Automatic Flood Detection System

by

Examination Roll: 193213

A Project Report submitted to the
Institute of Information Technology
in partial fulfillment of the requirements for the degree of
Professional Masters in Information Technology

Supervisor: Dr. Rashed Mazumder, Assistant Professor

Institute of Information Technology
Jahangirnagar University
Savar, Dhaka-1342

January 2023

DECLARATION

I am announcing that this project report is based on the results found by one and only myself. Resources for this report found by other researchers are mentioned herewith by reference. This project has not been submitted before for any course or degree partially or fully.

Roll: 193213

CERTIFICATE

The project titled "Automatic Flood Detection System" submitted by Sourav Das, ID: 193213, Session: Fall 2019, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Professional Masters in Information Technology on January 2023.

Dr. Rashed Mazumder
Supervisor

BOARD OF EXAMINERS

Dr. Mohammad Abu Yousuf Professor, IIT, JU	Coordinator PMIT Coordination Committee
Dr. M. Shamim Kaiser Professor, IIT, JU	Member, PMIT Coordination Committee & Director, IIT
Dr. M. Mesbahuddin Sarker Professor, IIT, JU	Member PMIT Coordination Committee
Dr. Mohammad Shahidul Islam Associate Professor, IIT, JU	Member PMIT Coordination Committee
Dr. Md. Sazzadur Rahman Associate Professor, IIT, JU	Member PMIT Coordination Committee

ACKNOWLEDGEMENTS

I'm grateful for the chance to convey my sincere gratitude and thanks to everyone who contributed to the creation of this report. This project is performed under the supervision of Dr. Rashed Mazumder, Assistant Professor, Institute of Information Technology (IIT), Jahangirnagar University (JU), Savar, Dhaka. During the whole working process, he supplied a lot of important advice and materials related to the current state of investigation. I would not have been able to complete the project work satisfactorily and on schedule without his assistance, courteous support, and generous time spans. I want to start by expressing my deep appreciation to him for his advice, insightful recommendations, encouragement, and friendly collaboration.

I also want to express my gratitude to the other IIT faculty members who have supported me in any way, whether directly or indirectly, in finishing this work. I want to thank all the various places I've gotten assistance. I owe thanks to everyone who contributed in some way to getting this job done, whether directly or indirectly.

Finally, I want to thank everyone at IIT, Jahangirnagar University, and my friends for supporting me and cooperating with me during the project.

ABSTRACT

Flooding is a major disaster that disturbs people's daily life. Based on this, there is an urgent need to perform rigorous environmental investigations on a regular basis. Physical modeling has now been replaced with mathematical modeling at all prediction levels. This paper, on the other hand, deals with flood alerting system strategies using the Internet of Things (IoT) an embedded system that would provide real-time calculation as well as Wireless Sensor Network (WSN) for computational processing, prediction, and analysis that would help to send an alert message to the nearby surroundings and reduce the time of risk. Here the Arduino Uno is connected to water sensor, rainfall sensor and waterflow sensor to analyse the water level. Further, these values will be passed to Arduino and it will pass the alert message to the IoT module (GSM module). Real time prediction and alerting can be done through this.

Keywords: GSM, Internet of Things, Water sensor, Arduino, Flood detection.

LIST OF FIGURES

Figure

TABLE OF CONTENTS

CHAPTER I

Introduction

1.1 Problem Statement

One of the most severe natural calamities that frequently strike Bangladesh is flooding. Flooding is the overflow of water from bodies of water like rivers and lakes that results in or threatens to result in damage. Some of the main problems regarding floods are:

- Water-borne diseases can spread if the warning is delayed.
- Delayed warnings can cause a threat to lives and livestock.
- Valuable belongings may be damaged.

So, there must be a way of ensuring early flood detection so that the described problems can be minimized.

1.2 Motivation

Floods are the most frequent natural disasters, and they are becoming a more prevalent occurrence in Asia, according to expert Abhas Jha. The following categories of people could find this study to be helpful.
Commuters: Notifications about impassable flooding roads will be provided through the SMS advice service. As a result, commuters won't get stranded on the route. The commuters will benefit from cost, time, and effort reductions.
Local Government: Once this innovation is adopted, the official in charge of the flood will find it easier to monitor and disseminate information to citizens. Officials may also provide further information or advise on preventive measures to take during the rainy season [1].

1.3 Objective & Scope

This project has several goals that must be reached, and the goals will serve as a roadmap for completing the project effectively.

- To use the microcontroller to develop a circuit and write a program.
- To create a system that can alert people to local flooding and monitor it in order to lessen the impact and expense of flood damage.
- To ensure the confirmation of a potential flood so that the civilians never get a wrong warning.
- Create a flood detection prototype system utilizing the included hardware and software.
- To comprehend communication system fundamentals and applications for the Arduino Uno.

1.4 Assumptions & Limitations

This research is done to find solutions for the issues caused by floods. The following characteristics must be present in the device: [2] It features a water sensor to determine the existence of water. The rainfall sensor and the waterflow sensor ensure that it is a flood. The technology has serial communication capabilities for text message alerts that provide information about the warning for floods. The three modules that make up the system are Users, Logs, and Contact Numbers. Admins can only change it. It is recommended that the device that contains the sensor should be positioned in a secured place where the water hits first after a flood has taken place. If the sensor is not positioned perpendicular to the floodwater, ultrasonic waves may reflect unevenly, leading to inaccurate measurement results. It is advised that the sensor be mounted on a pole that is between three and five meters tall. An 80k mAh solar power system will power the flood sensors and microcontrollers, ensuring continuous operation of water flood height detection and network data transfer [3].

1.5 Project Outline

The rest part of the report is structured like this:
Chapter II talks about the main system design and development process that is followed for solving the issue. In chapter III an experimental description is given. Here different types of hardware are explained in detail which we used in our project. In chapter IV, we will talk about interfacing the Arduino with the GSM module. Moreover, the working procedure of GSM is also described here. Then we have chapter V where the simulation part of the project is briefly highlighted. And in the last part (chapters VI and VII), the final results, discussion, and conclusion take place.

CHAPTER II

System Design and Development

2.1 System Environment

The project environment consists of water in a river, local residents, a water level sensor, a rainfall sensor, a waterflow sensor, a GSM module, a microcontroller, and other necessary pieces of equipment. All these elements have a relationship in some way here. For example, our sensor will be placed near a river. On a potential flood threat, the local residents will be notified with text messages. That's how civilians are also a part of our system environment [4][5].

