How to Interface GSM Module to Arduino-Send and Receive SMS

Last Updated on December 23, 2016 by jojo in Arduino with 59 Comments

In this article, we are going to see how to interface **GSM Module to Arduino**. There are different kinds of GSM modules available in market. We are using the most popular module based on **Simcom SIM900** and **Arduino Uno** for this tutorial. Interfacing a GSM module to Arduino is pretty simple. You only need to make 3 connections between the gsm module and arduino. So lets get to business!

A **GSM Module** is basically a GSM Modem (like SIM 900) connected to a PCB with different types of output taken from the board – say TTL Output (for Arduino, 8051 and other microcontrollers) and RS232 Output to interface directly with a PC (personal computer). The board will also have pins or provisions to attach mic and speaker, to take out +5V or other values of power and ground connections. These type of provisions vary with different modules.

Lots of varieties of GSM modem and GSM Modules are available in the market to choose from. For our project of connecting a gsm modem or module to arduino and hence send and receive sms using arduino – its always good to choose an ***arduino compatible GSM Module*** – that is a GSM module with TTL Output provisions.

GSM Module – Buyers Guide – are you looking to buy a GSM module? There are a handful of product variants for GSM module – like SIM900, SIM300, SIM800 etc. We have created this buyers guide to help you select the right GSM module for your project needs.

Notes on GSM Module

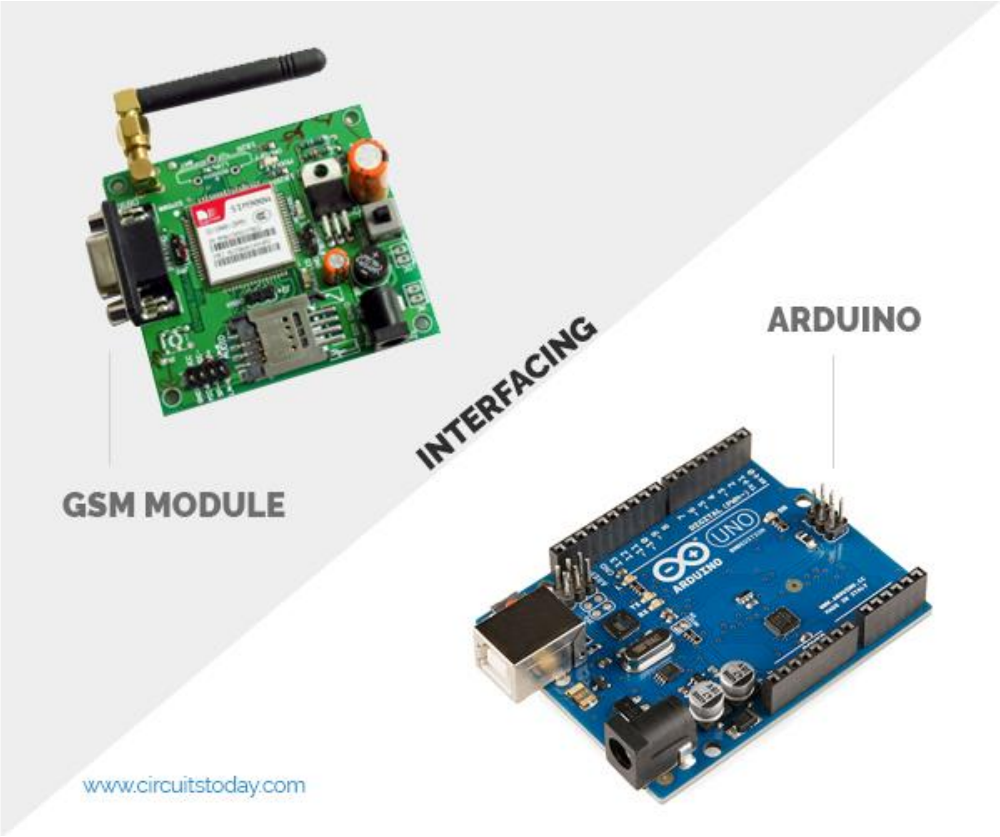
1. We use SIM900 GSM Module – This means the module supports communication in 900MHz band. We are from India and most of the mobile network providers in this country operate in the 900Mhz band. If you are from another country, you have to check the mobile network band in your area. A majority of **United States** mobile networks operate in 850Mhz band (the band is either 850Mhz or 1900Mhz). **Canada** operates primarily on 1900 Mhz band. Please read this wiki entry on **GSM Frequency Bands around the World**.

