Rochester Institute of Technology
Real Time and Embedded Systems

Project 6 – Voltage Indicator Using Servos

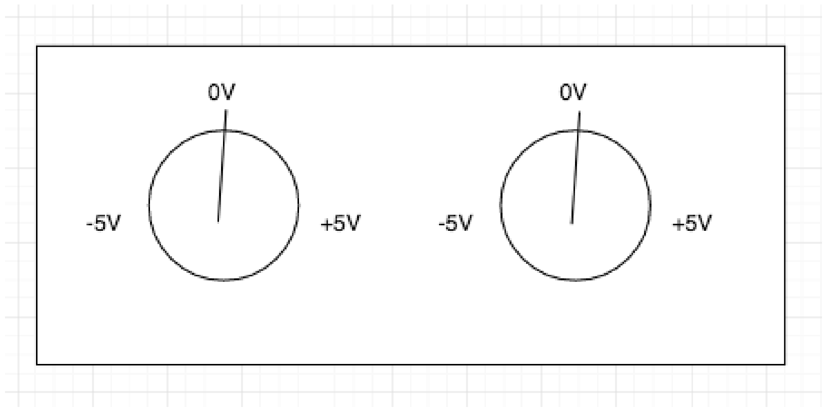
May 10, 2017

Zachary Weeden | zdw7287@rit.edu

Overview

This project is to demonstrate a communication between the QNX purple box and the STM development board. The action to demonstrate this involves the reception of a signal from a generator to the QNX system which is then processed by an onboard analog to digital converter and then passed onto the STM board which then deterministically resolves a PWM signal to drive a servo to an indicated position.

The servo's position indicated the polarity and magnitude of the input signal from the generator, more specifically, the converted A/D code seen by the STM. These ranged in magnitude from -5V to +5V. With the involvement of 2 subsystems, division of development was needed and there were efforts to determine the responsibility of each system and how they were to communicate with one another.



