**A PICAXE-based Antenna Switch**
for Icom transceivers

Essentially an automatic 'antenna versus band' switching concept...

( - or - No more manual antenna switching )

5 July 2010

*This is a photo of my completed unit. The background to it's development is detailed below.*
*This is NOT a kit. You can't buy one of these anywhere in the world.*
*In short, you have to construct it - but I have already done the hard work - the schematic, PCB layout, the software...*

It was made as a "companion unit" to the Icom IC-706Mk2G and is the same width within a few millimetres, is a very similar height but only just over half the depth.
(mouse over for a larger image)

*Why would you want - or need - one of these ??*

*Quite simply if you have multiple antennas and a multi-band Icom radio like a IC-706Mk2G (or similar) then remembering which antenna has to connected or switched to for any given band becomes a serious task.*
*The result if you transmit into the 'wrong' antenna could be PA transistor damage, maybe, at worst, but still possible.*
*At best you don't work anybody much because the selected antenna is not tuned and radiating !*
*This becomes even more of a problem if you are only an 'occasional' operator and use the radio gear only once a month - or every couple of months. (Which antenna is which ?)*

*My desire for one was for want of ease of operating in HF & VHF/UHF Field Days - where I do use multiple antennas, sometimes 2 for just one VHF or UHF band, and I wanted it to be quick and 'memory-free'.*
*Things do get busy at times during one of these style of events and the less you have to remember when you change bands, the better !*

*To achieve that outcome, I designed the electronics, made up the hardware and wrote control software for a PICAXE chip that automatically reads info from the Icom CIV data stream.*
*It switches relays as the frequency band changes, and in some cases where FM is selected instead of USB or CW.*
*In my configuration, that gives me 3 coaxial output connectors for HF antennas plus 2 for 50 MHz, 2 for 144 MHz and 2 for 430 MHz - and the switching happens automatically.*
*If your desire is for just HF ( & maybe 6m ) then the code can be altered slightly to give a choice of 5 possible ports, and 3 relays are omitted during construction..*

*My software is set up for use with the IC-706Mk2G at 9600 baud but only the CIV address 'byte' value requires changing to suit any other CIV-equipped radio.*
*It co-exists with other PC rig-control & logging applications like VKCL or HRD too.*

*For more information, read on.*

---

PICAXES : http://www.rev-ed.co.uk/picaxe/ (http://www.rev-ed.co.uk/picaxe/) In brief, a PICAXE is a PIC microcontroller preloaded with a variant of BASIC, is easy to program and does not require a dedicated programmer.

Those of you who have visited my other web site pages will have noted that I try to participate in the various Field Days (FDs) held here in Australia, particularly the VHF/UHF ones held 4 times a year.

I have been taking my Icom IC-7400 along on these events to use on HF, 6m SSB & FM, 2m SSB & FM, plus an IC-718 plus transverter for 70cm SSB (432.15 MHz) and a Yaesu VX7R for 70cm FM (439 MHz). In the field, these have been set up with a few different antennas depending on what bands and modes are in use. I recently bought an Icom IC-706MkIIG ( or IC-706MK2G to others) to replace the older IC-706 (Mk1) in the 4WD and it got me to thinking that with a little effort, maybe I could centre my FD operations around it instead - at least for the bands up to & including 430MHz. A little less power on 2m SSB & FM but a lot less aggravation in setting up, lower overall DC power consumption, easier logging with VKCL sampling the '706 etc.

That change of equipment would need the inclusion of an automatic antenna switching unit so that I wouldn't lose track of which antenna was in use at any one time during the 'heat of battle' - the RST/serial number and/or Grid Square exchange ! I could also use it in my shack at home if required - but my greatest need is probably in the field.

My initial thoughts were to 'sample' a few of the control voltages in the '706 and use those to select a relay to put an appropriate antenna on-line for the current band and mode. I traced the '706 schematic around and found points that were active on FM and not on any other mode, points that were active on 2m or 70cm as well, and on HF by sampling the BAND voltage on the accessory pin. I figured that these could be run to a small multipin socket fitted into the panel blank hole on the back of the 706Mk2G. Those sample points could be fed into a PICAXE to do the decoding of mode and band and then it would control the various relays to switch the antenna ports around. However it didn't take much realisation that this method would involve a fair amount of 'customisation' of the radio and was not really ideal.

The relay output control by the PICAXE idea was still ok but there had to be a better way to handle finding out which band and mode were in use. Of course it was staring me in the face : the Icom C-IV (or CIV ) bus has all of these details inherently available - and here I was planning to use a PICAXE to do some of the work, so why not all of the work ??? I figured that I wanted the maximum flexibility in the switching of antennas so came up with a list of what possible permutations I might need :

430 MHz : 2 ports, one for 432.15 SSB etc for horizontal polarisation, one for 439 FM for vertical polarisation.
144 MHz : 2 ports, one for 144.15 SSB etc for horizontal polarisation, one for 146.5 FM for vertical polarisation.
50 MHz : 2 ports, one for 50.15 SSB etc for horizontal polarisation, one for 52.525 FM for vertical polarisation.
HF : Up to 3 or 4 possible ports, preferably with some selection switching to decide which ports would be active on which bands.

That meant up to 9 or 10 output ports with the 2 input ports ( the HF/6m port and the 2m/70cm port) from the '706. I could get away with using around 7 relays by making the 2 ports on 6m, 2m & 70cm just a changeover and 3 ports on HF just 2 changeovers, with changeovers in between. The PICAXE coding would have to do the hard work in selecting the correct relay - or relays - depending on the band and mode currently in use..

I also wanted to be able to use just one antenna on each of the 6m, 2m & 70cm bands as well so that would mean that I had to be able to input that requirement and thus disable the "FM" sensing on a band-by-band basis. I would also want to be able to use just one HF antenna too, or any number up to the allowable 3, again on a band-by-band basis depending on how many HF antennas were available at the time and what frequencies each covered. Those requirements were something that would come later - but had to be kept in mind.

