Introduction to the Development of an Embedded System

Performance Check

______ (20) 1: IO Setup and Testing

______ (70) 2: Program skeleton

______ (90) 3: Total functionality of washing machine (procedure 4)

Introduction

In this exercise, you will write code to run a washing machine. This is a two week lab. Both Lab 4 and Lab 5 will receive the same score. This exercise requires ONLY the knowledge and experience you gained in ECET 179 and what you learned from Lab 1 - 3. Some reminders of where you might look to review your work are also included

Specifically:

- I/O PORT initialization. (ECET179 Labs #2&3)
- Bit masking (ECET179 Lab #2)
- Pattern Sequencing and Control Stepper Motor (ECET279 Lab #3)
- Generation of time delays (ECET179 Lab #11)
- Reading push-buttons (ECET179 Lab #2, ECET279 Lab#1)

Washing Machine Controller Operational Requirements:

Press Start to start the washing cycle sequence <u>**if**</u> door closed.					
Operation	**Drain Valve state**	**Stepper Motor state: Speed control using inter-step delay**	**Hot Water state**	**Cold Water state**	**Duration, seconds**
FILL	OFF	OFF	ON/OFF based on Temperature – see below		4
WASH	OFF	Agitate: CW/CCW	OFF	OFF	8
DRAIN	ON	OFF	OFF	OFF	4
FILL	OFF	OFF	ON/OFF based on Temperature – see below		4
RINSE	OFF	Agitate: CW/CCW	OFF	OFF	12
SPIN	ON	High Speed CW	OFF	OFF	9
DONE LED ON	OFF	OFF	OFF	OFF	
Door Closed switch OPEN		Return to start of wash cycle. Close door and push start to repeat wash cycle			

Temperature Settings based on any one of three switch settings (any other combination has no effect i.e. **trap any other combinations**):

HOT	Hot Water ON, Cold Water OFF
WARM	Hot Water ON, Cold Water ON
COLD	Hot Water OFF, Cold Water ON

Motor Actions:

Agitating (motor shaft slowly, briefly and repeatedly moving CW and CCW) using **bit pattern sequencing of stepper motor in half step mode**.	5ms inter-step delay 2s CW 2s CCW repeat for 8s wash cycle 12s rinse cycle
Spin (motor shaft moves fast CW continuously) Using bit pattern sequencing of stepper motor in **full-step mode**	3ms inter-step delay 9s total for spin cycle

Hardware Assignments:

L&S represents your own Light and Switch board or the Light and Switch board in the lab.
This hardware connection should be included at the top of your code, as hardware connections. Fill in the IO pins of your choice. Choose wisely, this can make a difference in how easy or difficult your project is. Consider reusing modules from your previous labs.

Inputs:	**PIN (ex. 'PINA.2')**
Temperature selection switches:	
1. Hot	
2. Warm	
3. Cold	
Door Open switch	
Start push-button	
Outputs:	**PORT (ex. 'PORTA.2')**
Motor Control:	
IN1	
IN2	
IN3	
IN4	
Drain valve	
Hot water valve	
Cold water valve	
Wash done LED	
Agitate LED	
Spin LED	

Timer assignments:	**Timer**
Stepper Motor sequencing and duration time delays	Use Timer 0 (can use _delay_ms)
HINT: Longer time delays can be generated by using multiples of short inter-step delays e.g. 50*20mS = 1S BUT the motor function must receive the desired time NOT how many repeats of short delays are required.	

The solution to this problem WILL be done in small stages. Since there are many alternative tactics to developing the solution, then one particular approach has been defined for ALL to follow.

Objectives

1. To review C programming and microcontroller hardware material from ECET179
2. To solve a programming problem using piecemeal development
3. To interpret a system flowchart (on the last page of this handout) and implement the first stage of code development
4. To develop a flowchart for the motor control section of the problem and implement stage 2 of the code development using the _delay_ms(); function.
5. Integrate all stages into a final solution using multi-modular programming.

