



**UNIVERSITA' POLITECNICA DELLE MARCHE**

**FACOLTA' DI INGEGNERIA**

Corso di Laurea triennale in Ingegneria biomedica

**Interfacciamento di sensori di campo magnetico su microcontrollori PIC per  
monitoraggio in ambienti di risonanza magnetica**

*Interface between magnetic field sensors and PIC microcontroller for monitoring of  
magnetic resonance environments*

Relatore: Chiar.mo

**Prof. Francesco Piva**

Tesi di Laurea di:

**Michele Girotti**

**A.A. 2021 / 2022**

# Indice

## **Introduzione**

## **Capitolo 1: Configurazioni tecniche**

- 1.1 Microcontrollori
- 1.2 Display
- 1.3 Interfaccia I2C
- 1.4 Interfaccia SPI
- 1.5 Touch Screen
- 1.6 Periferiche

## **Capitolo 2: Nozioni fondamentali**

- 2.1 Risonanza magnetica nucleare
- 2.2 Contesto e normative sull'inquinamento elettromagnetico
- 2.3 Effetti dell'inquinamento elettromagnetico
- 2.4 Logistica ed interferenze
- 2.5 Effetto Hall

## **Capitolo 3: Obiettivo della tesi**

## **Capitolo 4: Materiali e metodi**

- 4.1 Descrizione hardware
- 4.2 Configurazioni
- 4.3 Descrizione software

## **Capitolo 5: Risultati**

- 5.1 Listato di programmazione: main.c
- 5.2 Listato di programmazione: Risonanza.h
- 5.3 Foto prototipo

## **Capitolo 6: Discussione**

## **Appendice**

## **Bibliografia**

## **Ringraziamenti**

## **Indice delle figure**

- Fig.1 Schema a blocchi di un microcontrollore
- Fig.2 Schema a blocchi protocollo I2C
- Fig.3 Schema a blocchi interfaccia SPI
- Fig.4 Mode 0
- Fig.5 Mode 1
- Fig.6 Mode 2
- Fig.7 Mode 3
- Fig.8 Precessione del nucleo con spin parallelo e opposto
- Fig.9 Livelli energetici
- Fig.10 Linee di campo magnetico prodotte da una MRI
- Fig.11 Effetto Hall
- Fig.12 Schema a blocchi prototipo
- Fig.13 Microcontrollore
- Fig.14 Sensore magnetico
- Fig.15 Magnete di Halbach
- Fig.16 Linee di campo del magnete di Halbach
- Fig.17 Configurazione microcontrollore con sensore magnetico
- Fig.18 Configurazione microcontrollore con display
- Fig.19 Rappresentazione prototipo
- Fig.20 Segnale continuo
- Fig.21 Segnale alternato
- Fig.22 Illustrazione dati informativi
- Fig.23 Rappresentazione della FFT

## **Indice delle tabelle**

- Tab.1 Limiti di esposizione valutati per campi magnetici
- Tab.2 Limiti di esposizione valutati per campi elettrici indotti
- Tab.3 Valori massimi di tolleranza ad interferenze

## Introduzione

Macchinari di risonanza magnetica nucleare garantiscono tecniche diagnostiche molto efficienti e sono ormai molto diffuse nei centri ospedalieri. Sebbene queste apparecchiature siano molto utili nell'imaging, emettono campi magnetici molto intensi ed onde nel range delle radiofrequenze che possono interferire con il corretto funzionamento degli altri strumenti biomedicali ed essere pericolosi per l'organismo. È opportuno per questo motivo misurare i livelli di campo magnetico prodotti nelle sale di risonanza magnetica e se necessario trovare un modo per limitare la loro diffusione, ad esempio installando sistemi di schermatura o adottando soluzioni logistiche "ad hoc". Tale trattazione si pone lo scopo di spiegare nel dettaglio gli effetti del cosiddetto "inquinamento elettromagnetico", con particolare attenzione alle normative vigenti e successivamente sarà proposto un prototipo che dopo aver acquisito un segnale di campo magnetico prodotto da una risonanza magnetica, possa rendere i dati informativi e facilmente interpretabili per gli operatori sanitari. In particolare, inizialmente saranno illustrate le tecnologie elettroniche più diffuse nel mercato con particolare riferimento ai microcontrollori. Questi dispositivi oltre ad essere a basso costo, possono comunicare con un gran numero di periferiche, tanto da essere presenti ormai in quasi tutti gli strumenti biomedicali. Sebbene nei decenni scorsi l'elettronica analogica fosse molto utilizzata, con il passare degli anni il digitale ha soppiantato la precedente tecnologia e ha dato origine a nuovi sistemi, di cui i microcontrollori sono i pilastri. È opportuno quindi inizialmente dedicare parte della trattazione alla descrizione di questi elementi, citando anche le principali periferiche e i suoi possibili collegamenti. Successivamente nel secondo capitolo si solleverà il problema relativo all'emissione di campi elettromagnetici pericolosi per l'uomo e per gli altri strumenti, focalizzando la riflessione sugli ambienti ospedalieri e in particolare le sale di risonanza magnetica nucleare. Infine, si avanzerà una soluzione tramite la presentazione di un prototipo, il quale sarà costruito sfruttando alcune delle tecnologie descritte nel primo capitolo e sarà caratterizzato da un firmware, ovvero da un programma che porterà il dispositivo a compiere le istruzioni desiderate. Nel corso della trattazione si esporranno anche i

vantaggi dello strumento, i limiti di elaborazione e si presterà attenzione alle possibili modifiche da attuare in futuro nonché alle possibili evoluzioni a cui lo stesso potrebbe incorrere al fine di migliorare il suo funzionamento.

# Capitolo 1

## Configurazioni tecniche

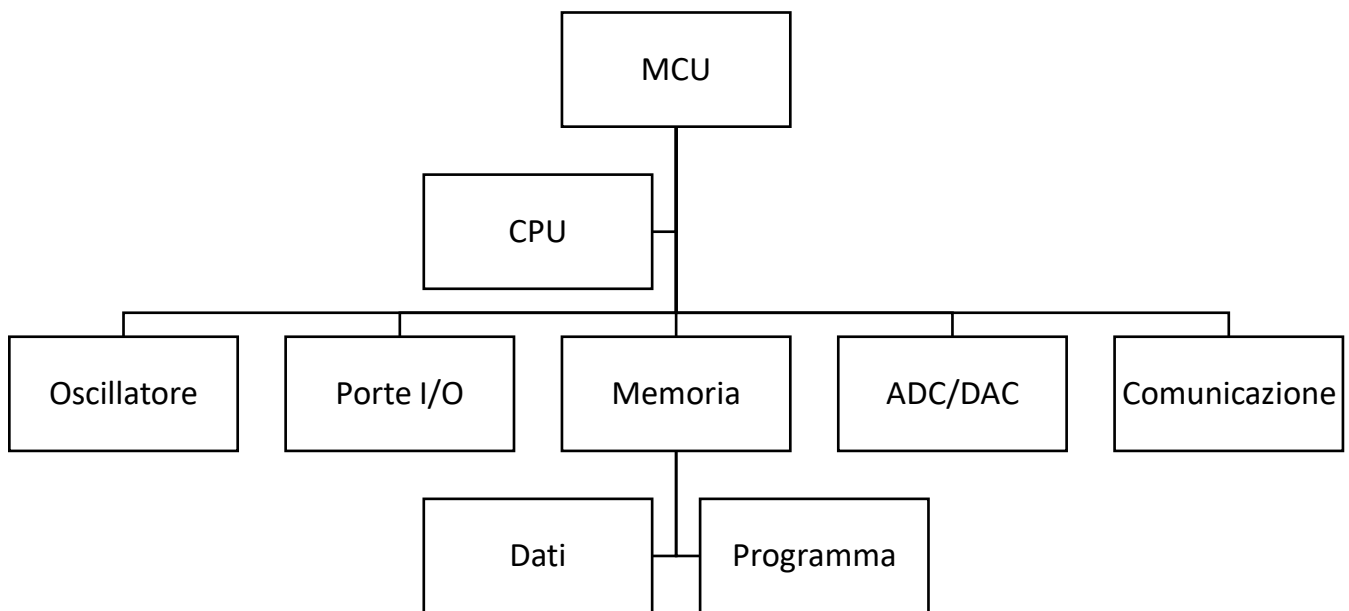
### 1.1 Microcontrollori

I microcontrollori (MCU) costituiscono una gamma di dispositivi hardware programmabili caratterizzati da un circuito integrato ed hanno la capacità di compiere innumerevoli funzioni, tra cui l'acquisizione dei dati, l'elaborazione delle informazioni e il pilotaggio delle periferiche. Gran parte della strumentazione anche in ambito biomedicale gode della loro presenza essendo molto efficienti, semplici da usare e facilmente programmabili tramite programmatore esterno. Infatti, sebbene non abbiano una potenza di calcolo comparabile con i moderni microprocessori che possono arrivare a qualche GHz, gli MCU hanno un ottimo rapporto qualità/prezzo che li rendono molto appetibili nel mercato. Attualmente esistono molti tipi di microcontrollori, i quali tuttavia, devono essere scelti opportunamente in base alle proprie esigenze tecniche; per compiere un'analisi esaustiva volta alla selezione del dispositivo bisogna valutare le singole componenti che lo caratterizzano:

- La CPU (Central Processing Unit) esegue le istruzioni volte al corretto funzionamento del dispositivo, elabora i dati presenti in memoria e coordina tutte le componenti del microcontrollore.
- La memoria permette di salvare tutte le informazioni del codice di programmazione ed in particolare compaiono una SRAM, ovvero una memoria volatile che memorizza i dati e i valori delle variabili, una FLASH che consente di salvare il programma da eseguire ed infine una EEPROM che costituisce una tipologia di memoria non volatile, il cui contenuto può essere riscritto mediante segnali elettrici.
- L'oscillatore interno ed esterno che emettono segnali d'onda a una precisa frequenza per la lettura/scrittura dei bit o più in generale per sincronizzare le attività del microcontrollore.

- Le porte input/output ovvero i singoli pin del dispositivo che possono essere configurati per pilotare periferiche esterne, convertire il segnale da analogico a digitale e viceversa, connettere il programmatore al dispositivo, fornire l'alimentazione, inviare la frequenza di oscillazione.
- Il modulo di conversione Analogico/Digitale ADC, DAC.
- I moduli di comunicazione che possono essere SPI, I2C, USB ecc.
- Moduli aggiuntivi

Fig.1 Schema a blocchi di un microcontrollore



La Microchip Technology è un'azienda americana, con sede in Arizona, leader nella produzione di microcontrollori e altri dispositivi elettronici. In particolare, i microcontrollori PIC costituiscono un gruppo di MCU con elevate potenzialità, costi sostenuti ed alta affidabilità tanto da soddisfare gli standard internazionali relativi a IEC 61508 e IEC 60730 che concernono la sicurezza dei sistemi elettronici. Attualmente esistono nel mercato diverse tipologie di prodotti offerti dalla Microchip

Technology, tra cui i PIC 18, 24, 32 e quelli “ds” che includono il digital signal processing del segnale in ingresso.

La programmazione del microcontrollore si sviluppa in tre particolari step: innanzitutto occorre scrivere il codice su un opportuno ambiente di programmazione, ad esempio, il software MPLAB che comprende sia un'interfaccia utente (MPLAB IDE) che consente di digitare le righe di codice, realizzare nuovi progetti e conoscere lo status del microcontrollore, sia un compilatore come XC8 Compiler associato al primo che traduce il codice sorgente in codice oggetto, ovvero in linguaggio macchina. Il secondo step corrisponde al lavoro di questo secondo software quindi alla compilazione senza la quale sarebbe impossibile comunicare con il PIC. Infine, come terzo ed ultimo step c'è la programmazione, ovvero il trasferimento dei dati e delle istruzioni alla memoria del microcontrollore grazie all'utilizzo di un programmatore esterno.

Gli MCU presentano dei registri, ovvero delle celle di memoria i cui valori possono essere configurati grazie a una programmazione esterna e che stabiliscono univocamente il lavoro della CPU. Esistono molti registri, alcuni dei quali devono essere configurati obbligatoriamente come ad esempio quelli relativi alle “configuration bits”: questi permettono di stabilire le tensioni di riferimento, le frequenze di oscillazione, la protezione della memoria e in generali tutte le istruzioni volte a inizializzare il dispositivo. Inoltre, per poter interagire con le porte del PIC occorre compilare i registri di porta TRIS, PORT e LAT. Il primo consente di selezionare la direzione del pin corrispondente, in particolare sarà posto a 0 se si invia un segnale in output oppure sarà posto ad 1 se si riceve un segnale in input; il secondo permette di settare il livello logico della porta; il terzo infine fa riferimento al valore in uscita del latch posto immediatamente prima al pin I/O.



## 1.2 Display

I microcontrollori hanno la possibilità di comunicare con dei display che a sua volta permettono di proiettare delle lettere, dei numeri, delle frasi, dei grafici cartesiani oppure interagire direttamente con il touch screen. Attualmente in commercio esistono diversi tipi di display ma quelli che godono di un ottimo rapporto qualità/prezzo sono i TFT. Questi sono un'evoluzione dei comuni LCD (Liquid cristal display) poiché hanno a disposizione una matrice attiva sulla quale giacciono dei transistor e dei condensatori che mantengono la carica in prossimità del pixel e migliorano la nitidezza dell'immagine, nonché la risoluzione. Quest'ultimo parametro dipende soprattutto dal numero di pixel, che a sua volta possono essere facilmente pilotati da un microcontrollore per quanto riguarda la luminosità e il colore (concorde allo standard RGB). Un problema non di poco conto quando si desidera comprare un display di questo tipo è che spesso non si conosce il driver che controlla le funzionalità del display e che deve quindi comunicare con il microcontrollore; perciò, è opportuno quando lo si sceglie, assicurarsi che sia presente il nome del driver corrispondente, in modo da poter configurare lo schermo arbitrariamente consultando il data sheet.

Un esempio di driver è l'ILI9341 il quale sarà descritto successivamente durante la presentazione del prototipo e consente di pilotare display aventi risoluzione 320x240. Sebbene questo controllore comunichi tramite una connessione seriale SPI (che sarà ampiamente discussa nella sezione 1.4), esistono altre forme di comunicazione, tra cui quella parallela. La configurazione di quest'ultima avviene richiamando l'indirizzo del registro desiderato o in forma binaria o in forma esadecimale e successivamente indicando il valore effettivo che deve assumere il registro stesso sempre usando un linguaggio binario o esadecimale. Un altro aspetto da considerare quando si setta un registro è inviare dei segnali standard ai pin di connessione del display, i quali sono elencati in seguito:

- RD (read) deve essere settato a 1 in modo da consentire al PIC di inviare segnali al display.
- WR (write) deve essere settato a 0 qualora si desideri scrivere nei registri oppure a 1 una volta terminata la scrittura.

- RS (reset) deve essere posto a 1 nel caso si voglia scrivere un preciso valore su un registro eliminando il precedente, oppure a 0 per selezionare un particolare registro.
- CS (chip select) deve essere posto a 0 per abilitare il driver del display alla configurazione e successivamente tornare a 1 una volta terminata la definizione del registro corrispondente.

