LOW LEVEL DESIGN AND IMPLEMENTATION DOCUMENT

IMPLEMENTATION OF PRECISION AGRICULTURE MONITORING SYSTEM USING RASPBERRY PI AND CROP PREDICTION USING MACHINE LEARNING ALGORITHM

UE17CS490B – Capstone Project Phase – 2

Submitted by:

Amala	**01FB17ECS703**
Anusha B	**PES1201701061**
Sharada G	**PES1201802412**
Veena K	**PES1201802492**

Under the guidance of

Prof. CHARANRAJ.B. R
Assistant professor
PES University

January - May 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

TABLE OF CONTENTS

Note:

Section 1	**Common for Prototype/Product Based and Research Projects**
Section 2 & 3	**Applicable for Prototype / Product Based Projects.**
Section 4	**Applicable for Research Projects.**
Appendix	**Provide details appropriately**

1. Introduction

1.1. Overview

This project is about precision agriculture monitoring system using raspberry pi. We are using sensors which senses the parameters details, those details are sent to think speak for visualization purpose. Sensors used are temperature and humidity, fire sensor, soil moisture, Relay for pumping water to fields whenever soil moisture content losses it moisture or value less than a fixed value of moisture. similarly, for temperature and humidity when temperature increases it temperature and humidity loses it humidity content then user gets the message and automatically pump gets on.

Here we are describing about the low level design in which we have divide the project in sublevel so that every processing steps are explained clearly such as class diagram, use case diagram sequence diagram, package development etc. These give us the overview of the project what we are going to do.

1.2. Purpose

Purpose of use precision agriculture is to get the accurate value of the parameters such as soil moisture sensor, humidity & temperature sensors and fire sensor for detection of fire in the farm. In low level design we see the progress of small part of the project, sequence diagram, use case diagram.

2. Design Constraints, Assumptions, and Dependencies

Constraints:

Network feasibility for GSM and things speak and productivity may or may not be more. We cannot estimate weather conditions as pollution is increasing gradually etc.

Software dependencies:

- Pycharm IDE
- Thing speak
- Fast2SMS

Hardware dependencies:

- Raspberry pi3
- DTH sensor
- Soil moisture sensor
- Relay
- Pump
- Power Supply

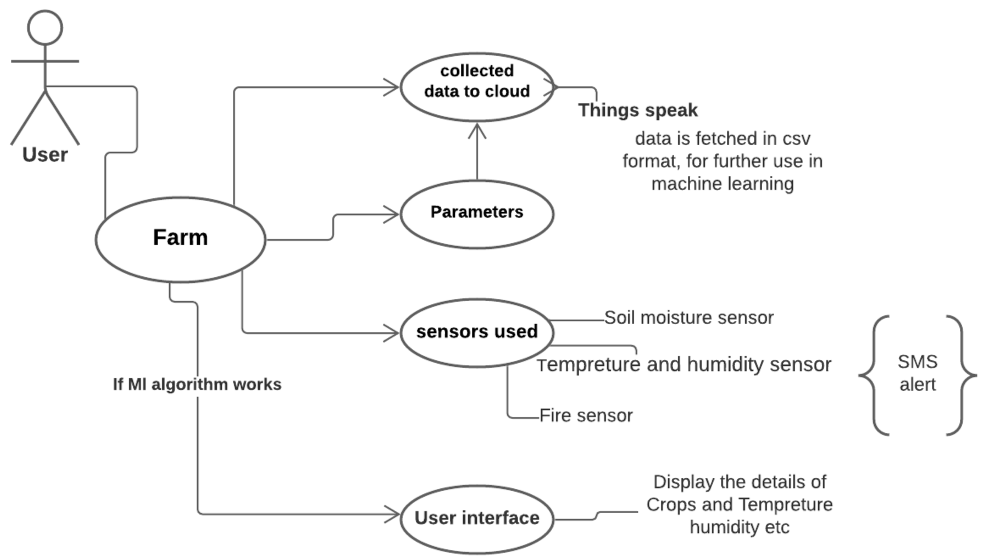
3 Master Class Diagram

3.1.1 Description

In this, whenever the user gets details from the farm those data are stored in the cloud using IOT i.e., Thing speak for now we are using a dataset from Kaggle website based on these data we are trying to predict what type of crop can be grown in these conditions.

