Misura di Temperatura mediante TMP36 [Tinkercad]

Obiettivo: Realizzare un controllo di temperatura mediante il dispositivo TMP36. Il TMP36 è il sensore di temperatura presente sul simulatore tinkercad.

Componenti elettronici:

- Arduino UNO
- Breadboard
- TMP36

Teoria: Il componente elettronico TMP36 è un dispositivo integrato ad alta precisione utilizzato per misurare la temperatura ambientale.
Dato il basso costo e l'ampia scala di valori ammissibili (ovvero da -40°C fino a 125°C) questi dispositivi sono particolarmente diffusi. E' inoltre importante considerare che non è necessaria nessuna operazione calibrazione per ottenere valori di accuratezza pari a ±1°C ad una temperatura di circa +25°C e ±2°C nel range di temperature –40°C to +125°C.

Questo dispositivo è caratterizzato da tre differenti pin ed un corpo semi-cilindrico. Guardando il lato piatto del dispositivo, il pin di sinistra è l'alimentazione (5V), il pin di destra la massa (GND), mentre sul pin centrale viene generata una tensione funzione della temperatura. La temperatura può essere pertanto misurata attraverso una lettura analogica sul pin centrale effettuata mediante il controllore Arduino.

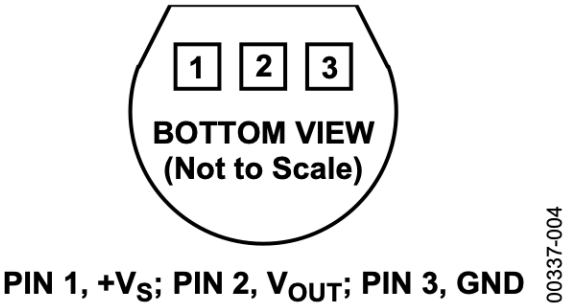


Figure 4. T-3 (TO-92)

TMP36 Package

Come riportato in precedenza è possibile utilizzare un pin di input analogico per ottenere il valore di temperatura mediante l'istruzione di analogRead. Nel caso specifico, osservando il grafico che riporta la caratteristica tensione/temperatura (per il **TMP36** la linea è evidenziata in rosso) per una tensione di uscita di 0.5V il sensore rileva la temperatura di 0°C. Pertanto valori di tensione inferiori a 0.5V indicano una temperatura sotto lo zero, mentre valori di tensione superiori a 0.5V indicano una temperatura positiva. Inoltre, è importante considerare che "una variazione di grado corrisponde ad una variazione di tensione di 10mV". Quindi, se sul pin di input analogico sono presenti 550mV significa che il sensore sta rilevando una temperatura di 5°C (550mV – 500mV = 50 mV variazione di 5°C).

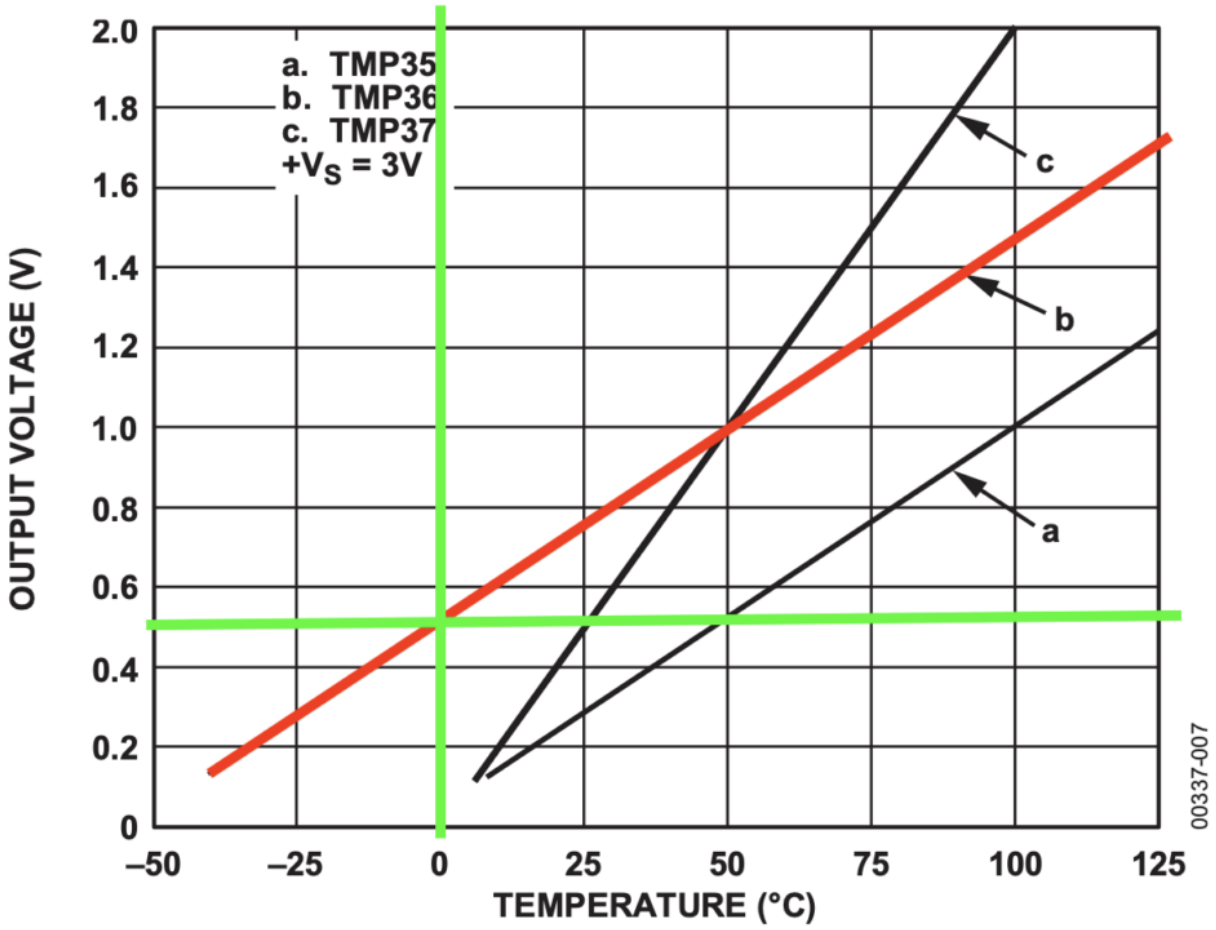
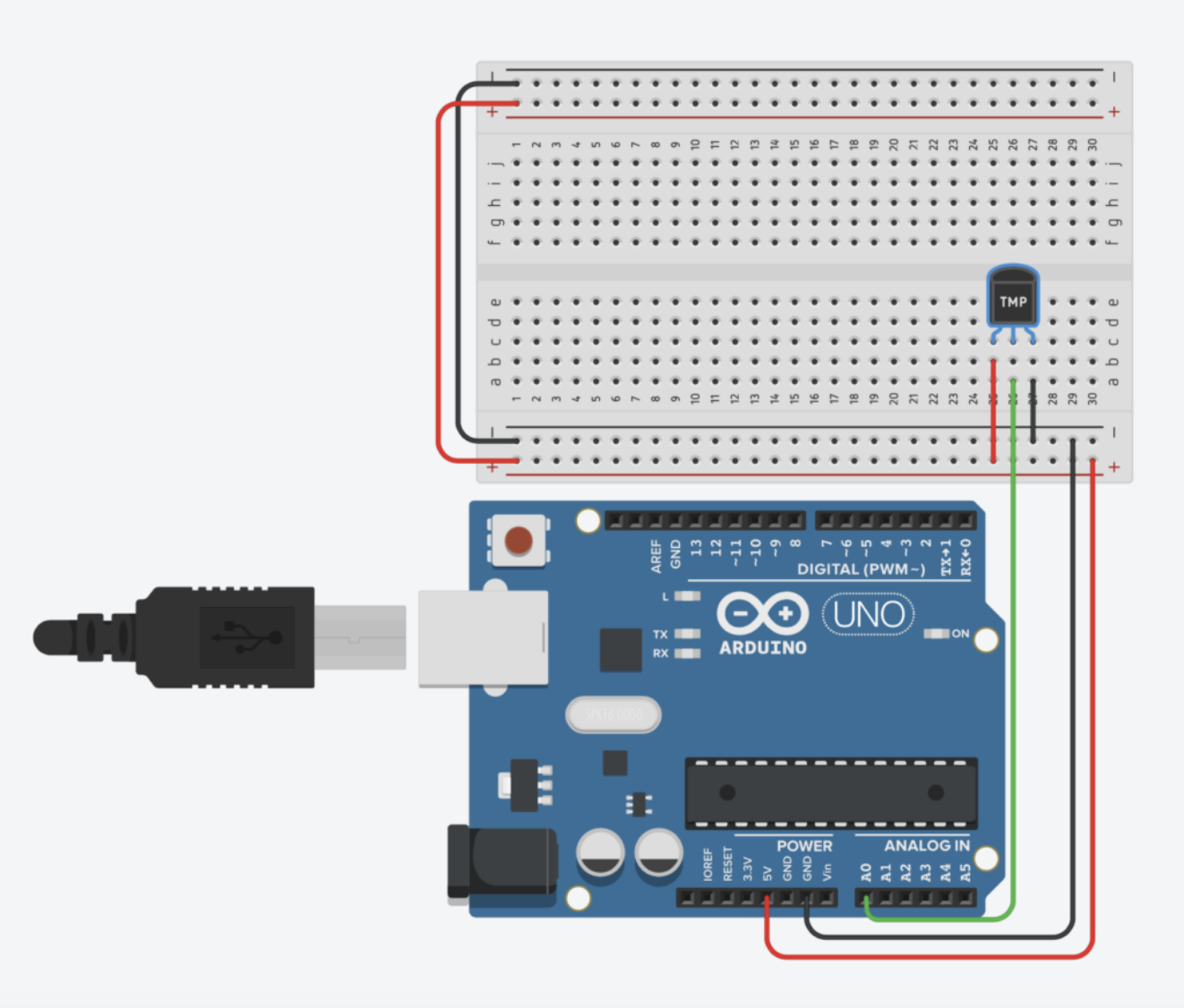


