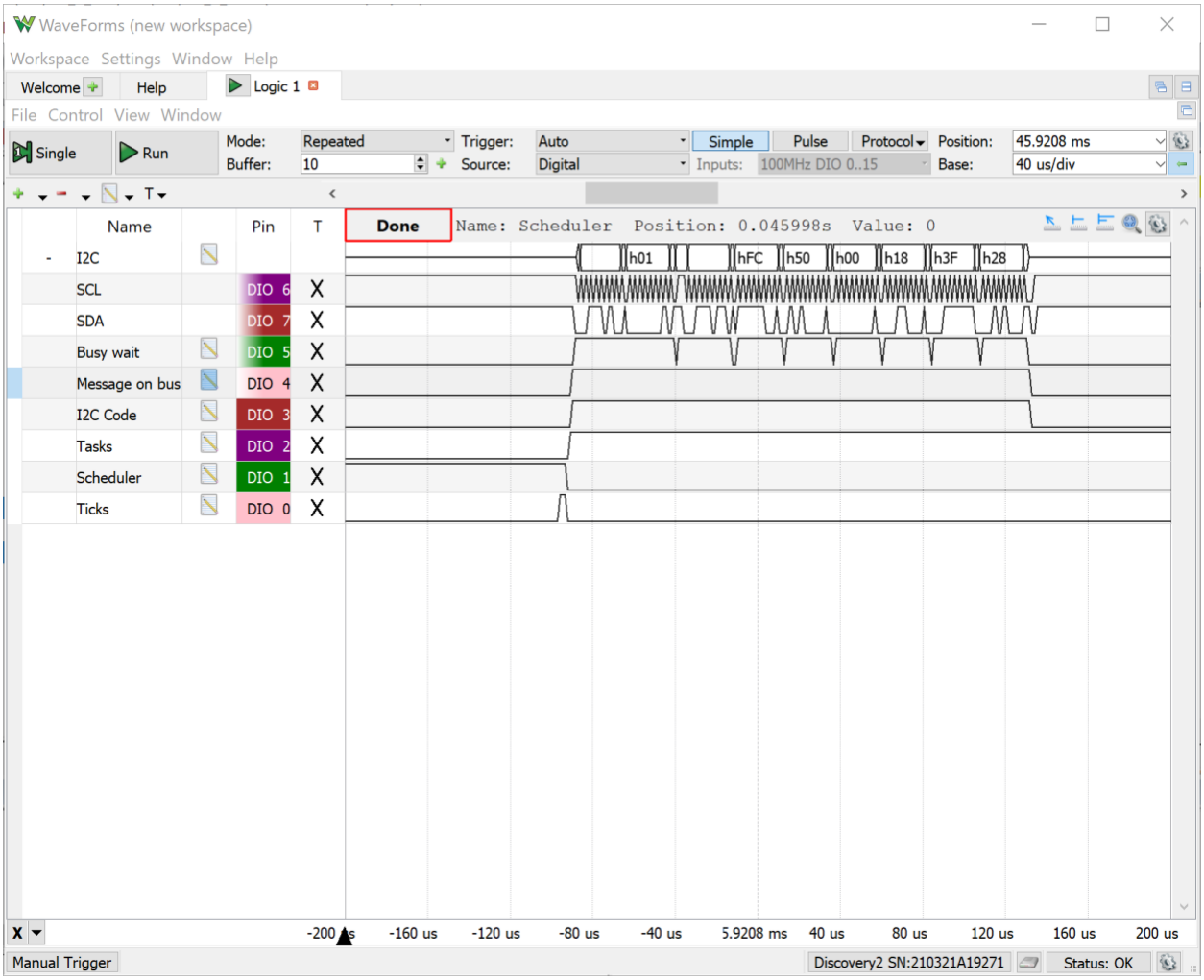
ECE 560 Project 1
Submitted By: Binoy Thomas
200260021

Q1. Screenshot of the read clearly shows 6 bits being read. The task debug bit is high, and the scheduler is low