2.2 Data Flow Diagram

Here is the data flow diagram/flowchart of the system. Here, the water sensor will keep taking reading its value and send it toward the microcontroller. Once received, the microcontroller analyses and compares its values with the predetermined value set into it. Then it takes the decision whether there is any water touching the device or not. If the result is positive i.e. the reading of the water sensor gets larger than a certain value, the microcontroller sends a signal so that the rainfall sensor and the waterfall sensor get activated. The rainfall sensor calculates the amount of water touching it for a time being and the waterflow sensor calculates the rate of water flow and the volume of water passed through it. If both of the sensor values reach a minimum level, the microcontroller triggers the GSM module so that nearby residents can be warned. It also triggers the alarm and buzzer to create alternate ways for the local people to be careful beforehand. On a negative result i.e. the reading of the water sensor stays lower than a certain value, the process just goes back to the previous step when the water sensor again starts taking its reading. Thus the whole process should be running continuously.

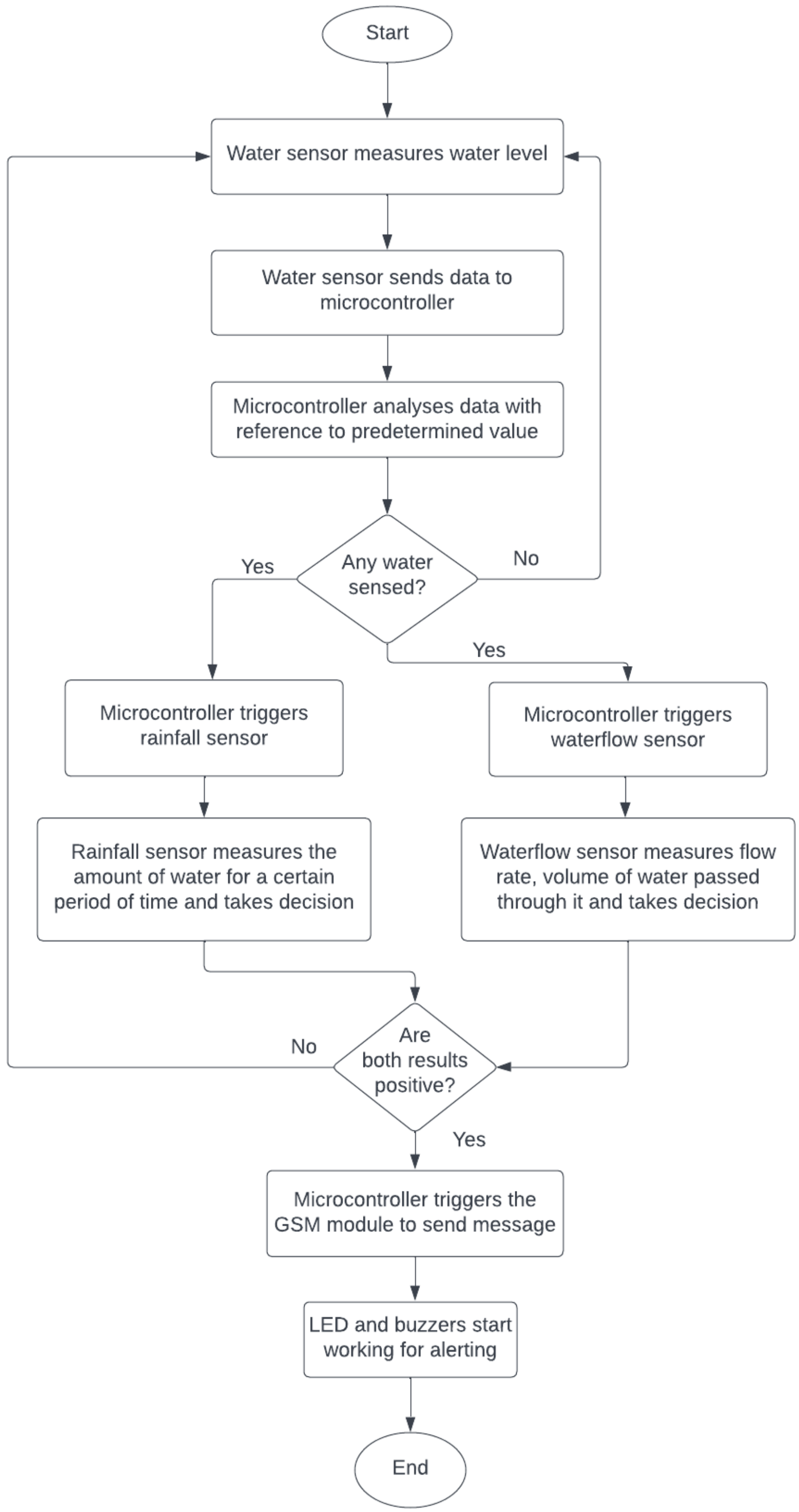

Figure 2.1: Data flow diagram of the system.

2.3 Development Model

Here is the overall block diagram of the process. The microcontroller (Arduino Uno) is in the center of it. On the upper side, it is connected to a mobile phone through GSM Module. On the left side, it is connected to Water Sensor. At the bottom, the rainfall sensor and the waterfall sensor are connected bidirectionally to the microcontroller.