Figure 6. Output Voltage vs. Temperature

Caratteristica tensione corrente

Collegamento Circuitale:

Viene in seguito riportato lo schema elettrico utilizzato per valutare la temperatura mediante il dispositivo elettronico TMP36



Collegamento Circuitale

<u>Codice</u>:

La temperatura viene determinata attraverso una lettura analogica ed una opportuna scalatura.
Nel dettaglio, la tensione prodotta dal componente TMP36 viene letta utilizzando il pin analogico A0 e mappata in un intervallo 0-1023. Considerando che alla temperatura di 0 gradi la tensione misurata è pari a 0.5V e che per ogni grado percepito si ha un incremento di tensione di 10mV è opportuno riportare il valore letto attraverso la funzione analogRead in mV ed eseguire un’opportuna scalatura. Pertanto se il valore letto mediante l’istruzione di analogRead è memorizzato in una

variabile denominata *valTMP* la temperatura può essere ottenuta mediante la seguente formula:

*temperatura = (((valTMP5.0)/1023.0)-0.5)*100*

Tinkercad

Personalizzazioni:

E’ possibile modificare il circuito aggiungendo una ventola che si accende in modo automatico superata una determinata temperatura.

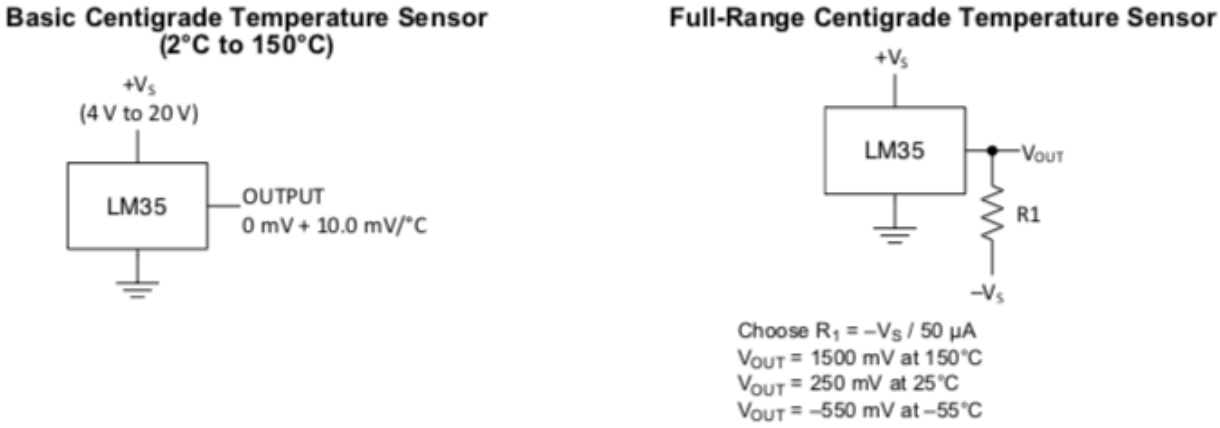
Controllo di Temperatura mediante LM35

Obiettivo: Realizzare un controllo di temperatura mediante il dispositivo LM35.

Componenti elettronici:

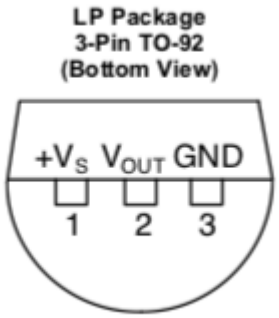
- Arduino UNO
- Breadboard
- LM35

Teoria: Il componente elettronico LM35 è un dispositivo integrato ad alta precisione utilizzato per misurare la temperatura ambientale.
Dato il basso costo e l'ampia scala di valori ammissibili (ovvero da -55°C fino a 150°C) questi dispositivi sono particolarmente diffusi. E' inoltre importante considerare che e nessun tipo di calibrazione esterna è richiesta.



