

## Architettura Degli Elaboratori 1 : Progetto Digispark KeyPad

### Obiettivi:

La periferica costruita consiste in un tastierino programmabile con due pulsanti e due modalità operative, intercambiabili tramite l'apposita leva. Nella prima modalità i tasti corrispondono ai comandi di uso frequente durante il debugging: step-into e step-over, mentre la seconda modalità permette l'avvio vero e proprio di script o programmi nella macchina alla quale è collegato secondo le modalità descritte in seguito.

### Funzionamento:

Al suo collegamento, tramite porta USB viene riconosciuto come una tastiera, questo espediente permette che alla pressione dei pulsanti possano essere inviati dei keystrokes.

Nella modalità 1, i due pulsanti sono pensati per poter aiutare un programmatore con il debug. Infatti, avremo sul tasto bianco la pressione del tasto F11, che nella maggior parte degli IDE quali Visual Studio corrisponde allo STEP INTO, mentre premendo il tasto giallo si simulerà la pressione del tasto F10, ovvero STEP OVER.

Nella modalità 2, alla pressione dei pulsanti verrà inviata da parte del micro-controllore una sequenza di comandi, in modo da permettere l'esecuzione di due script o programmi a scelta, presenti nella macchina in uso. In particolare il controllore invia la seguente combinazione di comandi sotto riportata e appositamente commentata:

Pulsante	Tasti Premuti	Azione corrispondente	
		Windows	Linux
Tasto Bianco	Win + R	Apertura menù di sistema "Run"	Apertura terminale <i>(previa installazione)</i>
	"script1"	Scrive all'interno del menu run la stringa "script1"	Scrive dentro terminale la stringa "script1"
	Enter	Preme invio per inviare il comando	Preme invio per inviare il comando
Tasto Giallo	Win + R	Apertura menù di sistema "Run"	Apertura terminale <i>(previa installazione)</i>
	"script2"	Scrive all'interno del menu run la stringa "script2"	Scrive dentro terminale la stringa "script2"
	Enter	Preme invio per inviare il comando	Preme invio per inviare il comando

Figura 1 tabella modalità 2

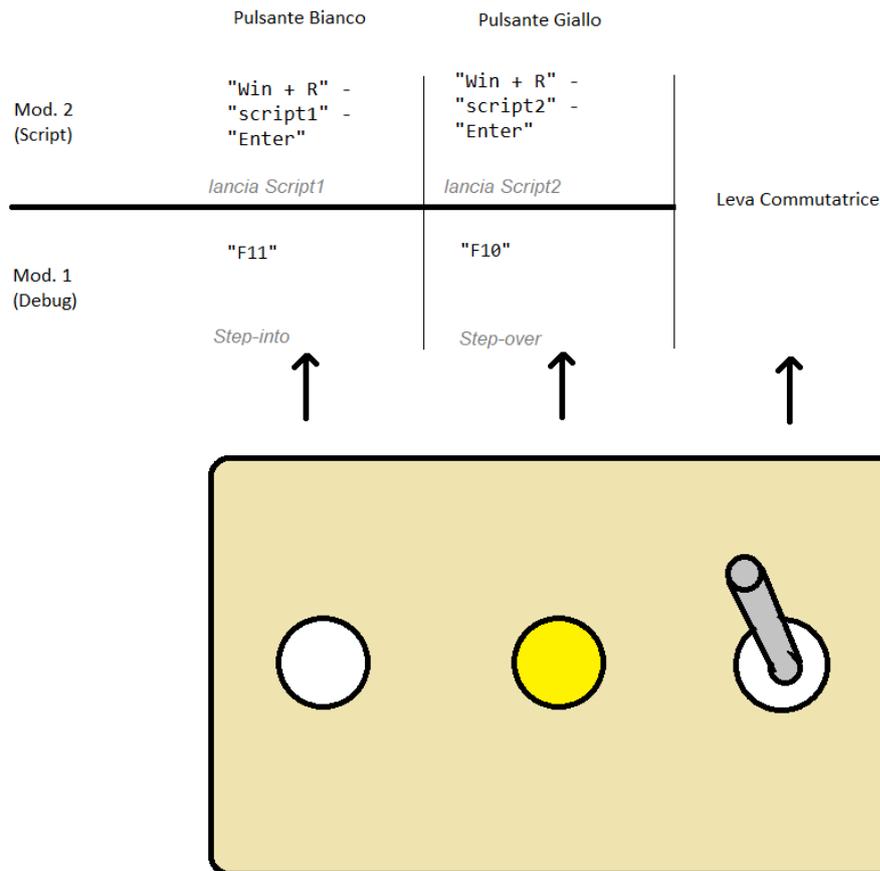


Figura 2 Schema recap. comandi

## Installazione

Per la modalità 1 è sufficiente collegare il dispositivo al computer tramite il cavo usb, e questo verrà automaticamente riconosciuto come una comune tastiera USB.

