Activity 12. EcoFarm

Project carried out and documented by:

- Ivan Sara Ioana, 10th grade D, mathematics-computer science specialization
- Voicu Natașa Elena, grade XI-a D, specialization in mathematics and computer science, National Pedagogical College "Constantin Cantacuzino" Târgoviște

Content

1.Introduction

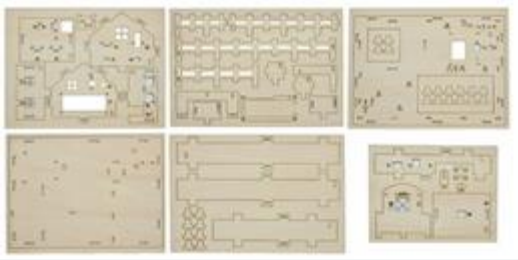
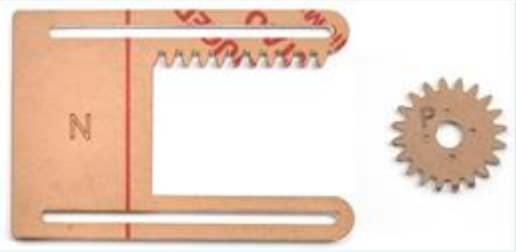
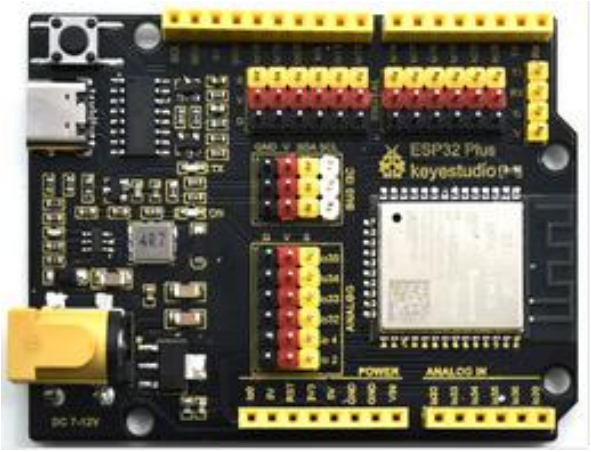
Project description

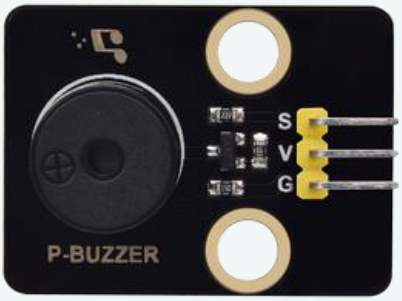
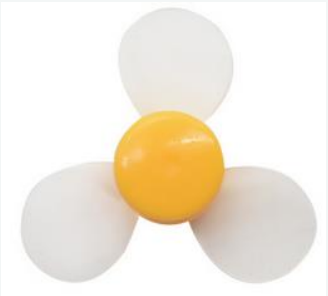
Based on the ESP32 Internet of Things, this Smart Farm kit integrates multiple sensors, realizing automatic and wireless manipulation as well as intellectual management. It is characterized as ***comprehensive***, due to the wide range of sensors it contains, ***innovative***, cultivates technological innovation and the ability to solve problems in agriculture and ***interactive***, having a multitude of cases and experimental practices.

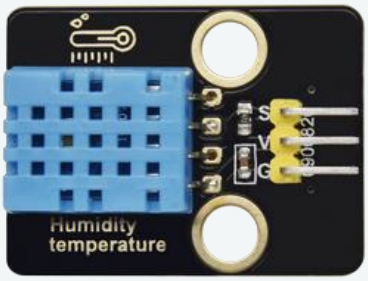
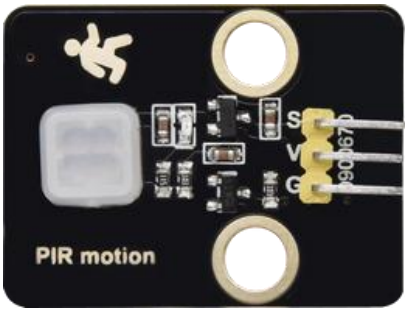
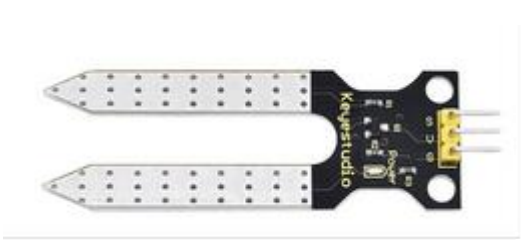
Project objectives

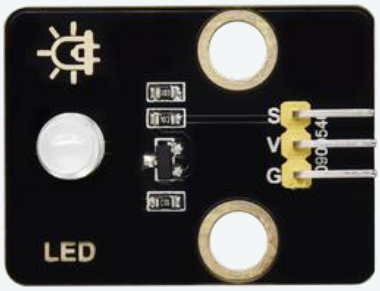
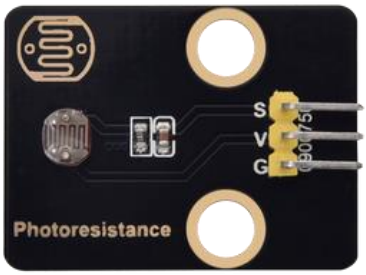
- mastering and acquiring knowledge of various sensors and wireless communication;
- in-depth understanding of how each sensor works and how it collects and processes data;
- stimulates learning enthusiasm by putting sensor elements into practice while creating a practical and easy-to-handle design.

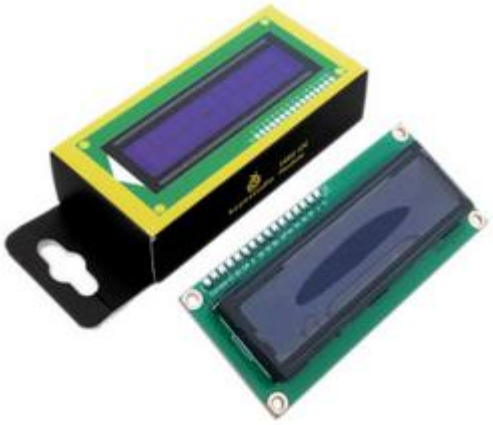
2.Required components

No.	Image	Name / Description	Quantity
1		Brick Wood Board	6 different
2		Patterned Brick Wood Board	1
3		Acrylic plate (a plate wrapped in a transparent plastic, with increased resistance and easy to work with adhesives or different solvents)	1
4		ESP32 PLUS development board (printed circuit board with a microcontroller or microprocessor mounted on it, with different metal pins)	1

No.	Image	Name / Description	Quantity
5		Button Mode (an actual button connected to the development board)	1
6		Button cover (attached to the button module so that the user can press it)	1
7		Passive buzzer (a device that generates a sound when an electric current passes through it)	1
8		130 Motor (a small motor which, if it receives a voltage between 3V and 6V, operates by rotating the intended element from its extremity)	1
9		Fan	1

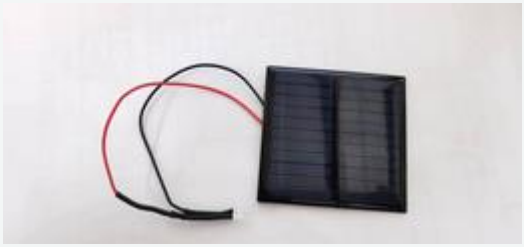
No.	Image	Name / Description	Quantity
10		Steam sensor (an analog sensor; when the moisture on its surface increases, the electrical voltage coming out will increase)	1
11		DHT11 temperature and humidity sensor (a device that converts temperature and humidity into electrical impulses in order to measure their values)	1
12		PIR motion sensor (used in motion detectors, such as self-triggering lighting devices and protection systems that measure devices that emit infrared light in their field of vision)	1
13		5V relay module (an automatic switch that is commonly used in an automatic control circuit and to control a high current using a low current signal)	1
14		Soil moisture sensor (measures and approximates the amount of water in the soil)	1

No.	Image	Name / Description	Quantity
15		Water level sensor (measures the level of water in a container)	1
16		SR01 V3 Ultrasonic Module (detects objects using ultrasonic waves, which are transmitted and received with a single piezoelectric disc)	1
17		White LED module (turns on/off a white LED when connected to ARDUINO)	1
18		Photoresistor (indicates the presence/absence of light and/or measures the intensity of light)	1
19		9G 180° servo (a motor that can rotate at a certain angle depending on the signal received)	1

No.	Image	Name / Description	Quantity
20		I2C 1602 LCD screen (a type of flat-panel display that uses liquid crystals in its main form of operation)	1
21		6-slot AA battery holder	1
22		DC 3V water pump with 15 cm wire for panel (transports water from one place to another, using a wire with pins/connectors at both ends)	1
23		USB Type-C Cable	1

No.	Image	Name / Description	Quantity
24		Self-tapping screw M1.4*6MM	4
25		M3 Lock Nut	4
26		Round Head Screw M4*8MM	18
27		M3*6MM Round Head Screw	8
28		M3*10MM Round Head Screw	16
29		M2*12MM Round Head Screw	2
30		M4 nut	18
31		M3 Nut	12

No.	Image	Name / Description	Quantity
32		M2 Nut	2
33		Double Bore Copper Pole M3*10MM	4
34		15 cm DuPont 3P FF wires (connected, 26 AWG, black-red-yellow) (DuPont wires are used to connect hardware such as sensors, Arduino boards, and breadboards)	4
35		20 cm DuPont 3P FF wires (connected, 26 AWG, black-red-yellow)	5
36		25 cm DuPont 3P FF Wires (Connected, 26 AWG, Black-Red-Yellow)	1
37		DuPont 20cm 4P FF Wires (Connected, 26AWG, Black-Red-Blue-Green)	1
38		20 cm DuPont 4P FF Wires (Split, 26 AWG, Black-Green-Blue-Red)	1
39		DuPont 20cm 4P FF Wires (Split, 26AWG, Black-Red-Blue-Green)	1
40		DuPont MF 15CM 40P Yarn	2
41		Phillips screwdriver 3.0*40MM	1
42		Notched screwdriver	1

No.	Image	Name / Description	Quantity
43		Plastic box (containers used at the water pump where the water will be stored)	2
44		Water Pump Pipe (6 cm)	1
45		Solar panel (collects clean renewable energy in the form of sunlight and converts this light into electrical energy that can then be used to provide power for electrical charges)	1
46		Stickers	1
47		Cross wrench (used for loosening and tightening nuts)	1

3. Working mode

a) Working parameters

Working Voltage: 5V

Battery voltage: 9V

Maximum Output Current: ≤1.5A

Maximum Power Consumption: ≤10W

WiFi Working Principle: 2.4G

Bluetooth Protocol: v4.2 Full Standard

Working temperature: -10°C~+65°C

Parameters and specifications of ESP32 development board

Voltage: 3.3V-5V

Current output: 1.2 A (maximum)

Maximum Output Power: 10W

Working Temperature: -10°C~50°C

Size: 69*54*14.5mm

Weight: 25.5 g

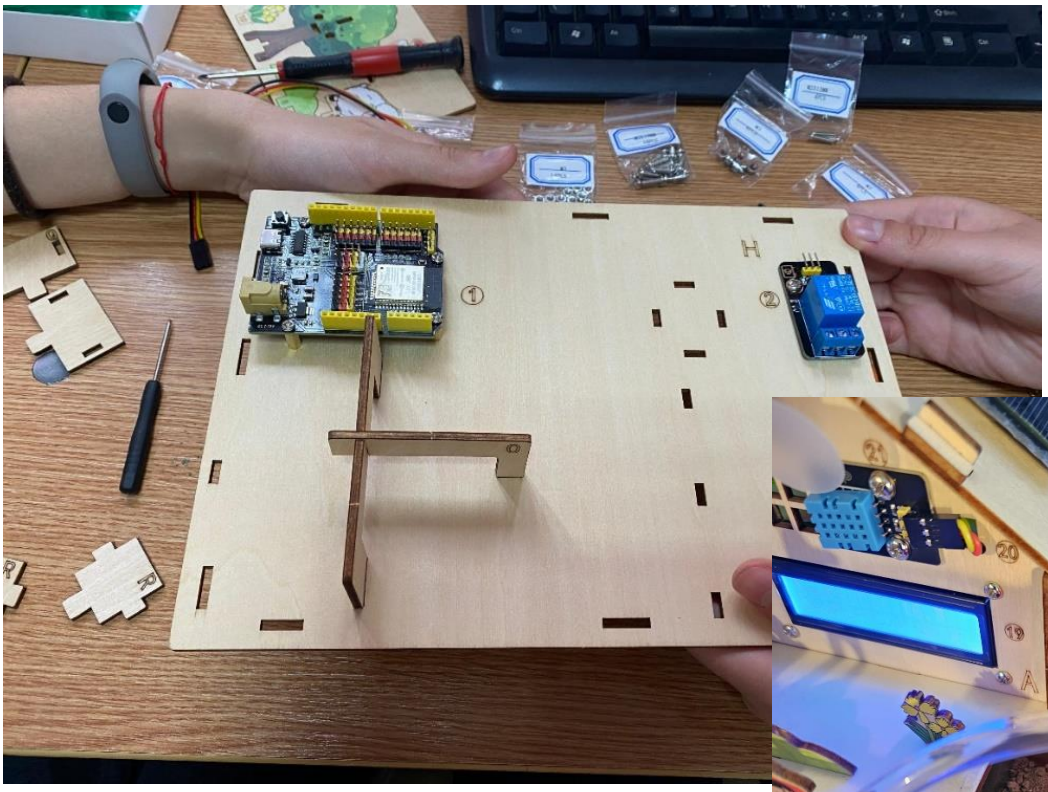
Skeleton assembly

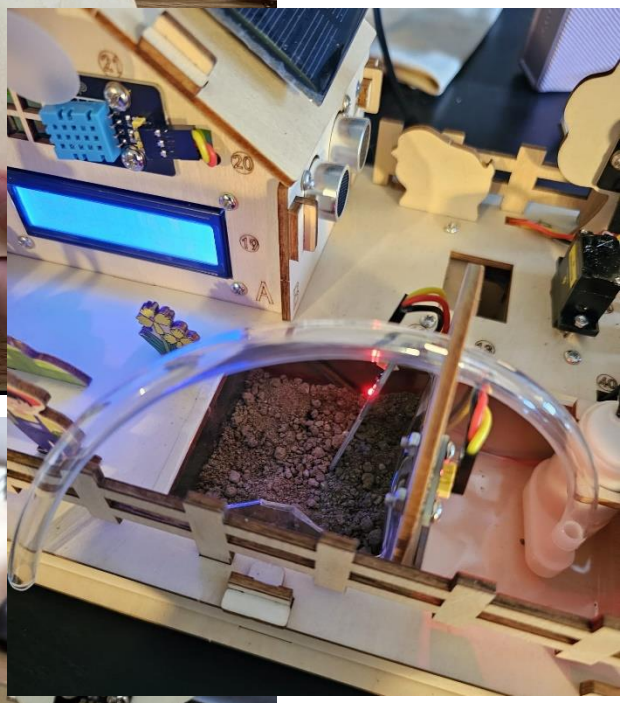
The assembly of the skeleton of the smart farm is done by carefully following each indication provided in the tutorial made available by the developer on the official website https://docs.keyestudio.com/projects/KS0567/en/latest/wiki/Arduino/arduino.html#assembling .

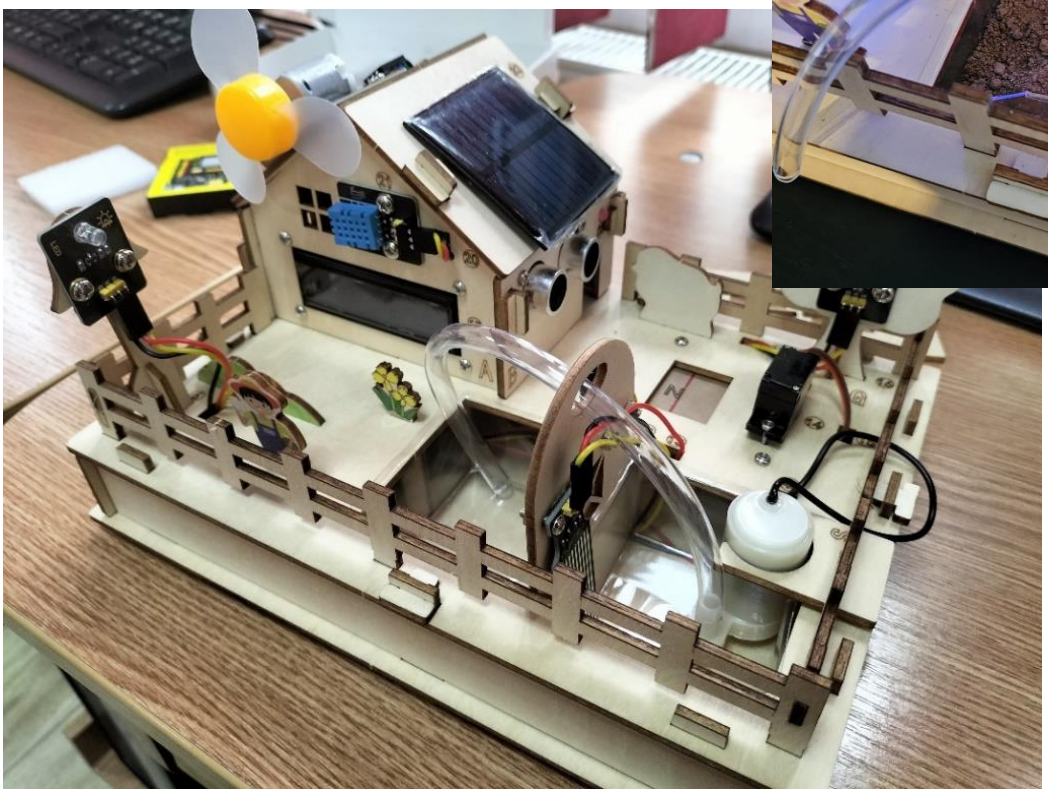
When opening the kit, make sure that no items are missing by consulting the list of components inside the box or the list attached above. The elements, taken out of the box, are individually packaged, as follows:

Avoid mixing nuts or screws with each other or exposing them to a surface where you can easily lose them, as they are small in size. Recommendation: place them in a container from which you can easily take them and use the tools in the kit (screwdrivers, cross wrench) that are intended for these types of screws.

Images during assembly







Wiring (Wiring)

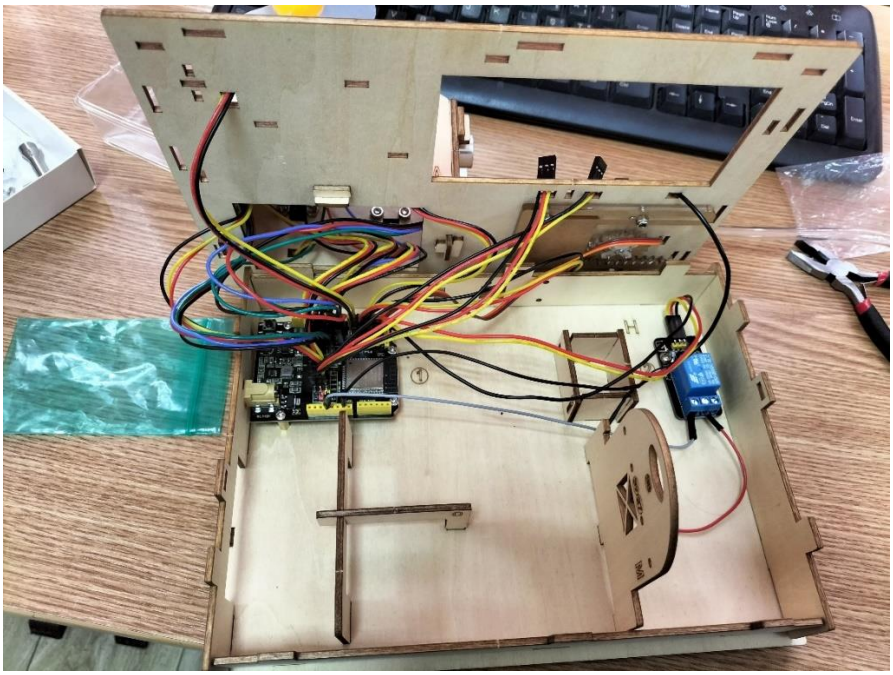
Required elements:

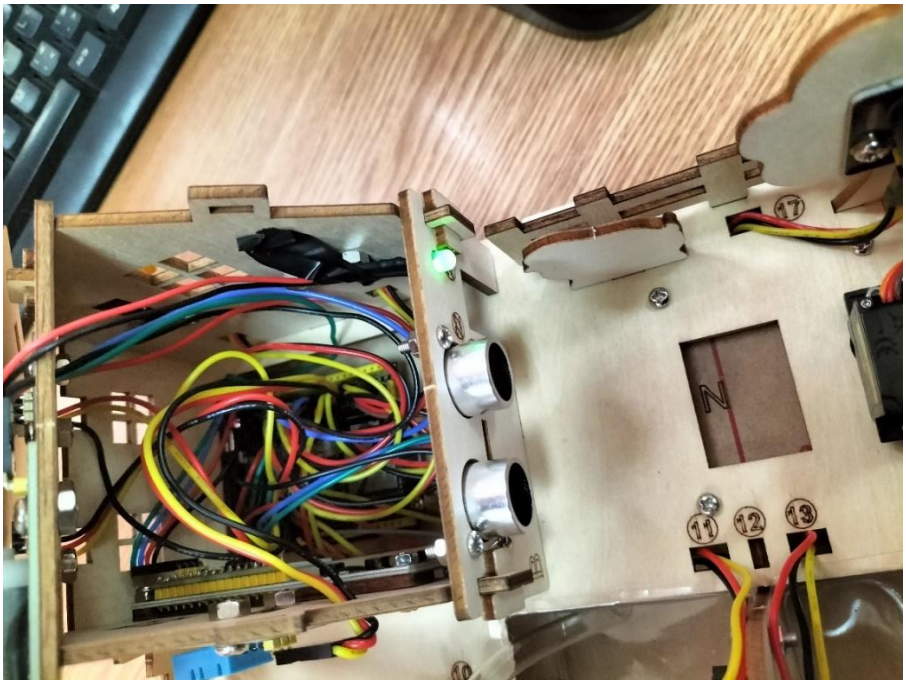
No.	Components	Hairs	Pins for ESP32 board
1	Fan	4 pins, **divided** black-red-blue-green	io18(IN-) | io19(IN+)
2	PIR Motion Sensor	3 pins 15 cm	io23
3	Button	3 pins 15 cm	io5
4	Ultrasonic Module	4 pins, **divided** black-green-blue-red	D12(TRIG) D13(ECHO)
5	LCD 1602	4-pin, **connected**	I2C
6	Temperature and humidity sensor	3 pins 20 cm	I17
7	Steam Sensor	3 pins 15 cm	io35
8	Photoresistor	3 pins 15 cm	io34
9	Servo	–	io26
10	Buzzer	3 pins 20 cm	io16
11	LED	3 pins 20 cm	io27
12	Water level sensor	3 pins 25 cm	io33
13	Soil moisture sensor	3 pins 20 cm	io32
14	Water Pump	3 pins 20 cm	io25

For this part, it is essential to be executed correctly because it gives "life" to the smart farm. By connecting each wire attached to the sensors to the development board in the indicated places, the sensors will receive the electrical current necessary for their operation, thus doing what they are programmed to do.

For the coupling of each sensor to the board, in the tutorial, in the section dedicated to this particular type of assembly, a circuit diagram is drawn, indicating where the wire must be coupled and what components are needed to make it.

In the end, all the cables attached inside the farm will look like this:





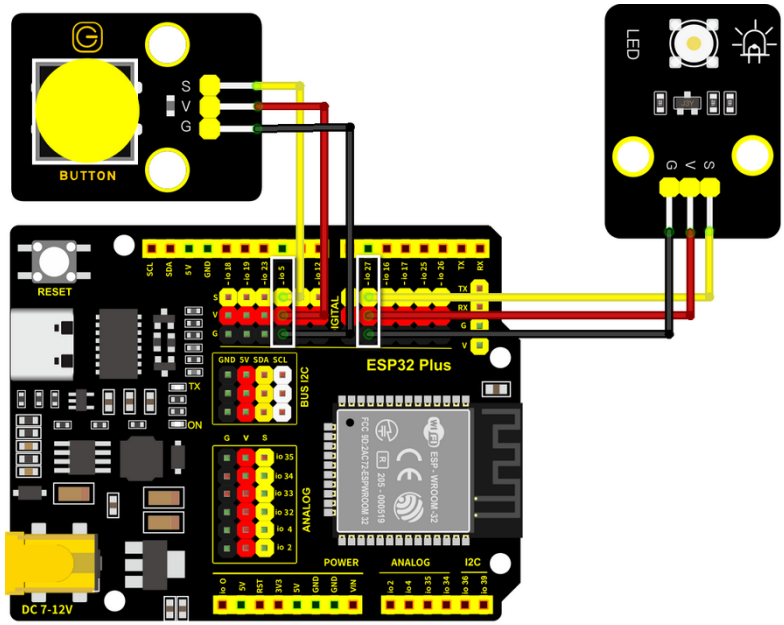
4.Component Functions

- Software used: ***Arduino IDE***

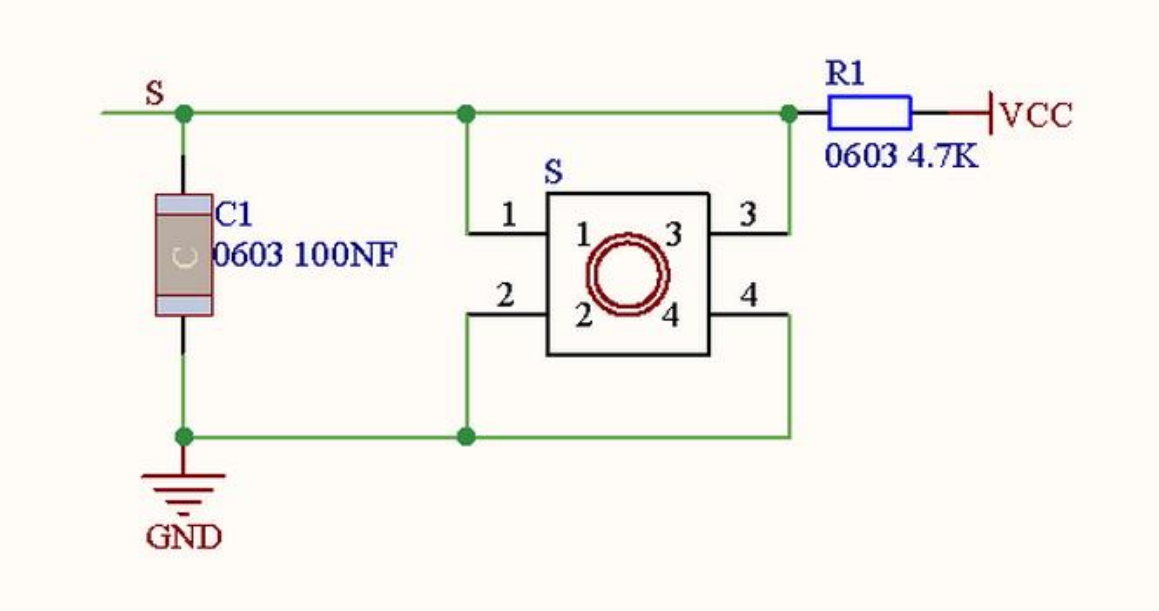
a) Light Control System

1) LED light up when the button is pressed

- ***Circuit diagram***



- ***Wiring diagram***



- ***Parameters***

  Voltage: 3-5 V

  Current: ≤1.1 mA

  Power: ≤5.5 mW

- ***Programming***

```
LED #define  27
#define ButtonPin 5
int value = 0;

void setup(){
  pinMode(led, OUTPUT);
  Serial.begin(9600); initialize serial port and set baud rate to 9600
  pinMode(ButtonPin, INPUT);
}

void ledONandOFF(){
  int ReadValue = digitalRead(ButtonPin);   Read button value

  if (ReadValue == 0) { // Button pressed
    delay(10); // Debounce
     if (digitalRead(ButtonPin) == 0) { // Confirm button press after
debounce
      value = !value;   Toggle the value
If value is true, turn on the LED, otherwise keep it off
      if (value) {
        digitalWrite(led, HIGH);
        Serial.println("LED ON");
      } else {
        digitalWrite(led, LOW);
      }
}
```

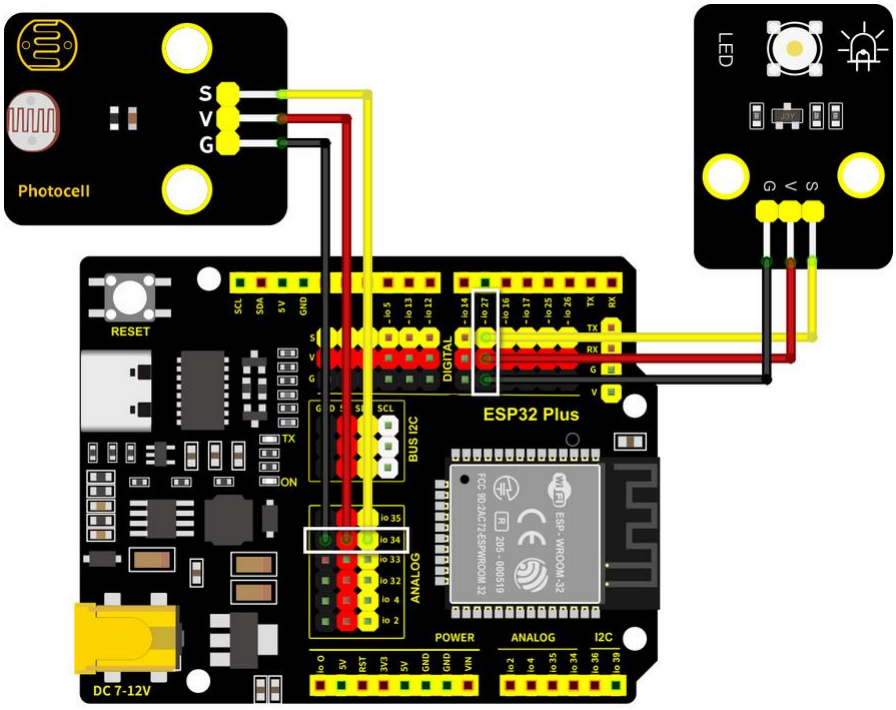
```
    while (digitalRead(ButtonPin) == 0);   // Wait until button is
released
  }
}

void loop(){
    ledONandOFF();
}
```

- ***Result***

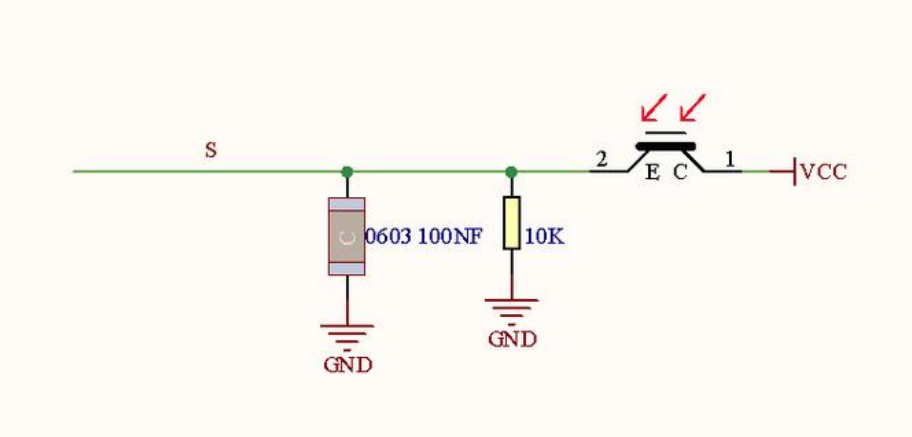
We can read the status of the digital input pin by programming to detect if the button is pressed. Thus, various interactive applications can be realized via a button module, such as turning an LED on/off or adjusting the brightness of the display.

2) LED Lights Up When Dark Is Detected

- ***Circuit diagram***



- ***Wiring diagram***



- ***Programming***

```
LED #define  27
#define PhotoceсllPin 34


void setup(){
  pinMode(led, OUTPUT);
  Serial.begin(9600);
  pinMode(PhotoceсllPin, INPUT);
}
void ledONandOFF(){
    int   ReadValuePhotocecll   =   analogRead(PhotoceсllPin);       Read
photoresistor value
  Serial.print("Photocecll value: ");
  Serial.println(ReadValuePhotocecll);


  if (ReadValuePhotocecll < 800) {  If light level is low and button has
toggled LED on
    digitalWrite(led, HIGH);   Keep LED on
    Serial.println("LED ON");
  } else if (!value) {
    digitalWrite(led, LOW);   Turn off LED if button has toggled it off
    Serial.println("LED OFF");
  }
```

```
  delay(100); // Short delay to stabilize readings
}
void loop(){
      ledONandOFF();
}
```

- ***Parameters***

  Voltage: 3~5V

  Current: 0.2mA

  Power: 1mW

  Maximum Spectrum Value: 540nm

  Light fastness (10 lux): 5~10KΩ

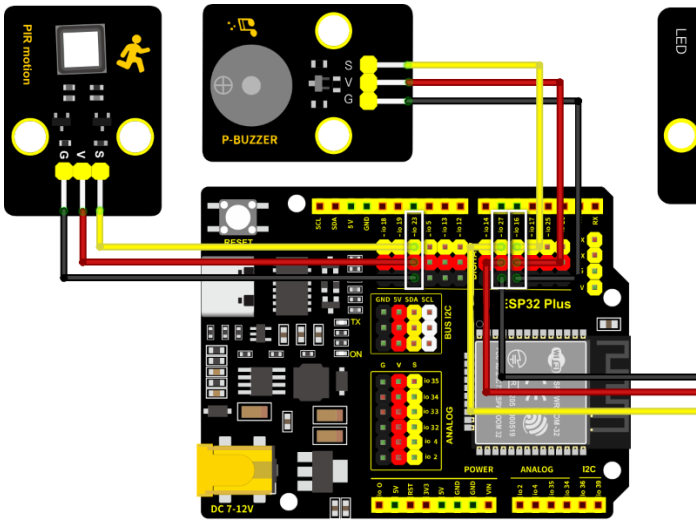
  Dark Resistance: 0.5MΩ

- ***Result***

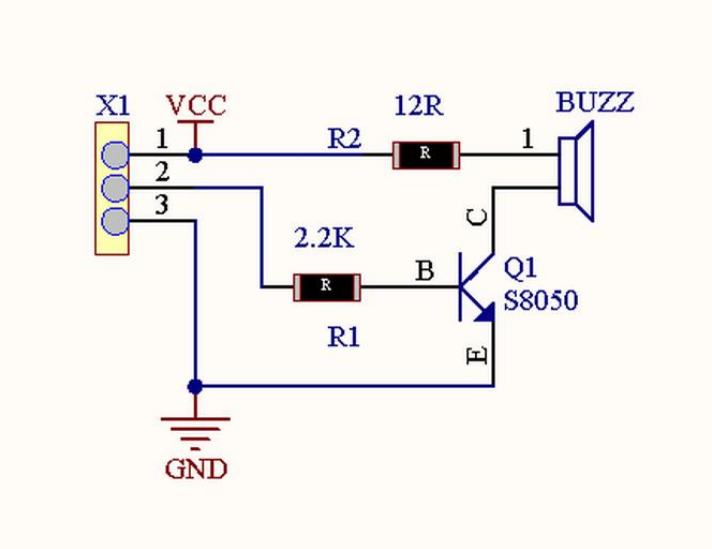
When the photoresistor detects that the ambient brightness is lower than the set value, the LED lights up. On the other hand, if the ambient light intensity is higher than the set value, the photoresistor will send a different signal to turn off the LED.

Motion sensor with alarm

- ***Circuit diagram***



- ***Wiring diagram***



- ***Programming***

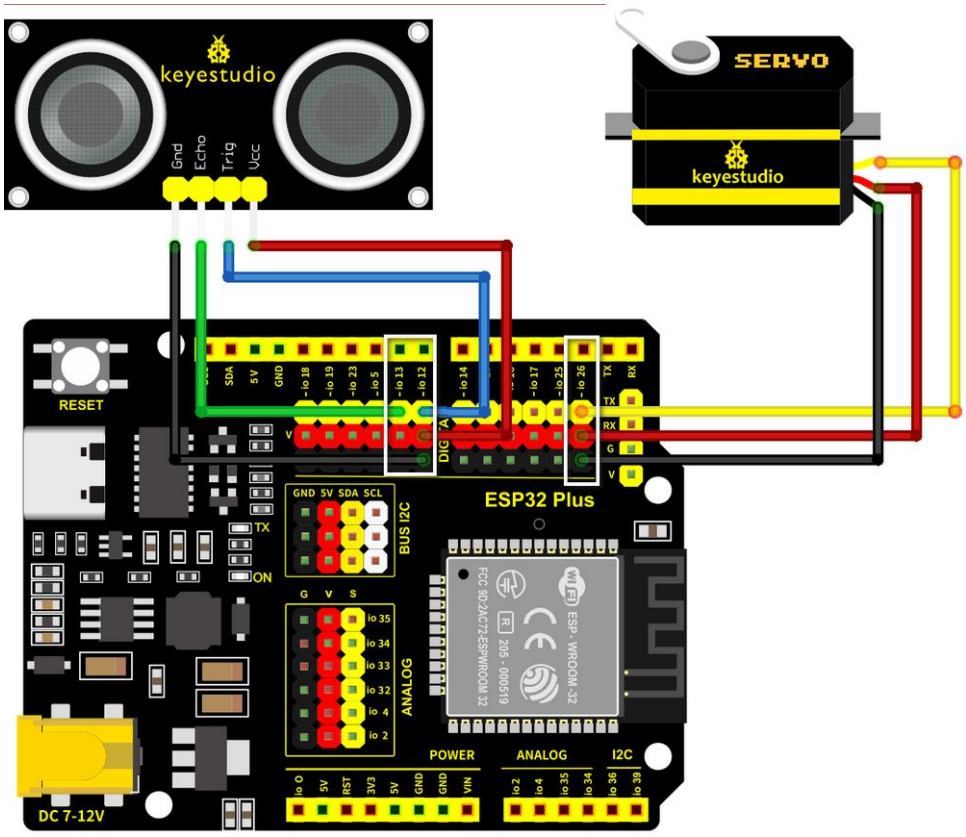
```
#define PyroelectricPIN 23
#define BuzzerPin 16

void setup(){
  pinMode(PhotocellPin, INPUT);
  pinMode(PyroelectricPIN, INPUT);
  pinMode(BuzzerPin, OUTPUT);
}

void nuTeMisca(){
  int ReadValuePyroelectric = digitalRead(PyroelectricPIN);
  if(ReadValuePyroelectric){
    Serial.println("Someone");
    tone(BuzzerPin,532); --C2 //
    delay(100);
    tone(BuzzerPin,587); //re --D3
    delay(100);
    tone(BuzzerPin,659); //mi --E3
    delay(100);
Alarm
```

```
   for(int i = 200; i<=1000; i+=10){
    tone(BuzzerPin,i);
    delay(10);
  }
Alarm
   for(int i = 1000; i>=200; i-=10){
    tone(BuzzerPin,i);
    delay(10);
   }
    noTone(BuzzerPin);
}
  else{
    Serial.println("No one");
}
  delay(100);
}

void loop() {
  nuTeMisca();
}
```

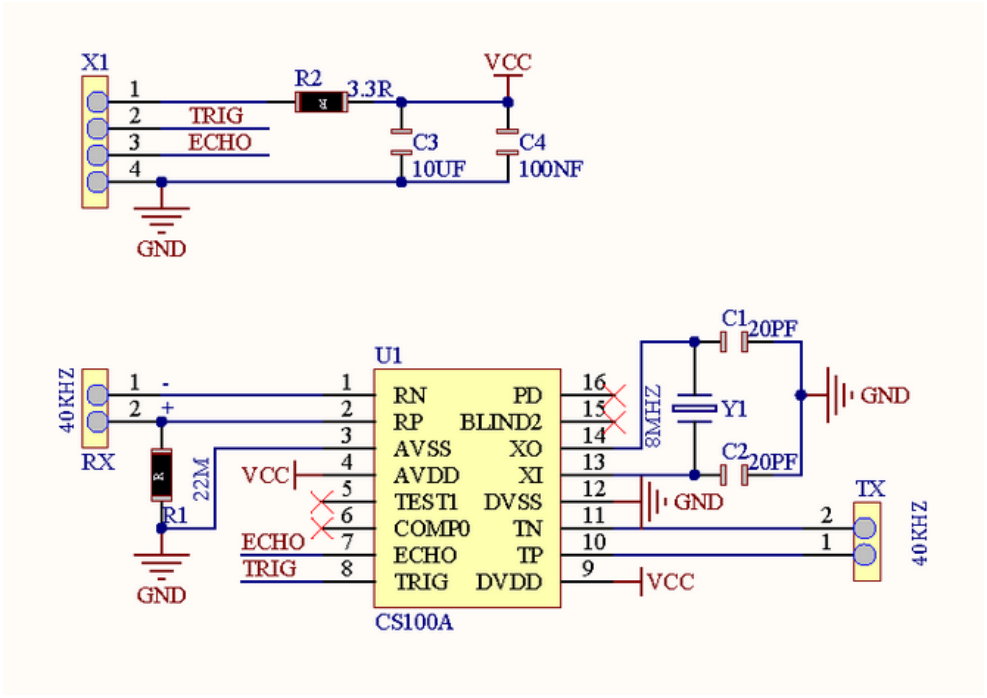
- ***Parameters***
  Voltage: 3~5V
  Current: ≤5mA
  Power: ≤25mW
- ***Result***
  When it detects movement, the doorbell emits an alarm signal.

Opening the box by servo motor

- ***Circuit diagram***



- ***Wiring diagram***



- ***Programming***

```
myservo servo;
int pos = 0;
int servoPin = 26;

#define TrigPin 12
#define EchoPin 13
float duration,distance;

void setup(){
  myservo.attach(servoPin);    attaches the servo on pin 26 to the servo object
  myservo.write(180);

  pinMode(TrigPin,OUTPUT);  set trig pin to output mode
  pinMode(EchoPin,INPUT);   set echo pin to input mode
}

float getDistance() {
  digitalWrite(TrigPin,LOW);
  delayMicroseconds(2);
  digitalWrite(TrigPin,HIGH);
  delayMicroseconds(10); //Trigger the trig pin via a high level lasting at least 10us
  digitalWrite(TrigPin,LOW);
   duration = pulseIn(EchoPin, HIGH); the time of high level at echo pin
   distance = (duration * 0.0343) / 2;convert into distance(cm)
  delay(50);

  Serial.println(distance);
  return distance;
```

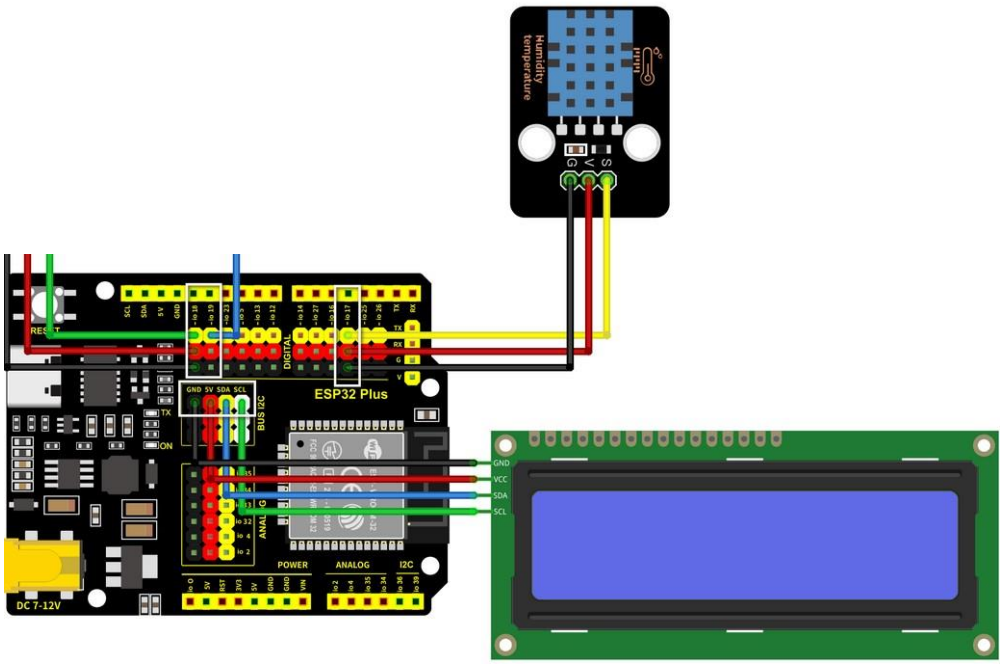
```
}


void servo(){
  float distance = getDistance();
  Serial.print("Distance: ");
  Serial.println(distance);
When the distance is detected within 2~7cm, open the feeding box. Or else, close.
  if (distance >= 2 && 7 >= distance) {
Servo rotates to 80° to open the box
    for (pos = 180; pos >= 80; pos -= 1) // goes from 80 degrees to 0 degrees
      myservo.write(pos);     tell servo to go to position in variable 'pos'
    delay(6000);
    myservo.write(180);
  }
}

void loop() {
servo();
}
```

- ***Result***

The ultrasonic module detects whether the animals are in the feeding area, and the Servo motor automatically opens the bird feed box.

Temperature and humidity detection + display

- ***Circuit diagram***



- ***Programming***

```
Initialize LCD 1602, 0x27 is I2C address
LiquidCrystal_I2C LCD(0x27,16,2);

#define DHTPIN 17 // Pin where the DHT11 is connected
#define DHTTYPE DHT11 // DHT 11
 DHT dht(DHTPIN, DHTTYPE);
Fast calculate the Dew Point, its speed is 5 times of dewPoint()
void setup(){
  Initialize LCD
  LCD.init();
Turn the (optional) backlight off/on
  LCD.backlight();
```

```
lcd.noBacklight();
Set the position o dcursor
  Lcd.setCursor(0, 0);
  Dht.begin();
}

double dewPointFast(double celsius, double humidity)
{
        double a = 17.271;
        double b = 237.7;
        double temp = (a * celsius) / (b + celsius) + log(humidity/100);
        double Td = (b * temp) / (a - temp);
        return Td;
}

Dew Point. The air is saturated and dews are produced under this
temperature.
double dewPoint(double celsius, double humidity)
{
        double A0= 373.15/(273.15 + celsius);
        double SUM = -7.90298* (A0-1);
        SUM += 5.02808 * log10(A0);
 SUM += -1.3816e-7 * (pow(10, (11.344*(1-1/A0)))-1) ;
        SUM += 8.1328e-3 * (pow(10,(-3.49149*(A0-1)))-1) ;
        SUM += log10(1013.246);
        double VP = pow(10, SUM-3) * humidity;
        double T = log(VP/0.61078);  // var temp
        return (241.88 * T) / (17.558-T);
}

double Kelvin(double celsius)
{
```

```
        return celsius + 273.15;
} //Convert Celsius degree to Kelvins

double Fahrenheit(double celsius)
{
        return 1.8 * celsius + 32;
} Convert Celsius degree to Fahrenheit degree

void lcdPrint(float humidity, float temperature){
First Screen: Humidity and Temperature in °C
  LCD.clear();
  LCD.print("Hum(%): ");
  LCD.print(humidity, 0); // 0 decimal places
  LCD.setCursor(0, 1);
  LCD.print("Temp(C): ");
  LCD.print(temperature, 1); // 1 decimal place
  delay(5000); // Display for 2 seconds

// Second Screen: Temperature in °F and Kelvin
  LCD.clear();
  LCD.print("Temp(F): ");
  LCD.print(Fahrenheit(temperature), 0); // 0 decimal places
  LCD.setCursor(0, 1);
  LCD.print("Temp(K): ");
  LCD.print(Kelvin(temperature), 0);  // 0 decimal places
  delay(5000); // Display for 2 seconds

Third Screen: Dew Point and Dew PointFast in °C
  LCD.clear();
  LCD.print("DewPt(C): ");
  LCD.print(dewPoint(temperature, humidity), 1); // 1 decimal place
  LCD.setCursor(0, 1);
```

```
  LCD.print("DewPtF(C): ");
  LCD.print(dewPointFast(temperature, humidity), 1); // 1 decimal place
  delay(5000);  // Display for 2 seconds
}

void temperatureSiHumidity() {
    Serial.println("\n");

    float humidity = dht.readHumidity();
    float temperature = dht.readTemperature();

Check if any reads failed and exit early (to try again).
    if (isnan(humidity) || isnan(temperature)) {
        LCD.println("Failed to read from DHT sensor!");
        return;
}

    lcdPrint(humidity, temperature);

  if (temperature >= 29) {
Turn left
    ledcWrite(1, 100);
    ledcWrite(3, 0);
  }else{
Stop
    delay(3000);
    ledcWrite(1, 0);
    ledcWrite(3, 0);
    delay(200);
}
}
```

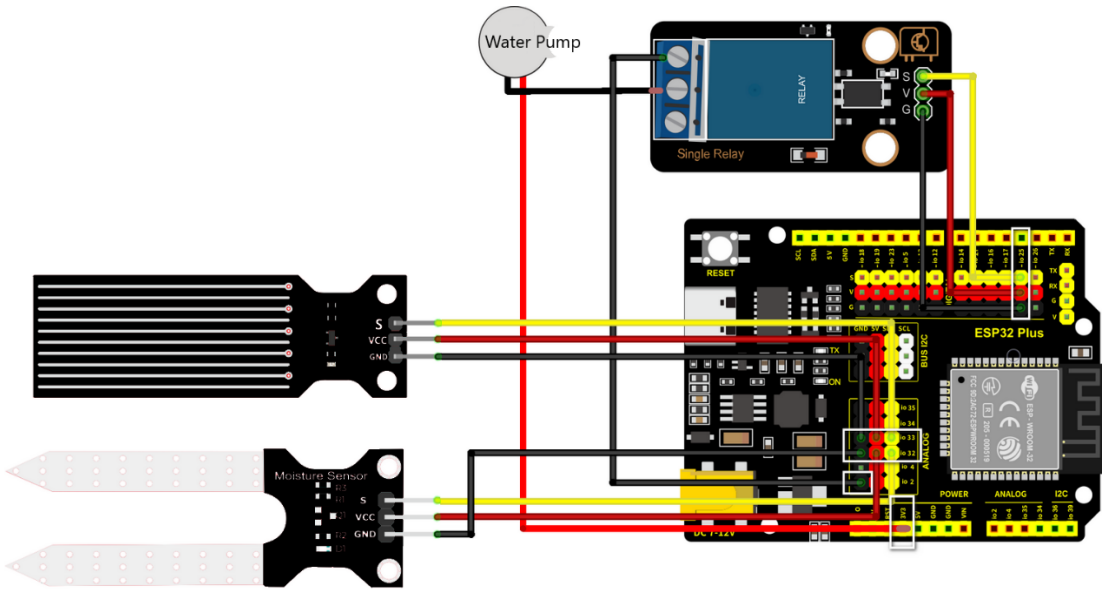
```
void loop() {
  temperatureSiHumidity();
}
```

- ***Result***

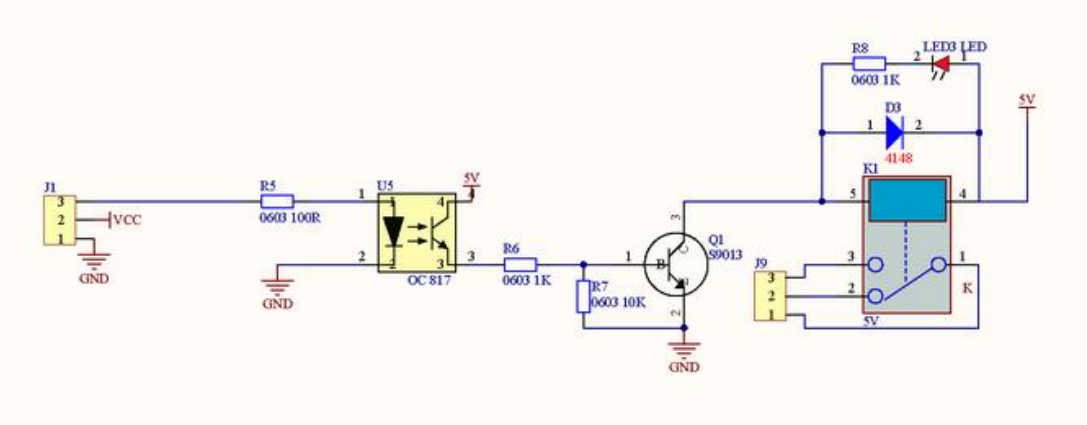
The DHT11 temperature and humidity sensor emits digital signals. It applies analog signal acquisition and conversion principles, as well as temperature and humidity sensing technology, thus providing long-term stability. In this context, we read the value of the DHT11 temperature and humidity sensor through communication, and the values will be displayed on the LCD. Overall, it is a practical system that helps farmers monitor and control status in real-time to improve production efficiency.

Intelligent irrigation system

- ***Circuit diagram***



- ***Wiring diagram***



- ***Programming***

```
#define SoilHumidityPin 32
#define WaterLevelPin 33
#define RelayPin 25
LiquidCrystal_I2C LCD(0x27,16,2);

void setup(){
Initialize LCD
  LCD.init();
Turn the (optional) backlight off/on
  LCD.backlight();
lcd.noBacklight();
Set the position o dcursor
  Lcd.setCursor(0, 0);

  pinMode(SoilHumidityPin,INPUT);
  pinMode(WaterLevelPin,INPUT);
  pinMode(RelayPin,OUTPUT);
}

float earth(){
  int ReadValueSoilHumidity = analogRead(SoilHumidityPin);
```

```
  return ReadValueSoilHumidity;
}

float apa(){
  int ReadValueWaterLevel = analogRead(WaterLevelPin);
  return ReadValueWaterLevel;
}

void waterEarth(){
  if(earth()<200) {
    if (apa()>0)
    {
      digitalWrite(RelayPin,HIGH);
      delay(400);irrigation delay
      digitalWrite(RelayPin,LOW);
      delay(700);
}
}
LCD.clear();
  LCD.print("Soil Hum: ");
  LCD.print(earth());  // 1 decimal place
  LCD.setCursor(0, 1);
  LCD.print("Water: ");
  LCD.print(apa());  // 1 decimal place
  delay(5000); // Display for 2 seconds
}

void loop() {
  watering the Earth();
}
```

- ***Parameters***
  Supply voltage: 5V
  Static current: 2mA
  Maximum Contact Voltage: 250VAC/30VDC
  Maximum current: 10A
- ***Result***

If the soil is very dry, the relay will turn on to control the water pump and irrigate the plants; And if the water level is too low, the water pump will not be able to work. They are displayed on the LCD screen. In this way, plant watering and water level control are automated, which increases production efficiency and reduces the time and effort of manual operations.

6.Conclusions

In conclusion, the realization of this smart farm is a good means for a student to acquire some beneficial qualities for a student who dreams of becoming a future programmer thanks to the coding part that will lay the foundations of a new programming language, easy to understand and practice by a passionate child.

However, the component assembly part can attract the interest of children who are passionate about architecture or engineering and increases their self-confidence the more they manage to carry out the instructions provided. Even their creativity is stimulated until the end of the experiment.

The wiring part attracts all electronics enthusiasts through the wire circuit, sensors with interesting functions and their connection to the main circuit.

Therefore, the Smart Farm kit will excite the students who will build it while learning different things in the areas mentioned above and will acquire useful skills in the future regardless of the field in which they will work.

7.Bibliographic references

- https://docs.keyestudio.com/projects/KS0567/en/latest/wiki/index.html#
- https://en.wikipedia.org/wiki/Main_Page
- https://docs.keyestudio.com/projects/KS0567/en/latest/wiki/Arduino/arduino.html#