#### Esempio 1: Configurazione registro da collegamento in parallelo

```

LCD_CS = 1;
LCD_RST = 1;
LCD_RS = 1;
LCD_RD = 1
LCD_WR = 1;
LCD_CS = 0;
Send_Command_8bit(0xB1); //Considero il registro B1 che gestisce il frame rate del display
Send_Data_8bit(0xD0); //Impongo il primo valore in esadecimale che corrisponde a 91,15Hz di frame rate
Send_Data_8bit(0x11); //Impongo secondo valore in esadecimale che corrisponde al clock
LCD_CS = 1;
__delay_ms(10);

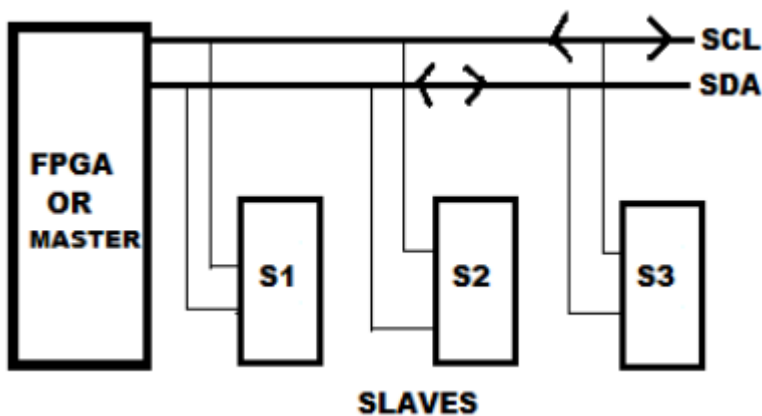
```

Oltre ai pin sopra citati occorre connettere i due fili dell'alimentazione e le porte dati deputati a trasmettere l'informazione dal PIC al display e viceversa. Il vantaggio è di avere a disposizione una velocità di trasmissione dei dati superiore rispetto ad una connessione seriale. Tuttavia, poiché le porte del microcontrollore sono limitate, una comunicazione parallela ha lo svantaggio di occupare un numero elevato di pin del PIC, per cui potrebbe essere un problema per progetti che necessitano di configurare più periferiche contemporaneamente. Occorre introdurre per questo motivo come alternativa due interfacce seriali, ovvero quella SPI e quella I2C.

### 1.3 Interfaccia I2C

La comunicazione I2C consiste in una tipologia di connessione seriale nella quale vengono usati solamente due fili, ovvero il Serial Clock (SCL) ed il Serial Data (SDA) e permette di connettere al microcontrollore molte periferiche esterne tra cui display, sensori o dispositivi biometrici garantendo affidabilità, convenienza economica, occupazione di pochi pin e buone prestazioni.

Fig.2 Schema a blocchi protocollo I2C



In particolare, i collegamenti sono “half duplex”, ovvero il microcontrollore (master) e le periferiche (slaves) possono comportarsi entrambi sia come ricevitori sia come emettitori con il limite però di non inviare i dati in entrambe le direzioni contemporaneamente. Esiste un iter specifico secondo cui i dati sono inviati per completare con successo una comunicazione I2C:

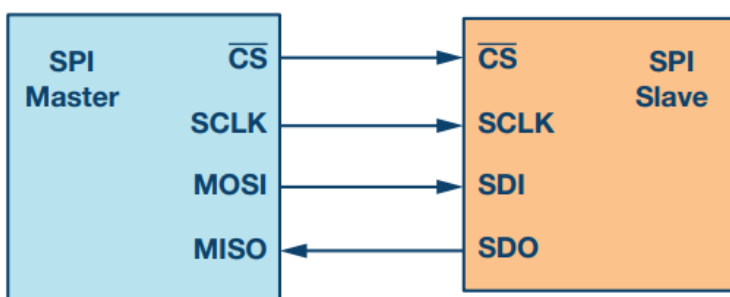
- Inizializzazione: occorre innanzitutto inviare un segnale di START per informare lo slave dell'imminente arrivo di un segnale dal master. Questo comando può essere inviato ponendo inizialmente SDA uguale a 0 e successivamente anche il SCK con valore logico basso. È fondamentale poi inviare un segnale di STOP per informare le altre periferiche che il bus è libero per inviare i dati. Questo comando invece si attua ponendo inizialmente il SDA alto e successivamente il SCK dello stesso livello logico.
- Trasmissione del byte allo slave: una volta terminata l'inizializzazione, un byte viene inviato dal master allo slave per specificare l'indirizzo della periferica a cui trasmettere i dati e la modalità dell'operazione.

- Ricezione del byte dallo slave: riconosciuto il byte in ingresso proveniente dal master, la periferica selezionata è pronta a ricevere i dati.
- Riconoscimento dei dati dallo slave: quando si invia un byte per selezionare un registro o imporre dei valori allo stesso, lo slave verifica la compatibilità del comando e il successo dell'operazione inviando al master un segnale di corretta ricezione.
- Riconoscimento dei dati dal master: il master rimane in ascolto fin quando non arriva il segnale di corretta ricezione e qualora questo non arrivasse il master interrompe la comunicazione. [1]

#### 1.4 Interfaccia SPI

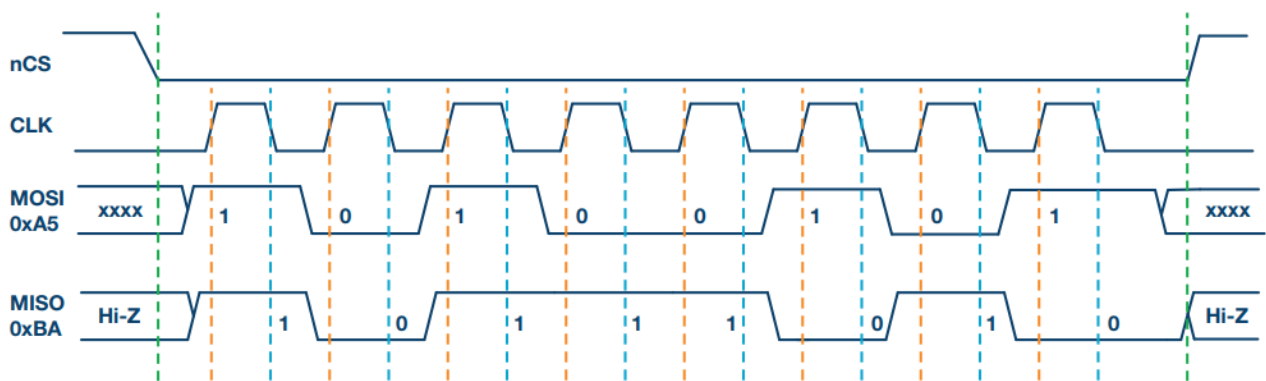
La comunicazione SPI è più veloce rispetto all'I2C e consente di comunicare con periferiche come display, sensori, memorie esterne sfruttando quattro fili di connessione. Questi sono “full duplex”, ovvero è possibile uno scambio bidirezionale dei dati in maniera simultanea, e sono in particolare il Clock (SCK) che permette di sincronizzare le istruzioni e la scrittura dei dati, il Chip Select (CS) che permette di abilitare la comunicazione con la periferica ed è un attivo basso (si inizializza la comunicazione quando è uguale a 0 e si interrompe quando assume un valore pari a 1), il Master output Slave input (MOSI) che abilita la trasmissione dei dati dal microcontrollore alla periferica e infine il Master input Slave output che definisce l'invio dei dati dalla periferica al PIC.

Fig.3 Schema a blocchi interfaccia SPI



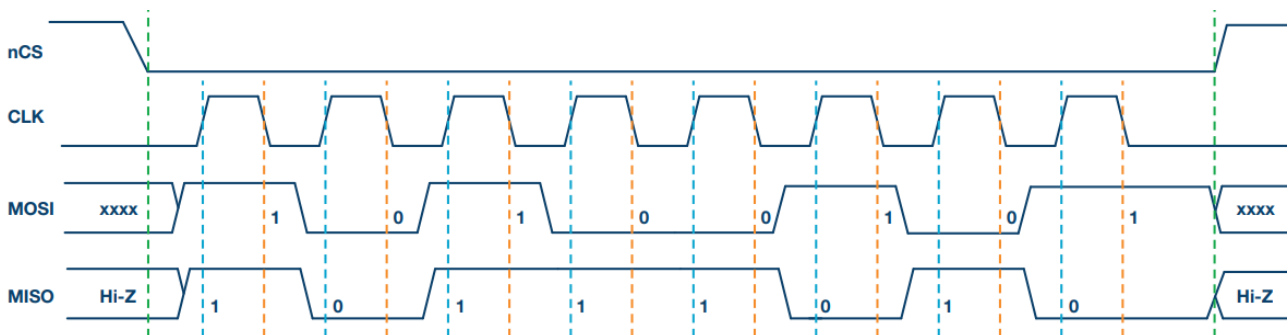
Prima di procedere con la configurazione bisogna introdurre altri due parametri ovvero il Clock Polarity (CPOL) e il Clock Phase (CPHA): il primo permette di definire a partire da quale valore logico del clock il CS può avviare la comunicazione; il secondo permette di stabilire rispetto a quale fronte d'onda del clock avviene il campionamento e lo scorrimento dei dati. Esistono in totale quattro possibili comunicazioni che si implementano scegliendo una delle quattro modalità disponibili nella configurazione SPI.

Fig.4 Mode 0



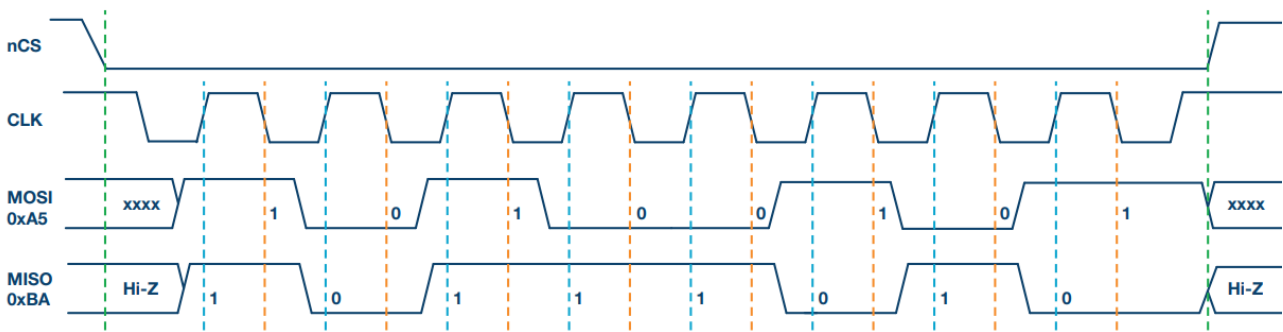
In questa modalità CPOL=0 dato che CS si attiva quando CLK=0; CPHA=0 poiché i dati sono campionati nel fronte di salita del CLK e shiftati in quello di discesa.

Fig.5 Mode 1



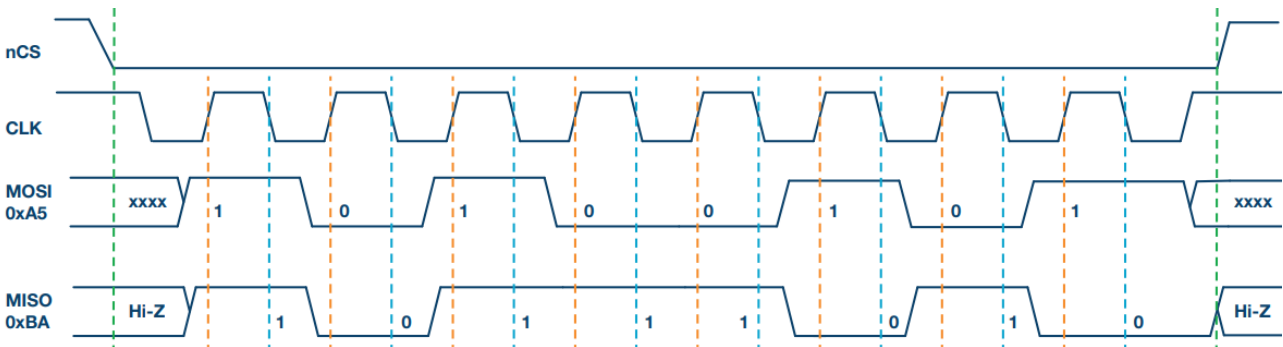
In questa modalità CPOL=0 dato che CS si attiva quando CLK=0; CPHA=1 poiché i dati sono campionati nel fronte di discesa del CLK e shiftati in quello di salita

Fig.6 Mode 2



In questa modalità CPOL=1 dato che il CS si attiva quando il CLK=1; CPHA=1 poiché i dati sono campionati nel fronte di discesa del CLK e shiftati in quello di salita.

Fig.7 Mode 3



In questa modalità CPOL=1 dato che il CS si attiva quando il CLK=1; CPHA=0 poiché i dati sono campionati nel fronte di salita del CLK e shiftati in quello di discesa. [2]

Così come per una comunicazione parallela, anche in un interfacciamento SPI occorre inviare dei segnali predefiniti sui pin della periferica per settare i registri. Se ad esempio consideriamo un display gestito dal driver ILI9341 occorre inviare i seguenti segnali, di cui i primi due costituiscono ulteriori pin di collegamento tra display e microcontrollore volti alla trasmissione dei dati.

- DC deve essere posto a 0 per selezionare un registro oppure ad 1 per imporre un valore preciso nel registro stesso.
- RST deve essere posto a 1 per poter scrivere nuovi dati.

- CS deve essere posto a 0 per abilitare il driver del display alla configurazione e successivamente tornare a 1 una volta terminata la definizione del registro corrispondente.

