University of Pittsburgh

Department of Electrical Engineering

EE/CoE 1188: Cyber-Physical Systems

Dr. Samuel Dickerson

LAB #1: DIGITAL PADLOCK

Intro to Microcontrollers, Assembly Language and Embedded Systems

DUE ON OR BEFORE FRDAY, JANUARY 20, 2017

PURPOSE

The general purpose of this laboratory is to familiarize you with the software development. You will learn how to perform digital input/output on parallel ports of the MSP432. Software skills you will learn include port initialization, logic operations, and unconditional branching.

LAB REQUIREMENTS

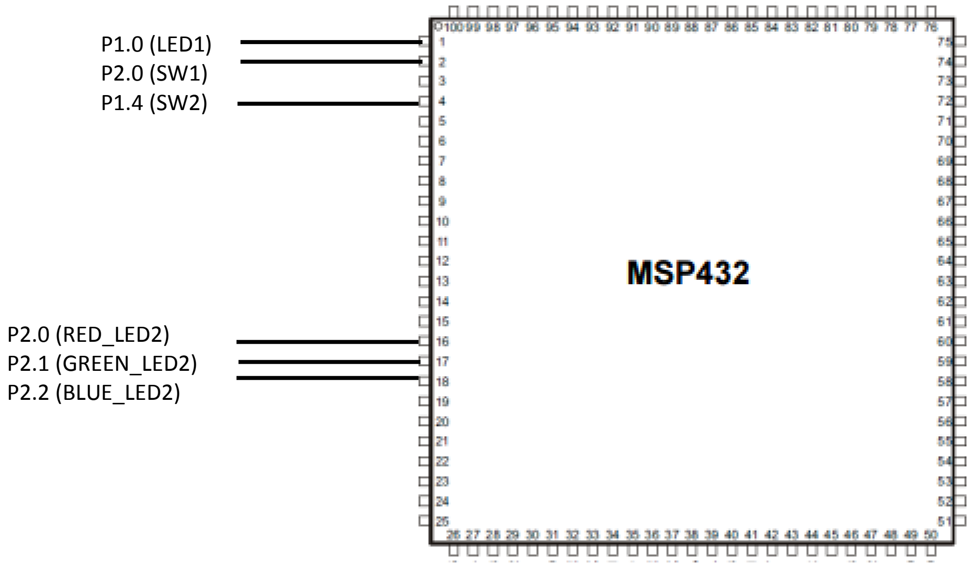
For this lab, you will create is a digital lock. The lock system has two switch inputs and one LED output that represents the state of the lock. The specification for your digital lock is as follows: If no buttons are pressed, the lock is in a standby mode and displays a blue light. If both buttons are pressed simultaneously, the device is unlocked and a green light is displayed. Finally, if an incorrect button combination is entered (i.e. only one of the two buttons are depressed) a red light is displayed.

PART A) PROJECT PREPARATION

In this project, you will become familiar with the MSP432 hardware, and also learn how to develop software for it. This will require that you have access to the board and

1) Included on the LaunchPad development board are switches and LEDs that can be accessed by the MSP432 chipset. Specifically, on the board are two user push buttons and four LEDs. One LED emits red (LT-C 190CKT) while the other is a Tri-Color RGB LED (EL-19-337) in a 3-input surface mount package. The third pushbutton on your board (at the middle left) is used as a RESET button for the board.

a) Sketch a schematic that represents the connections between the MSP432, the two user pushbuttons, and the LEDs. Include symbols and values for all relevant components (i.e. resistors, jumpers, voltage supplies, etc.) Also, make sure you label any ports and port pin numbers. Below is a symbol for the MSP432 you can use to get started. You will need to explore to the MSP432P401R LaunchPad Development Kit User's Guide in order to understand the layout of your board. MSP432.



