Roll Number:160116733063

IOTT LAB INTERNAL-2

3) Saving water is one of the most important responsibility of today's citizens. Use the devices and sensors to develop an efficient IOT model for grip irrigation system.

AIM: Use the devices and sensors to develop an efficient IOT model for grip irrigation system.

DESCRIPTION:

- Automated drip irrigation is the most beneficial approach for the farmers.
- This system reduces the extra manpower to the farmer for his farm work like supplying water to plants.
- The smart irrigation system is suitable and cost effective for advance water resources for agricultural production
- **Raspberry pi :** This board is popular nowadays that is used in mini project development . It is chosen because of its low cost and the optimum credit card-size. Raspberry Pi can act like a computer, which connects with a monitor, mouse and keyboard, and it operates in Linux.
- **Soil moisture sensor:** Known as hygrometer sensor and it detects moisture level of soil. This sensor detects the resistance of soil to get its moisture level. The soil with much water has less resistance while dry soil has more resistance.
- LED is used to notify.
- Buzzer sensor is used to ring when the moisture is low

ADVANTAGE:

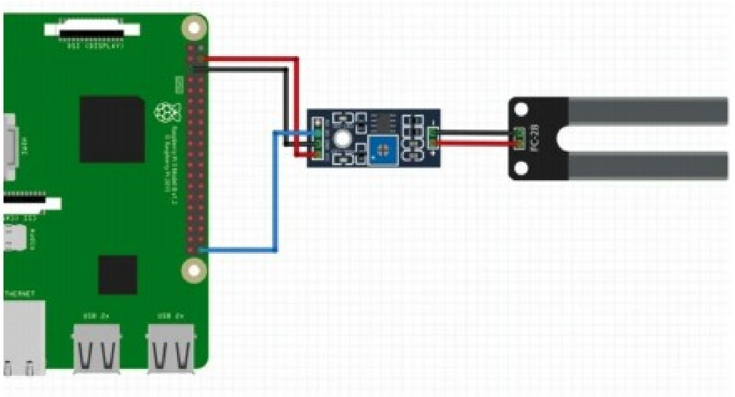
- Increase in productivity.
- Reduced water consumption.
- No man Power Required.
- Require smaller water sources.

REQUIREMENTS:

- Raspberry Pi 3
- 4 Amp Power Adapter
- Micro SD card
- Jumper Cables
- Soil Moisture Sensor

- Breadboard
- MCP3008
- LED

CIRCUTE DIAGRAM:



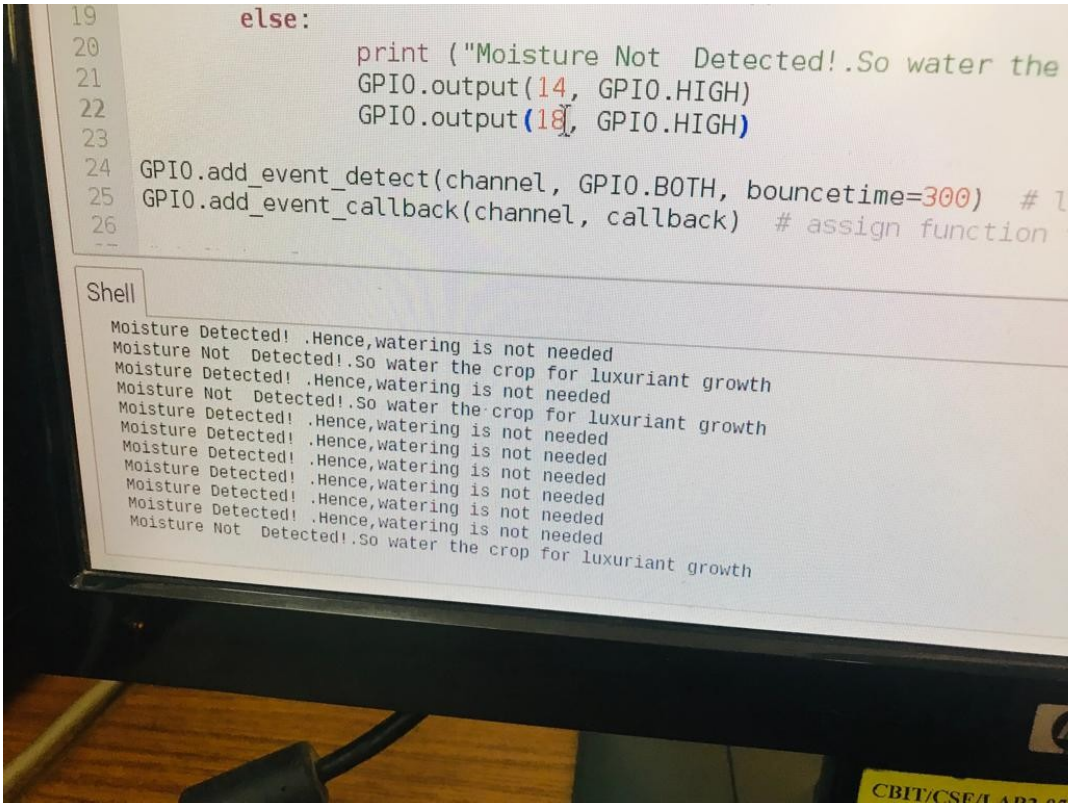
PROGRAM:

```
import RPi.GPIO as GPIO
import time
from datetime import datetime
#import mcp3008
#GPIO SETUP

channel = 21
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(channel, GPIO.IN)
GPIO.setup(14, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(18, GPIO.OUT, initial=GPIO.LOW)
print ("Grip Iriigation system")
print ("Reporting at")

print(datetime.now())
def callback(channel):
    if GPIO.input(channel):
        print ("Moisture Detected! .Hence,watering is not needed")
        GPIO.output(14, GPIO.LOW)
        GPIO.output(18, GPIO.LOW)# Turn on
#print(mcp3008.readadc(21))
    else:
        print ("Moisture Not  Detected!.So water the crop for luxuriant growth")
        GPIO.output(14, GPIO.HIGH)
        GPIO.output(18, GPIO.HIGH)
GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300)
GPIO.add_event_callback(channel,callback)
while True:
        time.sleep(1)
```

STEPS:

- Initialize the raspberry pi on the mointor and connecting soil moisture sensor using jumper wires at thier respective pins at GPIO 14 using channel 21
- The output is displayed accordingly, either water detected if the moisture content is more or moisture not detected
- If water is not detected then the LED cautions by switching on
- We can either use LED or buzzer sensor to indicate the content of the moisture in soil.
- Then the buzzer sensor is connected at GPIO 14 so that it rings when the moisture content is not detected likewise LED follows the same procedure

OUTPUT:



```
        else:
                print ("Moisture Not  Detected!.So water the
                GPIO.output(14, GPIO.HIGH)
                GPIO.output(18, GPIO.HIGH)

GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300)  # l
GPIO.add_event_callback(channel, callback)  # assign function
```

```
Moisture Detected! .Hence,watering is not needed
Moisture Not  Detected!.So water the crop for luxuriant growth
Moisture Detected! .Hence,watering is not needed
Moisture Not  Detected!.So water the crop for luxuriant growth
Moisture Detected! .Hence,watering is not needed
Moisture Detected! .Hence,watering is not needed
Moisture Detected! .Hence,watering is not needed
Moisture Detected! .Hence,watering is not needed
Moisture Detected! .Hence,watering is not needed
Moisture Detected! .Hence,watering is not needed
Moisture Not  Detected!.So water the crop for luxuriant growth
```

