

# Lab #4

## CSE-379

Ethan Zemanick  
ethanzem

Felix Wilking  
felixwil

Lab Section: 12:30 - 2:00  
March 7, 2023

## Table of Contents:

<b>Section 1: Division of Work</b>	<b>3</b>
<b>Section 2: Program Overview</b>	<b>3</b>
Section 2.1: Program Overview	3
Section 2.2: Program Summary	3
Section 2.3: High Level Flowchart	5
<b>Section 3: Subroutine Descriptions</b>	<b>6</b>
<b>Section 4: Subroutine Flowcharts</b>	<b>7</b>
read_tiva_push_button:	8
read_from_push_btns:	9
illuminate_LEDs:	10
illuminate_RGB_LED:	11
read_from_keypad:	11
output_character():	14
read_string():	14
output_string():	16
read_character():	17
int2string():	18
string2int():	19

# Section 1: Division of Work

We did the entirety of the code during two lab sessions plus one hour of open lab hours. We used Pair programming contribution for the code was 50/50. Both worked on flowsheets.

Felix: Usually inputting the code. Added comments to the code.

Ethan: Designed and coded the menu, initialization of the gpio ports, and assisted with thorough understanding of documentation.

## Section 2: Program Overview

### Section 2.1: Program Overview

1. Start running the program.
2. Directions on how to use it will be displayed in PuTTY.
3. Select the subroutine that you would like to test, by typing in a number 1 through 5.
4. For any of the selected subroutines, instructions on how to operate them will be displayed in PuTTY.
  - a. For `read_tiva_push_button`: Calling this subroutine will prompt you to press sw1 on the Tiva board, and the result of this press will be stored in r0, 1 for pressed 0 for not pressed.
  - b. For `read_from_push_btns`: Calling this result will prompt you to press the switches on the daughter board, the result of them being pressed will be stored in r0. The rightmost button corresponds to the first bit, leftmost corresponds to the fourth bit, where pressed is a 1 and not is a 0.
  - c. For `illuminate_LEDs`: Calling this will prompt you to enter a number 0 to 15, where the number will be displayed on the red LED's in binary format, with the rightmost being the first bit and the leftmost being the fourth bit.
  - d. For `illuminate_RGB_LED`: Calling this will prompt you to enter a number 0 to 5, where the number input will determine the color shown.
  - e. For `read_from_keypad`: Calling this will prompt you to hit a number on the keypad, with the result being stored in r0. However our implementation broke down in implementing the menu and trying to switch to using the keypad initialization from piazza
5. After the conclusion of the subroutine, press q to quit, or any other key to start again and select a subroutine to run.
- 6.

### Section 2.2: Program Summary

This lab is intended to add functionality of the daughter board keypad, push buttons, and LEDs, along with the Tiva push buttons and RGB Led, to our library. Our program starts by prompting the user to select a specific subroutine to run, and is selected by typing in the number of the selected subroutine into PuTTY. Once done, the user will again be given instructions or a prompt on how to test the subroutine and see its results.



