Realizzare un Voltmetro [0-5Volt] con Display a 7 Segmenti

Obiettivo: Realizzare un dispositivo per misurare tensioni nel range 0.5 V utilizzando il microcontrollore Arduino ed il display a sette segmenti.
Un progetto realizzato dagli alunni della classe 3AUT dell'Istituto Tecnico Industriale "Enrico Mattei" di Urbino:
– Amadori Federico
– Fucili Elia

Componenti elettronici:

- Arduino UNO
- Breadboard
- 14 Resistenze da 100 Ohm per I display a sette segmenti
- 2 Display a 7 Segmenti

Pre-requisiti:

1..2..3.. Il Display a 7 Segmenti

Le Funzioni digitalWrite, digitalRead, analogWrite e

analogRead

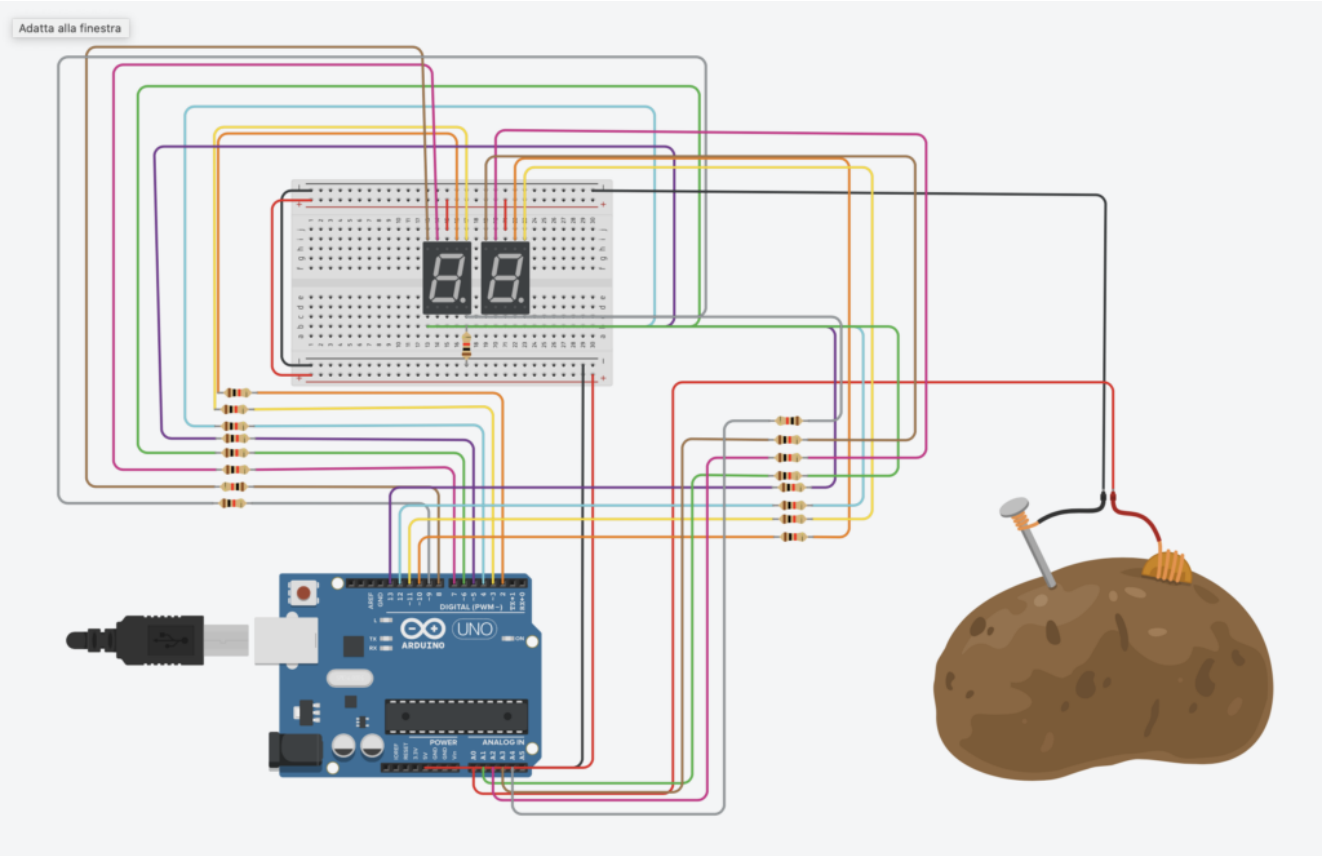
Teoria: Effettuare la misura di una tensione compresa nel range 0-5 V, utilizzando il controllore Arduino, è relativamente semplice. Per effettuare questa misura, basta collegare l'elemento di cui vogliamo analizzare la tensione ad un ingresso analogico (A0-A5) e utilizzare l'istruzione analogRead nell'apposito sketch.
A tale fine è importante ribadire che l'ingresso analogico compreso nel range 0-5V viene mappato utilizzato la funzione analogRead(A0) nel range 0-1023, pertanto è indispensabile utilizzare una corrispettiva funzione di conversione basta sulla seguente proporzione:

ValoreAnalogRead : 1023 = ValoreTensione : 5

In conclusione, il valore di tensione può essere semplicemente ottenuto dividendo per 1023 e moltiplicando per 5 il valore letto utilizzando la funzione analogRead(A0).

Tale valore può essere poi visualizzato su differenti tipi di display: monitor del computer, display LCD e display a 7 segmenti. Nell'esempio in questione vengono utilizzati due display a 7 segmenti (uno per la parte intera ed uno per la parte decimale) al fine di visualizzare la tensione misurata.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

Attraverso l’utilizzo di due display a sette segmenti è possibile visualizzare la tensione prodotta dalla “patata”. Nel caso specifico viene utilizzata una patata come generatore di tensione per verificare che il dispositivo fornisca una valutazione corretta della tensione misurata. Delle funzioni specifiche (e.g., zeroI, unoI, zeroD, unoD, etc) vengono utilizzate per visualizzare i vari numeri nei due differenti display.

Tinkercad:

Personalizzazioni: E’ possibile utilizzare altre tipologie di display per visualizzare la tensione misurata.

Utilizzare e Creare una Libreria per il Display a 7 Segmenti

Obiettivo: Utilizzare e creare una libreria (file header e cpp) per un Display a Sette Segmenti.

Puoi scaricare i file di libreria cliccando nel seguente link: http://www.arduinofacile.it/wp-content/uploads/2021/03/SevenSegment.zip
I file scaricati devono essere inseriti all’interno della cartella di progetto insieme al file .ino

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Display a 7 Segmenti
- 8 Resistenze

Pre-requisiti:

Creare funzioni con Arduino … per un Display a 7 Segmenti

Teoria: la realizzazione di funzioni di libreria permette di facilitare l'operazione di **riutilizzo del codice** rendendo più veloce e più rapido lo sviluppo. Nel caso specifico la funzione di libreria implementata sarà costituta da un file header (.h) e da un file sorgente (.cpp).

Un file header è un file di testo che contiene i prototipi dei metodi (funzioni) definite nel relativo file sorgente. Nel caso in questione il file header contiene anche la dichiarazione della classe "SevenSegment" utilizzata per modellare il display a sette segmenti.
Tale classe sarà caratterizzata da 10 attributi:

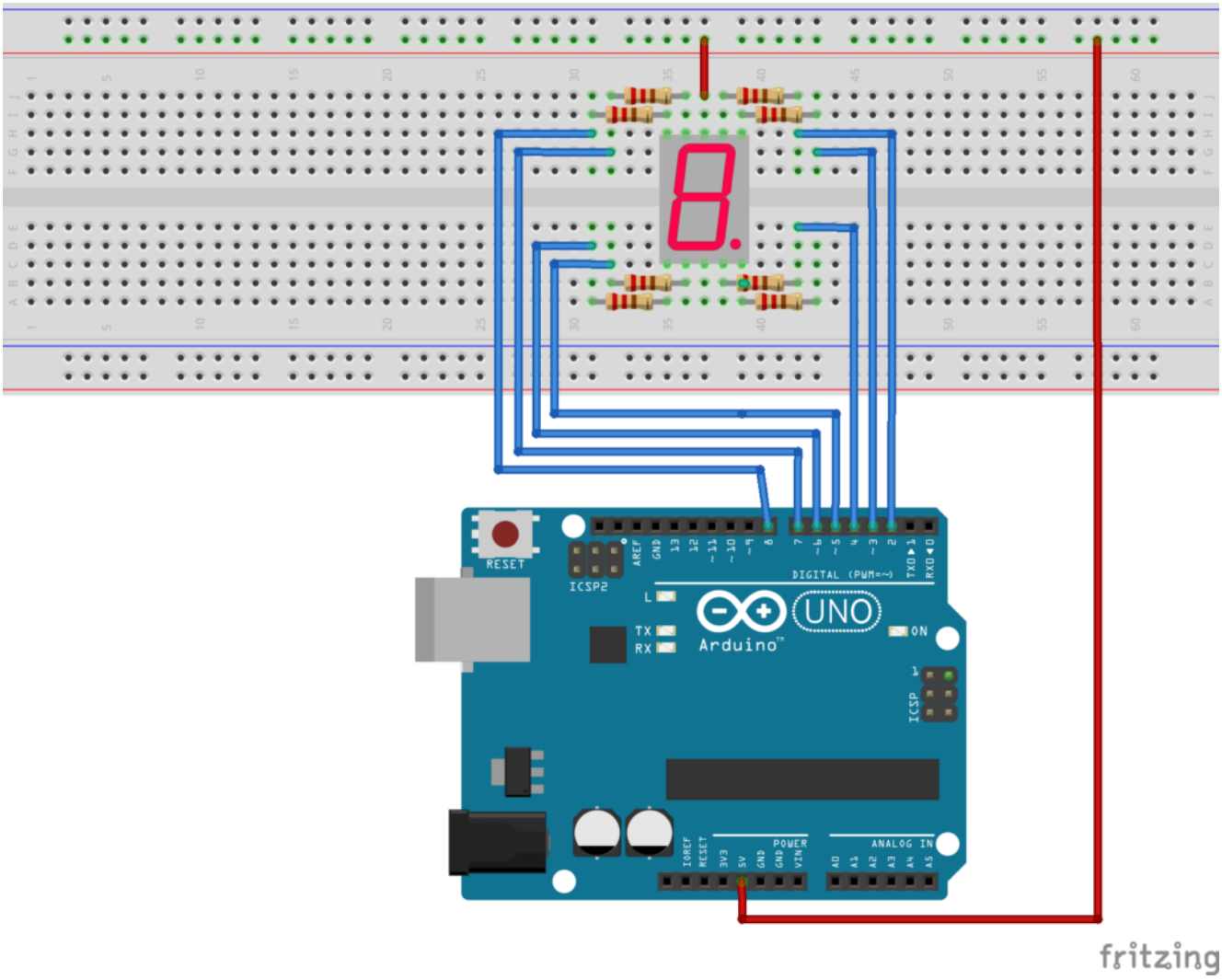
- **int pinA**: il pin A del display a sette segmenti
- **int pinB**: il pin B del display a sette segmenti
- **int pinC**: il pin C del display a sette segmenti
- **int pinD**: il pin D del display a sette segmenti
- **int pinE**: il pin E del display a sette segmenti
- **int pinF**: il pin F del display a sette segmenti
- **int pinG**: il pin G del display a sette segmenti
- **int pinDP**: il pin DP del display a sette segmenti
- **bool isCommonAnode**: indica se il display è di tipo anodo comune oppure no

e da 11 metodi

- **void Print0()**: metodo utilizzato per stampare il numero 0
- **void Print1()**: metodo utilizzato per stampare il numero 1
- **void Print2()**: metodo utilizzato per stampare il numero 2
- **void Print3()**: metodo utilizzato per stampare il numero 3
- **void Print4()**: metodo utilizzato per stampare il numero 4
- **void Print5()**: metodo utilizzato per stampare il numero 5
- **void Print6()**: metodo utilizzato per stampare il numero 6
- **void Print7()**: metodo utilizzato per stampare il numero 7
- **void Print8()**: metodo utilizzato per stampare il numero 8
- **void Print9()**: metodo utilizzato per stampare il numero 9
- **void CountDown()**: metodo utilizzato per eseguire il countdown.

Nel file sorgente viene invece riportata l'implementazione dei prototipi delle funzioni dichiarate nel file header.

Collegamento Circuitale:



Collegamento Circuitale

<u>Codice</u>:

Vengono in seguito riportate le tre porzioni di codice utilizzate per creare la funzione di libreria relativa al display a sette segmenti.

- File Header: contiene la definizione della classe con i propri attributi (i.e., pinA, pinB, etc) ed i prototipi dei relativi metodi.

- File Sorgente: contiene le implementazioni dei metodi riportati nel file header.

- File Arduino: Utilizzato per fornire un esempio di come utilizzare la libreria per la gestione del display a sette segmenti.

Se tutti i file sono correttamente posizionati sullo stesso livello all’interno della cartella di progetto, due nuove tab compariranno nell’ambiente di sviluppo utilizzato per programmare Arduino. Attraverso queste tab sarà possibile visionare e modificare il file sorgente (.cpp) ed il file header (.h)

IDE con l’utilizzo della libreria SevenSegment.h