
**Elettronica e Software di interfacciamento con il
Microscopio a Spettroscopia di Forza su Singola
Cellula, sviluppato nell'ambito del progetto
BIOJERKER**

*Marco Zora - IAMC-CNR U.O.S. di Capo Granitola
Alfonso Zaffuto - DAIMAR s.r.l.*

19 febbraio 2014

Indice

1	Introduzione	3
1.1	Quadrant Photodiode Front-End	3
1.2	Elaborazione Digitale dei segnali	4
1.3	SAD821	4
2	Specifiche Tecniche	5
2.1	Front End Analogici	6
2.2	Convertitore Analogico/Digitale	7
2.3	Front End Digitali	9
2.4	FPGA	10
2.4.1	JTAG	10
2.4.2	LEDS	11
2.4.3	RESET	11
2.5	ALIMENTAZIONE	12
2.6	PCB	12
2.7	Bus di Espansione	12
2.7.1	USB bus	12
2.7.2	Arduino bus	13
3	Usare la Scheda	13
3.1	Setup del Sistema	13
3.2	Protocollo Fast SPI	13
3.2.1	Elenco Comandi di Basso Livello	14
3.3	Acquisizione Segnali Analogici	15
3.4	Download dei Dati	15
3.5	Visualizzazione dei Dati	15
4	Software NetCom	16
4.1	Interfaccia Uomo/Macchina	16
4.2	Elenco Comandi	17
4.3	Codice sorgente	18

1 Introduzione

Il progetto BioJerker ha richiesto lo sviluppo di una piattaforma hardware dalle caratteristiche molto particolari. La misura fisica per la determinazione della posizione del cantilever si basa sul segnale elettrico prodotto da un sensore fotodiodo quattro quadranti. La luce generata da un laser opportunamente focalizzato nella regione terminale del cantilever, che si comporta come una superficie metallica riflettente, viene riflessa e deviata sull'area sensibile del fotodiodo. Il raggio luminoso investe il fotodiodo sulle sue quattro aree sensibili, denominate quadranti, che forniscono ognuno una corrente elettrica proporzionale alla quantità di luce incidente.

L'intensità della luce laser viene mantenuta bassa (qualche milliwatt) per non provocare deformazioni termiche sul cantilever e la corrente elettrica prodotta da ciascun quadrante del fotodiodo è dell'ordine di qualche micro ampere. Tale segnale deve essere amplificato di circa un milione di volte prima di poter essere utilizzato.

Il presente Report, descrive brevemente l'elettronica e il software di interfacciamento con la testa di misura del primo prototipo realizzato.

1.1 Quadrant Photodiode Front-End

Il circuito elettronico idoneo ad amplificare la debolissima corrente elettrica generata dal fotodiodo viene chiamato circuito di Front-End. Tale circuito è stato progettato e realizzato in modo tale da essere integrato con il fotodiodo stesso. Tale soluzione minimizza l'effetto del rumore elettronico/elettromagnetico normalmente presente in un ambiente di laboratorio, che degraderebbe in modo significativo la qualità del segnale da misurare. Sono state svolte diverse simulazioni per individuare la migliore configurazione circuitale e si è giunti ad una variante del circuito denominato amplificatore di transimpedenza. Di seguito viene riportato lo schema elettrico riferito a due dei quattro quadranti.

Le dimensioni del PCB (Printed Circuit Board) che è stato realizzato, sono di 22 mm x 22 mm. Il routing delle piste elettriche è stato svolto su 4 layers ed è stata richiesta una particolare attenzione per massimizzare gli effetti di schermatura elettrica e di resistenza elettrica mediante metallizzazione in oro delle pads.

1.2 Elaborazione Digitale dei segnali

Le soluzioni adottate in genere nei sistemi di ricostruzione delle componenti X e Y del raggio incidente sul fotodiodo sono di tipo analogico. I circuiti analogici adottati, hanno delle limitazioni in termini di rapporto segnale/rumore che si possono superare soltanto adottando un approccio nuovo di tipo digitale. Per tale ragione si è preferito spostare tutta la parte di analisi matematica per la ricostruzione delle coordinate X e Y su un dispositivo FPGA (Field Programmable Gate

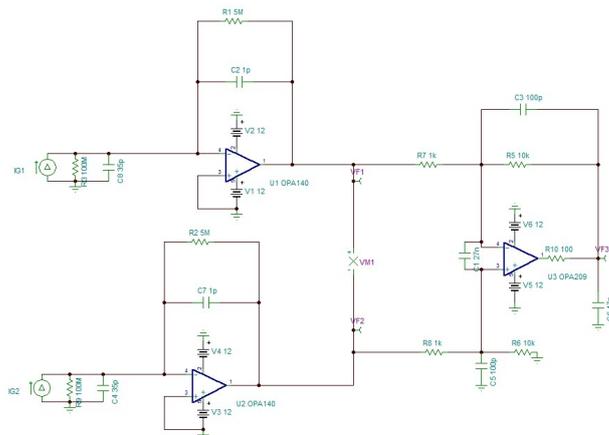


Figura 1: *Front-End per l'amplificazione della corrente generata dal fotodiode; Schema Elettrico.*

