

# Automating the Ameritron RCS-8 and RCS-10 Control Heads for Remote Coax Switches

Automating Ameritron RCS 8 and RCS 18 Control Heads.docx - 23-Sep-2016 13:37 - Terry G. Glagowski, Ph.D. / W1TR / AFA1DI

## INTRODUCTION

This document describes how the Ameritron RCS-8 and RCS-10 remote control head units can be modified so they are controlled by inexpensively available USB relay modules. Software can then read the exact frequency of the transceiver and decides which antenna to use. The USB relay module used is a SMAKN or SainSmart SRD-12VDC-SL-C 8 relay module (\$20-\$30).

- 1) <http://www.sainsmart.com/sainsmart-4-channel-12-v-usb-relay-board-module-controller-for-automation-robotics-1.html>
- 2) [https://www.amazon.com/SMAKN%C2%AE-Channel-Programmable-Computer-Control/dp/B00NTUE6Zl/ref=pd\\_sim\\_147\\_5?ie=UTF8&psc=1&refRID=TJC95EE6CN2P9T9S0PGN](https://www.amazon.com/SMAKN%C2%AE-Channel-Programmable-Computer-Control/dp/B00NTUE6Zl/ref=pd_sim_147_5?ie=UTF8&psc=1&refRID=TJC95EE6CN2P9T9S0PGN)

Ameritron makes several cost effective remote coax switches for amateur radio: the RCS-8 and RCS-10 models.

<http://www.ameritron.com/Product.php?productid=RCS-8V> (\$170).

<http://www.ameritron.com/Product.php?productid=RCS-10> (\$180).

They also make an RCS-12 controller which can select the antenna for a given band based on information provided by the transceiver.

<http://www.ameritron.com/Product.php?productid=RCS-12> (\$310)

The band information provided to the RCS-12 from the transceiver is an analog voltage for ICOM, RS-232 serial protocol for Kenwood, and 4 bit binary band code for Yaesu. This works well for HAM bands, but not for MARS frequencies or situations where one segment of the band is covered by one antenna, and another segment of the band is covered by a different antenna. Also, the RCS-12 is a bit pricey and the modification to the RCS-8 and RCS-10 can be performed for a fraction of the cost. Other manufacturers offer very high quality automatic switching as well, but their prices are in the stratosphere.

This document describes how you can easily and quickly modify the Ameritron RCS-8 and RCS-10 control heads to operate manually as they did originally, or alternatively be controlled by these USB Relay Control modules. Typically a control head can be modified in about 1-2 hours.

## OVERALL APPROACH

The overall approach to performing the modification is:

- 1) Test things after each of the following steps to simplify system testing at the end.
- 2) Remove the cover from the control head box.
- 3) Remove the PCB:
  - a. It is not necessary to disconnect any existing wires except as noted below.
  - b. Remove the knob from the antenna switch shaft
  - c. Remove the hex nut holding the antenna switch
  - d. Remove the hex head holding the power switch
  - e. For the RCS-10, remove the bottom screws that hold the PCB
  - f. Push the board away from the front panel
  - g. Save and put aside the hardware associated with the antenna switch and the power switch
- 4) For the RCS-10, create a mounting bracket for the USB Relay Module, mount it on the back panel of the chassis.
- 5) For the RCS-8, mount the USB Relay Module to the bottom of the chassis with standoff spacers.
- 6) Make a hole in the back panel for the USB connector
  - a. Drill a pilot hole then 3/8 inch hole on the rear panel in alignment with the USB connector
  - b. Use a 9 pin tube socket 3/4 inch Greenlee chassis punch to make the hole larger  
<https://www.amazon.com/Greenlee-Chassis-Punch-Holes-Greenlee/dp/B00BUG713C>
- 7) Replace the power switch which is DPDT with another which is DPDT center off.  
[https://www.amazon.com/Amico-125V-Position-Toggle-Switch/dp/B008DFYDNE/ref=pd\\_lpo\\_263\\_bs\\_t\\_1?ie=UTF8&psc=1&refRID=0X268FBHKTQTK2EM7MBX](https://www.amazon.com/Amico-125V-Position-Toggle-Switch/dp/B008DFYDNE/ref=pd_lpo_263_bs_t_1?ie=UTF8&psc=1&refRID=0X268FBHKTQTK2EM7MBX)
- 8) For the RCS-8, rewire the power switch so that the incoming power is on the common switch terminal and the outgoing power goes to the manual switches (switch up) or the USB relay board (switch down).
- 9) For the RCS-8, rewire the LED indicators to connect to the same switch terminals as the antenna relays. This can be accomplished by cutting traces from the antenna selection switch to the LEDs and wiring them instead to different ports on the switch (see the schematic).
- 10) Connect each of the USB Relay Board common relay terminals to the +12 supply terminal on the relay board.
- 11) Connect the +12V relay board terminal to the power switch active in automatic mode (switch down) on the switch PCB.
- 12) Connect the GND relay board terminal to the ground on the switch PCB.
- 13) Connect antenna relay control terminals to the normally open relay contacts, 1-5 for RCS-8, 1-8 for RCS-10.
- 14) Reassemble the box and install in the original operating position.
- 15) Connect the USB connectors to the computer, possibly through a hub.

## CONSTRUCTION DETAILS

Removing the PCB from the front panel RCS-8

Remove the Antenna Switch Knob RCS-8

Use a small HEX wrench.



