

Realizzazione di Microterminali Veicolari Terminali per la Piattaforma Informativa Veicolare

Rapporto Tecnico

1 - MOTIVAZIONE

La sperimentazione effettuata con terminali di tipo industriale (Symbol, DataLogic) e con terminali di tipo *consumer* (PALM IV e HP Jornada), hanno messo in evidenza alcuni limiti funzionali che si sono ritenuti cruciali per il dispiegamento della tecnologia del Tracing & Tracking di merci. I limiti possono essere riassunti nei seguenti punti:

Limiti di costi: sia i terminali di tipo *consumer* che i terminali di tipo industriale sono caratterizzati da un costo elevato: quando questi vengono dotati di tutte le periferiche necessarie (GPS, GSM e Bar code) il costo per pezzo (aggiornato all'ultimo trimestre del 2002) è dell'ordine di grandezza dei 2000 €. Questo limite è un forte ostacolo al dispiegamento del sistema presso i piccoli trasportatori o piccoli spedizionieri. Inoltre il fatto che tali dispositivi siano rivendibili per altre applicazioni rende l'eventualità del furto molto realistica specialmente nel caso in cui l'applicazione richieda di portare continuamente i terminali in viaggio.

Limiti di interfaccia: i terminali sono di tipo general purpose e quindi sono dotati di display evoluto (per esempio il PALM è dotato di touch screen a colori) e in alcuni casi anche di tastiera completa (es. Symbol): l'applicazione del tracing & tracking invece richiede la semplice pressione di un tasto per l'invio delle informazioni e un segnale visivo o acustico di riscontro per l'utente: ogni aggiunta a questa interfaccia minimale rischia di generare lentezza o errori nelle operazioni e tende quindi a scoraggiare l'utente.

Limiti di dimensioni: i terminali industriali per proteggere le componenti meccaniche quali il display e per includere la tastiera hanno dimensioni molto superiori a quelle di un dispositivo tascabile (es. telefonino). I terminali di tipo *consumer*, quando vengono dotati di tutte le periferiche necessarie (tramite il "jacket" di espansione) si impugnano con difficoltà e in alcuni casi (es. PALM) i dispositivi non possono essere tutti inseriti contemporaneamente nei loro alloggi PCMCIA per motivi di ingombro reciproco.

Limiti di robustezza: i terminali di tipo *consumer*, più piccoli e maneggevoli di quelli industriali sono caratterizzati da un grado di robustezza decisamente inaccettabile per essere continuamente portati in viaggio, principalmente a causa della fragilità del display la cui rottura compromette l'intero funzionamento

Limiti di consumo: i terminali industriali e i terminali *consumer* sono dotati di eccessiva potenza di calcolo rispetto alla necessità della applicazione ed alimentano dispositivi (es. il display) che non risulta assolutamente necessario.

Le considerazioni sopra riportate hanno motivato la scelta di progettare e realizzare un prototipo di terminale che svolgesse tutte le funzioni richieste, ossia lettura del codice a barre, identificazione della posizione geografica ed invio dei dati via GSM senza incorrere nei limiti sopra indicati.

L'obiettivo, raggiunto con questa sperimentazione, è stato quello di dimostrare che tutti i limiti sopra riscontrati sono effettivamente eliminabili progettando un nuovo oggetto integrato appositamente per l'applicazione. Questo risultato si ritiene di notevole interesse visto il fatto che attualmente la produzione di apparati elettronici ad hoc per singole applicazioni risulta sempre meno conveniente nel settore delle Information & Communication Technologies.

2 - REQUISITI

I requisiti, che hanno guidato la realizzazione e la progettazione del prototipo di terminale sviluppato derivano direttamente dai limiti riscontrati e possono essere riassunti nei seguenti punti.