Servo position indicating voltage

Areas of Focus

Zachary Weeden: All of project 6

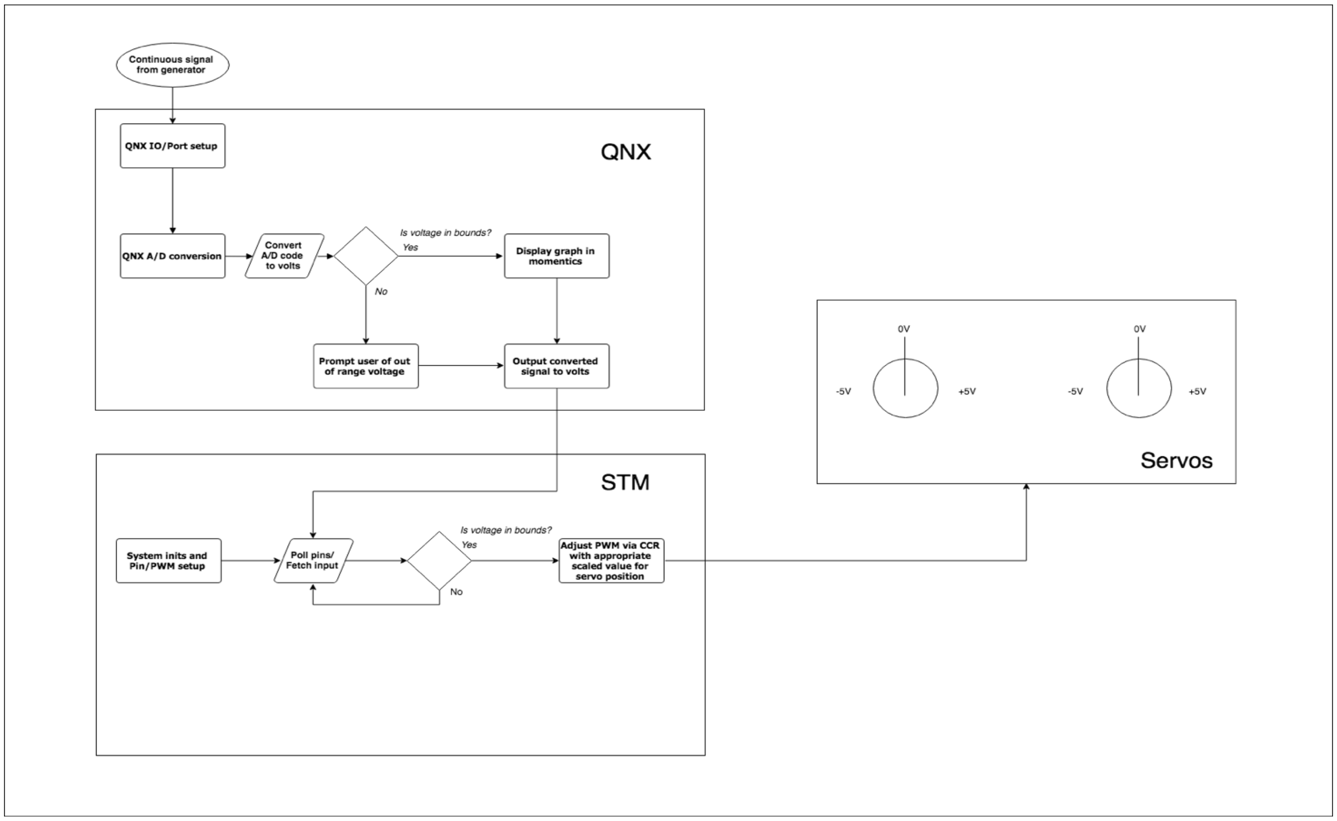
Analysis/Design

Because this project involved two different subsystems, isolation of functionality and determination of functional responsibility was required. Another design endeavor was the communications between the two systems.

Assuming that there would be a steady, continuous signal, be it square, sinusoidal, or saw-tooth wave, we setup A/D conversion functionality on the QNX system by writing and observing various handlers/pointers. As always in the QNX system, we need to get I/O access permission and to map ports desired into the address space prior to doing anything with said registers.

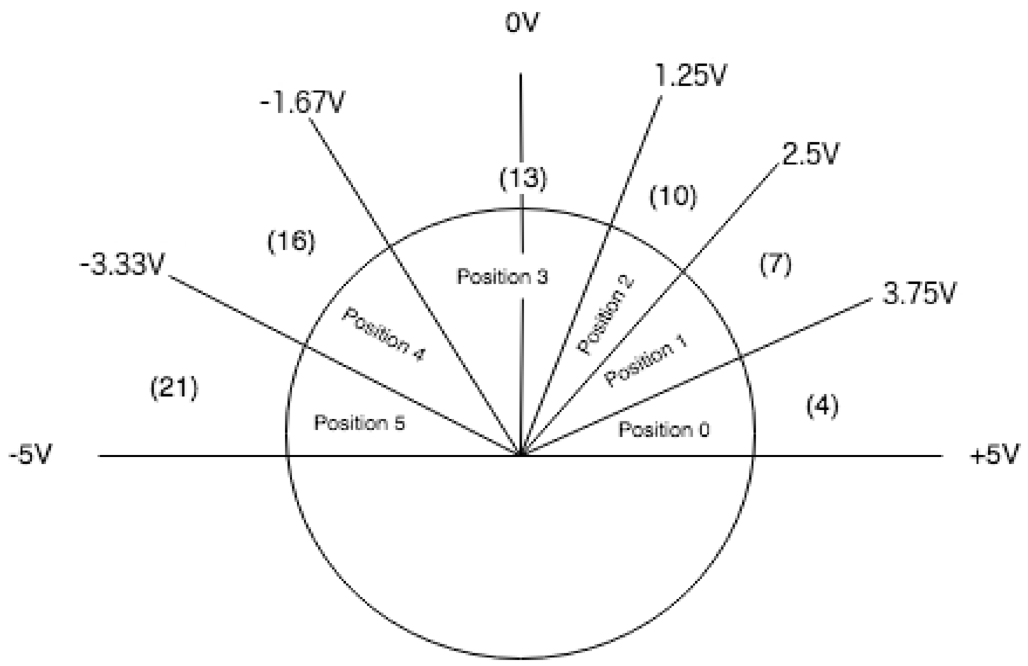
After that is all said and done we can begin writing to registers and in turn begin an A/D conversion. In an indefinite loop, we pass the necessary register handler pointers to a function that sets the proper values to the proper registers to perform a bipolar -5 to +5 A/D conversion. The resulting A/D code is then converted into a usable value, voltage. A conditional within the function logs if the voltage is out of the specified range. If the voltage is valid then a pseudo-bar graph is printed which should replicate the digitally converted analog in signal. Regardless of the validity of the voltage, the voltage is still passed through to the next system so that *both* systems can alert the user if the voltage is out of scope. It should be noted that the finer the measurements the smoother the servos movement will be at the expense of pin resources.

