Blinking Led Senza Delay: MILLIS()

Obiettivo: Realizzazione del classico blinking led senza utilizzare la funzione Delay

Componenti elettronici:

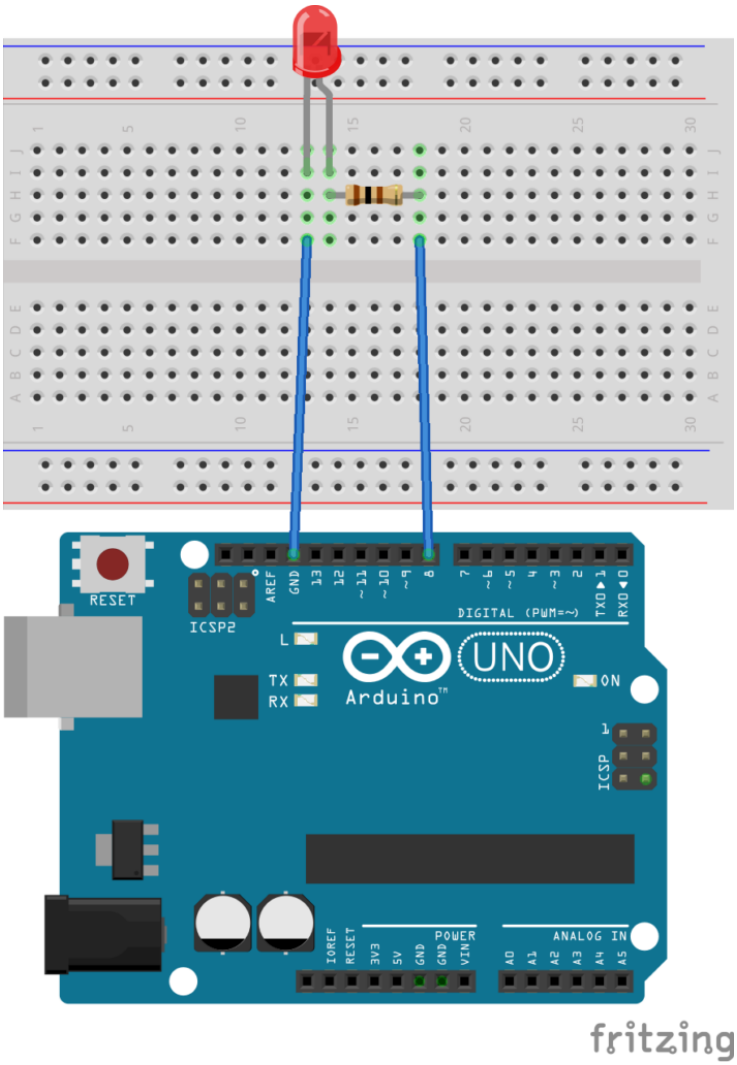
- Arduino UNO
- Breadboard
- Led
- Resistenza (100 Ohm)

Teoria: Se da un certo punto di vista l'impiego della funzione delay è particolarmente utile per la realizzazione di semplici applicativi; da un altro punto di vista molteplici potrebbero essere gli inconvenienti legati all'utilizzo di questa funzione.
E' infatti, molto importante, considerare che l'istruzione delay è un'istruzione bloccante. Questa istruzione "congela" Arduino nel suo stato corrente, pertanto negli istanti di delay non è possibile fare aggiornamenti, gestire input attraverso letture o comandare attuatori mediante scritture.
Ad esempio, nel caso della realizzazione di un semaforo per perdoni, dove è possibile effettuare la chiamata mediante la pressione di un pulsante, l'impiego della funzione delay è altamente sconsigliato per la gestione del semaforo perchè questo renderebbe invisibile la pressione del pulsate.
Per tutte queste ragioni è opportuno scrivere codici senza

l’impiego della funzione delay. Nel dettaglio è opportuno sostituire la funzione delay con la più funzionale ma meno pratica funzione **millis.**

La funzione millis restituisce il numero di millisecondi che sono passati da quando la board Arduino ha eseguito il programam corrente. Questo numero si riazzera dopo circa 50 giorni.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

A seguire viene riportato il codice utilizzato per fare lampeggiare il led senza delay. Nello specifico il codice memorizza l'ultimo istante in cui è avvenuta una azione (il led ha cambiato stato) nella variabile previousMillis mentre nella variabile currentMillis viene memorizzato il tempo corrente. Attraverso la differenza tra questi due tempi si comprendere se il led deve cambiare stato (accendersi o spegnersi). Nel dettaglio, se la differenza tra la variabile currentMillis e previousMillis è maggiore di 1000 millisecondi allora lo stato del led cambia.