The next step was to find out if the decoding of the CIV was feasible with a PICAXE. I still had an old PICAXE-28A plus a 4MHz resonator plugged into a breadboard so that would do as a starting point. The CIV signal was taken directly to an input pin ( since both are TTL-compatible or 0 to +5V ), the serial in/serial out port taken to a PC serial port, and some serial-style sampling code written into the PICAXE Programming Editor (http://www.rev-ed.co.uk/picaxe/progedit.htm) software and sunsequently to the 28A chip itself. The output ports were taken to +5V via a LED and series 390 ohm resistor as status indicators. By using the DEBUG feature in the Editor, I was able to see some of the data stream from the CIV - couldn't make sense of it - but could see it was happening. There was hope there.

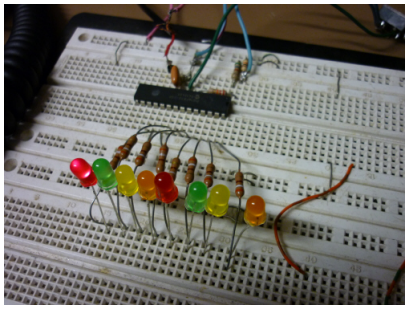
The 28A is an 'old-series' device and has been replaced by much faster and 'larger' ones - like the 28X1 and 28X2, or the 40X series - but if I could get the coding close then I would consider buying in a couple of these newer devices. I dropped the '706 CIV baud rate to 300 ( & the PICAXE I/O too of course ) and it started to make a bit more sense. The data pattern was consistent depending on whether I rotated the dial or pushed the radio's Mode button to select SSB, FM etc. Now I just had to actually decode it. It didn't take long to realise that I had initialised the serial

input function to 'inverted' mode rather than 'true' mode and by changing that, it actually made more sense on the DEBUG screen. A bit of BCD decoding started to show me frequency data - although in reverse order. A change to the PICAXE code and it was actually outputting things like 01 44 55 50 to screen when the dial was on 144.5550 MHz. I found the significant byte for FM contained a '5' ( other modes have values of 0, 1, 2, 4 etc) so started to look for that in the relevant data stream. A few more changes to the code and it was lighting up the LEDS on the output ports as I changed bands from 1.8 to 430 and as I pushed the Mode button successively. I BCD-coded 4 output port pins for the 9 HF bands, set one each for 50MHz, 144MHz and 430 MHz and the final output port pin (of 8) as active for FM.

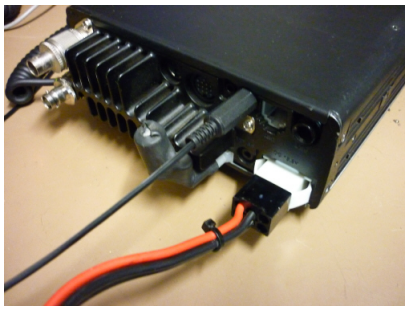
I tried the same arrangement at 1200 baud CIV data rate, with the PICAXE programmed to match - and met with failure. It was going to be a case of a faster PICAXE so that I could program it for decoding higher baud rates - or just leave the radio set to 300 baud. For interest, I unplugged the 706Mk2G and plugged in the older 706Mk1, spun the dial and pushed the Mode button and it all worked the same except that the 430MHz port LED did not light - and rightly so since this model doesn't have 430 MHz fitted ! All of the other band decoding and FM operation LEDs lit correctly - with no coding changes in the PICAXE. ( I don't look closely at the CIV device address in my current coding, just frequency and mode. )

The time expended to this point from when I first found the breadboard with the 28A and using the programming editor : about 4 to 5 working hours.
*By the way, the Australian agency for PICAXE devices is Microzed : http://www.microzed.com.au/ (http://www.microzed.com.au/)*

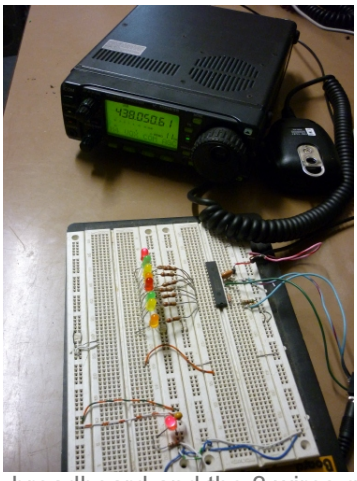
*Mouse over the images for greater detail.*



This is the breadboard with the PICAXE-28A plus the few resistors and LEDs used for the initial development phase



The input is derived from the CIV port on the back of the IC-706 (top black 3.5mm mono plug)



This is the basic test setup. The radio, the breadboard and the 3 wires going off to the serial port of the computer.

Different frequency bands - different LEDs light up. The amber one closest to the camera is used to indicate FM or non-FM.

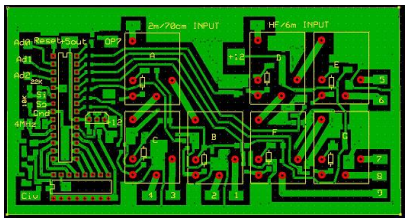
My plan is to eventually include an auto-ATU on the HF port between the radio itself and the antenna switch HF input port. It can make up for any impedance irregularities in the antenna switch plus anything from the HF and/or 6m antenna(s). I don't have an auto version tuner at the moment and most of my antennas are relatively low SWR so this isn't an imperative - but FD operations don't always have optimum installations so this should be kept in mind if you contemplate following my example with this project.

The next step is to create a PCB layout - and that means selecting a final PICAXE version, selecting suitable relays that can be fitted to the PCB, working out the remainder of the circuity and then optimising it all for good operation at radio frequencies. The PICAXE-to-relay interfacing has to be included since the PICAXEs run off +5V and the relays at +12V, the selection switching for FM-inhibit per VHF/UHF band and for the variation in selected HF antennas now has to be finalised.