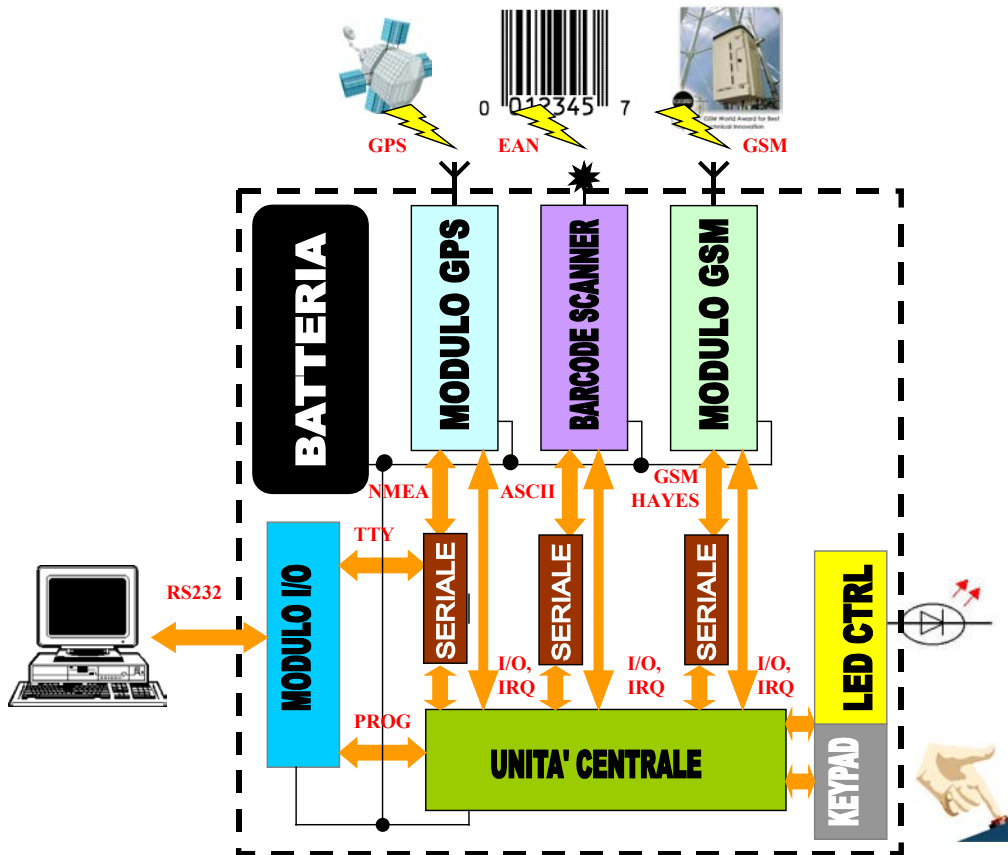
Basso costo: il terminale comprensivo di scanner laser per lettura del codice a barre, di ricevitore GPS di ultima generazione, di ricetrasmittitore GSM e di unità di controllo programmabile ha un prezzo che si aggira intorno ai 500 euro, ossia il 25% del costo dei terminali disponibili sul mercato ed ha la caratteristica di essere un oggetto poco appetibile se estrapolato dal contesto applicativo per cui è stato prodotto e quindi difficilmente soggetto a furti.

Interfaccia Efficiente: il terminale espone all'utente al più tre tasti per e gestisce autonomamente tutti i sottosistemi che incorpora senza chiedere l'intervento dell'utente, compresa l'accensione dell'unità centrale stessa e il suo spegnimento a compito terminato. Un led notifica all'utente il corretto funzionamento dell'apparato. La possibilità di interconnettere il dispositivo ad un terminale seriale lo rende poi facilmente configurabile tramite menù, non esponendo però la sua complessità all'utente.

Dimensioni ridotte: il terminale ha le dimensioni di un oggetto realmente tascabile e non ha elementi che possano essere estratti o smarriti. La sua compattezza è molto maggiore rispetto a quella riscontrata in tutti gli altri apparati disponibili sul mercato.

Robustezza: il terminale deve la sua robustezza alla sua compattezza, alla sua leggerezza, alla mancanza di display e al fatto di essere chiuso in un involucro di classe IP74, come i terminali di tipo industriale. La sperimentazione ha messo inevitabilmente in evidenza la sua robustezza.

Consumo: Il fatto di poter essere programmato con un software minimale perchè strettamente ritagliato sulla applicazione e il fatto di poter controllare lo spegnimento e l'accensione automatica che rendono il sistema dotato di notevole autonomia.



3 - ARCHITETTURA DEL TERMINALE:

Il sistema è composto da una unità centrale che si connette tramite linee seriali e linee di I/O e IRQ ai sottosistemi GSM, GPS e al lettore di codice a barre e si connette con linee di I/O e di IRQ al tastierino e all'indicatore luminoso come indicato in figura. Inoltre una quarta linea seriale consente di configurare e riprogrammare lo strumento.