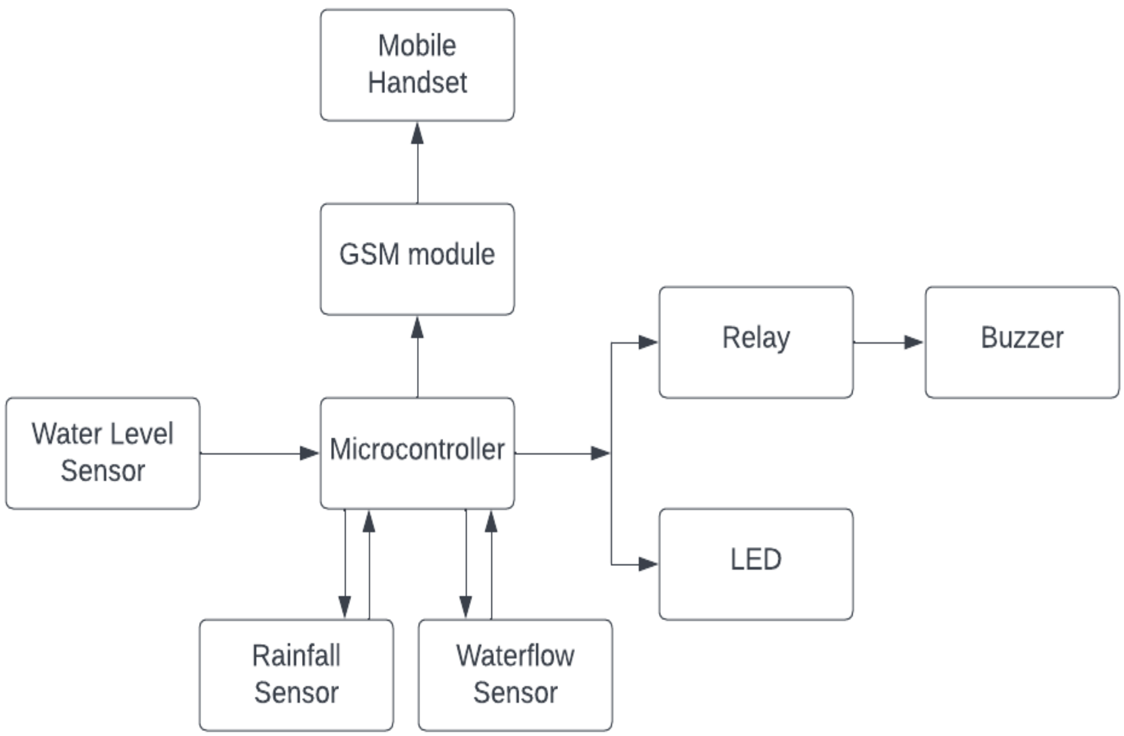

Figure 2.2: Block diagram of the system.

And finally, on the right side, it is connected to a buzzer and a led for output purposes. The buzzer is connected through a relay switch as it needs more voltage than the output voltage. So, the relay is used as an amplifier here [6][7].

CHAPTER III

Experimental Description

3.1 Hardware Configuration

For this investigation, an Arduino microcontroller R3 with operating voltages of 5V and 3.3V and a frequency of 16MHz is employed. It has three GNDs, five analog pins, 13 digital pins, and PWM respectively. Although they function as standard digital pins, these pins can also be utilized for a process known as pulse-width modulation (PWM). We use AREF for analog reference. We can generally leave this pin alone. The top limit for the analog input ports can occasionally be adjusted to an external reference voltage of 0-5 volts. The temperature sensors are attached to the analog pin (A0). The TX and RX pins of Arduino are linked to the RX and TX pins of the GSM module [8].

3.2 Hardware Details

We are going to need a bunch of hardware. Water level sensor, rainfall sensor, Water flow sensor, GSM module, and Arduino Uno are the mentionable items among them.

3.2.1 Water Level Sensor

For measuring or detecting the required amount of water or liquid in a space, a water level sensor or water level detector is a highly important tool. The water level indicator is relatively straightforward but does a superb job of monitoring the ideal level of any liquid. The water sensor is used here only to start the identifying process. Any splash of water can activate it. Once it is activated, the other two sensors come into work. Otherwise, it would be unwise to keep all three sensors

working simultaneously [9].
The water sensor is very sensitive to water and it gives increasing analog values when water or any of its splash is found. The threshold is set to be 50 here. It is a very low value relatively and it is used as some analog value that can be automatically generated without any sense of water. This may occur due to a bug in the water sensor. So, 50 is used as a bias here in order to work perfectly with it [10].

Figure 3.1: Water level sensor.

3.2.2 Rainfall Sensor

The rainfall sensor senses the amount of water that is in touch with the sensor. It has a maximum analog value of 1023. When the sensor is dry, it gives the value of 1023. As the water touches it, the value starts to decrease. The more amount of water keeps in touch with the sensor, the less value it gets. A threshold of 900 is set here. That means when the analog value of the sensor gets below 900, it votes in support of a flood occurrence. However, one vote isn't enough to take the decision. The other sensor and the third one must also have to give a positive result in order to detect the threat of a flood. [11].

Figure 3.2: Circuit diagram of water level sensor.

3.2.3 Waterflow Sensor

Waterflow sensor plays a great role, in fact, one of the most important roles in identifying a true or false result. The water level sensor and the rainfall sensor both may react positively even when there is no flood. There can be rain or any other event that makes the device wet. But it shouldn't be recognized as a flood. That's where waterflow sensor comes into play. It can measure the water flow rate as well as the volume of water passed through it. When the water level sensor gets activated, the waterflow sensor starts working and it examines for a certain period of time to calculate how much water passed through it. And according to this, it takes a decision whether an event is a flood or not.
There is a wheel inside the waterflow sensor and it rotates whenever water passes through it. And thus we can observe pulses from the sensor. Then, we calculated the volume(in mL) by multiplying 3.663 with the pulse rate. And the threshold is set to 70mL of water. So, after passing 70mL of water through this sensor, the event is finally identified as a potential flood. Because it is not usual that 70mL of water will pass through it in a random manner. So, our result will be more accurate and believable. [12].

Figure 3.3: Circuit diagram of motor driver IC.

3.2.4 GSM

The second generation of mobile technology is known as the Global System for Mobile (GSM) communication. Although third and fourth generation technology is becoming more and more common, GSM is currently the most successful and widely used communication technology.

Figure 3.4: GSM device.

In this project, the GSM module and MC microcontroller are interfaced. With the aid of MC, the message will be transmitted using AT instructions to a specific GSM mobile [13].

3.2.4.1 Operations of GSM

- Run the basic AT command.
- Note the GSM modem's IMEI (International Mobile Station Equipment Identifier) number.
- Establish a call connection to a GSM mobile phone (dial a number).
- Text the specified cellphone number.
- Using AT instructions, these actions are carried out. Four tactile switches have been used to provide the supply of these four operations. Each switch relates to one of the aforementioned purposes. With the AT+CMGF message format command, we may switch between text mode and PDU (Protocol Data Unit) mode. At+CMGS transmits a message to the network through GSM. On successful message delivery, the message reference value is returned to the GSM.
- The ESC key may be used to cancel sending, and ctrl-z is required to denote the end of the message body.

3.2.4.2 GSM SIM 900A Description

SIM900A Modem by SIMCOM is based on a Dual Band GSM/GPRS modem. It operates between 900 and 1800 MHz. The SIM900A can automatically search these two bands. AT Commands can also be used to set the frequency bands. Through the AT command, the baud rate is programmable between 1200-115200. To enable you to connect to the internet through GPRS, the GSM/GPRS Modem has an inbuilt TCP/IP stack. The wireless module SIM900A is incredibly small and dependable. This entire GSM/GPRS module is of the SMT variety, has an AMR926EJ-S core integrated single-chip CPU, and benefits from compact dimensions and low-cost solutions.

3.2.5 Arduino

Arduino is one of the most trending microcontrollers used nowadays. It is budget-friendly as well as easy to operate.

3.2.5.1 Overview

Arduino is a microcontroller and Uno is one of the various versions of it. It is based on the 8-bit ATmega328P microprocessor. It has some additional parts to work with the ATmega328P microprocessor which consists of a voltage regulator, crystal oscillator and serial connectivity. The Arduino Uno comes with a USB connection, a Power barrel jack, an ICSP header, 6 analog input pins, 14 digital input/output pins and six of which can be used as PWM (Pulse Width Modulation) outputs and other features [14].

