

SISTEMA DI LOCALIZZAZIONE ED IMPLEMENTAZIONE DI UN CRUSCOTTO TOUCH SCREEN BASATI SU ARDUINO

Introduzione

La tesina può dividersi in due parti, la prima delle quali consiste nel realizzare un sistema di localizzazione portatile basato su Arduino ed alimentato da una qualche forma di energia rinnovabile come quella elettromeccanica fornita da un sensore piezoelettrico.

Quindi in primo luogo si descriverà il sensore piezoelettrico passando poi a descrivere una sua applicazione come quella di realizzare un sistema di localizzazione per servizi di bike sharing che purtroppo però non è stato possibile realizzarlo in quanto non si rispettavano i requisiti prefissati.

La seconda parte invece consiste nel realizzare un progetto che verte nell'implementazione di un cruscotto touch che attraverso il microcontrollore Arduino, sia in grado di visualizzare, attivare/disattivare lo stato di un LED e contemporaneamente anche a visualizzare lo stato dei sensori connessi a tale sistema.

In particolare il sensore utilizzato nell'esperienza è di luminosità ottenuto attraverso una fotoresistenza.

In questo secondo caso oltre a presentare il microcontrollore Arduino ed un suo particolare modello che è quello utilizzato nell'esperienza, si descriverà anche il funzionamento di una fotoresistenza e si descriveranno le principali righe di codice che hanno dato modo di realizzare il progetto.

Il generatore piezoelettrico

Un generatore piezoelettrico è un oggetto che trasduce in energia elettrica disponibile a partire dalle vibrazioni meccaniche alle quali è soggetto.

Esso è quindi un dispositivo elettromeccanico che deve essere installato ad esempio su di una bicicletta in modo tale che le vibrazioni meccaniche percepite durante il suo movimento vengano appunto sfruttate per

convertirle in energia elettrica disponibile per alimentare un dispositivo portatile.

Quindi grazie alla sola eccitazione meccanica, i generatori piezoelettrici riescono a generare carica elettrica in quanto sono composti da materiali attivi.

Questa fonte di energia elettrica è utile pensarla applicata su dispositivi elettronici portatili ed a basso consumo energetico come ad esempio le lampade a LED, display e sensori.

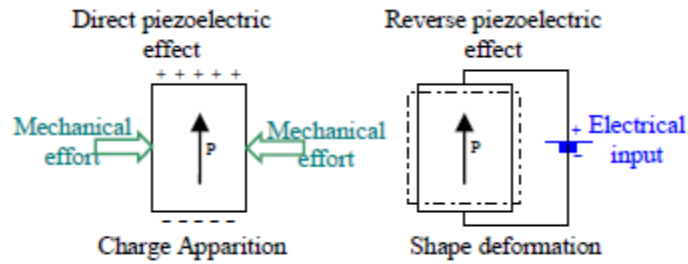
Nel caso particolare in cui il dispositivo portatile debba essere montato su una bicicletta si nota infatti che se anziché alimentarlo con una dinamo lo si alimentasse appunto con un generatore piezoelettrico, il ciclista non è più obbligato a fornire alcuno sforzo supplementare per compensare l'attrito della dinamo stessa grazie alle caratteristiche elettromeccaniche offerte dal materiale piezoelettrico.

In generale quindi un veicolo infatti nel suo movimento vibra in modo permanente nel tempo ed è quindi ciò favorisce appunto alla conversione elettromeccanica tramite i trasduttori piezoelettrici in modo tale che l'energia elettrica prodotta possa essere sfruttata per alimentare i dispositivi portatili a bordo del veicolo.

Fondamenti del materiale piezoelettrico

Per convertire l'energia meccanica in elettrica oltre alla comune dinamo, esistono altri fenomeni fisici come quelli che sfruttano l'effetto piezoelettrico.

Si intende effetto piezoelettrico diretto quello che converte l'energia meccanica in energia elettrica altrimenti esiste anche quello che è l'effetto inverso che è in grado di convertire un potenziale elettrico applicato in energia di deformazione meccanica come evidenzia la seguente figura:



In particolare l'effetto piezoelettrico diretto è dovuto a materiali capaci di funzionare come sensori mentre quello inverso funzionano come attuatori.

I materiali piezoelettrici esistenti sono il quarzo che però non è un materiale interessante per la produzione di energia elettrica e altri cristalli artificiali.

Caratteristiche

I materiali piezoelettrici appartengono alla classe dei materiali chiamati ferroelettrici.

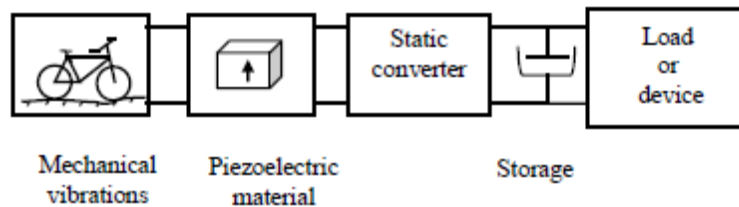
Un materiale ferroelettrico ha una struttura molecolare orientata e cioè costituita localmente da un dipolo elettrico.

L'insieme dei dipoli elettrici che costituiscono il materiale ferroelettrico sono orientati in maniera casuale ma se a tale materiale gli viene applicato un campo elettrico sufficientemente intenso allora i dipoli elettrici che lo costituiscono tendono a riorientarsi proprio rispetto al campo elettrico applicato; se poi il campo elettrico viene rimosso tali dipoli rimangono comunque orientati e quindi allineati rispetto al campo elettrico precedentemente applicato e quindi il sistema può ritenersi così polarizzato.

Al completamento della polarizzazione il materiale a cui è stato sottoposto esporrà il suo effetto piezoelettrico.

