* **Components:**
* Temperature and humidity sensor -> analogue pin
* Water pump -> needs relay -> analog pin
* Bluetooth -> pin 0 and 1
* Servo motor -> digital pin
* Soil moisture -> analogue or digital ([¯\\_(ツ)\_/¯](https://www.urbandictionary.com/define.php?term=%C2%AF%5C_%28%E3%83%84%29_%2F%C2%AF) )
* Rain sensor - > analogue pin

1. **Temperature and humidity sensor:**
   1. Model:

DHT11

* 1. Connection:

It has 3 pins: VCC, GND and DATA. Connect VCC to +ve rail on the breadboard, connect GND to -ve rail on the breadboard and connect DATA to an analogue pin in the Arduino (others say digital pin… guess only one way to find out but makes sense connecting it to the analogue!).

* 1. Functionality:

It measures humidity and temperature.

* 1. Code:

Note: add DHT Library to your Arduino IDE. Also, this is just a Demo.

#include <dht.h>

dht DHT;

//Constants

#define DHT11\_PIN 2 // DHT 22 (AM2302) - what pin we're connected to

//Variables

float hum; //Stores humidity value

float temp; //Stores temperature value

void setup()

{

Serial.begin(9600);

}

void loop()

{

int chk = DHT.read11(DHT11\_PIN);

//Read data and store it to variables hum and temp

hum = DHT.humidity;

temp= DHT.temperature;

//Print temp and humidity values to serial monitor

Serial.print("Humidity: ");

Serial.print(hum);

Serial.print(" %, Temp: ");

Serial.print(temp);

Serial.println(" Celsius");

delay(2000); //Delay 2 sec.

