

# APPLE-1 TERMINAL

v1.0

## 1. DESCRIZIONE

*Apple-1 Terminal* è un dispositivo che vi consente di utilizzare pienamente il vostro Apple-1 Originale, Full-Size Replica o Briel Computers Replica-1 Plus **mediante un terminale seriale USB**.

Sarà dunque possibile interagire con il computer senza bisogno di una tastiera esterna o di un monitor dato che l'output del computer verrà replicato sul terminale.

La scheda aggiunge anche altre funzionalità interessanti come, ad esempio, **la gestione degli escape codes ANSI** e altro ancora.

I possibili utilizzi per una simile soluzione sono molteplici, ad esempio:

- Ricerca di guasti, sia sulla sezione computer che su quella terminale,
- Utilizzo remoto della macchina,
- Sviluppo di programmi e applicazioni sfruttando o meno le funzionalità aggiuntive.

La scheda è disegnata per essere inserita sia sul connettore verticale che su quello orizzontale. In quest'ultimo caso sarà possibile continuare a utilizzare le schede aggiuntive come Juke-Box, microSD card, modem Wi-Fi, etc.

Se desiderate usare questa opzione, il connettore EDGE va montato in corrispondenza del riquadro bianco, dal lato opposto dei componenti che devono essere dunque sempre rivolti verso l'esterno.

Per il verso di inserzione riferirsi anche ai numeri/lettere guida 1, A e 22, Z che devono corrispondere a quelli presenti sulla motherboard.

Per funzionare, la scheda **non ha bisogno di alcuna modifica o predisposizione specifica** del computer ed è compatibile con qualunque configurazione di memoria.

Dopo aver assemblato/costruito il circuito, programmate il microcontrollore Arduino Nano con il codice sorgente disponibile sulla pagina di riferimento. È possibile che venga richiesta l'installazione della libreria EEPROM.

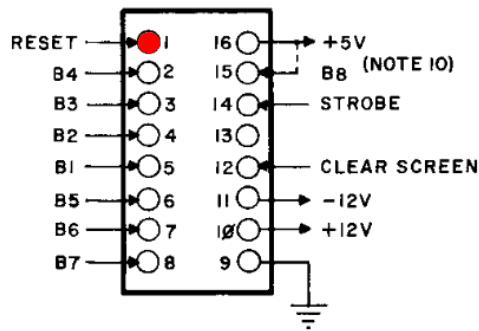
### AVVERTENZE:

- MAI lasciare il cavo USB collegato con il computer Apple spento.
- Fare estrema attenzione alla costruzione e all'orientamento della scheda stessa e dei connettori del cavo piatto della tastiera. Una errata costruzione o una errata inserzione portano al **sicuro danneggiamento immediato del computer** e dell'adattatore.
- Apple-1 non è una macchina progettata per gestire più schede di espansione contemporaneamente. L'uso di questa scheda in abbinamento con altre *potrebbe* in certi casi comportare malfunzionamenti e perdita di dati.

## 2. COLLEGAMENTO TASTIERA

*Apple-1 Terminal* interagisce con il computer tramite l'interfaccia tastiera esistente, che mette anche a disposizione l'essenziale funzione CLEAR SCREEN.

È necessario collegare opportunamente, a computer spento, la scheda terminale e il computer con un cavo piatto opportunamente intestato.



Il punto rosso nel disegno qua sopra evidenzia il pin numero 1. Sulla motherboard del computer e sull'adattatore **il pin 1 è contrassegnato da un punto bianco.**

*Un errore in tali collegamenti può portare a danni immediati e permanenti alla scheda terminale e al computer: controllate sempre più volte il lavoro svolto.*

Una volta assemblati PCB e flat cable, è opportuno controllare che non si siano commessi errori nella costruzione in questo modo:

- A computer spento: collegare il flat cable allo zoccolo "KEYBOARD",
- Rimuovere Microcontrollore e tutti i circuiti integrati, se presenti,
- Inserire la scheda terminale nel connettore designato,
- Collegare l'altra estremità del flat cable alla scheda terminale,
- Accendere il computer (deve comparire l'usuale schermata iniziale),
- Verificare la presenza di +5V tra il pin 20 (VCC) e il pin 10 (GND) del 74HC374.

Se tale tensione è presente spegnete il computer e procedete con l'assemblaggio, diversamente verificate cablaggi e saldature.

*Il dispositivo è sensibile all'elettricità statica, come il vostro computer, e potrebbe venirne danneggiato.*

*Prima di qualsiasi operazione sull'adattatore o sul computer 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.*

### 3. UTILIZZO

Accendete per primo il computer Apple, poi collegate il cavo USB.

Predisponete il programma terminale del vostro computer casalingo con i seguenti parametri:

Baud rate: **230400**, 8 bit di dati, 1 bit di stop, Nessuna parità.