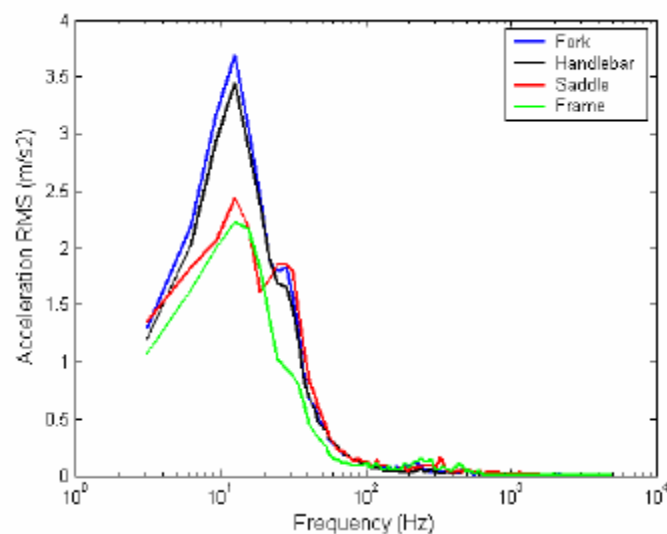
Principi del generatore piezoelettrico

Allo scopo di accumulare energia elettrica attraverso l'effetto piezoelettrico si mostra la seguente figura che rappresenta appunto la catena di conversione:

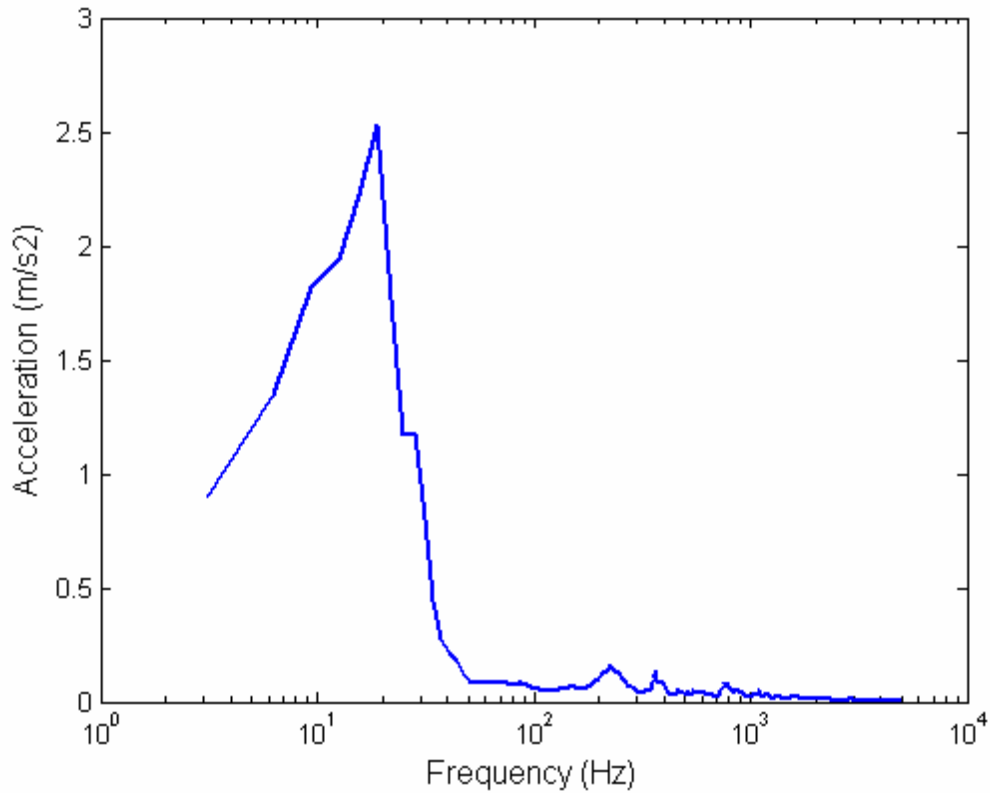


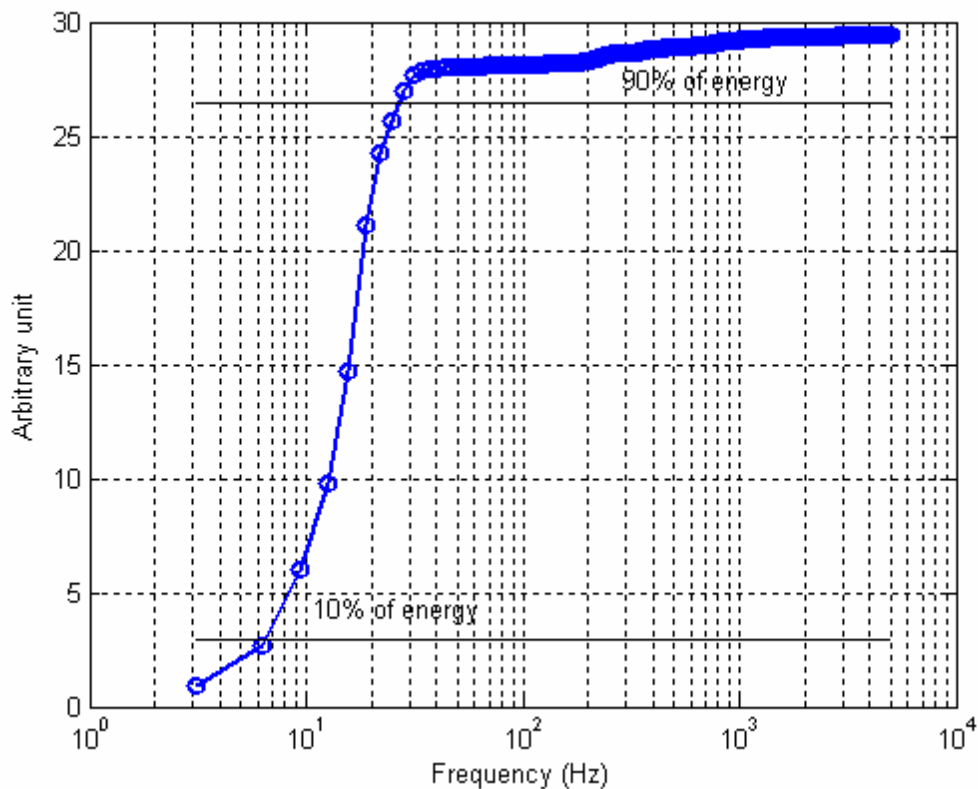
Tale catena è costituita appunto da un primo blocco che rappresenta una bicicletta come fonte di energia meccanica la quale grazie alle vibrazioni meccaniche generate durante il suo movimento va ad eccitare un materiale piezoelettrico la cui energia elettrica prodotta verrà ad essere immagazzinata ed utilizzato ad un eventuale carico posto a valle della catena complessiva.

Da analisi approfondite si è riscontrato che in una bicicletta sono la forcella e il manubrio quelli candidati a collocare il materiale piezoelettrico in quanto è in esso che si va a causare una maggiore quantità di vibrazioni (cfr la figura seguente)



Inoltre indipendentemente dalla velocità e dal tipo di pista, la banda di frequenze interessate alla maggior concentrazione dell'energia meccanica è contenuta si aggira approssimativamente tra i 10 e i 30 Hz come si evidenzia nelle due seguenti figure:





Ciò porta a concludere quindi che al fine di ottenere i migliori risultati dalla conversione elettromeccanica è conveniente progettare il generatore piezoelettrico che operi nel banda considerata e cioè che abbia appunto delle risonanze poste proprio nella gamma di frequenze che va dai 10 Hz ai 30 Hz.

