**Project 1: Smart Lighting with auto-dimming and GUI slider control**

1. Design an auto-dimming light that measures the illuminance of room, and adjust the light intensity
2. The measurement of illuminance must include photo-resistor and Analog-to-Digital Converter (ADC) circuit
3. The light should be using 220V power circuit with PWM-based power control
4. The current status of illuminance should be display on Python GUI display
5. Include a manual mode, where the light intensity is controller by slider (scale)

**Project 2: Simple Oscilloscope based on Raspberry Pi 3 and Python**

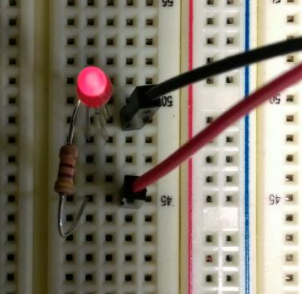
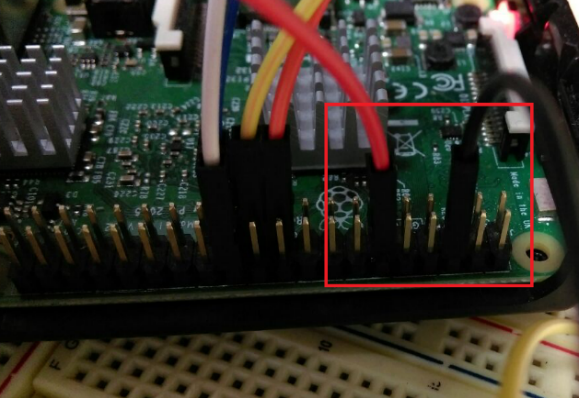
1. Design an oscilloscope that uses Analog-to-Digital converter to read the digital data of the given analog signal, and displays the analog signal on Python GUI frame.
2. Driver routines for ADC should be prepared in C programming language, and extended to be as a Python module

**First steps toward the interfacing hardware**

**Digital outputs. Turn on and off LED**

Materials: Breadboard, jumpers, LED, resistor 470 om

Connection: LED negative leg was connected to GND and the resistor was connected to GPIO 18.

To access the GPIO pins through *terminal* using RPi.GPIO:

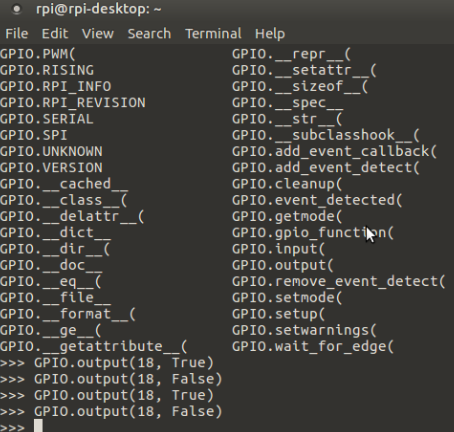
$sudo python3

$import RPi.GPIO as GPIO

$GPIO.setmode(GPIO.BCM)

$GPIO.setup(18, GPIO.OUT)