Estratto Datasheet LM35

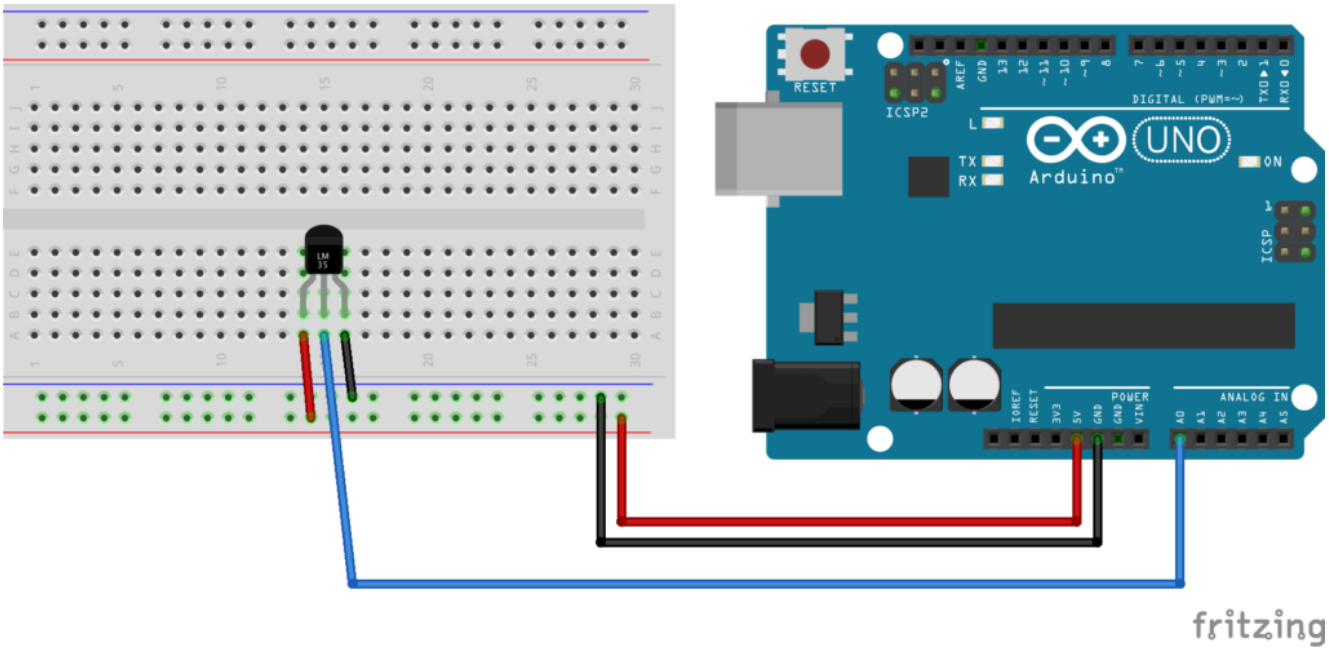
Questo dispositivo è caratterizzato da tre differenti pin ed un corpo semi-cilindrico. Guardando il lato piatto del dispositivo, il pin di sinistra è l'alimentazione (5V), il pin di destra la massa (GND), mentre sul pin centrale viene generata una tensione funzione della temperatura (10mV per ogni grado sopra lo zero). La temperatura può essere pertanto misurata attraverso una lettura analogica sul pin centrale effettuata mediante il controllore Arduino.



LM35 Package

Collegamento Circuitale:

Viene in seguito riportato lo schema elettrico utilizzato per valutare la temperatura mediante il dispositivo elettronico LM35



Collegamento Circuitale

Codice:

La temperatura viene determinata attraverso una lettura

analogica ed una opportuna scalatura.
Nel dettaglio, la tensione prodotta dal componente LM35 viene letta utilizzando il pin analogico A0 e mappata in un intervallo 0-1023. Considerando che per ogni grado percepito si ha un incremento di tensione di 10mV è opportuno riportare il valore letto attraverso la funzione analogRead in mV. Questa operazione può essere svolta dividendo il valore letto per 1023 e moltiplicando il risultato per 5000. I gradi sono infine ottenuti dividendo il risultato per 10.

Personalizzazioni:

E’ possibile modificare il circuito aggiungendo una ventola che si accende in modo automatico superata una determinata temperatura.

Controllare un LED mediante Smartphone

Obiettivo: Controllare un Led mediante Smartphone

Pre-requisiti:

App Inventor – Bluetooth HC06 – LED

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 modulo Bluetooth HC06
- 1 Smartphone Android
- 1 Resistenza da 220 Ohm
- 1 LED

Teoria: Partendo dall'applicativo Android, sviluppato nella lezione: "Appinventor – Bluetooth HC06 – LED" in questa lezione viene illustrato come utilizzare l'app creata per controllare un LED via smartphone attraverso il protocollo bluetooth.

Nel caso specifico il protocollo Bluetooth (abbreviato in BT) è uno standard di trasmissione dati per reti senza fili. Il BT fornisce un metodo standard, economico e sicuro per scambiare informazioni tra dispositivi diversi attraverso onde radio a corto raggio (qualche decina di metri). La comunicazione Bluetooth tra lo Smartphone ed Arduino avviene attraverso l'utilizzo di un modulo BT per Arduino denominato HC06.

Modulo BT HC06

Il Modulo HC06 implementa un convertitore da porta seriale UART a porta Bluetooth permettendo la comunicazione tra un microprocessore come una scheda Arduino e un dispositivo dotato di comunicazione Bluetooth (PC, Smartphone o Tablet). Il modulo è dotato di quattro differenti PIN descritti nella seguente tabella:

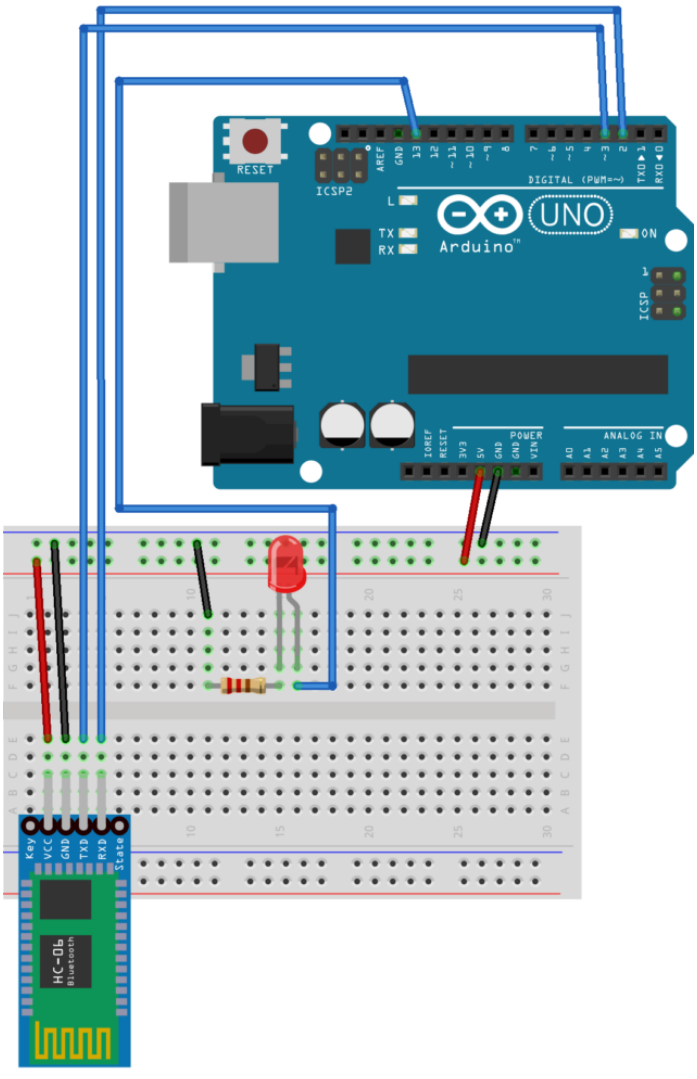
PIN	Descrizione
1	Vcc (Alimentazione, funziona correttamente anche con i 5 V di Arduino)
2	Gnd (Ground)
3	Txd (Pin di trasmissione seriale. Deve essere collegato ad un pin di ricezione)
4	Rxd (Pin di ricezione seriale. Deve essere collegato ad un pin di trasmissione)