SISTEMA DI LOCALIZZAZIONE APPLICATO A SERVIZI DI BIKE SHARING

Grazie all'esistenza dei sensori piezoelettrici una sua possibile applicazione è quella di realizzare un sistema di localizzazione ad esempio basato su Arduino che sia in grado di acquisire ed inviare la sua posizione geografica ad un database remoto mediante modulo GPS/GPRS/UMTS, tutto il sistema deve essere alimentato da una batteria che a sua volta deve essere sostenuta da un carica batterie basato sull'effetto di un sensore piezoelettrico.

Il sistema globale quindi può essere pensato da una catena costituita da un microcontrollore Arduino alimentato da una batteria caricata da un sensore piezoelettrico e connesso ad una scheda GSM ed una scheda GPS.

Il sistema quindi ogni lasso tempo prefissato dovrà attivarsi per essere programmato per comunicare i dati acquisiti riguardanti la posizione del veicolo su un database remoto.

L'intero sistema dovendo essere supportato da una bicicletta, richiede dei requisiti fondamentali che oltre ad essere a basso consumo energetico deve essere anche di piccole dimensioni e presentare un costo contenuto.

Conclusioni

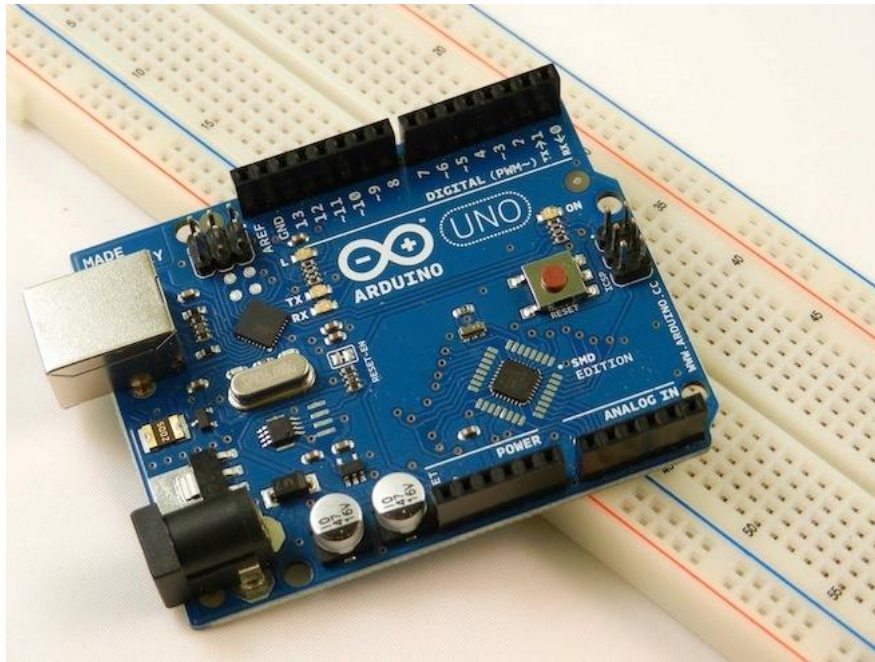
Il progetto non è stato possibile realizzarlo in quanto non rispettava i requisiti prefissati.

Infatti in prima analisi già nel dimensionamento si è riscontrato che il più economico dei sensori piezoelettrici ma comunque già ritenuto costoso ai fini dell'applicazione, non era in grado di fornire un'energia sufficientemente ad alimentare l'intero sistema.

Inoltre anche le dimensioni globali del sistema pensato appunto anche costituito dai moduli GSM e GPS, non sarebbero così molto ridotte da poter verificare il requisito prefissato.

Il microcontrollore Arduino

Arduino, è uno strumento open source che semplifica la progettazione e la prototipazione elettronica offrendo a studenti, insegnanti, artisti o semplici appassionati una piattaforma hardware e software per l'implementazione rapida di progetti interattivi, multimediali e multidisciplinari.



Arduino Uno, attuale modello “standard” della famiglia Arduino

La parte hardware è costituita da una scheda a singolo microcontrollore (che cambia a seconda dei modelli) capace di recepire input fisici dei più disparati, processarli e controllare motori, luci, suoni e molti altri elementi di output fisico.

La parte software di Arduino consiste invece in una IDE¹ cross-platform, disponibile cioè per tutti i sistemi operativi più diffusi.

Il linguaggio di programmazione di Arduino è una derivazione di Wiring e Processing, progetti fondamentali nell’ambito del physical computing e dell’interaction design.

Arduino, come molti altri progetti analoghi, nasce per semplificare il processo di programmazione di un microcontrollore, un’operazione solitamente riservata agli appassionati con conoscenze avanzate di elettronica. La scheda offre una piattaforma accessibile anche ai meno

¹ Integrated Development Environment (Ambiente di Sviluppo Integrato)

esperti con un sistema di pin di ingresso ed uscita ai quali si possono far corrispondere gli input e gli output fisici.

L'ambiente di sviluppo è ottimizzato per non spaventare coloro che non hanno familiarità con la programmazione ed offre strumenti che rendono più semplice ed intuitiva la scrittura del codice. Allo stesso tempo la IDE offre un'ampia versatilità che non limita i più esperti e consente di creare soluzioni via via più complesse.

Arduino non è l'unica piattaforma di prototipazione elettronica che si proponga di offrire una soluzione semplificata per la programmazione dei microcontrollori.

Il suo successo, però, è legato a fattori che distinguono questo progetto da tutti gli altri in quanto è ritenuto:

- *economico*: una scheda si può acquistare online per circa 30€ e il prezzo scende ulteriormente per il kit da saldare;
- *compatibile* con tutti i maggiori sistemi operativi sia a livello hardware sia a livello software ed è estremamente semplice da utilizzare;
- *semplice e versatile*: Arduino è Open Source e cioè "aperto" l'hardware, che si può modificare, riprogettare, estendere senza alcun limite; è aperto il software, che si può espandere tramite librerie C++ e si può integrare con "pezzi" di codice AVR-C, il linguaggio "nativo" del microcontrollore.

