VK5DJ Remote Controller Software
and modifications to the WW2R control board

Dave WW2R has produced a remote control board and this is described on his site:
http://users3.ev1.net/~g4fre/dtmfun.htm

The VK5DJ repeater controller, see http://vk5dj.mountgambier.org/Amateur_radio.html has a number of inbuilt remote controls that affect the operating conditions but it does not have facility for external functions. By putting the two boards together there is facility for even more control.

Dave's board uses a MT8870 DTMF decoder, a 16F84 PIC, a ULN2803 Darlington Driver, a bar LED and a handful of other bits. See his site for details.

My version of his board omits the MT8870 and its associated parts, including the crystal oscillator because all of this is already on the VK5DJ controller.

Here is how the remote controller works.

As there is no input for a COS from a receiver, the software in the 16F84 waits for a "*" character, then a three digit password sequence and then:
Digit 5 = Port number (1-8 are real ports on pins 11-13 of the ULN2803) while ports 9 and 10 are pseudo ports – see later).
Digit 6= Direction (0=low, 1=high, not used if port is 9 or 10)
Digit 7=Period (0=locked and stored, any other number defines a pulse, not used if port is 9 or 10)
Digit 8= # sets the control function.

Rules

- A "*" always restarts a sequence.
- A 3 digit password must always follow the "*" or the sequence will reset
- A real or pseudo port number must follow the password
- A direction byte follows (0=low, 1=high)
- A # always actions a sequence.
- If there is no tone for 7 secs then the sequence resets and awaits a "*".
- An actioned comand is always followed by an "R" sent in morse code. A pulse command has two "R"s, one when the command is accepted and once when the port finishes the pulse.

Notes
A successful control sequence is indicated with an "R" sent in morse code. This audio at about 1KHz appears on pin 3 of the PIC and is shaped with the components added to the circuit.

Imagine that the password is 123
To set all ports high send *1230# (note 6 digits- the direction and period digits are omitted)
To set all ports low send *1239# (note 6 digits) - the direction and period digits are omitted)

To set port 5 high send *123510# (note 8 digits: *=start sequence, 123=password, 5=port number, 1= set high, 0=hold and store)
To set port 5 low send *123500# (note 8 digits: *=start sequence, 123=password, 5=port number, 0= set low, 0=hold and store)

<u>For a positive going pulse on port 5</u> do the following:
Send *123511# (*=start sequence, 123=password, 5=port number, 1= set high, 1=period is 100mS, #=do it)

An “R” will be sent to acknowledge a correct sequence, then port 5 will go high for 100mS, then low to its memorised value, then another “R” is sent.

For a negative going pulse on port 5, the port must first be put into a default of high.
To set the port high send *123510# . The new condition is stored and displayed and becomes the default value of that port. It will be remembered even if the power goes off.
Next send:
123501# (=start sequence, 123=password, 5=port number, 0= set low, 1=period of 100mS, # = do it)
An “R” will be sent to acknowledge a correct sequence, then port 5 will go low for 100mS, then return to its memorised value of high, then another “R” is sent.

Note that if a port is memorised high then a high pulse will do nothing or if a port is memorised low then a low pulse will do nothing although you will hear an “R” to acknowledge a correct sequence was sent.

The remote controller does not respond if it is in the middle of timing a port operation. Eg if you decided to put port 2 ON for say 4 minutes then you will not be able to engage a new control for 4 minutes as the PIC is away doing a polled count. Wait for the second “R”.

Pulse number sent	Pulse length
1	100ms
2	200ms
3	300ms
4	400ms
5	500ms
6	1sec
7	2sec
8	3sec
9	4sec
#	1min
A	2min
B	3min
C	4min

The password is stored in EEDATA in the PIC at locations 1,2 and 3
In the attached file the password is 279. This may be altered with your programmer at burn time. The “R” is stored in EEDATA at locations 40,41,42 decimal. If you want a longer message then use the timings and numbers described in detail in the VK5DJ repeater controller manual. The acknowledge message finishes when it hits the hex 55.
Morse messages appear as audio, there is no PTT action so the assumption is that the operator is listening to the repeater and holding the key down.

Hardware
The WW2R board needs some modification.
Firstly do not install the MT8870 or the associated parts, except R4 should now use a pad that will enable a lead to go to pin 5 of the DTMF expansion socket on the VK5DJ board. Note that pin 1 of the DTMF socket is nearest the voltage regulator chip.

Secondly pin 16 of the PIC should go to a 33-47pF cap and then to pin 6 of the DTMF expansion board on the VK5DJ controller. It happily shares the crystal output of the MC145436 decoder chip with the other PIC.

Pins 17,18,1,2 of the PIC connect to the DTMF expansion socket pins 1,2,3,4 respectively.

Connect a 100K resistor in series with pin 3 of the PIC and a 10K pot to ground. Connect a 10K resistor to +5V from pin 3 of the PIC. Put a 0.1 uF cap across the top and bottom of the pot. From the wiper of the pot go through a 47K resistor to pin 9 of the 12 way socket on the repeater controller. If you want an audible response connect a small speaker, one lead to +5V via a 330 ohm resistor, the other lead through a 2uF cap to pin 3 of the PIC.

The unused lands on the MT8870 position will serve to mount all the extra components associated with smoothing the tone output and connection of a small speaker.

I’ve attached images of the original and the modified circuits.

Let me know if you need more information or any modifications to the software.

Best wishes
John VK5DJ