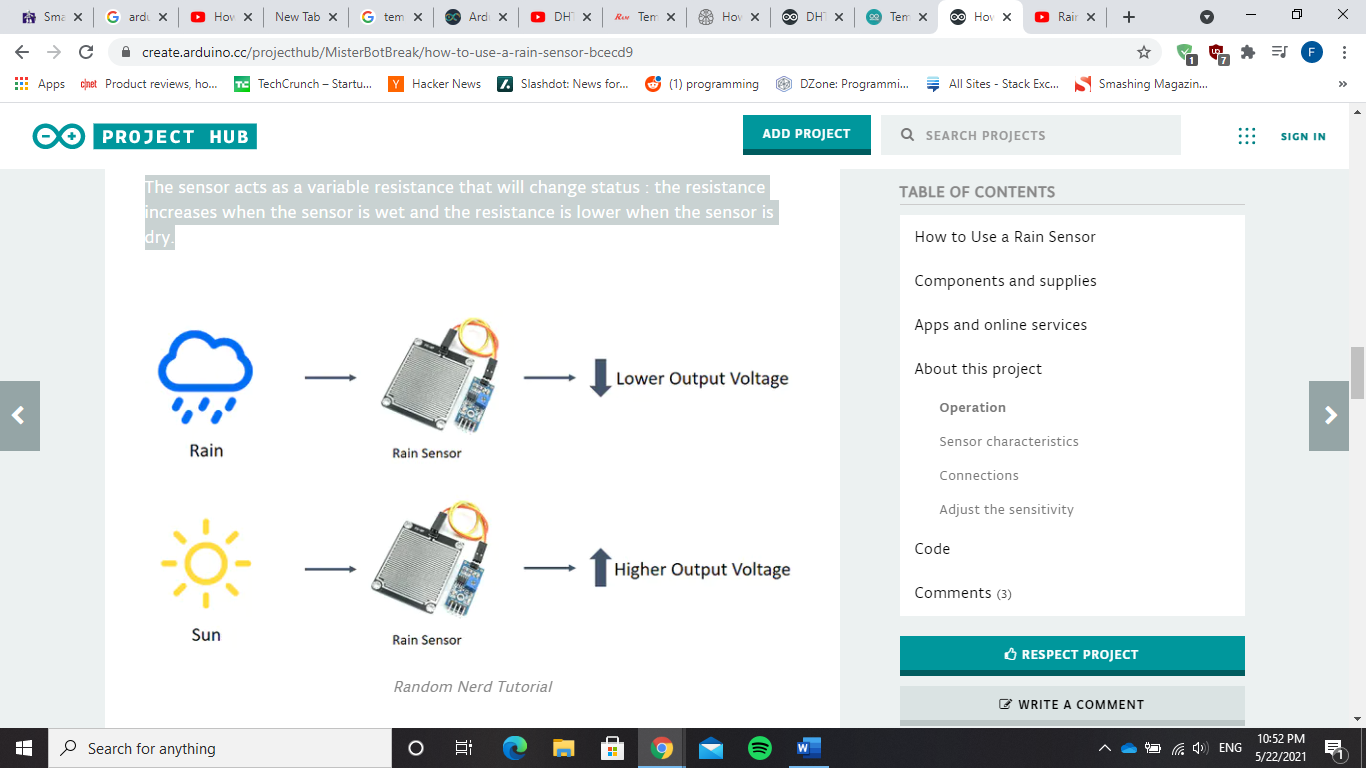
}

1. **Rain Sensor:**
   1. Connection:

It has 4 pins: VCC, GND, AC and DC. Connect VCC pin to the +ve rail on the breadboard, the GND pin to the -ve rail on the breadboard. We will connect the analogue pin for analogue output.

* 1. Functionality:

The rain sensor detects water that comes short circuiting the tape of the printed circuits. The sensor acts as a variable resistance that will change status: the resistance increases when the sensor is wet and the resistance is lower when the sensor is dry.



* 1. Code:

Note: This is only a demo.

const int capteur\_D = 4;

const int capteur\_A = A0;

int val\_analogique;

void setup()

{

pinMode(capteur\_D, INPUT);

pinMode(capteur\_A, INPUT);

Serial.begin(9600);

}

void loop()

{

if(digitalRead(capteur\_D) == LOW)

{

Serial.println("Digital value : wet");

delay(10);

}

else

{

Serial.println("Digital value : dry");

delay(10);

}

val\_analogique=analogRead(capteur\_A);

Serial.print("Analog value : ");

Serial.println(val\_analogique);

Serial.println("");

delay(1000);

}

1. **Soil Moisture**
   1. Connection:

Soil Moisture Sensor consist of two components. A two-legged Lead, that goes into the soil or anywhere else where water content has to be measured. This has two header pins which connect to an Amplifier/ A-D circuit which is in turn connected to the Arduino. The Amplifier has a Vin, GND, Analog and Digital Data Pins. This means that you can get the values in both Analog and Digital forms. Connect the two pins of the sensor to the two pins of the amplifier, connect VCC to the +ve rail on the breadboard, connect GND to the -ve rail on the breadboard. Connect Analogue pin.

* 1. Functionality:

The electrical resistance is measured between the two electrodes of the sensor. A comparator activates a digital output when an adjustable threshold is exceeded. It **estimates soil volumetric water content based on the dielectric constant (soil bulk permittivity) of the soil.** The dielectric constant can be thought of as the soil's ability to transmit electricity. The dielectric constant of soil increases as the water content of the soil increases. This response is due to the fact that the dielectric constant of water is much larger than the other soil components, including air. Thus, **measurement of the dielectric constant gives a predictable estimation of water content.**

* 1. Code:

void setup() {

// initialize serial communication at 9600 bits per second:

Serial.begin(9600);

}

// the loop routine runs over and over again forever:

void loop() {

// read the input on analog pin 0:

int sensorValue = analogRead(A0);

// print out the value you read:

Serial.println(sensorValue);

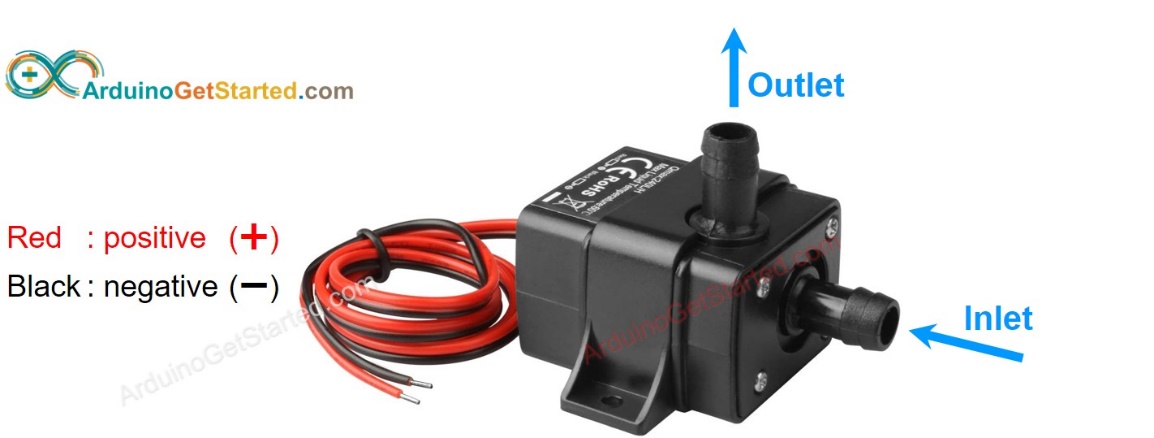
delay(1); // delay in between reads for stability

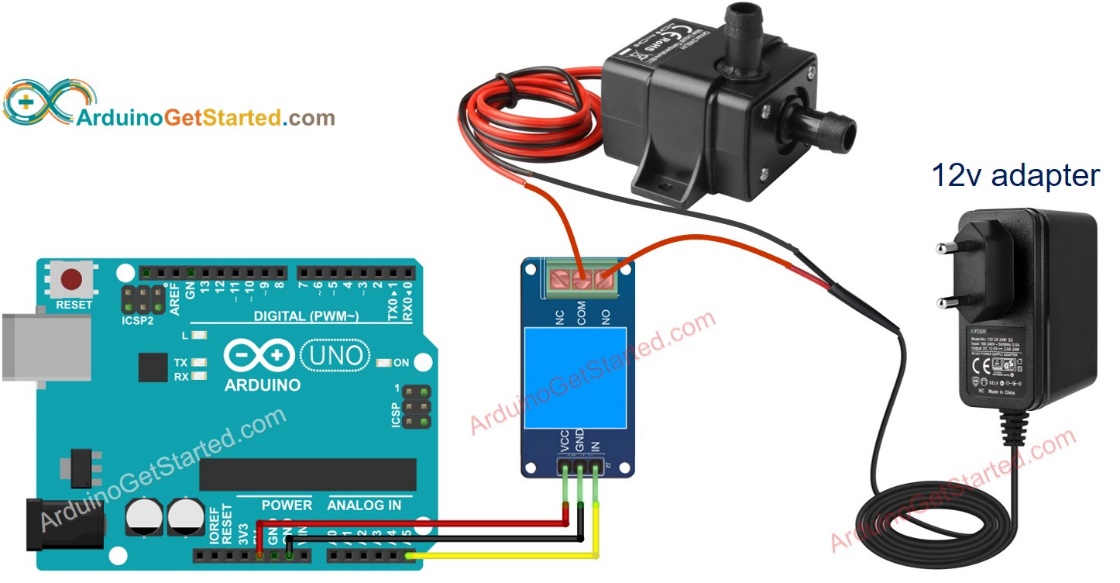
}

1. **Water Pump**
   1. Connection

The water pump cannot be directly connected to the Arduino, it needs a relay kit to connect to then the relay kit gets connected to the Arduino. The Relay got 3 pins in high voltage group (group connected to the power source): NC (Normally Closed), NO (Normally Open) and COM (Common Pin). We normally use COM and NO in normally open mode and use COM and NC in normally closed mode. It also consists of 3 inputs in the low voltage group (the group connected to the Arduino): VCC, GND and IN. Connect the VCC to the +ve rail on the breadboard, connect the GND to the -ve rail on the breadboard and connect the IN pin to a pin on the Arduino. Connect the +ve wire of the pump to the COM of the relay, connect the -ve wire of the pump to the negative input of the power source, and connect the positive wire of the power source to the NO of the relay.







