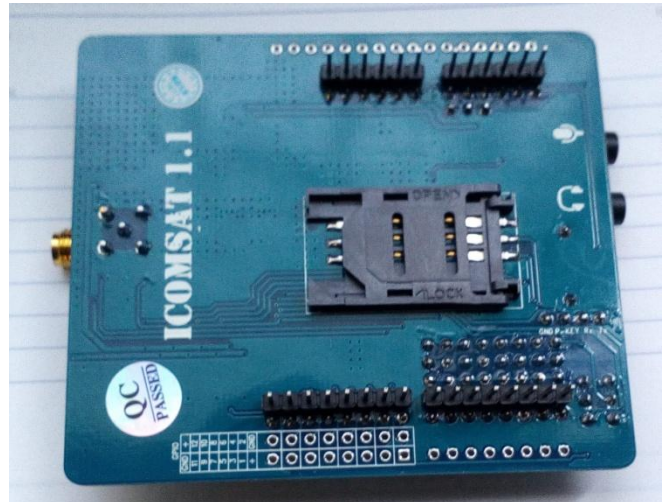


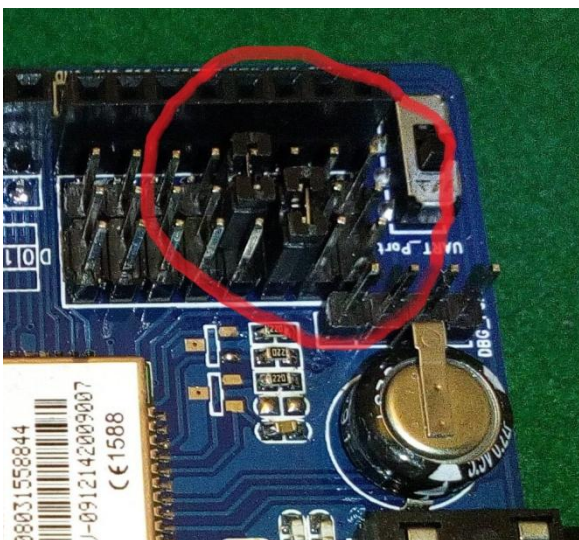
48- sim900 – Arduino & sms (some notes at end of this section)



L'idea che Arduino possa essere pilotato anche da grande distanza, tramite dei messaggi SMS, apre un certo numero di scenari di utilizzo. Non solo attivare e disattivare uno o più rele', ma anche ricevere informazioni locali, provenienti da eventuali sensori ambientali o da un modulo GPS o, sempre ad esempio, orientare una webcam che inquadra una strada, l'interno di un magazzino o altro.

Il modulo sim900 (il chip bianco e rosso al centro dell'immagine) è un modulo in grado di replicare le funzioni di un telefono. La sim900 può essere collegata ad Arduino, al quale trasmette le informazioni sugli sms in arrivo e dal quale riceve indicazioni sul destinatario e sul testo di eventuali sms da far partire. Può anche fungere da modem e quindi collegare Arduino ad internet, ma questa strada non è stata per ora esplorata.

Esistono diversi tipi di shield sulle quali è montata la sim900 e più o meno tutte dichiarano di essere compatibili con Arduino. Il modulo in figura non è diverso dagli altri, ma la shield sulla quale è montato si adatta perfettamente alle porte di Arduino, per cui il suo utilizzo elimina ogni problematica di collegamento. La shield ha, sul dorso, l'alloggiamento per la sim, indispensabile per compiere qualunque attività di tipo telefonico, compresa la ricezione e l'invio di chiamate vocali. È anche dotata di prese alle quali collegare microfono cuffia per cui, collegando ad Arduino una tastiera ed un display lcd, è possibile costruire un telefono cellulare, in grado di inviare e ricevere chiamate o sms e gestire una rubrica. La shield in figura è anche equipaggiata con un modulo RTC (Real Time Clock) in grado, all'occorrenza, di informazioni su data e ora.



La libreria che gestisce questa shield (e probabilmente anche le altre) utilizza i pin 2 e 3 per comunicare con Arduino per cui, prima di collegare arduino bisogna specializzare le porte 2 e 3 della shield alla trasmissione / ricezione dei dati. Per fare questo è necessario posizionare due ponticelli (in dotazione alla shield) come in figura.

Sempre prima di collegare la shield bisogna anche assicurarsi che il micro interruttore, a destra della fila dei ponticelli, sia posizionato su "UART".

