Concurrent Velocity & Position Control

Sheshan Patel

Submitted
April 27, 2021

Introduction

The purpose of this document is to give a brief overview of the basic functionality of the device design alongside a description of the design process. To begin with, the device created can control the position and velocity of an encoder/decoder with the aid of peripherals such as the PWM (Pulse-Width Modulation) which provides the direction and duty cycle at which the motor should move, Position Control Peripheral, which constantly compares the current position against the desired position and outputs the direction and duty cycle which the PWM should send to the motor, and a Velocity Control Peripheral, which performs the same functionality as the Position Control Peripheral but it also takes in the duty cycle specified by SCOMP as an additional input. We prioritized position and velocity control done by the hardware and keeping the software as a bridge that connects the different peripherals. This allowed us to not only meet the required specifications but also go beyond by controlling position and velocity concurrently.

Device Functionality

Position Control

The position control peripheral can be accessed using the IO address &H009. This peripheral is receiving the motor position directly from the Quadrature Decoder peripheral; the only necessary interaction between the user and the position control peripheral is to input the desired position and then the peripheral outputs the results to the Pulse-Width Modulation (PWM) peripheral, using IO address &H0F2, commanding the motor to move.

Velocity Control

The velocity control uses two peripherals: Velocity Parse and Velocity Control, using IO addresses &H00A and &H008 respectively. Velocity Parse takes in the desired speed given by the user and sends it to Velocity Control. The results of Velocity Control, containing the direction and duty cycle, must be outputted to the PWM peripheral, where it tells the motor how to move. Note, if the desired speed is lower than 30 rpm, the motor will not move due to it being too low of a duty cycle input.

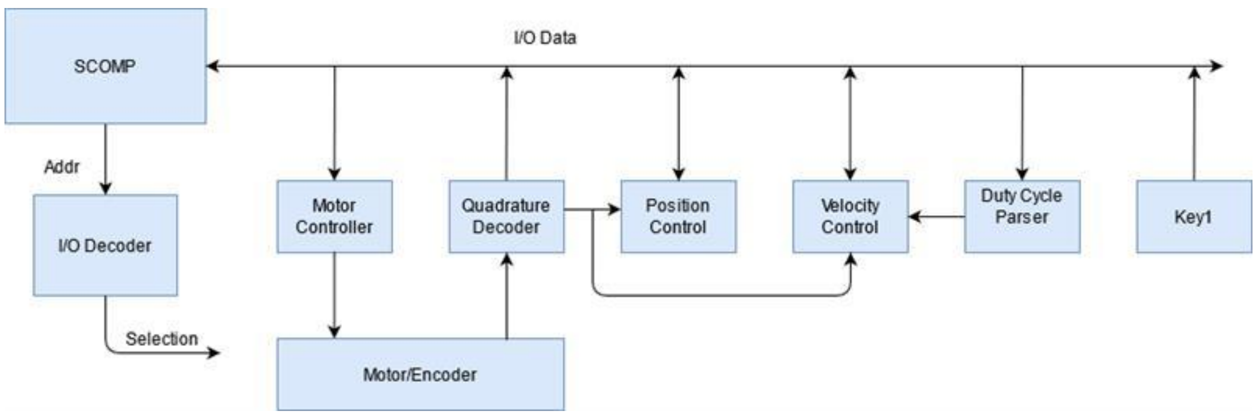

Figure 1. System peripheral overview.

Additional Functionality

The RPM peripheral, using IO address &H006, displays the velocity of the motor. Within our demonstration code, there are subroutines for concurrent position and velocity control that calculate velocity when given a time input to move over two positions.

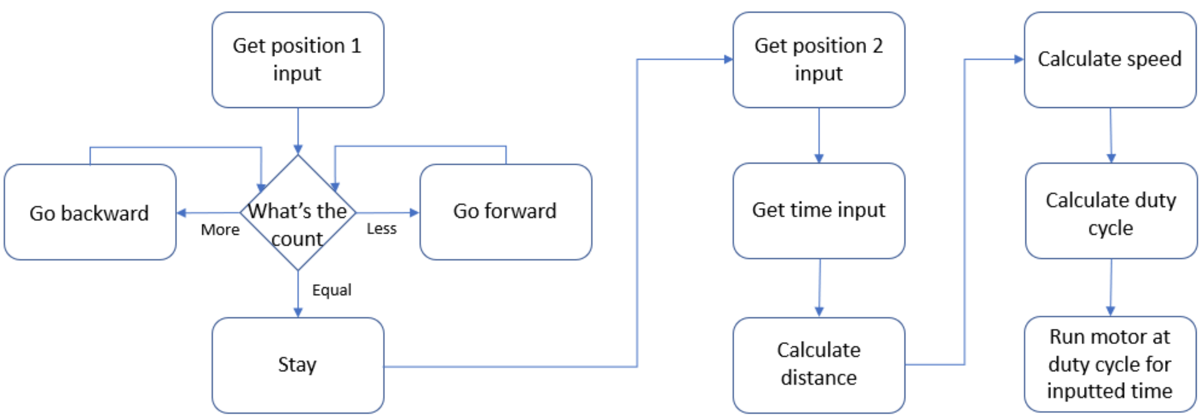

Figure 2. Position and velocity control procedure for device demonstration.

Design Decisions

Hardware and Software

Our team completed the requirements within peripherals so that the ability to add additional features within the software will be made simple. Once creating the individual peripherals to control velocity and position, we worked on the ability to use these controls concurrently. The only necessary software for the motor control implementation is to input to and output from these peripherals.

Modifications

The most significant modification our team made during this process was changing our approach to velocity control to allow for simultaneous position and velocity control. We originally intended to use bang-zero-bang control for both velocity and position control, however, our team was concerned with the effects of using the controls simultaneously with this method. Bang-zero-bang is a type of motor controller that switches between two states (sending maximum or minimum PWM commands) based on how close a current value, in our case the value of speed or position, is to the desired value. To ensure that concurrent controls would not interfere, we changed velocity control to take in the desired speed and matched it to the correct duty cycle.

Conclusions

Overall, the device surpassed all the requirements set, however, as with all designs, there can be improvements. Our device fails at moving at low RPM, so one way to overcome it is to have the motor start at 100% duty cycle and then decrease it. Another improvement would be to allow more than 11 bits input and allow the position to be negative. Overall, our device is a great initial design, but there’s room to improve upon.