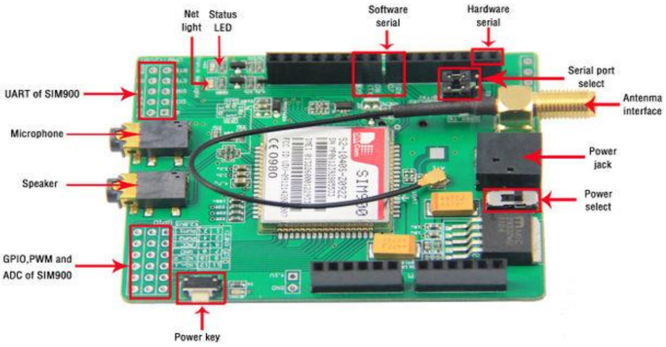
GSM Module and Arduino Uno

The GPRS Shield is based on SIM900 module from SIMCOM and compatible with Arduino and its clones. The GPRS Shield provides you a way to communicate using the GSM cell phone network. The shield allows you to achieve SMS, MMS, GPRS and Audio via UART by sending AT commands .The shield also has the 12 GPIOs, 2 PWMs and an ADC of the SIM900 module (They are all 2V8 logic) present onboard.

Hardware Diagram of GPRS Shield



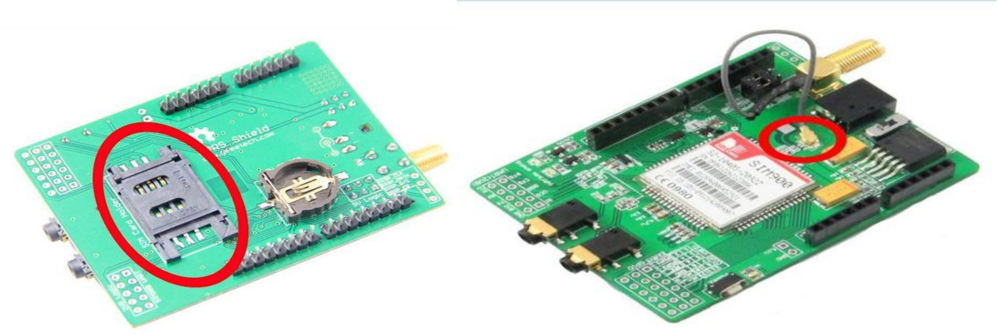
- Power select - select the power supply for GPRS shield(external power or 5v of Arduino)
- Power jack - connected to external 4.8~5VDC power supply
- Antenna interface - connected to external antenna
- Serial port select - select either software serial port or hardware serial port to be connected to **GPRS Shield**
- Hardware Serial - D0/D1 of Arduino
- Software serial - D7/D8 of Arduino
- Status LED - tell whether the power of SIM900 is on
- Net light - tell the status about SIM900 linking to the net
- UART of SIM900 - UART pins breakout of SIM900
- Microphone - to answer the phone call
- Speaker - to answer the phone call
- GPIO,PWM and ADC of SIM900 - GPIO,PWM and ADC pins breakout of SIM900
- Power key - power up and down for SIM900

Pin usage on Arduino

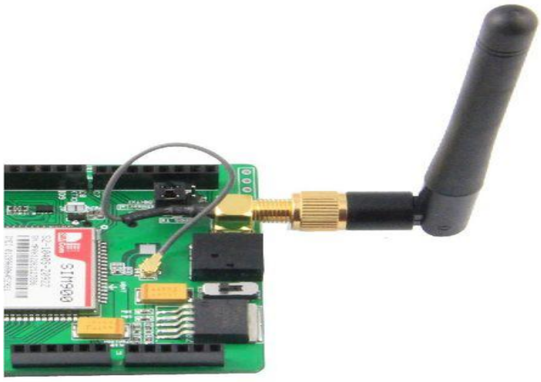
- Pin 7,8 for Communication and Pin 9 for control reset signal
- Pin 4,5,6 and 11 for Fan ,Street Lights ,Lamp ,And Door(Servo Motor) respectively .