After the initialization of the STM input and output pins and PWM setup, the STM continually monitors selected pins for a signal, these pins are the output of the QNX system. Determination of the compare capture register's (CCR) value is based off of the input signal. This is what drives the servo's duty cycle and therefore its position. We scaled the values so that if ~-5V is seen the CCR value is inflated to its greatest value moving the servo all the way to the left. A +5V meant that the servo should be moved to the right most position, position 0. This means that the CCR was set its smallest number. 0V was translated to position 3 in the middle. Validation of voltage is done on the STM as well and turns on a red LED if out of range and does not update the CCR value.


Continuous signal from generator
QNX
QNX IO/Port setup
QNX A/D conversion
Convert A/D code to volts
Is voltage in bounds?
Yes
No
Display graph in momentics
Prompt user of out of range voltage
Output converted signal to volts
STM
System inits and Pin/PWM setup
Poll pins/ Fetch input
Is voltage in bounds?
Yes
No
Adjust PWM via CCR with appropriate scaled value for servo position
0V
-5V
+5V
0V
-5V
+5V
Servos

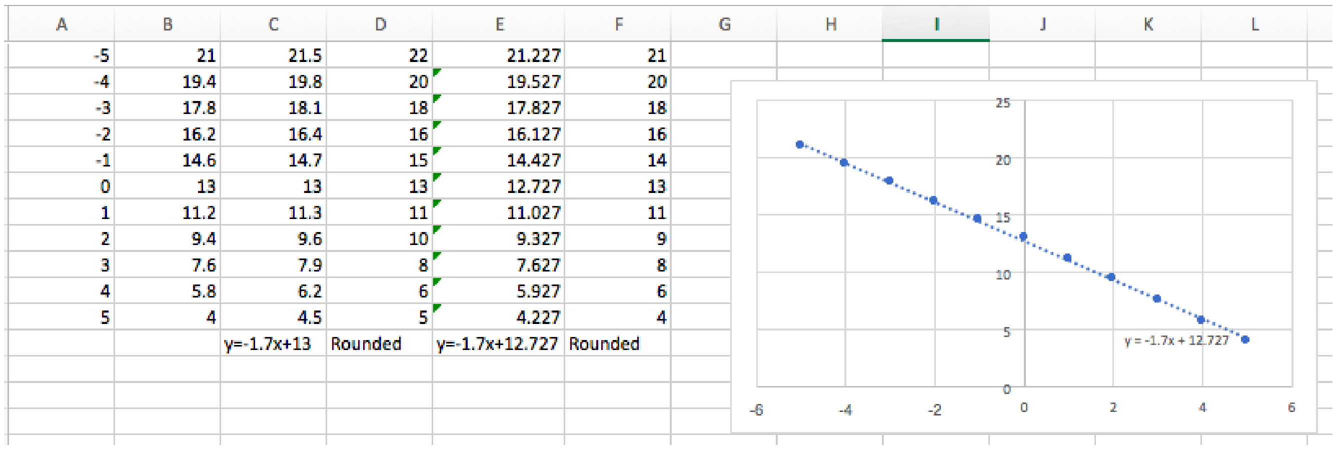
Block diagram showing general functionality and component isolation

To make the servo move, we strewn conditionals throughout the STM portion of the system. Below shows the scaling and the proportion of voltage vs. position vs. CCR value. We chose to simply divide evenly and set boundaries with typically the minimum boundary included in the current position with the maximum value included in the next position.



Servo division showing the allocation of position/CCR values compared to voltage

This design would cause for variance in the voltage and would not truly swing from position to position as it has a threshold for each position. This resulted in a "jerky" motion. To remedy this, we looked to excel and tried to resolve the value for CCR proportional to the voltage. To do this, we graphed the basic endpoints of the voltage and with averaging found the appropriate/scaled CCR values for each whole volt. A trend line was applied and then results from the equation were rounded to yield an integer value for the compare capture register.



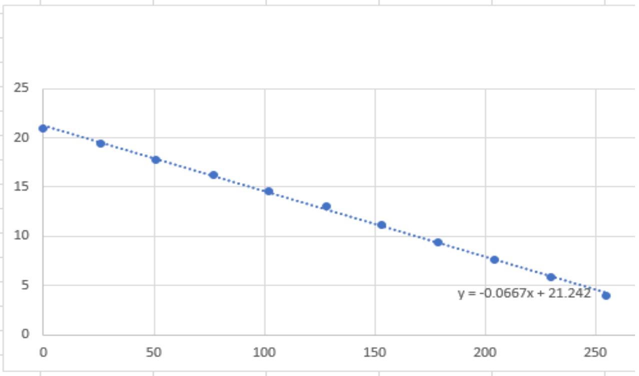
A	B	C	D	E	F
-5	21	21.5	22	21.227	21
-4	19.4	19.8	20	19.527	20
-3	17.8	18.1	18	17.827	18
-2	16.2	16.4	16	16.127	16
-1	14.6	14.7	15	14.427	14
0	13	13	13	12.727	13
1	11.2	11.3	11	11.027	11
2	9.4	9.6	10	9.327	9
3	7.6	7.9	8	7.627	8
4	5.8	6.2	6	5.927	6
5	4	4.5	5	4.227	4
		y=-1.7x+13	Rounded	y=-1.7x+12.727	Rounded

*CCR = [-1.7*voltage+12.727]*

This was a step closer to the solution but would cause for ambiguous data sent as the granularity isn't fine enough for our system. Back in the QNX portion, I chose to scale the voltage prior to output to digital I/O pins to range from 0 to 255, coincidently 1 byte; -5V represented by 0 and +5 by 255 with 0V being around 128. I went to excel to find the respective trend line to convert these numbers to appropriate CCR values. The graph below relates our new scale to respective CCR values.

Scaled +5 *25.5	Exact CCR	Rounded	Rounded
0	21	0	21
25.5	19.4	26	19
51	17.8	51	18
76.5	16.2	77	16
102	14.6	102	15
127.5	13	128	13
153	11.2	153	11
178.5	9.4	179	9
204	7.6	204	8
229.5	5.8	230	6
255	4	255	4

Rounded	Exact CCR				
0	21		21.242	21	
26	19.4		19.5078	20	x
51	17.8		17.8403	18	
77	16.2		16.1061	16	
102	14.6		14.4386	14	x
128	13		12.7044	13	
153	11.2		11.0369	11	
179	9.4		9.3027	9	
204	7.6		7.6352	8	
230	5.8		5.901	6	
255	4		4.2335	4	



*CCR = (-.0667*scaled_signal_from_pins) + 21.242*

This equation is to be implemented on the STM portion after reading the pins to converge on a CCR value.