Personalizzazioni: E' possibile modificare il comportamento del circuito in questione intervenendo sul valore della variabile *interval*. Modificando il suo valore infatti cambia la frequenza di lampeggiamento del LED.
E' inoltre possibile modificare il pin digitale utilizzato per pilotare il LED cambiando rispettivamente hardware e software.

L'utilizzo di una resistenza, in serie al LED, serve appunto per limitare la quantità di corrente presente sul diodo emettitore di luce.

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:

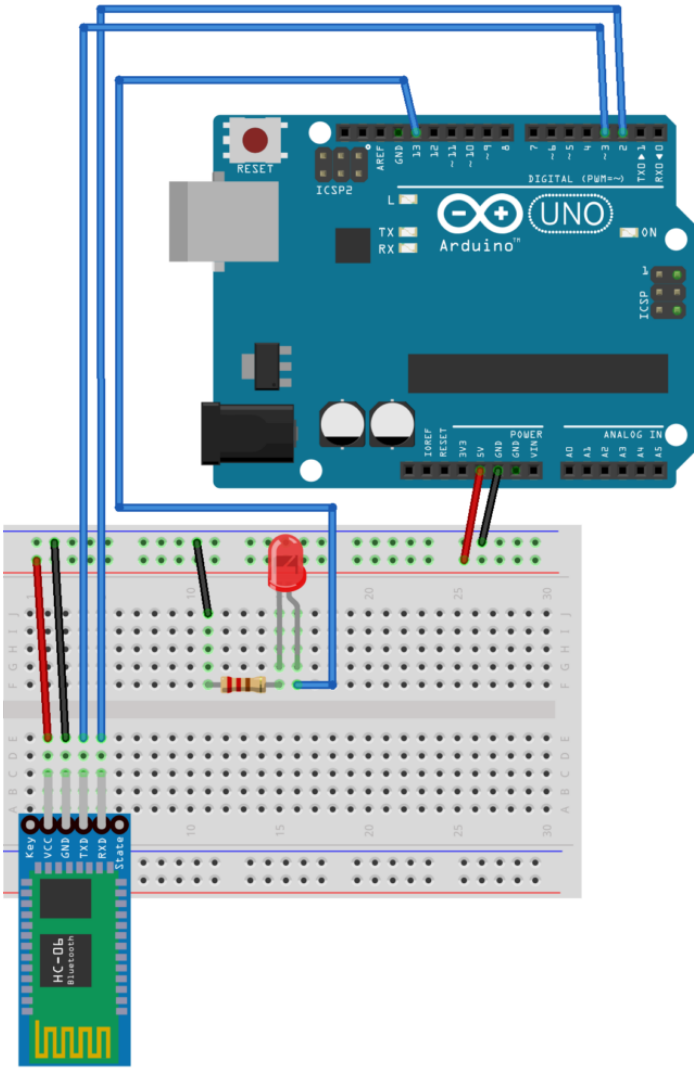
PIN	Descrizione
1	Vcc (Alimentazione, funziona correttamente anche con i 5 V di Arduino)
2	Gnd (Ground)
3	Txd (Pin di trasmissione seriale. Deve essere collegato ad un pin di ricezione)

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.

Il Motore passo-passo (Stepper)

Obiettivo: Pilotare un motore passo-passo tramite Arduino

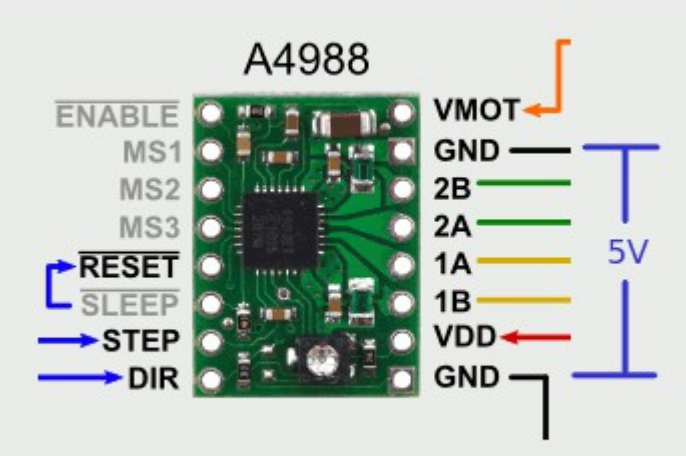
Componenti elettronici:

- 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:



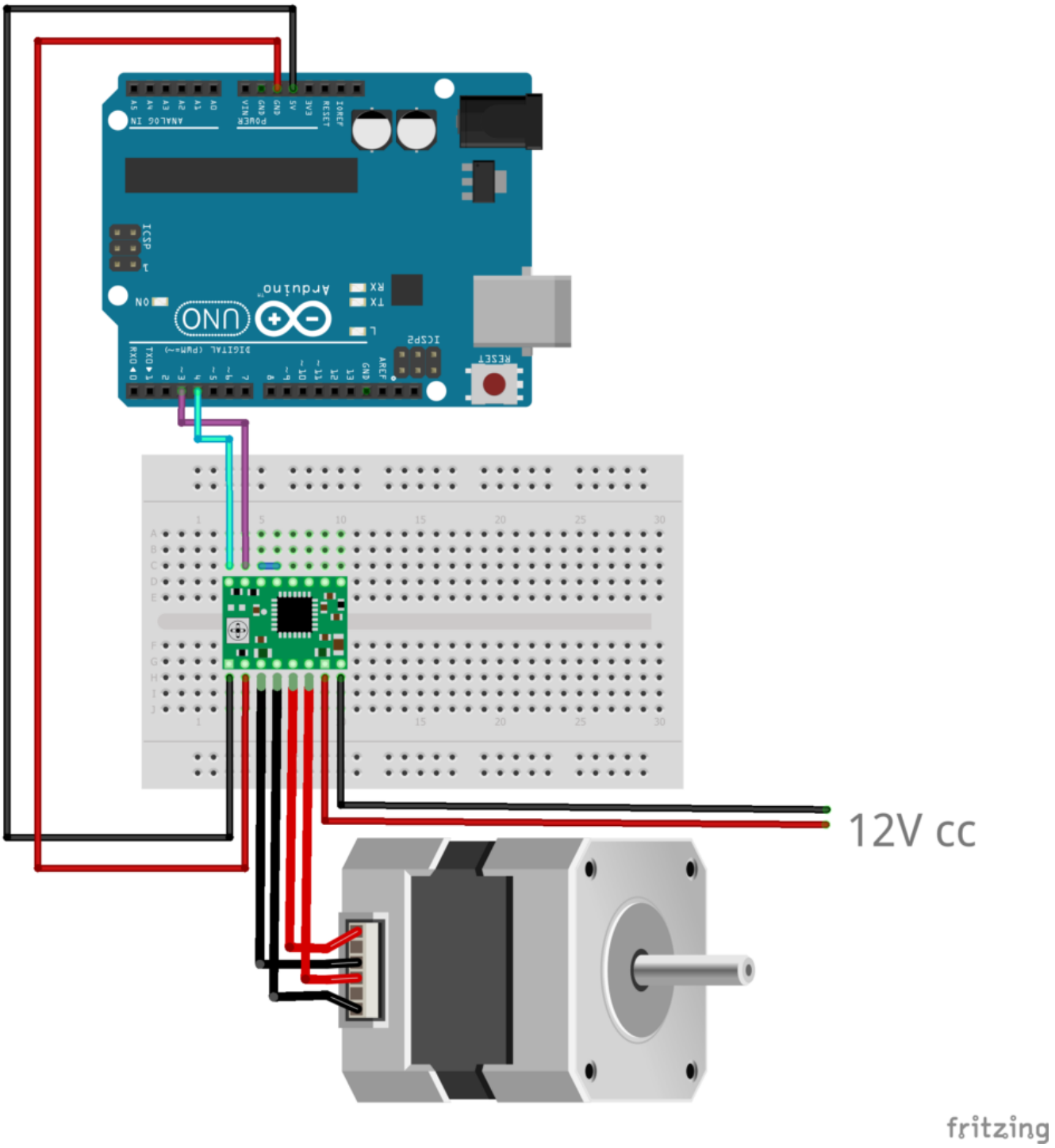
I pin STEP e DIR servono per controllare la rotazione (e la velocità) e la direzione e vanno collegati a due uscite digitali di Arduino. Sulla destra invece l'A4988 presenta i pin per il collegamento al motore. GND (in basso) e VDD servono per alimentare la scheda tramite Arduino. Fondamentali sono i 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.