Le ragioni per questo baud rate particolarmente elevato verranno esposte più avanti.

Dovrebbe comparire il seguente messaggio di riepilogo:

```
Apple-1 serial terminal by P-LAB  
-----
```

```
CTRL-L ==> Clear screen  
CTRL-R ==> Reset  
CTRL-O ==> Toggle:  
Uppercase for OUTGOING characters: ACTIVE  
CTRL-I ==> Toggle:  
Uppercase for INCOMING characters: not active
```

(nota: lo stato incoming/outgoing potrebbe essere differente)

Da questo momento è possibile utilizzare il computer Apple nella maniera tradizionale, come se si fosse davanti alla sua tastiera e al suo monitor.

I comandi sono intuitivi ma vengono qui riassunti i principali due:

- CONTROL-L           -CLEAR SCREEN- cancella lo schermo (sia quello reale che quello a terminale)
- CONTROL-R           -RESET- resetta il computer

Impartite dunque i comandi desiderati (tipicamente CONTROL-L e CONTROL-R) e operate normalmente.

I comandi relativi alle minuscole/maiuscole vengono descritti nel seguente paragrafo.

### 4. MAIUSCOLE e minuscole

Il videoterminale nativo di Apple-1 ha diverse limitazioni, una delle quali è sicuramente l'impossibilità di mostrare *nativamente* a schermo del testo minuscolo.

Sia il computer sia INTEGER BASIC, però, **sono perfettamente in grado di distinguere, memorizzare e utilizzare anche le lettere minuscole.**

Sul nostro terminale, e solo su di esso, sarete in grado di stampare a schermo stringhe e testo minuscolo come più vi aggrada. Esso continuerà ad essere visualizzato in maiuscolo solo sul monitor connesso al computer.

I comandi al WOZ monitor, le istruzioni, gli Statement del BASIC etc. dovranno essere sempre digitati in MAIUSCOLO o non verranno interpretati.

Per facilitare le operazioni sono state previste due voci in più che compaiono all'avvio: CONTROL-I e CONTROL-O. Esse operano in modalità alternata, ossia attivano o disattivano la funzione desiderata con lo stesso comando. Lo stato di funzionamento viene visualizzato da un breve messaggio a schermo.

Descrizione delle funzioni:

- CONTROL-I -INCOMING- forza o meno la visualizzazione in maiuscolo di tutto quello che viene mandato dal computer verso il terminale. Se qualche contenuto è memorizzato in minuscolo esso comparirà a terminale in maiuscolo. Questa funzione di fatto imita il comportamento del videoterminale originale.
- CONTROL-O -OUTGOING- forza o meno la traslazione in maiuscolo dei caratteri alfabetici digitati in minuscolo da terminale prima di inviarli al computer Apple. In questo modo sarà possibile programmare, ad esempio, delle stringhe contenenti caratteri minuscoli.

La vostra preferenza viene salvata nella EEPROM del microcontrollore, quindi alla successiva riaccensione del dispositivo la troverete come l'avete lasciata.

La combinazione di questi due comandi può indurre in errore, soprattutto se si è forzata la modalità "minuscola" per l'invio e quella "maiuscola" per la ricezione.

In questo caso i comandi e le istruzioni verranno effettivamente impartiti in caratteri minuscoli, visualizzati come maiuscoli ma non verranno interpretati o restituiranno un messaggio di errore.

## 5. VELOCITÀ, VELOCITÀ, VELOCITÀ

Un'altra forte limitazione del videoterminale nativo è la velocità di scrittura a schermo, che equivale a ben **1/60 di secondo per ogni carattere scritto sullo schermo.**

Quanto visto finora, seppur vantaggioso, rende l'utilizzo del computer tramite terminale esattamente uguale a quello che si avrebbe utilizzando tastiera e monitor nativi. La scheda terminale però offre qualcosa in più.

Essa può essere molto più veloce di così, a patto che si istruiscano programmi e applicazioni a non usare le routine di stampa a schermo del WOZ monitor o, in alternativa, a non usare proprio la PIA (Peripheral Interface Adapter - il grosso chip 6820 di fianco al microprocessore 6502) per la scrittura a schermo.

La scheda terminale, oltre ad "origliare" i caratteri che il microprocessore chiede alla PIA di stampare a schermo, rimane in ascolto anche per delle scritture dirette ed è proprio grazie a questa possibilità che è possibile **velocizzare enormemente la scrittura a terminale.**

Dato che la scheda non deve aspettare che il carattere venga stampato a schermo prima di accettare il successivo i programmi possono puntare a questo range di indirizzi per visualizzare il loro output.

Va da sé che in questa modalità di utilizzo il monitor principale non mostrerà nulla oltre quanto già presente.

