Using Simulink to Control the Robotic Arm

EECE 2160: Embedded Design - Enabling Robotics Final Project
Date Submitted: December 6, 2018

Section 08 Group 9: Elena Silva (silva.el@husky.neu.edu), Cassie Harbour (harbour.c@husky.neu.edu), Annie Waye (waye.a@husky.neu.edu)

Summary

The goal of this final project was to use the knowledge we gained through our course work and lab assignments to program and control the robotic arm in an easy, user-friendly manner. Our implementation allows the user to control all five joints of the arm with only three ZedBoard push-buttons. The user enables the center push-button to cycle through all five joints and select one joint to control. As the user chooses a joint, the LED that corresponds to the current joint is lit up, easily distinguishing which joint is active. The pivoting-movement of the joint is manipulated with the "UP" and "DOWN" push-buttons. Holding the UP button will simultaneously increase the angle of the joint and holding the DOWN button will decrease this angle.

Description

The top-level design of our Simulink Program "User809_Final_Project.slx" is zipped with our project submission and displayed in Figure 1. Each push-button is mapped to an input component and immediately de-bounced to avoid any adverse effects of the button signal oscillating when enabled. The de-bounce subsystem used for all buttons is displayed in Figure 2. It was designed to ensure that each button is pressed for at least 250ms before its value is read and passed to the rest of the circuit.

The state of the center button is used to enable a counter, beginning at 1 and stepping by one until it reaches 5 and repeats. The counter's output is sent to five "Compare to Constant" elements, that, together, determine which joint is active and which LED should light up.

The inputs of the UP and DOWN push-button are sent to each joint's subsystem to either increment or decrement its current angle. All joint subsystems are identical in logic, and Figure 3 shows the subsystem for the Bicep joint. When the up button is on, and the current angle is 170 degrees or less, the counter is enabled and the current angle is increased. When the down button is pressed, the current angle is checked to ensure it is not below 10 degrees, and then the angle is decreased. The increasing and decreasing is down in the "CurrentAngle" counter. The output of this counter is sent to the Angle To PWM subsystem, which produced the PWM to be mapped to the Zedboard input.