Come Collegare un Display LCD ad Arduino

Obiettivo: Utilizzare un Display LCD 16×2 (basato su un Driver Hitachi HD44780).

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Display LCD (e.g., 1602A) compatibile con Driver Hitachi HD44780
- 1 Potenziometro da 10 kOhm
- 1 Resistenza da 220 Ohm

Teoria: I display basati su Driver Hitachi HD44780 sono tra i più diffusi in ambito embedded. Caratterizzati da differenti formati (i.e., 8×1, 8×2, 16×2, 20×2, 16×3, etc), questi display permettono di visualizzare solo caratteri. Nella seguente tabella si riporta la descrizione dei **PIN** del dispositivo:

PIN	Descrizione
1	Vss (Massa)
2	Vcc (Genericamente 5 V)
3	Vee (Controllo contrasto, collegato in genere ad un potenziometrocon tensione che varia da 0 a 5v)
4	R/S (0 per selezionare l'invio di un comando, 1 per i dati)
5	R/W (0 per selezionare la scrittura di dati o comandi, 1 per la lettura dei dati o dello stato)
6	E (inizia il ciclo di scrittura o lettura, secondo R/S e R/W)
7	D0 (Bus dati)
8	D1 (Bus dati)
9	D2 (Bus dati)
10	D3 (Bus dati)
11	D4 (Bus dati)
12	D5 (Bus dati)
13	D6 (Bus dati)
14	D7 (Bus dati)
15	A (Vcc retroilluminazione, se presente)
16	K (Vss retroilluminazione, se presente)

Il Driver HD44780 si basa su una modalità di trasferimento dati di tipo parallelo. Nel dettaglio è supportato sia il

trasferimento di 8 bit (l'intero comando D0-D7) sia il trasferimento di 4 bit (D4-D7). Nel secondo caso, per trasmettere un byte vengono effettuati due trasferimenti.