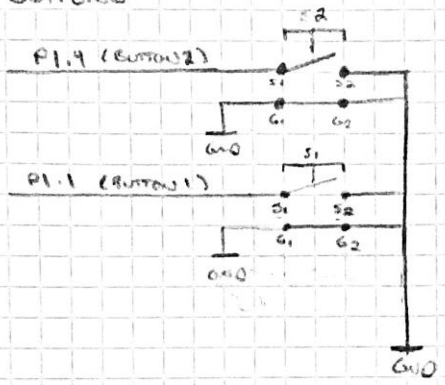
ANDREW ZUNDEL LAB 1 - DIGITAL PADLOCK DUE FRI, JAN 20, 2017

PART A

1.

a)

BUTTONS

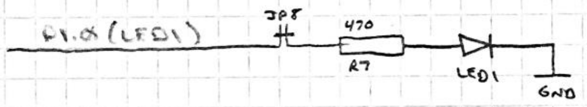


b) ACTIVE LOW BUTTONS

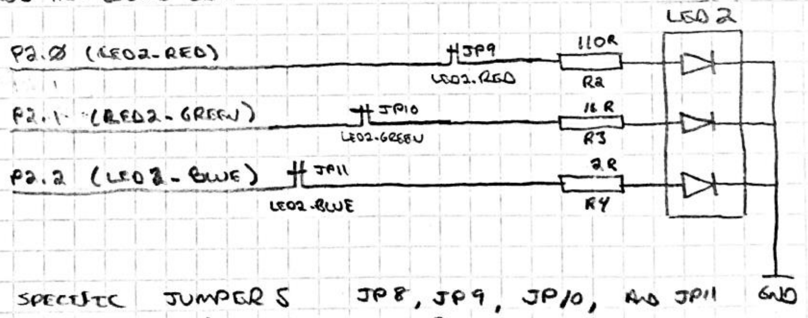
BUTTON PRESS COMPLETES CIRCUIT.

