

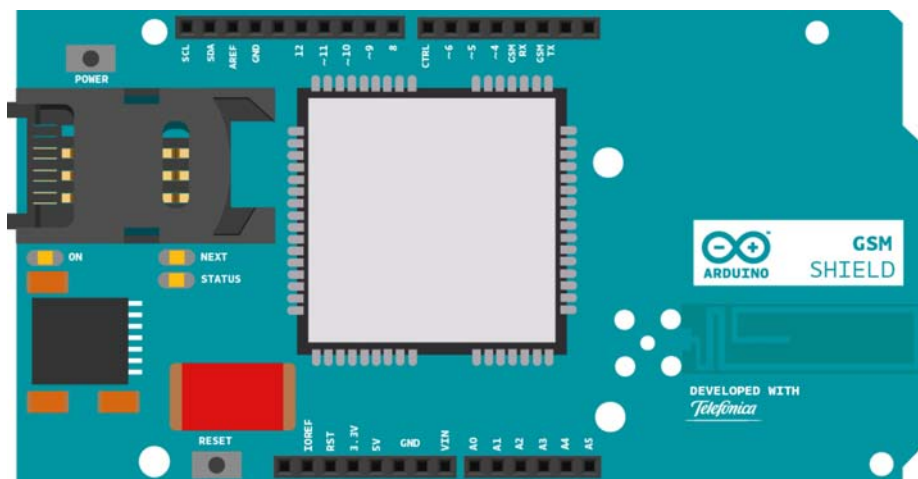
Make Voice Call

This sketch connects a voice call from your GSM shield and Arduino or Genuino to a remote phone number entered through the serial monitor. You'll need to attach a speaker and microphone ([//www.arduino.cc/en/Guide/ArduinoGSMShield#toc9](http://www.arduino.cc/en/Guide/ArduinoGSMShield#toc9)) to hear the connected phone and transmit your voice.

Hardware Required

- Arduino or Genuino Board
- Arduino + Telefonica GSM/GPRS Shield ([//www.arduino.cc/en/Main/ArduinoGSMShield](http://www.arduino.cc/en/Main/ArduinoGSMShield))
- Microphone and speaker attached to the GSM shield ([//www.arduino.cc/en/Guide/ArduinoGSMShield#toc9](http://www.arduino.cc/en/Guide/ArduinoGSMShield#toc9))
- SIM card

Circuit



([//www.arduino.cc/en/uploads/Tutorial/GSMShield_ArduinoUno.jpg](http://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 unlocks 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 `GSMVoiceCall` class.

```
GSM gsmAccess;  
GSMVoiceCall vcs;
```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=2>)

Create some variables to store the phone number you want to call :

```
String remoteNumber = "";  
char charbuffer[20];
```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=3>)

In `setup` , open a serial connection to the computer. You'll use this to send a phone number to the Arduino. After opening the connection, send a message to the Serial Monitor indicating the sketch has started.

```
void setup(){  
  
  Serial.begin(9600);  
  Serial.println("Make Voice Call");
```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=4>)

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] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=5>)

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] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=6>)

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

```
Serial.println("GSM initialized.");  
Serial.println("Enter phone number to call.");  
}
```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=7>)

The `loop` will accept incoming bytes from the serial monitor and connect your voice call.

First, check the serial buffer to see if there is any information waiting to be read. If there is, store it in a local variable :

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

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=8>)

If the buffer holds a newline character, check to see if the number entered is less than 20 digits long (theoretically, you'll never be able to dial a number with more digits than that).

```

if (inChar == '\n')
{
    if (remoteNumber.length() < 20)
    {

```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=9>)

Print out the number you're calling to the serial monitor.

```

Serial.print("Calling to : ");
    Serial.println(remoteNumber);
    Serial.println();

```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=10>)

The number to call will be stored in the String named `remoteNumber`. The `voiceCall()` function requires a `char` array. Copy the string to the array named `charbuffer`.

```

remoteNumber.toCharArray(charbuffer, 20);

```

To place the call, use `vcs.voiceCall()`, passing it the number you wish to reach. `voiceCall()` returns the status of the call; a `1` means it is connected. You can check the status of the connection with `getvoiceCallStatus()`.

