```
Import RPi.GPIO as GPIO
Import time
Import math
Import requests, json
Import Adafruit_DHT as dht
Import urllib.request as urllib2
Import Adafruit_GPIO.SPI as SPI
Import Adafruit_MCP3008

myAPI = ‘Z4HU0KS6AKT7H695’
# URL where we will send the data, Don’t change it
baseURL = ‘https://api.thingspeak.com/update?api_key=%s’ % myAPI

SPI_PORT   = 0
SPI_DEVICE = 0
Mcp = Adafruit_MCP3008.MCP3008(spi=SPI.SpiDev(SPI_PORT, SPI_DEVICE))

GPIO.setmode(GPIO.BCM)

GPIO.setup(13,GPIO.IN)
GPIO.setup(19,GPIO.OUT)
GPIO.setup(26,GPIO.IN)

Sensor = dht.DHT11
Dht11_pin = 4  # The Temperature And Humidity Sensor goes on digital port 2.
Light_sensor_pin = 13
Trig_pin = 19
Echo_pin = 26
```

```
Temp=0.0

Humidity = 0.0

Light_value = 0

Distance = 0.0

Duration = 0.0

Gas_value = 0.0

Moisture_value = 0.0


Print(“***********************”)

Print(“”)

Print(“IOT Development Kit”)

Print(“”)

Print(“***********************”)

Print(“”)

Print(“”)



While True:

   Try:

      #IR Snesor


      Print(“Scanning for sensors data….”)

      Time.sleep(2)

      Print(“--------------------“)

      #print()

      Humidity, temp = dht.read_retry(sensor, dht11_pin)


      If math.isnan(temp) == False and math.isnan(humidity) == False:
```

```
    #print(“Temparature and Humidity sensor value”)
    Print(“------------------“)
    Print(“Temparature = %.02f C”%(temp))
    Print(“------------------“)
    Print(“Humidity = %.02f%%”%(humidity))
    Print(“------------------“)

Light_value = GPIO.input(light_sensor_pin)
If light_value==0 or light_value==1:
    If light_value==0:
        Print(“------------------“)
        Print (‘Light Detected’)
        Print(“------------------“)
    Else:
        Print(“------------------“)
        Print (‘Light Not Detected’)
        Print(“------------------“)

GPIO.output(trig_pin, False)
Print(“”)
Print(“------------------“)
#Set TRIG as LOW
Print (“Waitng For Sensor To Settle”)
Time.sleep(2)                   #Delay of 2 seconds
Print(“------------------“)
GPIO.output(trig_pin, True)           #Set TRIG as HIGH
Time.sleep(0.00001)             #Delay of 0.00001 seconds
GPIO.output(trig_pin, False)          #Set TRIG as LOW
```

```
While GPIO.input(echo_pin)==0:             #Check whether the ECHO is LOW
    Pulse_start = time.time()            #Saves the last known time of LOW pulse

While GPIO.input(echo_pin)==1:             #Check whether the ECHO is HIGH
    Pulse_end = time.time()              #Saves the last known time of HIGH pulse

Pulse_duration = pulse_end – pulse_start #Get pulse duration to a variable

Distance = pulse_duration * 17150        #Multiply pulse duration by 17150 to get distance
Distance = round(distance, 2)            #Round to two decimal points

If distance > 2 and distance < 400:
    #Check whether the distance is within range
    Print(“Distance:”,distance-0.5,”cm”)
Else:
    Print(“------------------“)
    Print (“Out Of Range”)
    Print(“------------------“)
Gas_value = mcp.read_adc(0)
Moisture_value = mcp.read_adc(1)
Print(“------------------“)
Print(“Gas Value : “)
Print(“------------------“)
Print(gas_value)
Print(“------------------“)
Print(“soil Moisture Value : “)
Print(“------------------“)
Print(“moisture_value”)
Print(“------------------“)
```

```
        Conn = urllib2.urlopen(baseURL +
‘&field1=%s&field2=%s&field3=%s&field4=%s&field5=%s&field6=%s’ % (temp,
humidity,light_value,distance,gas_value,moisture_value))

        Print(“”)

        Print(“------------------“)

        Print(“Data Sent to cloud”)

        Print(“------------------“)

        Print(“”)

        Print(“”)

        Time.sleep(5)

    Except:

        Print(“exception”)

        Break
```