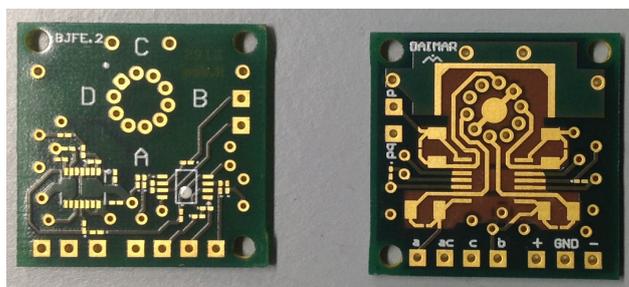


Figura 2: *Front-End per l'amplificazione della corrente generata dal fotodiode; Circuito Stampato (PCB).*

Array). I segnali elettrici preamplificati dal Front-End vengono digitalizzati ed elaborati senza passare da ulteriori stadi analogici. Le formule utilizzate per il calcolo delle componenti X e Y sono le seguenti:

$$Y = \frac{(A + B) - (C + D)}{A + B + C + B}$$

$$X = \frac{(A + D) - (B + C)}{A + B + C + B}$$

dove A, B, C, D sono i segnali elettrici generati dai quattro quadranti del fotodiode. La soluzione adottata richiede che tutti i segnali elettrici vengano acquisiti simultaneamente, perché altrimenti si introdurrebbe un errore nel calcolo. Ciò ha indirizzato la scelta di particolare componentistica elettronica idonea a soddisfare i requisiti di qualità dei segnali, velocità di campionamento e simultaneità.

1.3 SAD821

La scheda di acquisizione dati SAD821 è stata sviluppata per ottenere il massimo in termini di qualità dei dati e di integrazione con altri sistemi embedded. La sua architettura aperta permette l'interfacciamento con altri sistemi digitali tramite protocollo USB High-Speed o tramite protocollo SPI (Fast Serial Peripheral Interface). Tutte le periferiche hardware sono connesse ad un unico FPGA riprogrammabile tramite protocollo JTAG (Joint Test Action Group). Questa architettura semplifica lo sviluppo di soluzioni hardware molto diverse tra loro ed è fondamentale nella fase di ricerca e sviluppo.

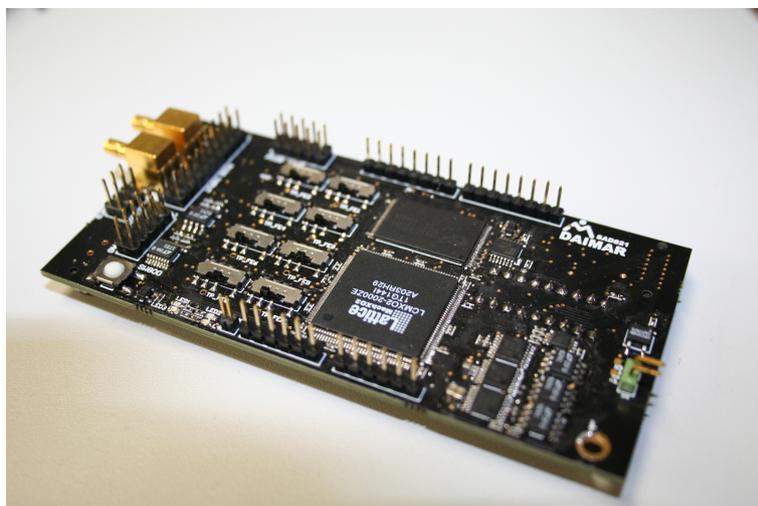
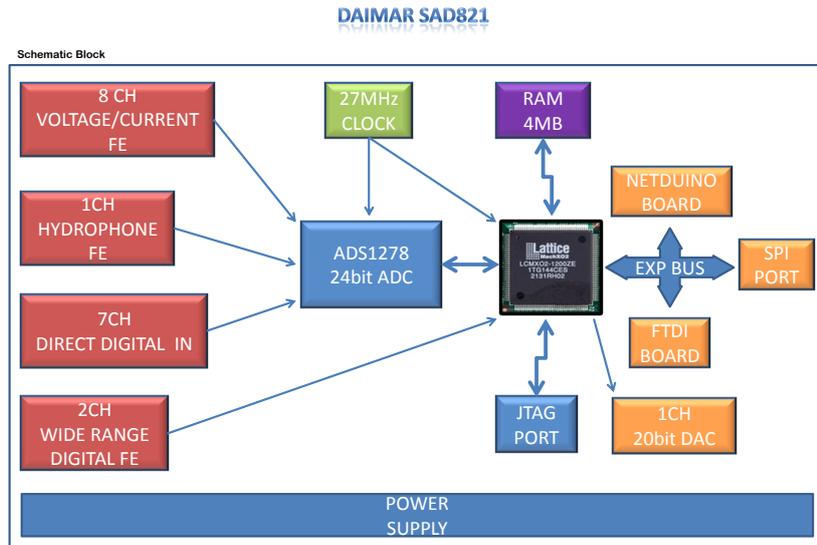


Figura 3: SAD 821, low power data acquisition board.

Con il modulo opzionale Netduino Plus 2, grazie al .NET Microframework della Microsoft, la scheda può essere comandata anche tramite connessione di rete e protocollo UDP/TCP-IP, i dati possono essere salvati su scheda di memoria SD card (4GB+) e molte delle funzioni possono essere programmate in C# nell'ambiente di sviluppo Visual Studio.