The relays I decided to use for HF / 6m ( and even 2m/70cm for a start ) are a Goodsky UDH-SS-112D, 12V single changeover model with contacts rated at 5A at 12V, size about 21mm x 17mm and 15mm high - available from Altronics under code S4202A or S4202B for around $3-$4 each, but there are probably others with essentially the same characteristics. These were picked based on price, availability and contact rating although others are probably just as readily available. I needed to make a choice to start the PCB layout - so that was the starting point. I know that they are not RF-style relays nor are they designed for 50 ohm work, however they are relatively small and should have relatively low inductance in the leads/contacts. The other thing is that the contacts will either be open or closed only on a band-by-band basis and will not chatter back and forth from transmit to receive. I know that these relays are not really suitable at UHF and once I have tested the relays in circuit on the PCB layout for loss at 2m & 70cm, I could always build up a smaller add-on PCB with more-suitable relays and just connect them with wires attached to the same relay drive points (ie +12 and the switching transistor collectors).

I chose to keep going with the 28-series PICAXEs and have ordered a couple of the 28X1 chips for future development of the project. The smaller components used in the PCB layout are all SMD in either 0805 or 1206 sizes depending on function and SOT23 transistors for the interfacing. My plan is to mount the 78L05 regulator for the PICAXE power supply, the relays and the PICAXE (in a socket) on the 'top' side of the board, the SMD parts on the other - but then actually mount it upside down so these main parts are on the bottom and the SMD parts and connections are exposed for easy termination. The DIL switch will also be socketed but on the new 'top-side' of the PCB. Using a socket here also allows the use of a header plus wiring to external switches (eg toggles) on the mounting case in lieu of the on-PCB switch.

The PCB will be made from double-sided PCB and the second side will form an earth plane, being attached to the 'earthy' parts of the main layout side with periodically placed wire links though the board (i.e. wired vias). The whole assembly will be probably be mounted upside down into a diecast box with the required number of BNC connectors along the sides. Yes, BNC - they are quick and easy to connect/disconnect and are far better than the so-called UHF/PL259 styles at 430MHz !




This was the layout in ExpressPCB, taking a bit over 4-5 hours. There has already been some 'tidying up' done to the first layout to create this last image. NOTE THAT THIS IS NOT THE FINAL VERSION LAYOUT.

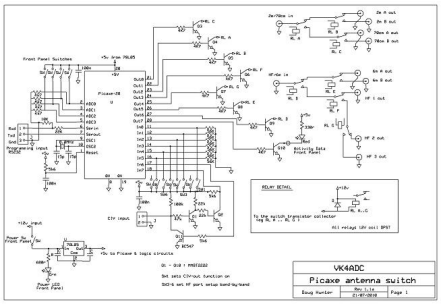
The full size layout is 126mm x 66mm, with much of that size determined by the relay packaging.

Layout : PICAXE 28 on the left, a 8 way DIL switch at bottom left, relays taking up much of the centre and right hand side.

A word of warning : this layout is designed for use with the likes of the IC-706 series radios - so the PCB layout and relays have been set up to cope with the maximum power from it - ie. 100W. It will not survive higher RF powers.

I did contemplate using the Omron G6Z '50 ohm' relays for the 2m/70cm work but these are rated only for 10 watts with a maximum SWR of 1.2:1. The '706 series outputs about 50w on 2m so it looked a bit 'fragile' if I was to use the G6Z's.

Post Note : I have done a daughterboard layout to suit these relays pending final loss measurements on the existing relay set. See further down the page.




This is the latest update of the schematic of the PCB ( now V1.1a) at left and now includes the external connections to the front panel toggle switches plus the LEDS.

The PCB actually has places left for the addition of RF bypass capacitors at various places and these are not included in the above schematic.

The places where the input & output signal, voltage and RF connections take place are actually square or rectangular pads on the PCB layer.

The layout also has 0 ohm SMD resistor links included and these are not shown on the schematic either.

Resistor and capacitor values are now listed on the schematic.

A full scale image of the schematic is available here (/~vk4adc/web/images/UserFiles/Image/antswitch/antswitch 2 a.JPG)
(right-click to download/save the 300K JPG, left-click to open in a new window)

This layout/schematic will accept the PICAXE 28A, 28X1 or 28X2 parts. The main difference between them will be a small change in the port configuration during the chip initialisation phase in the program - i.e. software only - no hardware...

These chips all require an external resonator to be fitted but the layout will also accept a HC-49U style crystal and SMD capacitors in lieu. That means that the clock rate can be set higher and thus the project's ability to decode higher CIV baud rates. Typical resonator/crystal values are 4, 8 or 16 MHz. Needless to say, the final software serial port routines have to reflect the value of the resonator fitted.

The memory limit of the 28A was reached during the experimentation phase on the breadboard and the final software version will require the larger X1 or X2 chip to contain the code segment to decode the DIL configuration switch settings. Given that, plus the 28A has been superseded, I do not recommend using one of these for the project.

In the event that no power is applied to the PCB - or no CIV input is supplied, the 2m/70cm port input will default to output #1 and the HF/6m input will default to output #5. Outputs 2, 3, 4, 6, 7, 8 & 9 will only be active with both CIV decoding and +12V power applied. For that reason, ports 1 & 5 should ALWAYS be used/terminated.

Transistors Q1 & Q2 are the CIV line input switches to isolate the PICAXE from any harmful external voltages. The two stages are used so that the signalling sense at the PICAXE input In7/Hserin remains unaltered (i.e. high = steady state, pulsing low with data).

