VK5DJ Remote Controller Software
and modifications to the WW2R control board
Mk II June 2019 modified from 2014

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.com 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. I have used a 16F628A.

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.

Alternatively you can make up the circuit as provided by George's circuit included with this file.

Here is how the remote controller works.

As there is no input for a COS from a receiver, the software in the 16F628A waits for a "*" character, then a three digit password sequence, and then a command section followed by # to indicate a command sequence is complete:

Six digit code (set or reset all ports)

*<pwd> command, #

This combination places ALL ports low or ALL ports high. A 9 command sets them high while a 0 sets them low. This is then memorized and after loss of power will reset to these levels when the power returns.

Example: *<pwd> 0 # puts all ports low while a *<pwd> 9 # puts all ports high.

Seven digit code (pulse mode)

*<pwd> portno,period, #

This combination sends the password, then the port number, then the period you want the designated port to pulse once, a # completes the sequence. The port pulses in the opposite direction to which it is currently set. So if the port is set high then a pulse command lowers the port for the period set in the table below.

Example: *123 2 5 pulses port 2 for 500msec

Eight digit code (single port set mode)

*<pwd> portno, period, direction, #, causes a single port to assume either a low or a high. The value is memorized to guard against loss of power.

Example: * 123 101 # where port number = 1, period = 0, and direction = 1 will cause port 1 to be high. A *123 100 # would set port 1 low.

Period **must be a 0** for this command or the command will fail.

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
If there is no tone for 7 secs then the sequence resets and awaits a "*".
A # always actions a sequence.
Commands other than 0 and 9 have a "MSG X" where "X" is the number of the port being manipulated sent in morse code before the command is actioned.
Two tones (a low and a high) indicate which way a port has been actioned (Low/High = port made high, a High/low indicates a port made low, while Low/High/Low or High/Low/High indicates a pulse.)

An actioned pulse command is followed by an “R” sent in morse code. For pulse commands another “R” is sent to indicate the command is complete.

Notes
Morse audio at about 1kHz appears on pin 17 of the PIC and is shaped with the components added to the circuit. PTT is available from T8 (active low).

Imagine that the password is 123

To set all ports high send *1239#	(note 6 digits- the direction and period digits are omitted)
To set all ports low send *1230#	(note 6 digits) - the direction and period digits are omitted)
To set port 5 high send *123501#	(note 8 digits: *=start sequence, 123=password, 5=port number, 0=period, 1=high and store)
To set port 5 low send *123500#	(note 8 digits: *=start sequence, 123=password, 5=port number, 0= period, 0=low and store)

Pulses work differently in Mark 2 of the code:
Pulses now invert the state of the indicated port for the period of the pulse.
This avoids the need to set a port to a particular state and adds flexibility to a setup. Perhaps preventing the need for an inverter transistor.

Send *12351# (*=start sequence, 123=password, 5=port number, 1=period is 100mS, #=do it)
A message will be sent to acknowledge the port to be manipulated, then port 5 will invert for 100mS, then return to its memorised value, then an “R” is sent to indicate completion of the pulse.

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 3 minutes then you will not be able to engage a new control for 3 minutes as the PIC is busy doing a polled count. Wait for the “R”.

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

This version of the software does not respond to port 8 commands other than acting as a PTT source (active low) to activate a transmitter. Morse code audio is derived from pin 3 of the PIC through a smoothing network and a level setting pot.

The password is stored in EEDATA in the PIC at locations 0,1 and 2
In the attached file the password is 123. This may be altered with your programmer at burn time. The “R” is stored in EEDATA at location 72. 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 FF.
Morse messages appear as audio.

Hardware
If you use George's circuit (DV2GAP) for a standalone remote control circuit then just follow his connections and ignore the comments below. Note that both circuit diagrams have an error in the ULN2803 connections. The wires to pins 3 and 4 should be swapped, as should the wires to pins 5 and 6 to ensure an increasing sequence on the LED bar.

Using the WW2R board with my Mk1 repeater controller
Some modification to work with my repeater controller.
Firstly do not install the MT8870 or the associated parts. PIC pins 3,2,1,18 now go to pins 5,4,3,2 respectively of JP1. Pin 7 has a 10K resistor to the cathode of a diode and then to pin 1 of JP1 of the VK5DJ controller board. Another 100K resistor is added between pin 17 and ground. Lastly another 100K resistor passes through a simple low pass filter to an output suitable for a volume control and microphone input. Note that pin 1 of the DTMF socket (JP1) is nearest the voltage regulator chip.

Lastly pin 16 of the PIC should go 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 (not applicable if your controller uses the 16F1827).

Although not shown on the circuit diagram, the tone out of the board could go to the top of a 10K potentiometer, the bottom to ground and the wiper (via a 47K resistor) to pin 9 of the 12 way socket on the 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 1uF cap to pin 3 of the PIC. I haven't tested this and it could cause a momentary high on pin 17 when first connected. This will be a short-lived problem at boot up only in that the PIC will look for a DTMF character. It will recover in 5 secs.

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

I've attached images of the original and the modified circuits.
The following table shows the different connections for the two supported tone decoders.

MC145436 pin	**M8870 pin**	**PIC16F828A**
12 (Data valid)	15 (Data valid)	15
13 (D3)	14 (Q4)	2
14 (D2)	13 (Q3)	1
1 (D1)	12 (Q2)	18
2 (D0)	11 (Q1)	17

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

Best wishes
John VK5DJ
Modified 30 July 2014
Update 23 June 2019