2 Specifiche Tecniche

- 8 canali di ingresso analogici selezionabili in tensione o corrente;
- 2 canali di ingresso per segnali modulati in frequenza;
- 8 Front-End Analogici in Tensione con range +10V,-10V;
- 8 Front-End Analogici in Corrente 0mA,20mA;
- 8 Comparatori a soglia programmabili;
- ADC 8 canali a campionamento simultaneo 24bit;
- Frequenza di Campionamento massima di 144 kSPS per canale;
- Interfaccia Real Time USB 2.0 High-Speed;



- Interfaccia Near Real Time Standard Arduino SPI;
- Lattice low power FPGA;
- 4MByte internal RAM Buffer.
- 4 modalità di funzionamento: High Speed, High Resolution, Low Power, Sleep Mode;
- 1 DAC 20 bit con uscita in tensione +10V,-10V.

2.1 Front End Analogici

La scheda SAD821 dispone di 8 canali di ingresso analogici. La presenza di 8 microinterruttori consente di selezionare una di tre tipologie di ingresso analogico: tensione, corrente, diretto.

In modalità corrente, l'ingresso analogico è in grado di misurare correnti comprese tra 0 e 25mA. Una resistenza da 100 Ω trasforma tale corrente in una tensione compresa tra 0 e 2,5V, il range di misura del convertitore analogico-digitale scelto. La resistenza è collegata ad un buffer di tensione realizzato con un amplificatore operazionale. Tale buffer disaccoppia l'ingresso in corrente dal convertitore analogico-digitale, che ha un'impedenza di ingresso differenziale che può raggiungere i 14 $k\Omega$. In parallelo alla resistenza da 100 Ω è collegato un condensatore da 680 pF che serve a ridurre la banda dello stadio analogico. Come risulta dalle simulazioni effettuate con LTSpice, la banda a 3 dB dello stadio di corrente è di 3,5 MHz. Come resistenza da 100 Ω si è scelto di utilizzare delle resistenze con elevata stabilità con la temperatura pari a $\pm 5\text{ppm}/^{\circ}\text{C}$ e con un'accuratezza di $\pm 0,01\%$. In modalità tensione, l'ingresso analogico converte tensioni comprese tra -10V e +10V in una tensione compresa tra 0 V e 2,5V. Il circuito di adattamento di tensione è realizzato con un amplificatore operazionale, cinque resistenze opportunamente dimensionate, due conden-

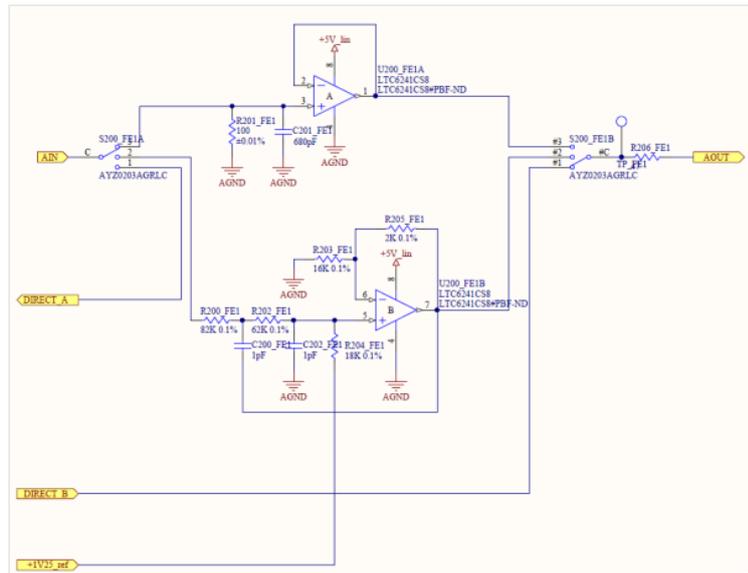


Figura 4: Schema elettrico di uno degli 8 canali di ingresso analogico.

satori e un riferimento di tensione da 1,25 V. Le resistenze adoperate hanno un'accuratezza di $\pm 0,1\%$. I due condensatori formano un filtro di Sallen-Key. Il filtro Sallen Key è un tipo di filtro attivo, noto e diffuso grazie alla sua semplicità. Il circuito fornisce una risposta a 2 poli (12dB/ottava) di tipo filtro passa-basso. Dalle simulazioni effettuate la banda passante a 3 dB del filtro è 3 MHz. La terza posizione del microinterruttore prevede un collegamento diretto dei piedini d'ingresso della scheda con il convertitore analogico-digitale.

