

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs  
CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

## Arduino GSM Shield

Shield-ul Arduino GSM permite unei placi Arduino sa se conecteze la rețeaua de Internet, sa trimita sau sa primeasca mesaje SMS si sa realizeze apeluri de voce. Shield-ul este compatibil cu placa Arduino UNO insa pentru placile Arduino Mega, Mega ADK si Leonardo sunt necesare cateva modificari minore. Momentan, placa Due nu este compatibila cu shield-ul.

In acest tutorial vei descoperi cum se realizeaza un apel de voce si cum se transmite un SMS, utilizand shield-ul Arduino GSM.



Pentru acest tutorial, vei avea nevoie de urmatoarele componente:

- O placa Arduino Mega - [http://www.robofun.ro/arduino\\_mega2560](http://www.robofun.ro/arduino_mega2560).
- Un shield Arduino GSM - [http://www.robofun.ro/arduino\\_gsm\\_gprs\\_shield](http://www.robofun.ro/arduino_gsm_gprs_shield).
- Cablu USB.
- Fire de conexiune tata-tata  
[http://www.robofun.ro/fire\\_conexiune\\_tata\\_tata-140mm](http://www.robofun.ro/fire_conexiune_tata_tata-140mm).
- Un alimentator extern pentru Arduino (9V @ 1A)  
<http://www.robofun.ro/alimentator-extern-arduino>.

- O pereche de casti cu microfon.

## Modificarea necesara pentru placa Arduino Mega.

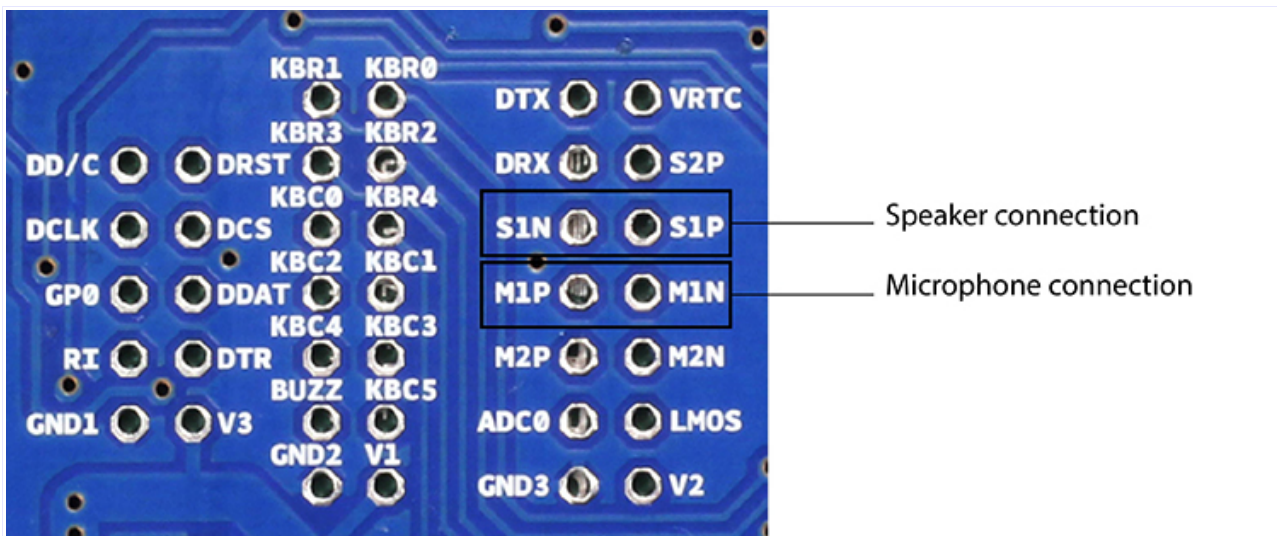
Pentru ca placa Arduino Mega sa comunice corect cu shield-ul GSM, sunt necesare 2 modificari si anume:

- pinul 2 al shield-ului se indoaie deoarece nu trebuie sa fie conectat cu placa Arduino.
- pinul 2 al shield-ului se conecteaza printr-un fir cu capete tata-tata la pinul 10.

Pentru celelalte placi, respectiv pentru placa Arduino UNO, nu este necesara nici o modificare. Nu trebuie decat sa conectezi shield-ul si atat. Pentru placa Leonardo, urmeaza aceeasi pasi ca mai sus cu diferenta ca pinul 2 se conecteaza la pinul 8.