Ultima ma non ultima indicazione riguarda tre microswitch che la equipaggiano (i tre rettangoli bianchi e grigi, che si scorgono in alto a sinistra (vedi foto in alto, sotto il titolo)).

Il primo accende o spegne la shield, il secondo la resetta ed il terzo esegue il reset di arduino. Durante le prove e' stata notata una certa criticita' nell'utilizzo di questi switch per cui, qualora si volesse replicare l'esercizio, prima di scoraggiarsi per l'apparente mancato funzionamento del sistema, bisognera' provare ad agire su detti switch (in particolare su quello per il reset di arduino).

La sequenza di attivazione della shield e' la seguente:

- collegare la shield (dotata di sim) ad Arduino. La sim deve essere stata preventivamente ripulita da eventuali sms letti ma non ancora cancellati. Eventuali sms presenti nella coda degli sms letti verranno interpretati dalla sim come sms in arrivo, e quindi letti ed interpretati;
- collegare Arduino ad una fonte di energia (pc o alimentatore o pacco batterie);
- accendere la shield (primo bottone a sinistra). Si accendera' un led verde (a fianco dei microswitch) mentre un altro iniziera' a lampeggiare velocemente, segno che la sim900 sta cercando di collegarsi alla rete telefonica. Una volta collegata alla rete il lampeggio si stabilizzera' e diventera' meno frequente (un lampeggio ogni 2 o 3 secondi).

Per quanto concerne le porte utilizzate dalla shield, non sono state reperite informazioni circa l'utilizzo di ulteriori porte oltre alla 2 e 3, utilizzate queste ultime in sostituzione delle canoniche 0 e 1, utilizzate dalla connessione usb per il caricamento del programma per l'eventuale utilizzo del serial monitor.

In questo esercizio ci limiteremo ad inviare degli sms per accendere e spegnere un led e per chiederne lo stato (acceso o spento). In quest'ultimo caso Arduino rispondera' con un sms nel quale esplicita la situazione. Nulla osta, ovviamente, che al posto del led sia collegato un rele', che a sua volta attiva o disattiva altri impianti. Gli sms da inviare ad Arduino possono contenere i seguenti testi:

- On accende il led
- Off spegne il led
- Stato chiede l'invio di un sms con la situazione del led (acceso o spento)

Il filmato del progetto e' reperibile qui: https://youtu.be/_oX603nhIWM

Prima di procedere alla compilazione del programma devono essere installate, se non gia' presenti, le seguenti librerie:

- Gsm libreria: 3.07.1 BETA_GSM_GPRS_GPS_IDE100_v307_1.zip, reperibile [qui](#)
- LiquidCrystal_I2C.h reperibile [qui](#)

Per installare le librerie e' necessario seguire la procedura illustrata nei precedenti progetti, e sintetizzabile in:

- download della libreria in formato compresso
- installazione, andando in IDE-> sketch-> includes Library-> add .zip library
- verifica l'avvenuta installazione (andando in IDE-> sketch-> includes Library-> Contributed library)

Nota: Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube.

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a giocarduino@libero.it

Here some notes about this project, translated by google translator



The idea that Arduino can be driven from a great distance, by an SMS messages, opens some new scenarios. Not only we can activate or deactivate one or more relay, but also receive local information, from any environmental sensor or by a GPS module or, for example, to orient a webcam on a road or inside a storage or elsewhere

The sim900 device (the white and red chips on picture) is able to replicate a phone functions. The sim900 can be connected to Arduino for information exchange about incoming and outgoing messages (the caller, the recipient and the text). It can also act as a modem, and then connect Arduino to web, but this road is not been explored here.

There are different shield types on which the sim900 is used and more or less all claim to be compatible with Arduino. The one in picture is one of them, but its shield perfectly fits the Arduino gates, so its use eliminates any problems connecting. The shield has, on its back, the housing for a sim, essential to fulfill any phone activity, including receiving and sending voice calls. It is also equipped with sockets for microphone and earphone and if you connect a keyboard and a LCD display, you can build a cell phone, to send and receive calls or text messages and manage an address book. The shield in figure is also equipped with an RTC (Real Time Clock) to get, if necessary, information about date and time.