2.2 Convertitore Analogico/Digitale

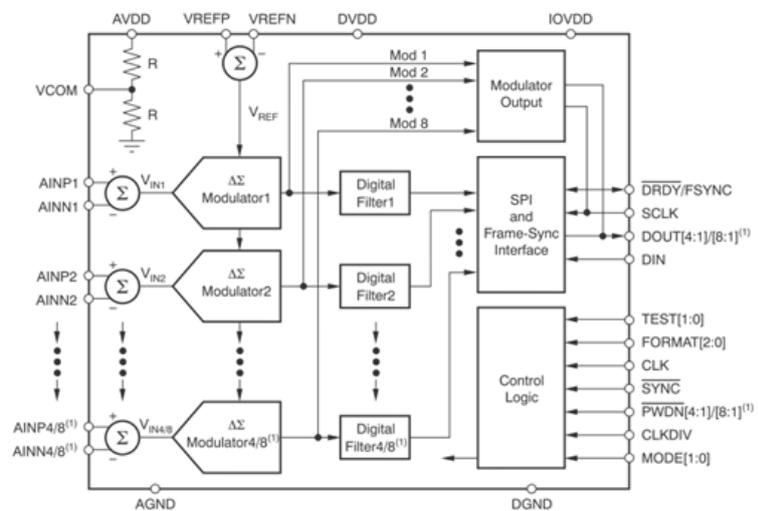


Figura 5: Schema di funzionamento del convertitore A/D delta-sigma ADS1278 (Texas Instruments).

La conversione A/D è affidata ad un convertitore delta-sigma con 8 canali di ingresso e una risoluzione di 24 bit. Il convertitore comprende 8 modulatori delta-sigma del sesto ordine, seguiti da altrettanti filtri FIR a basso ripple. I modulatori misurano la differenza tra il segnale d'ingresso differenziale $V_{IN} = (A_{INP} - A_{INN})$ e il segnale differenziale di riferimento, $V_{REF} = (V_{REFP} - V_{REFN})$. I filtri digitali ricevono il segnale dal modulatore e generano un'uscita digitale a basso rumore. Per consentire un trade-off tra velocità, risoluzione e potenza dissipata, sono supportate quattro modalità di funzionamento:

- High-Speed
- High-Resolution
- Low-Power
- Low-Speed

In modalità High-Speed la massima frequenza di campionamento è pari a 144 kSPS. In modalità High-Resolution l'SNR è pari a 111dB; in modalità Low-Power la potenza dissipata è di 31mW per canale; infine in modalità Low-Speed la potenza dissipata è di soli 7mW per canale, anche se la frequenza di campionamento è di soli 10.5 kSPS. I filtri digitali possono essere bypassati, consentendo un accesso diretto all'uscita del modulatore.

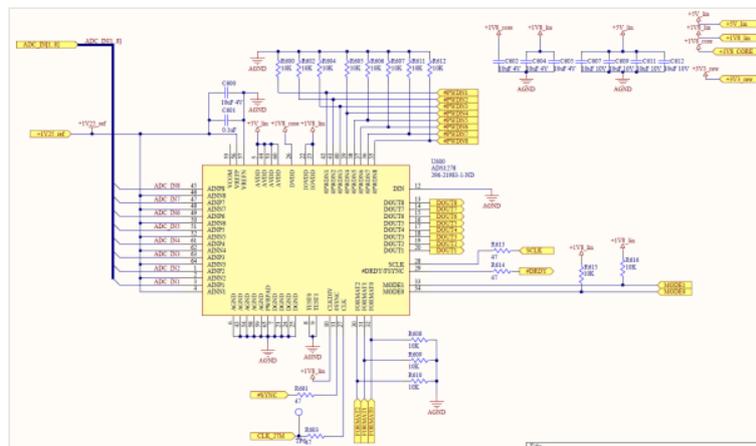


Figura 6: Schema elettrico relativo al convertitore A/D.

A_{INP_x} e A_{INN_x} sono gli ingressi differenziali della scheda, che vengono confrontati dal convertitore con l'ingresso differenziale di riferimento $V_{REF} = (V_{REFP} - V_{REFN})$. Il massimo valore positivo misurabile è $+V_{REF}$, che produce il codice digitale 7FFFFFFh. Al contrario, il massimo valore negativo misurabile è $-V_{REF}$, che produce il codice digitale 800000h. Dal momento che gli ingressi del convertitore sono differenziali, mentre i segnali che escono dal front-end del fotodiodo sono single-ended, si è deciso di fissare il valore dei piedini A_{INN_x} ad una tensione di riferimento pari a 1,25V. I piedini A_{INP_x} sono stati collegati alle 8 uscite dei front-end analogici. La tensione di riferimento ai capi

dei piedini VREFP e VREFN è fissata a 1,25V da un riferimento di tensione. Ai capi dei due piedini sono stati collegati due condensatori di by-pass da $0.1\mu\text{F}$ e $10\mu\text{F}$ in parallelo. In questo modo agli ingressi AIN-Px è possibile inviare segnali analogici di tensione compresi tra 0V e +2,5V. Il convertitore A/D richiede un segnale di clock per operare. Alla massima frequenza di campionamento, la frequenza del clock può essere sia 27 MHz che 13.5MHz nella modalità Low-Power, 27 MHz o 5.4 MHz nella modalità Low-Speed. Tale frequenza va impostata attraverso l'ingresso CLKDIV. Nella modalità High-Resolution, la massima frequenza è 27 MHz. Essendo necessario operare alla massima velocità possibile, il piedino CLKDIV è stato collegato al livello logico alto. Il piedino CLK è stato collegato ad un oscillatore al quarzo a 27 MHz. Gli ingressi MODE[1:0] consentono di impostare una delle quattro modalità di funzionamento e sono stati collegati all'FPGA. L'ingresso #SYNC, anch'esso collegato all'FPGA, stabilisce l'inizio della conversione analogico-digitale. Può essere utilizzato per sincronizzare più dispositivi. I dati possono essere letti dal convertitore con due protocolli e diverse opzioni riguardo il formato dei dati. Gli ingressi FORMAT[2:0], collegati all'FPGA, gestiscono queste opzioni. Il convertitore analogico-digitale comunica con l'FPGA tramite un'interfaccia SPI, che comprende gli ingressi SCLK, FSYNC, DIN e le uscite DOUT[8:1]. Tutti questi segnali, tranne DIN, sono stati collegati all'FPGA. I canali del convertitore possono essere spenti indipendentemente attraverso l'uso degli ingressi #PWDNx. Questi ingressi sono gestiti dall'FPGA ma sono anche collegati a massa tramite delle resistenze di pull-down. Il convertitore richiede tre alimentazioni diverse:

- AVDD è l'alimentazione analogica che accende il modulatore, collegata a +5V
- DVDD è l'alimentazione digitale, collegata a 1,8 V
- IOVDD è l'alimentazione degli I/O digitali, collegata a 1,8 V.

Tutti i piedini di alimentazione sono stati bypassati con condensatori da $0,1\mu\text{F}$ e da $10\mu\text{F}$.

2.3 Front End Digitali

Nella scheda SAD821 sono presenti due ingressi per segnali digitali. Gli ingressi digitali sono collegati ad un primo stadio analogico che funge da adattatore di livello di tensione. Tale circuito analogico, realizzato tramite amplificatori operazionali, trasforma un segnale di tensione compreso tra -10V e +10V in un segnale di tensione compreso tra 0V e + 2,5V. Questo segnale di tensione in uscita dal primo stadio viene confrontato con un valore di tensione analogico, impostato attraverso uno dei due canali del convertitore digitale-analogico. Per il confronto sono stati utilizzati i comparatori LT1721 della Linear Technology.

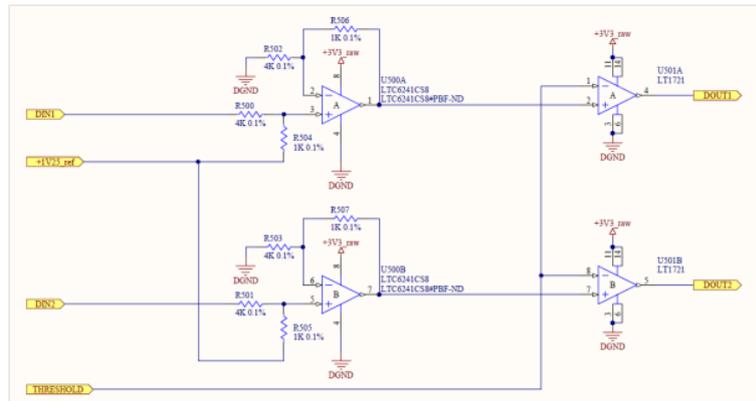


Figura 7: Schema elettrico ingressi per segnali in frequenza.

2.4 FPGA

L'FPGA scelto è il MACHXO2 della Lattice Semiconductor. Il principale vantaggio di questa famiglia di FPGA sono i bassi consumi e la notevole flessibilità nell'utilizzo. E' stata scelta la versione MachXO2-2000 con package da TQFP da 144 pin che dispone di 2112 look-up table, 1 PLL, interfacce hardware I2C e SPI e un contatore hardware. Inoltre è stata scelta la versione ultra low power (ZE) per ridurre i consumi dell'FPGA, che in stand-by possono raggiungere i $19 \mu\text{W}$.

2.4.1 JTAG

La programmazione dell'FPGA avviene tramite programmatore JTAG fornito dalla Lattice Semiconductor. Per questo motivo 9 piedini dell'FPGA sono stati portati su un connettore, mostrato in figura.

I piedini interessati dalla programmazione sono TCK, TDI, TDO e TMS. Si è deciso di portare sul connettore anche gli altri piedini del connettore che in genere non sono utilizzati: si tratta dei segnali PROG, JTAGEN, INITN e DONE per applicazioni future.

2.4.2 LEDS

La scheda elettronica contiene quattro LED che sono stati collegati all'FPGA tramite una resistenza di polarizzazione di 100Ω , come mostrato nella seguente figura.

2.4.3 RESET

All'FPGA è stato collegato un segnale di RESET, generato attraverso il chip MAX823 della Maxim Semiconductor.

Il chip genera un segnale di RESET in ciascuno dei seguenti tre casi:

- viene premuto il tasto di RESET presente nella scheda
- abbassamento della tensione di alimentazione

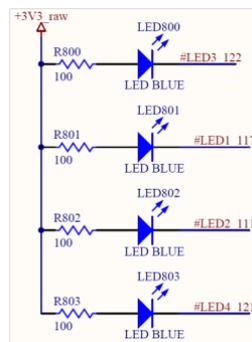


Figura 10: Collegamento elettrico dei quattro LED di servizio.

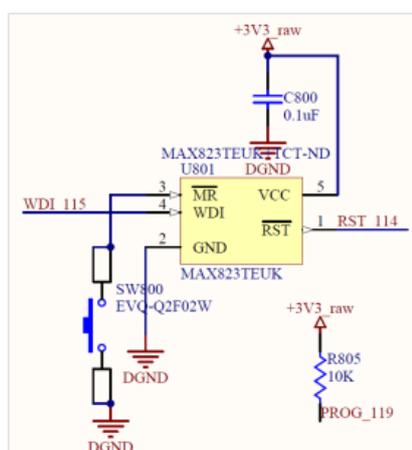


Figura 11: Schema elettrico del circuito di RESET.