After Removing Antenna Switch Knob RCS-8

Remove the Hex Nuts from the Potentiometer and the Power switch



After Removing Hex Nuts on Antenna Switch and Power Switch

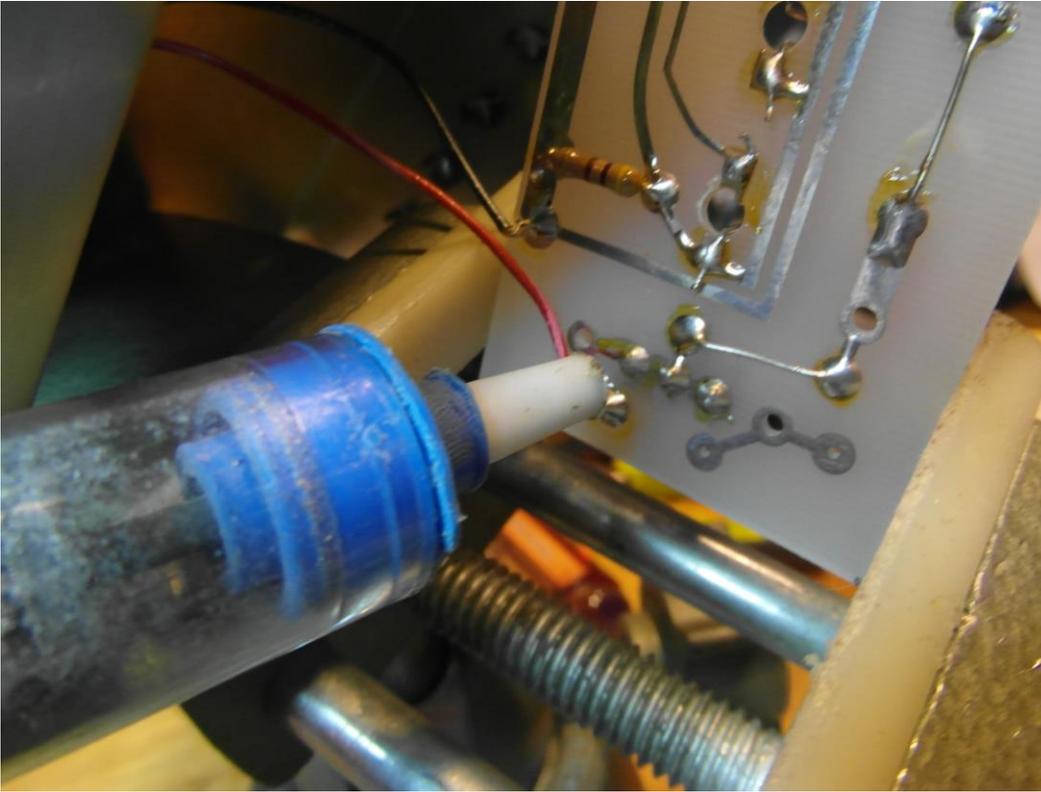


After Removing PCB from Front Panel



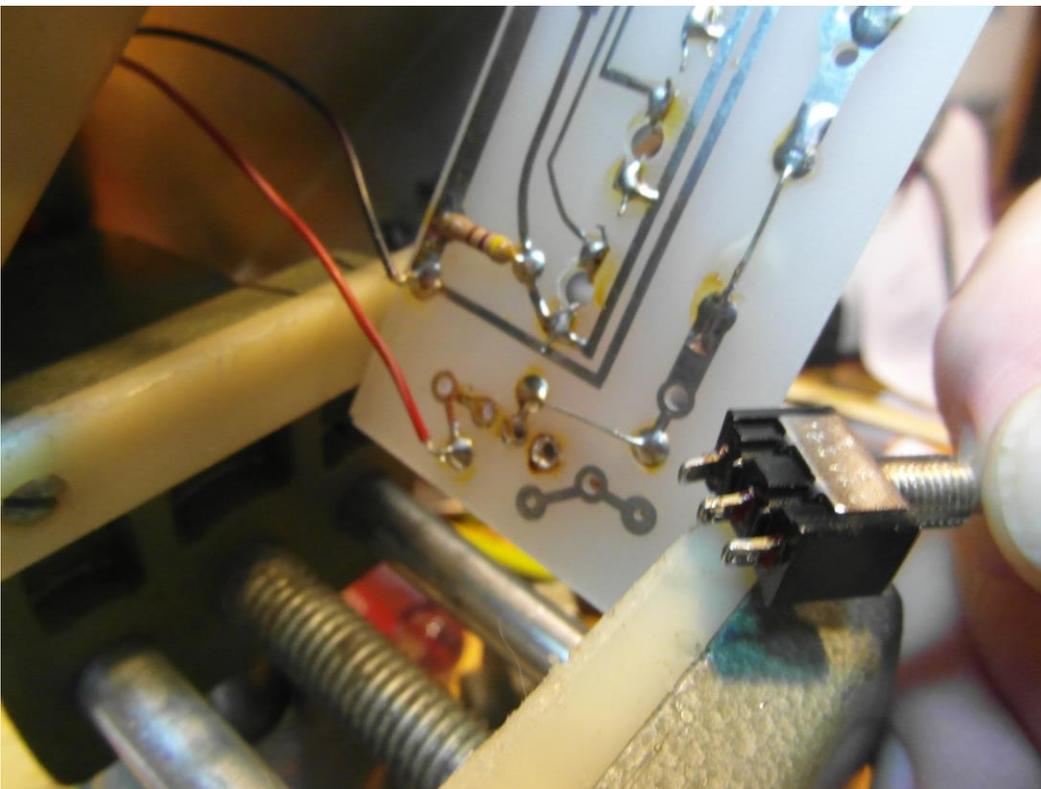
Replace Power Switch RCS-8

Desolder and Remove Power Switch RCS-8

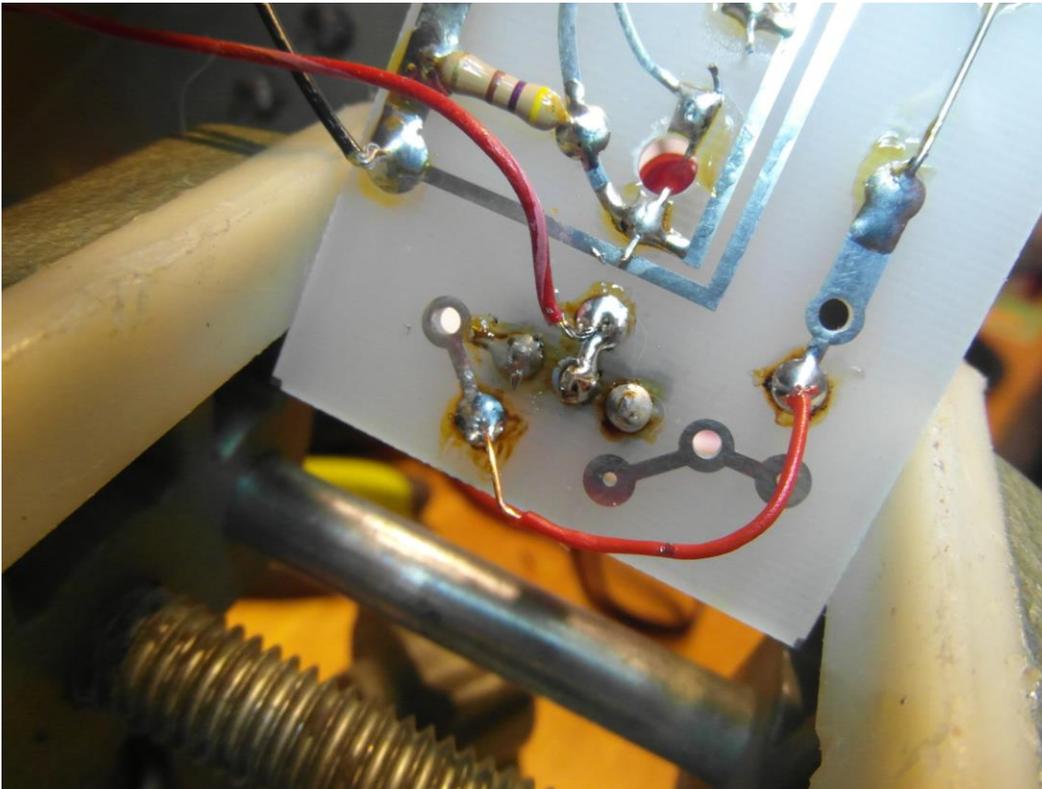


Power Switch Removed RCS-8

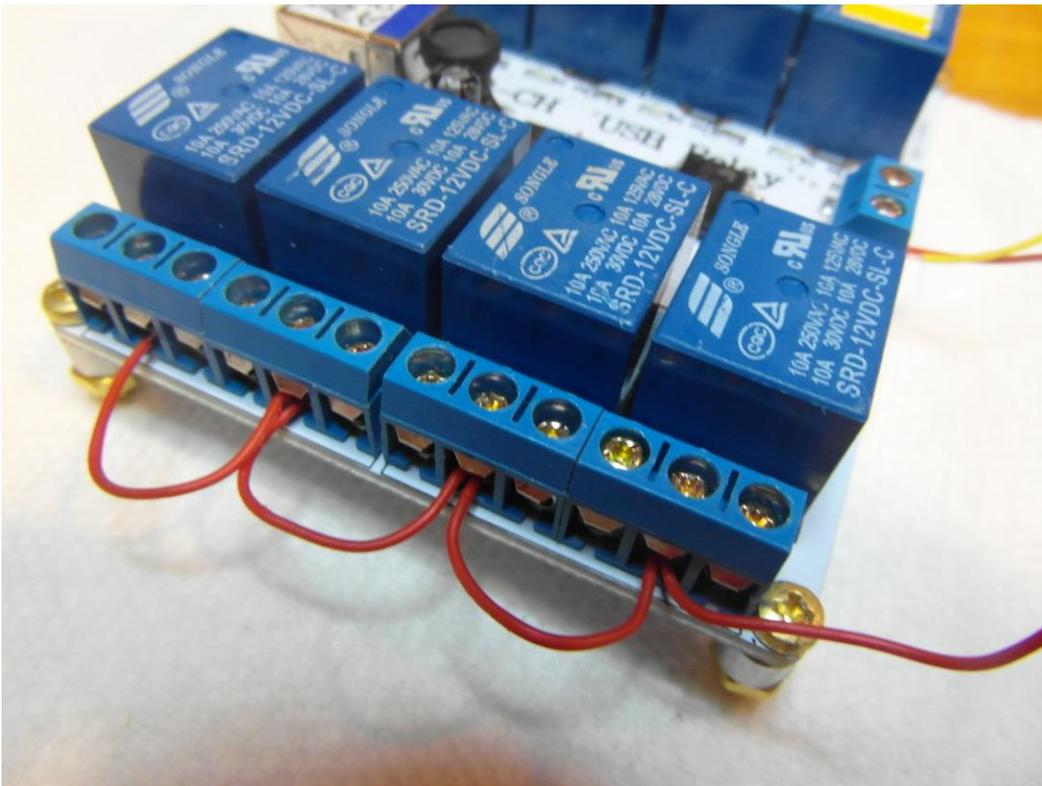
Now install an ON-OFF-ON center off two position ON, same size as the old switch



Rewiring Incoming Power to Switch Common Terminal, Antenna Switch to Power Switch UP RCS-8  
Originally, the incoming power went to one of the switch ON positions and the Common fed the circuitry.  
Bring the incoming power to the Common, and the UP position to the circuitry, the DOWN position to the USB module.