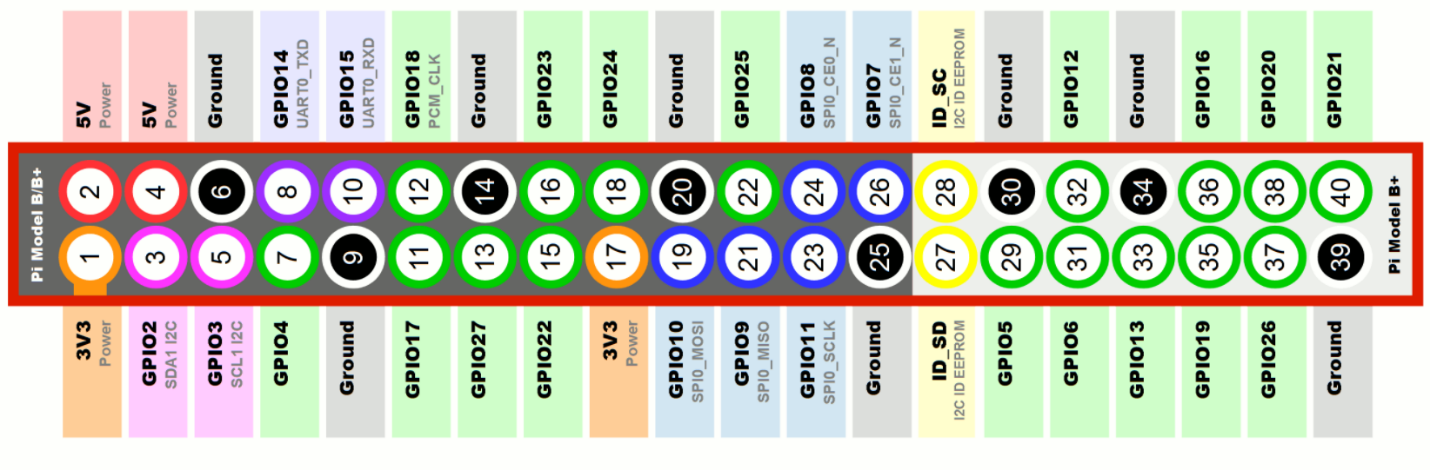
$GPIO.output(18, True) #False



**Note:** There are two GPIO standard pins for reading and controlling them. So, it is possible to set mode by pin names or pin position:

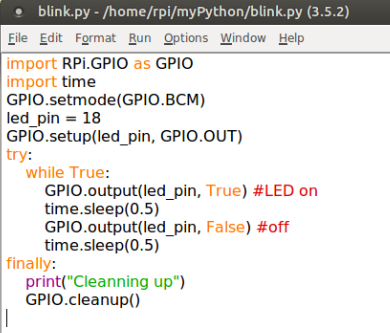
$ GPIO.setmode(GPIO.BCM) #Broadcom SOC channel

$ GPIO.setmode(GPIO.BOARD)

[4] The **GPIO.BOARD** option specifies that you are referring to the pins by the number of the pin the the plug - i.e the numbers printed on the board (e.g. P1) and in the middle of the diagrams below.

The **GPIO.BCM** option means that you are referring to the pins by the "Broadcom SOC channel" number, these are the numbers after "GPIO" in the green rectangles around the outside of the below diagrams:

To access the GPIO pins *on IDLE* using RPi.GPIO

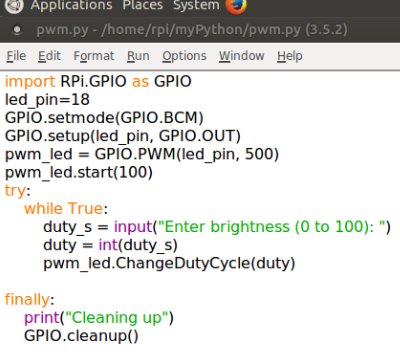
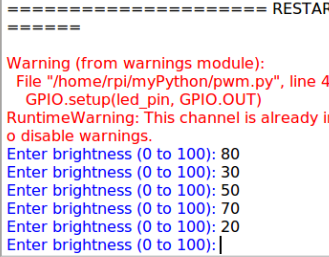


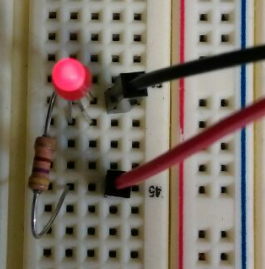
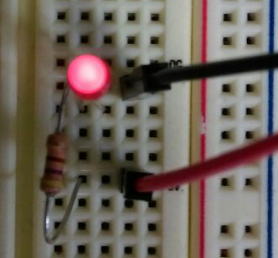
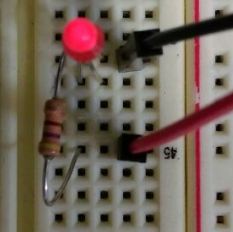
**Analog outputs**

