Real Time and Embedded Systems: Project 2b

To Design and implement the QNX “Purple box” program exhibiting multitasking characteristics in simultaneously controlling a pair of servo motors using a custom interpreted control language.

Harshdeep Singh Chawla
hxc3427@g.rit.edu

Deekshit Shenoy Manour
ds7194@rit.edu

Abstract

The objective of the project is to design and implement an embedded, stand-alone QNX Neutrino program to control a pair of servo motors. The system will be responsive to simultaneous independent, externally provided commands. Each servo will be controlled by its own recipe of commands in the custom interpreted control language.

Analysis/Design

In this project, we use 2 threads for two servos Servo1 and Servo2 .The main checks for the user input and whenever there is an input from the user the threads checks the condition and selects the corresponding mode. The duty cycle Values are set and loaded into the functions, which will be used by the function depending on the user command. We have used the usleep function and set the corresponding values depending on the duty cycles for different positions. The real-time clock resolution is set to 0.1 mili seconds thereby reducing timer jitters and thus ensuring the smooth running of servomotors. The recipes are loaded for demo in a function and the thread calls the function whenever there is an input. The motors are run by a function called which updates the servomotors position value after each execution. This position is used to move the motor right and left depending on the user command.

Design Constraints:

The approach used in the completion of this project is as follows:

- Understand the components and functionality of the Purple Box.
- Understand the functionality of the Servo motors.
- Calculate the duty cycle.
- Remove distortion from the the signal due to timer jitter.
- Write and debug program.

Areas of Focus:

Harshdeep:

- Hardware Configuration & Creation of Test Plan.
- Report work.

Deekshith:

- Design implementation of code on QNX neutrino.
- Implementation of code on QNX neutrino

Hardware

Servo Motor Used

Features:

- **Control System:** +Pulse Width Control 1520usec Neutral
- **Required Pulse:** 3-5 Volt Peak to Peak Square Wave
- **Operating Voltage:** 4.8-6.0 Volts
- **Operating Temperature Range:** -20 to +60 Degree C
- **Operating Speed (4.8V):** 0.23sec/60 degrees at no load
- **Operating Speed (6.0V):** 0.19sec/60 degrees at no load
- **Stall Torque (4.8V):** 44 oz/in. (3.2kg.cm)
- **Stall Torque (6.0V):** 56.8 oz/in. (4.1kg.cm)
- **Operating Angle:** 45 Deg. one side pulse traveling 400usec
- **Continuous Rotation Modifiable:** Yes
- **Direction:** Counter Clockwise/Pulse Traveling 1520-1900usec
- **Current Drain (4.8V):** 7.2mA/idle
- **Current Drain (6.0V):** 8mA/idle
- **Motor Type:** 3 Pole Ferrite
- **Potentiometer Drive:** Indirect Drive
- **Bearing Type:** Plastic Bearing
- **Gear Type:** All Nylon Gears
- **Connector Wire Length:** 12"
- **Dimensions:** 1.6" x 0.8"x 1.4" (41 x 20 x 36mm)
- **Weight:** 1.3oz. (37.2g)

OPERATION

The PWM signal from the purple box behaves as the control signal for the servos. By varying the duty cycle of the signal we actually can turn ON and Off the servos for the time required. The time period of the PWM signal used is 20ms. The servos also required a +5V signal to operate. The required duty cycle for various positions are as follows: -

- Position0 = 1.95%
- Position0 = 3.5%
- Position0 = 5.05%
- Position0 = 6.4%
- Position0 = 8.2%
- Position0 = 10.7%

Micro- Controller

We made use of the ‘Purple box’ which has an Athena microcontroller. Connections to the sensor was made through the data acquisition (DAQ).