La "openess" di Arduino è l'elemento caratteristico che più di ogni altro ne ha determinato il successo, garantendo al progetto il supporto crescente di una community online forte di decine di migliaia di utenti che condividono costantemente i propri progetti, il proprio codice e soprattutto la propria esperienza.

Per la sua versatilità, semplicità d'uso, compatibilità ed estrema economicità, Arduino è diventato rapidamente la piattaforma di

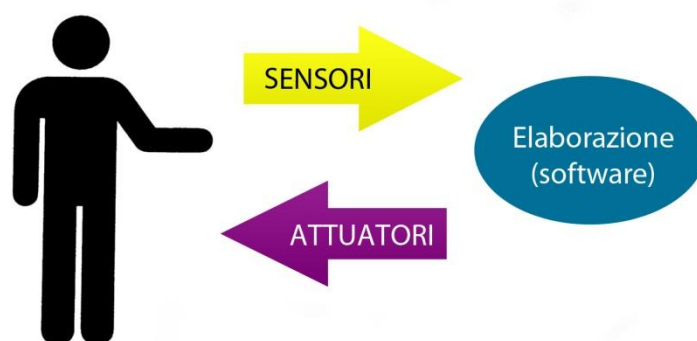
prototipazione elettronica per eccellenza nell'ambito dell'Interaction Design e delle Installazioni Multimediali. A fianco di un numero sempre crescente di applicazioni scientifiche o più prettamente legate all'hobbistica elettronica, Arduino è entrato a far parte della dotazione standard dell'artista multimediale contemporaneo. Poter imparare in poco tempo e in prima persona a programmare e installare un sistema di prototipazione elettronica che rende semplici e immediate interazioni che in passato richiedevano interventi professionali specifici.

La versatilità e la molteplicità delle numerose “destinazioni d'uso” di Arduino è rispecchiata dall'eterogeneità professionale e formativa della squadra che ha lavorato alla sua creazione.

Il software

Il microcontrollore, è il cervello di Arduino che è in grado di elaborare una vasta gamma di “stimoli” in ingresso, producendo segnali in uscita che possono comandare luci, suoni e più in generale degli elementi attuatori che operano sull'ambiente esterno.

Questo processo (Input → elaborazione → output) è, volendo semplificare molto, il modello di base di un dispositivo interattivo.



Per programmare il microcontrollore, il cervello di Arduino, abbiamo a disposizione una piattaforma di sviluppo (nome tecnico: IDE, Integrated Development Environment) che ci permette di scrivere il codice attraverso

il quale impartiremo al microcontrollore gli “ordini” da eseguire e le risposte da fornire ad un dato stimolo dell’ambiente.

In realtà, quello che la IDE ci fornisce è un ulteriore livello di traduzione e, volendo, trasduzione: il microcontrollore parla un linguaggio di basso livello, vale a dire più lontano dal linguaggio umano e più affine alla natura elettrica dei segnali che deve processare. La IDE di Arduino ci permette invece di impartire comandi utilizzando un linguaggio di programmazione di alto livello, impostato cioè su logiche che tendono ad avvicinarsi a quelle che sottendono alle strutture del linguaggio umano.

Prima dell’avvento di Arduino, programmare un microcontrollore significava scrivere codice in un linguaggio che ha ben poco a che fare con quello umano, solitamente Assembly o C, e conoscere a fondo il funzionamento di concetti di elettronica avanzata come gli interrupt e porte logiche. Con un sistema di collegamento seriale, si poteva poi “flashare” la memoria del microcontrollore per programmare fisicamente il componente.

Arduino ha semplificato tutto questo e ci permette, in sostanza, di imporre i nostri ordini al controllore semplicemente definendo un paio di funzioni, che verranno caricate nella memoria del chip con la pressione di un bottone dell’interfaccia della IDE.

La IDE di Arduino e il linguaggio di programmazione

Arduino nasce dal progetto Wiring e di quel progetto non solo ha preso “in prestito” gli intenti ma anche gli strumenti, in particolare il software dell’IDE, che è, nella sua forma attuale, una derivazione diretta dell’IDE di Wiring.



Le IDE di Arduino e Wiring a confronto

L'ambiente di sviluppo di Arduino è un software Open Source sviluppato in Java che può funzionare su qualsiasi sistema operativo e integra un editor di testo specializzato, che offre l'highlight della sintassi del linguaggio e un'ampia gamma di strumenti software per l'interfacciamento con le schede di Arduino esistenti sul mercato oggi.

Il legame con Wiring è rivelato non soltanto dall'aspetto dell'interfaccia di programmazione ma anche dal nome che designa la libreria software "fondamentale" ([Wiring.h](#), sviluppata in C++), grazie alla quale è possibile semplificare enormemente la scrittura di comandi basati su operazioni di input-ouput.

All'utente, infatti, viene richiesto in sostanza di definire due sole funzioni: una chiamata `setup()` che stabilisce quali sono le funzioni dei pin utilizzati dal "dispositivo interattivo" e l'altra, chiamata `loop()`, che invece contiene i veri e propri "comandi" e verrà eseguita in continuazione dal microcontrollore.

L'estrema prossimità tecnica di Arduino e Processing ha reso possibile un'integrazione pressoché totale fra i due progetti: controllare programmi scritti in Processing con Arduino è molto semplice e permette di interfacciare i sensori e il microcontrollore di Arduino con computer più potenti per la realizzazione di installazioni e progetti abbastanza ambiziosi

per complessità e dimensioni da richiedere la potenza di calcolo e le potenzialità di controllo di strumenti complessi (proiettori, riproduzione di video ad alta definizione, ecc...) che solo un PC ad oggi può fornire.

Alla prima apertura del software di Arduino l'utente viene accolto dall'interfaccia dell'editor degli sketch. E' qui che si scrive e modifica il codice da caricare sulla scheda.

I bottoni del menu in alto, illustrati in figura, offrono un rapido accesso alle operazioni di base necessarie per la stesura del codice, la sua validazione e il caricamento dello sketch realizzato su Arduino.



Il menu principale della IDE di Arduino

Da questa interfaccia è possibile accedere direttamente al monitor della porta seriale, un tool che consente di leggere in diretta valori numerici inviati dall'Arduino al computer sulla base dei quali è possibile programmare un feedback attuativo da parte del microcontrollore.

Hardware

Con il nome Arduino sono state prodotte, dal 2005 ad oggi, 19 schede elettroniche "ufficiali" o riconosciute dal team originale di Arduino. Differiscono fra di loro per alcune specifiche, ma condividono tutte delle caratteristiche costruttive comuni. Arduino è, di fatto, un'interfaccia semplificata per un microcontrollore, che è dunque il cuore di ogni "board".