To vary the brightness , it needs to create PWM channel:

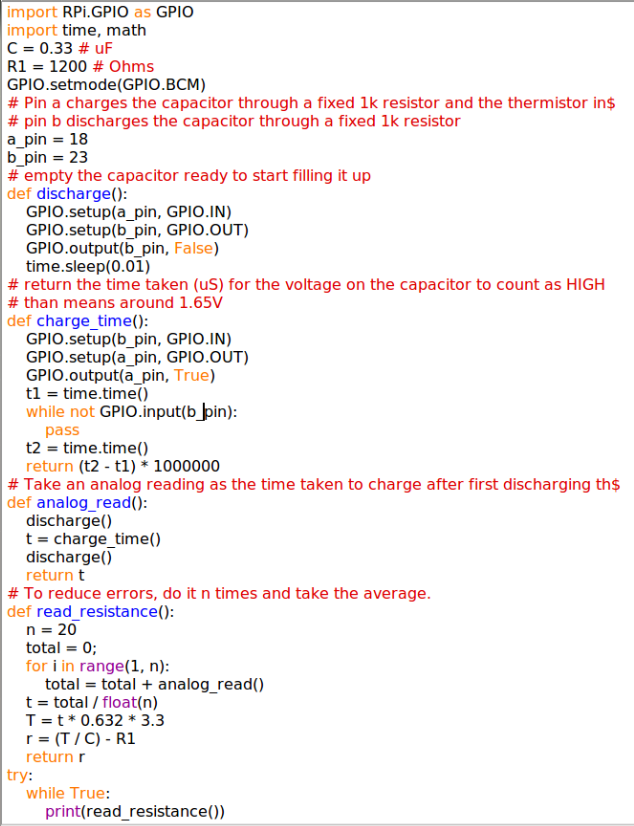
$pwm\_led = GPIO.PWM(led\_pin, 500)

500 specifies the number of pulses per second

**Digital and analog inputs**

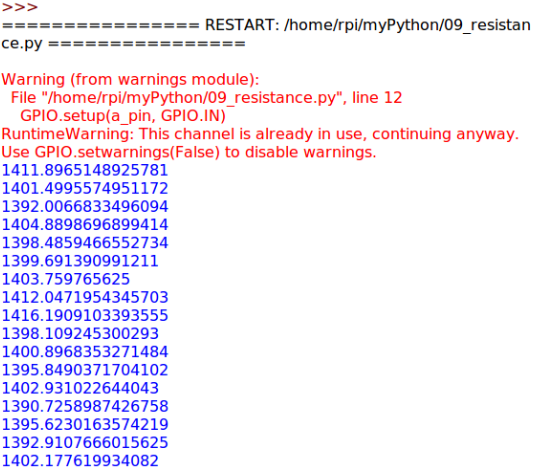


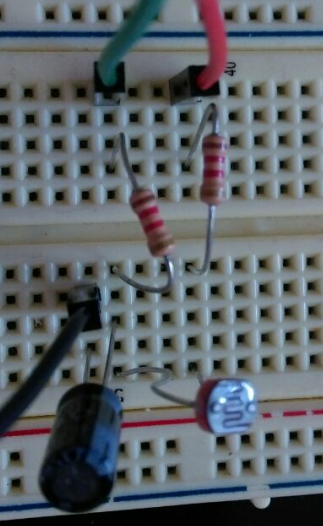
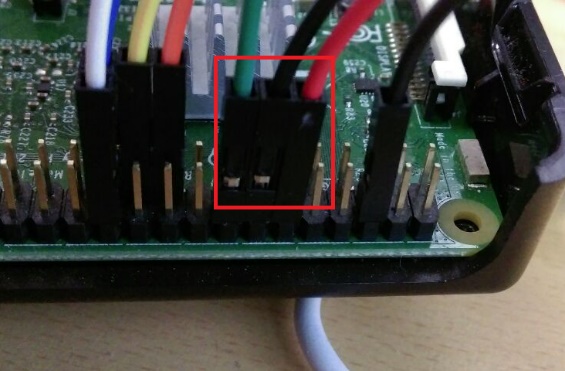
The photo-resistor’s resistance varies depending on the amount of light falling on it.

