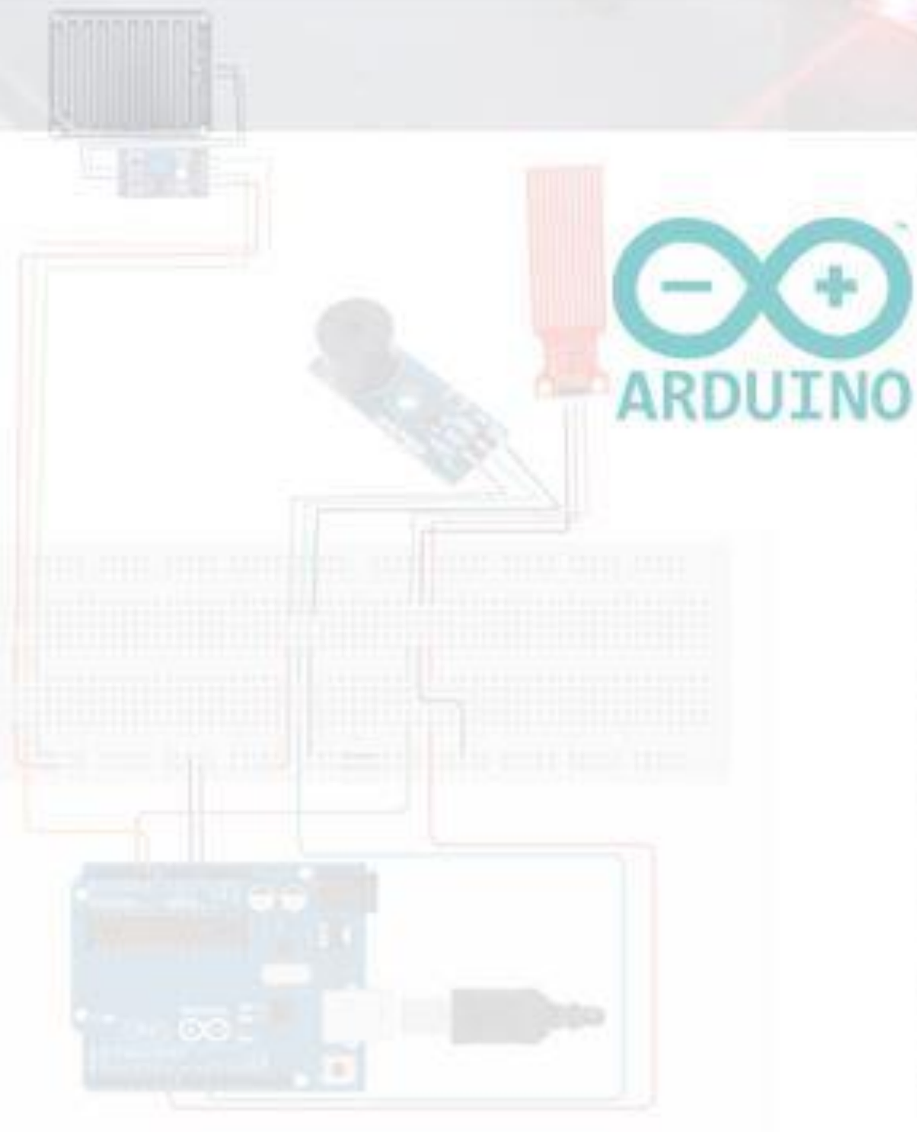
SIT 123 – DATA CAPTURE TECHNOLOGIES

FINAL PROJECT REPORT

RAIN DETECTOR AND EXCESSIVE RAIN ALARM



REPORTED BY:

CHIRAKSHI GUPTA

221397163

INTRODUCTION:

This project emphasizes on making people adapt to preventive and sustainable activities when it is raining. Experts have been predicting rainfall for a long time and advancing this field of utilising rainwater in efficient manners, but there are people who eventually had to suffer huge losses in terms of health as well as wealth when it comes to damages caused by heavy rains. Excessive rains lead to over-irrigation of farmlands, floods, slippery roads, soil erosions, etc. and many other disasters which can be avoided with suitable actions, only if one can know that rainfall has exceeded a certain amount and it's time to take some measures. Furthermore, sudden, and unpredicted rains at times when we are unprepared and unaware of it happening can cause many damages from daily chores like wetting our laundry drying in the open space, ruining outdoor plans, and unnecessarily excessive watering our plant pots to major ones including road accidents, agricultural losses, etc. The project focuses on the fact that people can prevent losses if they have the relevant information on time. The government informs people only when conditions get worse. So rather than looking for data for amount of rainfall from outside sources, why not collect our own data and make its best use by ourselves.

HYPOTHESIS: If this precipitation measuring system is brought into knowledge of common people at a vast level, then it can consequently reduce the problems faced due to excessive levels of rainfall and damages caused due to it.