2. Check the power requirements of GSM module – GSM modules are manufactured by different companies. They all have different input power supply specs. You need to double check your GSM modules power requirements. In this tutorial, our gsm module requires a 12 volts input. So we feed it using a 12V,1A DC power supply. I have seen gsm modules which require 15 volts and some other types which needs only 5 volts input. They differ with manufacturers. If you are having a 5V module, you can power it directly from Arduino’s 5V out.

Note:- GSM Modules are manufactured by connecting a particular GSM modem to a PCB and then giving provisions for RS232 outputs, TTL outputs, Mic and Speaker interfacing provisions etc. The most popular modem under use is SIM 900 gsm modem from manufacturer SIMCom. They also manufacture GSM Modems in bands 850, 300 and other frequency bands.

3. Check for TTL Output Pins in the module – You can feed the data from gsm module directly to Arduino only if the module is enabled with TTL output pins. Otherwise you have to convert the **RS232 data to TTL** using MAX232 IC and feed it to Arduino. Most of the gsm modules in market are equipped with TTL output pins. Just ensure you are buying the right one.

So that’s all about the gsm module basics. Now lets power it up!



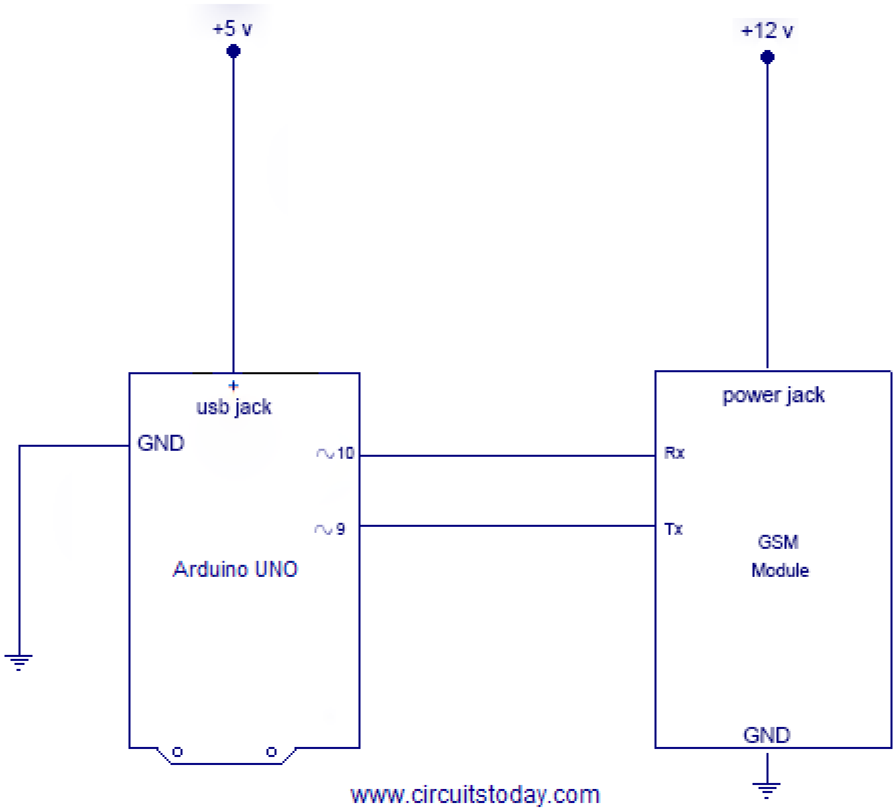
Booting the GSM Module!