A SOURCE VOLTAGE (VDD) IS MISSING IN THIS DIAGRAM AND ALSO A RESISTOR

RED LED

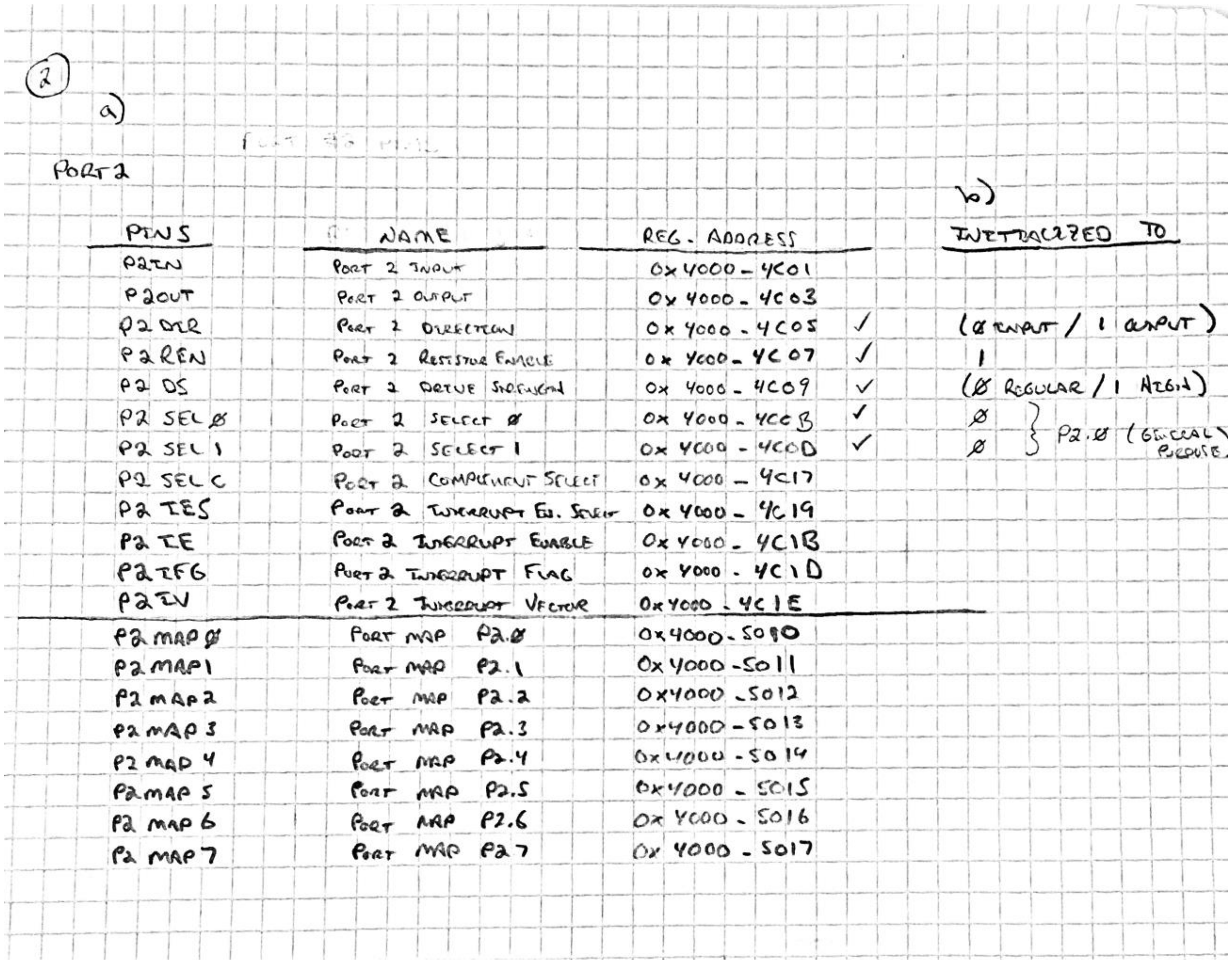


RGB TRI-COLOR LED



c) THE SPECIFIC JUMPERS JP8, JP9, JP10, AND JP11 CAN BE USED W/ AN EXTERNAL PROBE AS TO CIRCUMVENT HARDWIRED VALUES OF DEVELOPMENT BOARD

2) Find the memory map for the MSP432 in the MSP432P401R datasheet. On the MSP432, I/O ports are memory mapped just like program and data memory. This means that the software can access I/O simply by reading from or writing to the appropriate address, as if it were just another memory location.



2

a)