Un "microcontroller" è un dispositivo elettronico integrato su un unico chip, progettato appositamente per interagire con input esterni, analogici o

digitali, e restituire output analogici o digitali derivati dalle operazioni di processing interne determinate da un programma caricato nella memoria del chip.



Un microcontrollore Atmel ATMEGA8–16PI come quello usato sui primi prototipi di Arduino

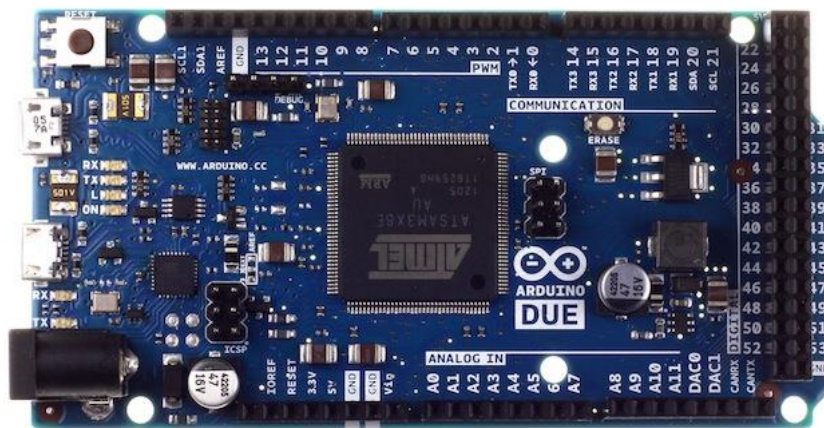
Sulle varie versioni di Arduino, si trovano i modelli di microcontrollore ATmega8, ATmega168, ATmega328 (Arduino Uno), ATmega1280, ATmega2560, ATmega32u4 (Arduino Leonardo e Arduino Esplora).

Arduino Due, lanciato nel 2012, è il primo modello di Arduino ad integrare una CPU Atmel SAM3X8E con infrastruttura ARM Cortex-M3.

Per permettere agli Arduino di comunicare con un computer tramite USB e renderne semplice la programmazione, ogni scheda integra un ulteriore chip per la conversione del segnale digitale da USB a seriale. L'Arduino Uno, il modello di riferimento di tutta la gamma, monta anche un chip AtMega16U2, programmato come convertitore USB-Seriale. Grazie a questo chip l'Arduino connesso ad un computer tramite USB può essere immediatamente riconosciuto come periferica senza configurazioni particolari. Nei modelli più recenti, in particolare sull'Arduino Leonardo e sull'Arduino Esplora, non è presente un ulteriore Chip per la conversione USB-seriale perché il microcontrollore Atmel ATmega32u4 integra già questa funzione. Grazie a questa caratteristica l'Arduino Leonardo e

l'Arduino Esplora sono riconosciuti dai computer come periferiche di input e possono essere utilizzati come fossero un mouse o una tastiera.

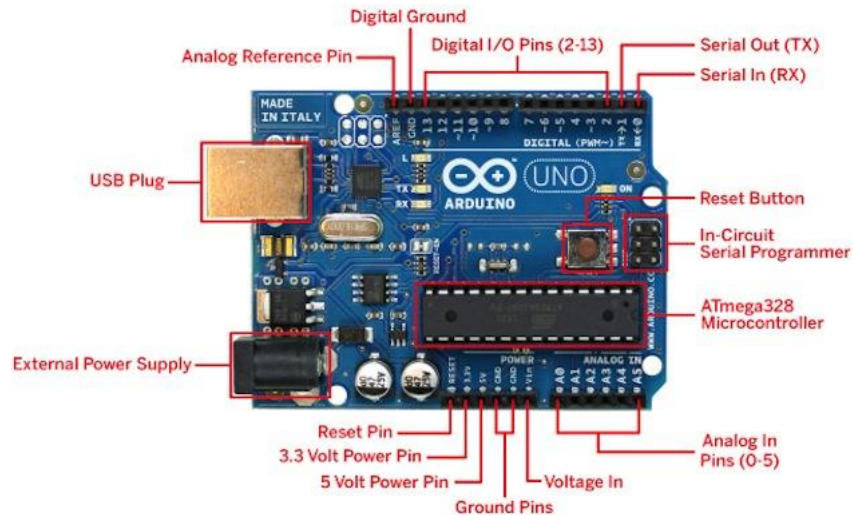
Quasi tutte le schede Arduino integrano poi altri componenti, alcuni specifici a seconda del modello, quali ad esempio un oscillatore ceramico, un pulsante di reset che fa ripartire la scheda in caso di stallo del programma in memoria, un jack per la connessione dell'alimentazione diretta.



Arduino Due è il primo modello ad implementare una CPU ARM Cortex

Su ogni board Arduino sono presenti infine dei pin (“piedini”) di input e output attraverso i quali il microcontrollore riceve le “informazioni” in ingresso e restituisce all’ambiente un segnale in uscita.

Tramite il software di Arduino i pin si possono programmare come porte di entrata o di uscita digitale o analogica e se ne può controllare il comportamento attraverso la programmazione ad alto livello dalla IDE di Arduino.



Arduino Uno: schema della componentistica e dei pin di Input/Output

E' questo il vero vantaggio di Arduino: operazioni semplici come l'accensione e lo spegnimento di un LED tramite programmazione di un microcontrollore in passato avrebbero richiesto la scrittura di un ampio quantitativo di codice Assembly, da "flashare" (ovvero registrare nella memoria) sul chip tramite apposito socket collegato tramite porta seriale ad un computer.

Grazie ad Arduino e la sua IDE di programmazione, la stessa operazione si può effettuare scrivendo poche righe di codice di alto livello e collegando i piedini del LED ai due PIN dell'Arduino programmati per emettere un segnale elettrico in uscita.

Il numero di ingressi ed uscite che si possono utilizzare, e dunque il numero di ingressi ed uscite disponibili su una scheda Arduino, dipendono dalle specifiche del microcontrollore. Il loro numero cambia a seconda del modello di chip che monta ogni specifica versione dell'Arduino.

La "revisione 3" dell'Arduino Uno, che ad oggi è considerato il modello base di riferimento, monta 14 pin di input/output digitale, dei quali 6 possono essere utilizzati come output PWM (modulazione della larghezza d'impulso), e 6 pin per l'input analogico, utili per la lettura di segnali elettrici analogici come quelli prodotti da un'ampia gamma di sensori. L'Arduino Due, la recente versione basata su CPU ARM Cortex-M3 a 32

bit, può permettersi un numero complessivo di pin molto più alto, con 54 piedini di input/output digitale (di cui 12 utilizzabili come uscite PWM), 12 input analogici ed anche 4 porte seriali hardware.

