

Università degli Studi di Padova

Facoltà di Ingegneria

Corso di Laurea in Ingegneria dell'Informazione

**Tesi di Laurea**

**INTERFACCIAMENTO DI UN MODULO GPS IN UNA  
RETE DI SENSORI WIRELESS**

**Relatore:** Prof. Leonardo Badia

**Correlatore:** Dott. Riccardo Manfrin

**Laureando:** Luca Tollot

**Anno Accademico 2011 / 2012**



# INDICE

<b>Sommario</b>	1
<b>Introduzione</b>	3
<b>CAPITOLO 1</b>	
Reti di sensori wireless	
1.1 Cos'è una rete di sensori wireless	5
<b>CAPITOLO 2</b>	
Il sistema GPS	
2.1 Cenni storici	7
2.2 Descrizione del sistema GPS	7
2.3 DGPS o GPS differenziale	8
<b>CAPITOLO 3</b>	
Hardware	
3.1 Scheda di sviluppo EVK-U12	11
3.2 Modulo wireless LISA-U120	12
3.3 Modulo GPS LEA- 6H	12
3.4 Scheda di supporto per microcontrollore LPC1768	12
3.5 Scheda visualizzazione dati GPS	13
3.6 Interfacciamento dispositivi	15
<b>CAPITOLO 4</b>	
Struttura software microcontrollore LPC1768	
4.1 Impostazione microcontrollore	17
4.1.1 Impostazione oscillatore	17
4.1.2 Impostazione timer	17
4.1.3 Impostazione seriali	17
4.1.4 Led di funzionamento	18
4.2 Funzionamento a regime del microcontrollore	18

## **CAPITOLO 5**

Configurazione modulo wireless LISA-U120

### **5.1 Comandi AT**

5.1.1 Struttura comandi AT 19

5.1.2 Comando disabilitazione ECO comandi 19

5.1.3 Comando accensione GPS 19

## **CAPITOLO 6**

Configurazione modulo GPS LEA-6H

6.1 Impostazioni di default 21

6.1.1 Struttura del protocollo NMEA 22

6.2 Messaggi di impostazione 24

6.2.1 Comando impostazione baud rate 24

6.2.2 Comando disabilitazione stringhe 25

## **CAPITOLO 7**

Acquisizione dei dati

7.1 Metodo per l'acquisizione dei dati 27

## **CAPITOLO 8**

Elaborazione stringhe NMEA

8.1 Classe C++ per elaborazione dati 29

8.1.1 Funzionamento del metodo della classe 30

## **CAPITOLO 9**

Conclusioni

9.1 Osservazioni 32

9.2 Conclusioni finali 33

**Bibliografia** 35

**Ringraziamenti** 37

# SOMMARIO

Questa tesi tratta l'interfacciamento di un dispositivo GPS in una rete di sensori wireless.

Nella tesi viene descritta prima di tutto la procedura utilizzata da un nodo sensore per impostare un modulo GPS e successivamente i passi per acquisire i dati che il modulo invierà al nodo. Tali dati verranno elaborati in maniera tale da massimizzare l'informazione in un nuovo pacchetto creato all'interno del nodo sensore che verrà reso disponibile e pronto per essere trasmesso dal nodo all'interno della rete.

Alcuni dei dati acquisiti dal sistema saranno visibili in un display LCD montato su un'apposita scheda.



# INTRODUZIONE

Il progetto che verrà descritto in questa tesi rientra nello studio di particolari reti chiamate: “*reti radio di sensori*” [2] le quali costituiscono una famiglia molto vasta di sistemi che possono differenziarsi in molti aspetti, pur presentando delle similarità nelle caratteristiche principali. L’elemento base di tali reti è il “*nodo sensore*” [2], termine generico con cui si indica un insieme di dispositivi studiati principalmente allo scopo di permettere l’osservazione del mondo fisico. L’interconnessione di tali nodi per mezzo di tecnologie radio, unitamente allo sviluppo di un’architettura protocollare che consenta la cooperazione tra le diverse entità, amplifica esponenzialmente la capacità dei singoli elementi e apre la strada ad un insieme vastissimo di nuove applicazioni.

Immaginiamo allora che ci sia, per esempio, la necessità di rilevare la posizione di auto in movimento allo scopo di monitorare zone ad alto traffico. La soluzione può essere rappresentata da un sistema che interfaccia un dispositivo GPS con una rete di sensori (WSN).

Per capire meglio di cosa si tratta, nei primi due capitoli verrà descritto brevemente cos’è una rete di sensori wireless e cos’è un GPS [3], mentre nei capitoli successivi verrà descritto il progetto vero e proprio, analizzando sia la parte hardware che quella software nei minimi dettagli.

Nell’ultimo capitolo invece verranno proposte alcune osservazioni e conclusioni con un breve sguardo anche ad applicazioni future.





# CAPITOLO 1

## Reti di sensori wireless

### 1.1 Cos'è una rete di sensori wireless

Una WSN (Wireless Sensor Network) è un sistema complesso che nasce dalla cooperazione tra diversi oggetti elementari detti nodi sensori [2]. Si tratta, nella sostanza, di veri e propri *sistemi embedded*, dispositivi elettronici in grado di svolgere in modo autonomo un certo insieme di operazioni più o meno complesse, di interagire con l'ambiente circostante e di cooperare tra loro per mezzo di opportune interfacce di comunicazione.

Ed è proprio l'interconnessione di questi oggetti in una WSN che amplia le potenzialità del sistema e apre la strada alla realizzazione di servizi avanzati e innovativi a partire dalle funzionalità elementari fornite da una serie di sensori di varia natura.

I possibili campi di impiego delle WSN sono numerosi ed eterogenei. Ad esempio nel contesto della tutela ambientale, le WSN possono essere utilizzate per rivelare tempestivamente incendi boschivi, oppure per monitorare l'inquinamento di falde acquifere e laghi o, ancora, per controllare aree a rischio, come discariche, impianti industriali e chimici, centrali nucleari. Un altro loro possibile impiego è nel campo della domotica, dove si possono sfruttare la WSN per realizzare un'interazione tra ambiente e persona. Scenari futuristici, inoltre, prevedono l'utilizzo delle WSN in ambito medico, attraverso l'impianto nell'organismo di minuscoli noti sensore in grado di monitorare i parametri vitali della persona e somministrare la terapia farmacologica in modo autonomo, quando necessario.

Per comprendere appieno le potenzialità di una WSN ne descriveremo genericamente l'architettura.

Le WSN sono solitamente costituite da dispositivi alimentati a batteria che comunicano tra di loro tramite onde radio. Tali dispositivi possono disporre di sensori di vario genere come per esempio sensori di temperatura, di pressione, accelerometri, sensori di luce ecc.. In particolare una dedicata sezione di ricerca (laboratorio SIGNET) [2] dell'università di Padova, ha in questi ultimi anni, concentrato i propri studi sulla tecnologia volta allo sviluppo di "*reti di sensori wireless*". Tutto questo soprattutto per la necessità di poter disporre di reti senza infrastruttura e senza un controllo

centralizzato, perciò semplici ed economiche da realizzare. L'effettivo sviluppo delle reti di sensori, come affermano Antonio Caruso, Stefano Chessa, Swades De e Alessandro Urpi nell'articolo "GPS Free Coordinate and Assignment and Routing Wireless Sensor Networks" [1], presenta una serie di sfide di ricerca che includono lo studio dei protocolli di routing attraverso la rete. L'obiettivo è quello di migliorare la gestione dei flussi di dati prodotti da cui ne deriva la necessità di integrare la tecnologia dei database con le reti di sensori wireless. L'integrazione di queste due tecnologie, nel contempo, richiede il supporto di efficienti e robusti protocolli di routing, più complessi rispetto a quelli per sostenere la semplice diffusione dei dati.

A tale scopo i protocolli di routing che si basano sulle informazioni geografiche dei nodi, sono stati proposti come una valida alternativa agli attuali protocolli per reti wireless ad hoc, al fine di ridurre l'onere di mantenere le tabelle di routing nei nodi.

L'instradamento geografico contribuisce a ridurre il routing overhead, a patto che i nodi siano consapevoli della loro posizione fisica. Queste informazioni possono essere ottenute dotando tutti i nodi con dispositivi GPS.

Veniamo quindi all'argomento vero e proprio di questa tesi. Quello che si è realizzato nel progetto, descritto nei dettagli nei prossimi capitoli, è equipaggiare un nodo della rete con un modulo GPS e utilizzare i suoi dati come informazioni da trasmettere, per esempio, ad un server centrale. In particolare le informazioni che si vogliono acquisire sono la latitudine, la longitudine e l'altitudine. Questi dati saranno successivamente messi a disposizione per essere inviati all'interno della rete oppure a disposizione del nodo stesso. Esistono poi altri motivi per cui si perviene ad utilizzare un modulo GPS, ad esempio il rilevamento della posizione e del movimento di veicoli (car tracking): tramite una rete di sensori è possibile monitorare la posizione di un'auto in movimento. È inoltre possibile monitorare il traffico su strade particolarmente transitate.

Qualora il ricevitore fosse dotato di apposito software e apposito link di comunicazione esso potrebbe calcolare e distribuire, in tempo reale, le correzioni differenziali DGPS (se calcolate per i codici) o RTK (se calcolate per le fasi) che consentono all'utente di determinare la propria posizione con precisione sub-metrica (DGPS), o centimetrica (RTK).

