**PROJECT REQUIREMENTS:**

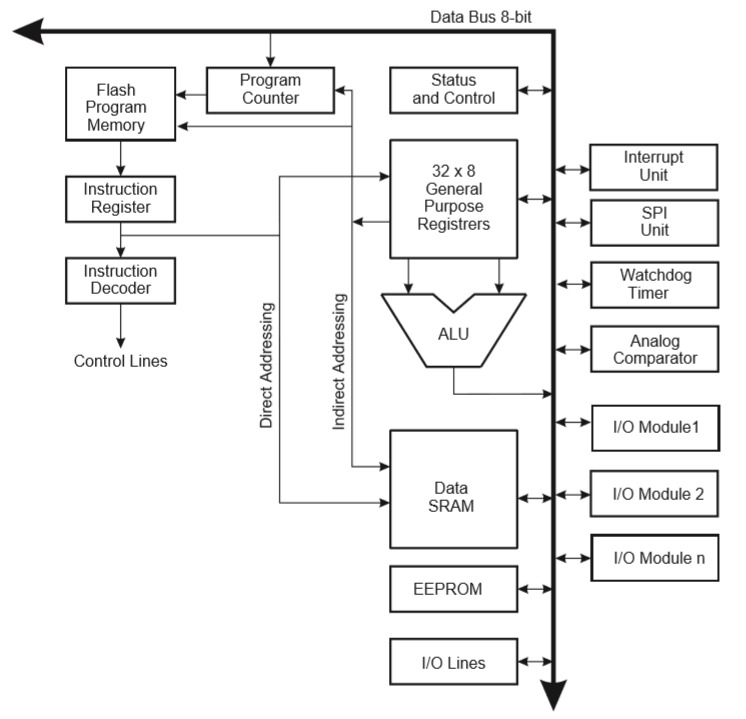
**ABOUT ARDUINO:**

****

Arduino is an open-source electronics platform based on easy-to-use hardware and software. [Arduino boards](https://www.arduino.cc/en/Main/Products) are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the [Arduino programming language](https://www.arduino.cc/en/Reference/HomePage) (based on [Wiring](http://wiring.org.co/)), and [the Arduino Software (IDE)](https://www.arduino.cc/en/Main/Software), based on [Processing](https://processing.org/).

The **Arduino Uno** is one kind of microcontroller board based on ATmega328, and Uno is an Italian term which means one. Arduino Uno is named for marking the upcoming release of microcontroller board namely **Arduino Uno Board 1.0.** This board includes digital I/O pins-14, a power jack, analog i/ps-6, ceramic resonator-A16 MHz, a USB connection, an RST button, and an ICSP header. All these can support **the microcontroller**for further operation by connecting this board to the computer. The power supply of this board can be done with the help of an AC to DC adapter, a USB cable, otherwise a battery.

**ARCHITECTURE OF ARDUINO:**



# \*Starting clockwise from the top centre:

* Analog Reference pin (AREF) (orange)
* Digital Ground (GND) (light green)
* Digital Pins 2-13 (green)
* Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (digital Read and digital Write) if you are also using serial communication (e.g. Serial.begin).
* Reset Button - S1 (dark blue)
* In-circuit Serial Programmer (blue-green)
* Analog In Pins 0-5 (light blue)
* Power and Ground Pins (power: orange, grounds: light orange)
* External Power Supply In (9-12VDC) - X1 (pink)
* Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
* USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

**DIGITAL PINS:**

The digital pins on an Arduino board can be used for general purpose input and output via the pinMode (), digitalRead(), and digitalWrite() commands. Each pin has an internal pull-up resistor which can be turned on and off using digitalWrite() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40 mA.

* **Serial**: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.
* **External Interrupts**: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* **PWM**: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means.
* **BT Reset**: 7. (Arduino BT-only) Connected to the reset line of the Bluetooth module.
* **SPI**: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).

Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two microcontrollers.

With an SPI connection there is always one master device (usually a microcontroller) which controls the peripheral devices. Typically, there are three lines common to all the devices:

* **MISO** (Master in Slave Out) - The Slave line for sending data to the master,
* **MOSI** (Master Out Slave In) - The Master line for sending data to the peripherals,
* **SCK** (Serial Clock) - The clock pulses which synchronize data transmission generated by the master and one line specific for every device:

**SS (Slave Select)** - the pin on each device that the master can use to enable and disable specific devices.

When a device's Slave Select pin is low, it communicates with the master. When it's high, it ignores the master. This allows you to have multiple SPI devices sharing the same MISO, MOSI, and CLK lines.