Da un punto di vista pratico, grazie all'impiego della specifica **libreria** SoftwareSerial inclusa nel pacchetto software di Arduino è possibile comunicare con il modulo Bluetooth ed il relativo dispositivo connesso. La libreria mette infatti a disposizione una classe **SoftwareSerial**

all’interno della quale sono definite le principali funzioni necessarie per utilizzare il modulo HC06. Quali:

- **Begin**: Inizializza l’interfaccia seriale definendo la velocità standard di comunicazione
- **Available**: Indica se dei dati sono stati inviati al modulo HC06 (dati disponibili da leggere).
- **Read:** Legge i dati ricevuti
- **Write:** Scrive i dati

Collegamento Circuitale:



Schema Circuitale

Codice:

Personalizzazioni: E’ possibile aggiungere più led modificando l’applicazione ed il codice sorgente.

Arduino Pirati dei Caraibi

Obiettivo: Riprodurre la melodia del film “Pirati dei Caraibi” utilizzando la piattaforma Arduino.

Componenti elettronici:

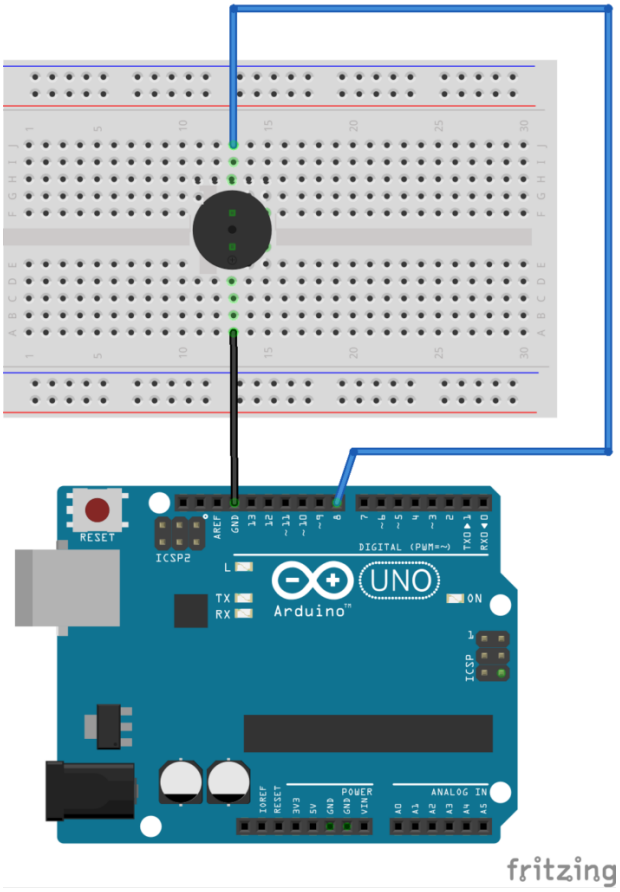
- Arduino UNO
- Breadboard
- 1 Buzzer

Pre-requisiti:

Buzzer Passivo

Teoria: Ogni melodia musicale è composta da note e pause. Se le pause possono essere riprodotte utilizzando la funzione **delay()** di Arduino (già vista negli esempi precedenti), le singole note possono essere facilmente generate grazie all'utilizzo di un buzzer passivo e della funzione **tone**(). Nel dettaglio, l'impiego della funzione tone permette di selezionare la frequenza riprodotta dal buzzer e la relativa durata della nota.
Nel caso in questione, l'intera melodia (comprensiva di note, durata delle note e pause, viene salvata in due differenti array (vettori) e riprodotta sequenzialmente come un vero spartito musicale.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

Arduino SuperMario

Obiettivo: Riprodurre la melodia del famoso videogioco utilizzando la piattaforma Arduino.

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Buzzer

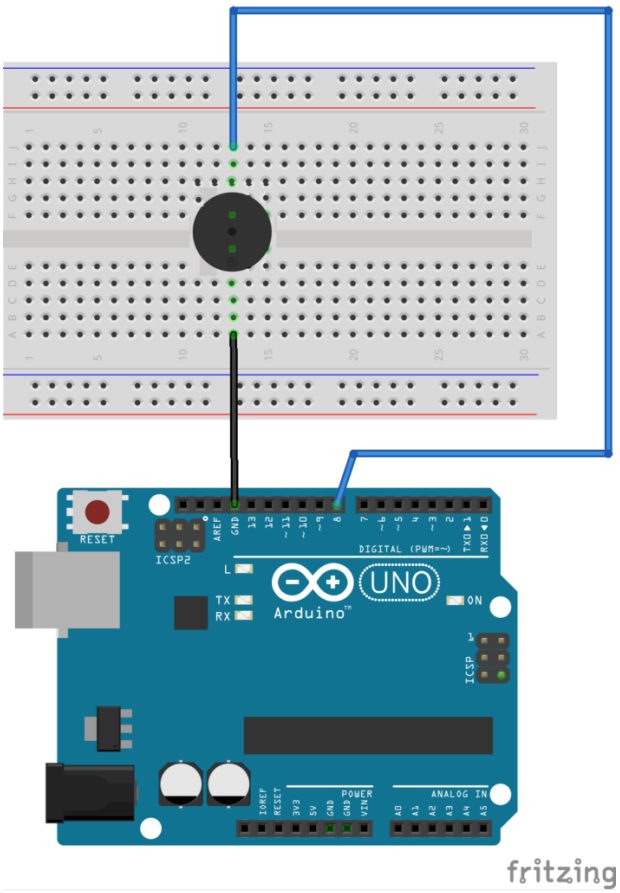
Pre-requisiti:

Buzzer Passivo

Teoria: Ogni melodia musicale è composta da note e pause. Se le pause possono essere riprodotte utilizzando la funzione **delay()** di Arduino (già vista negli esempi precedenti), le

singole note possono essere facilmente generate grazie all'utilizzo di un buzzer passivo e della funzione **tone**(). Nel dettaglio, l'impiego della funzione tone permette di selezionare la frequenza riprodotta dal buzzer e la relativa durata della nota.
Nel caso in questione, l'intera melodia (comprensiva di note, durata delle note e pause, viene salvata in due differenti array (vettori) e riprodotta sequenzialmente come un vero spartito musicale.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

Arduino Happy Birthday

Obiettivo: Realizzare un bigliettino di auguri di buon compleanno utilizzando la piattaforma Arduino.