L'unità centrale comunica con il modulo GSM tramite il protocollo Hayes GSM (Extended AT-commands), e consente di attivare e disattivare il ricetrasmittitore, controllare il livello di ricezione, scatenare l'invio di messaggi di tipo SMS o e richiedere la lettura degli SMS dalla memoria della SIM. L'unità centrale è in grado anche di accendere e spegnere il modulo GSM attraverso le linee di I/O e di ricevere notifiche dal modulo attraverso la linea di IRQ.

L'unità centrale comunica con il modulo GPS tramite il protocollo NMEA in ASCII ed effettua la lettura della posizione, dell'ora UTC, della velocità e della direzione ricavate dal sistema satellitare ed altri dati di controllo quali il numero di satelliti visibili e il rapporto segnale-disturbo presente. L'unità centrale è in grado anche di accendere e spegnere il modulo GSM attraverso le linee di I/O e di ricevere notifiche dal modulo attraverso la linea di IRQ.

L'unità centrale comunica con il modulo di lettura del codice a barre tramite lettura di caratteri ASCII che riportano direttamente il codice letto.

Il keypad attraverso segnala la pressione di un qualunque tasto al processore attraverso IRQ e il processore identifica i tasti premuti attraverso opportune porte di

I/O. L'avvisatore luminoso è pilotato attraverso porte di I/O connesse attraverso un circuito di switch.

Un terminale seriale può connettersi al sistema attraverso una linea seriale sia per interagire con il software del sistema sia per aggiornarlo.

Il programma risiede in una memoria flash e i dati di configurazione sono messi in una memoria di tipo EEPROM.

La batteria alimenta ogni modulo presente nel sistema.

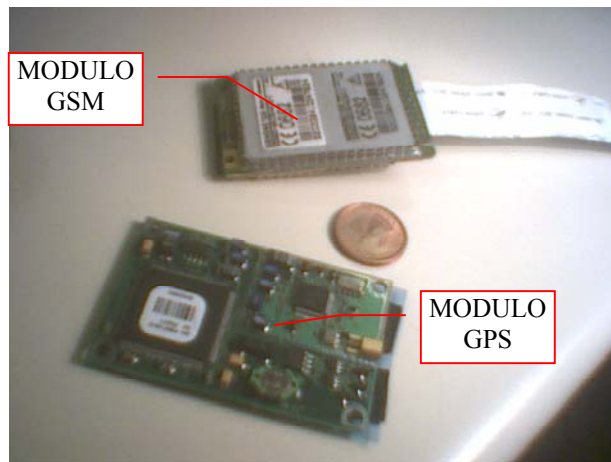
4 - IMPLEMENTAZIONE HARDWARE DEL TERMINALE

Elemento Architettureale	Prodotto Utilizzato	Note Tecniche
Unità Centrale	HITACHI H8S-F2134	20 MHz 60 linee di I/O Data BUS 8 bit 32 bit ALU 16 bit data path interno
Porte Seriali	-	Incorporata nella CPU: 3 seriali
Memoria Flash	-	Incorporata nella CPU 128 K x 8 bit
Modulo I/O	-	Incorporato nella CPU previa trasduzione da RS232 a TTL (con trasduttore MAX3232)
Memoria RAM	-	Incorporata nella CPU: 4K x 8 bit
Modulo GPS	Motorola Oncore M12	12 Satelliti Paralleli Mantenimento informazioni dell'ultima acquisizione
Modulo GSM	Siemens TC35	GSM dual-band Controllo di Carica della batteria
Letto di BARCODE	PSC LM500	Scan Engine laser Classe 2 46 scansioni/secondo
Batteria	Siemens	Li-Ion 750 mAh
Keypad		Flat Adesiva con feedback tattile

Tabella 1 – Principali componenti dei prodotti utilizzati

L'implementazione del sistema è avvenuta seguendo le seguenti fasi:

Identificazione dei moduli : in questa fase sono stati identificati i componenti per l'invio di messaggi SMS, per la lettura del codice a barre e per la lettura del GSM. I prodotti sono stati scelti in modo tale da rispettare il requisito di basso costo e allo stesso tempo per dotare il sistema della capacità di ricezione GSM e GPS massima e di lettore a codice a massima robustezza. Il microprocessore per il controllo del sistema è stato scelto dopo aver stimato le esigenze di calcolo per la gestione dei moduli identificati. Le componenti utilizzate per il prototipo sono riportate in Tab.1. In Fig. 2 si possono vedere i moduli GSM e GPS acquisiti ed incorporati nel sistema.



Progettazione della scheda madre del sistema: la scheda madre è l'elemento chiave del sistema al quale sono interconnesse tutte le componenti e sulla quale è montata l'unità centrale. Sulla base dello schema previsto dalla architettura si è proceduto alla stesura del circuito utilizzando l'ambiente di sviluppo ORCAD. Le parti fondamentali del circuito sono riportate nelle schermate di Fig. 2 e Fig. 3.

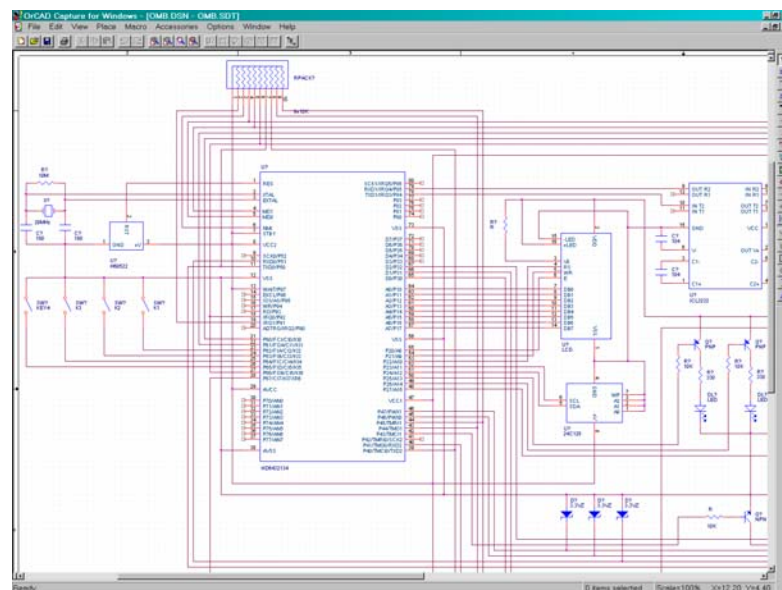


Figura 1 – Particolare del progetto della scheda madre: connessioni alla CPU

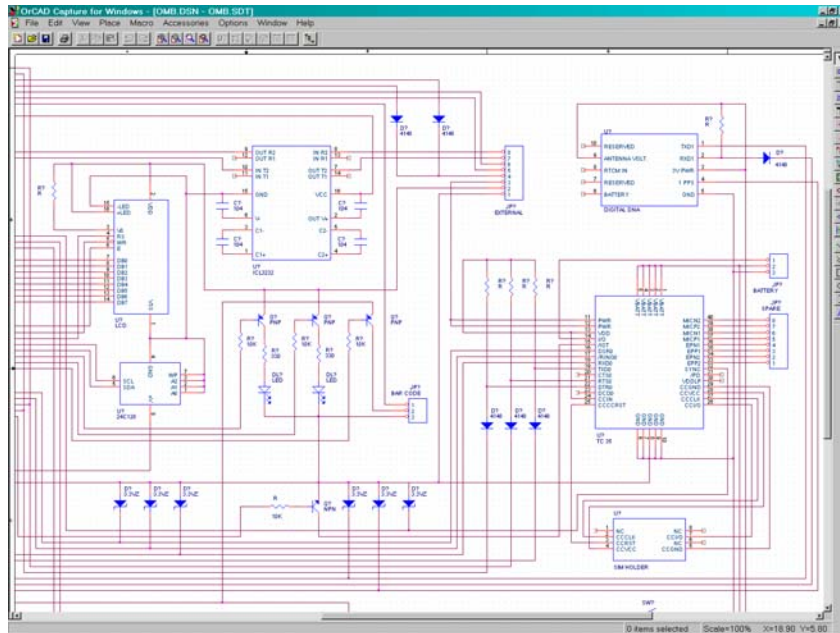


Figura 2 – Particolare del progetto della scheda madre: connessioni ai sottosistemi

Realizzazione della scheda madre: dopo aver progettato il circuito si è proceduti alla sua realizzazione avvalendosi di un produttore di circuiti stampati. Anche il montaggio dei componenti elettronici sulla scheda madre è stato effettuato con saldatrici automatiche visto il tipo di saldature necessarie.

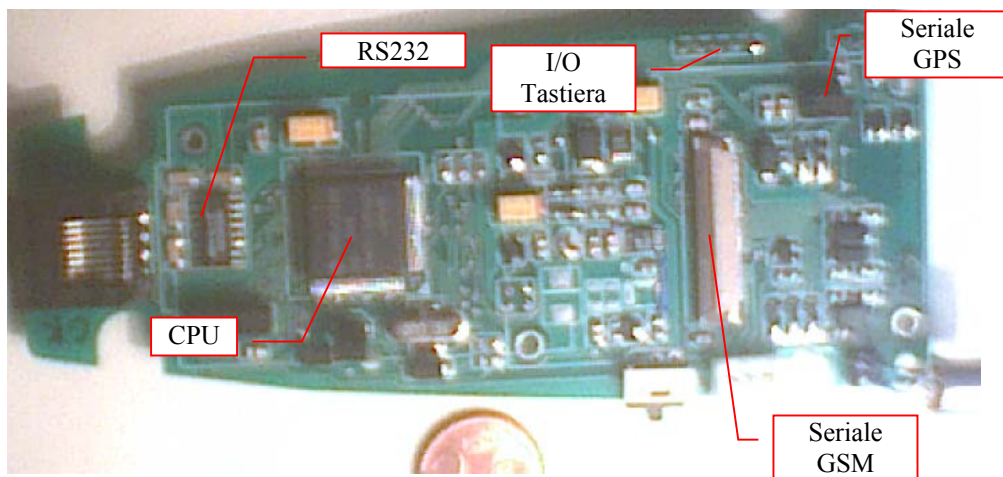


Figura 3 – Immagine della scheda madre del prototipo e indicazioni delle sue principali componenti

Montaggio del sistema: dopo aver realizzato il circuito si è proceduti al suo montaggio in un contenitore apposito ad elevata protezione d'urto. Il contenitore è stato scelto in modo che fosse facilmente impugnabile ed avesse il volume minimo per contenere tutte le componenti del sistema. L'aspetto più critico di questa fase è stata la schermatura delle componenti per ridurre al minimo le interferenze elettromagnetiche alle quali il modulo GPS è molto sensibile.

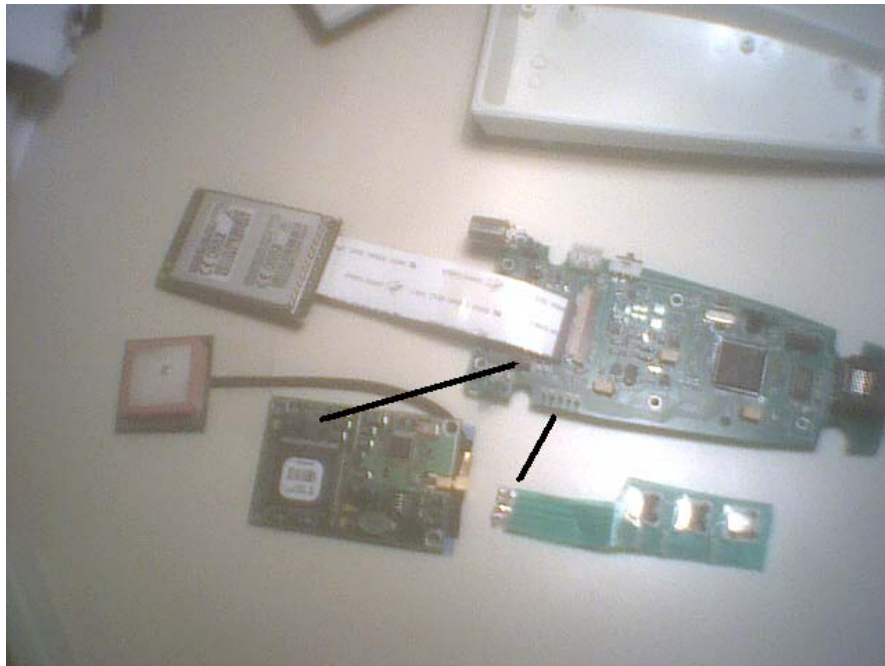


Figura 4 – Sistema smontato e contenitore



Figura 5 – Montaggio del sistema



Figura 6 – Sistema montato e sua impugnabilità

5 - SVILUPPO SOFTWARE

Il software del sistema è stato sviluppato in linguaggio C ed è stato compilato con l'ambiente di sviluppo IAR con il modulo linker specifico per l'architettura Hitachi S2000.