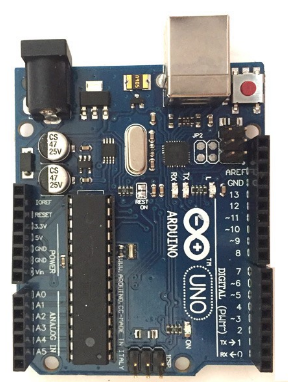

Figure 3.5: Arduino Uno.

3.2.5.2 Arduino Pin Configuration

The figure below describes the pin configuration for Arduino Uno. Here are 5 input pins (A0 to A5) and 13 output pins (PD0 to PD7 and PB0 to PB5). The PD0 and PD1 pins are used for transmitting and receiving signals. The PD2 and PD3 pins are used as 'interrupts' which get triggered after a specific task is done. There are also SS, MOSI, MISO, SCK, SCL and SDA pins in the input and output pins. Moreover, there are 3 GND (ground) pins, 3 voltage pins (1 voltage-in and 2 voltage-out), 1 RESET pin, and 1 AREF pin. All these pins have different functionalities, and they are briefly described in the next section.

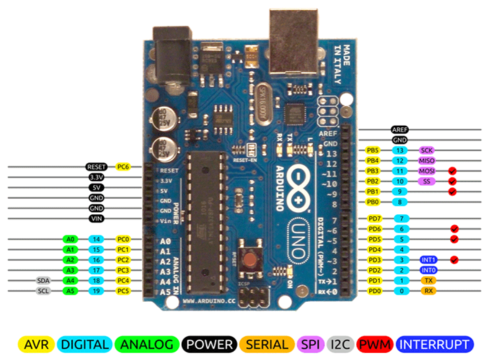

Figure 3.6: Arduino Uno pin configuration.

3.2.5.3 Using Arduino Board

When programming an Arduino, the pinMode(), digitalRead(), and digitalWrite() methods are usually used and they operate the digital input/output pins. There are fourteen of them. Each pin generally operates at five volts, but 3.3 volts is also a standard unit for this task. It has a built-in resistor(pull-up) of 20–50 kilo ohms. This resistor is usually placed as disconnected. It can conduct a current of 40mA (max). Some of these I/O pins have particular purposes and they are described below: [15]

- RX and TX pins are used to receive and send serial data respectively. The USB to TTL serial chip in the microprocessor is used to connect them.
- There are some interrupt pins. Of them, the external interrupt pins 2 and 3 may be used to trigger an interrupt by changing in input to low values, rising or falling edges, or value changes.
- Pulse Width Modulation pins are 3, 5, 6, 9, and 11 and they can generate an 8-bit PWM signal by analogWrite() method.
- There are four SPI pins. Pin 10 is SS, 11 is MOSI, 12 is MISO, and 13 is SCK. They are used to conduct SPI communication properly.

- Pin 13 is called the built-in LED pin which we used in our project experiment. The high state reflects that the LED is on and the low state signals that the LED is off.

- There are also six analog input pins. They can provide 10 bits of data which are as large as 1024 different values. These pins can calculate any value between 0 volt to 5 volts. We can also use the AREF pin and analog Reference() function to extend this limit.

- Analog pin 4 is SDA and pin 5 is SCA. They are responsible for TWI communication.

- The AREF pin is used to generate an analog voltage for the analog inputs. The analogReference() function is used therefore.

- The RESET pin is used to restart the microcontroller from scratch.

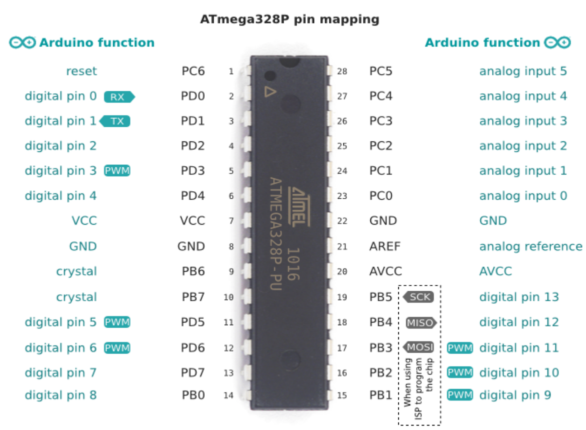

Figure 3.7: ATmega328P in Arduino board.

3.2.5.4 Communication

With the aid of an Arduino board, Arduino can communicate with a computer. The ATMega328P microcontroller lies at the heart of this Arduino board. It supports serial connection through pins 0 and 1. The Arduino software has a specific function called serial monitor. It facilitates data delivery and reception in text format from the Arduino board. When data is sent via the USB-to-serial chip and USB connection to the computer, the Arduino board's RX and TX LEDs blink. For serial communication, any digital pin on the Uno can be utilized. Furthermore, the ATmega328P

supports I2C/TWI and SPI connectivity. The Arduino software includes a library that simplifies the use of the I2C bus.
The pin mapping between the two is displayed in the image below when an ATmega328 chip can be used in place of an Arduino Uno, or vice versa.

3.2.5.5 Software

It's necessary to use the Arduino IDE (Integrated Development Environment) to program the Arduino Uno board. This software is open source and available on the internet for free. Though we can manually write the script in the text editor and compile it separately, we must use the IDE to insert the compiled code into the physical Arduino Uno microcontroller. So, this software is a must for any Arduino project. Moreover, this software has many library support which helps in many cases.

3.2.6 Overall Arrangement of Components

This is the overall arrangement of the components that have been used for building this project.

