

## 49- GPS: modulo VK2828U7G5LF (some notes at section end)



Un ricevitore GPS viene essenzialmente utilizzato per rilevare la posizione geografica ed e' quindi utile registrare un percorso o, se associato ad un sistema di comunicazione (ad esempio un telefono) per individuare la posizione di un apparato mobile, come ad esempio un veicolo, una barca o un animale.

Si tratta di una funzione nota a tutti e da tempo presente in ogni smartphone, ove viene utilizzata per istruire il navigatore o fornire indicazioni circa eventuali magazzini commerciali o servizi pubblici disponibili in zona

Il modulo in figura e' un ricevitore GPS che, tramite la libreria TinyGPS.h, interagisce facilmente con Arduino e fornisce, in tempi piuttosto brevi, le coordinate geografiche del punto in cui si trova.

Le specifiche tecniche di questo modulo sono reperibili [qui](#)

In questo progetto (derivato da un analogo progetto trovato sul web) si evidenzia su di un display lcd le coordinate, in gradi decimali, del punto in cui il modulo e' posizionato. [Qui](#) il film del progetto

Poiche' il modulo comunica con Arduino tramite un collegamento seriale posizionato sulle porte 0 ed 1 (e quindi sulle porte sulle quali transitano anche i segnali provenienti dal pc tramite la porta USB), e' opportuno che, nel momento in cui si carica il programma su Arduino, il modulo sia scollegato dall'alimentazione (e' cioe' opportuno che il modulo non sia operativo). Una volta caricato il programma ed una volta tolta la connessione USB, si puo' collegare il cavo rosso del modulo alla porta di alimentazione (5v) di Arduino, e procedere alla rilevazione della posizione.

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

- LiquidCrystal\_I2C.h reperibile [qui](#)
- TinyGPS.h reperibile [qui](#)

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

- download della libreria in formato complesso
- installare le nuove librerie andando in IDE-> sketch-> includes Library-> add .zip library (attenzione: per installare la TinyGPS bisogna prima disimpaccare il file ottenuto dal download, disimpaccare poi il file TinyGPS.rar, aprire la directory TinyGPS e copiare la directory ivi contenuta (che si chiama ancora TinyGPS) in Documenti/Arduino/libraries)
- verificare 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](mailto:giocarduino@libero.it)

## Here some notes about this project, translated by google translator



A GPS receiver is used to detect the geographical location and it is therefore useful to record a path or, if associated to any communication system (such as a phone), to locate the position of a mobile apparatus, such as a vehicle, a boat or an animal.

It is a feature known to anyone, present in any smartphone, where it is used to instruct the navigation system or to provide information about commercial warehouses or public services available in area

The device in figure is a GPS receiver that, through the TinyGPS.h library, easily interacts with Arduino and provides, in a fairly short time, the geographical coordinates.

[Here](#) you can find the data sheet.

In this project (derived from a similar project found on net) we highlight, on an LCD display, the coordinates in decimal degrees where the receiver is positioned. [Here](#) the project movie.

Since the gps device communicates with Arduino via a serial link placed on pins 0 and 1 (and then on pins on which pass signals from USB cable), is appropriate that, when you are loading program on Arduino, the device must be disconnected (the gps device must not be powered). After program loading, and after removing USB connection, you can connect the gps device red cable to the Arduino power pin (5V), and proceed to the position detection.

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

- LiquidCrystal\_I2C.h found [here](#)
- TinyGPS.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. Attention: to install the TinyGPS.h library, must first unzip the downloaded file, then unzip the TinyGPS.rar file, then open the directory and copy the contained directory (which is still called TinyGPS) in: Documents / Arduino / libraries
- 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.

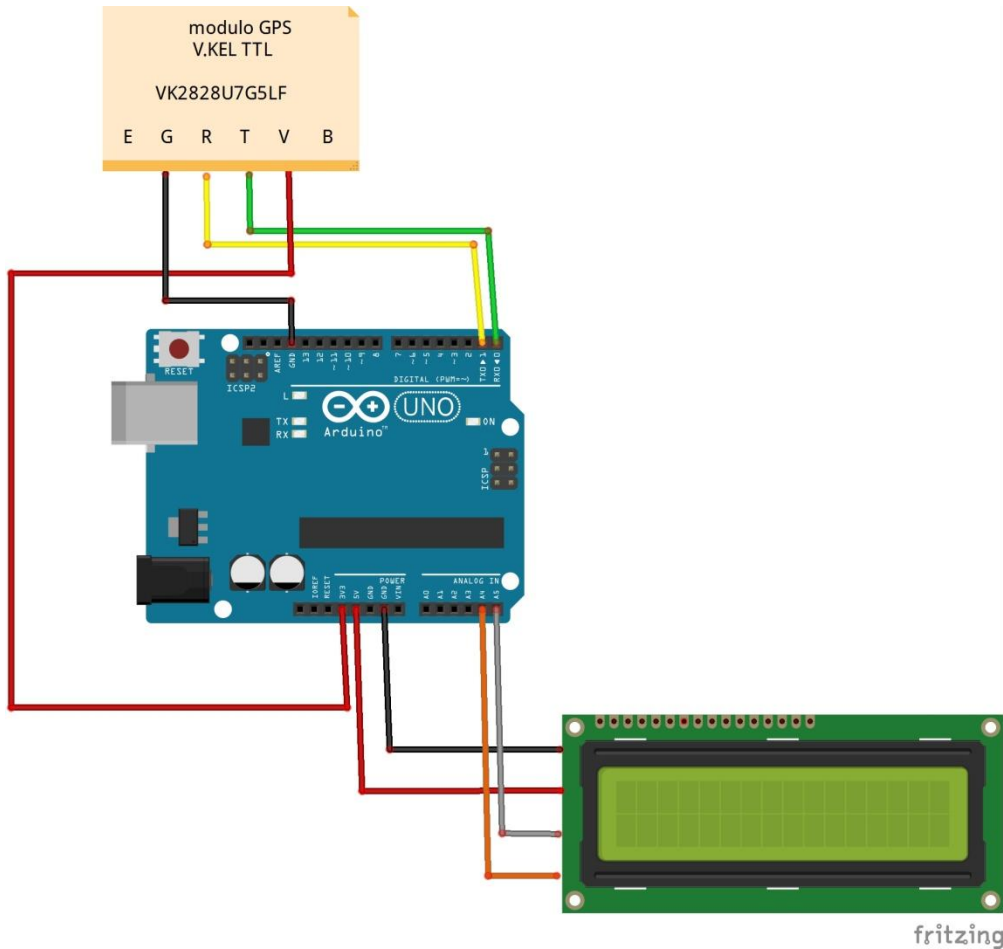
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