Test Plan

Hardware test:

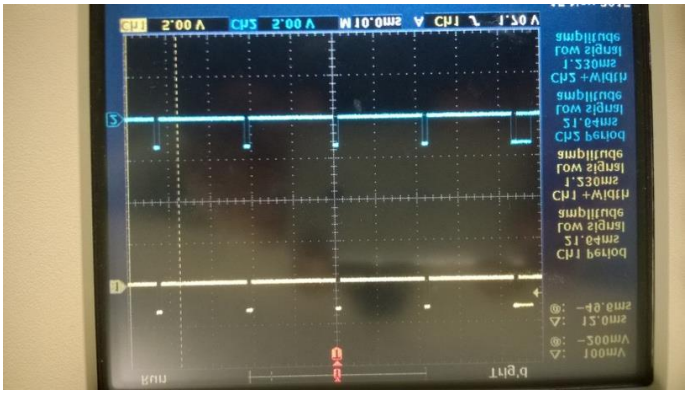
The proper functioning of the Servos was checked using the following method: -

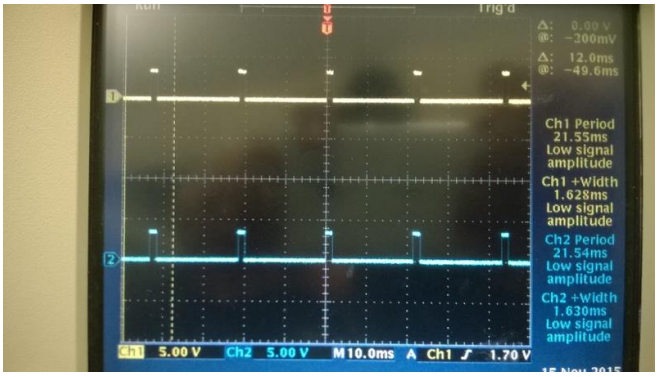
Using Ohm meter: Disconnect all power from machine. Check all three wires singly T1, T2, T3 (all three phases) to the ground wire. Readings should be infinite. If its zero or reads any continuity at all, then a problem exists with either the motor or cable. If it is go directly to the motor and disconnect from cable and check motor and the cable separately. Be sure to make sure leads on both ends are not touching anything including the other leads. Most servo motor shorts can be read with a regular quality meter.

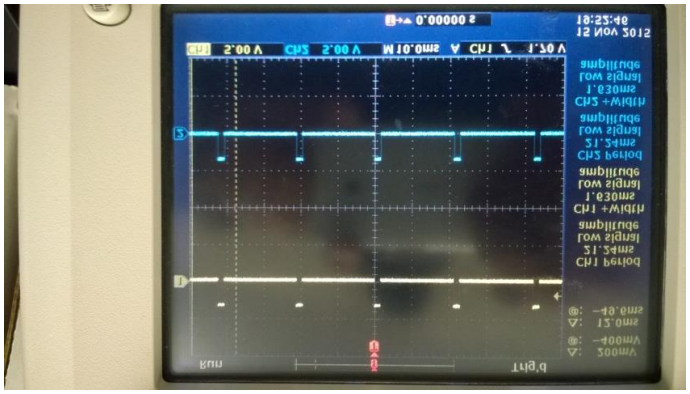
Software test:

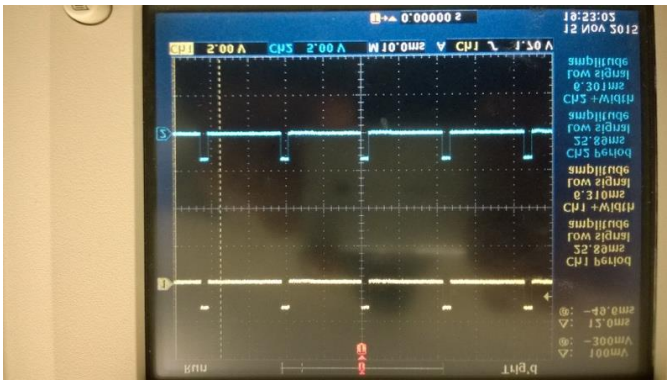
The software testing consisted of issuing proper trigger signals to the Servos and then noting the movement of the motors.

