**LDR interfacing with Raspberry pi**

The readLDR () function returns a count which is proportional to the light level. In this function the LDR pin is set to output and low and then to input. At this point the capacitor starts charging through the resistor (and a counter is started) until the input pin reads high (this happens when capacitor voltage becomes greater than 1.4V).  The counter is stopped when the input reads high. The final count is proportional to the light level as greater the amount of light, smaller is the LDR resistance and greater is the time taken to charge the capacitor.

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

ldr\_threshold = 1000

LDR\_PIN = 18

LIGHT\_PIN = 25

def readLDR(PIN):

reading = 0

GPIO.setup(LIGHT\_PIN, GPIO.OUT)

GPIO.output(PIN, false)

time.sleep(0.1)

GPIO.setup(PIN, GPIO.IN)

while (GPIO.input (PIN) ==Flase):

reading=reading+1

return reading

def switchOnLight(PIN):

GPIO.setup(PIN, GPIO.OUT)

GPIO.output(PIN, True)

def switchOffLight(PIN):

GPIO.setup(PIN, GPIO.OUT)

GPIO.output(PIN, False)

while True:

ldr\_reading = readLDR(LDR\_PIN)

if ldr\_reading < ldr\_threshold:

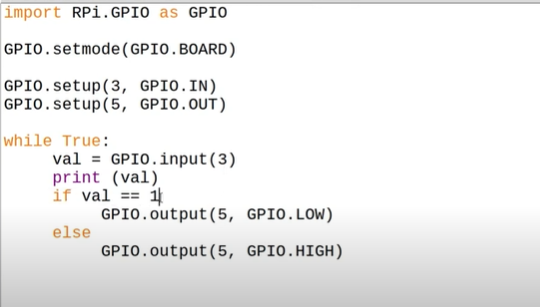
switchOnLight (LIGHT\_PIN)

else:

switchOffLight(LIGHT\_PIN)

time.sleep(1)

**IR sensor interfacing with raspberry pi**



**LCD interfacing with raspberry pi**

import Adafruit\_CharLCD as LCD

import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

lcd\_rs = 27

lcd\_en = 22

lcd\_d4 = 25

lcd\_d5 = 24

lcd\_d6 = 23

lcd\_d7 = 18

lcd\_columns = 16

lcd\_rows = 2

lcd\_backlight = 0 //gnd

lcd = LCD.Adafruit\_CharLCD (lcd\_rs, lcd\_en, lcd\_d4, lcd\_d5, lcd\_d6, lcd\_d7, lcd\_columns, lcd\_rows, lcd\_backlight)

# Print a two-line message

lcd. message('Hello\nworld!')

# Wait 5 seconds

time.sleep(5.0)

# Demo showing the cursor.

lcd.clear()

lcd.show\_cursor(True)

lcd.message('Show cursor')

time.sleep(5.0)

# Demo showing the blinking cursor.

lcd.clear()

lcd.blink(True)

lcd.message('Blink cursor')

time.sleep(5.0)

# Stop blinking and showing cursor.

lcd.show\_cursor(False)

lcd.blink(False)

**Temperature sensor interfacing with Raspberry pi**

import Adafruit\_CharLCD as LCD

import Adafruit\_DHT

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

lcd\_rs = 27

lcd\_en = 22

lcd\_d4 = 25

lcd\_d5 = 24

lcd\_d6 = 23

lcd\_d7 = 18

lcd\_columns = 16

lcd\_rows = 2

lcd\_backlight = 0

lcd\_backlight = 0 //gnd

lcd = LCD.Adafruit\_CharLCD (lcd\_rs, lcd\_en, lcd\_d4, lcd\_d5, lcd\_d6, lcd\_d7, lcd\_columns, lcd\_rows, lcd\_backlight)

while True:

humidity, temperature = Adafruit\_DHT.read\_retry (Adafruit\_DHT.DHT11, 2)

LCD. Clear ()

lcd. message (“Temp:” + str(temperature) +” C\n Humidity:” + str (Humidity) + “%”)

time. sleep (0.1)