On to the STM design, I chose to use port E and H due to their locality on the side of the board to the external ribbon cable. (More specifically pins 15 through 10 on E and 1 and 0 for H). I set them as inputs and brought in my PWM initialization used in prior labs to set PA0 in alternate function mode. The STM simply monitors the 8 pins and recomposes the signal by accumulating seen values and reconstructing the bit field. This value is returned.

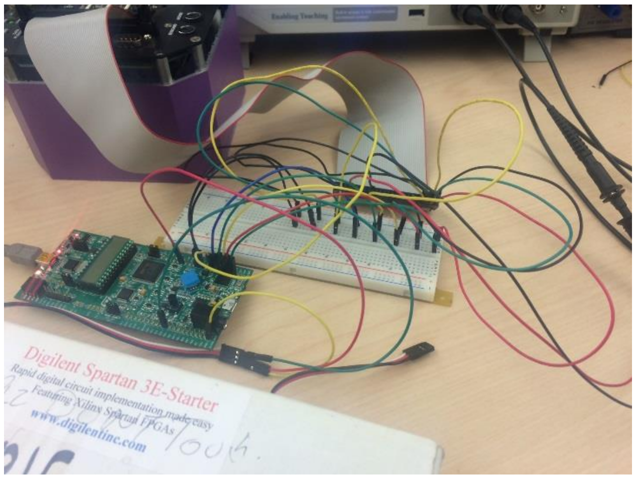
The equation is then used to convert the 255 scaled value to its matching CCR value. Because the trend line has variance and isn't 100% fit in mapping, the converted value is then rounded due to the floating point nature of the result of the equation. Because the math library wasn't an available resource to the STM, to round I casted the result as an integer only after adding .5.

Test plan

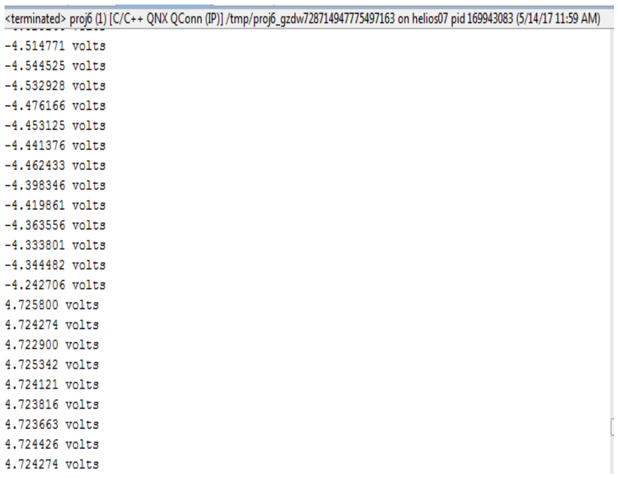
To test functionality, we tested various waveforms as specified in the document as well as at various frequencies. It should be known that at higher frequencies, anything over 1Hz resulted in the servos inability to swing the full range from -5V to +5V. Conceptually this makes sense, as it takes the servo 200ms to move 1 position. Given there are 5 positions, a full swing would take 1000ms which is the exact period of a wave at 1Hz.

Project Results

The results varied in stability depending on the frequency of the analog signal which was noted and expected. The granularity of movement in the servos could be seen as directly proportional to this factor as well as the arithmetic scaling and truncation of bits from the A/D code. Keeping the frequency low enough to provide the servo with enough time to sweep from position 0 to position 5 was another factor to the proper functionality of the application.