1. Insert the SIM card to GSM module and lock it.
2. Connect the adapter to GSM module and turn it ON!
3. Now wait for some time (say 1 minute) and see the blinking rate of 'status LED' or 'network LED' (GSM module will take some time to establish connection with mobile network)
4. Once the connection is established successfully, the status/network LED will blink continuously every 3 seconds. You may try making a call to the mobile number of the sim card inside GSM module. If you hear a ring back, the gsm module has successfully established network connection. Okay! Now let's see how to connect a gsm module to Arduino!

Connecting GSM Module to Arduino

There are two ways of connecting GSM module to arduino. In any case, the communication between Arduino and GSM module is serial. So we are supposed to use serial pins of Arduino (Rx and Tx). So if you are going with this method, you may connect the Tx pin of GSM module to Rx pin of Arduino and Rx pin of GSM module to Tx pin of Arduino. You read it right ? **GSM Tx –> Arduino Rx** and **GSM Rx –> Arduino Tx**. Now connect the ground pin of arduino to ground pin of gsm module! So that's all! You made 3 connections and the wiring is over! Now you can load different programs to communicate with gsm module and make it work.
Note:- The problem with this connection is that, while programming Arduino uses serial ports to load program from the Arduino IDE. If these pins are used in wiring, the program will not be loaded successfully to Arduino. So you have to disconnect wiring in Rx and Tx each time you burn the

program to arduino. Once the program is loaded successfully, you can reconnect these pins and have the system working!
To avoid this difficulty, I am using an alternate method in which two digital pins of arduino are used for serial communication. We need to select two **PWM enabled pins of arduino** for this method. So I choose pins **9** and **10** (which are PWM enabled pins). This method is made possible with the **SoftwareSerial Library** of Ardunio. SoftwareSerial is a library of Arduino which enables serial data communication through other digital pins of Arduino. The library replicates hardware functions and handles the task of serial communication.
I hope you understood so far! Lets get to the circuit diagram! So given below is the circuit diagram to connect gsm module to arduino – and hence use the circuit to send sms and receive sms using arduino and gsm modem.



Make the connections as shown! Now lets get to the coding part. The program has two objectives as described below:-

1) Send SMS using Arduino and GSM Module – to a specified mobile number inside the program
2) Receive SMS using Arduino and GSM Module – to the SIM card loaded in the GSM Module.

The Program

```
 #include <SoftwareSerial.h>




SoftwareSerial mySerial(9, 10);
```

```
  }
```

```
void setup()
{
  mySerial.begin(9600);   // Setting the baud rate of GSM Module
  Serial.begin(9600);    // Setting the baud rate of Serial Monitor (Arduino)
  delay(100);
}



void loop()
{
  if (Serial.available()>0)
   switch(Serial.read())
  {
    case 's':
      SendMessage();
      break;
    case 'r':
      RecieveMessage();
      break;
  }
```

```
 if (mySerial.available()>0)

   Serial.write(mySerial.read());

}




 void SendMessage()

{

  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode

  delay(1000);  // Delay of 1000 milli seconds or 1 second

  mySerial.println("AT+CMGS=\"+91xxxxxxxxxx\"\r"); // Replace x with mobile number

  delay(1000);

  mySerial.println("I am SMS from GSM Module");// The SMS text you want to send

  delay(100);

   mySerial.println((char)26);// ASCII code of CTRL+Z

  delay(1000);

}




 void RecieveMessage()

{
  mySerial.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a live SMS

  delay(1000);

 }
```

so that’s the program/code to make arduino send sms and receive sms using gsm module! Let’s get to explanation of program!

The Program Explanation

We begin by including **SoftwareSerial** library into the program. In the next line, we create a ***constructor*** of SoftwareSerial with name **mySerial** and we pass the digital pin numbers as parameters. The actual format is like **SoftwareSerial mySerial (Rx, Tx);**

So in our code, pin number 9 will act as Rx of Arduino and 10 will act as Tx of Arduino. Lets get to the configuration part of program inside setup. The first task is to ***set baud rates*** of SoftwareSerial library to communicate with GSM module. We achieve this by invoking ***mySerial.begin*** function. Our second task is to set the baud rate of Arduino IDE’s Serial Monitor. We do this by invoking ***Serial.begin***function. Both should be set at the same baud rate and we use 9600 bits/second here in our tutorial. Configuration part is over with setting baud rates and its good to give a small delay of 100 milli seconds.