Q3 is an inverter stage that allows the PICAXE to send CIV-formatted packets back onto the CIV bus. This could be isolated from the CIV line and both signal points taken to a RS232 converter chip (eg MAX232) for the likes of using the concept on a Yaesu CAT system

The relay contacts are arranged so that the switching will follow the pattern below :

| *Band* | *Mode* | *Input Port* | *Relays Activated* | *Output Port* | *Standard Output Port Selection* |
|---|---|---|---|---|---|
| 432 | Any | 2m/70cm | None | 1 | Standard 70cm output port ** ( default 2m/70cm port eg standard dual-band whip) |
| 439 | FM | 2m/70cm | B | 2 | Only if switched to enable alternate 70cm port, input on AD0 |
| 144 | Any | 2m/70cm | A | 3 | Standard 2m output port, enabled only by AD1 input |
| 146 | FM | 2m/70cm | A, C | 4 | Only if switched to enable alternate 2m port, input on AD2 gated with AD1 separate 2m enabling pin |
| 50 | Any | HF/6m | None | 5 | Standard 6m output port ** (default HF/6m port) |
| 52 | FM | HF/6m | E | 6 | Only if switched to enable alternate 6m port, input on AD3 from the 6m FM enabling switch |
| HF 1 | Any | HF/6m | D | 7 | As determined by logic & DIP switch settings |
| HF 2 | Any | HF/6m | D, F | 8 | As determined by logic & DIP switch settings |
| HF 3 | Any | HF/6m | D, F, G | 9 | As determined by logic & DIP switch settings |

The 'Output Port' number is the pad number on the PCB layout.
'Any' mode is anything except FM.

The outputs marked ** are the default - or no CIV/power - port connections so should always have suitable antennas/loads connected to them...

Outputs 2, 3 & 4 are a special case and have to be enabled depending on whether a single dual band 2m/70cm antenna is available or just two single band antennas or up to 4 discrete antennas. PICAXE input AD1 sets whether a separate 2m port is required at all, inputs AD0 and AD2 set whether the separate ports are wanted independently on either 70cm or 2m for FM use.

The HF1, HF2 and HF3 port selection will be as decided by the logic attached to the DIP switch settings. That would make it possible to set up (for example) so that all bands used just output 7 for a single all-band HF antenna, or can be set to split the antenna usage so that a specific band was on (for example) port 9 (eg 160m), a couple on port 8 (eg 40m/20m dual band dipole), and the balance (eg 10, 15 & 80m ) on port 7. At this stage, very little of my operations are on 10, 18 & 24 MHz so won't be prime candidates for separate switching in the logic flow... By altering the DIP switch settings, it should be possible to change port 9 to be 160m/80m etc... all depending on the logic behind it all.

The logic can always be re-programmed around the switch detection states if the need arises and the desired logic is not already in place and this can be done before the FD event or as shack fixed antenna changes are implemented. The main thing to focus on is which relay(/s) has/have to be activated to switch the HF port connection across.

Now it's just a case of re-writing the PICAXE code output logic to reflect all of the above... !!!!!!!!!!!

You may have noticed that the pin 28 output on the PICAXE is marked as spare, even though it is fed to a transistor switch via a series resistor. My current thinking is to attach a LED to it and make it blink to indicate correct CIV decoding and what it decoded ! I might even code the outcome in morse 'blinks'... eg 1, 3, 7, 10, 14, 18, 21, 24, 28, 50, F50, 144, F144, 432, F432. After all it is just a matter of setting up a code and calling a common 'morse-blink' subroutine. My original idea for this pin was to use it as a CIV-output function ( just wired across to the CIV input pin) to query the CIV bus if nothing was 'seen' happening for a time - and it might still happen that way..

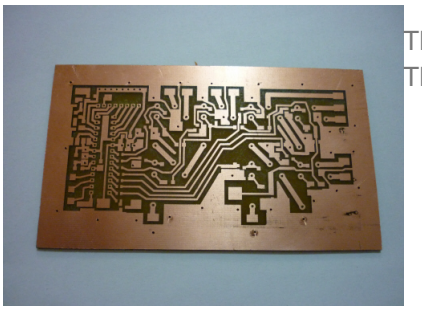
---

10 July 2010 : The PCB artwork was printed, exposed & developed. The whole of a RS Components #397-0104 photo PCB (1.6mm thick FR4 material) was used, only one side being exposed - the other side had the black plastic left on until after all the small holes were drilled.

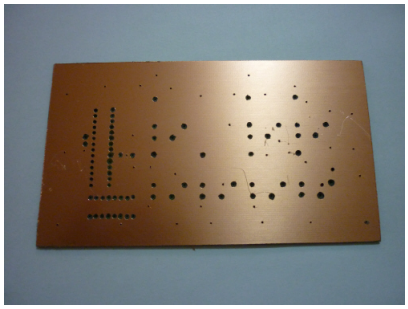
Note : ExpressPCB creates images that, when printed on clear laminate sheet/transparency, must be flipped over when creating PCBs off the 'bottom layer print'. That is you print on the 'top' of the sheet but expose from the other side (bottom/unprinted side) to get the actual layer in the true (rather than reversed) form.

---

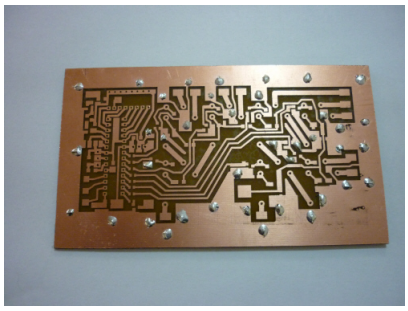
11 July 2010 : The PCB was trimmed to size, the wired 'vias' installed then the components added. The diecast box was marked out, drilled, filed & then undercoated.



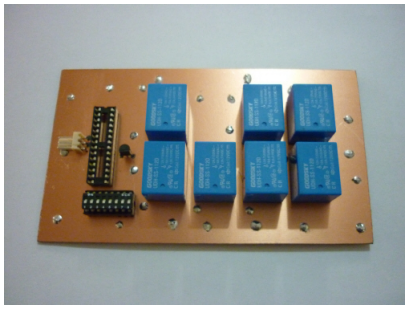
This is the main side of the PCB - basically all of the tracks are on this side. The blemish at bottom RHS was an 'oops' during the developing of the board.



