# USB TO TTL KEYBOARD SHIELD by P-LAB -- 2020



### **DESCRIZIONE DEL PROGETTO**

Questo Shield, disegnato e costruito per ARDUINO UNO, consente di collegare una comunque tastiera USB internazionale a qualunque computer vintage che usi una tastiera con interfaccia TTL. È obbligatorio l'uso congiunto con un USB Host Shield.

È stato testato solo il modello "ARCELI USB Host Shield for ARDUINO UNO", altri potrebbero risultare compatibili.



## ESCLUSIONE DI RESPONSABILITA' PER DANNI DA SCARICHE ELETTROSTATICHE

Le scariche elettrostatiche possono danneggiare i vostri computer e le loro periferiche. Prima di qualsiasi operazione sul dispositivo è necessario scaricare l'elettricità statica accumulata dal vostro corpo e prevenirne un nuovo accumulo. Non ci assumiamo alcuna responsabilità per danni, anche gravi o letali, causati a persone / cose / proprietà intellettuali durante l'installazione o l'utilizzo di questo dispositivo.

### PREREQUISITI

Per completare il progetto avete bisogno di:

- Arduino UNO R3 o compatibile,
- ARCELI USB Host Shield per ARDUINO UNO,
- USB to TTL Keyboard Shield (PCB versione 1.1),
- Un cavo adatto alla connessione tra il vostro computer e lo Shield,
- Strisce di contatti per collegare il PCB USB to TTL Keyboard Shield all'USB Host Shield.

## DI QUALI ALTRI COMPONENTI HO BISOGNO?

In realtà: nessuno. Dipende dalle vostre esigenze e dai vostri desideri.

Lo USB to TTL Keyboard Shield non fa troppe cose: aggiunge qualche connettore e qualche pulsante. La parte più interessante viene svolta dal firmware e verrà descritta in seguito.

Quindi: avete necessità di pulsanti di RESET e CLEAR SCREEN perché state pensando di usare lo Shield con un Apple-1? Montateli!

Volete invece vedere come i tasti che premete sono convertiti in bit e segnali elettrici? Montate il display a LED, le reti resistive e godetevi le luci lampeggianti!

A fine montaggio controllate sempre di non avere corti circuiti tra gli Shield.

I componenti obbligatori sono stati già elencati nel paragrafo PREREQUISITI.

## **COLLEGAMENTO AL COMPUTER**

Questa è la parte più complicata: seguite le indicazioni e tutto andrà bene.

Fate <u>estrema attenzione</u> ad ogni singolo collegamento o rischierete di danneggiare il computer.

Spesso infatti, su computer datati, sono presenti tensioni di +12V, -12V e -5V, in aggiunta al solito +5V: una connessione sbagliata potrebbe portare queste tensioni in posti sbagliati e procurare **ISTANTANEAMENTE** danni molto gravi.

Ancora: non siamo responsabili di qualunque danno causato alle vostre apparecchiature.

L'*USB to TTL Keyboard Shield* è stato creato avendo in mente macchine come Apple-1, Apple II e Apple II+, per questa ragione troverete due connettori DIP-16 specificatamente predisposti per questi computer.

La disposizione dei segnali e la loro nomenclatura ricalca quella dei computer originali, quindi non dovrebbe essere troppo difficile costruire il cavo adeguato.

Controllate almeno tre volte il cavo e le connessioni +5V e GND prima di collegare il resto.

Aiutatevi con un multimetro e assicuratevi che giungano sui giusti pin dello Shield.

I pin senza nome non sono connessi a nulla, considerateli dunque *sicuri* per attestarci eventuali tensioni non utilizzate dallo Shield (es. +12V, -12V, -5V): non causeranno alcun danno.

Notate che i Data bits sono chiamati **B1..B8** sui connettori Apple-1/Generic TTL e **D0..D6** sul connettore Apple II/II+. Apple II/II+ usa solo 7 bit.

Il connettore "Generic TTL" può essere usato per altri computer.

Tutto quello che dovrete fare sarà assemblare il cavo corretto basandovi sullo schema elettrico del computer e i vari segnali la cui funzione è indicata chiaramente sul PCB del USB to TTL Keyboard Shield.

MAI disconnettere il computer da USB to TTL Keyboard Shield mentre è acceso.

