Matteo Ghilardini – 12.06.2018



**Cassaforte**

**Progetto di fine modulo 121  
Il docente: Gianpaolo Sheuer**

# Sommario

[Sommario 1](#_Toc515352307)

[Introduzione al progetto 2](#_Toc515352308)

[Hardware e Datasheet componenti 3](#_Toc515352309)

[Arduino Mega 2560 3](#_Toc515352310)

[Matrice di Pulsanti 3](#_Toc515352311)

[Display 7 segmenti 4](#_Toc515352312)

[Led RGB 4](#_Toc515352313)

[Foto del montaggio 5](#_Toc515352314)

[Software 8](#_Toc515352315)

[Conclusione e Risultati 9](#_Toc515352316)

[Fonti 10](#_Toc515352317)

# Introduzione al progetto

Come lavoro di fine modulo 121, è stato assegnato ad ognuno di noi un diverso progetto ideato da noi, nel mio caso ho deciso di sviluppare il tutto sulla base della matrice di pulsanti 4x4, sul led RGB e sul display 7 segmenti singolo. Ciò mi ha portato a pensare di simulare la parte di sblocco di una cassaforte.

Infatti, la matrice di pulsanti corrisponde ad un tastierino numerico con valori esadecimali (0-9 & A-F), il display fa ciò che corrisponde alla realtà e il led simula il meccanismo di sgancio (apertura).

Oltre ciò ho deciso di integrare uno switch a 2 posizioni dove in una viene immesso il codice di sicurezza, mentre nella seconda, oltre ad accendere di blu il led (per distinguere facilmente le 2 posizioni), permette all’utente di immettere il suo codice; se i due corrispondono, verrà acceso il led verde, altrimenti quello rosso.

Quando il codice dell’utente è stato inserito, è possibile tramite un pulsante resettarlo (e quindi “ritentare”).

Ho aggiunto anche un potenziometro che permette di definire la dimensione del codice da un minimo di 1 carattere, ad un massimo di 9. Il valore è dunque stato mappato dal valore originale (0-1023) al valore personalizzato (1-9) e il valore ottenuto viene poi stampato su un display 7 segmenti (verde). Una volta modificato uno dei due codici, non sarà più possibile modificare il valore in questione. Per rendere chiara la conferma del numero, viene acceso il “DOT” del display.

In sintesi, i componenti utilizzati sono:

* 1 Arduino Mega 2560
* 1 bread-board
* 1 display 7 segmenti rosso
* 1 display 7 segmenti verde
* 1 matrice di pulsanti 4x4
* 1 led RGB anodo comune
* 1 potenziometro
* 1 switch
* 1 pulsante
* 3 resistenze da 330 Ω
* 3 resistenze de 10 kΩ
* 16 cavi m-m
* 19 cavi m-f

# Hardware e Datasheet componenti

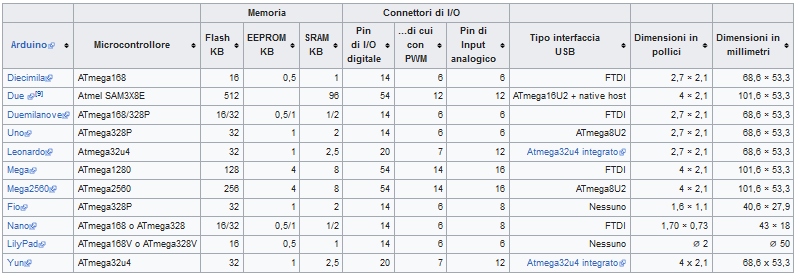
## Arduino Mega 2560

Il componente principale dell’hardware del progetto è l’Arduino nella versione Mega 2560.

Arduino è un microcontrollore composto da varie schede elettroniche di diverso tipo. È stato ideato come strumento per la creazione di prototipi sia hobbistici, sia didattici, sia professionali.

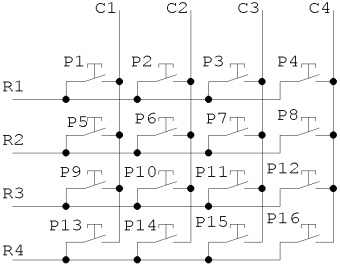
Ad Arduino vengono solitamente collegati più elementi che siano essi sensori (hardware di input) o motori e led (hardware di output). La componente software di Arduino è libero e gratuito, mentre l’hardware ha prezzi molto modesti considerate le potenzialità di esso.