Sono stati condotti molti test (casi estremi, di nessuna utilità pratica) sulla massima velocità raggiungibile, che sopravanza di gran lunga il numero massimo di interrupt e di scritture sulla porta seriale che il microcontrollore è in grado di gestire.

Tuttavia, i programmi scritti in BASIC (interpretati, dunque) ben si prestano a questa ottimizzazione in virtù del fatto che le scritture a schermo sono il risultato di molte operazioni e dunque sono sufficientemente diluite nel tempo "by design" da non rappresentare un grosso problema per il microcontrollore.

È stato deciso di provare a modificare INTEGER BASIC in virtù della grande disponibilità di programmi: modificando l'interprete tutti i programmi ne gioveranno.

La versione **FTBASIC (Fast Terminal)** dell'**INTEGER BASIC**, presente sul repository <https://p-l4b.github.io/terminal> è il solito BASIC di sempre ma esso, anziché scrivere a schermo tramite la PIA, scrive a terminale tramite la scheda.

La modifica è stata effettuata da **Antonino Porcino** e le routine aggiuntive non hanno influito sulle dimensioni del programma, che rimane di 4096 bytes. Egli ha infatti utilizzato delle aree di memoria non usate dal codice originale.

È stato scelto un baud rate pari a 230400 bit/s perché a valle delle modifiche è quello più sicuro per non perdere caratteri anche nelle condizioni più sfavorevoli.

Se volete dunque utilizzare gli stessi giochi o programmi di sempre, ma in maniera molto più veloce, semplicemente **caricate FTBASIC dalla vostra microSD card** (o dal file audio fornito nella stessa pagina) invece che il BASIC classico.

*Visitate di tanto in tanto il repository GitHub sopra menzionato per scoprire se altri interpreti o linguaggi sono stati “velocizzati”!*

Anche i programmi in linguaggio macchina possono essere velocizzati. Solitamente le scritte a schermo vengono effettuate da delle chiamate (JSR -Jump to SubRoutine) alla famosa routine ECHO del WOZ monitor, che risponde all'indirizzo \$FFEF:

```
JSR $FFEF → 20 EF FF
```

Dato che sia la ECHO che la scheda seriale visualizzano il contenuto dell'Accumulatore (ossia il codice ASCII del carattere che vogliamo scrivere a schermo) verrebbe spontaneo sostituire questi tre byte con una scrittura dell'Accumulatore sull'indirizzo mappato della scheda seriale \$DF12:

```
STA $DF12 → 8D 12 DF
```

Se però il codice effettua troppe scritte ravvicinate **si potrebbero perdere dei caratteri**, anche con Baud rate elevati (ad es. 2000000 bit/s).

A titolo di esempio: il programma di test dei caratteri ASCII ha bisogno di 20 istruzioni NOP (No Operation -per un totale di 40 cicli di clock) a valle dell'istruzione STA per evitare sovrapposizione tra interrupt proveniente dalla rete di Address Decoding e scrittura sulla porta seriale.

Nell'ultimo paragrafo questi tempi verranno illustrati con maggiore dettaglio.

Nel caso di “velocizzazione” di programmi in linguaggio macchina esistenti, o nuovi, fate dunque attenzione ad eventuali malfunzionamenti nella scrittura a schermo.

Se avete problemi di spazio per allocare questi ritardi considerate anche l'uso delle istruzioni PHA (Push Accumulator) e PLA (Pull Accumulator) che in coppia usano 7 cicli di clock a valle di due soli bytes occupati in memoria.

## 6. CONTROLLO DEL CURSORE – CODICI DI ESCAPE ANSI

L'ultima limitazione che la scheda terminale permette di superare è il controllo del cursore, ovvero la **capacità di scrivere in una determinata posizione dello schermo**.

Il videoterminale nativo è più simile a una telescrivente che a un terminale vero e proprio: l'unica differenza è che non usa la carta ma per il resto è equivalente: ciò che è stato scritto non può essere cancellato dallo schermo, il cursore va solo in avanti e non c'è una funzione/chiamata per la cancellazione dello schermo dato che essa viene fatta a livello hardware.

I codici di escape ANSI (o ANSI escape codes) sono delle sequenze predefinite e ben codificate di codici ASCII che, una volta interpretate dal terminale, consentono di **manipolare scritte e cancellazioni, il tipo di carattere, il colore e la posizione** di ciò che viene scritto.

Il nome "escape" deriva proprio dal codice ASCII del tasto ESCAPE, il cui codice ASCII è 27 in decimale. Tutte le sequenze di codici escape iniziano infatti con 27 decimale o \$1B esadecimale.

Il vostro programma terminale deve essere reso compatibile con tale standard nel caso non lo sia.

