Software brief

Write a C++ program that I can compile in Visual Studio 2012.

SW interrogates COM6, 38400 baud, 8 databits, no parity, 1 stop bit, no handshake.

Start Main function:

1. Open the port
2. Define a variable called 'Shutdown' and initialise to 0.
3. Define variables INTEGER x e f and initialise to 100.
4. Define an outputstream to a file called c:\autonomo\web\tmp\WEEDIT-DATA-<date>.m
   Example file name: WEEDIT-DATA-2017-02-21 note year is followed by month then day
   If file exists then append letter a or b or c or d, etc. Example: WEEDIT-DATA-2017-02-21a
   Clear the DATA.m file.

Then the program goes through a <u>continuous loop</u> as followed:

1. Send PX0<cr>
2. Get sensor response – see attachment for format of response
   -> Convert and write the data read from port into following INTEGER variables:
   x (Status) Example: '3' means Running
   e (Error) Example: '0' means No error
   f (Warnings) Example: '3' means Low Battery

   Notes:
   - above variables are being overwritten every time the loop is executed.
   - if no data for a variable is being read, eg. no Warnings, then set the variable value to 100.

3. Send BX0<cr>
4. Get sensor response – see attachment for format of response
   -> Convert and append the data read from port into DATA.m file:
   Here's where it gets a bit tricky as we may get any number of 0's and 1's returned, usually between 5 and maybe up to 200.
   The logic is something like this:

```
For (i=0,i<number of 0/1's read, i++)
{
        If(Returndata[i] = 1)  //Returndata is the array of 0's and 1's read from the port
        {
                Append to DATA file the following row:
                Date(yyyy-mm-dd);Time (HH:mm:ss.ss);number-value;1<cr>
                Example:
                2017-02-21;14:59:05.00;-10;1<cr>
        }
        else
        {
```

Append to DATA file the following row:
Date(yyyy-mm-dd);Time (HH:mm:ss.ss);number-value;0<cr>
Example:
2017-02-21;14:59:05.00;-10;1<cr>

}

// Now, the 'number-value' needs to be calculated as followed:
// ((i+1)-((number of 0's and 1's read from port +1)/2))*5
// Example: reading the 3rd value in a sequence of 15 0/1's returned (i=2!) =
// ((2+1)-((15+1)/2))*5 = -25

}

Therefore the DATA file may look something like this for one read of the data from the port:
2017-02-21;14:59:05.00;-15;1<cr>
2017-02-21;14:59:05.10;-10;1<cr>
2017-02-21;14:59:05.20;0;1<cr>
2017-02-21;14:59:05.30;5;1<cr>
2017-02-21;14:59:05.40;20;1<cr>
etc.

5. Sleep(500)

Loop is terminated once 'Shutdown' is set to 1 – you do not have to worry how it gets set to 1.

Close Port and any other cleanup tasks required.

Notes:

Code needs to check for any errors, eg. opening / closing the port, etc..

Protocol: see attached pdf