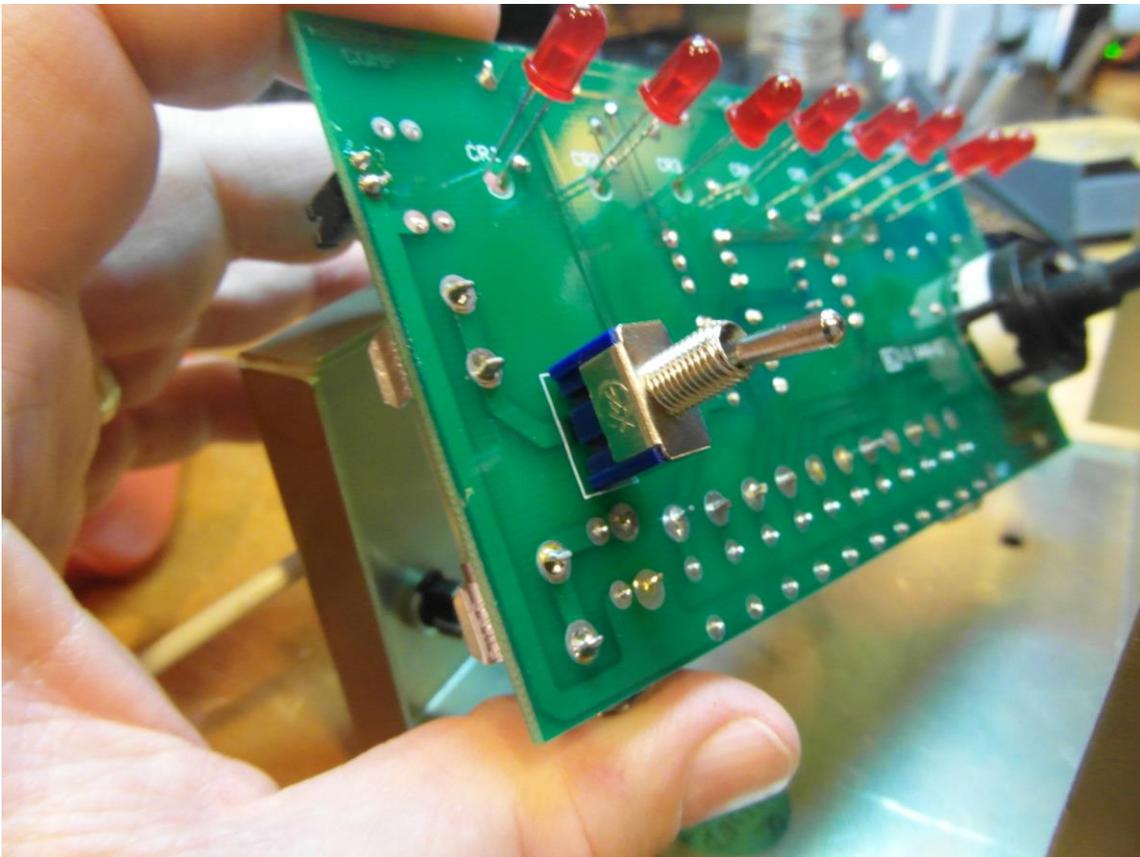
Connecting All Relay Common Terminals to +12V Vcc RCS-8  
The 12v comes from the DOWN position of the new power switch.



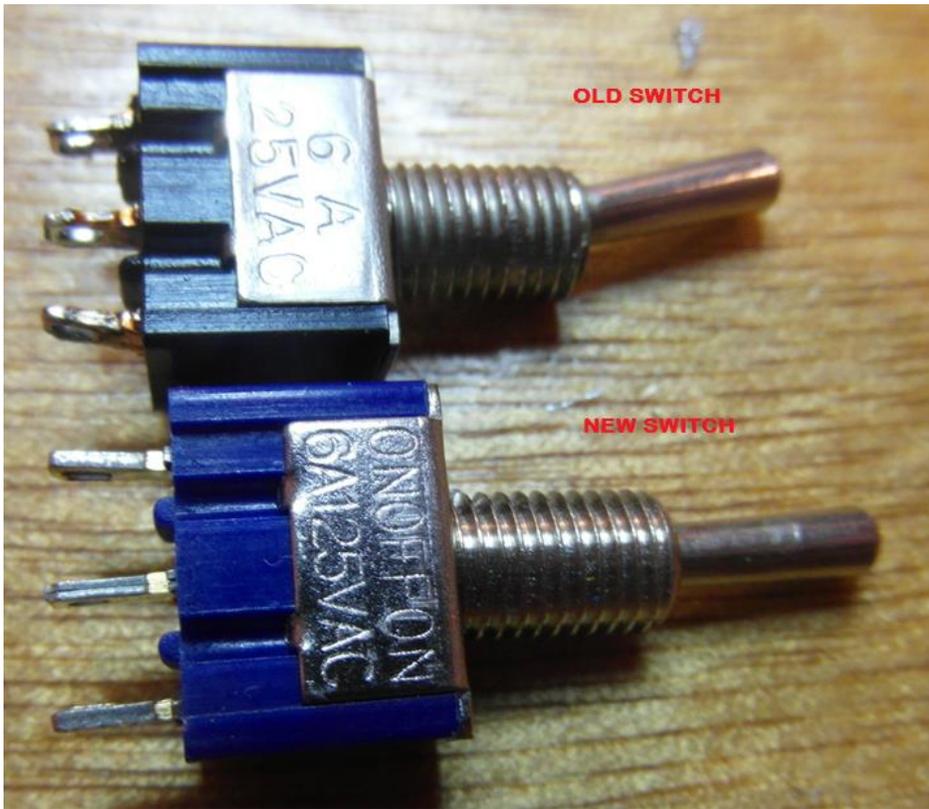
Removing the PCB from the front panel RCS-10



