SMART WATER MANAGEMENT

PRESENTED BY:

P.ABINAYA

K.VIJAYALAKSHMI

R.MOHANA

C.SAIPRAVEENA

INTRODUCTION

- WATER SCARCITY AND WATER MANAGEMENT TOPICS ARE VERY CLOSELY RELATED TO EACH OTHER. IMPROPER WATER MANAGEMENT LEADS TO WATER SCARCITY AND VICE VERSA. DON'T ASSUME THAT THIS PROJECT IS ALSO ONE OF THE 1000 SIMILAR PROJECTS BASED ON WATER MANAGEMENT ISSUES
- AM CONFIDENT THAT IF A PERSON GOES THROUGH THIS DOCUMENTATION, HE WILL GET TO KNOW WHY THIS PROJECT IS UNIQUE AND SMART FROM OTHER PROJECTS! NOT ONLY THIS PROJECT SOLVES THE OLD TRADITIONAL PROBLEM OF WATER MANAGEMENT IN A NEW WAY BUT IT ALSO PROVIDES A SMART WAY OF INTERACTING WITH ROBOTS

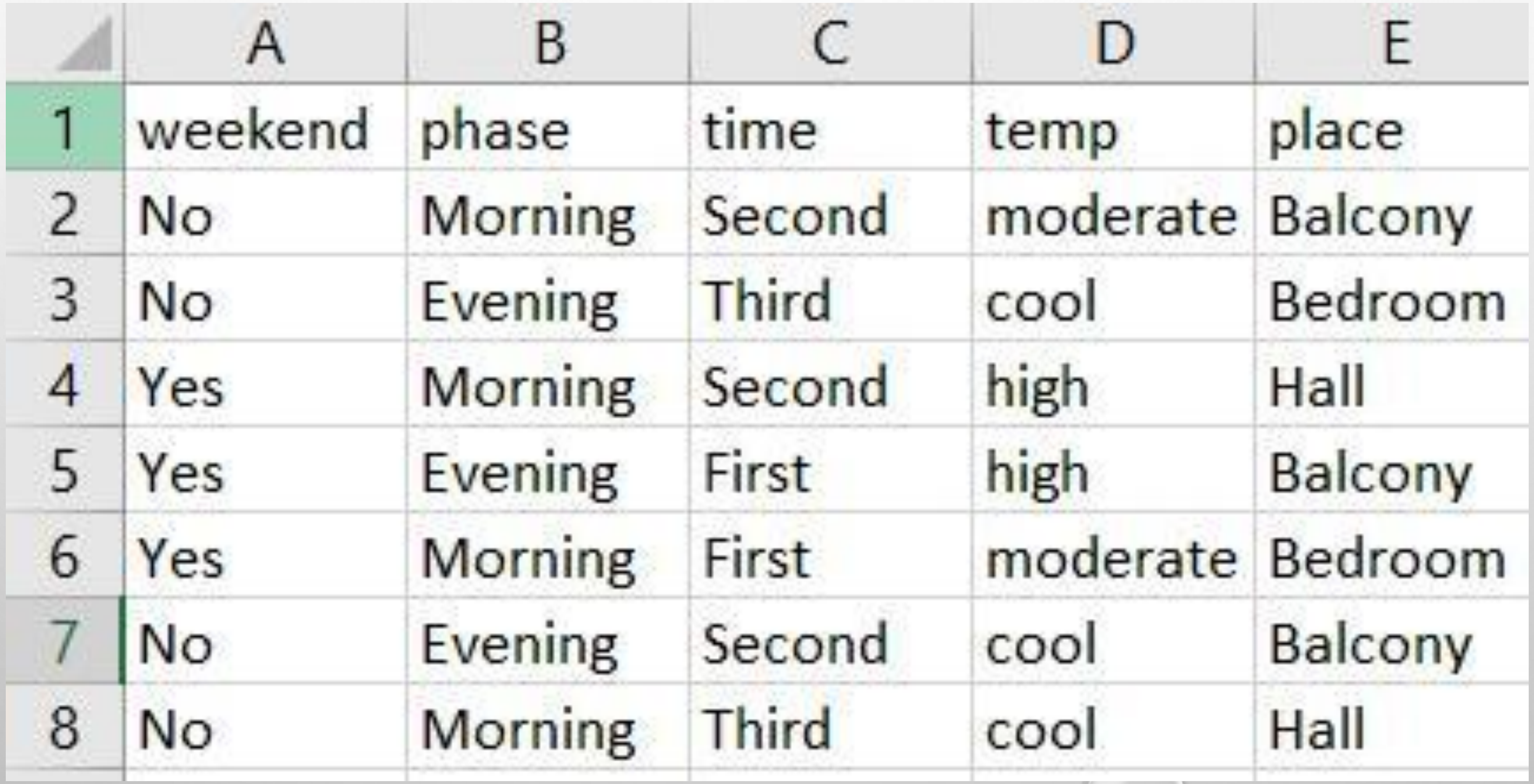
THINGS CAN BE USING IN THIS PROJECT

- HARDWARE COMPONENTS
- SENSORS USED WITH BOLTDUINO/ARDUINO
- 5V RELAY
- I2C LCD
- BOLTDUINO
- 9V BATTERY
- BOLT WIFI MODULE
- IRF540 MOSFET
- WATER FLOW SENSOR
- ULTRASONIC SENSOR X 2
- 1N4007 RECTIFIER DIODE

