AUTOMATIC IRRIGATION WITH DUAL PUMP CONTROLLING SYSTEM

Description:

The Automatic Irrigation system is designed to efficiently manage the watering of plants by monitoring soil moisture levels and water reservior levels It uses an Arduino UNO microcontroller interfaced with a GSM module, a moisture sensor and a ultrasonic sensor. Automatic irrigation with dual pump controlling system works when the tank fill 90% the tank pump motor OFF, otherwise it is in ON condition. When the moisture level is low the watering pump is ON, moisture level is going to 85% and above the watering pump is going to OFF.

Components Required:

- **Arduino Uno:** The central microcontroller responsible for reading sensor data and controlling the pumps.
- **GSM Module (e.g., SIM900D):** Enables the system to send and receive SMS messages for remote control and monitoring.
- **Moisture Sensor:** Measures soil moisture levels to determine when irrigation is needed.
- **Ultrasonic Sensor:** Measures water reservoir levels to prevent overfilling.
- **Relay Modules:** Control the two water pumps based on sensor inputs.
- **Water Pumps:** Distribute water to the plants as needed.
- **LCD Display:** It displays the condition of the two pumps whether it is in ON or OFF condition.

Project Operation:

In this project scenario, we’re utilizing an Arduino UNO board to perform the function of the two pumps. The moisture sensor continuously measures the soil's moisture level. When the moisture level falls below a certain threshold, it indicates that the soil is dry and in need of watering. The ultrasonic sensor monitors the water level in the reservoir or tank. If the water level is low, it suggests that the water source is running out or needs refilling. The microcontroller processes data from the moisture sensor and ultrasonic sensor. It decides whether to activate one or both pumps based on the soil moisture level and water availability. Users can send SMS commands to the system. For example, sending "START" could initiate irrigation, while sending "STOP" could halt it. The system can also send status updates, such as "IRRIGATION STARTED" or "WATER TANK LOW," back to the user. If both pumps are used, they can operate alternately to extend their lifespan. The microcontroller ensures that the pumps are not overworked and that the irrigation process is efficient. The system should have safety features to prevent over-irrigation, such as setting a maximum run-time for the pumps or monitoring the moisture levels to avoid waterlogging.

Block Diagram:

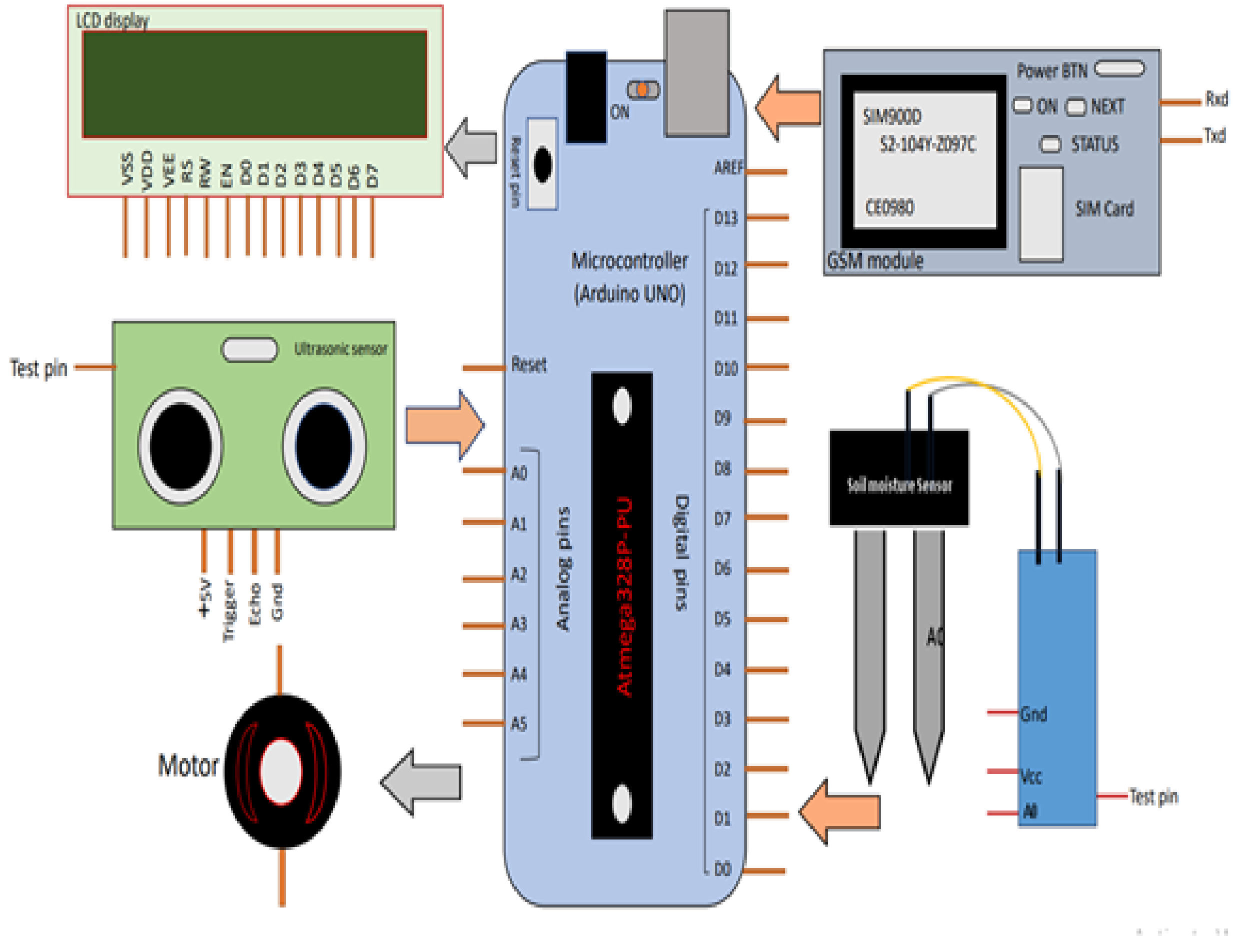
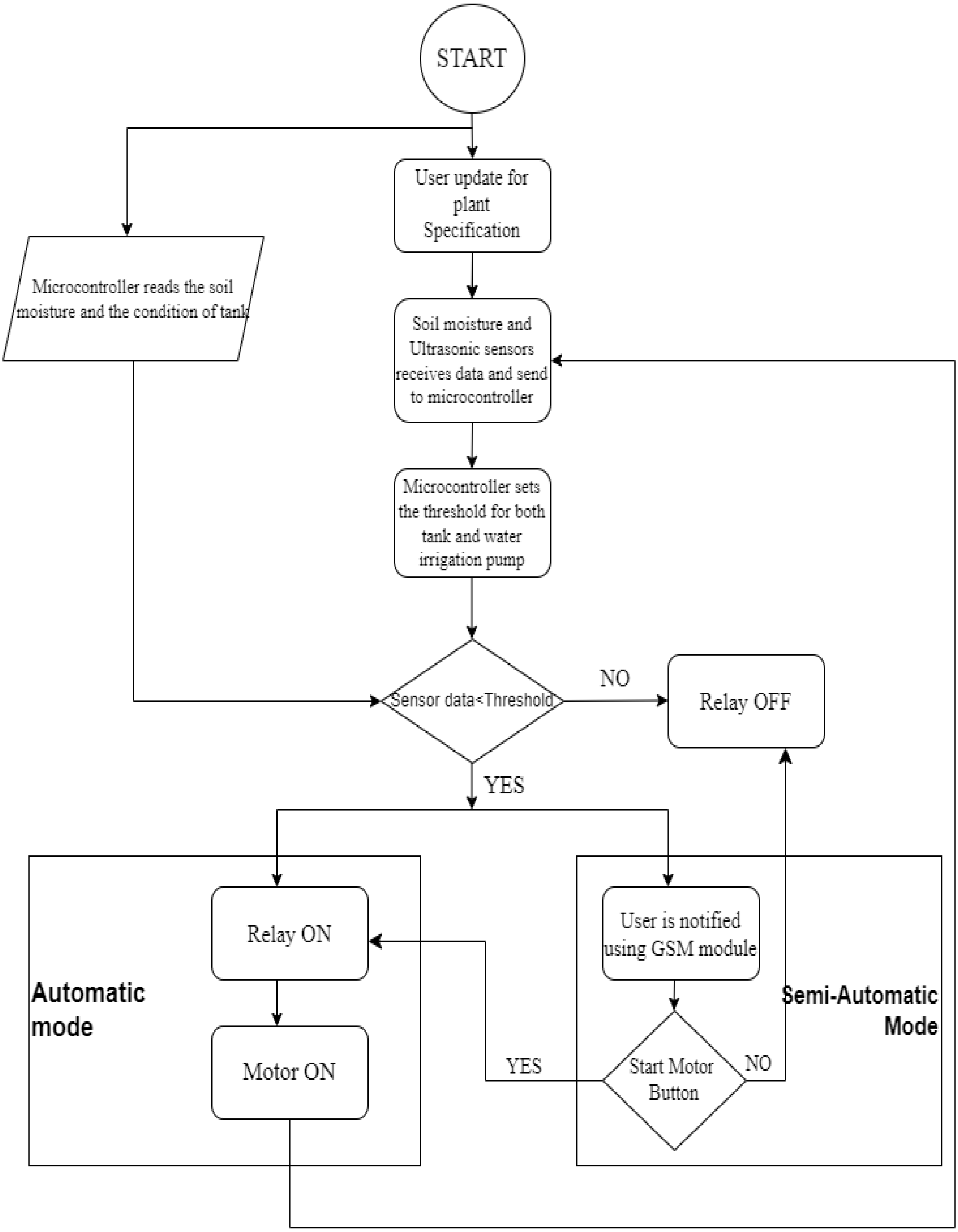


Figure: Block diagram of Automatic irrigation with dual pump controlling system

Input and Output:

Sl.no	Description	Name	Type	Data Direction	Specification	Remarks
1.	Soil moisture sensor Gnd	GND	INP	D1	Digital	Active High
2.	Soil moisture sensor Vcc	VCC	INP	D1	Digital	Active High
3.	Soil moisture sensor TP	RV2	INP	D1	Digital	Active High
4.	Soil moisture sensor A0	A0	INP	D1	Digital	Active High
5.	GSM Rxd	Rxd(1)	INP	D1	Digital	Active High
6.	GSM Txd	Txd(0)	INP	D1	Digital	Active High
7.	Ultrasonic sensor TP	RV3	INP	D1	Digital	Active High
8.	Ultrasonic sensor Vcc	VCC	INP	D1	Digital	Active High
9.	Ultrasonic sensor Gnd	GND	INP	D1	Digital	Active High
10.	Ultrasonic sensor Trigger	Trigger	INP	D1	Digital	Active High
11.	Ultrasonic sensor Echo	Echo	INP	D1	Digital	Active High
12.	LCD Data pin	D4	OUT	D0	Digital	Active High
13.	LCD Data pin	D5	OUT	D0	Digital	Active High
14.	LCD Data pin	D6	OUT	D0	Digital	Active High
15.	LCD Data pin	D7	OUT	D0	Digital	Active High
16.	LCD Data RST	RS	OUT	D0	Digital	Active High
17.	LCD Data EN	EN	OUT	D0	Digital	Active High
18.	LCD Data VSS	VSS	OUT	D0	Digital	Active High
19.	LCD Data VDD	VDD	OUT	D0	Digital	Active High
20.	LCD Data VEE	VEE	OUT	D0	Digital	Active High
21.	Motor	4	OUT	D0	Digital	Active High
22.	Motor	13	OUT	D0	Digital	Active High

Flowchart:


START
User update for plant Specification
Microcontroller reads the soil moisture and the condition of tank
Soil moisture and Ultrasonic sensors receives data and send to microcontroller
Microcontroller sets the threshold for both tank and water irrigation pump
Sensor data<Threshold
NO
Relay OFF
YES
Relay ON
Motor ON
Automatic mode
User is notified using GSM module
Start Motor Button
YES
NO
Semi-Automatic Mode

Source Code:

```
#include<LiquidCrystal.h>
#include <SoftwareSerial.h>

#define echo 9
#define trigger 10
#define tank_pump 4
#define watering_pump 13
#define moisture_sensor A0
long duration;
int distance;
int moisture_value;
int distance_percent;
int moist_percent;
SoftwareSerial SIM900(0,1);
LiquidCrystal lcd(12,11,8,7,6,5);
void setup () {
lcd.begin(20,4);
SIM900.begin(9600);
Serial.begin(9600);
pinMode(echo,INPUT);
pinMode(moisture_sensor,INPUT);
pinMode(trigger,OUTPUT);
digitalWrite(trigger,LOW);
pinMode(watering_pump,OUTPUT);
pinMode(tank_pump,OUTPUT);
```

```
digitalWrite(watering_pump,LOW);
digitalWrite(tank_pump,LOW);
lcd.setCursor(0,1);
lcd.print("AUTOMATIC IRRIGATION"    );
lcd.setCursor(0,2);
lcd.print("USING GSM,SONAR &");
lcd.setCursor(0,3);
lcd.print("MOISTURE SENSOR.....");
delay(500);
lcd.clear();
}
void loop(){
 // LEVEL SENSOR
 digitalWrite(trigger,LOW);
 delayMicroseconds(2);
 digitalWrite(trigger,HIGH);
 delayMicroseconds(10);
 digitalWrite(trigger,LOW);
 duration=pulseIn(echo,HIGH);
 distance=duration*0.017;
 distance_percent=map( distance,0,1023,0,100);
 moisture_value= analogRead(moisture_sensor);
moist_percent=map(moisture_value,0,1023,0,100);
 condition();
}
void sms(){
SIM900.print("AT+CMGF=1\r");
```

