VIT-AP UNIVERSITY

ENGINEERING CLINICS

AQUA CONTROLLER - Wireless pump control for Field Irrigation

Using Aurdino and GSM

FINAL REPORT(Working Prototype)

Guide

Prof Dr. Visalakshi Annepu

GROUP MEMBERS:

21BCE9584-GUDURU DASTAGIRI
21BCE9671-KARUMANCHI BHAVYA SRI
21BCE9406-AKURATHI VARSHITHA
21BEC7211-MUDIMI CHARAN
21BCE9149-NERELLA PAVAN
21BCE9561-MOLLETI MONIKA SAI

ABSTRACT:

Field irrigation plays a crucial role in agricultural practices where water supply to crops needs to be effectively managed. However, the traditional manual control of pumps used in field irrigation presents several challenges. Operators have to physically monitor and operate the pumps, resulting in increased labor costs, potential delays in responding to effect in crop production. To overcome these challenges, a wireless pump control system for field irrigation is required. The system should enable remote monitoring and control of the pumps, providing real-time information on their status and allowing efficient water management. Our project centers on the ingenious idea of controlling irrigation pumps remotely through a simple mobile call.

This system has several advantages over traditional methods of water pump control. It is more convenient for farmers. They no longer have to travel to the pump to turn it on or off. They can do it from anywhere with an simple mobile call. This saves them time and effort, and allows them to focus on other tasks.

Our objective is to empower farmers with the ability to remotely control irrigation pumps through a simple mobile call. This game-changing approach not only simplifies the irrigation process but also maximizes water utilization, minimizes manual intervention, and ultimately contributes to sustainable and eco-friendly agricultural practices. Through the fusion of telecommunications and agricultural engineering, our project redefines convenience and efficiency in the farming landscape.

Table of Contents

Chapter 1: **Introduction** 6

Chapter 2: **Background** 6

Chapter 3: **Problem defination** 6

Chapter 4: **Objectives** 7

Chapter 5: **Methodology** 7

Chapter 6: **Procedure** 8

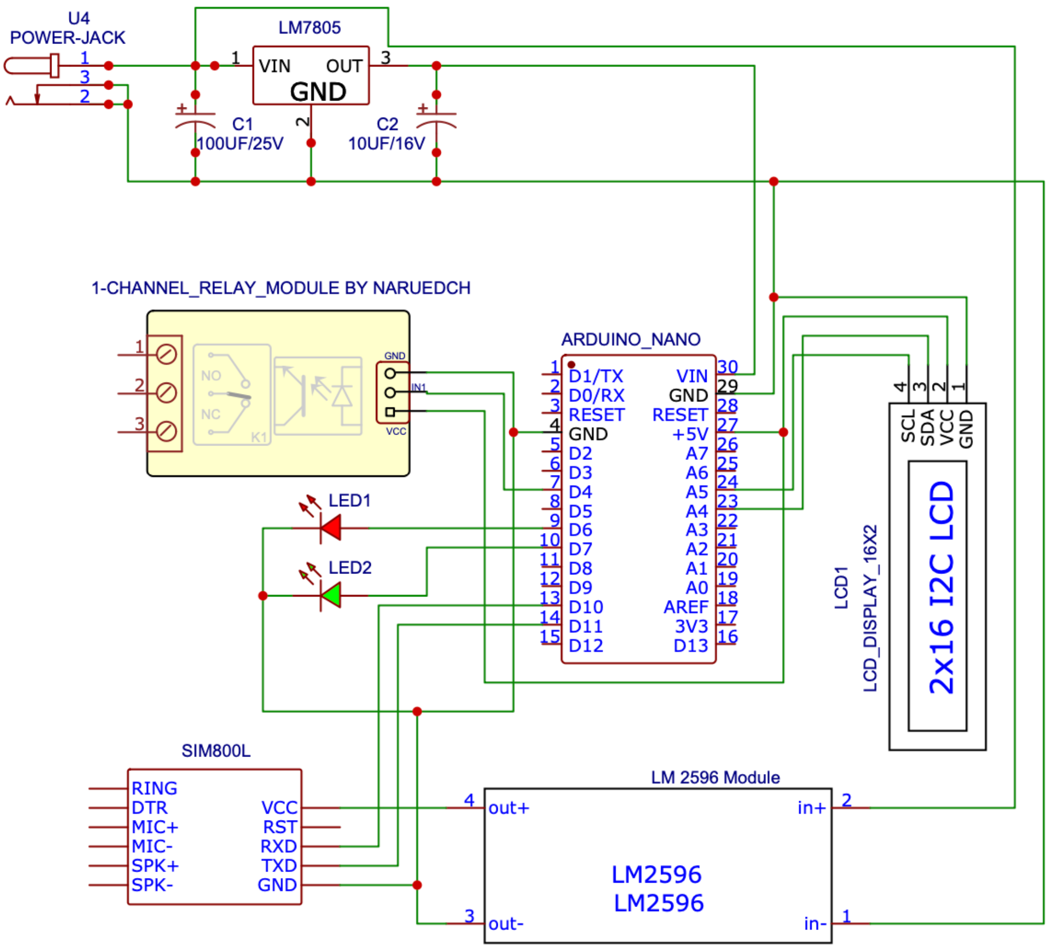
Chapter 7: **Results and Discussion** 10

