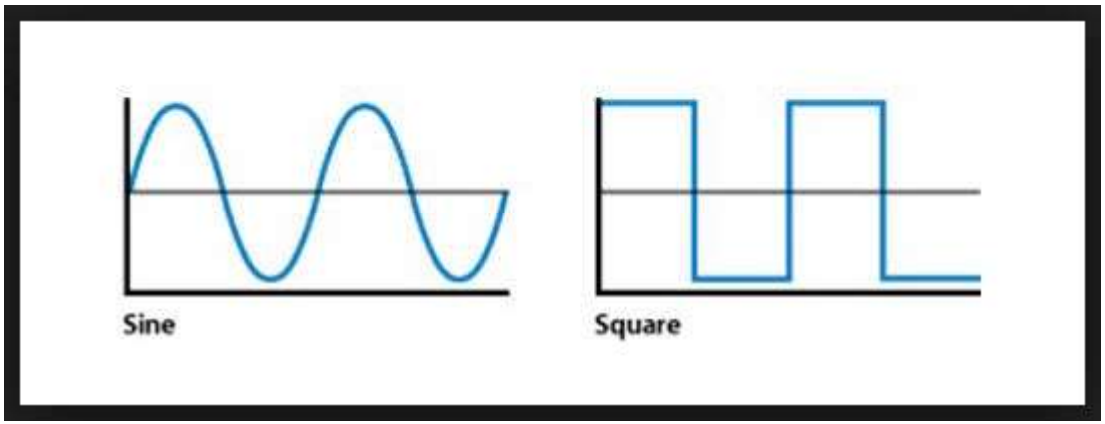
Attributes of a PL Tone

- Sine Waveform
- Specific Frequency
  - 110.9Hz or 9.017 milliseconds/cycle
- Frequency Tolerance of .5%
  - +/- .55 Hz
- Sufficient Amplitude to Drive the Repeater
  - 600 millivolts AC



The Challenge

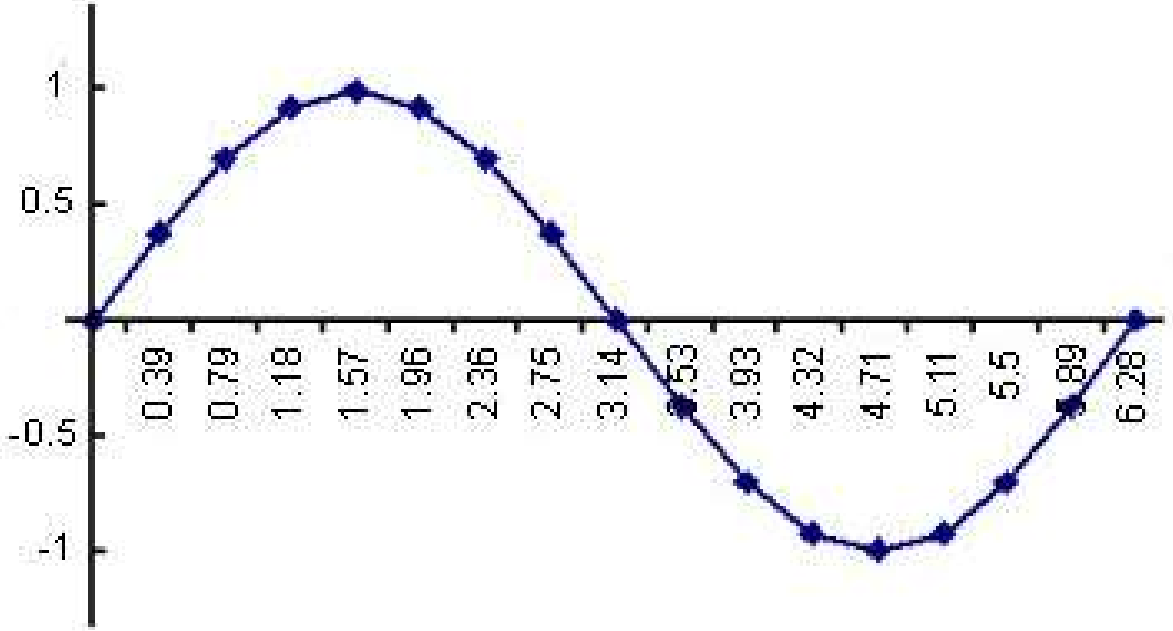
- How to make an Analog Waveform
- With a computer that only makes 1s and 0s?



The solution? Use a DAC!

How Will That Work?

- Reduce the sine wave to individual DC voltages



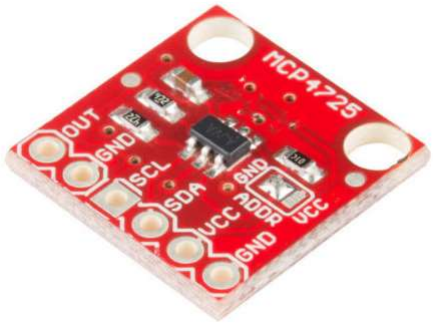
The plan:

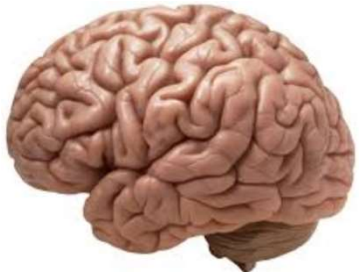
A computer calculates the amplitude and controls the timing.

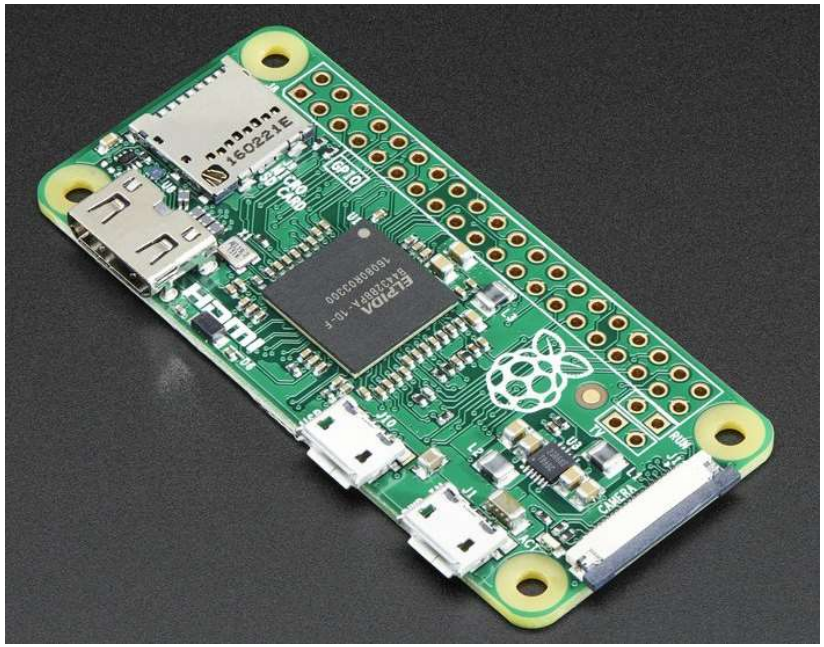
A digital -to- analog convertor produces the DC voltages.

