**CENG4480 Embedded System Development and Applications**

**Computer Science and Engineering Department**

**The Chinese University of Hong Kong**

**Laboratory 6: Motors Control**

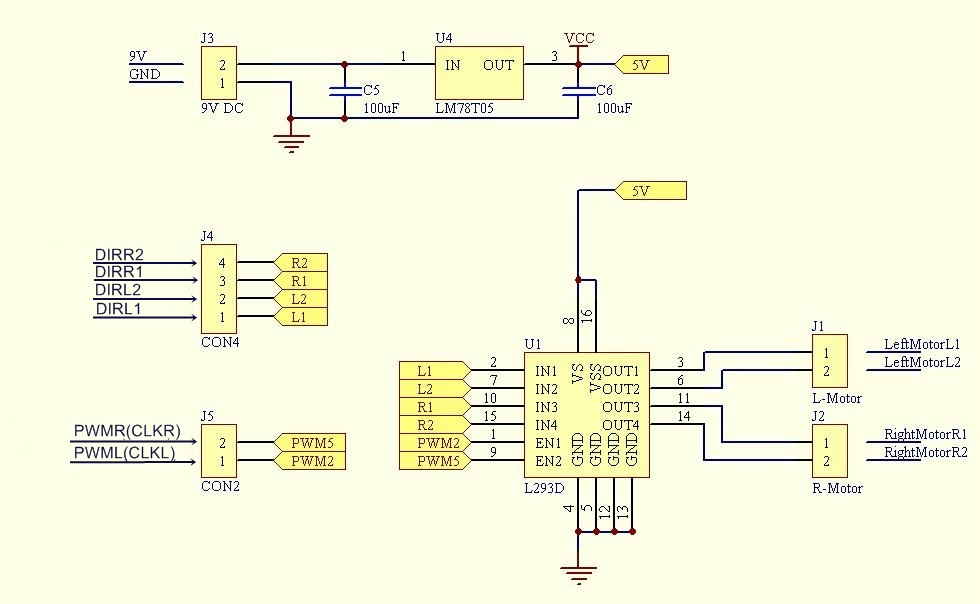
October, 2016

**Introduction**

In this exercise you will learn how to control Direct Current (DC) motors by using the Arduino microcontroller. This technique will be used in the coming Lab 7 and Lab 8.



**Figure 1. Block Diagram of Motors Control**



**Figure 2. Schematic Diagram of L293D Board**

**Objectives**

* To learn how to use motor driver L293D
* To learn how to use Pulse Width Modulation (PWM) method to control the motor speed
* To learn how to implement motor speed control by using programs in the Arduino microcontroller

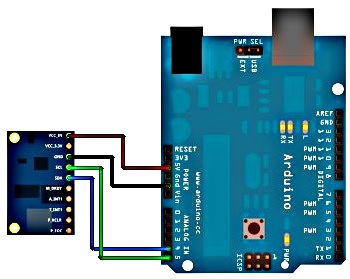
Procedures and what to submit:

Follow the procedures of each experiment. Submit a lab report sheet with your name and student ID, to the tutor after the lab. The lab report sheet should have the measurements or plots of your experiments, and answers of the questions asked in this lab manual. You may prepare the report using a computer document and use a camera to capture the waveforms and insert them in your report.

Experimental procedures

1. **Connect the IMU module to the Arduino board**

* Use dupont wires connect the VCC, GND, SDA, SDC of the IMU module to the Arduino board as shown on Figure 3. (refer to sample board found in the lab)

****

**Figure 3. Connection of IMU module**

1. **Attach the IMU module on the platform**

* Attach the IMU module on the experimental platform by using plastic tape.
* Make sure it can rotate freely along the x axis.

1. **Connect the L293D driver to Arduino board**

* Use male to female dupont wires to connect PWML, PWMR, DIRL1, DIRL2, DIRR1, DIRR2, Vin and GND of L293D driver board to Arduino as shown in Figure 1. (refer to sample)

1. **Connect motor to L293D driver board**

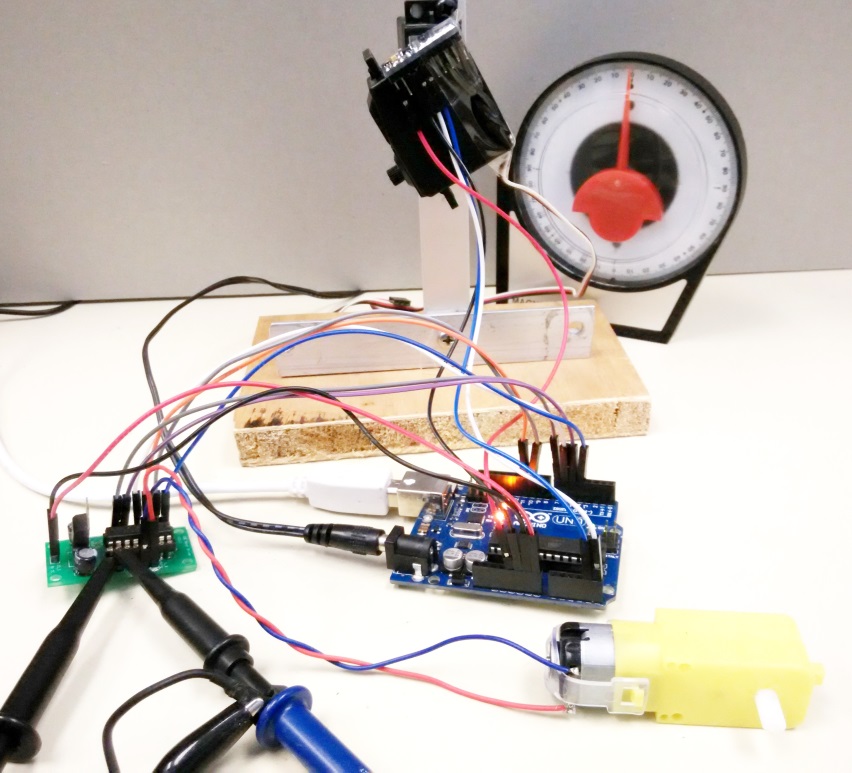
* Connect the two wire of motor to L293D driver board J1-1 and J1-2.

