10 Realizzazione Firmware

Il firmware del microcontrollore PIC deve provvedere ad una serie di operazioni:

1. Inizializzare la Card
2. Acquisire i dati dai due accelerometri
3. Gestire le temporizzazioni
4. Scrivere i dati sul file

Vediamo ora nel dettaglio come queste vengono effettuate.

10.1 Invio di comandi alla SDcard

Il Proton+ gestisce la comunicazione in modalità SPI; innanzitutto si dichiarano quali pin del PIC assolveranno la funzione di:

- **CS**: Chip Select
- **CLK**: Clock
- **SO**, **SI**: rispettivamente DataOut e DataIn del dispositivo seriale

CS, CLK, SI, SO sono, per il compilatore, delle parole chiave e non dei semplici identificatori. I primi tre devono essere settati come output e solo l'ultimo come input.
Il Proton+ utilizza per la comunicazione seriale le funzioni Shout e Shin.

10.1.1 Funzione Shout

Invia dati ad un dispositivo sincrono seriale; la sua sintassi è la seguente:

shout SI, CLK, MSBFIRST, [*dato1*\bit,*dato2*\bit,…]

Il PIC, per ciascun colpo di clock che genererà, invierà sul pin SI, (collegato al CMD della scheda) i dati inclusi nella parentesi quadra a partire dal bit più significativo a quello meno significativo (MSBFIRST). Dato1, dato2, … possono essere numeri o variabili.
bit è un parametro opzionale posto dopo *datox* che stabilisce quanti bit di *datox* devono essere inviati sul pin **SI**. Se omesso, il valore di default è 8.

Esempio:
SHOUT SI, CLK, MSBFIRST, [245\4]: invia 4 cicli di clock alla scheda ad ognuno dei quali corrisponde l'invio di uno dei 4 byte della parte meno significativa del numero 245 (in binario '1111<u>0101</u>'), nell'ordine 0, 1, 0, 1.

10.1.2 Funzione Shin

Analoga alla sua duale, riceve dati da un dispositivo asincrono seriale; la sua sintassi è la seguente:

shin SO,CLK,MSBPRE,[*dato1*\bit,*dato2*\bit,…]

ad ogni colpo di clock il PIC va a leggere il pin che corrisponde a SO (collegato alla linea dati dell'SDcard) e "immagazzina" il bit così acquisito a partire da *dato1*, finché non riempie tutte le variabili. MSBPRE significa che, come sempre, i bit ricevuti vanno dal più significativo al meno significativo e che vanno letti PRIMA di inviare il clock. Dato1, dato2, … possono essere solo variabili.

Esempio:
SHIN SO,CLK,MSBPRE,[var1\8,var2\16]: invia 24 fronti di clock e acquisisce altrettanti bit: i primi 8 verranno salvati in var1, i rimanenti in var2.

10.2 Acquisizione dei dati dall'accelerometro

Per poter determinare l'accelerazione in termini di *g* è necessario ricavare il *duty cicle* dell'onda quadra con la quale esce l'accelerometro (v. sezione *Accelerometri*). Per fare questo misuro per quanto tempo rimane alto il segnale tramite la funzione PULSIN, la cui sintassi è la seguente:

Variabile = PULSIN *pin*, *stato*

Dove:

- *Variabile* è dove verrà memorizzato il dato acquisito
- *Pin* è il piedino del PIC collegato all'accelerometro
- *Stato* può assumere il valore '0' o '1': dice quale fronte si deve attendere prima di iniziare la misurazione. Esempio: '1' attente che il segnale su *pin* assuma valore logico '1' e misura quanto questo duri prima che arrivi un fronte di discesa.

Variabile contiene il numero di *unità* misurate dal Pulsin: può essere di dimensione byte o word:

variabile	**Num. di bit**	**Num. di unità**
Byte	8	1 – 255
Word	16	1 - 65535

Il valore di *unità* dipende dalla velocità del clock inviato al PIC:

frequenza del clock	**Durata di un'unità**
4 MHz	10 μs
20 MHz	2 μs

Il tempo T_1 in cui rimane alto il segnale dell'accelerometro, è dato dalla formula inversa riportata nella sezione *Accelerometri*:

$$T_1 = (\frac{a(g)}{8} + 0{,}5) * T_2$$

Ricordando che il Memsic può variare tra –2g e +2g si evince che:

$$2{,}5\,\text{ms} \leq T_1 \leq 7{,}5\,\text{ms}$$

Utilizzando un clock a 4 MHz, servono quindi almeno 750 unità (corrispondenti a 7500 μs), che si potrebbero rappresentare con 10 bit: si è costretti, per soli 2 bit, ad utilizzare una variabile di tipo word: questo significa che in memoria si hanno 18 bit superflui per ogni acquisizione.
La grande disponibilità di memoria che ci fornisce la memory card rende però ininfluente questo problema.

