**PROJECT :** **CENTRALIZED REMOTE AGRICULTURE PUMP CONTROL SYSTEM FOR FARMER**

**TEAM NAME : MKS**

**MEMBERS**

1. **TNVD.MOUNICA**
2. **SHIRLENE ROSE**
3. **CH.KEERTHANA**

**ABSTRACT**

In the modern days ,as the usage of technology is increasing day by day,human beings are merely interested on remote monitoring equipment in each and every part of life . The need for electronically controlled water level in water pumps and soil in a field ,even in the absence of human beings ,to ensure constant water distribution is a serious demand . We therefore intend to provide a solution by constructing an electronic system in rural areas for farmers, that has a capability of monitoring the water level that flows through a pipe and also the amount of moisture in a soil and also to detect if the amount of water flowing through the pipe is less and inform the farmer immediately .

The project involves the use of a flow sensor, soil moisture sensor , GSM GPRS module and Arduino Uno. The flow sensor monitors the flow of water through the pipe and sends an SMS through the GSM GPRS module to the farmer asking him to turn on the pump and after the irrigation is done and a certain moisture level is reached , the soil moisture sensor notifies the farmer again through SMS to turn off the pump. This way we will be able to save water ,time and energy of the farmer.

**TOPIC :**

**Centralized Remote Agriculture Pump Control System For Farmers**

**Hardware**

* Arduino UNO
* GSM/GPRS module
* Push Buttons
* Relay
* Soil Moisture Sensor
* Flow Sensor

**Softwares**

* Arduino IDE

**ARDUINO UNO**

The **Arduino UNO** is an [open-source](https://en.wikipedia.org/wiki/Open-source) [microcontroller board](https://en.wikipedia.org/wiki/Microcontroller_board) based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc](https://en.wikipedia.org/wiki/Arduino).[[2]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-2)[[3]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-What_is_Arduino?-3) The board is equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various [expansion boards](https://en.wikipedia.org/wiki/Expansion_board) (shields) and other circuits.[[1]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-Makerspace-1) The board has 14 Digital pins, 6 Analog pins, and programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment) via a type B [USB cable](https://en.wikipedia.org/wiki/USB_cable).[[4]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-priceton-4) It can be powered by the USB cable or by an external [9-volt battery](https://en.wikipedia.org/wiki/9-volt_battery), though it accepts voltages between 7 and 20 volts. It is also similar to the [Arduino Nano](https://en.wikipedia.org/wiki/Arduino_Nano) and Leonardo.[[5]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-5)[[6]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-6) The hardware reference design is distributed under a [Creative Commons](https://en.wikipedia.org/wiki/Creative_Commons) Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

**APPLICATIONS**

## Features of the Arduino Uno Board:

* It is an easy USB interface. This allows interface with USB as this is like a serial device.
* The chip on the board plugs straight into your USB port and supports on your computer as a virtual serial port. The benefit of this setup is that serial communication is an extremely easy protocol which is time-tested and USB makes connection with modern computers and makes it comfortable.
* It is [easy-to-find the microcontroller](https://www.elprocus.com/microcontrollers-types-and-applications/) brain which is the ATmega328 chip. It has more number of hardware features like timers, external and internal interrupts, PWM pins and multiple sleep modes.
* It is an open source design and there is an advantage of being open source is that it has a large community of people using and troubleshooting it. This makes it easy to help in debugging projects.
* It is a 16 MHz clock which is fast enough for most applications and does not speeds up the microcontroller.
* It is very convenient to manage power inside it and it had a feature of built-in voltage regulation. This can also be powered directly off a USB port without any external power. You can connect an external power source of upto 12v and this regulates it to both 5v and 3.3v.
* 13 digital pins and 6 analog pins. This sort of pins allows you to connect hardware to your Arduino Uno board externally. These pins are used as a key for extending the computing capability of the Arduino Uno into the real world. Simply plug your electronic devices and [sensors](https://www.watelectrical.com/6-different-types-of-temperature-sensors-with-their-specifications/) into the sockets that correspond to each of these pins and you are good to go.
* This has an ICSP connector for bypassing the USB port and interfacing the Arduino directly as a serial device. This port is necessary to re-bootload your chip if it corrupts and can no longer used to your computer.
* It has a 32 KB of flash memory for storing your code.
* An on-board LED is attached to digital pin 13 to make fast the debugging of code and to make the debug process easy.
* Finally, it has a button to reset the program on the chip.