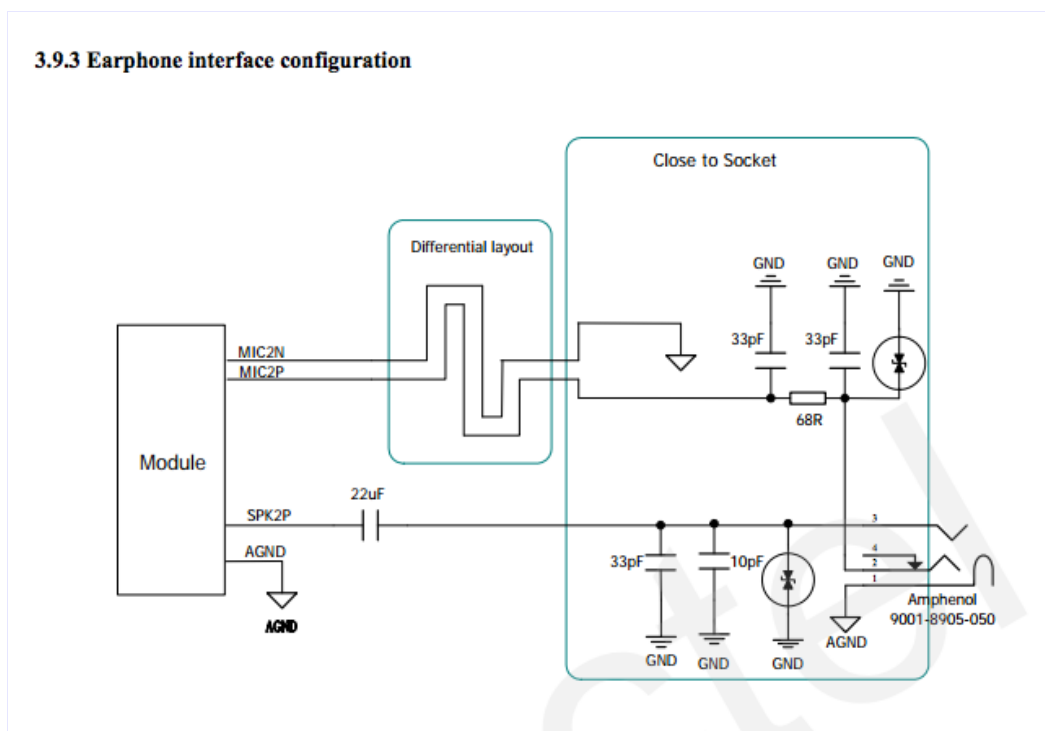
## Apel de voce.

Pentru a realiza apelul de voce, iti vor fi necesare castile si microfonul. Pentru conectarea celor doua, se vor urmari diagramele de mai jos:



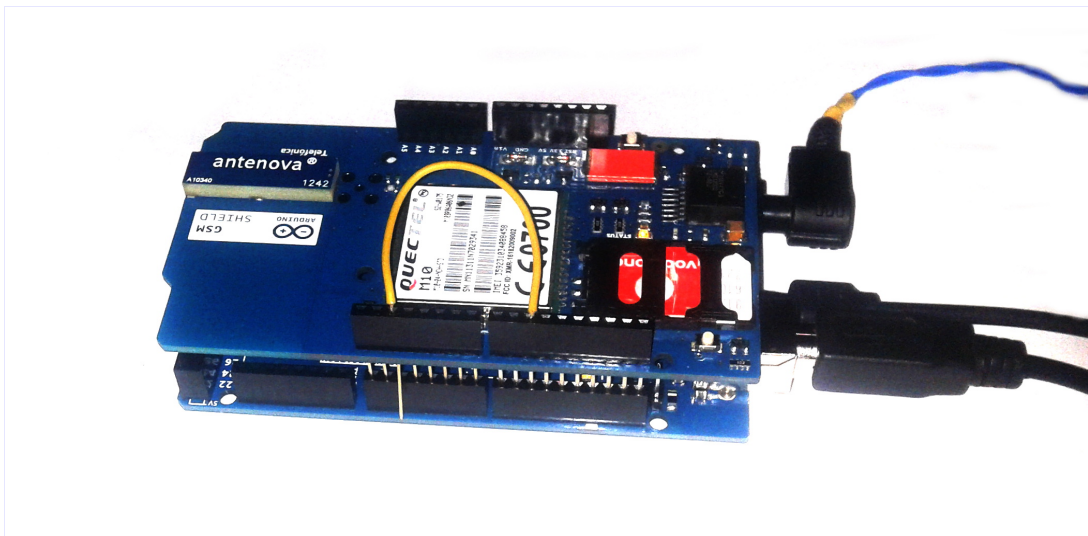
Imaginea de mai sus reprezinta partea interioara a shield-ului. Pe aceasta parte vei gasi pad-urile M1P / M1N fiind intrarea pozitiva si negativa pentru microfon si pad-urile S1P / S1N fiind iesirea pozitiva si negativa pentru speaker (difuzor).