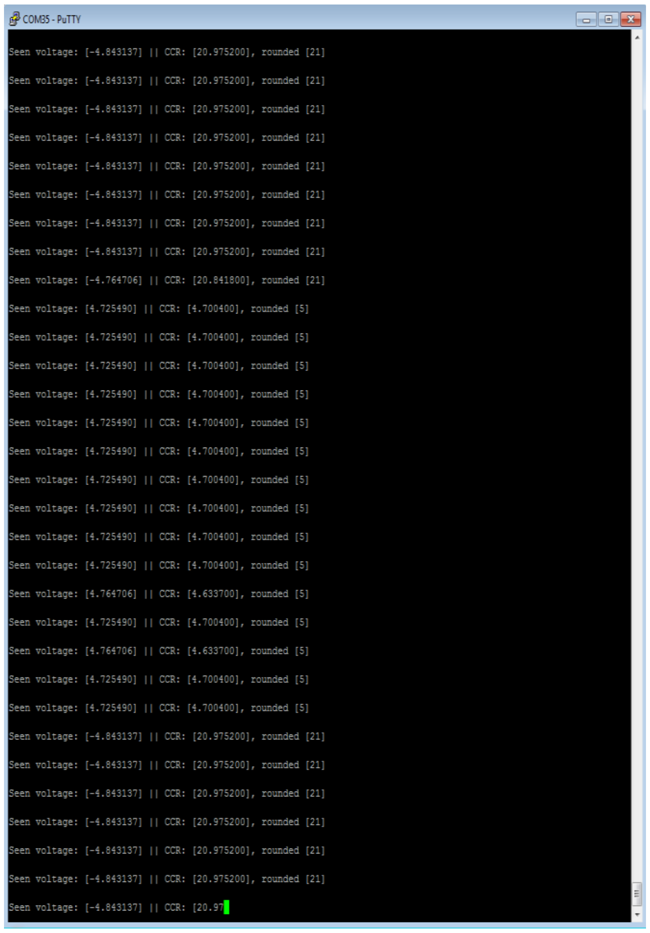

Digilent Spartan 3E-Starter
Rapid digital circuit implementation made easy
Featuring Xilinx Spartan FPGAs
www.digilentinc.com

Wiring of QNX to STM to servo

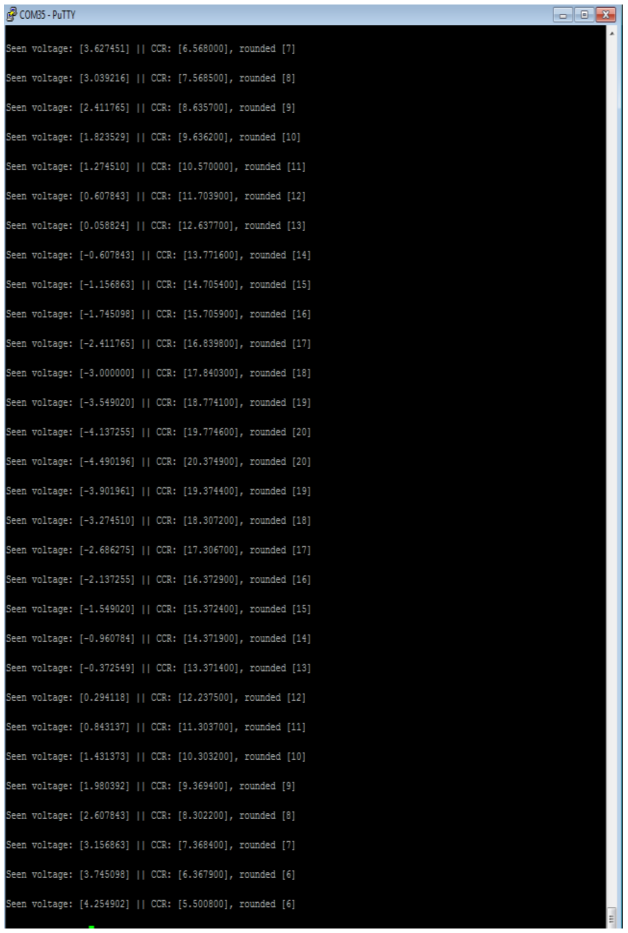


```
<terminated> proj6 (1) [C/C++ QNX QConn (IP)] /tmp/proj6_gzdw7287149477754971163 on helios07 pid 169943083 (5/14/17 11:59 AM)
-4.514771 volts
-4.544525 volts
-4.532928 volts
-4.476166 volts
-4.453125 volts
-4.441376 volts
-4.462433 volts
-4.398346 volts
-4.419861 volts
-4.363556 volts
-4.333801 volts
-4.344482 volts
-4.242706 volts
4.725800 volts
4.724274 volts
4.722900 volts
4.725342 volts
4.724121 volts
4.723816 volts
4.723663 volts
4.724426 volts
4.724274 volts
```