Controllate dunque nelle predisposizioni la rispondenza allo standard ANSI (o la compatibilità dell'emulazione con lo standard VT-100). I programmi: "minicom" per Linux, "PuTTY" e "Teraterm" per Windows sono solitamente compatibili di default.

Una volta predisposto il terminale, le parti di testo trasmesso dal computer Apple che rappresentano dei codici di escape validi non verranno visualizzati dal terminale **bensi interpretati per ottenere l'effetto corrispondente**.

Ci sono decine di codici di escape, magari personalizzati su famiglie specifiche di terminali ed esiste molta documentazione a riguardo. Questo può essere un punto di partenza per approfondire l'argomento:

<https://www2.ccs.neu.edu/research/gpc/VonaUtils/vona/terminal/vtansi.htm>

Tramite l'opportuna sequenza di codici il firmware del microcontrollore riuscirà pertanto a **riprodurre anche la funzione di Clear Screen**, replicando il vero comportamento del computer.

In questo caso specifico il microcontrollore invierà due comandi (<ESC> rappresenta il carattere di escape 27):

<ESC>[2J     che corrisponde alla cancellazione dello schermo visibile  
<ESC>[H     che posiziona il cursore nella posizione HOME in alto a sinistra

Tenete presente che possono esserci differenze di comportamento tra terminale e terminale a fronte degli stessi codici escape. Il programma "minicom", ad esempio, interpreta il primo comando sia come cancellazione schermo sia come posizionamento del cursore in HOME. "PuTTY" e "Teraterm" hanno invece bisogno anche del secondo comando per posizionare il cursore in HOME.

Un altro esempio: per posizionare il cursore in una specifica *riga/colonna* è possibile usare il comando:

<ESC>[*riga*;*colonna*f

(N.B. la lettera "f" finale è parte del comando)

Il carattere di ESCAPE, su Apple-1, può essere generato e mandato alla scheda seriale mediante una semplice istruzione che scriva all'indirizzo \$DF12 il valore \$1B (Oppure che, in INTEGER BASIC, scriva il valore 27 nella locazione -8446)

Questo rende finalmente **la funzione di Clear Screen disponibile per i programmi**. In INTEGER BASIC basterà prevedere poche semplici istruzioni, richiamabili magari con una GOSUB/RETURN:

```
10 POKE -8446,27
20 PRINT "[2J";
30 POKE -8446,27
40 PRINT "[H";
50 PRINT "A CLEAR SCREEN"
```

**Anche il controllo del cursore è possibile**, naturalmente. Per scrivere sulla riga 3, colonna 5 (ipotizziamo l'uso delle variabili R=3 e C=5) le istruzioni saranno:

```
10 POKE -8446,27
20 PRINT "[";R;" ";C;"f";
30 PRINT "THERE YOU GO"
```

Notate che:

- le istruzioni PRINT relative agli escape code si chiudono con il ";" per evitare che la PRINT "vera" vada a capo,
- l'ultima lettera dell'escape code nell'esempio del controllo del cursore è una "f" minuscola. Essa dovrà pertanto essere scritta usando l'opportuna combinazione di CONTROL-I e CONTROL-O.

La possibilità di usare codici escape apre molte strade a coloro che desiderano dedicarsi alla scrittura di programmi e giochi evoluti che vadano oltre le normali capacità del videoterminale nativo.

Con il terminale seriale cade anche il vincolo dello schermo 40x23. Potrete infatti ridimensionare lo schermo e usarlo delle dimensioni che preferite per i vostri programmi!

## 7. CARATTERISTICHE AC

Questa piccola carrellata di immagini e cifre ha lo scopo di quantificare i tempi necessari alla scheda per elaborare le informazioni ricevute.

I punti di misura utilizzati sono:

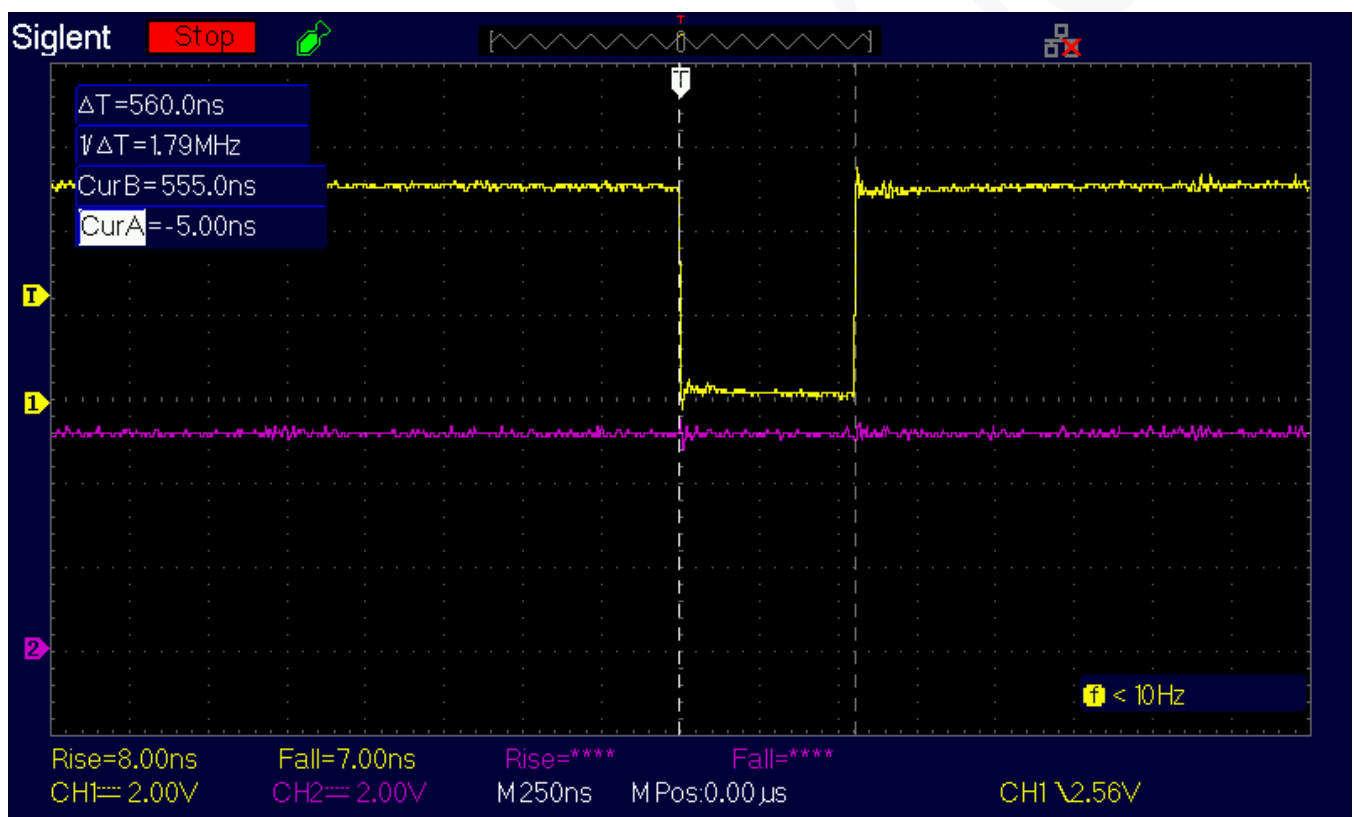
CANALE 1 – TRACCIA GIALLA, segnale di interrupt proveniente dalla logica di Address Decoding, Arduino pin “D2”

CANALE 2 – TRACCIA VIOLA, trasmissione dati seriali, Arduino PIN “TX”

Le specifiche del microcontrollore ATMEL 328P dichiarano che il tempo minimo per rilevare un interrupt sono 4 cicli di clock. A 16 MHz, ossia la frequenza del clock del microcontrollore, ciò equivale a **250 ns**.