**SOIL MOISTURE SENSOR**

Soil Moisture Sensor This sensor can be used to test the moisture of soil, when the soil is having water shortage, the module output is at high level, else the output is at low level. By using this sensor one can automatically water the flower plant, or any other plants requiring automatic watering technique. Module triple output mode, digital output is simple, analog output more accurate, serial output with exact readings.

**Features**

* convenient installation.

• Threshold level can be configured.

• Module triple output mode, digital output is simple, analog output more accurate, serial output with exact readings. Applications

• Agriculture

• Landscape irrigation Specifications Parameter Value Operating Voltage +5v dc regulated Soil moisture Digital value is indicated by out pin

• Sensitivity adjustable.

## Specifications

* Range: 0 to 45% volumetric water content in soil (capable of 0 to 100% VWC with alternate calibration)
* Accuracy: ±4% typical
* Typical Resolution: 0.1%
* Power: 3 mA @ 5VDC
* Operating temperature: –40°C to +60°C
* Dimensions: 8.9 cm × 1.8 cm × 0.7 cm (active sensor length 5 cm)
* Range: 0 to 45% volumetric water content in soil (capable of 0 to 100% VWC with alternate calibration)
* Accuracy: ±4% typical
* Typical Resolution: 0.1%
* Power: 3 mA @ 5VDC
* Operating temperature: –40°C to +60°C
* Dimensions: 8.9 cm × 1.8 cm × 0.7 cm (active sensor length 5 cm)

## Working Principle of Moisture Sensor

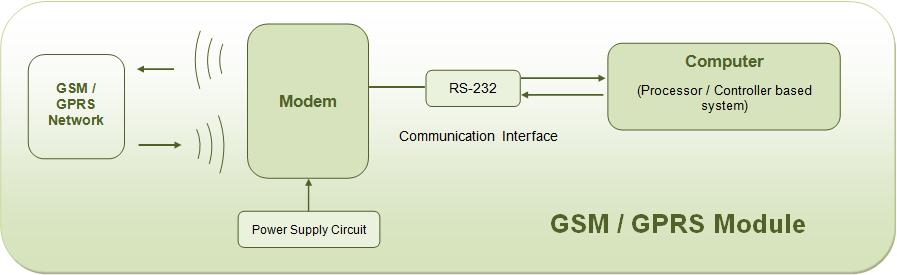
The Soil Moisture Sensor uses capacitance to measure dielectric permittivity of the surrounding medium. In soil, dielectric permittivity is a function of the water content. The sensor creates a voltage proportional to the dielectric permittivity, and therefore the water content of the soil. The sensor averages the water content over the entire length of the sensor. There is a 2 cm zone of influence with respect to the flat surface of the sensor, but it has little or no sensitivity at the extreme edges.The Soil Moisture Sensor is used to measure the loss of moisture over time due to evaporation and plant uptake ,evaluate optimum soil moisture contents for various species of plants,monitor soil moisture content to control irrigation in greenhouses and enhance bottle biology experiments.



**GSM GPRS MODULE**

**GSM/GPRS module** is used to establish communication between a computer and a **GSM-GPRS system**. **Global System for Mobile communication (GSM)** is an architecture used for mobile communication in most of the countries. **Global Packet Radio Service (GPRS)** is an extension of GSM that enables higher data transmission rate. **GSM/GPRS module consists of a GSM/GPRS modem assembled together with power supply circuit and communication interfaces** (like RS-232, USB, etc) for computer. The MODEM is the soul of such modules.

