Lab 5 - Stepper Motor

Objective

This lab has these major objectives: the understanding and implementing linked data structures; learning how to create a software system3; the study of real-time synchronization by designing a finite state machine controller. Software skills I used include advanced indexed addressing, linked data structures, creating fixed-time delays using the SysTick timer, and debugging real-time systems.

Introduction

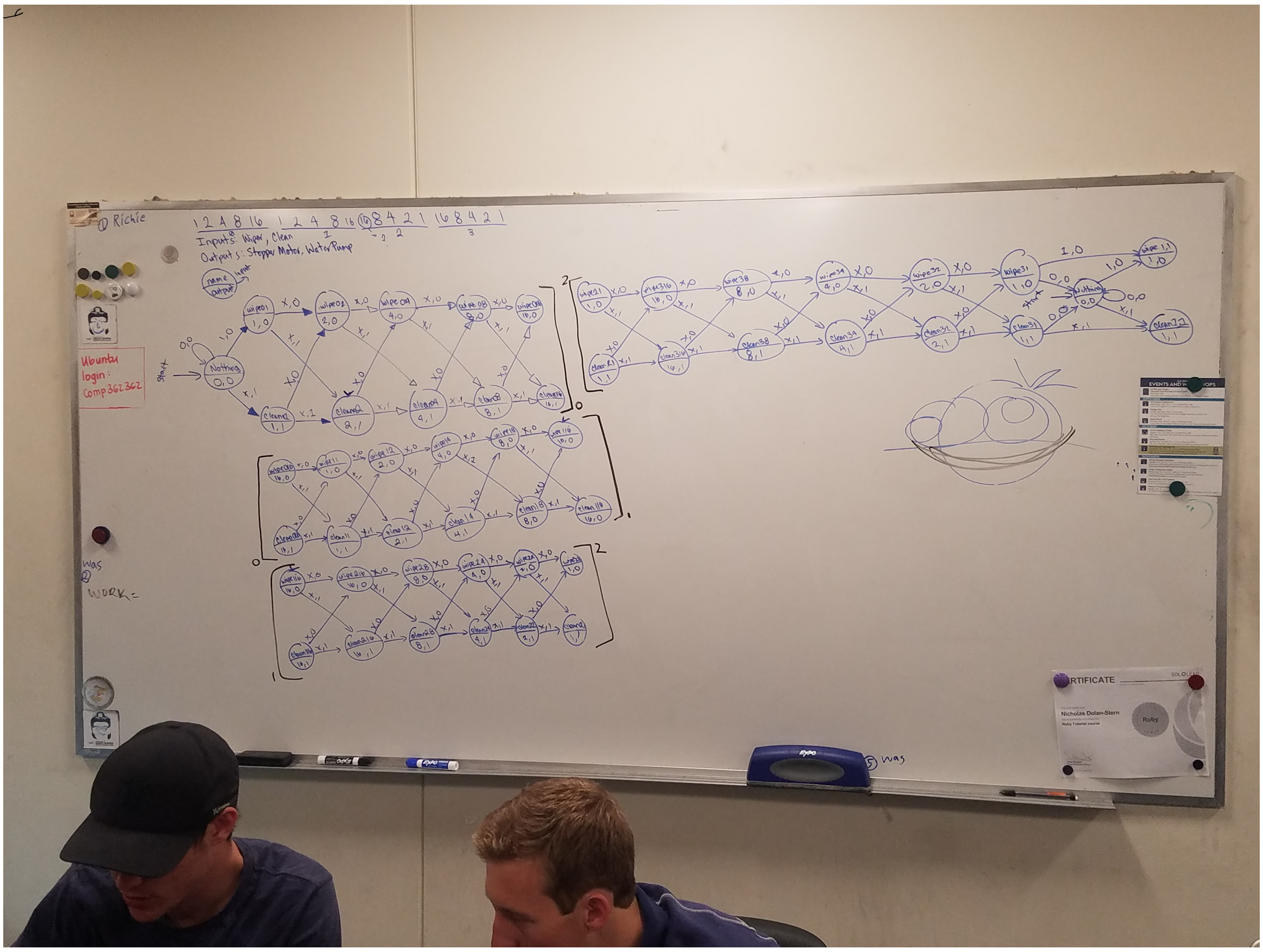
Consider a stepper interface circuit as shown below. The stepper motor will controlled by 5 output pins of the microcontroller. The ULN2003 driver must be used because the motor currents can be as high as 200mA. Notice also pin 9 of the driver must be connected to +5V (VBUS). With this circuit, outputs to PE4,PE3,PE2,PE1,PE0 will cause the motor to move. Each new output causes one step of the motor (4 degrees). Any five pins can be used. However, we recommend using one of the choices allowed by the simulator. You will also interface an LED to another output pin (not shown in Figure 5.1, you may use any pin for the LED interface, however we recommend using one of the choices allowed by the simulator). The user will control the windshield wiper system with two buttons. One button (wiper) will active just the wiper and the other button (clean) will active the wiper and the washer.