Le schede Arduino funzionano con una tensione in ingresso di 5V che può essere fornita direttamente dalla connessione USB. La maggior parte delle schede integra anche un connettore per l'alimentazione diretta con una tensione in ingresso che può arrivare fino a 9V. Unica eccezione è Arduino Due, che invece funziona con una tensione in ingresso di 3,3 V.

Arduino si può estendere attraverso le cosiddette “shield”, schede aggiuntive che si possono montare sui pin della scheda principale per ampliarne le funzionalità e controllare ad esempio dei servo motori (Motor Shield), riprodurre file WAV da una scheda di memoria (Wave Shield) o connettere l'Arduino ad una rete wireless (Wi-Fi Shield). Alle estensioni di Arduino è dedicata la terza sezione di questo capitolo.

I modelli di Arduino

In questa sezione ho raccolto alcune schede specifiche dei modelli più diffusi e più recenti di Arduino, con le principali caratteristiche tecniche, le destinazioni d'uso più probabili.

I modelli analizzati sono i seguenti:

- Arduino Uno
- Arduino Due
- Arduino Leonardo
- Arduino Esplora
- Arduino Mega 2560
- Arduino Nano
- LilyPad Arduino (modello realizzato da Sparkfun)

La linea completa di Arduino comprende ulteriori modelli, per un totale di 19, che non tratterò nello specifico. Fra questi meritano una menzione rapida anche:

- Arduino Robot (*board dotato di ruote e motori per le implementazioni di piccoli e grandi robot in grado di muoversi*)
- Arduino Mega ADK (*integra un chip MAX3421e che semplifica la connessione ad un terminale Android*)
- Arduino Ethernet (*stesse caratteristiche di un Arduino Uno, ma integra una porta Ethernet per la connessione ad una rete LAN*)
- Arduino BT (*simile all'Arduino Uno, integra un modulo Bluegiga WT11 per la connettività Bluetooth*)
- Arduino Micro (*concetto simile all'Arduino Nano, ma derivato da Arduino Leonardo. Ha un numero maggiore di pin di I/O rispetto ad Arduino Nano*)
- Arduino Pro (*scheda senza pin pre-saldati, pensata per l'installazione permanente in oggetti o opere d'arte*).

PROGETTO DI UN CRUSCOTTO TOUCH SCREEN IN ARDUINO

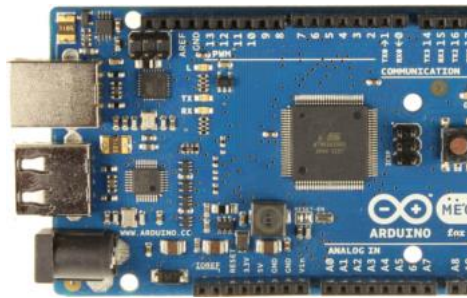
Si vuole implementare un display controllato da Arduino in grado di pilotare e visualizzare lo stato di un diodo LED ed inoltre visualizzare anche il valore misurato da un sensore connesso ad esso.

A tale scopo è stato utilizzato il seguente materiale:

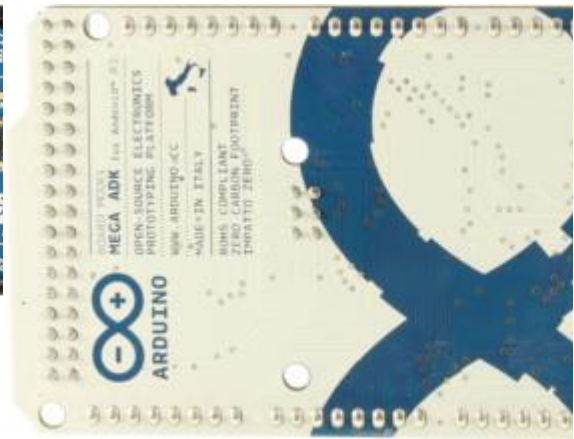
- Microcontrollore Arduino (Mega ADK);
- Display Touch Screen
- Bread board;
- Resistenza da 150 Ohm e diodo LED
- Resistenza e fotoresistenza da 10 KOhm (sensore di luminosità)
- Bread board e cavetti opportuni.

Arduino MEGA ADK

E' il modello di Arduino che è stato utilizzato nell'implementare il touch screen ed è per questo che se ne presentano alcune delle sue caratteristiche:



*Arduino MEGA ADK R3
frontale*



Arduino MEGA ADK R3 Indietro

Essa è una scheda basata sul microcontrollore ATmega2560.

Ha un'interfaccia utente di tipo USB per il collegamento con i cellulari basati su Android inoltre dispone 54 pin digitali di tipo I/O dove in particolare:

- 15 pin possono essere utilizzate come uscite PWM - Pulse With Modulation);
- 16 pin sono ingressi analogici dove 4 dei quali sono di tipo UART (ovvero porte seriali hardware)
- Un oscillatore da 16 MHz, un connettore USB, un jack di alimentazione, un Header ICSP (In Circuit Serial Programming) e un pulsante di reset.

Come detto in precedenza, Mega ADK è basato sull' ATmega2560, di seguito sono riportate sommariamente le caratteristiche principali:

Microcontroller	ATmega2560
Tensione di funzionamento	5V
Tensione in ingresso (consigliato)	7-12V
Tensione in ingresso (limiti)	6-20V
Pins Digital I / O	54 (di cui 15 forniscono PWM)
Pins di ingresso analogico	16
Corrente di CC per Pin O / I	40 mA
Corrente DC per 3.3V Pin	50 mA
Memoria Flash	256 KB di cui 8 KB utilizzati da bootloader
SRAM	8 KB
EEPROM	4 KB
Frequenza di clock	16 MHz
USB Host Chip	MAX3421E
Lunghezza	101,52 millimetri
Larghezza	53,3 millimetri
Peso	36 g

Alimentazione

Il MEGA ADK Arduino può essere alimentato attraverso la connessione USB o con un alimentatore esterno che vengono selezionate automaticamente dallo stesso dispositivo.

L'alimentazione esterna (non USB) può essere fornita da un alimentatore AC-DC o la batteria.

La scheda può operare su un alimentatore esterno da 5,5 a 16 volt. Se fornito con meno di 7V, tuttavia, il pin da 5V può fornire meno di cinque

volt e la scheda può essere instabile. Se si utilizzano più di 12 V, il regolatore di tensione può surriscaldarsi e danneggiare la scheda. Il range di tensione raccomandato è 7-12 volt.