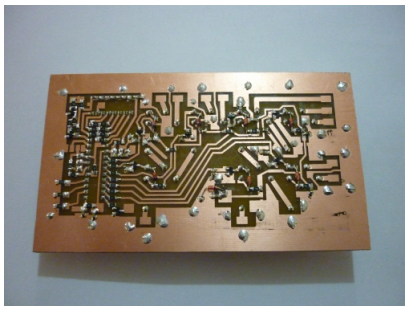
This is the top side of the PCB - as you can see there hasn't been any etching on this side & just the through-holes for the components have been 'cleared' away with a deburring tool.. There are lots of untouched "smaller" holes for the wired 'vias' to be fitted through.



The wire-through 'vias' in place on the PCB. Some are actually required for active connections, others are to make the top plane effective as a shield



The mostly-constructed top view showing the PICAXE programming input connection on the 3-pin header at left, the empty 28 pin socket awaiting a PICAXE, 78L05 regulator just to it's right, the lower 16 pin IC socket with an 8-way DIP switch installed and the 7 x 12V antenna connection switching relays.



The mostly-constructed bottom view. Only the resonator ( or crystal + 2 small capacitors ) has yet to be fitted. Most parts are SMD, the exception being the 1N4148 back-EMF diodes across the relay coils.

This is the front view of the diecast box that the above PCB will be built into. The 5 lower holes are for toggle switches and the 2 upper ones for LEDs.

The rear view shows the 11 x BNC flange-type socket mounting holes plus the power/CIV entry hole. The rectangular slot in the top/lid is for access to the internal 8-way DIP switch.

The inside view shows the 4 holes for mounting the PCB.

A bit of "spray-through" is obvious in the front RH corner.

---

12 July 2010 : The diecast box was painted with a 'Metallic Charcoal' colour from a spray can ( photos above ). The 28X1's arrived and one was inserted into this new PCB, powered up & then the 'fun' began. The 28X1 is quite significantly different to the 28A so requires changes to things like the serial comm's settings just to get it going. Quite a bit of the latter part of the day was expended in trying to get the PICAXE to reliably decode the CIV data at higher speeds. The external crystal attached to the PCB is 8.0 MHz at present though this may yet be changed for a 16.0 MHz one as the experimentation progresses. I have hopes of making the PICAXE auto-detect the CIV baud rate at the end of all this !!

---

13 July 2010 : Lots more time spent trying to improve the CIV detection process at higher speeds and improving the output relay port coding. The basic process works : the 70cm & 2m independent ports are selecting on both SSB & FM, as does the dual 6m ports. The 3 HF ports have been currently configured for so that the 1.8, 3.5 & 7 MHz is port 3 (eg for a multiband dipole/longwire etc..), 10, 18 & 24 MHz is port 2 (eg for a WARC band dipole) , and 14, 21 & 28 MHz is port 1 (eg for a triband yagi) - and they all work. I have moved the CIV data input to the "hserin" port (pin 17) { in parallel with the original pin } for now and am using the 'spare' inverted output from pin 28 as a CIV data source so that the attached radio can be queried as to mode and frequency. That all works ok at 9600 baud and slower (4800 & 1200) rates.

I have been experimenting with the background buffering of CIV-sourced input serial data to the PICAXE as against a foreground method and have yet to optimise the process. My whole attack now is to try to get the thing operating faster - the data from the radio is not always decoded the first time around ( and the critical word is always ) and I have yet to discover if it is a 'jam' signal on the CIV bus that is causing the delay in response.

---

14 July 2010 : The CIV comms is now entirely background using the Hserin & Hserout functions in the PICAXE. It took a while to 'discover' the right technique to reliably find the CIV data segment that I wanted - but I have now succeeded.

The version of the software is now at a fairly good working point so a diversion sideways. There were a couple of modifications done on the PCB that concerned cutting tracks and installing jumper wires - plus a BC547 transistor and a resistor down at the CIV input pad. After that was done, the device was tested again and found to be operating as desired. Next came the fitting of the PCB into the diecast box. All of the RF pads were terminated in RG174 miniature 50 ohm coax, a ribbon cable to the pads at pin 2, 3, 4, 5, the transistor collector fed from pin 28 and the +5V points. A +12 wire was run from the voltage regulator input pad to the realy 12V input pad then forward to the power switch. The CIV input was run in a shielded cable to go out to a 3.5mm mono plug.

My original plan was to mount the PCB upside down in the box but somehow I had managed to cut the DIL switch access hole in the loose top cover instead of the actual bottom of the box ! That meant I had to mount it right-way-up even though that made the coax runs to the BNC port connections longer. Past all that..., the PCB was mounted on metal standoffs onto the bottom of the box, all of the wires and coaxes were terminated, cross-checked then finally connected to the IC-706Mk2G and powered on. And it worked... The indicator LED was set up so that it flashed a code which represented which port was currently selected, and as I used the pre-programmed memory switch on the radio, the flashing port code followed it as the various bands were selected.

Photos to come. ( see below )

---

15 July 2010 : A quick check of the VHF/UHF port attenuation : 1dB at 144, around 7dB at 432 ? Have I got a crook relay or is it the style of relay or the PCB layout ??? To be checked as more time is available. The HF/6m values to be checked later. No too much else to do on this project.

For interest, the default HF/6m port allocations are all HF bands on port 1, 6m on either 6m port depending on ode - SSB or FM. Changing one DIP switch changes it so that it uses port 1 for 160, 80 & 40m then port 2 for 20, 15 & 10m then port 3 for the WARC bands of 10, 18 & 24 MHz. Lots of variations possible here though as many switch states can be selected.

There won't be any more work done on the project for a few days as I will be out of town. I hope to get back to it early next week as I really want to continue while my mind is still thinking PICAXEs and serial coding.