* Range: 0 to 45% volumetric water content in soil (capable of 0 to 100% VWC with alternate calibration)
* Accuracy: ±4% typical
* Typical Resolution: 0.1%
* Power: 3 mA @ 5VDC
* Operating temperature: –40°C to +60°C
* Dimensions: 8.9 cm × 1.8 cm × 0.7 cm (active sensor length 5 cm)



**GSM/GPRS MODEM**

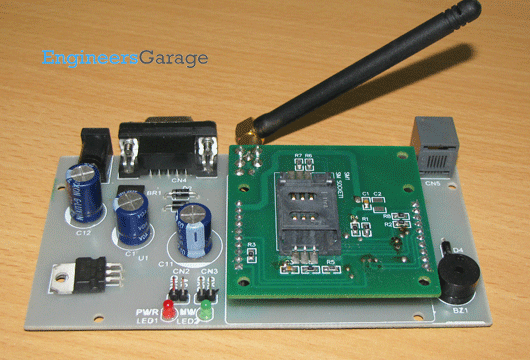
GSM/GPRS MODEM is a class of wireless MODEM devices that are designed for communication of a computer with the GSM and GPRS network. It requires a **SIM (Subscriber Identity Module)** card just like mobile phones to activate communication with the network. Also they have **IMEI** (International Mobile Equipment Identity) number similar to mobile phones for their identification. A GSM/GPRS MODEM can perform the following operations:

1.      Receive, send or delete SMS messages in a SIM.

2.      Read, add, search phonebook entries of the SIM.

3.      Make, Receive, or reject a voice call.

The MODEM needs **AT commands**, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the **GSM and GPRS cellular network**.

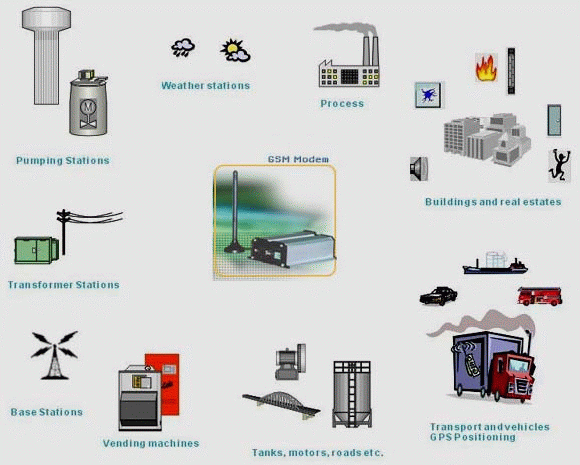


**Applications of GSM/GPRS module**

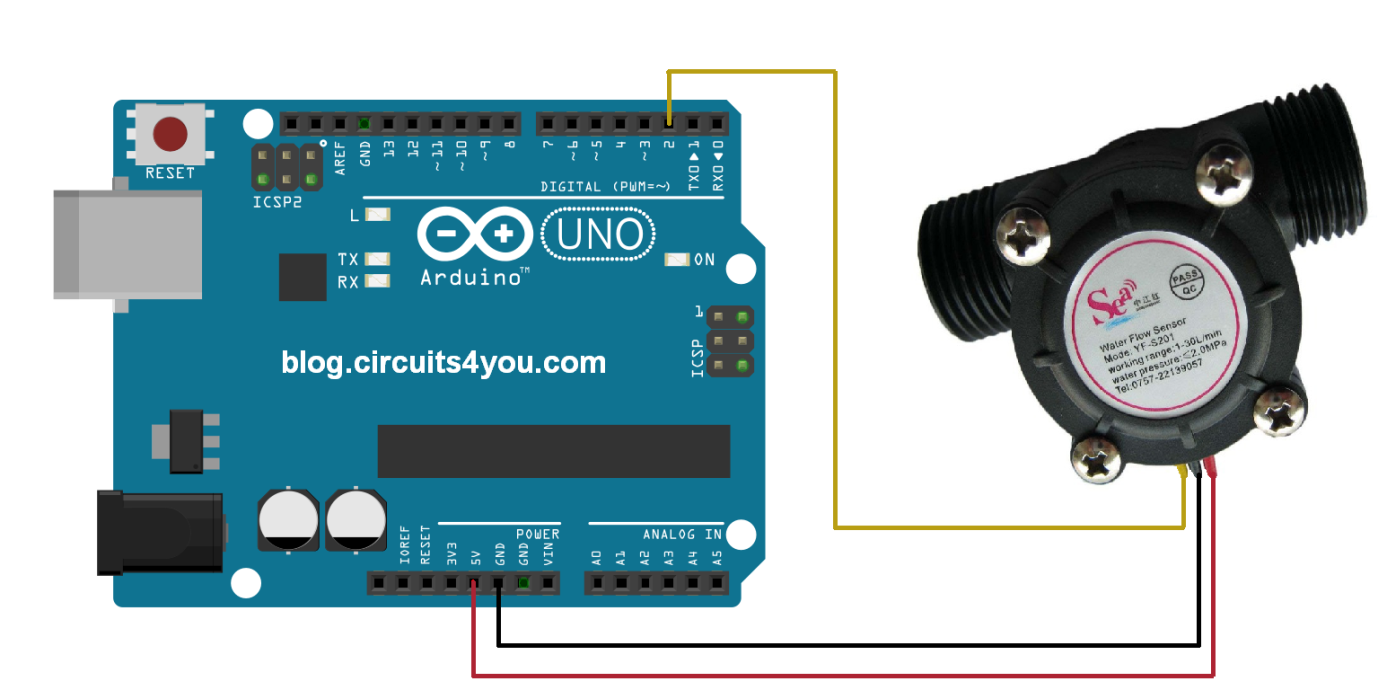
The GSM/GPRS module demonstrates the use of AT commands. They can feature all the functionalities of a mobile phone through computer like making and receiving calls, SMS, MMS etc. These are mainly employed for computer based SMS and MMS services.