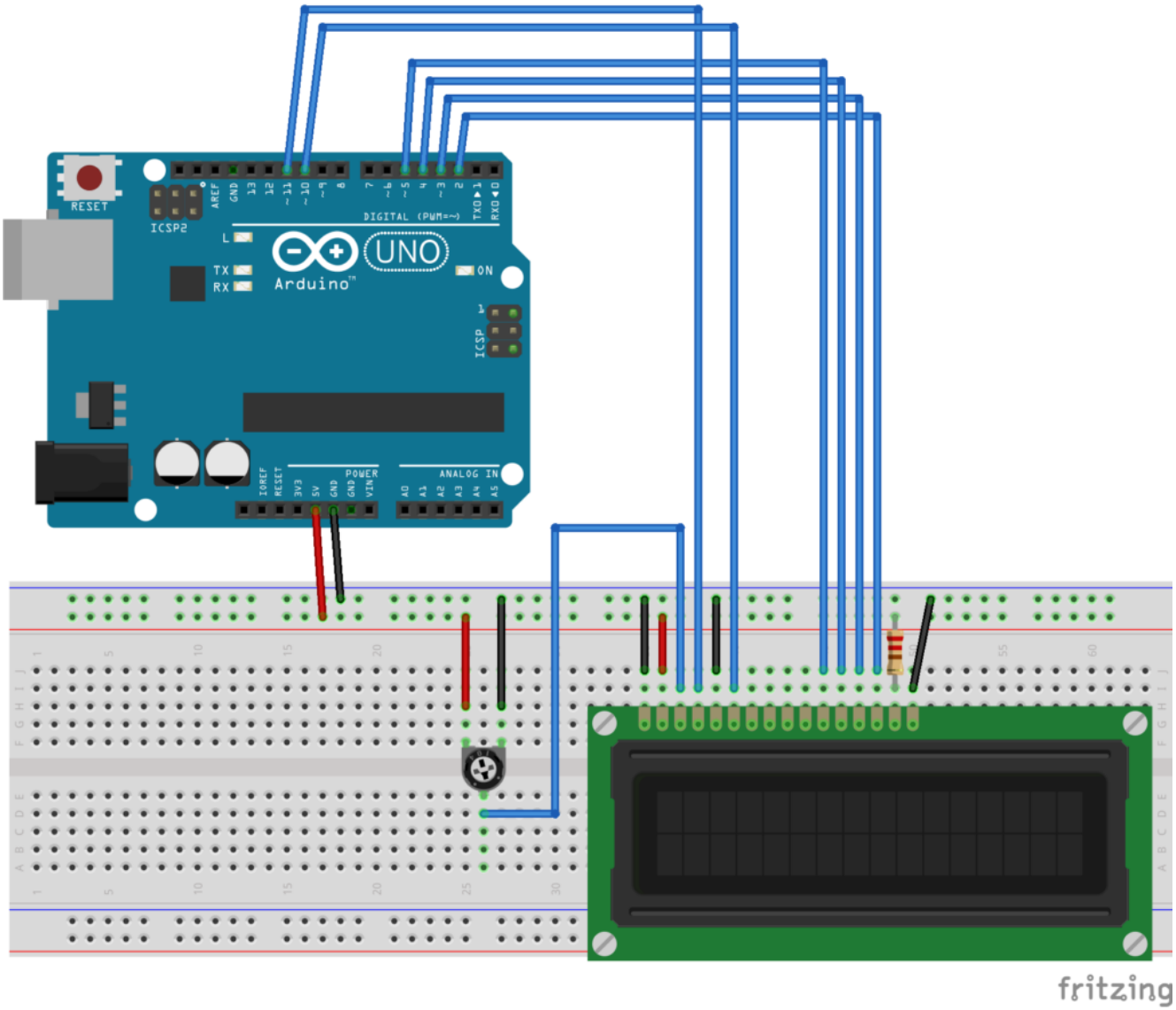
Per la scrittura di un singolo carattere è necessario rispettare il seguente "**protocollo di trasmissione**":

- Posizionare il cursore nella posizione desiderata
- Impostare a 1 il pin R/S e a 0 il pin R/W
- Inviare il codice ASCII del carattere
- Impostare ad 1 il pin E per un minimo di 450 nanosecondi ed, in seguito, riportarlo a 0

Da un punto di vista pratico, grazie all'impiego della specifica **libreria** **LiquidCrystal** inclusa nel pacchetto software di Arduino è possibile pilotare il display semplificando notevolmente la parte di gestione dei pin, dei comandi e delle relative temporizzazioni. La libreria mette infatti a disposizione una classe **LiquidCrystal** all'interno della quale sono definite le principali funzioni necessarie per utilizzare i display basati su Driver HD44780. Quali:

- **Begin**: Inizializza l'interfaccia del display LCD specificandone le dimensioni (larghezza ed altezza)
- **SetCursor**: Posizione il cursore LCD ovvero la posizione nella quale verrà visualizzato il testo scritto
- **Print:** Scrive il testo sul display LCD.

Collegamento Circuitale:



Codice:

Personalizzazioni: E’ possibile modificare il contrasto del display intervenendo sul potenziometro.

Collegare più Sensori Ultrasuoni ad Arduino

Obiettivo: Utilizzare simultaneamente due sensori a ultrasuoni (HC-SR04). Nel dettaglio, al fine di dimostrare il corretto funzionamento del circuito, ad ogni sensore è associato un led. Quando la distanza misurata dal sensore è inferiore ad una data soglia, il led associato si accende, mentre quando la distanza è superiore il led si spegne.

