Realizzazione di un Gioco a Quiz – La macchina a stati finiti

Obiettivo: Realizzare un gioco a Quiz mediante pulsanti e display LCD 16×2 (basato su un Driver Hitachi HD44780). L'attività prevede la realizzazione software di una macchina a stati.

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Display LCD (e.g., 1602A) compatibile con Driver Hitachi HD44780
- 1 Resistenza da 220 Ohm (per il display)
- 1 Trimmer da 10 kOhm (per il display)
- 2 Pulsanti
- 2 Resistenze da 1kOhm (per i pulsanti)

Pre-requisiti:

Come Collegare un Display LCD ad Arduino

LED e Pulsante

Teoria: L'utilizzo di una progettazione basata sul concetto di macchina a stati finiti permette la facile realizzazione di apparati elettronici e sistemi capaci di fornire output desiderati a partire da specifici input ricevuti. Questo paradigma si basa sulla definizione degli stati di funzionamento del dispositivo da realizzare e sul codice associato allo stato stesso. A seguire sono riportati alcuni esempi pratici di stati di sistemi generici:

- Semaforo: verde, giallo, rosso
- Ascensore: primo piano, secondo piano, etc
- Cancello: aperto, chiuso

Da un punto di vista pratico si potrebbe affermare che: "**Attraverso una macchina a stati è possibile avere tanti loop quanti sono gli stati del dispositivo in questione** ". Tuttavia è importante considerare che solamente uno di questi loop è attivo. (La macchina non può torvarsi in più stati contemporaneamente).
Nel caso specifico di un gioco a quiz gli stati sono rappresentati dalle:

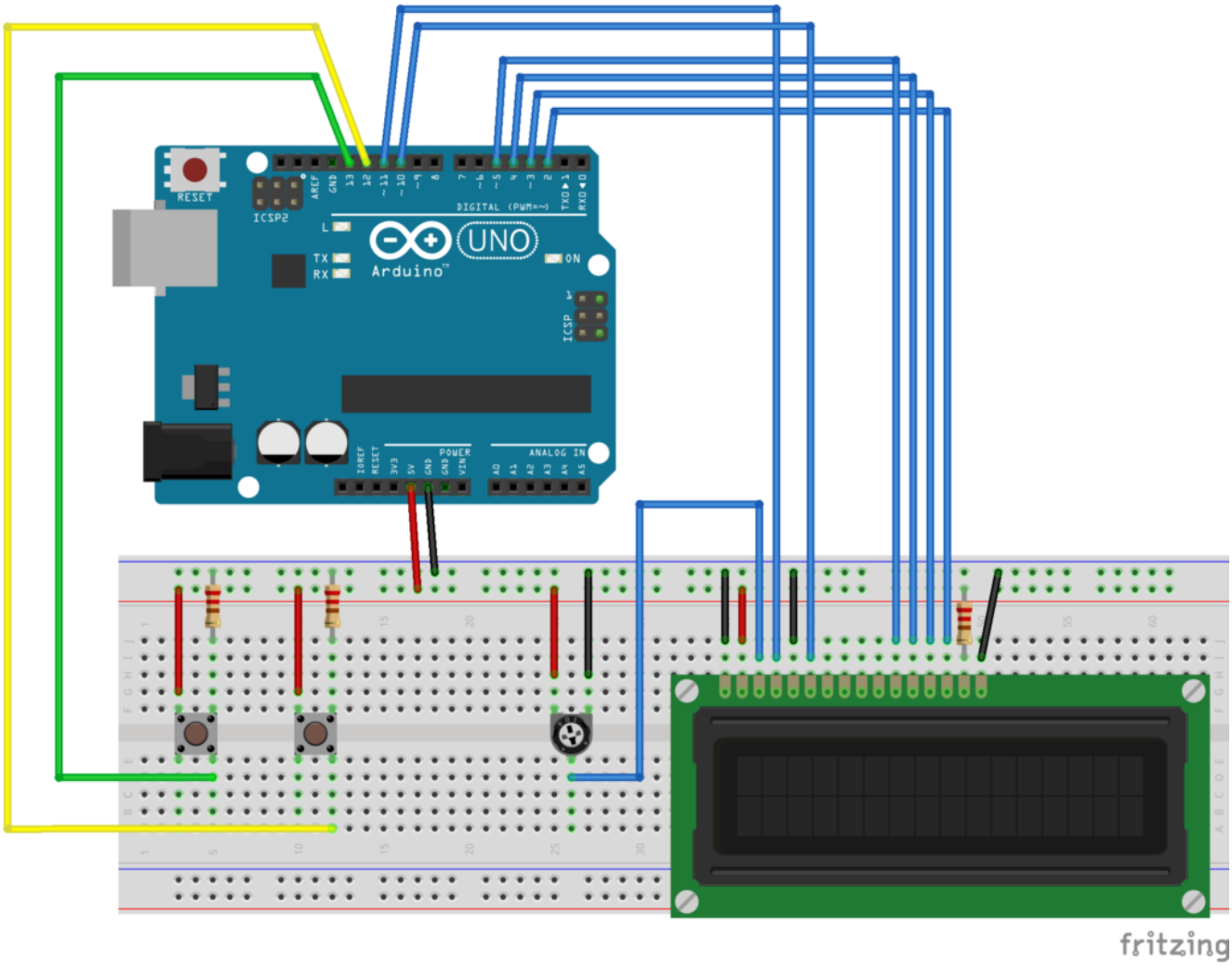
- Domande: la domanda deve essere visualizzata sul display
- Risposte: l'utente deve premere il pulsante e il controllore deve verificare se la risposta è corretta.

