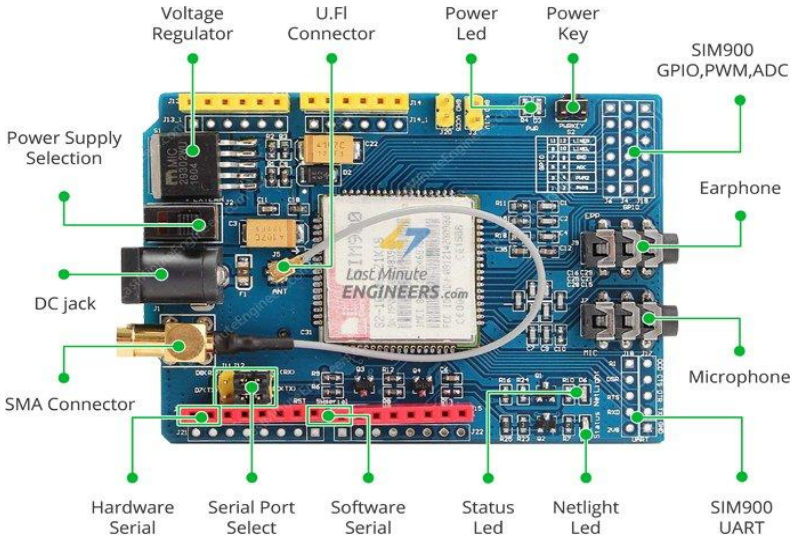
Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Foundation Skills in Sensor Interfacing (01CT11032)	**Aim:** To interface GSM SIM900 with Arduino.	
Experiment No: 11	**Date: 24-01-22**	**Enrolment No: 92100133020**

Aim: To interface GSM SIM900 with Arduino.

Apparatus: GSM SIM900, SIM card, Arduino UNO R3, USB-Cable, Jumper Wires.

Theory:

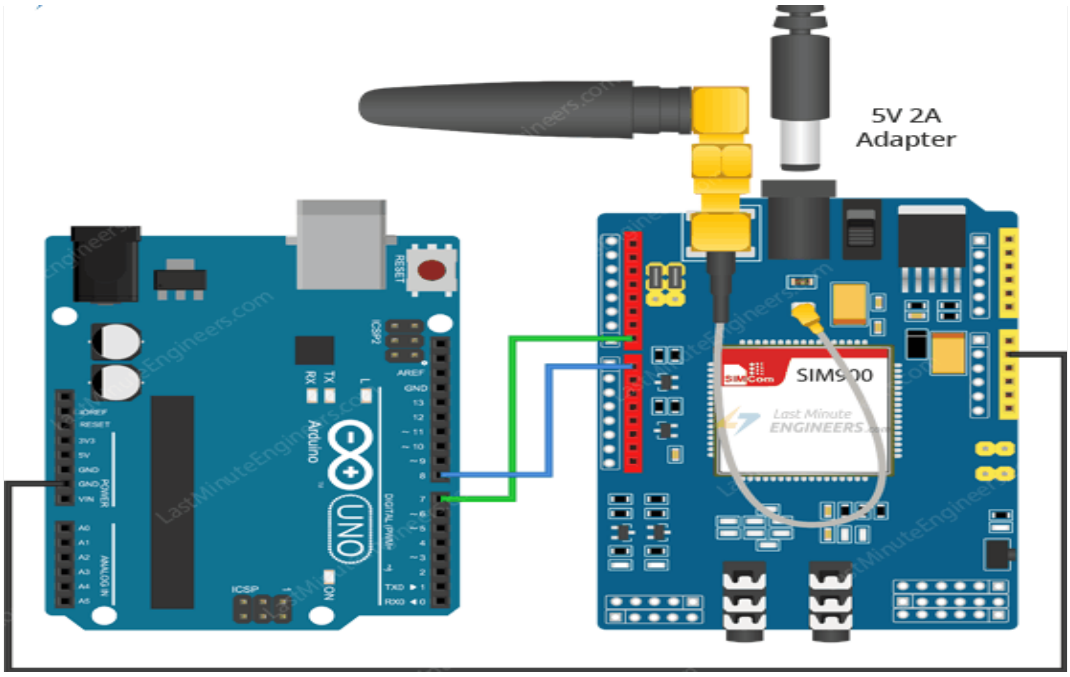


Talking about its working principle, it consists of some important parts which are responsible for connection of GSM SIM900 with Arduino and its working. As given below

- UART Communication: The SIM900 GSM/GPRS shield uses UART protocol to communicate with an Arduino.
- Earphone and Microphone Jack: It allows you to use SIM900's audio interface to make or receive voice calls.
- Antenna: An antenna is required to transmit and receive the frequencies or we can say to capture and receive the signal which are incoming and outgoing from the module.
- SIM Socket: There's a SIM socket on the back. Any activated, 2G full-size SIM card would work perfectly. As of now this module is not that much advance so that it can support 4G connection. To do that we have to use another module.
- RTC(Real Time Clock): This will keep the time even when the power is OFF. This needed because we need to keep reference of accurate time while we are establishing any communication.