RESEARCH QUESTIONS: This report answers following questions-

1. How can a sensor-based system detect rainfall?
2. How people can be alerted when there is unusually high rain?
3. How the system can be made so that it is open to all the developments?
4. What tests can be performed to improve the system's efficiency?
5. What data should be collected and how it should be used to improve the working of the product?

PROJECT DESCRIPTION:

"Rain detector and excess rain alarm" is a budget-friendly system that emphasizes on inexpensive and efficient hardware and software for productive results. Sensors have been chosen by me, keeping in mind that the sole purpose of this project is at lower scale like household levels and not some big industries, factories, or department-uses. However, the idea holds the capability of being extended to a bigger level as well.

HARDWARE:

The product is built up using following hardware:

1. **Arduino UNO Micro-controller Board**

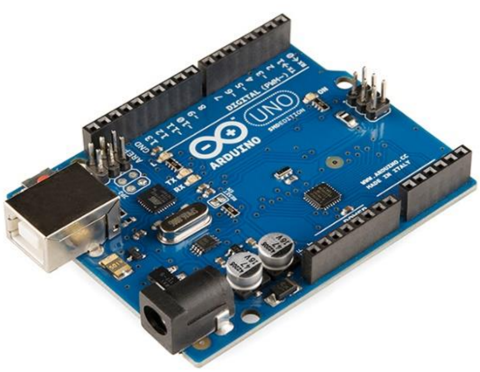


Figure 1 Arduino UNO Board
Source: Online

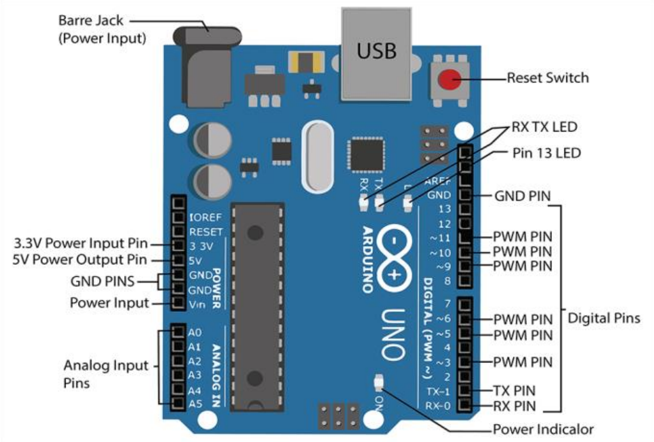


Figure 2 Pin Layout of Arduino Micro-controller Board
Source: Online

Arduino UNO is a micro-controller board, suitable for taking inputs, processing them, and sending them as outputs. It uses an interface called Arduino IDE that lets us write a sketch, basically a code program, which is uploaded on the board and the outputs can be read on serial monitor. These outputs can be passed to various devices or humans through technology. The pin layout of the board is given on right in detail.

2. **Active Buzzer Sensor Module**

Figure 3 Active Buzzer Module
Source: Online

The Active Buzzer module has three pins as can be seen in the figure. The I/O pin is used to send input to it in so that it can buzz. Like other sensors and equipments used, it has a ground pin as well as a VCC pin to connect it to Arduino Board.

3. **Rainwater Sensor Module**

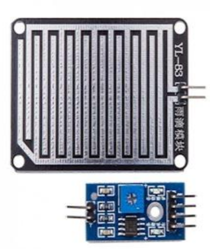


Figure 4 Rain Sensor Module
Source: Online

Rainwater Sensor Module is used to test if there is rain. It basically checks if water is present on its surface, so it must be exposed to rain and at an angle between 30 and 45 degrees for best results.

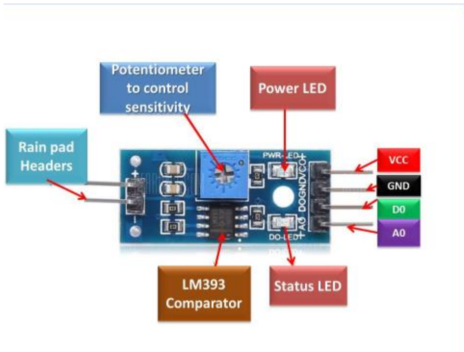


Figure 5 Pin layout for rain sensor
Source: Online

4. Water level Sensor

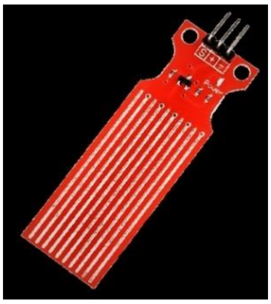


Figure 6 Water Level Sensor
Source: Online

It is used to check the level of water in a container. It is best for this project due to its cost and efficient use at low scale. It reads a value when dipped in water.

