Robo Waveplates Read Me

Version 0.1

Before You Start!

Please follow instructions from PCB Motor to set up your motors and familiarise yourself with commands via the TwinDemo software or Termite. Communication with the motors is possible using any terminal emulator. While PCB Motor does not supply software for Linux and Mac, communication is possible using a terminal emulator, such as CuteCom in Linux, and also through Matlab's Instrument Control Toolbox.

Overview

RoboWaveplates is designed to work with a PCB Motor kit using two external 60 mm stator motors with 200 line encoders. If you are using different motors, you may have to modify some of the code. RoboWaveplates assumes that each motor is fitted with an auxiliary QRE1113 sensor that outputs a low voltage when reflection is high, as well as an ADuC7020 micro-controller with the motor 3 and motor 4 auxiliary sensors connected to analog input channels 1 and 2, respectively. RoboWaveplates will work without the auxiliary sensors and micro-controller, but the zeroing function will not be available, and the micro-controller initialisation code must be commented out. It is also assumed that you have the Matlab Instrument Control Toolbox.

RoboWaveplates should work on any operating system, so long as the system is able to treat a USB connection as a virtual COM port. This version of RoboWaveplates was tested on Ubuntu 12.04, Windows 7, and Windows XP.

Setup

Hardware Setup

Please follow instructions from PCB Motor for assembling the motors, and refer to our accompanying paper for details on the mounting.

Channels 1 and 2 of the micro-controller are used for auxiliary sensor input. This leaves another two channels free for other applications. Conveniently, the micro-controller can run off the same 5 V power supply as the PCB motors. AVDD provides the power supply for the auxiliary sensors. Figure 1 illustrates the connection scheme.

Software Setup

Windows: The PCB Motor documentation will guide you through the driver installation process for Windows. Note that Windows Vista is not supported. If the motor has installed successfully, it should appear as a COM port. The microcontroller is connected to the computer via an RS232 to USB adapter cable. When plugged in without the micro-controller, it should automatically install a driver. If the computer has an RS232 port, then the adapter cable may be omitted.