Nel corso della sua storia, Arduino ha subito molte modifiche di vario tipo passando semplicemente da una diversa estetica, ad una diversa struttura delle componenti hardware, ad una totalmente diversa area d’impiego (come ad esempio Arduino LillyPad, flessibile e utilizzato nelle applicazioni su tessuti), di seguito una tabella riassuntiva dei vari modelli (noi utilizziamo Arduino Mega2560):



## Matrice di Pulsanti

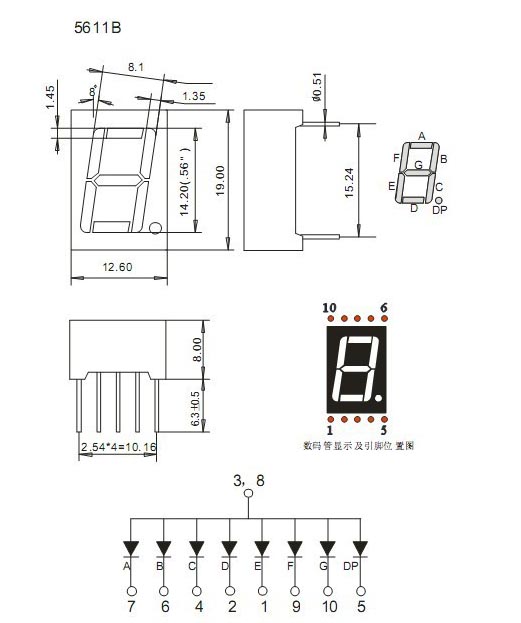
Per immettere il codice, ho utilizzato una matrice di pulsanti simile ad un tastierino numerico come questa (vedi immagine a lato).

Il funzionamento della matrice, visto la schema elettrico può risultare molto semplice, ma per implementare il codice in modo che ogni pulsante venga letto correttamente, è stato ,al contrario, molto complesso infatti ho dovuto utilizzare un estratto di codice trovato in internet per poi modificarlo secondo le mie necessità.

La matrice simula dei valori HEX (ossia numeri da ‘0’ a ‘9’ e lettere da ‘A’ a ‘F’) seguendo l’ordine indicizzato dei pulsanti ed il codice ha una lunghezza di 6 caratteri.

Ogni pin della matrice viene collegato ad Arduino tramite dei cavi m-f ai pin digitali da 4 a 11.

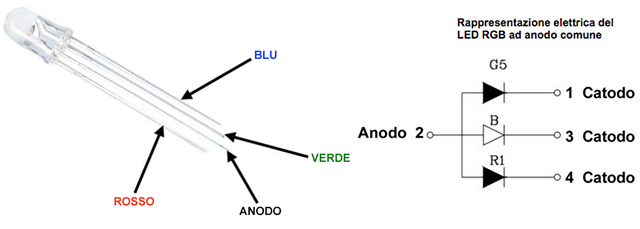
## Display 7 segmenti

Nel modello definitivo del progetto ho utilizzato un semplice display 7 segmenti per permettere all’utente di visualizzare i valori immessi tramite la matrice di pulsanti.

Per impedire fraintendimenti riguardo l’output del display (visto che ci sono lettere che se scritte in un certo modo risulterebbero uguali a numeri) alcune lettere come la ‘B’ e la ‘D’ sono lette correttamente ma ritornate sotto forma di ‘b’ e ‘d’ dal memento che la ‘B’ sarebbe scritta in maniera uguale a ‘8’ e la ‘D’ uguale allo ‘0’.

Il pin ‘3’ del display è collegato tramite una resistenza da 330 Ω al GND, mentre gli altri pin sono collegati ai pin digitali di Arduino da ‘37’ a ‘53’ (solo pin dispari).