### **DESCRIZIONE DEI SEGNALI**

SEGNALE	SIGNIFICATO
5V	+5V: Alimentazione proveniente dal computer, alimenta l'intero dispositivo.
GND	Massa.
~RST	/RESET: Segnale Negato (Attivo basso). Normalmente +5V. Va a 0V quando viene premuto il tasto reset o quando richiesto dal firmware.
CLR	CLEAR SCREEN: Segnale Positivo (Attivo alto). Normalmente 0V. Va a +5V quando viene premuto il tasto CLEAR SCREEN o quando richiesto dal firmware.
STB	Segnale di STROBE, gestito dal firmware. Normalmente 0V, va a +5V per 1 millisecondo quando un tasto valido viene premuto.
B1B8	Data Bits, B1 è il meno significativo, B8 il più significativo. Questi segnali sono chiamati D0D6 sul connettore Apple II/II+.

## ALTRE INFORMAZIONI SUI COMPONENTI AGGIUNTIVI

Se non avete intenzione di installare pulsanti e LED potete saltare questo paragrafo. Diversamente tenete in mente che:

- I pulsanti sono Normalmente Aperti, i contatti devono chiudere sul lato corto.
- La barra LED necessita di due Reti Resistive <u>e</u> di due resistenze da 470 Ohm (RD1 e RD2).
- La sagoma della barra LED ricalca il modello MV57164, altri potrebbero risultare compatibili.
- Ogni Rete Resistiva SIL è composta da 4 resistenze da 470 Ohm, senza pin comune. Ossia:



In alternativa è possibile usare 8 resistenze singole da 470 Ohm 1/8W, quattro per ogni rete resistiva.

Esse andranno disposte come in figura.

- Per abilitare i LED bisogna cortocircuitare le piazzole "LED ENA" Se questa operazione non viene fatta, i LED rimarranno isolati dalla massa e non si accenderanno.

#### SIGNIFICATO DEI LED

#### LED SIGNIFICATO

- READY Se questo LED è acceso l'USB to TTL Keyboard Shield è pronto e in attesa dei tasti che premerete. All'avvio ci vorranno alcuni secondi affinché ciò avvenga, a causa del processo di riconoscimento della tastiera da parte dello Shield USB HOST. Se la tastiera USB viene disconnessa il LED si spegnerà. Quando la tastiera USB verrà riconnessa, dopo pochi secondi il LED si accenderà. La tastiera USB può essere scollegata e ricollegata senza spegnere il computer.
- **STB** Segnale di STROBE, come descritto in precedenza. La luminosità potrebbe apparire debole a causa della breve durata dell'impulso. Questo non è indice di malfunzionamento. Non modificate il valore di RD2 per aumentarne la visibilità per non incorrere in sovraccarichi.
- **DATA** Questi otto LED si accenderanno in funzione del valore binario corrispondente al codice ASCII del tasto che avete premuto. Il LED più a destra rappresenta il bit meno significativo (B1/D0), quello più a destra il più significativo (B8).

## **USO DEL DISPOSITIVO**

In condizioni normali il firmware si occupa di tradurre i tasti che premete sulla tastiera USB in un formato comprensibile al computer che avete collegato.

Esso tiene però <u>anche</u> conto di ciò che accade sulla porta seriale/USB di Arduino.

Sì: È possibile collegare un terminale alla porta USB di arduino e usarlo come tastiera remota.
Sarà sufficiente collegare Arduino al vostro computer personale tramite il cavo USB e usare un qualunque programma terminale (PuTTY, minicom, etc) con i seguenti parametri:
9600 baud, No parità, 8 bit di dati, 1 bit di stop.

Sarete così in grado di scrivere sul vostro computer vintage via terminale **e** tramite tastiera USB locale (non ESATTAMENTE nello stesso momento, naturalmente...). I due metodi possono coesistere.

Questa funzione è utile per automatizzare l'inserimento di programmi che avete scritto e memorizzato altrove (magari su file di testo), senza temere errori di battitura. Ci arriveremo tra poco.

Come già detto, questo prodotto è stato pensato principalmente per Apple-1, quindi alcune caratteristiche sono dedicate proprio a questo computer.

Il richiamo delle funzioni è leggermente diverso se si opera da una tastiera USB locale o da un terminale remoto. Vediamo in dettaglio:

#### TASTI SPECIALI

#### CLEAR SCREEN [solo Apple-1]

La funzione viene richiamata con i seguenti tasti o combinazioni:

- **PRTSCN** (Print Screen/SysRq) sulla tastiera USB locale
- F11 sulla tastiera USB locale
- **Control+C** via terminale

### RESET

La funzione di RESET viene richiamata con i seguenti tasti o combinazioni:

- Pause/Break sulla tastiera USB locale
- F12 sulla tastiera USB locale
- Control+R via terminale

#### **FUNZIONI SPECIALI**

#### SCHERMATA DI AIUTO

Una breve schermata di aiuto può essere richiamata con i seguenti tasti o combinazioni:

- F1 sulla tastiera USB locale
- Control+Q via terminale

#### PROGRAMMA DI TEST DEI CARATTERI [solo Apple-1]

Un piccolo programma di test, che mostra a schermo ciclicamente tutti i caratteri, verrà automaticamente digitato e lanciato con i seguenti tasti o combinazioni:

- **F2** su tastiera USB locale
- Control+W via terminale

Per terminare l'esecuzione del programma, resettare il computer.