---

18 July 2010 : I took my notebook computer away with me (plus the source code developed so far) and found some time to sit and comment my source code for this project. I found that I had made an 'oops' in the source code with an 'Endif' statement including a 'Loop' command - where the 'Loop' should have been after the 'Endif'. What does that

mean ??? It caused a fault which occasionally caused the PICAXE to lock up and required a quick power-up to reset. I re-loaded the altered PICAXE code on my return today and - voila - it no longer locks up at all. I also created a different HF port table while I was away and incorporated it into the current PICAXE coding :

| Band (MHz) -> | 1.8 | 3.5 | 5 | 7 | 10 | 14 | 18 | 21 | 24 | 28 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BCD value | Port | Port | Port | Port | Port | Port | Port | Port | Port | Port | Typical use |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | all bander |
| 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | tribander |
| 2 | 1 | 1 | 1 | 1 | 3 | 2 | 3 | 2 | 3 | 2 | WARC ant + tribander |
| 3 | 3 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 160m |
| 4 | 3 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 160 + low + high |
| 5 | 3 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | all + 60m + 160m |
| 6 | 1 | 1 | 3 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 60m + tribander |
| 7 | 1 | 1 | 3 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 60m ant + WARC |
| 8 | | | | | | | | | | | |
| 9 | | | | | | | | | | | |
| 10 | | | | | | | | | | | |
| 11 | | | | | | | | | | | |
| 12 | | | | | | | | | | | |
| 13 | | | | | | | | | | | |
| 14 | | | | | | | | | | | |
| 15 | | | | | | | | | | | |

This is the latest HF switching matrix. The first column is the BCD value created from 4 of the DIP switches ( SW3 through 6 ), 0 being none selected, 15 being all 4 selected.
You may notice that only half of the possible table value rows have been set up at this time thus leaving space for future permutations. Typical usage is shown at right.
To implement it, the row number (1..15) value is used in the "Case" statement in the PICAXE program flow
( eg. see more further below :-

```
bn28: ' Select For BAND 28 MHz
Select readinputs
Case 5
Goto hfport1
Case 1,2,3,4,6,7
Goto hfport2
.....
)
```

To use a specific setup/configuration in practice, it is simply a matter of changing the DIP switch settings - no need to power off the unit - and the next time the band is changed then it will switch HF ports accordingly.

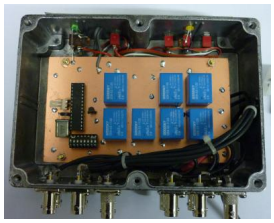
Updated the schematic above to version 1.1 to incorporate the external switch & LED connection details plus the changes to reflect the latest alterations on the CIV serial input & output functions to the PICAXE chip. The PCB layout image has not yet been altered to reflect some of these changes although the actual layout has been modified. If someone is contemplating building up a similar unit then I can make the corrected layout available.

---

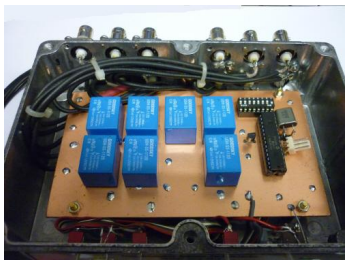
19 July 2010 : The project is all but finished. The front panel decal was created and attached. The port guide label for the top back of the box ditto. I still have to do more loss testing from port-to-port at various frequencies and, if necessary, I can lay out a small PCB that attaches to the top of the existing board ( in place of the 3 VHF/UHF relays ) to accommodate some smaller relays (eg G6Z's ) to reduce the through loss at VHF & UHF. I regard the latter possibility as fairly simple to do anyway.

Front view with the toggle switches & LEDs mounted. Inside view of rear BNC sockets.

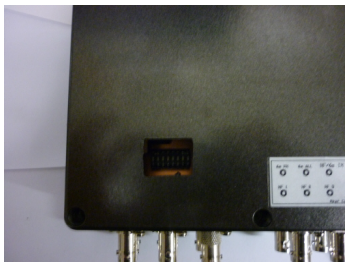
Rear view of BNC sockets etc.. A different style of flange mount socket was used for the input sockets just to make it easier.



Inside view after mounting the PCB & installing the wiring. The RG174 has been cable tied to form it into a stable wiring format.



Angled view of cabling to the rear panel.

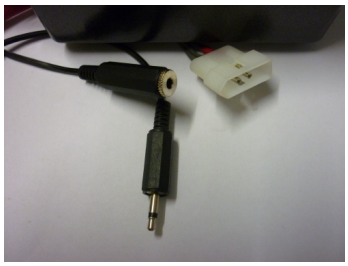


The DIP switch is accessible through the top slot. Though not visible, the slot actually has a clear mylar flap fitted to the underneath & it needs to be pushed down to access the switches.

Top view of the box with the labelling of the rear port attached -0 to make it easier to connect the coax cables up.

Front panel - this was originally supposed to be Scotchcal ( as I still have some old stock but no 3M developer left ). I had to fall back to an adhesive paper label ( using 1/4 of a A4 page) plus a clear 'cold laminate' adhesive sheet over the top to preserve the panel.

Anybody know what the chemical in the liquid developer for the Scotchcal / Dynamark 8008 metal labels was based on ???? It is no longer available and while I have tried a number of solvents, the only thing that makes any impression on the black coating layer is Acetone - but not a good enough result to use in a project panel.



The power for the unit comes from the 4 pin polarised antenna tuner socket on the back of the Icom radio, the 3.5mm plug goes into it's CIV / Data socket and the 3.5mm in-line socket allows connection of a CT-17 (or equivalent) RS232->CIV adapter & hence to the computer.

This is the view of the completed unit.

*Two weeks from contemplation to virtual conclusion.*

Front panel switching explanation :

---

LHS - VHF/UHF Single / Dual : 'Single' is used where a single dual-band antenna is used *and no other switching is required*. 'Dual' sets for separate antennas for each band. The default 2m ALL port is active in 'Single' mode.

