

Automated irrigation system using relay module and ir remote control

This project aims to develop an irrigation system in two modes: manual, using an IR remote-controlled sprinkler, with more modes of irrigation controlled by a remote; automatic, the relay been activated by multiple sensors like, soil humidity, temperature and air humidity.

- Objectives

- *Learn to use ir remote control from distance.
- *Automation an irrigation system, based on extern factors.
- *Using multiple sensors, relay, and water pump.
- *Increase knowledge about Arduino, and microcontrollers.

- Components

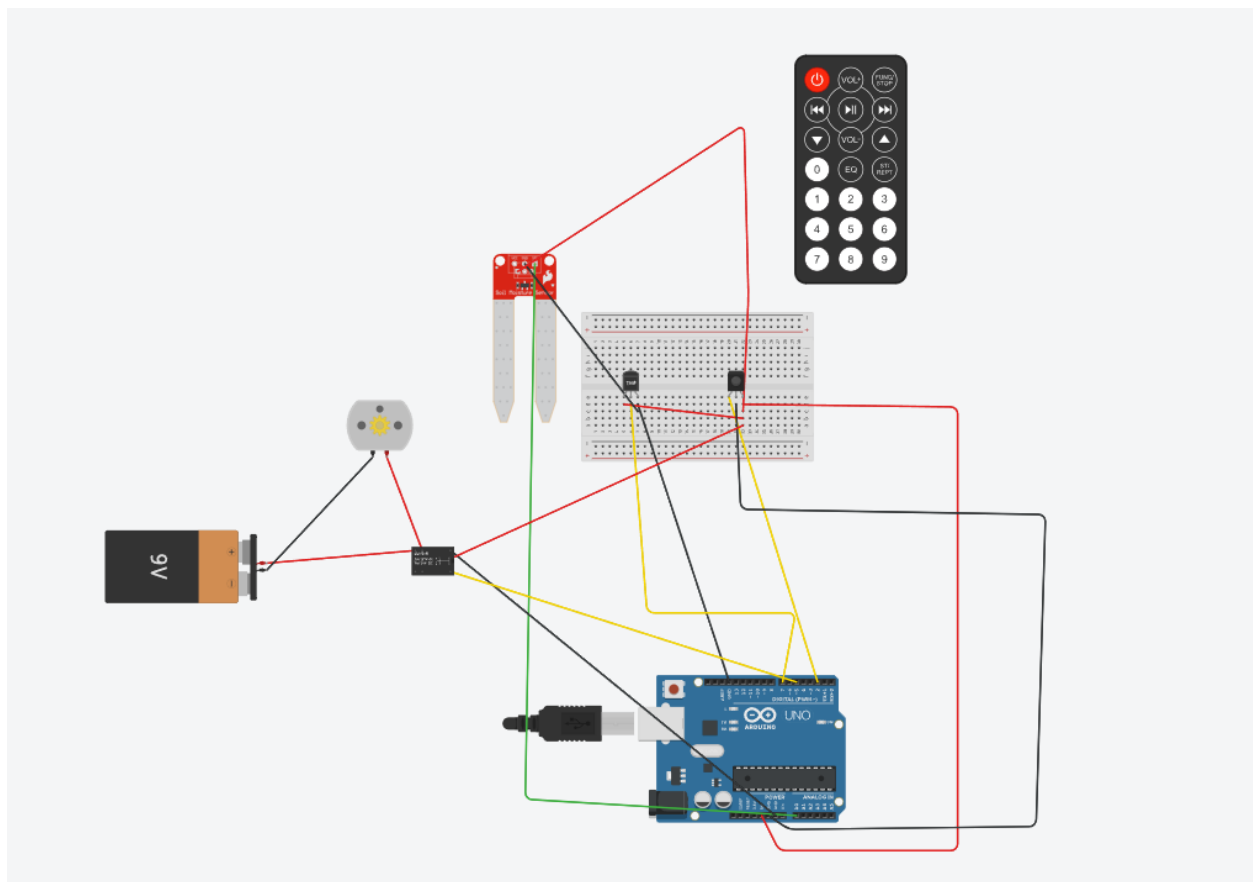
- Arduino Mega 2560
- Breadboard
- Humidity soil sensor
- IR remote + Receiver
- External 9V source
- Submergence water pump
- Temperature and humidity (DHT11)
- More cables
- 5V relay module

- Circuit diagram

In thinkercard water pump doesn't exist,so I used in diagram a 5V DC motor,which has same principle of functioning.

Same for DHT11 sensor,I used a temperature sensor.

Because in thinkercard,relay module doesn't exist,I used one simple relay.



- Code

Setup all sensors,relay module,and water pump;

```

void setup() {
  Serial.begin(115200);
  IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK);
  Serial.println("Pornit receptor IR...");

  pinMode(pinSenzor, INPUT);
  pinMode(releu, OUTPUT);
  digitalWrite(releu, HIGH);

  dht.begin();
}

```

Read all factors, else we can't continue to run the project

```

uint32_t cod = IrReceiver.decodedIRData.decodedRawData;

//start-stop for relay
if (cod == 0x184C) {
  digitalWrite(releu, HIGH);
} else if (cod == 0x104C) {
  digitalWrite(releu, LOW);
}

```

Manual mode:

```

//the relay is activated for 5 seconds, after it's turn off
if (cod == 0x184D || cod == 0x104D) {
  digitalWrite(releu, LOW);
  delay(5000);
  digitalWrite(releu, HIGH);
  delay(2000);
}

```

```

//the relay is intermittent
if (cod == 0x1875 || cod == 0x1075) {
    digitalWrite(releu, LOW);
    delay(2000);
    digitalWrite(releu, HIGH);
    delay(2000);
}

manual = true;
timer = millis();

IrReceiver.resume();

```

For more efficiency, automatic mode is priority, so for manual mode I put a timer to run after I press a button, after that automatic mode is reactivated.

```

//automatic mode for irrigation
if(manual && millis() - timer > 100000)
{
    Serial.println("Automatic mode for irrigation: ");
}

```

Automatic mode:

In this mode, relay module activate water pump based on more factors like soil humidity, air temperature, and air humidity. After these elements I make some smart irrigation programs to activate relay on different conditions of weather.

```

//mode one,for high temperature,and dry soil and moderate humidity
if (temperatura > 30 && valoare >= 750 && umiditate <= 40) {
    digitalWrite(releu, LOW);
    delay(15000);
    digitalWrite(releu, HIGH);
    delay(500);
}

```

```

//mode two
if (temperatura >= 20 && temperatura <= 30 && valoare > 750 && umiditate >= 40 && umiditate < 60) {
    digitalWrite(releu, LOW);
    delay(10000);
    digitalWrite(releu, HIGH);
    delay(500);
}

```

```

//mode 3
if(temperatura >=30 && valoare >= 650 && valoare <= 750 && umiditate <= 40)
{
    digitalWrite(releu, LOW);
    delay(5000);
    digitalWrite(releu, HIGH);
    delay(500);
}

```

```

//mode four,the soil is very wet,so we don't need extra factors
if(valoare < 350)
{
    digitalWrite(releu,HIGH);
    delay(500);
}

```

- Results and tests

The program run very good,more soft and hard problems was solved.The tests demonstrated stable power supply.

- Conclusions and improvements

The project demonstrated hardware-software integration in a simple embeeded project,very useful for future Ir remote project,automation projects,and learn to work with multiple sensors.

Improvements:

- *Control the project with and application
- *Create a irrigation mode based on photoresistor
- *Water sensor for plant to prevent overflowing.