**APPLICATIONS:**

**Machine-to-Machine Communications**  
M2M communication is about integrating real time date from remote assets with business processes to gain business advantage  
  
**Benefits of M2M**  
Reduced Costs : M2M solutions reduces cost by making the operations efficient, reducing the onsite vists, automating processes etc.   
  
Improved Service : By building intelligence with in the devices, M2M solutions can greatly improve the service aspect of the devices through remote management.  
  
**Increases Revenue:   
  
M2M Eco System**  
  
**Modules** : Wireless modules that operate on the cellular networks enable remote communication on M2M devices. They support different communication methods such as IP, SMS etc. Bandwidth and latencies of wireless modules vary depending on the network technologies that they support. Wireless modules come in different form factors such as PCIe, LGA, BGA and other proprietary form factors. The cost of wireless modules have been significantly coming down over the past few years.  
  
**Devices** : Wireless modules are embedded in variety of devices across various vertical markets such as security, industrial, retail, healtcare, transportation, buildings, energy etc.  
  
**Network** : Many M2M devices operate on the same cellular networks that are used by the everyday mobile devices such as Smartphone, tablets etc. Getting M2M devices certified on carrier network is a challenging task because of the onerous approval process established by network carriers. As voice subscribers are getting saturated, wireless carriers are looking more in to M2M devices for future growth.  
  
**Software** : The real benefit of M2M solutions comes from taking data from a remote device and integrating with backend systems.



**WATER FLOW SENSOR**



**Specification**

Working voltage 5V-24V

Maximum current 15 mA（DC 5V）

Weight 43 g

External diameters 20mm

Flow rate range 1～30 L/min

Operating temperature 0℃～80℃

Liquid temperature <120 C

**WORKING PRINCIPLE**

The Arduino flow meter works on the principle of the Hall effect. According to the Hall effect, a voltage difference is induced [in a conductor](https://eepower.com/textbook/vol-i-foundations-power-design/chapter-1-introduction-power/resistors-and-ohm-s-law) transverse to the electric current and the magnetic field perpendicular to it. Here, the Hall effect is utilized in the flow meter using a small fan/propeller-shaped rotor, which is placed in the path of the liquid flowing.

The liquid pushes against the fins of the rotor, causing it to rotate. The shaft of the rotor is connected to a Hall effect sensor. It is an arrangement of a current flowing coil and a magnet connected to the shaft of the rotor, thus a voltage/pulse is induced as this rotor rotates. In this flow meter, for every liter of liquid passing through it per minute, it outputs about 4.5 pulses. This is due to the changing magnetic field caused by the magnet attached to the rotor shaft as seen in the picture below. We measure the number of pulses using an Arduino and then calculate the flow rate in liters per hour (L/hr) using a simple conversion formula explained in step 2.