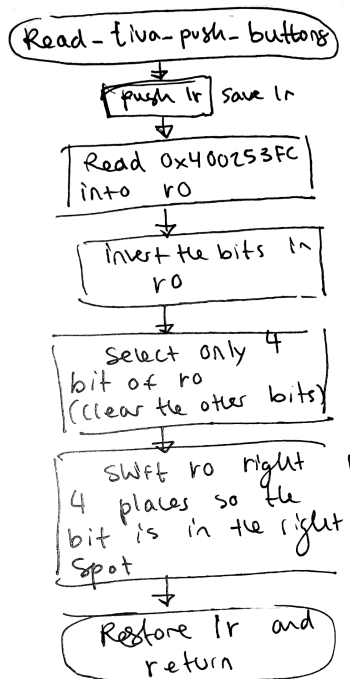
## Section 3: Subroutine Descriptions

Name	Arguments	Return Values	Description
lab4	none	none	Main routine that calls all the other subroutines. May be executed repeatedly if the user desires.
read_tiva_push_button	none	r0	Outputs the value of the push button on the tiva board into r0 (pressed/unpressed, 1/0)
read_from_push_btns	none	r0	Returns the values of the 4 push buttons on the baseboard in r0. Bit 0-3 each represent a pressed/unpressed state of the 4 buttons.
illuminate_LEDs	r0	none	Sets the values of the 4 LEDs on the base board.
illuminate_RGB_LED	r0	none	Sets the color of the RGB Led on the Tiva board. Uses the first 3 bits of r0 corresponding to 0-R, 1-G, 2-B
read_from_keypad	none	r0	Returns the number that was pressed on the keypad as an integer.
read_string:	pointer to char buffer (char*)	none	Reads a string from the serial port. Characters are echo'd back as they are typed. The result is stored at the address provided in r0
output_string:	pointer to string to output (char*)	none	Output a string to the serial port.
uart_init:	none	none	Sets a bunch of mmio

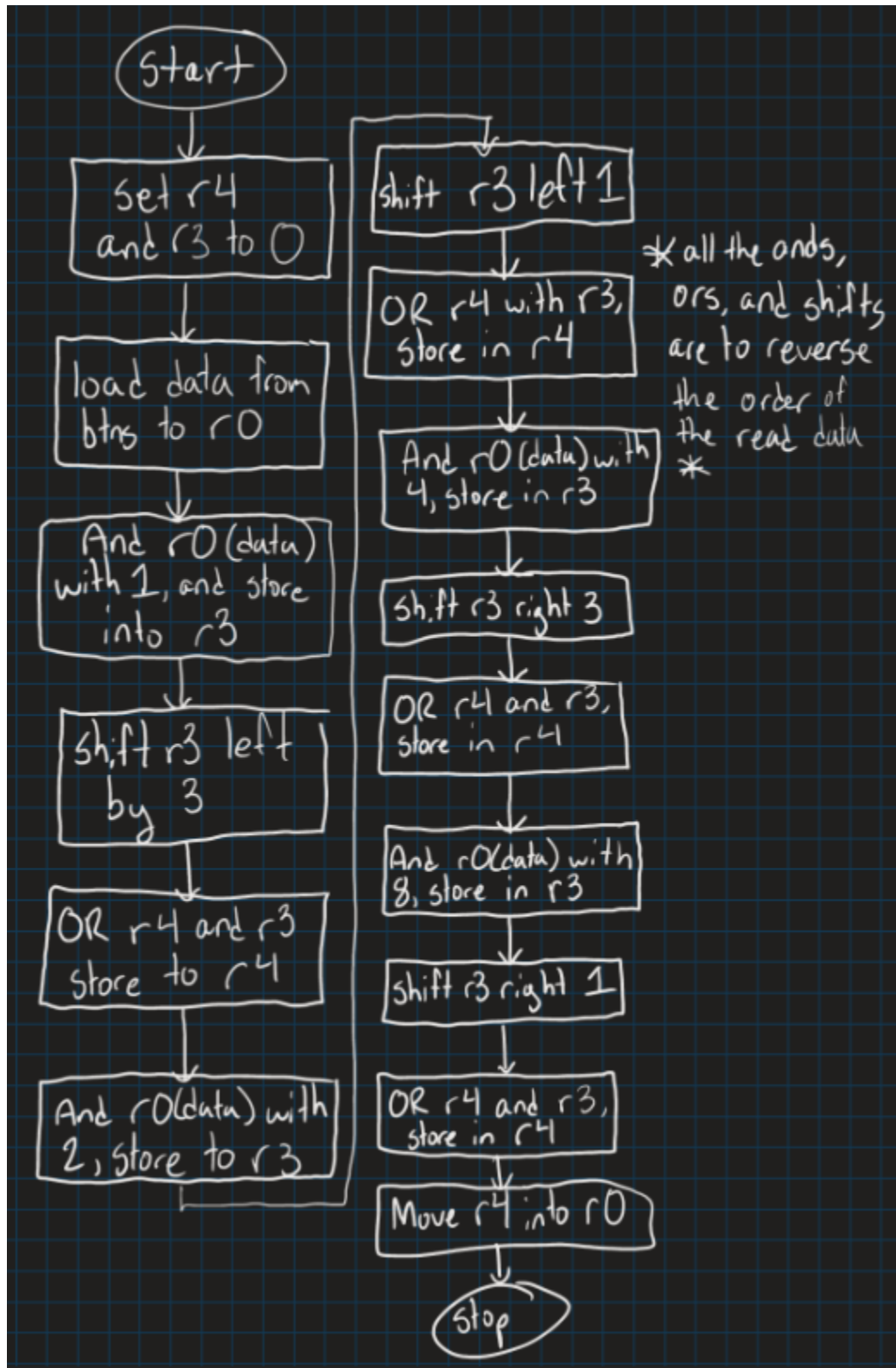
			addresses to specific values to setup the serial connection.
int2string:	integer to convert (int), pointer to char buffer (char*)	none	Converts the provided integer into a string stored at the provided address.
string2int:	pointer to string to convert (char*)	the integer (int)	Converts the provided string into an integer and returns it.
output_character:	character to output (char)	none	Sends a character over uart. (called repeatedly by output_string)
read_character:	none	character (char)	Receives a character over uart. (called repeatedly by read_string)

