GOVERNMENT COLLEGE OF ENGINEERING

ERODE

(FORMERLY KNOWN AS IRTT)

BE., ELECTRONICS AND COMMUNICATION ENGINEERING

TECHNOLOGY NAME **: INTERNET OF THINGS**

ENVIRONMENTAL MONITORING

AUTOMATIC SOIL IRRIGATION SYSTEM

NAME OF THE STUDENT	REGISTER NUMBER
TEAM MEMBER	
M.DEVAKI	731121106004

PHASE 5

Under the mentor of

Dr.M.POONGOTHAI ME, Phd.,

DEPARTMENT OF INFORMATION TECHNOLOGY(IT)

Government College of Engineering, Erode

Near Vasavi College Post,

TamilNadu – 638316.

Affiliated to Anna University , Chennai.

AUTOMATIC SOIL IRRIGATION SYSTEM

INTRODUCTION:

Agricultural lands require water in certain levels.If Water is under minimum level, the plants may die due to the absence of proper watering. In **Automatic Soil Irrigation System**, the **Arduino IDE** will irrigate the plants automatically and keep the plants healthy even when Man power is out of the cultivation lands for weeks or months. It reduces man power and Controls the wastage of water. By using several components of LED, Microcontroller ESP8266 and Sensors project work gets easier.

OBJECTIVES:

The objectives of an Automatic Soil Irrigation System are:

Efficient Water Management:

Ensure efficient and optimal use of water resources by delivering the right amount of water to the soil based on its moisture levels, thereby preventing over-irrigation or under-irrigation.

Crop Health and Yield Improvement:

Maintain consistent soil moisture levels to promote healthy plant growth and increase crop yields.

Water Conservation:

Conserve water resources by avoiding wasteful irrigation practices, which is particularly important in regions facing water scarcity.

Labor and Time Savings:

Automate the irrigation process to reduce the need for manual labor and save time for farmers.

Remote Monitoring and Control:

Allow farmers to monitor and control the irrigation system remotely, enabling timely adjustments and interventions as needed.

Minimize Environmental Impact:

Minimize the environmental impact of agriculture by reducing runoff, leaching, and the use of fertilizers and pesticides through controlled irrigation.

Data Collection and Analysis:

Gather data on soil moisture, weather conditions, and crop health, which can be used for analysis and decision-making to optimize irrigation strategies.

Energy Efficiency:

Implement energy-efficient mechanisms for pumping and distributing water.

Scalability and Adaptability:

Design the system to be scalable and adaptable to different crop types, soil conditions, and environmental factors.

Cost Savings:

Reduce operational costs associated with irrigation by automating and optimizing the process.

Environmental Sustainability:

Promote sustainable agriculture practices by reducing the environmental footprint of irrigation.

User-Friendly Interface:

Provide an easy-to-use interface for farmers to set preferences and receive alerts or recommendations.

An automatic soil irrigation system aims to address these objectives to improve the efficiency, productivity, and sustainability of agriculture.

COMPONENTS REQUIRED:

- Embedded C Language
- Arduino IDE
- ESP8266 microcontroller
- DTH11 Sensor
- Soil moisture sensor
- OLED Display
- Relay
- Pump
- Wi fi
- Thing speak sensor server
- 5v relay module
- Jumper wire

IOT DEVICE SETUP :

ESP8266 Microcontroller :

ESP8266 is the main component used in our system .It controls the sensors to start the process by sending commands to sensors. ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

Arduino IDE:

Arduino IDE used to write software programs to implement the functions in hardware components.It is functioned by uploading the c program to arduino by using pc .

Soil moisture sensor

Soil moisture sensor is a type of sensor that senses the moisture of soil and gives information about the moisture content level to microcontroller. The soil moisture sensor (SMS) is a sensor connected to an irrigation system controller that measures soil moisture content in the active root zone before each scheduled irrigation event and bypasses the cycle if soil moisture is above a user- defined set point.

DHT11 sensor

DHT11Sensor senses the Temperature and Humidity of atmosphere cultivation land. DHT-11 **Digital Temperature And Humidity Sensor** is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed).

OLED DISPLAY

OLED is connected to display the amount of Soil moisture level, Temperature and Humidity. OLED stands for **organic light-emitting diode** and is one of the newer technologies available for TVs and other digital displays. OLED technology is called an "emissive" technology because it uses millions of pixels that emit their own light rather than relying on a separate backlight.

