

Firmware per matrice sensorizzata e accelerometri, descrizione architettura e sensori

Versione: 3_SERIALE_TOLIFE

Tipologia comunicazione: Seriale

Scheda: Arduino DUE

Freq. Camp. teorica matrice sensorizzata: 8 Hz

Freq. Camp. teorica accelerometri: 128 Hz

Filtraggio on-board: nessuno (per acquisizione di dati grezzi)

Valori inviati: costanti per debug

La scheda scelta, Arduino DUE, gestisce la comunicazione seriale e la lettura dei sensori che costituiscono il **Physiological Data Collector** (PDC). In particolare il PDC è una parte costituente dell’architettura del sistema per l’analisi del sonno all’interno del progetto Europeo TOLIFE. In Figura 1 un’immagine esplicativa dell’architettura scelta.

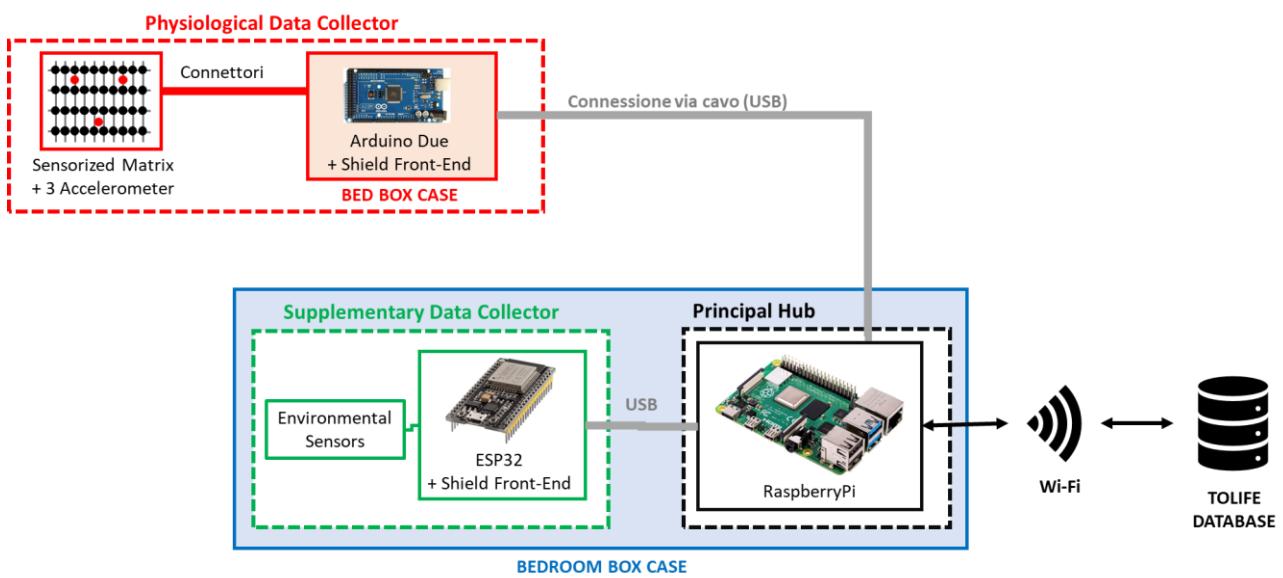


Figura 1: Architettura sistema analisi del sonno.

I sensori che costituiscono il PDC sono una matrice sensorizzata e tre accelerometri. In particolare, la **matrice sensorizzata** è composta a sua volta da **40 elementi sensibili alla pressione**, ed è un esempio di sensoristica distribuita su superfici. La matrice è di piccole dimensioni 40 x 50 cm, e sarà posizionata all’altezza del torace dei pazienti. La matrice, infatti, sarà locata sul materasso dei soggetti. I singoli elementi sensibili vengono letti con la tecnica del *textile pressure mapping* nell’ordine indicato in Figura 2. Dalla matrice sensorizzata è possibile ricavare informazioni quali:

- Posizione del soggetto sul letto,

- Movimento,
- Attività respiratoria,
- Eventi respiratori anomali quali colpi di tosse, apnee notturne ecc..

I singoli elementi sensibili della matrice sensorizzata sono inviati come valori campionati dall'ADC della scheda di acquisizione. Essendo l'ADC on-board di 12 bit, ogni singolo dato avrà quindi dimensione di 2 byte. L'intera matrice acquisita avrà dimensione di 80 byte. Non è necessario eseguire ulteriori trasformazioni del dato in quanto all'interno dello studio siamo interessati alle variazioni del segnale e non ai valori assoluti. Idealmente i dati della matrice sensorizzata sono campionati ad 8 Hz.

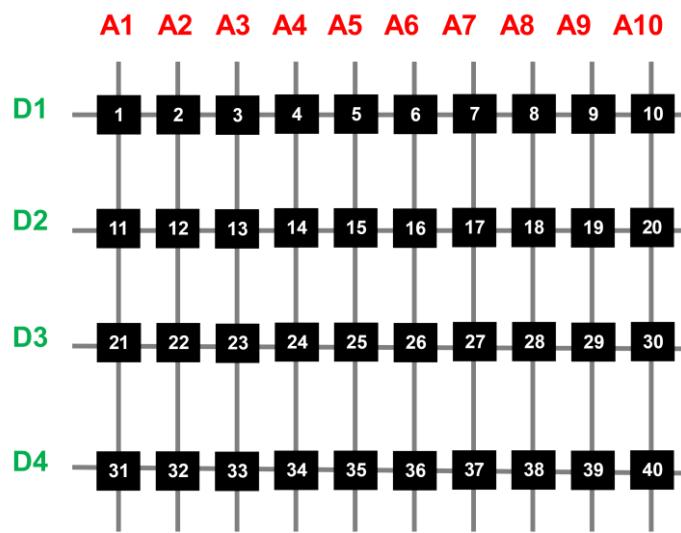


Figura 2: Ordinamento lettura punti sensibili.

I **tre accelerometri**, invece, vengono utilizzati per eseguire analisi **BCG** ovvero analisi ballistocardiografica. L'accelerometro scelto è l'**ADXL362** della Analog Device che fornisce dati digitali. Il singolo accelerometro preleva accelerazioni nel range [-2;+2] g lungo l'asse X, Y, Z e aggiunge anche un'informazione sulla temperatura T. Gli accelerometri sono posizionati sulla matrice sensorizzata in modo da massimizzare l'informazione acquisita . Gli accelerometri forniscono valori digitali, anch'essi a 12 bit, quindi ogni valore avrà dimensione di 2 byte (sia per le accelerazioni lungo X, Y, Z che per la T). Idealmente i valori degli accelerometri sono campionati a 128 Hz.

L'architettura della comunicazione è di tipo **master-slave**, in cui il master è rappresentato dalla scheda RaspberryPi e lo slave dalla scheda Arduino DUE. L'invio e la ricezione dei dati avviene tramite seriale con invio di *byte*. Di seguito le **impostazioni della COM**:

- 115200 bps baudrate
- 8 data bits
- 1 stop bit
- No parity bit
- Big-endian bit order

Il **protocollo** di comunicazione di **ricezione** per lo slave si articola come segue:

- ‘s1’ o ‘0x7331’ : permette di avviare l’invio dei dati,
- ‘s0’ o ‘0x7330’ : permette di stoppare l’invio dei dati.
- ‘r’ o ‘0x72’: permette di resettare via software la scheda.

Lo slave (Arduino DUE), a seguito della ricezione della stringa ‘s1’ o la sequenza di byte ‘0x7331’ inizierà l’invio dei dati. Allo stesso modo, a seguito della ricezione della stringa ‘s0’ o la sequenza di byte ‘0x7330’ verrà stoppata l’acquisizione ed invio dei dati. A seguito della ricezione della stringa ‘r’ o la sequenza di bit ‘0x72’ la scheda esegue il reset via software, la chiamata è asincrona e può avvenire anche durante la ricezione dei dati.

I **dati inviati** dallo slave al master seguono l’ordinamento dei byte di tipo big-endian quindi l’MSB precede sempre l’LSB.

I dati sono inviati come byte secondo la seguente struttura [**TAG | PAYLOAD | CRC**], vedi Tabella 1.

Tabella 1

TAG	PAYLOAD	CRC
1 byte	574 byte	2 byte

Dove:

- **TAG:** byte che indica la tipologia e le caratteristiche del payload,
 - o **0x00:** indica che il payload è completo di tutti i dati, matrice sensorizzata e accelerometri stanno funzionando correttamente,
 - o valori differenti del TAG indicano che ci sono dei problemi con i sensori e che i dati ad essi associati nel payload saranno nulli, vedi Tabella 2 per dettaglio degli errori. Si osserva che nella versione v3 i messaggi di errore ancora NON sono stati implementati, dalla v3.1 del firmware sono presenti.
- **PAYLOAD:** vettore di byte che contiene l’argomento del messaggio, ovvero indice del campione, dati della matrice sensorizzata e degli accelerometri;
- **CRC:** 2 byte che indicano il valore unico di CRC.

Tabella 2

TAG	TAG_NAME	TIPOLOGIA ERRORE
0x01	ERROR_ACC1	ACC1 non funzionante
0x02	ERROR_ACC2	ACC2 non funzionante

0x03	ERROR_ACC3	ACC3 non funzionante
0x04	ERROR_ACC1_2	ACC1, ACC2 non funzionanti
0x05	ERROR_ACC1_3	ACC1, ACC3 non funzionanti
0x06	ERROR_ACC2_3	ACC2, ACC3 non funzionanti
0x07	ERROR_ACC1_2_3	ACC1, ACC2, ACC3 non funzionanti

In particolare il campo PAYLOAD è strutturato come segue [**INDICE** | **MATRICE** | **ACC**], vedi Tabella 2.

Tabella 2

INDICE	MATRICE	ACC
4 byte	84 byte	486 byte

Dove:

- **INDICE**: è una parola di 4 byte che indica l'indice del campione inviato in seriale contenente i valori della matrice e degli accelerometri.
- **MATRICE**: è un vettore di 84 byte che indica i valori acquisiti della matrice sensorizzata. Come specificato, i punti sensibili associati sono 40 (M_1, ... , M_40) acquisiti con un ADC di 12 bit, ogni singolo valore quindi è una parola da 2 byte. Inoltre viene aggiunto anche il valore del timestamp (M_t) nel quale è acquisita la matrice. Vedi Tabella 3 per maggiori informazioni.
- **ACC**: è un vettore di 576 byte che indica i valori acquisiti dai 3 accelerometri; dal momento in cui gli accelerometri sono campionati a 128 Hz si accumulano i valori acquisiti in un buffer e si inviano successivamente alla frequenza di campionamento della matrice sensorizzata (ovvero a 8 Hz). Il vettore ACC quindi sarà costituito dai valori dei 3 accelerometri (ACC1, ACC2, ACC3), che a loro volta saranno costituiti dai 16 campionamenti (per esempio per ACC1 si ha ACC1_1, ACC1_2, ..., ACC1_15, ACC1_16). Ogni campionamento degli accelerometri è costituito in ordine: dal timestamp associato a quella particolare acquisizione, dai valori delle accelerazioni in X, Y, Z (per esempio per ACC1 nel primo campione acquisito si ha ACC1_1_t, ACC1_1_X, ACC1_1_Y, ACC1_1_Z). Infine, si aggiunge al valore degli accelerometri il valore della temperatura T calcolato come media delle 16 acquisizioni effettuate a 128 Hz (per esempio per ACC1 si avrà ACC1_1, ACC1_2, ... ACC1_15, ACC1_16, ACC1_T).

Più in dettaglio il messaggio inviato dalla scheda Arduino DUE è strutturato come nelle Tabelle 3, 4, 5. Si osserva che i valori HEX indicati specificano i bit di interesse per ogni elemento del messaggio (0 indica che quei bit saranno sempre nulli, F indica che possono modificarsi con le letture).

Tabella 3

INDICE 4 byte				MATRICE 84 byte								
MSB		LSB		M_t 4 byte				M_1 2 byte		...	M_40 2 byte	
				MSB		LSB		MSB	LSB		MSB	LSB
FF	FF	FF	FF	FF	FF	FF	FF	OF	FF	...	OF	FF

Tabella 4

ACC 486 byte											
ACC1 162 byte				ACC2 162 byte				ACC3 162 byte			
ACC1_1 10 byte	..	ACC1_16 10 byte	ACC1_T 2 byte	ACC2_1 10 byte	..	ACC2_16 10 byte	ACC2_T 2 byte	ACC3_1 10 byte	..	ACC3_16 10 byte	ACC3_T 2 byte

Ogni campione ACCx_y con x = 1:3 e y = 1:16 degli accelerometri è composto come segue (indicato come esempio per ACC1_1).

Tabella 5

ACC1_1 10 byte									
ACC1_1_t 4 byte				ACC1_1_X 2 byte		ACC1_1_Y 2 byte		ACC1_1_Z 2 byte	
MSB		LSB		MSB		LSB		MSB	
FF	FF	FF	FF	OF	FF	OF	FF	OF	FF

I valori dei sensori inviati, per un migliore debugging, sono **costanti**; al contrario, i valori di indice e timestamp sono variabili in base alle acquisizioni. In particolare:

- Per la matrice sensorizzata: M_1, ..., M_40 è rappresentato da un vettore 0:39 che indica gli indici dei singoli elementi sensibili;
- Per gli accelerometri: l'accelerazione lungo X vale 1, lungo Y vale 15, lungo Z vale 255 e la T vale 2000.

Il CRC utilizzato è il **CRC16-CCITT** o **CRC16/XMODEM**:

- Polinomio standard = $x^{16} + x^{12} + x^5 + 1$
- Rappresentazione del polinomio = 0x1021
- CRCIn / Init = 0x000
- ReverseIn = false
- ReverseOut = false

- XorOut = 0x0000

La libreria Arduino utilizzata si trova al seguente link su GitHub <https://github.com/RobTillaart/CRC> (v0.3.3).