Pertanto si avranno tanti stati quante sono il numero di domande moltiplicato per due.

E’ infine importante considerare che, nel caso della realizzazione di un quiz a risposta multipla, l’impiego di una macchina a stati permette di risolvere il problema annoso del delay. Questa funzione può infatti essere tranquillamente utilizzata negli stati associati alle domande (l’utente deve avere tempo di leggere la domanda) mentre non deve essere assolutamente impiegata negli stati relativi alle risposte.

La realizzazione di una macchina a stati mediante il controllore Arduino è relativamente semplice. Infatti, il concetto di stato viene implementato grazie all’utilizzo di una variabile globale (tipicamente denominata state) e attraverso una serie di if o mediante uno switch case si seleziona il codice da eseguire in quello specifico stato.

Collegamento Circuitale:



<u>Codice</u>: Attraverso la variabile *"state"* è possibile specificare il "loop" che Arduino dovrà eseguire. Nel dettaglio, valori pari della variabile "*state*" sono utilizzati per riprodurre la domanda sul display, mentre valori dispari sono impiegati per gestire la risposta. E 'importante notare che non sono presenti delay nel loop di gestione della risposta (ad eccezione di quello utilizzato per visualizzare se la risposta data è corretta oppure errata).

<u>Personalizzazioni</u>: E' possibile aggiungere un numero maggiore di domande ed un numero maggiore di pulsanti per gestire più

risposte.

Controllo del Contrasto di un Display LCD mediante PWM

Obiettivo: Controllare il contrasto di un Display LCD 16×2 (basato su un Driver Hitachi HD44780) mediante PWM. (Se non possiedi un Trimmer puoi utilizzare questa strategia basata su PWM e filtro passa-basso).

Componenti elettronici:

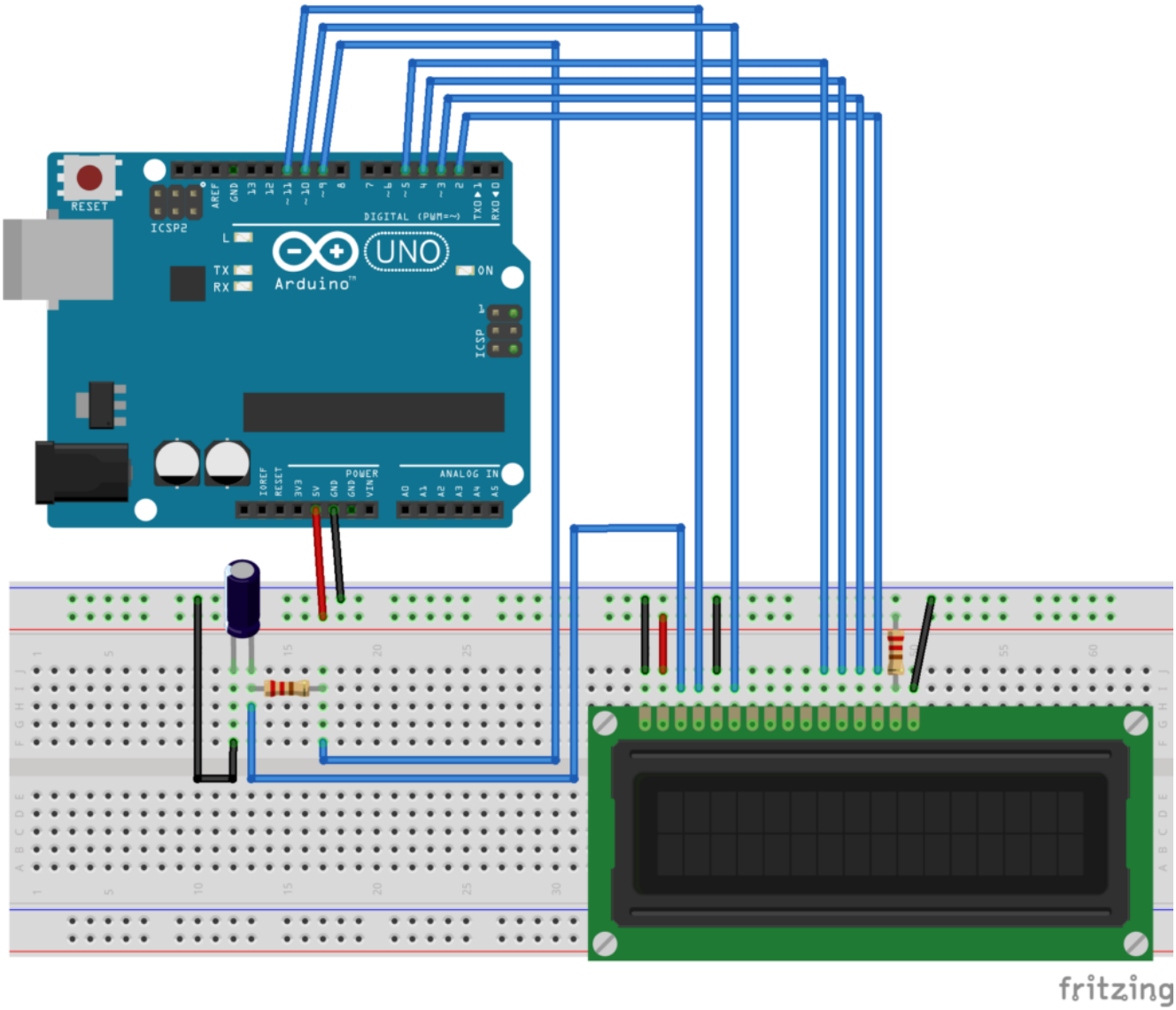
- Arduino UNO
- Breadboard
- 1 Display LCD (e.g., 1602A) compatibile con Driver Hitachi HD44780
- 1 Resistenza da 220 Ohm
- 1 Resistenza da 1kOhm (per filtro passa basso)
- 1 Condensatore elettrolitico da 22uF (per filtro passa basso)

Pre-requisiti:

Come Collegare un Display LCD ad Arduino

Teoria: In un display LCD basato su Driver HD44780 il pin numero 3 è utilizzato per gestire il contrasto. Questo pin viene tipicamente collegato ad un potenziometro con tensione variabile nel range 0 a 5v. Pertanto variando la posizione del Trimmer cambia il livello di contrasto del display. Il contrasto è un parametro di fondamentale importanza nella gestione di una dispositivo elettronico che utilizza un display LDC. Un'errata regolazione del livello di contrasto può rendere un'immagine troppo o poco dettagliata con il rischio che le aree più chiare e/o quelle più scure possano scomparire rendendo il testo non leggibile. Pertanto, nel caso in cui non si possedesse un Trimmer, l'utilizzo di un display LCD potrebbe essere facilmente compromesso a causa della incapacità di settare un livello di contrasto corretto.
Tuttavia, è importante considerare che, esistono delle alternative all'utilizzo di un trimmer per generare una tensione variabile compresa tra 0 e 5 Volts. Nel dettaglio, in questo articolo viene presentata una tecnica basata sull'utilizzo della PWM e di un filtro passa basso. L'impiego della tecnica PWM permette di generare un segnale con un duty cycle regolabile. Questa tecnica viene utilizzata anche nell'istruzione analogWrite per creare dei segnali apparentemente analogici partendo da segnali digitali. Il segnale PWM viene in seguito filtrato utilizzando un filtro RC passa basso del primo ordine. Attraverso questa operazione è infatti possibile ottenere la componente continua del segnale PWM necessaria per regolare il contrasto.

