Abstract:

Determine the current state of affairs for the Copernicus robot. Investigate and assess existing hardware and software for the project, prepare Kinect control program for future use with robot. Outline the current status of Copernicus robot and provide helpful, tangible information for the next team, furthering the project.

Immediate tasks:

-Install robot body onto roving base.

-Assess motor driver “black-box.”

-Attempt control of base motors.

-Powerpoint program for presentation related to Copernicus robot.

-PowerPoint program outlining project.

Attempted:

Leverage software provided by J.R. Kerr website. The software is written for Visual Studio 2003. We attempted to recompile the libraries, DLLs and example projects from source, we were unable to complete this task in the several hours dedicated to it. As we were unable to compile this source into a useable program, we were not able to merge it directly with our Kinect program and thereby control the robot via the Kinect as intended. The motors, serial communication and motor driver boards all tested functional when using the NMC test utility. The NMC test utility program does work on current versions of Windows, however, as it is a \*.exe Windows executable file.

An outline of the required experimental equipment, platform, etc. follows below:

Provided by: J.R Kerr LLC (mfg. of motor controller/RS-232 Serial device)

Software:  
<http://www.jrkerr.com/software.html>

Linux:  
<http://www.praecogito.com/~brudy/blue_cube/code/blue_move/>

Documentation:

<http://www.jrkerr.com/picsrvsc.pdf> ← (command summary included here)  
<http://www.jrkerr.com/pioboard.pdf>  
<http://www.jrkerr.com/piodata.pdf>  
<http://www.jrkerr.com/boards.html>

Installed Software:

FTDI Drivers:  
<http://www.ftdichip.com/Drivers/D2XX.htm>

Visual Studio 2015 Enterprise→ Install VC++ tools through Setup

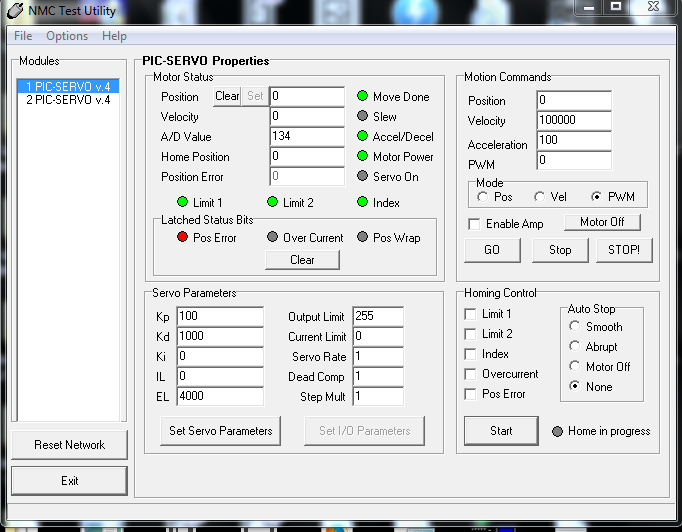
NMC Test Utility:

Figure 1: Screenshot of NMC Test Utility

Wireshark USB sniffing: (via USBPcap)