Replacing the Power Switch RCS-10



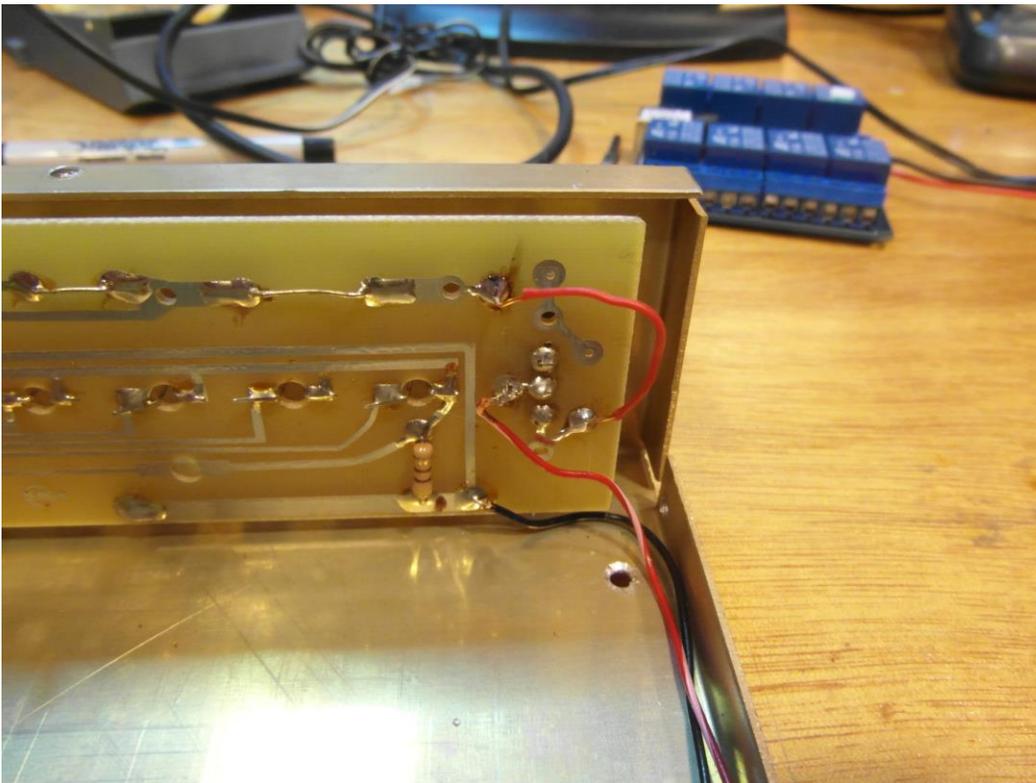
## OLD vs NEW Power Switch



[https://www.amazon.com/Amico-125V-Position-Toggle-Switch/dp/B008DFYDNE/ref=pd\\_lpo\\_263\\_bs\\_t\\_1?ie=UTF8&psc=1&refRID=0X268FBHKTQTK2EM7MBX](https://www.amazon.com/Amico-125V-Position-Toggle-Switch/dp/B008DFYDNE/ref=pd_lpo_263_bs_t_1?ie=UTF8&psc=1&refRID=0X268FBHKTQTK2EM7MBX)

## Revised Power Wiring on RCS-8

Wire incoming power to power switch common, rest of circuitry to UP position, USB module to DOWN Position.

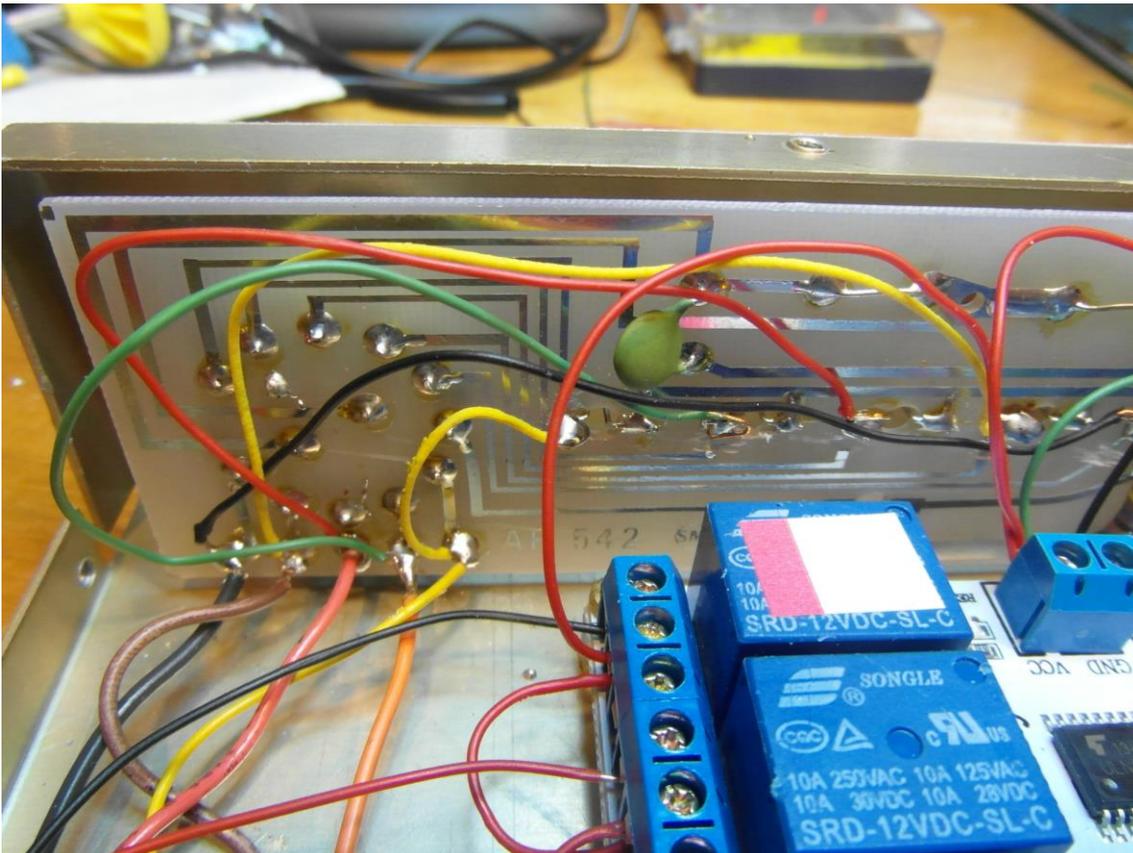


## Revised Wiring to Indicator LEDs RCS-8

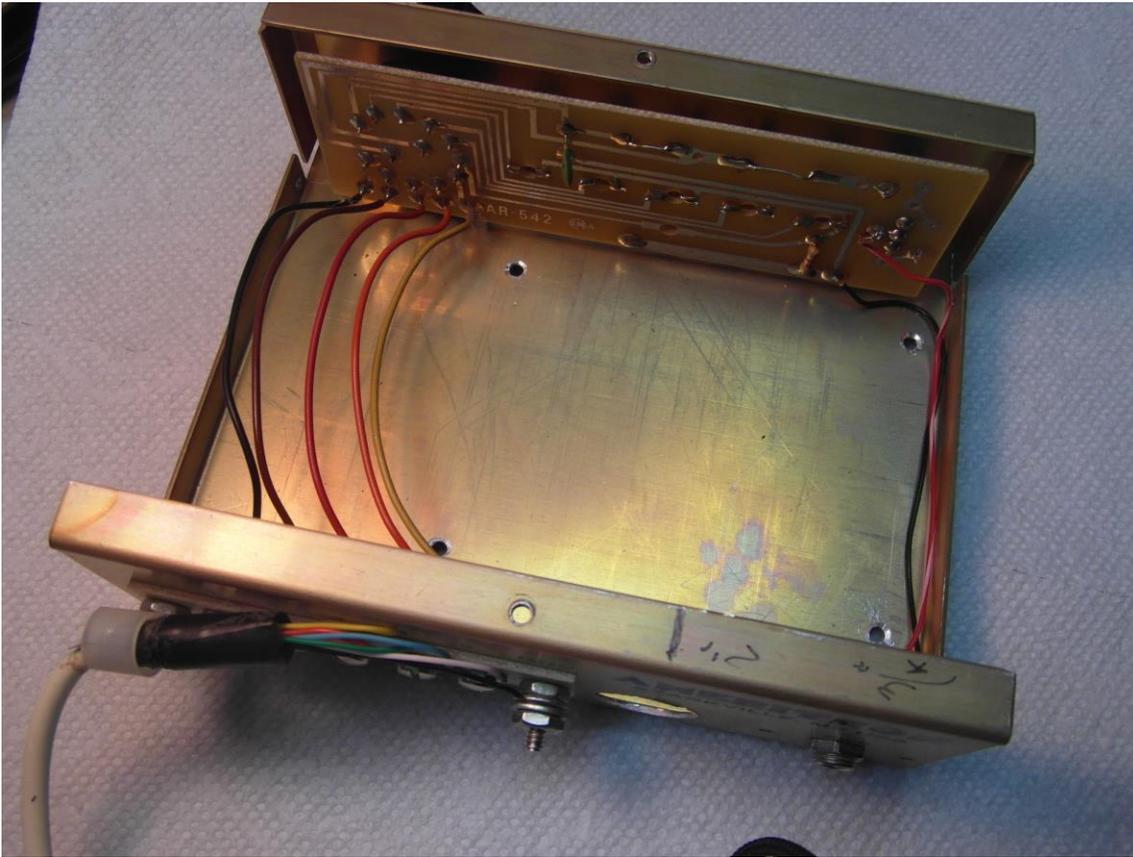
Traces Cut to rewire Indicator LEDs to Antenna Relay Control Terminals.  
This will allow the LEDs to indicate what the USB module is doing.