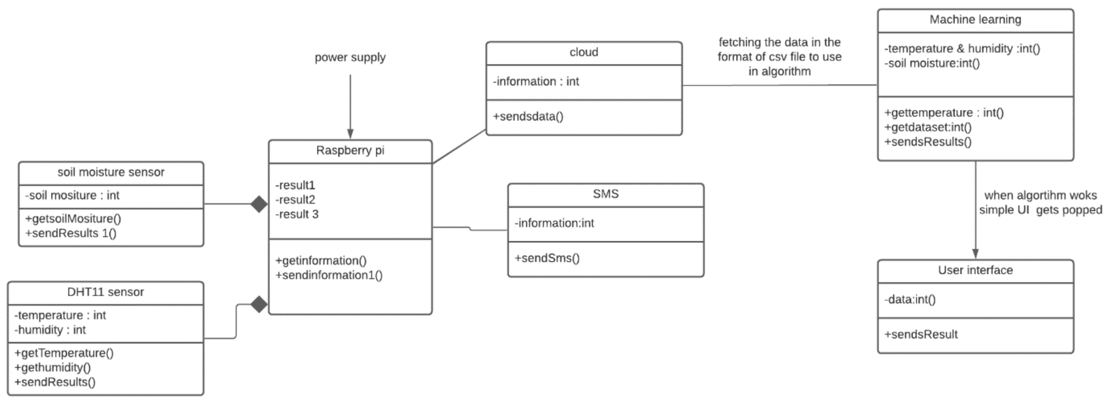
In real time we can take readings every day and can be stored in Thing speak and then after few months, we can extract the files in CSV/XML format which was taken every day based on that we can also predict which type of crop is growing there to that suitable condition. When the algorithm takes data and display in simple UI. This dependent on the algorithm.