To disconnect your call, send a newline character to trigger `hangCall()`.

```

if(vcs.voiceCall(charbuffer))
{
    Serial.println("Call Established. Enter line to end");
    while(Serial.read()!='\n' && (vcs.getvoiceCallStatus()==TALKING));
    vcs.hangCall();
}

```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=11>)

Once the call has been completed, clear the variable that stored the phone number :

```

Serial.println("Call Finished");
    remoteNumber="";
    Serial.println("Enter phone number to call.");
}

```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=12>)

If the number you entered in the serial monitor is longer than 20 digits, clear the `remoteNumber` String and start again :

```

else
{
    Serial.println("That's too long for a phone number. I'm forgetting it");
    remoteNumber = "";
}
}

```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=13>)

When reading information from the serial monitor, if the incoming character is not a newline or carriage return, add it to the `remoteNumber` String and close up the loop.

```

else
{
    // add the latest character to the message to send:
    if(inChar!='\r')
        remoteNumber += inChar;
}
}
}

```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=14>)

Once your code is uploaded, open the serial monitor. Once you see the message "Enter phone number to call", type a phone number and press "return". Make sure the serial monitor is set to only send a newline character on return.

The complete sketch is below.

```
/*
  Make Voice Call

  This sketch, for the Arduino GSM shield, puts a voice call to
  a remote phone number that you enter through the serial monitor.
  To make it work, open the serial monitor, and when you see the
  READY message, type a phone number. Make sure the serial monitor
  is set to send a just newline when you press return.

  Circuit:
  * GSM shield
  * Voice circuit.
  With no voice circuit the call will send nor receive any sound

  created Mar 2012
  by Javier Zorzano

  This example is in the public domain.
  */

// libraries
#include <GSM.h>

// PIN Number
#define PINNUMBER ""

// initialize the library instance
GSM gsmAccess; // include a 'true' parameter for debug enabled
GSMVoiceCall vcs;

String remoteNumber = ""; // the number you will call
char charbuffer[20];

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("Make Voice Call");

  // 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.");
  Serial.println("Enter phone number to call.");

}

void loop() {

  // add any incoming characters to the String:
  while (Serial.available() > 0) {
    char inChar = Serial.read();
    // if it's a newline, that means you should make the call:
    if (inChar == '\n') {
      // make sure the phone number is not too long:
      if (remoteNumber.length() < 20) {
        // let the user know you're calling:

```

```

Serial.print("Calling to : ");
Serial.println(remoteNumber);
Serial.println();

// Call the remote number
remoteNumber.toCharArray(charbuffer, 20);

// Check if the receiving end has picked up the call
if (vcs.voiceCall(charbuffer)) {
    Serial.println("Call Established. Enter line to end");
    // Wait for some input from the line
    while (Serial.read() != '\n' && (vcs.getVoiceCallStatus() == TALKING));
    // And hang up
    vcs.hangCall();
}
Serial.println("Call Finished");
remoteNumber = "";
Serial.println("Enter phone number to call.");
} else {
    Serial.println("That's too long for a phone number. I'm forgetting it");
    remoteNumber = "";
}
} else {
    // add the latest character to the message to send:
    if (inChar != '\r') {
        remoteNumber += inChar;
    }
}
}
}
}

```

[Get Code] (<http://www.arduino.cc/en/Tutorial/GSMExamplesMakeVoiceCall?action=sourceblock&num=1>)

