

Title Page

Lab #: Lab #6

Course: CSE379

Partner Names: Solomon Richards, Jah Ulerie

Partner Usernames: solomonr, jahsiemu

Lab Section: R1

Date: 04/05/23

Table of Contents

Section 1.....	3
Division of Work.....	3
Section 2.....	3
Program Overview.....	3
Program Summary.....	3
High Level Flowchart.....	4
Section 3.....	4
Subroutine Descriptions.....	4
Section 4.....	5
Subroutine Flowchartst.....	5

Section 1

Division of Work

Solomon: uart_interrupt_init, UART0_Handler, simple_read_character, Switch_Handler

Jah: gpio_interrupt_init, Switch_Handler, timer_interrupt_init, Timer_Handler

Section 2

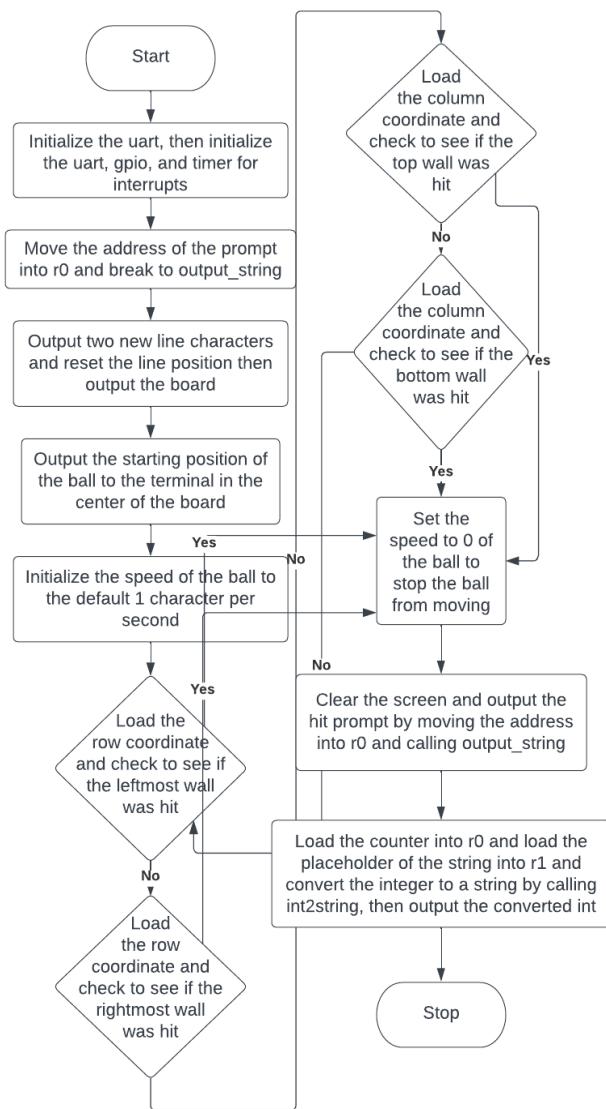
Program Overview

- Plugin Tiva board, open putty, and run instructions to set up the input and output terminal.
- Once you're set up you will be prompted to press either the sw1 button on the Tiva board or any key on the keyboard or q to quit. The putty terminal will display the number of times you pressed a key and the sw1 button along with bar graphs to represent them.
- You can also use any of the subroutines that were previously defined in the labs prior to this (uart_init, output_character, read_character, read_string, output_string, int2string).

Program Summary

This lab applies the use of general-purpose I/O, loading and storing to and from memory, type conversion, and serial communication to output the number of times you've pressed certain keys and the sw1 button. We also use these components to represent the number of times you pressed the key or buttons as bar data allowing the user to compare it.

High-Level Flowchart



Section 3

Subroutine Descriptions

uart_init - initializes the uart for the user

uart_interrupt_init - initializes the uart for interrupts to occur when a key is pressed

gpio_interrupt_init - initializes the sw1 button for interrupts to occur when it is pressed

timer_interrupt_init - initializes the timer for interrupts to occur when the clock ticks

output_character - character passed in r0 is output to the terminal

simple_read_character - reads the character from the uart and stores into r0 and outputs it to the terminal in putty

read_string - uses read_character and the base address passed into r0 to store the string in another specified memory address and calls outputs that string to the terminal in putty

output_string - uses output_character and the base address passed into r0 to print the entire string to the terminal

int2string - converts the integer in r0 to string and stores the string at the address placed in r1

UART0_Handler - handles the interrupt when a key is pressed, in this case it loads the direction that the ball should be moving into memory for the timer to use

Switch_Handler - handles the interrupt when sw1 is pressed, in this case it increases the speed at which the ball should move and stores the new interval into General Purpose Timer Interval Load Register.