3.1.2 Use Case Diagram

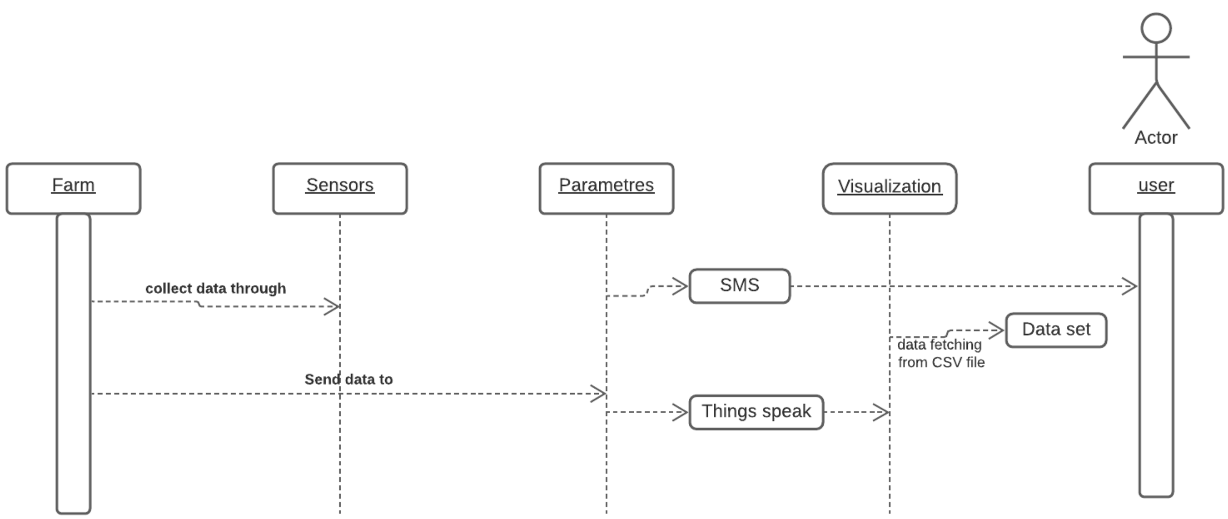


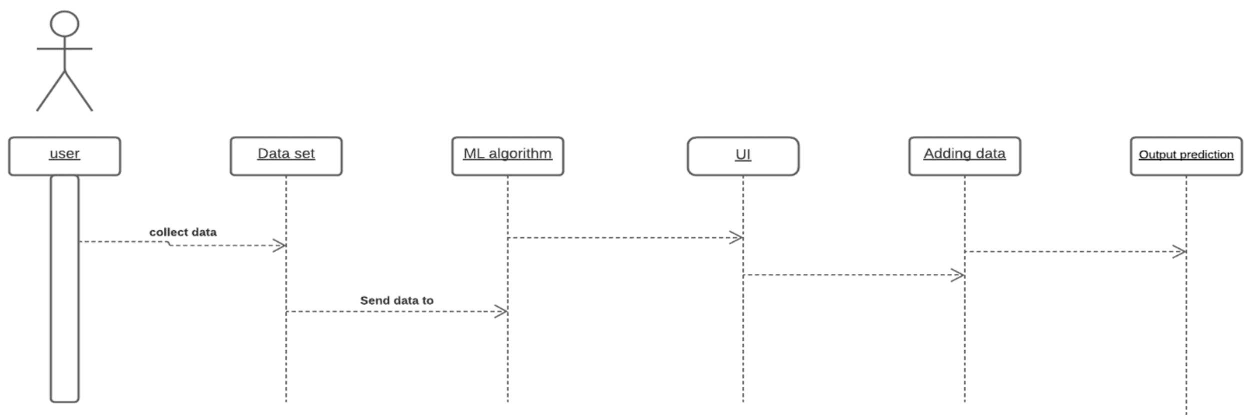
Use case Item	Description
User	User can control the sensor using mobile when get alert sms whenever there is fluctuation in the field
Thing speak	It is used to store data based on daily reading and can get visualization of data as per the requirement.
Algorithm	Process where the data is taken and processed such as classification, splitting the data.
GUI	User interface, in this it is just a simple UI when the algorithm works properly it executes and parameters details should be give so that it will predicts which crop can be grown in that suitable conditions

3.1.3 Class Diagram



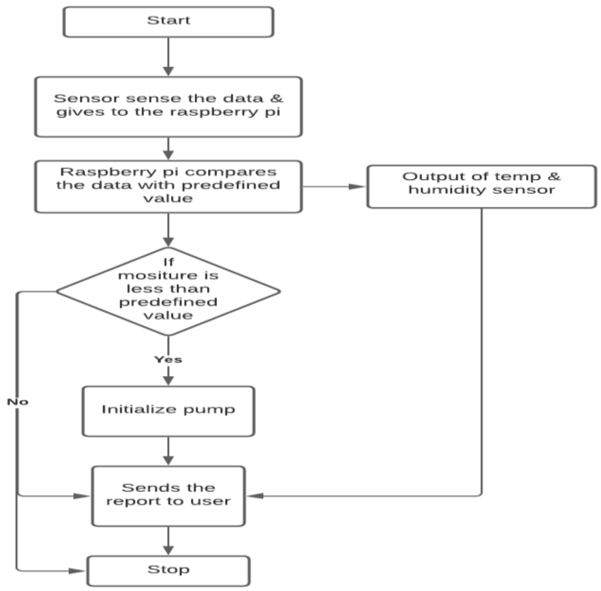
3.1.4 Sequence Diagram





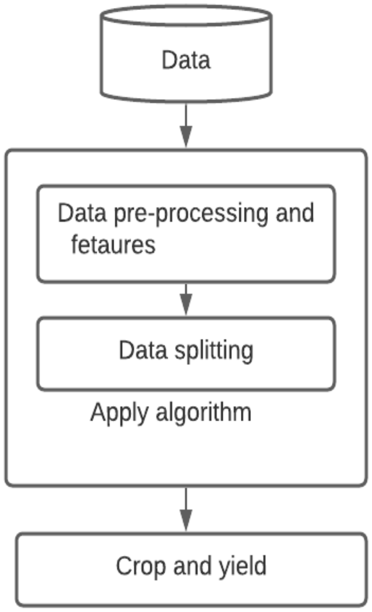
- Hardware should be connected properly to the respective pin connection.
- Sensor will sense the data and those are collected into the cloud that is Things peak is an IOT platform then visualization is done using those data and then dataset is fetched in the format of CSV/XML then further use in algorithm.
- If in case there is any change in the parameter detected value and predefined value, then SMS will be sent to the user saying that increase/decrease in temperature, humidity similarly soil moisture loses its moisture content then user gets the SMS saying that there is no water detected and automatically pump will get ON.
- Raspberry Pi OS installation process is properly done and dump the code to SD card then the System will start.
- After all the connections done sensors will sense.
- The dataset which is fetched in Things peak can be used now in the algorithm (for now we had taken the dataset from Kaggle website).
- If algorithm works properly simple UI will pop up, which is done using Tkinter and the data should be added to that required field.
- Then it predicts the Output