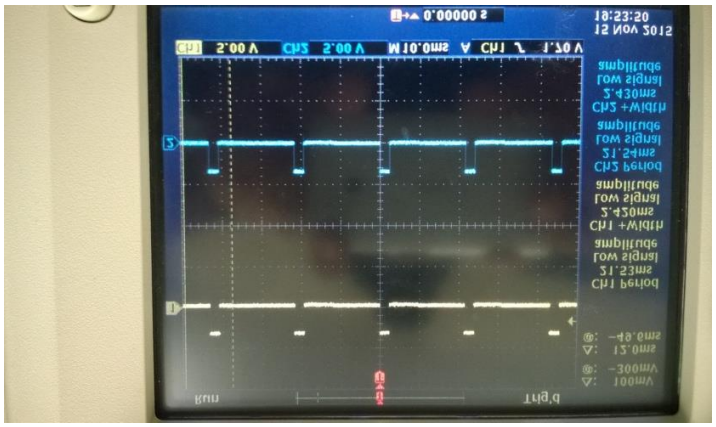
Signals Triggered were checked by the help of oscilloscope:-

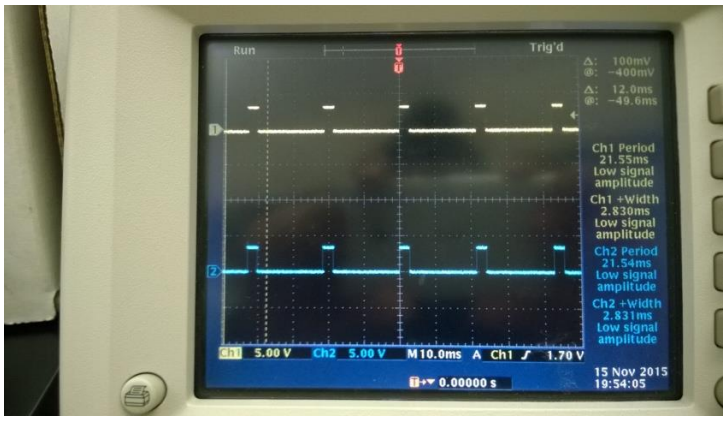




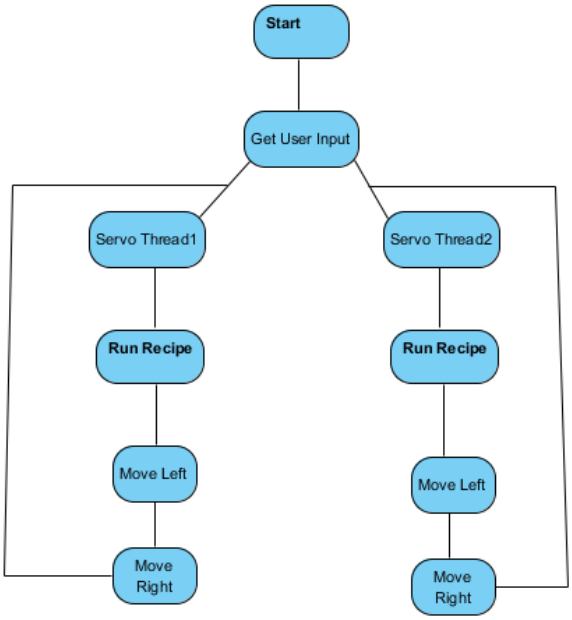








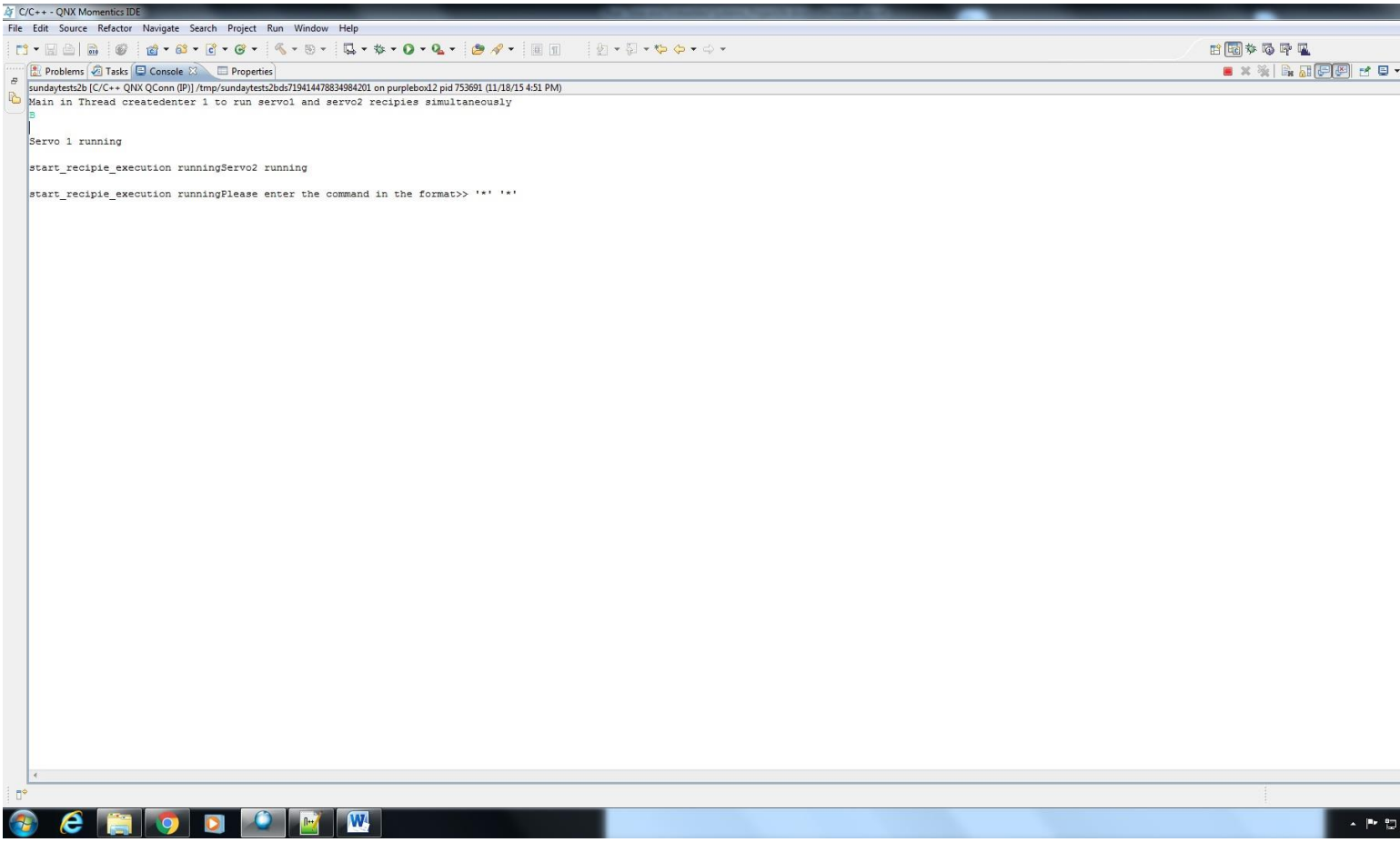
Flow Chart



Lessons Learned

In this project we learned and implemented the basic features of the QNX such as setting timer resolution of the real time clock for more precision of timer ticks and thereby reducing jitters . Thus, enabling smooth running of the motors.. As it was the second time we have used the servo motors in our project, the test plan was easy to make. But a major challenge was determining the information we needed to know to run the motors using purple box. However, once an initial general understanding of the hardware was achieved, this process became straightforward. Another challenge encountered was adequately dividing the work and responsibilities. Developing cross discipline skills is important as a student and it later becomes essential when entering industry. After this was achieved, the code was developed and debugged to obtain the proper output. However, the desired results were not obtained, as instead of obtaining accurate values the sensor output showed “out of bounds” every time.

Output


C/C++ - QNX Momentics IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Tasks Console Properties
sundaytests2b [C/C++ QNX QConn (IP)] /tmp/sundaytests2bds719414478834984201 on purplebox12 pid 753691 (11/18/15 4:51 PM)
Main in Thread createdenter 1 to run servo1 and servo2 recipies simultaneously
Servo 1 running
start_recipie_execution runningServo2 running
start_recipie_execution runningPlease enter the command in the format>> '*' '*'