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Buzzer

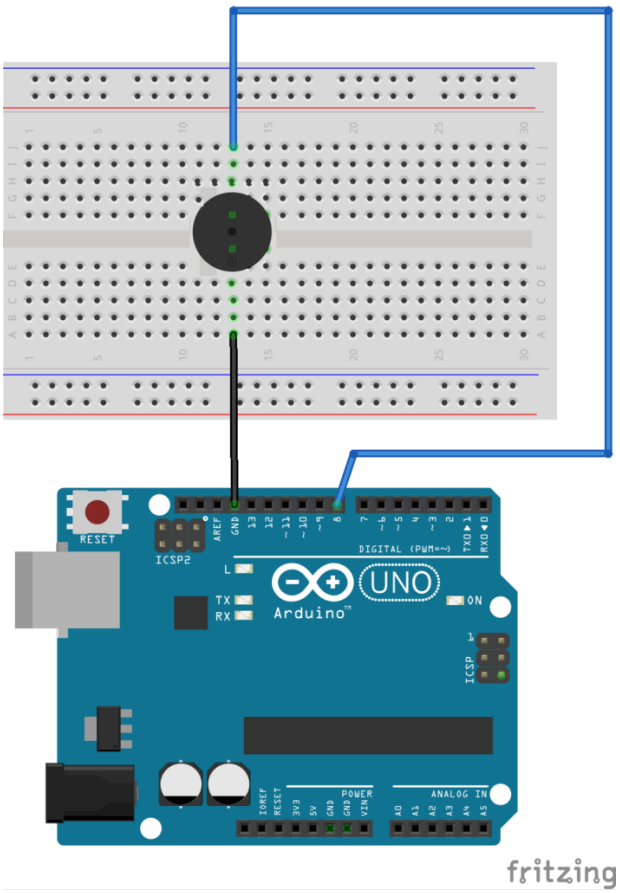
Pre-requisiti:

Buzzer Passivo

Teoria: Ogni melodia musicale è composta da note e pause. Se le pause possono essere riprodotte utilizzando la funzione **delay()** di Arduino (già vista negli esempi precedenti), le singole note possono essere facilmente generate grazie all'utilizzo di un buzzer passivo e della funzione **tone**(). Nel dettaglio, l'impiego della funzione tone permette di selezionare la frequenza riprodotta dal buzzer e la relativa durata della nota.

Nel caso in questione, l’intera melodia (comprensiva di note, durata delle note e pause, viene salvata in due differenti array (vettori) e riprodotta sequenzialmente come un vero spartito musicale.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

Il Sensore di Parcheggio

Obiettivo: Realizzare un sensore di parcheggio

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Sensore Ultrasuoni (e.g., HC-SR04)
- 1 Buzzer

Pre-requisiti:

Il Sensore a Ultrasuoni

Buzzer Attivo

Teoria: Il **sensore di parcheggio** rappresenta una delle moderne tecnologie che permettono al guidatore di un autoveicolo di essere a conoscenza della distanza tra la propria automobile ed un eventuale altro veicolo. Questo permette di gestire in modo facile e rapido l'operazione di parcheggio evitando danneggiamenti o urti agli autoveicoli in questione.
Nel caso specifico, il sensore di parcheggio si basa su un

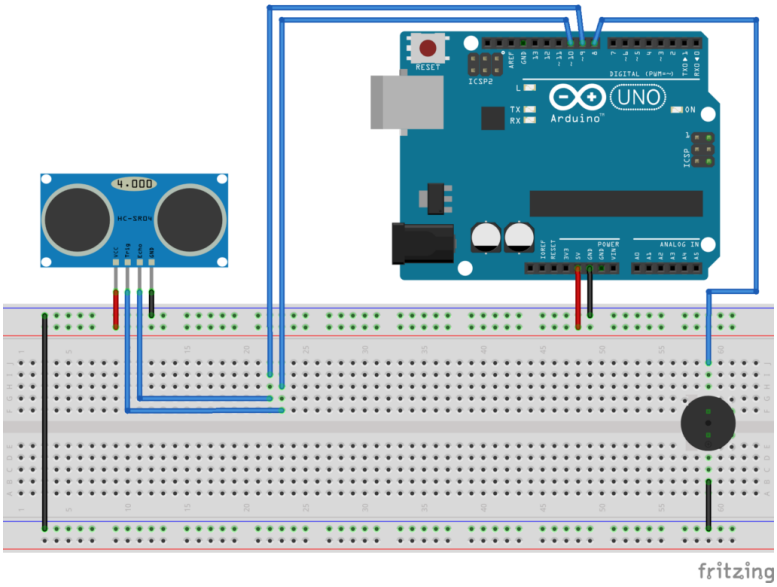
sensore di prossimità ad ultrasuoni (INPUT) ed un buzzer (OUTPUT) utilizzato come segnalatore acustico.
Nel dettaglio, maggiore è la distanza maggiore sarà il ritardo tra una segnalazione acustica e la successiva, analogamente, minore è la distanza minore sarà il ritardo tra una segnalazione acustica e la successiva. In particolare, la relazione ingresso uscita che lega la distanza al ritardo impiegato nella segnalazione acustica è funzione dei seguenti quattro parametri:

- DistanzaMassima: la distanza massima dopo la quale non viene più segnalato un'ostacolo.
- DistanzaMinima: la distanza minima per la quale il buzzer emette un tono continuo
- RitardoMassimo: Il ritardo tra una segnalazione acustica e la successiva nel caso di massima distanza.
- RitardoMinimo: Il ritardo tra una segnalazione acustica e la successiva nel caso di minima distanza.

Questi valori vengono utilizzati al fine di determinare l'equazione fondamentale per il calcolo del ritardo:

$$ritardo = distanza*((RitardoMax-RitardoMin)/(DistanzaMax-DistanzaMin))$$

Collegamento Circuitale:



Collegamento Circuitale

Codice:

Personalizzazioni: E’ possibile inserire un led al fine di avere un segnalatore visivo che indichi la distanza dall’ostacolo.

Il Motore passo-passo (Stepper)

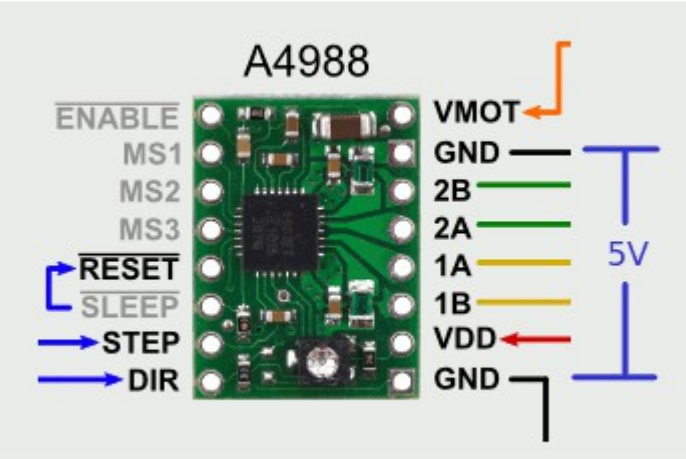
Obiettivo: Pilotare un motore passo-passo tramite Arduino

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Motore Stepper (e.g. Nema17 1.7A 1.8° 42MM Stepper Motor – 42shdc3025-24b)
- 1 Driver A4988
- 1 Alimentatore a 12 V e 2 A a corrente continua

Teoria: Il motore passo-passo, detto anche stepper, è un motore elettrico sincrono in corrente continua, senza spazzole, che permette la suddivisione della rotazione in piccoli angoli detti step. E' un motore molto preciso e veloce e facilmente controllabile tramite una scheda elettronica, denominata "driver". Esso viene collegato ad Arduino, e rende il controllo del motore molto facile, permettendo, con soli due uscite digitali, di controllare la velocità, la direzione e l'angolo di rotazione.