* 1. Functionality

When activated it pumps water from a container using a PVC tube to the other side.

* 1. Code

const int RELAY\_PIN = A5; // the Arduino pin, which connects to the IN pin of relay

void setup() {

pinMode(RELAY\_PIN, OUTPUT);

}

void loop() {

digitalWrite(RELAY\_PIN, HIGH); // turn on pump 5 seconds

delay(5000);

digitalWrite(RELAY\_PIN, LOW); // turn off pump 5 seconds

delay(5000);

}

1. **Bluetooth**
   1. Connection:

Connect the transmitter pin with receive pin in Arduino and vice versa.

* 1. Functionality:

Sends Bluetooth signals.

* 1. Code:

Note: Works only on Android

**char** data = 0; //Variable for storing received data

**void** **setup**()

{

Serial.begin(9600); //Sets the data rate in bits per second (baud) for serial data transmission

pinMode(13, OUTPUT); //Sets digital pin 13 as output pin

}

**void** **loop**()

{

**if**(Serial.available() > 0) // Send data only when you receive data:

{

data = Serial.read(); //Read the incoming data and store it into variable data

Serial.print(data); //Print Value inside data in Serial monitor

Serial.print("\n"); //New line

**if**(data == '1') //Checks whether value of data is equal to 1

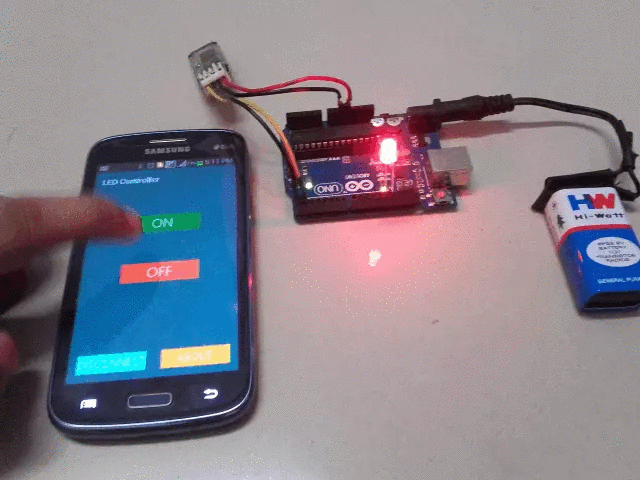
digitalWrite(13, HIGH); //If value is 1 then LED turns ON

**else** **if**(data == '0') //Checks whether value of data is equal to 0

digitalWrite(13, LOW); //If value is 0 then LED turns OFF

}

}



* **The System**

The whole system will be connected on the 5V pin. The system supposedly will test the moisture/ humidity/ temperature of the soil and at certain values it will detect when it will need to be watered, so it will activate the relay to activate the water pump to water the pump and send a notification to the phone to notify the user and keep them up to date.

**That’s it! :D**

* **References**

<https://www.ardumotive.com/how-to-use-dht-11-sensor-en.html#:~:text=and%20humidity%20sensor.-,It%20uses%20a%20capacitive%20humidity%20sensor%20and%20a%20thermistor%20to,right%20most%20pin%20to%20ground>.

<https://create.arduino.cc/projecthub/techno_z/dht11-temperature-humidity-sensor-98b03b>

<https://create.arduino.cc/projecthub/MisterBotBreak/how-to-use-a-rain-sensor-bcecd9>

<https://www.instructables.com/Arduino-Soil-Moisture-Sensor/>

<https://create.arduino.cc/projecthub/MisterBotBreak/how-to-use-a-soil-moisture-sensor-ce769b>

<https://arduinogetstarted.com/tutorials/arduino-relay>

<https://arduinogetstarted.com/tutorials/arduino-controls-pump>

https://maker.pro/arduino/tutorial/bluetooth-basics-how-to-control-led-using-smartphone-arduino#:~:text=Arduino%20Pins%20%7C%20Bluetooth%20Pins&text=Connect%20an%20LED%20positive%20to,jumper%20wires%20and%20a%20connector.