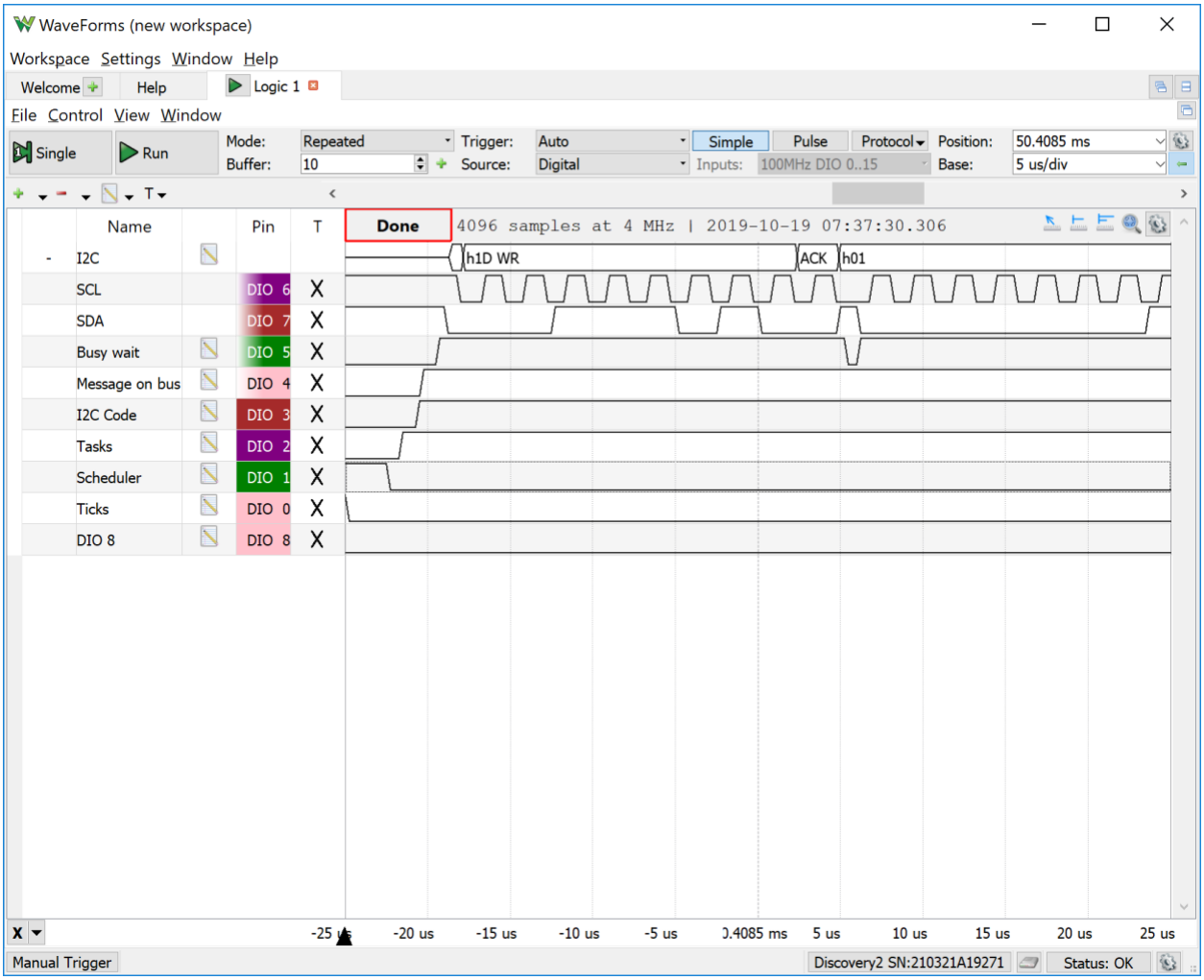
Task Motion Sensor is active for 122.775 ms