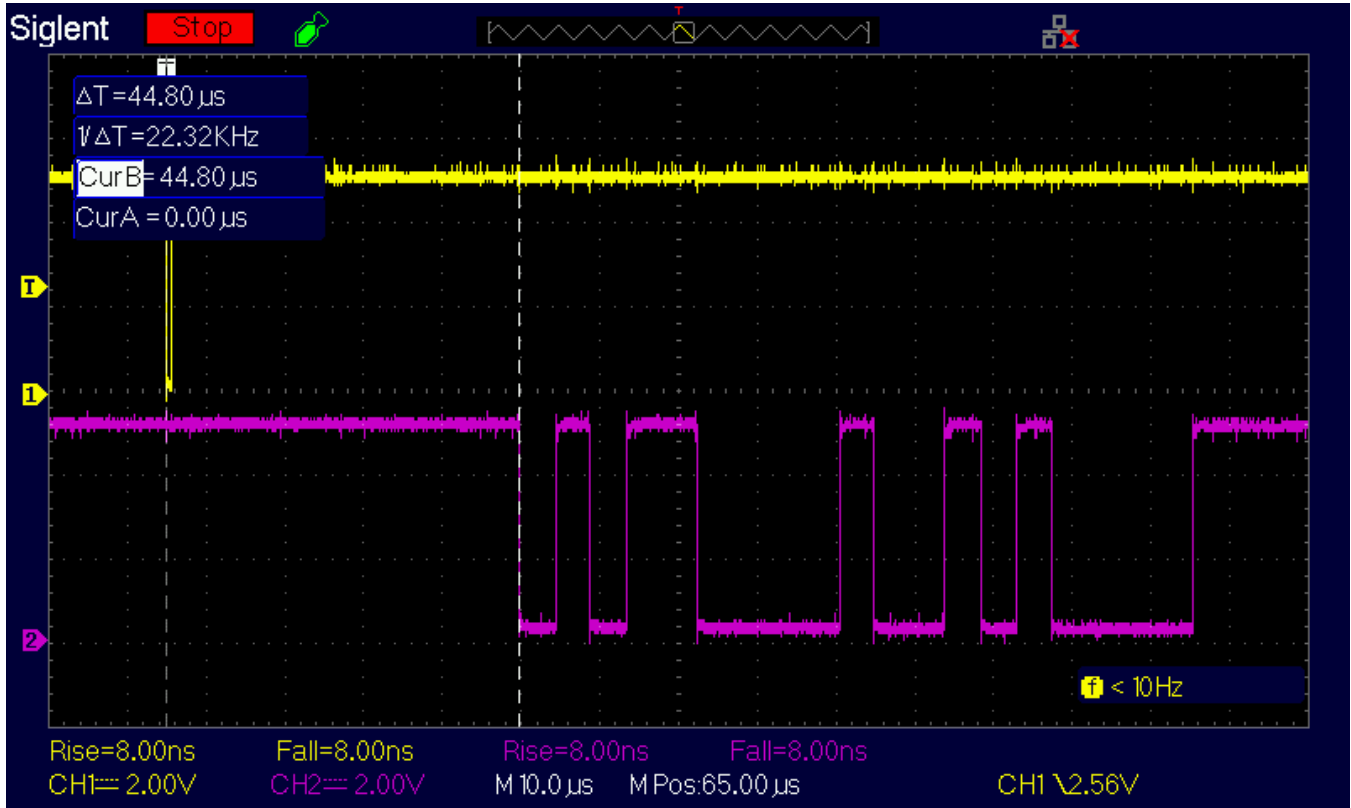
*Nelle immagini seguenti le linee tratteggiate verticali rappresentano l'intervallo di tempo misurato.*

Segnale di interrupt proveniente dalla logica di Address Decoding: **560 ns** (maggiore della soglia minima di 250ns)