2m All / FM : Where the LHS switch is in the 'Dual' position, this selects whether the same antenna port is used for 'ALL' *2m* modes, or a separate one just for 'FM'.

70cm All / FM : Where the LHS switch is in the 'Dual' position, this selects whether the same antenna port is used for 'ALL' *70cm* modes, or a separate one just for 'FM'.

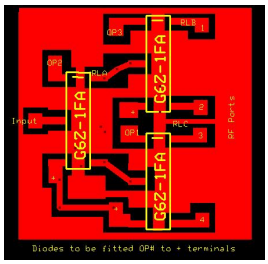
6m All / FM : This selects whether the same antenna port is used for 'ALL' *6m* modes, or a separate one just for 'FM'.

RHS - Power On/Off : self explanatory.

---

Things that MIGHT get done :

- The 9600 baud CIV rate and the Icom CIV device address details are presently hard-coded for the IC-706Mk2G. I might eventually change the startup processes so that it will auto-baud and detect the CIV address byte to automate the compatability with other radios - but that's a maybe.
- The VHF/UHF relays MIGHT get replaced by the daughterboard with lower loss relays - depends on final loss values ( Followup note - now it WILL be done ).
- Maybe re-write the code to suit the Yaesu CAT configuration as an alternative use/possibility ????




19 July 2010 :This is a PCB layout to replace the VHF/UHF relays with Omron G6Z-1FA-DC12 devices. (These particular relays are available from Minikits (http://www.minikits.com.au/) in S.A. ) The PCB will fit in the space released by removing the other 3 relays from the main PCB and soldering this PCB ( & relays ) in the space released.

{ No attempt has been made to try to make the relay coil connections on the main PCB line up with the matching connection pads on this one - they will have to be made using flexible wires, probably around the edges of the PCB. }

Now I just have to buy a few of these relays and print the PCB.... (again in 1.6mm double-sided FR4 material, one side only printed ).
There will also be SMD bypass capacitors required across all relay coil connections to the earth/ground plane, back-EMF diodes across each relay coil, plus wire links to connect front and back planes of the PCB material together.

---

21st July 2010 : I have re-measured and confirmed the losses in the relay contact arrangement.

At HF & 6m, it is not measurable because it is so low - for all intents and purposes it is 0dB.
At 2m, again it is hard to measure because it is less than 1dB.
At 70cm, it is about 7dB. This is too high to be acceptable.
I have ordered a few Omron G6Z relays and will print off the above circuit board and install it as soon as the relays actually arrive.

I also noted a bit of relay 'chatter' when putting 70cm SSB into it. I suspect that it is because of the loss (and therefore radiation) of several watts of RF in the contact wiring/arrangement producing high RF levels around the 78L05 voltage regulator (VR). The +5v from the regulator is varying with RF drive & the PICAXE output port voltages are dropping as a result. The changeover to the daughterboard ( with minimal losses) and the fitting of a shielding piece of PCB or tinplate between the relay segment and the VR on the PCB should solve the problem. I cannot duplicate the effect at 2m, 6m or HF even with 100w of RF into it.

---

22 July 2010 : Just a few code snippets added to this web page to show part of how the PICAXE 28X1 source software is set up. The actual full source code is just under 700 lines long, including some comment lines, at this stage.

An alternate CIV address :
civaddr = $58 ' preset CIV address for 706mk2g in hex format, change as necessary for different model

---

The decoded frequency selection breakdown :

' break down frequency selection bands into blocks by MHz defines - NO GAPS
' each Case will jump out to the output subroutine & return via it

' variable *freqz* contains the thousand, hundreds, tens and units of MHz digits value. 100's of KHz and below are ignored

Select freqz

Case 1 to 2
Goto bn1 ' band 1.8 MHz
Case 3 to 4
Goto bn3 ' band 3.5 MHz
Case 5
Goto bn5 ' band 5 MHz
Case 6 to 8
Goto bn7 ' band 7 MHz
Case 9 to 12
Goto bn10 ' band 10.1 MHz
Case 13 to 15
Goto bn14 ' band 14 MHz
Case 16 to 18
Goto bn18 ' band 18.1 MHz
Case 19 to 22
Goto bn21 ' band 21 MHz
Case 23 to 25
Goto bn24 ' band 24.9 MHz
Case 26 to 30
Goto bn28 ' band 28 MHz
Case 31 to 99
Goto bn50 ' band 50 MHz
Case 100 to 199
Goto bn144 ' band 144 MHz
Case 400 to 470
Goto bn432 ' band 432 MHz

Endselect

---

The HF port selection code :

' DIL Switches 2 - 5 are Low & pulled High by operation of the switch to set up 4 bit BCD value !
' giving range of 0 to 15 - thus 16 configuration possibilities

' port config from XLS file : band matrix antsw 1.xls

' variable *readinputs* contains a value from 0 to 15 that is used to set up any particular port configuration for HF

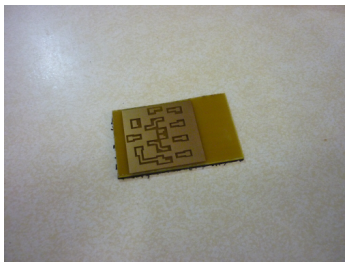
bn28: ' Select For BAND 28 MHz
Select readinputs
Case 5
Goto hfport1
Case 1,2,3,4,6,7

```
Goto hfport2
Case 8,9
Goto hfport3
Else
Goto hfport1
Endselect
Return

' ***************----------------------------------------


bn24: ' Select For BAND 24.9 MHz
Select readinputs
Case 1,5,6
Goto hfport1
Case 3,4,7
Goto hfport2
Case 8,9
Goto hfport3
Else
Goto hfport1
Endselect
Return

' ***************----------------------------------------

etc...

------------------------

An actual relay switching subroutine :

' ----------------------------------------

hfport1: ' output 7
Low RLG
Low RLF
High RLD ' must be High for any HF port


For temp = 1 to 1 ' flash the LED
High 7
Pause 200
Low 7
Pause 200
Next


Return
```

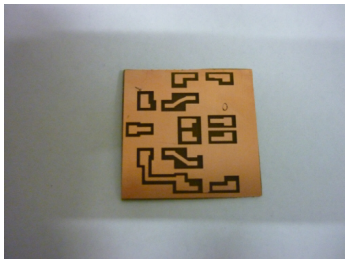
23 July 2010 : These are the steps used in the construction of the small relay PCB. The main PCB was created using the same process.

