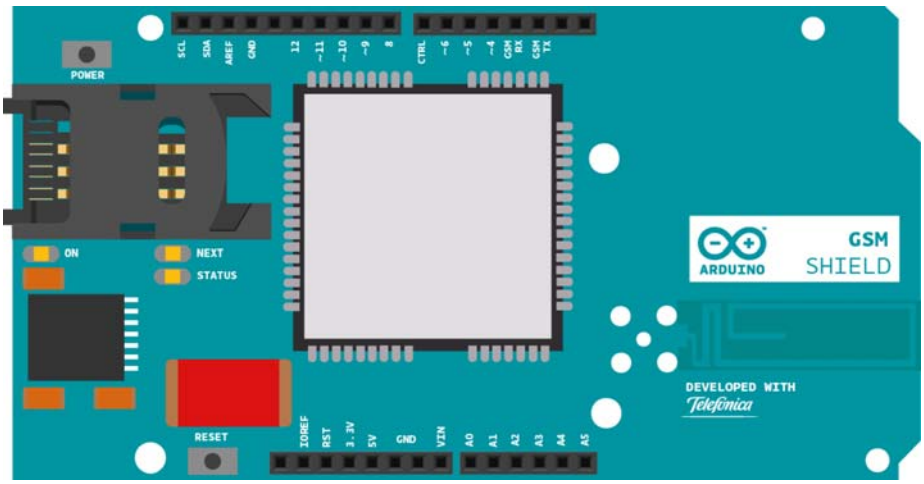
Send SMS

This sketch send a SMS message from an Arduino or Genuino board equipped with a GSM shield. Using the serial monitor of the Arduino Software (IDE), you'll enter the number to connect with, and the text message to send.

Hardware Required

- Arduino or Genuino Board
- Arduino + Telefonica GSM/GPRS Shield (//www.arduino.cc/en/Main/ArduinoGSMShield)
- SIM card

Circuit



(//www.arduino.cc/en/uploads/Tutorial/GSMShield_ArduinoUno.jpg)
image of the Arduino GSM Shield on top of an Arduino or Genuino board

Code

First, import the GSM library

```
#include <GSM.h>
```

SIM cards may have a PIN number that enables their functionality. Define the PIN for your SIM. If your SIM has no PIN, you can leave it blank :

```
#define PINNUMBER ""
```

Initialize instances of the classes you're going to use. You're going to need both the GSM and GSM_SMS class.

```
GSM gsmAccess;
GSM_SMS sms;
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=2)

In `setup`, open a serial connection to the computer. After opening the connection, send a message indicating the sketch has started.

```
void setup(){
  Serial.begin(9600);
  Serial.println("SMS Messages Sender");
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=3)

Create a local variable to track the connection status. You'll use this to keep the sketch from starting until the SIM is connected to the network :

```
boolean notConnected = true;
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=4)

Connect to the network by calling `gsmAccess.begin()`. It takes the SIM card's PIN as an argument. By placing this inside a `while()` loop, you can continually check the status of the connection. When the modem does connect, `gsmAccess()` will return `GSM_READY`. Use this as a flag to set the `notConnected` variable to `true` or `false`. Once connected, the remainder of `setup` will run.

```
while(notConnected)
  {
    if(gsmAccess.begin(PINNUMBER)==GSM_READY)
      notConnected = false;
    else
    {
      Serial.println("Not connected");
      delay(1000);
    }
  }
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=5)

Finish `setup` with some information to the serial monitor.

```
Serial.println("GSM initialized.");
}
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=6)

Create a function named `readSerial` of type `int`. You'll use this to iterate through input from the serial monitor, storing the number you wish to send an SMS to, and the message you'll be sending. It should accept a `char` array as an argument.

```
int readSerial(char result[])
{
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=7)

Create a variable to count through the items in the serial buffer, and start a `while` loop that will continually execute.

```
int i = 0;
  while(1)
  {
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=8)

As long as there is serial information available, read the data into a variable named `inChar`.

```
while (Serial.available() > 0)
    {
      char inChar = Serial.read();
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=9)

If the character being read is a newline, terminate the array, clear the serial buffer and exit the function.

```
if (inChar == '\n')
      {
        result[i] = '\0';
        Serial.flush();
        return 0;
      }
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=10)

If the incoming character is an ASCII character other than a newline or carriage return, add it to the array and increment the index. Close up the `while` loops and the function.

```
if(inChar!='\r')
      {
        result[i] = inChar;
        i++;
      }
    }
  }
}
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=11)

In `loop`, create a `char` array named `remoteNumber` to hold the number you wish to send an SMS to. Invoke the `readSerial` function you just created, and pass `remoteNumber` as the argument. When `readSerial` executes, it will populate `remoteNumber` with the number you wish to send the message to.

```
Serial.print("Enter a mobile number: ");
  char remoteNumber[20];
  readSerial(remoteNumber);
  Serial.println(remoteNumber);
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=12)

Create a new `char` array named `txtMsg`. This will hold the content of your SMS. Pass `txtMsg` to `readSerial` to populate the array.

```
Serial.print("Now, enter SMS content: ");
  char txtMsg[200];
  readSerial(txtMsg);
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=13)

Call `sms.beginSMS()` and pass it `remoteNumber` to start sending the message, `sms.print()` to send the message, and `sms.endSMS()` to complete the process. Print out some diagnostic information and close the `loop`. Your message is on its way!

```
Serial.println("SENDING");
  Serial.println();
  Serial.println("Message:");
  Serial.println(txtMsg);

  sms.beginSMS(remoteNumber);
  sms.print(txtMsg);
  sms.endSMS();
  Serial.println("\nCOMPLETE!\n");
}
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=14)

Once your code is uploaded, open the serial monitor. Make sure the serial monitor is set to only send a newline character on return. When prompted to enter the number you wish to call, enter the digits and press return. You'll then be asked to enter your message. Once you've typed that, press return again to send it.

The complete sketch is below.