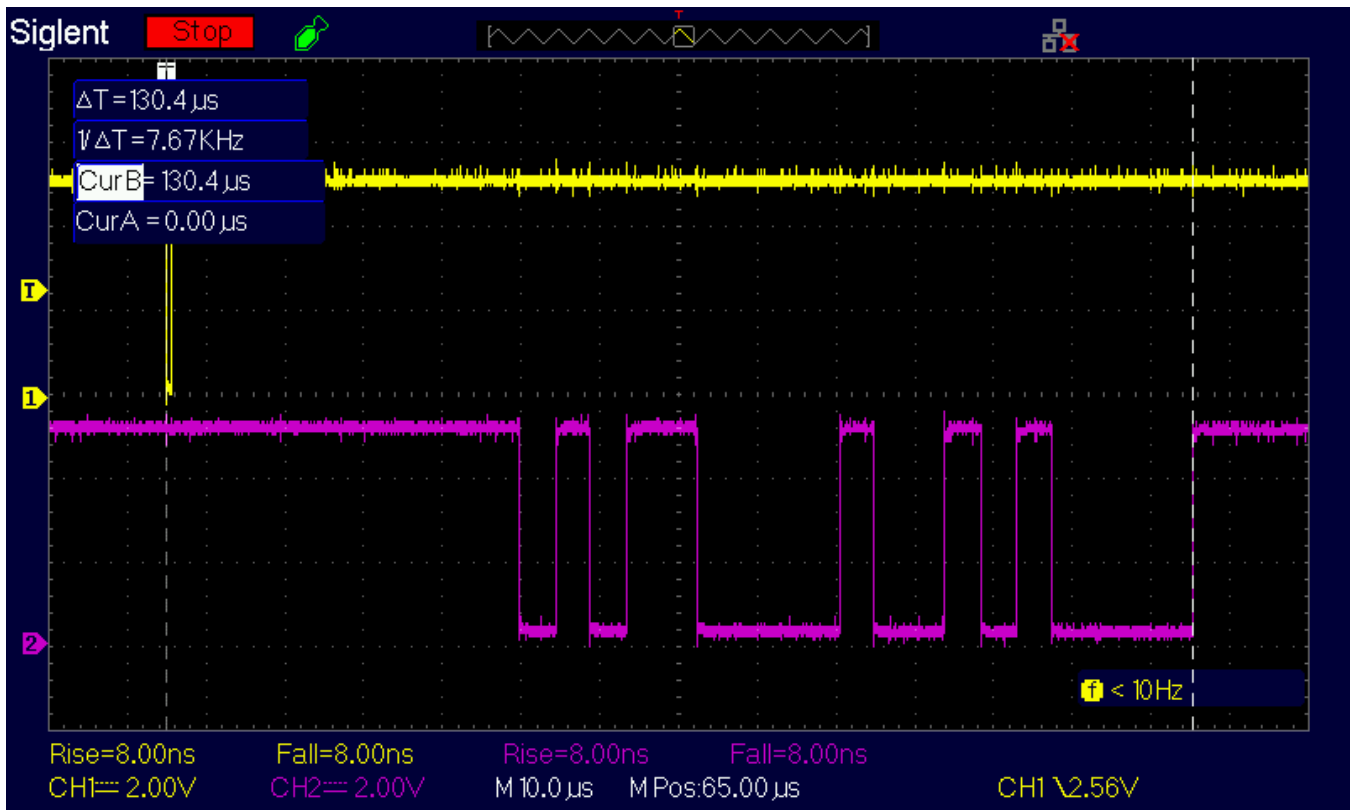


Tempo di elaborazione del microcontrollore prima di iniziare la trasmissione seriale a 230400 baud: ~45  $\mu$ s



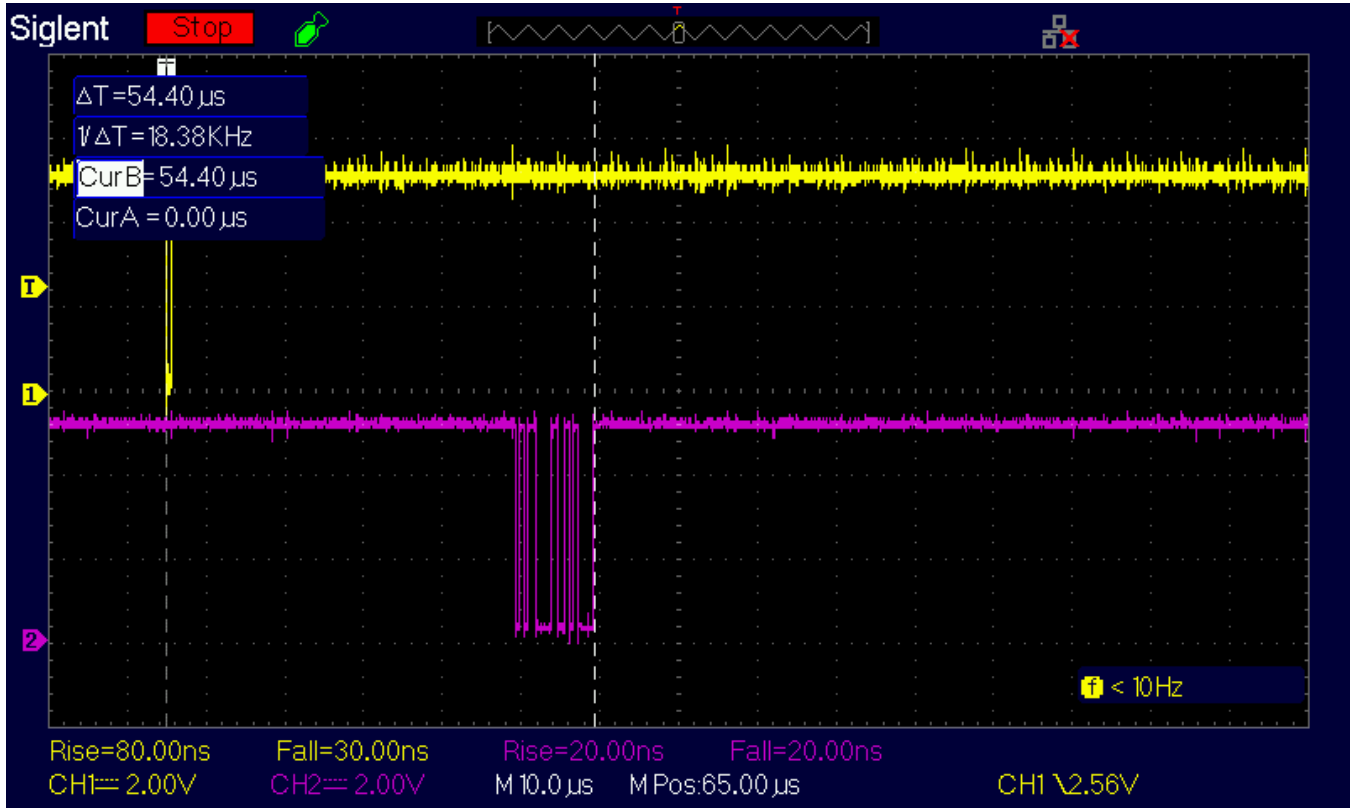
Questo è il caso peggiore in quanto riguarda la pressione del tasto ENTER, che da solo genera due caratteri sulla porta seriale: CR (13 decimale, \$0D in esadecimale) e LF (10 decimale, \$0A in esadecimale).