1. **Connect oscilloscope inputs to L293D driver board**

* Connect Channel 1 input of oscilloscope to L293D driver board J2-1
* Connect Channel 2 input of oscilloscope to L293D driver board J2-2

1. **Connect USB cable from PC to Arduino board**
2. **Connect power adapter to Arduino board**

* The whole setup of the experiment is as shown in Figure 4.



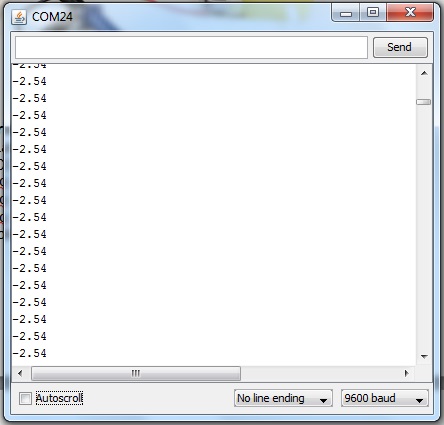
**Figure 4. Experiment setup for Lab 6**

1. **Upload the program Lab6.ino to Arduino board**

* Double click *Lab6.ino*
* Observe the COM port number from Device Manager
* On the Arduino IDE **Tools**, **Serial Port** select the correct port number
* On the Arduino IDE **Tools**, **Board select** the **Arduino Uno**
* On the Arduino IDE press the Upload button
* Wait for the uploading program finish

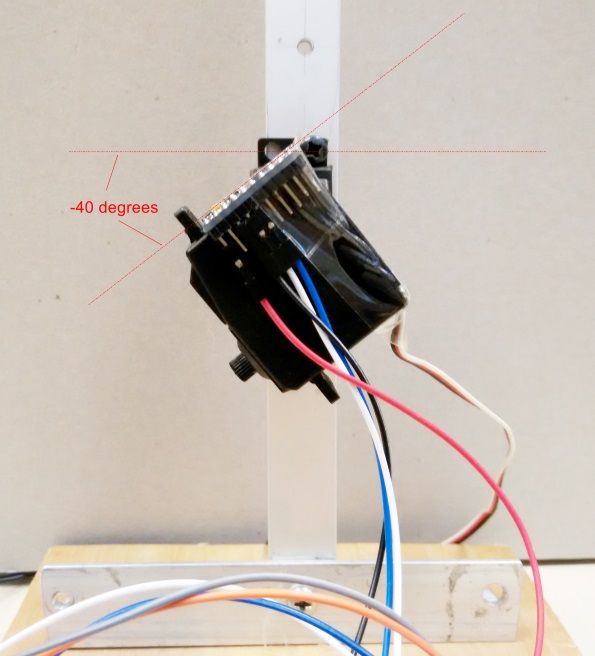
1. **Calibrate the 0 degree of the platform**

* Rotate the platform to the 0 degree position (horizontal)
* On the Arduino IDE **Tools**, select **Serial Monitor**, you should see the following window



* Use the number displayed on the COM window as the offset value (in this example = -2.54) and change the offset value in the program to this value (for example: #define offset -2.54)
* Upload the program to the Arduino board again
* You should see the value displayed on the COM window is now close to zero

1. **Rotate the platform to -40 degrees, record the waveform on the oscilloscope screen and measure the duty cycle of the waveform. What is the direction (clockwise ‘+’ or anticlockwise ‘-’) and speed of motor?**

****

1. **Repeat step 9 for -30, -20, -10, 0, +10, +20, +30 and +40 degrees. Fill in the following table. Enter your results in the lab report sheet.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Platform Angle**  **(Degree)** | **Picture of waveform on oscilloscope** | **PWM Duty cycle** | **Direction**  **(+ or -)** | **Speed** |
| **-40** |  | **100%** | **+** | **++++** |
| **-30** |  |  |  |  |
| **-20** |  |  |  |  |
| **-10** |  |  |  |  |
| **0** |  |  |  |  |
| **+10** |  |  |  |  |
| **+20** |  |  |  |  |
| **+30** |  |  |  |  |
| **+40** |  |  |  |  |

**END**