Pump

Water pump will starts working as on and off conditions so water is poured to cultivation lands. Water pumps are a type of impulse turbine that works using a spinning wheel to convert potential energy from pressure into kinetic energy. The water is pumped from a lower elevation to a storage area, where it is collected until it's needed.

Wi fi

Wifi is connected with a help of router to make it as IOT massively. Wi-Fi is a wireless networking technology that uses radio waves to provide wireless high-speed Internet access. A common misconception is that the term Wi-Fi is short for "wireless fidelity," however Wi-Fi is a trademarked phrase that refers to IEEE 802.11x standards.

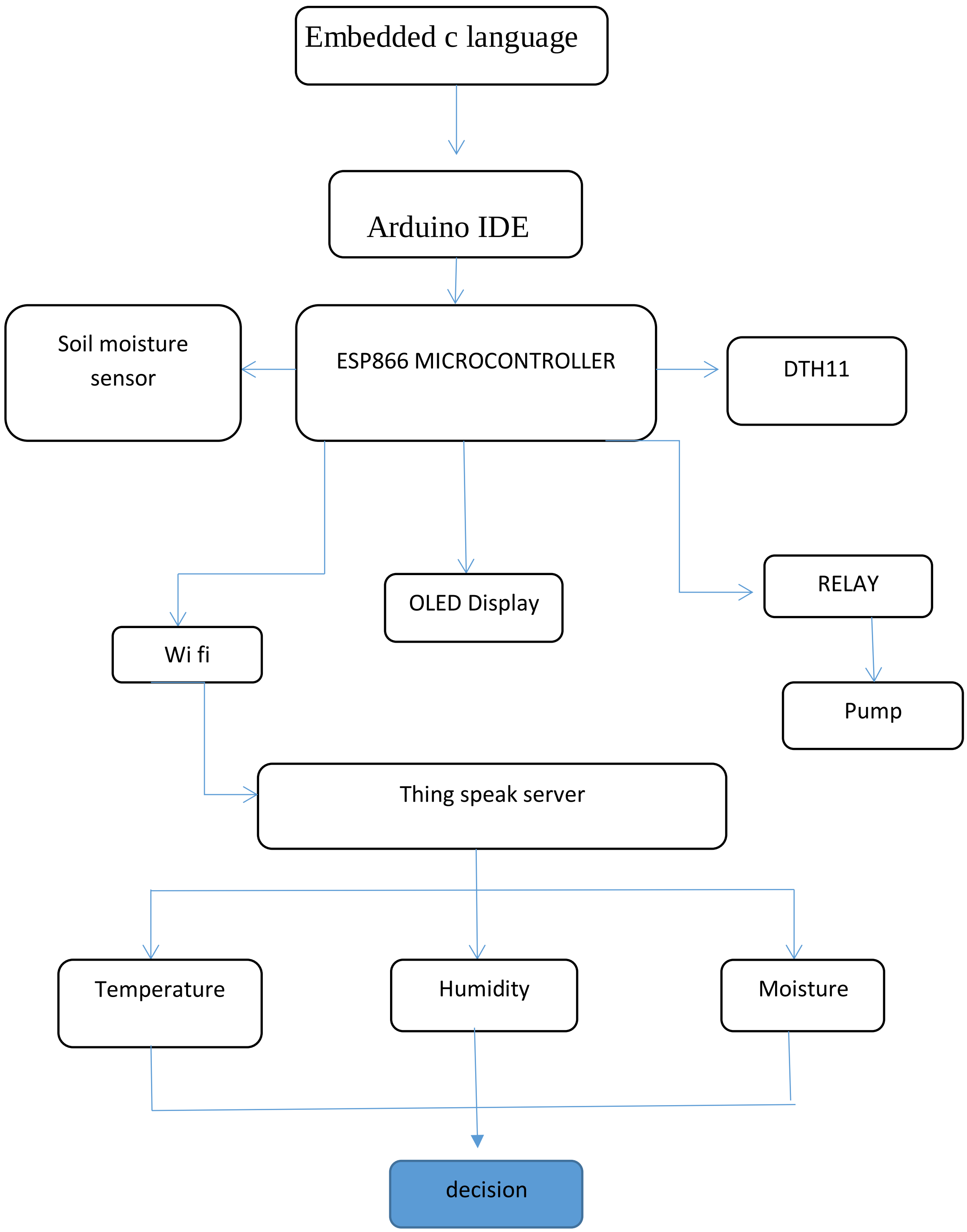
Thing speak server

Thing speak server allows to aggregate, visualize and analyze live data stream in the cloud . ThingSpeak is an IoT analytics platform service that allows user to aggregate, visualize, and analyze live data streams in the cloud. User can send data to ThingSpeak™ from their devices, create instant visualizations of live data, and send alerts using web services like Twitter® and Twilio®.