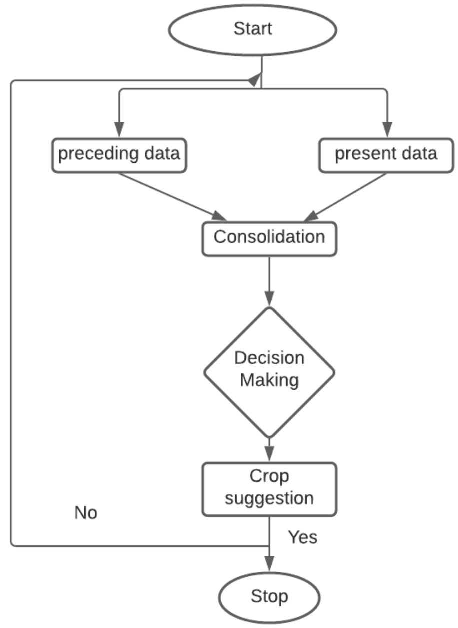
4 Proposed Methodology / Approach



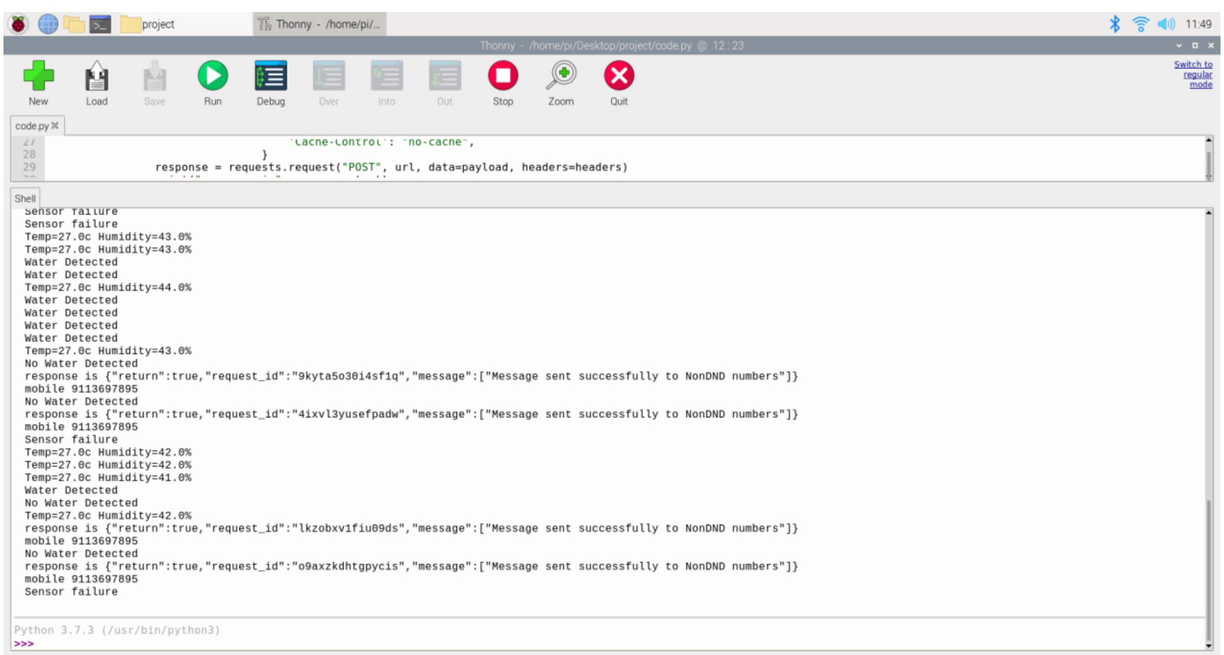
To enhance the productivity of the crop there by supporting both farmer and nation we have to use the technology which estimates the quality of crop and giving suggestions. • Wireless sensor network are sensors of different types are used to collect the information of crop conditions and environmental changes these information is transmitted through network to the farmer or devices that initiates corrective action. It also helps in collecting information about conditions like weather, moisture, temperature and fertility of soil, level of water, pest detection, animal intrusion in to the field, crop growth, agriculture. The proposed model aims at developing a smart system that would provide an ideal environment for the crops. The sensors sense the soil moisture and the humidity levels. This reduces human effort to a great extent and also ensures that an optimal environment is provided for the crops thus improving crop quality