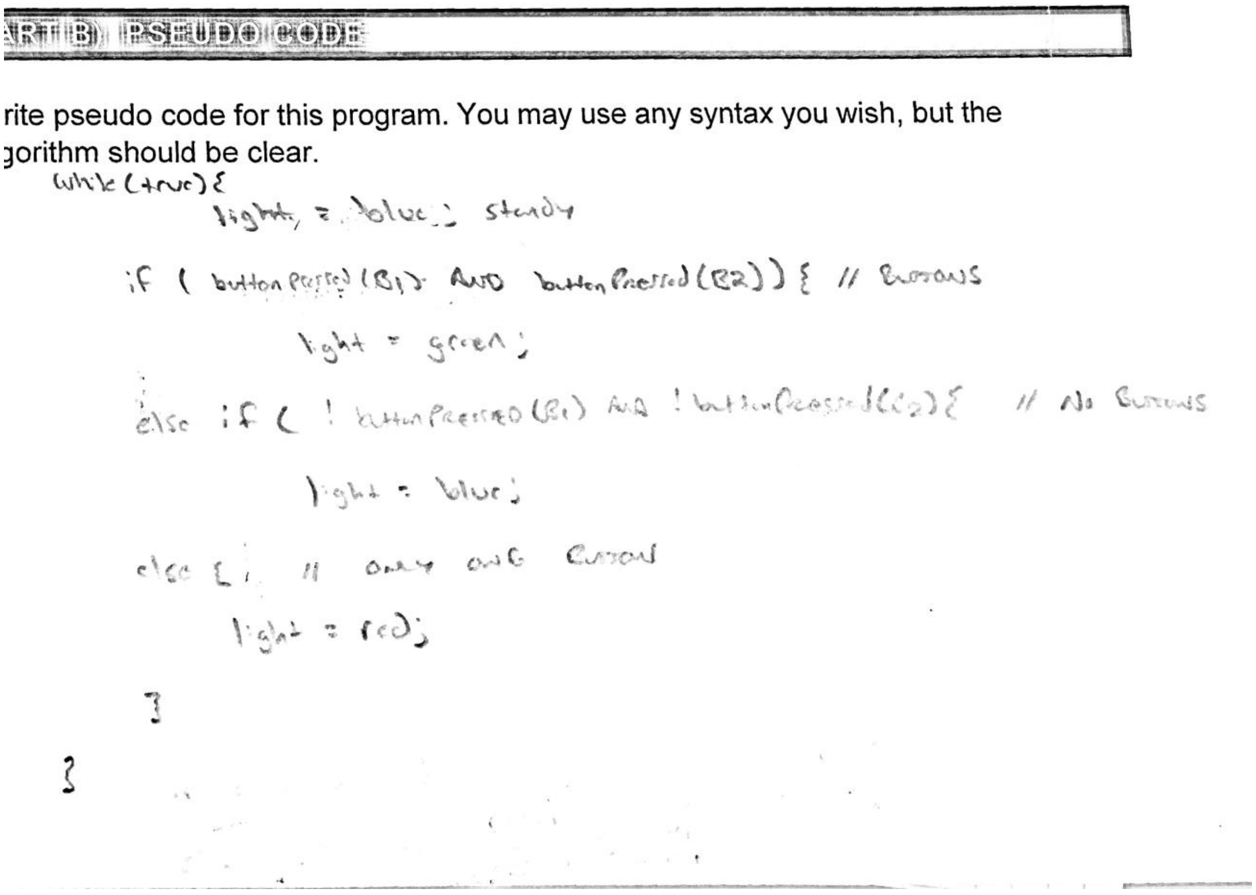
PORT 2

b)

PINS	NAME	REG. ADDRESS		INITIALIZED TO
P2IN	PORT 2 INPUT	0x 4000 - 4C01		
P2OUT	PORT 2 OUTPUT	0x 4000 - 4C03		
P2 DIR	PORT 2 DIRECTION	0x 4000 - 4C05	✓	(Ø INPUT / 1 OUTPUT)
P2REN	PORT 2 RESISTOR ENABLE	0x 4000 - 4C07	✓	1
P2 DS	PORT 2 DRIVE STRENGTH	0x 4000 - 4C09	✓	(Ø REGULAR / 1 HIGH)
P2 SEL Ø	PORT 2 SELECT Ø	0x 4000 - 4C0B	✓	Ø } P2.Ø (GENERAL PURPOSE
P2 SEL 1	PORT 2 SELECT 1	0x 4000 - 4C0D	✓	Ø
P2 SEL C	PORT 2 COMPLEMENT SELECT	0x 4000 - 4C17		
P2 IES	PORT 2 INTERRUPT ED. SELECT	0x 4000 - 4C19		
P2 IE	PORT 2 INTERRUPT ENABLE	0x 4000 - 4C1B		
P2IFG	PORT 2 INTERRUPT FLAG	0x 4000 - 4C1D		
P2IV	PORT 2 INTERRUPT VECTOR	0x 4000 - 4C1E		
P2 MAP Ø	PORT MAP P2.Ø	0x4000 - 5010		
P2 MAP1	PORT MAP P2.1	0x 4000 - 5011		
P2 MAP 2	PORT MAP P2.2	0x4000 - 5012		
P2 MAP 3	PORT MAP P2.3	0x4000 - 5013		
P2 MAP 4	PORT MAP P2.4	0x 4000 - 5014		
P2 MAP 5	PORT MAP P2.5	0x4000 - 5015		
P2 MAP 6	PORT MAP P2.6	0x 4000 - 5016		
P2 MAP 7	PORT MAP P2.7	0x 4000 - 5017		

PART B) PSEUDO CODE

Write pseudo code for this program. You may use any syntax you wish, but the algorithm should be clear.



