

## ESET 349 Microcontroller Architecture

### Lab 4: Selecting LEDs based on input (switch and photosensor response)

#### Objectives

1. Understand pull-up and pull-down circuits, and the function of MSP432's internal input resistors.
2. Build a simple circuit on a breadboard for switches and interface inputs with an MSP432.
3. Write a program in assembly language to read input from the external circuit.
4. Implement conditional statements in a program.

#### Your Tasks

1. Program an MSP432 microcontroller to light up two LEDs based on the state of a switch. Read an input coming from breadboard switch. If the input bit is low, light up LED1 on the breadboard. Else (input bit must be high), light up LED2 on the breadboard. You may use P5.4 as input pin and P5.1 and P5.0 as output pins for the two LEDs. Decide on whether you will use a pull-up or pull-down resistor for the input pin. The pull-up and pull-down resistor settings are shown in Fig. 1.
2. Replace the push switch input with a photoresistor input to light up the LEDs based on the detection of light.

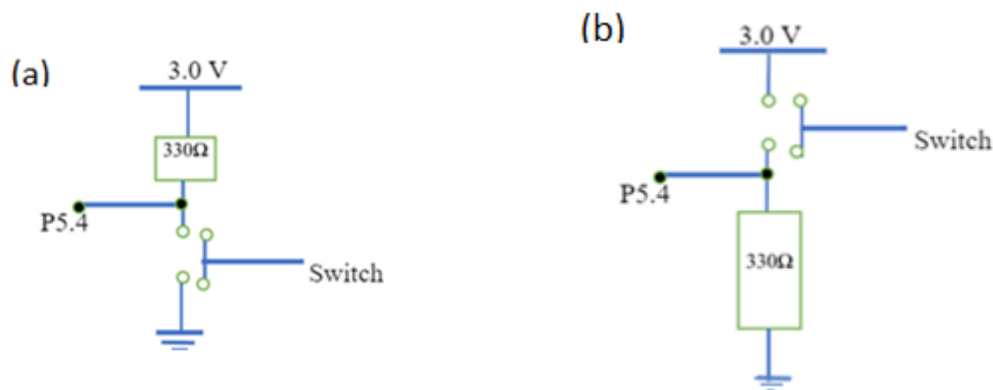


Figure 1. (a) Pull-up and (b) pull-down resistor settings

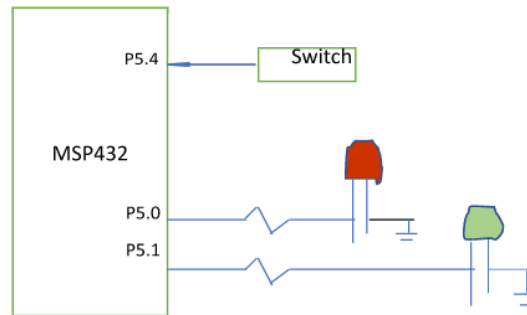


Figure 2. Circuit sketch

## Flowchart

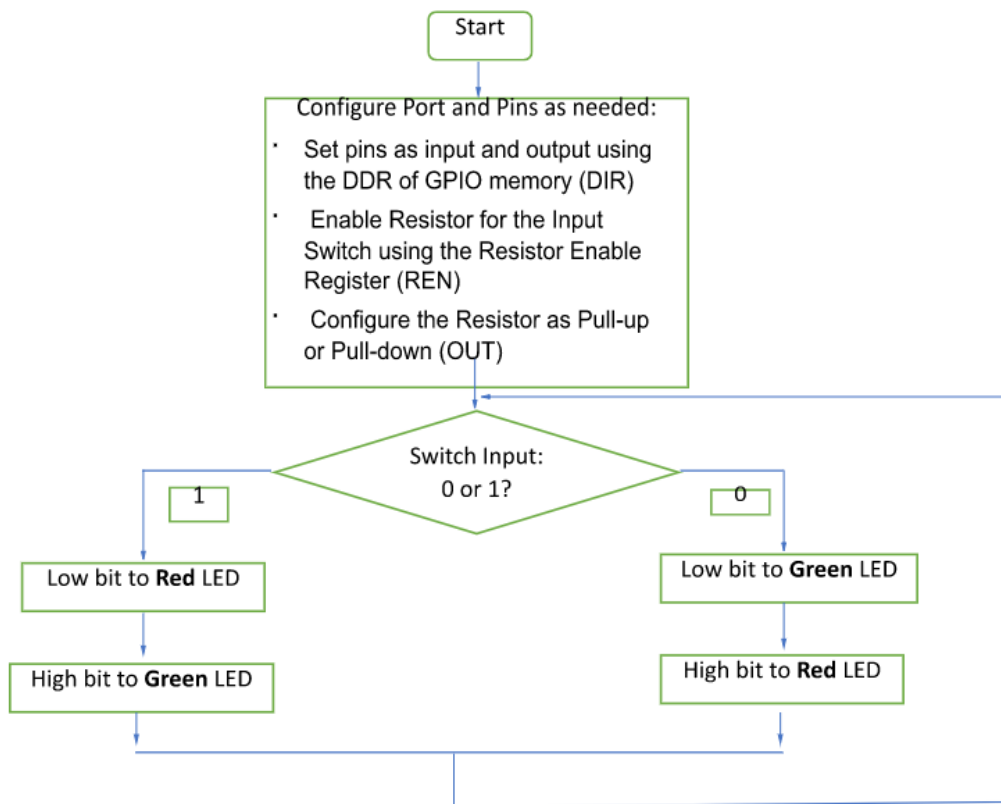


Figure 3. A flowchart to implement output LED selection based on switch input

## Task 2: Photoresistor Circuit Design

(The symbols with the circles are the photoresistors.)

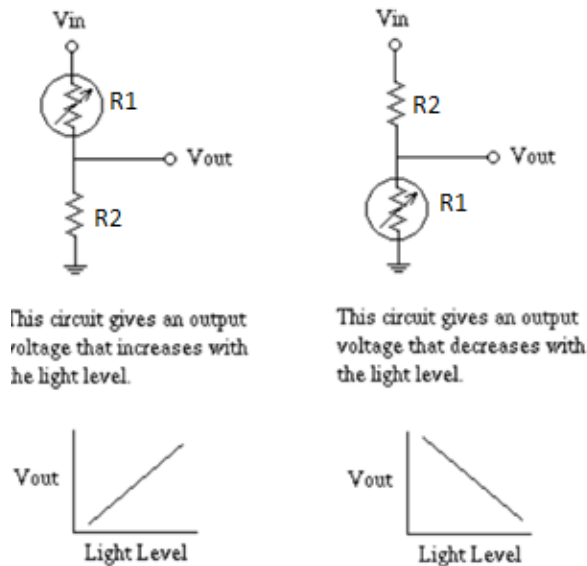


Figure 4. Functioning of a photoresistor as a light-sensitive switch

A photocell or photoresistor is a light sensitive device. Its resistance changes with respect to changes in the ambient light intensity. A photocell can be used as a smart device in numerous situations, such as turning ON streetlights at night without human interaction.

