FT-817 remote display

The FT-817 has a serial port on the rear panel, which provides information on the status of the radio, frequency, mode ‘S’ meter, etc. This information is detailed in the user guide in the section for the CAT interface

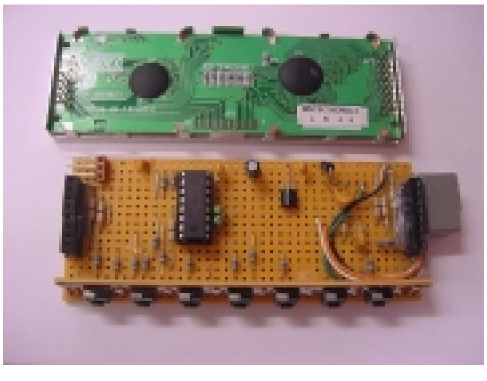
Harry ZL1BK asked me if it would be possible to make a remote display that would allow him to change frequency and mode whilst his radio was in his backpack.

After reading the user guide and the information on the CAT interface we decided it would indeed be possible we just needed to decide on what functions would be accessible form the remote display, out came the “wish” list, all the things that he would like to adjust whilst hiking up the mountain (or big hill)
To start there was the requirement that it contributed as little as possible to the total weight, had a big display that could be seen in daylight, was as small as possible and consumed little or no battery power. None of these requirements on their own was difficult but all in the one unit was a big ask. (And not to forget it had to have total control of all the radio functions).

To get back to reality a few of the requirements had to have some compromise to start there would be some power required (about 10ma) this could be sourced from the FT817 itself. The weight would depend on the size of the display as LCD displays are made of glass an it would be the heaviest part a compromise on size to weight but also on the availability of LCD displays (it was decided in the end that we would use the 2 line by 16 character one from DSE although any standard Hitachi chip set type could be used it should be noted that most ‘key’ phones use this type of display).
The number of functions that we could control on the radio from a remote display is limited to the commands listed in the user guide.

The next step was to build a prototype this was duly done using a LCD from an old telephone it was about 110mm by 35mm and ‘looked’ about the right size,. We started with a rotary encoder for the frequency adjustment and a few buttons at one end of the display but Harry didn’t like it as it made the board about 150mm long (“too big” was his comment) Back to the proto type and bit of head scratching …..We could mount buttons along the top of the display this would only add a few millimetres to the height of the unit. This was decision we only had to find the switches. I had some vertical mounting TACT switches but they required about 7mm additional space to mount them (Harry said too much, doesn’t want much eh!!) last idea was to mount the TACT switches on another small piece of ‘vero’ board and fit it to the top of the display this was done and approved by Harry.

Prototype ‘dumb’ display

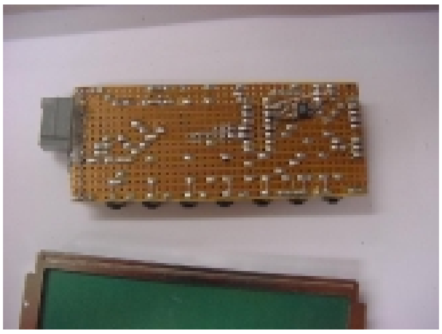


Initial hardware ready for testing

Next decision was what to use to control all this easy! a micro controller! any of the available micro controllers would do; AVR, PIC, etc, just pick the one you can write

code for. Okay so not everyone likes writing code for micro controllers, to make it an easy choice we will use the one that we already have the code for a Motorola MC68HC908KX8 it is a 16 pin dual inline device, the general specification for this device are obviously its from the 908 family of devices has 8Kbytes of Flash memory, 128 bytes of RAM, 14 input or 13 output pins, one of these pins can only be an input where all the others can be programmed as input or output, onboard UART and internal clock generator. So what does all this mean?
8kbytes of Flash this is the program memory (the code that makes it do the right things)
UART is the serial communication device and saves a lot of 'bit bashing' type code.
RAM is the volatile memory used for temporary storage of data (adding numbers)
The only thing we don't have in the micro is some changeable non-volatile memory or as we would know it as EEPROM we have selected a 24LC64 an 8kbyte I2C serial device this will be used to store some display messages, setup parameters, user memories, VFO information and anything else that may need changing during operation
That is the actual hardware more or less complete apart from some resistors and a 3 terminal voltage regulator, a total of some 18 resistors, 7 switches, 6 capacitors, 3 ICs, 1 LCD, cable/connector to connect to the radio and the 'Vero' board.
What can we make this little bit of hardware do for us? Just about anything that is controllable form the CAT interface. A few prelim decisions must be made before we get stuck into the programming.
Do we want to be able to turn the radio on~ off from the display?
Will we require a 'lock' function for the buttons?
Do we want 'memories' available and how many?
Do we want to be able to change operating mode and frequency?
Should we have 'band switching like the FT-817?
What step size should we use if we change frequency?
The answers to all the above is either yes or adjustable (and this means some nifty software has to be generated)