The “resistive” sensors can be used with a Raspberry Pi by timing how long it takes for current to flow through the resistive sensor and charge up a capacitor to the extent that it crosses the threshold of a digital input so that the input counts as HIGH rather than LOW.

Pin 18 – charging

Pin 23 - uncharging



1. **Project 1**

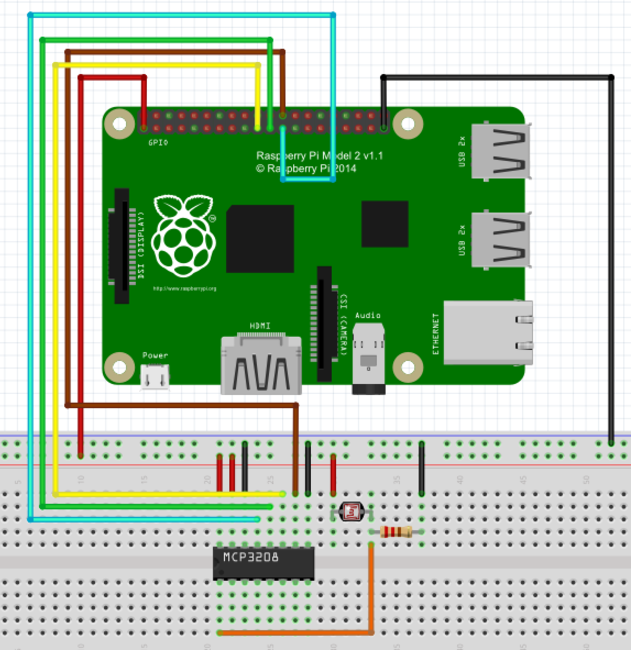
**Raspberry Pi gets light strength data with photo-resistor**

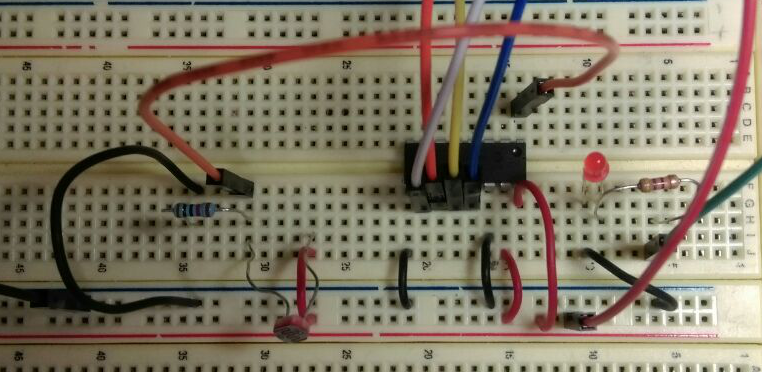
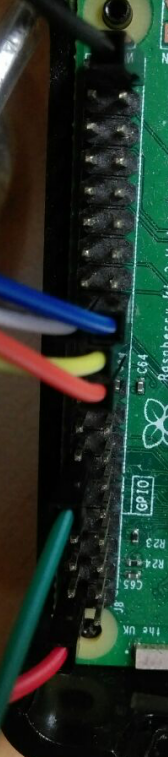
As raspberry pi can not handle analog input, It is needed an ADC adapter to convert analog voltage into digital signal.

Firstly, I connected the raspberry pi with MCP3208 ADC converter following this circuit connection graph [6]. Some others resources are also available such as [7] and [8]. After the photo-resistor (1kOhm) and resistor (10kOhm) were jointed to the analog pin side.