#### TEST DELLA MEMORIA RAM [solo Apple-1]

È possibile caricare un programma di test della memoria RAM con i seguenti tasti o combinazioni:

- **F3** su tastiera USB locale
- Control+E via terminale

Alla fine della digitazione automatica, prima di lanciare il test è necessario inserire l'indirizzo di partenza e l'indirizzo di fine +1 inserendoli nelle locazioni di memoria 0-3 (in formato Little-Endian).

Esempio: per testare da \$E000 a \$EFFF digitare:

0: 00 E0 00 F0 {ENTER} Esempio 2: per testare da \$0500 a \$0700 digitare: 0: 00 05 01 07 {ENTER}

#### Per eseguire il programma di test:

280R {ENTER}

Dopo qualche tempo, in funzione della quantità di memoria che avete deciso di testare, appariranno i risultati. Riferitevi al link in Bibliografia per ulteriori dettagli riguardo l'autore o la metodologia di test.

#### MENU CFFA1 [solo Apple-1 + scheda CFFA1]

Si può richiamare il programma di gestione della scheda CFFA 1 mediante i seguenti tasti o combinazioni:

- **F4** sulla tastiera USB locale
- **Control-T** via terminale

L'indirizzo 9000R verrà digitato automaticamente e il prompt CFFA1> apparirà immediatamente.

# **OTTAVO BIT / MAIUSCOLE / MINUSCOLE...**

Alcuni computer vintage hanno la caratteristica di interpretare solo una parte dell'intero set di caratteri ASCII.

Per esempio: Apple-1 elabora/interpreta solo lettere maiuscole e può visualizzare solo lettere maiuscole. Se si digita una lettera minuscola, essa verrà visualizzata a schermo (maiuscola) ma non sarà interpretata. Inoltre: Apple-1 richiede che l'ottavo dei Data bit sia a livello logico 1, altrimenti non interpreterà alcun comando, anche se sembrerà comparire correttamente sullo schermo.

La situazione appena descritta è chiaramente non standard, e il fatto che i caratteri siano comunque visualizzati a schermo riflette scelte tecniche progettuali che non sono in discussione in questo documento.

Il firmware può gestire l'ottavo bit in modo da creare o meno la mappatura richiesta da Apple-1:

- F7 su tastiera USB (default al boot) → l'ottavo bit è forzato a 1, e ogni lettera minuscola digitata verrà automaticamente resa maiuscola. Il LED corrispondente all'ottavo bit sarà acceso).
- F8 su tastiera USB → l'ottavo bit sarà libero. Non ci sarà conversione da minuscole a maiuscole.
   In questa modalità Apple-1 non interpreterà alcun comando ma quanto digiterete apparirà comunque su schermo.

Questa modalità potrebbe essere utile per computer più aderenti agli standard.

FUNZ.	+ -   	PREMO	+ -   ,	USCITA	·+- 	VALORE	+•	APPLE-1 MOSTRA	+ -   	INTERPRETATO
F7	+ - 	a	+ - 	11000001	- + -	193	+ · 	 А	+ <b>-</b> 	SI
F7		A		11000001		193		A		SI
F8		а		01100001		97		A		NO
F8		A		01000001		65		A		NO

#### ESEMPI DI MAPPATURA CON DIFFERENTI SETTAGGI DELL'OTTAVO BIT

Se state usando un Apple II/II+, predisponete F7/F8 secondo le vostre preferenze. Apple II/II+ non richiede l'ottavo bit, ma la conversione minuscole/maiuscole potrebbe comunque rivelarsi utile.

## **TASTIERE SENZA FILI**

Alcune tastiere economiche senza fili si sono rivelate perfettamente funzionali. In alcuni casi, tuttavia, i tasti PRINT SCREEN e PAUSE/BREAK sono risultati inesistenti. Per tali funzioni sono stati pertanto predisposti i tasti F11 e F12, come descritto in precedenza.

## TASTIERE LOCALIZZATE

Le tastiere con tasti specifici per una particolare lingua (ad es. Italiano/Spagnolo etc.) verranno comunque mappate come internazionali, quindi alcuni tasti potrebbero non corrispondere. Per risolvere questo inconveniente potrebbe essere necessario modificare le librerie USB HOST di Arduino.

Questa operazione non è trattata in questo documento ma ogni contributo è il benvenuto.