PLATFORM DEVELOPMENT:

ThingSpeak server is used for Automatic Soil Irrigation System. Login for the process of functioning of the system. Channel is require for sending and receiving information so that new channel is created .In this channel, Four fields are used . They are Moisture, Humidity, Temperature, Motor status. After this channel will be saved. Chart will be created on the basis of channel. Motor status is indicated by an LED, when soil moisture level is less than 30% Motor will ON and Water will be poured into the cultivation land . If the soil moisture level is above 30% Motor is in its OFF condition. API key is used for the codes. Working of the system is by declaring nodemc 1.0 , port is selected for execution of program. Specific devices are functioned by declaring the components. Hotspot name is entered with password. Checking these steps program is uploaded in the ThingSpeak server with connected devices.

FLOW CHART



CODE IMPLEMENTATION:

```
#include <ESP8266WiFi.h>

#include <SPI.h>

#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <DHT.h>  // Including library for dht

#include <Fonts/FreeSerif9pt7b.h>  // Add a custom font

#include <Fonts/FreeMono9pt7b.h>  // Add a custom font

#define SCREEN_WIDTH 128 // OLED display width, in pixels

#define SCREEN_HEIGHT 64 // OLED display height, in pixels

#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)


#define DHTPIN D4          //pin where the dht11 is connected

DHT dht(DHTPIN, DHT11);


String apiKey = "*************"; // Enter your Write API key from ThingSpeak

const char *ssid = "***********";     // replace with your wifi ssid and wpa2 key

const char *pass = "***********";

const char* server = "api.thingspeak.com";


Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);


const int AirValue = 790;   //you need to replace this value with Value_1

const int WaterValue = 390;  //you need to replace this value with Value_2
```

```
const int SensorPin = A0;
int soilMoistureValue = 0;
int soilmoisturepercent=0;
int relaypin = D5;
WiFiClient client;
void setup()
{
  Serial.begin(115200); // open serial port, set the baud rate to 9600 bps
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //initialize with the I2C addr 0x3C (128x64)
  display.clearDisplay();
  pinMode(relaypin, OUTPUT);
   dht.begin();
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
    Serial.println("");
    Serial.println("WiFi connected");
    delay(4000);
}
```

```
void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  Serial.print("Humidity: ");
  Serial.print(h,0);
  Serial.println("%");
  Serial.print("Temperature: ");
  Serial.print(t,0);
  Serial.println("%");
  delay(100);
  soilMoistureValue = analogRead(SensorPin);  //put Sensor insert into soil
  //Serial.println(soilMoistureValue);
  soilmoisturepercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);
if(soilmoisturepercent >= 100)
{
  Serial.print("soilmoisture: ");
  Serial.println("100%");

  display.setCursor(0,20);  //oled display
  display.setTextSize(0);
  display.setFont(&FreeMono9pt7b);  // Set a custom font
  display.setTextColor(WHITE);
  display.print("Moist:");
```

```
  display.setTextSize(0);
  display.print("100");
 display.println("%");
  display.setCursor(0,40);  //oled display
  display.setTextSize(0);
  display.print("Humid:");
  display.setTextSize(0);
  display.print(h,0);
  display.println("%");
  display.setCursor(0,60);  //oled display
  display.setTextSize(0);
  display.print("Temp:");
  display.setTextSize(0);
  display.print(t,0);
  display.println("*C");
  display.display();

  delay(100);
  display.clearDisplay();
}


else if(soilmoisturepercent <=0)
{
  Serial.print("soilmoisture: ");
```