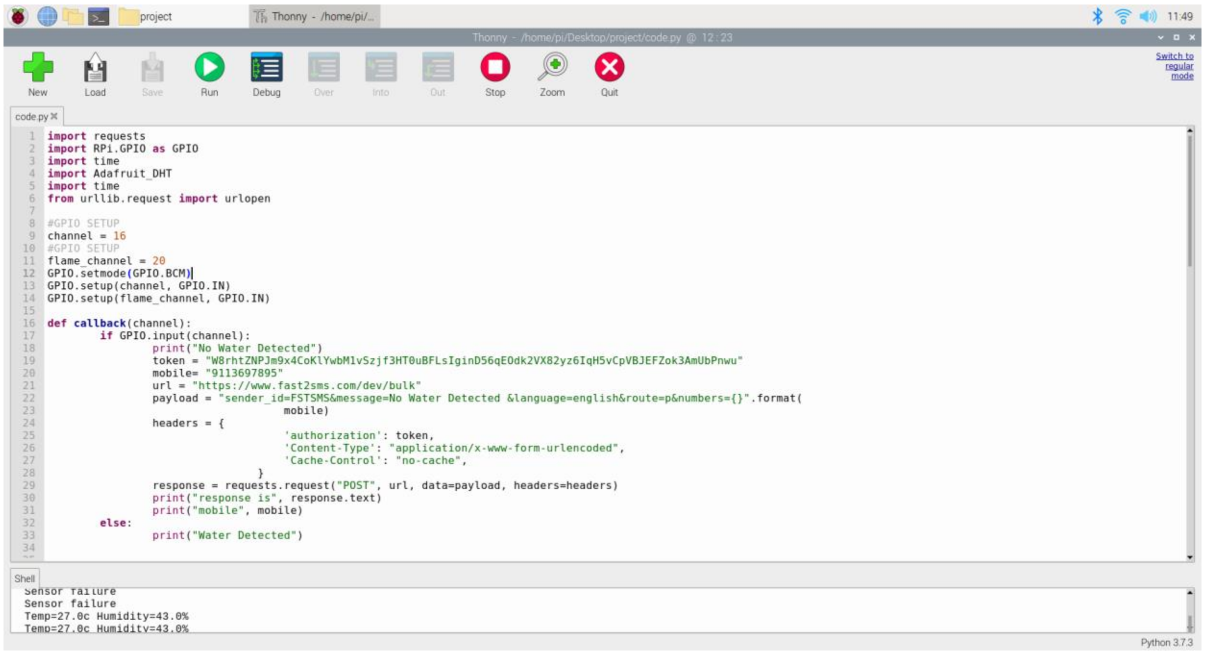
4.1 Algorithm and Pseudocode

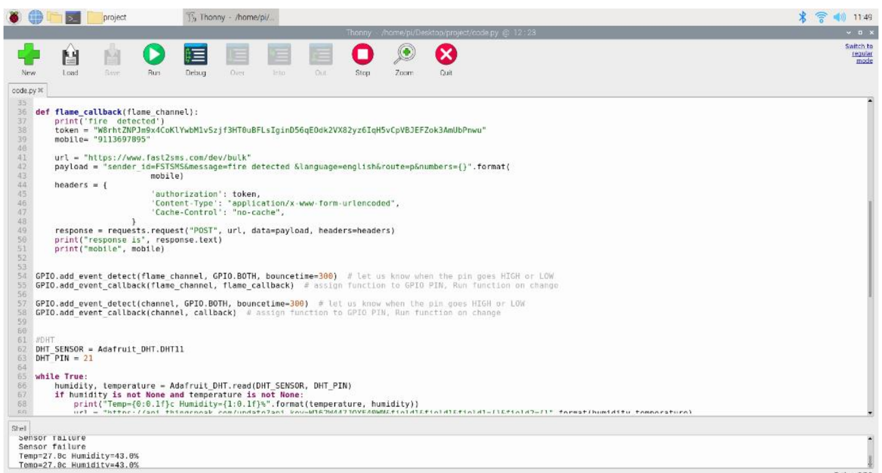


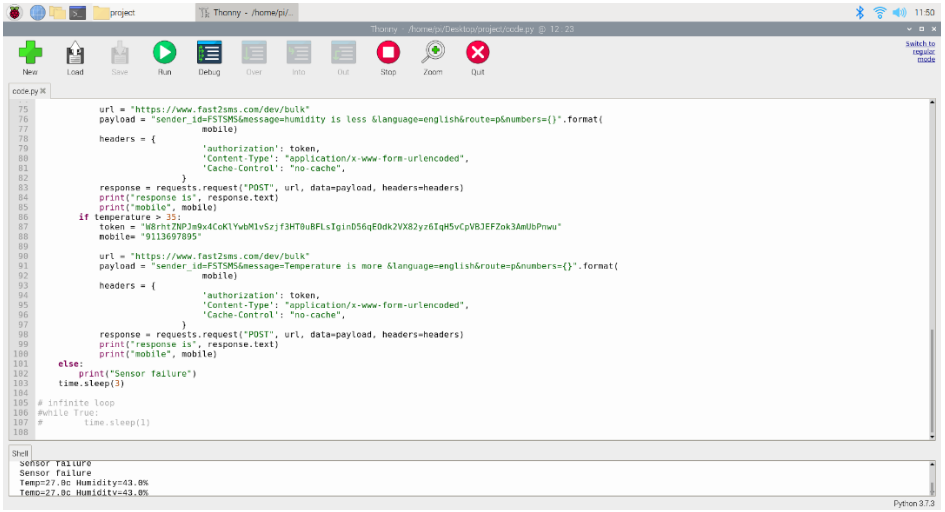


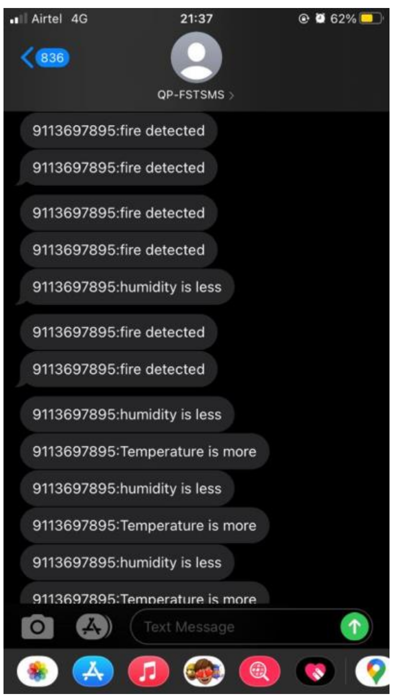
4.2 Implementation and Results



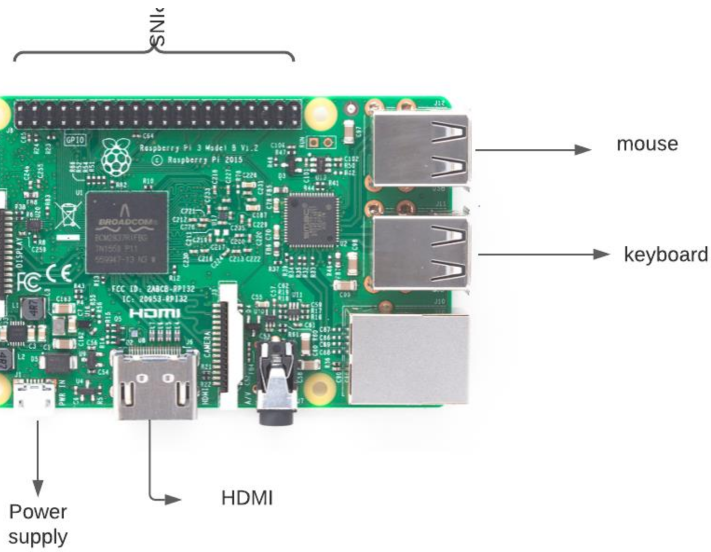

project
Thonny - /home/pi/...
11:49
Thonny - /home/pi/Desktop/project/code.py @ 12 : 23
New
Load
Save
Run
Debug
Over
Into
Out
Stop
Zoom
Quit
Switch to regular mode
code.py
import requests
import RPi.GPIO as GPIO
import time
import Adafruit_DHT
import time
from urllib.request import urlopen
#GPIO SETUP
channel = 16
#GPIO SETUP
flame_channel = 20
GPIO.setmode(GPIO.BCM)
GPIO.setup(channel, GPIO.IN)
GPIO.setup(flame_channel, GPIO.IN)
def callback(channel):
if GPIO.input(channel):
print("No Water Detected")
token = "W8rhtZNPJm9x4CoKlYwbM1vSzjf3HT0uBFLsIginD56qEOdk2VX82yz6IqH5vCpVBJEFZok3AmUbPnwu"
mobile= "9113697895"
url = "https://www.fast2sms.com/dev/bulk"
payload = "sender_id=FSTSMS&message=No Water Detected &language=english&route=p&numbers={}".format(
mobile)
headers = {
'authorization': token,
'Content-Type': "application/x-www-form-urlencoded",
'Cache-Control': "no-cache",
}
response = requests.request("POST", url, data=payload, headers=headers)
print("response is", response.text)
print("mobile", mobile)
else:
print("Water Detected")
Shell
Sensor failure