Per il funzionamento della modalità 2 su Windows l'installazione consiste semplicemente nel posizionare due eseguibili o collegamenti ad essi dentro la cartella di Windows sotto i rispettivi nomi di "C:\Windows\script1" (pulsante bianco) e "C:\Windows\script2" (pulsante giallo). È possibile infatti creare dalle più piccole e semplici utilities in .bat ai più complessi programmi in qualsivoglia linguaggio purché siano eseguibili con il nome riportato, ciò consente per l'appunto enorme flessibilità all'utilizzatore di poter configurare come preferisce le funzionalità dei due script, andando a modificare in autonomia i due file.

Su Linux e Mac è necessario inoltre associare alla combinazione di tasti Win + R l'apertura del terminale, infatti non è lo shortcut di default. Per configurare la combinazione Win + R è suggerito controllare le istruzioni sui rispettivi siti della distribuzione in uso, più precisamente dell'ambiente grafico in uso (GNOME, MATE, KDE, ecc), ad esempio per ubuntu le operazioni da seguire sono le seguenti: <https://help.ubuntu.com/stable/ubuntu-help/keyboard-shortcuts-set.html.en>

Per associare uno script bisogna poi creare due alias sul terminale chiamati “script1” e “script2”. La creazione di un alias è ottenibile editando il file “.bashrc” presente in “/bin/bash” con il proprio editor testuale preferito, creando due righe con la seguente sintassi:

```
Script1=' [command 1]; [command 2]; [command 3]; ..... [command n]'.  
Script2=' [command 1]; [command 2]; [command 3]; ..... [command n]'.
```

Se si vuole creare uno script in bash è possibile lanciare direttamente i comandi nel file bashrc, alternativamente è suggerito lanciare un programma con “./myCustomProg”.

Essendo il microcontrollore adottato una periferica USB-HID compatibile, è stato possibile implementare le funzioni descritte di seguito emulando una tastiera, col vantaggio che il dispositivo viene riconosciuto senza che sia necessaria l’installazione di alcun driver o software supplementare.

Nel nostro caso a puro scopo illustrativo, abbiamo inserito due semplici script già pronti in .bat, uno genera una forma ellittica, l’altro genera un codice “matrix”.

Questo il codice dello script2:

```
@echo off  
Cls  
Color 02  
:top  
Echo %%random%%random%%random%%random%%random%%  
Goto top
```

### Componenti utilizzati:

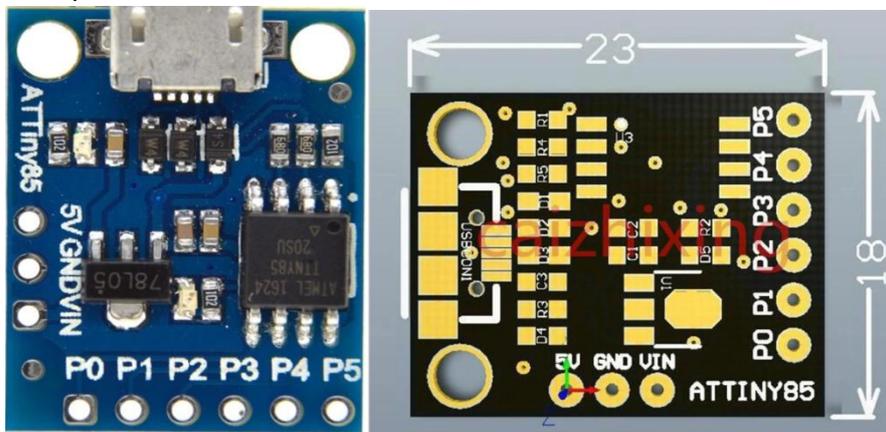
- **Digispark mini pro**

Il **Digispark** è una piccola scheda di sviluppo di piccolissime dimensioni.

E' basata sul microcontrollore **Attiny85**, simile all' ATMEGA montato sui comuni arduini, in versione più economica, più piccolo e con risorse molto limitate.

