# Lab 2: Configuring a Photoresistor with RPi

A code example has been prepared for this lab and can be reached here:

<https://github.com/gabrielastudillo/Internet-of-things-2/blob/main/photoresistor.py>

## Introduction

## The photoresistor module is a resistor module designed based on the principle of photoconductive effect of semiconductors, of which the resistance varies with the intensity of incident light. The resistance of the photoresistor we use decreases with stronger incident light and increases with weaker light.

## List of Components

* 1 \* Raspberry Pi
* 1 \* GPIO Extension Board
* 1 \* 40-Pin GPIO Cable
* 1 \* Breadboard
* 1 \* ADC0832 Module (2 channel, 8 bits)
* 1 \* Potentiometer Module
* 1 \* Photoresistor Module
* 2 \* 3-Pin Wires
* 1 \* 5-Pin Wires

## Experimental Principle

A picture containing schematic

Description automatically generatedIn this experiment, by programming the Raspberry Pi, we collect the analog values output by the photoresistor module through pin CH0 of the ADC0832, convert it to digital values and display them on the terminal or light a led depending on the light intensity.

## Experimental Procedures

### Step 1: Build the circuit

Note that on the manufacturer manual, 5V is selected as source of power, ***choose 3.3V instead as it is the recommended voltage for the ADC0832.***



### Step 2: Read and edit

1. Read the code with your favorite text editor, look for errors and edit accordingly.

#!/usr/bin/env python

import ADC0832

import time

def init():

    ADC0832.setup()

def loop():

    while True:

        res = ADC0832.getResult()

        vol = 3.3/255 \* res

        print ('analog value: %03d  ||  voltage: %.2fV' %(res, vol))

        time.sleep(0.2)

if \_\_name\_\_ == '\_\_main\_\_':

    init()

    try:

        loop()

    except KeyboardInterrupt:

        ADC0832.destroy()

        print ('The end !')

### Step 3: Run

$ sudo python photoresistor.py

1. Now, when you cover to the photoresistor, you will find that the value displayed on the screen decreases. On the contrary, when you shine the photoresistor with a strong light, the value displayed will increase.

### Step 4: Output on the terminal

We have learned that any value below 10 Lux is considered as a dark room. So, modify the code so you can print on the terminal “dark” when the luminosity is below 10 lux and “light” when the luminosity is over 10 lux. Test and tune your code as needed.

### Step 5: Adding a LED

Now add a LED to one of the GPIO ports and modify your code so every time that someone turns on the lights on your room an alarm is triggered (the LED is ON) and when your room is dark again the alarm is closed (LED OFF), you can also add a buzzer, so the alarm also generates an annoying sound (optional).