* **LED**: 13. On the Diecimila and Lilypad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

**ANALOG PINS:**

The analog input pins support 10-bit analog-to-digital conversion (ADC) using the analogRead() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

**POWER PINS:**

* **VIN** (sometimes labelled "9V"). The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.5V.
* The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
* **3V3**. (Diecimila-only) A 3.3-volt supply generated by the on-board FTDI chip.
* **GND**. Ground pins.

**OTHER PINS:**

* **AREF**. Reference voltage for the analog inputs. Used with [analogReference](https://www.arduino.cc/en/Reference/AnalogReference)().
* **Reset**. (Diecimila-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

**WHY ONLY ARDUINO UNO?**

Arduino Uno is the most standard board available and probably the best choice for a beginner. It is a good all-purpose board that has enough features for a beginner to get started with. Some of its better features are:

1. Its biggest advantage is that we connect the board to the computer via a USB cable which does a dual purpose of supplying power and acting as a Serial port to interface the Arduino and the computer.
2. It can also be powered by a 9V-12V AC to DC adapter.
3. The ATmega328 chip can be newly bought, removed and replaced if damaged which is not possible with other versions.
4. The board operates at 5V throughout, i.e. digital pins output or read 5v and analog pins read in the range 0-5V.
5. Lots of example code and projects are done using Arduino Uno, hence will get good support.
6. The Uno features 14 Digital I/O pins and 6 Analog I/O pins.
7. Lot of extra Add-on hardware is built for Uno. Special hardware is available for Internet, Bluetooth, Motor control etc.

**GSM SIM800A MODULE:**

|  |  |
| --- | --- |
| **KEY FEATURES** • Dual frequency 900 / 1800MHz  • GPRS multi-slot class 12/10  • GPRS mobile station class B  • meet GSM 2/2 + standards    – Class 4 (2W @900MHz)    – Class 1 (1W @1800MHz)  • Size: 24 \* 24 \* 3mm  • Control via AT commands (GSM 07.07, 07.05 and SIMCOM enhanced AT command set)  • SIM Application Toolkit  • Supply voltage range: 3.4 ~ 4.4V  • Low power consumption  • Operating temperature range: -40 ° C ~ + 85 ° C   **Data transmission** • GPRS class 12: Maximum 85.6 kbps (downlink rate)  • Support PBCCH  • Coding schemes CS 1, 2, 3, 4  • PPP-stack  • CSD up to 14.4 kbps  • USSD  • non-transparent **SMS** • MO and MT  • SMS Broadcast  • Text and PDU mode   **Software Features** • 0710 MUX protocol  • embedded TCP / UDP protocol  • FTP / HTTP  • MMS  • Email  • DTMF detection  • Interference Detection  • Recording  • SSL  • Bluetooth 3.0 (optional) | **VOICE** • Tricodec   - Half Rate (HR)   - Full rate (FR)   - Enhanced Full Rate (EFR)  • AMR   - Half Rate (HR)   - Full rate (FR)  • Support echo suppression algorithms   **Interface** 68-pin SMT  • one analog audio interface  • Support RTC  • USB Interface  • Serial Interface  • SIM card interface (3V / 1.8V)  • GPIO  • ADC  • PWM  • I2C  • GSM antenna pin  • BT antenna pin   **Compatibility** • AT cellular command interface   **Authenticate** • EC  • ROHS  • REACH  • CCC  • TA  • CTA (network test) (on going) |

**KIT DETAILS:**

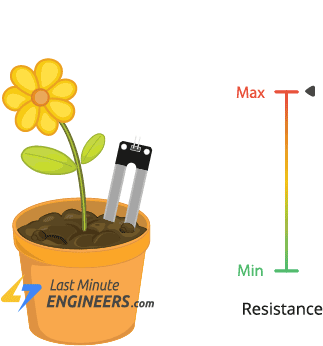
The following are the kit details of this Arduino based project which are hardware components:

**Relay(12v):**

1. A **relay** circuit is **used** to realize logic functions. They play a very important role in providing safety critical logic.
2. **Relays** are **used** to provide time delay functions.
3. **Relays** are **used** to control high voltage circuits with the help of low voltage signals.
4. They are also **used** as protective **relays**.

Here it is used to control the voltage to passed through the circuit and also to switch the water cooler pump on or off.

**GSM module 800A:** The SIM800A Quad-Band GSM/GPRS Module with RS232 Interface is a complete Quad-band GSM/GPRS solution in an **LGA** (Land grid array) type which can be embedded in the customer applications. SIM800A support Quad-band 850/900/1800/1900 MHz, it can transmit Voice, SMS and data information with low power consumption.