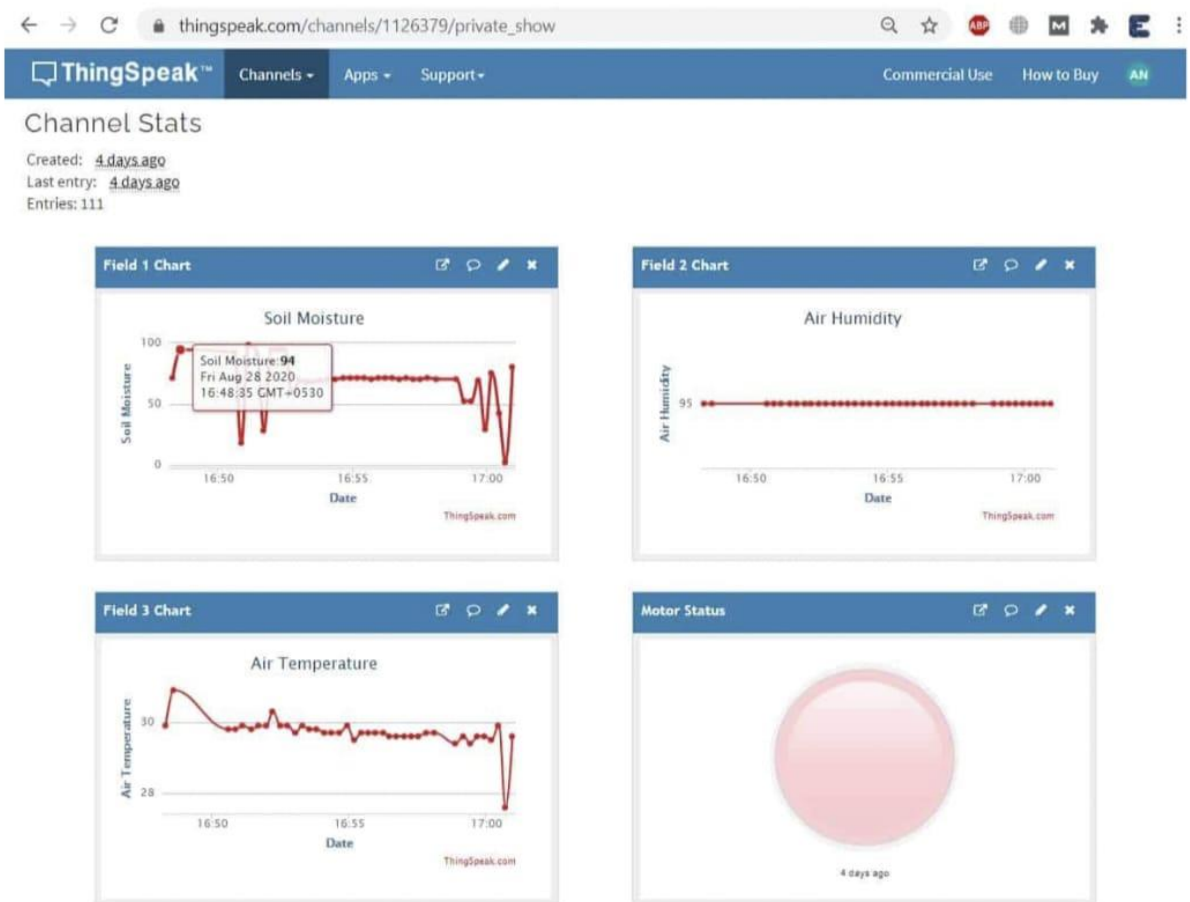
```
Serial.println("0%");

display.setCursor(0,20);  //oled display
display.setTextSize(0);
display.setFont(&FreeMono9pt7b);  // Set a custom font
display.setTextColor(WHITE);
display.print("Moist:");
display.setTextSize(0);
display.print("0");
display.println("%");
display.setCursor(0,40);  //oled display
display.setTextSize(0);
display.print("Humid:");
display.setTextSize(0);
display.print(h,0);
display.println("%");
display.setCursor(0,60);  //oled display
display.setTextSize(0);
display.print("Temp:");
display.setTextSize(0);
display.print(t,0);
display.println("*C");
display.display();

delay(100);
```

```
  display.clearDisplay();
}



else if(soilmoisturepercent >=0 && soilmoisturepercent <= 100)
{
  Serial.print("soilmoisture : ");
  Serial.print(soilmoisturepercent);
  Serial.println("%");

  display.setCursor(0,20);  //oled display
  display.setTextSize(0);
  display.setFont(&FreeMono9pt7b);  // Set a custom font
  display.setTextColor(WHITE);
  display.print("Moist:");
  display.setTextSize(0);
  display.print(soilmoisturepercent);
  display.println("%");
  display.setCursor(0,40);  //oled display
  display.setTextSize(0);
  display.print("Humid:");
  display.setTextSize(0);
  display.print(h,0);
  display.println("%");
  display.setCursor(0,60);  //oled display
```

```
  display.setTextSize(0);
  display.print("Temp:");
  display.setTextSize(0);
  display.print(t,0);
  display.println("*C");
  display.display();

  delay(100);
  display.clearDisplay();
}

  if(soilmoisturepercent <= 49)
  {
    digitalWrite(relaypin, LOW);
    Serial.println("Motor is ON");
   Serial.println("-------------------");
  }
  else if (soilmoisturepercent >49 && soilmoisturepercent <= 100)
  {
    digitalWrite(relaypin, HIGH);
    Serial.println("Motor is OFF");
   Serial.println("-------------------");
  }

  if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com
```