2.6 PCB

La progettazione del PCB della scheda SAD821 si sviluppa su 6 strati. Le linee guida principali che sono state seguite riguardano la separazione dei circuiti analogici da quelli digitali. In particolare è stata separata la massa analogica da quella digitale per evitare che la corrente di ritorno dei circuiti digitali introduca del rumore nei segnali analogici. Nel PCB della scheda i piani di massa analogica e digitale si uniscono in un solo punto.

2.7 Bus di Espansione

Nella scheda è presente un bus di espansione (denominato NETBUS) condiviso tra diverse periferiche. Il NETBUS collega direttamente l'FPGA con una interfaccia SPI esterna, il connettore Arduino like, lo slot USB e un connettore Board to Board. Questa soluzione permette di condividere dati con dispositivi esterni mantenendo limitato il numero di piedini necessari dell'FPGA, semplificando anche il routing delle piste elettriche.

2.7.1 **USB bus**

Il NETBUS può essere interfacciato ad un modulo USB tipo DLP-USB1232H della FTDI. In questo modo è possibile scambiare dati con un PC tramite interfaccia USB High Speed ad una velocità superiore a 4 MB/s.

2.7.2 **Arduino bus**

Il NETBUS è accessibile da un connettore standard tipo Arduino. In questa modalità, la scheda SAD821, diventa a tutti gli effetti uno SHIELD per Arduino. La comunicazione tra un microcontrollore esterno e la scheda avviene tramite protocollo SPI.

3 **Usare la Scheda**

Il funzionamento della scheda SAD821 dipende dal firmware che viene caricato all'interno dell'FPGA. E' stato sviluppato un primo firmware base dove sono state implementate le funzioni essenziali per utilizzare la scheda come sistema di acquisizione dati e un secondo firmware per l'acquisizione delle curve di forza sulle cellule. In entrambi i casi, i comandi vengono impartiti alla scheda tramite il NETBUS ed in particolare tramite protocollo SPI. Per ampliare le funzionalità del sistema si è utilizzata una scheda di espansione Netduino Plus 2. Nel caso in cui si vogliono prelevare e/o elaborare i dati in real time tramite un PC allora bisogna ricorrere all'interfaccia USB.

3.1 **Setup del Sistema**

Il sistema di acquisizione base è costituito da tre moduli principali.

- Alimentatore con tripla uscita 12V, 6V, 3.3V
- Scheda SAD821 con firmware NET rev 1.0 o superiore
- Scheda Netduino plus 2

La scheda Netduino è dotata di memoria di massa SD-Card e interfaccia di rete TCP/IP. La programmazione della scheda avviene tramite il Microframework della Microsoft. E' possibile scaricare l'ambiente di sviluppo Visual Studio C# e seguire le istruzioni fornite sul sito Netduino per riprogrammare il sistema secondo le proprie esigenze.

3.2 **Protocollo Fast SPI**

Il protocollo di basso livello per la comunicazione con la scheda SAD821 è il Fast SPI. Questo non è altro che un protocollo standard SPI dove la frequenza del clock può raggiungere i 50MHz. Il massimo throughput raggiungibile è quindi 50 Mbit/s.

Ci sono due canali di comunicazione che condividono lo stesso bus e che sono individuati da due segnali di selezione CS1 e CS2 (CS1=NETBUS pin D9, CS2=NETBUS pin D10). CS1 seleziona la scrittura dei comandi

mentre CS2 seleziona la lettura dei dati.

I comandi sono formati da parole di 16 bit ciascuna. All'inizio di una transazione il segnale di selezione CS1 deve passare dal livello basso al livello alto. Il bit più significativo viene allora letto per primo, sul fronte di salita del clock. Dopo 16 bit il segnale CS1 deve ritornare basso per terminare la transazione del comando.

Per leggere i dati dalla memoria RAM bisogna per prima cosa portare il segnale CS2 dal livello basso al livello alto. Sul fronte positivo del segnale di clock, i bit vengono presentati sul bus a partire dal più significativo. Al termine della transazione il segnale CS2 deve tornare al livello basso.

3.2.1 **Elenco Comandi di Basso Livello**

I comandi di basso livello sono quelli più vicini all'hardware e vengono inviati tramite bus SPI selezionando la linea CS2. Ogni comando è diviso in due parti da 8 bit ciascuna. La prima parte rappresenta il codice operativo e la seconda parte il parametro.

Codice Operativo	Parametro	Descrizione
0	X	NOP. No Operation.
1	0:OFF 1:ON	RADDR-CLR. Clear Read Address internal flag.
2	0:OFF 1:ON	WADDR-CLR. Clear Write Address internal clag.
3	0:OFF 1:ON	ADC-EN. Enable Analog/Digital Converter chip.
4	0:High-Speed 1:High-Resolution 2:Low-Power 3:Low-Speed	ADC-MODE. Set ADC acquisition mode.
5	0...7	ADC-CH-SEL. Set ADC single channel.
6	0:SINGLE 1:ALL	ADC-CH-SEL-ALL. Set ADC all channels.
7	0:OFF 1:ON	RAM-EN. Enable Ram chip.
8	0:CH1 1:Temperature 2:DAC-Volt 3:Threshold-Volt	ADC-CH1-MUX. Select input for ADC channel 1.
9	0:OFF 1:ON	EN-12V. Enable 12 Volt front-end for hydrophone and DAC output.
10	0:OFF 1:ON	EN-5V. Enable 5 Volt front-end, voltage reference and ADC power.
11	0:Entire Ram Memory 1:One Ram Block only	BLOCK-STOP-EN. Enable write memory stop bit.