Chapter 8: **Conclusion and Future Scope** ..10

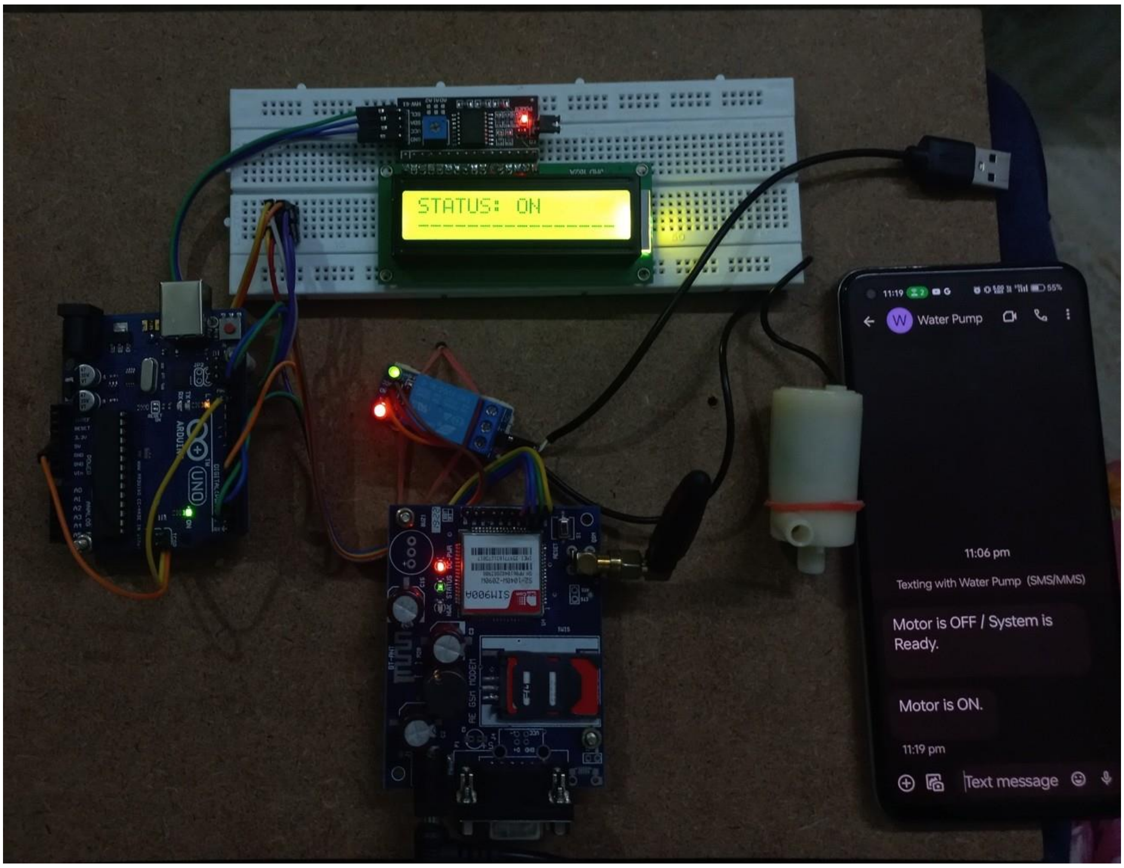
Chapter 9: **References** 10

Chapter 10 :**Codes** 11

Circuit Diagram



Working Prototype



This is our Project Working Prototype initially it performs an eeporam test Which analyses code and make sure it is error free or not after that it checks Model connectivity. First, it checks if everything is okay with its memory. Then, it looks at a special part that talks to phones. If that part is working well, the prototype sends a message to the farmer's phone, telling them the water pump can be turned on. When the farmer gets the message and calls the prototype, the water pump starts. The prototype also sends a message back to say it's started. If the farmer calls again, the water pump stops, and the prototype sends another message. Even if the power goes off while the pump is on, when the power comes back, the pump starts again by itself. This smart system helps farmers water their crops from far away and deals with power problems too.