Esempio: configurazione registro da collegamento SPI

```
void sendCommand(uint8_t cmd) {           //Inizializzo una funzione con cui possa selezionare registro
    TFT_DC = 0;                          //quando DC è a 0 seleziono registro
    SPI1_WriteByte(cmd);                  //Scrivo byte per la selezione univoca del registro
    TFT_DC = 1;                          //quando DC è a 1 scrivo un dato
}

void sendData(uint8_t cmd) {             //Inizializzo una funzione con cui scrivere valore nel registro
    TFT_DC = 1;                          //quando DC è a 1 scrivo un dato
    SPI1_WriteByte(cmd);                  //Scrivo byte per imporre valore nel registro
}
```

Inizializzate le due funzioni sopra rappresentate possiamo settare i registri desiderati, in questo caso la luminosità.

```
TFT_RST = 1;                            //Porre RST=1 per scrivere nuovi dati
TFT_CS = 0;                             //quando CS è pari a 0 eseguo configurazione
sendCommand(0x51);                       //Selezione registro di luminosità
sendData(0xFF);                          //Impongo valore massimo di luminosità
TFT_CS = 1;                              //quando CS è pari a 1 termino configurazione
__delay_ms(10);
```

## 1.5 Touch screen

Una delle applicazioni più diffuse che ha permesso di migliorare l'interazione dell'uomo con gli schermi è sicuramente il touch screen. Molti display in commercio sono dotati di questa opzione, tuttavia bisogna assicurarsi che gli stessi dispongano del driver deputato al controllo del touch. Un esempio di controllore è XPT2046 il quale modulo comunica con una connessione SPI e dispone di 5 pin di collegamento, ovvero i 4 tipici della comunicazione seriale di tipo SPI e uno chiamato IRQ necessario per gestire l'interrupt: quando il pennino del touch screen tocca il display, a prescindere dall'azione svolta dal microcontrollore, questa viene interrotta in modo da prestare attenzione

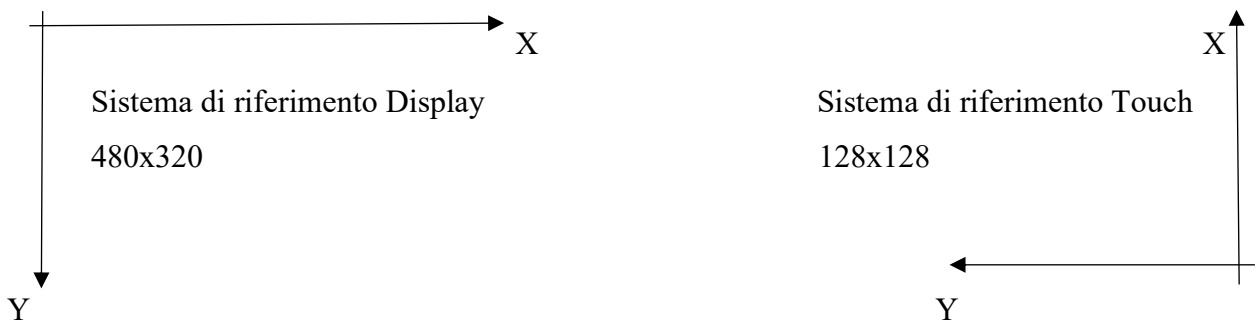
esclusivamente al touch. Per avere un corretto riconoscimento del segnale proveniente dal tocco del pennino sullo schermo, sono necessari 24 cicli di clock: in conformità con il data sheet, i primi 8 cicli servono per completare l'acquisizione ed inviare i dati al controller; i successivi 12 cicli consentono la conversione A/D del segnale di input in modo da poter risalire non solo alle coordinate X e Y dello schermo ma ad altre due variabili Z1 e Z2 associate alla pressione esercitata dal pennino; infine gli ultimi 4 cicli permettono di terminare l'azione con l'aggiunta di zeri ignorati dal controllore. Per interagire con il touch, oltre a valutare i periodi temporali di trasmissione del segnale, occorre considerare anche i dati inviati: in particolare, si trasmette un "control byte" il quale può essere scomposto nel seguente modo a partire dal MSB (Most significant bit) fino al LSB (Less significant bit).

- "Start bit": il primo bit viene settato a 1 per iniziare la lettura del touch screen.
- "Addressing bits": i 3 bit successivi sono digitati per selezionare la variabile che si vuole ottenere in uscita dal controllore tra le 4 presenti, X, Y, Z1, Z2.
- "Mode bit": questo valore deve essere basso per consentire una conversione a 12 bit mentre deve essere alto per averla a 8 bit. La prima permette di avere una migliore risoluzione a discapito della velocità di trasmissione del segnale, viceversa la seconda ha una peggior precisione aumentando però la velocità.
- "Control bits": gli ultimi 3 bit servono per stabilire particolari controlli relativi all'elaborazione del byte.

Il control byte viene salvato su una memoria temporanea ovvero un buffer, che a sua volta può essere letto per risalire alle coordinate X e Y o Z1, quest'ultima usata per stabilire la pressione (Z2 può essere escluso tra i parametri utilizzati). In realtà Z1 è cruciale per il funzionamento del touch screen poiché consente di stabilire una soglia al di sotto della quale il display ignora eventuali comandi: a causa dell'errore dovuto al rumore di fondo o a causa di disturbi esterni si possono infatti generare dei comandi touch screen indesiderati. Allora, stabilire una pressione



minima al di sotto della quale non si registrano attività, costituisce un'ottima soluzione per risolvere il problema e inviare solamente comandi esterni desiderati. Un altro problema associato al touch screen è dovuto al fatto che le coordinate X e Y relative al tocco del pennino si riferiscono ad un sistema di riferimento diverso rispetto a quello volto alla scrittura sul display. Inoltre, qualora sia utilizzata una conversione ad 8 bit per il touch, ci sarebbe una risoluzione di 128 livelli sia per l'asse X sia per l'asse Y e quindi, considerando ad esempio un display TFT 480x320, si incorrerà non solo a una disomogeneità per quanto riguarda la direzione degli assi ma anche per la differente risoluzione. Possiamo riassumere i seguenti concetti valutando gli schemi seguenti.



Nell'appendice è possibile trovare un listato di programmazione sviluppato per risolvere questi due problemi, nonché per consentire il funzionamento del touch. [A1]

## 1.6 Periferiche

Oltre ai display un microcontrollore può interfacciarsi con molti altri dispositivi esterni soddisfacendo particolari protocolli di comunicazione come SPI, I2C ecc. Esempi di periferiche con cui instaurare un collegamento sono ad esempio sensori di vario tipo, il convertitore analogico/digitale esterno, le memorie esterne ecc.

Per quanto riguarda i primi, ne esistono di moltissime tipologie differenti e ci permettono di misurare una particolare grandezza fisica tra le tante esistenti: spazio, velocità, accelerazione, campo magnetico, forza, pressione, temperatura, umidità, deformazione, suono ecc.

In base al progetto che si intende sviluppare, è fondamentale scegliere il sensore prestando molta attenzione alle caratteristiche statiche e dinamiche dello stesso, nonché alla spesa economica al fine di trovare il miglior compromesso sul rapporto qualità/prezzo. Tra le caratteristiche statiche da tener conto ci sono il campo di misura e quindi il fondo scala che determina il range di misurazione dello strumento; la sensibilità che è definita come il rapporto tra il segnale di output del sensore e quello di input, coincide con il coefficiente angolare della curva di taratura e si auspica sia lineare per limitare al minimo l'errore; la risoluzione che coincide con la più piccola variazione misurabile dalla grandezza in ingresso; la soglia ovvero il più piccolo ingresso misurabile con il sensore ed il range di temperatura al di fuori della quale lo strumento potrebbe avere delle anomalie. Inoltre, è necessario valutare la banda passante, il tempo di assestamento, la stabilità e l'accuratezza.

I moduli di conversione A/D sono periferiche esterne che permettono di compiere una prima elaborazione del segnale acquisito, in particolare si procede con un campionamento iniziale che avviene ad una precisa frequenza e con una quantizzazione dei valori in esame. Maggiore sarà il numero di bit deputati alla conversione A/D e maggiore sarà la precisione dei dati digitali: ad esempio una conversione ad 8 bit consente di avere 256 livelli di quantizzazione disponibili, mentre una a 10 bit permette di avere ben 1024 livelli e di conseguenza risulta essere molto più precisa della prima. Attualmente, gran parte dei microcontrollori in commercio dispongono di convertitori A/D integrati che permettono di acquisire un segnale analogico direttamente sui pin del PIC che sono stati progettati per questo tipo di acquisizione (nel datasheet è possibile visualizzare quali pin possono ricevere ingressi analogici e spesso sono contrassegnati con la sigla AN). Tuttavia, in fase di programmazione per raggiungere tale scopo occorre portare a termine due passaggi fondamentali:

- Configurazione dei registri per la conversione analogica-digitale con lo scopo di definire le modalità con cui si esegue il processo. Con i microcontrollori PIC più avanzati, ad esempio il PIC18F47Q10 (il quale sarà utilizzato nella realizzazione del prototipo), occorre configurare ben 19 registri inizialmente, mentre per quelli più semplici sono sufficienti solo alcuni di

questi. In genere durante la configurazione dei registri occorre selezionare la porta di acquisizione, stabilire le tensioni di riferimento, scegliere la frequenza di campionamento, definire il tempo di attesa tra l'acquisizione e la conversione, decidere il tipo di giustificazione e infine come memorizzare i dati appena digitalizzati.

- Una volta configurati i registri si può avviare la conversione salvando i dati sui registri di uscita. Da un punto di vista hardware occorre aspettare un certo periodo di tempo per completare il processo, per questo motivo è fondamentale l'impiego di un condensatore che possa mantenere la tensione di input. Per conservare correttamente la carica si apre un interruttore a monte (generalmente un mosfet) il quale a sua volta è caratterizzato da un'alta impedenza di ingresso. La frequenza con cui si apre e si chiude l'interruttore determina la frequenza di campionamento.

Per quanto riguarda il PIC18F47Q10 in appendice sono implementati gli algoritmi per risolvere entrambi i passaggi citati sopra. [A2]

## Capitolo 2

### Nozioni fondamentali

#### 2.1 Risonanza magnetica nucleare

Tale trattazione si pone lo scopo di presentare un prototipo il cui lavoro consiste nell'effettuare controlli in ambienti di risonanza magnetica. Prima di spiegare l'importanza di questa attività, è necessario illustrare innanzitutto, il funzionamento di una risonanza magnetica nucleare e il conseguente inquinamento elettromagnetico prodotto dalla stessa. La NMR (Nuclear Magnetic Resonance) anche chiamato in ambito clinico MRI (Magnetic Resonance Imaging), è uno strumento biomedicale molto comune negli ambienti ospedalieri, che consente di riprodurre immagini del nostro corpo al fine di diagnosticare una malattia come, ad esempio, un tumore oppure monitorare un trattamento medico. Sebbene esista anche l'applicazione chimica che si pone lo scopo di studiare la struttura molecolare di un campione, la MRI costituisce uno dei metodi più avanzati di diagnostica attualmente in circolazione e pone le sue radici nel 1937 quando il professor Isaac Rabi presso il suo laboratorio fu in grado di osservare per primo il fenomeno della risonanza, cosa che gli valse il premio Nobel per la fisica nel 1944. Grazie alle nuove teorie della meccanica quantistica e grazie al lavoro di molti scienziati tra cui lo stesso Rabi, i decenni successivi furono cruciali per lo sviluppo di questa tecnica spettroscopica che iniziò ad avere implicazioni cliniche solamente dal 1970. [3]

Il principio fisico su cui si basa tale tecnica consiste nella capacità di alcuni nuclei atomici di assorbire onde elettromagnetiche provenienti dall'esterno. In particolare, l'apparecchio biomedicale presenta un magnete permanente o un elettromagnete a superconduttore che genera un campo magnetico statico molto forte (fino a 3T) volto a interagire con i nuclei atomici dei tessuti biologici. Così come gli elettroni possiedono un moto di spin, allo stesso modo anche le particelle subnucleari ruotano intorno ai propri assi determinando un preciso vettore di magnetizzazione. Inizialmente, questi ultimi godono di una direzione casuale, tuttavia in presenza di un campo magnetico esterno, sono ammissibili solo due particolari configurazioni: uno spin parallelo al campo esterno con conseguente

minimizzazione dell'energia oppure uno spin antiparallelo che comporta un'energia maggiore. Generalmente compaiono più i primi, in modo da garantire una maggior stabilità del sistema e da un punto di vista quantitativo si può dire che all'aumentare del modulo del campo magnetico esterno, tende ad aumentare anche il numero dei nuclei con spin parallelo. Oltre a modificare la direzione del vettore magnetico, si induce un moto di precessione intorno alla direzione del campo magnetico esterno (come il moto di una trottola), tuttavia la rotazione risulta sfasata rispetto agli altri nuclei adiacenti.

Fig.8 Precessione del nucleo con spin parallelo e opposto

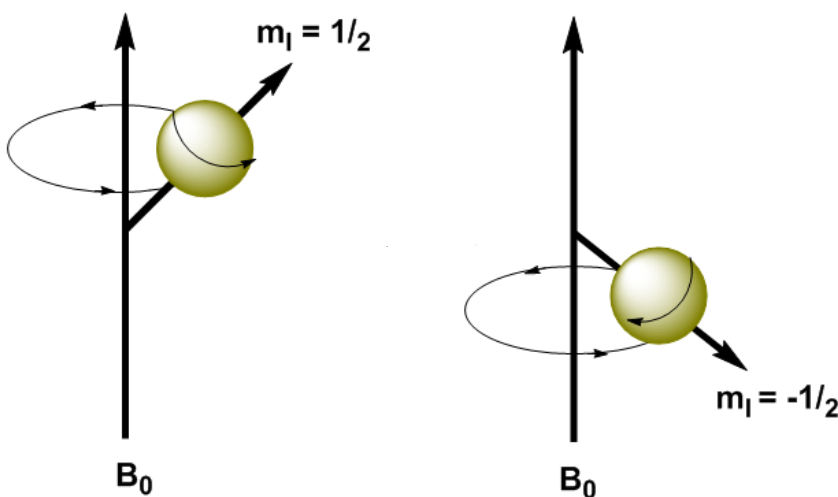
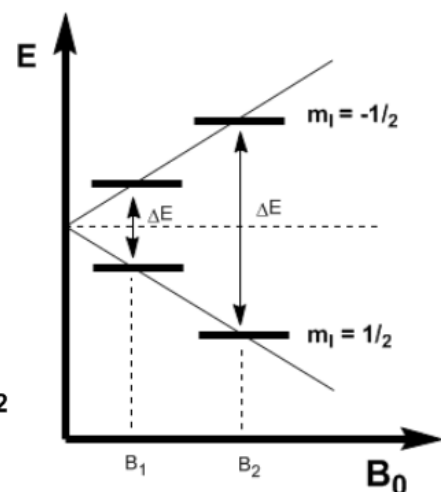


Fig.9 Livelli energetici



Sottoponendo una radiazione elettromagnetica alla frequenza di Larmor, ovvero pari a quella del moto di precessione (nell'ordine dei Mhz e quindi nel range delle radiofrequenze), innanzitutto avviene una sincronizzazione in fase dei moti rotazionali dei nuclei e successivamente il vettore di magnetizzazione tende a piegarsi di un angolo sempre maggiore fino a disporsi trasversalmente rispetto al campo magnetico esterno. Supponendo quindi che quest'ultimo coincida con l'asse z di un sistema di riferimento cartesiano, allora il vettore di magnetizzazione dopo un preciso arco temporale giacerà sul piano xy compiendo un angolo di  $90^\circ$ ; non a caso la radiazione che consente tale configurazione è anche chiamata anche "impulso di  $90^\circ$ ". Esiste anche la possibilità di fornire un'intensità o una durata doppia rispetto alla prima generando in questo caso un "impulso di  $180^\circ$ ". Questo metodo consente di avere la cosiddetta risonanza, ovvero il fenomeno per cui il segnale in

uscita risulta essere amplificato in modulo e quindi maggiormente predisposto a successive elaborazioni. Cessato l'impulso in radiofrequenza si ha in primo luogo, una desincronizzazione del moto di precessione dei nuclei e contemporaneamente, un progressivo riallineamento del vettore di magnetizzazione con il campo magnetico esterno con annesso rilascio di energia elettromagnetica. Grazie a dei rilevatori chiamati "bobine di gradiente", è possibile ricevere un segnale di uscita, il quale spettro presenta molte frequenze diverse. Il motivo per cui avviene tale configurazione è che tessuti differenti i quali sono caratterizzati da nuclei differenti entrano in risonanza con impulsi diversi. Allora, una volta applicata la trasformata di Fourier che ricostruisce lo spettro in frequenza del segnale si può applicare un opportuno algoritmo per ricostruire l'immagine, delimitare i confini dei tessuti adiacenti e allocare i singoli elementi nella posizione corretta.