3.3 Acquisizione Segnali Analogici

Quando viene lanciata una sessione di acquisizione, i dati vengono memorizzati in una RAM interna ad alta velocità. Quando la RAM è piena, o quando viene memorizzato il numero previsto di campioni, l'acquisizione si interrompe e la scheda si porta in modalità di attesa. I dati vengono scaricati dalla RAM tramite SPI ed elaborati dal Netduino per essere spediti in rete o memorizzati su SD-Card.

3.4 Download dei Dati

Il download dei dati viene fatto dalla RAM della scheda SAD821 al Netduino e poi dal Netduino ad una postazione remota collegata in rete. I dati vengono memorizzati nella RAM un Byte per volta iniziando dall'indirizzo 000000 all'indirizzo esadecimale 400000. Ogni cam-

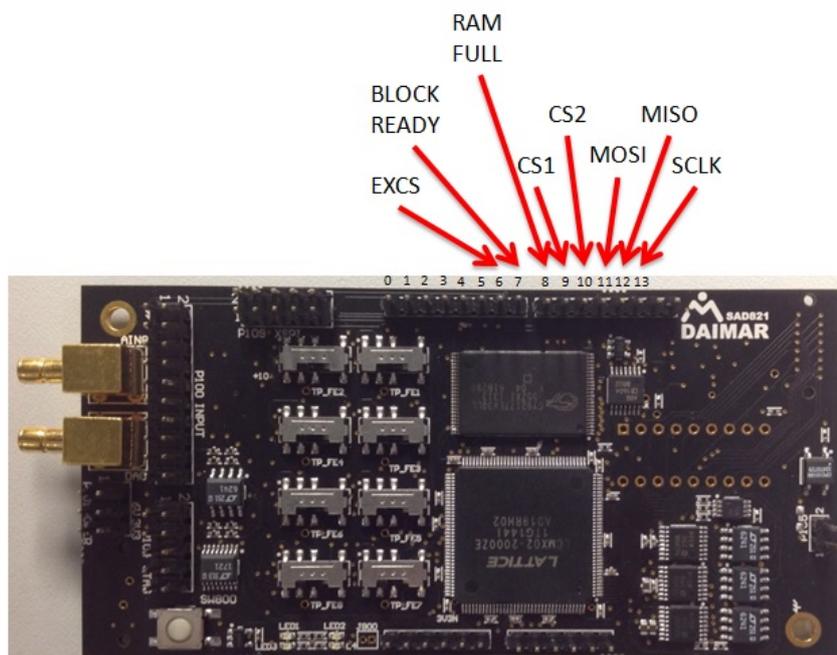


Figura 12: SAD 821 NETBUS pinout.

pione da 24 bit viene memorizzato in 3 Bytes consecutivi iniziando dal meno significativo e dal canale 8.

3.5 Visualizzazione dei Dati

I dati salvati nella RAM devono essere convertiti in un formato diverso prima di essere visualizzati. La procedura corretta è quella di creare campioni da 24 bit raggruppando 3 bytes adiacenti per volta. Il dato così ottenuto è di tipo SIGNED. Se si vogliono ottenere campioni in formato standard 32bit signed bisogna effettuare una estensione di segno sul byte MSB.

4 Software NetCom

Il software NetCom è stato sviluppato per semplificare l'uso della scheda SAD821 da parte dell'utente standard. Tramite questo software, per sistemi operativi Windows, è possibile impartire comandi alla scheda, cambiare le modalità di funzionamento, eseguire acquisizioni, salvare su SD-Card i dati acquisiti ed infine scaricare i dati su un PC. Ovviamente tali funzioni possono essere facilmente espanso o modificate da utenti con adeguate competenze tecniche.

4.1 Interfaccia Uomo/Macchina

L'interfaccia Uomo/Macchina Biojerker è stata realizzata con l'ambiente di sviluppo LabView che si integra perfettamente con il software NetCom. In questa interfaccia vengono visualizzati in tempo reale i

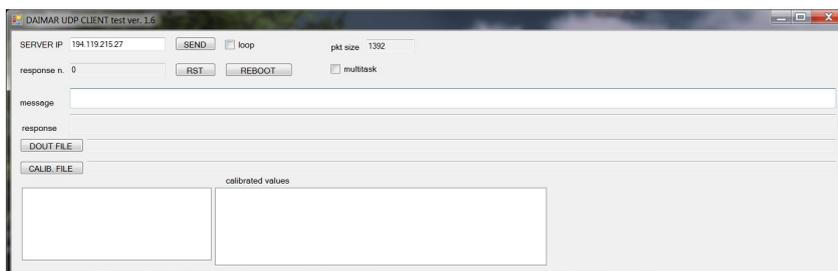


Figura 13: *Interfaccia Software NetCom per la comunicazione UDP tra il PC e la scheda Netduino/SAD821.*

valori di deflessione (Y) e torsione (X) del cantilever. La posizione del raggio laser sul fotodiodo viene visualizzata mediante un cerchio rosso la cui intensità di colore è proporzionale alla potenza del laser incidente. Dopo aver allineato il fascio laser con il cantilever, agendo sulle movimentazioni meccaniche del fotodiodo, si agisce in modo da portare il cerchio rosso nel centro dei quattro quadranti. Lo stesso software permette di eseguire una calibrazione del cantilever partendo da uno spostamento nanometrico noto rispetto ad una superficie di contatto.

