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