See Also

- [GSM](http://www.arduino.cc/en/Reference/GSMConstructor) (<http://www.arduino.cc/en/Reference/GSMConstructor>) Constructor
 - [GSM.begin](http://www.arduino.cc/en/Reference/GSMBegin) (<http://www.arduino.cc/en/Reference/GSMBegin>())
 - [GSM.shutdown](http://www.arduino.cc/en/Reference/GSMShutdown) (<http://www.arduino.cc/en/Reference/GSMShutdown>())
 - [GSMVoiceCall](http://www.arduino.cc/en/Reference/GSMVCSConstructor) (<http://www.arduino.cc/en/Reference/GSMVCSConstructor>) Constructor
 - [getVoiceCallStatus](http://www.arduino.cc/en/Reference/GSMVCSGetVoiceCallStatus) (<http://www.arduino.cc/en/Reference/GSMVCSGetVoiceCallStatus>())
 - [ready](http://www.arduino.cc/en/Reference/GSMVCSReady) (<http://www.arduino.cc/en/Reference/GSMVCSReady>())
 - [voiceCall](http://www.arduino.cc/en/Reference/GSMVCSVoiceCall) (<http://www.arduino.cc/en/Reference/GSMVCSVoiceCall>())
 - [answerCall](http://www.arduino.cc/en/Reference/GSMVCSAnswerCall) (<http://www.arduino.cc/en/Reference/GSMVCSAnswerCall>())
 - [hangCall](http://www.arduino.cc/en/Reference/GSMVCSHangCall) (<http://www.arduino.cc/en/Reference/GSMVCSHangCall>())
 - [retrieveCallingNumber](http://www.arduino.cc/en/Reference/GSMVCSRetrieveCallingNumber) (<http://www.arduino.cc/en/Reference/GSMVCSRetrieveCallingNumber>())
-
- [Arduino GSM Shield](http://www.arduino.cc/en/Main/ArduinoGSMShield) (<http://www.arduino.cc/en/Main/ArduinoGSMShield>) – Complete product description.
 - [Getting started with the GSM Shield](http://www.arduino.cc/en/Guide/ArduinoGSMShield) (<http://www.arduino.cc/en/Guide/ArduinoGSMShield>) – Get everything set up in minutes.
 - [GSM library](http://www.arduino.cc/en/Reference/GSM) (<http://www.arduino.cc/en/Reference/GSM>) – Your reference for the GSM Library.
-
- [GSMExamplesReceiveVoiceCall](http://www.arduino.cc/en/Tutorial/GSMExamplesReceiveVoiceCall) (<http://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](http://www.arduino.cc/en/Tutorial/GSMExamplesReceiveSMS) (<http://www.arduino.cc/en/Tutorial/GSMExamplesReceiveSMS>) - How to receive an SMS message.
 - [GSMExamplesSendSMS](http://www.arduino.cc/en/Tutorial/GSMExamplesSendSMS) (<http://www.arduino.cc/en/Tutorial/GSMExamplesSendSMS>) - How to send an SMS entering number and text through the serial monitor.
 - [GSMExamplesWebServer](http://www.arduino.cc/en/Tutorial/GSMExamplesWebServer) (<http://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](http://www.arduino.cc/en/Tutorial/GSMExamplesWebClient) (<http://www.arduino.cc/en/Tutorial/GSMExamplesWebClient>) - Connect to the Arduino.cc home and print the contents on the serial monitor window.

- [GSMToolsTestGPRS \(//www.arduino.cc/en/Tutorial/GSMToolsTestGPRS\)](http://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



©2017 Arduino [Copyright Notice \(//www.arduino.cc/en/Main/CopyrightNotice\)](http://www.arduino.cc/en/Main/CopyrightNotice) [Contact us \(//www.arduino.cc/en/Main/ContactUs\)](http://www.arduino.cc/en/Main/ContactUs)
[About us \(//www.arduino.cc/en/Main/AboutUs\)](http://www.arduino.cc/en/Main/AboutUs) [Careers \(//www.arduino.cc/Careers\)](http://www.arduino.cc/Careers)

<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>