8)

```
import RPi.GPIO as GPIO
import requests
import time

api_key = "your_api_key_here"
channel_id = "your_channel_id_here"

LDR_PIN = 7
LED_PIN = 13

GPIO.setmode(GPIO.BOARD)
GPIO.setup(LDR_PIN, GPIO.IN)
GPIO.setup(LED_PIN, GPIO.OUT)

try:
    while True:

        ldr_value = GPIO.input(LDR_PIN)

        GPIO.output(LED_PIN, ldr_value)



        params = {'api_key': api_key, 'field1': ldr_value}

        try:

            response = requests.get(f"https://api.thingspeak.com/update", params=params)
            print(f"LDR Value: {ldr_value}, LED State: {'ON' if ldr_value else 'OFF'}, Response: {response.text}")
        except Exception as e:
            print(f"Failed to update ThingSpeak: {e}")



        time.sleep(5)

except KeyboardInterrupt:
    print("Program terminated")

finally:

    GPIO.cleanup()
```

----

7)

```
import RPi.GPIO as GPIO
import smtplib
from email.mime.text import MIMEText
import time

EMAIL_ADDRESS = 'your_email@example.com'
EMAIL_PASSWORD = 'your_password'
TO_EMAIL = 'recipient_email@example.com'

SENSOR_PIN = 11
```

```
BUZ_PIN = 13

GPIO.setmode(GPIO.BOARD)
GPIO.setup(SENSOR_PIN, GPIO.IN)
GPIO.setup(BUZ_PIN, GPIO.OUT)

def send_email():

    msg = MIMEText('Rain/Water detected!')
    msg['Subject'] = 'Alert: Rain/Water Detected'
    msg['From'] = EMAIL_ADDRESS
    msg['To'] = TO_EMAIL

    try:
            with smtplib.SMTP_SSL('smtp.gmail.com', 465) as server:
            server.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
            server.send_message(msg)
            print("Email sent successfully")
    except Exception as e:
        print(f"Failed to send email: {e}")

try:
    print("Rain/Water Sensor Test")
    time.sleep(2)
    print("System Ready")

    while True:
        if not GPIO.input(SENSOR_PIN):
            print("Rain/Water detected!")
            GPIO.output(BUZ_PIN, GPIO.HIGH)
            send_email()
            time.sleep(5)
        else:
            GPIO.output(BUZ_PIN, GPIO.LOW)

except KeyboardInterrupt:
    print("Program terminated")

finally:
    GPIO.cleanup()
```

----

6)

```
import time
import adafruit_dht
import board

dht_sensor = adafruit_dht.DHT11(board.D4)

while True:
    try:
        temperature = dht_sensor.temperature
        humidity = dht_sensor.humidity

        print(f'Temperature: {temperature:.1f}°C')
        print(f'Humidity: {humidity:.1f}%')

    except RuntimeError as e:
        print(f"Reading error: {e}")

    time.sleep(2)
```

----

5)

```
import RPi.GPIO as GPIO
import time

PIR_SENSOR_PIN = 11
LED_PIN = 13

GPIO.setmode(GPIO.BOARD)
GPIO.setup(PIR_SENSOR_PIN, GPIO.IN)
GPIO.setup(LED_PIN, GPIO.OUT)

try:
    while True:
        if GPIO.input(PIR_SENSOR_PIN):
            print("Motion detected!")
            GPIO.output(LED_PIN, GPIO.HIGH)
        else:
            print("No motion")
            GPIO.output(LED_PIN, GPIO.LOW)
        time.sleep(2)

except KeyboardInterrupt:
    print("Program terminated")

finally:
    GPIO.cleanup()
```