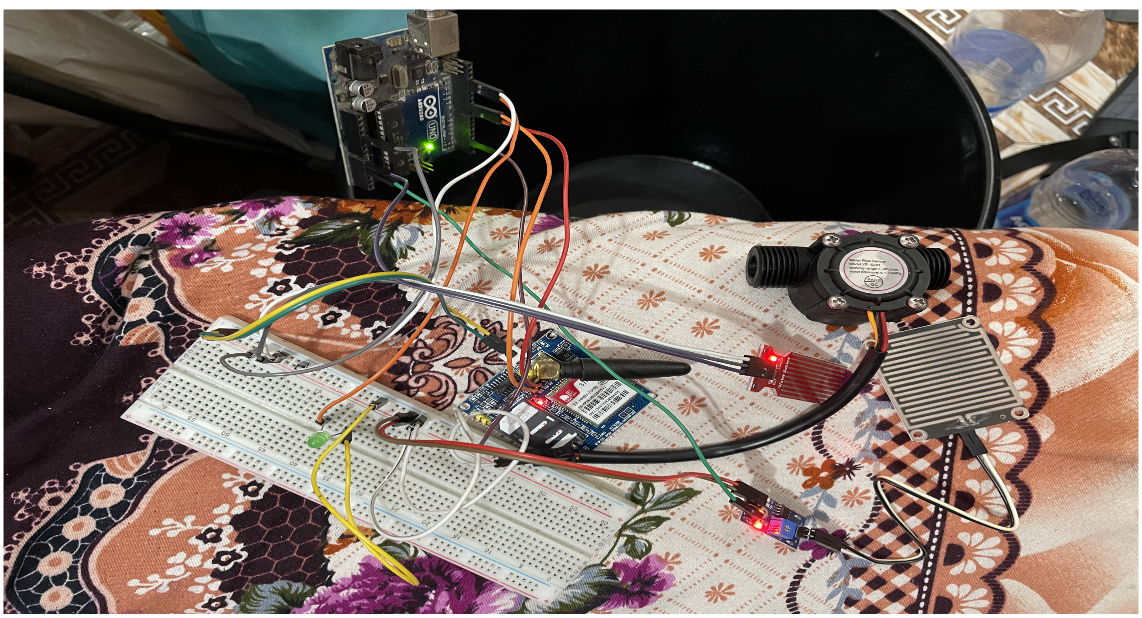

Figure 3.8: Arrangement of components.

The Arduino is connected to all the sensors either directly or indirectly through the breadboard. A laptop is used as a power source for Arduino and the 5v pin of

Arduino is used as a power source for the rest of the sensors and other components. A combination of three sensors is used to detect floods with high precision. And a GSM module is used for doing the final task i.e., sending messages to the people concerned. The other staff components used here are jumper wires, led, and buzzer.

CHAPTER IV

Interfacing Arduino with GSM SIM900A Module

Material Preparation

We are going to need:

- Arduino Uno with connector
- GSM SIM900A
- 5V power adapter
- SIM card
- Some wires and breadboard

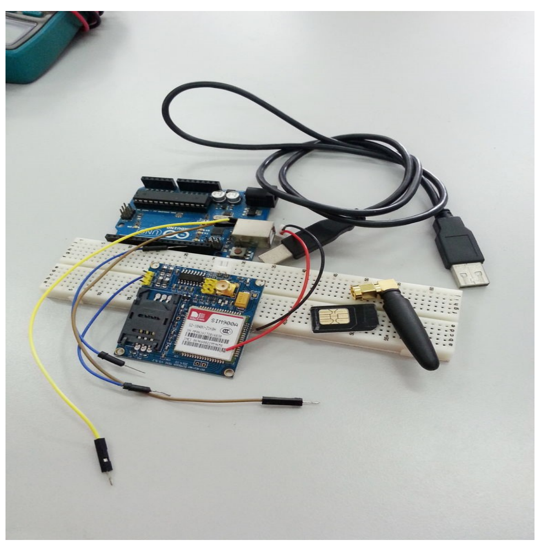

Figure 4.1: Material preparation.

Booting Up GSM module

First, the SIM card is inserted and secured into the GSM module. The GSM module is linked to the antenna.

Figure 4.2: Booting up GSM SIM900A.

GSM is activated by connecting it to the Arduino's 5V and GND.

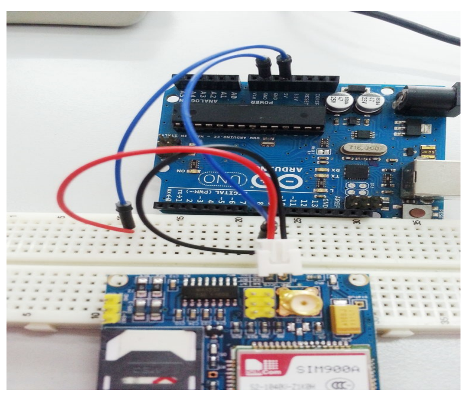

Figure 4.3: Connecting to Arduino.

Now we wait for a while to watch how fast the LED blinks. This is due to the fact that the GSM module often takes some time to establish a connection with the mobile network. When the connection is properly established, the LED will blink continuously every 3 seconds.

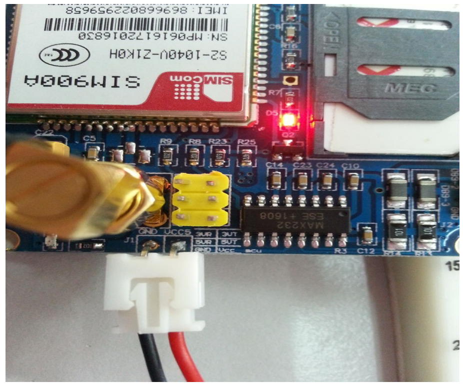

Figure 4.4: GSM SIM900A is powered up.

We might try calling the sim card's mobile number from within the GSM module. If we get a ring back, we know the GSM module has successfully established a network connection.

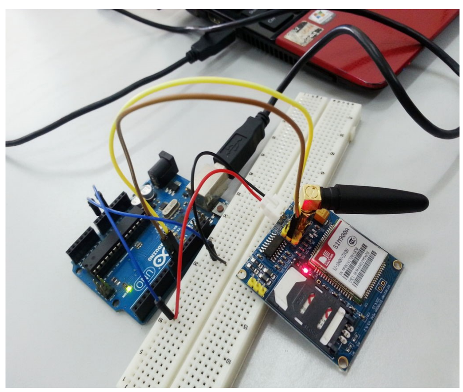

Figure 4.5: Overall view of GSM SIM900A connected to Arduino Uno.

Pin Connection

On the SIM900A, a TTL pin with the following pin numbers can be seen: 3VR, 3VT, 5Vr, 5VT, VCC, and GND. For serial communication between Arduino and the SIM900A module, we must connect GSM's 5VT to Arduino D9 and GSM's 5VR to Arduino D10. The TX pin of one must be connected to the RX pin of the other in order to complete the transmission and receiving process. In our case, the line that plays the most important role is the TX pin of SIM900A connected to the RX pin of Arduino Uno as the signal will go from SIM900A to Arduino Uno as an input signal. [13]

Library

An Arduino library called SoftwareSerial enables serial data transfer via the board's additional digital pins. The library manages serial communication as well as hardware functionality replication. We must download and extract this library into your Arduino's libraries in order to connect. This library will also be necessary for the simulation as this can't be found as a built-in option. So, we must configure it manually before working with this.

