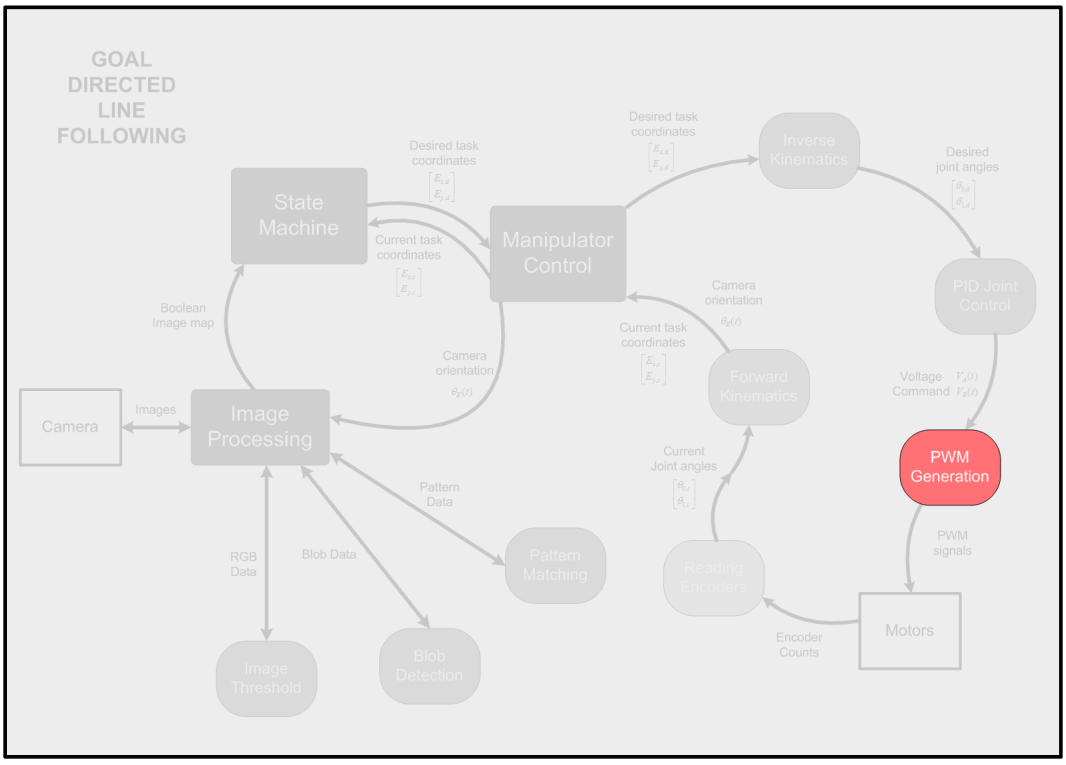
Pulse Width Modulation

Topics Covered

- Pulse Width Modulation (PWM).
- Using LabVIEW™ to develop a Pulse Generator VI.
- Using PWM to regulate the dimness of LEDs.
- Exploring PWM to regulate motor speed.

Prerequisites

- The QNET Mechatronic Systems is set up according to the Quick Start Guide.



1 Background

Pulse Width Modulation (PWM) is a technique used to regulate the effective voltage provided to a motor, thereby changing it's speed. Changing the voltage across a motor using a potentiometer does not work because the motor resistance is much smaller than the potentiometer, and the latter uses up all the power delivered by the voltage source. The Duty Cycle of a pulsing wave is the percentage duration for which the pulse is 'on'. Consider a pulsing wave $V = V(t)$ with Duty Cycle D and time period T s. Using Mean Value Theorem, the average value of this waveform over one period is:

$$\bar{V} = \frac{1}{T}\int_0^T V(t)dt \tag{1.1}$$

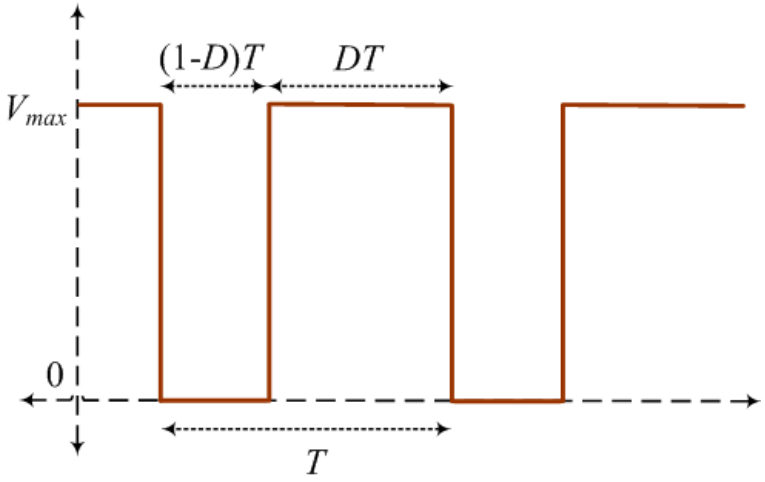


Figure 1.1: A Pulse Waveform

If the pulse wave has a maximum value V_{max} for DT s, and minimum value of V_{min} for $(1-D)T$ s, the average value from Equation 1.1 becomes:

$$\bar{V} = DV_{max} + (1-D)V_{min} \tag{1.2}$$

In the case of the QNET Mechatronic Systems, the maximum voltage V_{max} is 24V. The minimum voltage V_{min} is 0V. Thus, the average motor voltage is,