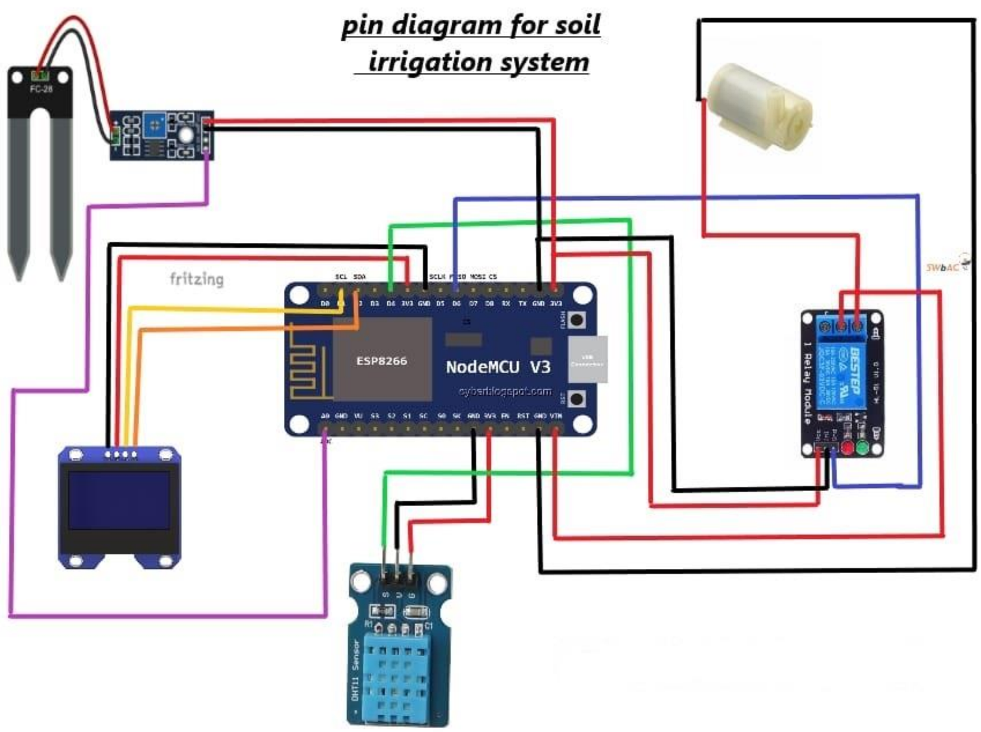
```
  {
    String postStr = apiKey;
      postStr += "&field1=";
      postStr += String(soilmoisturepercent);
      postStr += "&field2=";
      postStr += String(h);
      postStr += "&field3=";
      postStr += String(t);
      postStr += "&field4=";
      postStr += String(relaypin);
      postStr += "\r\n\r\n\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);

  }
    client.stop();
}
```

OUTPUT:



The above output shows the live data of temperature,moisture level,humidity and motor status.

CIRCUIT DIAGRAM:



CONCLUSION:

An automatic soil irrigation system in IoT (Internet of Things) offers numerous advantages for efficient and smart management of agricultural irrigation. In conclusion, it enhances precision and conserves water resources by:

Soil Moisture Monitoring: IoT-based sensors continuously monitor soil moisture levels, ensuring that plants receive the right amount of water precisely when needed. This prevents overwatering or underwatering.

Remote Control and Automation: The system allows farmers to remotely control irrigation, adjust schedules, and respond to changing conditions, reducing the need for manual labor and increasing operational efficiency.

Data Analysis and Optimization: Collected data on soil moisture, weather conditions, and plant requirements can be analyzed to optimize irrigation strategies, leading to improved crop yields and resource conservation.

Water Conservation: By delivering water only when necessary, the system conserves water resources, which is particularly critical in regions with water scarcity.

Savings: Automatic irrigation systems reduce operational costs, labor, and water expenses, contributing to increased profitability for farmers.

In conclusion, an IoT-based automatic soil irrigation system improves agricultural practices by enhancing precision, reducing resource consumption, and optimizing crop growth, ultimately benefiting both farmers and the environment.