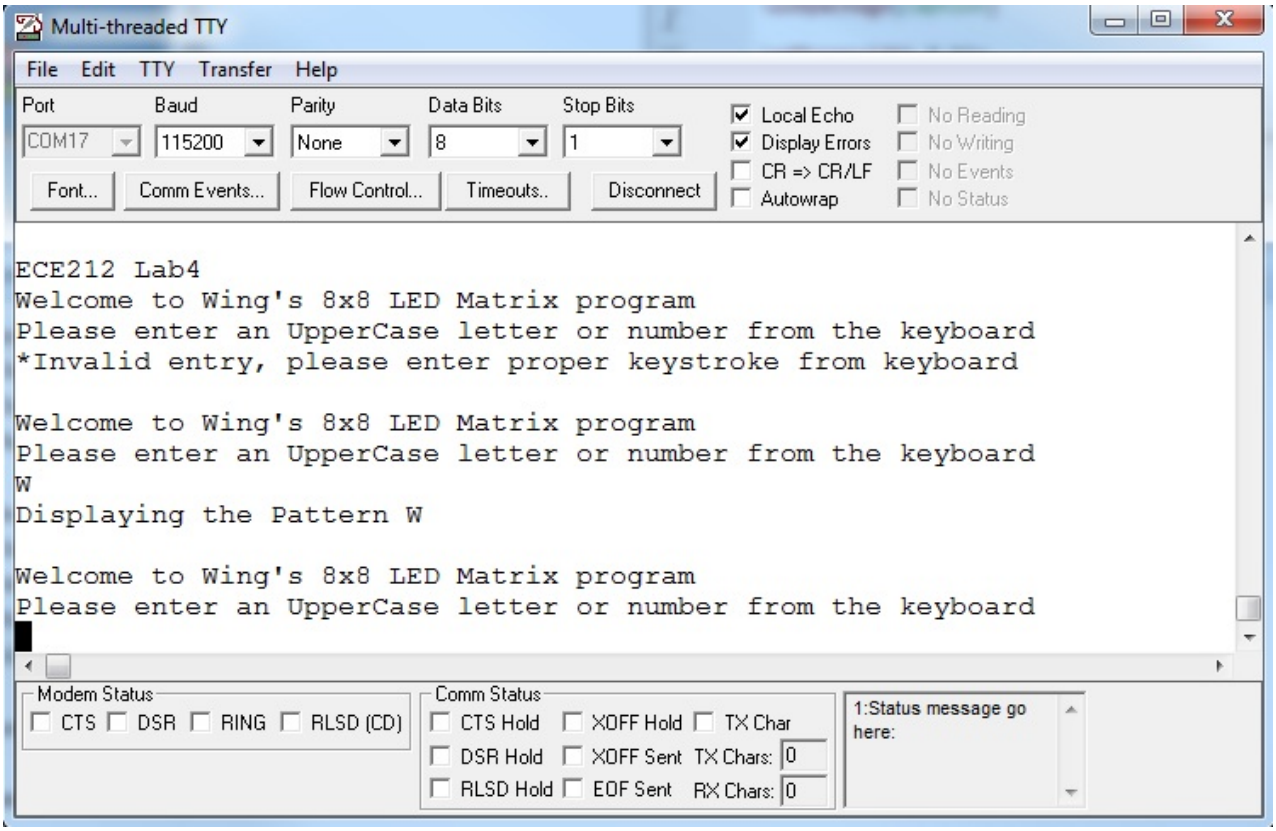
Lab 4: Introduction to Hardware Interfacing



Lab Dates

Refer to the schedule on the ECE212 Laboratories page for the latest schedule

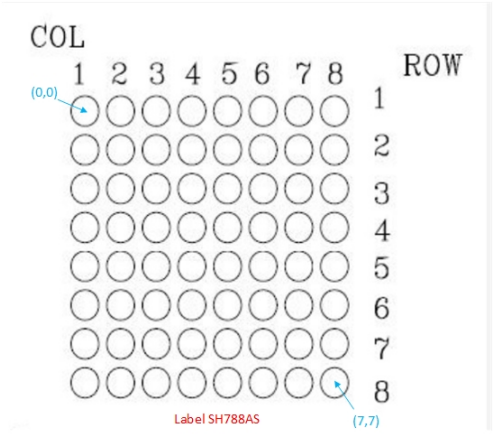
Introduction

In this lab, the students will be using the Nucleo-L432KC board to interface it with a custom 8X8 LED array PCB(printed circuit board). As background information, the LED PCB consist of one 8by8 LED matrix, two 74LS138 decoders, two 74HC04 inverters and eight 1KΩ resistors. The PCB will be powered from the +5V supply from the Nucleo board. The schematic/layout of the design is provided on Eclass for reference. A hardware verification program is also provided so that the students may verify their circuit connection before progressing to the lab requirements.