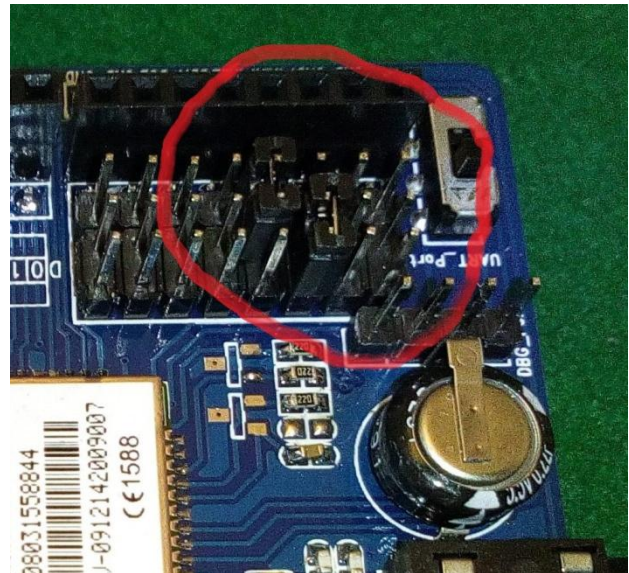
The library that handles this shield (and probably other) uses pins 2 and 3 to communicate with Arduino so, before connecting to Arduino, you must specialize the shield pins 2 and 3 to transmission / reception of data. To do this we need to place two jumpers (supplied with the shield) as shown below.

Before shield connection you must also ensure that the micro switch, on the right of jumpers row, is set to "UART".

The last indication concerns three microswitches (the three white and gray rectangles, which you can see on the top left (see picture at the top, under the title)).

The first turns on or off the shield, the second resets it and the third is the Arduino resets.

in testing we have seen some criticality in use of these switches so, if you try to replicate the example, before you get discouraged by the apparent failure, act on said switches (in particular on the arduino reset one).



The shield activation sequence:

- connect shield (with a sim) to Arduino. The sim must have been previously cleared of any sms read but not yet deleted. Any sms in queue will be treated as incoming sms, and then read and interpreted;
- connect Arduino to a power source;
- turn shield on (top left button). Will light a green led (alongside the microswitch) while another will start to flash rapidly, indicating that sim900 is trying to connect to the telephone network. Once connected and stabilized, flashing becomes less frequent (one flash every 2 or 3 seconds).

Arduino: sim900 & sms

With regard to pins used by shield, we have not find any information about use of other pins, in addition to the 2 and 3 (used in place of 0 and 1, reserved to program loading and for serial monitor communications).

In this project, we send sms to turn on and off a LED and ask for its status (on or off). In the latter case, Arduino will answer with a text message. Instead a led you can, of course, connect a relays, which in turn activates or deactivates other apparatus.

Text messages to be sent to Arduino can contain the following text:

- On turns led on
- Off turns led off
- State requests sending an sms, with the led situation

The project movie: https://youtu.be/_oX603nhIWM

Before proceeding to program compilation must be installed, if not already done, the libraries:

- Gsm library: **3.07.1 BETA_GSM_GPRS_GPS_IDE100_v307_1.zip**, found [here](#)
- LiquidCrystal_I2C.h found [here](#)

For library installation, see process shown in previous projects, and summarized in:

- library download in compressed form;
- Installation via IDE-> sketch-> includes Library-> add .zip library
- After installation please verify the library. It must be present in IDE-> sketch-> includes Library-> Contributed library