## **2.2 Contesto e normative sull'inquinamento elettromagnetico**

Sebbene la MRI sia diventato uno strumento diagnostico molto comune nei centri ospedalieri, questa risulta essere una fonte non trascurabile di inquinamento elettromagnetico. A maggior ragione negli ospedali, dove risiedono persone fragili, è necessario valutare i rischi associati a una eccessiva esposizione a radiazioni esterne. Con l'espressione "inquinamento elettromagnetico" si intende la produzione artificiale di campi elettromagnetici e in particolare la diffusione di radiazioni non ionizzanti classificabili in onde a bassa frequenza (da 0 a 300Hz) e quelle a radiofrequenza (da 300Hz a 300GHz). Sebbene non esista alcuna documentazione che attesti la reale pericolosità delle radiazioni a lungo termine, la comunità scientifica è spaccata tra coloro che argomentano rischi non trascurabili per l'esposizione prolungata a tale inquinamento e coloro che non lo considerano un problema. Sono infatti nate due particolari visioni sul tema: la posizione "conservativa" ritiene gli effetti a breve termine (ustioni, emorragie, necrosi) quelli più pericolosi e quindi sostiene di porre dei limiti legislativi per minimizzare i rischi; la posizione "cautelativa" presta pari attenzione agli effetti biologici acuti così come a quelli cronici, i quali possono insorgere a distanza di diversi anni (effetti

mutageni e quindi cancerogeni). Di fronte a tale problema la seconda visione, oltre a stabilire dei limiti da rispettare per legge, invita ad avere cautela nella continua installazione di antenne o altri strumenti trasmissivi, perlomeno fino a quando non ci saranno documenti scientifici che dimostrino la reale pericolosità delle radiazioni. [5] Le istituzioni stanno sempre più sposando il secondo fronte tanto che il WHO (World Health Organization) ha dichiarato che l'inquinamento elettromagnetico costituisce una delle quattro principali criticità che occorre risolvere nel nuovo millennio. L'IARC (International Agency for Research on Cancer) ha inserito le radiazioni non ionizzanti sia in bassa sia in alta frequenza nella categoria 2B, la quale comprende tutti gli agenti possibilmente cancerogeni per l'uomo. Il sistema giuridico italiano impone che lo Stato abbia il compito di delineare i limiti di esposizione ai campi elettromagnetici specialmente nei luoghi di maggior intensità, di misurare le radiazioni generate e di memorizzare tutte le sorgenti di onde installate nel Paese grazie ad un catasto nazionale.

Per quanto concerne i forti campi magnetici generati dai macchinari di risonanza magnetica, in Italia è avvenuta l'attuazione della direttiva 2013/35/UE riguardante le disposizioni in materia di salute e sicurezza per i lavoratori esposti a campi elettromagnetici, grazie alla pubblicazione in Gazzetta Ufficiale del decreto legislativo 1 agosto 2016, n. 159 che modifica il Dlgs 81/08.

Tale direttiva europea che abroga quella precedente 2004/40/CE obbliga il datore di lavoro ad effettuare una valutazione dei rischi provocati dai campi elettromagnetici e definisce due particolari soglie da rispettare:

- "VLE (Valori limite di esposizione) relativi agli effetti sanitari", ovvero le soglie "al di sopra dei quali i lavoratori potrebbero essere soggetti a effetti nocivi per la salute, quali il riscaldamento termico o la stimolazione del tessuto nervoso o muscolare".
- "VLE (Valori limite di esposizione) relativi agli effetti sensoriali", ovvero le soglie "al di sopra dei quali i lavoratori potrebbero essere soggetti a disturbi transitori delle percezioni sensoriali e a modifiche minori nelle funzioni cerebrali".

Per quanto riguarda frequenze inferiori a 1Hz si possono vedere i limiti di esposizione ai campi magnetici dalla tabella 1.

	<b>VLE relativi agli effetti sensoriali [T]</b>
Condizioni di lavoro normali	2
Esposizione localizzata degli arti	8
	<b>VLE relativi agli effetti sanitari [T]</b>
Condizioni di lavoro controllate	8

Tab.1 Limiti di esposizione valutati per campi magnetici

Con riferimento invece ai campi elettrici indotti nel corpo umano a causa di onde elettromagnetiche con frequenze superiori a 1Hz si osservi la tabella.

<b>Intervallo di frequenza</b>	<b>VLE relativi agli effetti sanitari [<math>Vm^{-1}</math>] (valori di picco)</b>
$1Hz \leq f < 3kHz$	1.1
$3kHz \leq f \leq 10MHz$	$3.8 \times 10^{-4} f$
<b>Intervallo di frequenza</b>	<b>VLE relativi agli effetti sensoriali [<math>Vm^{-1}</math>] (valori di picco)</b>
$1Hz \leq f < 10Hz$	$0.7/f$
$10Hz \leq f < 25Hz$	0.07
$25Hz \leq f \leq 400Hz$	$0.0028f$

Tab.2 Limiti di esposizione valutati per campi elettrici indotti

Generalmente i campi elettrici sono indotti a causa del movimento dell'operatore sanitario all'interno della sala di risonanza magnetica dove di solito il campo non è mai uniforme oppure a causa dell'accensione delle "bobine di gradiente" (i rilevatori della MRI) fondamentali per l'acquisizione del segnale nel processo di imaging.



### **2.3 Effetti dell'inquinamento elettromagnetico**

Una tale premura da parte delle istituzioni deriva dai risultati di alcuni studi epidemiologici che assegnano una probabilità non indifferente di effetti permanenti sull'uomo. Possiamo in particolare distinguere effetti a breve termine, i quali sono facilmente osservabili e quelli a lungo termine che sono più difficili da studiare e necessitano di indagini approfondite e durature per dimostrare la loro presenza. Per quanto riguarda i primi, esistono innanzitutto gli effetti faradici dovuti a radiazioni con frequenze inferiori a 1kHz che provocano contrazioni muscolari indesiderate a causa dell'interazione con cellule elettricamente eccitabili; in secondo luogo, compaiono effetti termici che comportano il riscaldamento cellulare. Quest'ultimo effetto è facilmente osservabile, ad esempio, quando si effettua una chiamata al telefono per un periodo prolungato e con il passare del tempo si avverte una sensazione di calore crescente sull'orecchio. Gli effetti cronici, ovvero quelli che si manifestano a distanza di anni, secondo alcuni studi epidemiologici potrebbero sfociare in mutazioni geniche e quindi nel crescente rischio di sviluppare un tumore. A livello cellulare infatti, si è notato un'alterazione dei meccanismi di riparazione del DNA, un cambiamento del trasporto ionico transmembrana e una maggior formazione di radicali, specie reattive dell'ossigeno che interferiscono sempre con il DNA. Nelle sale di MRI si manifestano dei campi magnetici molto intensi che possono essere statici o dinamici a causa sia del magnete permanente sia dell'impulso a radiofrequenza generato per ottenere il fenomeno di risonanza. I primi modificano l'orientamento delle molecole biologiche che costituiscono l'organismo e provocano degli effetti elettrolitici, ovvero mutano la disposizione degli ioni all'interno e all'esterno della cellula. Nel caso siano presenti dispositivi impiantati quali pacemaker o defibrillatori interni c'è un forte rischio che le linee di campo possano interagire con questi e provocare dei malfunzionamenti con conseguente possibilità di fibrillazione. La FDA (Food and Drug Administration) ha infatti stabilito il limite di massima esposizione al campo magnetico pari a 5G per i portatori di tali dispositivi. Inoltre, ci possono essere anche effetti indiretti che riguardano oggetti metallici ad esempio protesi o impianti innestati nel paziente o qualsiasi altro

utensile presente nella stanza; questi infatti, possono interagire con il campo magnetico statico esterno, acquistando velocità e portando il paziente ad avere anche seri infortuni. In particolare, da un punto di vista normativo si considera la linea isomagnetica pari a 0.5T come soglia oltre alla quale esiste tale rischio. È opportuno quindi, prima di entrare all'interno di sale di MRI, dove è possibile sfiorare tale limite, verificare che tutti gli oggetti pericolosi siano stati tolti ed occorre utilizzare dei dispositivi che possano verificare l'intensità del campo magnetico e scoprire la presenza di questi oggetti. I campi magnetici variabili nel tempo in accordo con la legge di Faraday possono indurre delle correnti nei tessuti provocando effetti termici indesiderati, anomalie nella conduzione neuromuscolare o l'apparizione di effetti luminosi sulla retina. Radiazioni ad alta frequenza come quelle generate dagli impulsi alla frequenza di Larmor, possono causare disturbi metabolici e nervosi a causa dell'aumento della temperatura; il SAR (Specific Absorption Rate) è un parametro che suggerisce gli effetti termici subiti dai tessuti biologici a causa delle onde elettromagnetiche e si misura in W/kg. Si possono osservare le prime anomalie dell'organismo quando la temperatura sale di 1°C, ovvero con un assorbimento di 2W/kg, mentre con un SAR di 10W/kg si rischia di avere conseguenze irreversibili. Inoltre, una prolungata esposizione a radiazioni esterne ad alte frequenze può limitare la motilità degli spermatozoi con conseguente riduzione della fertilità, può provocare alterazioni dell'attività enzimatica, della concentrazione di calcio all'interno e all'esterno della cellula, delle proteine di membrana e dei trasporti ionici. Sebbene vi sia anche la componente psicologica, alcuni soggetti particolarmente sensibili all'inquinamento elettromagnetico soffrono dolori, fastidi, depressione, mal di testa, insonnia e in alcuni casi anche convulsioni e crisi epilettiche.

[6]

#### **2.4 Logistica e interferenze**

Dati i seri problemi di salute dovuto all'esposizione ai campi elettromagnetici prodotti da una risonanza magnetica nucleare è fondamentale delineare da un punto di vista logistico sale ad accesso programmato:

- Zona di pericolo: a causa dei campi magnetici dispersi dal macchinario si possono avere linee di campo che toccano i 10mT e di conseguenza impongono operatori sanitari e pazienti a giacere nella suddetta stanza solo per l'attuazione della diagnosi e non per un tempo superiore. Inoltre, occorre evitare di portare oggetti metallici che possono diventare dei veri e propri proiettili con campi magnetici così intensi.
- Zona di accesso controllato: si riferisce alle aree adiacenti in cui si può riscontrare un campo di 0,5mT, ovvero 5G coincidente con il limite di tolleranza per portatori di dispositivi impiantabili. All'interno di queste stanze, oggetti magnetici possono interagire mentre circuiti elettronici tendono ad avere dei guasti, per cui è necessario inserire delle schermature che possano proteggerli da malfunzionamenti.
- Zone di rispetto: tutte le aree che presentano un campo da 0,1 a 0,5mT. I corridoi e gli ingressi di ospedale che conducono alle sale di risonanza magnetica devono appartenere a questa categoria.

Inoltre gli strumenti ospedalieri sono molto sensibili, in quanto devono rilevare segnali biologici la cui intensità in genere è molto modesta e di conseguenza sono spesso oggetto di interferenze causate da radiazioni prodotte da altra strumentazione o da impianti di telecomunicazione. Stanze in cui risiedono sistemi di risonanza magnetica nucleare non fanno eccezione, data la generazione di campi molto intensi, per cui è fondamentale valutare i possibili rischi ad essi associati. Oltre all'esposizione alle radiazioni non ionizzanti, è necessario osservare i cosiddetti EMI (ElectroMagnetic Interference) ovvero i disturbi elettromagnetici che possono provocare anomalie, malfunzionamenti o guasti ad altri dispositivi, nonché situazioni di pericolo per i pazienti coinvolti. Nell'ambito della risonanza magnetica, tali disturbi possono essere provocati dallo stesso macchinario verso altre apparecchiature ospedaliere e viceversa. I disturbi che bisogna tenere in considerazione sono i seguenti:

- Linee di distribuzione di energia elettrica che erogano corrente a 50Hz (nel nord America 60Hz).

- Interferenze di cariche elettrostatiche che sono a frequenza nulla e nascono a causa dell'accoppiamento di un campo elettrico esterno con un circuito.
- Scariche elettrostatiche dovuto all'accumulo di cariche opposte in poli diversi.
- Interferenze a radiofrequenza causato da dispositivi per diatermia, risonanze magnetiche, sistemi di comunicazione radio, wireless e bluetooth, sistemi di telefonia cellulare.
- Interferenze causate da altre apparecchiature biomedicali che essendo possessori di circuiti elettrici generano campi elettromagnetici che possono interferire con altri macchinari.

Le interferenze si suddividono in condotte e irradiate: le prime si propagano via cavo attraverso le connessioni con altri dispositivi; le seconde si manifestano senza l'ausilio di fili ma tramite onde elettromagnetiche. Inoltre, Poiché spesso un disturbo su una apparecchiatura ospedaliera è frutto della somma di più disturbi che interagiscono con la stessa, è difficile spesso stabilire la natura eziologica del problema e quindi distinguere le diverse fonti di rumore elettromagnetico.[7]

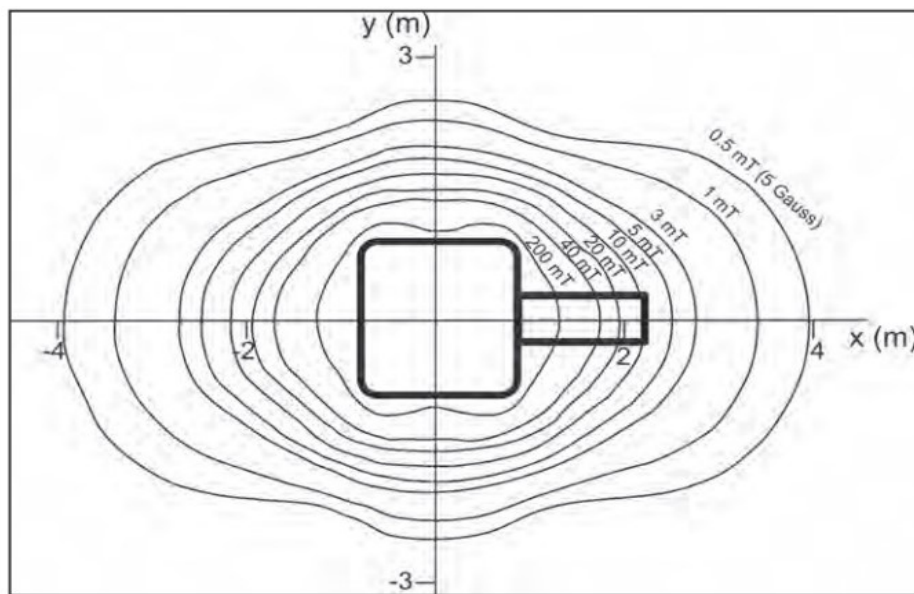
Nel decreto ministeriale 02/08/1991 sono specificati valori limite oltre le quali particolari apparecchiature riscontrano dei malfunzionamenti come mostrato nella tabella.

<b>Apparecchiature e supporti di dati</b>	<b>Valore limite di campo magnetico</b>
Video-terminali, dischi magnetici, carte di credito	2-3mT
Unità disco, tubi radiogeni, apparecchi ad ultrasuoni	1mT
pacemaker cardiaco, camere multiformato	0.5mT
Intensificatori di immagini a raggi X, gamma-camere, TAC, acceleratori lineari, microscopi elettronici	0.1mT

Tab.3 Valori massimi di tolleranza ad interferenze

La MRI è la fonte che sviluppa il tipo di interferenza più intensa per cui è importante valutare le linee di campo magnetico prodotte con gli annessi valori di intensità per stabilire la disposizione logistica di altre apparecchiature elettroniche.