Some Assembly Required…

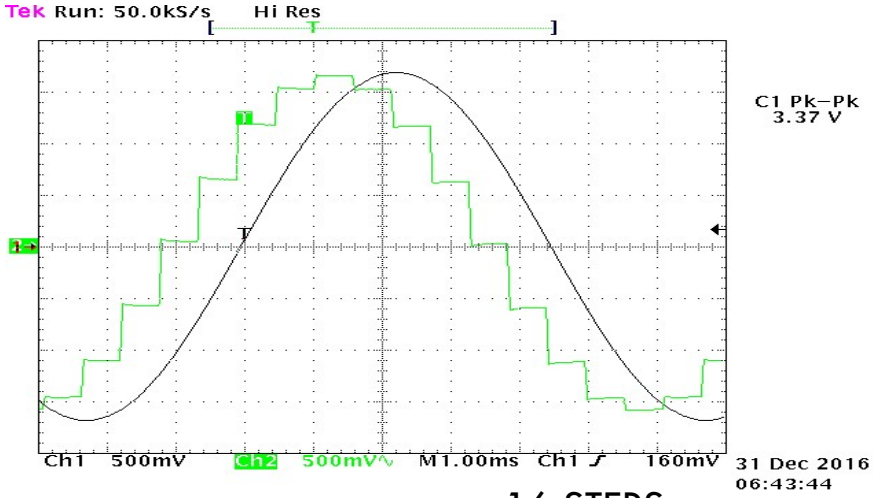
- Bill of Materials
  - A Raspberry Pi "Zero" ($.99)
  - An MCP4725 D-to-A converter ($1.11)
  - A lot of programming…