Getting Started

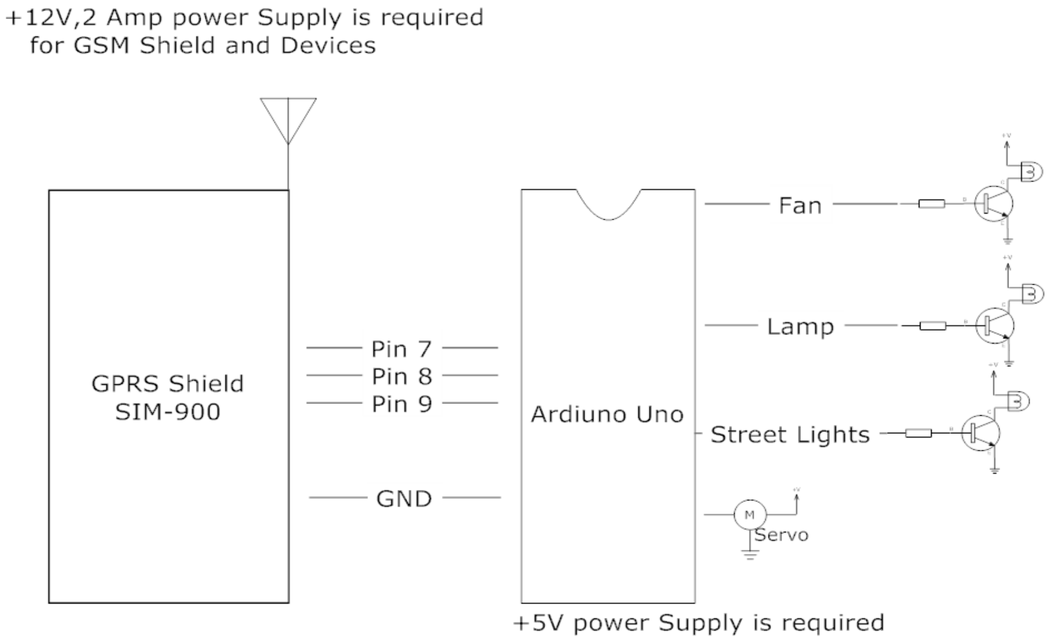
- **Insert a unlock SIM card and Make sure the antenna pad buckled properly**



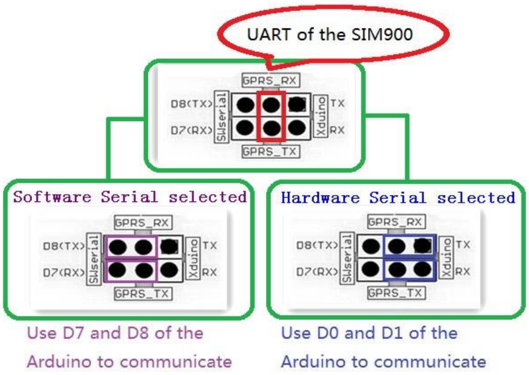
- **Install the GSM antenna**



Assemble the Circuit as shown below


+12V,2 Amp power Supply is required
for GSM Shield and Devices
GPRS Shield
SIM-900
Pin 7
Pin 8
Pin 9
GND
Ardiuno Uno
Fan
Lamp
Street Lights
M
Servo
+5V power Supply is required

Choose communication port properly using the jumpers



Monitor /Observe the LED status

Press power key about 2 seconds to turn on the GPRS shield

Indicator LEDs

There are three indicator LEDs (PWR, Status, Network) on the GPRS Shield, users can know about the working state of the shield based on the three indicator LEDs. Detailed information please refer to the following table:

LEDs	Status	Description
Power	ON	When the Power is available ,Otherwise OFF
Status	ON	When the GPRS shield is ON otherwise OFF

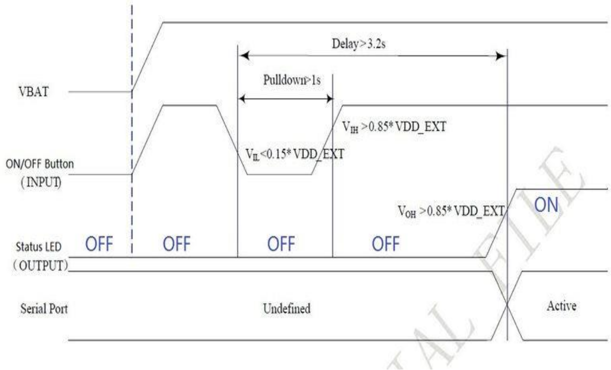
Net Work	64mS ON /800mS OFF 64mS ON /3000mS OFF 64mS ON /300mS OFF OFF	SIM900/GPRS has not Logged on the Network SIM900/GPRS has Logged on the Network SIM900/GPRS Communication OFF