Figure 7 Pin layout of water level sensor
Source: Online

Alternate: **Ultrasonic sensor**

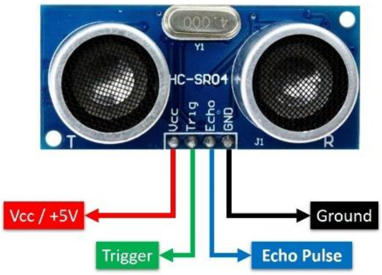


Figure 8 Pin layout for Ultrasonic sensor
Source: Online

Water level sensor has its own limitations, and it would be better to use an ultrasonic sensor when extending this project to higher scale. It can be set at the top of rain gauge to check the water level in it.

Figure 9 Ultrasonic Sensor
Source: Online

5. Breadboard

Thin plastic board used to put together all the electrnic components while building a system or product. These boards are re-usable and quite efficient in their working. I used it to connect multiple sensors to arduino and also the buzzer.

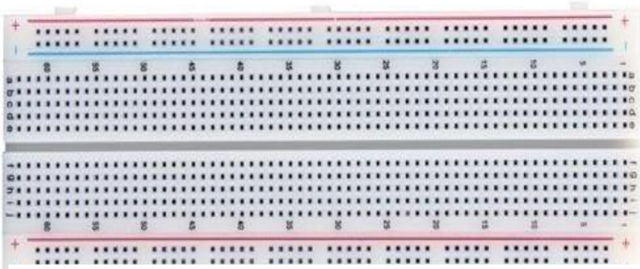


Figure 10 Breadboard - 800 pins
Source: Online

6. **Jumper wires**:

   i. Male to Male

   Male to Male wires have both the ends same with a pin pointing out that can be inserted into a slot on Arduino, breadboard or any other equipment. I have used them in my project to make connections on my breadboard.

Figure 11 Male to Male Jumper wires

ii. Male to Female

These are the most suitable wires when we have to connect sensors to the Arduino Board whether it is a direct connection or through a breadboard. The two ends are different as shown in the picture and fulfil their purpose when required to.

Figure 12 Male to Female Jumper wires
Source: Online

iii. Female to Female

These type of wires have both ends that can insert pins and are a requirement in many projects. I have used them to connect the rain sensor pad to its chip within the rain sensor module.

Figure 13 Female to Female Jumper wires
Source: Online

7. **Power Supply:**

I have used my laptop computer as a source of power, but external power can always be supplied to Arduino when it comes to practically implementing and applying any project idea. The power input pin, also called barre jack, is used to fulfil this purpose.

8. **Rain Gauge**

Figure 14 Professional Rain Gauge
Source: Online

Rain gauge is an instrument which can measure the amount of rainfall. It collects water over an area of 1 square metre into a cylindrical container which has markings on it in millimetres. The depth of water shows the amount of precipitation. When it comes to highest accuracy and big budgets, these are preferred for this system along with ultrasonic sensors, but that would make the product high in costs. Therefore, there is an alternate instrument which is widely used at low scale for measuring rainfall amount. It is shown in picture on right. It is a simple glass tube with scale marked on it to take measurements of rainwater.

Figure 15 Alternate for Rain Gauge
Source: Online

SOFTWARE:

The software used for this project is Arduino IDE, which can not only upload a sketch to the Arduino Board, informing it how to run, but also receive input and use it to show the information or perform some action, like in this case, make the buzzer beep when required. All the code to make our system work according to our needs can be written in form of a sketch in this software by including necessary libraries and using numerous other features. I have used C++ programming language to write the code.

Figure 16 Arduino Logo
Source: Online

ORIGINAL SETUP:

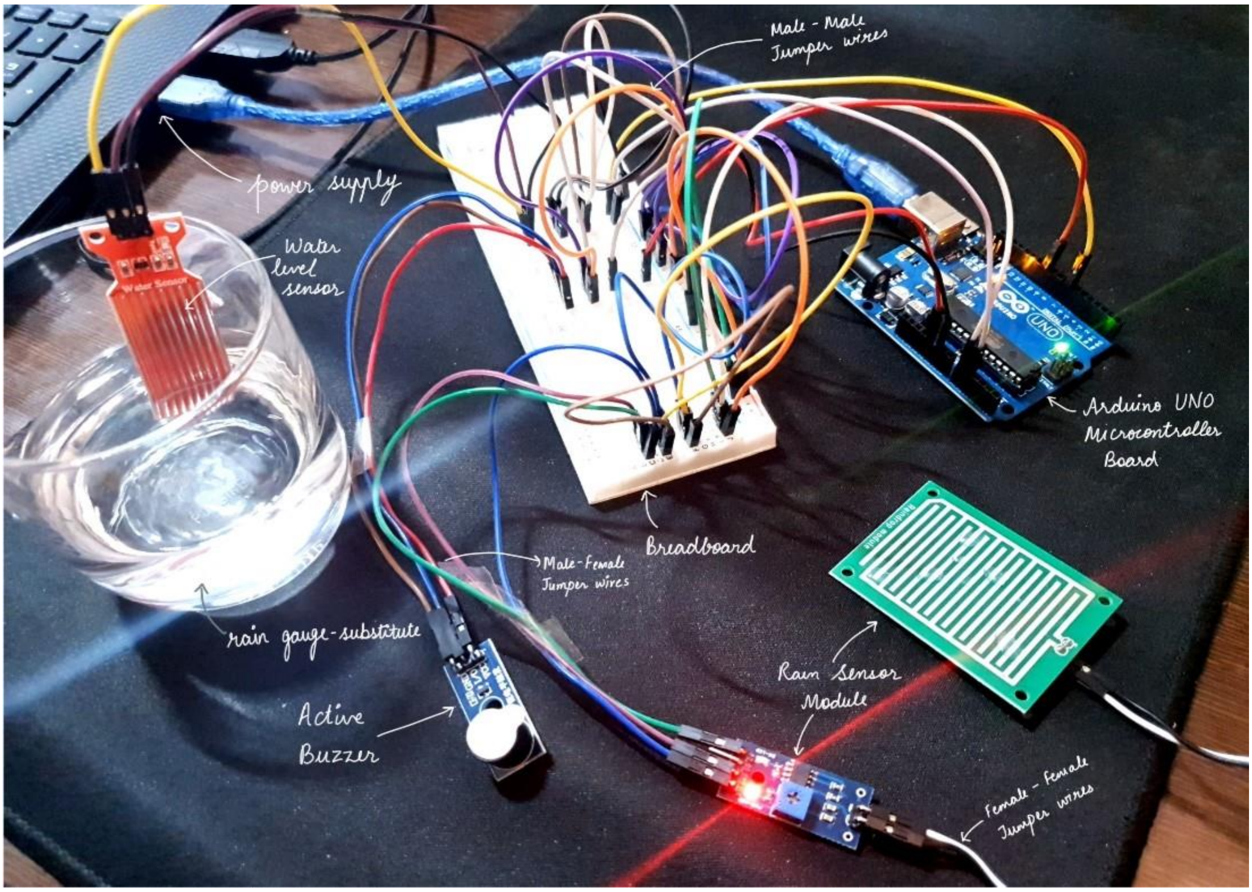


Figure 17 Original setup made *Source: Self*

ALTERNATE SETUP:

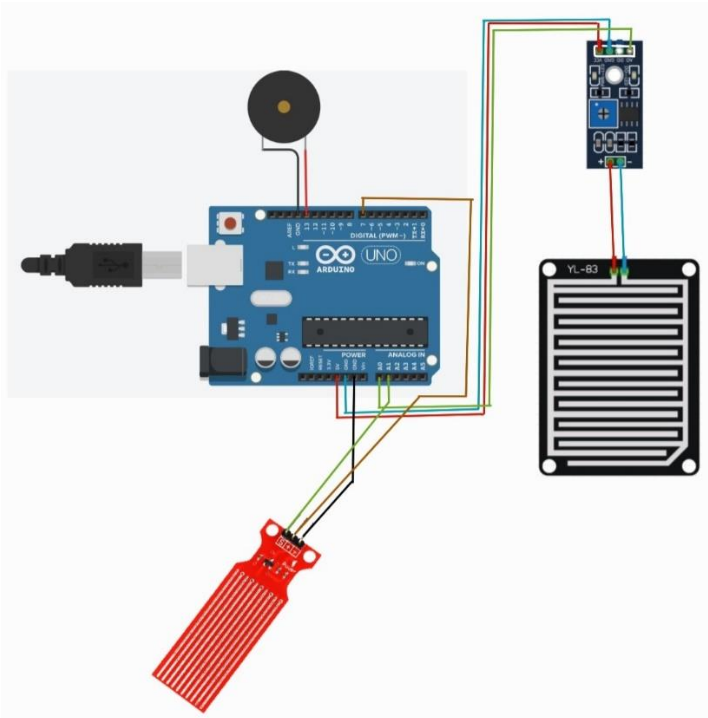


Figure 18 Alternate Setup with connections *Source: Self*

There are two setups possible for this system:
1. Setup 1 – Figure 1 – My product
2. Alternate Setup – Figure 2 –Omitted

REASONS:
Although, the setup 2 is cheaper than 1, it cannot be extended in the same cost and will need a lot of modifications in design if any additions are to be made. Also, the cost for any modifications will be comparatively high in setup 2 because of changes that will have to be made in the existing model. Moreover, as my device is open to all the modifications, it must be able to include any developments and get along with upcoming technologies, therefore, first setup is better. Its design seems a little complicated, but it is actually more efficient and suitable for small-scale project with best outcomes.
To go briefly over second setup, it uses an Arduino directly connected with two sensors and piezo buzzer but will not welcome any other sensor additions without code complexities or a breadboard.