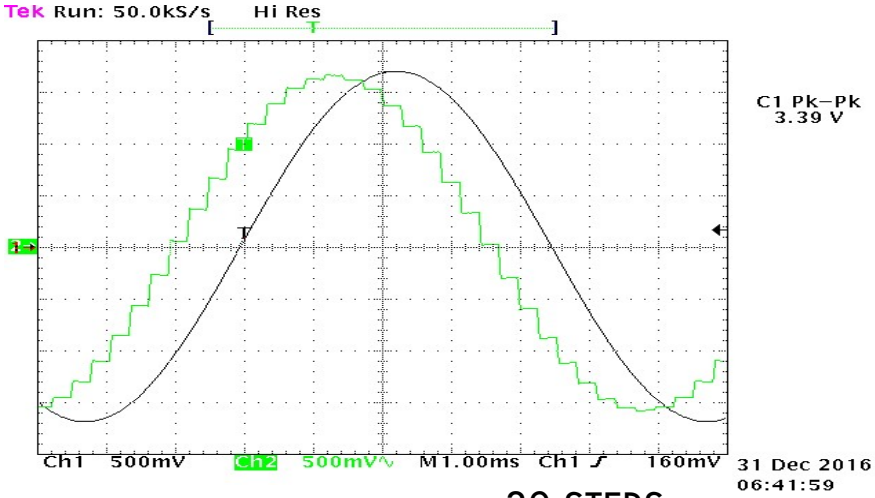




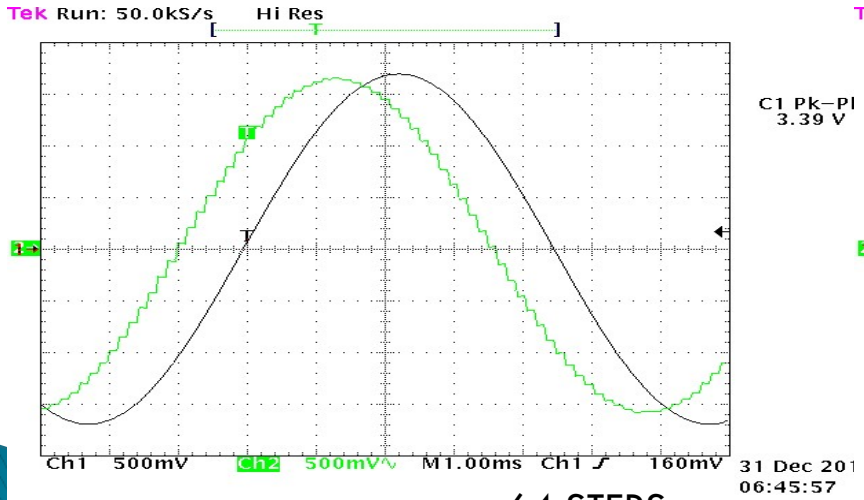
From Square Waves to a Sine Wave



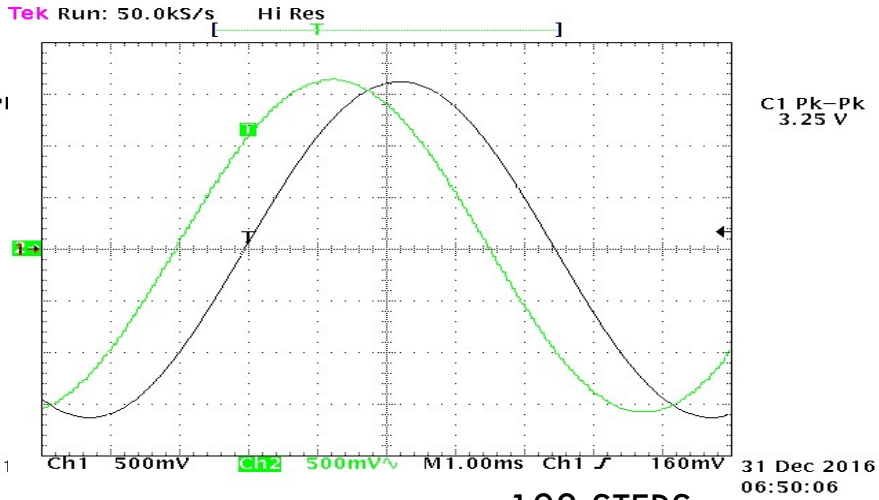
16 STEPS



32 STEPS



64 STEPS



128 STEPS

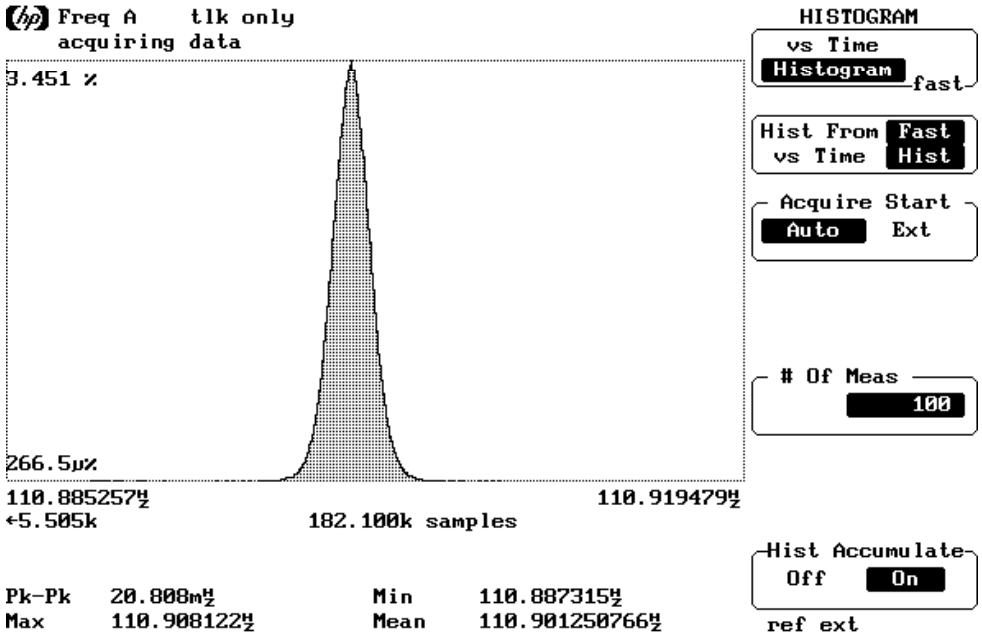
But the Steps are Expensive

- 16 steps: interrupt every 563 microseconds
- 32 steps: interrupt every 281 microseconds
- 64 steps: interrupt every 140 microseconds
- 128 steps: interrupt every 70 microseconds

Expense affects frequency stability and performance of other applications on the Pi

PiTone Configuration

- A compromise:
  - Produce a sine wave from 32 steps
  - Shape the waveform using an op amp and filters
  - Add stability by assigning Real-time priority to PiTone
  - Result:
    Rock-solid stability
    within 12 mHertz



PiTone for the N8BHT Repeater Summary

- Command line options
  - 16, 32, 64 or 128 steps (we use 32 steps)
  - Any tone from 50 to 180 Hz (we use 110.9 Hz)
  - Output settable from 100% to 10% of VCC (nominally 3.3 VDC) centered on ½ VCC.
  - Reverse burst option

- Tested with Allstar, MABEL and PiTone running on the same rPi3
  - Must run PiTone as realtime process, i.e. “chrt -r 99 ./pitone 110.9 32 100 &”
  - Within .02 Hz of desired frequency
  - Less than +/_ 0.1 Hz deviation
  - No effect on Allstar or MABEL performance
  - Max CPU utilization less than 7% of one core

- N8BHT Repeater Configuration of PiTone uses custom MABEL DR-1X Interface board
  - Header to install MCP4725 breakout board ($1.12 on eBay from goodmodule)
  - 3 pole low pass DC coupled active filter with a gain of one.
  - 10K ohm potentiometer to set output level
  - AC coupled output