**Soil moisture sensor:** The fork-shaped probe with two exposed conductors, acts as a variable resistor (just like a potentiometer) whose resistance varies according to the water content in the soil. This resistance is inversely proportional to the soil moisture:

* The less water in the soil means poor conductivity and will result in a higher resistance.

The sensor produces an output voltage according to the resistance, which by measuring we can determine the moisture level.

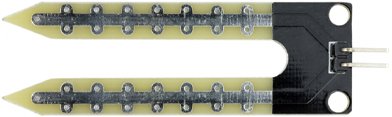
**Hardware Overview**

A typical soil moisture sensor has two components.

The Probe

The sensor contains a fork-shaped probe with two exposed conductors that goes into the soil or anywhere else where the water content is to be measured.

Like said before, it acts as a variable resistor whose resistance varies according to the soil moisture.



The Module

The sensor also contains an electronic module that connects the probe to the Arduino.

The module produces an output voltage according to the resistance of the probe and is made available at an Analog Output (AO) pin.

The same signal is fed to a LM393 High Precision Comparator to digitize it and is made available at an Digital Output (DO) pin.

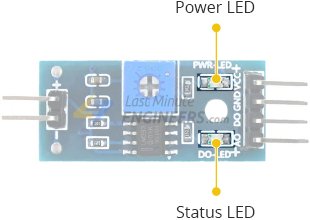


The module has a built-in potentiometer for sensitivity adjustment of the digital output (DO).

You can set a threshold by using a potentiometer; So that when the moisture level exceeds the threshold value, the module will output LOW otherwise HIGH.

This setup is very useful when you want to trigger an action when certain threshold is reached. For example, when the moisture level in the soil crosses a threshold, you can activate a relay to start pumping water. You got the idea!

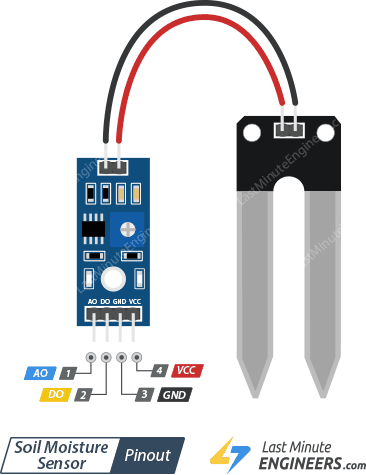
Tip: Rotate the knob clockwise to increase sensitivity and counterclockwise to decrease it.



Apart from this, the module has two LEDs. The Power LED will light up when the module is powered. The Status LED will light up when the digital output goes LOW.

Soil Moisture Sensor Pinout

The soil moisture sensor is super easy to use and only has 4 pins to connect.



AO (Analog Output) pin gives us an analog signal between the supply value to 0V and will be connected to one of the analog inputs on your Arduino.

DO (Digital Output) pin gives Digital output of internal comparator circuit. You can connect it to any digital pin on an Arduino or directly to a 5V relay or similar device.

VCC pin supplies power for the sensor. It is recommended to power the sensor with between 3.3V – 5V. Please note that the analog output will vary depending on what voltage is provided for the sensor.

GND is a ground connection.

Sensing Soil Moisture using Analog Output

As you know that the module provides both analog and digital output, so for our first experiment we will measure the soil moisture by reading the analog output.

Wiring

Let’s hook the soil moisture sensor up to the Arduino.

First you need to supply power to the sensor. For that you may connect the VCC pin on the module to 5V on the Arduino.

However, one commonly known issue with these sensors is their short lifespan when exposed to a moist environment. Having power applied to the probe constantly speeds the rate of corrosion significantly.

To overcome this, we recommend that you do not power the sensor constantly, but power it only when you take the readings.

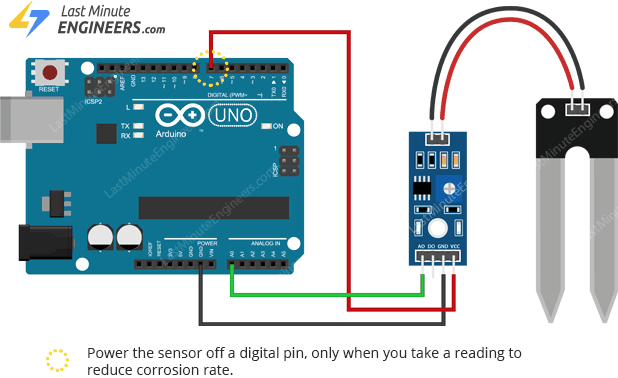
An easy way to accomplish this is to connect the VCC pin to a digital pin of an Arduino and set it to HIGH or LOW as per your requirement.

Also, the total power drawn by the module (with both LEDs lit) is about 8 mA, so it is okay to power the module off a digital pin on an Arduino.

