Michael Eller

ECE 3430

Lab 8

26 October 2015

Rotary Encoder

Design Approach

Encoder:

Using the debouncing code created in the previous lab, the rotary encoder was defined as two switches and then continuously debounced in the main loop. When the encoder completes a physical click, there are actually four transitions that occur. The screen capture below shows a full clockwise rotation.

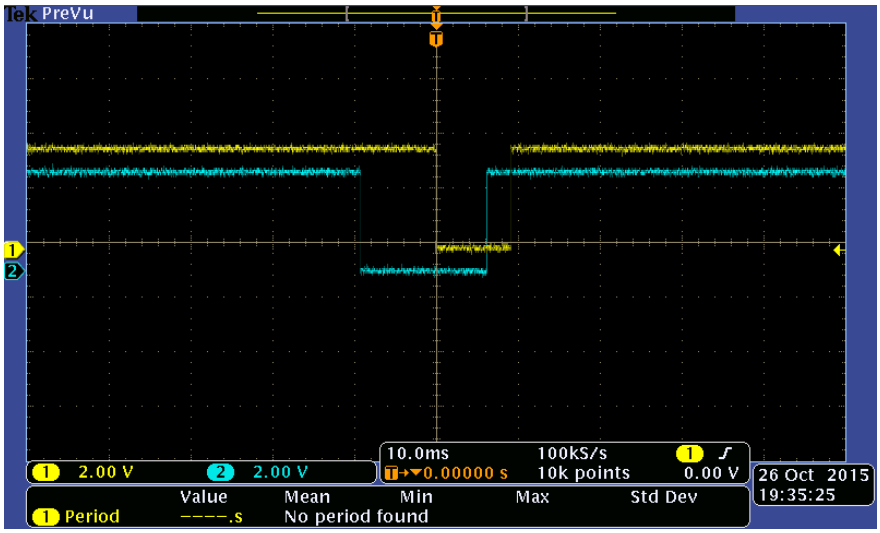

Figure 1, oscilloscope screenshot showing the four transitions of the encoder

A state machine was then created to track each transition and verify a full clockwise or counter clockwise click. Each transition incremented (or decremented if it was rotating counter clockwise) a count variable, so each physical click incremented or decremented the count by four. This count variable was used later to keep track of the value to display on the LEDs.

LED Display:

The LED display was controlled by a constant current LED driver. The driver operated by taking two bytes of data representing hexadecimal numbers and then turning on the appropriate LEDs. The main display control was performed in *UpdateDisplay()*. Using the SPI code provided and a character array of all 16 op-codes (for showing hexadecimal value), *UpdateDisplay()* used the

count variable updated by the state machine to determine the op-code to use for each 7-segment display. The count variable was manipulated to return an index to the correct op-code in the character array. Those codes were then sent using the SPI code and shown on the LED displays.

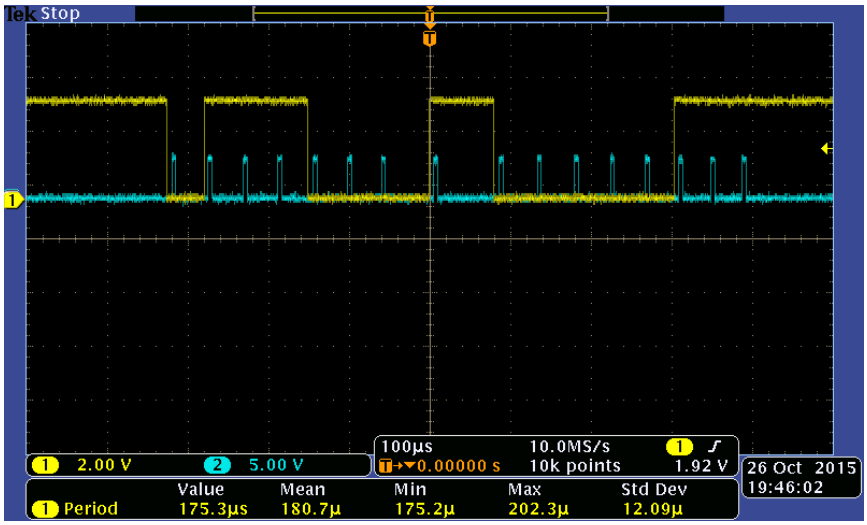

Figure 2, oscilloscope screen capture showing the SPI op-codes for 0x7F

The screen capture above shows the op-code for 0x7F. The code for F (0111 0001) was sent before the code for 7 (0000 0111) because the driver received the byte for the second 7-segment display first.

PWM:

The pulse width modulation for the green LED was completed in the same manner as the previous PWM lab. Timer A0 was unchanged except for turning on the green LED. Timer A1 interrupt was added to create the pulse width modulation. The limit of Timer A1, which turned the green LED off, was incremented when the rotor completed a full clockwise click, making the LED brighter. The limit was also decreased when the rotor completed a full counter clockwise click, making the LED dimmer.

Reset Button:

The reset button was implemented by using an interrupt service routine on pin 1.3, as in previous labs. The interrupt simply resets the display and corresponding count variable to 0x7F as well as the duty cycle (brightness) of the green LED.