Fig-1:Display of the output in shell after execution of the code

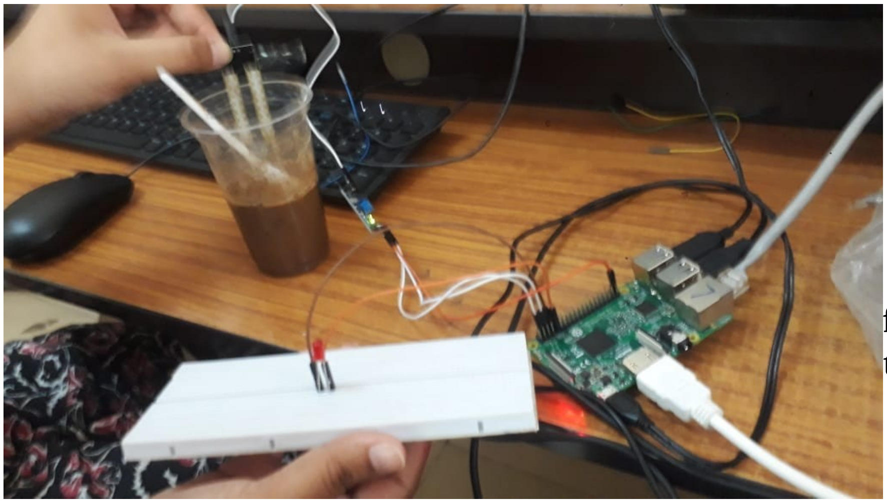


fig-2:LED is switched off as the moisture content is high

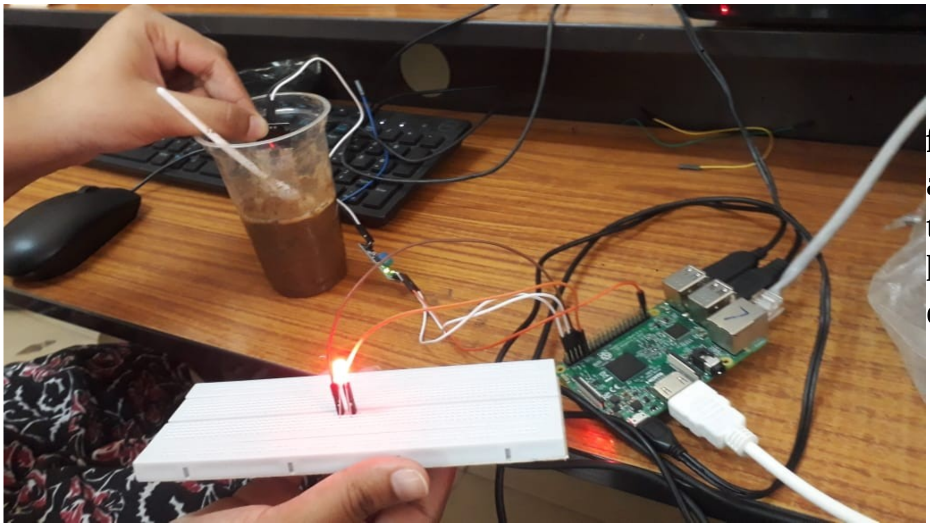


fig-3:LED is switched on automatically to indicate that moisture content is low, so watering should be done