CONNECTIONS:

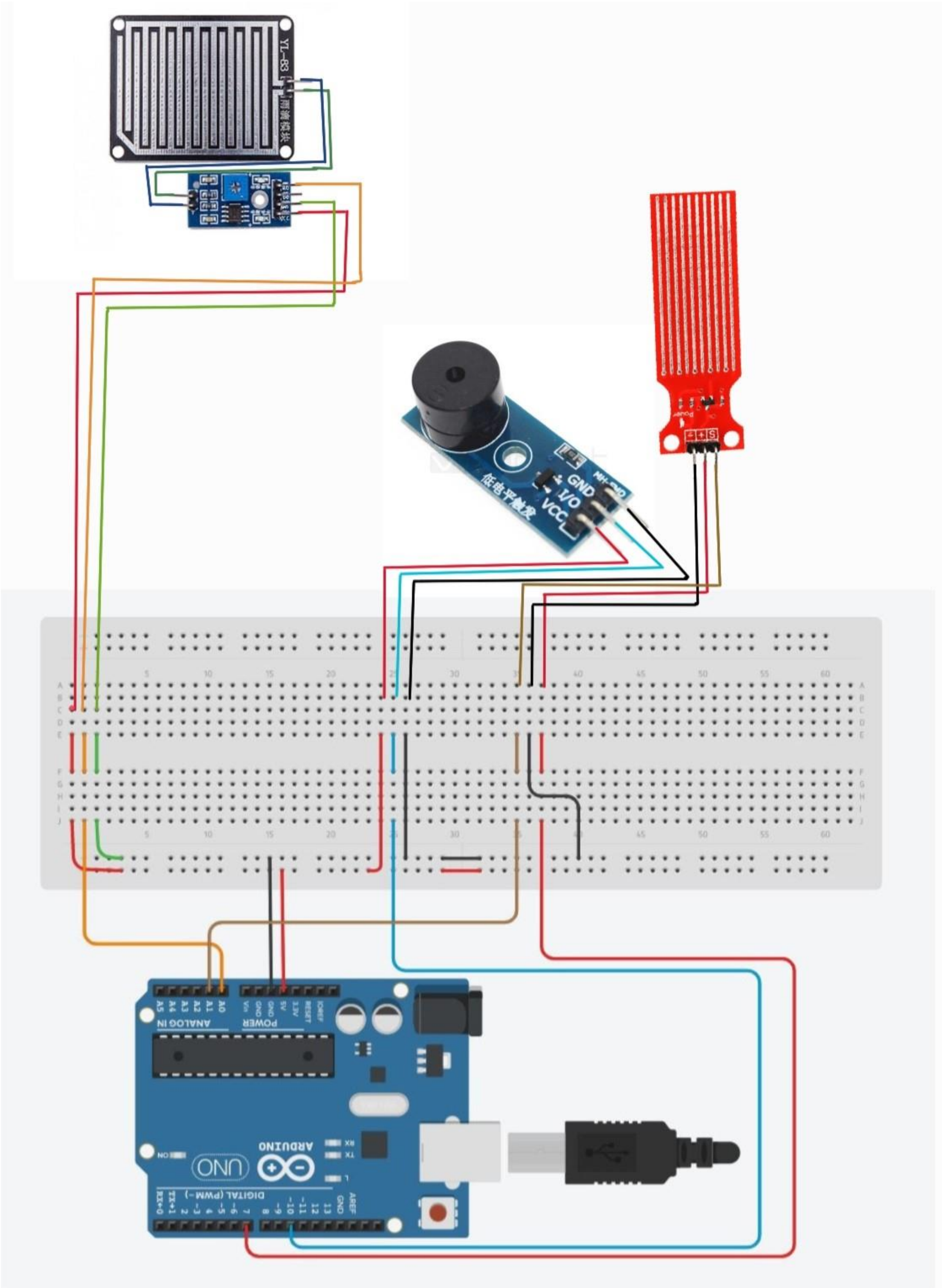


Figure 19 Connections in original setup
Source: Self

UNDERSTANDING IN TERMS OF SENSE-THINK-ACT PARADIGM

Considering in terms of Sense-Think-Act, the requirements and functioning of the setup is as follows:

- Sensing Requirement: There are two different sensors that are collecting data in this case and include Rain sensor and water level sensor. They capture data from surroundings and fulfil the sensing requirement of the system as it has to sense data from its environment.
- Thinking Requirement: The sketch uploaded to the Arduino micro-controller controls how the system will respond. Receiving inputs from the sensors and sending output to the buzzer is all a part of thinking done by the system. All these connections run through the Arduino.
- Acting Requirements: The beep of buzzer when rain sensor detects rain or three beeps when water level sensor finds amount of rainfall exceeding are all results of acts performed by system. It is done through the piezo active buzzer.

WORKING:

When the drops of rainwater fall on the surface of the rain sensor surface, it sends input to the Arduino through its chip. The Arduino checks the value and makes the alarm beep by sending it a signal, if the value is under a certain value, indicating that rain has started because there is some water on the sensor. If there is no water or just moisture, no alarm goes off and it keeps on checking the value continuously. If the first beep is there, then the system will check for the amount of water collected. In case excess of water is collected in rain gauge, implying that the amount of rainfall has exceeded a certain level, then the alarm starts beeping thrice until the user cuts off the power - supply.

