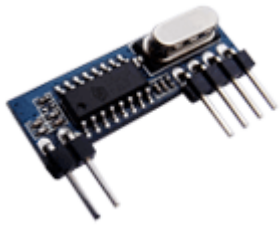
# Wanderbug Radio Control Assembly Instructions

**You will need:** Soldering iron, solder, side cutters, 35cm of single-strand, insulated wire.

## A Receiver expansion board

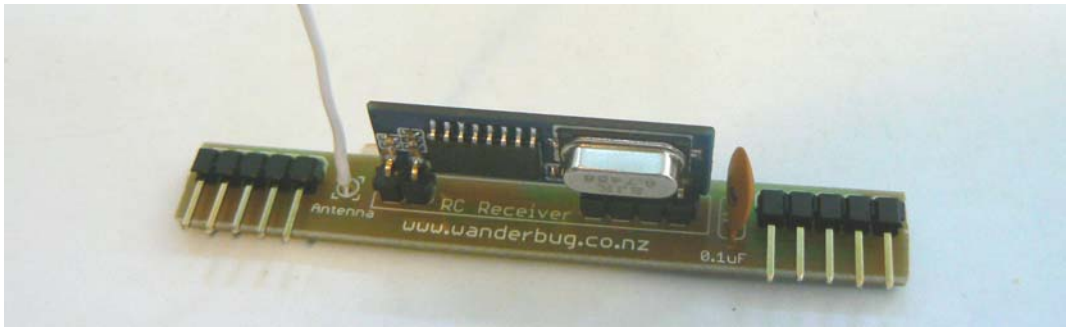
1. Cut the six-pin, right-angled header into a two-pin and four-pin header using side cutters and solder to the **Dorji** receiver unit.



2. Fit the right-angle headers in place, solder and snip off the excess pins.


3. Fit the 0.1uF capacitor (either way around), the **Dorji** receiver module, solder in place and snip off the excess leads.

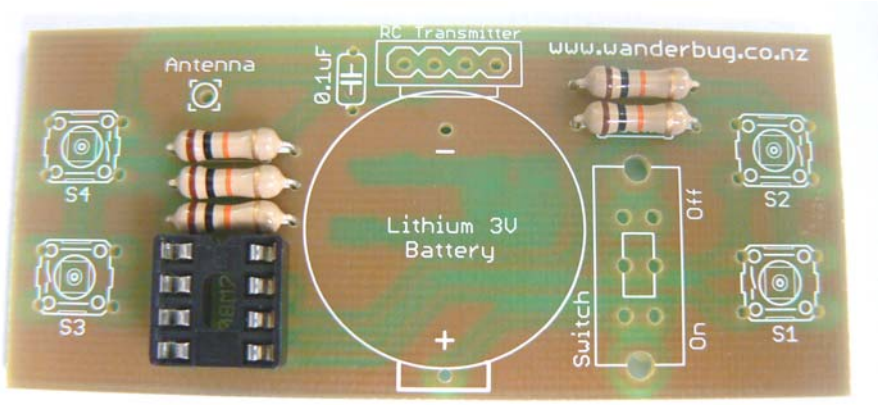
   Cut a length of single strand insulated wire 17.5 cm long, strip one end and solder in place as an antenna.



## B Transmitter board

1. Solder the five 10kΩ resistors in place (10kΩ is brown-black-orange). Snip off the excess leads.

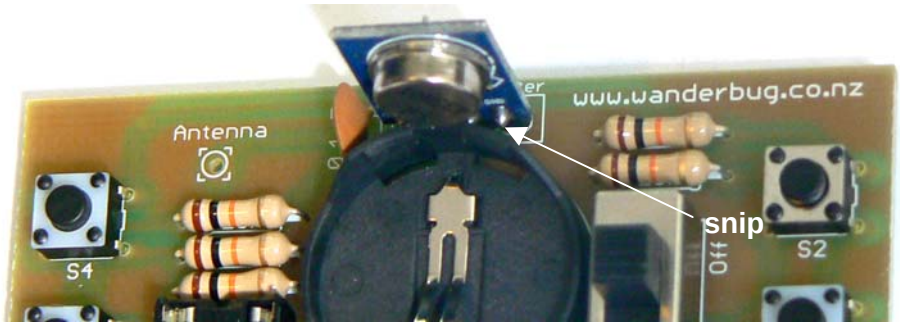
   Solder the 8 pin IC socket in place with the dimple to the top



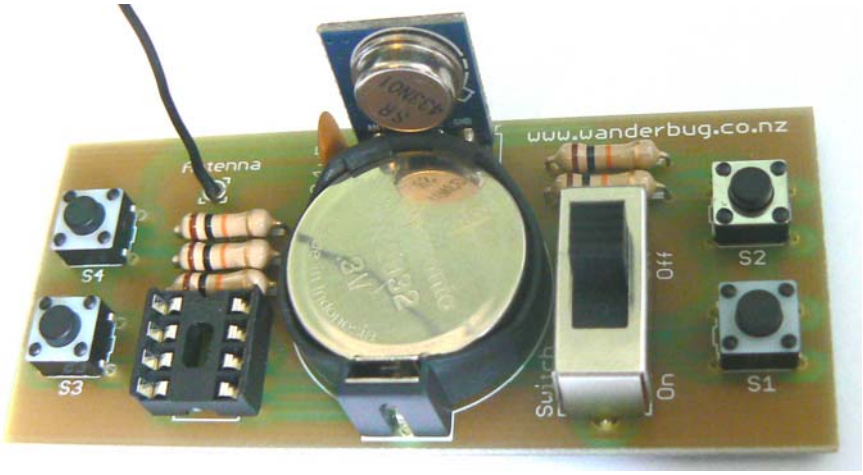

2. Solder the battery holder and 0.1uF capacitor in place. Snip off the excess leads.


3. Fit and solder the main switch and the four push-button switches in place. Make sure all switches are pushed in fully before soldering.

4. Snip off the excess pins on the **Dorji** transmitter module so that it will fit snuggly in beside the battery holder, and solder in place.

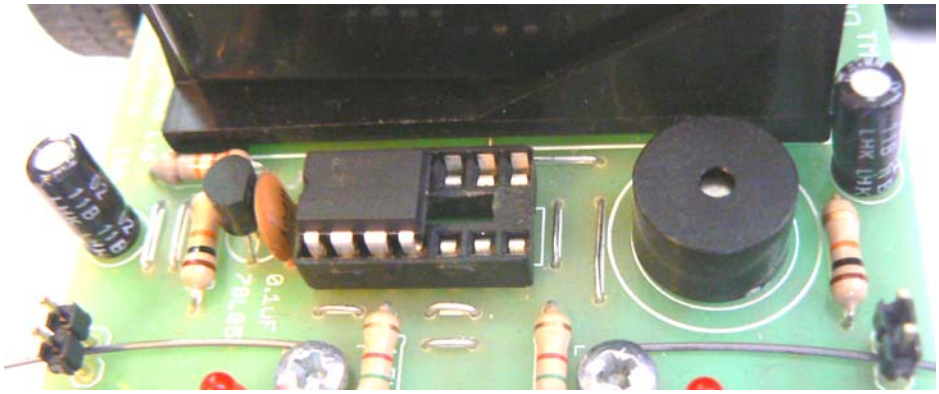



5. Cut a length of single strand insulated wire 17.5 cm long, strip one end and solder in place as an antenna. Make sure the switch is in the off position and fit the battery.




6. The PICAXE 08M2 chip has been programmed with code $00. There are 16 codes available so that the 433MHz channel can be shared by up to 16 different Wanderbugs. The codes available are $00, $10, $20,…,$e0, $f0, counting in hexadecimal.

If the code needs to be changed the 08M2 chip needs to be inserted into the Wanderbug 14M2 socket as shown, and the line:

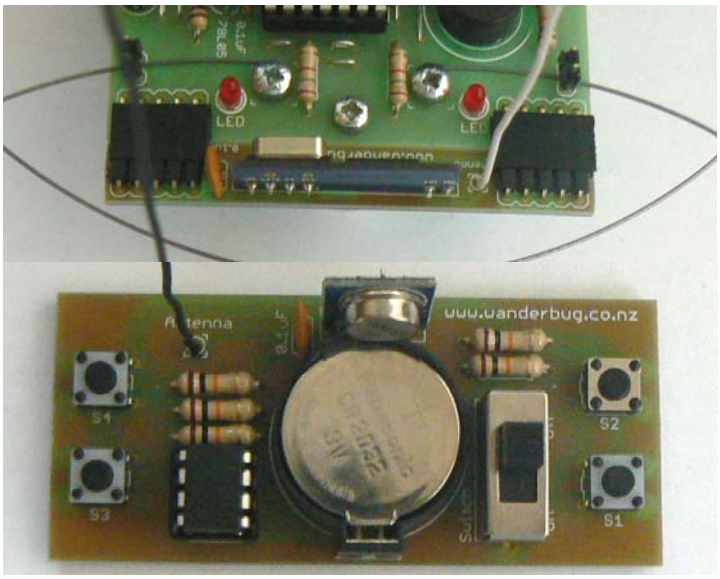


```
symbol code = $00                          '$00 to $f0, 16 unit codes
```

can be changed and programmed into the PICAXE 08M2. This is then inserted into the transmitter IC socket.

If the code has been changed in the 08M2, then the matching code must also be changed in the 14M2 so the receiver commands will be accepted.

The completed modules look like this:




| Wanderbug Radio Control kit parts list | | | | | |
|---|---|---|---|---|---|
| **Receiver** | | | **Transmitter** | | |
| **Item No.** | **Qty** | **Description** | **Item No.** | **Qty** | **Description** |
| 1 | 1 | Receiver PCB | 6 | 1 | Transmitter PCB |
| 2 | 1 | Dorji receiver | 7 | 1 | Dorji transmitter |
| 3 | 1 | 6M R/A Pin Header | 8 | 5 | 10kΩ resistors |
| 4 | 2 | 5M R/A Pin Headers | 9 | 1 | 8 pin IC socket |
| 5 | 1 | 0.1uF Capacitor | 10 | 1 | 3V Battery holder |
| | | | 11 | 1 | 0.1uF Capacitor |
| | | | 12 | 1 | On-Off switch |
| | | | 13 | 4 | Push-button switches |
| | | | 14 | 1 | PICAXE 08M2 IC |
| | | | 15 | 1 | CR2032 3V battery |

## Sample Code Receiver

```
                                        'Wanderbug RC Receiver
                                        'PICAXE 14M2

#picaxe 14M2

let dirsb = %00111111                   'configure Port B pins as outputs
let dirsc = %00000010                   'configure Port C pins as inputs

symbol code = $00                       'this must be the transmitter code
symbol incode = b0                      'incoming code from transmitter
symbol name = b1                        'incoming ID code

main:                                   'main loop starts here
if pinc.3 = 0 then left                 'left bumper has hit an object
if pinc.4 = 0 then right                'right bumper has hit an object

serin [100,main],c.0,n1200_4,(127),incode 'wait for correct signal
name = incode & $f0                     'check if it's my code
if name = code then                     'yes it is, set the outputs
      pinb.2 = bit0                     'to suit the commands received
      pinb.3 = bit1
      pinb.4 = bit3
      pinb.5 = bit2
endif
goto main                               'carry on

left:                                   'left bumper object routine
low b.2:high b.3                        'reverse right wheel
low b.4:low b.5                         'stop left wheel,
high b.1                                'left LED on
high c.1                                'sound the buzzer
pause 500                               'pause for 500 milliseconds
low c.1: low b.1                        'buzzer off, left LED off
low b.3                                 'stop right wheel
goto main                               'look for new signal

right:                                  'right bumper object routine
low b.2:low b.3                         'stop right wheel
low b.4:high b.5                        'reverse left wheel
high b.0                                'right LED on
high c.1                                'sound the buzzer
pause 500                               'pause for 500 milliseconds
low c.1: low b.0                        'buzzer off, right LED off
low b.5                                 'stop left wheel
goto main                               'look for new signal
```

## Sample Code Transmitter

```
                                        'Wanderbug RC Transmitter
                                        'PICAXE 08M2

#picaxe 08M2

dirs = %00000001                        'configure 8M2

symbol code = $00                       '$00 to $f0, 16 unit codes
symbol buts = b0                        'button code
symbol oldbuts = b1                     'last button code

main:                                   'main loop
buts = pins                             'read the buttons command
pause 20                                'slow the loop
if buts = oldbuts then main             'no change, read buttons again
oldbuts = buts                          'store the new command
buts = buts/2                           'shift right one bit
buts = buts & $0f                       'clear the top nibble
buts = buts + code                      'add the unit code
serout 0,n1200_4,(85,85,85,85,127,buts) 'new command, send it
goto main                               'check for another command
```