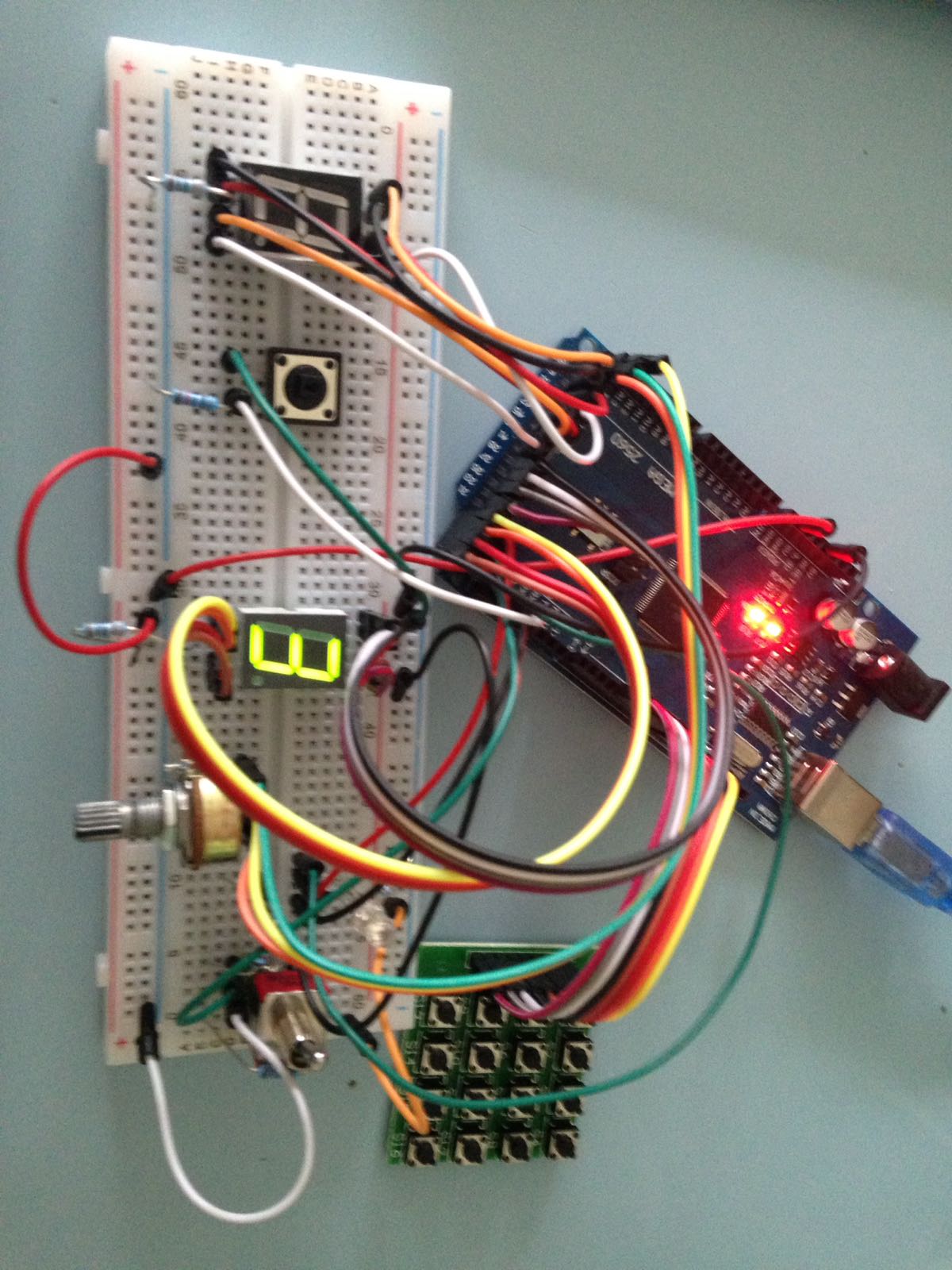
## Led RGB

Per indicare la funzione attiva della “cassaforte” ho deciso di utilizzare un led RGB anodo comune (+5V in comune) come quello accanto.