L'hardware è molto semplice infatti ogni pulsante è collegato direttamente al pin sebbene si sarebbe potuto implementare qualche strategia per collegare più pulsanti su un solo pin usando una lettura “analogica” con un partitore di tensione o usando una matrice di tasti.

Il Software fa largo uso della libreria DigiKeyboard la quale permette di inviare comandi come se questi provenissero da una comune tastiera usb. E' presente un ciclo di polling continuo con ascolto sulla pressione dei tasti in quanto non è possibile l'uso di interrupt su tutti i pin usati.



- Resistenze da 10k
- 2 push button
- 1 switch a leva

## Sketch:

```
#include "DigiKeyboard.h"
#define N_PINS 5
#define MAX_CMD_LEN

const int BIANCO = 0;
const int GIALLO = 1;
const int LEVA = 2;

void preventMissingFirstCharAfterDelay()
{
  DigiKeyboard.sendKeyStroke(0);
}

void setup()
{
  pinMode(BIANCO,INPUT);
  pinMode(GIALLO,INPUT);
  pinMode(LEVA,INPUT);
  pinMode(3,INPUT);
  pinMode(4,INPUT);
  pinMode(5,INPUT);
}

void loop()
{
  preventMissingFirstCharAfterDelay();

  DigiKeyboard.delay(200);

  if(digitalRead(LEVA)==LOW) //se sono in modalit  DEBUG
  {
    if(digitalRead(BIANCO)==HIGH)
    {

      //qua facciamo STEP INTO F11
      DigiKeyboard.sendKeyStroke(KEY_F11);

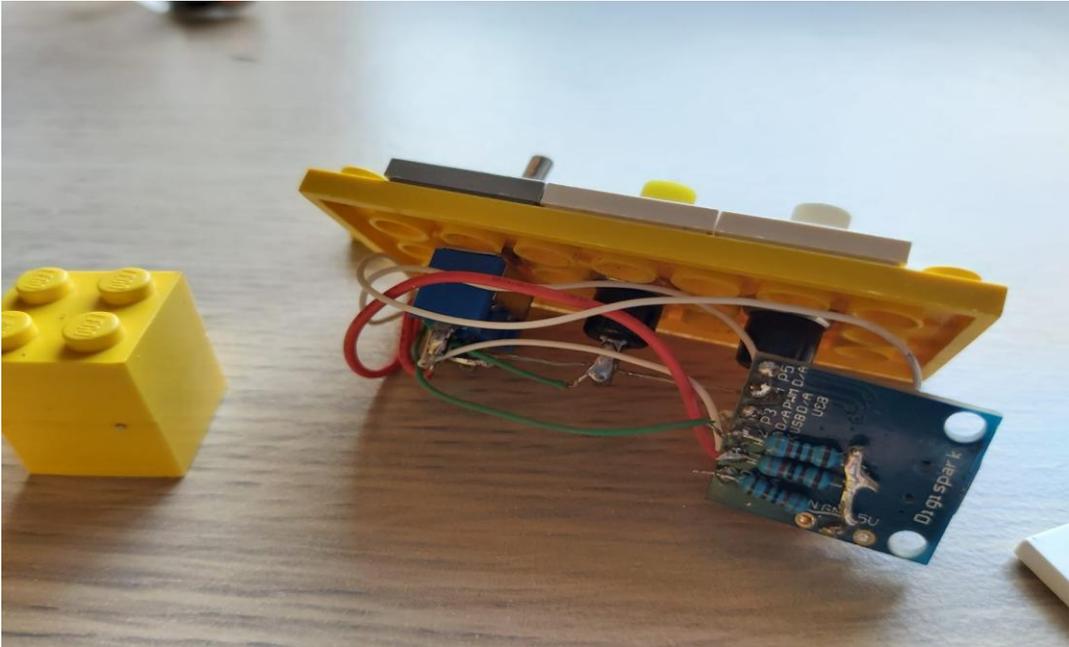
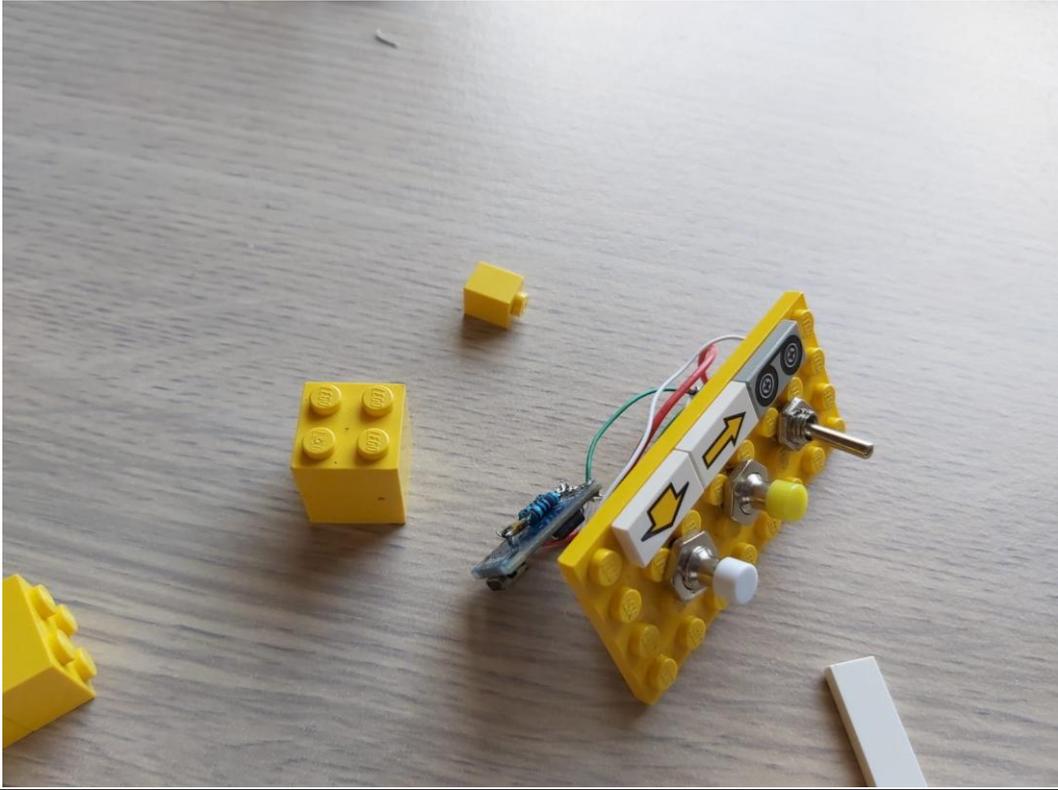
    }
    else if(digitalRead(GIALLO)==HIGH)
    {
      //qua facciamo STEP OVER F10
      DigiKeyboard.sendKeyStroke(KEY_F10);
    }
  }

  else if(digitalRead(LEVA)==HIGH) //se sono in modalit  SCRIPT
  {
    if(digitalRead(BIANCO)==HIGH)
    {

      //qua facciamo script1
      DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
      DigiKeyboard.delay(250);
      DigiKeyboard.println("script1");

    }
    else if(digitalRead(GIALLO)==HIGH)
    {
      //qua facciamo script2
      DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
      DigiKeyboard.delay(250);
      DigiKeyboard.println("script2");
    }
  }
}
```