- SENSORS USED WITH BOLTDUINO/ARDUINO
- NODEMCU
- PIEZO BUZZER
- IR SENSOR X 2
- DC MOTORS X 2
- 12V DC ADAPTER
- TCS3200 COLOR SENSOR
- CAPACITIVE TOUCH SENSOR
- ESP8266 MOTOR DRIVER SHIELD
- ANALOG MULTIPLEXER IC – CD4051

- SOFTWARE APPS AND ONLINE SERVICES
- ARDUINO IDE
- BOOTSTRAP STUDIO
- SPYDER (ANACONDA)
- TWILIO
- CANVA
- HOSTINGER
- INTEGROMAT
- MEGA CREATOR
- **PICHON** (ICONS8

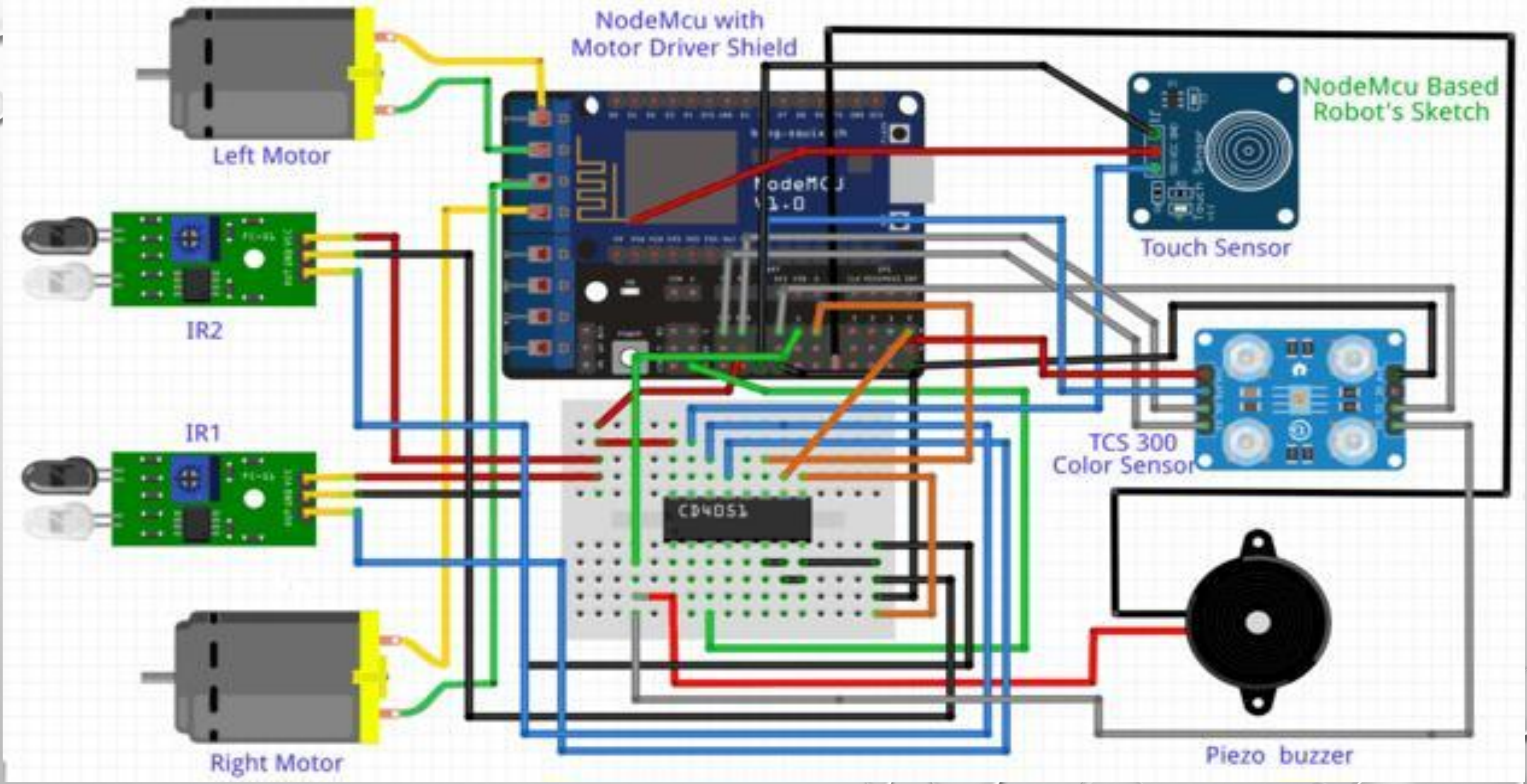
TABULATION



	A	B	C	D	E
1	weekend	phase	time	temp	place
2	No	Morning	Second	moderate	Balcony
3	No	Evening	Third	cool	Bedroom
4	Yes	Morning	Second	high	Hall
5	Yes	Evening	First	high	Balcony
6	Yes	Morning	First	moderate	Bedroom