Fig.10 Linee di campo magnetico prodotte da una MRI

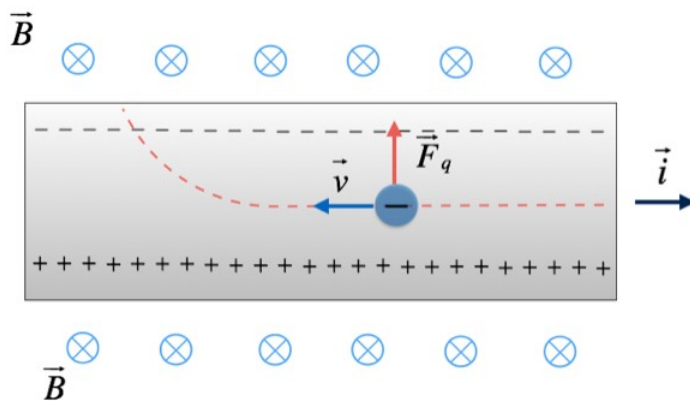


In figura sono rappresentate le linee di campo prodotte della MRI “Avanto” prodotta dalla Siemens che ha un magnete di 1.5T. Date le specifiche dei dispositivi elettronici relative alla tolleranza a sopportare intensi campi magnetici esterni, si può determinare una posizione in cui possano compiere correttamente le loro funzioni: ad esempio display, televisori possono continuare a funzionare solo se compare un campo magnetico inferiore a 1G; addirittura a 50G si possono smagnetizzare hard disk, carte di credito. Si ricorda di nuovo il limite di 5G oltre la quale persone non autorizzate non possono entrare e che risulta la soglia per portatori di pacemakers. Qualora non sia possibile mantenere le distanze ottimali, occorre utilizzare delle schermature contro campi magnetici. Sebbene il piombo sia largamente utilizzato in ambito sanitario, ad esempio, per schermare apparecchiature a raggi X, tale materiale ha una scarsa permeabilità magnetica e di conseguenza scarsa efficienza nell'attività di shielding. Al contrario acciai legati a silicio o carbonio (i primi sono più economici) riescono ad assorbire campi magnetici e fungono da ottimi schermi.

## 2.5 Effetto Hall

La rilevazione dei campi magnetici in prossimità di un macchinario di risonanza magnetica può essere effettuata da sensori che sfruttano l'effetto Hall. Tale fenomeno da un punto di vista fisico consiste nel sottoporre un campo magnetico  $\mathbf{B}$ , entrante nel piano, ad una lamina metallica. A questo punto se impongo una certa corrente sulla lamina grazie ad un generatore di tensione si instaura una differenza di potenziale in senso trasversale rispetto al moto delle cariche a causa della forza di Lorentz. Più nello specifico si osservi la seguente figura.

Fig.11 Effetto Hall



La forza di Lorentz è definita tramite la seguente formula

$$\mathbf{F} = q \cdot \mathbf{v} \times \mathbf{B} \quad [1]$$

Poiché facciamo riferimento ad una corrente elettrica, dobbiamo valutare il movimento degli elettroni sapendo che il valore della carica è  $q = 1,6022 \times 10^{-19} \text{C}$ . Il prodotto vettoriale è risolto usando la regola della mano destra: il pollice sarà rivolto verso il senso opposto al moto degli elettroni essendo la carica negativa (nella figura il dito deve essere rivolto verso destra), l'indice sarà concorde al vettore di campo magnetico  $\mathbf{B}$  entrante nel piano e allora il vettore della forza di Lorentz sarà orientato verso l'alto e avrà come modulo la seguente espressione (essendo  $\mathbf{v}$  e  $\mathbf{B}$  perpendicolari).

$$F = q \cdot v \cdot B \quad [2]$$

Una volta raggiunto l'equilibrio si instaura un campo elettrico  $\mathbf{E}$  tale che il suo modulo è pari a

$$E = \frac{F}{q} \quad [3]$$

Allora supponendo che la distanza tra le cariche addensate nei poli opposti della lamina sia pari a  $d$ , la differenza di potenziale  $\Delta V_h$  assume il seguente valore

$$\Delta V_h = E \cdot d \quad [4]$$

Allora dalla [2], [3], [4] posso sostituire e dire che

$$\Delta V_h = v \cdot B \cdot d \quad [5]$$

## Capitolo 3

### Obbiettivo della tesi

Lo scopo che vuole porsi tale trattazione consiste nella realizzazione di un prototipo che abbia la capacità di rilevare i campi magnetici presenti in una stanza e in particolare in un ambiente di risonanza magnetica nucleare. Il dispositivo infatti, grazie a un sensore che sfrutta l'effetto Hall e un microcontrollore PIC, campiona il segnale in ingresso e proietta lo stesso su un display dopo ulteriori elaborazioni. In particolare, avvenuta l'acquisizione, i dati sono resi informativi calcolando il valor medio, lo zero crossing, la componente continua o alternata e alla fine si analizzano le frequenze grazie ad una FFT. Quest'ultima è fondamentale per capire la natura dei disturbi elettromagnetici presenti nella stanza in quanto si ottiene lo spettro di tutte le possibili interferenze, tra cui quella di rete 50Hz (in nord America 60Hz) o quelle prodotte dagli altri macchinari che possiedono circuiti elettronici. In ambienti di risonanza magnetica questo prototipo potrebbe aiutare gli operatori logistici a delineare le zone di pericolo distinguendole dalle zone ad accesso limitato e da quelle di rispetto e inoltre potrebbe essere molto utile per la sperimentazione di nuove MRI e per verificare il loro inquinamento elettromagnetico. Attualmente infatti, sono in fase di sperimentazione strumenti di risonanza magnetica portatili e a basso costo che consentirebbero di diagnosticare particolari problemi direttamente da un'ambulanza per poi inviare i dati all'ospedale che ospiterà il paziente. In ogni caso tutte le apparecchiature biomedicali non devono superare i limiti imposti dalle normative, altrimenti sarà necessaria una nuova progettazione o un sistema di schermatura aggiuntivo che possa proteggere pazienti, operatori sanitari e altra strumentazione. Il prototipo oltre a testare che i valori limite non siano superati, come possibile evoluzione futura potrebbe quantificare le onde elettromagnetiche assorbite da ogni paziente e dal personale sanitario, analogamente al funzionamento di un dosimetro, in modo da avere dei dati più esaustivi sulla reale esposizione alle radiazioni e favorire la ricerca per valutare i possibili effetti a lungo termine di questo tipo di inquinamento.



## Capitolo 4

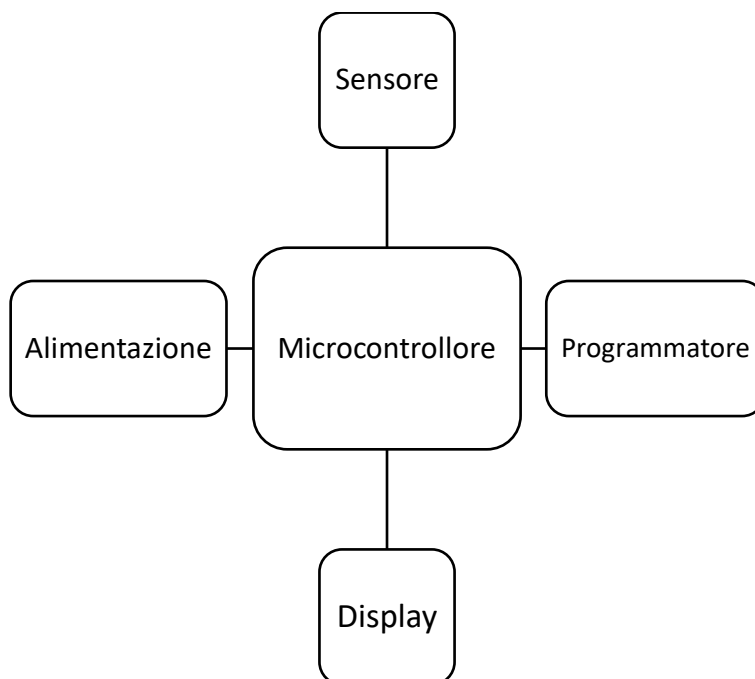
### Materiali e metodi

Gli obiettivi citati nel capitolo 3 possono essere realizzati sfruttando il lavoro di particolari componenti hardware che con la giusta programmazione possono eseguire importanti elaborazioni e garantire risultati interessanti. Il prototipo nasce dalla configurazione hardware e software: la prima si compie eseguendo le giuste connessioni tra il microcontrollore e il sensore e tra il microcontrollore e il display; la seconda consiste nella stesura di un codice di programmazione tramite il software MPLAB IDE cosicché il prototipo possa eseguire una serie di istruzioni.

#### 4.1 Descrizione hardware

La connessione delle componenti hardware per eseguire la programmazione del PIC e per garantire il funzionamento del prototipo è riassumibile dalla figura sottostante.

Fig.12 Schema a blocchi prototipo



In particolare, la lista delle componenti utilizzate è elencata di seguito:

1. Microcontrollore PIC18F47Q10
2. Sensore magnetico UGN3503 ad effetto Hall
3. Magnete di Halbach
4. Display TFT 320x240 pixel con driver ILI9488
5. Cavo di alimentazione e cavi di collegamento e programmatore

- **Microcontrollore Pic18F47Q10**

Il PIC 18F47Q10 opera a 64 MHz di frequenza, ha una memoria di programma FLASH a 128kB, una memoria dati implementata da una SRAM a 3615B e una EEPROM a 1024B. Inoltre dispone di 25 interrupt, 5 porte I/O (A,B,C,D,E), 40 pin di collegamento, 35 canali ADC con conversione a 10 bit.

Fig.13 Microcontrollore



- **Sensore magnetico UGN3503 ad effetto Hall**

Il sensore magnetico UGN3503 è un prodotto della Allegro Microsystem che rileva i campi magnetici sfruttando l'effetto Hall. Esso presenta tre terminali: uno di "supply" ovvero quello da dove occorre fornire la tensione di funzionamento pari a 5V, uno di "ground" cioè il collegamento a terra e uno di "output" da cui giunge il segnale elettrico di risposta proporzionale al campo magnetico rilevato. Tra le caratteristiche tecniche il sensore emette una tensione pari alla metà di quella fornita, ovvero 2,5V nel caso in cui il campo magnetico B sia nullo. Inoltre esso presenta una sensibilità di 1.30mV/G, una scala da -900G a +900G, una banda del segnale rilevabile che può arrivare a 23kHz e un range di temperatura nella quale il sensore continua a funzionare pari a [-20; +85] °C. Il segnale che è inviato

dal sensore è analogico per cui è fondamentale campionare tramite il microcontrollore il segnale stesso e quantizzarlo grazie ad una scala da 0 a 1028, sfruttando quindi 10 bit di conversione A/D.

Fig.14 Sensore magnetico



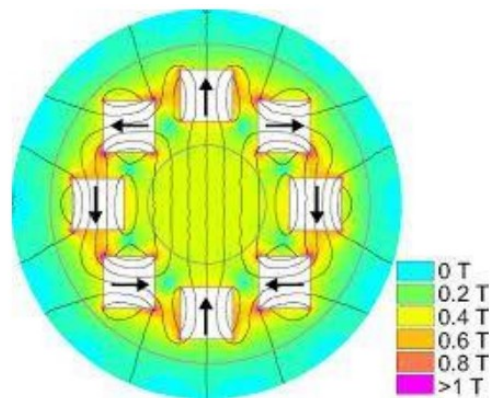
#### 4.4 Magnete di Halbach

Per poter calibrare il sensore e analizzare le sue risposte è stato utilizzato un magnete di Halbach caratterizzato da una precisa disposizione di magneti permanenti, la quale rafforza il campo magnetico lungo una precisa direzione. Tramite il magnete è stato possibile verificare il funzionamento del sensore simulando la presenza di forti campi magnetici indotti da una risonanza magnetica nucleare. Come illustrato nella figura 16 le linee di campo si dispongono lungo una precisa direzione all'interno del magnete con un'intensità di circa 0.3T, mentre in prossimità dei singoli magneti permanenti che costituiscono insieme tutto il sistema, ci sono campi molto più intensi che possono arrivare fino a 0.8T.

Fig.15 Magnete di Halbach



Fig.16 Linee di campo del magnete di Halbach



#### **4.5 Display TFT 320x240 pixel con driver ILI9341**

Il display in questione avente una risoluzione di 320x240 sfrutta la tecnologia TFT garantendo elevate prestazioni, alta risoluzione e un contrasto di colori marcato. Lo schermo comunica con il microcontrollore tramite una connessione SPI ed oltre ai pin della comunicazione seriale (CS, SCK, SDI, SDO) si connettono i pin di alimentazione (Vcc, Gnd), RESET, DC, LED questi ultimi fondamentali per la comunicazione con i registri. Per quanto riguarda il driver ILI9341, la configurazione dei registri è stata ampiamente discussa nel Capitolo 1 nella sezione “Interfaccia SPI”. Occorre tuttavia presentare due problemi che sono nati durante l’uso del display: innanzitutto la scarsa velocità del display nell’eseguire le istruzioni inviate dal microcontrollore e in secondo luogo la bassa luminosità. Per quanto riguarda il primo problema si è deciso di cambiare il quarzo del microcontrollore inizialmente a 20MHz e sostituirlo con un oscillatore a 40MHz in modo da eseguire le istruzioni del PIC più velocemente. Successivamente si è intervenuti sui registri modificando i settaggi; in particolari selezionando il registro B1 relativo al frame rate sono stati fatti dei tentativi al fine di vedere quale impostazione fosse la più efficiente. Prima si è provato a selezionare la frequenza massima possibile ovvero quella pari a 119Hz ma comparivano problemi di risoluzione. Scendendo gradualmente il valore di frame rate, lo scopo era trovare un compromesso tra velocità e risoluzione dello schermo; alla fine la decisione è ricaduta in un valore di 100Hz dato che garantiva entrambi i requisiti. Per risolvere il problema della bassa luminosità si è intervenuti su due particolari registri: il registro numero 51 denominato “brightness” a cui è stato dato il valore associato alla massima luminosità possibile; inoltre, il registro C5 chiamato “VCOM control” il quale consente di stabilire la tensione di riferimento per tutti i pixel. In particolare, è stato scelto -2V rispetto a 0V in modo da avere una differenza di potenziale più alta e quindi una maggior risoluzione a patto che si accetti un maggior consumo energetico.

#### **4.6 Cavo di alimentazione, cavi di collegamento e programmatore**

Il prototipo necessita di un'alimentazione pari a 3.3V per il corretto funzionamento mentre per compiere i collegamenti opportuni si usano dei cavetti che consentono di connettere il sensore al

microcontrollore, nonché quest'ultimo con il display. Per effettuare la programmazione si utilizza un Pickit3 che permette di resettare la vecchia configurazione e iniziarne una nuova.

