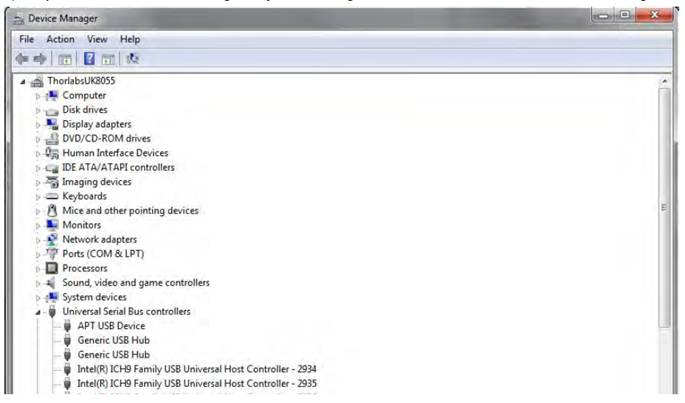
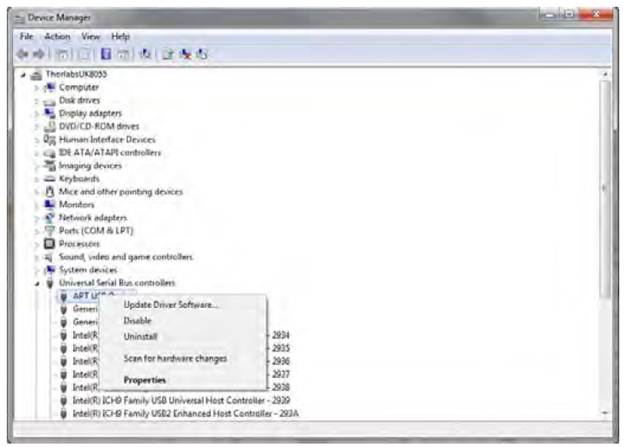
**Using a Virtual Comm Port**

When using the low level communications protocol messages to develop client applications outside of the APT software, a virtual comms port can be configured as follows:

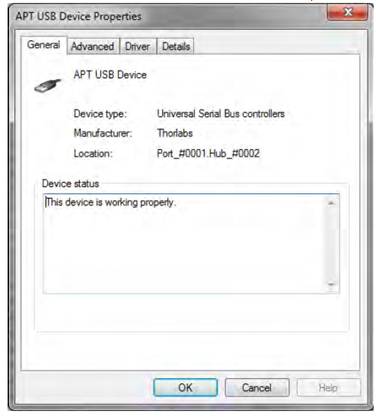
1)      Open the device manager by selecting Start/Control Panel/Device Manager/



2)      Click ‘USB Serial Bus Controllers’ and select the APT USB Device to be configured, then right click and select ‘Properties’.



3)      The ‘USB Device Properties’ window is displayed.



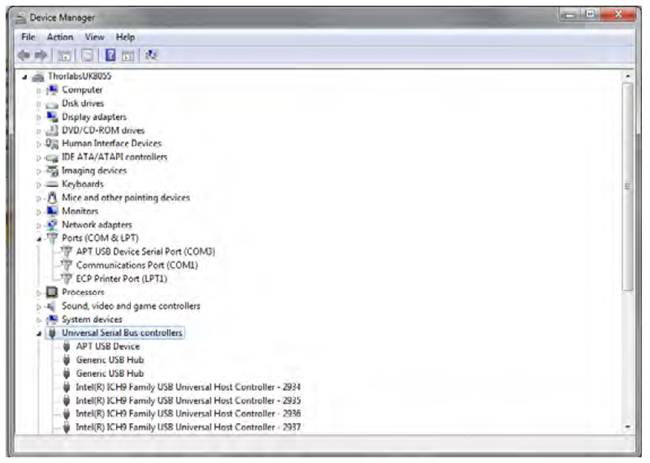
4)      Select the ‘Advanced’ tab, and check the ‘Load VCP’ box.



5)      Click OK, then power cycle the device being configured.

6)      In the device manager, click ‘Ports (COM & LPT)’, and note the ‘APT USB Device Serial Port’ COM port number (e.g. COM3).

7)



This COM port can then be used for low level protocol messages.

We do not have any specific examples for using our protocol document but we do have some general tips that will help you find the correct byte messages to send.

1)            Com port settings baud rate that should be used is 115,000. Without this being set correctly odd bytes will be seen on the return and the controller will not respond correctly.

In addition;

•             Flow control = RTS/CTS

•             Bits per word = 8

•             Number of stop bits = 1

•             Parity = NONE

2)            It is also worth clearing the buffers to ensure there are no part messages left in there which would prohibit any further communication. To clear the buffers before calling the commands please use.

// Pre purge dwell 50ms.

Sleep(uPrePurgeDwell);

// Purge the device.

ftStatus = FT\_Purge(m\_hFTDevice, FT\_PURGE\_RX | FT\_PURGE\_TX);

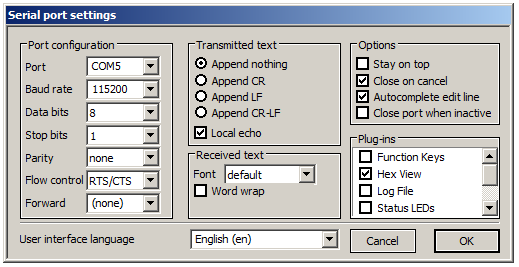
// Post purge dwell 50ms.

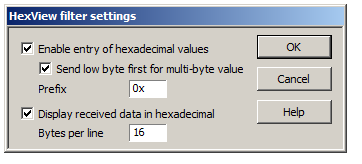
Sleep(uPostPurgeDwell);

3)            As a tip, to find the correct command string and message exchange, simply open APT user as normal and open the system log from the view menu. In here is the event log. This will allow you to see the hardware communication to and from the device, so for example when you click enable it will show the bytes sent when this command is clicked, or for example when a move is made it will show the byte string sent. This helps remove the need for frustrating reverse engineering. If hardware communication does not show up in this event log, then you will need to open APT config, click on the server tab, and check the ‘enable hardware communications’ box first.

4)            Also as a general suggestion, one thing that you can try to aid with your debugging (and does not rely on any specific hardware or software to be functional) is to download a terminal emulator that allows hexadecimal input (for example Termite, there are quite a few and most of them are freeware), enable the virtual comms port on the FTDI USB port (Control Panel->System->Hardware->Device Manager->Universal Serial Bus Controllers->APT USB Device->(right click)->Properties->Advanced->Tick the Load VCP box), and then you can directly send the commands, exactly as described in the APT Communications Protocol document, to the controller. We use this all the time when somebody has problems with a particular command because we can try the same byte sequence and see what happens or how the controller responds.

For termite to work you need to ensure you have the correct setting enabled, see below. Also if you have sent something to the controller and not got a response in termite it is often necessary to power cycle the controller, this is because you can mess up the USB connection and fill the buffers with nonsense and the only way to reinitialise the USB port is to restart the controller and connection. This is a USB issue not a controller issue.





When you enter data you should use this format