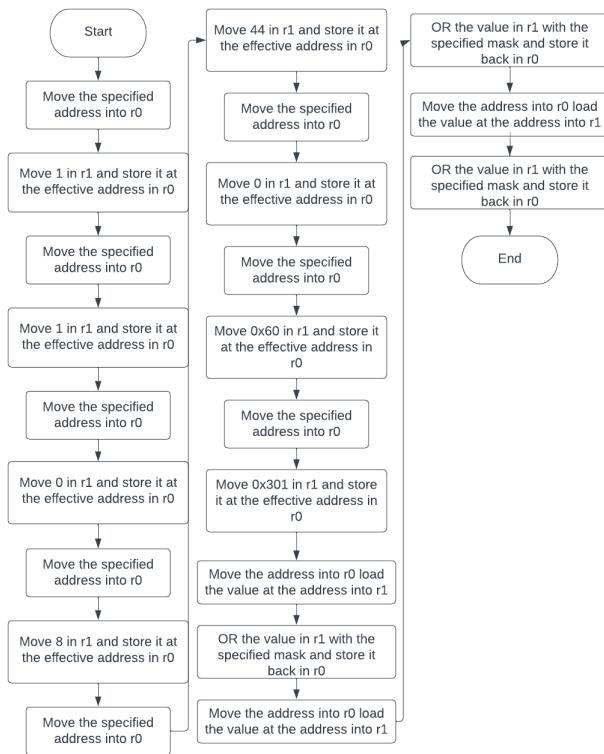
Timer_Handler - Moves the ball at the desired speed and direction everytime the clock ticks

lab6 - initializes the uart, and both interrupts, prompts the user on how to play the game, displays the board, and starts the ball at 1 character per second at moving to the right. Waits for the ball to hit on eof the walls and outputs the final number of moves to the terminal