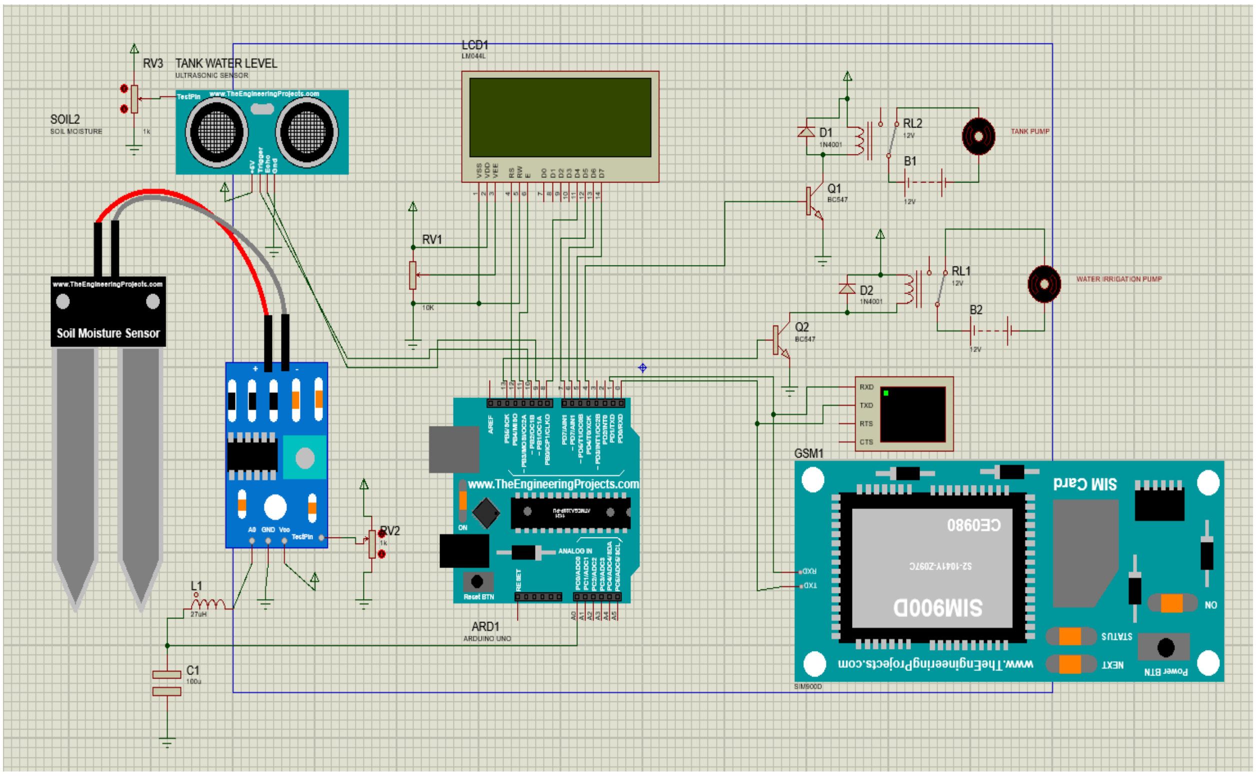
```
delay(200);
 SIM900.println("AT + CMGS = \"+91xxxxxxxxxx\"");// recipient's mobile
number
SIM900.println("WATERING PUMP IS OFF");// message to send
Serial.println("WATERING PUMP IS OFF");
SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
Serial.println((char)26);
//delay(200);
SIM900.println();
}
void sms1(){
SIM900.print("AT+CMGF=1\r");
 SIM900.println("AT + CMGS = \"+91xxxxxxxxx\"");// recipient's mobile
number
SIM900.println("TANK PUMP IS OFF"); // message to send
Serial.println("TANK PUMP IS OFF");
SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
Serial.println((char)26);
//delay(200);
SIM900.println();
}
void sms2(){
SIM900.print("AT+CMGF=1\r");
 SIM900.println("AT + CMGS = \"+91xxxxxxxxxx\"");// recipient's mobile
number
SIM900.println("WATERING PUMP IS ON"); // message to send
Serial.println("WATERING PUMP IS ON");
//delay(200);
```

```
SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
Serial.println((char)26);
//delay(200);
SIM900.println();
}
void sms3(){
SIM900.print("AT+CMGF=1\r");
 delay(200);
 SIM900.println("AT + CMGS = \"+91xxxxxxxxxx\"");// recipient's mobile number
SIM900.println("TANK PUMP IS ON"); // message to send
Serial.println("TANK PUMP IS ON");
//delay(200);
SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26
Serial.println((char)26);
//delay(200);
SIM900.println();
}
void condition(){
if (distance_percent>65 &&moist_percent<85){
LCD_3();
digitalWrite(tank_pump,LOW);
digitalWrite(watering_pump,HIGH);
 sms1();
 sms2();
delay(500);
}
else if (distance_percent<65 &&moist_percent>85)
```

```
{
LCD_2();
digitalWrite(tank_pump,HIGH);
digitalWrite(watering_pump,LOW);
sms3();
sms();
delay(500);
}
else if (distance_percent>65 &&moist_percent>85)
{
LCD_4();
digitalWrite(tank_pump,LOW);
digitalWrite(watering_pump,LOW);
sms1();
sms();
delay(500);
}
else if (distance_percent<65 &&moist_percent<85)