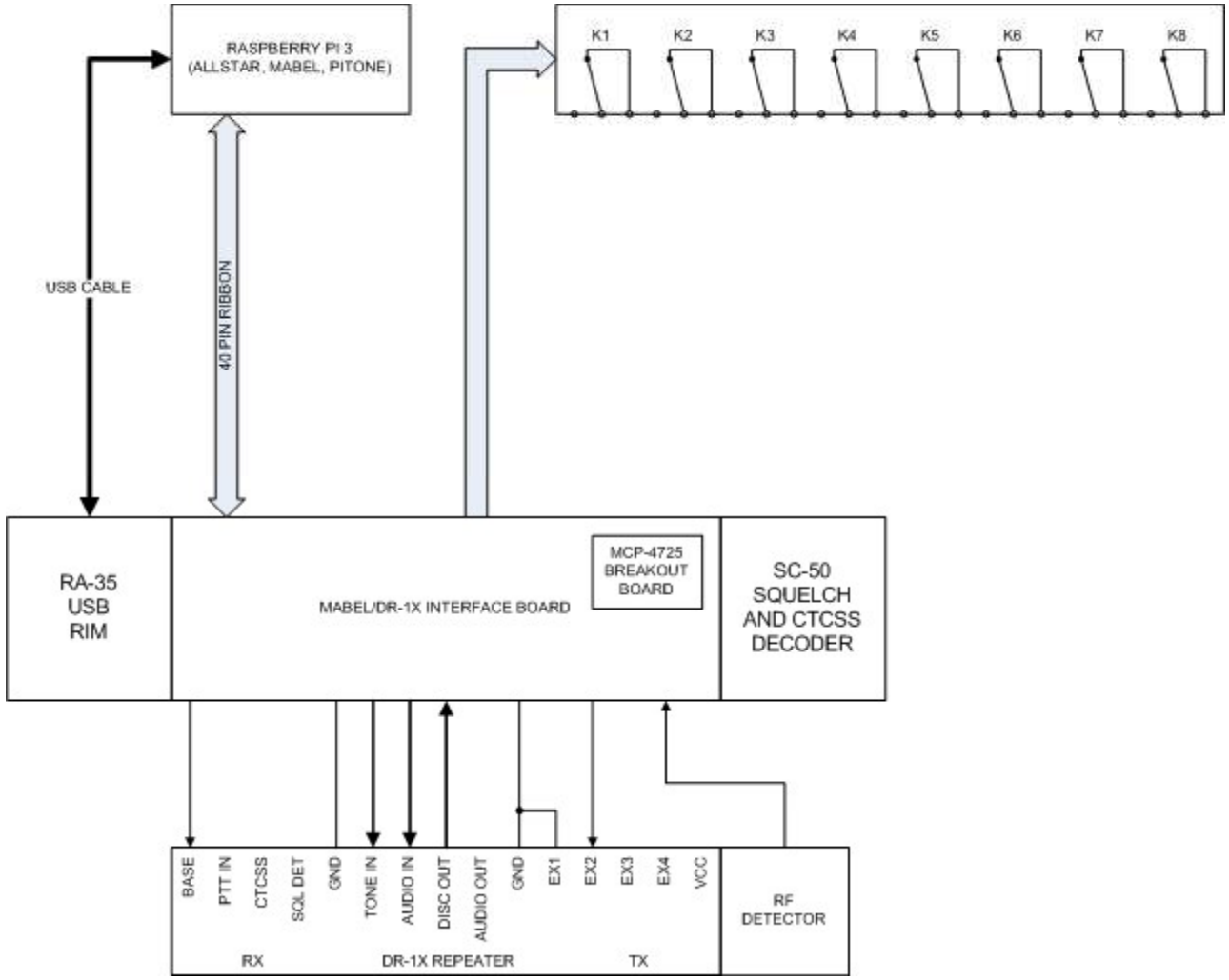
MABEL/DR-1X Interface Electronics

- In addition to the SC-50 and RA-35 from Masters Communications (http://www.masterscommunications.com/) our second prototype utilized an rPi HAT from Lo-Tech electronics, a PL generator from COMSPEC, a breadboard low pass PL filter, a MCP-4725 board and a whole lot of wiring.

- We also wanted to interface a Sainsmart 8 relay module for other control functions.

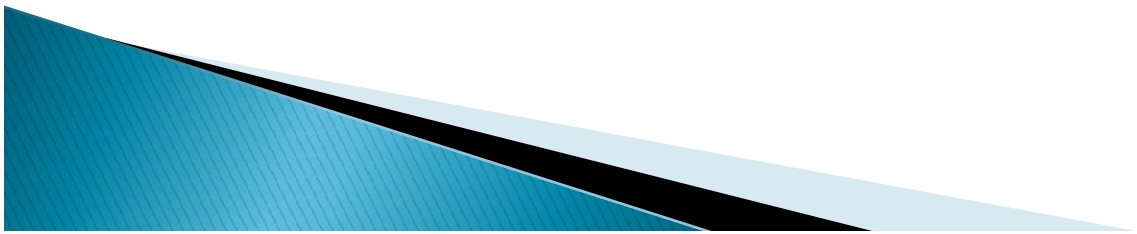
- We decided to design a custom interface board. The SC-50, RA-35, MCP-4725 with a low pass filter, rPi and Sainsmart Relay board would all plug into this board. Screw terminal blocks would be provided for all wire connections to the DR-1X control connector and the RF detector.

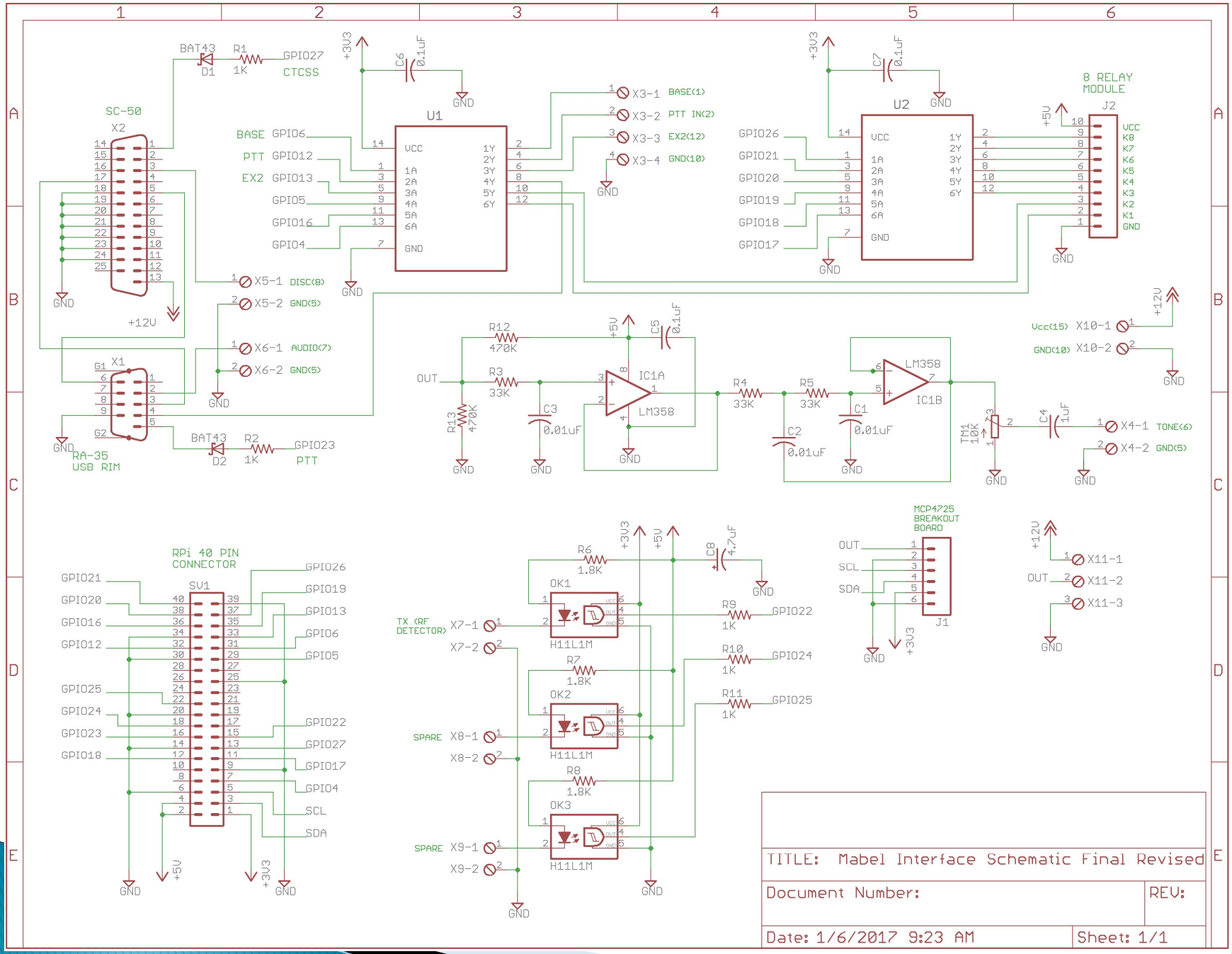
MABEL/DR-1X Interface Electronics (continued)