Componenti elettronici:

- Arduino UNO
- Breadboard
- 2 Sensori Ultrasuoni (e.g., HC-SR04)
- 2 Led (utilizzati per dimostrare il funzionamento del circuito)
- 2 Resistenze (100 Ohm) (indispensabili per il corretto funzionamento dei led)

Pre-Requisiti:

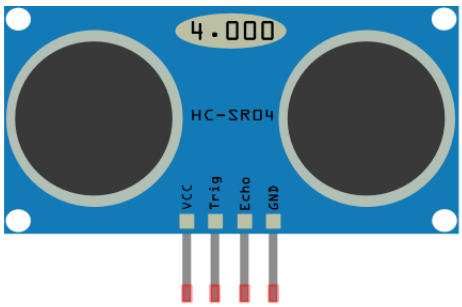
Il Sensore a Ultrasuoni

Teoria: Il sensore di prossimità è un dispositivo che permette di rilevare la presenza di oggetti nelle immediate vicinanze, senza che vi sia un effettivo contatto.
Nel caso specifico, il sensore di prossimità ad ultrasuoni sfrutta il principio del Sonar. Degli impulsi sonori (ultrasonici) vengono emessi dal dispositivo il quale attraverso l'eventuale eco di ritorno permette di rilevare la presenza di un oggetto all'interno della portata nominale. Esempi pratici di sensori ad ultrasuoni sono i sensori di retromarcia e di parcheggio utilizzati nelle moderne automobili.
Nel dettaglio, le principali caratteristiche tecniche del sensore ad ultrasuoni HC-SR04 sono:

- Alimentazione: +5V DC
- Angolo di misura: < 30°
- Distanza di rilevamento: da 2cm a 400cm
- Risoluzione: 1cm
- Frequenza: 40kHz

Il sensore, dispone di 4 pin: Vcc (+5V), Trigger, Echo, GND.

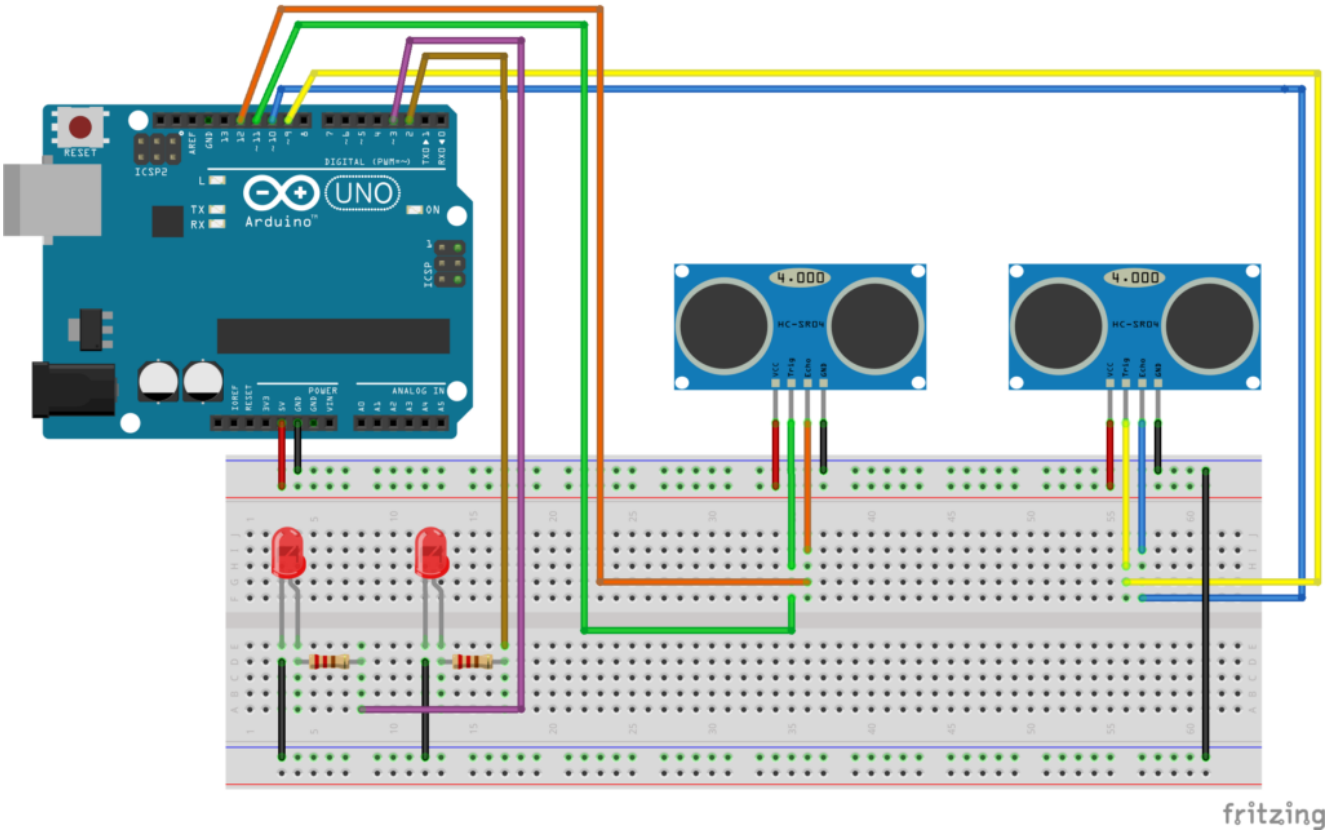
- VCC: +5V DC
- Trigger: Genera l'impulso ultrasonico
- Echo: Rileva il segnale ultrasonico di ritorno
- GND: 0V Ground