Procedure

The basic approach to this lab will be to first develop and debug your system using the simulator and then interface with an actual stepper motor and switches on a physical TM4C123. As you have experienced, the simulator requires different amount of actual time as compared to simulated time.

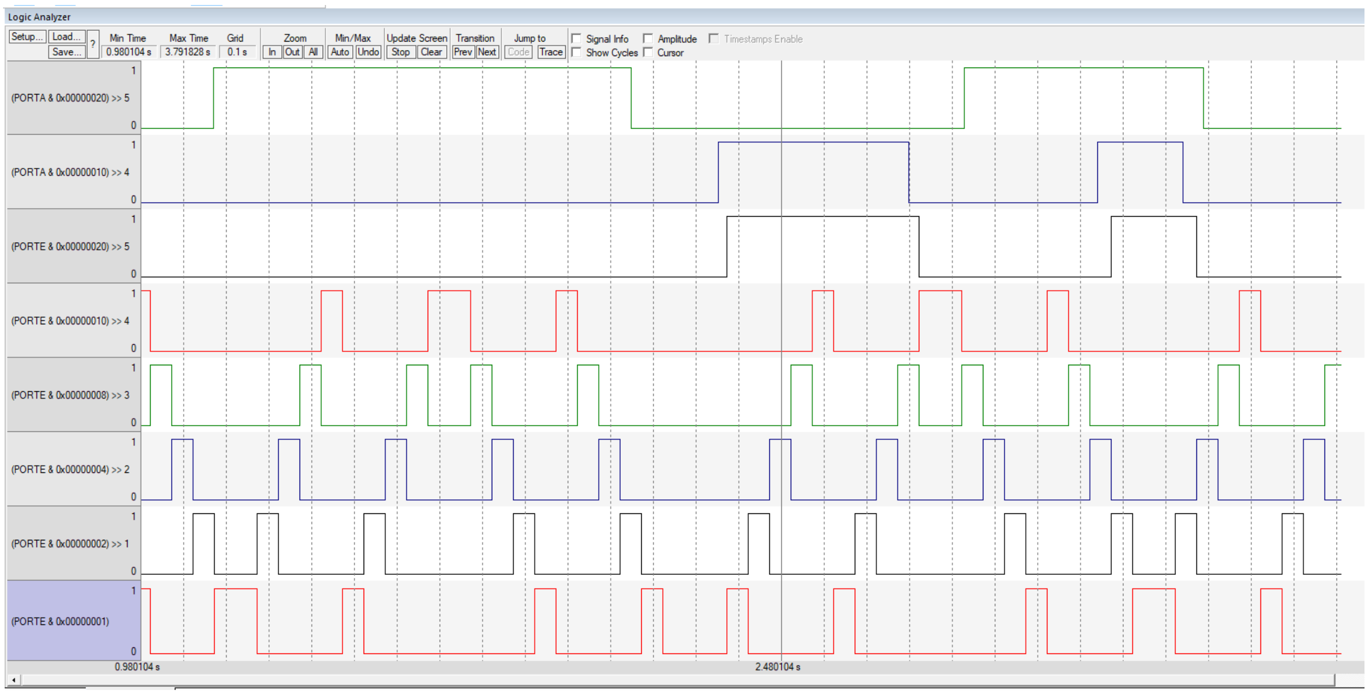
On the other hand, the correct simulation time is maintained in the SysTick timer, which is decremented every cycle. The simulator speed depends on the amount of information it needs to update into the windows and the speed of your personal computer.All software in this lab must be developed in C. The Lab 5 starter file has the appropriate files needed for Lab 5. The call to TExaS_Init will activate 80 MHz PLL. The function passed to TExaS_Init will specify which bits will be sent to the logic analyzer.

Finite State Machine



Implemented this in C.

Analyzer



This ended out working perfectly

Test Circuit