Department of Engineering Technology and Industrial Distribution
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 STM32's internal input resistors.
2. Build a simple circuit on a breadboard for switches and interface inputs with an STM32.
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 STM32 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 PC0 as the input pin and PC1 and PC2 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.

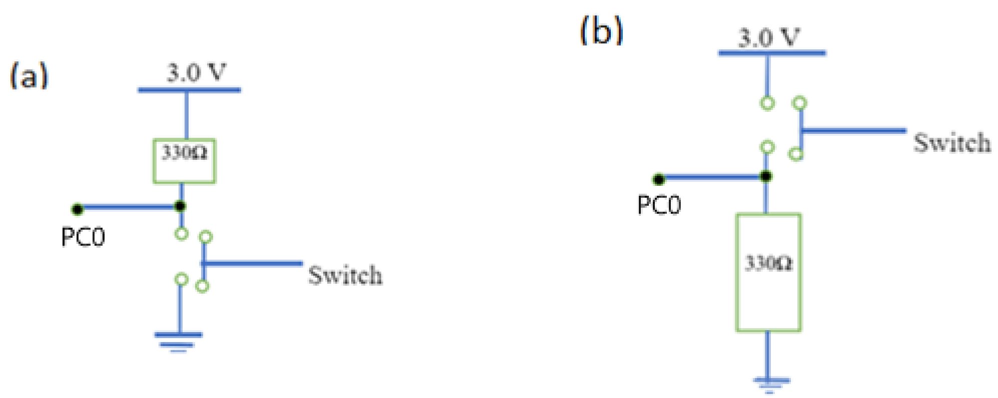

Figure 1. (a) Pull-up and (b) pull-down resistor settings

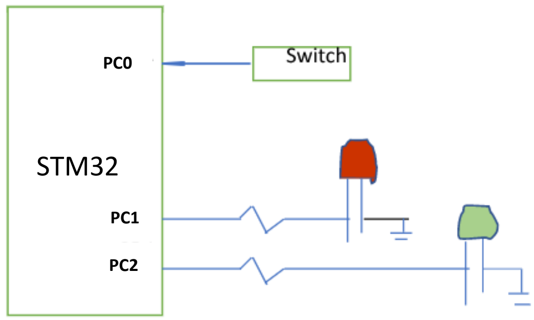

Figure 2. Circuit sketch

Flowchart

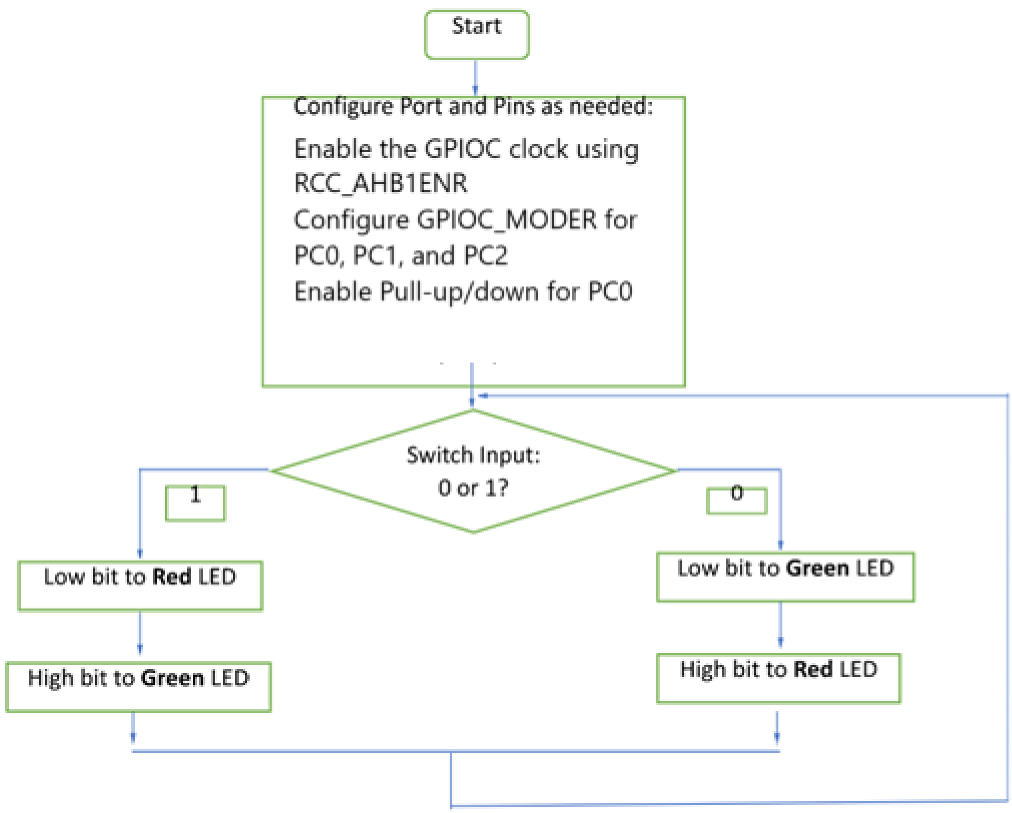

Figure 3. A flowchart to implement output LED selection based on switch input

Task 2: Photoresistor Circuit Design

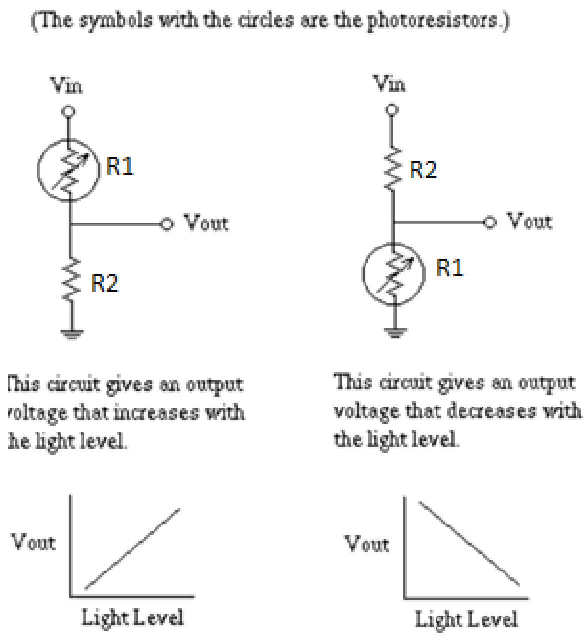

Figure 4. Functioning of a photoresistor as a light-sensitive switch

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 **Ω**. 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)	10kΩ	10kΩ
Photocell Resistor (R1)	5 kΩ	50kΩ (depending on cover)
Input to P5.4, V	$V = \frac{10}{10+5} * 3V = 2.2$V	$V = \frac{10}{10+50} * 3V = 0.5V$ (0)

Program Sketch

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

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.

```
            AREA Lab4, CODE, READONLY
            EXPORT __main

__main      PROC

            ; Enable GPIOC clock
            LDR R0, =0x40023830       ; RCC_AHB1ENR address
            MOV R1, #0x00000004       ; Enable GPIOC (bit 2)
            STR R1, [R0]              ; Write to RCC_AHB1ENR

            ; Configure GPIO
            LDR R0, =0x40020800       ; GPIOC base address
            ; ADD CODE TO CONFIGURE PC0 AS INPUT AND PC1, PC2 AS OUTPUTS

            ; Main loop
repeat      LDR R2, [R0, #0x10]       ; Load GPIOC_IDR (input data register)
            ; ADD CODE TO CHECK INPUT STATE OF PC0
            ; ADD CODE TO SEND HIGH OR LOW SIGNALS TO PC1 AND PC2

            B repeat                  ; Loop indefinitely

            ENDP
            END
```