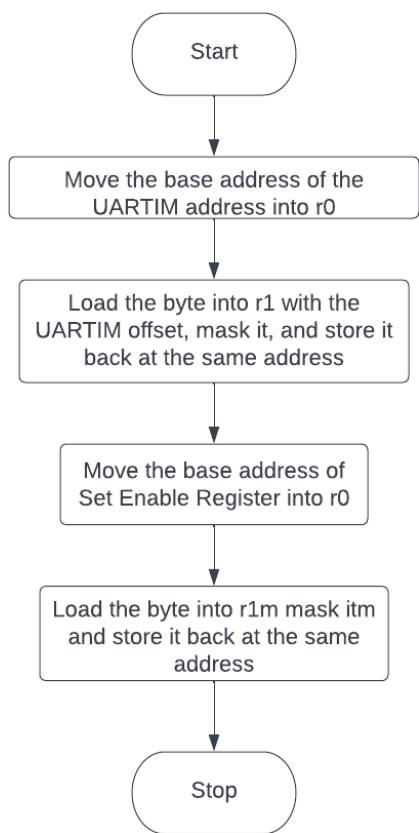
Section 4

Subroutine Flowcharts

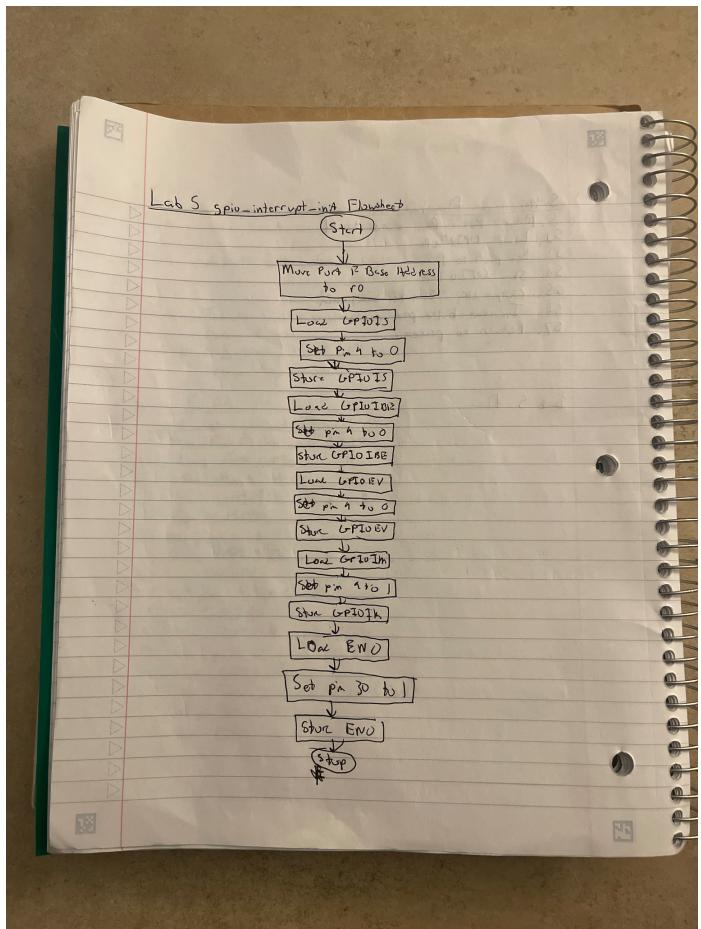
uart_init



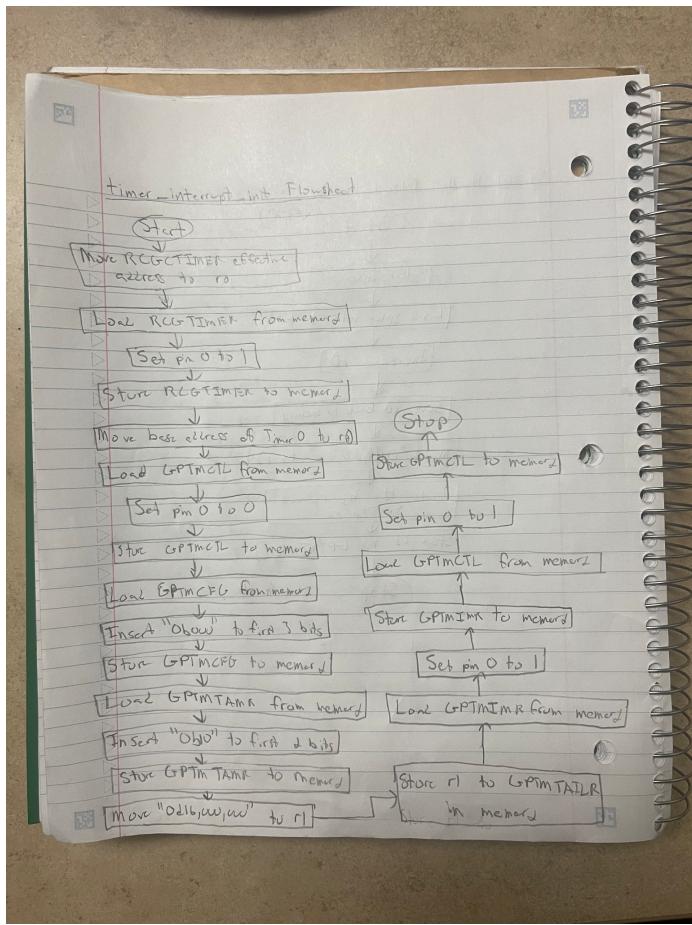