Collegamento Circuitale:



Collegamento Circuitale

<u>Codice</u>: Attraverso la variabile *contrast* (regolabile nel range 0-255) è possibile modificare il valore del contrasto da software fino ad ottenere l'effetto desiderato.

<u>Personalizzazioni</u>: E' possibile introdurre due pulsanti per modificare il contrasto del display utilizzando il valore PWM (0-255). Un pulsante può incrementare il valore della variabile *contrast* mentre l'altro pulsante può decrementare il

suo valore.

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.

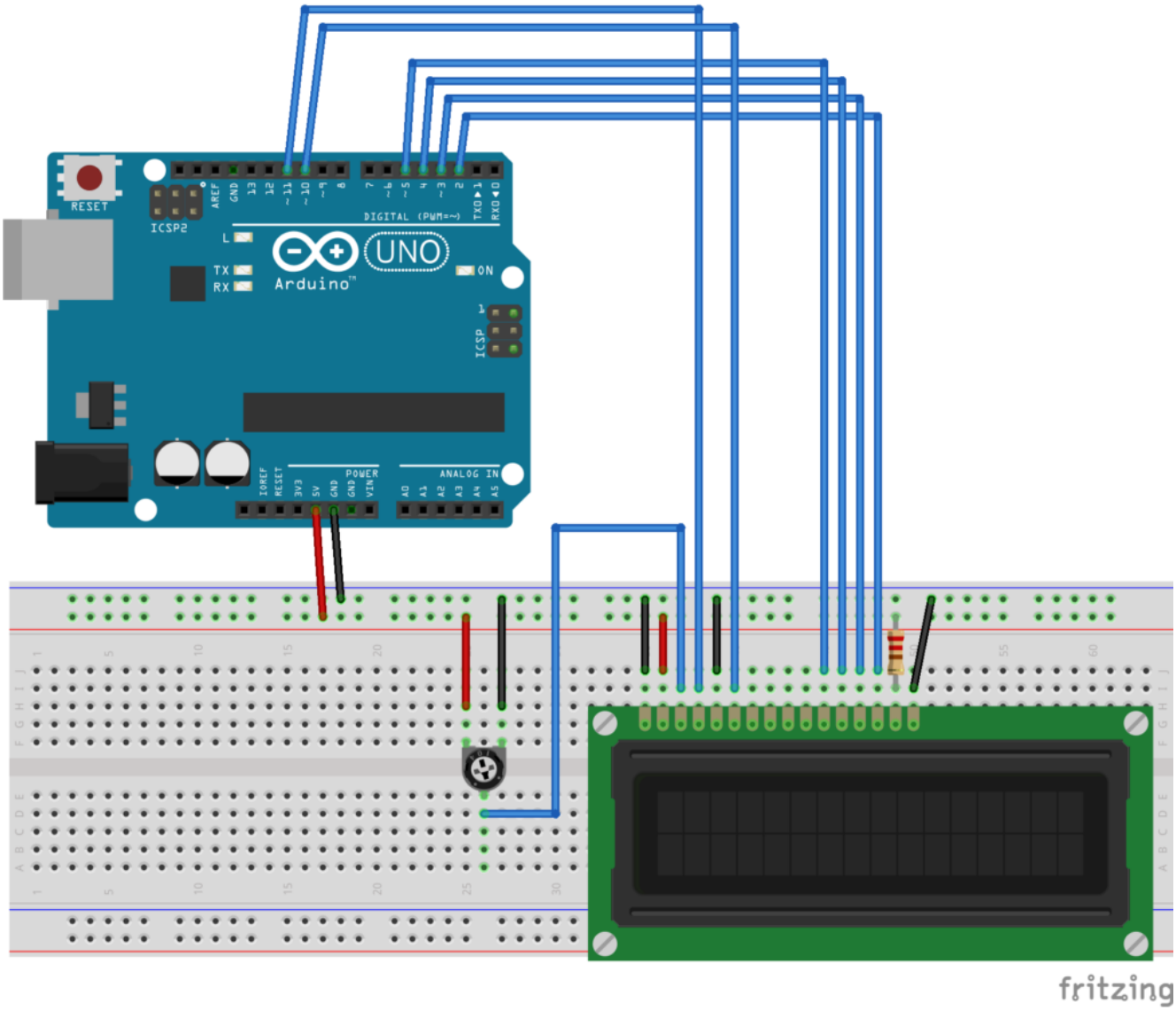
Per la scrittura di un singolo carattere è necessario rispettare il seguente “**protocollo di trasmissione**“:

- 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.