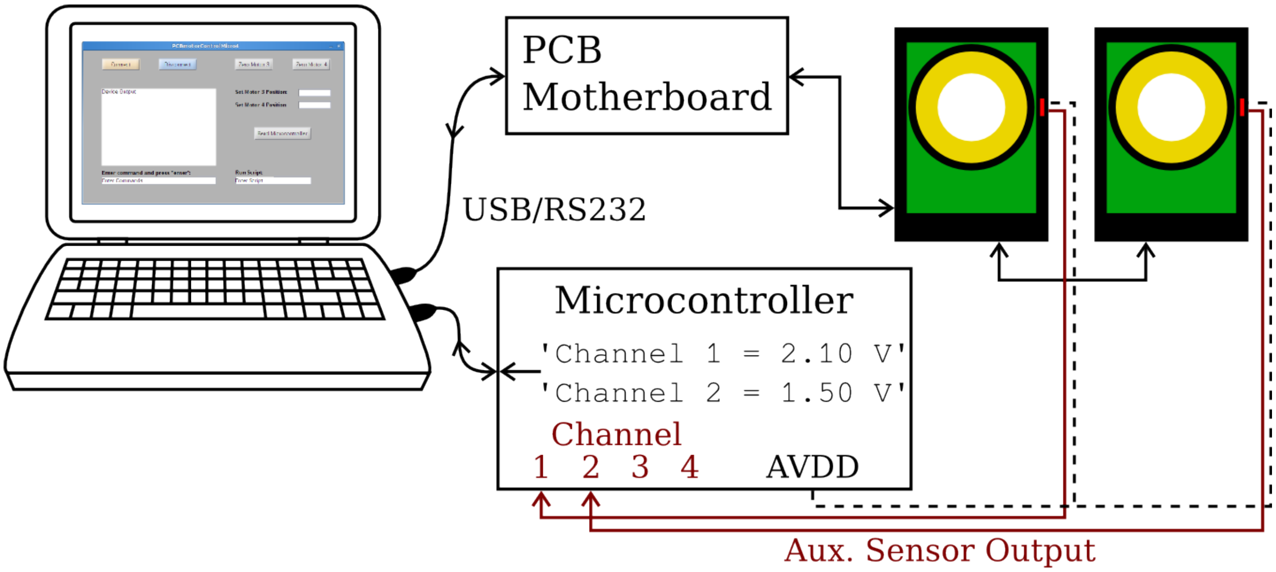

Figure 1

Linux: To establish RS232 communication over USB for Linux, please refer to the following guide. http://pensacola-tech.com/pensacola/2010/06/01/how-to-enable-usb-serial-port-adapter-on-ubuntu-2/
The motor and microcontroller will appear as /dev/ttyUSB0 and /dev/ttyUSB1.

Mac: Establishing communication with the motor and microcontroller should be possible, but has not been tested.

Programming the micro-controller

Using RoboWaveplates

1. Check which instrument is connected to which port. This is easily done using the tmtool in Matlab and scanning for serial instruments. Try connecting to the instruments. The micro-controller works using default settings, but the PCB motors require specific settings (see line 73 or the PCB Motor documentation). In Linux, you may need to run Matlab as root (e.g. "sudo matlab" in Ubuntu).

2. Before running RoboWaveplates, edit lines 57-59, 64-66, 80-82, and 87-89 (as appropriate) in RoboWaveplates.m to address the correct port for each instrument.

3. Click the "Connect" button. Type "h" in the "Enter Commands" text box and press enter. If the help file is displayed in the "Device Output" frame, then the motor is properly connected. The Device Output window is not big enough to display the entire help file.

4. Motor commands can be entered in the "Enter Commands" text box. The same command format is used as in Termite and TwinDemo.

5. Each motor can be zeroed independently using the "Zero Motor" buttons. You may need to adjust the cut-off value for the voltage sensed by the auxiliary sensors on lines 257 and 340. This value depends on the distance between the motor and reflective surface as well as the type of surface. If your waveplate mount is aluminium with a dark line indicating the zero position, the zeroing loop must stop when the voltage is high, and vice-versa for an anodised mount with a reflective strip indicating the zero position. Note that the motor moves only a single step at a time in a clockwise direction, so the zeroing procedure may take a while.

6. Motor positions may also be set using the "Set Motor Position" text boxes. Setting a position in this manner ensures that the motor always rotates clockwise. This is used to overcome an issue we observed during testing (see following section). If you are not using a 200 line encoder in half-step mode, you must edit lines 428, 435, 504, and 511.

7. To display voltages on all of the micro-controller analog input channels in the "Device Output" frame, click the "Read Microcontroller" button.

8. Additional scripts may also be run using the "Run Script" text window.

9. When you are done using the motors, click the "Disconnect" button.

Known bugs and other issues

1. Commands in text boxes are executed when pressing enter. If a command is partially written and the user clicks to a different part of the GUI, unpredictable motor movements may happen. Most likely, the motor will take a single step.

2. If the user attempts to enter a command in the "Enter Commands" text window while another function is being executed (e.g. Zero Motor), the motor sometimes starts spinning at maximum speed without stopping next time a command is issued. To stop this, disconnect the USB connector or power supply. In Linux, you may have to restart the computer to avoid the devices being assigned to different ports.

3. When the motor changes direction, it has a tendency to "miss" a step by taking a smaller step size than normal. Therefore, issuing the commands "s100" followed by "s-100", for example, will not return the motor to the original position. Instead, it will be one step off. This is a PCB Motor issue and is out of our control. The "Set Motor Position" command text boxes are a workaround to this problem.

4. When disconnecting the motor, it is important to unplug the USB cable prior to turning off the power supply. If this is not done, data in the EEPROM may be lost. PCB Motor are aware of this problem and are working on it.

5. If the motor is told to do a sweep of more than around 320 steps immediately followed by another movement, the motor moves by approximately designated amount, briefly stops, then moves again, coming to rest at the wrong location. The built-in sensor also fails to track the movement. This is a PCB Motor bug.