uart_interrupt_init



gpio_interrupt_init

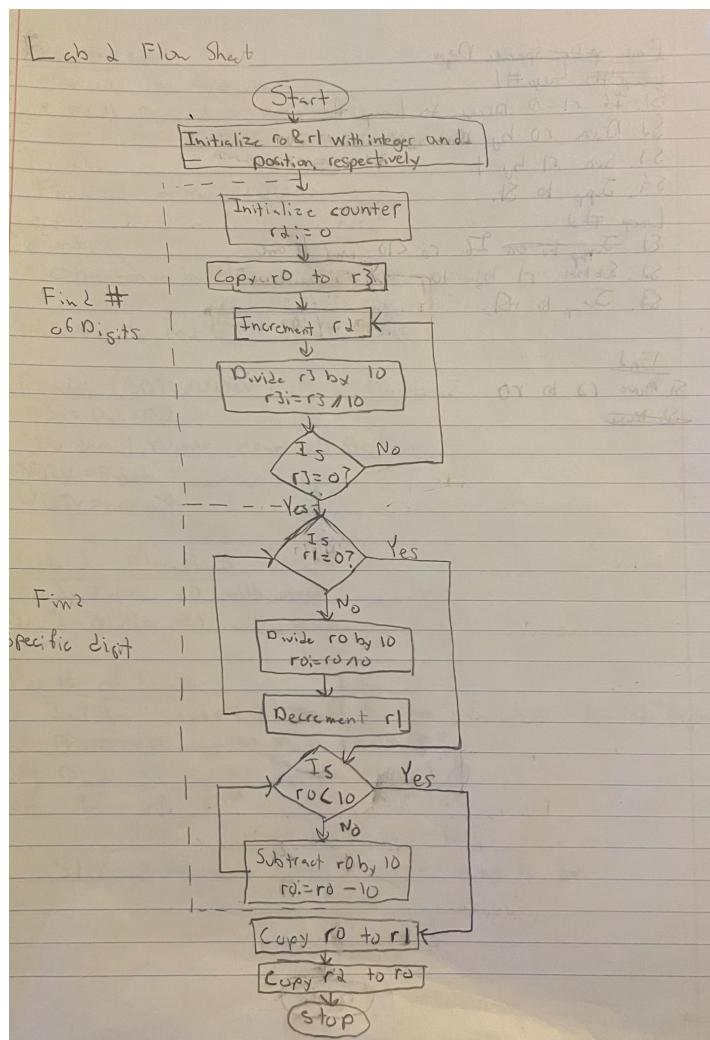


timer_interrupt_init

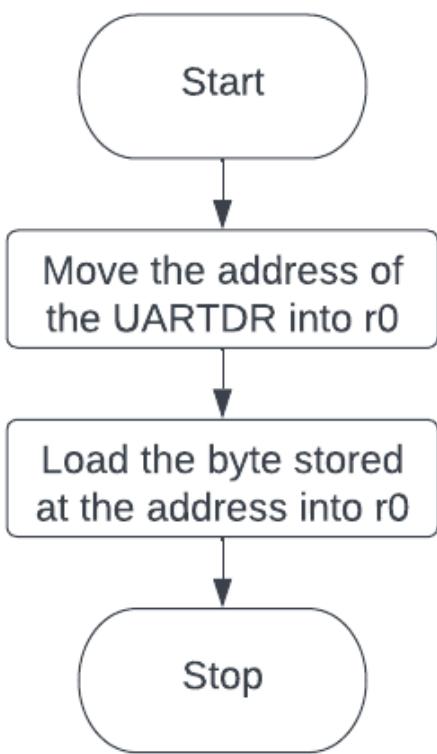


output_character