7	No	Evening	Second	cool	Balcony
8	No	Morning	Third	cool	Hall

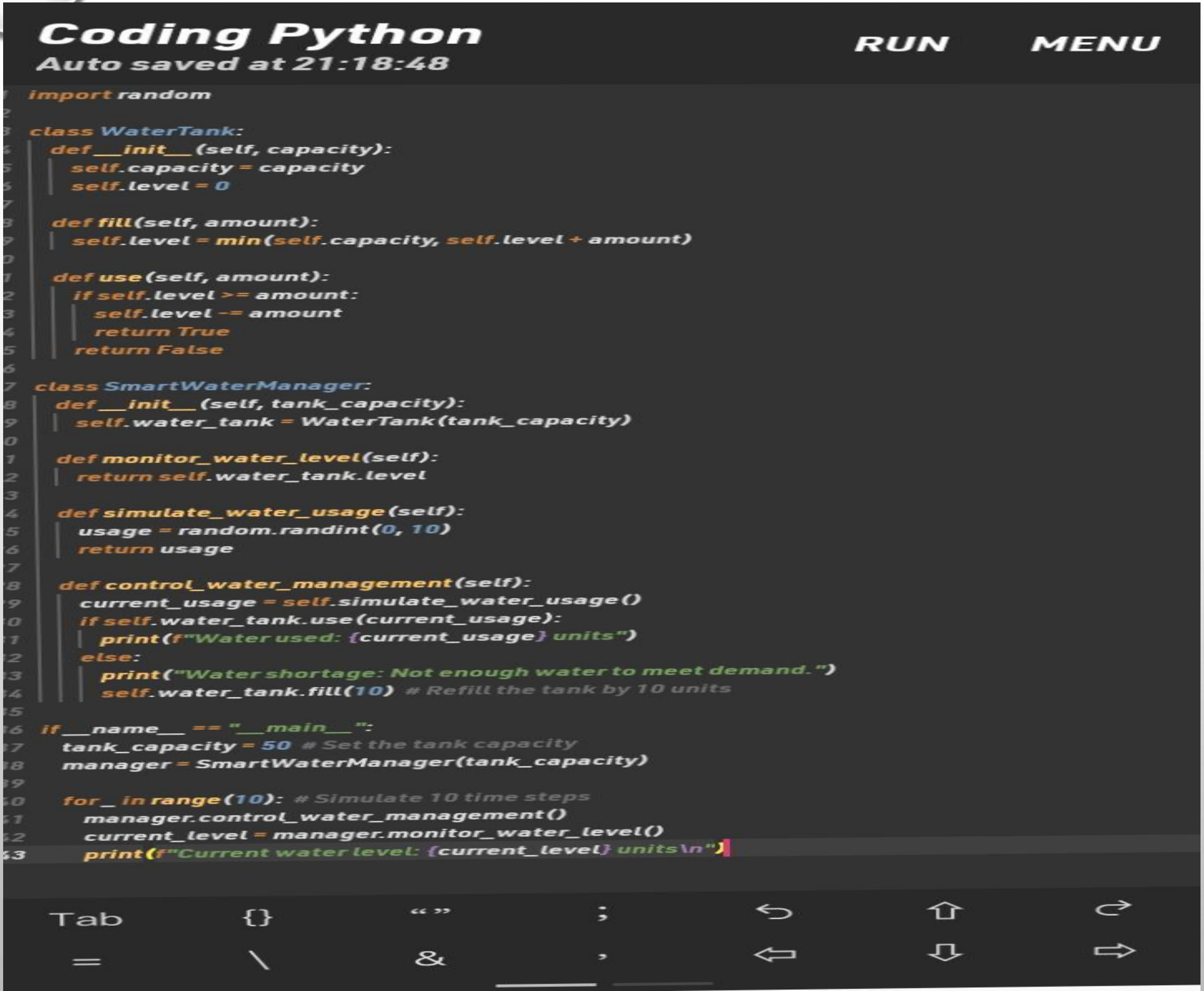
- THE FIRST WORK FOR THE ROBOT IS TO FETCH THE ATTRIBUTE SO THAT IT CAN FIND THE PLACE WHERE IT HAS TO GO. ONCE THE ROBOT FETCHES THAT VALUE THEN THE TCS3200 COLOR SENSOR COMES INTO THE PICTURE. BASICALLY IN THE DATASET, THERE ARE 03 VALUES FOR PLACE ATTRIBUTE WHICH ARE FROM THE STARTING POINT OF THE ROBOT TO THE FINAL DESTINATION I HAD LAID DOWN 03 DIFFERENT COLORED LINES SO THAT THE COLOR SENSOR WILL FOLLOW A PARTICULAR COLORED LINE AFTER GETTING THE PREDICTED VALUE. THE PLACE HAS THE FOLLOWING COLORS:
- RED – BEDROOM
- BLUE – BALCONY
- GREEN – HALL