<https://wiki.wireshark.org/CaptureSetup/USB>

USBPcap: <http://desowin.org/usbpcap/>

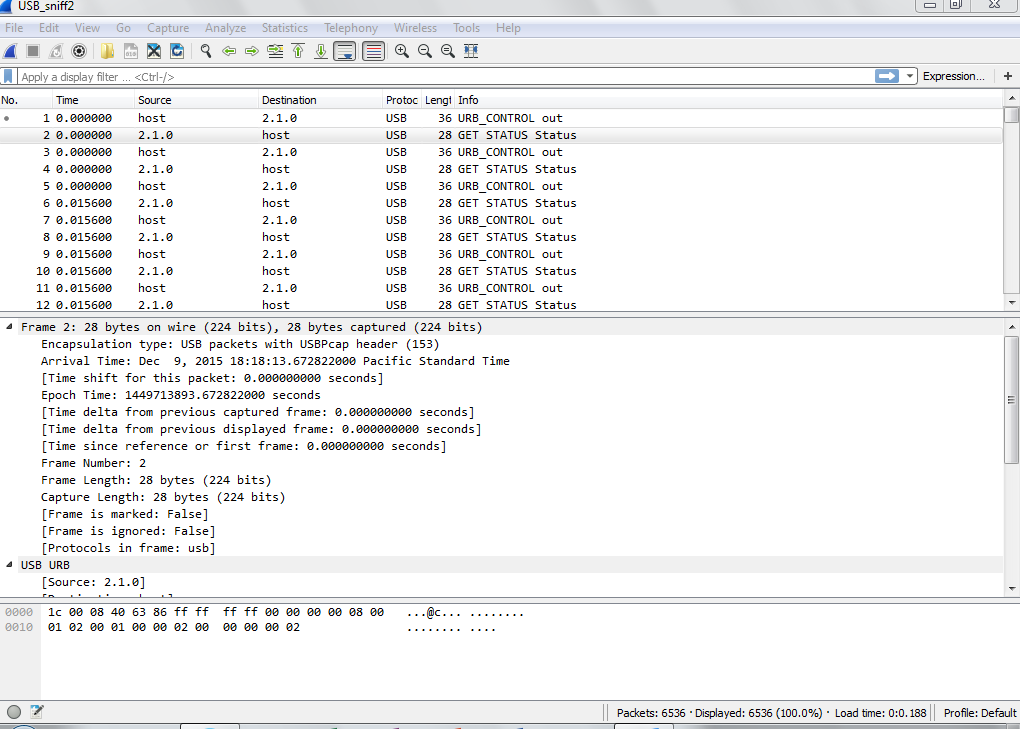


Figure 2: Analyzing USB packet data captured with USBPcap in Wireshark

Software: (continued)

RealTerm:

Serial console for Windows. Can be used to transmit/receive bytes with I/O board, capture serial data, etc.

<http://sourceforge.net/projects/realterm/>

Hardware platform: Evolution Robotics servo controller.

(3 total) Internal boards:

(1) RS-232 → Serial comm board

(2) PIC-SERVO v.4, PIC\_ENC v.1 boards

(2) Motors provided, mounted on base chassis

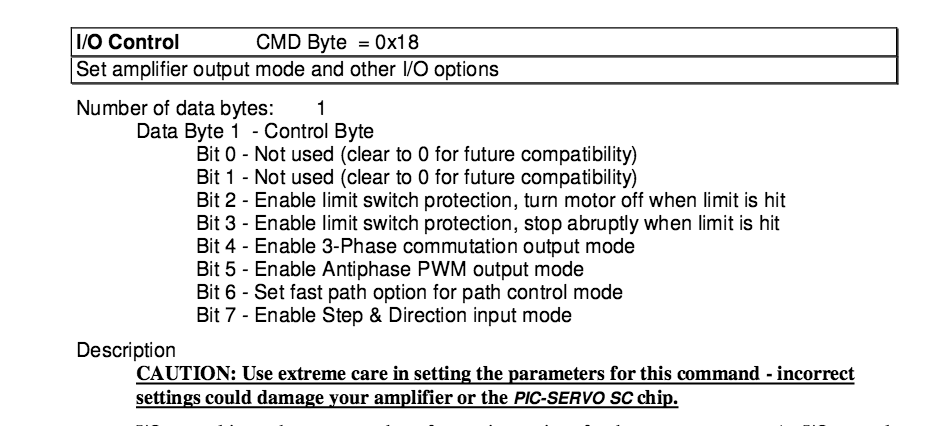


Figure 3: Excerpt from CMD outline in documentation from J.R. Kerr

Power for servo control unit:

9V, 3-4 Amp.connector polarity GND -> - ( + )

Operating System:

Windows 7 Professional (Purely due to Kinect development)

Results:

We have created operating code for the Kinect using both gestures and transparent button lay-overs to exercise control programmatically. This code can be coupled with any

Running the NMC test program not only allowed us to verify the functional state of the motors and control unit, but also to sample the communications data providing unique insight into the protocols. This advantage, paired with the outlined information in the documentation from the manufacturer this allows for a more complete view of the protocol.

As of current, the robot body is not mounted on the base as it is unclear whether or not the base will indeed be used. However, based on the documentation now located and the to verify the USB communication in transit, we do believe it possible, (though non-trivial), to build an interpreter program and revive the hardware's usability. Given that multiple of these devices exist in the lab, it may prove a worthwhile undertaking. Given this newly assembled information, toolset and workflow, we estimate approximately 40-60 man hours would be necessary for that section of the project to completion, without attempting to repair the deprecated software from the manufacturer.