```
/*
 SMS sender

 This sketch, for the Arduino GSM shield,sends an SMS message
 you enter in the serial monitor. Connect your Arduino with the
 GSM shield and SIM card, open the serial monitor, and wait for
 the "READY" message to appear in the monitor. Next, type a
```

```
 message to send and press "return". Make sure the serial
 monitor is set to send a newline when you press return.

 Circuit:
 * GSM shield
 * SIM card that can send SMS

 created 25 Feb 2012
 by Tom Igoe

 This example is in the public domain.

 http://www.arduino.cc/en/Tutorial/GSMExamplesSendSMS

 */

// Include the GSM library
#include <GSM.h>

#define PINNUMBER ""

// initialize the library instance
GSM gsmAccess;
GSM_SMS sms;

void setup() {
  // initialize serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  Serial.println("SMS Messages Sender");

  // connection state
  boolean notConnected = true;

  // Start GSM shield
  // If your SIM has PIN, pass it as a parameter of begin() in quotes
  while (notConnected) {
    if (gsmAccess.begin(PINNUMBER) == GSM_READY) {
      notConnected = false;
    } else {
      Serial.println("Not connected");
      delay(1000);
    }
  }

  Serial.println("GSM initialized");
}

void loop() {

  Serial.print("Enter a mobile number: ");
  char remoteNum[20];  // telephone number to send sms
  readSerial(remoteNum);
  Serial.println(remoteNum);

  // sms text
  Serial.print("Now, enter SMS content: ");
  char txtMsg[200];
  readSerial(txtMsg);
  Serial.println("SENDING");
  Serial.println();
  Serial.println("Message:");
  Serial.println(txtMsg);

  // send the message
  sms.beginSMS(remoteNum);
  sms.print(txtMsg);
  sms.endSMS();
  Serial.println("\nCOMPLETE!\n");
}

/*
  Read input serial
```

```
  */
int readSerial(char result[]) {
  int i = 0;
  while (1) {
    while (Serial.available() > 0) {
      char inChar = Serial.read();
      if (inChar == '\n') {
        result[i] = '\0';
        Serial.flush();
        return 0;
      }
      if (inChar != '\r') {
        result[i] = inChar;
        i++;
      }
    }
  }
}
```

[Get Code] (//www.arduino.cc/en/Tutorial/GSMExamplesSendSMS?action=sourceblock&num=1)

See Also:

- GSM (//www.arduino.cc/en/Reference/GSMConstructor) Constructor
- GSM.begin (//www.arduino.cc/en/Reference/GSMBegin)()
- GSM.shutdown (//www.arduino.cc/en/Reference/GSMShutdown)()
- GSM_SMS (//www.arduino.cc/en/Reference/GSMSMSConstructor) Constructor
- beginSMS (//www.arduino.cc/en/Reference/GSMSMSBeginSMS)()
- ready (//www.arduino.cc/en/Reference/GSMSMSReady)()
- endSMS (//www.arduino.cc/en/Reference/GSMSMSEndSMS)()
- available (//www.arduino.cc/en/Reference/GSMSMSAvailable)()
- remoteNumber (//www.arduino.cc/en/Reference/GSMSMSRemoteNumber)()
- read (//www.arduino.cc/en/Reference/GSMSMSRead)()
- write (//www.arduino.cc/en/Reference/GSMSMSWrite)()
- print (//www.arduino.cc/en/Reference/GSMSMSPrint)()
- peek (//www.arduino.cc/en/Reference/GSMSMSPeek)()
- flush (//www.arduino.cc/en/Reference/GSMSMSFlush)()

- Arduino GSM Shield (//www.arduino.cc/en/Main/ArduinoGSMShield) – Complete product description.
- Getting started with the GSM Shield (//www.arduino.cc/en/Guide/ArduinoGSMShield) – Get everything set up in minutes.
- GSM library (//www.arduino.cc/en/Reference/GSM) – Your reference for the GSM Library.

- GSMExamplesMakeVoiceCall (//www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall) - How to make a voice call with mic and speaker.
- GSMExamplesReceiveVoiceCall (//www.arduino.cc/en/Tutorial/GSMExamplesReceiveVoiceCall) - The call is received and connected, the number that is calling is shown on serial monitor and then the call is hung up.
- GSMExamplesReceiveSMS (//www.arduino.cc/en/Tutorial/GSMExamplesReceiveSMS) - How to receive an SMS message.
- GSMExamplesWebServer (//www.arduino.cc/en/Tutorial/GSMExamplesWebServer) - A web server that when gets a request from a client sends back the value read on Analog inputs.
- GSMExamplesWebClient (//www.arduino.cc/en/Tutorial/GSMExamplesWebClient) - Connect to the Arduino.cc home and print the contents on the serial monitor window.
- GSMToolsTestGPRS (//www.arduino.cc/en/Tutorial/GSMToolsTestGPRS) – Tries to access the internet over GPRS with supplied APN and credentials.

Last revision 2015/08/13 by SM

Share

NEWSLETTER

Enter your email to sign up →

(https://twitter.com/arduino) (https://www.facebook.com/official.arduino) (https://plus.google.com/+Arduino)
(https://www.flickr.com/photos/arduino_cc) (https://youtube.com/arduinoteam)