Note: the circuit power was connected to 5V.

|  |
| --- |
| MCP3208 VDD -> 5V (pin 2) |
| MCP3208 VREF -> 5V (pin 2) |
| MCP3208 AGND -> GND (pin 39) |
| MCP3208 CLK -> GPIO11 SPIO\_SCLK (pin 23) |
| MCP3208 DOUT -> GPIO09 SPIO\_MISO (pin 21) |
| MCP3208 DIN -> GPIO10 SPIO\_MOSI (pin 19) |
| MCP3208 CS -> GPIO08 SPIO\_CEO\_N (pin 24) |
| MCP3208 DGND -> GND (pin 39) |





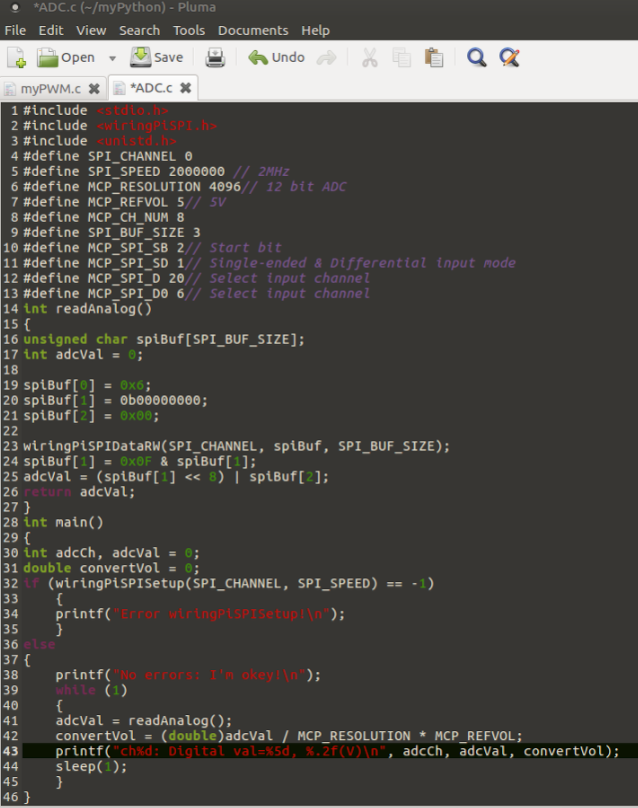
The code was used from [6] and modified by me. The line 42 and 43 in the ADC.c file converts the data related to this reasoning:

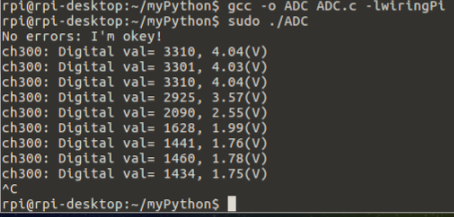
The MCP3208 is a 12-bit ADC. That means it will read a value from 0 to 4095 (212 = 4096 values) where 0 is the same as "ground" and "4095" is the same as "5 volts".  We do not convert the number to voltage, although it’s easy to do that by multiplying the number by (5 / 4095).

