RTOS: Stepper and Temperature Project Plan

Dan Blanchette

February 27, 2023

1 Tasks Break Down

1.1 Number of Tasks

I am thinking of using 4 tasks with priorities 1-5. In the following sub-sections, I will describe each task and its ranking in the architecture. My idea is that we have 4 devices. One is a shared resource by itself(the 7-segment display), two share the same I2C bus (stepper motor and the temp/humidity sensor), and the remaining 3 devices are interrupt buttons, each acting as a switch.

2 Priorities

2.1 Priority 5 and Priority 4 (The 7-SEG Displays)

The 7-segment display will have the two highest priorities as its refresh rate is paramount to maintain throughout the software run. A priority of 5 is given to the left digit and 4 for the second digit, controlled by a queue passing draw values between them, and is scheduled using semaphores.

3 Priority 3 The three Buttons

I want to ensure that the buttons have a high enough priority to operate as interrupt routines. these devices need a high enough priority to affect the underlying device (temp/hum sensor and stepper motor). While the 7-Segment display will be the highest,... when the buttons are pressed, I can use a queue to update values to the displays after they are registered and should update accordingly based on my hierarchy.

For noise reduction, I plan on using vTaskDelays based on the count of bounces registered (0 to 1 back to 0) per port-tick. I hope to develop a routine to approximate how many milliseconds per port-tick I need to deal with each button press for the program to ignore the input until a 1 or 0 is consistently registered.

3.1 Priority 2 Temp/Humidity Sensor

I'm thinking that the best option for the I2C bus device is to get a reading from the temp/humidity sensor. This is because the stepper motor's functionality relies on having the values parsed from the temp/hum sensor. It makes this device a prime candidate to use a queue for data transfer to both the 7-segment displays and the stepper motor values that will instruct its rotation.

4 Priority 1 The Stepper Motor

The stepper motor will be set to a priority of one. This ensures the data queue is not empty from the temp/humidity sensor's reading before it begins its rotation. It is also to schedule the I2C bus to have an opportunity (perhaps by semaphores) to be blocked on the stepper motor until the sensor has finished its read and can relay the information.

5 State Diagram

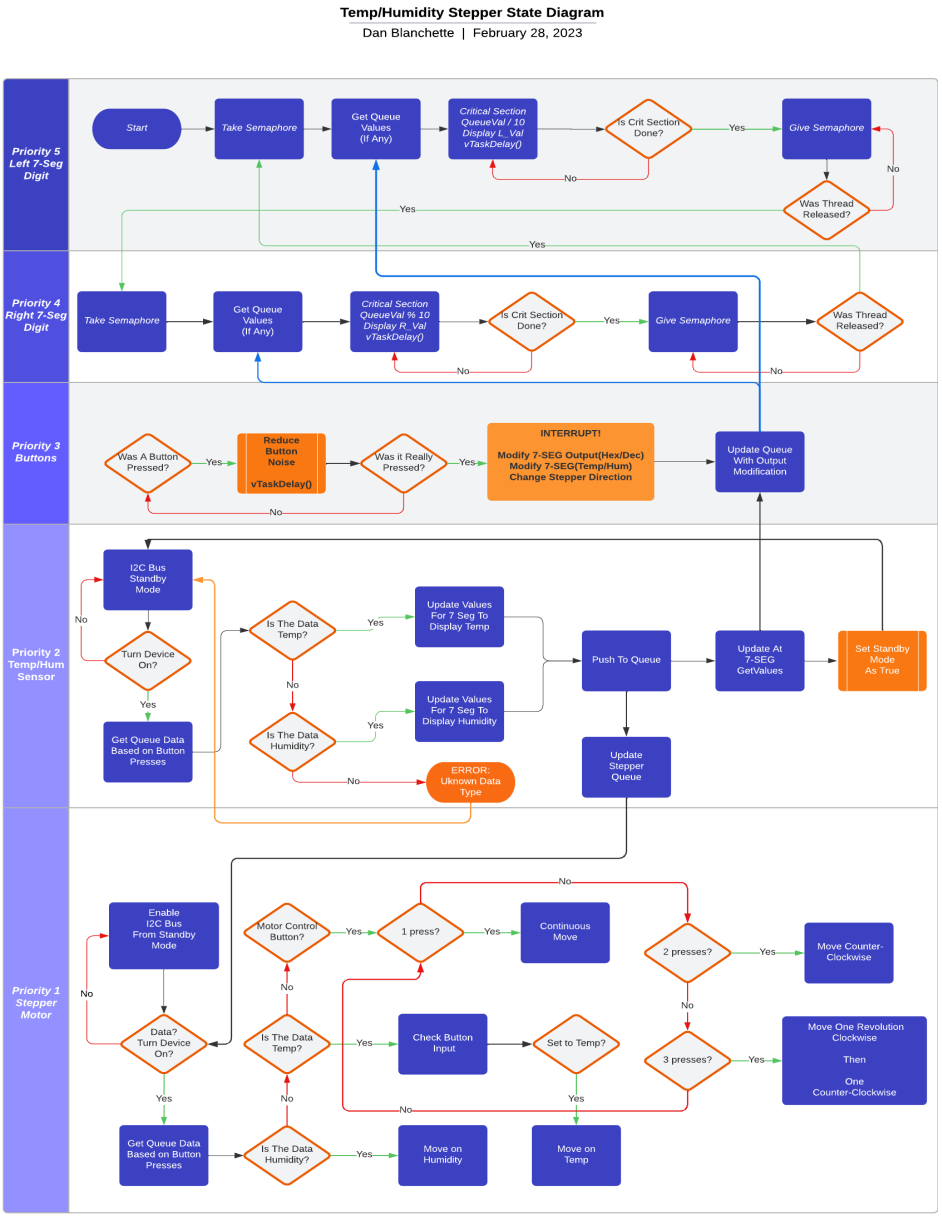

Figure 1: Temperature Humidity Stepper Motor State Diagram