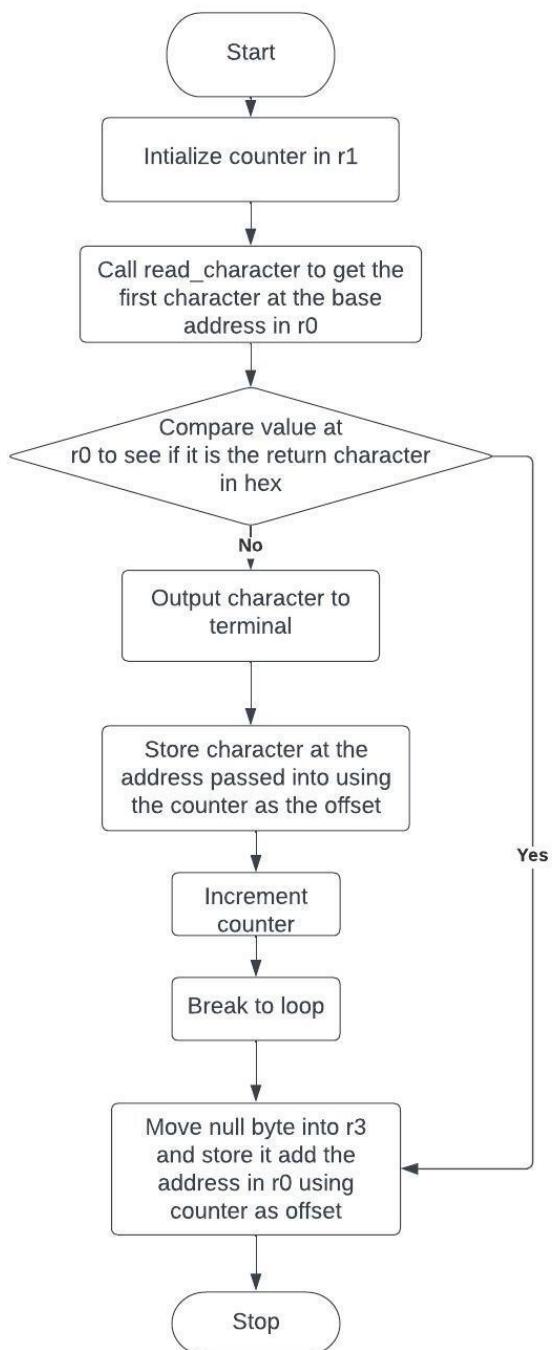
Lab 2 Flow Sheet



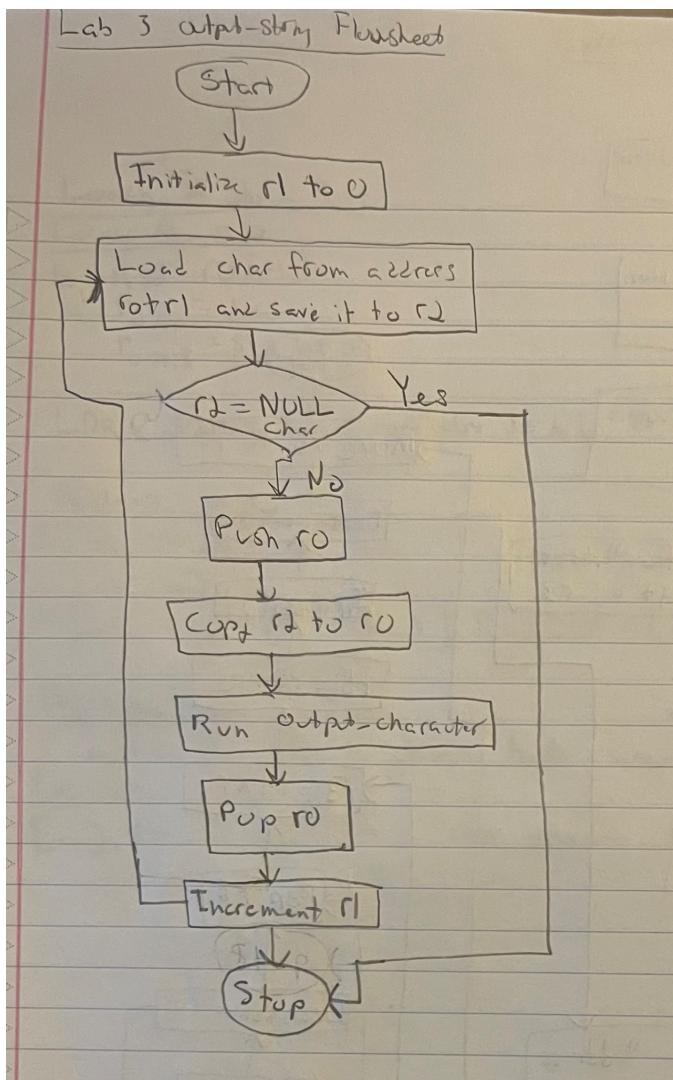
simple_read_character



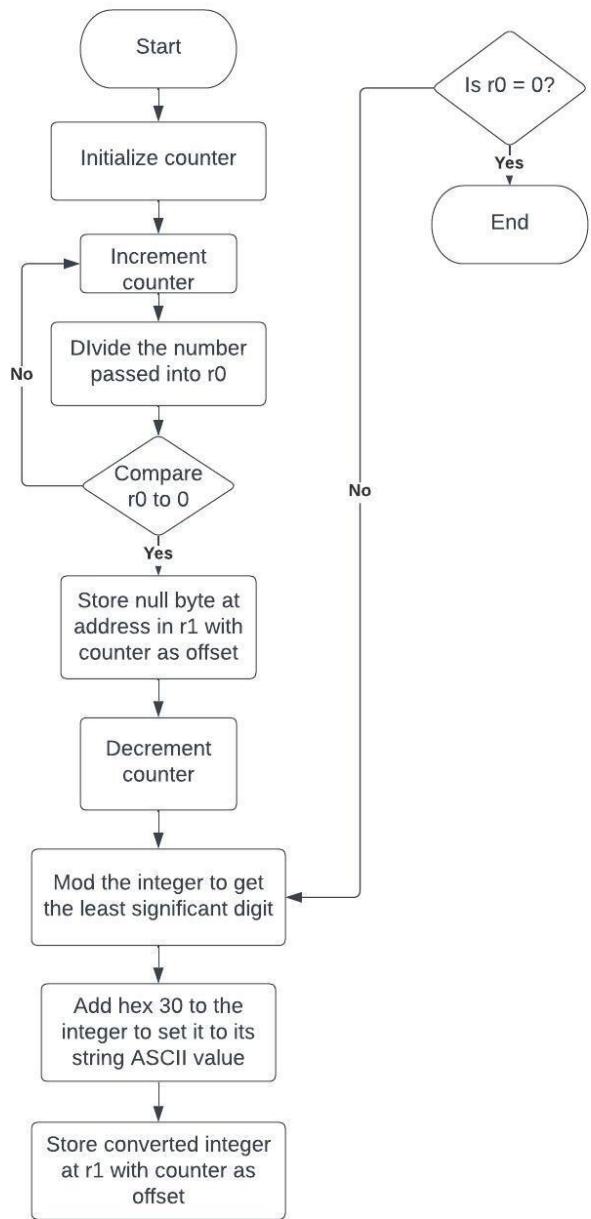