The task motion runs one cycle every 18.45 ms

The message on bus is active for 225 us for one message

The busy wait is active for 208.75 us when there is a message on bus

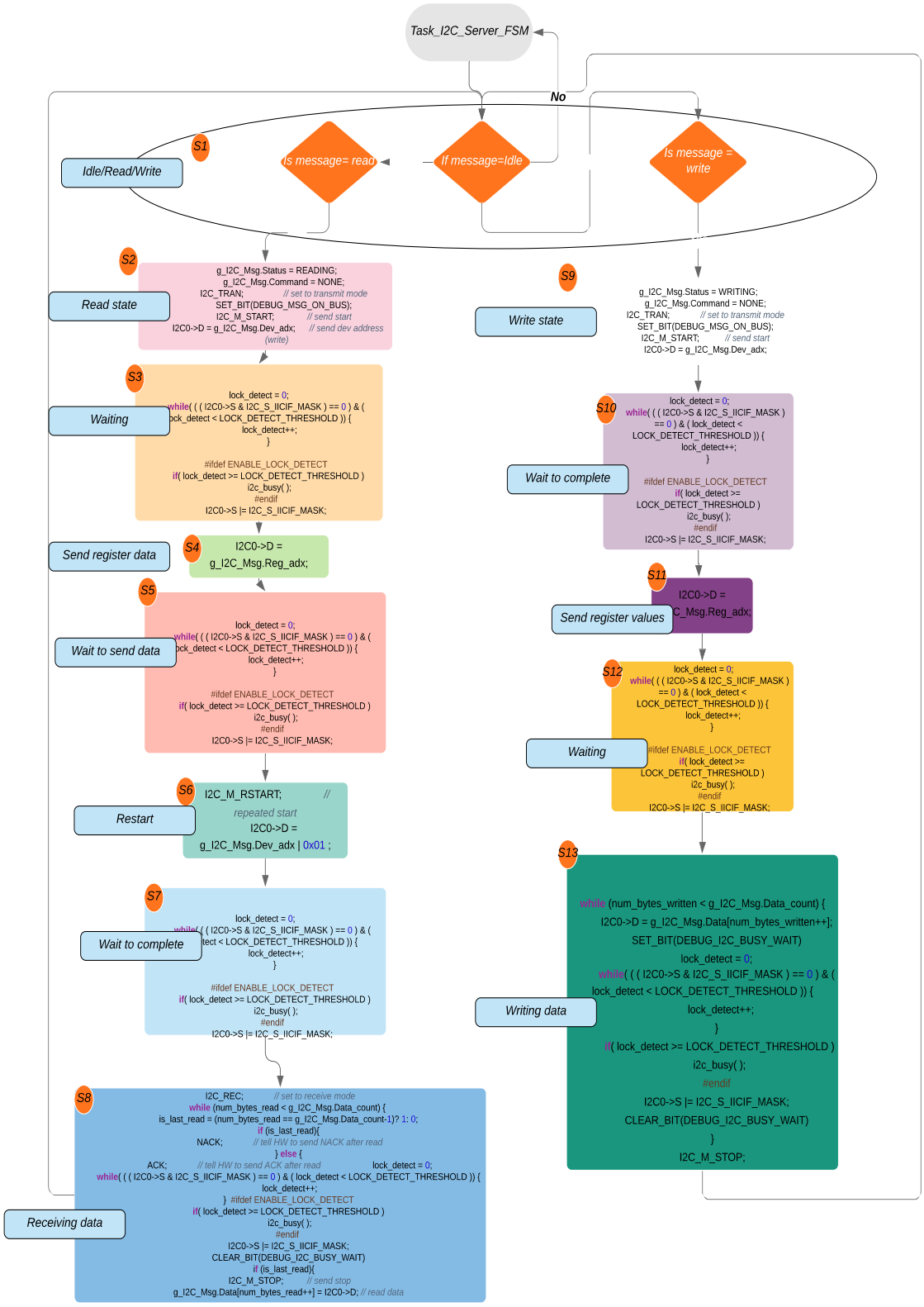
Q2. Screenshot of the write operation.



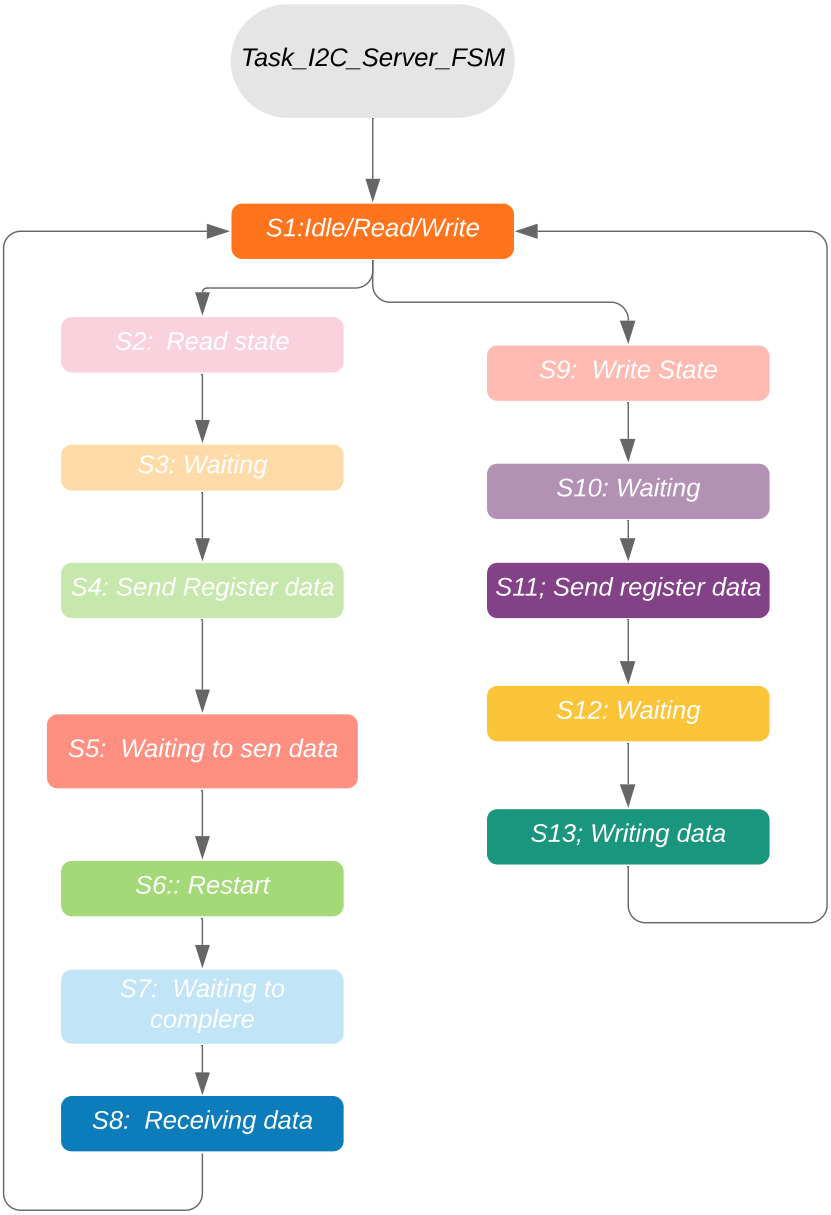
The message on bus is active for 71,4 us for one message