Wires installed to connect LEDs to Antenna Relay Control Terminals.



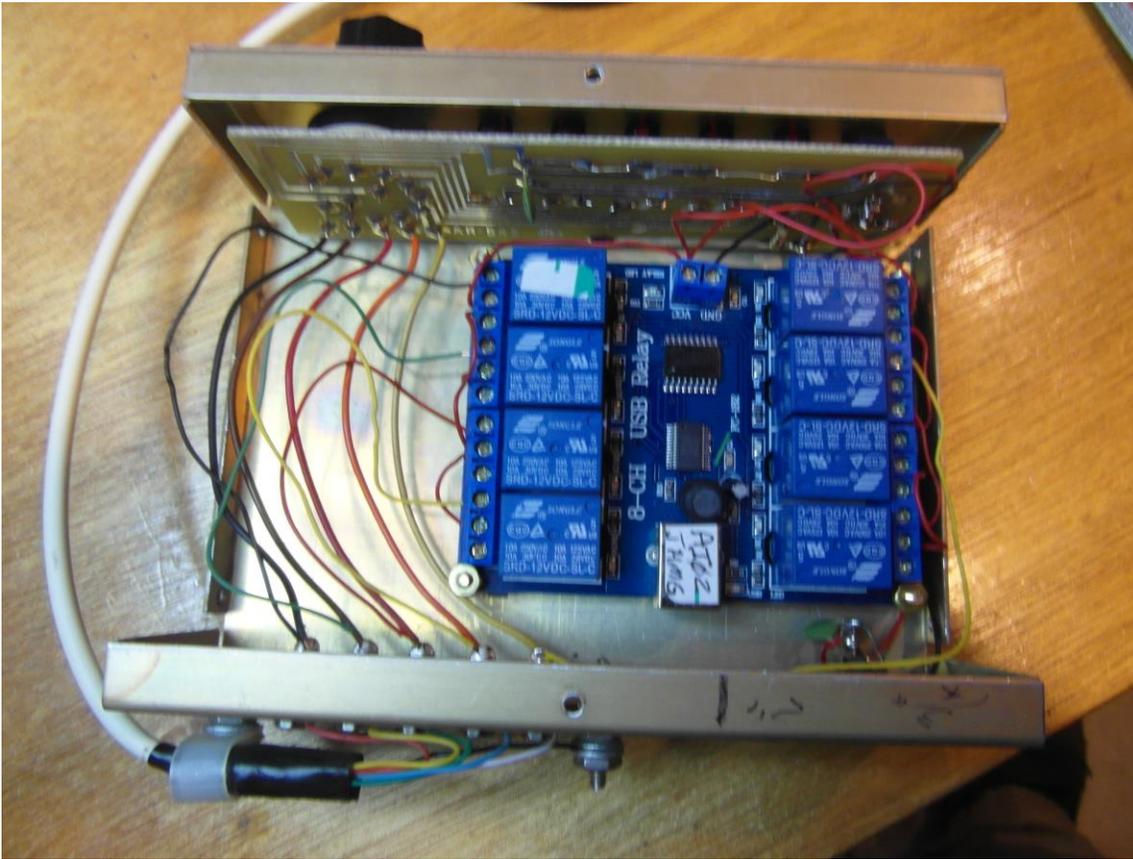
Mounting Holes for USB Relay Control Board in RCS-8



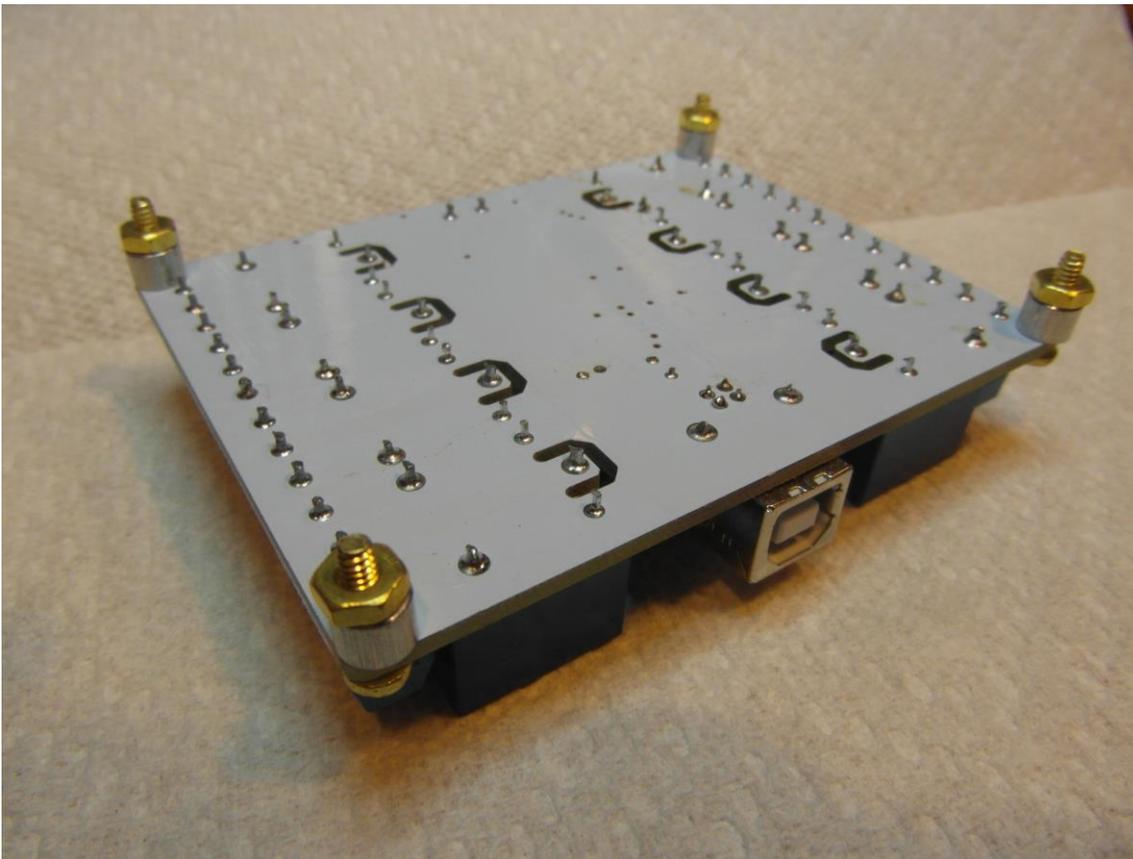
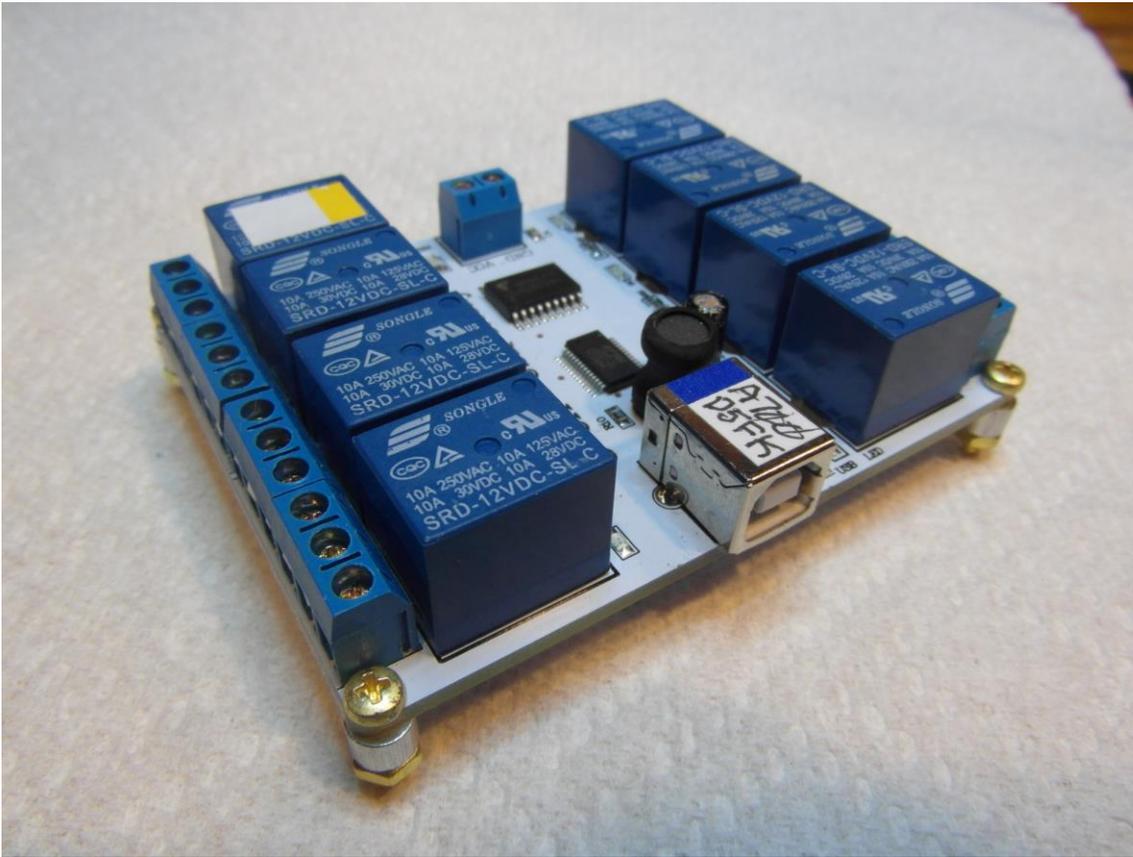
Hole Punched for USB Connector in RCS-8