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Foundation Skills in Sensor Interfacing (01CT11032)	**Aim:** To interface GSM SIM900 with Arduino.	
Experiment No: 11	**Date: 24-01-22**	**Enrolment No: 92100133020**

Interfacing Diagram:



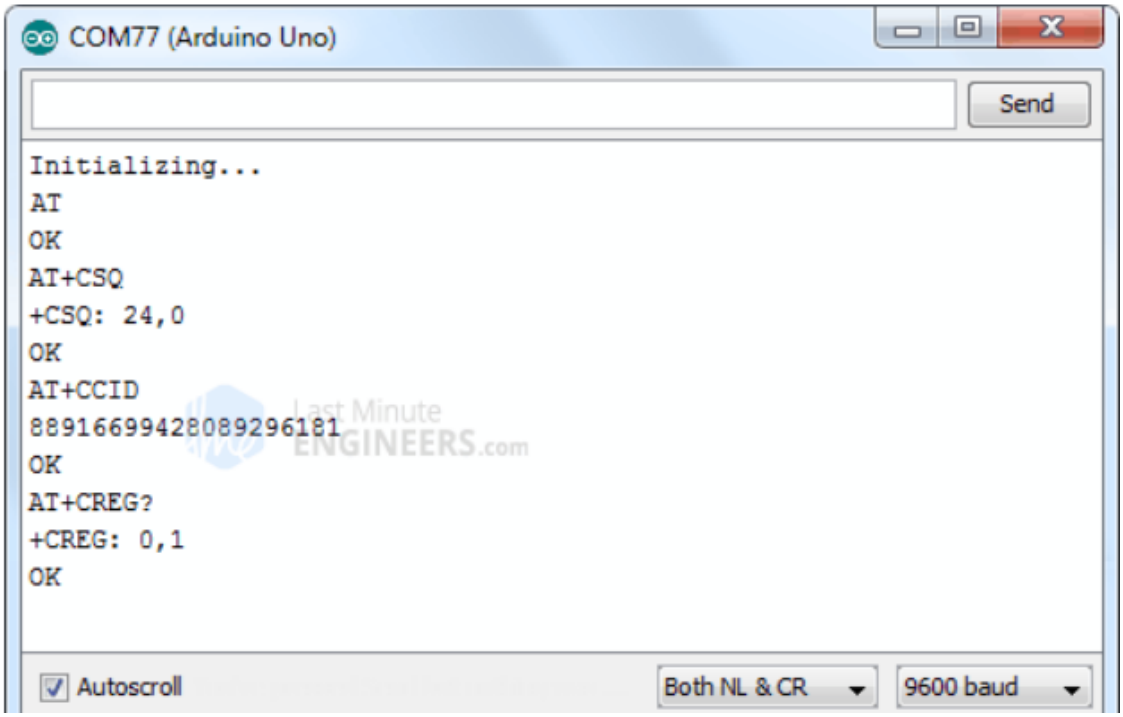
Code:

```
#include<SoftwareSerial.h>
//create software serial object to communicate with SIM900
SoftwareSerial mySerial (7,8);  // connect 7th and 8th pin as TX and RX to Arduino

void setup() {
 Serial.begin(9600); // begin serial communication with Arduino and IDE
 mySerial.begin(9600); //begin serial communication with Arduino and SIM900
 Serial.println("Initializing..."); // It will be displayed until the SIM900 is initialized
 delay(1000);
 mySerial.println("AT"); //Check that SIM900 is working or not
```

```
 updateSerial();
 mySerial.println("AT+CSQ"); //check signal quality, it gives any number between 0 to 31
 updateSerial();
 mySerial.println("AT+CCID"); //check SIM information, it gives SIM card number
 updateSerial();
 mySerial.println("AT+CREG?"); //check whether it has registered in network or not (give 0 or 1)
 updateSerial();
}

void loop() {
 updateSerial(); //Gets the data from SIM900 and send it to serial monitor
}
void updateSerial();
 {
  delay(500);
  while(Serial.available()) // this function gets only available data or value from Serial monitor
  {
    mySerial.write(Serial.read()); //Forward to Software Serial or SIM900 whatever serial monitor(Arduino) has received
  }
}
}
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Foundation Skills in Sensor Interfacing (01CT11032)	**Aim:** To interface GSM SIM900 with Arduino.	
Experiment No: 11	**Date: 24-01-22**	**Enrolment No: 92100133020**

Observations:



Conclusion:

From the above experiment I got to know about GSM SIM900 and its working principle. Also I got an idea to how to implement this experiment in our regular projects. We can also use this type of communication device in home automation projects, automatic water supply project, etc. We can also perform tasks from the very long distance. For eg. If I want to water my farm automatically, then I have to just setup that kind of arrangement that when I do a call to this module it will turn ON or turn HIGH the relay attached to it and the water pump start. At a specific time interval it will turn OFF the relay and hence the water pump is stopped. To perform this type of tasks we can utilize this module.

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Foundation Skills in Sensor Interfacing (01CT11032)	**Aim:** To interface GSM SIM900 with Arduino.	
Experiment No: 11	**Date: 24-01-22**	**Enrolment No: 92100133020**

Post Session Exercise:

TASK 1

Explain SoftwareSerial, mySerial.available, mySerial.read and mySerial.write.

Serial.available() : Get the number of bytes (characters) available for reading from the serial port. This is data that's already arrived and stored in the serial receive buffer (which holds 64 bytes)
mySerial.available() : Get the number of bytes (characters) available for reading from a software serial port. This is data that's already arrived and stored in the serial receive buffer.
Serial.read() : The Serial.read( ) in Arduino reads the incoming serial data in the Arduino. The int data type is used here. It returns the first data byte of the arriving serial data. It also returns -1 when no data is available on the serial port.