10.3 Frequenza di acquisizione

Si è detto più volte che la frequenza alla quale si vuole acquisire è 12Hz: questo significa che la durata della funzione di acquisizione deve essere di 1/12Hz = 83,33 ms.
Le operazioni effettuate in questa funzione sono le seguenti:

Operazione	**Durata**
3 Pulsin (uno per ogni asse)	30 ms
Scrittura di 6 byte sulla card	4 ms
'toggle' di un piedino di controllo[1]	1 ms

Per far tornare i conti è necessario un delay di circa 48 ms. Questo ha creato un problema inaspettato di non facile soluzione: con 48 ms la durata di una acquisizione era di circa 81 ms e variando il delay si verificava la seguente situazione:

Delay	**Durata acquisizione (t)**	**Frequenza (1/t)**
47 ms	80,8 ms	12,375 Hz
48 ms	81,20 ms	12,315 Hz

[1] questa operazione è stata importante in ambito di debugging in quanto, al piedino in questione, veniva collegato un led che così lampeggiava ad ogni scrittura. Terminato il prototipo non è stato più possibile togliere questa istruzione perché si rischiava di cambiare la durata del periodo di acquisizione senza aver modo di misurarlo.

Delay	Durata acquisizione (t)	Frequenza (1/t)
49 ms	81,20 ms	12,315 Hz
50 ms	84 ms	11,905 Hz
51 ms	84 ms	11,905 Hz

Questo è accaduto perché la funzione delay opera, a livello Assembly, mettendo una sequenza di NOP (No OPeration) di durata prestabilita e stabile. Il compilatore, però, utilizza degli algoritmi di ottimizzazione del firmware che ne riduce le dimensioni cambiando l'ordine delle istruzioni e causando quindi un'alterazione delle temporizzazioni.

E' stato impossibile risolvere il problema, in quanto si sarebbe dovuto metter mano al codice macchina (operazione lunga e impegnativa); si è preferito cambiare la frequenza di acquisizione, fissandola così a 12,315 Hz.

10.4 Ulteriori temporizzazioni

Come detto nella sezione dedicata alla SDcard la scrittura avviene a blocchi di 512 byte: scrivendo 6 byte alla volta (2 byte per ogni asse) si eseguono 85 scritture, per un totale di 510 byte. Gli ultimi 2 byte vengono riempiti con il valore 0000(hex), dopodiché si chiude il blocco e si ricomincia con un altro:

Operazione	Durata
85 scritture (81,20ms l'una)	6,9020 sec
Scrittura di 2 byte	2,6 msec
Chiusura di un blocco e riapertura di un altro	5,0 msec
Totale	**6,9096 sec**

Questo significa che, in media, perdo un'acquisizione ogni 10,7 blocchi circa, ovvero ogni 73,8 secondi.

Tuttavia questo non risulta essere molto preoccupante in quanto, nell'arco di 24 ore, perdo circa 95 secondi di acquisizioni che, ai nostri fini, non costituiscono un problema.

10.5 Creazione e gestione del File

Come già detto l'actigrafo realizzato effettua la scrittura dei dati acquisiti in maniera seriale attraverso la modalità SPI. Lo scaricamento dei dati viene effettuato con un comune lettore di Card, collegabile al PC tramite porta USB, il quale utilizza come protocollo di comunicazione l'SDcard Bus Protocoll per poter trasferire dati alla velocità massima (*v. specifiche SDcard*).

Naturalmente i dati sull'SDcard devono essere sotto forma di file.

Per poter fare ciò è stato necessario creare una File Allocation Table (FAT) sulla SDcard, semplicemente formattando la stessa in ambiente Microsoft Windows XP. In seguito utilizzando **WinHex 11.15** della X-ways Software si è creato un file della dimensione desiderata e si è andati a vedere la locazione di memoria corrispondente all'inizio di tale file: da questa locazione si inizieranno a scrivere i dati acquisiti.

L'indirizzo dell'*header* di un file appena creato, per una carta da 128Mb formattata, è statico e corrisponde a **000F2200h;** il campo dati inizia alla locazione **000F2400h** e per default tutti i suoi byte sono posti a 00h.

Dopo ogni acquisizione si copia il file sul PC con il programma di elaborazione dei dati e con WinHex si cancella il contenuto del file sulla memory card la quale, a questo punto, è pronta per una nuova acquisizione.

La dimensione del file è strettamente legata al periodo di acquisizione che si desidera effettuare:

Periodo di acquisizione	**Dimensione del file**
12 h	3,05 MB (3124 KB)
24 h	6,11 MB (6252 KB)
2 giorni	12,21 MB (12504 KB)
1 settimana	42,74 MB (43765 KB)

Attualmente la dimensione del file è 12,21 MB, in quanto siamo vincolati ad acquisire al massimo per 28 ore, a causa delle batterie utilizzate.