{
LCD_1();
digitalWrite(tank_pump,HIGH);
digitalWrite(watering_pump,HIGH);
sms3();
sms2();
delay(500);
}
}
```

```
  void LCD_1()
  {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("TANK LEVEL=  ");
  lcd.print(distance_percent);
  lcd.print("%");
  lcd.setCursor(0,1);
  lcd.print("MOIST CONTENT= ");
  lcd.print(moist_percent);
  lcd.print("%");
   lcd.setCursor(0,2);
  lcd.print("W-PUMP STATUS  ");
  lcd.print("  ON");
  lcd.setCursor(0,3);
  lcd.print("T-PUMP STATUS  ");
  lcd.print("  ON");
  }
void LCD_2(){
  lcd.clear();
  lcd.setCursor(0,0);
   lcd.print("TANK LEVEL=  ");
  lcd.print(distance_percent);
  lcd.print("%");
  lcd.setCursor(0,1);
  lcd.print("MOIST CONTENT= ");
  lcd.print(moist_percent);
lcd.print("%");
 lcd.setCursor(0,2);
lcd.print("W-PUMP STATUS ");
lcd.print("  OFF");
lcd.setCursor(0,3);
lcd.print("T-PUMP STATUS ");
lcd.print("   ON");
}
void LCD_3(){
lcd.clear();
lcd.setCursor(0,0);
 lcd.print("TANK LEVEL=  ");
lcd.print(distance_percent);
lcd.print("%");
lcd.setCursor(0,1);
lcd.print("MOIST CONTENT= ");
lcd.print(moist_percent);
lcd.print("%");
 lcd.setCursor(0,2);
lcd.print("W-PUMP STATUS ");
lcd.print("  ON");
lcd.setCursor(0,3);
lcd.print("T-PUMP STATUS ");
lcd.print("  OFF");
}
void LCD_4(){
lcd.clear();
lcd.setCursor(0,0);
 lcd.print("TANK LEVEL=  ");
lcd.print(distance_percent);
lcd.print("%");
lcd.setCursor(0,1);
lcd.print("MOIST CONTENT= ");
lcd.print(moist_percent);
lcd.print("%");
 lcd.setCursor(0,2);
lcd.print("W-PUMP STATUS");
lcd.print("  OFF");
lcd.setCursor(0,3);
lcd.print("T-PUMP STATUS");
lcd.print("  OFF");
}
```

Schematic:



RV3
TANK WATER LEVEL
ULTRASONIC SENSOR
SOIL2
SOIL MOISTURE
www.TheEngineeringProjects.com
Soil Moisture Sensor
LCD1
LM044L
RV1
RV2
D1
1N4001
RL2
12V
TANK PUMP
B1
Q1
BC547
D2
1N4001
RL1
WATER IRRIGATION PUMP
B2
Q2
BC547
RXD
TXD
RTS
CTS
GSM1
SIM900D
SIM Card
ARD1
ARDUINO UNO
L1
27uH
C1
100u