Modello Sensore HC-SR04

Collegamento Circuitale:

La realizzazione hardware di circuiti che utilizzano più sensori ad ultrasuoni è relativamente facile. Ognuno dei pin di Echo e di Trig del sensore deve essere semplicemente collegato ai pin di input/output di Arduino.



Collegamento Circuitale

Codice:

Nel caso specifico, in cui si vogliano utilizzare più sensori ad ultrasuoni per misurare simultaneamente la distanza in più punti è opportuno gestire i sensori in modo corretto attraverso il relativo codice. Nel dettaglio, è importante considerare che inizialmente entrambi i sensori generano l'impulso utilizzato per determinare la distanza. In seguito le letture di echo (effettuate mediante l'istruzione pulseIn) sono eseguite in modo sequenziale (una dopo l'altra).

Il Sensore a Ultrasuoni

Obiettivo: Utilizzare un Sensore a Ultrasuoni (HC-SR04) per misurare la distanza.

Componenti elettronici:

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

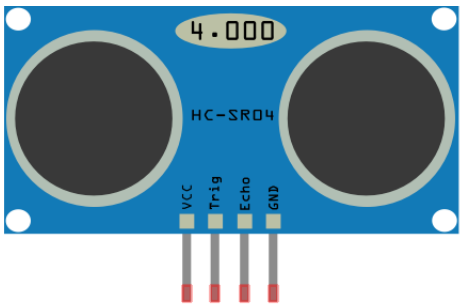
Teoria: Il sensore di prossimità è un dispositivo che permette di rilevare la presenza di oggetti nelle immediate vicinanze, senza che vi sia un effettivo contatto.

Nel caso specifico, il sensore di prossimità ad ultrasuoni sfrutta il principe del Sonar. Degli impulsi sonori (ultrasonici) vengono emessi dal dispositivo il quale attraverso l'eventuale eco di ritorno permette di rilevare la presenza di un oggetto all'interno della portata nominale. Esempi pratici di sensori ad ultrasuoni sono i sensori di retromarcia e di parcheggio utilizzati nelle moderne automobili.
Nel dettaglio, le principali caratteristiche tecniche del sensore ad ultrasuoni HC-SR04 sono:

- Alimentazione: +5V DC
- Angolo di misura: < 30°
- Distanza di rilevamento: da 2cm a 400cm
- Risoluzione: 1cm
- Frequenza: 40kHz

Il sensore, dispone di 4 pin: Vcc (+5V), Trigger, Echo, GND.

- VCC: +5V DC
- Trigger: Genera l'impulso ultrasonico
- Echo: Rileva il segnale ultrasonico di ritorno
- GND: 0V Ground



Modello Sensore HC-SR04

Pertanto considerando la formula che lega velocità, spazio e tempo:

$$s = v*t$$

e la velocità del suono pari a 343 m/s (che espressa in microsecondi diventa 0,0343 m/uS), si ottiene:

$$s = 0,0343*t$$

Considerando inoltre che il suono percorrerà due volte la distanza da misurare (dal sensore all'oggetto e dall'oggetto al sensore); il tempo t ottenuto deve essere diviso per due, ottenendo:

$$s = 0,0343*(t/2)$$

$$s = 0,01715*t$$

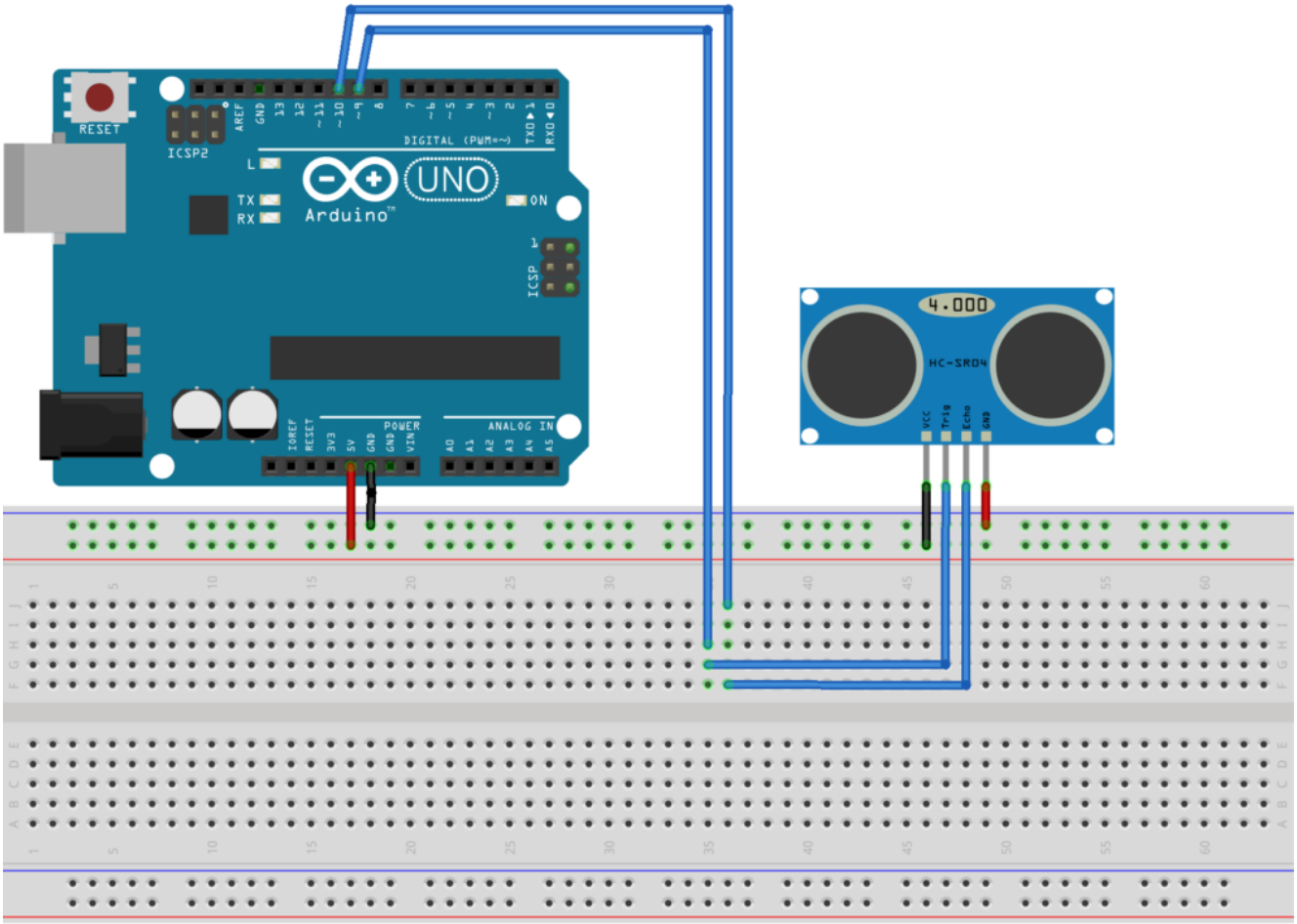
$$s = t/58,31$$

Per valutare la presenza di un oggetto è necessario rispettare il seguente "**protocollo**":

- Il PIN Trigger deve essere dettato alto (HIGH value) per almeno 10microsecondi.
- In automatico il modulo HC-SR04 invierà 8 impulsi ultrasonici ad una frequenza pari a 40kHz.

- Il PIN Echo viene posto in ascolto. Calcolando il tempo di arrivo/ritorno dell’impulso ultrasonico.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

[crayon-663881fa9dce0279801458/]

Personalizzazioni: E’ possibile modificare il contrasto del display intervenendo sul potenziometro.