#### 4.7 Configurazioni

Fig.17 Configurazione microcontrollore con sensore magnetico

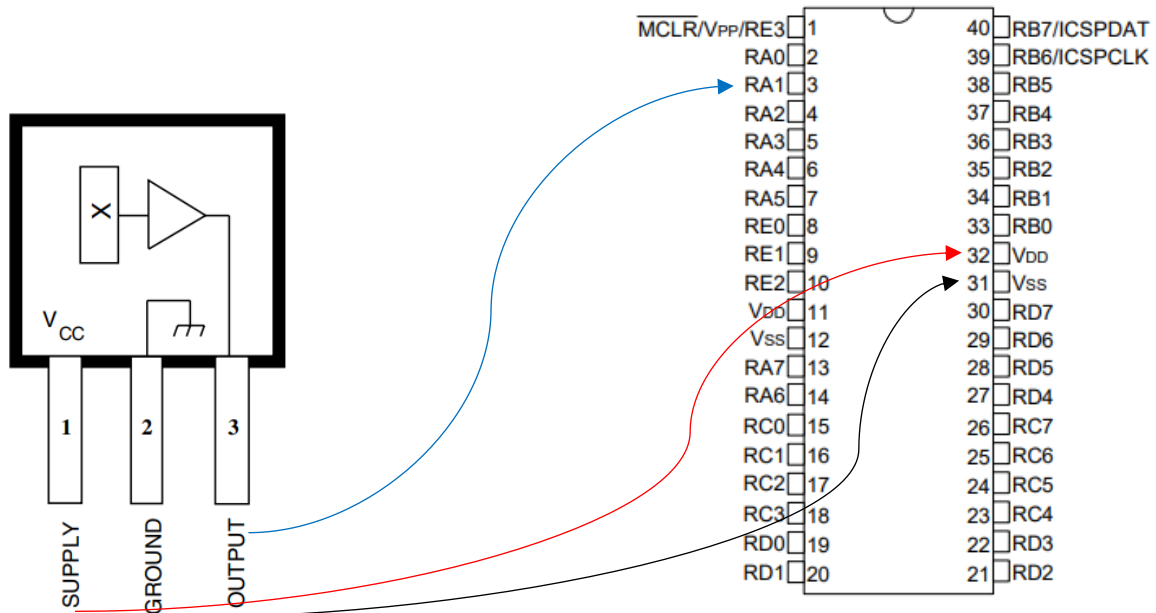
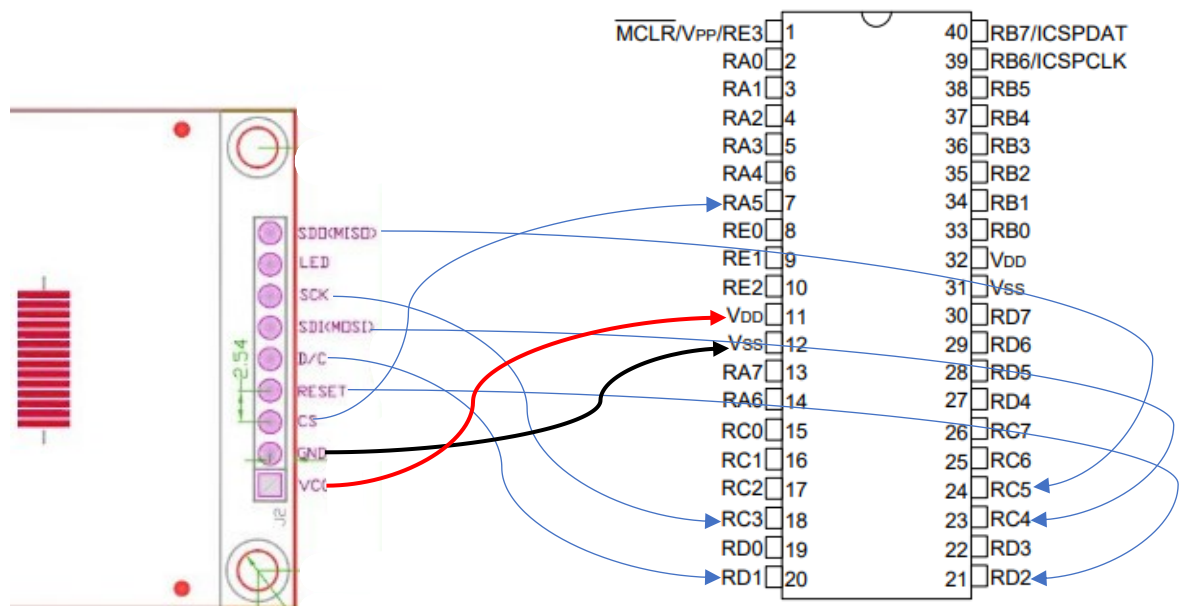


Fig.18 Configurazione microcontrollore con display



Le configurazioni illustrate in figura sono una delle tante possibili combinazioni che possono essere effettuate per il corretto funzionamento del prototipo. Consultando il datasheet del microcontrollore è possibile, infatti, vedere quali pin sono capaci di rilevare segnali analogici, quali sono deputati al

collegamento seriale, quali fungono da interrupt e quali ricevono l'alimentazione. Ovviamente ad una configurazione hardware dei dispositivi deve corrispondere anche un opportuno settaggio software e quindi del codice di programmazione. Considerando i collegamenti illustrati in figura, il capitolo successivo si dedicherà alla descrizione del listato di programmazione.

#### **4.8 Descrizione software**

Una volta compiute le corrette connessioni tra i dispositivi hardware si procede con la scrittura del codice di programmazione in C. Lo scopo del programma è acquisire 640 campioni tramite il sensore magnetico, proiettarli sottoforma di punti su schermo, rendere i dati informativi specificando la media, la presenza di componenti continue o alternate, il numero di volte che la funzione taglia l'offset e quindi lo zero crossing e infine implementare una FFT (Fast Fourier Transform) in modo da ricavare lo spettro in frequenza discreto del segnale. In particolare, il listato si sviluppa nelle seguenti fasi:

- **Inizializzazione del codice:** in questa fase occorre includere tutte le librerie che si desidera utilizzare e inizializzare le variabili. Oltre alla configurazione della comunicazione seriale, è necessario introdurre un vettore che possa contenere 640 campioni. Infine, bisogna inizializzare il display settando i registri (la funziona addetta è chiamata "Init9341" ed è presente in appendice [A4]) e impostando lo sfondo di colore nero.
- **Acquisizione e conversione analogico-digitale:** la seconda fase si occupa di compiere correttamente l'acquisizione settando tutti i registri come descritto in Appendice [A2] e memorizzando i singoli valori ormai campionati nel vettore.
- **Analisi informativa:** una volta terminata l'acquisizione occorre ricavarsi dei dati informativi che dovranno poi essere mostrati su schermo, ovvero la tipologia di segnale (continuo, alternato), la media e lo zero crossing. Per ricavarsi il primo tramite delle funzioni innanzitutto si ottiene il massimo e il minimo del segnale e successivamente si deduce che ci sarà un segnale continuo qualora la loro differenza sia pari a 0, mentre sarà non continuo se è diverso da 0. In realtà, poiché possono insorgere dei disturbi si considera un margine di errore pari a

3, oltre alla quale il segnale è considerato alternato. Per avere il secondo parametro, ovvero la media basta applicare la somma di tutti i campioni per poi dividere per il numero degli stessi. Infine, per calcolare lo zero crossing si utilizza un contatore che aumenterà di una unità ogni volta che un campione si trova sopra il valore di offset e il successivo al di sotto dello stesso e viceversa, oppure quando un campione ricade esattamente sulla media.

- Proiezione su schermo: la quarta fase è deputata grazie all'impiego di librerie grafiche a scrivere su schermo i punti (rappresentati grazie a un grafico cartesiano) e i dati informativi precedentemente ricavati. In particolare, la proiezione dei punti avviene tracciando due volte 320 campioni dato che il display presenta una risoluzione di 320x240 e in quanto si dispone di 640 campioni.
- Calcolo FFT e proiezione su schermo: infine come ultima fase, tramite una funzione ad hoc, si ricavano i valori della FFT i quali saranno poi proiettati su schermo tracciando delle barre verticali. In appendice [A3] si può vedere la funzione per implementare la FFT in codice C.

Nella sezione successiva è presente il listato di programmazione che consente di realizzare il firmware per il funzionamento del dispositivo. Tuttavia, per avere una visione completa e per comprendere appieno le scelte fatte nella scrittura del codice bisogna descrivere sinteticamente alcune librerie utilizzate, specificando le funzioni in esse contenute.

- “def.h”: innanzitutto all'interno di questa libreria si specificano le connessioni del collegamento seriale scegliendo i nomi ritenuti più opportuni.
- “Add\_9341.h”: giacciono in questa sezione una serie di funzioni di cui una che consente di illuminare un pixel specifico denominata “write\_pixel”, la quale riceve in ingresso le coordinate del punto e i tre colori RGB.
- “Set\_9341.h”: al suo interno compaiono funzioni che consentono di selezionare registri e di imporre i valori sugli stessi, una che permette di settare le impostazioni principali del display, tra cui frequenza, accensione, velocità e molte altre ed è chiamata “Init9341”, ed infine

un'altra che permette di imporre uno sfondo nero al display ed è chiamata "Set\_Background1".

- "gfx.h": equivale alla libreria grafica e quindi consente di scrivere un testo grazie a "display\_puts", di scrivere un numero tramite "display\_putn", e in particolare utilizzando "display\_setTextColor", "display\_setTextFont", "display\_setTextSize" e "display\_setCursor" si imposta rispettivamente il colore, lo stile, la grandezza e la posizione del carattere.
- mcc\_generated\_files/mcc.h: questa coincide con una cartella dove risiedono molte librerie diverse caratterizzate a sua volta da molte funzioni differenti tra loro. Occorre ricordare "SYSTEM\_Initialize" che abilita tutte le funzioni della cartella, "ADCC\_Initialize" e "ADCC\_StartConversion" che rispettivamente configura i registri di conversione analogica/digitale (espressa nella prima parte di [A2]) e inizia la conversione a 10 bit.



## Capitolo 5

### Risultati

Il seguente capitolo illustrerà prima di tutto il listato di programmazione e successivamente saranno mostrate delle foto per vedere il reale comportamento del dispositivo e le interconnessioni compiute tra i vari elementi hardware. Per quanto riguarda la struttura del codice è presente un file “main.c” coincidente con la prima parte del listato, un file “Risonanza.h” dove sono state scritte funzioni opportune per non appesantire eccessivamente il codice principale e infine un file “fft.h” esplicitato in appendice [A3]. Sempre in appendice si può trovare l’implementazione di alcune funzioni citate nel codice tra cui quella per il settaggio dei registri della conversione A/D e quelli relativi alla configurazione del display.

#### 5.1 Listato di programmazione: main.c

```
#include "mcc_generated_files/mcc.h"           //Includo i file e le librerie di interesse
#include "def.h"
#include "Set_9341.h"
#include "Add_9341.h"
#include "gfx.h"
#include "Risonanza.h"
#include "fft.h"
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

unsigned int campioni[640];                    //Inizializzo vettore dove acquisire 640 campioni
bool wave=0;                                  //Inizializzo un booleano per indicare se il segnale è continuo o no
int max=0;                                    //Inizializzo variabile per trovare il massimo
int min=0;                                    //Inizializzo variabile per trovare il minimo
int media=0;                                  //Inizializzo variabile per trovare la media in Gauss
float mediamT=0;                              //Inizializzo variabile per trovare la media in mT
unsigned int zeri=0;                          //Inizializzo variabile per trovare gli attraversamenti sulla media

void main(void)
```

```

{
SYSTEM_Initialize(); //Abilito funzioni della cartella mcc_generated_files/mcc.h
SSP1CON1bits.SSPEN = 1; //Attivo la porta seriale SPI
__delay_ms(50); //Aspetto 50ms per consentire l'attivazione della porta seriale
Init9341(); //Setto funzioni del display
__delay_ms(50); //Aspetto 50 ms per consentire l'attivazione di tutte le funzioni
Set_Background1(); //Setto il background
ADCC_Initialize(); //Setto tutti i registri per la conversione A/D
ADCC_StartConversion(1); //Abilito conversione
INTCONbits.GIE = 0; //Disattivo tutti gli interrupt

display_setTextSize(2); //Setto la dimensione del carattere
display_setTextFont(5,7,1); //Setto lo stile del carattere
display_setTextColor(255,255,255,0,0,0); //Setto il colore del carattere
graficoCartesiano(); //Traccio grafico cartesiano

for(int i=0; i<640; i++){ // acquisisco 640 campioni
    ADCON0bits.ADGO = 1; //Inizio conversione A/D:
    while(ADCON0bits.ADGO == 1); //Attendo il termine della conversione, ovvero quando ADGO è 0
    campioni[i]=(ADRESH*256) | (ADRESL); //Terminata la conversione, memorizzo i dati digitali nel vettore
}

max=findMax(campioni,640); //Trovo il massimo tra i punti del vettore
min=findMin(campioni,640); //Trovo il minimo tra i punti del vettore
media=offset(campioni); //Trovo la media tra i punti del vettore
if((max-min)>3) { //Considero un errore pari a 3 per stabilire un segnale continuo
    wave=1;
}

max=round((float)(max*1800/481.104)-1923); //Trasformo il livello digitale quantizzato a 10 bit nel valore reale
min=round((float)(min*1800/481.104)-1923); //Trasformo il livello digitale quantizzato a 10 bit nel valore reale
media=round((float)(media*3.7414)-1923); //Trasformo il livello digitale quantizzato a 10 bit nel valore reale
mediamT=media/10; //Cambio unità di misura sapendo che 1G = 0.1mT
zeri=zerocrossing(campioni,media); //Ottengo numero di attraversamenti sulla media del segnale

int i=0; //Inizializzo primo contatore
int k=0; //Inizializzo secondo contatore

```

```

for(int j=0; j<2; j++){ //Primo ciclo per eseguire 2 schermate da 320 pixel
    while(i < 320+(j*320)){ //Secondo ciclo per tracciare i 320 valori dell'array su schermo
        write_pixel(((i-(j*320))), 240-round(((float)campioni[i])*0.234),255,255,255); //Traccio punti bianchi
        i++; //Aumento contatore
    }
    __delay_ms(1000); //Attendo 1 secondo per vedere segnale dei primi 320 campioni
    while(k < 320+(j*320)){ //Elimino punti già tracciati sovrascrivendoli con punti neri
        write_pixel(((k-(j*320))), 240-round(((float)campioni[k])*0.234),0,0,0); //Traccio punti neri
        k++; //Aumento contatore
    }
}
cancellaGraficoCartesiano(); //Elimino grafico cartesiano
display_setTextColor(255,255,255,0,0,0); //Selezione colore carattere
display_setCursor(20,20); //Selezione posizione carattere
display_setTextSize(2); //Selezione dimensione carattere
if(wave==0){ //Riporto presenza di un segnale continuo valutando il booleano
    display_puts("Continuo");
}
else display_puts("Non continuo"); //Altrimenti riporto la presenza di un segnale non continuo
display_setCursor(20,50); //Selezione posizione carattere
display_puts("Media[G] "); //Riporto media in G
display_putn(media);
display_puts(" G");
display_setCursor(20,80); //Selezione posizione carattere
display_puts("Media[mT] "); //Riporto media in mT
display_putn(mediamT);
display_puts(" mT");
display_setCursor(20,110); //Selezione posizione carattere
display_puts("Zero crossing "); //Riporto numero di attraversamenti sulla media
display_putn(zeri);
__delay_ms(3000); //Attendo 3 secondi per leggere i dati informativi
display_setCursor(20,20); //Elimino i dati prima scritti
display_puts(" ");
display_setCursor(20,50);
display_puts(" ");
display_setCursor(20,80);

```

```

display_puts("                ");
display_setCursor(20,110);
display_puts("                ");

FFT(campioni,3);                //Compio FFT del segnale
display_setCursor(20,20);       //Scrivo FFT in alto a sinistra dello schermo
display_puts("FFT");
for(int i=0; i<16; i++){        //Per ogni punto traccio barre verticali e in particolare 16
    writeLineVertical(i*20,campioni[i]);
}
while(1);
}