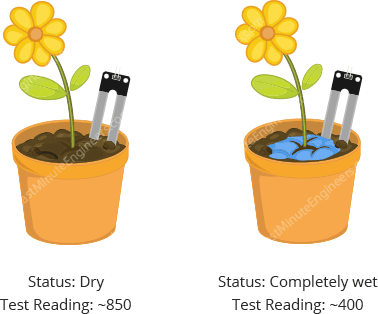
So, let’s connect the VCC pin on the module to the digital pin #7 of an Arduino and GND pin to ground.

Finally, connect the AO pin on the module to the A0 ADC pin on your Arduino.

The following illustration shows the wiring.



Calibration

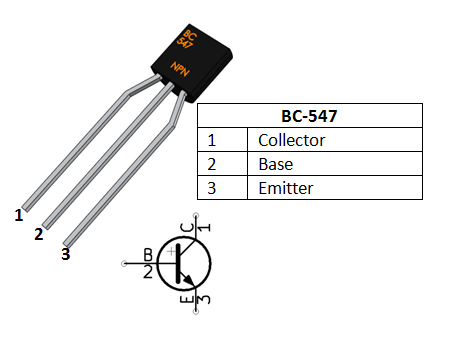
To get accurate readings out of your soil moisture sensor, it is recommended that you first calibrate it for the particular type of soil that you plan to monitor.

Different types of soil can affect the sensor, so your sensor may be more or less sensitive depending on the type of soil you use.

Before you start storing data or triggering events, you should see what readings you are actually getting from your sensor.

Use the below sketch to note what values your sensor outputs when the soil is as dry as possible -vs- when it is completely saturated with moisture.

**Transistor (BC 547):**

A Transistors acts as an Amplifier when operating in **Active Region**. It can amplify power, voltage and current at different configurations. Here it is used as Amplifier.

Some of the configurations used in amplifier circuits are

1. Common emitter amplifier
2. Common collector amplifier
3. Common base amplifier

Of the above types common emitter type is the popular and mostly used configuration. When uses as an Amplifier the DC current gain of the Transistor can be calculated by using the below formulae

**DC Current Gain = Collector Current (IC) / Base Current (IB)**

**Applications**

* Driver Modules like Relay Driver, LED driver etc.
* Amplifier modules like Audio amplifiers, signal Amplifier etc.

**Resistor:** It is used to control or adjust the resistance in the circuit.

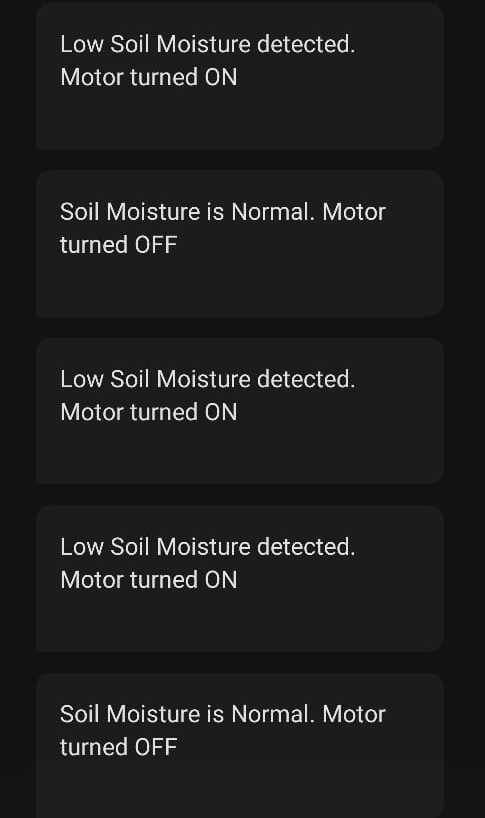
**Water motor (220-240 v):** It is used to pump the water from the source to the plants.



**PROJECT STATEMENT:**

Our project is an Arduino based Automatic irrigation system through which the plants get irrigated automatically, depending upon the moisture levels of the soil. This would be done by automatic switching of motor pump between ON/OFF according to the soil moisture levels regarding which an SMS would be sent to the user with the help of GSM.

**SAMPLE RESULTS FROM THE TESTING PROCESS OF THE PROJECT:**



**PROCESS OF EXECUTION:**

The moisture sensor probe that is placed in the soil senses the moisture and gives the digital reading (1 or 0) to the Arduino. According the code that is written, when the digital read is 0, it indicates that the soil moisture is normal and the motor gets turned OFF and an SMS will be sent to the registered mobile number. Same happens when the moisture is low. As the digital read goes to 1, the motor gets triggered and turned to ON and an SMS would be sent to the mobile.