Note: This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

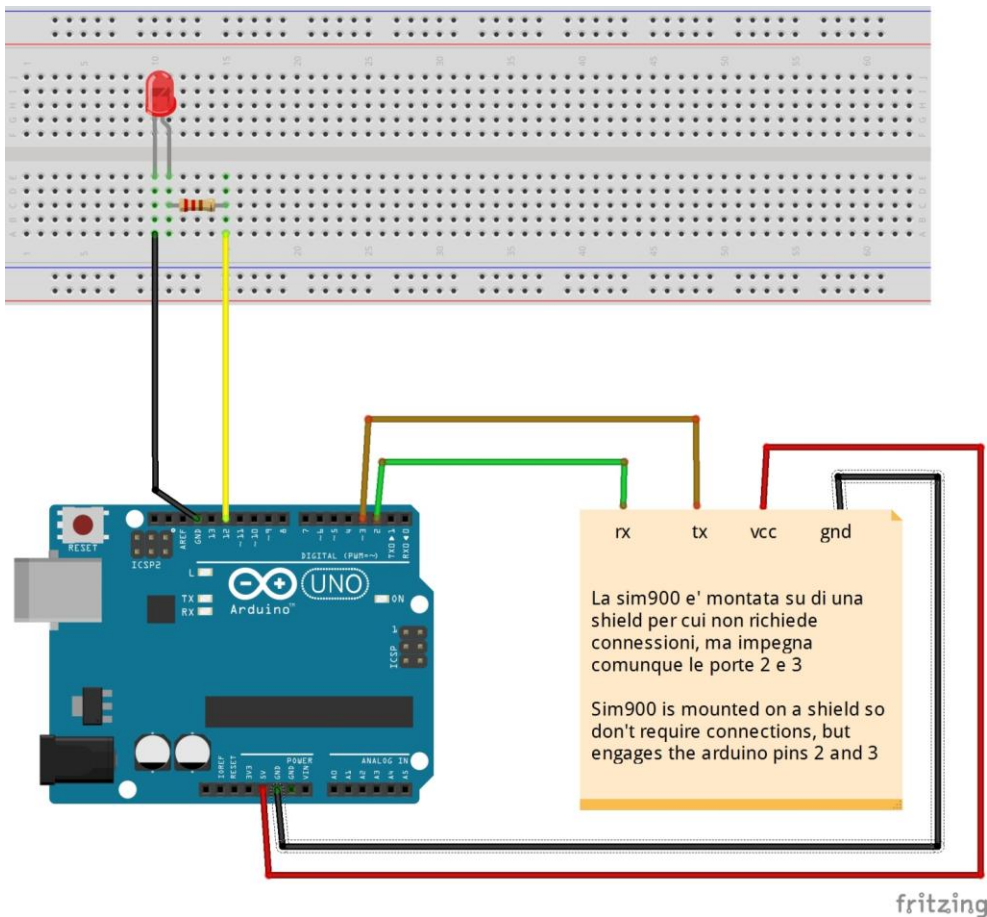
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

For any questions or suggestions about this note (and on its english translation), please write to giocarduino@libero.it (simple words and short sentences, please)

Materiali

- Un modulo Sim900, montato scheda icomsat 1.1
- Un Display lcd 1602 con driver I2C
- Una sim attiva
- Un led
- Una resistenza da 220 ohm
- Un po' di cavetteria

Schema



Programma

```
/* Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo. Per
 * rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il
 * trasferimento nell'IDE, premendo CTRL+T.
 *
 * Le basi di questo programma sono state reperite in rete
 *
 * Questo programma utilizza uno shield sim900 per ricevere degli sms che, una volta interpretati
 * da Arduino, provocano l'esecuzione di alcune azioni.
 *
 * In pratica accende e spegne un led o manda un messaggio, a seconda del testo contenuto
 * nell'SMS ricevuto. Se il testo dell'sms e':
 * On => Accende il led
 * Off => Spegne il led
 * Stato => Stato del led
 *
 * il programma utilizza anche un display lcd per visualizzare il nome del mittente, il testo del
 * messaggio ricevuto ed eventuali altre informazioni. Il display deve essere del tipo 1602 con
 * driver I2C e deve essere collegato, oltre che all'alimentazione ed al gnd, anche al bus
 * SCL/SDA (pin A4 ed A5)
 *
 * la shield sim900 sms deve essere predisposta all'uso dei pin 2 e 3 per la gestione
 * della comunicazione tra arduino e la shield. Si tratta di spostare due ponticelli
 * seguendo quanto indicato nella scheda tecnica di questo progetto
 *
 * Il led deve essere collegato alla porta 12
 *
 * -----
 * Warning: cut&paste from PDF to IDE loses formatting. to restore it press CTRL + T.
 *
 * The basics of this program have been found on the net
```

