

ROVER CODE

Step 1

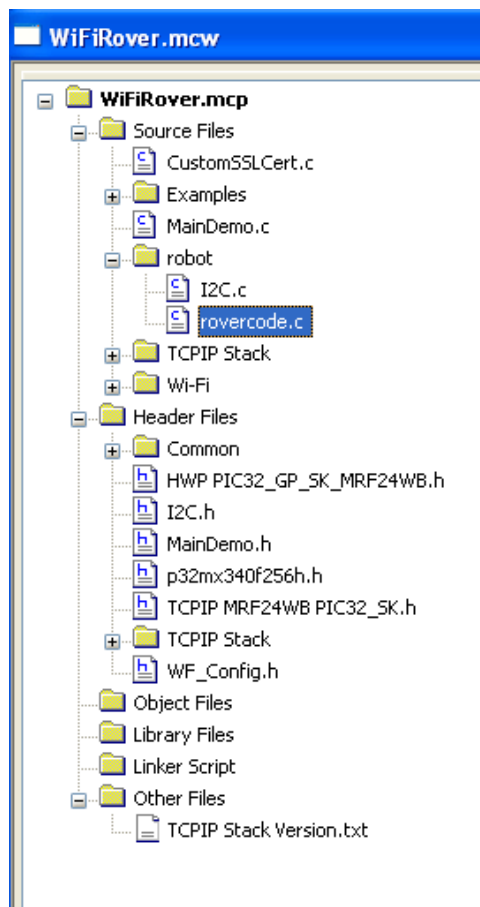
Extract to your drive.

Step 2

Navigate to \WIFIROVER\TCPIP\Demo App and double click WiFiRover.mcw

This should load MPLAB

Then all the interesting bits (that you will change) are in rovercode.c.



Don't change anything else unless you are sure!

The code receives TCP messages and responds with TCP messages.

```
void processcommand(void)           // the main routine which processes commands
```

There is code for all the commands specified in 2013 DATATYPES.docx on the Z: drive..

In order to respond use the following procedures:

```
POSTTCPhead(LEN,ID); // prepares a response of TAG ID and length LEN
```

```
POSTTCPchar(c);      // adds a char to the TCP packet. Packet gets sent when processcommand() finishes.
```

There is a routine:

void InitializeBoard(void)

which sets up ports etc. of the board. You might need to change this if you change what is attached to the board.

There are a number of Interrupt routines which either cope with timing or devices:

- void __attribute__((interrupt(ipl3), vector(_ADC_VECTOR), nomips16))) **_ADCInterrupt**(void)
This reads the two ADCs channels used for the Line following. The routine also decodes the IR bit stream from the crashed rocket (uses the fact that it is called at a regular rate).
- void __attribute__((interrupt(ipl4), vector(_INPUT_CAPTURE_1_VECTOR), nomips16))) **_IC1Interrupt**(void)
This records the movement of one of the wheels.
- void __attribute__((interrupt(ipl4), vector(_INPUT_CAPTURE_3_VECTOR), nomips16))) **_IC3Interrupt**(void)
This records the movement of other wheel.

Contained In other c files, to make sure you don't modify or use them)!

- in TICK.C
 - void __attribute__((interrupt(ipl2), vector(_TIMER_1_VECTOR), nomips16))) **_T1Interrupt**(void)
Timer 1 is used by the system (WIFI etc) to measure time
- in WF_Eint.c
 - void __attribute__((interrupt(ipl3), vector(_CHANGE_NOTICE_VECTOR), nomips16))) **_WFInterrupt**(void)
The change_notice interrupt is used as part of the handler for the WIFI module

OTHER PERIPHERALS of the PIC that are used:

- Input captures 1 to 4.
- SPI PORT 1
- I2C port 2
- Timer 2
- Timer 3
- Output compares 2 to 5

Examples of cases in processcommand()

number after case matches ID number

```
case 2:if (commandlen==1)           //Set LEDs
    { // OK we have 1 byte of info
      // look at individual bits in data using AND and set the port pin.
      if (nextcommand[1] & 2) LED1_IO=0; else LED1_IO=1;
      if (nextcommand[1] & 1) LED0_IO=0; else LED0_IO=1;
      POSTTCPhead(0,2); // send a null ACK with ID 2
    }
    break;
case 3: // get internal counter
    POSTTCPhead(2,3); // set up return packet to contain 2 bytes with an ID code of 3
    POSTTCPchar(REQCOUNT & 0xff); // send the lower 8 bits of count
    POSTTCPchar((REQCOUNT>>8) & 0xff); // send the upper 8 bits of count
    REQCOUNT++; // increment the count
    break;
```