QNX output of A/D conversion on square wave

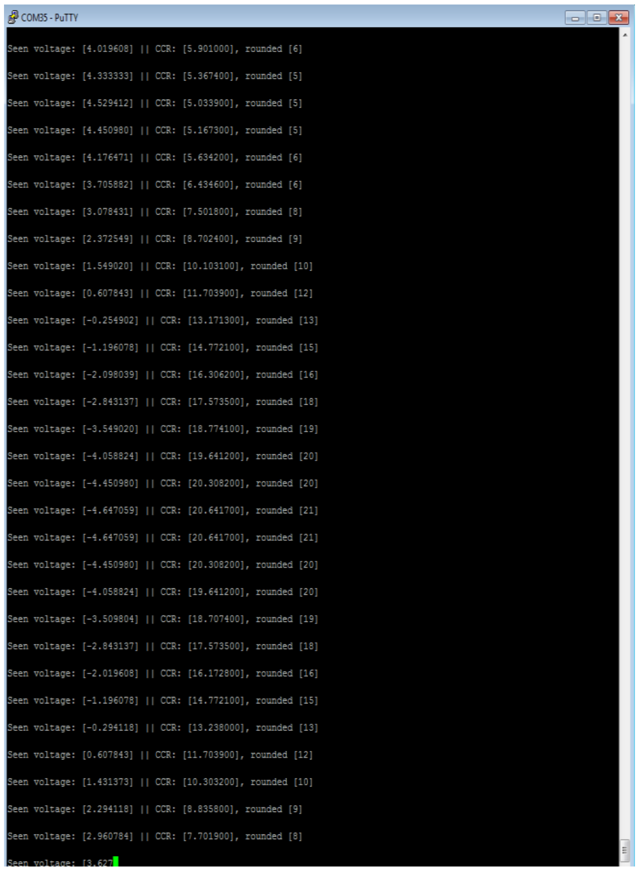


```
COM35 - PuTTY

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.764706] || CCR: [20.841800], rounded [21]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.764706] || CCR: [4.633700], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.764706] || CCR: [4.633700], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [4.725490] || CCR: [4.700400], rounded [5]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.975200], rounded [21]

Seen voltage: [-4.843137] || CCR: [20.97
```



```
COM35 - PuTTY

Seen voltage: [3.627451] || CCR: [6.568000], rounded [7]

Seen voltage: [3.039216] || CCR: [7.568500], rounded [8]

Seen voltage: [2.411765] || CCR: [8.635700], rounded [9]

Seen voltage: [1.823529] || CCR: [9.636200], rounded [10]

Seen voltage: [1.274510] || CCR: [10.570000], rounded [11]

Seen voltage: [0.607843] || CCR: [11.703900], rounded [12]

Seen voltage: [0.058824] || CCR: [12.637700], rounded [13]

Seen voltage: [-0.607843] || CCR: [13.771600], rounded [14]

Seen voltage: [-1.156863] || CCR: [14.705400], rounded [15]

Seen voltage: [-1.745098] || CCR: [15.705900], rounded [16]

Seen voltage: [-2.411765] || CCR: [16.839800], rounded [17]

Seen voltage: [-3.000000] || CCR: [17.840300], rounded [18]

Seen voltage: [-3.549020] || CCR: [18.774100], rounded [19]

Seen voltage: [-4.137255] || CCR: [19.774600], rounded [20]

Seen voltage: [-4.490196] || CCR: [20.374900], rounded [20]

Seen voltage: [-3.901961] || CCR: [19.374400], rounded [19]

Seen voltage: [-3.274510] || CCR: [18.307200], rounded [18]

Seen voltage: [-2.686275] || CCR: [17.306700], rounded [17]

Seen voltage: [-2.137255] || CCR: [16.372900], rounded [16]

Seen voltage: [-1.549020] || CCR: [15.372400], rounded [15]

Seen voltage: [-0.960784] || CCR: [14.371900], rounded [14]

Seen voltage: [-0.372549] || CCR: [13.371400], rounded [13]

Seen voltage: [0.294118] || CCR: [12.237500], rounded [12]

Seen voltage: [0.843137] || CCR: [11.303700], rounded [11]

Seen voltage: [1.431373] || CCR: [10.303200], rounded [10]

Seen voltage: [1.980392] || CCR: [9.369400], rounded [9]

Seen voltage: [2.607843] || CCR: [8.302200], rounded [8]

Seen voltage: [3.156863] || CCR: [7.368400], rounded [7]

Seen voltage: [3.745098] || CCR: [6.367900], rounded [6]

Seen voltage: [4.254902] || CCR: [5.500800], rounded [6]
```