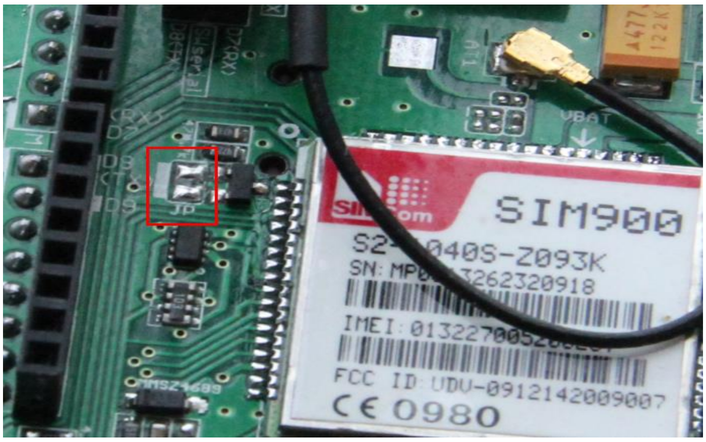
Power Up and Power down the GPRS Shield

Power up the GPRS Shield

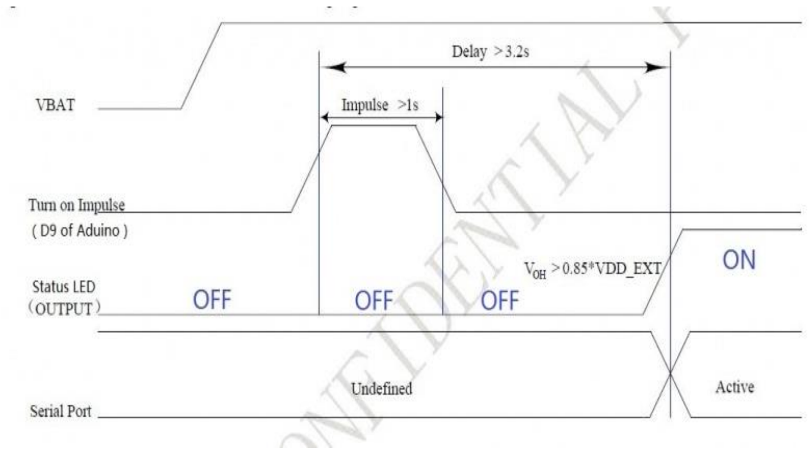
The GPRS Shield can be turned on by two ways:

Hardware Triger; Press the ON/OFF Button about two seconds. The power up scenarios illustrates as following figure:





Software Triger; If use this way to power up the GPRS Shield, JP need to be soldered, then Digital Pin 9 of the Arduino will act as Software Triger port and Digital Pin 9 cannot be use as other purpose. Then give Digital Pin 9 a Turn on Impulse can power up the GPRS Shield. The power up scenarios illustrates as following figure:



The following code is power up subroutine for Arduino if using software triger:

```
void powerUp()
{
  pinMode(9, OUTPUT);
  digitalWrite(9,LOW);
  delay(1000);
  digitalWrite(9,HIGH);
  delay(2000);
  digitalWrite(9,LOW);
  delay(3000);
}
```

When power on procedure completes, the SIM900 will send out following result code to indicate the GPRS shield is ready to operate; When set as fixed baud rate, the SIM900 will send out result code: RDY This result code does not appear when auto baud rate is active.