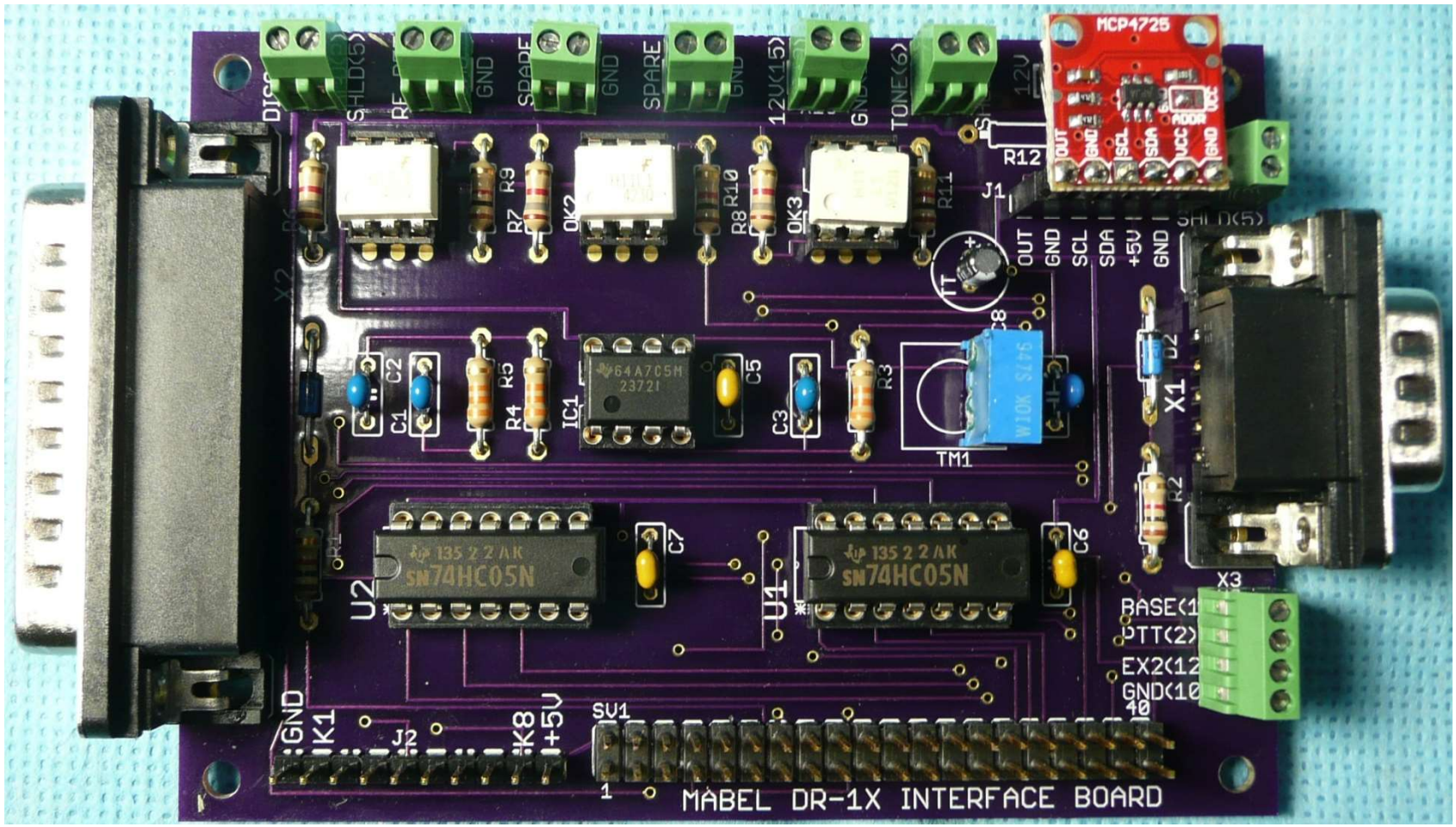
MABEL/DR-1X Interface Electronics Features

- All required interface electronics between the rPi, SC-50 and RA-35 and the DR-1X repeater
- One optically isolated input for the RF detector.
- Two spare optically isolated inputs.
  - Revision 2 board provides more flexibility on these inputs
- 8 spare open collector outputs (74HC05)
  - 10 pin header matches SainSmart 8 relay board
- SC-50 CTCSS/Squelch board and RA-35 plug into mating d-subs
- rPi plugs into mating 40 pin connector via a ribbon cable
- MCP-4725 breakout board for PiTone PL plugs into 6 pin header
- Interface wires to the DR-1X and RF detector use screw terminals
- 3 pole active lowpass filter with output poteniometer for PiTone PL
  - Can also be used to filter a PL signal from an external board