Converted number (V) =

$gcc -o spi-test spi-test.c –lwiringPi

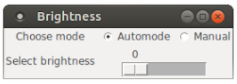
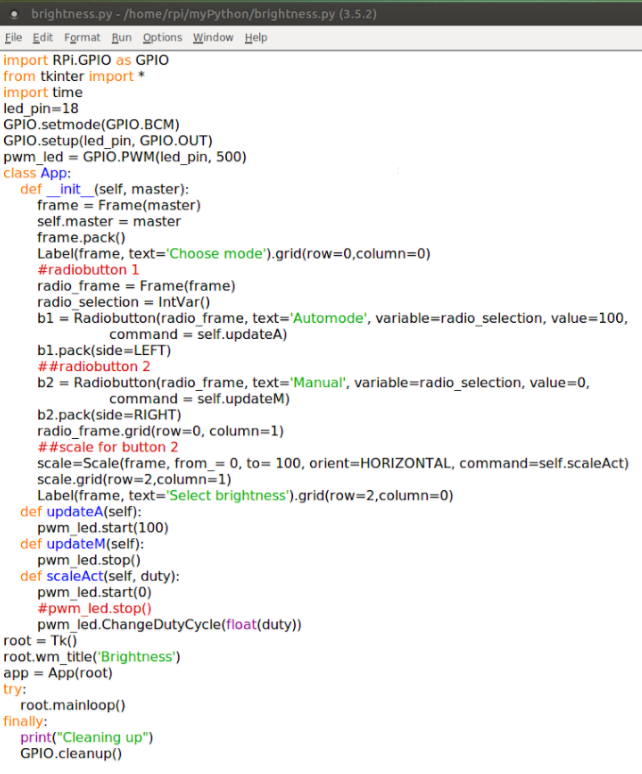
$sudo ./file

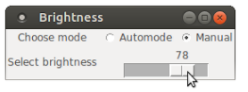
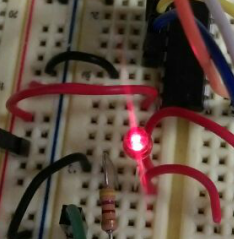


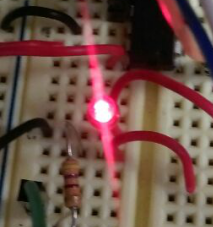


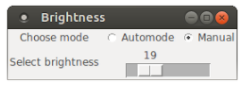
Connect LED to any channels of MCP3208. I chose channel 7 and GPIO 8.

**Python GUI control**









Raspberry Pi has only digital inputs.

The MCP3208 is a digital and analog convertor with 8 analog inputs.

Raspberry pi queries 4 digital pins.

**VDD** (power) and **DGND**(digital ground) to power the MCP3008 chip. We also need four "SPI" data pins: **DOUT** (Data Out from MCP3008), **CLK** (Clock pin), **DIN** (Data In from Raspberry Pi),  and /**CS** (Chip Select).  Finally of course, a source of analog data. We'll be using the basic 10k trim pot.

The MCP3008 has a few more pins we need to connect: **AGND** (analog ground, used sometimes in precision circuitry, which this is not) connects to **GND**, and **VREF** (analog voltage reference, used for changing the "scale" - we want the full scale, so tie it to **3.3V**).

Below is a wiring diagram. Connect the 3.3V cobbler pin to the left + rail and the GND pin to the right - rail. Connect the following pins for the MCP chip

MCP3008 VDD -> 3.3V (red)

MCP3008 VREF -> 3.3V (red)

MCP3008 AGND -> GND (black)

MCP3008 CLK -> #18 (orange)

MCP3008 DOUT -> #23 (yellow)

MCP3008 DIN -> #24 (blue)

MCP3008 CS -> #25 (violet)

MCP3008 DGND -> GND (black)

The MCP3008 is a 10-bit ADC. That means it will read a value from 0 to 1023 (210 = 1024 values) where 0 is the same as "ground" and "1023" is the same as "3.3 volts".  We don't convert the number to voltage, although its easy to do that by multiplying the number by (3.3 / 1023).

Wiring – проводка, соединение

Lots of people have asked about the spi.xfer2 line. This sends 3 bytes to the device. The first byte is 1 which is equal to 00000001 in binary.

“8+channel” is 00001000 in binary (where channel is 0). “<<4” shifts those bits to the left which gives 10000000. The last byte is 0 which is 00000000 in binary.

So “spi.xfer2([1,(8+channel)<<4,0])” sends 00000001 10000000 00000000 to the device. The device then sends back 3 bytes in response. The “data=” line extracts 10 bits from that response and this represents the measurement.