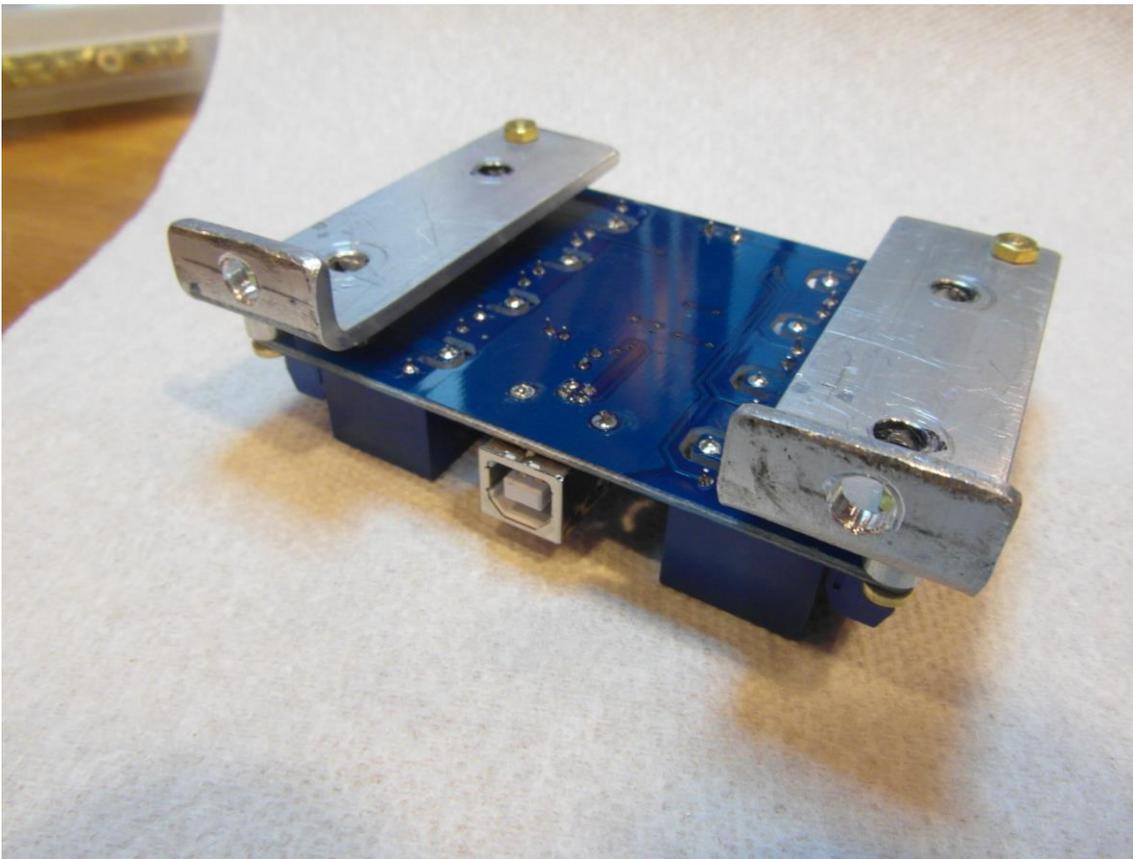
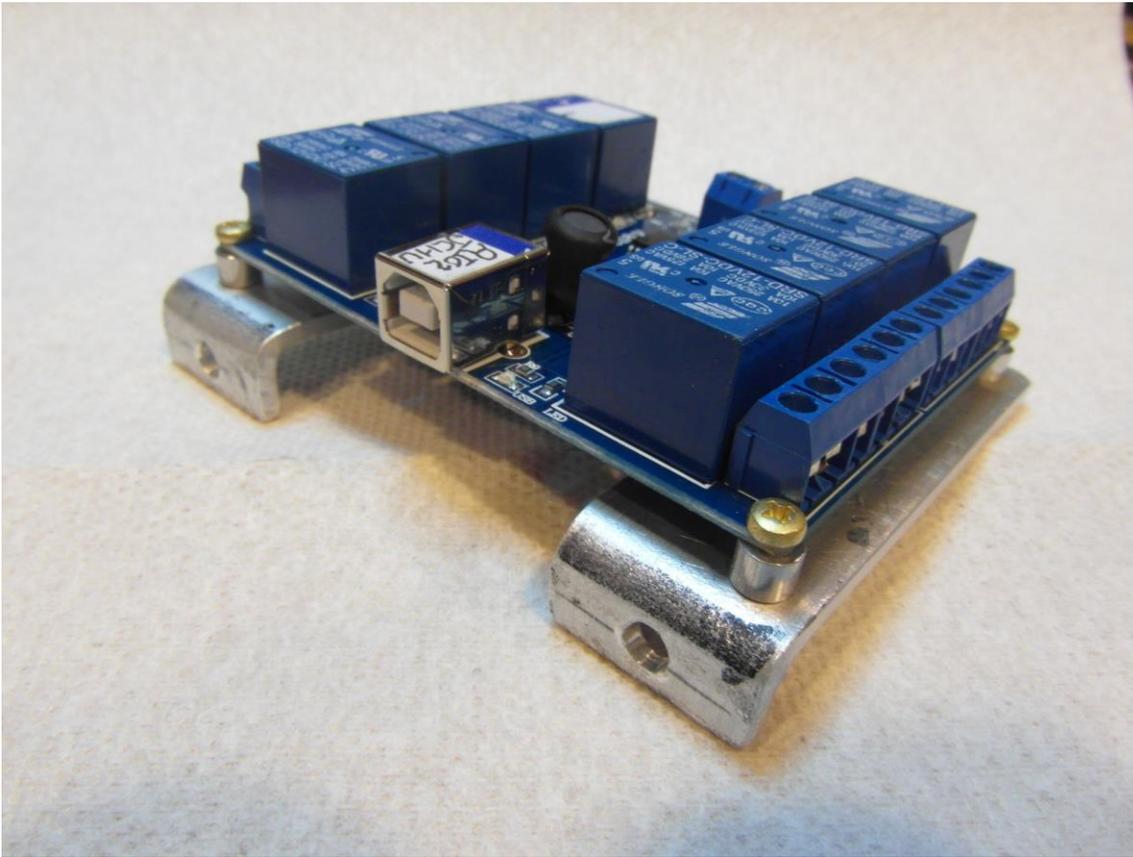
USB Relay Board Installed in RCS-8