read_string



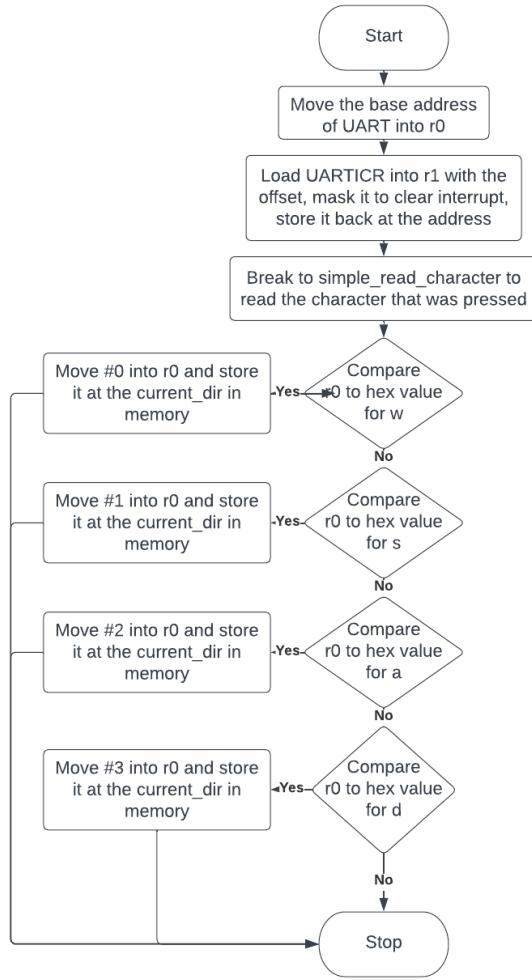
output_string



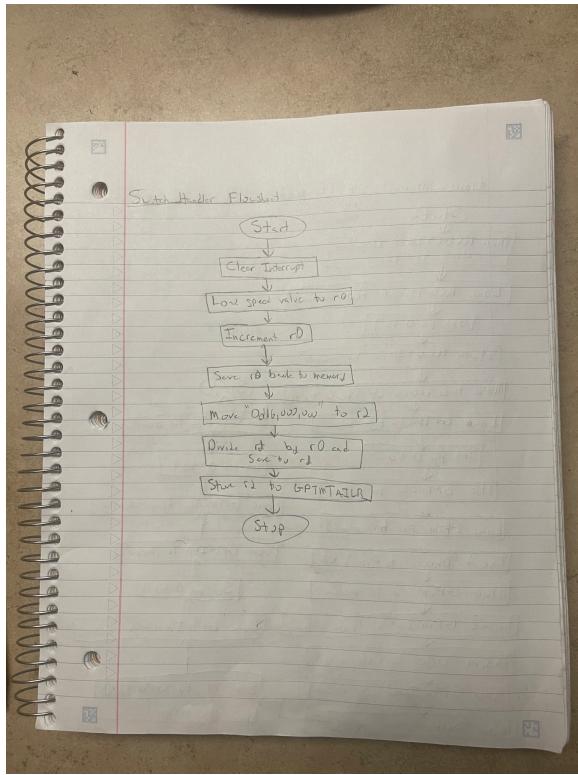
int2string



UART0_Handler



Switch_Handler



Timer_Handler

