IS4151/IS5451 – AIoT Solutions and Development

AY 2024/25 Semester 2

Practical Lab 03 – Single-board Computer (I)

Part 1 – Basic Programming

PE03-1-1 – Fancy LEDs Light

Recall that in Lecture 06, we have set up a circuit to control three LEDs with three push buttons (slide 54-56). Sample code `src05.py` turns on a particular LED only when the respective push button is pressed and remains depressed. The LED turns off immediately once the push button is released.

Perform the following tasks:

a. Set up the circuit as shown in the lecture note and run sample code `src05.py` to observe the output.

b. Rewrite sample code `src05.py` such that pressing a particular push button will toggle the respective LED on or off. That is, if the LED is currently turned off, then pressing the push button once will turn it on and the LED will remain turned on until the same push button is pressed again.

c. Redesign and set up a circuit that takes in a fourth push button. Extend the revised code from (b) to program the fourth push button to toggle the three LEDs to light up in the following pattern mode:

   i. User presses fourth push button to toggle into pattern mode.
   ii. Turn off all three LEDs
   iii. Pause 1 sec > Turn on red LED only > Pause 1 sec > Turn on green LED only > Pause 1 sec > Turn on blue LED only
   iv. Pause 1 sec > Turn off all three LEDs
   v. Pause 0.3 sec > Turn on all three LEDs > Pause 0.3 sec > Turn off all three LEDs.
   vi. Repeat the blinking sequence in (v) 3 times.
   vii. Repeat from (ii) until user presses the fourth push button again to toggle out of the pattern mode.

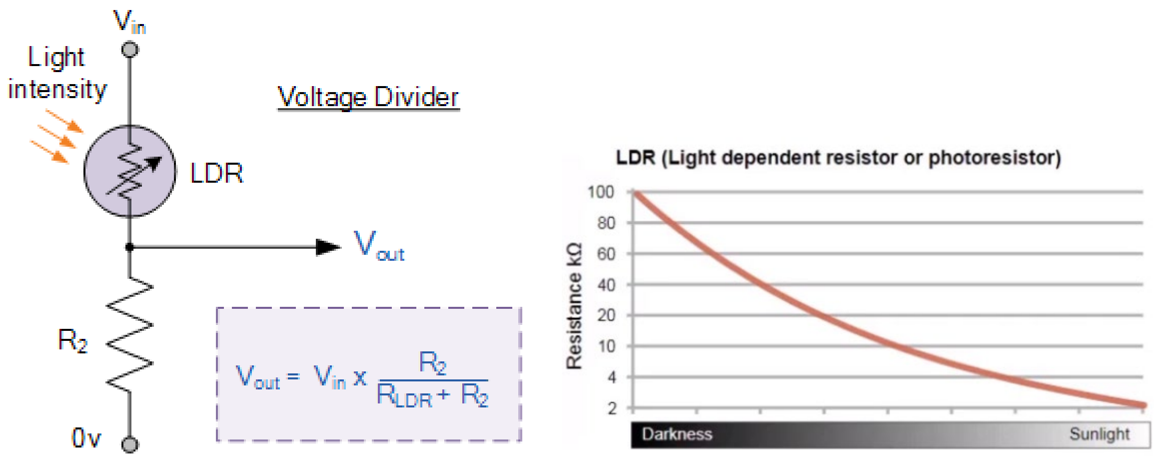
PE03-1-2 – Automatic Lighting System

The Raspberry Pi kit that is issued to you comes with a photocell, which is a sensor that can detect light. A photocell is also known as CdS cell (it is made of Cadmium-Sulfide), light-dependent resistors (LDR), or photoresistor.

We discussed the basic working principle of an LDR light sensor in Lecture 05. In that lecture, we used a Grove light sensor module that is already soldered onto a breakout board with the Grove connector and all the other required electronic components. Moreover, a micro:bit has an onboard Analogue to Digital (ADC) converter, which makes it much easier to work with an analogue sensor such as the light sensor. In contrast, the Raspberry Pi does not have an onboard ADC and we need to attach an external ADC such as the MCP3008, which is a 10-bit ADC (0 to 1,023).

The easiest way to use the photocell with the Raspberry Pi is to connect one end of the photocell to 3V3 Power (note that we use 3V3 for the MCP3008 too), and the other end to a pull-down resistor then to ground. The point between the fixed pull-down resistor and the variable photocell resistor is connected to one of the 8 available connectors on the MCP3008.

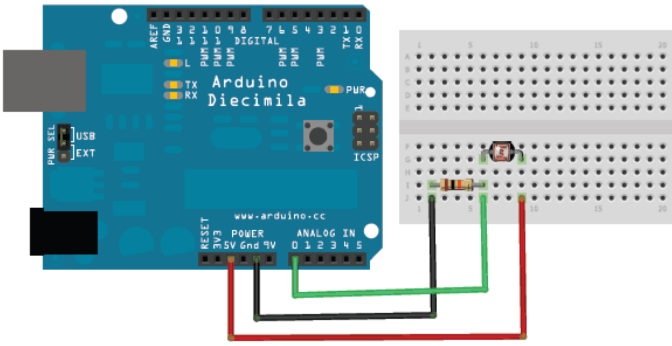
This setup is known as a voltage divider network. Connecting a LDR in series with a standard resistor across a single DC supply voltage has one major advantage, i.e., a different voltage will appear at their junction for different levels of light. The amount of voltage drops across the series resistor R_2 is determined by the resistive value of the LDR, i.e., R_{LDR}. This ability to generate different voltages makes a voltage divider network a very useful circuit.