Hints:

A list of simple steps is given below:

1. Read the Start PB and check the door switch. Use an LED to indicate the state i.e. if LED is on then code is ready to read the desired temperature setting. This covers the first small section of the flow chart. Check operation.
2. Extend 1 to read the temperature setting, illuminate the appropriate hot and or cold valves (LEDs) for 4 seconds and then turn them off. Check operation.
3. Turn on an LED to indicate motor activity and leave it on for 4 seconds, repeat for another 4 seconds. Turn it off and now illuminate the drain LED (valve) for 4 seconds. Check operation.
4. Read temperature setting (you may decide to change it during the delays of stage 3), turn on the appropriate LEDs (valves) for 4 seconds and then turn them off. Check operation.
5. Continue development gradually building the stages as shown in the flow chart. The cycle should repeat only after the door switch has indicated that the door has been opened and re-closed.
6. It is simpler to set up a function to pass a character (‘A’ or ‘S’) and a time in seconds into the function and generate the spin sequence if an ’S’ is passed and simply light some other LED if an ‘A’ is passed.
7. Extend the function to produce the forward and reverse pattern for a specific time if an ‘A’ is passed.
8. Now add the motor control function to the code of steps 1 thru 5 above where LEDs were used before to indicate motor activity.
9. Add final touches to code comments (you should comment as you develop your code – **DO NOT** wait until the code works and then go back), get your check-offs and enjoy a coke, coffee or beer depending on your taste – you’ve earned it!

Procedures

Procedure 1: IO Setup and Testing

1. Create a folder named ECET279_Lab4_Wash_Embed_Sys_Dev
2. Create a project for the ATmega2560 called Wash_MC_Sys_Dev
3. Rename the c file to Control_stg_1.c
4. Add a comment header at the top of the .c file containing the following information in the order indicated below:
    a. Project name:
    b. .c filename:
    c. Author: *your name*
    d. Date:
    e. Brief yet complete overview description of what your code does
    f. A list of ATmega2560 peripherals used, their purpose and what PORT pins are connected to which pin on any external hardware device. (info in the hardware table)
5. Create a function to initialize all I/O PORTs used and set all outputs such that all external hardware devices are initially OFF.
6. Add the 'Debugger.c' and 'Debugger.h' files from lab 3 into the same folder as your project's 'main.c' file. You may find the ATMega Debugger 1.2 zip file from the resource folder. If you have already downloaded it from lab 3, don't need to download it again. Simply copy the two files into your folder. Note: serial.exe file takes about a minute to load and make sure you load your code from previous steps before you start serial.exe.
7. Add the Debugger module to your project (If you don't remember, refer back to lab 3, procedure 1).
8. Label the hardware switches, pushbuttons, and LEDs with a paper and placed next to the hardware. Make it easy to identify the functionality of the inputs and outputs.
9. Connect your hardware, and ensure the correct inputs turn green in the debugger and the output ports are configured correctly in the debugger.

Instructor Check off 1
1. Show IO Table and hardware labelled.
2. Demonstrate pushbuttons work with the Debugger
3. Demonstrate output configuration in the Debugger correspond to the hardware table

Procedure 2: Program Skeleton

1. Create the program based on the flow chart shown at the end of this document but <u>exclude</u> the motor control function and the delay function using Timer0. Instead place a comment at the various points in the code where the motor function would be called and temporarily use _delay_ms() to generate the 2, 3, and 4 second delays. It may be prudent to use an LED to indicate the points at which the motor function is called. Reference the tables in the Introduction section while writing to code.
2. Comment out the _delay_ms() lines and then use simulator to verify the operations.
3. Obtain a washing machine lab part from your lab instructor and connect the door wires to your input. Verify that it works using the debugger.

4. Load the code to your controller and verify with the hardware

Instructor Check-off

1. Use hardware to demonstrate the sequence of operation, hardware should be labelled.
2. Demonstrate the various temperature switch combinations and observer the correct valve operations (any other combination has no effect).
3. Demonstrate door switch operation (machine starts only after door is closed). After cycle complete, open the door to turn off the Done LED. Machine should not start again until door is opened.