Memoria

La MEGA ADK ha 256 KB di memoria flash per memorizzare il codice (di cui 8 KB è utilizzato per il bootloader), 8 KB di SRAM e 4 KB di EEPROM (che può essere letta e scritta con la libreria EEPROM).

Ingresso e uscita

Ciascuno dei 50 pin digitali della MEGA ADK può essere usato come ingresso o uscita, utilizzando le funzioni `pinMode ()`, `digitalWrite ()`, e `digitalRead ()`. Essi operano a 5 volt. Ogni pin può fornire o ricevere un massimo di 40 mA ed ha una resistenza di pull-up interno (disconnessa di default) di 20-50 kOhms. Inoltre, alcuni pin hanno funzioni specializzate.

La MEGA ADK ha 16 ingressi analogici, ciascuno dei quali fornisce 10 bit di risoluzione (cioè 1024 valori differenti). Per default misurano da terra a 5 volt, anche se è possibile cambiare l'estremità superiore del loro intervallo utilizzando il pin AREF e la funzione `analogReference ()`.

Comunicazione

Arduino MEGA ADK ha una serie di servizi per la comunicazione con un computer, un altro Arduino o altri microcontrollori.

L' ATmega2560 fornisce quattro hardware UART per TTL (5V) di comunicazione seriale.

Un ATmega8U2 su uno di questi canali a bordo oltre alla USB fornisce una porta COM virtuale per il software sul computer (i computer Windows

avranno bisogno di un file inf, ma le macchine OSX e Linux riconoscerà la scheda come una porta COM automaticamente.

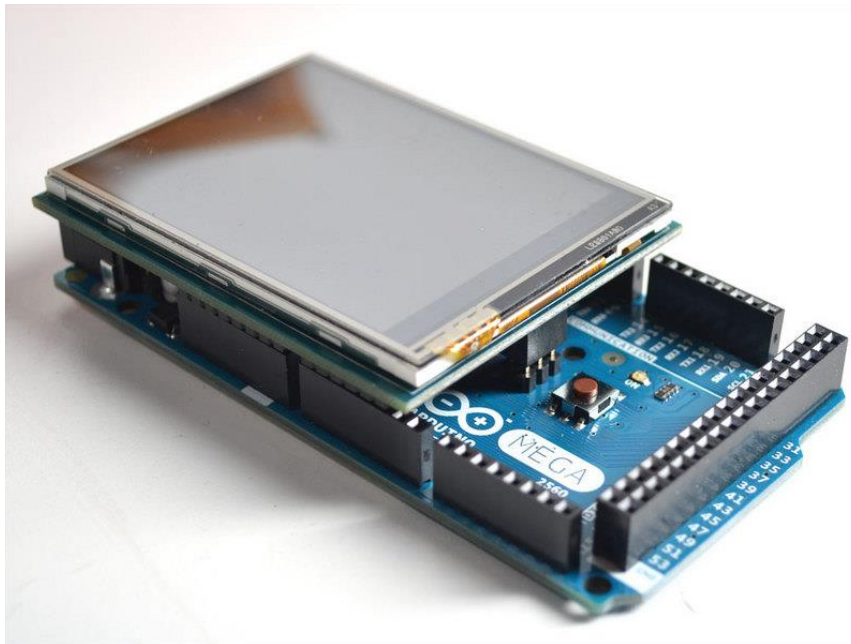
L'interfaccia utente USB permette ad Arduino MEGA ADK di connettersi e interagire a qualsiasi tipo di dispositivo che dispone di una porta USB. Ad esempio, consente di interagire con molti tipi di telefoni, controllare fotocamere Canon, interfacciandosi con tastiera, mouse e controller giochi come Wiimote e PS3 .

Arduino - Display TFT LCD

Un display TFT LCD è una variante dei display a cristalli liquidi (LCD) che utilizza la tecnologia dei Thin-Film Transistor (TFT) e, a differenza di altri LCD a matrice passiva, si avvale di uno schermo a matrice attiva nel quale ogni singolo pixel è collegato a un transistor e ad un condensatore che ne mantengono attivamente lo stato. Questo tipo di gestione, aumentando il numero di pixel che lo schermo è in grado di supportare anche il contrasto, rende possibile migliorare alcune delle caratteristiche delle immagini visualizzate.

Lo schermo TFT LCD preso in considerazione è da 2.8 pollici (7.1 cm) ed è un multicolore con un touch screen resistivo a 4 conduttori. Inoltre i vari modelli esistenti in commercio sono compatibili con i sistemi Arduino /Arduino Mega e tale integrazione è resa possibile da connettori alla base di supporto del display compatibili con quelli presenti sulla scheda del microcontrollore.

Di seguito è mostrato il display touch screen supportato dal microcontrollore Arduino che è stato utilizzato nell'esperienza:



Nella tabella seguente vengono mostrate le sue caratteristiche tecniche del modello adottato nel progetto:

2.8" TFT LCD Touch Shield V2.0 [[SeeedStudio](#)]

Caratteristiche	Valore	Unità
Alimentazione	5	Volt
Assorbimento	250	mA
Dimensioni LCD	2.8"	Pollici
Angolo di visione	60~120	Gradi
Risoluzione	320x240	Pixel
Colori LCD	65k	
Retroilluminazione	LED	
LCD driver IC	ILI9341	
Interfaccia	SPI	
Touch Screen	4-Wire resistive touch screen	
Area Attiva	43.2*57.3	mm
ESD contact discharge	±4	KV
ESD air discharge	±8	KV
Dimensioni	72.5x54.7x18	mm
Peso	24±2	g

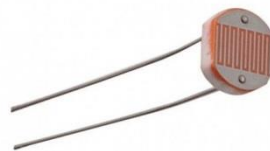
Nell'esperienza lo schermo è stato applicato sul microcontrollore Arduino che a sua volta è stato connesso nei suoi pin con una bread board sulla quale sono stati montati (al netto delle opportune resistenze) sia un diodo LED che una fotoresistenza.

Sia il diodo che la fotoresistenza sono stati connessi ai pin di Arduino per consentire la comunicazione con il microcontrollore stesso e consentire al display Touch Screen le funzionalità congiunte nell'attivare/disattivare il LED tramite un opportuno tasto posto sul display e visualizzare il valore misurato dal sensore di luminosità che è appunto la fotoresistenza.