- SO, THE ROBOT WILL FIND THE PATH THROUGH THE CUSTOM-MADE DIRECTION FUNCTIONS. WHEN THE ROBOT WILL REACH TO THE END THEN IT WILL WAIT FOR 20 SECONDS AND THEN IT WILL CHECK WHETHER THE USER CLICKED ON THE CAPACITIVE TOUCH SENSOR. IF YES, THEN THE NODEMCU WILL SEND AN API REQUEST TO THE BOLT WIFI CLOUD AND IN THIS MANNER, THE WATER MOTOR WILL BE SWITCHED OFF IN TIME
- THE SKETCH FOR THE HARDWARE CONNECTIONS FOR THE MOBILE ROBOT IS AS FOLLOWS:

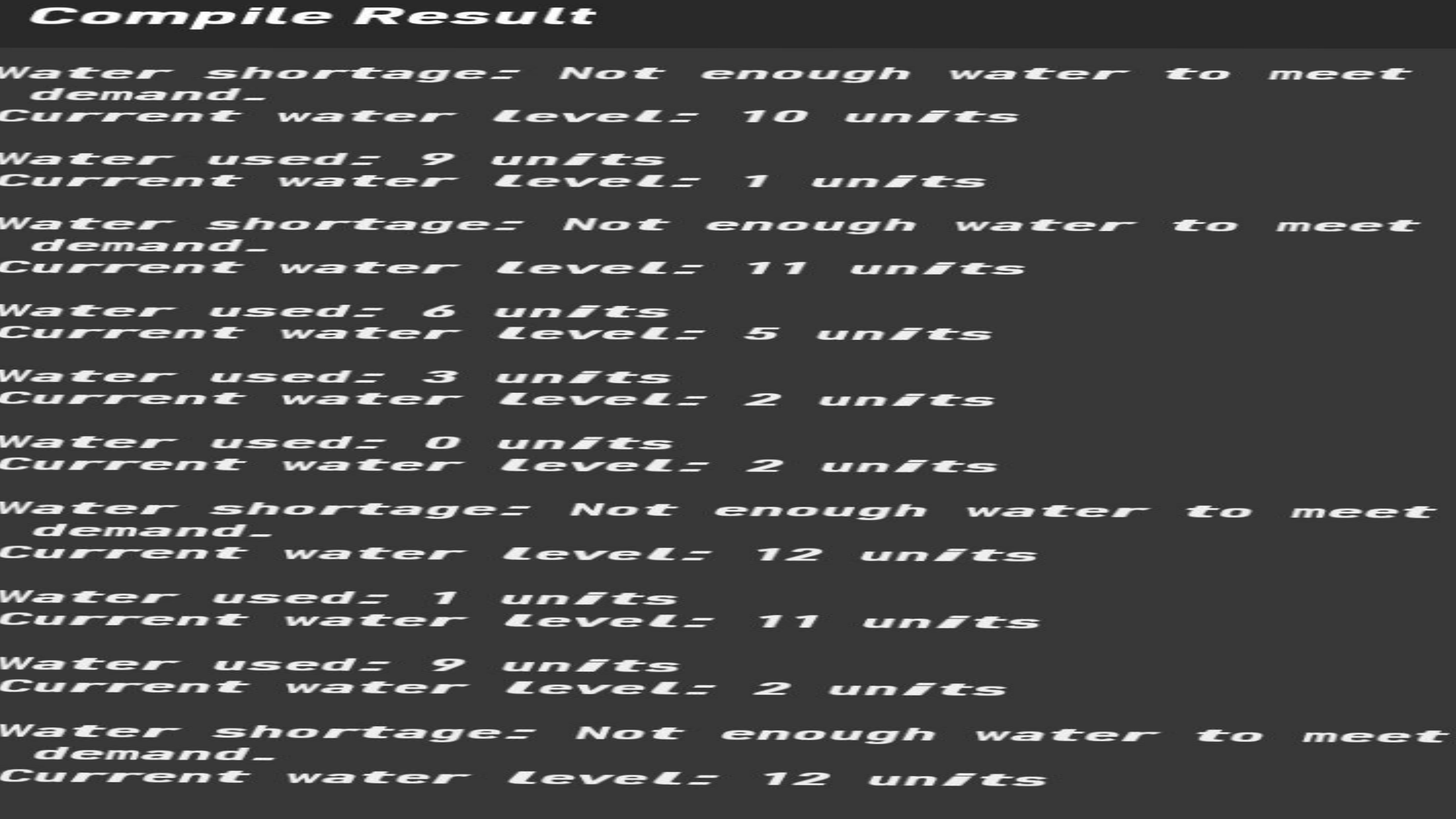

NodeMcu with
Motor Driver Shield
NodeMcu Based
Robot's Sketch
Left Motor
NodeMCU
V1.0
Touch Sensor
IR2
IR1
TCS 300
Color Sensor
CD4051
Right Motor
Piezo buzzer

PROGRAMMING CODE USING PYTHON

- SMART WATER MANAGEMENT IS A COMPLEX TASK THAT TYPICALLY INVOLVES DATA COLLECTION, ANALYSIS, AND CONTROL SYSTEMS. WHILE I CAN PROVIDE A SIMPLE EXAMPLE OF A PYTHON PROGRAM FOR WATER LEVEL MONITORING, IT'S IMPORTANT TO NOTE THAT REAL-WORLD SMART WATER MANAGEMENT SYSTEMS ARE MUCH MORE INVOLVED AND MAY REQUIRE THE USE OF SENSORS, DATABASES, AND MORE SOPHISTICATED CONTROL MECHANISMS. HERE'S A BASIC PYTHON PROGRAM TO SIMULATE WATER LEVEL MONITORING

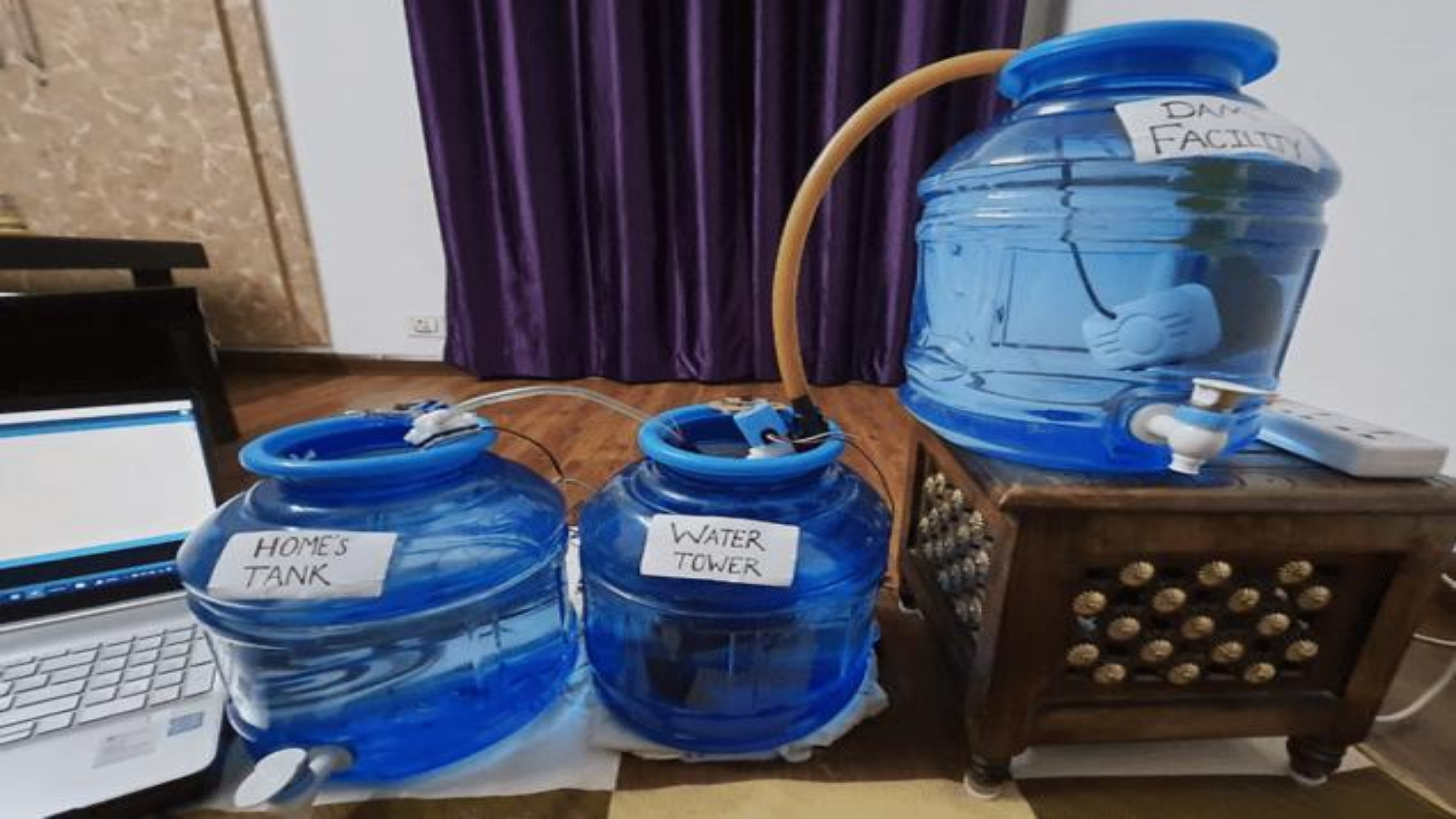


```python
import random

class WaterTank:
    def __init__(self, capacity):
        self.capacity = capacity
        self.level = 0

    def fill(self, amount):
        self.level = min(self.capacity, self.level + amount)

    def use(self, amount):
        if self.level >= amount:
            self.level -= amount
            return True
        return False

class SmartWaterManager:
    def __init__(self, tank_capacity):
        self.water_tank = WaterTank(tank_capacity)

    def monitor_water_level(self):
        return self.water_tank.level

    def simulate_water_usage(self):
        usage = random.randint(0, 10)
        return usage

    def control_water_management(self):
        current_usage = self.simulate_water_usage()
        if self.water_tank.use(current_usage):
            print(f"Water used: {current_usage} units")
        else:
            print("Water shortage: Not enough water to meet demand.")
            self.water_tank.fill(10)  # Refill the tank by 10 units

if __name__ == "__main__":
    tank_capacity = 50  # Set the tank capacity
    manager = SmartWaterManager(tank_capacity)

    for _ in range(10):  # Simulate 10 time steps
        manager.control_water_management()
        current_level = manager.monitor_water_level()
        print(f"Current water level: {current_level} units\n")
```