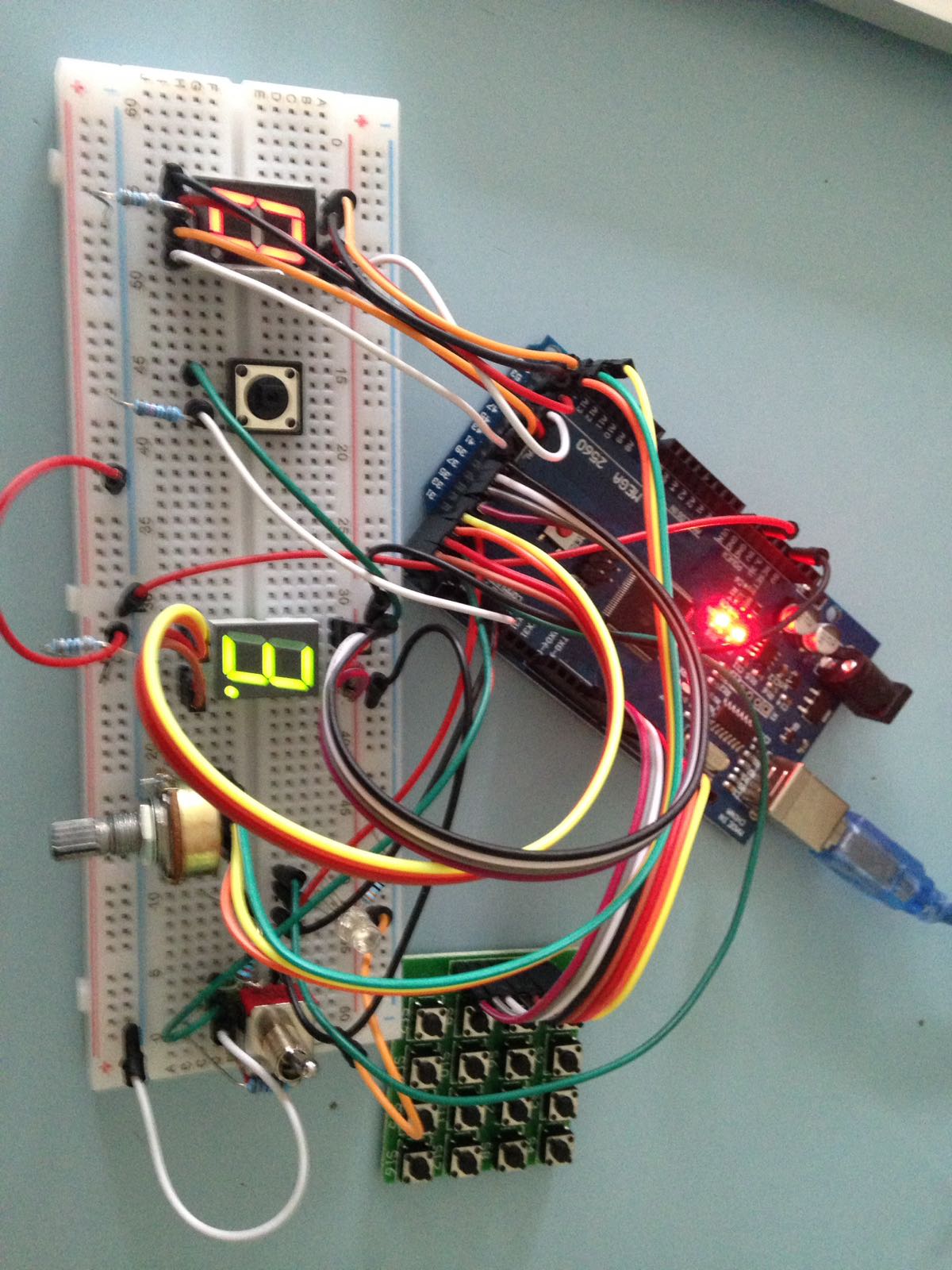
Come suggerisce il nome, ho collegato l’anodo comune al +5V tramite una resistenza dal 330 Ω. In seguito ogni colore corrisponde ad una determinata funzione:

* Blu: il catodo corrispondente al Blu è collegato al pin ‘24’ di Arduino e viene settato a ‘0’ (quindi acceso) quando lo switch che definisce se il codice è in lettura o in scrittura (input pin ‘22’ di Arduino): se il led è spento il codice è in scrittura (=corrisponde al codice di sicurezza attivo); se invece è acceso di colore Blu il codice è in lettura (=corrisponde al codice immesso dall’utente).
* Verde: il catodo corrispondente al Verde è collegato al pin ‘26’ di Arduino e viene settato a ‘0’ (quindi acceso) quando il codice di sicurezza e dell’utente corrispondono
* Rosso: il catodo corrispondente al Rosso è collegato al pin ‘20’ di Arduino e viene settato a ‘0’ (quindi acceso) quando il codice di sicurezza e dell’utente non corrispondono.