# CAPITOLO 2

## Il sistema GPS

### 2.1 Cenni storici

Il programma per la realizzazione del sistema è stato creato dal dipartimento della difesa degli USA nel 1973 per consentire a navi, aeroplani e veicoli terrestri delle forze armate degli USA di determinare, in qualsiasi istante, la propria posizione in qualunque parte del globo si trovino.



Il GPS è stato creato e realizzato dal Dipartimento della Difesa statunitense. I primi 11 satelliti di tipo sperimentale sono stati lanciati dal 1978 al 1985 e sostituiti, successivamente, da quelli operativi (da 24 a 32 satelliti), a partire dal 1989. Infine il sistema è diventato pienamente operativo nel 1994.

Nel 1991 gli USA resero disponibile il servizio con il nome SPS (*Standard Positioning System*), con specifiche differenziate da quello militare denominato PPS (*Precision Positioning System*). La differenza tra i due sistemi consisteva nella introduzione della Selective Availability (SA) dove venivano introdotti errori intenzionali nei segnali satellitari allo scopo di ridurre l'accuratezza della rilevazione, consentendo precisioni solo nell'ordine di 100-150 m. Tale degradazione del segnale è stata disabilitata nel maggio del 2000, grazie a un decreto del presidente degli Stati Uniti Bill Clinton, mettendo così a disposizione degli usi civili la precisione attuale di circa 10-20 m.

Pur essendo stato concepito per scopi militari, i progettisti del sistema ne hanno consentito l'uso anche ai civili.

### 2.2 Descrizione del sistema GPS

Il sistema GPS (*Global Positioning System*), come viene descritto in [3], rappresenta una tecnologia utilizzata principalmente per la geolocalizzazione, cioè per determinare la propria posizione sulla superficie

terrestre. Il sistema di GPS più noto ed utilizzato è quello americano che si chiama NAVSTAR (quello che comunemente chiamiamo GPS).

Dal punto di vista dell'utente, il sistema è costituito da due elementi: *la costellazione dei satelliti ed il ricevitore GPS*.

I satelliti GPS sono collocati in orbita intorno al nostro pianeta, ad un'altezza di circa 20000 km rispetto alla superficie terrestre, quindi su un'orbita non geostazionaria. Questo vuole dire che se da terra potessimo osservare ad occhio nudo un qualunque satellite lo vedremmo "muovere" lungo la sua orbita ed effettuare ogni giorno due interi giri intorno alla terra. La costellazione è formata da numerosi satelliti collocati in modo tale da garantire in qualunque ora della giornata ed in qualunque luogo della terra la "visione" (o ricezione) di almeno 4 satelliti. L'orbita di ciascun satellite (e quindi la sua posizione nello spazio) è nota e calcolabile utilizzando alcuni parametri caratteristici chiamati effemeridi. Sapendo esattamente dove si trovano i satelliti possiamo determinare con esattezza la nostra posizione, a patto di conoscere a che distanza ci troviamo da ogni satellite ricevuto: questo processo si chiama trilaterazione. Per conoscere la nostra distanza dal satellite si usa un metodo abbastanza semplice. A bordo di ogni satellite è installato un orologio molto preciso (si tratta di un orologio atomico) ed una volta al secondo, ogni satellite trasmette l'ora esatta. Per un principio fisico, le onde radio non si trasmettono istantaneamente, quindi il segnale orario arriverà al nostro ricevitore GPS con un certo ritardo (si ricorda che il segnale orario deve percorrere circa 20000 km prima di raggiungerci). Tenendo quindi conto di questo ritardo e sapendo a che velocità viaggia un segnale radio, possiamo calcolare quanta strada ha percorso, e quindi la nostra distanza da ogni singolo satellite.

Per quanto riguarda l'altra componente del GPS a cui accennavamo, il ricevitore, diciamo che contemporaneamente deve essere in grado di ricevere i segnali orari generati dai vari satelliti; di misurare con quanto ritardo arrivano i segnali ricevuti e quindi di elaborare tutti i dati in tempi molto brevi per poter fornire la posizione esatta.

## **2.3 DGPS o GPS differenziale**

Come viene affermato nell'articolo [4], dopo l'introduzione della Selective Availability (SA), l'errore del sistema GPS crebbe in maniera elevata, tanto da stimolare la ricerca di un modo per compensarlo.

La soluzione che venne trovata richiedeva l'uso di almeno 2 ricevitori GPS, uno, F, posizionato stabilmente in una località fissa L con coordinate note e l'altro, M, mobile, da usare normalmente in una generica posizione P. Se i due ricevitori non sono troppo lontani l'uno dall'altro, vedranno gli stessi satelliti e riceveranno gli stessi segnali. Inoltre si troveranno all'incirca nelle stesse condizioni atmosferiche. A questo punto entrambi potranno determinare la propria posizione che sarà affetta dall'errore E dovuto alla SA. In prima approssimazione possiamo pensare che questo errore E sia uguale per entrambi i ricevitori, e quindi possiamo sfruttare questa circostanza per correggere l'errore del secondo ricevitore. La posizione data dal GPS F è pari alle coordinate "reali" L (che sono note), più l'errore E. Quindi, sottraendole alla lettura del GPS F, si ottiene per differenza l'errore E. Se allora F comunica l'errore E al secondo GPS M, quest'ultimo dovrà semplicemente sottrarre E dalla propria lettura per ottenere la propria posizione esatta.



# CAPITOLO 3

## Hardware

### 3.1 Scheda di sviluppo EVK-U12

La scheda di sviluppo utilizzata nel progetto è la EVK-U12 3.75G HSPA Wireless evaluation kit fornita dalla U-BLOX.

In questa scheda sono presenti in particolare un modulo wireless e un modulo GPS, i quali verranno utilizzati nel progetto.

Le porte di comunicazione che saranno utilizzate di questa scheda di sviluppo sono la seriale con standard RS232 del modulo wireless e la seriale asincrona del GPS.

Tramite lo switch SW403 si attiva la seriale RS232 del modulo wireless disponibile nel connettore DB9 mentre con lo switch SW304 si attiva la comunicazione tra il modulo wireless e il modulo GPS tramite il protocollo I2C.

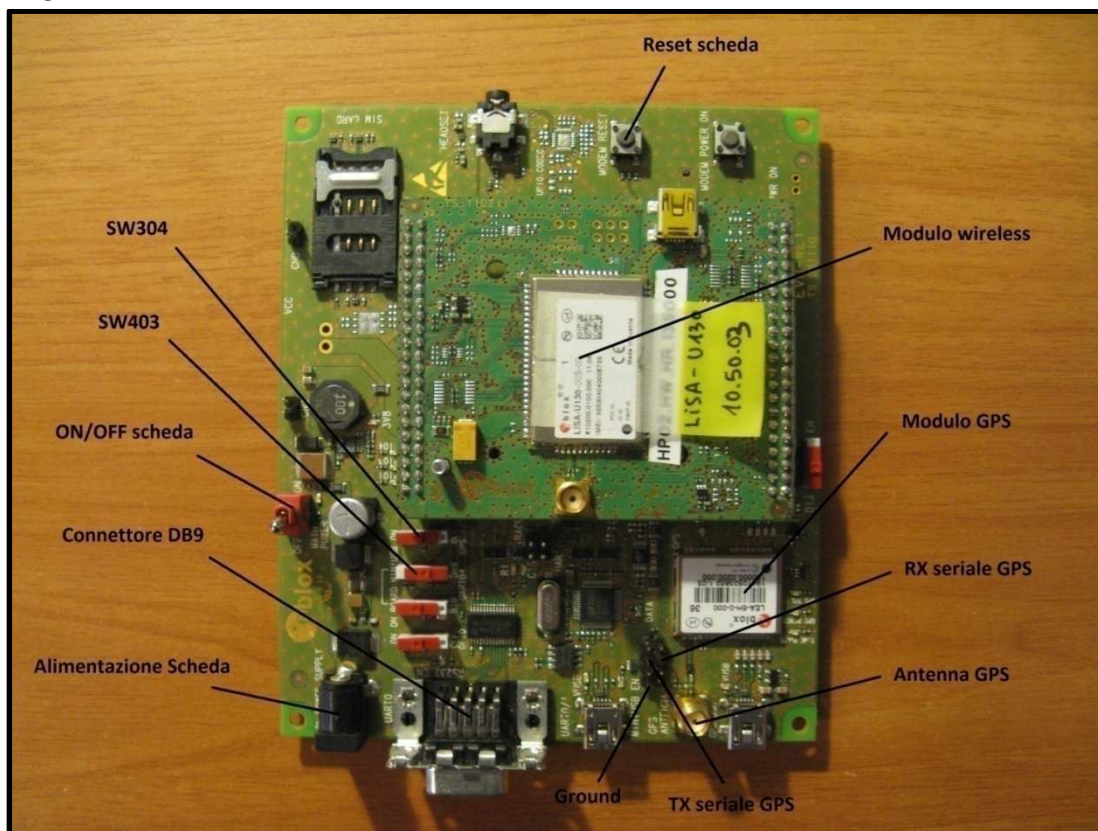


Figura 3.1

## **3.2 Modulo wireless LISA-U120**

Il modulo LISA-U120 è un modulo wireless GSM/UMTS, che in questo progetto è stato utilizzato per attivare l'alimentazione del modulo GPS e quindi accenderlo o spegnerlo. Questa procedura è stata eseguita inviando un comando AT al modulo LISA-U120 (vedi cap. 3.3). Dopo l'accensione, la seriale asincrona del modulo Gps risulta di default immediatamente attiva e trasmette di continuo (ogni secondo) i dati disponibili sulla linea "Tx" mentre la linea "Rx" può essere usata per impostare il GPS. Contemporaneamente viene attivata la seriale I2C tramite la quale il modulo wireless, con esso direttamente collegato, invia o su richiesta riceve dei dati dal GPS.

## **3.3 Modulo GPS LEA-6H**

Il modulo GPS LEA-6H utilizzato nel progetto è un GPS di ultima generazione con un alto grado di performance in varie condizioni ambientali, inoltre è caratterizzato da un basso consumo. La seriale asincrona è raggiungibile dalla scheda tramite un doppio connettore a pettine con passo 2,54 [mm] come indicato in figura 3.1. L'accensione e lo spegnimento possono essere attivati solo tramite il modulo wireless perché la scheda non è dotata di un interruttore manuale per l'alimentazione del GPS. Questo vuol dire che quando la scheda viene accesa, di default il GPS non è alimentato.

Il modulo, per ricevere i segnali dai satelliti, deve essere collegato tramite l'apposito connettore visibile in figura 3.1 all'antenna. Questa è preferibile posizionarla all'esterno degli edifici con lo scopo di avere un buon segnale e quindi dei dati validi, altrimenti i dati potrebbero non essere ottimali per i ritardi che i segnali (inviati dai satelliti) potrebbe subire prima di arrivare al ricevitore e questo a causa dei numerosi ostacoli che potrebbero dare origine a ripetute riflessioni e ad attenuazioni.

## **3.4 Scheda di supporto per microcontrollore LPC1768**

Il microcontrollore usato nel progetto è un LPC1768 della NXP. È dotato di un microprocessore Arm Cortex-M3 con una frequenza massima di 100 MHz. Esso è dotato di quattro porte seriali asincrone, due delle quali sono state utilizzate nel progetto. Nella scheda di supporto sono inoltre stati aggiunti



due led collegati a due GPIO del microcontrollore. La funzione di questi led verrà illustrata nel prossimo capitolo.

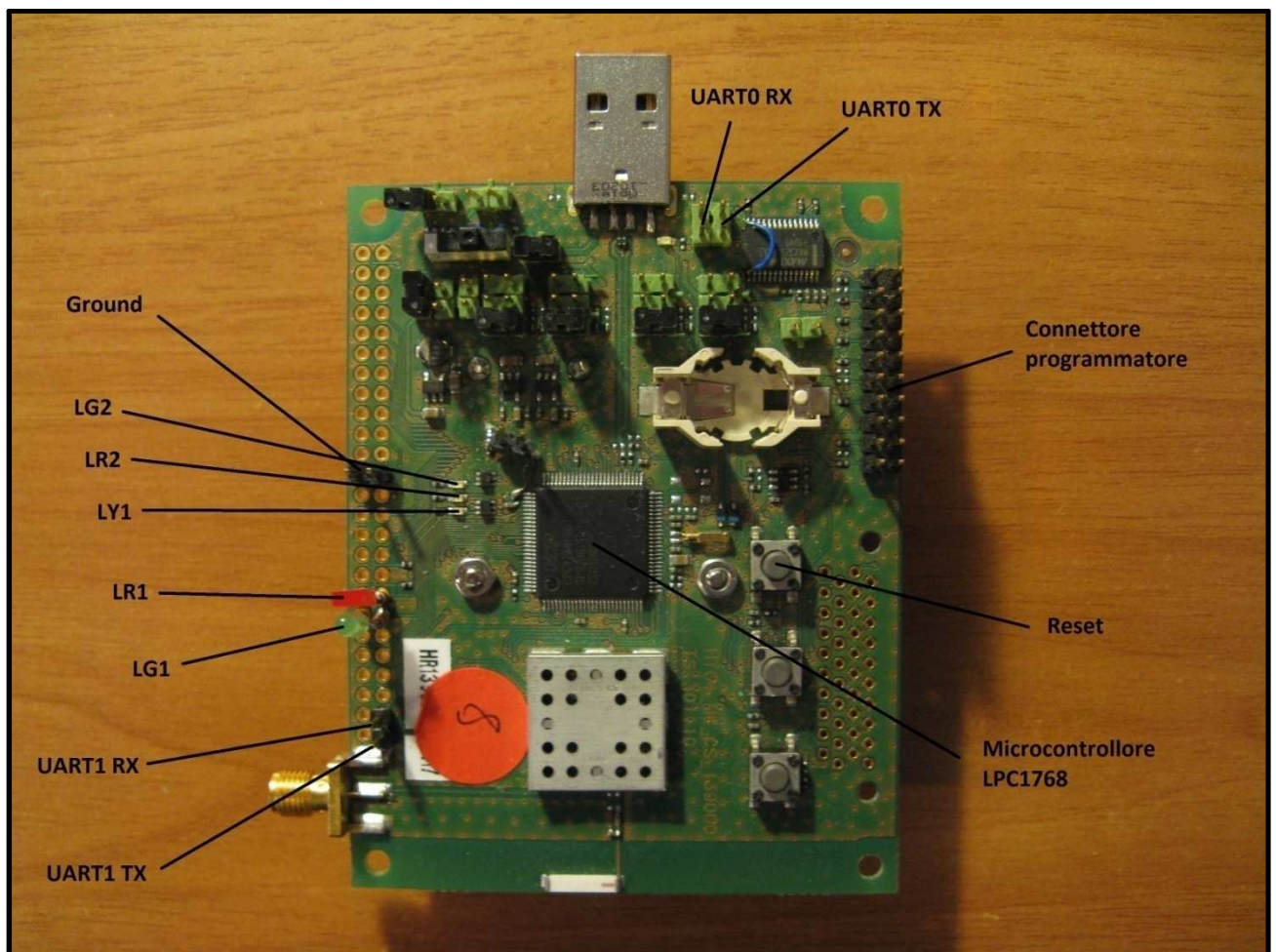


Figura 3.2

### 3.5 Scheda visualizzazione dati GPS

Questa scheda, che ho realizzato personalmente per lo sviluppo di vari progetti, in origine è stata dotata di un modulo GPS (JUPITER32), per il quale è stato creato un menu apposito per la visualizzazione di alcuni suoi dati. Per il progetto descritto nella tesi la scheda è stata però opportunamente modificata, scollegando il modulo GPS (JUPITER32) e collegando il modulo GPS LEA-6H della scheda di sviluppo EVK-U12. Ho quindi modificato il firmware per il microcontrollore MC9S12XDP512 (freescale) per acquisire e visualizzare i dati richiesti tramite il display LCD.

La scheda presenta un grande vantaggio rappresentato dalla possibilità di creare varie interfacce grafiche, inserendo delle immagini all'interno di una

libreria creata appositamente. Tramite la tastiera è inoltre possibile selezionare le voci visualizzate nel display spostandosi nei menu e sottomenu relativi. Le immagini che appaiono possono anche rappresentare brevi animazioni.

Le figure da inserire nella libreria, rigorosamente in formato bitmap monocromatico, possono essere realizzate tramite un qualsiasi editor d'immagini di uso comune; successivamente, grazie ad un software di mia realizzazione, si possono convertire i file bitmap in un nuovo formato, che io stesso ho ideato allo scopo di minimizzare lo spazio occupato dall'immagine. Tutto il codice di mia proprietà, utilizzato per la realizzazione delle interfacce nel display LCD è possibile trovarlo al seguente indirizzo:

<http://www.123homepage.it/lucatollot>

Nella scheda sono presenti anche dei sensori, in particolare un sensore magnetico, utilizzato come bussola. La funzione di bussola è presente nell'applicazione, caricata nella scheda e visualizzabile tramite apposita voce presente nel menu dal quale è possibile accedere anche ad una particolare sezione che permette di effettuare la taratura del sensore che si effettua facendo compiere un giro di 360° alla scheda intorno a se stessa. Nel tempo necessario per effettuare il giro, la scheda acquisisce le letture del sensore, permettendo di determinare i riferimenti per individuare dove si trova il nord magnetico.

Oltre al sensore magnetico è possibile anche collegare un accelerometro con il quale si può determinare l'accelerazione impressa al sensore lungo tre assi ortogonali fra loro.

Infine tutta la scheda risulta alimentata da una batteria da 7.2 V. Questa tensione è stata modificata, creando un semplice ma efficace circuito che permette di ottenere i 5 V per alimentare il microcontrollore e il display e 3.3 V per alimentare i vari sensori e il GPS (JUPITER32).

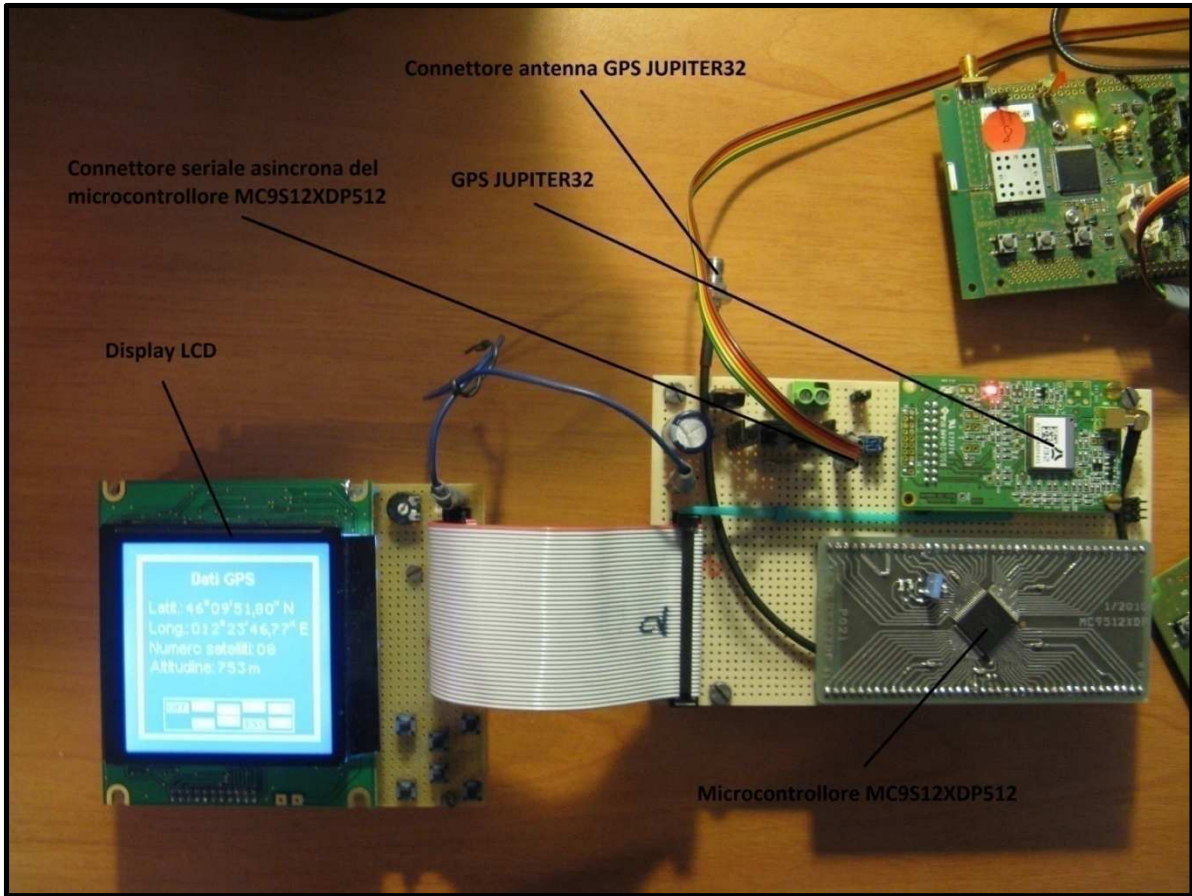


Figura 3.3

### 3.6 Interfacciamento dispositivi

Nella figura 3.4 si possono vedere i collegamenti tra i vari dispositivi utilizzati nel progetto. La UART0 del microcontrollore è collegata alla seriale RS232 del modulo wireless mentre la UART1 è collegata alla seriale asincrona del GPS. In alternativa la seriale del GPS può essere collegata alla seriale della scheda di visualizzazione (figura 3.3) per permettere la visualizzazione dei dati nel display. Da osservare anche l'antenna per il GPS che è stata collegata nell'apposito connettore della scheda.

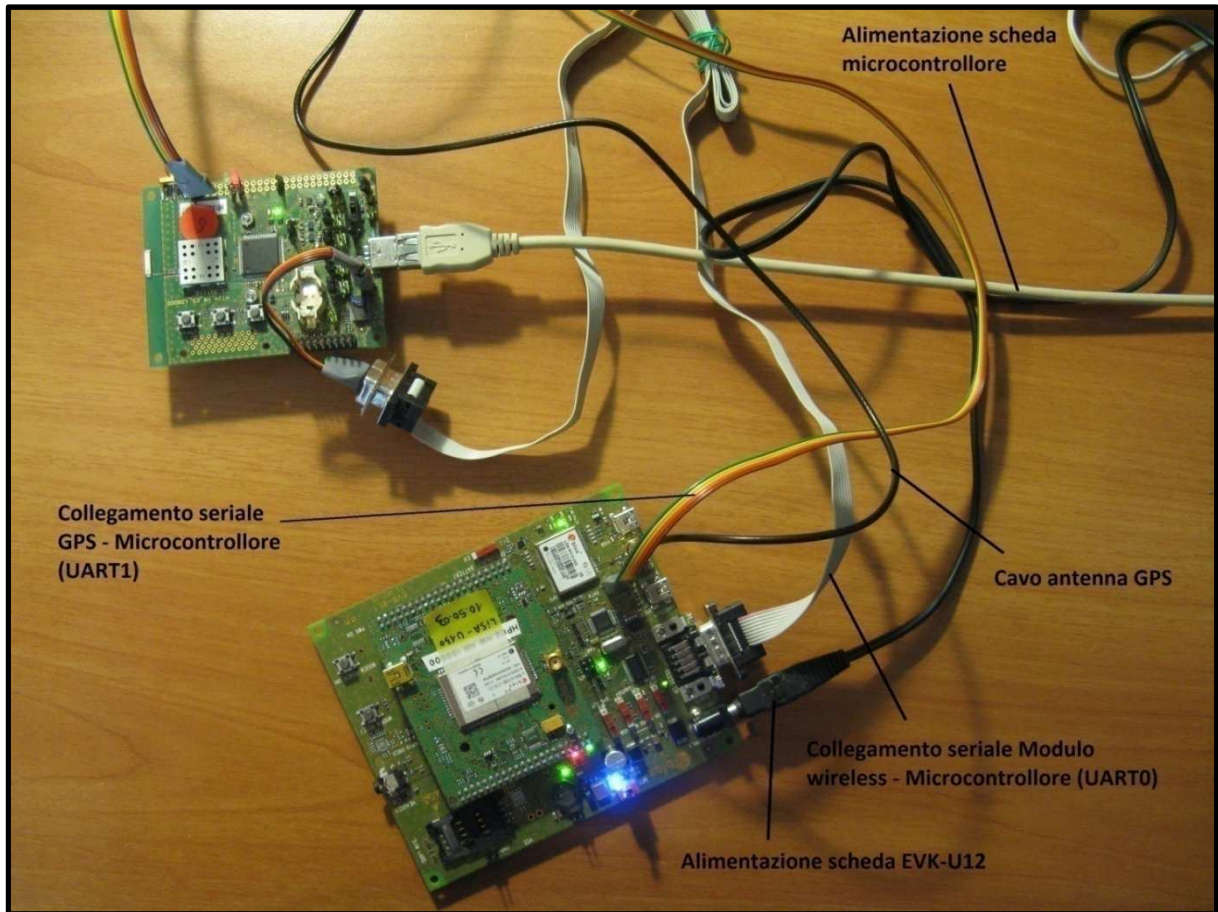


Figura 3.4

# CAPITOLO 4

## Struttura software microcontrollore LPC1768

### 4.1 Impostazione microcontrollore

Di seguito verranno illustrate le impostazioni iniziali del microcontrollore:

#### 4.1.1 Impostazione oscillatore

La sorgente di clock utilizzata dal microcontrollore è fornita da una oscillatore presente nella stessa scheda di supporto dove è posizionato il microcontrollore ed ha esattamente una frequenza di clock di 1 MHz. Successivamente tramite il PLL presente nel microcontrollore si porta questa frequenza a 100 MHz, la quale verrà in seguito utilizzata come sorgente di clock per il microprocessore. Per tutte le altre periferiche presenti nel microcontrollore questa frequenza verrà prima divisa per quattro da un divider interno e poi potrà essere utilizzata, ottenendo così una frequenza di 25 MHz.

#### 4.1.2 Impostazione timer

All'interno del microcontrollore sono presenti quattro timer che possono generare un interrupt dopo un certo intervallo di tempo prestabilito in funzione della sorgente di clock a cui si appoggiano. Nel progetto è stato utilizzato il timer 0 per generare un interrupt ogni millisecondo e questo per avere un riferimento del tempo all'interno del microcontrollore. La sorgente di clock a cui si è appoggiato il timer è quella descritta nel paragrafo precedente e cioè quella con una frequenza di 25 MHz.

#### 4.1.3 Impostazione seriali

Le seriali utilizzate nel microcontrollore sono la UART0 e la UART1. La prima è collegata alla seriale del modulo wireless ed è impostata alla velocità di 115200 bit/s

con 1 bit di stop e senza controllo di parità mentre la seconda è collegata alla seriale del GPS ed è impostata alla velocità di 9600 bit/s sempre con 1 bit di stop e senza controllo di parità.

#### **4.1.4 Led di funzionamento**

Per la posizione dei led fare riferimento alla figura 3.2.

Il led LG1 individua il corretto funzionamento del microcontrollore e normalmente lampeggia con una frequenza di 1.25 Hz. Esso sta ad indicare che il microcontrollore è acceso e sta funzionando correttamente con la giusta frequenza di clock. Se non lampeggia con questa frequenza il microcontrollore non funziona correttamente e probabilmente sarà necessario eseguire un reset.

Il led LR1 identifica l'avvenuta ricezione da parte del microcontrollore della stringa GGA in formato NMEA.

Il led LG2 conferma l'avvenuta accensione del GPS secondo la risposta fornita dal modulo wireless.

Il led LY1 lampeggia con una frequenza di 2.5 Hz se i dati ricevuti non sono validi (fix non effettuato), mentre lampeggia con frequenza di 0.5 Hz se i dati sono validi (fix effettuato).

## **4.2 Funzionamento a regime del microcontrollore**

Una volta eseguite tutte le impostazioni descritte precedentemente, comprese quelle che riguardano i moduli GPS e wireless, le quali verranno descritte in dettaglio nei prossimi due capitoli, il programma entra in un loop infinito in attesa di elaborare i dati in arrivo dal GPS, attraverso una classe scritta in C++.

# CAPITOLO 5

## Configurazione modulo wireless LISA-U120

### 5.1 Comandi AT

La maggior parte dei modem utilizza i comandi AT Hayes, uno specifico insieme di comandi originalmente sviluppato per il modem Hayes Smartmodem da 300 bit/s. Questa serie di comandi prepara il modem per la comunicazione, impostando caratteristiche come il tipo di connessione, i tempi di attesa, la rilevazione del segnale di occupato, eccetera.

#### 5.1.1 Struttura comandi AT

La struttura di questi comandi è la seguente:

```
AT<command_name><string><CR>
```

AT: prefisso di ogni comando

<command\_name>: nome del comando e può talvolta essere preceduto dal “+”

<string>: sequenza dei parametri del comando

<CR>: carattere di terminazione

#### 5.1.2 Comando disabilitazione ECO comandi

Questo comando viene inviato al modulo wireless per dirgli che nella risposta che invierà ai successivi comandi, ometta l’eco del comando appena inviato.

Il comando inviato è il seguente: ATE0<CR>

#### 5.1.3 Comando accensione GPS

Questo comando viene inviato al modulo wireless per accendere il modulo GPS.

Una volta inviato il comando il modulo wireless tramite un suo GPIO attiverà un transistor MOS che fornirà la tensione adeguata al modulo GPS e permetterà a quest'ultimo di entrare in funzione con le impostazioni di default.

Il comando inviato è il seguente: AT+UGPS=1,0<CR>

Nel seguito è riportata la sequenza dei comandi inviati al modulo wireless, con le relative risposte inviate dal modulo al microcontrollore. In alto si hanno i comandi mentre in basso le risposte. Il comando più a sinistra è quello per la disabilitazione dell'eco dei comandi mentre il comando più a destra è quello per l'accensione del GPS.

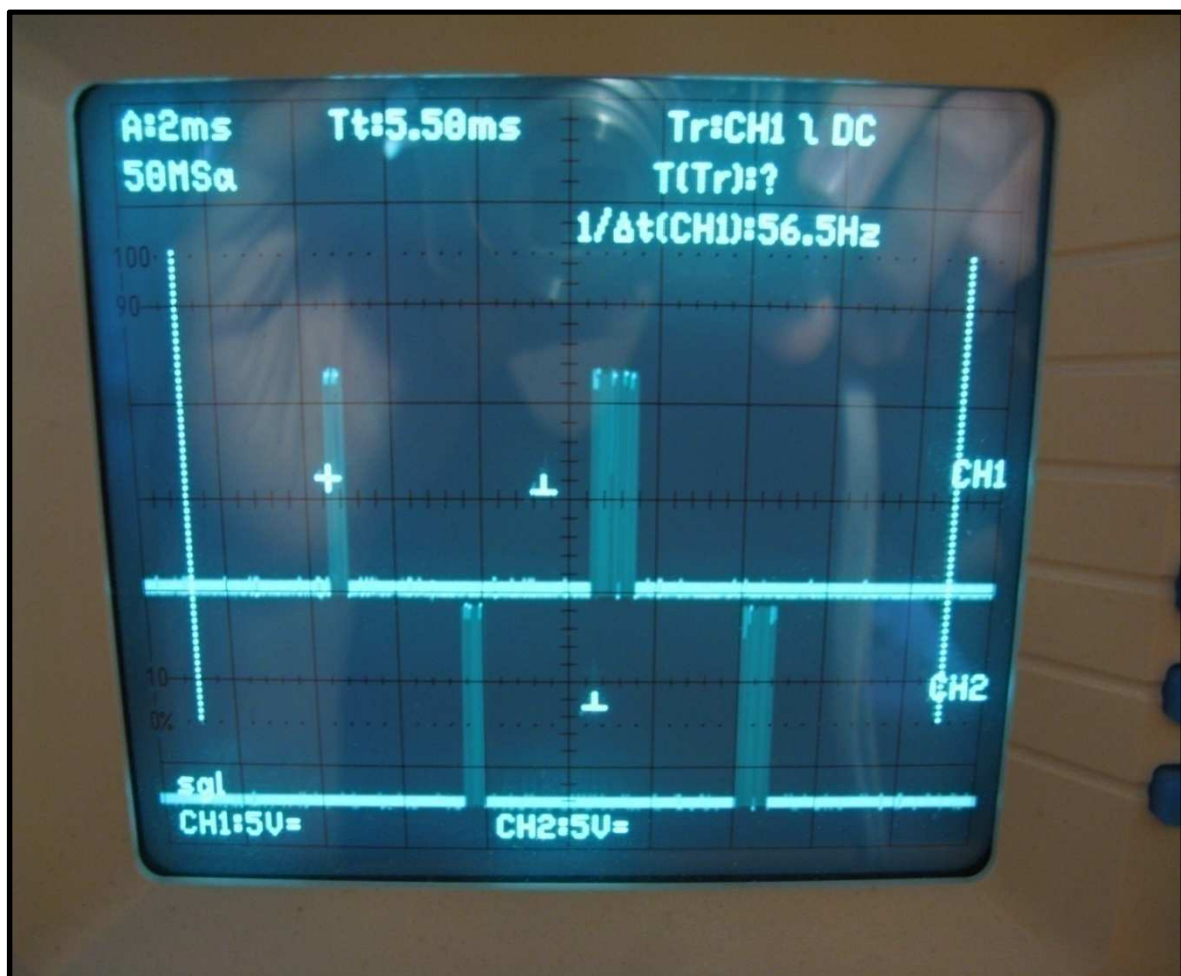


Figura 5.1



# CAPITOLO 6

## Configurazione modulo GPS LEA-6H

### 6.1 Impostazione di default

Ogni volta che il modulo GPS viene acceso, carica alcune impostazioni di default che usa per inviare i dati che ha elaborato ai dispositivi a cui è collegato. In particolare, la seriale asincrona viene impostata per trasmettere e/o ricevere dati alla velocità di 9600 bit/s con un bit di start, uno di stop e nessuno bit di parità. Ogni secondo e senza pause vengono inviate sulla linea “Tx”, in uno speciale formato standardizzato dal National Marine Electronics Association (NMEA), le seguenti stringhe di dati:

1. GGA (dati fissi per il Global Position System)
2. GLL (posizione geografica – Latitudine/Longitudine)
3. GSA (degradazione di accuratezza e il numero di satelliti attivi nel sistema di navigazione satellitare globale (GNSS))
4. GSV (satelliti in vista nel GNSS)
5. RMC (minimo GNSS raccomandato)
6. VTG (direzione, velocità orizzontale)
7. TXT (informazioni della versione software del modulo)

Di seguito è riportato un esempio dei dati trasmessi dalla seriale del GPS:

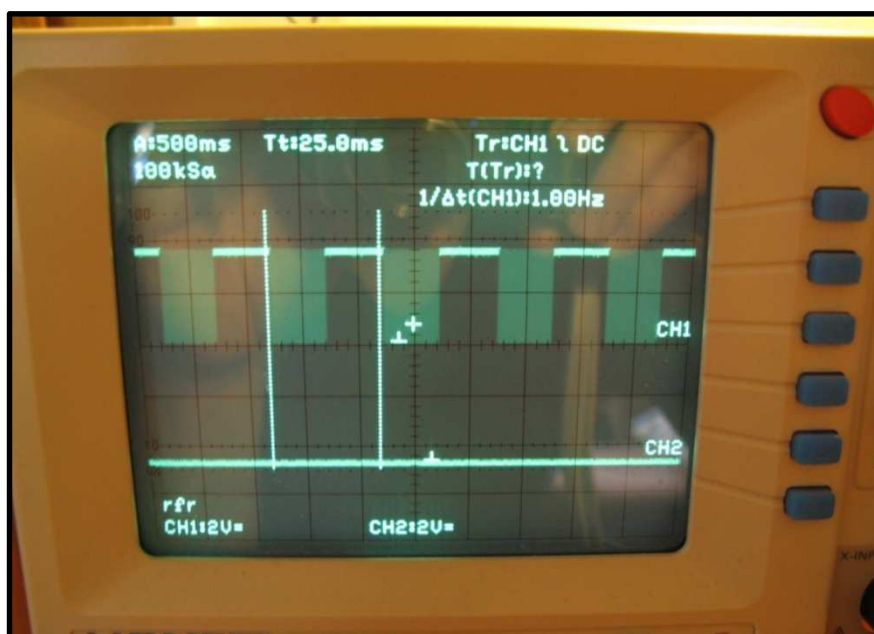


Figura 6.1

### 6.1.1 Struttura del protocollo NMEA

I dati vengono trasmessi nello standard NMEA, usando caratteri ASCII a 8 bit. La trasmissione inizia con un bit di start (zero logico), seguito da 8 bit di dati e un bit di stop (uno logico), aggiunto alla fine. Il bit di parità non è usato. Ogni set di dati provenienti dal GPS, è strutturato allo stesso modo, come ad esempio la stringa

“GGA”: \$GPGGA, inf\_1, inf\_2, inf\_3,..., inf\_n\*CS<CR><LF>.

Le funzioni dei caratteri individuali o dei set di caratteri è spiegata in tabella:

Campo	Descrizione
\$	Inizio del set di dati
GP	Informazioni provenienti dal GPS
GGA	Identificatore del set di dati
inf_1... inf_n	Informazioni contenute nel set
,	Separatore fra le informazioni
*	Separatore per il checksum
CS	Checksum
<CR><LF>	Fine del set di dati

Tabella 6.1

Il numero massimo di caratteri usati non può essere superiore a 80. Nel conteggio del numero di caratteri non vengono inclusi il carattere di inizio \$ e quelli di fine <CR><LF>.

Le seguenti stringhe sono ottenute dalla ricezione di un messaggio proveniente dal GPS in formato NMEA:

\$GPRMC,130303.0,A,4717.11578,N,00833.91254,E,000.03,043.4,2006 01, 01.3,W*7D<CR><LF>
\$GPZDA,130304.2,20,06,2001,,*56<CR><LF>
\$GPGGA,130304.0,4717.11578,N,00833.91254,E,1,08,0.94,00499,M,0 47, M,,*59<CR><LF>
\$GPGLL,4717.115,N,00833.912,E,130304.0,A*33<CR><LF>
\$GPVTG,205.5,T,206.8,M,000.04,N,000.08,K*4C<CR><LF>

\$GPGSA,A,3,13,20,11,29,01,25,07,04,,,,,1.63,0.94,1.33*04<CR><L>
\$GPGSV,2,1,8,13,15,208,36,20,80,358,39,11,52,139,43,29,13,044,36*42<CR><LF>

Tabella 6.2

Nella tesi è presa in considerazione solo quella identificata da GGA, poiché essa contiene tutte le informazioni di nostro interesse.

Esaminiamo la stringa GGA come esempio:

\$GPGGA,184936.00,4508.43100,N,00737.21937,E,1,06,1.69,278.4,M,47.2,M,,\*5F<CR><L>

Campo	Descrizione
GGA	Identificativo protocollo GGA
184936.00	Ora UTC (hhmmss.ss)
4508.43100	Latitudine (ddmm.mmmmm)
N	N = nord; S = sud
00737.21937	Longitudine (dddmm.mmmmm)
E	E = est; W = ovest
1	Qualità del Fix: 0 = Invalido 1 = GPS fix 2 = DGPS fix 3 = Fix GPS PPS 4 = RTK (Real Time Kinematic) intera 5 = RTK float 6 = Navigazione Stimata (dead reckoning) 7 = Input Manuale 8 = Simulazione
06	Satelliti in vista
1.5	HDOP
278.4	Altitudine rispetto al livello del mare
M	Unità di misura metri
47.2	Separazione geoidale
M	Unità di misura metri
vuoto	Tempo dall'ultimo aggiornamento DGPS
vuoto	Id della stazione DGPS
*5F	Checksum
<CR><LF>	Caratteri di terminazione

Tabella 6.3

## 6.2 Messaggi di impostazione

È possibile inviare dei comandi al GPS attraverso il protocollo NMEA per eseguire delle impostazioni, come disabilitare l'invio di alcune stringhe o cambiare il bit rate della seriale.

La struttura del comando è quella riportata di seguito:

```
$PUBX,N,par_1,..., par_N,*CS<CR><LF>
```

Campo	Descrizione
\$PUBX	UBX protocol header
N	Tipo di comando
par_1,..., par_N	Parametri del comando
CS	Checksum
<CR><LF>	Caratteri di terminazione

Tabella 6.4

Il checksum delle stringhe NMEA è calcolato iniziando da 0x00 ed eseguendo l'XOR fra tutti i caratteri a partire dal primo dopo il carattere di inizio messaggio "\$" e fino all'ultimo prima del terminatore "\*".

Il risultato calcolato è formattato in un due caratteri ASCII che rappresentano un numero esadecimale.

Nel software scritto per il microcontrollore è presente una funzione apposita per calcolarlo.

### 6.2.1 Comando impostazione baud rate

Per aumentare l'efficienza del sistema è stata cambiata la velocità di trasmissione della seriale del GPS, da 9600 b/s a 115200 b/s. In questo modo, avendo abilitato solo la stringa GGA, i dati arrivano in meno di 5,6 ms. Il comando per impostare la velocità a 115200 è il seguente:

```
$PUBX,40,1,0007,0003,115200,0*18<CR><LF>
```

## 6.2.2 Comando disabilitazione stringhe

Durante l'impostazione del GPS sono stati inviati alcuni comandi per disabilitare le stringhe NMEA non necessarie per il progetto; in particolare sono state disabilitate tutte tranne la GGA.

Qui di seguito è riportato il comando per disabilitare la stringa GLL:

```
$PUBX,40,GLL,0,0,0,0,0,0*5C<CR><LF>
```

Nel seguito è riportata la sequenza dei comandi inviati al modulo GPS. A destra si può notare una zona più densa di dati che sta ad indicare il cambio di velocità da 9600 b/s a 115200 b/s.

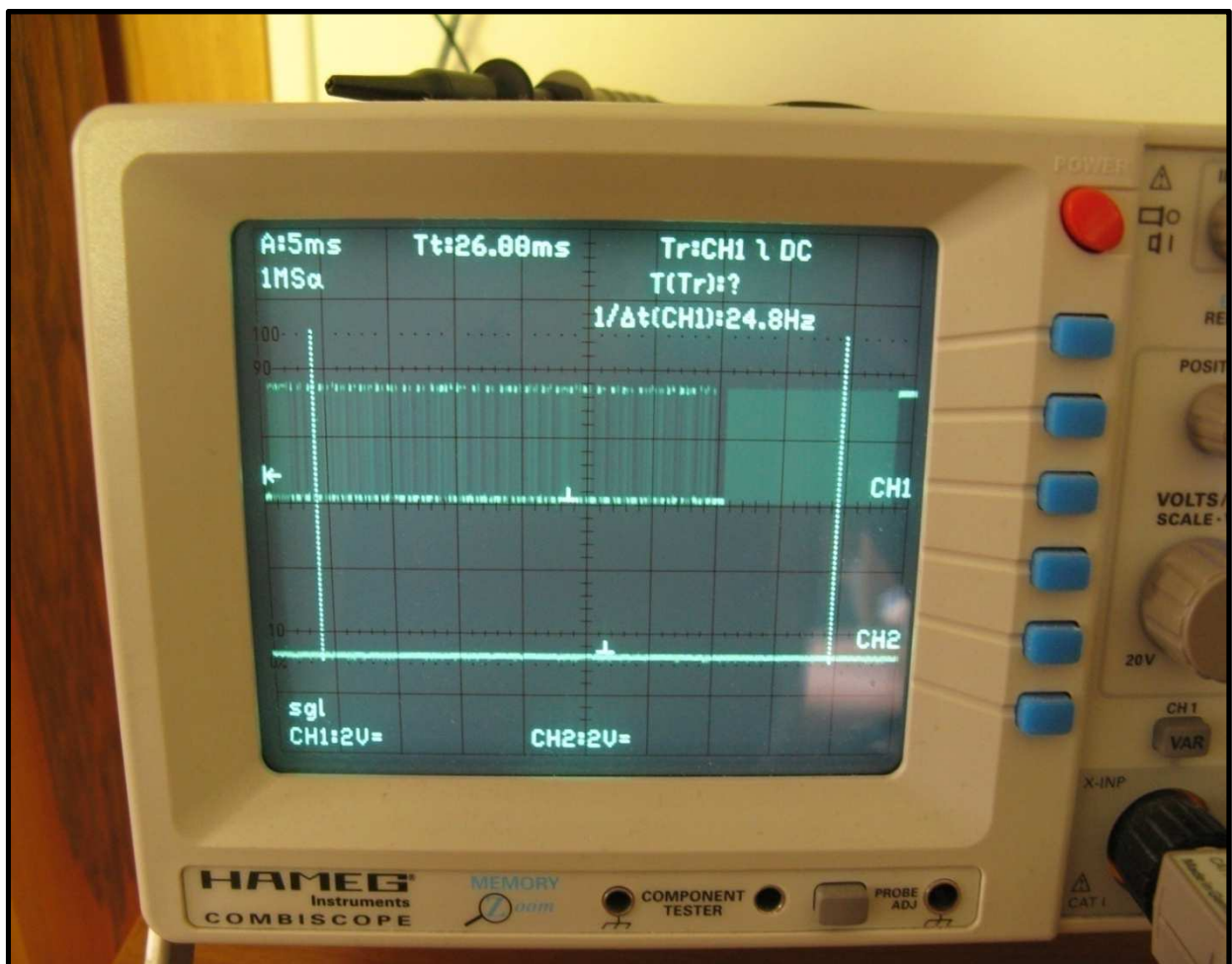


Figura 6.2



# CAPITOLO 7

## Acquisizione dei dati

### 7.1 Metodo per l'acquisizione dei dati

Verrà ora qui descritta la procedura per l'acquisizione dei dati provenienti dal GPS tramite il microcontrollore.

Dopo le impostazioni eseguite il GPS trasmette sulla linea Tx della seriale asincrona, ogni secondo, alla velocità di 115,2 kb/s, solo la stringa GGA. La linea Tx del GPS è stata collegata al pin Rx del microcontrollore. Con il firmware è stato abilitato l'interrupt di ricezione che viene attivato dopo l'arrivo di un carattere formato da 10 bit. L'inizio della ricezione di ogni stringa "GGA" viene individuata dal firmware (ogni carattere della stringa viene trasmesso dal GPS in modo consecutivo con un ritardo dal precedente inferiore alla sua durata) e memorizzata in un vettore di lunghezza opportuna. In interrupt di ricezione, dopo l'arrivo di un dato, viene impostato a uno un flag di controllo e contemporaneamente viene azzerata la variabile "temp\_fine\_RX". Nell'interrupt del timer, generato ogni millisecondo, viene eseguito il controllo di fine ricezione della stringa "GGA" e se il flag di controllo è a uno la variabile "temp\_fine\_RX" viene incrementata. A questo punto per sapere quando la ricezione è terminata, sempre all'interno dell'interrupt timer è presente una funzione che controlla se la variabile "temp\_fine\_RX" ha un valore maggiore o uguale a due. Se questa condizione è rispettata la ricezione è sicuramente terminata e il flag di controllo viene azzerato. Considerato che un carattere ha una durata di 86,8 us circa e che l'invio dei dati in seriale è continuo, se non vengono ricevuti dati per più di un millisecondo, la ricezione può considerarsi conclusa. Una volta conclusa la ricezione il buffer usato nell'interrupt di ricezione della seriale viene svuotato e i dati vengono trasferiti in un altro buffer, disponibile per essere utilizzati.

Questo viene fatto per aumentare il tempo a disposizione per operare sui dati ricevuti, infatti se questo passaggio non venisse effettuato i dati presenti nel buffer di ricezione non potrebbero essere manipolati a piacere perché se iniziasse un'altra trasmissione i dati presenti in quest'ultimo verrebbero sovrascritti.

In ogni caso prima di poter manipolare i dati bisogna controllare che non sia in corso il trasferimento dei dati dal buffer di ricezione al secondo buffer.

Quest'ultimo controllo viene effettuato all'interno della classe per l'elaborazione dei dati.



# CAPITOLO 8

## Elaborazione e visualizzazione stringhe NMEA

### 8.1 Classe C++ per elaborazione stringhe NMEA

Nel software è stata scritta una classe c++ apposta per l'elaborazione dei dati provenienti dal GPS. Questa classe consiste in unico metodo che quando viene invocato, richiede come unico parametro l'indirizzo del buffer contenente i dati del GPS e restituisce un puntatore ad un vettore contenente i dati elaborati e organizzati in gruppi di otto bit, per avere un vettore di byte di lunghezza ridotta e pronto per essere inviato attraverso la rete di sensori.

Nella tabella seguente è illustrata la struttura di questo vettore:

Posizione	Descrizione
0	Stato dei dati
1	Gradi latitudine (GG)
2	Primi latitudine (MM)
3	Secondi latitudine (SS)
4	Decimi e centesimi di secondo latitudine (DC)
5	Millesimi di secondo latitudine (M)
6	Nord/Sud
7	Gradi longitudine (GGG)
8	Primi longitudine (MM)
9	Secondi longitudine (SS)
10	Decimi e centesimi di secondo longitudine (DC)
11	Millesimi di secondo longitudine (M)
12	Est/Ovest
13	Satelliti
14	Altitudine_H
15	Altitudine_L
16	Ore
17	Minuti
18	Secondi

Tabella 8.1

Il primo byte del vettore si riferisce allo stato dei dati. Se il valore assunto è “-1” i dati non sono disponibili e tutti i valori che seguono impostati a zero. Se il valore assunto è “0” significa fix non effettuato da parte del GPS. Dal valore “1” in poi i dati sono validi. Dalla posizione 1 alla posizione 5 ci sono i byte relativi alla latitudine che è espressa nel seguente formato: GG°MM’SS,DCM”. Nella posizione 6 si ha l’indicazione riguardante in quale emisfero ci si trova. Dalla posizione 7 alla posizione 11 ci sono i byte relativi alla longitudine che è espressa nel seguente formato: GGG°MM’SS, DCM”. Nella posizione 12 si ha l’indicazione riguardante se ci si trova a est o a ovest di Greenwich. Nella posizione 13 si ha il numero di satelliti utilizzati in quel momento. Dalla posizione 14 alla posizione 15 si ha l’altitudine divisa in due byte dove “Altitudine\_H” sta a indicare il byte più significativo e “Altitudine\_L” il byte meno significativo. Infine dalla posizione 16 alla posizione 18 si ha l’ora corrente.

### **8.1.1 Funzionamento del metodo della classe**

Il metodo della classe che restituisce i dati elaborati, come detto prima riceve come parametro l’indirizzo del buffer contenente tutti i dati ricevuti dal GPS. Esso esegue successivamente una ricerca all’interno del buffer della stringa GGA per mezzo dell’algoritmo KMP (Knuth-Morris-Pratt) descritto nel libro “Strutture dati e Algoritmi in java” di Goodrich Michael T. e Roberto Tamassia. Se la trova prosegue con l’extrapolazione dei dati altrimenti ritorna come valore nella posizione 0 del vettore il valore “-1” che sta ad indicare la non disponibilità dei dati. C’è anche un altro caso in cui può ritornare questo valore e cioè quando nell’interrupt del timer è in corso il passaggio dei dati ricevuti, dal buffer di ricezione al buffer passato a questo metodo.

Se la stringa viene trovata il passo successivo è controllare che sia avvenuto il fix del GPS.

Nel caso il fix non sia effettuato, ciò equivale al valore “0”, viene restituito uno “0” nella posizione iniziale del vettore.

Se invece il valore è maggiore di 1 i dati sono validi e quindi si procede con l’elaborazione della latitudine.

Le informazioni che saranno elaborate in seguito saranno rappresentate in formato numerico decimale, perciò è importante sapere che i dati ricevuti sono caratteri ASCII e quindi se un carattere indica un numero bisogna sottrargli 48 per ottenere il numero reale che rappresenta.

A questo punto il formato della latitudine nella stringa GGA è il seguente: GG°MM',XXXXX e viene convertito nel formato che segue: GG°MM'SS, DCM", dove GG, MM, SS, DC e M occupano rispettivamente un byte ciascuno per un totale di 5 byte.

Questi byte, più un altro byte per rappresentare il nord o il sud, il quale rappresenta invece un carattere ASCII, vengono successivamente caricati nel vettore che il metodo restituisce.

Per ricostruire poi il valore dei secondi si usa la seguente formula:

$$\text{secondi di latitudine} = \text{SS} + (\text{DC} * 10 + \text{M}) * 10^{-3}$$

In modo analogo si procede per l'elaborazione e la memorizzazione della longitudine.

In seguito viene caricato il byte relativo al numero di satelliti utilizzati e due byte sono invece utilizzati per l'altitudine dove per ricostruire il valore bisogna usare la seguente formula:  $\text{altitudine} = \text{Altitudine\_H} * 100 + \text{Altitudine\_L}$ .

Infine altri tre byte sono utilizzati per memorizzare l'ora UTC.

È da notare che questo sistema di rappresentazione delle informazioni è stato scelto per immagazzinare tutti i dati nel minor spazio possibile.

# CAPITOLO 9

## Conclusioni

### 9.1 Osservazioni

Attraverso la scheda di visualizzazione (figura 9.1), dov'è presente un display LCD, è stato possibile visualizzare alcuni dati provenienti dal GPS, in particolare la latitudine, la longitudine, il numero di satelliti e l'altitudine. Nell'immagine che segue si possono osservare i dati visualizzati tramite il display LCD.

Osservando i dati per un certo periodo si è notato che, mantenendo la posizione del GPS fissa, i dati non variavano di molto. Se invece si fanno apparire i dati relativi al GPS JUPITER32 si può notare che i dati variano molto frequentemente. Questa differenza è dovuta al fatto che il modulo usato in questo progetto, essendo più recente utilizza la modalità DGPS [4] per funzionare, mentre l'altro no. Questo ovviamente porta ad una differenza di precisione dei dati forniti, che è maggiore nel GPS di ultima generazione.

In questo modo quindi si è potuto anche mettere a confronto due diversi tipi di moduli GPS.



Figura 9.1

Nella figura 9.2 si può osservare l'intero sistema in funzione.

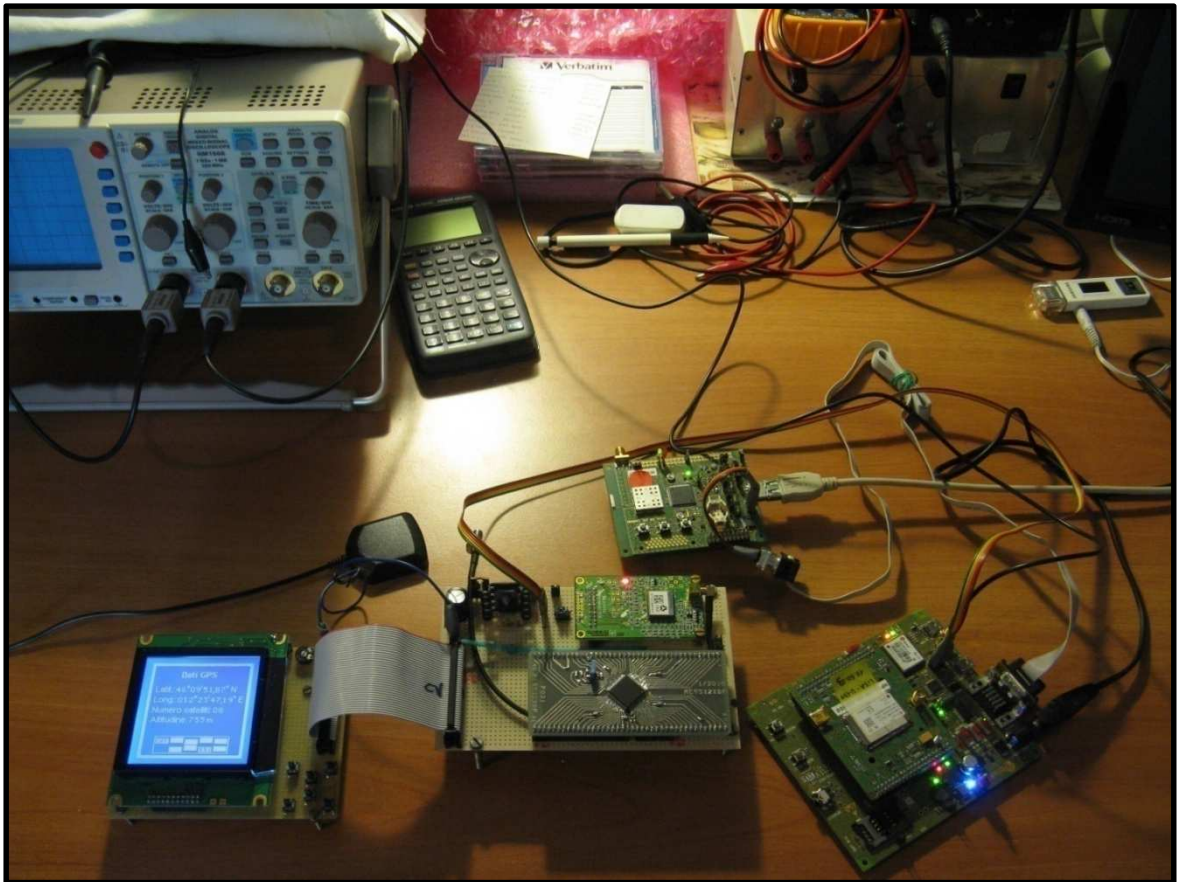


Figura 9.2

## 9.2 Conclusioni finali

Lo svolgimento del progetto si è diviso in varie fasi; anzitutto l'apprendimento della strumentazione utilizzata (hardware e software), successivamente lo studio del problema per cercare la soluzione migliore e infine la realizzazione di un prototipo sperimentale con relativo software. Tutte le fasi sono state portate a termine con successo, perseguendo così il risultato finale di acquisire i dati dal GPS, elaborarli e prepararli per un'eventuale trasmissione all'interno delle reti di sensori.

La prima fase, di apprendimento, è stata sicuramente la più lunga in quanto ho dovuto apprendere il funzionamento del microcontrollore con architettura ARM e contemporaneamente il funzionamento dell'ambiente di sviluppo linux, con il quale non avevo mai lavorato. In modo molto più rapido, grazie ad una certa esperienza che ho maturato nell'ambito dell'utilizzo dei moduli GPS, sono riuscito a portare a termine la fase di

realizzazione del prototipo vale a dire l'impostazione del microcontrollore e l'utilizzo del modulo GPS.

In conclusione ritengo che il mio progetto rappresenti per il futuro, uno sbocco interessante nel campo delle telecomunicazioni, in particolare per applicazioni riguardanti il tracciamento della posizione di un nodo mobile in una rete di sensori outdoor [12].

Per quanto riguarda invece il mio futuro non è da escludere che in vista di una tesi per la laurea magistrale, possa riprendere questo progetto per svilupparlo ulteriormente.

## BIBLIOGRAFIA

- [1] Antonio Caruso, Stefano Chessa, Swades De, Alessandro Urpi, "GPS Free Coordinate and Assignment and Routing Wireless Sensor Networks", *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 150-160 volume 1, 13-17 marzo 2005
- [2] Andrea Zanella, Michele Zorzi, "Reti di sensori: dalla teoria alla pratica", *Notiziario tecnico Telecom Italia*, Anno 15 n. 1, pp. 104- 116, Giugno 2006
- [3] Mauro Bertolini, "Corso di aggiornamento in navigazione moderna", *documento redatto nel corso di aggiornamento svolto a Viareggio su alcuni temi di Navigazione Moderna in data 1997/98*
- [4] Marco Mussini, "Come funziona il ricevitore GPS", *articolo disponibile online all'indirizzo: [http://www.01net.it/itechstudio/articoli/0,1254,4s5009\\_ART\\_72628,00.html](http://www.01net.it/itechstudio/articoli/0,1254,4s5009_ART_72628,00.html)*, acceduto online il giorno 25/08/2012
- [5] Goodrich Michael T. e Roberto Tamassia, "Strutture dati e Algoritmi in java", *ed. Zanichelli*, Maggio 2007
- [6] Datasheet modulo GPS, *disponibile online all'indirizzo: [http://www.u-blox.com/images/downloads/Product\\_Docs/LEA-6\\_DataSheet\\_%28GPS.G6-HW-09004%29.pdf](http://www.u-blox.com/images/downloads/Product_Docs/LEA-6_DataSheet_%28GPS.G6-HW-09004%29.pdf)*, acceduto online il giorno 25/08/2012
- [7] Manuale dei protocolli del modulo GPS, *disponibile online all'indirizzo: [http://www.u-blox.com/images/downloads/Product\\_Docs/u-blox6\\_ReceiverDescriptionProtocolSpec\\_%28GPS.G6-SW-10018%29.pdf](http://www.u-blox.com/images/downloads/Product_Docs/u-blox6_ReceiverDescriptionProtocolSpec_%28GPS.G6-SW-10018%29.pdf)*, acceduto online il giorno 25/08/2012
- [8] Manuale dei comandi AT, *disponibile online all'indirizzo: [http://www.u-blox.com/images/downloads/Product\\_Docs/u-blox\\_AT\\_Commands\\_Manual\\_%28WLS-SW-11000%29.pdf](http://www.u-blox.com/images/downloads/Product_Docs/u-blox_AT_Commands_Manual_%28WLS-SW-11000%29.pdf)*, acceduto online il giorno 25/08/2012

- [9] Manuale utente della scheda EVK-U12, *disponibile online all'indirizzo:*  
[http://www.u-blox.com/images/downloads/Product\\_Docs/EVKU20-EVKU23\\_GettingStarted\\_%283G.G3-EK-11002%29.pdf](http://www.u-blox.com/images/downloads/Product_Docs/EVKU20-EVKU23_GettingStarted_%283G.G3-EK-11002%29.pdf), acceduto online il giorno 25/08/2012
- [10] Manuale utente del microcontrollore LPC1768 della NXP, *disponibile online all'indirizzo:*  
[http://www.nxp.com/documents/user\\_manual/UM10360.pdf](http://www.nxp.com/documents/user_manual/UM10360.pdf), revisione 2.0, agosto 2010, acceduto online il giorno 25/08/2012
- [11] Manuale utente del microcontrollore MC9S12XDP512V2 della Freescale, *disponibile online all'indirizzo:*  
[http://cache.freescale.com/files/microcontrollers/doc/data\\_sheet/MC9S12XDP512RMV2.pdf?fpsp=1&WT\\_TYPE=Data%20Sheets&WT\\_VENDOR=FREESCALE&WT\\_FILE\\_FORMAT=pdf&WT\\_ASSET=Documentation](http://cache.freescale.com/files/microcontrollers/doc/data_sheet/MC9S12XDP512RMV2.pdf?fpsp=1&WT_TYPE=Data%20Sheets&WT_VENDOR=FREESCALE&WT_FILE_FORMAT=pdf&WT_ASSET=Documentation), revisione 2.21, ottobre 2009, acceduto online il giorno 25/08/2012
- [12] Tesi di dottorato, Paolo Sperandio, *“Algoritmi di localizzazione per reti di sensori wireless”*, Università di Roma “Tor Vergata”, Dipartimento di ingegneria elettronica, Marzo 2007



# Ringraziamenti

*Desidero ringraziare il prof. Leonardo Badia, il dott. Riccardo Manfrin e la mia famiglia per il contributo dato alla realizzazione di questa tesi.*

*Padova, 09 settembre 2012*

---

*Tollot Luca*