Arduino: sim900 & sms

```
*
* This program is using a shield sim900 to receive text messages which, once interpreted by
* Arduino, causes certain actions execution.
*
* In practice it turns on and off a LED or sends a message, depending on received text content. If
* the sms text is:
* On => LED on
* Off => LED off
* State => send an sms with the LED state
*
* Program also uses an LCD display to show the sender name, the message received and any additional
* information. The display must be a 1602 type with I2C driver and must be connected to power, to
* ground, and to the SCL / SDA bus (A4 and A5 pins)
*
* Shield sim900 uses pins 2 and 3 as communication channel with Arduino. For do that you must set
* two jumpers as specified in project note
*
* The LED must be connected to pin 12
*-----
*/
#include "SIM900.h"           // libreria per la gestione degli sms
#include <SoftwareSerial.h>   // necessaria alla libreria gsm.h richiamata in sms.h
#include "sms.h"             // Libreria per la gestione degli SMS
#include <Wire.h>            // libreria wire presente, di fault, nell'IDE
#include <LiquidCrystal_I2C.h> // libreria di gestione del display lcd
// . . . . . addr,en,rw,rs,d4,d5,d6,d7,bl,b1pol
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // definisce la tipologia del
// display
MSGSMS sms;
int rele = 12;              // Il led e' collegato al pin 12
int numdata;
boolean started = false;
char smsbuffer[160];
char Mittente[20];
char mittenteregistrato [13] = "+393350000000"; // numero del mittente registrato (conosciuto)
char nomemittente [16] = "Giocarduino ";
char sediciirt [16];       // zona di memorizzazione dei primi sedici caratteri del messaggio ricevuto
int i = 0;                // indice utilizzato nei cicli di for
int semaforo = 0;         // semaforo riconoscimento numero telefono: 0 = numero riconosciuto;
// 1 = numero sconosciuto
//
void setup()
{
  Serial.begin(9600);
  lcd.begin(16, 2); // inizializza il display (16 caratteri per due righe) e accende
// l'illuminazione dello sfondo
  for (int i = 0; i < 2; i++) // fa lampeggiare per due volte lo sfondo

  {
    lcd.backlight();
    delay(250);
    lcd.noBacklight();
    delay(250);
  }
  lcd.backlight(); // termina mantenendo illuminato lo sfondo
  lcd.print("buongiorno ");

  pinMode(rele, OUTPUT); // imposta il pin 12 come pin di OUTPUT.
  digitalWrite(rele, LOW); // spegne inizialmente il led
  lcd.setCursor (0, 1);
  // Inizializzo la connessione GSM impostando il baudrate
  if ( gsm.begin(2400) )
  {
    lcd.print ("sim900 pronta");
    started = true;
  }
  else
    lcd.println("sim900 ko");
}
//
void loop()
{
  char inSerial[50];
  char position;
  if (started)
  {
    position = sms.IsSMSPresent(SMS_ALL); // legge i messaggi in coda
    if (position)
```

Arduino: sim900 & sms

```
{
  lcd.backlight(); // illumina il display lcd
  lcd.clear ();
  sms.GetSMS(position, Mittente, smsbuffer, 160); // Leggo il messaggio ed il mittente
  lcd.print ("sms in arrivo da");
  lcd.setCursor (0, 1);
  semaforo = 0;
  for (i = 0; i <= 12; i++)
  {
    if (!(Mittente[i] == mittenteregistrato [i]))
      semaforo = 1;
  }
  if (semaforo == 0)
    lcd.print (nomemittente);
  else
    lcd.print (Mittente);
  delay (3000); // attende tre secondi per consentire la lettura su lcd del mittente
  lcd.clear ();
  lcd.print ("testo sms:");
  lcd.setCursor (0, 1);
  for (i = 0; i <= 15; i++)
    sedicicrt [i] = smsbuffer [i]; // memorizzo i primi 16 caratteri del messaggio ricevuto
  lcd.print (sedicicrt); // espongo su lcd il messaggio ricevuto
  if (strcmp(smsbuffer, "On") == 0)
  {
    digitalWrite(rele, HIGH); // Attiva il RELE
  }
  else if (strcmp(smsbuffer, "Off") == 0)
  {
    digitalWrite(rele, LOW); // disattiva il RELE
  }
  else if (strcmp(smsbuffer, "Stato") == 0)
  {
    if (digitalRead(rele) == HIGH)
    {
      sms.SendSMS(Mittente, "STATUS: RELE Attivo"); // restituisce true se invia l'SMS
    }
    else
    {
      sms.SendSMS(Mittente, "STATUS: RELE disattivo"); // restituisce true se invia l'SMS
    }
  }
  else
  {
    lcd.setCursor (0, 0);
    lcd.print ("sms non valido");
  }
  delay (3000); // attende tre secondi per consentire la lettura del messaggio
  sms.DeleteSMS(position); // Elimina l'SMS appena analizzato
}
delay (5000); // attende 5 secondi per consentire la lettura del display
lcd.clear ();
lcd.print ("sim900 pronta");
lcd.setCursor (0, 1);
lcd.print ("in attesa sms");
delay (5000);
lcd.noBacklight();
}
```