STM UART showing voltage and CCR for a square wave and saw tooth respectively. Note the linear increase and decrease in the saw tooth resolution (seen right)



```
Seen voltage: [4.019608] || CCR: [5.901000], rounded [6]

Seen voltage: [4.333333] || CCR: [5.367400], rounded [5]

Seen voltage: [4.529412] || CCR: [5.033900], rounded [5]

Seen voltage: [4.450980] || CCR: [5.167300], rounded [5]

Seen voltage: [4.176471] || CCR: [5.634200], rounded [6]

Seen voltage: [3.705882] || CCR: [6.434600], rounded [6]

Seen voltage: [3.078431] || CCR: [7.501800], rounded [8]

Seen voltage: [2.372549] || CCR: [8.702400], rounded [9]

Seen voltage: [1.549020] || CCR: [10.103100], rounded [10]

Seen voltage: [0.607843] || CCR: [11.703900], rounded [12]

Seen voltage: [-0.254902] || CCR: [13.171300], rounded [13]

Seen voltage: [-1.196078] || CCR: [14.772100], rounded [15]

Seen voltage: [-2.098039] || CCR: [16.306200], rounded [16]

Seen voltage: [-2.843137] || CCR: [17.573500], rounded [18]

Seen voltage: [-3.549020] || CCR: [18.774100], rounded [19]

Seen voltage: [-4.058824] || CCR: [19.641200], rounded [20]

Seen voltage: [-4.450980] || CCR: [20.308200], rounded [20]

Seen voltage: [-4.647059] || CCR: [20.641700], rounded [21]

Seen voltage: [-4.647059] || CCR: [20.641700], rounded [21]

Seen voltage: [-4.450980] || CCR: [20.308200], rounded [20]

Seen voltage: [-4.058824] || CCR: [19.641200], rounded [20]

Seen voltage: [-3.509804] || CCR: [18.707400], rounded [19]

Seen voltage: [-2.843137] || CCR: [17.573500], rounded [18]

Seen voltage: [-2.019608] || CCR: [16.172800], rounded [16]

Seen voltage: [-1.196078] || CCR: [14.772100], rounded [15]

Seen voltage: [-0.294118] || CCR: [13.238000], rounded [13]

Seen voltage: [0.607843] || CCR: [11.703900], rounded [12]

Seen voltage: [1.431373] || CCR: [10.303200], rounded [10]

Seen voltage: [2.294118] || CCR: [8.835800], rounded [9]

Seen voltage: [2.960784] || CCR: [7.701900], rounded [8]

Seen voltage: [3.627
```

STM UART for sine wave – gradual around extremes.

Lessons Learned

Communication between systems is vital in embedded systems as there may be external hardware with functionality needed. Another lesson learned/concept debated on was the tradeoff of precision at the expense of hardware and the number of pins used/required to provide a fine enough sample for smooth movement of the servo. The arithmetic and scaling of the post A/D code conversion voltage and number of pins used can be attributed to the “jerkiness” of the servo as well as the frequency of the incoming signal.