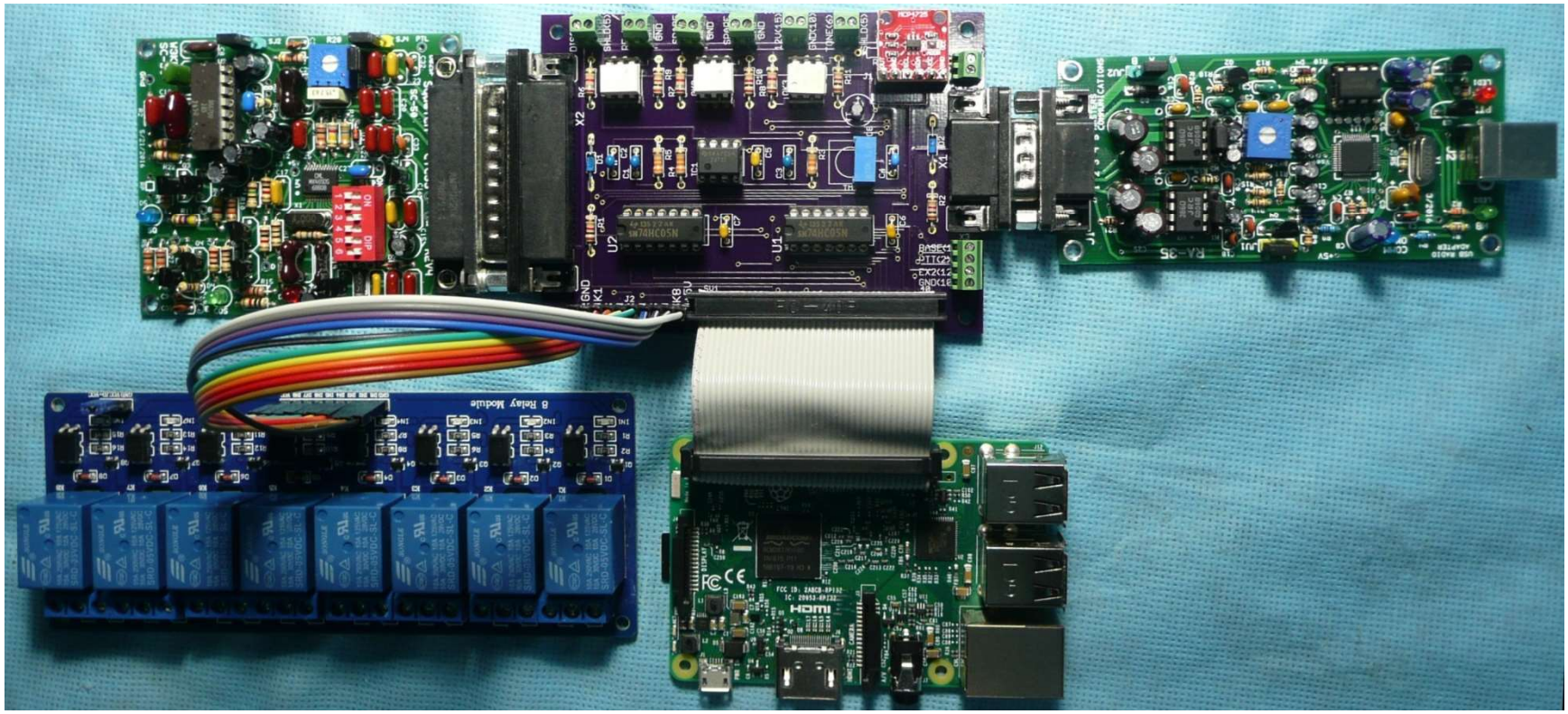



SC-50
X2
BAT43
D1
R1
1K
GPIO27
CTCSS
+3V3
C6
0.1uF
GND
U1
BASE GPIO6
PTT GPIO12
EX2 GPIO13
GPIO5
GPIO16
GPIO4
VCC
1A
2A
3A
4A
5A
6A
GND
1Y
2Y
3Y
4Y
5Y
6Y
X3-1 BASE(1)
X3-2 PTT IN(2)
X3-3 EX2(12)
X3-4 GND(10)
U2
C7
GPIO26
GPIO21
GPIO20
GPIO19
GPIO18
GPIO17
8 RELAY MODULE
J2
+5V
VCC
K8
K7
K6
K5
K4
K3
K2
K1
GND
X5-1 DISC(8)
X5-2 GND(5)
X6-1 AUDIO(7)
X6-2 GND(5)
+12V
X1
G1
G2
RA-35
USB RIM
BAT43
D2
R2
1K
GPIO23
PTT
R12
470K
R3
33K
R13
470K
C3
0.01uF
C5
0.1uF
IC1A
LM358
OUT
R4
33K
R5
33K
C2
0.01uF
C1
0.01uF
LM358
IC1B
TM1
10K
C4
1uF
X4-1 TONE(6)
X4-2 GND(5)
Vcc(15) X10-1
GND(10) X10-2
RPi 40 PIN CONNECTOR
SV1
GPIO21
GPIO20
GPIO16
GPIO12
GPIO25
GPIO24
GPIO23
GPIO18
GPIO26
GPIO19
GPIO13
GPIO6
GPIO5
GPIO22
GPIO27
GPIO17
GPIO4
SCL
SDA
R6
1.8K
C8
4.7uF
OK1
H11L1M
TX (RF DETECTOR) X7-1
X7-2
R9
1K
GPIO22
R7
1.8K
OK2
R10
1K
GPIO24
SPARE X8-1
X8-2
R11
1K
GPIO25
R8
1.8K
OK3
SPARE X9-1
X9-2
MCP4725 BREAKOUT BOARD
OUT
SCL
SDA
J1
X11-1
X11-2
X11-3
TITLE: Mabel Interface Schematic Final Revised
Document Number:
REV:
Date: 1/6/2017 9:23 AM
Sheet: 1/1