Please connect the two boards using the following pin assignments.

PA8 (Nucleo-L432KC) $\longrightarrow$ Pin1(C1)(8x8 LED PCB) = LSB that controls column selection
PA9 (Nucleo-L432KC) $\longrightarrow$ Pin2(C2)(8x8 LED PCB) = 2nd bit that controls column selection
PA10(Nucleo-L432KC) $\longrightarrow$ Pin3(C3)(8x8 LED PCB) = MSB that controls column selection
PB0 (Nucleo-L432KC) $\longrightarrow$ Pin4(R1)(8x8 LED PCB)) = LSB that controls row selection
PB1 (Nucleo-L432KC) $\longrightarrow$ Pin5(R2)(8x8 LED PCB) = 2nd bit that controls row selection
PB4 (Nucleo-L432KC) $\longrightarrow$ Pin6(R3)(8x8 LED PCB) = MSB that controls row selection
PA11(Nucleo-L432KC) $\longrightarrow$ Pin7(EN)(8x8 LED PCB) = Enable pin for both 74LS138.
GND(Nucleo-L432KC) $\longrightarrow$ Pin8(GND)(8x8 LED PCB)
5V(Nucleo-L432KC) $\longrightarrow$ Pin9(+5VDC)(8x8 LED PCB)

Note: From the LED array, think of every light as being on a 8 by 8 grid system. The orientation of the LED array in the schematic/layout gives the orientation of the rows and columns. The upper left-hand corner of the array is row 0, column 0.



Example: To turn on the the LED at coordinates (0,0), we need to set the logic level on pins P1-P3(column) and P4-P6(row) low and P7 high. To turn on the the LED at coordinates (7,7), we need to

set the logic level on pins P1-P3(column), P4-P6(row) and P7 high.

For convenience we have provided five subroutines row, col, onled, offled and HAL_Delay that will simplify this procedure of turning on an individual LED. The steps are as follows.

1. Move the LED row number into register R0. Jump to row subroutine
2. Move the LED col number into register R0. Jump to col subroutine
3. Jump to onled subroutine to turn on the LED
4. Move the Delay number into register R0. Jump to HAL_Delay subroutine
5. Jump to offled subroutine to turn off the LED

Objectives:

1. To gain experience in interfacing external circuitry(PCB) to the Nucleo-L432KC Board
2. To gain experience in using the Nucleo-L432KC I/O (GPIO ports) along with serial communcation from the keyboard.
3. To gain more experience in writing subroutines

Prelab and Preparation:

- Read the lab prior to coming to your lab section.
- Do the online prelab Quiz

View the datasheets of the various components:

- 74LS138
- 74LS04
- SN788AS
- Pin schematic/layout of LED PCB

Lab Work and Specifications

1. Inspect the PCB provide on the breadboard. Make sure it's mounted on correctly.
2. View the schematic/layout and hook up the PCB to the Nucleo board.
3. Download the hardware verification program for this lab. Load the program onto the Nucleo board and verify that the PCB board functions correctly. For further instructions on performing this task, please read the HOW-TO pdf on Eclass.
4. Download the template files
    - main_L432KC.c
    - Lab4_L432KC.s

- Lab4a_L432KC.s
- Lab4b_L432KC.s
- Lab4c_L432KC.s
- pattern.s
- getpatternaddress.s

Specifications

If you recall from the Lab1, the 'main.c' is the standard project template that is used to initialize and call standard functions/subroutines including our 'AssemblyProgram'. Do not modify any of the parameters in this file. 'Lab4_L432KC.s' has been provided to call your subroutines. Do not modify any of the parameters in this file. Lab4a_L432KC.s, Lab4b_L432KC.s and Lab4c_L432KC.s are provided for you to write your subroutines. Each subroutines has certain passed parameters and certain conditions. Carefull attention should be paid. A more detail description is provided below.