Procedure 3: Motor operations

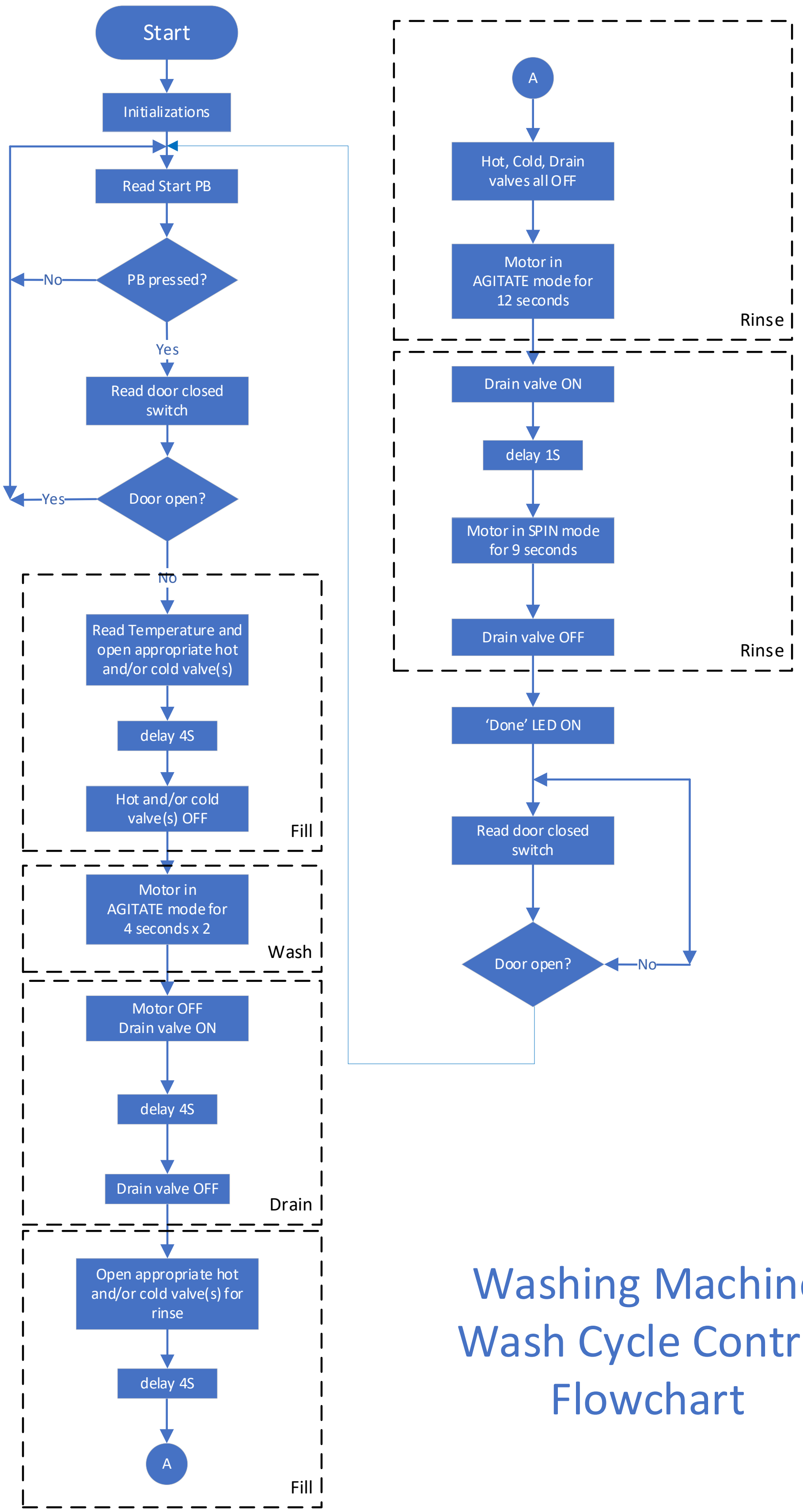
1. It is highly recommended that you **make a new project**, called Control_stg_2. This allows you to fall back to a known state when troubleshooting.
2. Add all the modules from previous steps and add the stepper module you developed from Lab 3 (you will modify one of the functions in this module).
3. Add a function in the stepper module for the motor control function based on the following requirements, inside the stepper module:
    a. The function should require **2 parameters** to be passed, namely:
        i. The **mode** i.e. 'A' for agitate, 'S' for spin, OFF is the default.
        ii. The required **time in seconds** for either the agitate or the spin mode to be on.
    b. In agitate mode, **the motor reverses direction every 2s** and the motor is driven in half step mode with a delay of 6ms between each step
    c. In spin mode, the motor runs in a clockwise direction in full step mode with a delay of 3ms between each step.
4. Download the code to the hardware and verify that the motor turns appropriately according to the Motor Action table at the beginning of this lab.

Procedure 4: Total functionality of washing machine

1. Incorporate the motor control functions from previous step into your project in Procedure 2.
2. Demonstrate the complete functionality of the system, including the motor control.

Instructor Check-off

1. Demonstrate complete functionality of the system with the hardware labelled
2. Code is fully commented



Washing Machine Wash Cycle Control Flowchart