Contents

1 Overview

1.1 Objective

The goal of Lab 4 is to allow the user to manipulate various inputs and outputs on the ARM Processor. The user should be able to use the push buttons to display a decimal number on putty, the RGB LED's to display a specific color, and the seven-segment display to display a single hexadecimal digit. The inputs and outputs are to be manipulated through a series of subroutines that exist in a library.s file.

1.2 Usage

A menu displayed on Putty will allow the user to choose between displaying a decimal number through the push buttons, outputting a specific color through the RGB LEDs, and displaying a single hexadecimal digit on the seven-segment display board. User input will be taken through Putty to control the RGB LEDs and the seven-segment display board, while the push buttons on the processor will take in a binary number from the user and show the converted binary number in decimal on Putty.

1.3 Division of Work

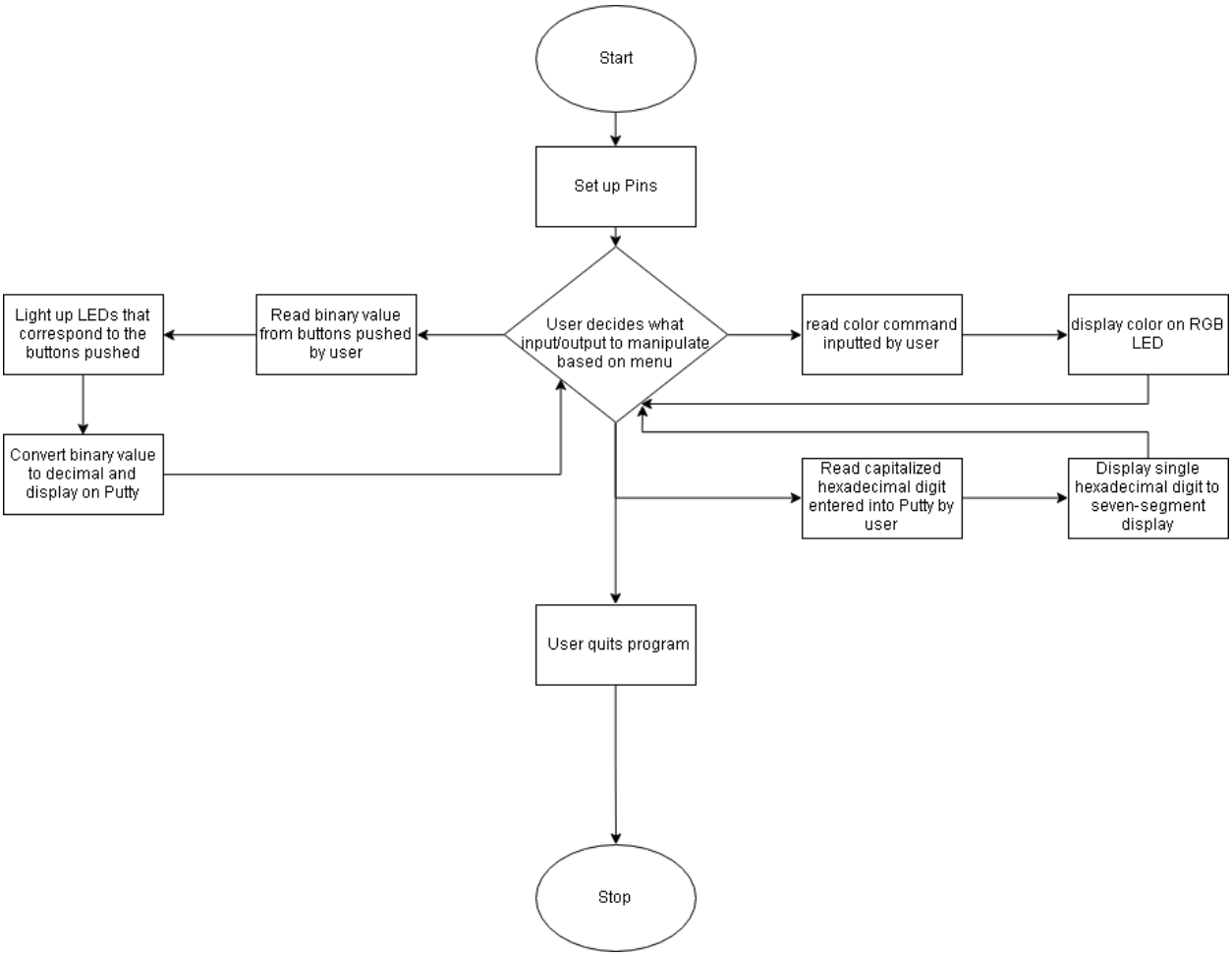
ccaballe readjusted lab3 code to work with current code (read string, write string, etc.). tylerhan completed code that handled LEDs and the push buttons. Everything else was developed together.

