Rochester Institute of Technology

Real Time and Embedded Systems

Project 6 – Voltage Indicator Using Servos

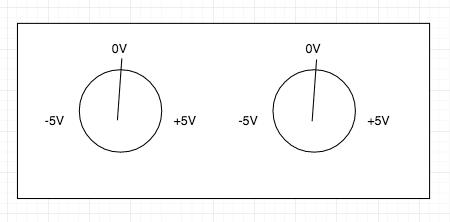
May 10, 2017

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*

**System Block Diagram/Workflow**

**Hours of Operation Timer**

This is the main control of the program and used in determining if more customers should be generated and queued for servicing. This individual component is emulating the hours of operation; open and close of the bank.

**Customer Queue**

This is a structure to organize the arrival of customers. This can be of the array data type and be in it’s own thread. The queue will continually be dequeuing by the 3 teller threads throughout program execution and customer completion as well as be appended to with the new arrival of customers until the close time.

**Bank Teller X**

These are 3 separate threads that will be used to handle and process customers from the customer queue. Each teller thread will lock once a customer is taken into that thread via a semaphore/mutex to indicate that this particular teller is unavailable.

After the randomly generated transaction time of the specific customer has been exceeded, the customer is deemed to be handled and is completed with their business. Metadata of the customer’s transaction will be retained for processing of overall/average metrics at the program’s completion and the thread will then unlock indicating that a customer can be seen/taken in by this teller thread.

**Software Design**

In regards to answering listed questions in the report document:

1. The customer queue can simply just to be thought of as an array of elements – I plan on making these elements the randomly generated transaction times of that particular customer.
2. On the topic of generation – I plan to do something like specify a range for valid values and generate a uniform distribution with something like:

*return lower\_bound+(rand()/(RAND\_MAX/(upper\_bound+1-lower\_bound));*

1. Customer-Teller assignment for the first of each teller is to simply take in the first 3 customers to each one of the tellers. From there, the unlocking of a thread will indicate precedence and more importantly, availability to accept another customer.
2. Metrics will be handled at the end of the program but individual customer metrics will be stored as soon as each customer has been completed.