For code, refer attached code file or text file.

DATA COLLECTION PROCESS AND TESTING:

This project is a trigger-based system that does not need a continuous recorded data for analysis or actions to take place, rather it works on the current environment and conditions. The alarm beeps once when the rain starts and thrice when the amount of rainfall exceeds a particular limit set by the user until the user is alerted and the power supply is cut-off.

Serial Monitor readings (Rainwater)

```
08:41:58.392 -> Rain Sensor Reading: 381
08:41:58.392 -> Water level: 0
08:41:58.771 -> It's Raining!
08:41:58.817 -> Enjoy the rain.
08:41:59.799 ->
08:41:59.799 -> Rain Sensor Reading: 378
08:41:59.799 -> Water level: 0
08:42:00.176 -> It's Raining!
08:42:00.223 -> Enjoy the rain.
08:42:01.209 ->
08:42:01.209 -> Rain Sensor Reading: 374
08:42:01.209 -> Water level: 286
08:42:01.589 -> It's Raining!
08:42:01.635 -> Enjoy the rain.
08:42:02.625 ->
08:42:02.625 -> Rain Sensor Reading: 371
08:42:02.625 -> Water level: 311
08:42:03.003 -> It's Raining!
08:42:03.049 -> Enjoy the rain.
08:42:04.033 ->
08:42:04.033 -> Rain Sensor Reading: 368
08:42:04.033 -> Water level: 413
08:42:04.411 -> It's Raining!
08:42:04.458 -> Enjoy the rain.
08:42:05.444 ->
08:42:05.444 -> Rain Sensor Reading: 366
08:42:05.444 -> Water level: 514
08:42:05.820 -> It's Raining!
08:42:06.624 -> Excessive Rain!
08:42:07.660 ->
08:42:07.660 -> Rain Sensor Reading: 363
08:42:07.660 -> Water level: 487
08:42:08.036 -> It's Raining!
08:42:08.835 -> Excessive Rain!
08:42:09.863 ->
08:42:09.863 -> Rain Sensor Reading: 363
08:42:09.863 -> Water level: 473
08:42:10.286 -> It's Raining!
```

```
08:41:46.918 -> Rain Sensor Reading: 1023
08:41:46.964 -> Water level: 0
08:41:46.964 -> It's not raining.
08:41:51.946 ->
08:41:51.946 -> Rain Sensor Reading: 1022
08:41:51.993 -> Water level: 1
08:41:51.993 -> It's not raining.
08:41:56.977 ->
08:41:56.977 -> Rain Sensor Reading: 386
08:41:56.977 -> Water level: 0
08:41:57.355 -> It's Raining!
08:41:57.403 -> Enjoy the rain.
08:41:58.392 ->
08:41:58.392 -> Rain Sensor Reading: 381
08:41:58.392 -> Water level: 0
08:41:58.771 -> It's Raining!
08:41:58.817 -> Enjoy the rain.
08:41:59.799 ->
08:41:59.799 -> Rain Sensor Reading: 378
08:41:59.799 -> Water level: 0
08:42:00.176 -> It's Raining!
08:42:00.223 -> Enjoy the rain.
08:42:01.209 ->
08:42:01.209 -> Rain Sensor Reading: 374
08:42:01.209 -> Water level: 286
08:42:01.589 -> It's Raining!
08:42:01.635 -> Enjoy the rain.
08:42:02.625 ->
08:42:02.625 -> Rain Sensor Reading: 371
08:42:02.625 -> Water level: 311
08:42:03.003 -> It's Raining!
08:42:03.049 -> Enjoy the rain.
08:42:04.033 ->
08:42:04.033 -> Rain Sensor Reading: 368
08:42:04.033 -> Water level: 413
08:42:04.411 -> It's Raining!
08:42:04.458 -> Enjoy the rain.
08:42:05.444 ->
```

Figure 20 Serial Monitor Readings *Source: Self*

To collect this type of data, it would have been best to test rainwater at different places, but as that was not possible under current circumstances, so I took following measures to collect data for my product's testing:

1. Chose water from three different sources.
2. Took readings by water level sensor when the height of water is 1cm for each source to improve my sketch.
3. Took readings by rain sensor for each source. That gave me the conclusion that for different regions, my product will have to be modified depending on quality of rainwater.
4. The most important part of collecting data was actually testing the system and checking manually if it behaves in a desired way. For this, I put drops of water on my rain sensor surface and checked if the alarm went off. To make it work properly I had to put data collected by the sensor on the serial monitor, so I could analyse that the system beeps when it is supposed to. I dipped water level sensor in collected rainwater to check that it beeps after a certain level. And then put it altogether to test working of complete system.

The following data was collected by me for the test:

1. Different Sources of Water
2. Value read by each sensor for each source
3. True Positive and True Negative Rates

Data has been put into tables with use of 'color' as the visual variable, for efficient analysis.