Figure 1: Simulink Design

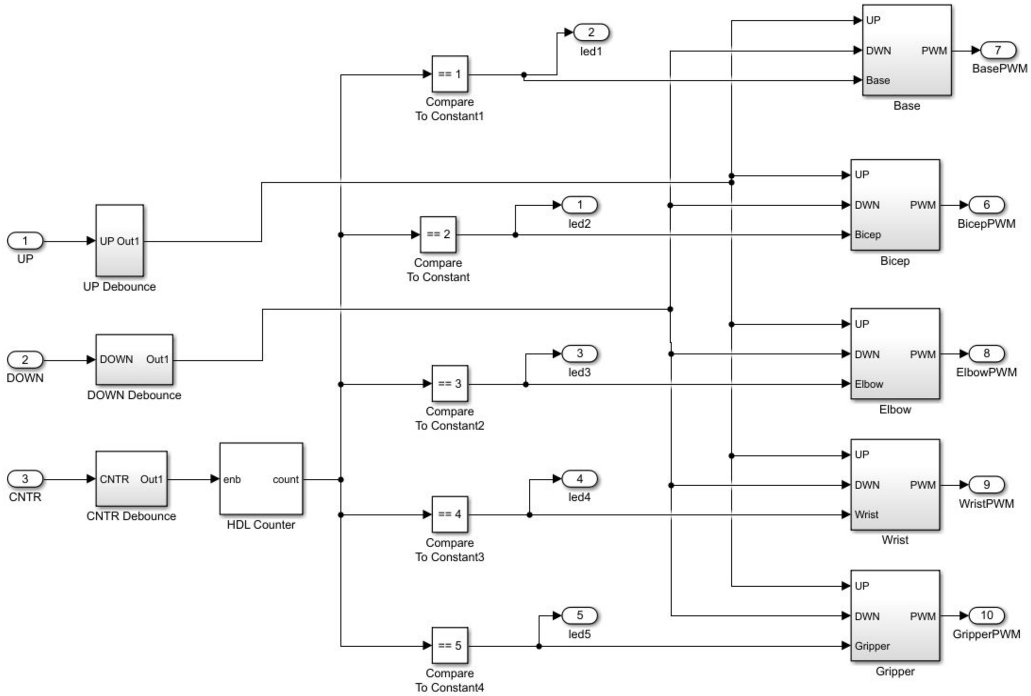


Figure 2: De-bouncer subsystem

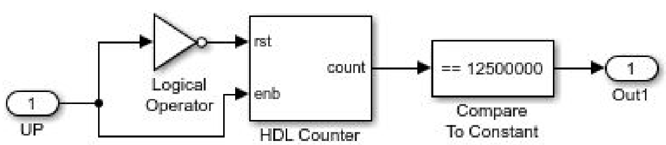
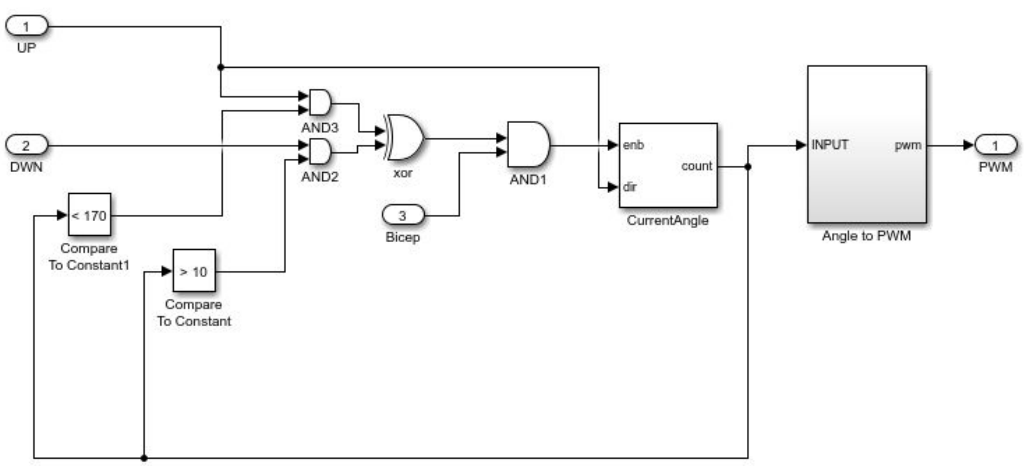


Figure 3: Bicep Joint Subsystem



Relevance in Couse

Our project involves most of the skills we obtained in lab and the understanding we gathered from lectures. We learned in the first lecture, how software and hardware are configured to constitute a computer system. Thus, our project relies on digital logic to control and interact with hardware components. As early as Lab 4, we were combining software and hardware to design a product with a specific purpose. The ZedBoard was our main piece of hardware, as it has many hardware pieces to work in conjunction with our programs. Simulink was the prominent software component in our assignment, as opposed to programming with C++ based software, because based on past lab assignments, we believed Simulink was more consistent and trustworthy than programming with MobaXterm. We wanted to design an interactive, in real time, system that can easily tailor to many situations, rather than a system that runs a specific sequence of steps or movements to complete one function.

We learned in class that the robotic arm operates through Pulse Width Modulation (PWM) signals. The exact position of each joint is determined by the periodic signal created in Simulink. The Lab 6 handout illustrates different "duty cycles" and corresponding servo arm positions. In Lab Assignment 11, inputs were integer values representing servo arm angles and speeds. We redesigned the program to operate at a constant speed, controlled by the "Current Angle" counter step-value. Additionally, the angle was increased or decreased relative to the joint's current angle rather than targeting a specific angle value. We maintained the portion of Lab 11 logic that converts the integer angle values into PWM signals with a fixed frequency in subsystem "Angle to PWM."

Roles

Our group worked as a team throughout the completion of this project. The lab assignments were all completed as a group, and the reports were also written together, so we were very prepared and accustomed to working as a unit. Together, we brainstormed and choose our topic based on what we most enjoyed working on during the labs. The implementation and testing of the Simulink project was done together, which made the process efficient because we are used to working this way. Cassie and Annie operated the robot during our demo for the video, while Elena took the footage. We also worked together to edit and upload the video. This was helpful as none of us had much experience creating videos. For the report, we simply created a shared Google document and each contributed.

YouTube Link

"EECE2160 Final Project (Silva, Harbour, Waye)" https://youtu.be/j4hNuhg5iBY