Introduction:

The Mobile-Controlled Water Pump System for Farmers is a innovative solution tailored to address the unique challenges faced by farmers in remote areas. These challenges include the distance between fields and homes, as well as limited access to a stable power supply.

The primary objective of this project is to empower farmers with the ability to remotely manage and control water pumps within their fields, leveraging the convenience and efficiency of mobile technology. By implementing this system, farmers can optimize irrigation practices, minimize water wastage, and reduce the need for manual intervention.

Back ground:

The project "Wireless Pump Control for Field Irrigation" aims to provide a convenient and efficient solution for controlling irrigation pumps remotely using mobile phones. This technology addresses the need for automated irrigation systems that can be controlled From a distance, which is especially important in agricultural settings where timely and precise irrigation can greatly impact crop yield and water usage efficient By using a wire less pump control system, farmers can save money on water and energy costs. They can also save time and labor costs by not having to travel to the pump to turn it on or off. Farmers can respond quickly to emergencies such as sudden changes in weather conditions or unexpected water shortages. They can remotely start the pump to mitigate potential damage to crops. If a farmer manages multiple fields or farms at different locations, they can remotely control the irrigation of each site using a single mobile phone.

Problem definition:

The primary issue this project aims to solve is the lack of convenient and efficient irrigation management for farmers in remote areas. The need for manual intervention and the absence of a reliable power supply further exacerbate the problem, leading to suboptimal water usage and reduced agricultural productivity.

Objectives of the Proposed Work:

The key objectives of the proposed work include:

Developing a system that enables farmers to remotely control water pumps using their mobile phones.

Enhancing irrigation management to ensure optimal water usage and crop health.

Reducing the dependency on manual intervention and mitigating the impact of limited power supply.

Methodology/Procedure:

The Mobile-Controlled Water Pump System utilizes the following components:

- Arduino Uno or similar microcontroller board
- GSM module for mobile communication
- 16x2 LCD display
- Relay module for controlling the water pump
- Power supply
- Beard bord
- Jumper wires
- Water pump

Arduino Uno:

Arduino Uno is a microcontroller board based on the ATmega328P chip. It is one of the most widely used and beginner-friendly boards in the Arduino family. The board comes with digital and analog input/output pins that allow you to connect various sensors, actuators, and other electronic components to create interactive projects. It also has a USBinterface, power jack, and an onboard voltage regulator.

Key features of Arduino Uno:

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Digital I/O Pins: 14 (of which 6 can be used as PWM outputs)
- Analog Input Pins: 6
- Flash Memory: 32 KB
- SRAM: 2 KB
- EEPROM: 1 KB

- Clock Speed: 16 MHz

Arduino Uno is programmed using the Arduino IDE, which is an open-source development environment that simplifies the process of writing, compiling, and uploading code to the board.

GSM Module:

GSM modules enable communication over the GSM network. They allow you to send and receive SMS messages, make phone calls, and access the internet (limited capabilities compared to modern smartphones). GSM modules usually come in the form of small boards that can be easily integrated into various projects.

Pump Control Relay:

An electronic switch used to turn the water pump ON or OFF based on commands received from the GSM module.

The system integrates the Arduino board with the GSM module and relay module. When a Farmer initiates a call from their mobile phone to the GSM module, the system verifies the user's identity and processes the command to turn the water pump ON or OFF. The relay module, connected to the Arduino, controls the pump's power supply accordingly.
The system's status is displayed on the LCD display, providing real-time information to the user.

Hardware Setup:

This project involves the integration of an Arduino uno, which functions as a mini CPU, and a GSM module, which enables the sending and receiving of SMS messages, phone calls, and internet access (limited in comparison to smartphones). The GSM module is connected to the Aurdino uno via the UART(SERIAL) interface. Generally, the TX (transmission) of the module will be connected to the RX (receive) of the Aurdino uno and vice-versa. Additionally, a relay module is used when the pump operates on a different voltage or requires a higher current than the Arduino can provide. So at that time relay module is used to control the power supply of the pump or motor driver. Connect the pump or the motor driver to the relay module or directly to the Arduino and Connect the relay module to one of the Arduino's digital pins. All connections should be secure, and the wiring should be double-checked before continuining.