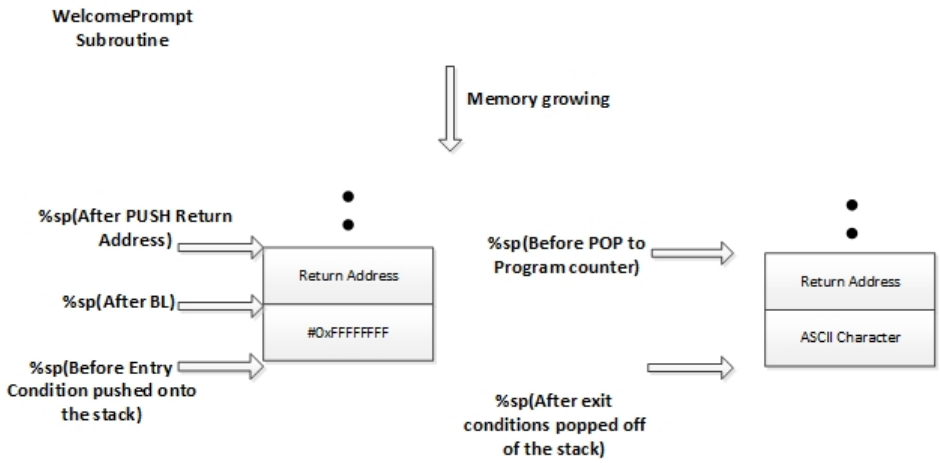
The requirements on each subroutine are indicated below.

1. WelcomePrompt

   In the Lab4a_L432KC.s template file, you are to code a subroutine called *WelcomePrompt* that prompts the user to enter a keystroke from the keyboard. The keystroke entered by the user must be an upper case letter or number. If the condition is not met, your program will reprompt the user to enter a proper keystroke. The valid keystroke(word) which is stored as its Ascii code will be passed out through the stack.

   Entry Condition = space allocated for the keystroke on stack(word-32bits)

   Exit Condtion = ASCII Code for keystroke on stack(word-32bits)



   Example - Your program should look something like this:

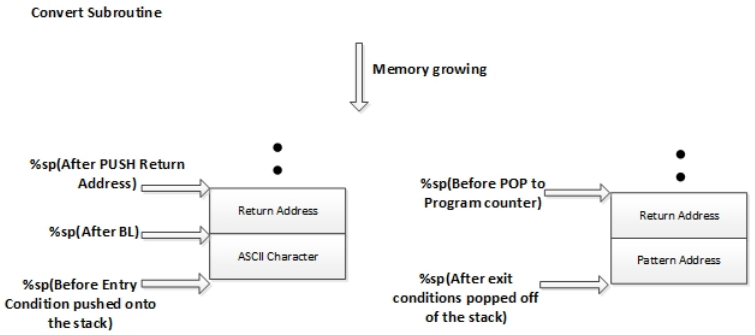
- Display a welcome string on startup.
  Welcome to Wing's LED Display
- Display a string prompting the user to enter the number of entries.
  Please enter an UpperCase letter or Number from the keyboard
- Display a string indicating invalid entry if an improper keystroke was entered.
  Invalid entry, please enter proper keystroke from keyboard

2. Convert

   In the Lab4b_L432KC.s template file, you are to code a subroutine called *convert* which takes the keystroke from the stack and replaces it with the pattern address. The actual pattern is stored as 2 words(8bytes). For example, the keystroke 'A' might have its pattern stored starting at memory location 0x20001000 to 0x20001007. A subroutine called convert1 has been provided to shorten the workload. Make use of it. The pattern address(word-32bits) will be passed out through the stack. This is a short subroutine and should take no more than 10 lines of code.

   Entry Condition = ASCII Code for keystroke on stack(word-32bits)

   Exit Condition = Pattern address on stack(word-32bits)

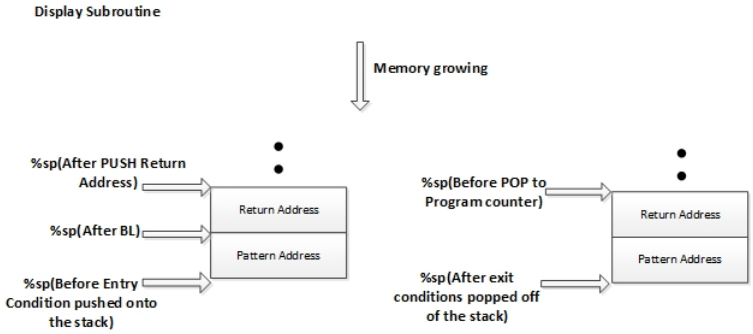


3. Display

   In the Lab4c_L432KC.s template file, you are to code a subroutine called *Display* that takes the pattern address from the stack and displays the pattern on the LED array by lighting up the proper LEDs. To turn on the LEDS on the 8X8 LED segement display, use the 5 subroutines provided *row*, *column*, *onled*, *offled* and *HAL_Delay*. Only one LED can be turned on at a time. However, by changing the timing in the delay subroutine, one can simulate having them all on by cycling through the LEDS at a fast speed(Refresh Rate).