With the hardware built a test program was required to make sure the display and buttons where working this was done just before I was due to leave for my annual holidays I had the hardware working and made a connecting cable to fit between the radio and prototype. Remember the telephone display well the handset cord from the same telephone was used for the connecting cable I fitted the socket from the telephone on the prototype and wired the other end directly into the 8 pin mini Din plug that fitted the radio. I packed it with a programmer for the MC68HC908KX8, a few spares a PC and radio just in case the weather was not in our favour
Off on holiday for 2 weeks to Australia's Gold Coast. Arrived during electrical storms and lots of rain



Programming board for the micro
Display to 817 cable
DB25 is a programming lead for the EEPROM

display 'vero' board

At this stage I will have to admit that a large portion of the software was written in VK land so it was done by VK2IZX.With the weather not being so good it produce some interesting discussions on the local repeater regarding lighting protection.

I thank ***The Gold Coast Amateur Radio Society*** for these interesting discussions it gave me the needed breaks in programming (I had a few problems with my software turning the radio off).

Back to the software writing having proved that the prototype hardware was working the next step was to get the communications between the radio and prototype working This turned out to be a simple task I was concerned about the accuracy of the internal clock generator on the micro would not be good enough to have reliable communication at 38400 baud but it was. I tried some of the calibration routines as described by Motorola but they didn't seem to work only because it was as accurate as it could be before calibration, I have since tried several devices and found the same results, no calibration was required. At this stage I was able to read the frequency and mode from the 817 and display it on the LCD a good start I could now talk to the radio and receive the data from it. Next item to look at was the 'wish list' and allocate buttons to functions or functions to buttons initial allocation was, from left to right,

Button 1 this button is connected to the interrupt pin on the micro and can be used as a special purpose button, more on this later.

Button 2 at this stage it is not used I have plans to use it with one of the auto antenna tuners

Button 3 is menu select

Button 4 is band down/memory select

Button 5 is band up/VFO select

Button 6 is frequency down/memory down/ menu item select/confirm memory erase

Button 7 is frequency up/memory up/menu item select/confirm memory write

The software was written to accommodate the functions as listed above although some of the functions where not on the original 'wish list' they just evolved

The first function that I had to get operational was simple frequency up and down, looking at the commands available in the 817 there is no frequency up or down command how do I do that? Simple or so I though, I'll just get it to add or subtract to or from the frequency I have just read from the radio and then send it back to the radio I was able to read the frequency from the radio and save it into ram but its in BCD format I could add in BCD format but was not able to subtract in BCD format "Bright Idea" I'll go to the internet Café and do a search for BCD routines on a Motorola 908 series micro, results, nothing I got wet for nothing (yes it was still raining in sunny Queensland) I seemed to have a mental block on how to do BCD math on a 908 micro. I couldn't get stuck on this so the next best way around this problem is to convert to binary, add or subtract and convert back to BCD this done and working the next question, what to add? The step size should depend on mode of operation we don't want 10Hz steps on FM! My simple add or subtract just go a lot more complicated I now had to find the mode of operation before I went up or down in frequency so the correct step size could be added or subtracted. I wonder if the engineers at Yeasu thought about this when they wrote the software for the 817 as it sends mode information when you ask it what frequency its on. I now had to make a decision on the step size referring to the user guide for the 817 it does give step sizes for different modes just use them, this done I was now able to select a step size for each mode of operation (this turned out to be 3 sets I fixed WFM at 50kHZ)

I now had the basic operation of up and down in frequency working. This was a big step forward I had now proved that I could change the frequency and my BCD add and subtract worked (or so I thought more about this later)

With the step size adjustable and a added function of 1mHz steps it was looking good move on to the next function Band selection up and down, a telephone call to Harry to discuss how this function should work and arrange a sked with him using a wire antenna and my FT817 he was going to do the same in Auckland.

The band up and down function was a little more difficult than frequency as I had to check what frequency the radio was on and if it was in a valid ham band (or within the limits I had set for the ham bands) if it was within the ham band then it was to be saved for the next time that band was selected. This is what the EEPROM is used for. If the last used frequency was outside the ham band limits then it would not be saved and the next band up or down would be retrieved from the EEPROM and sent to the radio. Another problem was found here, as we changed band we could also change mode FM to SSB or any other mode the problem comes with changing from any mode to WFM the sequence of sending data to the radio was send mode first then frequency I done it this way to avoid the frequency changing when CW or DIG modes are selected as both of these mode have an offset added when selected. If changing band to FM broadcast (76~108 MHz) the mode can only be WFM if any other mode is sent it turns the radio off! At each band change a check was made to see if it was going to WFM and the correct mode information was sent (in fact no mode information was sent as the radio automatically went to WFM when this band was selected). Another leap forward in the software I now had both frequency and band up/down functions working quite usable at this stage also the rain in sunny Queensland had stopped so no more time to write software the weather was fine until my return to NZ all further software has been written by ZL1BQE Thanks again to ***The Gold Coast Amateur Radio Society*** for their interesting nets twice a day a very active club!