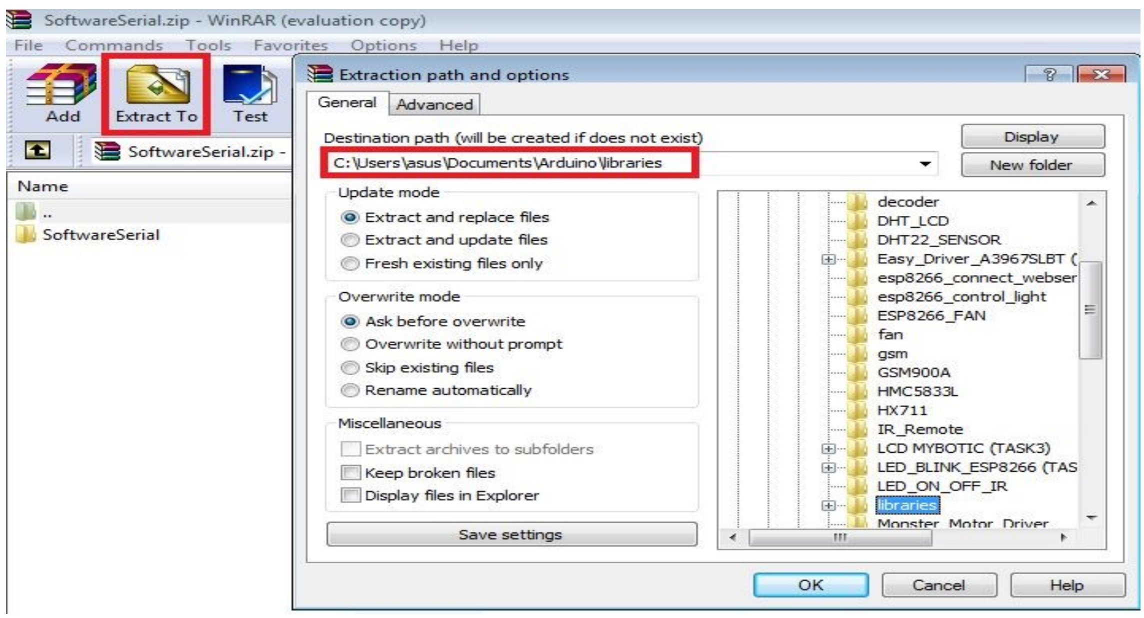

Figure 4.6: Arduino library.

Sample Source Code

The main source code can be found in the provided GitHub repository [16]. A sample is attached herewith in the figure below. In gist, the code looks like a C program. Here, we declared some global variables. The setup() function initializes all the variables that we used in our code. And the loop() function is called continuously. So, the main task is written in the loop() function. The main task here is to check if the water sensor is triggered. This check is done continuously in this loop() function. If the check becomes true, some necessary changes are made. Such as, transferring the signal to Arduino Uno. And when the Arduino Uno gets this information, it first rings the buzzer along with the alarm. Moreover, it also sends text messages to the numbers that were saved previously in our database. Another function named sendsms() is used to complete this task.

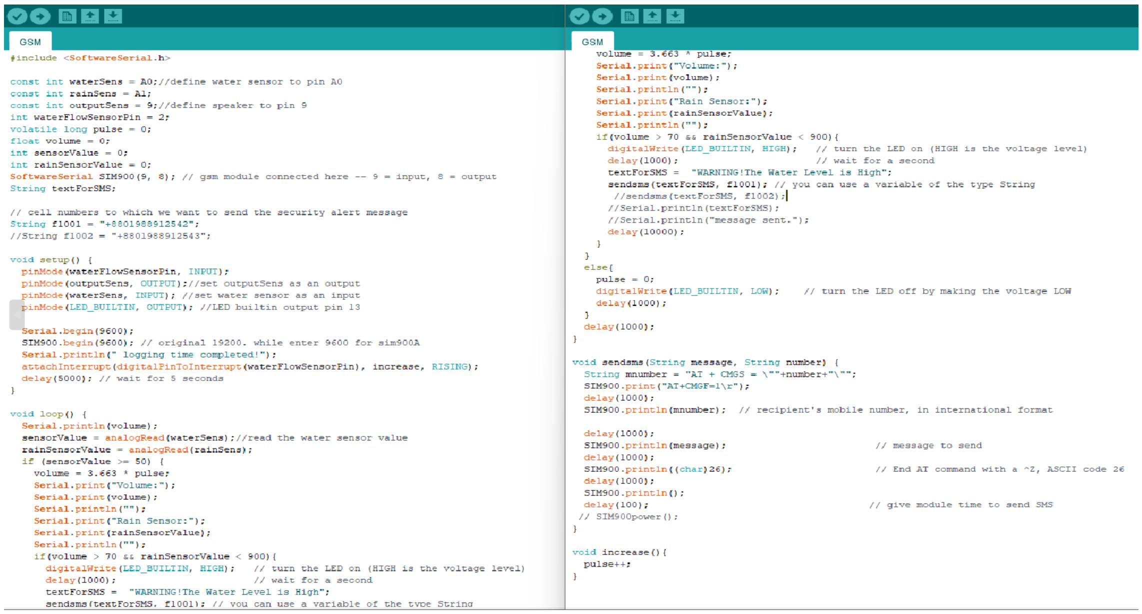

Figure 4.7: Sample source code.

Here we used some delay so that we can get some time to adjust with the situation which means the GSM Module will get enough time to operate i.e., send messages to the targeted persons.

With the help of the GSM module, we can easily send messages to our desired person very easily. A demo can be found in the figure below. The sendsms() function is responsible for sending and receiving messages. The receiver has nothing to do in order to receive the message. However, the sender GSM Module must stay in the range of the network area of the sim that we used here.

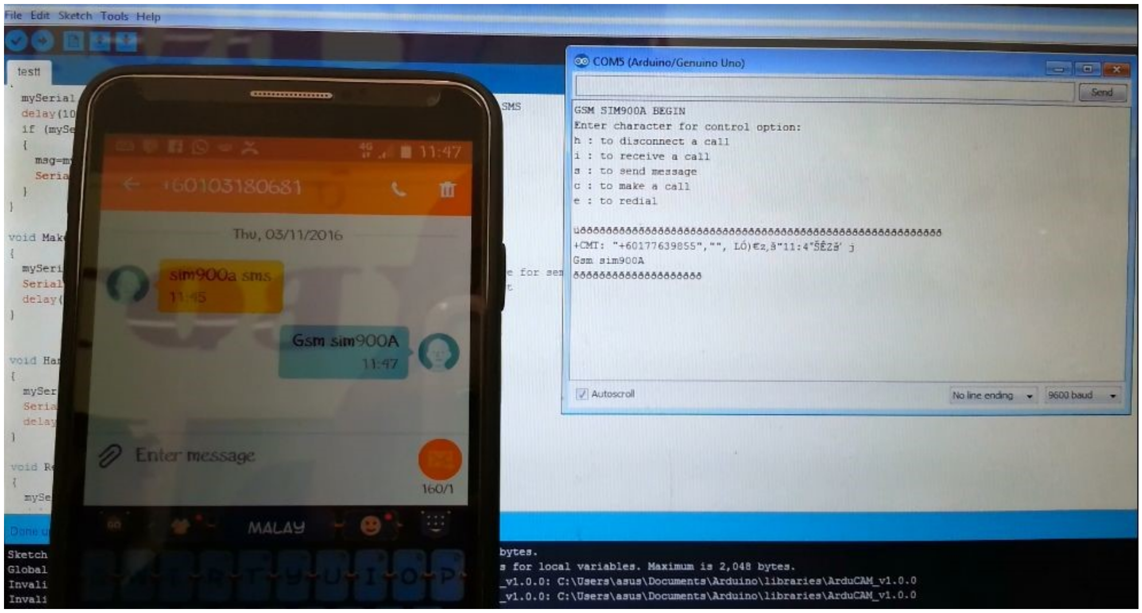

Figure 4.8: Sending and receiving SMS through GSM SIM900A.

We must be careful about the network coverage area. It is very possible that flood-prone area doesn't have a high coverage network of every sim company in our country. So, we must make a trial session before deploying this system for working. The sim card with the highest accuracy rate should be taken into consideration.

CHAPTER V

Simulation

5.1 Overview

Along with the physical project, we also did the same thing virtually. We used Proteus 8.4 SPO version as simulation software. The overall circuit setup looks like the figure below. Proteus is one of the most used simulation software nowadays. Though it doesn't provide a built-in solution for all the components we used, it has high community support. Thus, the libraries provided by the external sources were used here.

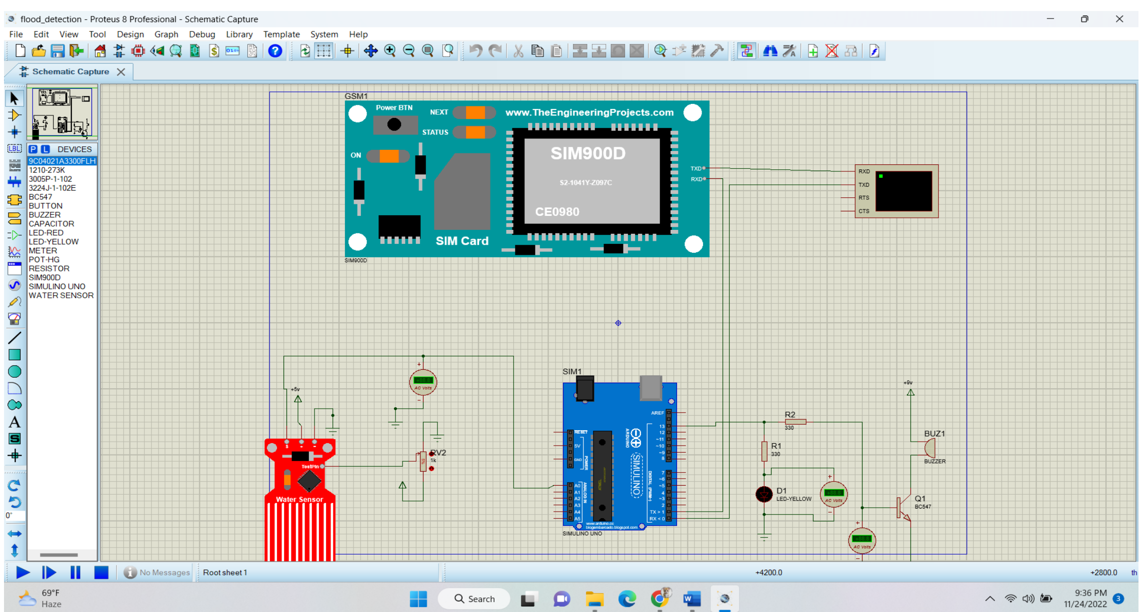

Figure 5.1: Circuit setup for simulation.

The other sensors that we used outside of these are a rainfall sensor and a waterflow sensor. As these two sensors are new, proteus 8.4 doesn't have the library support for these two sensors. As a result, these two sensors can't be simulated here. Here we assumed that the water sensor alone did the task of all three sensors.

5.2 Description

The main components used here (Arduino, GSM, and Water sensor) weren't present in the proteus. So, we had to use external libraries. Most of them were collected from [10]. We used a virtual terminal to show the messages that people will see on their mobile phones.

We know the water sensor gets triggered when drops of water fall on it. But it isn't possible to show in simulation. For that reason, we used a variable resistor in order to get the water sensor triggered by manually providing voltage to it.

We used some voltmeters in order to observe the changes in voltage. We also had to use some extra resistors so that the components don't get over-voltage or under-voltage.

After a successful detection in the water sensor, the outcome looks something like the figure below.

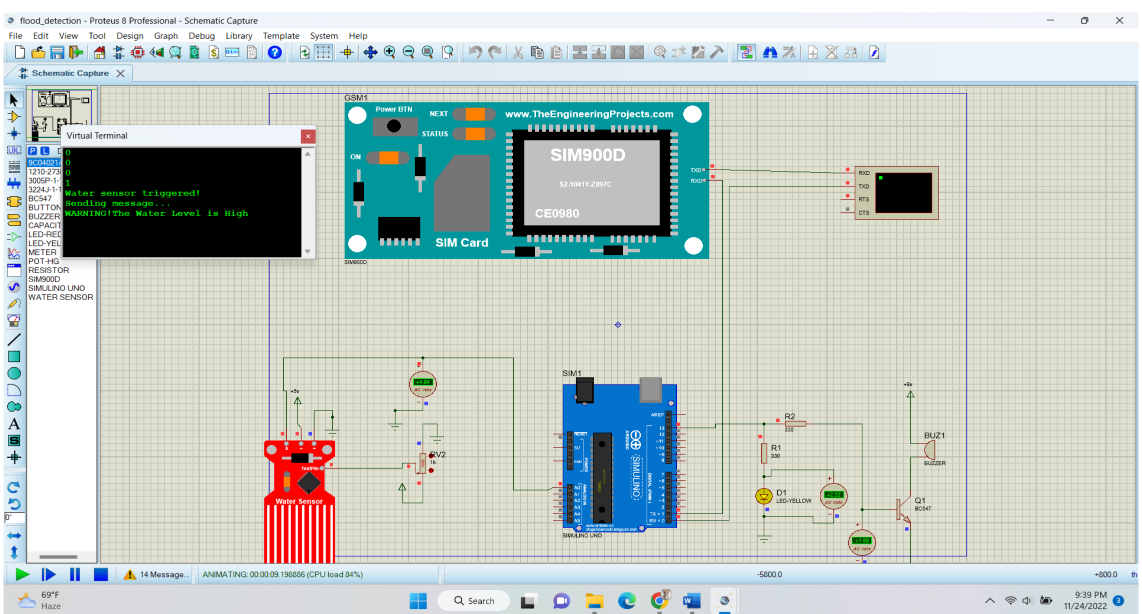

Figure 5.2: Running the simulation.