   Entry Condition = Pattern address on stack(word-32bits)

   Exit Condition = Pattern address on stack(word-32bits)



To get started with this subroutine, you should initialize your pointer to the coordinates(0,0) on the LED array and turn off the led. From there you can start displaying your pattern one LED at a time. Cycle through your pattern slowly by delaying it and once it works, decrease the delay. Repeat displaying the pattern between 50-100 times to see the pattern more clearly.

Note: For each subroutine, only the information provided can be used. No assumptions can be made. In addition, each subroutine must preserve the register values and the stack must be cleaned up if necessary.

Provided Subroutines
An example of how to turn on a LED will be provide at the end of this section.

1. convert1

   This subroutine is written in the template file getpatternaddress.s. It takes the ASCII character from keyboard and returns the address of where the character pattern is stored.

   Entry Condition = ASCII Code on the stack(word-32bits)

   Exit Condition = Pattern Address for character on the stack(word-32bits)

   Example of pseduocode usage:
   Push ASCII character onto the stack
   Jump to convert1 subroutine
   Pop pattern address off of the stack

2. onled

   This subroutine outputs a logic one(3.3V) on pin PA11(Nucleo-L432KC). The logic one(3.3V) is enough to trigger the enable pins on the two 3by8 logic decoder chips. Essentially, turning both decoder chips on.

   Entry Condition = None

   Exit Condition = None

Example of pseduocode usage:
Jump to onled subroutine

3. offled

   This subroutine outputs a logic zero(0V) on pin PA11(Nucleo-L432KC). The logic zero(0V) disables the enable pins on the two 3by8 logic decoder chips. Essentially, turning both decoder chips off.

   Entry Condition = None

   Exit Condition = None

   Example of pseduocode usage:
   Jump to offled subroutine

4. row

   This subroutine selects which row the LED will be turned on. For example, if you want LED in row 3 to be turned on, the value 3 would be moved into the register R0 and the subroutine would be called.

   Entry Condition = Decimal value in register R0(word-32bits) Valid entries are 0 to 7. (0=row0,7=row7)

   Exit Condition = None

   Example of pseduocode usage:
   move row number into register R0
   Jump to Row subroutine

5. column

   This subroutine selects which column the LED will be turned on. For example, if you want LED in column 3 to be turned on, the value 3 would be moved into the register R0 and the subroutine would be called.

   Entry Condition = Decimal value in register R0(word-32bits)
   Valid entries are 0 to 7. (0=column0,7=column7)

   Exit Condition = None

   Example of pseduocode usage:
   move column number into register R0
   Jump to Column subroutine

6. HAL_Delay

   This subroutine delays the program a certain amount of time(ie 1=1ms). When a proper delay value is chosen between turning on/off the Led as you cycle through your leds, it will give the illusion that the pattern is on at the same time. An incorrect delay chosen will cause the pattern to flicker.

Entry Condition = Decimal value in register R0(word-32bits)

Exit Condition = None

Example of pseduocode usage:
Move Decimal value into register R0
Jump to HAL_Delay subroutine

7. printf

   Prints a string to the monitor. No carriage return or linefeed is invoked after calling this function.

   Entry Condition = address of the string in register R0

   Exit Condition = None

   Example of pseduocode usage:
   move StringLabel address into register R0
   Jump to printf subroutine

8. cr

   A function that generates a carriage return and linefeed

   Entry Condition = None

   Exit Condition = None

   Example of pseduocode usage:
   Jump to cr subroutine

9. getchar

   A function that prompts the user to enter a stroke from the keyboard. The data entered is stored as an ASCII code character in the data register R0. For example, if the keystroke 'A' was entered, R0 would contain the data 0x41. No carriage return or linefeed is invoked after calling this function.

   Entry Condition = None

   Exit Condition = Ascii keystroke stored in register R0

   Example of pseduocode usage:

   Jump to getchr subroutine

10. putchar

    Prints an ASCII character to the monitor. No carriage return or linefeed is invoked after calling this function.