Tempo totale di gestione dell'interrupt sommato alla trasmissione seriale a 230400 baud: **130.4  $\mu$ s**



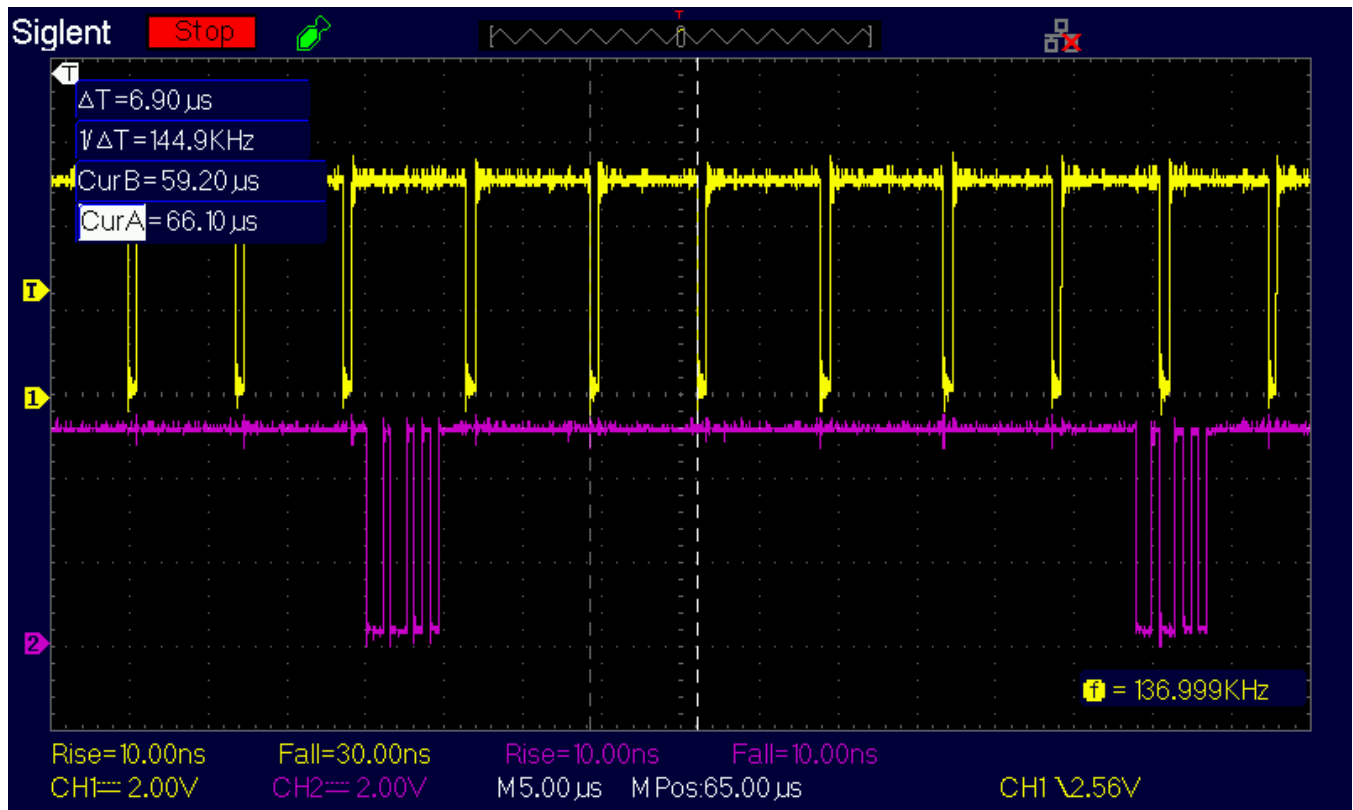
Questo è il caso peggiore in quanto riguarda la pressione del tasto ENTER, che da solo genera due caratteri sulla porta seriale: CR (13 decimale, \$0D in esadecimale) e LF (10 decimale, \$0A in esadecimale).

Tempo totale di gestione dell'interrupt sommato alla trasmissione seriale a 2000000 baud: **54.40  $\mu$ s**



Questo è il caso peggiore in quanto riguarda la pressione del tasto ENTER, che da solo genera due caratteri sulla porta seriale: CR (13 decimale, \$0D in esadecimale) e LF (10 decimale, \$0A in esadecimale).

Overlap tra segnali di interrupt e trasmissione seriale a 2000000 Baud ovvero perdita di caratteri.



Ci auguriamo che tu possa divertirti ad utilizzare *Apple-1 Terminal!*

**APPLE-1 TERMINAL**

INFO | ORDINI | SUPPORTO: [p-l4b@protonmail.com](mailto:p-l4b@protonmail.com)

**P-L4B @ PROTONMAIL.COM**