This is my old hinged exposure frame - after about 35 years of use and which was originally used for silk screening PCBs. The black 'bag' has cardboard inside and is used to control against accidental exposure to light and covers the whole of the wooden aperture.

The frame has the artwork ( in this case an inkjet transparency printed from ExpressPCB ) inserted on top of the PCB material (not shown, and with the black adhesive cover removed from the top side only) and the assembly is latched down with the simple metal hook at front. The PCB is then exposed to direct sunlight through the artwork for about 20 - 25 seconds by removing the 'covering piece' to do the timing.

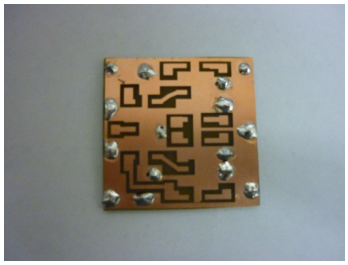


After exposure, the PCB is developed in a weak caustic soda (NaOH - Sodium Hydroxide) solution then etched in ferric chloride with this result. The ragged black tendrils on the edge underneath are actually the second black plastic adhesive covering (on the PCB as supplied). This is left on to prevent the second side from actually being etched.

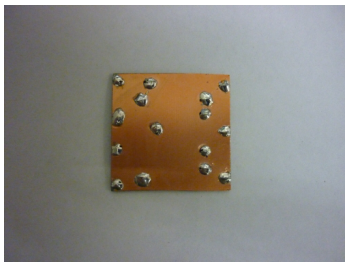


After trimming to size, cleaning off the PCB positive resist layer, the cover off the second side & drilling for the 'wire-through' vias.

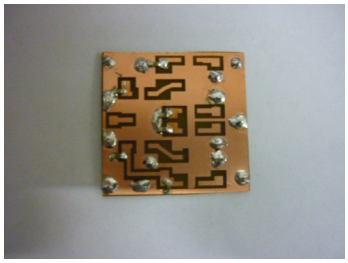
Actual PCB size is 40mm x 42mm.



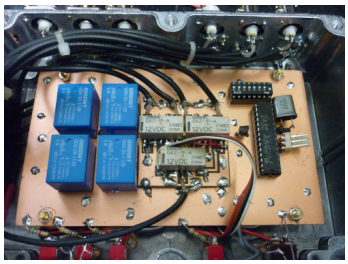
Wired 'vias' in place - top view.



Rear view with vias in place. Note that the PCB has the entire copper area available as a ground plane.



The 6 x 0.1uF (100nF) SMD bypass capacitors in place on the relay connection pads



The daughter-board mounted on the top of the main PCB after the original relays had been de-soldered and removed. The RG174 coax flyleads to the BNC sockets in place.

27 July 2010 : The results of installing the daughter-board :

As a matter of interest I did a quick measurement on 439 FM just using the S-meter on the 706, increasing the signal generator until it just indicated S1 when fed via a direct coax. I then interspersed this switcher box and I couldn't really pick any change in signal to noise and the S-meter still read S1 ! I then jumped to 146.5 FM and repeated the test - no noticeable S/N change there either.

Power loss testing was done at 439.000 FM so that a steady carrier was available. The transmitter power was measured at 14 watts.

Initially, the loss was such that output power from the 70cm FM port was about 8 watts (nearly 3dB loss). By changing the length of the cable between the radio & this switcher, the power varied between 7 and 9 watts..
To confirm where the loss was taking place, I de-soldered the 439FM port coax from the daughter-board and soldered a coax flylead directly to te PCB pad - essentially same power as via the panel BNC jack so that wasn't the major loss point.

I then replaced the incoming RG174 coax with a short length of teflon-series 50 ohm coax and the output power went up to 11.5 watts ( ie about 0.8dB loss through 2 relay contacts plus PCB losses plus coax losses).

Obviously if I change all of the VHF & UHF coaxes to solid PTFE-inner UT141 (or even more of a PTFE-style (teflon) coax) then the losses at UHF would be even lower - but is it worth the trouble given the termination method to the back panel BNC's ??

NO, not really. Not when you can have antenna gain to compensate !

THAT ESSENTIALLY ENDS THE PROJECT DEVELOPMENT. AT SOME LATER STAGE I MIGHT REVISIT THE CODING IN THE PICAXE TO SET UP A DIFFERENT PORT SWITCHING MATRIX BUT OTHERWISE THE BOX LID WILL REMAIN SCREWED DOWN..

I HOPE THAT MY PROJECT DESCRIPTION HAS BEEN INFORMATIVE AND USEFUL FOR YOU, AND, IF FOR NO OTHER REASON, MAKE YOU REALISE THAT PROJECTS LIKE THIS CAN BE DONE OUTSIDE OF A FACTORY.

Doug Hunter VK4ADC - 27 July 2010

---

*Project Note : I will NOT be publishing my full PICAXE code on this web page. The main reason is that, in the short term, the code is like to change as issues with it are identified and corrected - plus functional improvements are introduced.*

*Serious constructors will be able to obtain it directly from me by dropping an enquiry to my normal email address.*

*The other thought in the back of my mind is that someone else could use my ideas and R&D work for their own commercial gain....*

---

Estimated cost of parts for the project is around the $AUD100 - $AUD150 mark
- but, since you can't BUY one of these, anything that does what this does anywhere in the world, final cost is irrelevant.