    Entry Condition = ASCII code in register R0(word-32bits)

    Exit Condition = None

Example of pseduocode usage:
Move ASCII code into the register R0
Jump to putchr subroutine

Example of code snipet that turns on and off the LED at column 3, row 4 on the 8by8 LED array

mov r0,#3 /*move #3 into register R0 */
bl Column /*Jump to column subroutine */

mov r0,#4 /*move #4 into register R0 */
bl Row /*Jump to row subroutine */

bl onled /*jump to TurnOnLed subroutine */

mov r0,#10 /* move delay number into register R0*/
bl HAL_Delay /*jump to delay subroutine*/

bl offled /*turn off LED*/

Questions

1. Why were the two 74LS138 decoders used in the circuit instead of directly connecting the pins to the array? Discuss how removing the decoders would affect the function of the LED array in the context of this lab.

Marking Scheme

Lab 4 is worth 20 % of the final lab mark.

Demo and Report

There is no report for this lab. The demo is due before 5PM of the students scheduled lab session.

Appendix

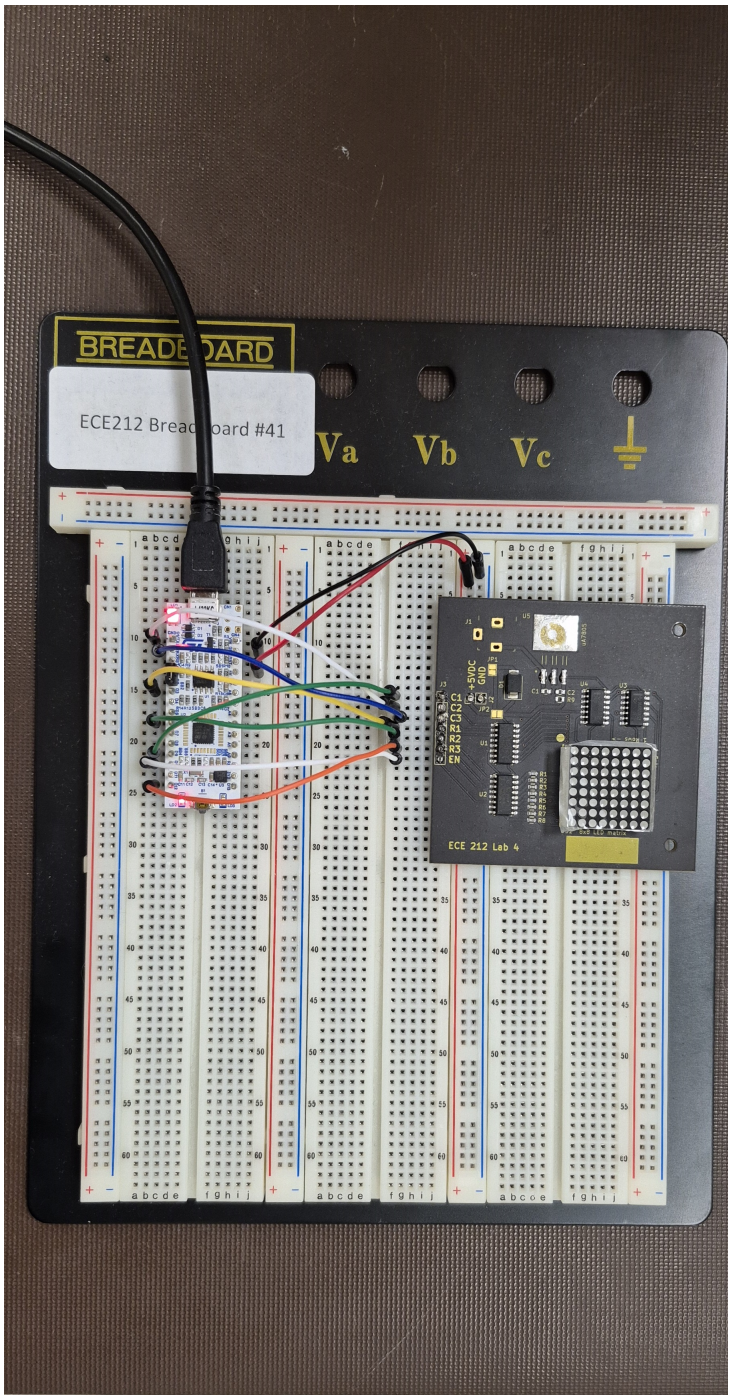


Figure 1: BreadBoard Layout