Il programma è costituito da un ciclo infinito che effettua un polling sui possibili eventi (es. pressione di un tasto, invio di carattere dal modulo RS232, etc.). Gli eventi possono essere interni (es. lettura di variabili di programma) od esterni (lettura di linee di I/O).

Altri eventi esterni sono invece gestiti asincronamente attraverso appositi *interrupt handlers*. Per tali eventi si è utilizzato il supporto del compilatore per la definizione di interrupt handlers chiamati automaticamente nel momento in cui si verifica un interrupt.

Le azioni principali svolte dal software sono

- **Comunicazione con terminale esterno** (se collegato): questa azione consiste nell'interpretare i caratteri ricevuti dalla seriale RS232 nel modulo main. La medesima seriale è utilizzata per comunicare con l'esterno e per riprogrammare il sistema. Il significato di tali caratteri cambia quindi in funzione di ponticelli nel connettore di connessione al terminale esterno, così da identificare il device connesso alla seriale esterna.
- **Gestione dei moduli (output)** comprende le funzioni di interfaccia gestite in interrupt con buffers di dati sia in ingresso che in uscita. Le funzioni che intervengono direttamente sui moduli periferici (Gsm, Gps, Barcode scanner) cambiano lo stato dei dispositivi ed aggiornano lo stato nelle variabili comuni di stato che possono essere lette in ogni momento da qualsiasi funzione.

- **Gestione dei moduli (input)** le funzioni interrupt dei timers incorporano il codice per la gestione dei lampeggi del Led, per la gestione del segnale del GSM e per le funzioni da eseguire a tempo.

5.1 Descrizione del software di base

Il software di base si compone di

- uno strato di funzioni a basso livello per gestire le funzionalità elementari quali la scrittura nella memoria EEPROM, gestione del Display LCD di debugging, etc.), di funzione di comunicazione con i moduli. I prototipi commentati delle principali funzioni che compongono questo strato sono riportate in Tab. 2
- uno strato di funzioni per la comunicazione con i moduli ed in particolare le gestioni delle seriali interne e delle linee di IRQ. I prototipi commentati delle principali funzioni che compongono questo strato sono riportate in Tab.3
- uno strato di funzioni a livello intermedio per la gestione delle stringhe. I prototipi commentati delle principali funzioni che compongono questo strato sono riportate in Tab.4
- uno strato di funzioni ad alto livello per la gestione dei dispositivi (es. mandare un SMS). I prototipi commentati delle principali funzioni che compongono questo strato sono riportate in Tab.5

Prototipo funzione	Commento
void byteout(unsigned char dato);	invia un byte al chip di memoria EEPROM
unsigned char bytein(void);	legge un byte dal chip di memoria EEPROM
void leds(char);	accende / spegne i leds di segnalazione
void init_lcd(void);	inizializza display LCD (il display è aggiunti opzionalmente per debugging sul campo)
void curs_lcd(unsigned char l_add);	posiziona cursore al valore assoluto di memoria del display
void char_lcd(unsigned char clcd);	manda un char al display
void pos_lcd(unsigned char n_row,unsigned char n_col);	posiziona per riga e colonna
void curs_lcd(unsigned char l_add);	posiziona cursore al valore assoluto di memoria del display
void scroll_lcd(void);	scorre display in verticale
void spr_lcd(char *sptr_lcd);	invia una stringa stringa su LCD

Tabella 2 – Funzioni a basso livello

Prototipo funzione	Commento
void Init_Serials(void);	init porte seriali
void rx0(void);	interrupt RX seriale GSM
interrupt [SCI_ERI0] void rx0e(void);	interrupt errore seriale GSM
interrupt [SCI_TXI0] void tx0(void);	interrupt TX seriale GSM
interrupt [SCI_TEI0] void te0(void);	interrupt END TX seriale GSM
void Init_Serials(void);	init porte seriali
interrupt [SCI_RXI0] void rx0(void);	interrupt RX seriale GSM