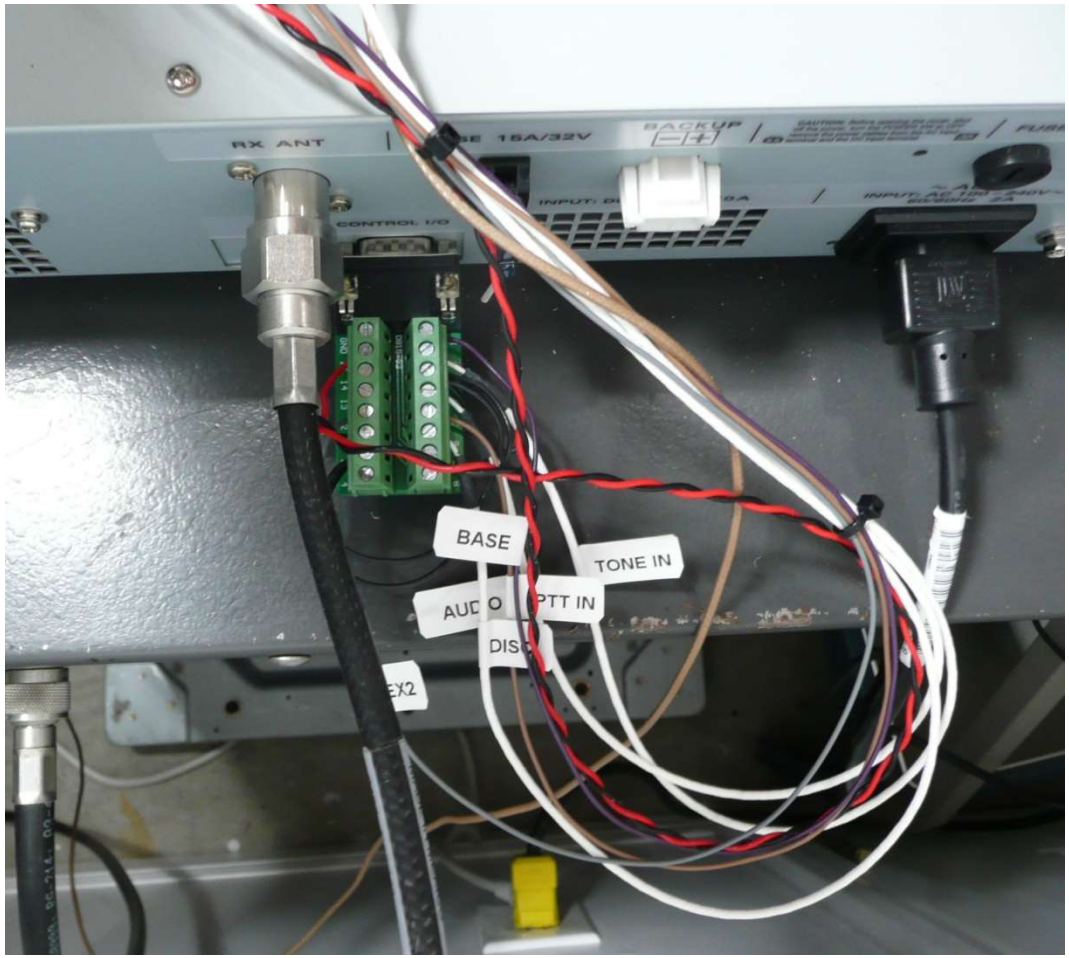
MABEL/DR-1X Interface PCB



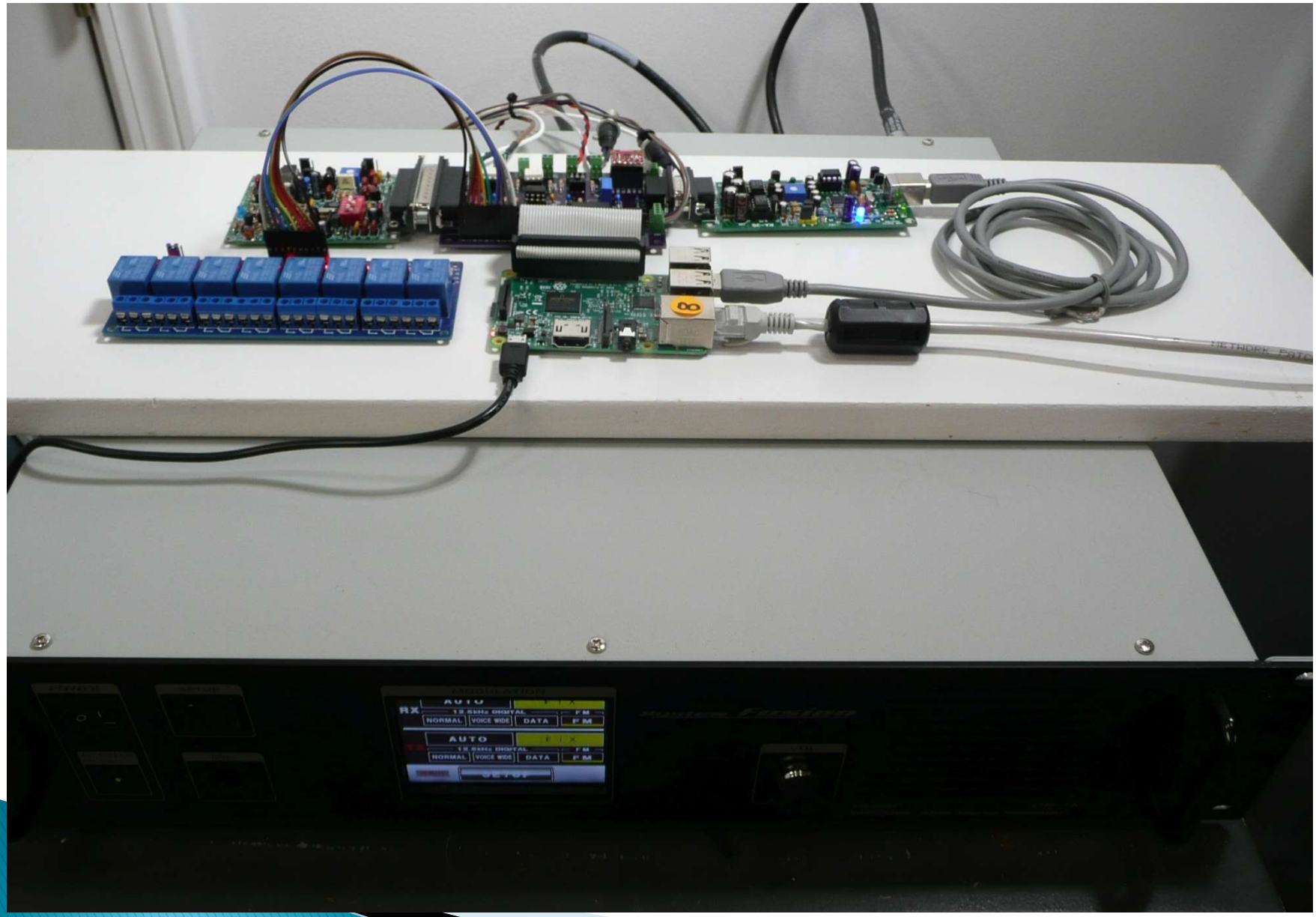
MABEL/DR-1X ALLSTAR CONTROLLER



CONNECTIONS AT THE DR-1X REPEATER

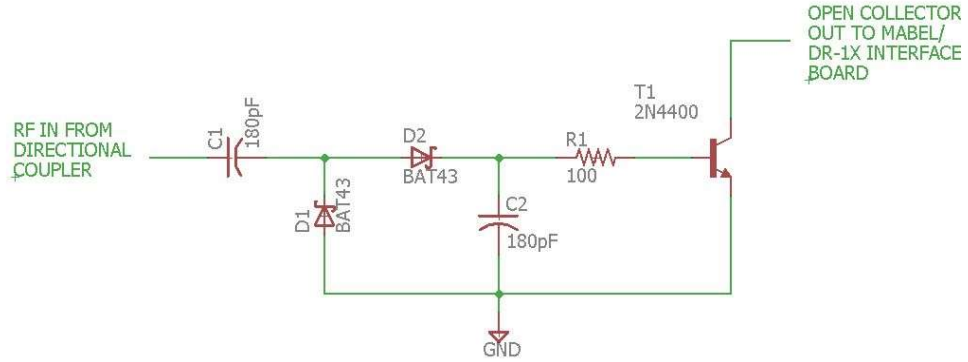


MABEL/DR-1X ALLSTAR CONTROLLER IN OPERATION



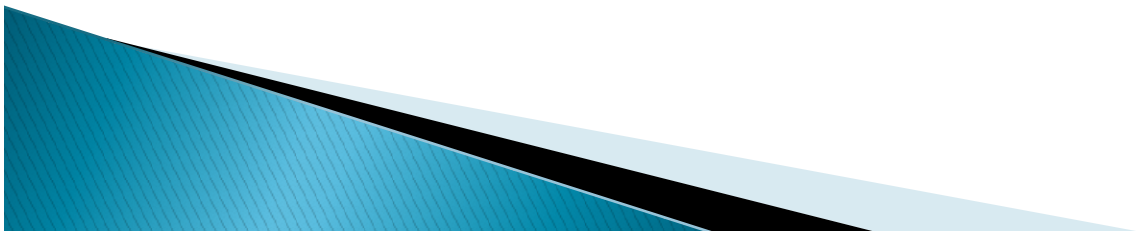
THE RF DETECTOR

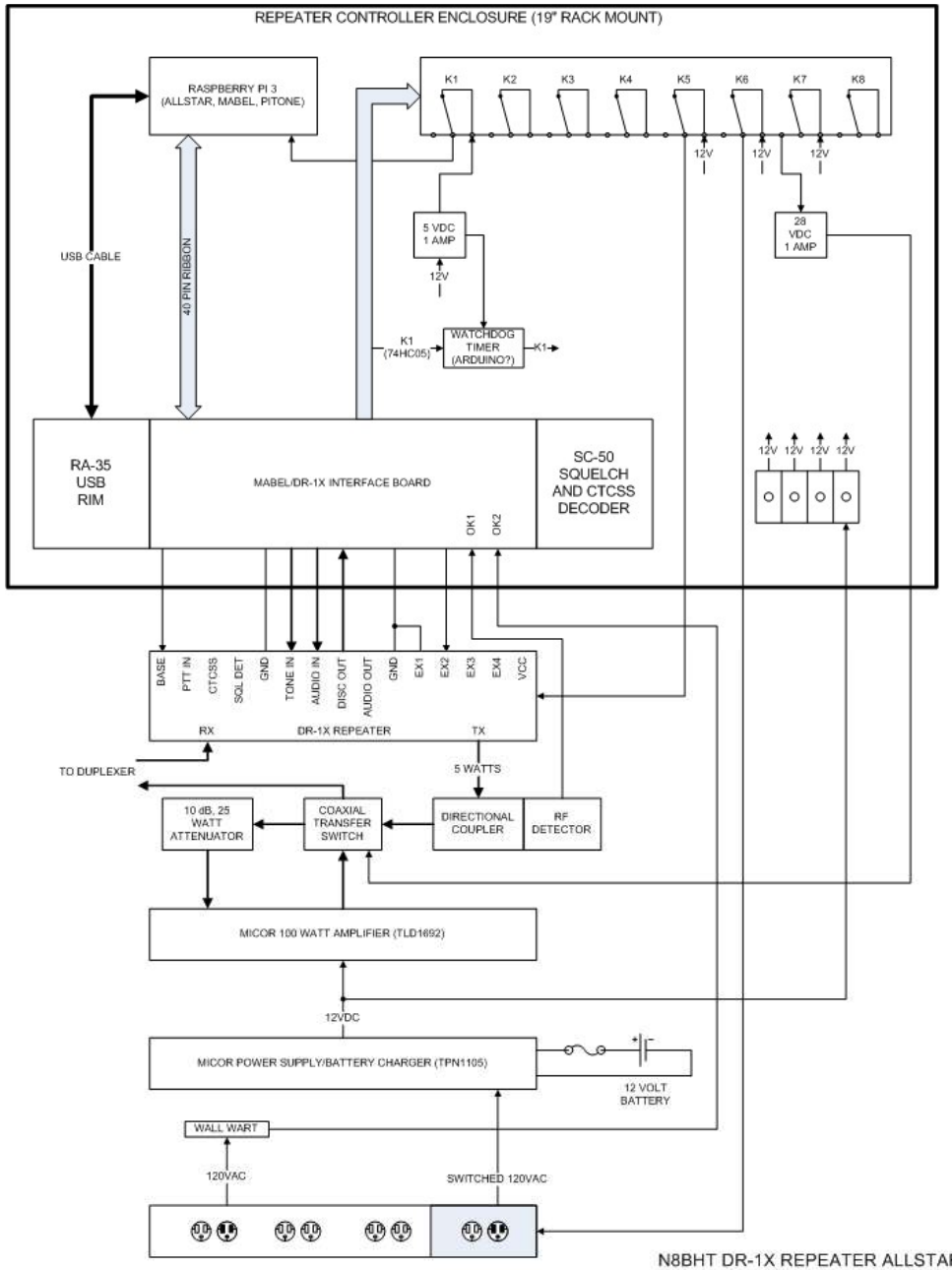
- Use a directional coupler and an RF detector with open collector output



THE PROPOSED N8BHT REPEATER CONFIGURATION

- Motorola Micor power supply with automatic battery backup switchover
- Motorola 100 watt amplifier driven by 5 watt DR-1X output through a 10 dB attenuator
- Direction coupler and RF detector routed to MABEL
- Transfer relay to switch to 5 watt output when running on battery power (activated by SainSmart relay)
- AC power loss detected on spare optically isolated input
- Sainsmart relay to turn off power to the DR-1X
- SainSmart relay to turn off AC power for testing purposes
- AT-tiny board used as a watchdog to reset the rPi using a SainSmart relay
- All relays controlled using bash scripts and the gpio feature of wiringpi. Control via SSH over the internet or DTMF over the air





N8BHT DR-1X REPEATER ALLSTAR
CONTROLLER BLOCK DIAGRAM
1/4/2017

Thank You!

Steve, N8AR and Larry, K8UT