Intre casti/microfon si pad-urile shield-ului este necesar un circuit intermediar, dat de diagrama de mai jos.



In final, conecteaza alimentatorul de 9V in mufa placii Arduino. Atentie ! Nu se recomanda alimentarea shield-ului direct din USB, fara alimentator. Modemul de pe placa shield-ului consuma mult mai mult decat poate sa genereze portul USB. Acest lucru poate duce la o functionare incorecta sau , in cel mai rau caz, la o defectiune.

Dupa realizarea tuturor conexiunilor, tot ce iti ramane este sa infigi shield-ul in placa Arduino si sa fixezi cartela SIM in slot. Urmeaza imaginile de mai jos.



In imaginea de mai sus se observa si modificarea de care am vorbit mai devreme.

## Sketch-ul pentru apelul de voce

Incarca sketch-ul de mai jos in placa Arduino. Deschide Serial Monitor si asteapta pana cand ti se va cere numarul de telefon. Tasteaza numarul si asteapta pana cand se realizeaza apelul. Pentru a incheia apelul, nu trebuie decat sa tastezi un caracter de la tastatura si sa il trimiti prin Serial Monitor.

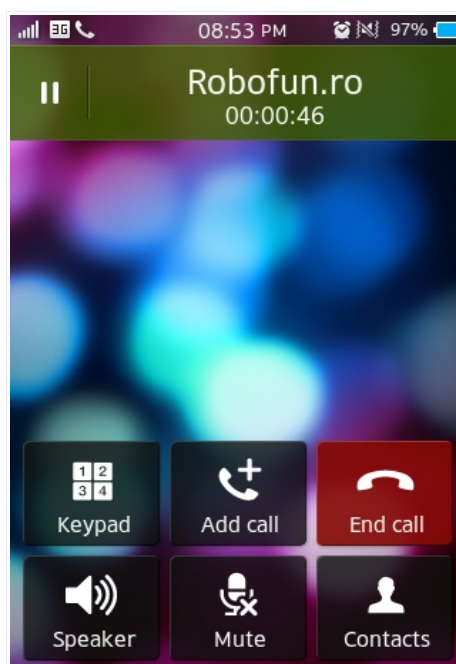
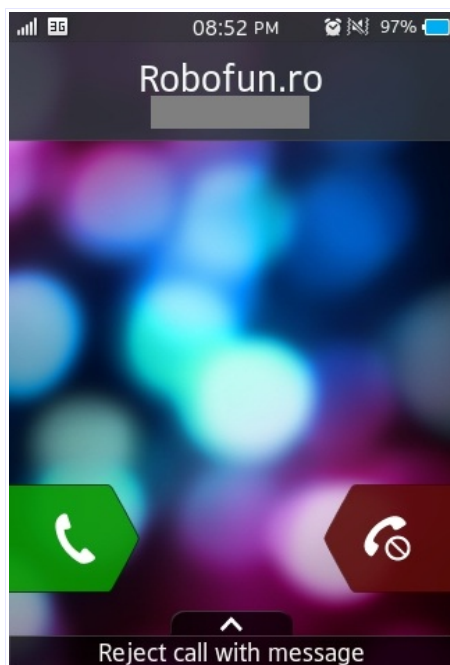
```
COM17

Realizeaza apel de voce
Modul GSM initializat
Tasteaza nr. de telefon:
Apelez : 0720[redacted]

Apel realizat. Tasteaza o noua linie pentru a inchide.
Apel incheiat.
Tasteaza nr. de telefon:
Apelez : 0720[redacted]

Apel incheiat.
Tasteaza nr. de telefon:
Apelez : 0720[redacted]

Apel realizat. Tasteaza o noua linie pentru a inchide.
```



```

#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
    Serial.begin(9600);

    Serial.println("Realizeaza apel de voce");

    // 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("Lipsa conexiune");
            delay(1000);
        }
    }

    Serial.println("Modul GSM initializat");
    Serial.println("Tasteaza nr. de telefon: ");

}

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)
        {
            // show the number you're calling:
            Serial.print("Apelez : ");
            Serial.println(remoteNumber);
            Serial.println();

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

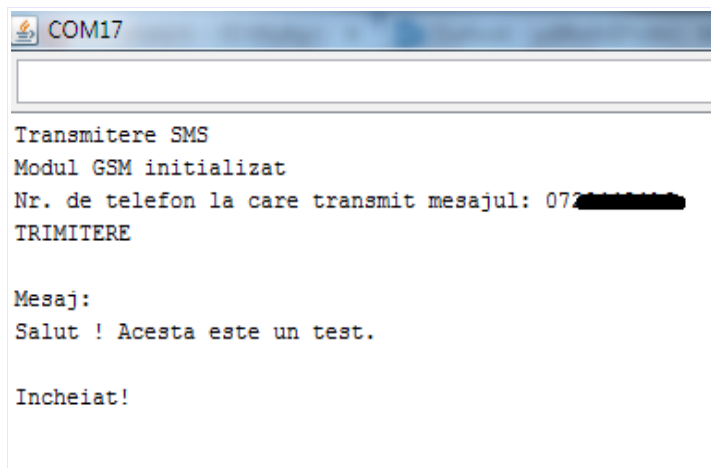
            // Check if the receiving end has picked up the call
            if(vcs.voiceCall(charbuffer))
            {
                Serial.println("Apel realizat. Tasteaza o noua linie
pentru a inchide.");
                // Wait for some input from the line
                while(Serial.read()!='\n' &&
(vcs.getvoiceCallStatus()==TALKING));
                // And hang up
                vcs.hangCall();
            }
            Serial.println("Apel incheiat.");
            remoteNumber="";
            Serial.println("Tasteaza nr. de telefon: ");
        }
        else
        {
            Serial.println("Reintrodu nr. de telefon.");
            remoteNumber = "";
        }
    }
    else
    {
        // add the latest character to the message to send:
        if(inChar!='\r')
            remoteNumber += inChar;
    }
}
}