### SCRITTURA DA REMOTO TRAMITE TERMINALE E TRASFERIMENTO FILES

Quando si usa un comune programma terminale, come minicom o PuTTY, non ci sono da prestare particolari attenzioni. In buona sostanza quello che digitate sulla tastiera appare sul computer remoto. Tuttavia, se avete intenzione di caricare qualche programma o routine in LM, magari da un file di testo che avete salvato, è necessario inventarsi qualcosa per evitare di perdere informazioni nel trasferimento. Questo succede perché naturalmente nessun computer è progettato per accettare input da tastiera oltre una certa velocità. Se la velocità di scrittura diventa "super-umana" si cominceranno a perdere caratteri, tanto che poi il programma scritto sarà inutilizzabile.

Abbassare il Baud Rate della porta seriale, oltre a non essere sufficiente, avrebbe reso la scrittura da remoto troppo lenta e innaturale.

È altresì perfettamente chiaro che questo non è il modo corretto per trasferire programmi ad un computer. Tuttavia, visto che c'è la possibilità... perché non fare un tentativo?

Questo piccolo script BASH per Linux manda al DEVICE appropriato il file indicato al momento del lancio, carattere per carattere, interponendo dei ritardi (ricavati sperimentalmente) tra i caratteri.

I ritardi indicati sono calibrati per Apple-1 INTEGER BASIC. Lo script può ovviamente essere migliorato, ma già così funziona correttamente.

Potrebbe essere necessaria una ulteriore "finitura" del file prima di mandarlo al computer, specialmente per quel che riguarda la gestione del terminatore di linea. Lo script riportato lavora bene con LF (\$0A, che è tipico dei sistemi Linux) come terminatore di linea.

Se il vostro file proviene da un sistema Windows, potrebbe usare CR+LF (\$0D + \$0A) come terminatore di linea. Questo potrebbe causare, con lo script sotto riportato, l'effetto di un doppio ENTER.

Ogni contributo migliorativo (specialmente per utenti Windows) è il benvenuto.

Quindi: se avete chiamato il lo script BASH 'sendfile.sh' ed il file che volete inviare al computer si trova nella stessa directory ed è chiamato 'myprogram.txt' il comando da dare sarà: sudo ./sendfile.sh myprogram.txt

Nota: lo script per il trasferimento dei files va necessariamente <u>lanciato da una nuova shell mentre il</u> <u>programma terminale continua a girare nella propria</u>. Diversamente, Arduino entrerà in un loop ciclico di reset. Naturalmente non dovrete premere nulla nella shell dove è in funzione il programma terminale.

```
#!/bin/bash
DEVICE="/dev/ttyACM0"
stty -F $DEVICE 9600 cs8 -cstopb -parenb
INPUT=$1
while IFS= read -r -n1 char
do
       if [ "$char" = "" ];
               then
               echo -n -e '\x8d' > $DEVICE
              sleep 0.5
              echo
       fi
       echo -n "$char" > $DEVICE
       echo -n "$char"
       sleep 0.06
done < "$INPUT"
```

Siate consapevoli che il trasferimento è lento e dovrebbe essere usato solo per file piccoli.

È stato calcolato che con la migliore ottimizzazione, in termini di lunghezza delle righe e temporizzazioni, 4 kBytes verrebbero trasferiti in 4-5 minuti.

L'interfaccia cassette trasferisce la stessa quantità di dati in circa 30 secondi.

#### <u>È IMPERATIVO DISCONNETTERE IL CAVO USB DA ARDUINO <mark>PRIMA</mark> DI SPEGNERE IL COMPUTER.</u>

Se ciò non viene fatto, la tensione di 5V proveniente dalla porta USB del vostro computer personale continuerebbe ad alimentare l'altro computer, rischiando seriamente di danneggiarlo.

## **BIBLIOGRAFIA / RICONOSCIMENTI**

Video ufficiale USB to TTL Keyboard Shield https://youtu.be/ZIYEnyh8RC0

Apple-1, Manuale delle Operazioni https://www.applefritter.com/files/a1man.pdf

Memory Test per 6502 di Mike Willegal https://www.willegal.net/appleii/6502mem.htm

Pagina della scheda CFFA1 di R&D Automation's. http://dreher.net/?s=projects/CFforApple1&c=projects/CFforApple1/main.php

## ADDENDUM

Per funzionare correttamente i pad su "ARCELI USB Host Shield for ARDUINO UNO" devono essere predisposti (saldati o aperti manualmente, se necessario) come appare nell'immagine sottostante:

In particoare:



Pad VBUS PWR 3.3V:	APERTO
Pad VBUS PWR 5V:	CHIUSO
Pad OTHER 3.3V PAD:	CHIUSO
Pad OTHER 5V PAD:	CHIUSO