Now lets get to the actual program inside loop(). To make things simpler, I have developed a user input based program. The program seeks user input via **serial monitor of Arduino**. If the input is **‘s’** the program will invoke function to **send an sms from GSM module**. If the user input is **‘r’**, the program will invoke the function to **receive a live SMS** from GSM module and display it on serial monitor of Arduino. The whole program is as simple as that!

Serial.available() – checks for any data coming through serial port of arduino. The function returns the number of bytes available to read from serial buffer. If there is no data available, it returns a -1 (value less than zero).

Serial.read() – Reads all the data available on serial buffer (or incoming serial data if put otherwise). Returns the first byte of incoming serial data.

mySerial.available() – checks for any data coming from GSM module through the SoftwareSerial pins 9 and 10. Returns the number of bytes available to read from software serial port. Returns a -1 if no data is available to read.

mySerial.read() – Reads the incoming data through software serial port.

Serial.write() – Prints data to serial monitor of arduino. So the function Serial.write(mySerial.read()) – prints the data collected from software serial port to serial monitor of arduino.

Lets get the functions **SendMessage()** and **RecieveMessage()**

These are the functions in which we actually send commands to GSM module from Arduino. These commands to communicate with GSM module are called AT Commands. There are different commands to perform different tasks using the GSM module. You can read complete AT Commands Library to understand all that is possible with GSM module.

SendMessage() – is the function we created in our arduino sketch to send an SMS. To send an SMS, we should set our GSM module to Text mode first. This is achieved by sending an AT

Command “AT+CMGF=1” We send this command by writing this to SoftwareSerial port. To achieve this we use the mySerial.println() function. mySerial.println writes data to software serial port (the Tx pin of our Software Serial – that is pin 10) and this will be captured by GSM module (through its Rx pin). After setting the GSM module to Text mode, we should the the mobile number to which we shall send the SMS. This is achieved with AT command “AT+CMGS=\”+91xxxxxxxxxx\”\r” – where you may replace all x with the mobile number.
In next step, we should send the actual content of SMS. The end of SMS content is identified with CTRL+Z symbol. The ASCII value of this CTRL+Z is 26. So we send a char(26) to GSM module using the line mySerial.println((char)26); Each and every AT command may be followed by 1 second delay. We must give some time for GSM module to respond properly. Once these commands are send to GSM module, you shall receive an SMS in the set mobile number.

RecieveMessage() – is the function to receive an SMS (a live SMS). The AT command to receive a live SMS is “AT+CNMI=2,2,0,0,0” – we just need to send this command to GSM module and apply a 1 second delay. Once you send this command, try sending an SMS to the SIM card number put inside GSM module. You will see the SMS you had sent displayed on your Arduino serial monitor. There are different AT commands for different tasks. If you want to read all SMS’s stored in your SIM card, send the following AT Command to gsm module – “AT+CMGL=\”ALL\”\r”

Okay! That’s all and you are done. And you have learnt how to use arduino to send sms and receive sms message with example code.

I shall summarize this tutorial on how to send/receive a text message using arduino and gsm module with the following notes:-

AT Commands to Send SMS using Arduino and GSM Module

```
AT+CMGF=1 // Set the GSM module in text mode

AT+CMGS=\"+YYxxxxxxxxxx\"\r // Input the mobile number| YY is country code

“the message” with stopping character (char)26 // ASCII of ctrl+z
```

AT Commands to Receive SMS using Arduino and GSM Module

```
AT+CMGF=1 // Set the GSM Module in text mode

AT+CNMI=2,2,0,0,0 // AT Command to receive live sms
```

Read the AT commands library and start playing with your GSM module and Arduino!