The I2C busy wait is active for 101.1 us

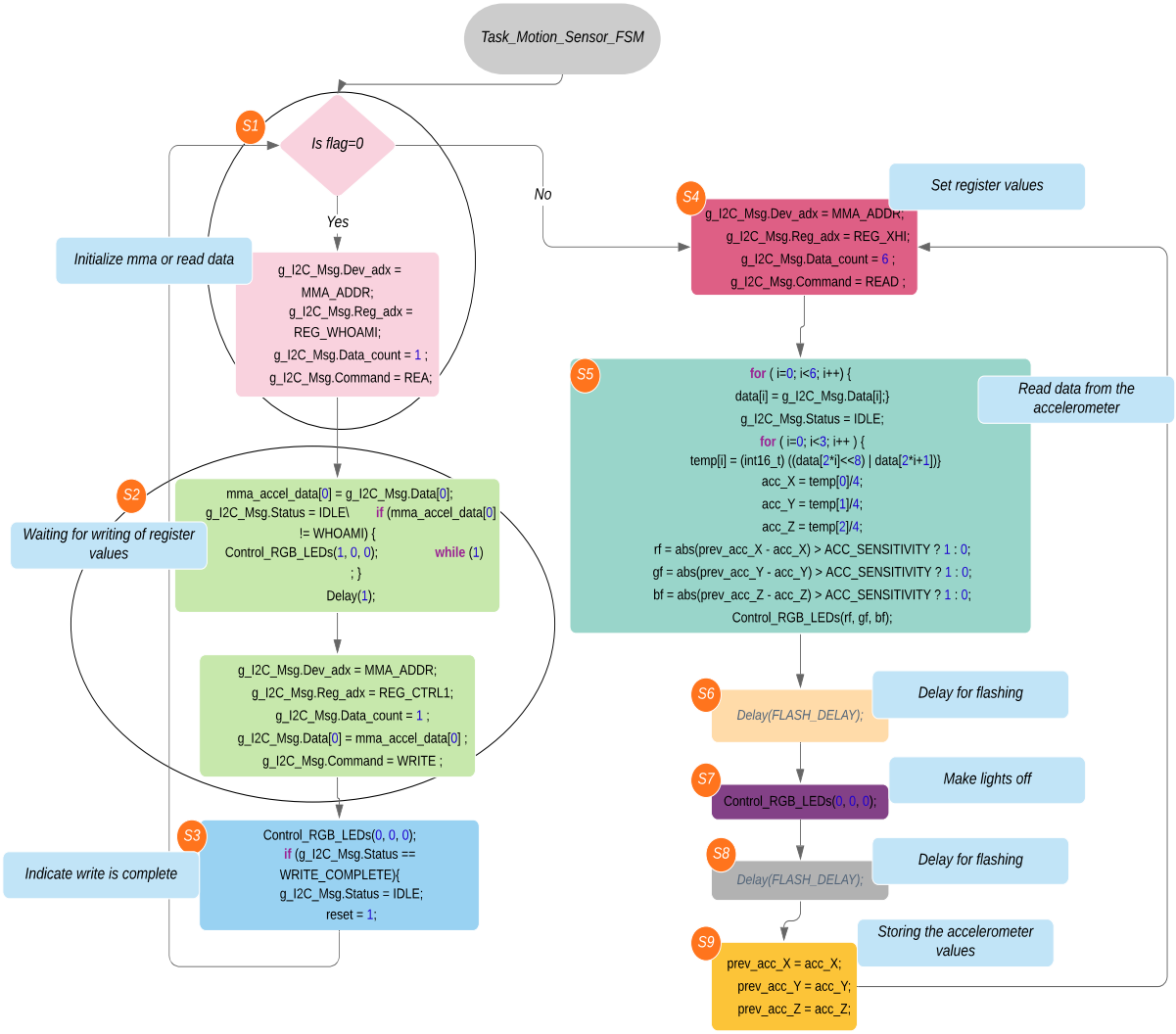
Q3.
Flowchart of Task_I2C_Server_FSM



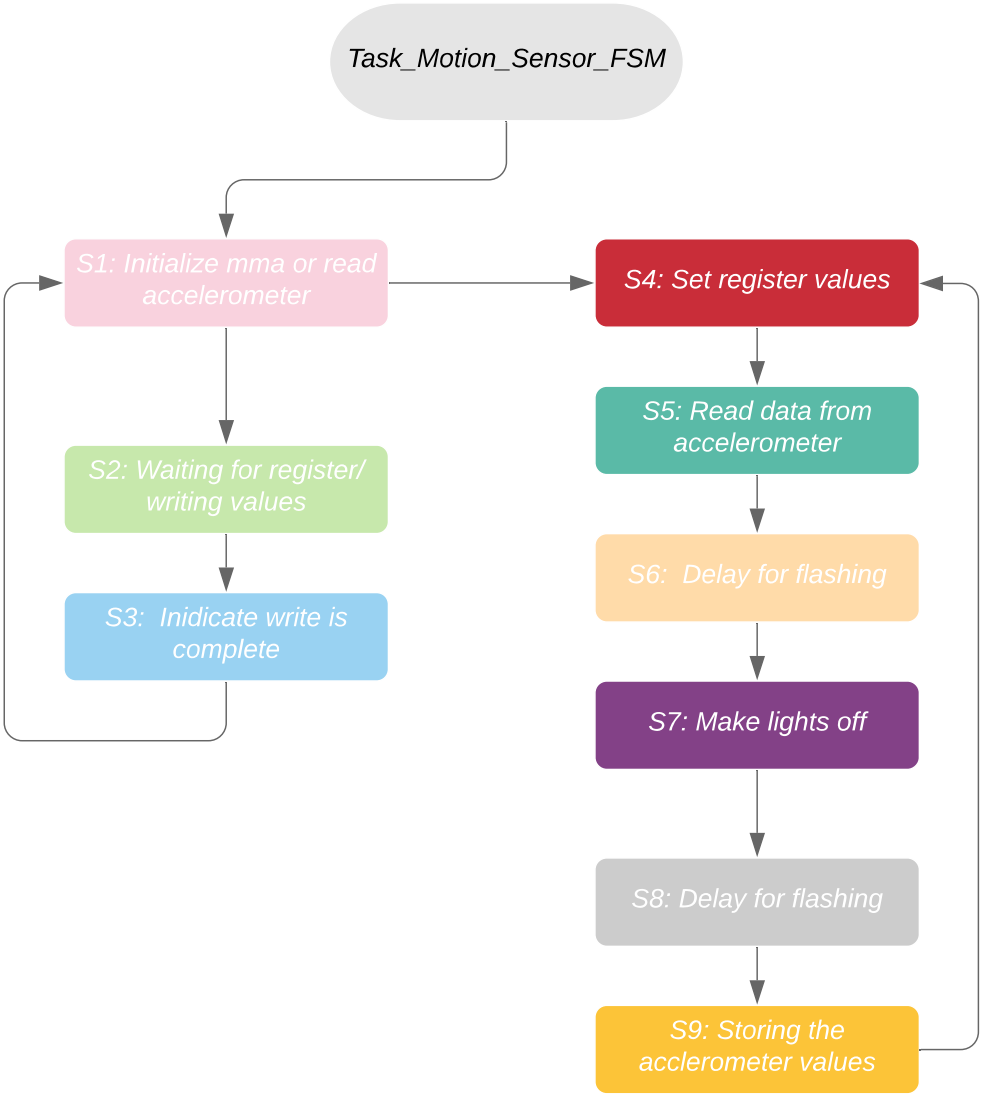
State Transition diagram of Task_I2C_Server_FSM