## Section 4: Subroutine Flowcharts

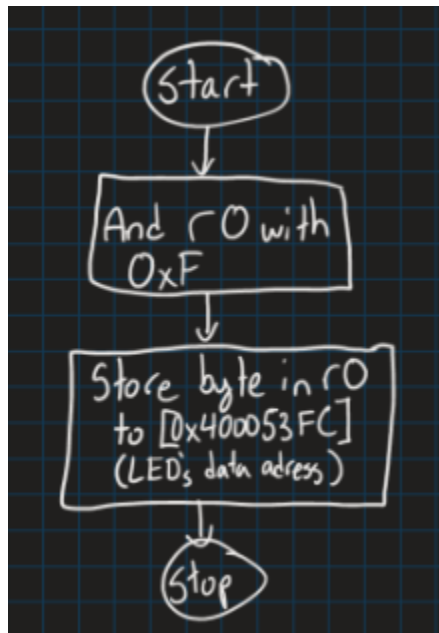
read\_tiva\_push\_button:



read\_from\_push\_btns:

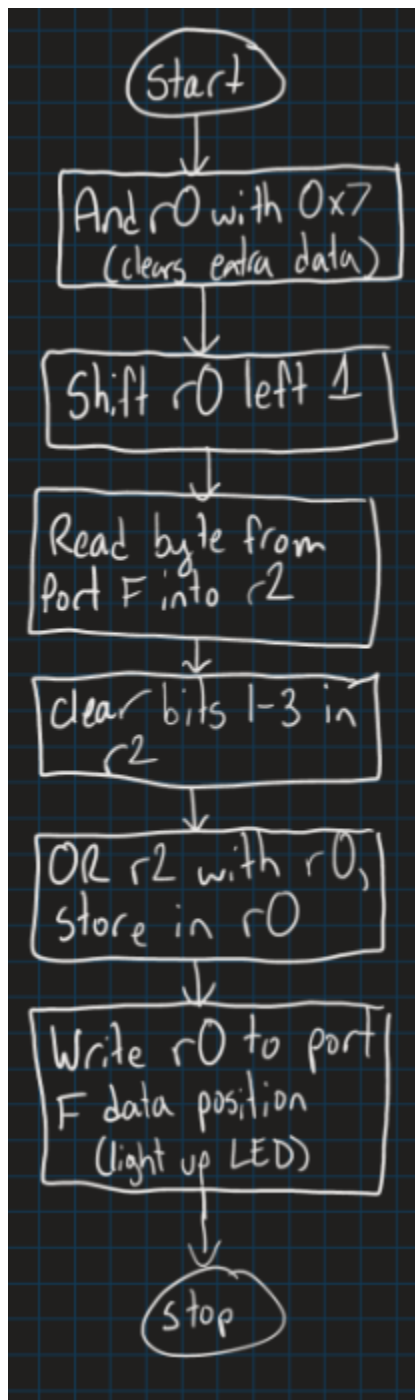


illuminate\_LEDs:

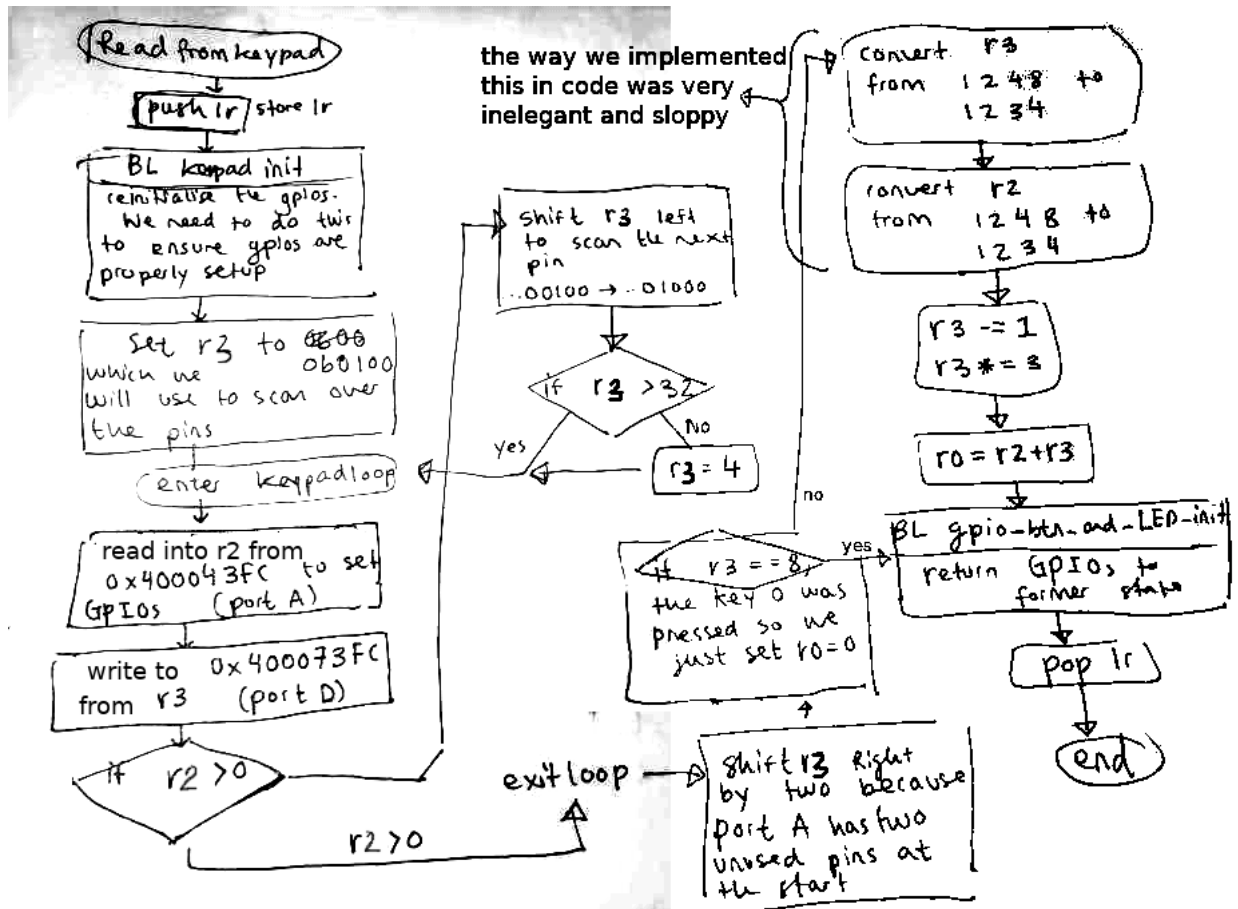