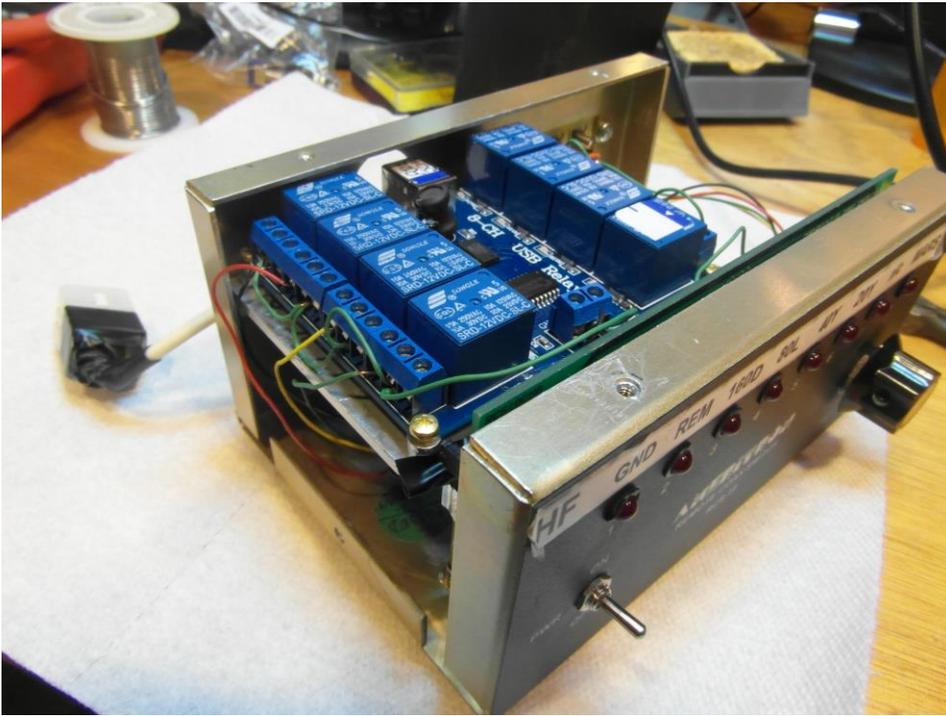
Mounting Bracket for USB Relay Board in RCS-8



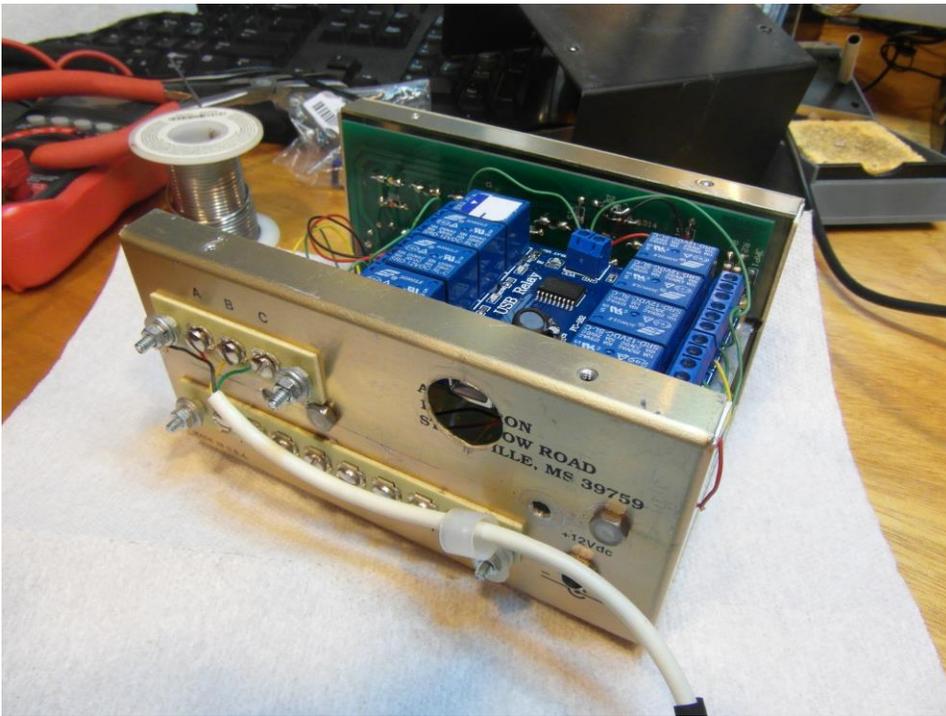
Mounting Bracket for USB Relay Board in RCS-10



# USB Relay Board Mounted in RCS-10



The bolts are attached via tapped holes in the mounting brackets.



## SOFTWARE

Details of software programming for this USB to Relay Board are beyond the scope of this document.

However some brief information is provided to get you started on the proper path.

Software **has** been developed by the author.

The client and server software is almost available now, it is in Beta status, awaiting documentation updates.

Source and executables for a test application that uses the FTDI FTD2XX.dll to control USB Relay Modules **is** available.

### Software Design Approaches:

There are two viable approaches for software programming:

- 1) Treat the USB Relay Board as a Standard COM port  
Open(ComXX), output 0xFF 8 bit data to control the relays, Close()
- 2) Use FTDI FTD2DXX\_NET.dll  
GetDeviceList(), Open(SerialNr), SetBitMask(0xFF, 0x04), Write(0xFF), Close()

### Test Programs

Source and executables for a test application that uses the FTDI FTD2XX.dll to control USB Relay Modules is at this URL:

<http://www.glagowski.org/radio/Software.htm#HFAUTO>

The item of interest is at the very bottom of list.

There are some test programs available that can be used to check your hardware installation.

[http://k5fr.com/DDUtilV3wiki/index.php?title=USB-8\\_Relay\\_Board](http://k5fr.com/DDUtilV3wiki/index.php?title=USB-8_Relay_Board)

<http://denkovi.com/usb-8-relay-manager>

<http://denkovi.com/drm-software>

<http://usb-8-relay-manager1.software.informer.com/>

## Standard COM Port Approach

Use the Windows Device Manager to set the COM port to an acceptable value that doesn't conflict with something else.

Use a terminal program such as HyperTerm, PuTTY, or RealTerm to connect to the COM port.

Use 8 bit byte, no parity, 1 stop bit, baud rate does not seem to matter.

Output 8 bit bytes, each bit position is assigned to control the relay associated with that bit.

Consult the ASCII to HEX or Binary table for what bits are ON / OFF for a given character on the keyboard.

For software programming, use standard serial port classes in the Windows .NET or Win32 API.

### Programming Logic:

- 1) Open COM Port
- 2) Output Data
- 3) Close COM Port

### References:

<https://www.youtube.com/watch?v=SKw3iZdFOo0> YouTube Video

This shows how you can connect HyperTerm or other terminal emulator program such as PuTTY or RealTerm to the COM port for the device and enter data.

## [FTDI FTD2XX\\_NET.dll Approach](#)

Using this DLL, we expect to see an FTDI-245 chip not FTDI-232 chip.

The chip on the relay board is accessed by Serial Number, not COM port NR as when you access it as a serial device.

Use Device Manager to assign the COM ports to acceptable numbers not in conflict with anything else.

Have a copy of the FTDI FTD2XX\_NET.dll in the Visual Studio project, and include it as a reference.