mySerial.read() : Return a character that was received on the RX pin of the software serial port. Note that only one SoftwareSerial instance can receive incoming data at a time (select which one with the listen() function.
Serial.write() : Writes binary data to the serial port. This data is sent as a byte or series of bytes; to send the characters representing the digits of a number use the print() function instead.
mySerial.write() : Prints data to the transmit pin of the software serial port as raw bytes. Works the same as the Serial.write() function.
Explain the functions of all the LED present in SIM900.
 PWR: This LED is connected to the shield's power supply line. If this LED is on, the shield is receiving power. Status: This LED indicates SIM900's working status. If this LED is on, the chip is in working mode.
 Status: This LED indicates SIM900's working status. If this LED is on, the chip is in working mode.
 Net light: This LED indicates the status of your cellular network. It'll blink at various rates to show what state it's in.

 Difference between hardware serial and software serial.

 A hardware serial, as the name suggests, denotes that a dedicated piece of hardware (UART) enables Serial communication. In Arduino Uno, for instance, pins 0 and 1 have UART support, and they are connected to the USB via a USB-to-UART converter. That facilitates communication between your computer/laptop and the Arduino. While Arduino Uno has a single Hardware Serial, other boards like Mega have multiple. They are accessed using Serial, Serial1, Serial2, and so on.

Software serial is a library that replicates the hardware serial behaviour on other digital pins of the Arduino, using (you guessed it) software. You can have more than one software serials running in parallel, with the limitation being that only one can receive data at a time.

	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Foundation Skills in Sensor Interfacing (01CT11032)	**Aim:** To interface GSM SIM900 with Arduino.	
Experiment No: 11	**Date: 24-01-22**	**Enrolment No: 92100133020**

Code :

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(7, 8); //SIM900 Tx & Rx is connected to Arduino #7 & #8
void setup()
{
//Begin serial communication with Arduino and Arduino IDE (Serial Monitor)
Serial.begin(9600);
//Begin serial communication with Arduino and SIM900
mySerial.begin(9600);
Serial.println("Initializing...");
delay(1000);
mySerial.println("AT"); //Handshaking with SIM900
updateSerial();
mySerial.println("AT+CMGF=1"); // Configuring TEXT mode
updateSerial();
mySerial.println("AT+CMGS=\"+91\"");// after execution of this an sms will be send on this
number”9172944661”
updateSerial();
```

	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Foundation Skills in Sensor Interfacing (01CT11032)	**Aim:** To interface GSM SIM900 with Arduino.	
Experiment No: 11	**Date: 24-01-22**	**Enrolment No: 92100133020**

```
mySerial.print("Code is successful"); //whatever written in double quotes in mySerial.print function will be send as an sms to the above number
updateSerial();
mySerial.write(26);//CTRL+Z
}
void loop()
{
}
void updateSerial()
{
delay(500);
while (Serial.available())
{
mySerial.write(Serial.read());//Forward what Serial received to Software Serial Port
}
while(mySerial.available())
{
Serial.write(mySerial.read());//Forward what Software Serial received to Serial Port
}
}
```