

## Laboratory Worksheet for Control Algorithm and Type Casting

The following is an exercise in the functionality of the speed control algorithm and the importance of using proper type casting in the program. The speed control algorithm can be written as:

$$\text{temp\_motorpw} = \text{motorpw} + (k_i + k_p) * (\text{desired} - \text{actual}) - k_p * \text{prev\_error}$$

$$\text{prev\_error} = \text{desired} - \text{actual}$$

temp\_motorpw is the calculated pulsewidth to be implemented;

motorpw is the pulsewidth from the last calculation;

$k_i$  is the integral gain constant of the control algorithm (use a value of 150 for the calculations);

$k_p$  is the integral gain constant of the control algorithm (use a value of 150 for the calculations);

desired is the reading from the speed potentiometer;

actual is the latest speed measurement;

prev\_error is the previous calculation of the difference between desired and actual speeds

Case 1: motorpw=10000, prev\_error=-20, desired=15, actual=35 (car is going too fast)

$$\text{temp\_motorpw} = \underline{\hspace{2cm}}$$

Case 2: motorpw= 10000, prev\_error=20, desired=35, actual=15 (car is going too slow)

$$\text{temp\_motorpw} = \underline{\hspace{2cm}}$$

Case 3: motorpw=1000, prev\_error=-20, desired=5, actual=25 (car is going too fast)

$$\text{temp\_motorpw} = \underline{\hspace{2cm}}$$

Case 4: motorpw=63000, prev\_error=20, desired=35, actual=15 (car is at full power, but is running slower than desired speed )

$$\text{temp\_motorpw} = \underline{\hspace{2cm}}$$

- Consider the four cases given above and calculate the appropriate temporary motor pulsewidth for each.
- Once you have completed the calculations, go to the *Sample Code* page on the Embedded Control website (<http://litec.rpi.edu>) and copy the sample code pertaining to *Type Casting and Algorithms* to a new *Program* sheet. Compile and download the code to the EVB.
- You should see the results of each version of the control algorithm equation (shown below) displayed on your screen. Note the different ways of type casting.

Which version should be used for your *Smart Car*? Why don't the others always work?

/\* 1st control algorithm equation \*/

```
temp_motorpw = motorpw + (kp+ki) * (desired - actual) - kp*prev_error;
```

/\* 2nd control algorithm \*/

```
temp_motorpw = motorpw + (int)(kp+ki) * (desired - actual) - kp*prev_error;
```

/\* 3rd control algorithm equation \*/

```
temp_motorpw = (long)(motorpw + (kp+ki) * (desired - actual) - kp*prev_error);
```

/\* 4th control algorithm equation \*/

```
temp_motorpw = motorpw + (kp+ki)*(int)(desired - actual) - kp*(int)prev_error;
```

/\* 5th control algorithm equation \*/

```
temp_motorpw = (long)motorpw + (long)((kp+ki)*(desired - actual)) - (long)(kp*prev_error);
```

/\* 6th control algorithm equation \*/

```
temp_motorpw = (long)motorpw + (long)(kp+ki) * (long)(desired - actual) - (long)kp *
(long)prev_error;
```