**\*NOTE: due to inaccuracies in the light detection of the photoresistors, you may need to cover the device or use a phone flashlight to observe changes in light sensitivity.**

A typical value for R2 in the potential divider circuit shown in Fig. 4. could be 10k  $\Omega$ . If you do not find the suggested value optimal, measure the resistance of the photocell (R1) when it is covered and exposed to ambient light with a multimeter. Once you have the values, re-calculate the required resistance of R2 as shown in Table 1 below.

Table 1: Voltage Divider Calculations

	Photocell Exposed to Ambient light	Photocell Covered
Ordinary Resistor (R2)	10k $\Omega$	10k $\Omega$
Photocell Resistor (R1)	5 k $\Omega$	50k $\Omega$ (depending on cover)
Input to P5.4, V	$V = \frac{10}{10+5} * 3V = 2.2V$	$V = \frac{10}{10+50} * 3V = 0.5V (0)$

## Program Sketch

Incomplete program as a guide only; please complete the code. Notice this skeleton program only shows how to read input from port 5.

Select the input mode to use (pull-up or pull-down) and complete the configuration code. Add code to check the input state and then control the output LEDs, as shown in Figure 3.

Once you have the push button switch program working, demonstrate to your TA. Then, replace the switch circuit with a photoresistor circuit as suggested in Figure 4.

If you suspect your hardware is not functional, try using a different set of three pins, or reach out to your TA for assistance.

```
1 ; P5.4 connected to input
2 ; P5.1 and P5.0 connected to two LEDs
3
4
5     area Lab4, code, readonly
6     export __main
7
8 __main proc
9
10        ; Configure GPIO
11        LDR R0, =0x40004C00      ; Port 1 base address
12        ADD R0, #0x40            ; Port 5 base address
13
14        ; ADD CODE TO CONFIGURE P5DIR, P5REN AND P5OUT
15
16        ; Loop, read input, toggle LEDs
17 repeat LDRB R2, [R0, #0x00]      ; Load input register value to R2
18        AND R2, #0x10            ; Mask pin 4
19
20        ; ADD CODE TO CHECK INPUT, SEND HIGH OR LOW SIGNALS TO 5.0 AND 5.1
21
22        B repeat                  ; Loop indefinitely
23
24        endp
25        end
```