**Used details**

|  |  |  |  |
| --- | --- | --- | --- |
| MCP3208 | 2.7V 4-Channel/8-Channel 12-Bit A/D Converters with SPI™ Serial Interface.  This chip will add 8 channels of 10-bit analog input to your microcontroller or microcomputer project. It's super easy to use, and uses SPI so only 4 pins are required. We chose this chip as a great accompaniment to the Raspberry Pi computer, because its fun to have analog inputs but the Pi does not have an ADC. | Resolution: 12 Bits  A/D: 8 input channels  D/A: No  Interface: SPI  Sampling rate: 100 ksps at VDD = 5V  50 ksps at VDD = 2.7V | <http://ww1.microchip.com/downloads/en/DeviceDoc/21298c.pdf> |
| R2501 HJ341 | Opto-coupler (HIGH ISOLATION VOLTAGE, SINGLE TRANSISTOR TYPE, MULTI PHOTOCOUPLER SERIES) |  | <http://www.cel.com/pdf/datasheets/ps2501.pdf> |
| Power MOSFET  IRF830A | Switch mode power supply  Uninterruptable power supply  High speed power switching |  | <http://www.vishay.com/docs/91061/91061.pdf> |
|  |  |  |  |
|  |  |  |  |

**Appendix**

**Optocoupler**

An optical coupler, also called opto-isolator, optocoupler, opto coupler, photocoupler or optical isolator, is a passive optical component that **can combine or split transmission data** (optical power) from optical fibers. It is an electronic device which is designed to transfer electrical signals by using light waves in order to provide coupling with electrical isolation between its input and output. The main purpose of an optocoupler is to prevent rapidly changing voltages or high voltages on one side of a circuit from distorting transmissions or damaging components on the other side of the circuit. An optocoupler contains a light source often near an LED which converts **electrical input signal into light**, a closed optical channel and a photosensor, which detects incoming light and either modulates electric current flowing from an external power supply or generates electric energy directly. The sensor can either be a photoresistor, a silicon-controlled rectifier, a photodiode, a phototransistor or a triac [1].

**ADC**

Analog-to-digital converter (ADC) is a system that converts an [analog signal](https://en.wikipedia.org/wiki/Analog_signal), such as a sound picked up by a [microphone](https://en.wikipedia.org/wiki/Microphone) or light entering a [digital camera](https://en.wikipedia.org/wiki/Digital_camera), into a [digital signal](https://en.wikipedia.org/wiki/Digital_signal_(signal_processing))

Resolution in this context refers to the conversion of an analog voltage to a digital value in a computer (and vice versa). A computer is a digital machine and thus stores a number as a series of ones and zeroes. If you are storing a digital 2-bit number you can store 4 different values: 00, 01, 10, or 11 [2].

A 12-bit digital value can represent 4096 (212) different numbers [3]

**PWM**

Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a modulation technique used to encode a message into a pulsing signal. Although this modulation technique can be used to encode information for transmission, its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors.

References:

1. <https://www.futureelectronics.com/en/optoelectronics/optocouplers.aspx>
2. <https://labjack.com/support/faq/what-does-12-or-16-bit-resolution-mean>
3. <http://www.ni.com/white-paper/4806/en/>
4. <https://raspberrypi.stackexchange.com/questions/12966/what-is-the-difference-between-board-and-bcm-for-gpio-pin-numbering>
5. <https://stackoverflow.com/questions/8683217/when-do-i-need-to-call-mainloop-in-a-tkinter-application>
6. <http://osoyoo.com/2016/12/16/raspberrypi-photoresistor/>
7. <http://www.icbanq.com/PBLOGER/BOARD_VIEW.ASPX?number=269>
8. <https://pimylifeup.com/raspberry-pi-adc/>
9. <https://www.westfloridacomponents.com/blog/what-is-an-oscilloscope-and-how-does-it-work/>
10. <http://whatis.techtarget.com/definition/oscilloscope>