CMPE-110
Intro to Computer Engineering
Fall 2019
Laboratory Exercise Eight
Robot Servos and Encoders

This laboratory exercise investigates the use of servos to make the robot move and make use of encoders to determine how far the robot actually moved. The investigation begins with a provided sample program.

Robot Usage

A few key reminders:

1) This robot will be with you for the next several weeks. Treat it with care!
2) Ensure that the robots power supply is off
3) Please be careful plugging in the USB cable to avoid damaging the robot.
4) If you do not know what code is in the robot, you want to be sure you are holding the robot (securely and off any surface) when you power on the robot.
5) While writing code (i.e. the robot is sitting on the workbench), please turn off the power to the robot to preserve the batteries.

Controlling servos

Many mobile robots use hobby servo motors to drive their wheels. Servo motors are small DC motors with an integrated gearbox and built-in circuitry to rotate the servos' output shaft to specific position and hold that position. A typical servo is controlled by periodically pulsing the servo's control signal at a specific frequency (e.g. 50Hz).

As you recall, period T=1/frequency, so for a servo that needs a 50Hz frequency, the pulse rate is 1/50Hz or 20 milliseconds (ms). This pulse going into the servo is a square wave that is high for only a portion of the 20ms period (e.g. 1ms to 2ms), and low for the rest of the time. The speed of the servos is controlled by varying the duration the pulse is high, this is called pulse width modulation (PWM).

Our servos are connected to a DRV8838 motor driver chip on the Romi 32U4 board. This chip takes care of the PWM signal, so the code just sets a speed value from 0 to 100. Another output is used to set this direction to forward or backward. A value of 0 will stop spinning the motors. Sending a 0 to these motors stops sending the PWM signal.

Determine robot normal speed

You will determine how far the robot travels, in 2 seconds, at different speeds. You will need a measuring tape (one can be found at the front of the room). This will help you determine a good speed for controlling your robot, which you will need for the robot project in later labs.

1. Start the Energia program. We should find in in the Window's Program list, or search for it.
2. Select menu item File→New.
3. Replace all the default code that appears in this new sketch with the code from the Lab8_Speed.ino.txt file on myCourses (Ctrl-A will select all the code, then hit Delete).
4. Select menu item File→Save As… menu item. Navigate to a folder on your thumb drive in which you want to create the new sketch and select that folder.
5. Enter Lab8_Speed for the File name and then click Save.
6. Click the Verify button, then the Upload button to load the sketch onto the robot.
7. Measure distance for speed 30:
    a. Place the robot on the ground and place the tape next to it so you can measure the distance it travels.
    b. Press the 'Start' button on the Arduino board.
    c. When the robot stops moving, measure how many centimeters it traveled and enter that on your worksheet.
8. Repeat previous step for all speeds (program will automatically increment speeds from 30 to 80, in steps of 10 (i.e. you will take 6 total measurements).

Encoder

Now you will learn about the wheel encoders.

First, let's determine how many encoder counts you will get per wheel revolution:

1. Select menu item File→New.
2. Replace all the default code that appears in this new sketch with the code from the Lab8_Encoder.ino.txt file on myCourses (Ctrl-A will select all the code, then hit Delete).
3. Select menu item File→Save As… menu item. Navigate to a folder on your thumb drive in which you want to create the new sketch and select that folder.
4. Enter Lab8_Encoder for the File name and then click Save.
5. Click the Verify button, then the Upload button to load the sketch onto the robot.
6. Open the Serial Monitor.
7. Rotate the wheel one complete revolution, being careful to not spin it too far. As you spin the wheel, the count will appear in the Serial Monitor. If you want to start over, just hit the 'Reset' button on the Arduino board.
8. You may want to do this a couple times to make sure you have an accurate count. Once you have the count, enter it on your worksheet.

Now, use the tape measure to determine the number of counts needed to travel 30 centimeters, by measuring the outer circumference of the wheel.
☞Enter the numbers on your worksheet.

Drive

Now you will train your robot to drive in a 30cm square.

First, you will need to train your robot to turn 90 degrees. Start with Lab8_Drive.ino.txt from myCourses. In addition to fwd()/rev()/turnLeft()/turnRight()/stopMotors()/wait(), there are three new methods: waitEncoderLeft(), waitEncoderRight() and clearEncoders(). Instead of using wait() to just wait a certain amount of time, these new methods will allow you to wait until a certain number of encoder counts have occurred. Use clearEncoders() to clear the counts before you start to look at the counts again.

With the speed set to 120, determine how many counts are need to get a 90 degree turn (i.e. in loop(), modify the parameter for waitEncoderLeft()).

Each press of 'Start' will execute the commands. This will allow you to press the button 4 times to see if the robot did a complete 360.

☞Enter the count on your worksheet.

Now train your robot to drive 30cm. In loop(), replace the call to turnLeft() with fwd(), and replace the parameter for waitEncoderLeft() to use the value you calculated earlier in this lab. Tune the count until you robot drives 30cm.

☞Enter the count on your worksheet.

Now update the sketch so you robot will drive in a 30cm square.

Demonstrate this to your instructor/TA to get your 'Drives 30cm square' worksheet line initialed.

Submission

One per team—*make sure team member names are written on each item*:

___ Completed laboratory worksheet.
___ A printed copy of portion of the 'Drives 30cm square' Arduino sketch that you changed (e.g. the file comment block and loop() function; no need to print the code that was provided to you); also, do not forget to comment your code.
___ Make sure the robot's toggle switch is in the OFF (left) position.
___ Robot was put away.