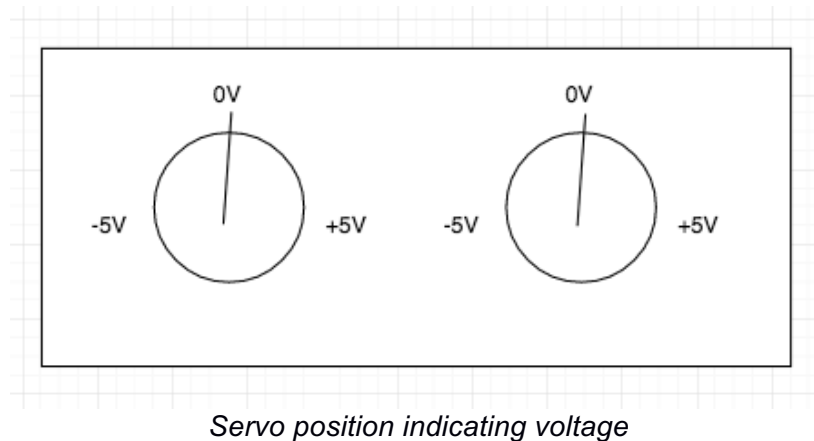


Rochester Institute of Technology
Real Time and Embedded Systems
Project 6 – Voltage Indicator Using Servos
May 10, 2017

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.



Areas of Focus

Zachary Weeden: Report, QNX A/D conversion and port/pin setup, STM register update
Dinesh Bashkaran: STM pin setup and voltage validation

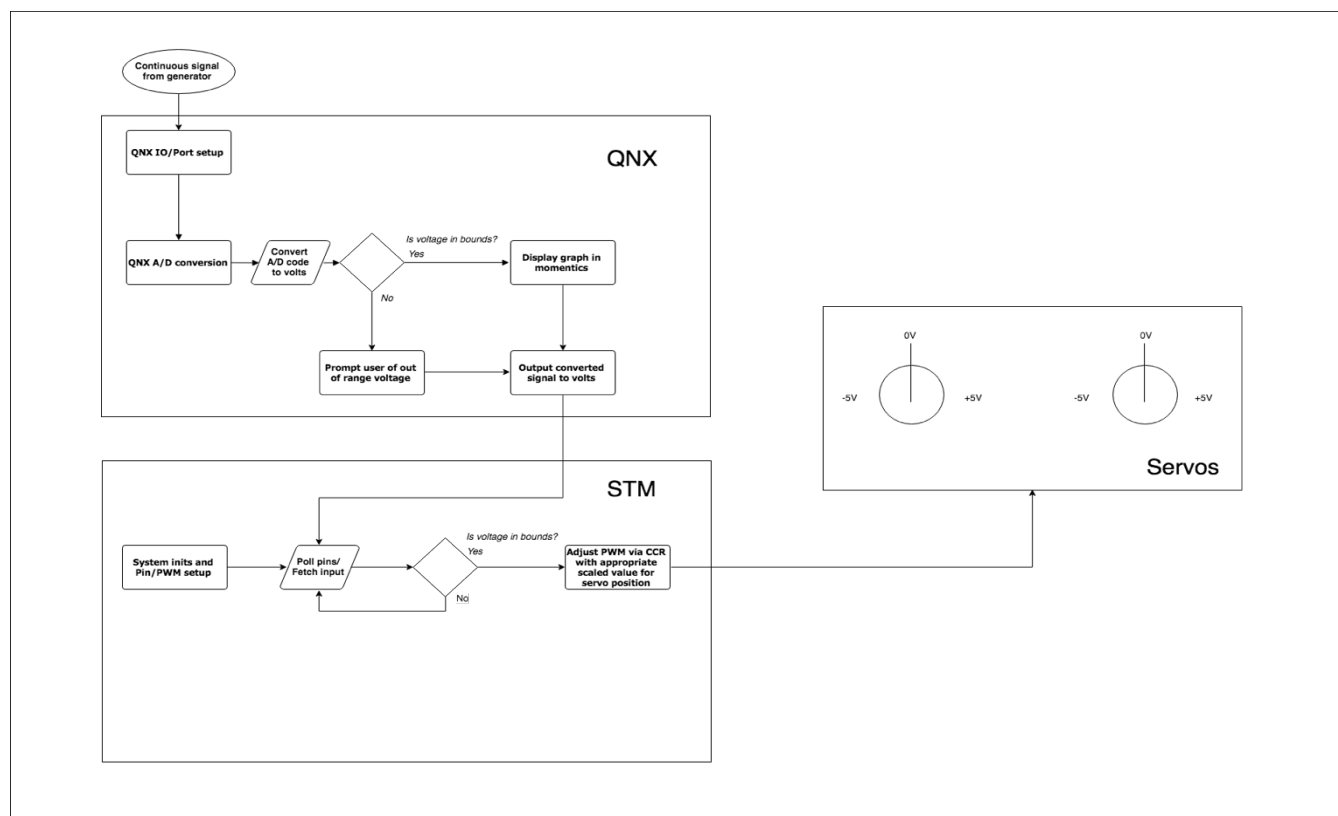
Analysis/Design

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.

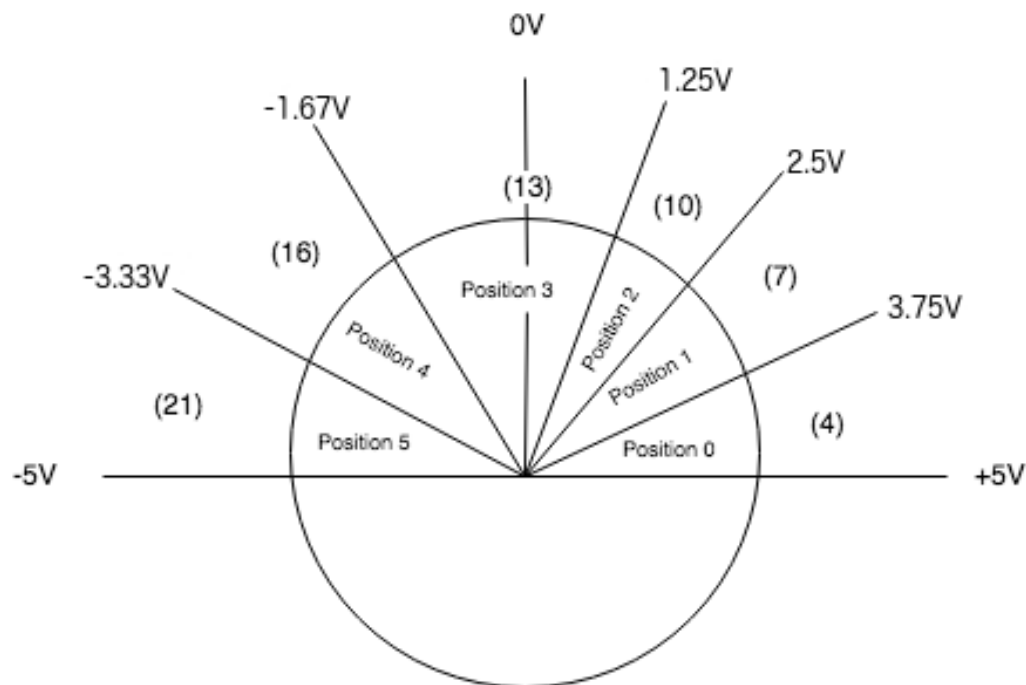
After that is all said and done we can begin writing to registers and in turn begin an A/D conversion. In an indefinite loop, we pass the necessary register handler pointers to a function that sets the proper values to the proper registers to perform a bipolar -5 to +5 A/D conversion. The resulting A/D 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 it's 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.



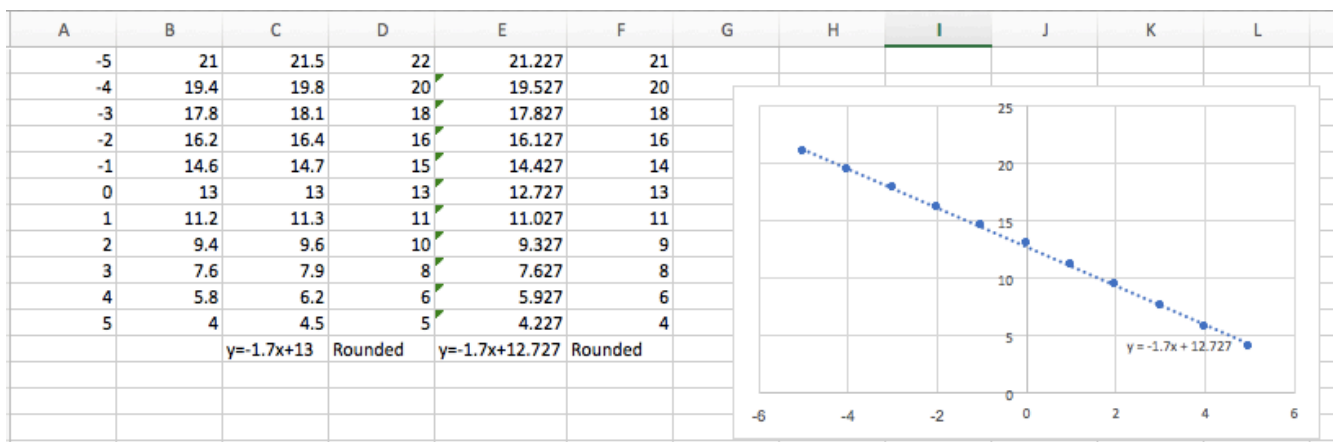
Block diagram showing general functionality and component isolation

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.



$$CCR = \lfloor -1.7 * voltage + 12.727 \rfloor$$

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.

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.