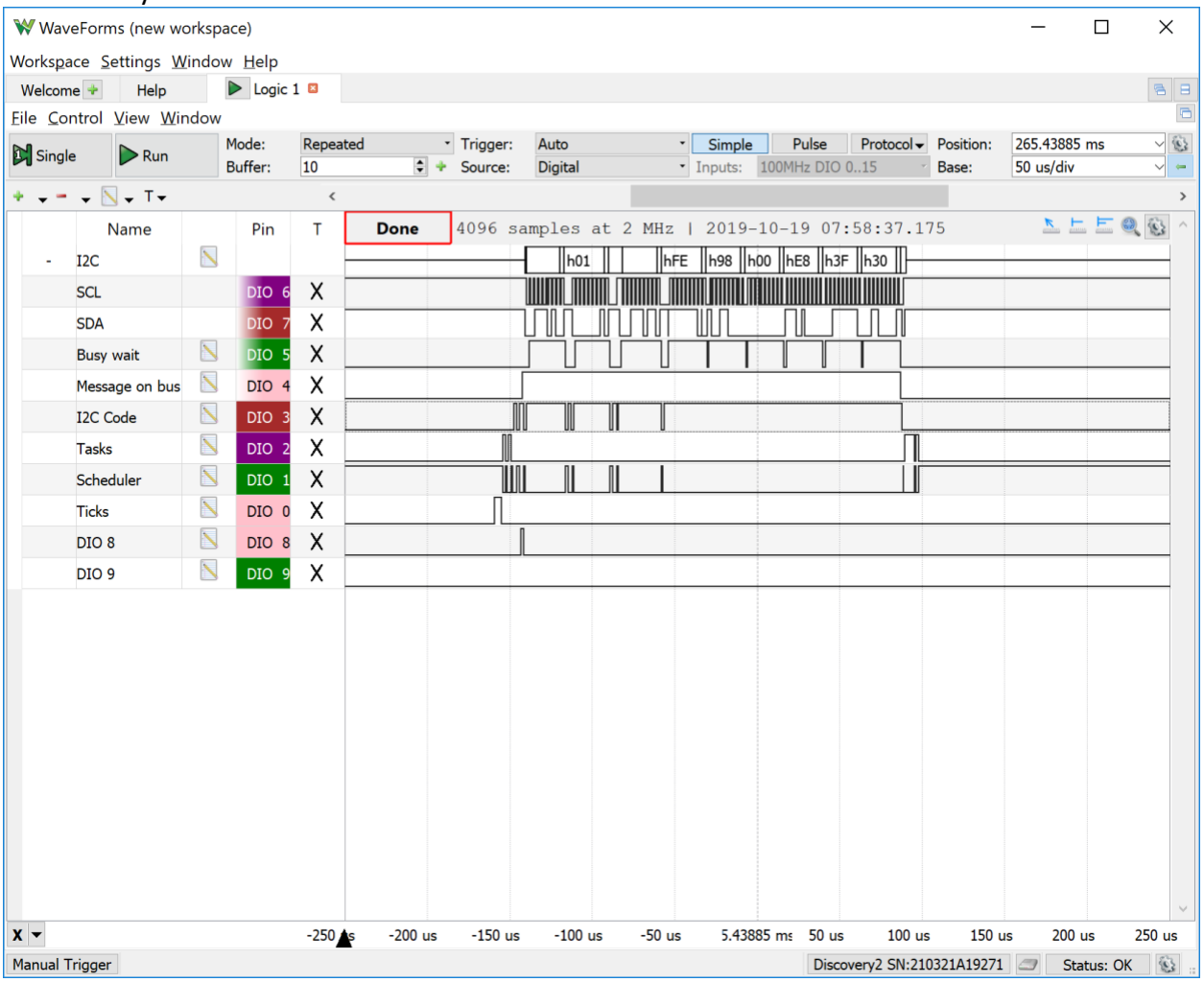
Q4. Flowchart of Task_Motion_Sensor_FSM