Temp=27.0c Humidity=43.0%
Temp=27.0c Humidity=43.0%
Python 3.7.3


project
Thonny - /home/pi/...
11:49
Thonny - /home/pi/Desktop/project/code.py @ 12 : 23
New
Load
Save
Run
Debug
Over
Into
Out
Stop
Zoom
Quit
Switch to regular mode
code.py
def flame_callback(flame_channel):
print('fire detected')
token = "W8rhtZNPJm9x4CoKlYwbM1vSzjf3HT0uBFLsIginD56qEOdk2VX82yz6IqH5vCpVBJEFZok3AmUbPnwu"
mobile= "9113697895"
url = "https://www.fast2sms.com/dev/bulk"
payload = "sender_id=FSTSMS&message=fire detected &language=english&route=p&numbers={}".format(
mobile)
headers = {
'authorization': token,
'Content-Type': "application/x-www-form-urlencoded",
'Cache-Control': "no-cache",
}
response = requests.request("POST", url, data=payload, headers=headers)
print("response is", response.text)
print("mobile", mobile)
GPIO.add_event_detect(flame_channel, GPIO.BOTH, bouncetime=300) # let us know when the pin goes HIGH or LOW
GPIO.add_event_callback(flame_channel, flame_callback) # assign function to GPIO PIN, Run function on change
GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300) # let us know when the pin goes HIGH or LOW
GPIO.add_event_callback(channel, callback) # assign function to GPIO PIN, Run function on change
#DHT
DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 21
while True:
humidity, temperature = Adafruit_DHT.read(DHT_SENSOR, DHT_PIN)