(**Source**: https://www.electronics-tutorials.ws/io/io_4.html)

The current through a series circuit is common and as the LDR changes its resistive value due to the light intensity, the voltage present at V_{OUT} that is connected to the ADC will be determined by the voltage divider formula. The resistance, i.e., R_{LDR}, of the photocell in our Raspberry Pi kit can vary from about 5-10KΩ under light to 200KΩ in darkness.

The figure on the next page shows a photocell in a voltage divider network with V_{OUT} connected to the analogue input 0 of an Arduino board.



(**Source**: https://learn.adafruit.com/photocells/using-a-photocell)

Perform the following tasks:

a. Set up the circuit as shown in slide 67 of Lecture 06 to connect the MCP3008 ADC to the Raspberry Pi.

b. Set up a voltage divider network on the same breadboard using the photocell given to you and connect V_{OUT} to connector 0 of the MCP3008 ADC.

c. Set up one red LED and one green LED on the same breadboard. The positive end of each LED should be connected to an available GPIO pin on the Raspberry Pi.

d. Write a Python program to implement an automatic lighting system with the circuit setup in (a-c):

   i. The brightness of the green LED should show the intensity of the current ambient lighting condition, i.e., the green LED should be brightened when the ambient lighting is bright and dimmed when the ambient is darkened.

   ii. When the ambient lighting has darkened to less than 20% brightness, turn on the red LED. Otherwise, turn off the red LED.

<u>PE03-1-3 – Temperature Warning Gauge</u>

Using the Adafruit's version of the Bosch BME280 environmental sensor, set up a circuit and write a Python program to implement a temperature warning gauge. The gauge consists of 1 green LED and 3 red LEDs. The LEDs should be turned on according to the ambient temperature that is detected by the sensor:

- Temperature <= 30 – Turn on green LED
- Temperature > 30 – Turn on first red LED
- Temperature > 40 – Turn on second red LED
- Temperature > 50 – Turn on third red LED

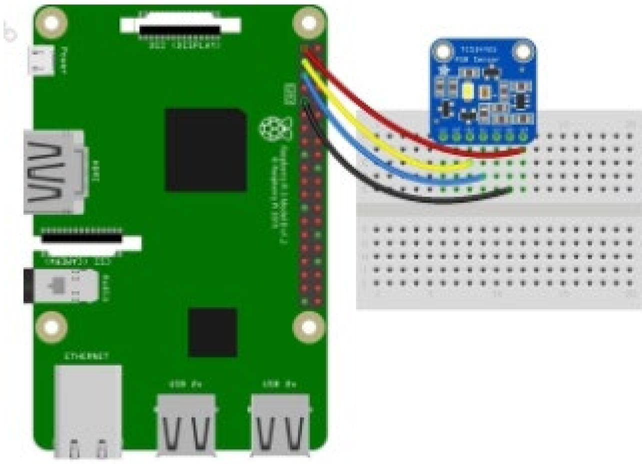
Do **NOT** use water or naked flame to trigger the temperature changes. The safest way is to use a hair dryer.

Did you notice any problem with the use of the BME280 sensor for detecting ambient temperature?

PE03-1-4 – RGB Colour Sensor

The Raspberry Pi kit that is issued to you comes with the Adafruit TCS34725 RGB Color Sensor. Find out more about the sensor on the website of Adafruit. Thereafter set up a circuit and write a Python program to implement a simple RGB colour recognition system with the help of the Adafruit CircuitPython library for the TCS34725 colour sensor.

Hint on setting up the circuit board:



- Pi 3V3 to sensor VIN
- Pi GND to sensor GND
- Pi SCL to sensor SCL
- Pi SDA to sensor SDA

To install the required Python library, use the following command:

```
sudo python -m pip install adafruit-circuitpython-tcs34725
```

Compare the RGB colour code that is detected by the sensor using this online tool – https://www.color-hex.com. Is the sensor accurate?

Part 2 – Intermediate Programming

PE03-2-1 – Weather Station

Set up a circuit and write a Python program to implement a simple home weather station. The weather station consists of a temperature indicator gauge and a humidity indicator gauge for monitoring the ambient temperature and humidity level using a Raspberry Pi and the Bosch BME280 environmental sensor. Each gauge manifests as a separate set of three LEDs, one green, one blue and one red.

a. On start-up, record the initial ambient temperature and humidity level detected by the BME280 sensor. This is the baseline temperature and humidity level.

b. Check the ambient temperature every one minute and:

    i. Turn on a green LED if the current temperature is equal to the baseline temperature.
    ii. Turn on a blue LED if the current temperature is lower than the baseline temperature.

iii. Turn on a red LED if the current temperature is higher than the baseline temperature.

c. Check the ambient humidity level every one minute and:

i. Turn on a green LED if the current humidity level is equal to the baseline humidity level.
ii. Turn on a red LED if the current humidity level is lower than the baseline humidity level.
iii. Turn on a blue LED if the current humidity level is higher than the baseline humidity level.

d. If user presses a reset push button, the baseline temperature and humidity level will be set to the respective current value, and the green LEDs should be turned on.

When checking whether a current value is equal to the baseline value, you should define a suitable threshold for the variance instead of making a strict equal comparison. For instance, if the threshold for ambient temperature is ±0.5˚C, then 24.3˚C is considered to be equal to 24.0˚C, i.e., any temperature value in the range [23.5˚C, 24.5˚C] is considered to be equal.

File Transfer to Raspberry Pi

If VNC server does not supports file transfer. you should use SFTP from your laptop's command prompt or terminal:


```
sftp pi@raspberrypi.local
<replace host name with ip address if need to, and then enter password, which should be raspberry>

cd /home/pi/Documents

lcd "<local path on laptop to be quoted due to spaces>"

put <file name>

quit
```

The file will be uploaded to your Raspberry Pi home directory, i.e., /home/pi/Documents

Multiple “put” commands can be issued within a single SFTP login session.