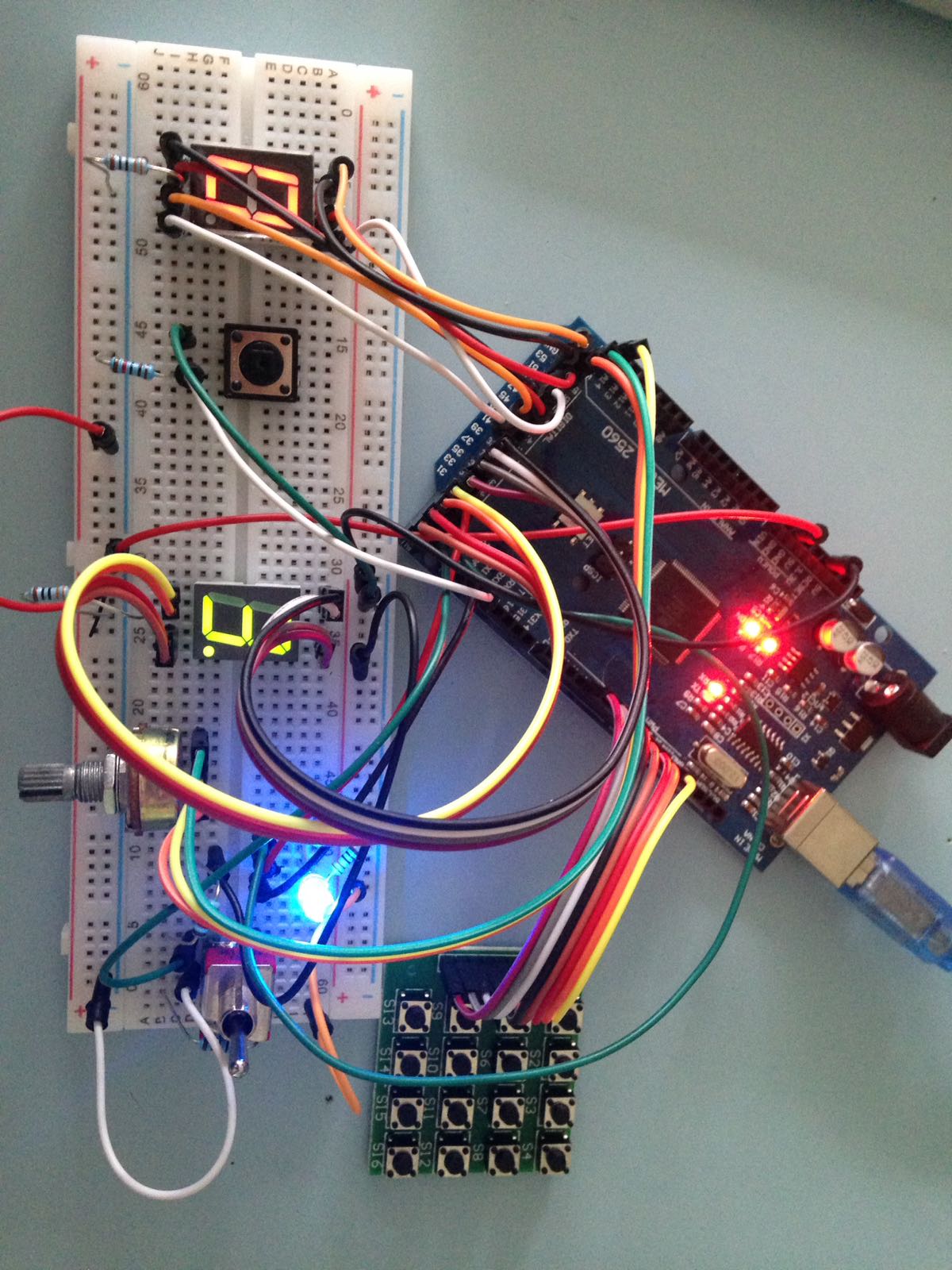
## Foto del montaggio



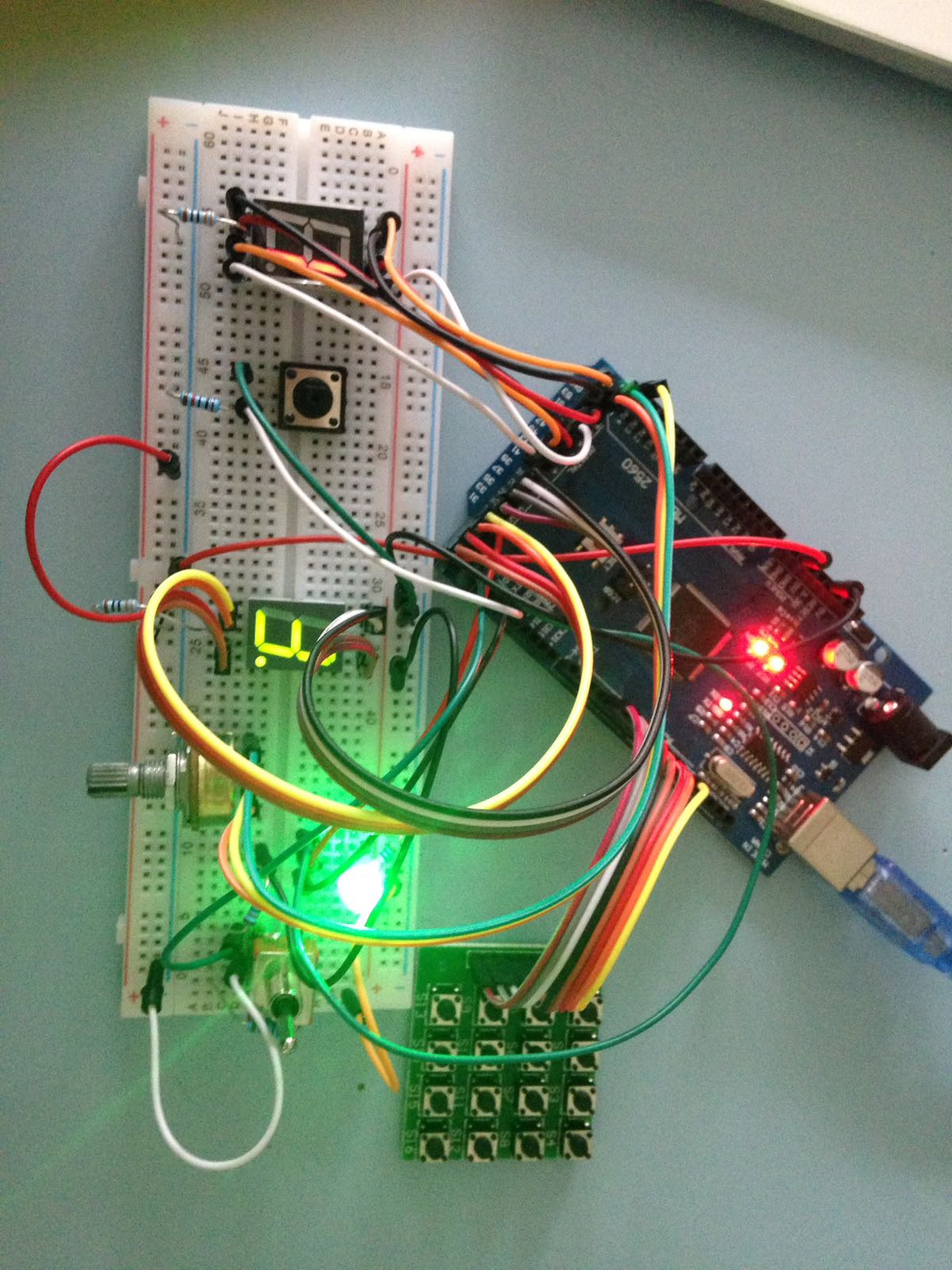
CodeLegnth



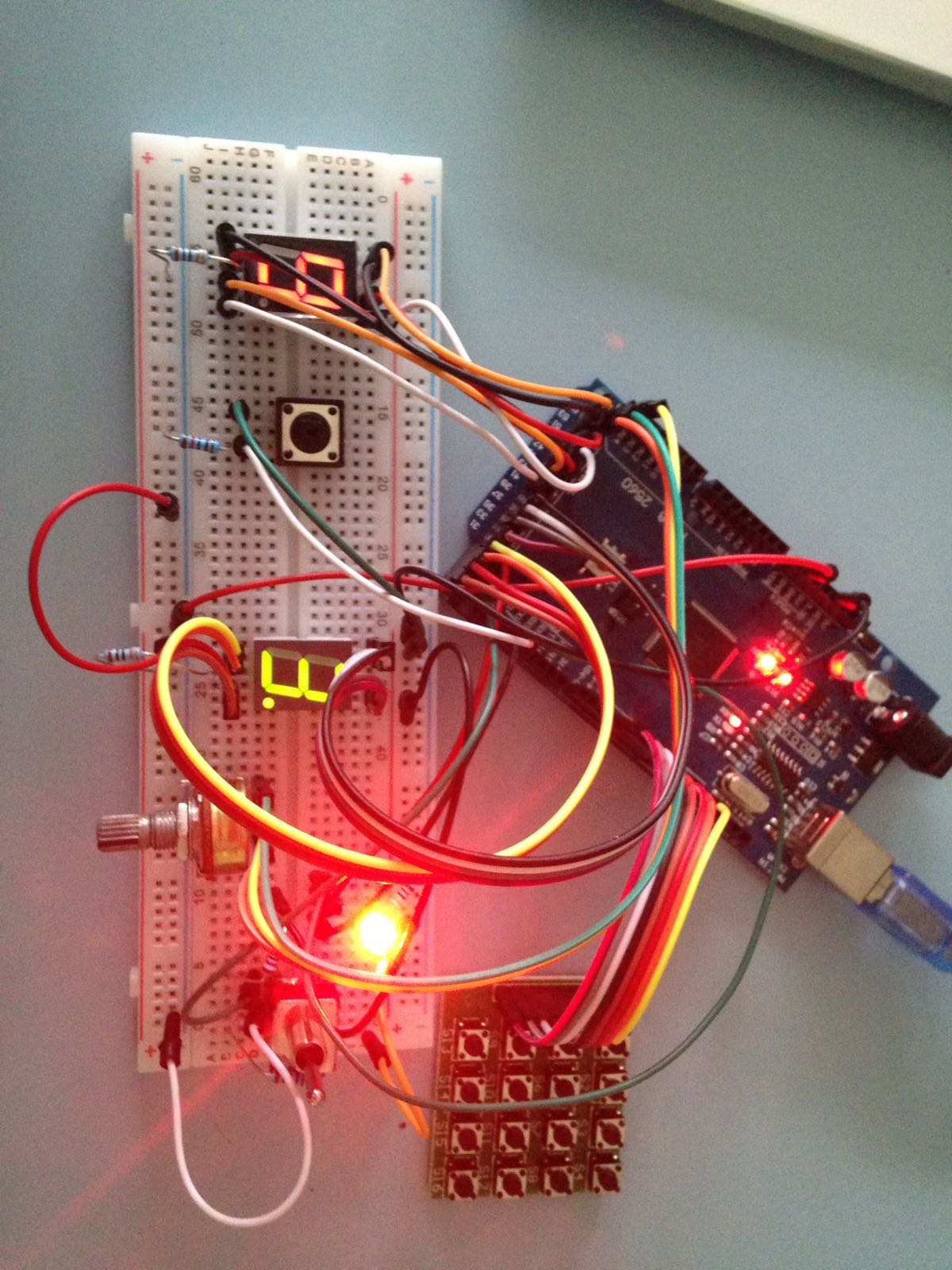
CodeLength was confirmed & SecurityCode is writable



UserCode Insert



UserCode is equal to SecurityCode



UserCode is not equal to SecurityCode

# Software

Il software del progetto è stato sviluppato sulla base di un codice trovato su internet ([haneefputtur.com](http://www.haneefputtur.com)) il quale serviva a leggere i valori emessi dalla matrice di pulsanti, ossia distinguere quale pulsante fosse stato premuto. Mentre il seguito del codice è stato ideato completamente da me.

Pur avendo estrapolato il codice da un esterno, mi sono sforzato di comprendere il suo funzionamento e la sua sintassi; nonostante ciò, alcune linee di codice mi risultano estranee, ad esempio il contenuto   
dell’if (no\_press\_flag==1){…}.

All’interno della funzione loop(){…}, per prima cosa, leggo il valore del potenziometro analogicamente e   
ri-mappo il risultato così ottenuto in modo da ottenere un risultato compreso fra 1 e 9. Questo valore indica la quantità di caratteri che comporranno entrambi i codici. Per informazione dell’utente stampo il valore su di un display 7 segmenti verde. Se è stato premuto un qualunque pulsante della matrice non sarà più possibile modificare la dimensione del codice, per dimostrare questa conferma verrà acceso il DOT del display.