if humidity is not None and temperature is not None:
print("Temp={0:0.1f}c Humidity={1:0.1f}%".format(temperature, humidity))
Shell
Sensor failure
Temp=27.0c Humidity=43.0%
Temp=27.0c Humidity=43.0%
Python 3.7.3


Thonny - /home/pi/Desktop/project/code.py
New
Load
Save
Run
Debug
Over
Into
Out
Stop
Zoom
Quit
code.py
url = "https://www.fast2sms.com/dev/bulk"
payload = "sender_id=FSTSMS&message=humidity is less &language=english&route=p&numbers={}".format(
mobile)
headers = {
'authorization': token,
'Content-Type': "application/x-www-form-urlencoded",
'Cache-Control': "no-cache",
}
response = requests.request("POST", url, data=payload, headers=headers)
print("response is", response.text)
print("mobile", mobile)
if temperature > 35:
mobile= "9113697895"
url = "https://www.fast2sms.com/dev/bulk"
payload = "sender_id=FSTSMS&message=Temperature is more &language=english&route=p&numbers={}".format(
mobile)
headers = {
'authorization': token,
'Content-Type': "application/x-www-form-urlencoded",
'Cache-Control': "no-cache",
}
response = requests.request("POST", url, data=payload, headers=headers)
print("response is", response.text)
print("mobile", mobile)
else:
print("Sensor failure")
time.sleep(3)
# infinite loop
#while True:
time.sleep(1)
Shell
Sensor failure
Temp=27.0c Humidity=43.0%
Temp=27.0c Humidity=43.0%
Python 3.7.3