Data Tables

Four types of test results are:

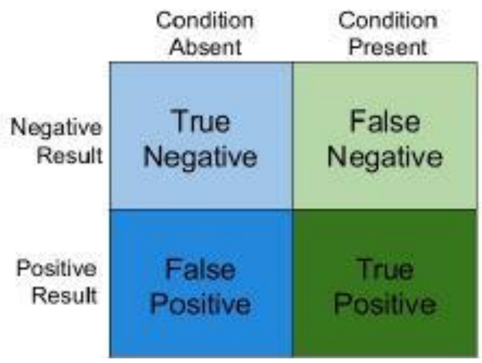


Figure 21 Test Results
Source: Deakin

- *TP (True Positive) - Detects the condition when it is present.*
- *FN (False Negative) - Does not detect the condition when it is present.*
- *TN (True Negative) - Does not detect the condition when it is absent.*
- *FP (False Positive) - Detects the condition when it is absent.*

Table for Sensor readings in particular conditions for different sources			
Water Source	Rain Water Sensor Reading	Water Level Sensor Reading	
Purified Drinking Water	500 - 600	330 - 345	
Rainwater	570 - 750	295 -305	Chosen for this experiment
Running Tap water	350 - 450	335 - 345	

Table 1 Sensor Readings in particular condition for different sources
Source: Self

Table for testing Rain sensor's accuracy - TPR (True Positive Rate)													
Water Presence on Sensor	Reading 1	Reading 2	Reading 3	Reading 4	Reading 5	Reading 6	Reading 7	Reading 8	Reading 9	Reading 10	Sum of TP	Total readings	TPR
YES	1	1	1	1	1	1	1	1	1	1	9	10	10 / 0+10 = 1 = 100%

***Note that:*

Reading is 1 if serial monitor reading changes to less than 750 and buzzer beeps, otherwise it is noted as zero
TPR is calculated for Rain sensor
Formula used: TPR = TP / TP + FN
TP stands for True Positive, i.e., when reading is 1
FN stands for False Negative, i.e., when reading is zero

Table 2 Testing sensor's accuracy – TPR for rain sensor
Source: Self

Table for testing Rain sensor's accuracy - TNR (True Negative Rate)													
Water Presence on Sensor	Reading 1	Reading 2	Reading 3	Reading 4	Reading 5	Reading 6	Reading 7	Reading 8	Reading 9	Reading 10	Sum of FN	Total readings	TNR
NO	0	0	0	0	0	0	0	0	0	0	10	10	10 / 0+10 = 1 = 100%

***Note that:*

Reading is 1 if serial monitor reading changes to less than 750 and buzzer beeps, otherwise it is noted as zero
TNR is calculated for Rain sensor
Formula used: TNR = TN / TN + FP
TN stands for True Negative, i.e., when reading is 0
FP stands for False Positive, i.e., when reading is 1

Table 3 Testing sensor's accuracy – TNR for rain sensor
Source: Self

Table for testing Water Level sensor's accuracy - TPR (True Positive Rate)													
Sensor in required range	Reading 1	Reading 2	Reading 3	Reading 4	Reading 5	Reading 6	Reading 7	Reading 8	Reading 9	Reading 10	Sum of TP	Total readings	TPR
YES	1	1	1	1	1	1	1	1	1	1	9	10	10 / 0+10 = 1 = 100%

***Note that:*

Reading is 1 if serial monitor reading changes to 350 or more and buzzer beeps, otherwise it is noted as zero
TPR is calculated for Water Level sensor
Formula used: TPR = TP / TP + FN
TP stands for True Positive, i.e., when reading is 1
FN stands for False Negative, i.e., when reading is zero

Table 4 Testing sensor's accuracy – TPR for water level sensor
Source: Self

Table for testing Water Level sensor's accuracy - TNR (True Negative Rate)													
Sensor in required range	Reading 1	Reading 2	Reading 3	Reading 4	Reading 5	Reading 6	Reading 7	Reading 8	Reading 9	Reading 10	Sum of FN	Total readings	TNR
NO	0	0	0	0	0	0	0	0	0	1	10	10	9 / 1+9 = 0.9 = 90%

***Note that:*

Reading is 1 if serial monitor reading changes to 350 or more and buzzer beeps, otherwise it is noted as zero
TPR is calculated for Water Level sensor
Formula used: TNR = TN / TN + FP
TN stands for True Negative, i.e., when reading is 0
FP stands for False Positive, i.e., when reading is 1

Table 5 Testing sensor's accuracy – TNR for water level sensor
Source: Self

Also refer to the attached excel file.

ETHICAL CONCERNS

Ethics are the rules or principles which cast light on what practices are considered right or wrong, depending on if it is an individual, a group of people or an organisation that involves third-party as well. There is a code of conduct that explicitly mentions all the rules and values to be followed and may lead to strict actions if violated. One of the ethics is plagiarism – copying someone's work without consent, which is basically stealing and against correct practice of producing original work. Privacy and ownership are other major concerns related to data, giving a person access to all the collected information. Overall, ethics are all about being fair to another person, especially when it comes to collecting their personal or daily information. However, this project does not involve any data collection from people and is just a trigger-based system, so, it is not risked by any major ethical concern.