Nell’immediata successione ho integrato la possibilità di resettare il UserCode in modo che sia possibile “ritentare” senza bisogno di ricaricare il programma tramite il pulsante di reset di Arduino e quindi senza dover ri-inserire anche il SecurityCode. Ciò avviene tramite un semplice pulsante.

In seguito verifico se il codice che verrà letto dai bottoni rappresenta il codice di sicurezza o il codice immesso dall’utente e questa selezione viene rappresentata con il led RGB che in un caso (securityCode) è spento, mentre nell’altro (userCode) è acceso di colore blu, la selezione avviene tramite uno switch.

Nelle stesse linee di codice leggo anche i valori verificando che rispettino la lunghezza definita del codice, se vengono premuti un numero maggiore di pulsanti rispetto a quelli concessi, vengono presi in considerazione solo i primi valori (Es.: se la lunghezza massima del codice è 3 e digito “0”,”7”,”A”,”F”,”6”, il codice sarà “07A”).  
Ad ogni pressione di un qualunque pulsante verifico se siamo nello stato del SecurityCode o in quello dell’UserCode.

Nelle ultime linee del codice viene verificato se i due codici combaciano, in caso positivo viene acceso il colore verde del led RGB, mentre in caso negativo viene acceso il colore rosso.

Sparse per tutto il codice si possono trovare linee di stampa tramite interfaccia seriale utilizzate per dei test in fase di sviluppo, nonostante ciò ho deciso di mantenerle per rendere più chiaro il funzionamento del codice.

# Conclusione e Risultati

All’inizio ero convinto si trattasse di un progetto abbastanza banale in quanto non utilizzavo molti componenti a me estranei (non conoscevo solo la matrice di pulsanti e il led RGB), ma sin dai primi istanti, mi sono reso conto che sarebbe stato più complesso di quanto non credessi. Infatti come prima cosa ho tentato di capire il funzionamento della matrice di pulsanti, il quale si è rivelato una vera sfida ma fortunatamente sono riuscito a trovare molte informazioni su internet tra cui una parte di codice che si occupa della lettura degli impulsi ricevuti. In questo modo sono riuscito ad adattare il codice alle mie necessità.

Il circuito funziona correttamente, inoltre, nel corso dello svolgimento ho deciso di implementare nuove funzioni oltre a quelle definite in fase di progettazione. Eventuali sviluppi futuri potrebbero consistere nel rendere più realistico il progetto, integrando magari un display con più valori per la visione completa del codice immesso, oppure l’integrazione di un servo per rendere effettivo lo sblocco (della serratura), oppure ancora l’implementazione di un ulteriore sistema di sicurezza come potrebbe essere un lettore per il badge.

Come nuovi argomenti teorici del mondo della programmazione in generale non ne ho riscontrati, però ho utilizzato per la prima volta gli array in Arduino.

# Fonti

* Display 7 segmenti – Datasheet: <http://www.wayjun.com/Datasheet/Led/Segment%20Digit%20LED%20Display.pdf>
* 4x4 Matrix Membrane Keypad (#27899) – Datasheet: <https://www.parallax.com/sites/default/files/downloads/27899-4x4-Matrix-Membrane-Keypad-v1.2.pdf>
* RGB Led common Anode – Datasheet: <https://www.kitronik.co.uk/pdf/5mm_RGB_led_common_anode.pdf>
* 5 kΩ potentiometer – Datasheet: <https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwizjf3ntKrbAhWCxLwKHQ3zDywQjRx6BAgBEAU&url=https%3A%2F%2Fwww.futurlec.com%2FPotentiometers%2FPOT5KBSWITCH.shtml&psig=AOvVaw3-wIQq8WsSEKEmjNe17sEB&ust=1527665663251581>
* Button – Datasheet: <https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiv_LTZtKrbAhXCXrwKHUu_AJQQjRx6BAgBEAU&url=http%3A%2F%2Firishelectronics.ie%2FSMD-Tact-Switch-Push-Button-6631&psig=AOvVaw2pf5_Jr68BsUHOVztgUdmq&ust=1527665785322911>
* Switch – Image: <https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwjh2pC7tqrbAhXGerwKHW89AugQjRx6BAgBEAU&url=https%3A%2F%2Fwww.jaycar.com.au%2Felectromechanical-components%2Fswitches%2Fc%2F220&psig=AOvVaw2Fzh-w80c_5jdORPAB023C&ust=1527666159801632>