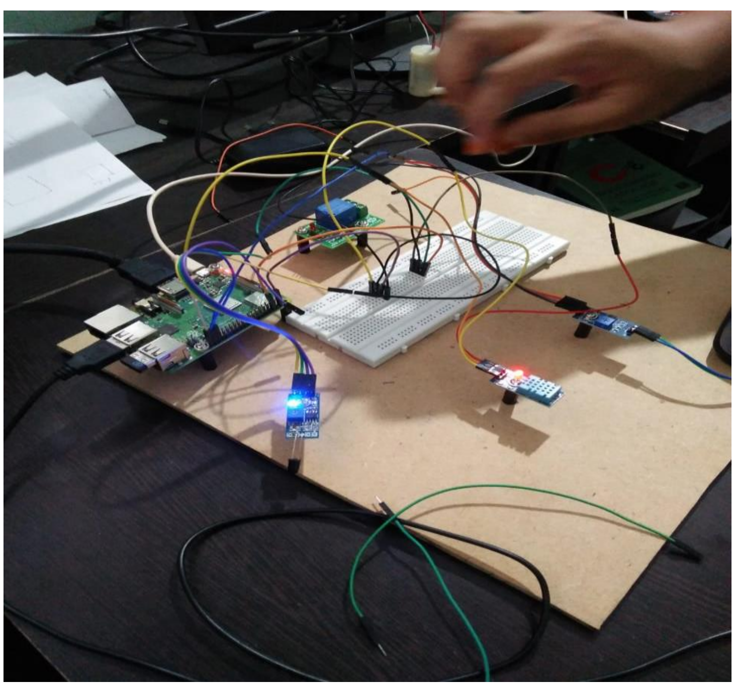

Airtel 4G
21:37
62%
836
QP-FSTSMS
9113697895:fire detected
9113697895:fire detected
9113697895:fire detected
9113697895:fire detected
9113697895:humidity is less
9113697895:fire detected
9113697895:fire detected
9113697895:humidity is less
9113697895:Temperature is more
9113697895:humidity is less
9113697895:Temperature is more
9113697895:humidity is less
Text Message

HARDWARE CONNECTION:



PIN CONNECTION

1)TEMPERATURE AND HUMIDITY
-VCC->5V
-SIG->GPIO21
-GND->GND
2)FLAME SENSOR
-VCC->5V
-SIG->GPIO20
-GND->GND
3)SOIL MOSITURE SENSOR
-VCC->5V
-GND->-GND
-SIG->GPIO16



Appendix : References

- Crop Prediction using Machine Learning Proceedings of the Third International Conference on Smart Systems and Inventive Technology (ICSSIT 2020) IEEE Xplore Part Number: CFP20P17-ART; ISBN: 978-1-7281-5821-1
- Crop Prediction Using Machine Learning Kevin Tom Thomas1 , Varsha S2 , Merin Mary Saji3 , Lisha Varghese4 , Er. Jinu Thomas5 1,2,3,4 UG students Department Computer Engineering, SAINTGITS College of Engineering, APJ ABDUL KALAM TECHNOLOGICAL University, Kerala, India 5Asst. Prof. Department Computer Engineering, SAINTGITS College of Engineering, APJ ABDUL KALAM TECHNOLOGICAL University, Kerala, India
- Wireless Agriculture Monitoring using Raspberry Pi International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERTV6IS050217 (This work is licensed under a Creative Commons Attribution 4.0 International License.) Published by : www.ijert.org Vol. 6 Issue 05, May – 2017
- Raspberry pi based real time monitoring of Agriculture & Irrigation Using IOT Athira P. Shaji MTech Student Computer Science and Engineering, School Of Computer Sciences, MG University , Kottayam , india© IJEDR 2018 | Volume 6, Issue 2 | ISSN: 2321-9939
- Internet of Things for Precision Agriculture Applications 2019 Fifth International Conference on Image Information Processing (ICIIP)
- IOT BASED SMART CROP-FIELD MONITORING AND AUTOMATION IRRIGATION SYSTEM Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018) IEEE Xplore Compliant - Part Number:CFP18J06-ART, ISBN:978-1-5386-0807-4; DVD Part Number:CFP18J06DVD, ISBN:978-1-5386-0806-7