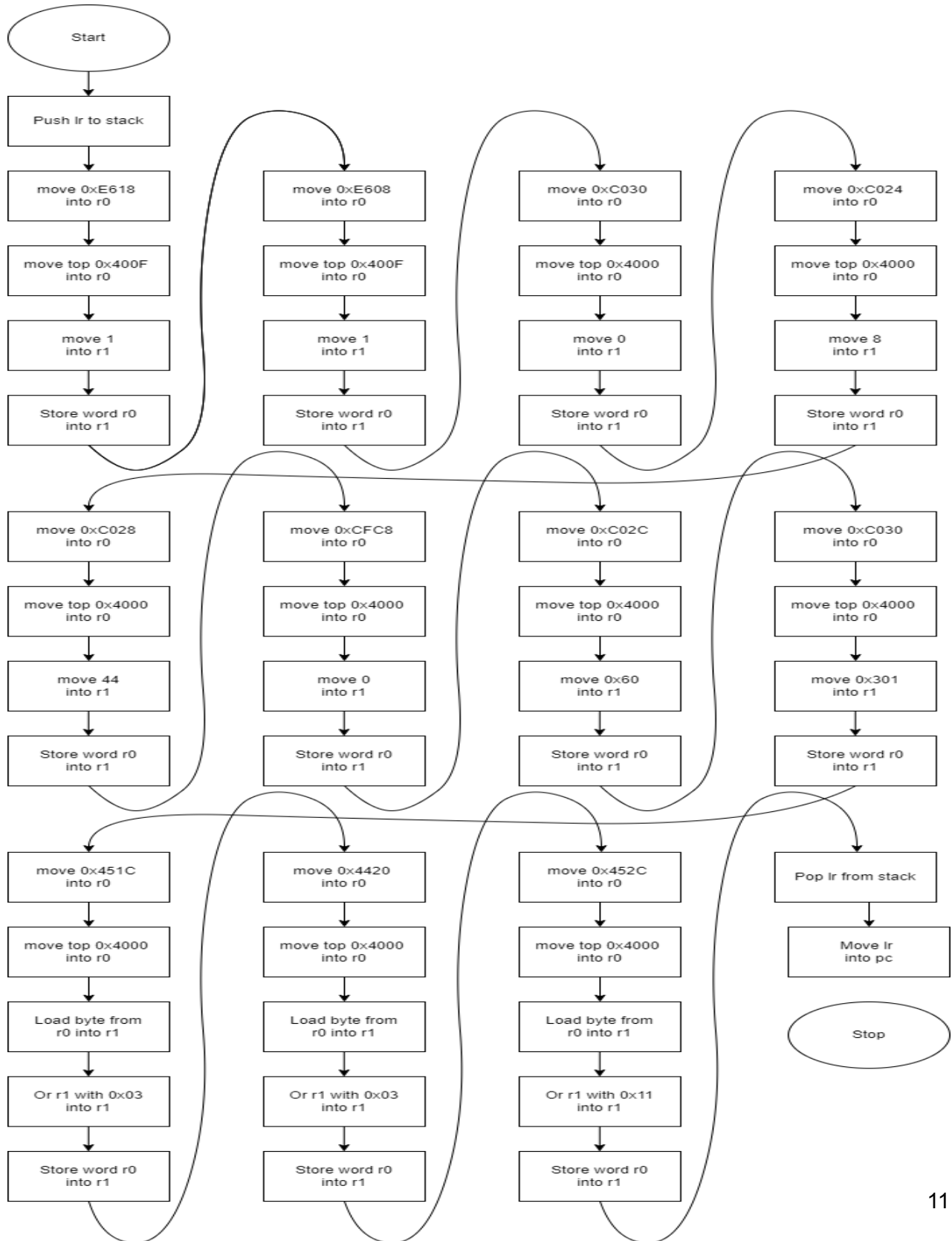
illuminate\_RGB\_LED:



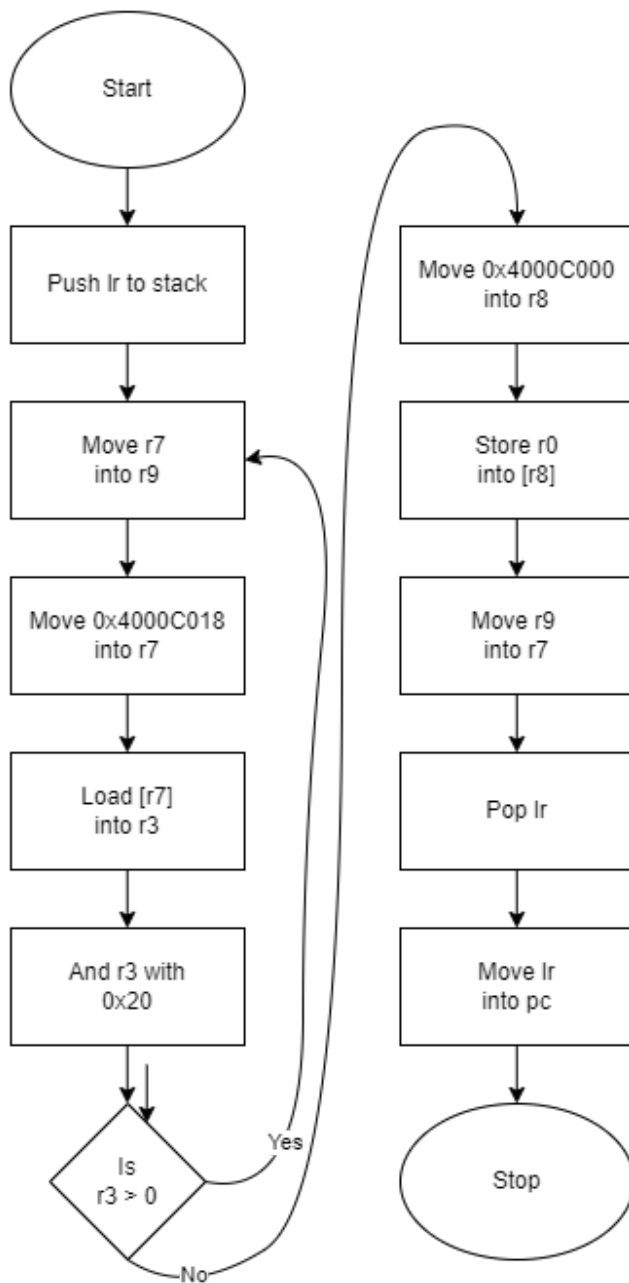
read\_from\_keypad:



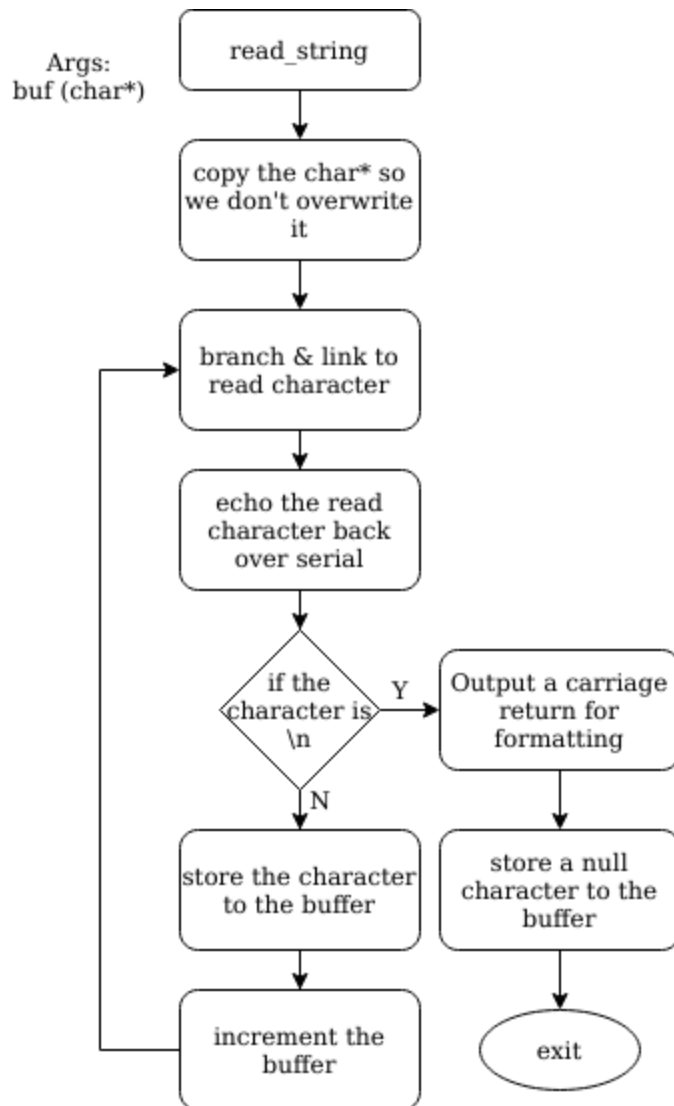
uart\_init():