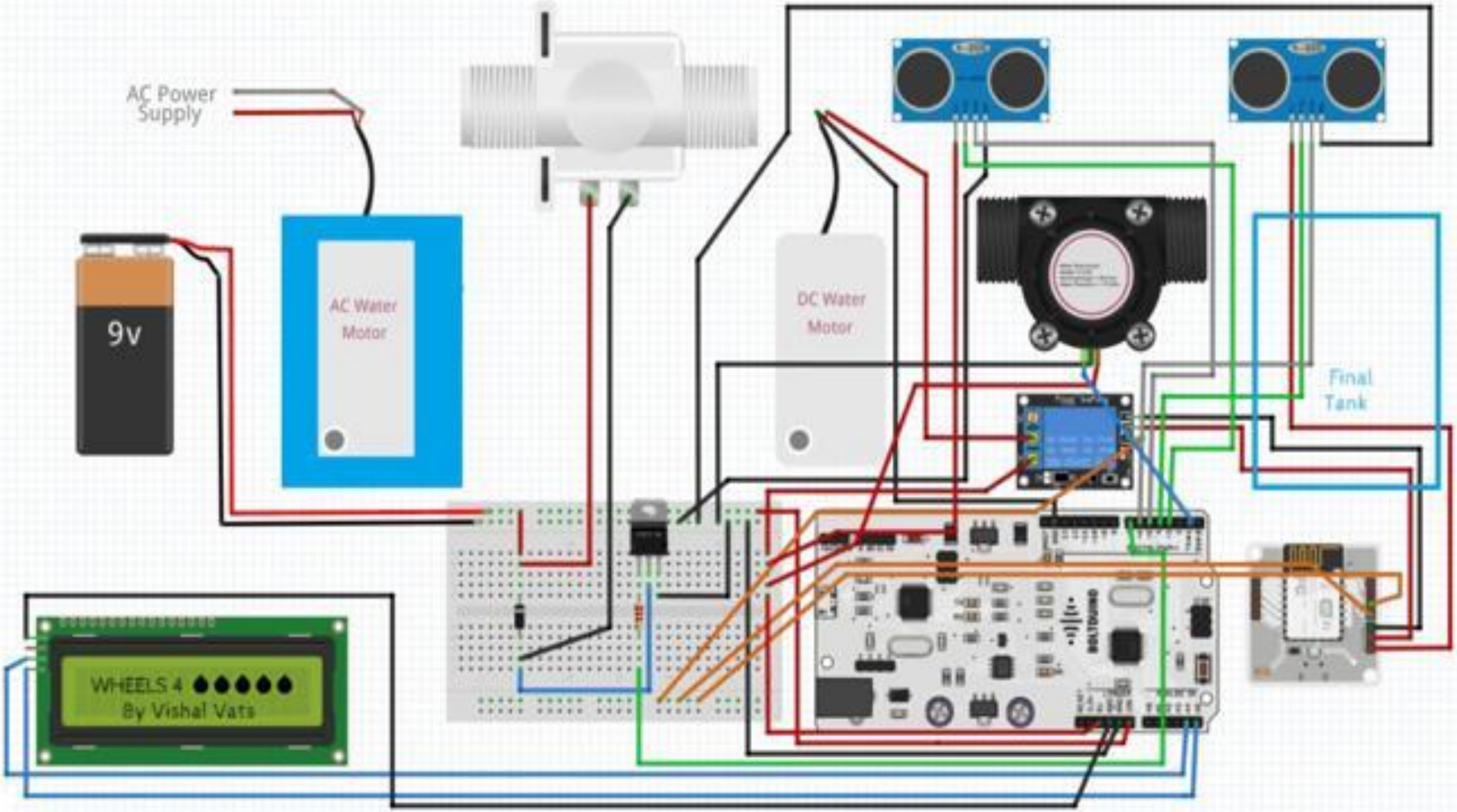


Compile Result

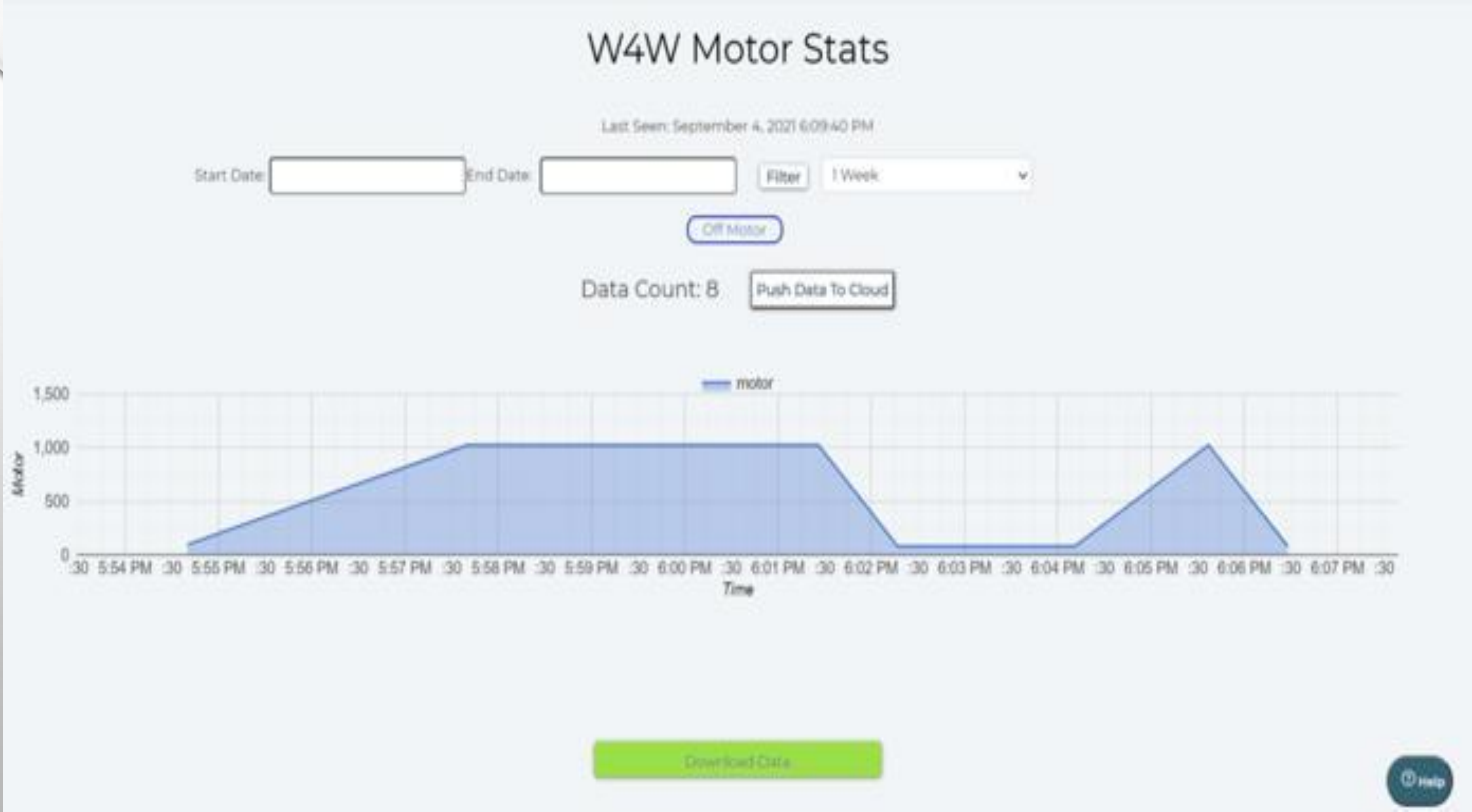
```
Water shortage: Not enough water to meet demand.
Current water level: 10 units

Water used: 9 units
Current water level: 1 units

Water shortage: Not enough water to meet demand.
Current water level: 11 units

Water used: 6 units
Current water level: 5 units

Water used: 3 units
Current water level: 2 units

Water used: 0 units
Current water level: 2 units

Water shortage: Not enough water to meet demand.
Current water level: 12 units

Water used: 1 units
Current water level: 11 units

Water used: 9 units
Current water level: 2 units

Water shortage: Not enough water to meet demand.
Current water level: 12 units
```


HOME'S TANK
WATER TOWER
DAM FACILITY

- JUST WANT TO TELL YOU THAT THE I2C LCD WILL BE UPDATED SEVERAL TIMES DURING THE COURSE OF THE PROJECT. BE IT UPDATING AND PRINTING THE PREDICTED VALUE OR DISPLAYING THE TOTAL TIME TO FILL, ONE CAN EASILY KEEP A TRACK OF IMPORTANT PARAMETERS THROUGH THE LCD.
- THERE COULD BE MANY POSSIBILITIES IN WHICH THE USER MAY NOT BE ABLE TO TURN THE MOTOR OFF. IN THOSE SCENARIOS, THE ARDUINO WILL USE THE TTF (TIME TO FILL) TO TURN OFF THE MOTOR, WHEN THE TTF IS ABOUT TO BE REACHED. SO, IN THIS WAY, THE PROJECT BECOMES QUITE INTERACTIVE AND THE USER ALSO GETS TIMELY UPDATES REGARDING THE WHOLE PROJECT.


AC Power Supply
9v
AC Water Motor
DC Water Motor
Final Tank
WHEELS 4 ●●●●●
By Vishal Vats
BOLTDUINO

- TO UNDERSTAND THE SOLUTION FOR THIS PROBLEM LET'S START FROM THE INITIAL STEP IE CREATING THE PRODUCT ON THE CLOUD. FOR MY PRODUCT, I OPTED INPUT DEVICE & GPIO COMMANDS DURING THE PRODUCT CREATION PHASE. SECONDLY, IN THE HARDWARE SECTION OF THE PRODUCT, I ASSIGNED PIN A0 THE NAME MOTOR. AS I HAVE USED A RELAY TO CONNECT THE BOLT WIFI MODULE AND ARDUINO SO I CONNECTED THE VIN PIN OF THE RELAY TO THE BREADBOARD AND FROM THAT BREADBOARD'S PIN I ALSO CONNECTED THE A0 PIN OF WIFI MODULE
- SO, IN THIS MANNER, THE AO PIN'S VALUE INCREASES FROM 0 AND THUS IN THE DASHBOARD, A GRAPH WAS FORMED WHICH WILL DROP DOWN TO 0 IF THE MOTOR IS OFF AND WHENEVER THE MOTOR IS ON THEN THE GRAPH CLIMBS UP.
- NOW AS I HAVE EXPLAINED EVERYTHING TO YOU I AM SHARING THE SCREENSHOT OF THE SAME SO THAT YOU CAN HAVE A BRIEF IDEA ABOUT WHAT I WAS TELLING.



W4W Motor Stats

Last Seen: September 4, 2021 6:09:40 PM

Start Date: End Date:

1 Week

Data Count: 8

Filter
Off Motor
Push Data To Cloud
motor
1,500
1,000
500
0
Motor
:30 5:54 PM :30 5:55 PM :30 5:56 PM :30 5:57 PM :30 5:58 PM :30 5:59 PM :30 6:00 PM :30 6:01 PM :30 6:02 PM :30 6:03 PM :30 6:04 PM :30 6:05 PM :30 6:06 PM :30 6:07 PM :30
Time
Download Data
Help

CONCLUSION

- 1. AS ALREADY MENTIONED I WILL BE CREATING 4 CUSTOM OPTIONS FOR THE PROJECT AND I DID THAT. I HAD CREATED THE WEBSITE, INCORPORATED THE CAPACITIVE TOUCH SENSOR, AUTOMATED THE VOICE CONTROL THING, AND CREATED THE CLOUD DASHBOARD.
- 2. THIS IS A PROJECT WHICH CAN BE DEPLOYED IN THE MARKET AS IT IS ALMOST OPTIMIZED. IT IS ALSO COST-EFFICIENT AS THE WHOLE HARDWARE SETUP COSTS AROUND $130. AS IN THE REAL WORLD, THE MAGNITUDE OF THE SENSORS WILL BE INCREASED, SO THAT THE COST WILL ONLY REACH UP TO $300 (WHICH IS A QUITE REMARKABLE FEAT).
- THEREFORE, I MAGNIFICENTLY SOLVED AN OLD-AGE PROBLEM OF WATER MANAGEMENT BY INCORPORATING ABOUT 30 ELECTRICAL COMPONENTS TOGETHER AND MAKING A PROJECT WHICH CAN BE EITHER USED BY A SINGLE PERSON OR BY THE ENTIRE COMMUNITY.

Thank
You!