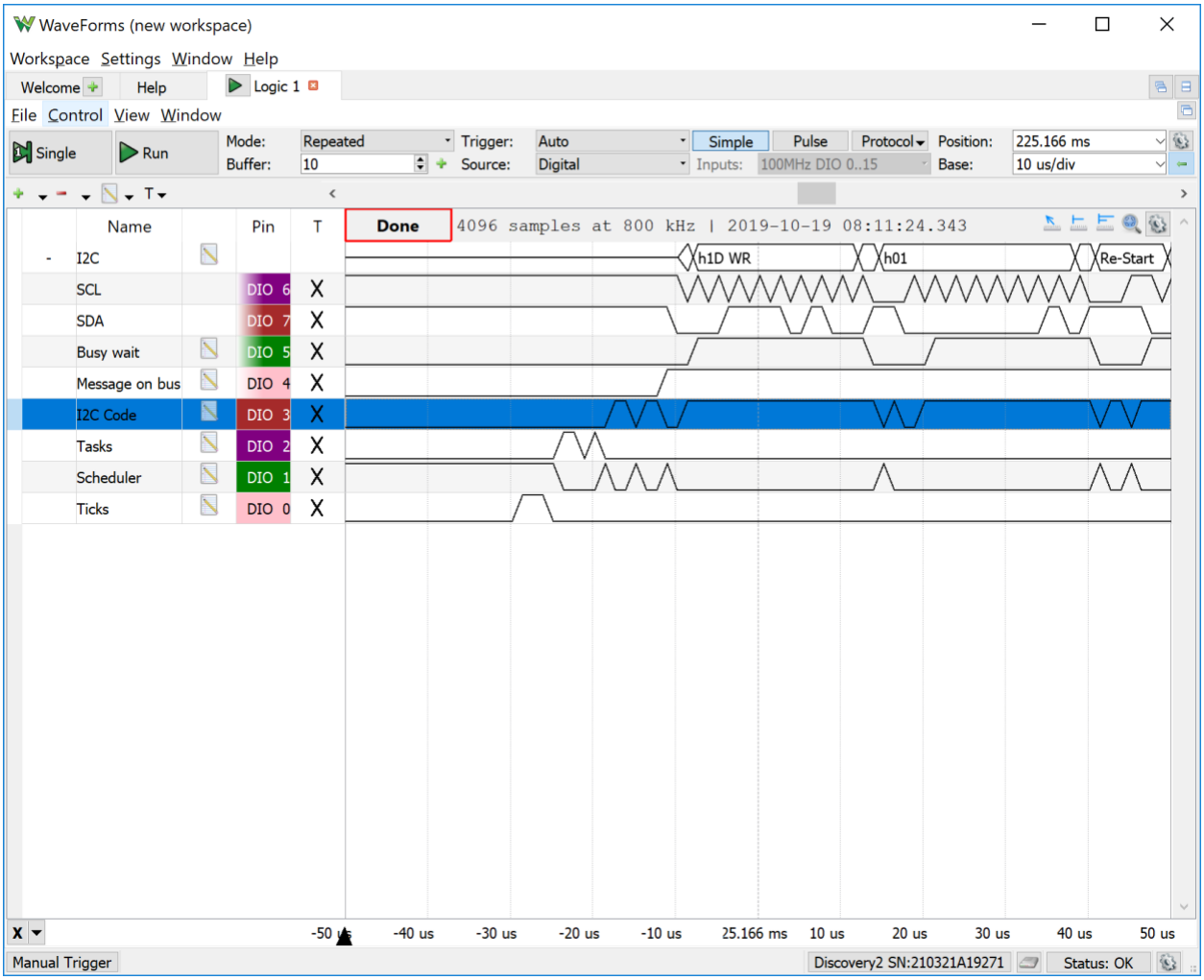
State Transition diagram of Task_Motion_sensor_FSM