```



## Cum transmit un SMS ?

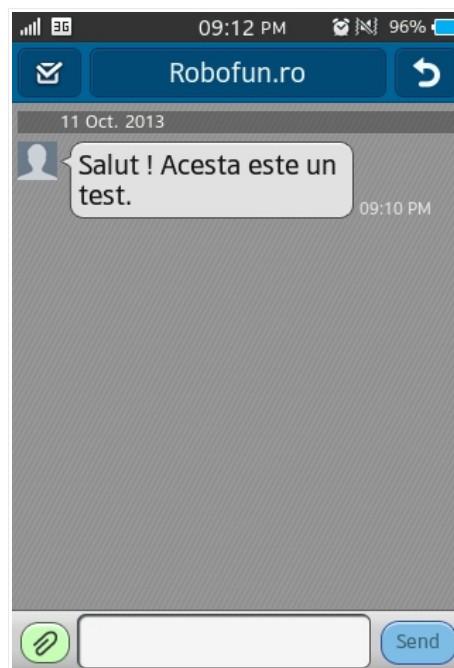
Shield-ul GSM este capabil de a trimite si SMSuri. Sketch-ul de mai jos realizeaza o conexiune cu retea GSM si transmite un mesaj catre un numar pe care il specificati in program.



```
COM17
Transmitere SMS
Modul GSM initializat
Nr. de telefon la care transmit mesajul: 072[redacted]
TRIMITERE

Mesaj:
Salut ! Acesta este un test.

Incheiat!
```



```
#include <GSM.h>

#define PINNUMBER ""

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

// char array of the telephone number to send SMS
// change the number 1-212-555-1212 to a number
// you have access to
char remoteNumber[20]= "9999999999";

// char array of the message
char txtMsg[200]="Salut ! Acesta este un test.";

void setup()
{
    // initialize serial communications
    Serial.begin(9600);
```

```

Serial.println("Transmitere SMS");

// 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("Lipsa conexiune");
        delay(1000);
    }
}
Serial.println("Modul GSM initializat");
sendSMS();
}

void loop()
{
    // nothing to see here
}

void sendSMS(){

    Serial.print("Nr. de telefon la care transmit mesajul: ");
    Serial.println(remoteNumber);

    // sms text
    Serial.println("TRIMITERE");
    Serial.println();
    Serial.println("Mesaj:");
    Serial.println(txtMsg);

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

```