References:

The FTD2XX\_NET DLL:

[http://en.pudn.com/downloads454/ebook/detail1912142\\_en.html](http://en.pudn.com/downloads454/ebook/detail1912142_en.html)

WEB Search: FTDI FTD2XX\_NET.dll

<http://www.ftdichip.com/Support/SoftwareExamples/CodeExamples/CSharp.htm>

<http://www.ftdichip.com/Support/Documents/ProgramGuides.htm>

[http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX\\_Programmer's\\_Guide\(FT\\_000071\).pdf](http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX_Programmer's_Guide(FT_000071).pdf)

<http://www.codeproject.com/Articles/37982/Interact-with-FTDI-chip>

<http://stackoverflow.com/questions/18900862/using-ftd2xx-net-managed-net-wrapper-class-to-read-real-time-temperature-data>

Programming Logic:

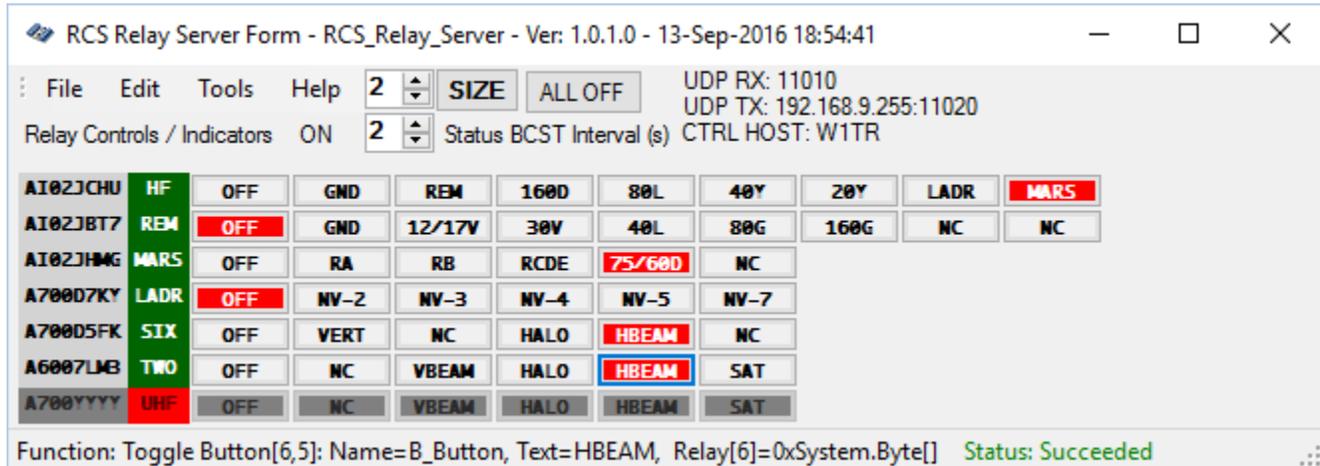
- 1) Use `GetDeviceList()` to get the list of available devices, which will include COM port and serial number information.
- 2) Create an FTDI object for each relay board using the Device List
- 3) `OpenBySerialNumber(SerialNr)` for each FTDI object in the Device List as needed.
- 4) `SetBitMode(0xff, 0x04)` for each FTDI object.
- 5) `Write(0xXX)` an 8 bit byte value for each FTDI object as required for your program logic.
- 6) `Close()` each FTDI object when program is about to exit.

## W1TR RCS-8 / RCS-10 ANTENNA SWITCH SOFTWARE

The software used at W1TR / AFA1DI to control Ameritron RCS-8 and RCS-10 control heads with USB Relay Modules is built in two parts, communications is via Ethernet UDP packets:

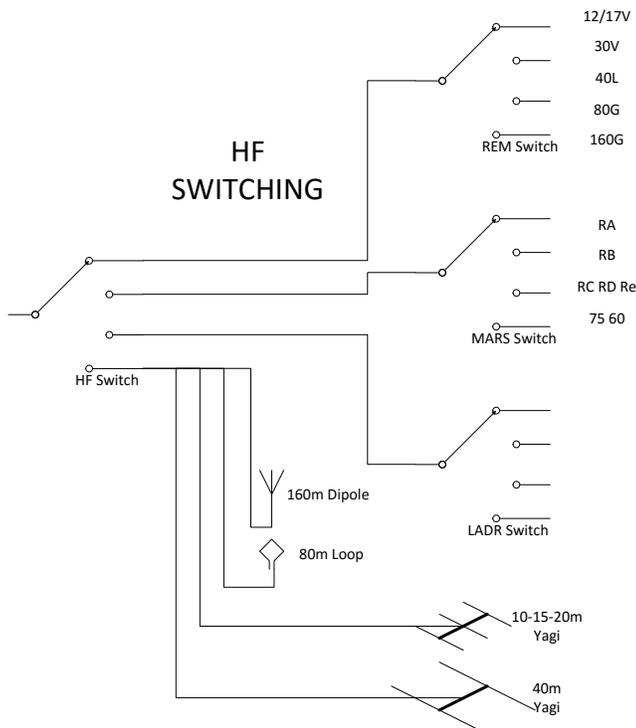
- 1) A USB Relay Server program that resides on the computer connected by USB to the Relay Control Boards in the RCS 8/10 Control Heads.
- 2) A Client program that can be on the same computer as above, or a different computer on the LAN (or even WAN) so multiple operating positions / computers can be used to switch the antennas.

### GUI for USB Relay Server Program

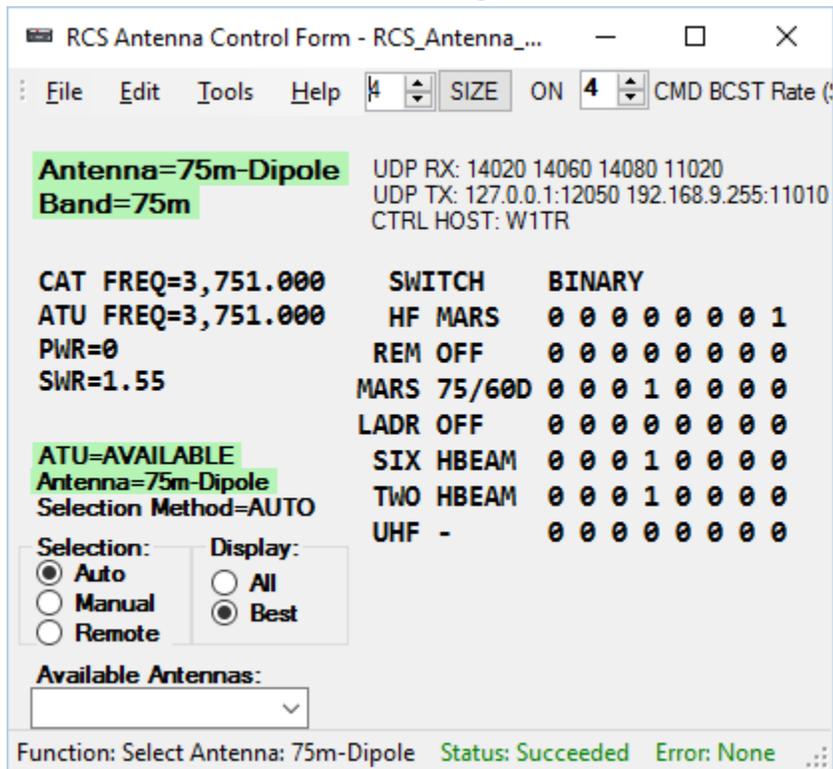


To understand this GUI, one needs to understand that the radio station is connected to the HF switch (RCS-10) Three other switches are connected in cascade, MARS (RCS-8) and REM (RCS-10), and LADR (custom relays for ladder line) are connected to ports on the HF switch.

An HF antenna is selected by giving combinations of switch settings for HF, MARS, REM, LADR as required. SIX and TWO are not connected to the HF switch but directly to the station.



## GUI for Antenna Switch Client Program



## CONCLUSION

For some time now, it has been my pipe dream to have total station automation so that changing the frequency on the transceiver will control the amplifier, antenna tuner (if needed), and the antenna switches. The ATU control and Amplifier control components of the automated station were developed during the last year and the antenna switch component was completed during the summer of 2016. It is very satisfying to have one second QSY capability, for DX chasing, Contesting, NTS Traffic Net Operations, or MARS operations. It was also educational and a great creative outlet to build this system.