Fotoresistenza

Le fotoresistenze (o fotoresistori) sono resistenze composte da uno strato semiconduttore che reagisce alla radiazione luminosa, sia quest'ultima emessa dal sole o da altra fonte luminosa.

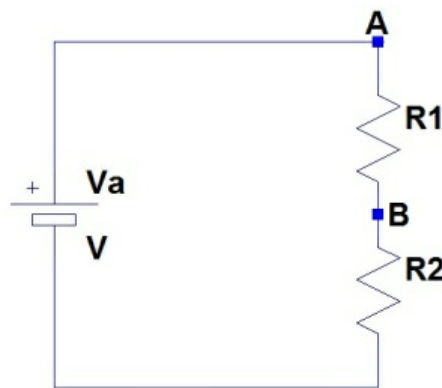
E' quindi un componente elettronico la cui resistenza è inversamente proporzionale alla quantità di luce che lo colpisce. Si comporta come un tradizionale resistore, ma il suo valore in Ohm diminuisce mano a mano che aumenta l'intensità della luce che la colpisce. Ciò comporta che la corrente elettrica che transita attraverso tale componente è proporzionale all'intensità di una sorgente luminosa. In tale maniera si realizza una sorta di potenziometro attuabile tramite la luce anziché tramite forze meccaniche o segnali elettrici.



Il funzionamento di una fotoresistenza, è semplice varia il valore da 0Ω alla sua resistenza massima $10,50,100K\Omega$, in base al tipo di fotoresistenza, quando passa dalla condizione di illuminazione piena a buio.

In pratica se la luce è sufficientemente forte il suo valore in ohm è 0, man mano che la luce diminuisce il suo valore aumenta fino alla resistenza massima.

Ai fini di visualizzare i valori della fotoresistenza tramite il microcontrollore Arduino si realizza il circuito come un semplice partitore di tensione connettendo la tensione misurata ai capi di essa con uno dei pin analogici di Arduino.



il principio di funzionamento è: due resistenze, di eguale valore, collegate come nello schema, dividono in parti uguali la tensione erogata dalla batteria, nel punto B si misurerà $Volt/2$, in questa configurazione R1 ed R2 devono essere identiche.

Nell'esperienza infatti la fotoresistenza utilizzata è di 10 KOhm , cosicché nel partitore considerato nei casi limite in cui essa è colpita dalla massima intensità luminosa, allora nel punto B verrà a trovarsi esattamente 0 V mentre quando ci sarà buio al punto B avrai $V/2 = 5\text{V} / 2 = 2,5\text{ V}$ in quanto Arduino è alimentato da una tensione nominale pari a 5 Volt .

Ovviamente a seconda della luminosità misurata ci saranno anche delle situazioni intermedie e quindi in generale assumendo i 5 Volt della tensione nominale, nel punto B si misureranno valori di tensione che variano nell'intervallo $V \in [0, 2.5\text{ V}]$.

Tutti questi valori verranno visualizzati sul display in una scala che varia da a 0 a 1023 in quanto il microcontrollore opera con 10 bit.

Implementazione di un cruscotto touch screen in Arduino

Per consentire l'utilizzo del display, nello sketch di Arduino prima della routine di void setup() oltre alle usuali librerie occorre aggiungere le seguenti librerie:

```
1 #include <stdint.h>
2 #include <TFTv2.h>
3 #include <SPI.h>
```

Grazie a queste librerie si rendono possibile sia le funzioni grafiche del display che l'utilizzo del Touch Screen.

A seguire dopo aver definito anche le opportune variabili, nella routine di setup il codice è strutturato come segue:

- Definizione dei pin opportuni sia analogici che digitali ai quali è connesso il display sul microcontrollore Arduino;

In particolare il LED è stato connesso al pin 14 di tipo digitale mentre il sensore è stato connesso al pin 15 di tipo analogico ed entrambi di tipo output

- Inserimento della seguente funzione che consente l'inizializzazione della libreria TFT

```
void setup()
{
  Tft.TFTinit();
  // altre inizializzazioni
}
```

Si prosegue con la routine di loop() dove risiede il cuore delle istruzioni da far eseguire al microcontrollore.

In particolare in tale routine vengono inserite delle funzioni opportune che consentono il puntamento e la registrazione dei punti (x,y) rappresentanti il piano cartesiano del display e poi a seguire sono state scritte le opportune

condizioni che consentono la visualizzazione del tasto di accensione/spengimento che risiede nel semipiano superiore del display mentre in quello inferiore le istruzioni necessarie per la visualizzazione del valore riportato dal sensore.

Di seguito si riportano le righe di codice principali impiegate nel progetto:

```
/ punta un oggetto e registra le coordinate x y e z
```

```
Point p = ts.getPoint();
```

```
p.x = map(p.x, TS_MINX, TS_MAXX, 0, 240);
```

```
p.y = map(p.y, TS_MINY, TS_MAXY, 0, 320);
```

```
/ considera il semipiano superiore per attivare e disattivare il LED tramite il tasto di accensione chiamato buttonRED
```

```
if (p.x > 0 && p.x < 80 && p.y > 160 && p.z > 300 && buttonRED == LOW)
```

```
{
```

```
digitalWrite(14, HIGH);
```

```
buttonRED = digitalRead(14);
```

```
Tft.fillScreen(0, 239, 0, 160, GREEN);
```

```
} else if(p.x > 0 && p.x < 239 && p.y > 160 && p.z > 300 && buttonRED== HIGH)
```

```
{
```

```
digitalWrite(14, LOW);
```

```
buttonRED = digitalRead(14);
```

```
Tft.fillScreen(0, 239, 0, 160, RED);
```

```
}
```

Per quanto riguarda la parte di codice inerente alla fotoresistenza, le righe sono le seguenti

```
// fotoresistenza
```

```
// visualizza stringa o arrays di caratteri:
```

```
Tft.drawString(h1, 40,200, 3, YELLOW);
```

```
sensorValue = analogRead(15); // leggi il pin 15
```

```
Serial.println(sensorValue);
```

```
// refresh
```

```
// Tft.fillScreen(z1,x2, y1, y2,COLOUR);
```

```
Tft.fillScreen(0,240, 240, 320,BLACK);
```

```
// visualizza numero intero : Tft.drawNumber(integer, x, y, size, colour);
```

```
Tft.drawNumber(sensorValue,80, 240, 4, YELLOW);
```

```
}
```

Conclusioni

Questa seconda esperienza che è appunto quella di realizzare un cruscotto touch screen basata su Arduino, può ritenersi fattibile in quanto rispetta gli obiettivi prefissati.