ART B) PSEUDO CODE

rite pseudo code for this program. You may use any syntax you wish, but the
gorithm should be clear.

```
while (true){
        light = blue; steady

    if ( buttonPressed (B1) AND buttonPressed (B2)) { // Buttons

            light = green;

    else if ( ! buttonPressed (B1) AND ! buttonPressed (B2) {    // No Buttons

            light = blue;

    else { // Only one Button

          light = red;

    }

}
```

SOURCE CODE

```
.thumb
.text
       .align 2

       .global main
       .thumbfunc main

main: .asmfunc

       ; Base Address of Port Mappings
    MOV R0, #0x4c00
    MOVT R0, #0x4000

       ;Settings for Port 1                                [000S00SL]
       MOV R1, #0xFE                 ;                            2  1
       MOV R2, #0x00
       STRB R2, [R0, #0x04]          ; P1DIR       => [00000000] = 0 (Input Mode
for SWITCHes)
       STRB R1, [R0, #0x06]          ; P1REN  =>  [00010010] = 1 (Registers for
each SWITCH and LED)
       STRB R2, [R0, #0x08]          ; P1SEL0 => [00000000] = 0 --> General
Purpose Register Option
       STRB R2, [R0, #0x0A]          ; P1SEL1 => [00000000] = 0 -/

       ;Settings for Port 2                                [00000BGR]
       MOV R1, #0x07
       MOV R2, #0x00
       STRB R1, [R0, #0x05]          ; P2DIR       => [00000111] = 1 (Output Mode
For LEDs)
       STRB R1, [R0, #0x07]          ; P2REN  =>  [00000111] = 1 (Registers for
each position of LED)
       STRB R2, [R0, #0x0B]          ; P2SEL0 => [00000000] = 0 --> General
Purpose Register Option
       STRB R2, [R0, #0x0D]          ; P2SEL1 => [00000000] = 0 -/

       ; Initialize P2OUT to Display BLUE LED
       MOV R1, #0x04                 ; [00000100] B__
       STRB R1, [R0, #0x03]          ; P2OUT => 0x40004C03

running ; Start main loop of padlock program

       ; Check both input switches
       LDRB R7, [R0, #0x00]      ; P1IN => 0x40004C00
       AND R7, R7, #0x12


       ; If R7 contains 0x12 no SWITCH is pressed
       MOV R6, #0x00
       SUB R6, R7, #0x12
       CBZ R6, blue
       ; If R7 contains 0x10 SWITCH 2 is being pressed
       MOV R6, #0x00
       SUB R6, R7, #0x10
```

```
        CBZ R6, red
        ; If R7 contains 0x02 SWITCH 1 is being pressed
        MOV R6, #0x00
        SUB R6, R7, #0x02
        CBZ R6, red
        ; If R7 contains 0x00 Both SWITCHes are pressed
        CBZ R7, green


blue
        MOV R5, #0x04                       ; [00000100] B__
        STRB R5, [R0, #0x03]                ; P2OUT => 0x40004C03
        B running

green
        MOV R5, #0x02                       ; [00000010] _G_
        STRB R5, [R0, #0x03]                ; P2OUT => 0x40004C03
        B running

red
        MOV R5, #0x01                       ; [00000001] __R
        STRB R5, [R0, #0x03]                ; P2OUT => 0x40004C03
        B running

.end
```