RESULTS AND CONCLUSIONS:

The following interpretations can be made from the data collected for the project:

1) Accuracy of working of sensors.
2) Standard Readings for rainwater.
3) Fact that quality of water will affect the system.

This table shows final results, refer attached excel file for more.

Results and Conclusions				
Water Source	Sensor Name	Original sensor readings (for chosen standard)	TPR	TNR
Rainwater	Rain Water Sensor	570 - 750	100%	100%
Rainwater	Water Level Sensor	295 -305	100%	90%

Table 6 Final table with interpreted outcomes
Source: Self

The above data shows that the system is very reliable and fulfils its purpose with accuracy, precision, and efficiency.

CHALLENGES AND LEARNINGS:

I faced following challenges in completion of this project:

1. It was hard to understand how to use multiple sensors with one Arduino board and then a breadboard to implement my idea, but I eventually understood it.
2. The buzzer initially produced a continuous sound, so I had to figure out how to make it beep. It made me realise that a single piece of equipment can be used in multiple ways.
3. Water level sensor used in the project is budget-friendly but gets easily malfunctioned if used continuously for a long period. So, I had to write a piece of code that would

turn the sensor on and off as per the requirement. This taught me that projects need to inculcate the needs of its equipment to give best outputs.

4. I had to manually test the working of the system, along with accuracy of each sensor till I received desired results.

PROJECT EXTENSION IDEAS:

The best thing about this project is that it has widest possibilities if adopted. Instead of making it buzz an alarm, we can link it to one of the advanced technologies, such as:

1. Smart Homes: They can take appropriate measures like collecting laundry, closing doors, checking drain systems, checking plants' moisture levels, etc. when there are signs of rain.
2. Flood Monitor: In areas where floods are often, one can know when it has reached a certain level that can be harmful and take precautions before the government declares emergency. Even in areas where rainfall is moderate, the system will alert the user if rain exceeds the usual amount, giving time for preparations.
3. Rainwater Conservation: One of the best features of the device that it can be used at larger scale as the product can be set to activate any system related to rainwater preservation as soon as it detects rain.
4. Weather detection: This system can be included in a weather station, where it can not only detect rain but also deduce conclusions by analysing the data for amount of rainfall, which has wide range of applications in the industry.

REFERENCES:

1. Image Sources:
    - Arduino UNO
        - https://en.wikipedia.org/wiki/Arduino
        - https://learn.thestempedia.com/courses/introductory-course-on-arduino/lessons/getting-started-5/topic/different-parts-of-arduino-board/
    - Active Buzzer Sensor Module
        - https://veerobot.com/active-buzzer-module-33v-5v
    - Rainwater Sensor Module
        - https://microcontrollerslab.com/raindrop-sensor-arduino-detector/
    - Water level Sensor
        - https://www.roboelements.com/product/highly-sensitive-water-level-moisture-water-droplet-sensor-for-arduino/
        - https://www.circuituncle.com/product/water-liquid-level-sensor-india/
    - Ultrasonic Sensor
        - https://www.hackster.io/SiSi/intro-to-ultrasonic-sensor-0c88fe
        - https://microcontrollerslab.com/hc-sr04-ultrasonic-sensor-interfacing-with-arduino-distance-measurement-example/
    - Breadboard
        - https://www.amazon.in/REES52-Solderless-Breadboard-Points-MB102/dp/B01IN2TSAO/ref=sr_1_fkmr4_2?dchild=1&keywords=Robotbanao+Solderless+MB102+Breadboard+830+Tie+Points&qid=1632452497&s=industrial&sr=1-2-fkmr4
    - Rain Gauge
        - https://www.walmart.com/ip/Rain-and-Snow-Gauge-Stratus-RG202-Long-Term-Professional-Gauge/163253590
        - https://www.ironaccents.com/products/rain-gauge-replacement-glass
    - Jumper Wires
        - https://www.elementzonline.com/male-to-male-jumper-wire-132
        - https://robocraze.com/products/f2m-jumper-wires-20cm-20pcs
2. Buzzer beep
    - https://www.youtube.com/watch?v=D2bhEGfZYCQ
3. Multiple Sensor Use
    - https://www.etechnophiles.com/how-to-use-multiple-sensors-with-arduino-at-once/
    - https://www.youtube.com/watch?v=fsy5X1b6WuY
4. Code Help
    - https://create.arduino.cc/projecthub/Shubham_Desai/rain-detector-alarm-using-arduino-uno-e819e6
    - https://lastminuteengineers.com/water-level-sensor-arduino-tutorial/#:~:text=In%20the%20loop%20section%2C%20we,))%3B%20delay(1000)%3B