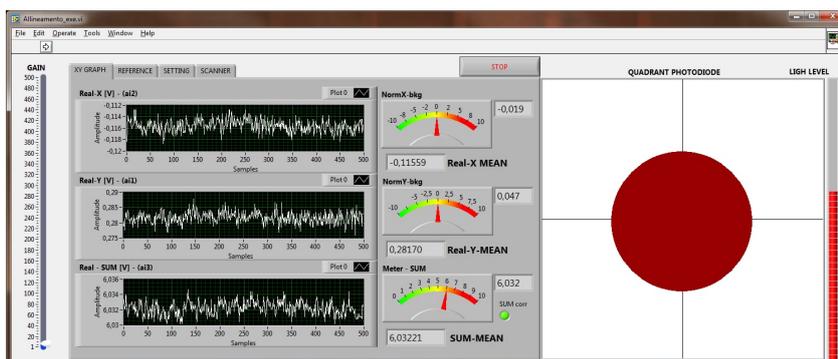


Figura 14: *Interfaccia di allineamento BioJerker.*

Gli spostamenti nanometrici del cantilever vengono visualizzati come spostamenti in Y e in X del cerchio rispetto alla sua posizione di equilibrio. Con il primo prototipo si è riusciti a misurare uno spostamento di ± 1 nm, ma è possibile migliorare questa sensibilità anche di un fattore 10 ottimizzando la progettazione delle parti meccaniche che risultano molto sensibili alle vibrazioni acustiche e alle variazioni di temperatura ambientali.

4.2 Elenco Comandi

I comandi vengono inviati alla scheda Netduino tramite interfaccia ethernet come semplici stringhe di testo, utilizzando il protocollo di comunicazione UDP. Inviare datagrammi UDP risulta molto più veloce rispetto a messaggi TCP/IP ed è possibile visualizzare la posizione del cantilever in tempo reale con una frequenza di acquisizione di circa 10 kSPS.

Esistono comandi con parametro e comandi senza parametro. I comandi con parametro sono del tipo: "comando = parametro". Il segno "=" separa il comando dal parametro. Nella tabella seguente sono riportati alcuni comandi di esempio:

Comando	Parametro	Descrizione
setbufsize=	numero campioni (0 → 1398101)	definisce il numero di campioni da salvare nella memoria RAM interna durante un'acquisizione.
myip=	indirizzo IP	setta l'indirizzo IP del computer host al quale la scheda Netduino risponde.
read	xxx	inizia una acquisizione dati. I dati vengono salvati nella memoria RAM interna.
savetxt	xxx	salva tutti i dati acquisiti nella memoria di massa SD-card della scheda Netduino. I dati vengono salvati in formato ASCII.
saveraw	xxx	salva tutti i dati acquisiti nella memoria di massa SD-card della scheda Netduino. I dati vengono salvati in formato binario RAW.
led_on	xxx	accende il LED di debug della scheda Netduino.
led_off	xxx	spegne il LED di debug della scheda Netduino.

Il set di comandi può essere modificato intervenendo sul codice in C# della scheda Netduino.

4.3 Codice sorgente

Il codice sorgente dell'applicazione NetCom è scritto in C# ed è stato compilato con l'ambiente di sviluppo gratuito Microsoft Visual Studio Express. Di seguito si riporta, a titolo esemplificativo, parte del listato che mostra come inviare i comandi e ricevere le risposte dal server Netduino-SAD821.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;

using System.Net;

using System.Text;
using System.Windows.Forms;

using Class_datagram_IO;

namespace PC_datagram_IO_2
{
    public partial class MainForm : Form
    {
        private NetIO netio;

        // Inizializza la classe netio
        // con un oggetto tipo "NetIO"
        // la gestione del protocollo TCP/IP-datagram
        // e' automatica
        public MainForm()
        {
            InitializeComponent();
            netio = new NetIO();
        }

        // tReceiverIP.Text = IP del PC HOST collegato
        // al Netduino
        // tSend.Text = comando inviato al Netduino
        // netio.SendDatagram(...) = Invio del datagram
        // netio.ReceiveDatagram() = Resta in attesa
        // di ricevere una risposta dal Netduino
        // richReceived.Text = messaggio di risposta
        private void fSendMessage()
        {
            netio.SendDatagram(tReceiverIP.Text, tSend.Text);
            richReceived.Text = netio.ReceiveDatagram();
        }
    }
}
```

La funzione "netio.ReceiveDatagram()" rimane in attesa fino a quando Netduino non ha fornito una risposta. Se il sistema non risponde entro un tempo prefissato (2 secondi), il sistema invia un messaggio che indica un errore di time-out.