Word Count: 1095

Video, Working with LDR and PIR Sensors

In this video, we will learn about the working principle behind popular sensors like LDR and PIR, later we will learn how to interface them to the Raspberry Pi, and create some simple projects.

An LDR, also known as a Light Dependent Resistor, is a passive electronic component, basically, a resistor that has a resistance that varies depending on the light intensity.

So, how does an LDR work?

.

It has two wire leads which terminate in the face of the light detector at the two metal dots you see on its face. The main body of the light detector component is made of an insulator called ceramic. On the face of the ceramic, a thin strip of cadmium sulfide is coated in a zig-zag pattern to maximize the length of the strip while keeping the component small. This is connected at each end to an electrode. The front face is then coated in clear plastic, epoxy, glass, or similar for protection. Cadmium-sulfide is a photoconductive material, which means that photons of light hitting it with sufficient energy will release electrons from their atomic bonds. The higher the light intensity, the more photons of light are hitting the Cadmium sulfide strip, the more electrons are freed, the more electricity can flow through the light detector. This is the working principle of an LDR.

If we want to use analog sensors like the LDR with the Raspberry Pi 4, we would need to be able to measure the resistance of the sensor. Unlike the Arduino, the Raspberry Pi's GPIO pins are unable to measure resistance and can only sense if the voltage supplied to them is above 2 volts. To overcome this issue, you could use an Analogue to Digital Converter (ADC), or you could use a relatively cheap capacitor instead. Here we will be using a 1 uF Electrolytic capacitor.

We will build a simple project using an LDR and an LED. When the LDR detects that it is dark, the LED should be ON. Thus, lets continue from our last circuit. You just have to remove any extra LEDs. Only the Red LED connected to the GPIO 17 shall remain. Now fix the LDR on the breadboard in such a way that one of its legs is connected to a 3V pin on the Pi. Just like a resistor, LDR doesn’t have any polarity. Thus you can use any legs. Next, take the 1uF electrolytic capacitor and fix the longer leg or the positive leg to the other leg of the LDR. The negative leg of the capacitor should go to the ground. Finally, connect a jumper from the junction point between the LDR and the Capacitor and wire it up to the GPIO 18 pin. You can find the circuit diagram in the resources.

In the Thonny IDE, open the LDR LED.py script from the repository and run the script. Now you block the face of the LDR, and you can see that the LED is ON. There are three key differences in the code as opposed to the Button LED.py code. One is that we have imported the LightSensor class from the gpiozero library. The second is the input arguments of the class when the object is created. The first one is obviously the GPIO pin, the second is the threshold, which can be set from 0 to 1. Basically, the Pi sees the LDR values as floating numbers from anywhere between 0 to 1. Here, If the average of the LDR values within a default time limit is above 0.2, then its detected as “light”, if it's below its detected as dark. The detection part comes with the light\_detected method, which is used in a conditional branching kept inside an infinite loop to detect and respond. Please check out the resources section to find all the methods available for the LightSensor Class.

Now let us look at the PIR sensor.

The term PIR is the short form for Passive Infra-Red. The term “passive” indicates that the sensor does not emit the referred IR signals itself, rather passively detects the infrared radiation coming from the human body in the surrounding area.

PIRs are basically made of a pyroelectric sensor and a Fresnel lens. The pyroelectric sensor in a motion detector is actually split into two halves. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low. So when a human body or any animal passes by, it intercepts one slot of the PIR Sensor. This causes a positive differential change in between the two halves. When the subject leaves the sensing area, the sensor generates a negative differential change between the two halves. The Fresnel lens is used so that the two halves of the sensor can detect motion from longer distances.

Next, we will interface the PIR Sensor and detect motion. The PIR Sensor is a self-contained sensor with all the necessary circuitry inbuilt on the PCB. Thus, we need to wire it up directly. A PIR sensor will have three pins for interfacing. VCC, Ground, and D out. D out means Digital out. Unlike the LDR sensor, PIR sensor will only return either a 1 or 0. As the pins of the PIR sensor are male header types, we need a female to female jumper cable to interface.

First Connect the VCC pin to a 3V3 pin on the Pi, next connect the Ground pin to any of the ground pins of the pi. Finally, connect the D out pin to GPIO pin 4 of the Pi. Now open the PIR LED.py code in Thonny IDE and run the script. Move your hand in front of the PIR Sensor, and you can see the LED lighting up. The code is almost similar to the LDR LED.py code. The only difference is that we have imported the MotionSensor class, used it to create an object name “pir” and finally detected the motion with the “motion\_detected” method. We have added a half-second delay to avoid the LED to flicker on false positive detections.

Summary

In this video, we have covered the following

● Working Principle of an LDR

● Interfacing of an LDR with the Pi

● Working Principle of a PIR

● Interfacing of a PIR with the Pi

In the next video, we will learn to interface and work with Relays.