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

## Materiali

- Un modulo GPS VK2828U7G5LF
- Un display 1602 con driver I2C
- Un po' di caverteria

## 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.
 *
 * Questo programma (derivato da un analogo programma presente sul web) sperimenta l'utilizzo di un
 * modulo VK2828U7G5LF per la ricezione dei segnali GPS il programma riceve i segnali GPS, li
 * decodifica attraverso le funzioni rese disponibili dalla libreria TinyGPS e li espone
 * (latitudine e longitudine in gradi decimali) su di un display LCD.
 * Maggiori informazioni, comprensive di schema dei collegamenti e note descrittive del progetto,
 * sono reperibili in http://giocarduino.altervista.org/e49-gps.pdf
 *
 *-----
 * Warning: cut&paste from PDF to IDE loses formatting. To restore it press CTRL + T.
 * This program (found on web) uses a VK2828U7G5LF module to receive GPS signals. Program receives
 * GPS signals, decodes them through some functions offered by the TinyGPS library and exposés
 * latitude and longitude in decimal degrees, on an LCD display. More information are available in:
 * http://giocarduino.altervista.org/e49-gps.pdf
 *-----
 */
#include <TinyGPS.h>
#include <Wire.h> // libreria wire presente, di default, nell'IDE
#include <LiquidCrystal_I2C.h> // libreria di gestione del display lcd
//. . . . . addr, en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // definisce la tipologia del display
```

## Arduino: GPS - VK2828U7G5LF device

```
TinyGPS gps;
long lat;
long lon;
unsigned long fix_age;
int DEG;
int MIN1;
int MIN2;
int c;
//
//***** routine di esposizione della latitudine (in gradi decimali) *****
//
void LAT()
{
  DEG = lat / 1000000;
  MIN1 = (lat / 10000) % 100;
  MIN2 = lat % 10000;
  lcd.setCursor(0,0);
  lcd.print("LAT:");
  lcd.print(DEG);
  lcd.write(0xDF);
  lcd.print(MIN1);
  lcd.print(".");
  lcd.print(MIN2);
  lcd.print("'  ");
}
//
// ***** routine di esposizione della longitudine (in gradi decimali) *****
//
void LON()
{
  DEG = lon / 1000000;
  MIN1 = (lon / 10000) % 100;
  MIN2 = lon % 10000;
  lcd.setCursor(0, 1);
  lcd.print("LON:");
  lcd.print(DEG);
  lcd.write(0xDF);
  lcd.print(MIN1);
  lcd.print(".");
  lcd.print(MIN2);
  lcd.print("'  ");
}
//
//
void setup()
{
  Serial.begin(9600); // definisce la velocita' (il baudrate) dei dati dal ricevitore GPS
  lcd.begin(16, 2); // inizializza il display
  pinMode(13, OUTPUT); // led attraverso il quale osservare la frequenza dei segnali dal GPS
  // (utile probabilmente anche per rallentare la successione dei cicli)
  lcd.setCursor(0, 0); // posiziona il cursore del display all'inizio della prima riga
  lcd.print(" benvenuto");
  lcd.setCursor(0, 1); // posiziona il cursore del display all'inizio della seconda
  lcd.print(" uso GPS");
  delay(3000); //attende tre secondi per consentire la lettura del display
}
//
//
void loop()
{
  while (Serial.available())
  {
    digitalWrite(13, HIGH);
    c = Serial.read(); // legge i dati provenienti dal modulo GPS
    if (gps.encode(c) // apparentemente inutile, forse decodifica i dati provenienti dal GPS
        {
          // inserire qui una eventuale codifica di trattamento del segnale ricevuto
        }
    }
    digitalWrite(13, LOW);
    gps.get_position(&lat, &lon, &fix_age); // lat/long (segnati) in millesimi di grado decimale
    LAT(); // lancia la routine di esposizione della latitudine
    LON(); // lancia la routine di esposizione della longitudine
  }
}
```