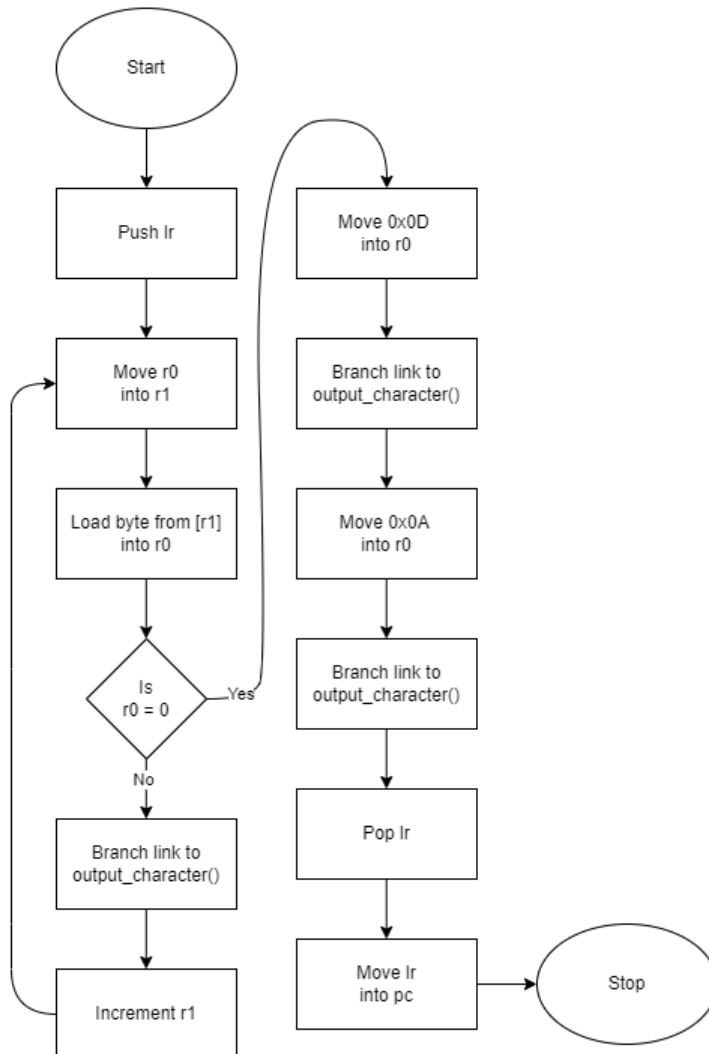
output\_character():



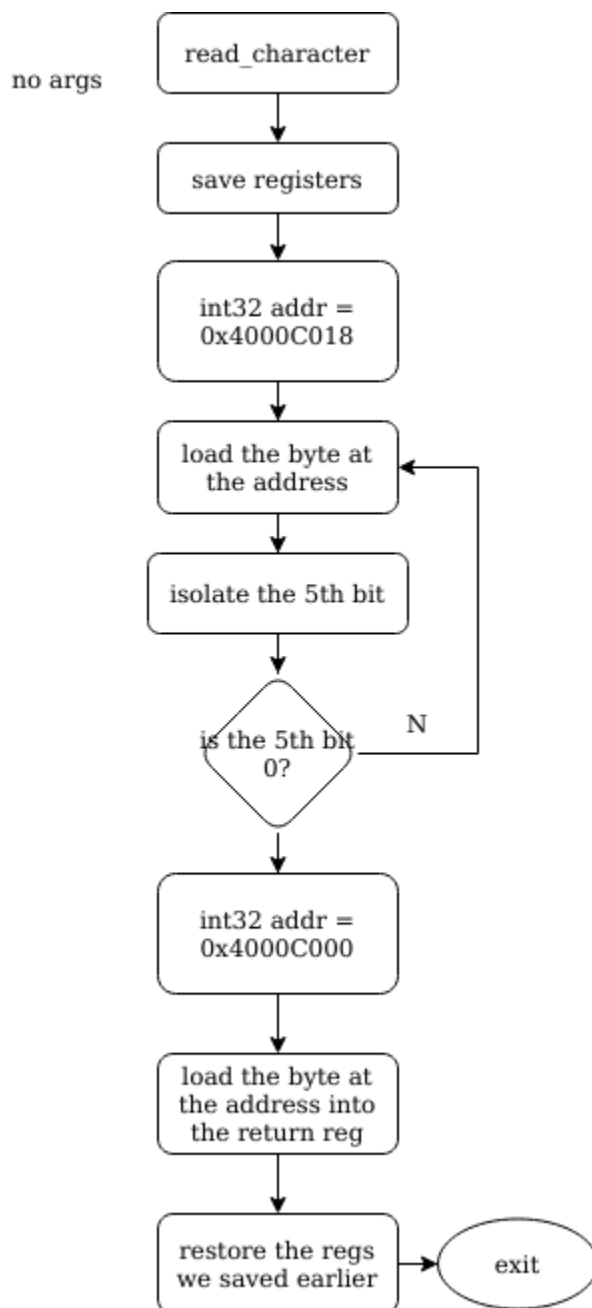
read\_string():



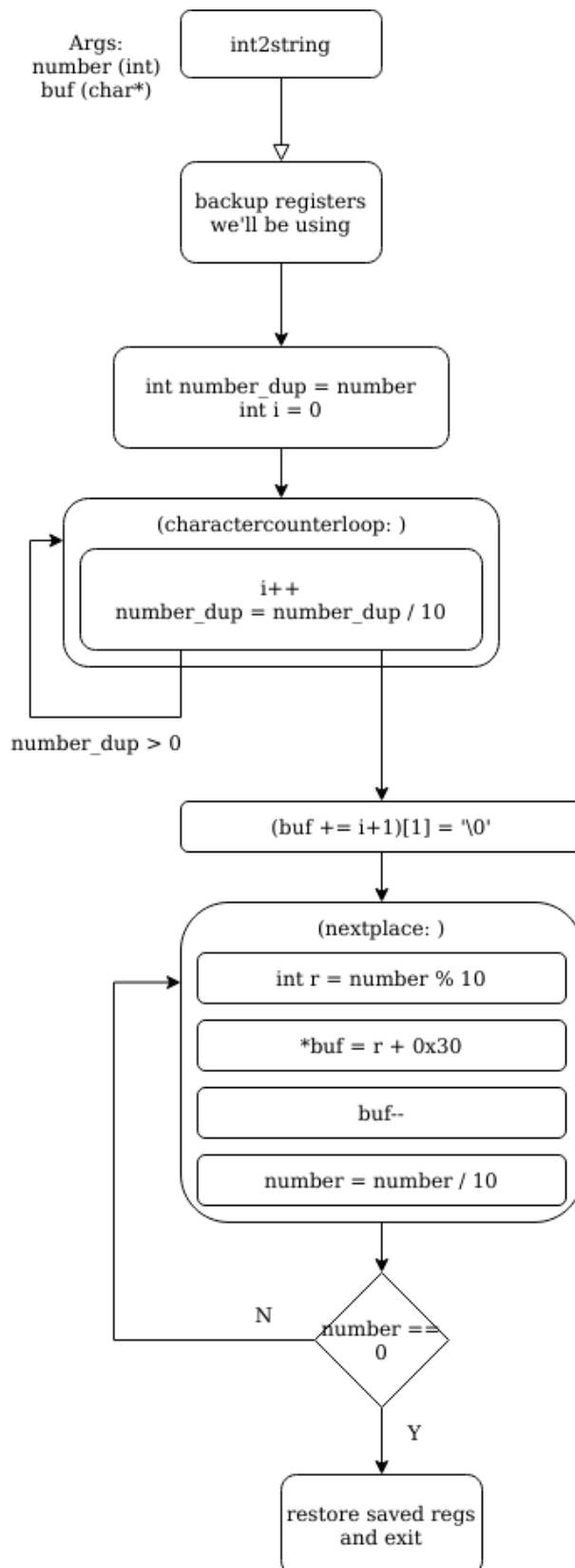
output\_string():



read\_character():



int2string():





string2int():