interrupt [SCI_ERI0] void rx0e(void);	interrupt errore seriale GSM
interrupt [SCI_TXI0] void tx0(void);	interrupt TX seriale GSM
interrupt [SCI_TEI0] void te0(void);	interrupt END TX seriale GSM
interrupt [SCI_RXI1] void rx1(void);	interrupt RX Seriale Esterna
interrupt [SCI_ERI1] void rx1e(void);	interrupt errore Seriale Esterna
interrupt [SCI_TXI1] void tx1(void);	interrupt TX Seriale Esterna
interrupt [SCI_TEI1] void te1(void);	interrupt END TX Seriale Esterna
interrupt [SCI_RXI2] void rx2(void);	interrupt RX seriale GPS
interrupt [SCI_ERI2] void rx2e(void);	interrupt errore seriale GPS
interrupt [SCI_TXI2] void tx2(void);	interrupt TX seriale GPS
interrupt [SCI_TEI2] void te2(void);	interrupt END TX seriale GPS
void Init_Timers(void);	inizializzazione timers
interrupt [TM_OVI0] void tempo1(void);	interrupt timer 1.
interrupt [TM_OVI1] void tempo2(void);	interrupt timer 2.
interrupt [IRQ_6] void tastpres(void);	interrupt di pressione tasto
unsigned char getcrx0(void);	legge un carattere dal buffer RX della seriale 0
unsigned char getcrx1(void);	legge un carattere dal buffer RX della seriale 1
unsigned char getcrx2(void);	legge un carattere dal buffer RX della seriale 2
void txc0(unsigned char);	mette un carattere nel buffer TX della seriale 0
void txc1(unsigned char);	mette un carattere nel buffer TX della seriale 1
void txc2(unsigned char);	mette un carattere nel buffer TX della seriale 2
void txcs0(unsigned char*);	mette una stringa nel buffer TX della seriale 0
void txcs1(unsigned char*);	mette una stringa nel buffer TX della seriale 1
void txcs2(unsigned char*);	mette una stringa nel buffer TX della seriale 2
unsigned char gettasto(void);	legge il valore del tasto premuto

Tabella 3 – Funzioni di gestione moduli in input e output

Prototipo funzione	Commento
void e2write(unsigned char,unsigned char,unsigned char);	scrive un byte in EEPROM
void e2swrite(unsigned char ,unsigned char ,unsigned char *);	scrive una stringa max 64 bytes fino al terminatore NULL nella E2prom
unsigned char e2read(unsigned char,unsigned char);	legge un unchar da EEPROM
char datel(unsigned char *cosa,unsigned char perquanto);	attende da colloquio GSM la stringa puntata da *cosa per il tempo indicato in perquanto (1 = 1 sec. circa) rende 1 se ricevuto , 0 se non ricevuto nel tempo previsto
void codabar(void)	attiva il lettore di barcode, legge un codice fino al completamento oppure fino al timeout (600 mSec) nel buffer bufco

Tabella 4 – Funzioni di livello intermedio

Prototipo funzione	Commento
void accspegsm(char);	accende o spegne Gsm secondo char (0=spigne, 1 accende) con invio e controllo del Pin code e dell'effettiva connessione al gestore aggiorna lo stato comune GSM
void inviosms(void);	invia sms al n. telefono in Istr_ing
void accspegps(char);	accende e spegne GPS
void decodGPS(void);	aggiorna lo stato GPS e' chiamata quando l'interrupt segnala la presenza nel buffer della seriale di una stringa completa NMEA
void decodeSMS(void);	decodifica un SMS ricevuto chiamata quando viene rilevato un nuovo SMS ricevuto

Tabella 5 – Funzioni di alto livello

5.2 Descrizione del software applicativo

La programmabilità del dispositivo ha fatto sì che il software applicativo potesse essere sviluppato in diverse versioni a seconda dei requisiti del dominio di utilizzo. In particolare si sono identificati due domini applicativi:

- Tracing & Tracking del singolo pacchetto: per questa applicazione il terminale deve avere massima autonomia poichè dev'essere asportato dal camion: in questo caso la pressione di un tasto (l'unico esposto all'utente) causa la lettura di un codice a barre e l'invio del SMS contenente i dati del codice a barre e le coordinate geografiche. Se i dati provenienti dal GPS non sono sufficientemente recenti l'invio viene differito spegnendo il dispositivo e riaccendendolo periodicamente in attesa di disporre della copertura GPS. Nel momento in cui la copertura viene nuovamente rilevata gli SMS differiti vengono finalmente inviati.
- Tracing & Tracking di un carico: per questa applicazione il terminale non deve necessariamente abbandonare il veicolo e quindi può rimanere continuamente collegato all'alimentazione. In questo caso il GPS può rimanere costantemente attivo. Nel momento in cui il veicolo ha effettuato l'intero scarico (es. un container) il conducente preme il tasto senza leggere alcun codice a barre e vengono inviate le coordinate geografiche che si riferiscono all'ultima acquisizione GPS.