At that time, the LED gets lightened, and the buzzer starts beeping.
The details of the simulation, necessary codes along with a video demonstration can be found in the provided GitHub repository [16].

CHAPTER VI

Results and Discussion

6.1 Results

In gist, the main procedure for both real-life and simulation works like this:
After the successful detection of water, the signal passes to the Arduino through a line. And for this successful detection mechanism, 2-step verification is used. In the first step, the water level sensor senses the presence of water and thus activates the other two sensors. And in the second step, the rainfall sensor measures how much amount of water is in touch with it and the waterflow sensor measures the water flow rate and volume. If both of these two measurements reach a certain value, the event is detected as a potential flood and this is reported to the Arduino as a signal.
This signal does two tasks at a time. Firstly, it makes the LED lighten and beeps the buzzer. Secondly, it triggers the GSM so that the GSM device can send messages to nearby residents. The mobile numbers of the residents should be kept in a database. This warning alert can be sent to as many people as we want. In this way, our project can be a great help to the people of flood-prone areas as they will be notified earlier about a potential flood occurrence.

6.2 Discussion

To work on this project, a lot of challenges had to be faced. We are going to discuss them thoroughly.

- Firstly, for checking if the sensors are working, we had to write demo codes for them and check whether they are actually working or not. These type of sensors usually gets damaged very often according to my experience. But in this case, I was lucky enough that all of the sensors worked perfectly on the first attempt. To check if the sensors are working, their output values are printed in the terminal of the Arduino IDE.

- Then, the GSM module is another damage-prone device and it's very difficult to find one that actually works. The first three of my four GSM modules didn't work. However, the fourth one worked properly and is implemented in the project. But this GSM module sometimes takes more time to send the messages. It may also occur due to the bad signal of the service provider of the sim card that we used.

- The water level sensor and the rainfall sensor almost do the same work. But here I tried to use them as separate functions. The 2-step verification is the concept used here. The water level sensor is used in the first step and the rainfall sensor in the second step. I tried to optimize the battery consumption and this process helps a lot here. The rainfall and the waterflow sensor comes into work only when the water level sensor tells them to do so.

- If any battery is used to provide power to the Arduino, it gets discharged very fast as the quality of these batteries that we find in the market isn't up to the mark. That's why I had to use the laptop as a power source for the demonstration of the project. But it is recommended that we use a good quality battery as a power source for this device so that the device can be easily portable.

- In the simulation, a virtual terminal is used to show the messages that are actually sent to the people concerned. I had to use different codes for the simulation and the real project.

CHAPTER VII

Conclusion and Future Works

7.1 Conclusion

In this report, we tried to implement a hardware-based flood detection system that can be named “Automatic Flood Detection System”. After the study is over, the objectives have been developed as the basis for working. The "Automatic Flood Detection System ", which was intended to assist the civilians of a flood-prone area, achieved our overall goal. The commuters will benefit from saving money, time, and effort. One of the quickest ways to check for and monitor flooding is the Automatic Flood Detection System.

7.2 Future Works

We would like to suggest that the Local government unit of flood control see the study "Flood Detector System using Arduino" as a critical source of flood information based on the outcomes of the findings and conclusions acquired. We would also seek to provide households with access to a website that contains information on the flood level. Additionally, it is expected that the established system serve our entire country. Finally, we would like to advise continuing the study or progressively enhancing it for the benefit of upcoming researchers.

References

[1] U. K. Panchal, H. Ajmani, and S. Y. Sait, "Flooding level classification by gait analysis of smartphone sensor data," *Ieee Access*, vol. 7, pp. 181678–181687, 2019.

[2] Z. Wu, Y. Zhou, and H. Wang, "Real-time prediction of the water accumulation process of urban stormy accumulation points based on deep learning," *IEEE Access*, vol. 8, pp. 151938–151951, 2020.

[3] M. A. Baballe and Z.-A. Abbati, "A review of flood detection systems,"

[4] A. Islam, M. Uddin, M. Shamrat, and M. Rahaman, "Flood and flood management in bangladesh," *A term paper, Depertment of Disaster Science and Management, Faculty of Earth and Environmental Sciences, University of Dhaka*, 2017.

[5] M. H. Kabir and M. N. Hossen, "Impacts of flood and its possible solution in bangladesh," *Disaster Adv*, vol. 12, no. 10, pp. 48–57, 2019.

[6] N. Mahzan, N. M. Enzai, N. Zin, and K. Noh, "Design of an arduino-based home fire alarm system with gsm module," in *Journal of Physics: Conference Series*, vol. 1019, p. 012079, IOP Publishing, 2018.

[7] A. G. D. L. C. V. EDWIN DE GUZMAN, VALERIE SHANE CUADRA, "Flood detector system using arduino." `https://docplayer.net/52575716-Flood-detector-system-using-arduino.html`.

[8] "Components 101." `https://components101.com/tags/arduino`.

[9] "Simple water level sensor or liquid level detector." `http://www.circuitdiagram.org/simple-water-level-sensor-liquid-detector.html`.

[10] D. Watson, "The engineering projects." `https://components101.com/tags/arduino`. Accessed: 07-2020.

[11] "Rain sensor working and its applications." https://www.elprocus.com/rain-sensor-working-and-its-applications/.

[12] M. Kumar, "Measuring water flow rate and volume using arduino and flow sensor." https://circuitdigest.com/microcontroller-projects/arduino-based-water-flow-sensor#:~:text=Arduino%20Water%20Flow%20Sensor%20Working&text=In%20this%20condition%2C%20the%20code,are%20generated%20from%20the%20sensor.

[13] "Working with gsm module." https://www.fabian.com.mt/viewer/36294/pdf.pdf.

[14] vandenbrande, "Arduino - flood alarm - using a simple water level sensor." https://www.instructables.com/Arduino-FLOOD-ALARM-Using-a-Simple-Water-Level-Sen/.

[15] "Atmega328p pin mapping." https://www.pinterest.com/pin/414401603184128742/.

[16] S. Das, "Flood detection using arduino." https://github.com/piy4l/Flood-Detection-using-Arduino.