**APPLICATIONS**

Flow sensors are devices which are used to measure a flow rate of Fluid. These sensors are generally part of a flow meter that would help to measure the flow rate.

Nowadays many different types of flow sensors are available which are used for different purpose.

Flow measurement is the diverse area and it is necessary for different engineering operations. There are different types of flow sensors which are designed to measure mass flow and pressure sensing in different types of applications such as gas meter, chemicals, process auto-control, medical, HVAC, food, and beverages.

**FINAL CODE**

int flowPin = 2; //This is the input pin on the Arduino

double flowRate; //This is the value we intend to calculate.

volatile int count; //This integer needs to be set as volatile to ensure it updates correctly during the interrupt process.

#include <GSMSim.h>

#define RX 8

#define TX 7

#define RESET 2

#define BAUD 9600

const int soil\_sensor = A0; // Analog input pin that the soil moisture sensor is attached to

int sensorValue = 0;

#include <SoftwareSerial.h>

SoftwareSerial mySerial(9, 10);

GSMSim gsm;

void setup() {

// put your setup code here, to run once:

pinMode(flowPin, INPUT); //Sets the pin as an input

attachInterrupt(0, Flow, RISING); //Configures interrupt 0 (pin 2 on the Arduino Uno) to run the function "Flow"

Serial.begin(9600); //Start Serial

gsm.start();

}

void loop()

{

// put your main code here, to run repeatedly:

count = 0; // Reset the counter so we start counting from 0 again

interrupts(); //Enables interrupts on the Arduino

delay (1000); //Wait 1 second

noInterrupts(); //Disable the interrupts on the Arduino

//Start the math

flowRate = (count \* 2.25); //Take counted pulses in the last second and multiply by 2.25m

flowRate = flowRate \* 60; //Convert seconds to minutes, giving you mL / Minute

flowRate = flowRate / 1000; //Convert mL to Liters, giving you Liters / Minute

Serial.println(flowRate); //Print the variable flowRate to Serial

if(flowRate>=20)

{

Serial.println("GSMSim Library - SMS Example");

Serial.println("");

delay(1000);

gsm.start(); // baud default 9600

//gsm.start(BAUD);

Serial.println("Changing to text mode.");

gsm.smsTextMode(true); // TEXT or PDU mode. TEXT is readable :)

char\* number = "8019665124";

char\* message = "WATER IS COMING!!! SWITCH ON THE PUMP"; // message lenght must be <= 160. Only english characters.

Serial.println("Sending Message --->");

Serial.println(gsm.smsSend(number, message)); // if success it returns true (1) else false (0)

delay(2000);

Serial.println("Listing unread message(s).");

Serial.println(gsm.smsListUnread()); // if not unread messages have it returns "NO\_SMS"

Serial.println("Read SMS on index no = 1");

Serial.println(gsm.smsRead(1)); // if no message in that index, it returns IXDEX\_NO\_ERROR

}

else

{

sensorValue = analogRead(soil\_sensor);

// print the sensor results to the serial monitor:

Serial.print("Moisture Value = " );

Serial.println(sensorValue);

// delay of one second

delay(1000);

if(sensorValue<=200)

{

Serial.println("GSMSim Library - SMS Example");

Serial.println("");

delay(1000);

gsm.start(); // baud default 9600

//gsm.start(BAUD);

Serial.println("Changing to text mode.");

gsm.smsTextMode(true); // TEXT or PDU mode. TEXT is readable :)

char\* number = "8019665124";

char\* message = "WATER STOPPED!!! SWITCH OFF THE PUMP"; // message lenght must be <= 160. Only english characters.

Serial.println("Sending Message --->");

Serial.println(gsm.smsSend(number, message)); // if success it returns true (1) else false (0)

delay(2000);

Serial.println("Listing unread message(s).");

Serial.println(gsm.smsListUnread()); // if not unread messages have it returns "NO\_SMS"

Serial.println("Read SMS on index no = 1");

Serial.println(gsm.smsRead(1)); // if no message in that index, it returns IXDEX\_NO\_ERROR

}}}

void Flow()

{

count++; //Every time this function is called, increment "count" by 1

}