$$\bar{V} = DV_{max} \tag{1.3}$$

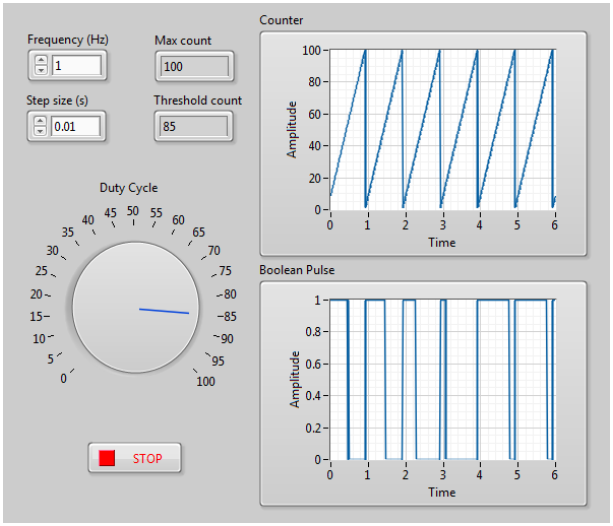
2 In-Lab Exercises

2.1 Pulse Generation

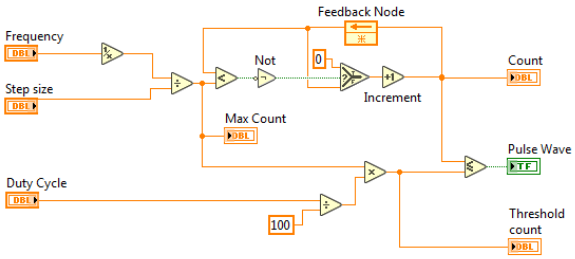
You can create a manual pulse generator with any desired frequency and duty cycle. In your system's explorer, browse to your project folder. In your system's explorer, browse to the your project folder. Under the folder `Subsystems | Investigation Controllers | PWM Generation`, and open `PWM Generation.vi`.

Note: Ensure that the file is opened through your system's explorer, and not `Mechatronic Systems.lvproj`. This will make sure that the processor from your CPU is used, and not that on the NI ELVIS RIO Control Module.

The front panel should look similar to that in Figure 2.1a. Browse to the block diagram, and open the sub-VI labelled `Pulse Generator.vi`. Complete it according to the block diagram shown in Figure 2.1b. With a step size input, this sub-VI steps a counter value from 0 to a Max Count, which depends on the Frequency. The Threshold count is the product of the Max Count and Duty Cycle. This pulse generator creates a TRUE signal whenever the counter value is below the Threshold count, and FALSE signal otherwise.



(a) Front panel



(b) Pulse generator

Figure 2.1: LabVIEW™ VI to generate a pulse signal according to desired specifications

1. Back in `PWM Generation.vi`, what does changing the Duty Cycle dial do?
2. Set the Frequency to 1 Hz, and the Step size to 0.01 s. Run the VI and attach figures to show the counter and boolean pulse for various values of the Duty cycle dial. Does the response match what you expected?

3. Set the Frequency to 1 Hz, the Duty Cycle to 50%, and the Step size to 0.01 s. Run the VI, and change the Frequency to 2 Hz at some point. Attach your response. Does the signal vary as expected?

Close the files `Pulse Generator.vi` and `PWM Generation.vi`.

2.2 PWM Counts with the QNET Mechatronic Systems

In the project `Mechatronic Systems.lvproj`, under `Quanser ELVIS RIO | Subsystems`, open `Pulse Width Modulation.vi`. Select the PWM on LEDs page in the Tab Control.

1. Set the Frequency to 1 Hz, and the Duty Cycle to 50%. Run the VI. Does the behaviour of the LEDs expected with varying Duty Cycle values?
2. Stop the VI. Set the Frequency to 1000 Hz, and vary the Duty Cycle from 0 to 100%. Run the VI again. What do you observe?
3. Stop the VI, and switch to the PWM on Motors page. Describe what the Voltage to PWM Counts gain does, given that a maximum count of 2000 is used for the maximum motor voltage of 24 V.

Note the sub VI `H-bridge.vi`. Instead of setting a command of 0 counts when the motor is idle, the PWM command at both ends of the motor is set to high (2000 counts). This prevents ground bounce. When a 6 V signal is desired, `H-bridge.vi` drops the PWM count by 25% (6 out of 24 V), to 1500, while the other is maintained at 2000. It also helps reverse motor-direction, i.e. to apply -6 V, the same count commands are applied at the opposite terminals.

In the project `Mechatronic Systems.lvproj`, under `Quanser ELVIS RIO | Drivers`, open `ELVIS-RIO Customized FPGA.vi` and take a look at `Simple PWM.vi` under DIO/PWM, as seen in Figure 2.2. This is very similar to the Pulse Generator designed in section 2.1. While the motors can be controlled with a 1 KHz frequency PWM at the Processor, this is difficult to achieve consistently while carrying out other tasks such as Image Processing in parallel threads. Carrying out PWM on the FPGA can give consistent motor control due to it's faster speeds, and allow the processor to carry out more important tasks, such as image processing and path planning.

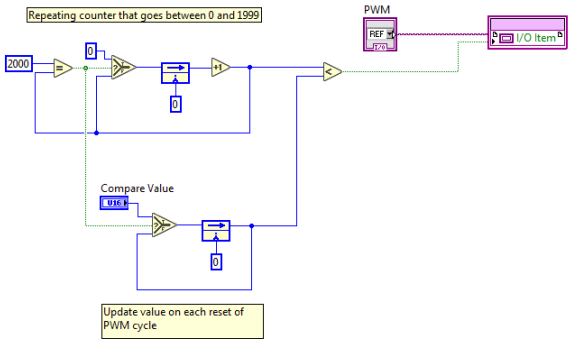


Figure 2.2: Pulse Width Modulation in the QNET Mechatronic Systems's FPGA code

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com