Upload Sketch/Code to Arduino

```
/*------------------------------------------------------------------------
  * GSM(sMs) based peripherals controll
  * Author: Malik
  * Purpose : Openday Demonsteration
  ------------------------------------------------------------------------*
/#include <Servo.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(7, 8);

String Check="OK";
String Recieved_data="";
String CMD="";

String Codedstring[5][2]={
                                                {"#EED 1 OFF","#EED 1 ON"},
                                                {"#EED 2 OFF","#EED 2 ON"},
                                                {"#EED 3 OFF","#EED 3 ON"},
                                                {"#EED 4 OFF","#EED 4 ON"},
                                                {"STUPID","#EED ALL OFF"}
                                           };


const int Lights =4;        //Pin 4
const int Fan =5;               //Pin 5
const int Lamp=6;               //Pin6
//int SIZE=0;
Servo Door_servo;               //Pin 11

void RecieveMessage();
void powerUp();
void All_Off();
void crazy();
void RecievedData();

void setup()
{
    String Responce="";
    pinMode(Lights,OUTPUT);// For Lights
    pinMode(Fan,OUTPUT);// for Fan drive
    pinMode(Lamp,OUTPUT);// for Lamp drive
//    pinMode(Door_servo,OUTPUT);// for Door drive

    Door_servo.attach(11);
    pinMode(9,OUTPUT);// for SiM900 to get reset Signal through the PIN

    All_Off();
    powerUp();// To force SIM900 to log-on the network by resting method

    mySerial.begin(19200);          // Setting the baud rate of GSM Module

    Serial.begin(19200);              // Setting the baud rate of Serial Monitor (Arduino)

    delay(100);


    do{

          Responce="";

          mySerial.println("AT");

          delay(2000);

          Responce=mySerial.readString();

          Serial.println(Responce);

        }while(Responce==Check);

        ;

    do{

          Responce="";

          mySerial.println("AT+CMGF=1");

          delay(2000);

          Responce=mySerial.readString();

          Serial.println(Responce);

        }while(Responce==Check);


  RecieveMessage(); // AT Command to receive a live SMS

  mySerial.flush() ;


}


/*     ************************************************************************

  *                                                                  Main Loop

  ************************************************************************/

void loop()

{

  Recieved_data="";


  if(mySerial.available()>0)

  {

    RecievedData();
```

```
if(CMD == Codedstring[0][0] )                    //Turn OFF
        {
          digitalWrite(Lights, LOW);
          Serial.print(CMD);CMD="";
          }
else if(CMD == Codedstring[0][1])         //Turn ON
        {
          digitalWrite(Lights, HIGH);
          Serial.print(CMD);CMD="";
          }

else if(CMD == Codedstring[1][0])              //Turn OFF
        {
          digitalWrite(Fan, LOW);
          Serial.print(CMD);CMD="";
          }
        else if(CMD == Codedstring[1][1])      //Turn ON
        {
          digitalWrite(Fan, HIGH);
          Serial.print(CMD);CMD="";
          }
else if(CMD == Codedstring[2][0])         //Close the Door
      {
        Door_servo.write(0);
        Serial.print(CMD);CMD="";
        }
else if(CMD == Codedstring[2][1])         //Open the Door
        {
          for(int i=0; i<=110; i+=5){
                  Door_servo.write(i);delay(50);}
            Serial.print(CMD);CMD="";
          }
else if(CMD == Codedstring[3][0])         //Turn OFF
      {
        digitalWrite(Lamp, LOW);
        Serial.print(CMD);CMD="";
```

```
            }
      else if(CMD == Codedstring[3][1])         //Turn ON
            {
              digitalWrite(Lamp, HIGH);
              Serial.print(CMD);CMD="";
              }
      else if(CMD == Codedstring[4][0])         //Call a function crazy
          {
            crazy();
            Serial.print(CMD);CMD="";
            }
      else if(CMD == Codedstring[4][1])         //Call a function All_Off
            {
              All_Off();
              Serial.print(CMD);CMD="";
            }

      else
        {
          Serial.println("No/Wrong CMD"); //Do Nothing
          Serial.print(CMD);CMD="";
          mySerial.flush() ;
          }
  }

else ;
//mySerial.println("AT+CMGD=1,4"); // delete all SMS

}

/*      **************************************************************************
  *                           End of main Loop
  ****************************************************************************/



  /*void SendMessage()
```

```
{
    mySerial.println("AT+CMGF=1");          //Sets the GSM Module in Text Mode
    delay(1000);    // Delay of 1 second
    mySerial.println("AT+CMGS=\"+971551008980"\r"); // mobile number
    delay(1000);
    mySerial.println("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+CMGF=1");
    //delay(1000);
    mySerial.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a live SMS
    delay(1000);
  }

void All_Off()
{
    digitalWrite(Lights,LOW);// turn OFF
    digitalWrite(Fan, LOW); // turn OFF
    digitalWrite(Lamp, LOW);// Turn OFF
    Door_servo.write(0);// To Keep Close the Door initially
}

void All_On()
{
    digitalWrite(Lights,HIGH);// turn OFF
    //digitalWrite(Fan, HIGH); // turn OFF
    digitalWrite(Lamp, HIGH);// Turn OFF
    Door_servo.write(110);// To open the door fully
}
```

```
void crazy()
{
    while(1)
    {
        All_Off();
        delay(1000);//0.9Sec
        All_On();
        delay(900);//0.9Sec
        if(mySerial.available()>0)
              RecievedData();
        if(CMD == "SORRY") break;
        else;
    }
}

void RecievedData()
{
    //Recieved_data=mySerial.readString();
    //Serial.print(Recieved_data);

    CMD=(mySerial.readString().substring(49));
    //CMD=Recieved_data.substring(49);
    CMD.trim();
    //     SIZE=CMD.length();
    //     Serial.println("Size of CMD is Size");
    //       Serial.println(SIZE);
}

  void powerUp()
{

  digitalWrite(9,LOW);
  delay(1000);
  digitalWrite(9,HIGH);
  delay(2000);
  digitalWrite(9,LOW);

  delay(3000);

}
```