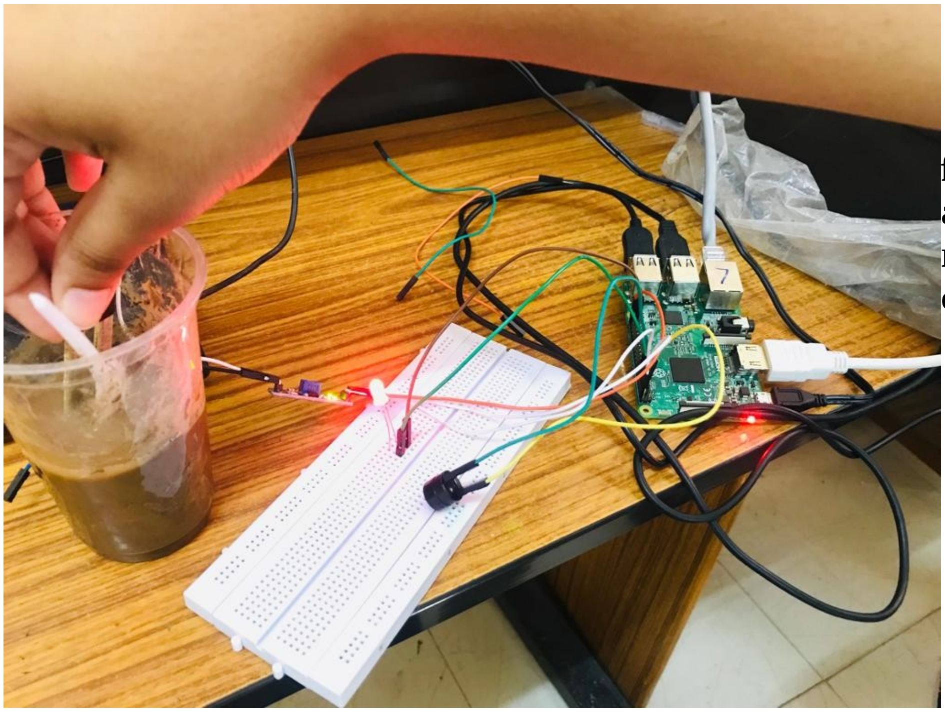
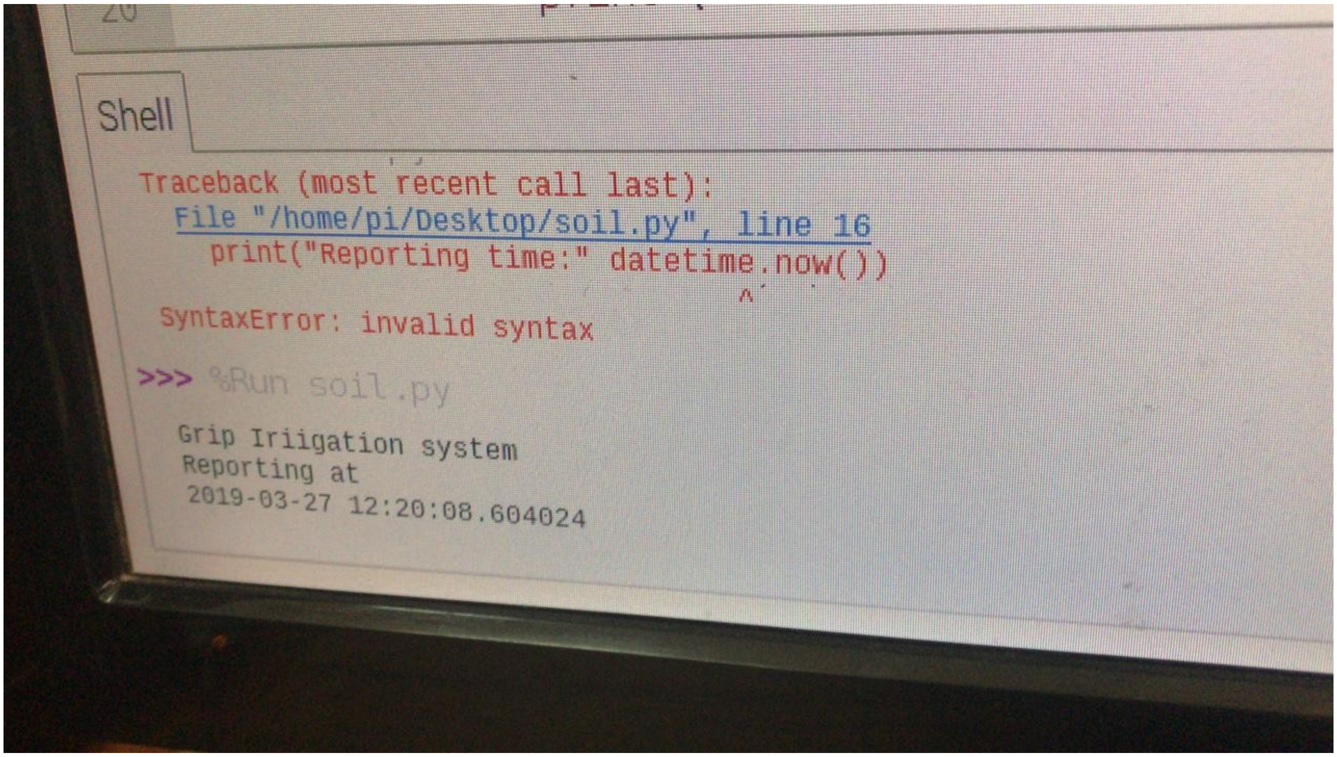


fig-4:LED is switched on and the buzzer sensor rings as the soil moisture content is low.



```
Traceback (most recent call last):
  File "/home/pi/Desktop/soil.py", line 16
    print("Reporting time:" datetime.now())
                                    ^
SyntaxError: invalid syntax

>>> %Run soil.py

  Grip Iriigation system
  Reporting at
  2019-03-27 12:20:08.604024
```

System time is given