Q5. Screenshot of the read operation for the FSM model clearly shows the scheduler being called every time the code returns to the task table.



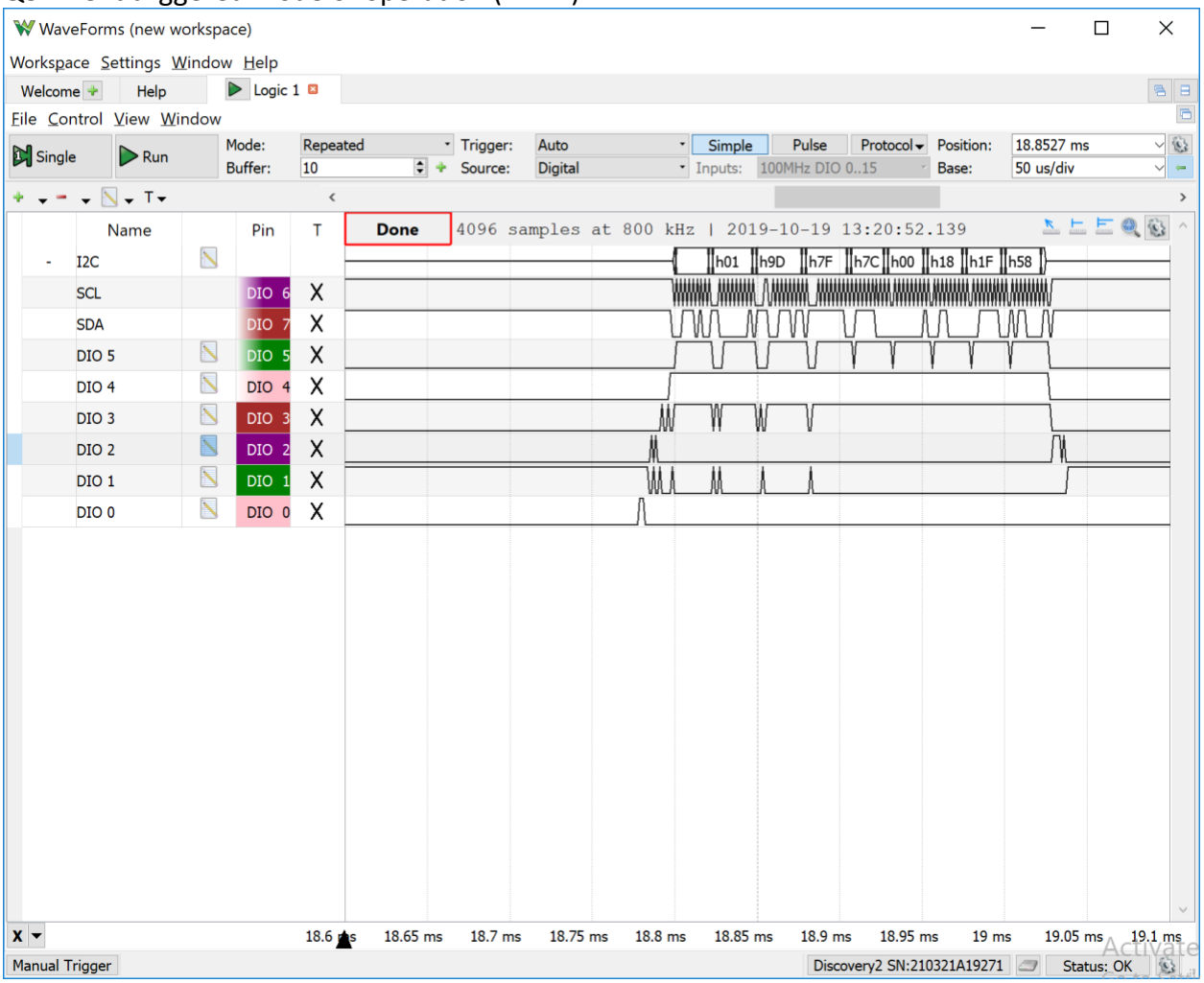
Q6. Write command for the Finite state machine