Back to New Zealand and also back to work this would slow the progress on the software, but the main part was already done by VK2IZX just the added extras like memories and better frequency stepping etc.

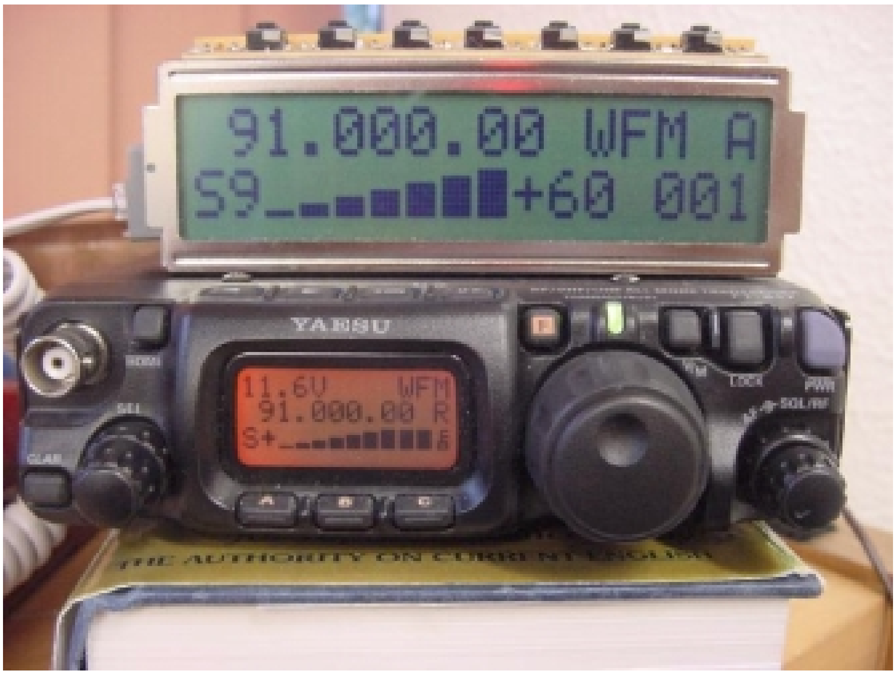
Harry's prototype was the next project he had an untested one and I needed to update his software so he could try the 'new' functions (all his would do was change display information when the buttons where pushed). By now Harry had time to think about the 'wish list' and after trying the new software a few things got added and a few changes made. It would be nice if, when changing frequency the button was held down it would change faster (or take bigger steps) a quick look at the software and yes we could make it do that just had to make the buttons appear to be released after a preset time and the up/down software would step one more ("Bright idea" If I counted how many steps where taken and at say the 5th step multiply the step size by 10 it would appear to be stepping faster, this was done and it did appear to work well now if I could do the same at the 10th and 15th steps and multiply by 100 and 1000 that would be even better. Done! This counter was reset when the button was released so pushing the button momentarily the frequency only increased or decreased by one step size at a time)

Another tick on the wish list and move on to the next item this continued until there was only 1 item left 'SCAN' I'm still looking at this one and don't know if it will be added it will depend on the demand and if Harry is persistent about it. The other functions that are working: memory save and erase, power on and off, lock function to stop inadvertent frequency change for both the radio and push buttons on the display (it should be noted that as soon as the display is connected to the radio and turned on the radio is put into lock mode) step size selection and mode selection.

The last things that need to be completed is not software but hardware it needs a 'House' to live in and it does radiate some noise from the serial cable (I left this for Harry to fix).

If your are interested in making a display please email keith@kc.net.nz if there is enough interest I will continue with the project and make a complete PCB and add more features to it.

As with all software there is no warranty or guarantees that it does all the things you want it to do and not do things it should not do. I would suggest that you record the 'settings' in your FT-817 prior to connecting this display to it. With all the mistakes in writing the software I have only managed to do a 'reset' on the radio but not a 'factory' reset, if you are unsure how to record these settings then do not connect the radio to the display until you are. (as in all text I have read on using the serial port on the FT-817 "you have been warned")



On the display top line left to right frequency, mode, VFOA(or B or M or L memory and lock) 2nd line 'S' meter (in TX power meter) and memory number 001~250 in this space it will display the menu items memory write, erase, mode, lock, repeater +/- step size and maybe Scan
Buttons along the top left to right Function, auto tuner, menu, band down, band up, freq/mem down and freq/mem up.
Function + bandup changes to memory
Function + band down selects VFO A or B
Function +freq up/down step 1MHz
Function +menu hold to power off
Function hold for power on

Thanks to Harry ZL1BK for all the testing Simon ZL1SWW for drawing the circuit