**UART PROGRAM**

**https://www.circuitbasics.com/how-to-set-up-uart-communication-for-arduino/**

**ADC INPUT DATA READ OPERATION**

int dValue, analogInputV;

void setup() {

**Serial**.begin(9600);

}

void loop() {

 dValue = analogRead(A0);

**Serial**.print("Digital value: ");

**Serial**.print(dValue);

 analogInputV = dValue \* 4.887;

**Serial**.print(" , Analog Input voltage: ");

**Serial**.print(analogInputV);

**Serial**.println(" mV");

 delay(1000);

}

**MAP FUNCTION**

int dValue, analogInputV;

void setup() {

**Serial**.begin(9600);

}

void loop() {

 dValue = analogRead(A0);

 analogInputV = map(dValue, 0, 1023, 0, 5000);

**Serial**.print(" , Analog Input voltage: ");

**Serial**.print(analogInputV);

**Serial**.println(" mV");

 delay(1000);

}

**TEMPERATURE SENSOR CALIBRATION FOR ARDUINO**

In the LM35 temperature sensor, 10 mV denotes 1 degree centigrade. So if the voltage value divided by 10 it will give us output in degree centigrade.

int a = analogRead(A0);

a = a \* 4.887;

temperature = a / 10;

int a = analogRead(A0);

a = a \* 3.2235;

temperature = a / 10;

//actual Analog to digital converted Value OR Digital Value

int a = analogRead(A0);

// actual Analog Value OR Input Voltage at A0

Vout = a \* 4.887;

// actual source voltage

Vin = a \* (3000 + 1000) / 1000;

Soil sensor with Arduino in Analog mode

The soil moisture sensor consists of two probes that are immersed in soil to measure the moistness content of water. Current passes from one probe to reach another probe through/via soil and then it gets the resistance value which is then calibrated and we get moisture level.

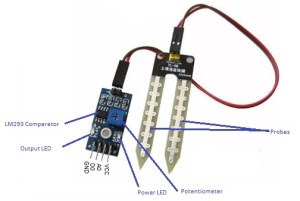
When there is more water (moisture level will be higher), the soil will conduct more electricity which means that there will be less resistance.

Dry soil conducts electricity poorly, so when there will be less water (moisture level will be lower), then the soil will conduct less electricity which means that there will be more resistance.

As soil moisture sensor can be used in both modes analog and digital. Let first connect it in Analog mode and then we will use it in Digital mode.

### **PINOUT – SOIL MOISTURE SENSOR**

The soil Moisture sensor FC-28 has four pins

* VCC: For power
* A0: Analog output
* D0: Digital output
* GND: Ground
* The Module also contains a potentiometer (variable resistor) which is used to set the threshold value and then this threshold value will be compared by the LM393 comparator. The output LED will light up and turn off according to this threshold value.
* [](https://i0.wp.com/pijaeducation.com/wp-content/uploads/2019/09/soil-moisture-sensor.jpg?ssl=1)

int sensor\_pin = A0;

int output\_value ;

void setup() {

**Serial**.begin(9600);

**Serial**.println("Reading From the Sensor ...");

 delay(2000);

}

void loop() {

 output\_value = analogRead(sensor\_pin);

 delay(200);

 //output\_value = map(output\_value, inputLow, inputHigh, outputLow, outputHigh);//

 output\_value = map(output\_value, 0, 1023 , 10, 550);

 output\_value = map(output\_value, 550, 10, 0, 100);

**Serial**.print("Moisture : ");

**Serial**.print(output\_value);

**Serial**.println("%");

 delay(1000);

}

Soil sensor with Arduino in Digital mode

int led\_pin = 13;

int sensor\_pin = 8;

int sensor\_output;

void setup() {

 pinMode(led\_pin, OUTPUT);

 pinMode(sensor\_pin, INPUT);

}

void loop( ) {

 sensor\_output = digitalRead(sensor\_pin);

 if (sensor\_output == HIGH) {

   digitalWrite(led\_pin, HIGH);

 }

 else {

   digitalWrite(led\_pin, LOW);

 }

 delay(1000);

}

Tinkercad simulation: https://www.tinkercad.com/things/8ohm5gHNQLN-soil-moistutre-sensor-interface

ULTRASONIC SENSOR

/ defining pins and variables

long duration;

int distance, led = 3;

const int trigPin = 8, echoPin = 7;

// setup pin mode and begins serial communication with baud rate 9600

void setup() {

 pinMode(led, OUTPUT);

 digitalWrite(led, LOW);

 delay(100);

 pinMode(trigPin, OUTPUT);

 pinMode(echoPin, INPUT);

**Serial**.begin(9600);

}

void loop() {

 // calling a user defined function "calculateDistance()" which returns distance in integer form

 distance = calculateDistance();

 // if distance is less than 20 light up led and print distance as well

 if (distance < 20) {

   digitalWrite(led, HIGH);

**Serial**.print("distance = ");

**Serial**.println(distance);

   delay(500);

 }

 // if distance is greater than 20 and less than 400 turn Off led and print distance as well

 else if (distance > 20 && distance <= 400) {

   digitalWrite(led, LOW);

**Serial**.print("distance = ");

**Serial**.println(distance);

   delay(500);

 }

 // if distance is less than 2 and greater than 400 print message "Out of Range"

 else if (distance < 2 || distance > 400) {

**Serial**.println("Out of range");

 }

}

/\* defining a function calculateDistance() of int type, which means it must return the value of int type.

  then set trigger Pin LOW for 2 microseconds, so that no noise present initially

  then set trigger Pin HIGH for 10 us, so that ultrasonic sensor transmit a sound wave of 40KHz for 10 us as in datasheet

  then set trigger Pin LOW and immediately without delay call a function pulseIn and make echo Pin high which returns time and stored in variable duration

  then convert it by speed formula and returns the distance where function calculateDistance() called

\*/

int calculateDistance() {

 digitalWrite(trigPin, LOW);

 delayMicroseconds(2);

 digitalWrite(trigPin, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin, LOW);

 duration = pulseIn(echoPin, HIGH);

 distance = duration \* 0.034 / 2;

 return distance;

}

**DAC PROGRAMS**

[Build a simple DAC for your Arduino | Arduino Project Hub](https://projecthub.arduino.cc/Arduino_Scuola/build-a-simple-dac-for-your-arduino-5b72f5)

[Arduino DAC Tutorial: Interfacing MCP4725 12-Bit Digital-to-Analog Converter with Arduino (circuitdigest.com)](https://circuitdigest.com/microcontroller-projects/arduino-dac-tutorial-interfacing-mcp4725-dac)