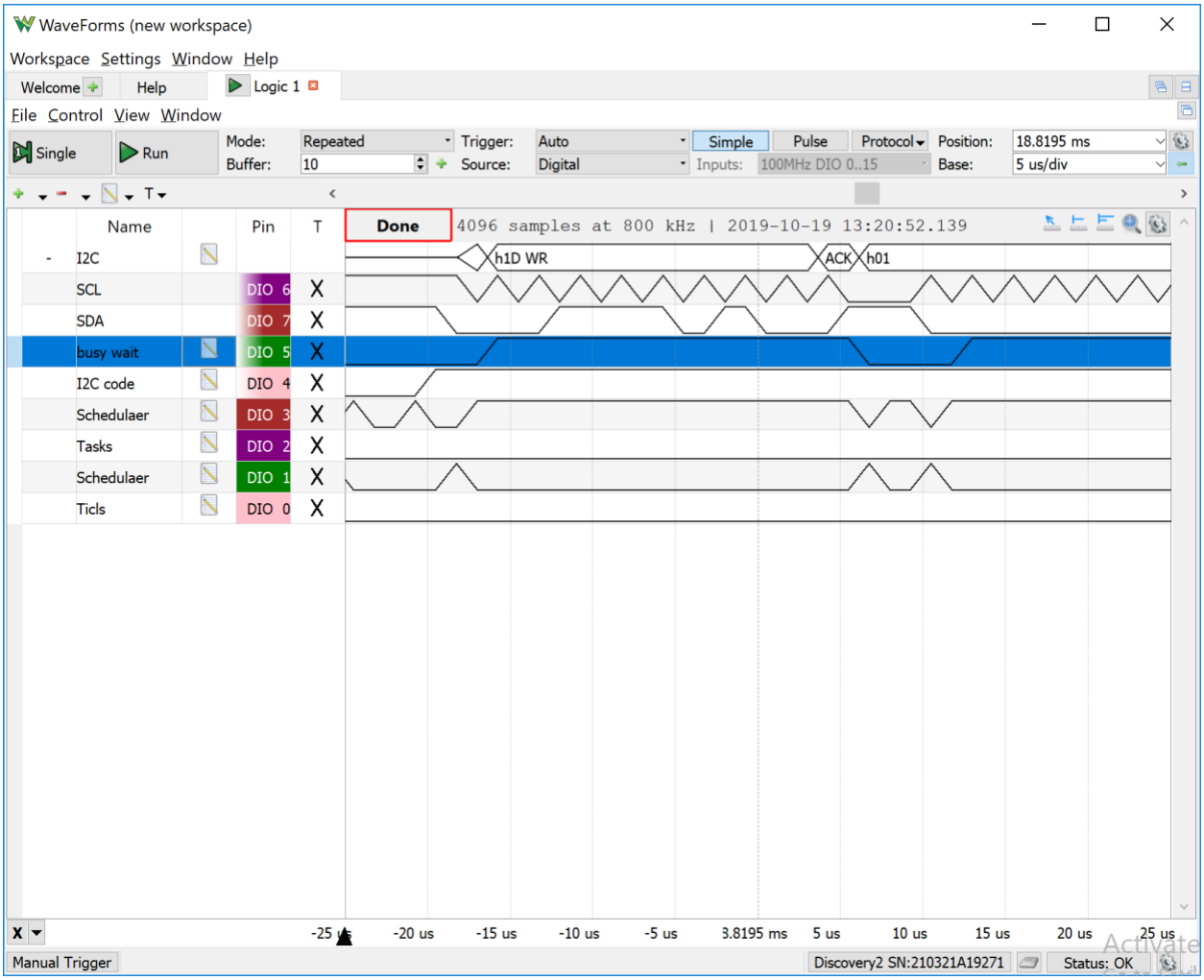
Q7.

The basic approach to triggering the tasks when the event occurs is to increase the releases pending for the particular task as soon as the event is triggered rather than periodically. For this purpose, I added RTCS_Task_Table[1].ReleasesPending++ during every state as mentioned in the design. I did not use the g_I2C_Message_Client. Instead I used an approach by which the period of the task was set to zero that is the period doesn't become set to the number of ticks but the task is released as soon as it reaches a state where it is ready for the next state. This approach reduces the time needed for processing the state and goes to the next state immediately.

Q8. Event triggered mode of operation (READ)



Q9. Event triggered mode of operation(WRITE)



Q10.

The delay in question 5 was found to be 78.1 ms and the delay in question 8 was found to be around 180 us. The excess delay was basically because in the normal FSM code ticks take place every 2 ms because of the frequency of the ticks that was set. In event triggered mode the task is released as soon as the state is observed.