```

## 5.2 Listato di programmazione: Risonanza.h

*/\*Funzione per trovare il massimo: attraverso un ciclo scandisco il vettore e ogni qual volta trovo un valore superiore a quello di riferimento, aggiorno il massimo.\*/*

```

unsigned int findMax(unsigned int v[], unsigned int dim){
    int max=0;
    for(int i=0; i<dim; i++){
        if(v[i]>max){
            max=v[i];
        }
    }
    return max;
}

```

*/\* Funzione per trovare il massimo: attraverso un ciclo scandisco il vettore e ogni qual volta trovo un valore inferiore a quello di riferimento, aggiorno il minimo.\*/*

```

unsigned int findMin(unsigned int v[], unsigned int dim){
    int min=1023;
    for(int i=0; i<dim; i++){
        if(v[i]<min){
            min=v[i];
        }
    }
    return min;
}

```

*/\*Funzione per tracciare grafico cartesiano: attraverso due cicli traccio gli assi x e y arrivando rispettivamente a 320 per il primo e 240 per il secondo. Successivamente scrivo i nomi degli assi.*

```

void graficoCartesiano(){
    for(int i=0; i<320; i++){
        write_pixel(i,120,255,255,255);
    }
    for(int j=0; j<240; j++){
        write_pixel(160,j,255,255,255);
    }
    display_setCursor(170,8);
    display_putc('B');
    display_setCursor(10,135);
    display_puts("campioni");
}

```

/\*Funzione per cancellare grafico cartesiano: applico lo stesso ragionamento della funzione precedente ma invece di tracciare dei punti bianchi, li metto di colore nero come lo sfondo.\*/

```

void cancellaGraficoCartesiano(){
    for(int i=0; i<320; i++){
        write_pixel(i,120,0,0,0);
    }
    for(int j=0; j<240; j++){
        write_pixel(160,j,0,0,0);
    }
    display_setTextColor(0,0,0,0,0,0);
    display_setCursor(170,8);
    display_putc('B');
    display_setCursor(10,135);
    display_puts("campioni");
}

```

/\*Funzione per trovare la media: è possibile trovare questo parametro sommando uno ad uno gli elementi del vettore per poi dividere per il numero di punti.\*/

```

unsigned int offset(unsigned int s[]){
    unsigned short i;
    float tot = 0;
    for(i = 0; i<320; i++){
        tot = tot + s[i];
    }
    return round(tot/320);
}

```

/\*Funzione per trovare il numero di attraversamenti sulla media: tramite delle condizioni posso stabilire se il segnale ha tagliato o meno l'offset. In particolare, se il campione i-esimo è maggiore dell'offset mentre quello immediatamente successivo è minore allora aumento il contatore. Allo stesso modo viceversa se un campione è minore della media e quello dopo è maggiore. Inoltre se il campione i-esimo è pari all'offset mentre quello i-1 e quello i+1 sono rispettivamente minore e maggiore della media o viceversa, anche in questo caso aumento il contatore.\*/

```
unsigned short zerocrossing(int s[], unsigned int offset){
    unsigned short zeros = 0;
    unsigned short i;
    for(i = 1; i<639; i++){
        if(s[i]>offset){
            if(s[i+1]<offset){
                zeros = zeros + 1;
            }
        }
        else if(s[i]<offset){
            if(s[i+1]>offset){
                zeros = zeros + 1;
            }
        }
        else if(s[i]==offset){
            if(s[i-1]>offset && s[i+1]<offset){
                zeros = zeros + 1;
            }
            else if(s[i-1]<offset && s[i+1]>offset){
                zeros = zeros + 1;
            }
        }
    }
    return zeros;
}
```

/\*Funzione per tracciare le linee verticali: è possibile tracciare le linee verticali per la FFT attraverso un ciclo che da 0 al valore esplicito della trasformata illumina una serie di pixel. Per maggiore impatto visivo scrivo 4 barre adiacenti della larghezza di un pixel ciascuno in modo da avere dei rettangoli più spessi.\*/

```
void writeLineVertical(unsigned int x, unsigned int y){
    for(int i=0; i<=y; i++){
        write_pixel(x,240-i,255,255,255);
        write_pixel(x+1,240-i,255,255,255);
        write_pixel(x+2,240-i,255,255,255);
        write_pixel(x+3,240-i,255,255,255);
    }
}
```

}  
}

### 5.3 Foto prototipo

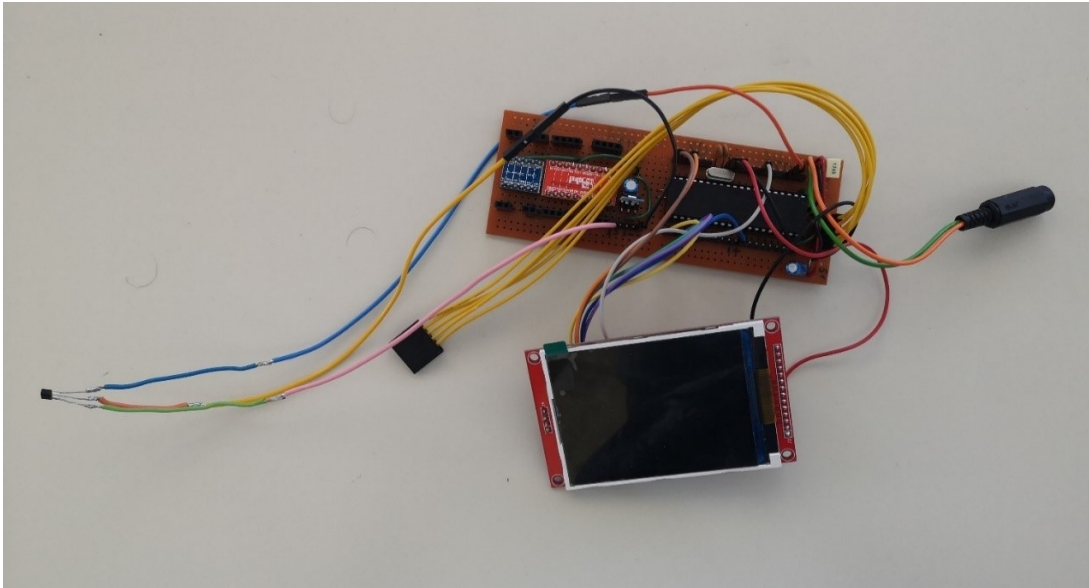


Fig.19 Rappresentazione prototipo

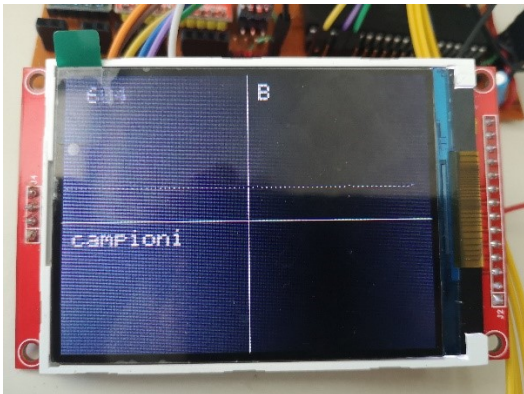


Fig.20 Segnale continuo

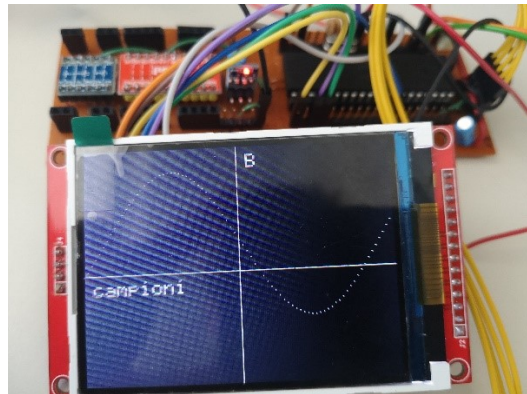


Fig.21 Segnale alternato



Fig.22 Illustrazione dati informativi

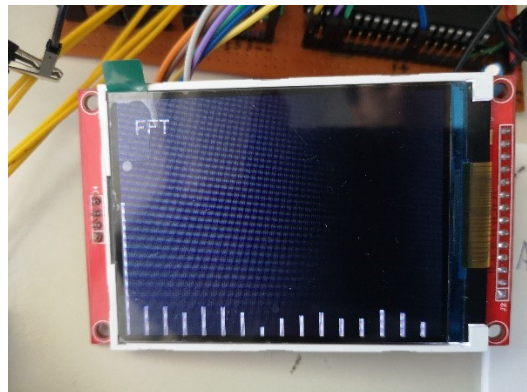


Fig.23 Rappresentazione della FFT

## Capitolo 6

### Discussione

Il prototipo descritto nei capitoli precedenti è un esempio di strumento biomedicale che acquisisce segnali di campo magnetico al fine di avvertire eventuali operatori sanitari del possibile rischio di interferenze con altri macchinari o dell'eccessiva esposizione al campo stesso. L'implementazione hardware del dispositivo si basa sulla scelta delle componenti elettroniche che garantiscano buone performance e siano a un buon prezzo di mercato. In futuro sarebbe possibile migliorare alcuni aspetti del prototipo magari sostituendo o aggiungendo degli elementi che assicurano una migliore acquisizione ed elaborazione dei dati, nonostante tale scelta richieda un maggior esborso economico. Ad esempio, una delle modifiche che potrebbe essere portata a termine, sarebbe l'utilizzo di un modulo di conversione analogico-digitale a 12 bit o ancora meglio a 16 bit. Aziende come la stessa Microchip o la Texas Instruments vendono questi prodotti anche a un costo modesto e inoltre il loro acquisto garantirebbe una maggior precisione nella conversione analogica-digitale in quanto offrono un maggior numero di livelli di quantizzazione. Un'altra modifica che si potrebbe implementare è la configurazione di una memoria RAM esterna che permetterebbe di memorizzare un maggior numero di campioni e di conseguenza favorirebbe una miglior elaborazione dei dati. Infatti, la memoria RAM risulta essere molto utile per migliorare la FFT: il nostro prototipo a causa della scarsa memoria interna, attualmente a partire da 640 campioni garantisce soltanto 16 punti ottenuti dalla trasformata discreta di Fourier quando invece la massima informazione possibile sarebbe ricavata dalla metà dei campioni in ingresso, ovvero 320 punti. Essendo la FFT infatti simmetrica e ridondante, si deve considerare esclusivamente la prima metà dei valori disponibili. Una memoria RAM assicurerebbe un maggior spazio in memoria per salvare molti più valori ottenuti dalla FFT e di conseguenza fornirebbe come output un maggior contenuto informativo. Rappresentare 320 barre sul display come risultato della Fast Fourier Transform significherebbe avere la massima risoluzione dello spettro in frequenza a partire dai dati disponibili, ovvero avere a disposizione una sequenza di punti



caratterizzati dal minimo errore. Infine nel caso sia presente un driver apposito sul display si potrebbe configurare un sistema touch in cui grazie all'utilizzo di un pennino si possa arbitrariamente selezionare delle impostazioni e osservare i dati informativi desiderati. L'implementazione del touch screen è già stato descritto nella sezione 1.5 e in particolare in appendice [A1], per cui basterebbe contestualizzare tale codice ai fini del prototipo; nella trattazione è descritto un codice riferito a un display con risoluzione 480x320 per cui occorre compiere una calibrazione geometrica, essendo il prototipo 320x240. Un'evoluzione futura del prototipo potrebbe essere la sua trasformazione in un dosimetro personalizzato, ovvero in uno strumento che sia portato da tutti gli operatori sanitari nonché dai pazienti che devono sottoporsi ad esami di risonanza magnetica, affinché si rilevi l'assorbimento dei campi elettromagnetici da parte dei tessuti biologici. L'idea sarebbe proporre un'analisi quantitativa e qualitativa: nel primo caso per ottenere il valore numerico riferito all'intensità dell'onda elettromagnetica a cui l'organismo si espone, nel secondo caso per comprendere quali frequenze caratterizzano i campi elettromagnetici negli ambienti ospedalieri. In questo modo oltre a garantire una maggior sicurezza per il personale sanitario e i pazienti, si potrebbero portare a termine molti studi epidemiologici che ricercano eventuali disturbi o correlazioni dell'inquinamento elettromagnetico con patologie croniche che si manifestano a lungo termine.

## Appendice

[A1] Libreria per touch SPI: driver XPT2046

Innanzitutto, occorre definire le porte del driver associandole a quelle del microcontrollore.

```
#define T_CS  PORTBbits.RB0
#define T_SDI PORTBbits.RB3
#define T_SDO PORTBbits.RB2
#define T_SCK PORTBbits.RB1
#define T_CS_DIR  TRISBbits.TRISB0
#define T_SDI_DIR TRISBbits.TRISB3
#define T_SDO_DIR TRISBbits.TRISB2
#define T_SCK_DIR TRISBbits.TRISB1
```

Si inizializza una funzione per definire le direzioni di porta (input/output).

```
void InitTouch_XPT2046(){
    T_CS_DIR = 0;           //CS output
    T_SDI_DIR = 0;         //SDI_touch output
    T_SDO_DIR = 1;         //SDO_touch input
    T_SCK_DIR = 0;         //SCK output
}
```

Si utilizzano tre funzioni volte a definire la posizione X, Y e il valore di pressione Z1.

```
uint8_t ReadX_8bit(){
    uint8_t Xtouch;        //Definisco variabile per contenere posizione X
    T_CS = 0;              //Pongo CS = 0 per scrivere buffer
    SSP2BUF=0xD8;          //Invio il control byte 0b11011000 al buffer per leggere posizione "X"
    while(!PIR3bits.SSP2IF); //Attendo che l'invio del comando sia terminato, quando avviene ciò il flag è settato a 1
    PIR3bits.SSP2IF=0;     //Pulisco il flag settandolo a 0
    SSP2BUF=0x00;          //Pulisco il buffer
    while(!PIR3bits.SSP2IF); //Attendo che l'invio del comando sia terminato, quando avviene ciò il flag è settato a 1
    PIR3bits.SSP2IF=0;     //Pulisco il flag settandolo a 0
    Xtouch =(SSP2BUF);     //Assegno il valore letto che ora si trova nel buffer alla variabile Xtouch
    return Xtouch;        //L'output della funzione coincide con il valore di X
}
```

Si esegue la stessa funzione per calcolare Y e Z1, ricordando nel primo caso di mandare come control byte 10011000 che in esadecimale equivale a 98, mentre nel secondo caso 10111000 che in esadecimale coincide con B8.