Foto:



### Utilizzo pratico:

Questo progetto semplice e veloce risolve nel personale caso per cui è stato costruito il sistematico problema che durante le sessioni di debugging premendo senza guardare i tasti F11 e F12 si finisce frequentemente per sbagliare, con l'effetto di perdere la riga di interesse e spesso si deve ricominciare la sessione senza poter tornare indietro. Nella modalità 2 invece l'idea è che gli script permetteranno di collegarsi a 2 differenti VPN senza dover manualmente disconnettersi da una e collegarsi all'altra, risparmiando tempo e pazienza quando si deve passare frequentemente dall'una all'altra.

### Conclusioni e possibili sviluppi:

Il presente progetto ha un costo di realizzazione davvero irrisorio, il componente principale ovvero il digispark mini ha un costo di poco più di 1€ su store quali AliExpress e i pulsanti hanno anche essi un costo molto basso e se si considera che possono essere recuperati da qualsiasi elettrodomestico ci si rende conto di quanto sia bassa la spesa.

Il case è stato realizzato con mattoncini lego per il fascino che gli stessi esercitano e per dare una nuova ad un oggetto ludico dell'infanzia.

Questo semplice progetto, realizzato ad hoc per l'uso che noi ne facciamo, potrebbe essere spunto per chi volesse realizzare delle periferiche di supporto, ai fini di semplificare/rendere accessibile l'uso del computer a persone con disabilità motorie agli arti superiori.

### Video test:

<https://youtu.be/6GLmR5ORxpQ>