1.4 Program Flow Chart

1.4.1 Summary

After the initial setup of the inputs and outputs, the user decides what to manipulate. They have the option of controlling the push buttons and LEDs to display a decimal value to Putty. They can also display a specific color on the RGB LED or display a single valid hexadecimal digit to the seven-segment display. The inputs to display to the RGB LED or the seven-segment are entered through Putty by the user, while buttons on the processor are used to display a decimal value onto Putty. The user can perform any of these tasks until they decide to quit the program.

1.4.2 Flow Chart



2 Subroutines in library.s

2.1 uart init

Initializes the UART.

2.2 read string

Takes in user inputted string from Putty and stores in memory.

2.3 terminate string

Adds a null terminator to the user inputted string in memory.

2.4 read character

Reads a character typed into Putty and stores it in the receive register.

2.5 output character

Displays the character from the transmit register onto Putty.

2.6 output string

Displays the string stored in memory to Putty.

2.7 new line

Outputs the characters that are necessary for creating a new line in Putty.

2.8 setup pins

Initializes the parts of the board that will be manipulated. This includes: seven-segment display, RGB LEDs, LEDs, and momentary push buttons.

2.9 read from push btns

Reads values from the momentary push buttons by loading the address of I0PIN and accessing pins (bits) 20 to 23.

2.9.1 lXX pushed

Routines that decide which LEDs to light up for every button pressed.

2.9.2 read num from btns

Converts binary numbers attained from momentary push buttons into a decimal value and stores decimal value in register. Steps through each bit and adds the exponential value (2^n where n is the bit position) to the final decimal value. This step is accomplished by the set of subroutines that have 'rnf' prefix (i.e. rnf add 1)

2.9.3 output to decimal

Converts each digit of the decimal value into its ascii representation and displays it on Putty in the correct order. Subroutines with the prefix 'otd' handle the two possible digit positions that may be displayed on Putty.

2.10 change display

Displays a valid single hexadecimal digit that the user inputted on the seven-segment display.

2.11 clear display

Clears the seven-segment display of any single hexadecimal digit that may be on the seven-segment display.

2.12 illuminate ” ” subroutines

The subroutines write to the appropriate pins to activate specific colors (red, green, blue, yellow, purple, white) on the RGB LED.

2.12.1 illuminate reset

Resets the RGB LED to default state.

2.13 led set

Sets LEDs that correspond to push button to a specific state by writing to clear register or set register, which are the high and low subroutines respectively.

2.13.1 clear led

Resets LEDs to default state, off.

3 Subroutines in lab4.s

3.1 lab4

Sets up the inputs and outputs of the processor, as well as the default settings. Default settings include: illuminating white on the RGB LED, all LEDs illuminated, and cleared seven-segment display.

3.2 loop

Waits for the user to input a command that will allow them to utilize a specific input or output on the processor.

3.3 display menu

Interactive menu displayed on Putty to instruct user on how to use program.

3.4 init seven segment

Lets user know they are utilizing seven-segment display output.

3.5 loop seven segment

Routine to decide if user has entered a valid hexadecimal digits and converts user input to properly display on seven-segment display. Routines with prefix "lss" convert the input into proper ascii value.

3.6 init led

Initializes the routines required for displaying a decimal digit to Putty and lighting up the LEDs that correspond to the buttons pressed by the user.

3.7 init color

Lets user know they are utilizing RGB LED.

3.8 loop color

Waits for user to input a valid color command and lights up the RGB LED to the appropriate color.

3.9 color " " subroutines

Lets user know what color they chose and illuminates the light corresponding to the user's choice.