Per risolvere il problema dato dalla disomogeneità dei sistemi di riferimento, a causa sia delle diverse direzioni degli assi X e Y, sia dalla differente risoluzione tra il sistema touch e quello display, si presti attenzione alle seguenti funzioni deputate a compiere una trasformazione di coordinate. In primo luogo, occorre trasformare la coordinata Y del touch in coordinata X del display e analogamente quella X del touch in quella Y del display; successivamente è necessario calibrare i valori effettivi delle coordinate sapendo che nel touch vi è una risoluzione pari a 128x128 mentre nel display è pari a 480x320. Allora sapendo che il conteggio dei pixel viene effettuato da 0 a 479 per l'asse X e da 0 a 319 per l'asse Y, presupposto che i livelli di quantizzazione del touch presenta una scala da 0 a 127 in entrambe le direzioni, è possibile trovare un fattore di scala che possa permettere la corretta conversione. In particolare, per trovare la coordinata X del display tale fattore sarà pari a  $479/127=3.77$ , mentre per scovare la coordinata Y del display coinciderà con il risultato di  $319/127=2.51$ .

```
int xpos_convert(uint8_t ytouch){ //Funzione per convertire Y del touch in X del display
    int xdisplay; //Definisco variabile di uscita
    xdisplay = 479 - round(((float) ytouch)*3.77); //Effettuo conversione
    return xdisplay; //In uscita invio la coordinata X del display
}
```

```
int ypos_convert(uint8_t xtouch){ //Funzione per convertire X del touch in Y del display
    int ydisplay; //Definisco variabile di uscita
    ydisplay = 319 - round(((float) xtouch)*2.51); //Effettuo conversione
    return ydisplay; //In uscita invio la coordinata Y del display
}
```

Per trovare una soluzione invece, al dover trovare una pressione minima al di sotto della quale non si registra alcuna attività in modo da trascurare disturbi esterni e rumore di fondo, si può osservare il seguente algoritmo: la condizione “if” consente di valutare esclusivamente le coordinate X e Y che sono associate ad una pressione superiore a 1.

```

void drawWithPen(uint8_t dim, uint8_t r, uint8_t g, uint8_t b){ //Funzione per disegnare su schermo con un pennino
    uint8_t xtouch, ytouch, pressure; //Definisco variabili
    xtouch = ReadX_8bit(); //Leggo coordinata X
    ytouch = ReadY_8bit(); //Leggo coordinata Y
    pressure = ReadZ1_8bit(); //Leggo coordinata Z1
    int xdisplay, ydisplay; //Definisco variabili per conversione
    if(pressure > 1){ //Definisco soglia pressione
        xdisplay = xpos_convert(ytouch); //Converto la Y nel sistema di riferimento display
        ydisplay = ypos_convert(xtouch); //Converto la X nel sistema di riferimento display
        drawSquare(xdisplay, ydisplay, dim, r, g, b); //Traccio quadratino per ogni tocco del pennino
    }
}

void drawSquare(int xpos, int ypos, uint8_t dim, uint8_t r, uint8_t g, uint8_t b){
    int vertex, verty; //Definisco posizione vertice quadrato
    vertex = xpos - (dim/2); //Ricavo vertex conoscendo x del centro rispetto sistema riferimento display
    verty = ypos - (dim/2); //Ricavo verty conoscendo y del centro rispetto sistema riferimento display
    for(int i = vertex; i<vertex+dim; i++){
        for(int j = verty; j<verty+dim; j++){
            write_pixel(i,j,r,g,b);
        }
    }
}

```

## [A2] Conversione A/D\_10bit\_PIC18F47Q10

### Settaggio registri per inizializzare conversione A/D

```
ADLTHL = 0x00;           // ADC Lower Threshold LSB
ADLTHH = 0x00;           //ADC Lower Threshold MSB
ADUTHL = 0x00;           //ADC Upper Threshold LSB
ADUTHH = 0x00;           //ADC Upper Threshold MSB
ADSTPTL = 0x00;          //ADC Threshold Setpoint Least Significant Byte
ADSTPTH = 0x00;          //ADC Threshold Setpoint Most Significant Byte
ADRPT = 0x00;            //ADC Repeat Threshold bits
ADPCH = 0x00;            //ADC Positive Channel Selection Register
ADCAP = 0x00;            //ADC Additional Sample Capacitor Selection bits
ADPRE = 0x00;            //Precharge Time Select bits
ADCON1 = 0x00;           //ADC Control Register 1
ADCON2 = 0x10;           //ADC Control Register 2
ADCON3 = 0x00;           //ADC Control Register 3
ADSTAT = 0x00;           //ADC Status Register
ADREF = 0x00;            //ADC Reference Selection Register
ADACT = 0x00;            //ADC AUTO Conversion Trigger Source Selection Register
ADCLK = 0x03;            //ADC Clock Selection Register
ADCON0 = 0x84;           //ADC Control Register 0
ADACQ = 0x0A;            //ADC Acquisition Time Control Register
PIR1bits.ADIF = 0;       //ADC Interrupt Flag bit
PIE1bits.ADIE = 1;       //ADC Interrupt Enable bit
```

### Algoritmo di conversione

```
ADCON0bits.ADON = 1;      //Abilito il modulo ADC
INTCONbits.GIE = 0;       //Disattivo tutti gli interrupt
for(int i=0; i<320; i++){ //Acquisisco 320 campioni
    ADCON0bits.ADGO = 1;   //Inizio conversione A/D:
    while(ADCON0bits.ADGO == 1); //Attendo il termine della conversione, ovvero quando ADGO va a 0
    campioni[i]=(ADRESH*256) | (ADRESL); //Terminata la conversione, memorizzo i dati digitali in un array
}
```

### [A3] Funzione per calcolo della FFT

```
#include <stdio.h>
#include <math.h>
#define Nc 32

const double PI = 3.14159265358;
double h, b;
int lettura[Nc];

unsigned Log2() {
    unsigned k = Nc;
    unsigned i = 0;
    while(k) {
        k >>= 1;
        i++;
    }
    return i - 1;
}

unsigned reverse(unsigned Log_N, unsigned n)
{
    unsigned j, p = 0;
    unsigned limite = Log_N;
    for(j = 1; j <= limite; j++) {
        if(n & (1 << (limite - j)))
            p = p | 1 << (j - 1);
    }
    return p;
}

void ordina(int *lettura){
    unsigned i, temp, nuovo_indice;
    unsigned Log_N = Log2();
    for(i = 0; i < Nc; i++) {
        nuovo_indice = reverse(Log_N, i);
        if((i != nuovo_indice) && (i < nuovo_indice)){
```

```

temp = lettura[i];
lettura[i] = lettura[nuovo_indice];
lettura[nuovo_indice] = temp;
}
}
}

void prodotto_c(double re1, double im1, double re2, double im2)
{
h = (re1 * re2) - (im1 * im2);    //parte reale
b = (re1 * im2) + (im1 * re2);    //parte immaginaria
}

void pot_c(double re, double im, unsigned expp) {
unsigned k;
if (expp == 0) {
h = 1;
b = 0;
}
else {
h = re;
b = im;
for(k=1; k < (expp); k++) {
prodotto_c(h, b, re, im);
}
}
}

void transform(int *lettura) {
double t_re, t_im, Temp_Re, Temp_Im;
double f[Nc][2];    //matrice f_re e f_im, campioni temporali
double W[(Nc/4)+1][2];    //matrice dei coefficienti
unsigned t, i, j, z, n = 1, a = Nc / 2, Log_N = Log2(); //0..65K
ordina(lettura);
for(i = 0; i < Nc; i++) {
f[i][0] = (double) lettura[i];

```

```

f[i][1] = 0;
}
W[0][0] = 1; //Re
W[0][1] = 0; //Im
W[1][0] = cos(-2 * PI / Nc); //Re
W[1][1] = sin(-2 * PI / Nc); //Im

for(i = 2; i < (Nc / 4 + 1); i++) {
    pot_c(W[1][0], W[1][1], i); //Re del coeff 1; Im del coeff 1;
    W[i][0] = h;
    W[i][1] = b;
}
for(j = 0; j < Log_N; j++) {
    for(i = 0; i < Nc; i++) {
        if(!(i & n)) {
            t_re = f[i][0];
            t_im = f[i][1];
            z = (i * a) % (n * a);
            if(z > (Nc / 4)) {
                Temp_Re = -1 * W[Nc/2-z][0]; //Re
                Temp_Im = W[Nc/2-z][1]; //Im
            }
            else {
                Temp_Re = W[z][0]; //Re
                Temp_Im = W[z][1]; //Im
            }
            prodotto_c(Temp_Re, Temp_Im, f[i + n][0], f[i + n][1]);
            f[i][0] = t_re + h;
            f[i][1] = t_im + b;
            f[i + n][0] = t_re - h;
            f[i + n][1] = t_im - b;
        }
    }
    n = n * 2;
    a = a / 2;
}

```



```

for(t = 0; t < Nc; t++) {
    t_re = f[t][0];
    t_im = f[t][1];
    lettura[t] = (int) 20 + 10*(log10(sqrt(pow(t_re, 2) + pow(t_im, 2))/Nc));
}
}

```

```

void FFT(int *lettura, int x) {
    int m;
    transform(lettura);
    for(m = 0; m < Nc; m++) {
        lettura[m] *= x;
    }
}

```

#### [A4] Funzione “Init9341” per settaggio dei registri del display

```

void Init9341(void){           //Funzione per settare funzioni principali display
    TFT_RST = 1;             //Reset è segnale di output
    TFT_RST_DIR = 0;
    __delay_ms(100);
    TFT_RST = 0;
    __delay_ms(100);
    TFT_RST = 1;             //Porre RST=1 per scrivere nuovi dati
    __delay_ms(200);
    TFT_CS_DIR = 0;          //CS è segnale di output
    TFT_DC_DIR = 0;          //DC è segnale di output
    TFT_SDI_DIR = 0;         //SDI è segnale di output
    TFT_SDO_DIR = 1;         //SDO è segnale di input
    TFT_SCK_DIR = 0;         //SCK è segnale di output

    TFT_CS = 0;              //Porre CS=0 per modificare registro
    sendCommand(0x01);       //Software Reset
    TFT_CS = 1;              //Porre CS=1 per terminare modifica
    __delay_ms(10);          //Attendere 50ms per nuova modifica
}

```

```

TFT_CS = 0;
sendCommand(0x11);           //Sleep OUT
TFT_CS = 1;
__delay_ms(10);

TFT_CS = 0;
sendCommand(0x3A);          //Interface pixel format
sendData(0x66);
TFT_CS = 1;
__delay_ms(10);

TFT_CS = 0;
sendCommand(0x36);          //Memory access control
sendData(0xE8);
TFT_CS = 1;
__delay_ms(10);

TFT_CS = 0;
sendCommand(0xC2);          //Power control 3 for normal mode
sendData(0x0A);              //Frequenza circuito elevatore di tensione, maggiore è la frequenza e
TFT_CS = 1;                  // maggiore è la qualità del display, ma a scapito del consumo di corrente
__delay_ms(10);

TFT_CS = 0;
sendCommand(0xB1);          //Frame rate: 100Hz
sendData(0x00);
sendData(0x13);
TFT_CS = 1;
__delay_ms(50);

TFT_CS = 0;                  //VCOM control: Setto Vref o a 0V o -2V
sendCommand(0xC5);          //Nel secondo caso ho una maggior risoluzione
sendData(0x00);
sendData(0x00);
sendData(0x00);
sendData(0x00);

```

```

TFT_CS = 1;
__delay_ms(10);

TFT_CS = 0;
sendCommand(0x13);           //Normal mode ON
TFT_CS = 1;
__delay_ms(10);

TFT_CS = 0;
sendCommand(0x29);         //Display ON
TFT_CS = 1;
__delay_ms(10);

TFT_CS = 0;
sendCommand(0x53);         //Write CTRL display value
sendData(0x2C);
TFT_CS = 1;
__delay_ms(10);

TFT_CS = 0;
sendCommand(0x51);         //Brightness
sendData(0xFF);
TFT_CS = 1;
__delay_ms(10);

TFT_CS = 0;
sendCommand(0x2A);         //Column Address Set
sendData(0x00);
sendData(0x00);
sendData(0x01);
sendData(0xDF);
sendCommand(0x00);
TFT_CS = 1;
__delay_ms(10);

TFT_CS = 0;

```

```
sendCommand(0x2B);          //Page Address Set
sendData(0x00);
sendData(0x00);
sendData(0x01);
sendData(0x3F);
sendCommand(0x00);
TFT_CS = 1;
__delay_ms(10);

TFT_CS = 0;
sendCommand(0x38);          //IDLE mode OFF
TFT_CS = 1;
__delay_ms(10);
}
```

## **Bibliografia**

- [1] Mankar, Jayant, et al. "Review of I2C protocol." *International Journal of Research in Advent Technology* 2.1 (2014).
- [2] Dhaker, Piyu. "Introduction to SPI interface." *Analog Dialogue* 52.3 (2018): 49-53.
- [3] Domenici, Valentina, and Carlo Alberto Veracini. "Risonanza Magnetica Nucleare: passato, presente e futuro di una tecnica spettroscopica che ha cambiato la chimica e non solo. Parte I." (2007): 13-22.
- [4] Giammarile, F., F. Podo, and F. Iorio. "Nuclear magnetic resonance and positron emission tomography: diagnostic potential in tissue characterization." *Annali Dell'istituto Superiore di Sanita* 25.2 (1989): 253-307.
- [5] Di Martino, Antonio. "L'inquinamento elettromagnetico."
- [6] Fiore, Maria, et al. "Salute e campi elettromagnetici: stato delle conoscenze."
- [7] Tonazzo, Marco. "Effetti delle interferenze elettromagnetiche sulle apparecchiature elettromedicali e l'uomo: aspetti normativi e tecnologici."
- [8] Bogi, A., et al. "Campi elettromagnetici in ambito sanitario: Valutazione e prevenzione del rischio. Risultati di una ricerca condotta a livello regionale. Pisa, Grafiche Caroti, 2013."

## **Ringraziamenti**

Il periodo universitario è stato piacevole e stimolante poiché ho avuto la possibilità di confrontarmi con nuove persone, arricchire le mie conoscenze e crescere personalmente. La realizzazione della tesi e più in generale il superamento di tutto il percorso di laurea triennale deriva oltre che da sforzi personali anche dal contributo di molte persone intorno a me senza le quali non sarebbe stato possibile il raggiungimento di tale traguardo.

Innanzitutto, vorrei ringraziare il relatore di tesi, il professor Francesco Piva che mi ha aiutato a comprendere il funzionamento e le potenzialità dei microcontrollori PIC nonché altri strumenti di laboratorio. Con grande disponibilità ha chiarito dubbi e ha condiviso le sue conoscenze e la sua preziosa esperienza permettendomi di portare a termine il prototipo.

Un forte ringraziamento mi piacerebbe dedicarlo soprattutto ai miei genitori, mio padre Mauro e mia madre Raffaella che mi hanno sempre sostenuto nei momenti difficili e mi hanno dato la forza per superare questo percorso triennale di università, insegnandomi valori quali sacrificio, passione e rispetto. Non posso non ringraziare ovviamente mio fratello Riccardo con cui ho condiviso sorrisi e litigi e con cui ho trascorso gran parte del mio tempo.

Vorrei a questo punto ringraziare una persona speciale, Alessia che è sempre stata al mio fianco in questi tre anni e che mi ha stimolato continuamente a dare il meglio di me nel raggiungere i miei obiettivi. Un grazie molto sentito anche ai miei compagni di viaggio di università Lorenzo e Matteo con cui mi sono confrontato per studio ed esami e insieme abbiamo avuto tante soddisfazioni. Al di fuori dell'università ringrazio ovviamente tutti gli amici che mi hanno strappato una risata nei momenti di svago, tra cui mi piacerebbe nominare Francesco, Nicola, Giulia e Alessandra per la loro simpatia e intraprendenza. Infine mi piacerebbe ringraziare tutti i parenti che hanno creduto in me tra cui mia nonna Maria sempre gentilissima e curiosa dei miei risultati, le zie, gli zii, cugine e cugini che invece mi hanno spesso dato consigli di vita sia in ambito universitario sia al di fuori di esso.