Il driver è il vero protagonista del controllo del motore passo-passo. Si pone tra Arduino e il motore e viene collegato come dal seguente schema:



I pin STEP e DIR servono per controllare la rotazione (e la velocità) e la direzione e vanno collegati a due uscite digitali di Arduino. Sulla destra invece l'A4988 presenta i pin per il collegamento al motore. GND (in basso) e VDD servono per alimentare la scheda tramite Arduino. Fondamentali sono i pin 1A, 1B, 2A e 2B che vanno collegati alle fasi del motore, come descritto più avanti. Infine, i pin GND e VMOT riguardano l'alimentazione del motore e vanno collegati all'alimentatore a 12 V.

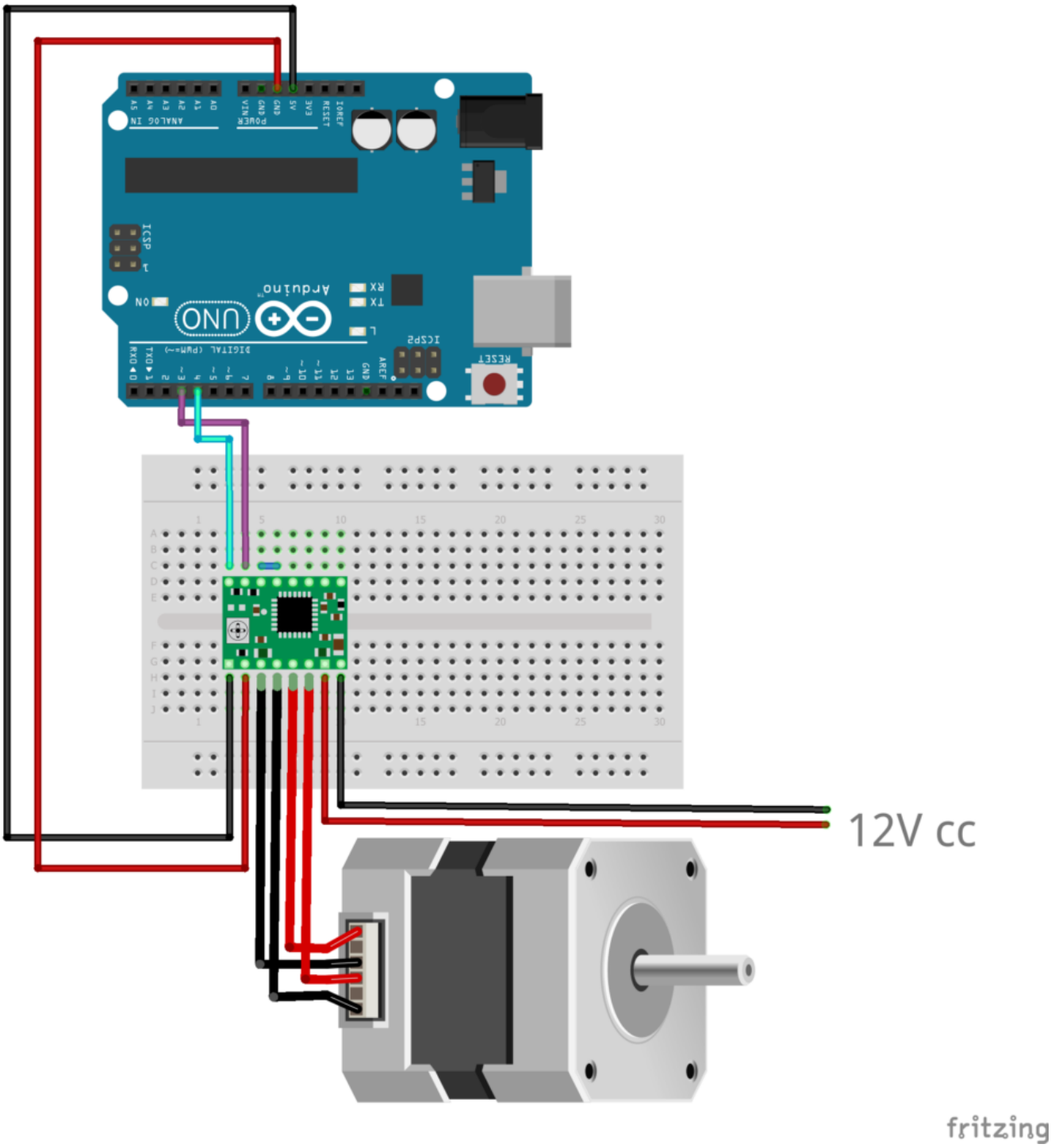
4988-DatasheetScarica

Il motore necessita di 200 impulsi per completare un giro. Il programma si basa su cicli che gestiscono la rotazione. La velocità di rotazione dipende dal tempo di attesa tra un impulso e un altro all'interno dei singoli cicli di rotazione.

La pausa tra un impulso e un altro (e quindi la gestione della velocità di rotazione) dipende dalla funzione

delayMicroseconds presente all’interno del ciclo di rotazione.

Collegamento Circuitale:



Codice:

Personalizzazioni: E’ possibile collegare un potenziometro e un pulsante per gestire la direzione e la velocità di rotazione.

Controllare un Servo Motore tramite Joystick

Obiettivo: Come controllare due servomotori utilizzando un Joystick per Arduino.

Pre-requisiti

Il Servomotore

Componenti elettronici:

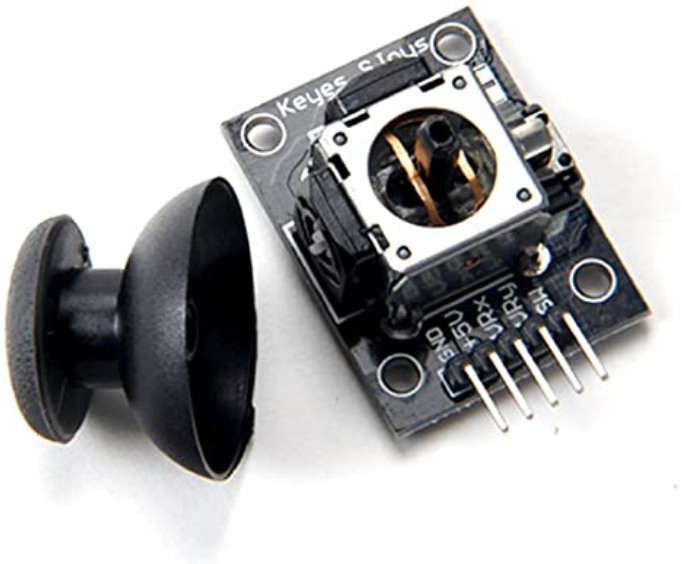
- Arduino UNO
- Breadboard

- 2 Servomotori
- 1 Joystick KY-023K

Teoria: Il **Joystick** è una dispositivo elettronico che trasforma i movimenti di una leva manovrata dall'utente in una serie di segnali elettrici, questi segnali possono essere in seguito utilizzati per controllare un software (e.g., videogame), un'apparecchiatura o un attuatore meccanico. Possono esistere due differenti tipologie di Joystick:

- Joystick Digitale: Rileva solamente la direzione dell'inclinazione della leva.
- Joystick Analogico: Rileva anche l'ampiezza dell'inclinazione.