Software Setup

1. Begin by installing the Arduino IDE on your computer.
2. Connect the Arduino board to your computer using the USB cable.
3. Launch the Arduino IDE and paste the provided code for the Mobile-ControlledWater Pump System.
4. Verify the code for any potential errors to ensure smooth operation.
5. Upload the verified code to the Arduino board, allowing it to take control of thesystem.
6. Ensure that the SIM card is securely inserted into the GSM module and has been activated correctly.
7. Power on the system and double-check that the GSM module establishes a connection with the mobile network.

Operations:

1. Once the setup is on, it tests the eeporm and performs the required diagnosis, And sends a message that “motor is on/ System is Ready”.
2. When a farmer makes a call from their mobile phone to the GSM module, the system verifies the number (It only allows authenticated users).and turn On the pump and an acknowledgement will be sent.
3. The Arduino board processes the incoming calls sequences and extracts the command to turn the water pump ON or OFF.
4. Based on the received command, the Arduino board controls the relay module, which in turn switches the water pump ON or OFF.
5. The status of the water pump (ON or OFF) is displayed on the LCD display.

8. Begin by installing the Arduino IDE on your computer.
9. Connect the Arduino board to your computer using the USB cable.
10. Launch the Arduino IDE and paste the provided code for the Mobile-Controlled Water Pump System.
11. Verify the code for any potential errors to ensure smooth operation.
12. Upload the verified code to the Arduino board, allowing it to take control of the system.
13. Ensure that the SIM card is securely inserted into the GSM module and has been activated correctly.
14. Power on the system and double-check that the GSM module establishes a connection with the mobile network.

Results and Discussion:

The implementation of the Mobile-Controlled Water Pump System has yielded several positive outcomes. Farmers now have the capability to remotely control water pumps, significantly improving irrigation management. This has led to more efficient water usage, reduced manual labor, and enhanced crop health. The integration of modern technology has bridged the gap between remote agricultural fields and the farmers' homes, streamlining the irrigation process.

Conclusion and Future Scope:

In conclusion, the Mobile-Controlled Water Pump System for Farmers has demonstrated Its effectiveness in addressing the challenges faced by farmers in remote areas. By providing remote control capabilities through mobile phones, the system has transformed irrigation management.

Future enhancements could include:

1. Development of a mobile application for more user-friendly interaction.
2. We want to include the features like protection against single phasing, over-current, dry running.
3. By adding different sensors like humidity, temperature, moisture, light etc.
   According to sensor values graph will be display on Smart phone side and by using this graph user can on or off motor.
   To decrease the work for farmers and to increase the automation in the irrigation.
   The irrigation time depending on the temperature and humidity reading from sensors and type of crop and can automatically irrigate the field.

Reference

1. V. Ahmed and S. A. Ladhake, "Innovative Cost Effective Approach for
   i. Cell Phone Based Remote Controlled Embedded System for Irrigation," 2011 International Conference on Communication Systems and Network Technologies, Katra, India, 2011, pp. 419-422, doi: 10.1109/CSNT.2011.93.
2. Adjardjah, Winfred, David Bright Kofi Arthur, Alex Ewuam, and Kingsley Nunoo. "The Design of a Mobile Phone-Based Remote-Control Application to Submersible Motor for Effective Water Supply." Journal of Sensor Technology 12, no. 2 (2022): 19-31.
3. SUMAN, B., REDDY, B.S. and RAMANI, S., 2019. Smart Irrigation System using Mobile Phone.
4. Gautam I, Reddy SR. Innovative GSM bluetooth based remote controlled embedded system for irrigation. International Journal of Computer Applications. 2012 Jun;47(13):1-7.
5. Kumar, G.S., Nagaraju, G., Rohith, D. and Vasudevarao, A., 2023. Design and Development of Smart Irrigation System Using Internet of Things (IoT)-A Case Study. Nature Environment and Pollution Technology, 22(1), pp.523-526.

Code:

```
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <EEPROM.h>
SoftwareSerial gsm(2,3); // RX, TX
LiquidCrystal_I2C lcd(0x27, 16, 2);

int address = 0;
int wr_call_add = 1;
int f_address = 2;
int eeprom_dead = 3;
int wr_call_val = 0;
int f_value = 70;
int value = 0;
int ring = 0;
int i = 0;
int var = 0;
int eeprom_tst = 0;
int eeprom_tst_ack = 0;
int tst_var = 100;
int wr_call_tst = 100;
int eeprom_dead_val = 0;
String number = "";
String string = "";
const int output = 5;
const int red = 7;
const int buzzer = 4;
const int green = 6;
boolean wait = true;
boolean at_flag = 1;
boolean net_flag = 1;
char str[]="+917815880448";
void setup()
{
Serial.begin(9600);
lcd.init();
lcd.backlight();
gsm.begin(9600);
pinMode(output, OUTPUT);
pinMode(red, OUTPUT);
pinMode(buzzer, OUTPUT);
pinMode(green, OUTPUT);
if (EEPROM.read(f_address) != f_value)
{
EEPROM.write(f_address, f_value);
EEPROM.write(address, value);
EEPROM.write(wr_call_add, wr_call_val);
eeprom_dead_val = 0;
```

```
EEPROM.write(eeprom_dead, eeprom_dead_val);
}
if (EEPROM.read(eeprom_dead) == 1)
{
while (true)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("EEPROM Error.");
lcd.setCursor(0, 1);
lcd.print("System Disabled.");
delay(1500);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Reload the code");
lcd.setCursor(0, 1);
lcd.print("with new address");
delay(1500);
}
}
if (EEPROM.read(address) == 0)
{
digitalWrite(output, HIGH);
digitalWrite(red, HIGH);
digitalWrite(buzzer, LOW);
delay(2000);
digitalWrite(green, LOW);
gsm_init();
lcd.clear();
if (EEPROM.read(wr_call_add) == 0)
{
lcd.setCursor(0, 0);
lcd.print("Sending SMS");
lcd.setCursor(0, 1);
lcd.print("Acknowledgement.");
delay(1000);
gsm.println("AT+CMGF=1");
delay(500);
gsm.print("AT+CMGS=");
gsm.print("\"");
gsm.print(str);
gsm.println("\"");
delay(1000);
gsm.println("Motor is OFF / System is Ready.");
delay(100);
gsm.println((char)26);
}
if (EEPROM.read(wr_call_add) == 1)
```

```
{
wr_call_val = 0;
EEPROM.write(wr_call_add, wr_call_val);
}
}
if (EEPROM.read(address) == 1)
{
eeprom_test();
output_begin();
}
}
void(* resetFunc) (void) = 0;
void loop()
{
serialEvent();
if (ring == 1)
{
number = "+917815880448";
var = string.indexOf("+CLIP: \"");
if (var > 0)
{
number += string.substring(var + 8, var + 13 + 7);
}
if (number[0] == str[0] && number[1] == str[1] && number[2] == str[2] && number[3]
== str[3]
&& number[4] == str[4] && number[5] == str[5] && number[6] == str[6] && number[7] ==
str[7]
&& number[8] == str[8] && number[9] == str[9] && number[10] == str[10] && number[11]
== str[11])
{
gsm.println("ATH");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Incomming call");
lcd.setCursor(0, 1);
lcd.print("Number Verified.");
delay(2000);
if (EEPROM.read(address) == 0)
{
EEPROM.write(address, 1);
}
else if (EEPROM.read(address) == 1)
{
EEPROM.write(address, 0);
}
resetFunc();
}
```

```
if (!(number[0] == str[0] && number[1] == str[1] && number[2] == str[2] && number[3]
== str[3]
&& number[4] == str[4] && number[5] == str[5] && number[6] == str[6] && number[7] ==
str[7]
&& number[8] == str[8] && number[9] == str[9] && number[10] == str[10] && number[11]
== str[11]))
{
gsm.println("ATH");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Unknown number");
lcd.setCursor(0, 1);
lcd.print("Call Rejected.");
wr_call_val = 1;
EEPROM.write(wr_call_add, wr_call_val);
delay(2000);
resetFunc();
}
}
if (EEPROM.read(address) == 0)
{
serialEvent();
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("System Standby");
lcd.setCursor(0, 1);
lcd.print("Status: OFF");
delay(1000);
}
else if (EEPROM.read(address) == 1)
{
lcd.setCursor(0, 0);
lcd.print("STATUS: ON");
lcd.setCursor(0, 1);
lcd.print("----------------");
}
}
void gsm_init()
{
lcd.print("System booting....");
lcd.setCursor(0, 1);
lcd.print("initiating......");
delay(1500);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Checking Module");
lcd.setCursor(0, 1);
lcd.print("Connectivity....");
```

```
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Module");
lcd.setCursor(0, 1);
lcd.print("Connection: OK");
delay(1500);
eeprom_test();
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("System is Ready");
lcd.setCursor(0, 1);
lcd.print("----------------");
delay(1500);
}
void serialEvent()
{
while (gsm.available())
{
char read_char = gsm.read();
string += read_char;
i++;
if (string[i - 4] == 'R' && string[i - 3] == 'I' && string[i - 2] == 'N' && string[i
- 1] == 'G' )
{
ring = 1;
}
}
}
void output_begin()
{
digitalWrite(red, LOW);
digitalWrite(output,LOW);
digitalWrite(buzzer, HIGH);
delay(2000);
digitalWrite(buzzer, LOW);
digitalWrite(green, HIGH);
if (EEPROM.read(wr_call_add) == 0)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Sending SMS");
lcd.setCursor(0, 1);
lcd.print("Acknowledgement.");
delay(1000);
gsm.println("AT+CMGF=1");
delay(500);
gsm.print("AT+CMGS=");
gsm.print("\"");
```

```
gsm.print(str);
gsm.println("\"");
delay(1000);
gsm.println("Motor is ON.");
delay(100);
gsm.println((char)26);
}
if (EEPROM.read(wr_call_add) == 1)
{
wr_call_val = 0;
EEPROM.write(wr_call_add, wr_call_val);
}
lcd.clear();
}
void eeprom_test()
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Testing EEPROM");
lcd.setCursor(0, 1);
lcd.print("Memory.");
delay(1500);
eeprom_tst = EEPROM.read(address);
eeprom_tst_ack = EEPROM.read(wr_call_add);
EEPROM.write(address, tst_var);
EEPROM.write(wr_call_add, wr_call_tst);
if (EEPROM.read(address) == tst_var && EEPROM.read(wr_call_add) == wr_call_tst)
{
EEPROM.write(address, eeprom_tst);
EEPROM.write(wr_call_add, eeprom_tst_ack);
if (EEPROM.read(address) != eeprom_tst || EEPROM.read(wr_call_add) !=
eeprom_tst_ack)
{
digitalWrite(output,HIGH);
digitalWrite(red, HIGH);
digitalWrite(buzzer, LOW);
delay(2000);
digitalWrite(green, LOW);
eeprom_dead_val = 1;
EEPROM.write(eeprom_dead, eeprom_dead_val);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("EEPROM Test is");
lcd.setCursor(0, 1);
lcd.print("Unsuccessful.");
delay(1500);
}
```

```
else if (EEPROM.read(address) == eeprom_tst && EEPROM.read(wr_call_add) ==
eeprom_tst_ack)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("EEPROM Test is");
lcd.setCursor(0, 1);
lcd.print("Successful.");
delay(1500);
}
}
else if (EEPROM.read(address) != tst_var || EEPROM.read(wr_call_add) != wr_call_tst)
{
digitalWrite(output,HIGH);
digitalWrite(red, HIGH);
digitalWrite(buzzer, LOW);
delay(2000);
digitalWrite(green, LOW);
eeprom_dead_val = 1;
EEPROM.write(eeprom_dead, eeprom_dead_val);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("EEPROM Test is");
lcd.setCursor(0, 1);
lcd.print("Unsuccessful.");
delay(1500);
}
}
void eeprom_sms()
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Sending SMS");
lcd.setCursor(0, 1);
lcd.print("Acknowledgement");
delay(1000);
gsm.println("AT+CMGF=1");
delay(500);
gsm.print("AT+CMGS=");
gsm.print("\"");
gsm.print(str);
gsm.println("\"");
delay(1000);
gsm.println("EEPROM error. System disabled. Please reload the code with new
address.");
delay(100);
gsm.println((char)26);
{
```

```
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("EEPROM Error.");
lcd.setCursor(0, 1);
lcd.print("System Disabled.");
delay(1500);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Reload the code");
lcd.setCursor(0, 1);
lcd.print("with new address");
delay(1500);
}
}
```