

Coding con Arduino...

Introduzione

Coding

Il “coding” è un termine inglese che può essere tradotto in italiano con la parola “programmazione”, ovvero digitare codice; questo termine è ovviamente legato al mondo dell’informatica, ma in questo corso non verrà semplicemente insegnata dell’informatica nel senso tradizionale del termine.

Obiettivo del corso

Questo corso non ha come scopo quello di insegnare solamente un linguaggio di programmazione, ma bensì quello di fornire degli strumenti trasversali che potranno essere applicati a diverse discipline.

In particolare attraverso l’utilizzo di un microcontrollore verranno affrontati argomenti come quello di algoritmo, sequenza, variabile, eventi, “cicli” e “scambio di messaggi”.

Microcontrollore: Arduino

Gli argomenti citati nel precedente paragrafo verranno visti avvalendosi di un microcontrollore opportunamente programmato.

Un microcontrollore è un dispositivo elettronico integrato su un singolo chip; ovvero è un oggetto di dimensioni relativamente ridotte che al suo interno contiene delle memorie per i dati e per i programmi, delle porte di input/output per interfacciarsi con il mondo esterno e tutto il necessario per eseguire delle computazioni.

Le caratteristiche di questo tipo di dispositivo sono il consumo ridotto di energia che richiede per il suo funzionamento, il fatto che sia veloce, ri-programmabile e che possa essere utilizzato per un’ampia gamma di applicazioni.

Arduino è un progetto italiano nato nel 2005 che ha prodotto una piattaforma di prototipazione elettronica open-source; tale piattaforma include componenti hardware come i microprocessori “Arduino UNO” e software, come gli ambienti di sviluppo per programmarli.

Essendo un progetto open-source, le caratteristiche della struttura dei microprocessori Arduino sono di dominio pubblico quindi numerose aziende hanno creato dei microprocessori con la medesima architettura ma con nomi diversi.

In questo corso infatti verrà usata una scheda denominata “Elegoo UNO R3” che presenta le medesime caratteristiche di Arduino ma ha un costo inferiore.

IDE: Snap4Arduino

Come accennato nel precedente paragrafo il progetto di Arduino include gli ambienti di sviluppo per programmarlo: questi sono dei programmi particolari sui quali viene scritto il codice da caricare sul microcontrollore.

La caratteristica principale di questi programmi è che aiutano i programmatori a scrivere il codice. Ogni linguaggio di programmazione ha delle “parole chiave” che se scritte hanno delle funzionalità particolari e questi programmi aiutano a riconoscerle e scriverle correttamente.

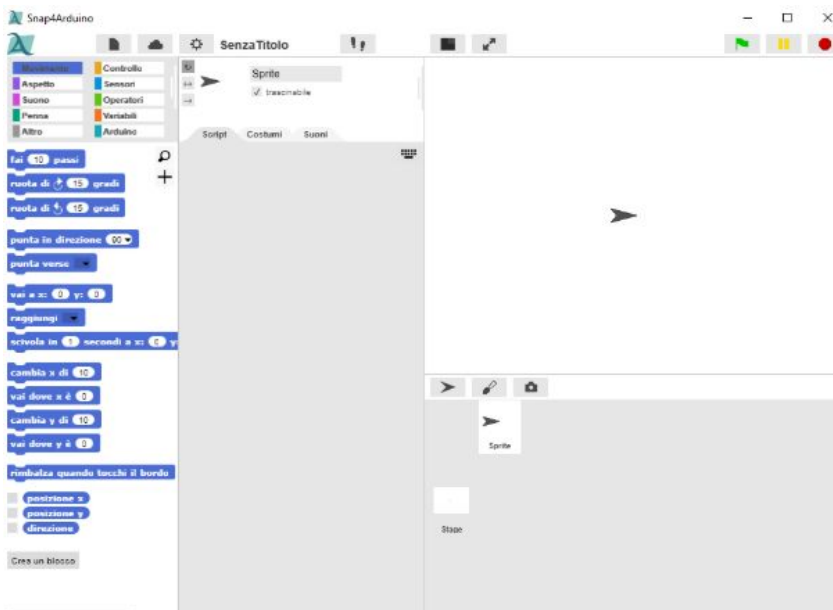
Per esempio, in Instagram quando si vuole “taggare” una persona si deve mettere una “@” prima del nome e l’applicazione suggerisce i contatti che possono essere taggati.

In un IDE quelle parole che hanno delle funzioni particolari vengono evidenziate, per far capire al programmatore che quella parola è diversa dalle altre, inoltre come in Instagram l’ambiente può suggerire come completare un’istruzione, o sottolinearla se si accorge in anticipo che è stata scritta in maniera errata.

Gli IDE sono degli strumenti particolarmente potenti ed utili quando si programma.

Normalmente Arduino viene programmato con “Arduino IDE”, basato a sua volta su un altro IDE chiamato “Wiring”, tuttavia tale piattaforma richiede che i programmi vengano scritti in un linguaggio di programmazione denominato “C”. Lo scopo di questo corso non è quello di insegnare il linguaggio di programmazione C, ma quello di dare le basi dei principali concetti utili per programmare Arduino e quello di esplorare le sue possibili applicazioni. Per questo si è scelto di programmare Arduino tramite Snap4Arduino.

Snap! è un popolare ambiente di programmazione che utilizza un linguaggio di programmazione di tipo visuale. Snap4Arduino è una sua estensione che contiene una categoria di istruzioni per poter programmare i microcontrollori.



In Snap4Arduino le istruzioni che il microcontrollore deve eseguire non devono essere scritte “a mano”, ma basterà sfruttare i “blocchi” pre-esistenti concatenandoli tra loro per costruire l’algoritmo.

Ogni blocco appartiene ad una delle nove categorie definite da Snap!, ciò non toglie si possono creare dei nuovi blocchi che descrivono un nuovo comportamento e per creare un linguaggio specifico per il micromondo nel quale si sta lavorando.

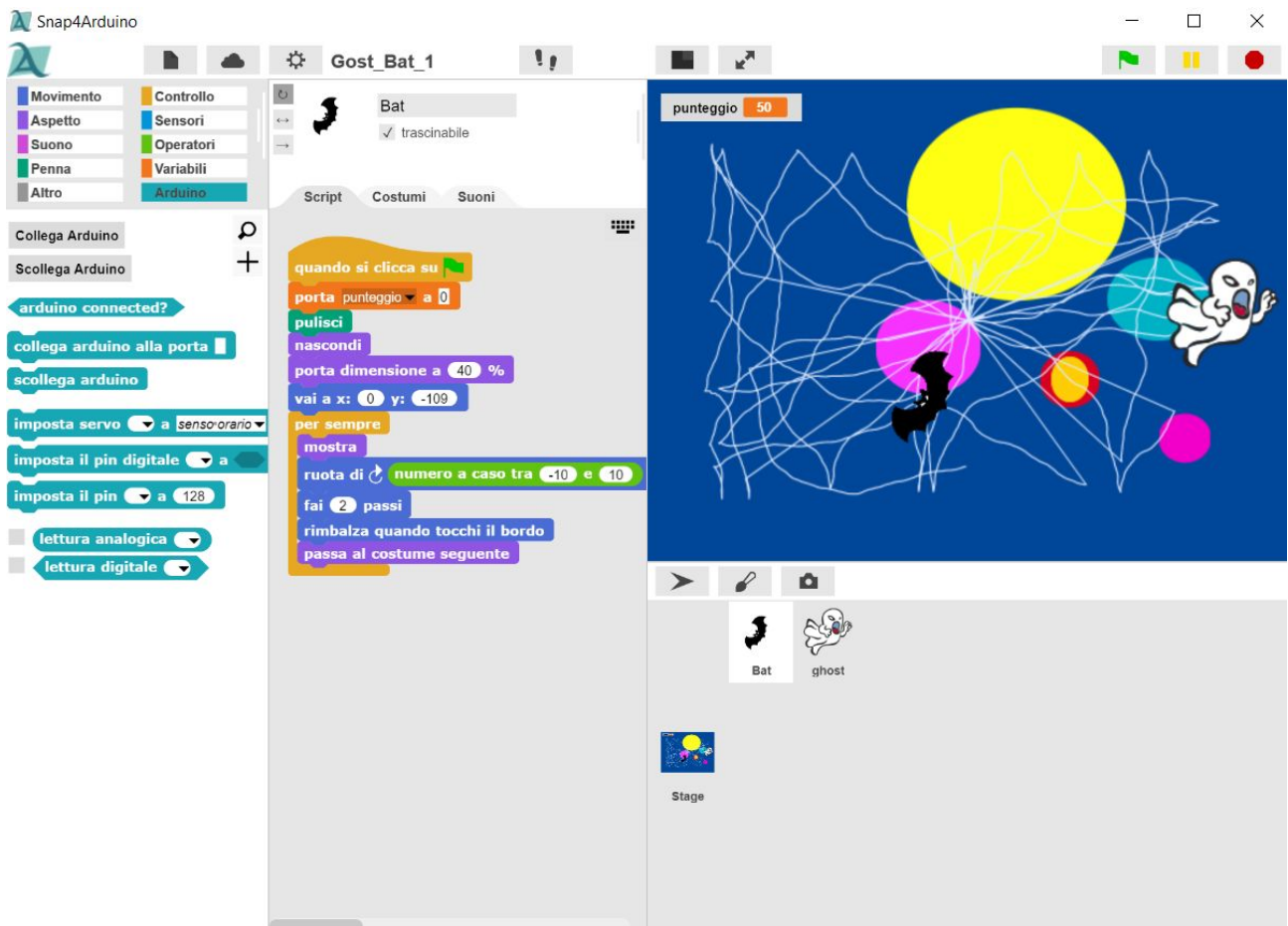
Snap!: Micromondi

Ogni progetto creato su Snap!, può essere visto come un piccolo universo a sé stante dove chi lo crea si può sentire libero di esplorare qualsiasi argomento, si parla di “micromondo”.

Un micromondo può essere usato per creare un gioco, come Arkanoid e Snake, per fare dello StoryTelling, simulazioni, biglietti di auguri personalizzati, o qualsiasi altro tipo di progetto.

Snap non ha un target specifico di progetti che possano essere caricati, ma lascia gli utilizzatori liberi di esplorare ed esprimere la propria fantasia come meglio credono.

Uno degli obiettivi del corso è spingere i ragazzi a creare dei micromondi che non siano limitati al lato software, ma che siano capaci di interagire con il mondo fisico rappresentato dal microprocessore e da tutti i suoi sensori e attuatori.



Snap4Arduino: Arduino

Snap4Arduino permette, come già detto in precedenza, di programmare un microcontrollore tramite i blocchi contenuti nella sezione denominata "Arduino".

Svolte tutte le operazioni preliminari contenute nel blocco: "Come usare Arduino IDE e Snap4Arduino", per programmare da Snap4Arduino è necessario: premere sul blocco nella sezione precedentemente citata chiamato "Collega Arduino", se una volta fatto, appare la scritta "Arduino connesso, buona sperimentazione", allora si può procedere a programmare, in caso contrario, si veda il documento sopra citato per la risoluzione del problema.

Risolti tutti gli eventuali inconvenienti, si può iniziare a pensare al programma che si intende creare. In generale, come già visto in precedenza i componenti che possono essere collegati al microcontrollore sono di due tipi: output, in caso di attuatori e input, in caso di sensori.

Per i dispositivi di output in generale le operazioni più sensate da compiere sono quelle riguardano il "dire al componente cosa fare": che si traducono nei blocchi di Arduino con:

imposta il pin ▼ a ○ per i pin analogici

imposta il pin digitale ▼ a ◀ per i pin digitali

Al contrario per i dispositivi di input è "interessante" leggere il valore che essi rilevano tramite:

lettura analogica ▼

lettura digitale ▼

Tuttavia non bisogna dimenticarsi che l'IDE necessita di sapere quando deve fare eseguire le operazioni al microcontrollore. Tale informazione gliela si può impartire tramite un blocco "quando...":

quando si clicca su 🚩

quando sono cliccato ▼ *

quando si preme il tasto spazio ▼ *

*Questi due blocchi permettono una serie di opzioni, basta premere sulla freccia nera per vederle.

quando ◀ blocco generico dove può essere inserita una qualsivoglia condizione, come ad esempio:

quando ▶ **arduino connected?**

Algoritmo

L'algoritmo è uno degli aspetti fondamentali che verranno affrontati in questo corso; con questo termine si intende un procedimento composto da una serie finita di passaggi che hanno lo scopo di risolvere un determinato tipo problema o di far raggiungere un obiettivo.

Questo argomento come precedentemente accennato non è solamente legato all'informatica, ma è alla base di molteplici attività della vita quotidiana: qualsiasi problema che ci si ponga può essere risolto tramite un algoritmo più o meno complesso.

Per esempio, il fatto di dover prepararsi per andare a scuola puntuali può essere visto come un problema che può essere risolto eseguendo le seguenti operazioni: puntare la sveglia e preparare lo zaino la sera prima, alzarsi quando suona, fare colazione, lavarsi, vestirsi, prendere lo zaino ed uscire di casa.

Tutte queste azioni compongono l'algoritmo per prepararsi per andare a scuola.

Sequenza

La sequenza è un altro costrutto fondamentale per qualsiasi disciplina; questo termine in italiano ha molteplici significati, ma per questo corso con questa parola si intendono un insieme di istruzioni disposte in un ordine prestabilito.

Questo aspetto è così importante perché stabilire l'ordine in cui compiere le azioni è alla base di ogni compito svolto nella giornata.

Per esempio, quando inizia una lezione e bisogna scrivere quello che dice il professore i passaggi da compiere possono essere i seguenti: tirare fuori il quaderno, aprirlo e poi iniziare a scrivere.

Non è possibile prendere appunti senza aver prima aperto il quaderno.

Variabile

La variabile è un contenitore; ossia un oggetto che può contenere uno o, più valori che possono cambiare nel tempo. Ogni variabile è definita quindi da un contenuto e da un nome, che ne definisce il contenuto.

Per esempio, il contenitore che contiene i biscotti può essere visto come una variabile. Il suo nome (etichetta) potrebbe essere "Scatola dei biscotti". Il suo contenuto è dato dal numero di biscotti al suo interno e questo valore può cambiare in base a quanti biscotti si mangiano o a quanti biscotti si aggiungono.

Le variabili sono particolarmente utili perché forniscono un nome per usare un determinato elemento e quindi facilitano la scrittura del codice da parte del programmatore e ne aiutano anche la lettura.

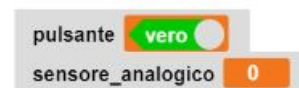
Note: Le variabili, appena create, hanno un valore pari a zero, sono vuote.

Il loro valore cambierà quando il programmatore deciderà di riempirle.

Nei linguaggi di programmazione classici, le variabili possono contenere un solo tipo di valore (numero, testo, "vero/falso") e appena si crea la variabile bisogna indicare quale tipo di valore può contenere.

In Sanp4Arduino questo non è necessario, perché il tipo di dato viene riconosciuto in automatico appena la variabile viene riempita.

Questo significa che se alla variabile viene associata la lettura del valore di un pulsante il suo valore potrà essere solo "vero", o "falso", al contrario se ad essa viene associata la lettura di un sensore analogico essa avrà un valore di tipo intero.

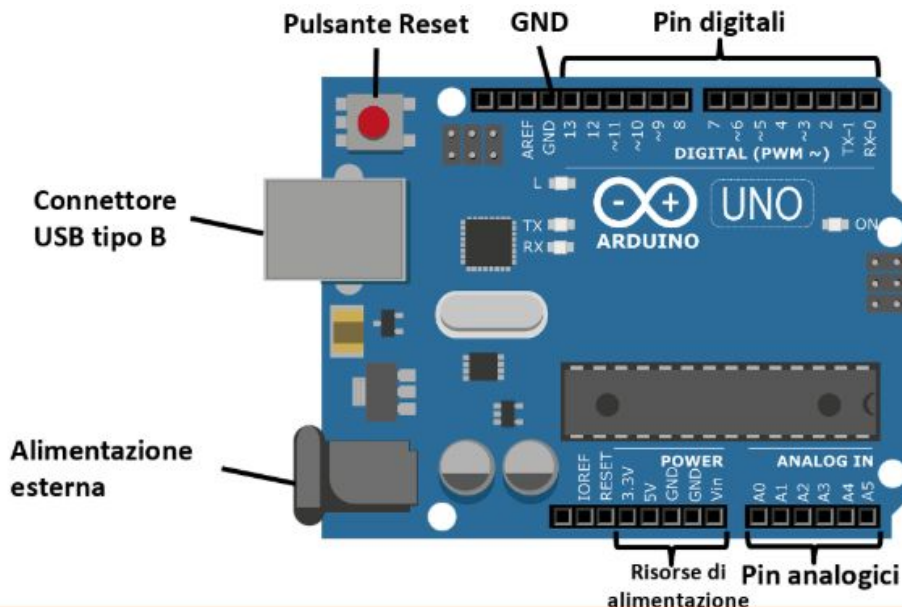


Debugging

Il debugging è l'operazione effettuata dal programmatore quando analizza il codice alla ricerca degli errori, o di quelli che potrebbero essere dei possibili errori futuri.

Arduino : struttura fisica

Arduino è un microcontrollore che funziona a 16 MhZ.



Il microprocessore che verrà utilizzato ha diverse componenti tra le quali:

- Un pulsante di reset: permette il riavvio della scheda;
- Una serie di pin:
 - GND: sono tre e forniscono un'alimentazione a 0 Volt ("messa a terra");
 - Vin: vi si può collegare una batteria esterna per alimentare il circuito;
 - 3.3V: se collegato al circuito gli fornisce 3.3 Volt di corrente;
 - 5V: se collegato al circuito gli fornisce 5 Volt di corrente;
 - Analogici: sono sei, vanno da A0 a A5:
 - Operano a una tensione di 5 Volt;
 - gestiscono segnali di tipo continuo che possono assumere un serie infinita di valori tra Vcc e Vin;
 - Digitali: sono quattordici, da 0 a 13,
 - operano a una tensione di 5 Volt;
 - gestiscono dei segnali "discreti", possono assumere solo due valori: 0 e 1;
 - Tra di questi ce ne sono tre: il 9, il 10 e l'11 (quelli con "~" "vicino) sui quali può essere attuata una tecnica di pilotaggio dei pin detta "PWM".
- Un connettore USB di tipo B: che permette di connettere la scheda al computer in modo tale accendere la scheda e da caricarvi sopra il programma creato.
- Una porta per l'alimentazione esterna; in modo tale da non dovere mantenere il microcontrollore attaccato al computer.

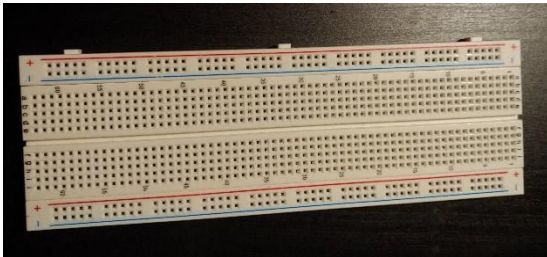
Componenti di base

Breadboard

La breadboard è una basetta di plastica con dei fori alla quale possono essere collegati dei componenti fisici come i led che poi possono essere collegati attraverso i “jumper” al microcontrollore.

Ci sono due tipi di fori:

- quelli al centro sono collegati verticalmente e generalmente vengono usati per collegare i dispositivi;
- quelli laterali (tra le righe rosse e azzurre) sono collegati orizzontalmente e vengono usati per collegare le alimentazioni;



breadboard

Jumper

I jumper sono dei cavetti particolari che permettono al microcontrollore di mandare o ricevere segnali ai diversi componenti fisici.

Possono avere diverse dimensioni e colori ma questo non incide sul loro funzionamento interno; a livello fisico sono tutti esattamente uguali

Ve ne sono di tre macro tipologie diverse: quelli maschio-maschio, quelli maschio-femmina e femmina-femmina; nel kit vi sono solo i primi due.



Jumper maschio-maschio

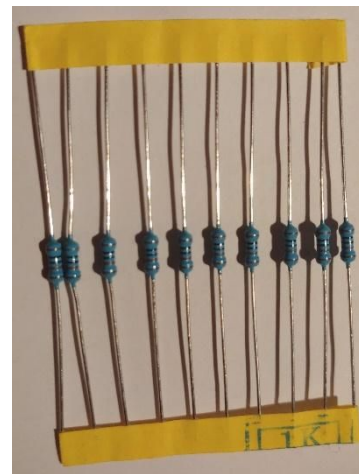
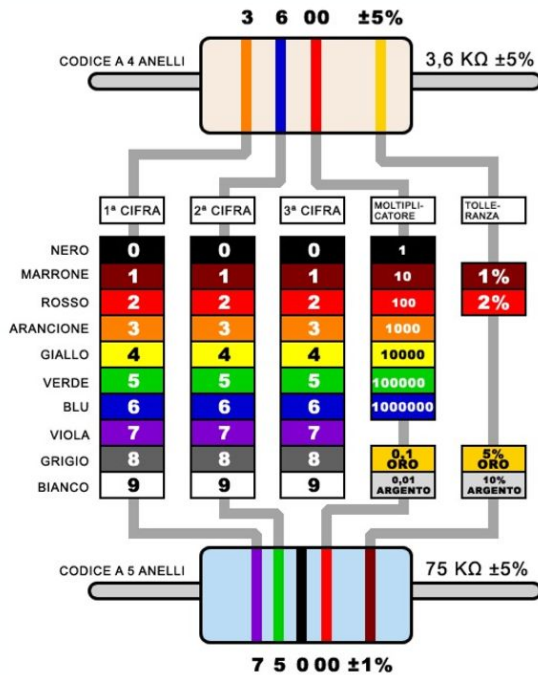


jumper maschio-maschio

Resistore

La resistenza dal punto di vista fisico è una grandezza che misura la tendenza di un conduttore ad opporsi al passaggio di una corrente elettrica quando è sottoposto a una tensione.

Per quello che riguarda questo corso i resistori sono dei particolari dispositivi fisici che a seconda della loro grandezza provocano una caduta di potenziale rispetto all'energia che dalla scheda forniamo al circuito; in altre parole dal momento che a tutti i componenti serve una quantità di corrente elettrica per funzionare, le resistenze permettono di dosarla in modo da far sì che i componenti ricevano solo quella che gli serve e non si "brucino".



resistore

Sensori e Attuatori

I dispositivi visti in questo corso appartengono a due categorie: sensori e attuatori.

I sensori sono dei dispositivi che rilevano un qualche tipo di dato dal mondo fisico e lo restituiscono al microcontrollore, gli attuatori invece a seconda di come vengono programmati producono un risultato sul mondo fisico.

Attuatori:

Led

Light Emitting Diode: è un componente fisico che se correttamente alimentato emette luce.

Questo dispositivo dal punto di vista è dotato di due terminali che hanno una polarità: quello più lungo è il terminale positivo e quello più corto è quello negativo, questa sua caratteristica fa sì che la corrente che gli viene fornita possa attraversare il led solo per un solo verso: in caso si invertano i terminali il led non si accende.



RGB

Il led RGB è un particolare tipo di led dotato di quattro piedini che gli permettono di variare il colore che assume.

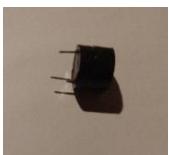
In particolare tre piedini vanno collegati a tre pin di input e uno a GND, a seconda del valore che viene dato a ciascuno dei tre pin che controllano il colore cambierà il colore assunto dal led.



Buzzer

Il buzzer è un dispositivo di output che opportunamente programmato produce un suono.

È dotato di due pin che vanno collegati a GND e a un pin digitale.



Fotoresistenza

La fotoresistenza è una resistenza composta da un particolare materiale capace di reagire alla radiazione luminosa, sia essa artificiale (lampada, torcia del telefono), o naturale (sole).

Questo dispositivo è dotato di due piedini che possono essere collegati in modo indifferente uno a 5V e l'altro a un pin analogico.

Su quest'ultimo pin potrà essere letta la luminosità rilevata dal sensore.



Sensori

Pulsante

Il pulsante è un sensore che può assumere due valori: basso e alto a seconda che sia stato premuto o meno.

È dotato di quattro piedini; che funzionano a coppie; per evitare malfunzionamenti esso va messo a cavallo del solco che divide a metà la breadboard in modo che i piedini non subiscano interferenze.

A questo punto considerando una coppia di piedini uno va collegato a GND e l'altro a un pin digitale.

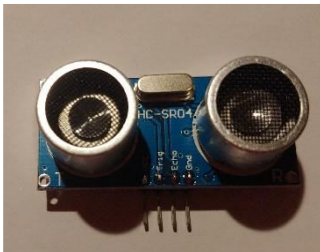


Sensore ad ultrasuoni

Il sonar è un dispositivo di prossimità che misura la distanza degli oggetti nelle vicinanze del suo "lato sensibile" del sensore stesso, senza che vi sia un contatto fisico.

È dotato di quattro pin:

- VCC: pin che deve essere collegato a +5 V;
- Trig: Trigger è il pin di input;
- Echo: Echo è il pin di output, attraverso questo pin verrà restituita la distanza rilevata;
- GND: pin che deve essere collegato a GND;



Potenzimetro:

Il potenziometro è sensore che restituisce un valore compreso tra X e Y a seconda di come viene girata la manopola.

È dotato di tre pin che vanno collegati a:

- GND;
- 5V;
- un pin analogico del microcontrollore.



Joystick

Il joystick non è altro che un attuatore che al suo interno ha due potenziometri e un pulsante.

Il rilevamento dei movimenti direzionali viene eseguito dai potenziometri, che misurano lo spostamento rispetto all'asse x e y del piano cartesiano; l'intervallo dei valori per ogni direzione va da 512 a 0 e da 512 a 1023.

La pressione della leva al contrario viene rilevata dal pulsante che viene azionato quando viene premuto il tasto di navigazione verso il basso.

È dotato di 5 pin che vanno rispettivamente collegati a:

- GND: GND;
- +5V: +5V;
- VRx: a un pin analogico;
- VRy: a un pin analogico;
- SW: a un pin digitale: rileva il segnale proveniente dal pulsante connesso alla leva del joystick;