Nello specifico in questo articolo viene illustrato il funzionamento di uno dei controller più tipicamente utilizzati e presenti nei vari kit Arduino: il "Dual Axis Joystick Module **KY-023**", Questo dispositivo, basato sul controller della PlayStation2, utilizza due potenziometri bi-assiali per controllare l'asse X e l'asse Y. Inoltre è possibile premere il controller per attivare uno switch. Nello specifico, la tensione di funzionamento del dispositivo è compresa nel range 3.3 – 5 V. Mentre le dimensioni sono pari a 2.6 x 3.4 cm.



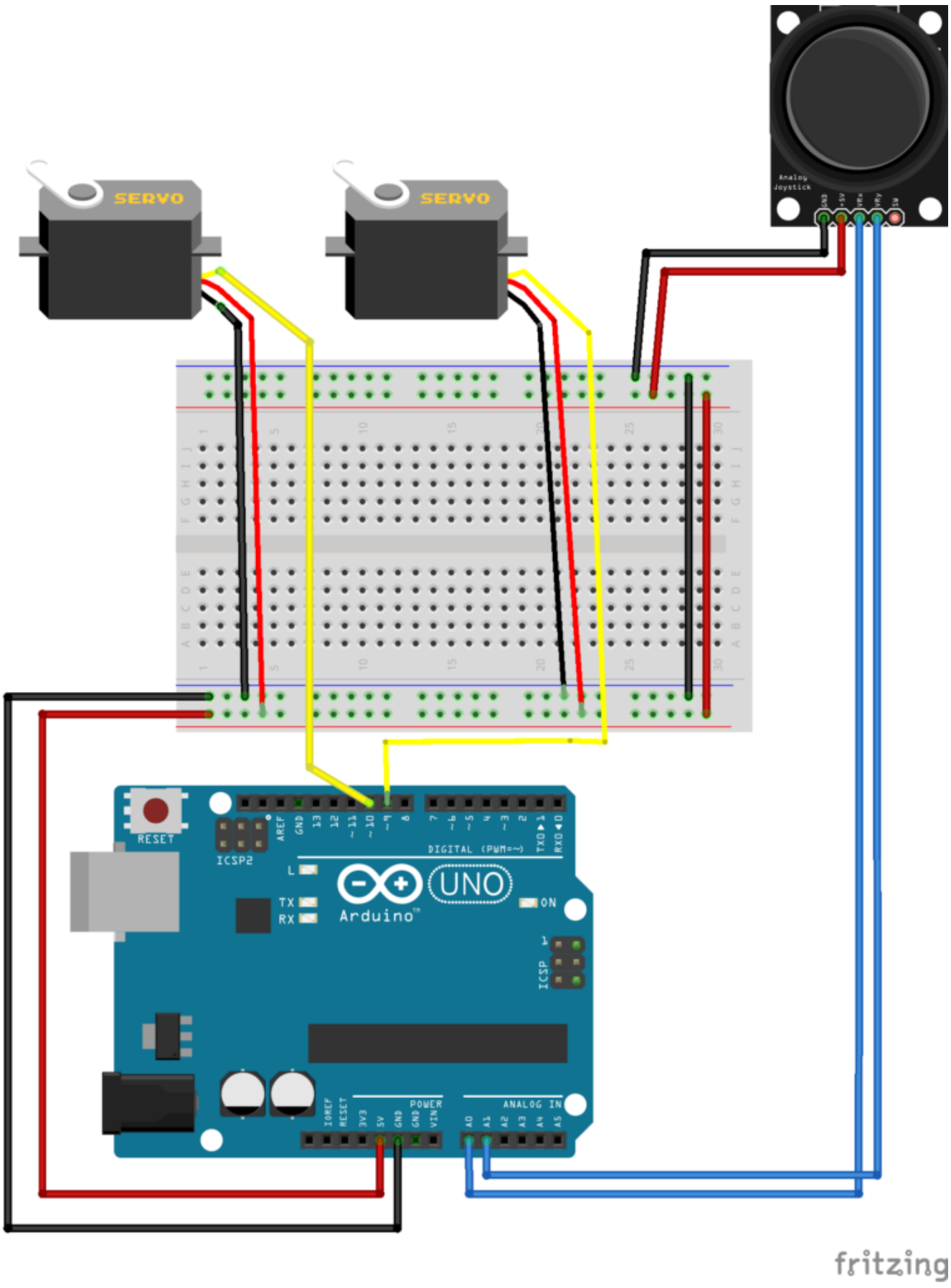
Dual Axis Joystick Module **KY-023**

Nella tabella a seguire è riportata la piedinatura utile per collegare in modo corretto il Joystick KY-023.

KY-023 PIN	**Descrizione**
GND	Ground
+5V	Alimentazione Vcc
VRx	Uscita Analogica (Asse X)
VRy	Uscita Analogica (Asse Y)
SW	Switch

In questo articolo il Joystick viene utilizzato per comandare due differenti servomotori. Un servo è associato all’asse X ed un altro è invece associato all’asse Y. La posizione di riposo dei due servomotori è per entrambi 90 gradi. Spostando il Joystick lungo l’asse X si può modificare la posizione del servo associato all’asse X di un angolo variabile da 0 a 180 gradi. Lo stesso accade modificando la posizione del joystick lungo l’asse y.

Collegamento Circuitale:



Codice:

Pilotare un Servo Motore tramite Potenziometro

Obiettivo: Ruotare un Servo Motore tramite un potenziometro

Pre-requisiti

Il Servomotore

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Servomotore
- 1 Potenziometro

Teoria: Il Servomotore è uno particolare tipo motore ampiamente utilizzato sia in contesti industriali sia nell'ambito del modellismo. Nel dettaglio, il servomotore è impiegato in tutte le applicazioni che prevedono il controllo della posizione di in motore in corrente continua ed il raggiungimento di un determinato angolo in modo preciso indipendentemente dalla posizione iniziale. Le caratteristiche principali del servomotore sono:

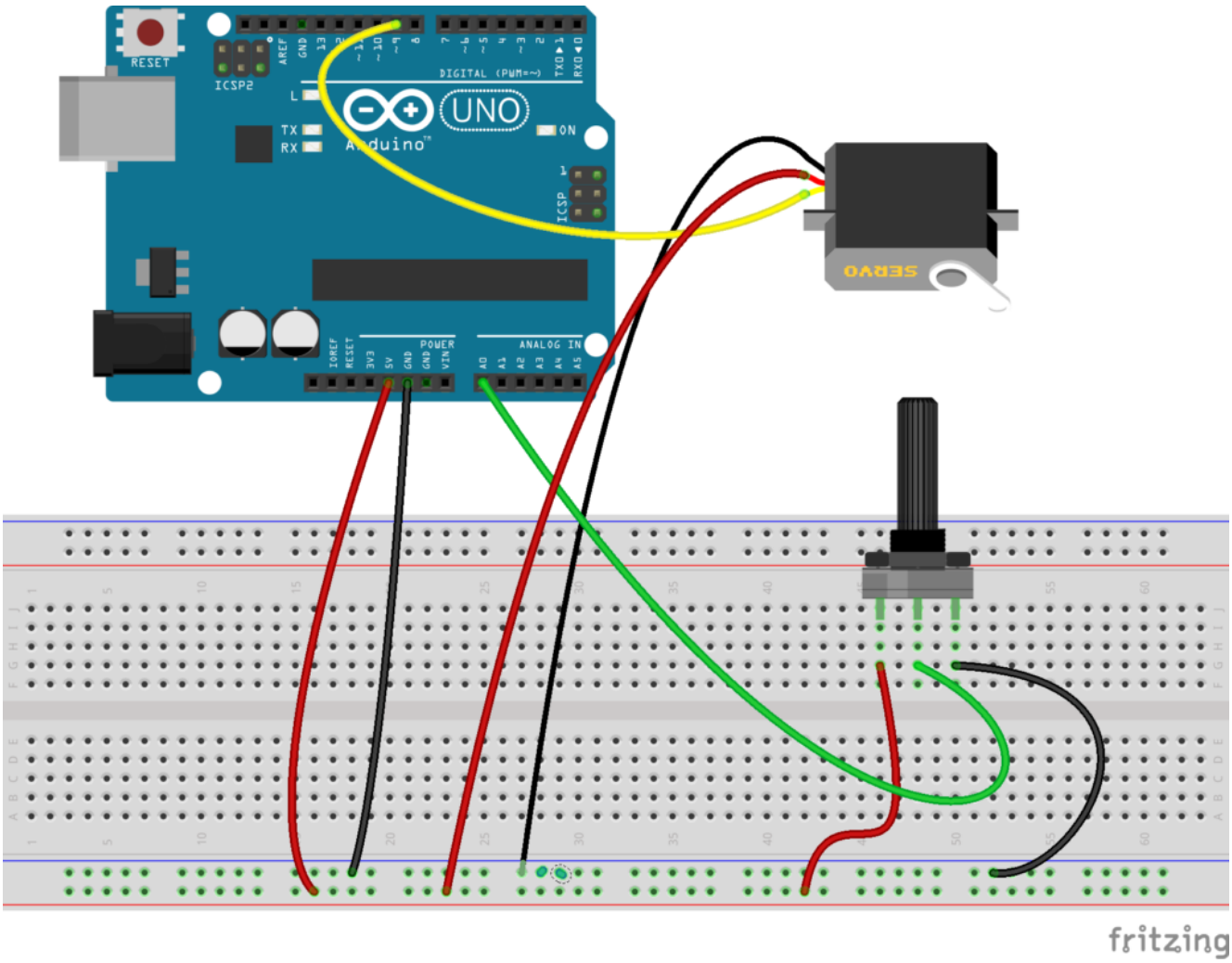
Come già accennato, i servo
motori sono dispositivi molto utilizzati in svariati ambiti perché permettono
la rotazione del proprio albero in base ad un angolo prestabilito.

I servomotori sono stati utilizzati per la prima volta nel mondo del modellismo RC, generalmente per controllare lo sterzo delle auto RC o i flap su un aereo RC o per aprire botole su un drone. Con il tempo, hanno trovato il loro uso anche nella robotica, nell'automazione e in svariati progetti Arduino.

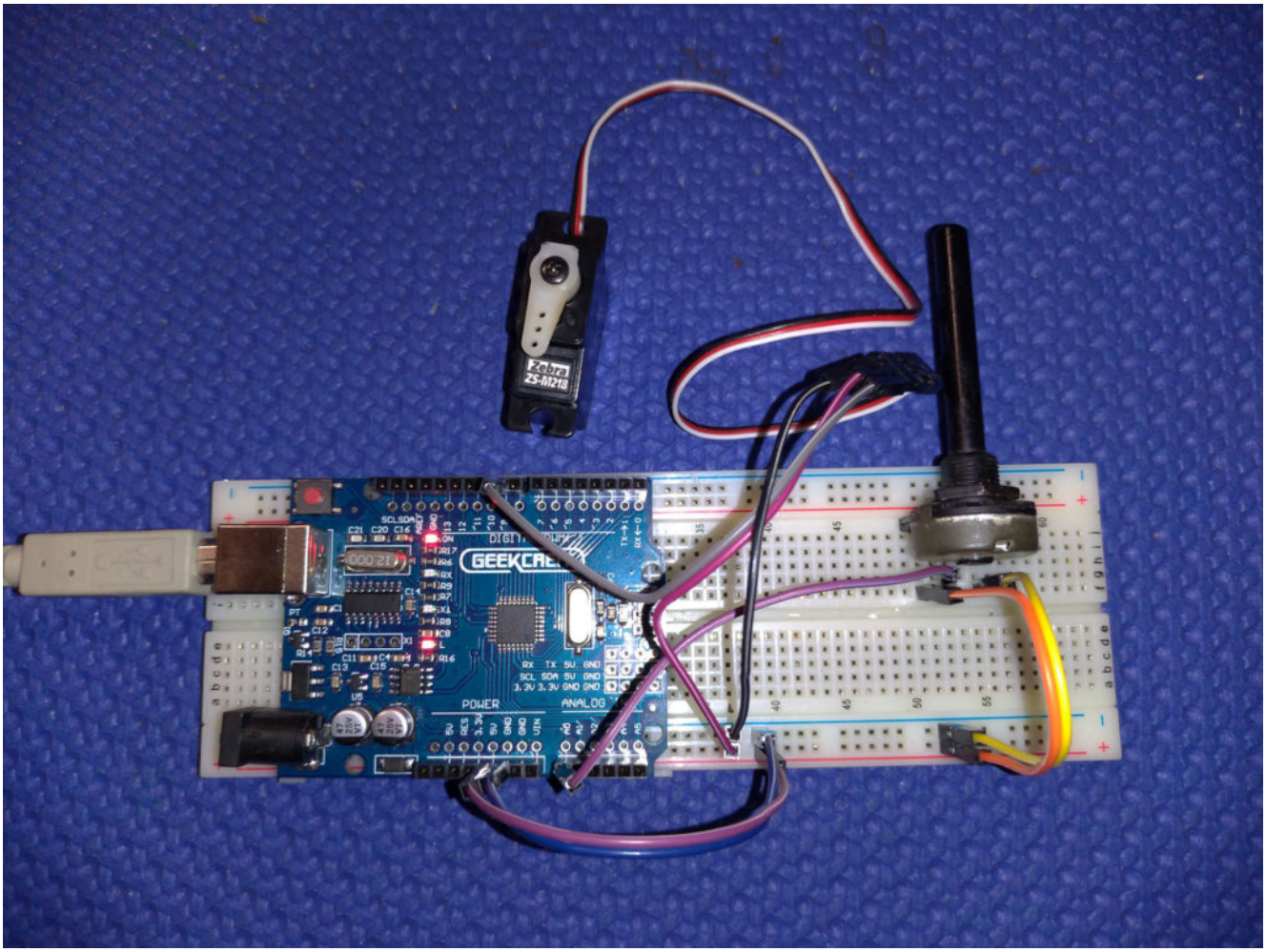
Solitamente l'albero può ruotare da 0 a 180 gradi e usando Arduino, possiamo dire a un servo di andare in una posizione specificata

In questo tutorial vedremo come interconnettere il servo motore ad Arduino e come farlo ruotare tramite un potenziometro con pochissime istruzioni.

Collegamento Circuitale:


RESET
ICSP2
DIGITAL (PWM=~)
UNO
ON
TX
RX
ICSP
SERVO
POWER
ANALOG IN
fritzing

Risultato:



Codice: