



HACKING E REVERSE ENGINEERING DI UNA MACCHINA CNC

Cricut, FREEcut! Chi sono / cosa faccio

- Federico Braghioli (federico.braghioli@gmail.com)
- Software engineer sistemi embedded
- Sviluppo firmware su sistemi ARM (cortex M3/M4) e AVR
- Felice utente Linux
- PCOfficina
 - <http://www.pcofficina.org>
 - Siamo qua al Linux Day 2018
 - Ogni martedì sera, dalle 20.00 alle 22.30 in via Pimentel 5 a Milano
 - pcofficina@gmail.com

Cricut, FREEcut! Il progetto / perchè

- Reverse engineering di una macchina CNC a livello hobbistico
- Opporsi all'obsolescenza programmata
- Liberare le capacità dell'hardware
- Perchè è divertente! (Il piacere del sano hacking)
- Questo progetto si ispira al lavoro svolto da Matt Williams
<https://github.com/seishuku/TeensyCNC>

Cricut, FREEcut!

Il progetto / argomenti trattati

- Presentazione del prodotto
- Termine del supporto
- Punti di attacco
- Analisi, studio e hacking hardware
- Obiettivi firmware di controllo
- Stm32 bluepill board
- Sistema operativo utilizzato sulla board (NuttX)
- Implementazione firmware e tecniche di controllo
- Demo
- Futuri migliorie

Cricut, FREEcut! Che cos'è?



CRICUT MINI - PROVO CRAFT

Un sistema CNC ad uso hobbistico per il taglio di vari materiali (carta, cartoncino, tessuto)



Cricut, FREEcut! Che cos'è?



UTENSILE

Contiene la lama per effettuare il taglio. È fissato al supporto ma si può rimuovere e sostituire.

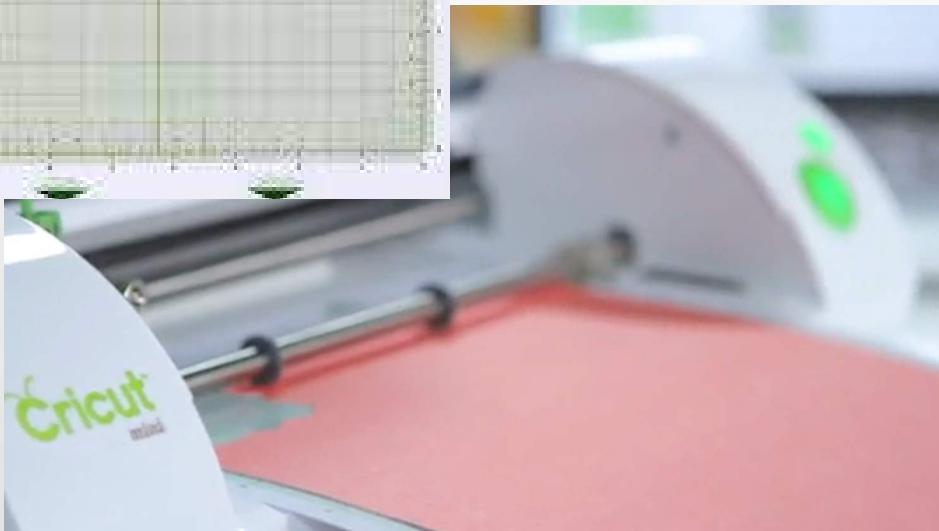


Cricut, FREEcut! Che cos'è?

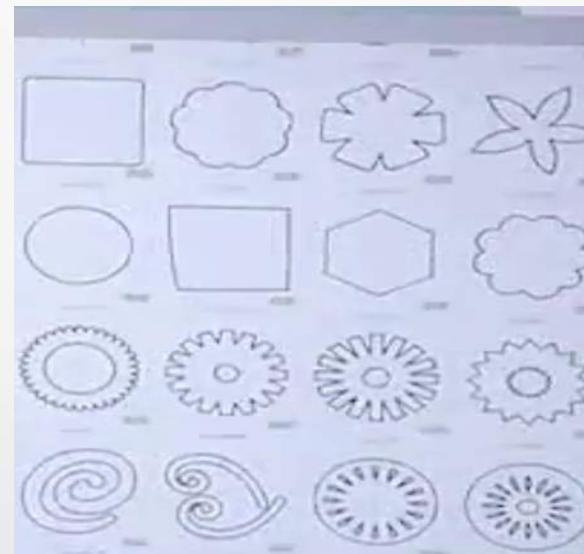
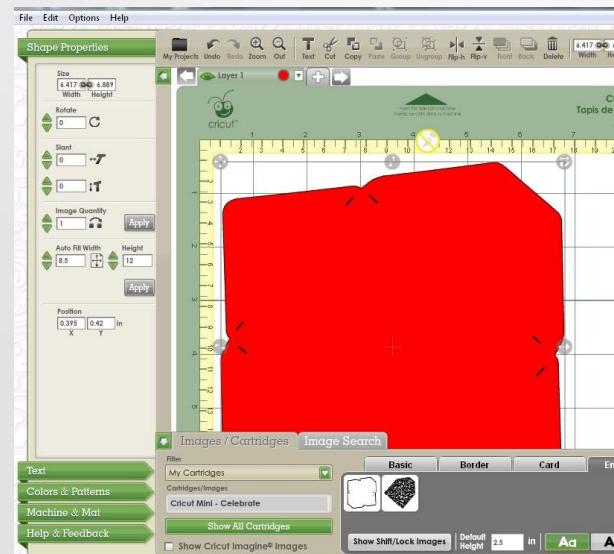


SUPPORTO

Sorregge il materiale da tagliare, lo tiene fermo grazie ad un strato attacca/stacca



Cricut, FREEcut! Che cos'è?



CRICUT CRAFT ROOM

È il software (Win o MAC).
Funziona con la “cartuccia”
che, se inserita nella macchina,
propone tramite l'interfaccia
grafica una serie di forme da
poter tagliare.



Cricut, FREEcut!

Termine del supporto

- Luglio 2018: la decisione di Provocraft di terminare il supporto
- La nota: “*As of 9am MT today, Cricut Craft Room has been officially shutdown and is no longer available for legacy machine owners to access.*” “*This means you will no longer be able to access Cricut Craft Room on your desktop if you still use a legacy machine, including Cricut Personal, Create, Expression, Expression 2, Mini, Cake, Cake Mini, and Imagine.*”
- La soluzione proposta da ProvoCraft: “*To help with this transition, we have a special offer for legacy machine owners who have not yet upgraded to our latest machines: You'll get \$75 off any Explore machine or bundle on Cricut.com.*”

Cricut, FREEcut!
Punti di attacco

Cosa fare?

Cricut, FREEcut!
Punti di attacco

Opzione 1

Cricut, FREEcut! Punti di attacco



Cricut, FREEcut!
Punti di attacco

Opzione 2

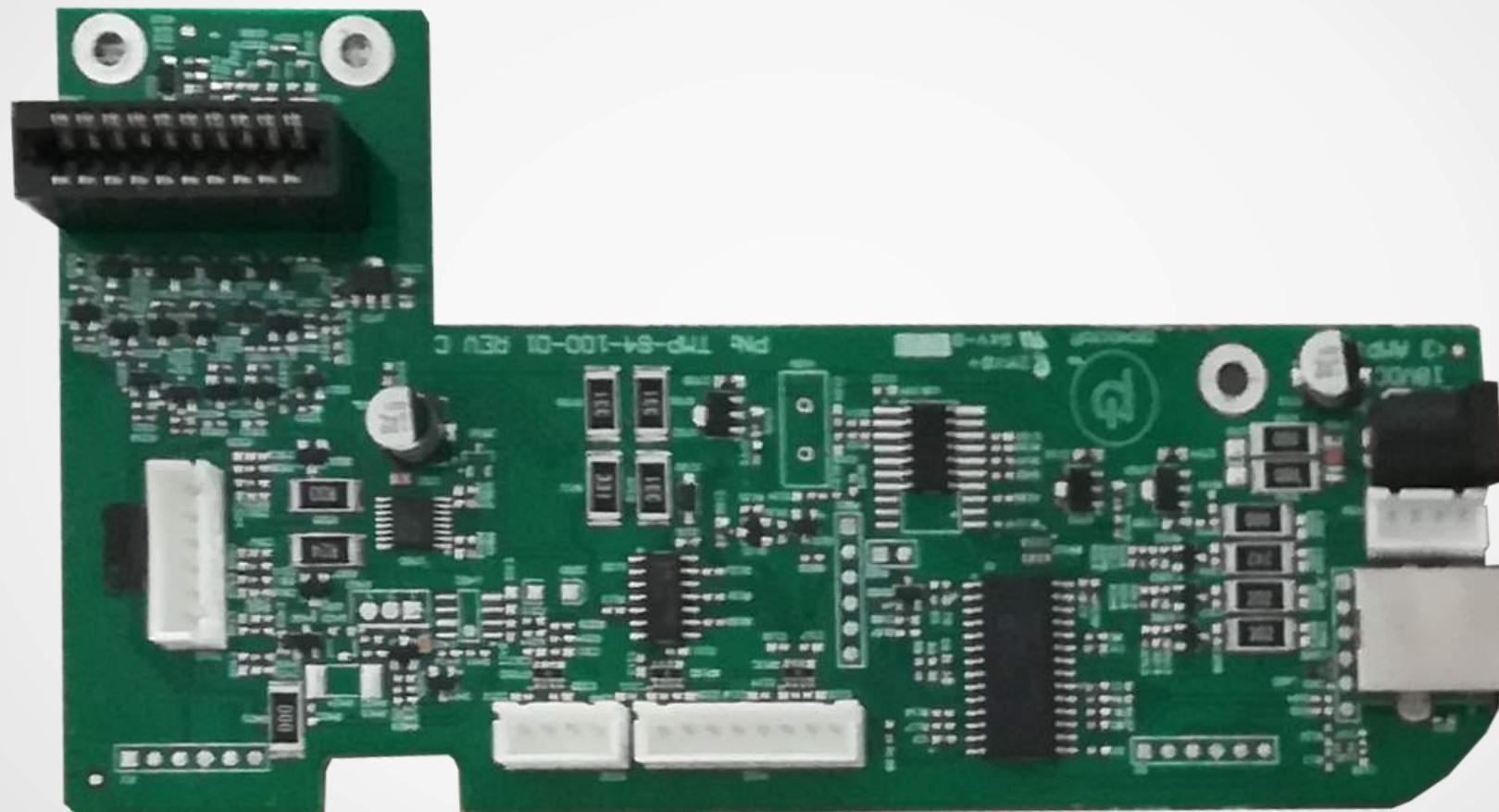
Cricut, FREEcut! Punti di attacco

- Hacking del programma di controllo Cricut Craft Room
 - Reverse engineering protocollo di comunicazione su USB
 - Server di autenticazione
- Pro:
 - Nessuna modifica hardware
 - “Garanzia” non invalidata
- Contro:
 - Windows
 - Complesso (disassembler, ...)
 - Limiti software rimangono

Cricut, FREEcut!
Punti di attacco

Opzione 3

Cricut, FREEcut! Punti di attacco



Cricut, FREEcut! Punti di attacco

- Modifica / riscrittura firmware microcontrollori
- Pro:
 - Nessuna modifica hardware
- Contro:
 - Accessori per programmare su circuiti
 - Non conosco i PIC
 - PIC sono limitati
 - profondità stack
 - PIC16LF1823 8bit, 2K Flash, 128B Sram, @ 24MHz
 - PIC24FJ64GB002 16bit, 32K Flash, 8K Sram, @32MHz

Opzione 4

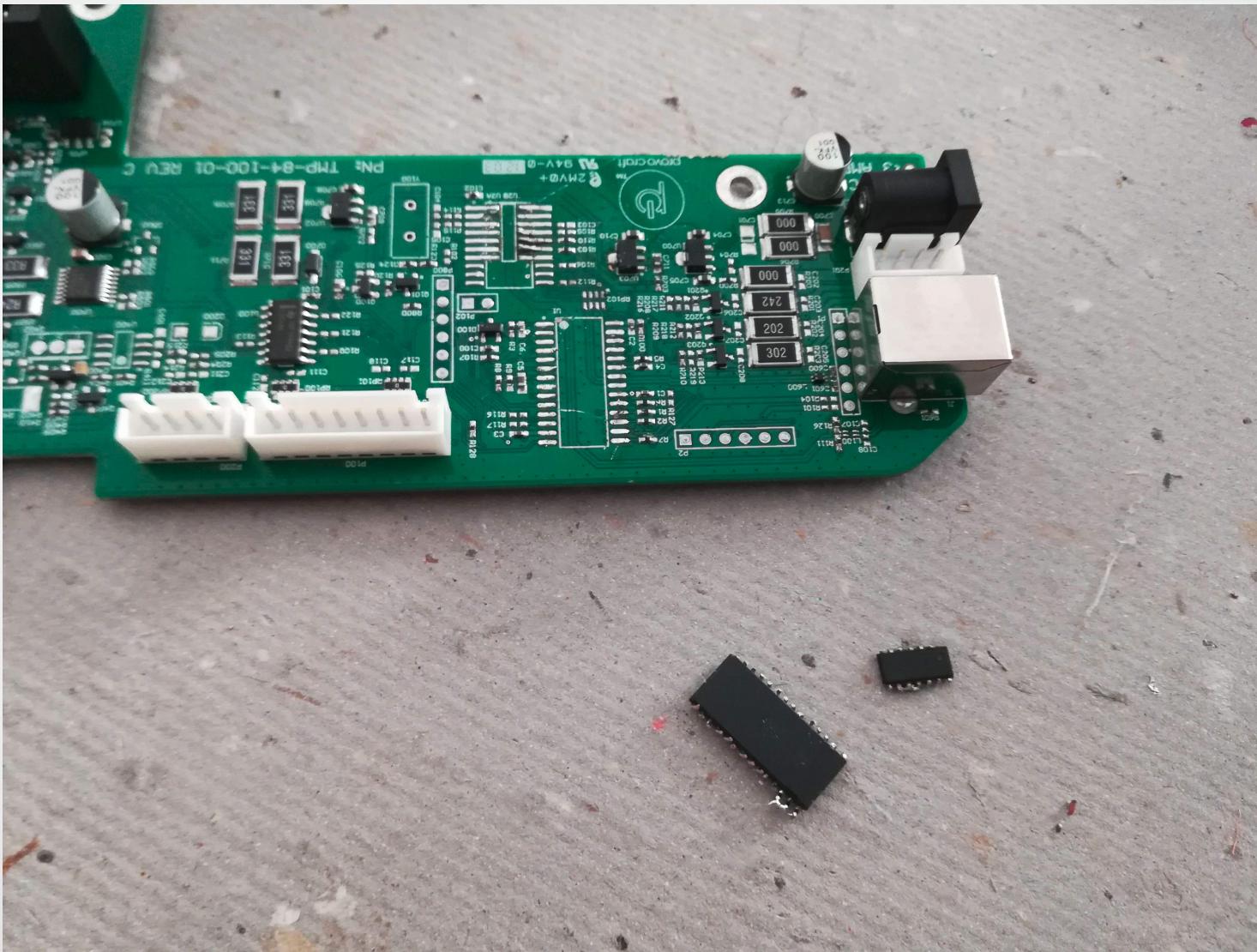
Cricut, FREEcut! Punti di attacco

- Reverse engineering dei segnali di controllo e utilizzo di una board dedicata
- Pro
 - Totale libertà di controllo
 - Piattaforma / supporto software conosciuto
- Contro
 - Manualità con strumentazione (saldatore, aspirastagno, multmetro)
 - Modifiche hardware
 - Affidabilità?

Cricut, FREEcut!
Punti di attacco

Quindi?

Cricut, FREEcut! Punti di attacco



Cricut, FREEcut!
Analisi hardware

Analisi hardware

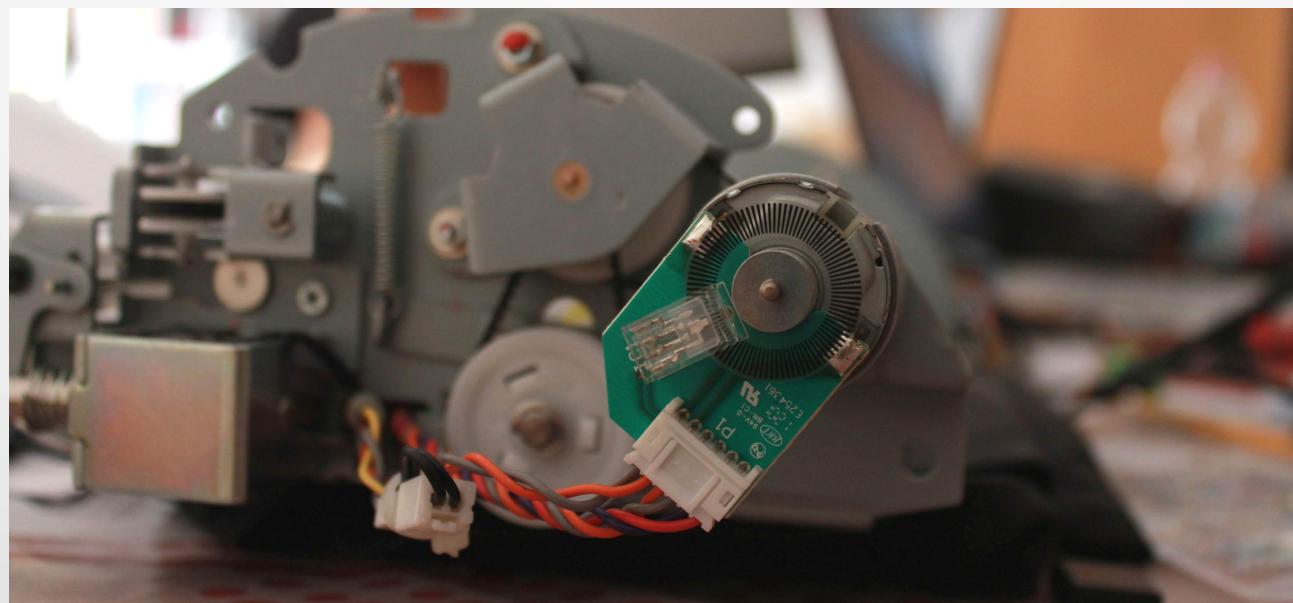
Cricut, FREEcut! Analisi hardware

- Due motori DC con relativi encoder
- Due pulsanti (power e feed)
- Solenoide per la testa di taglio
- MCU master (usb, coda comandi, controllo motori)
- MCU slave (encoder, interrupt)
- Comunicazione master/slave via SPI

Cricut, FREEcut!

Analisi hardware - Motori

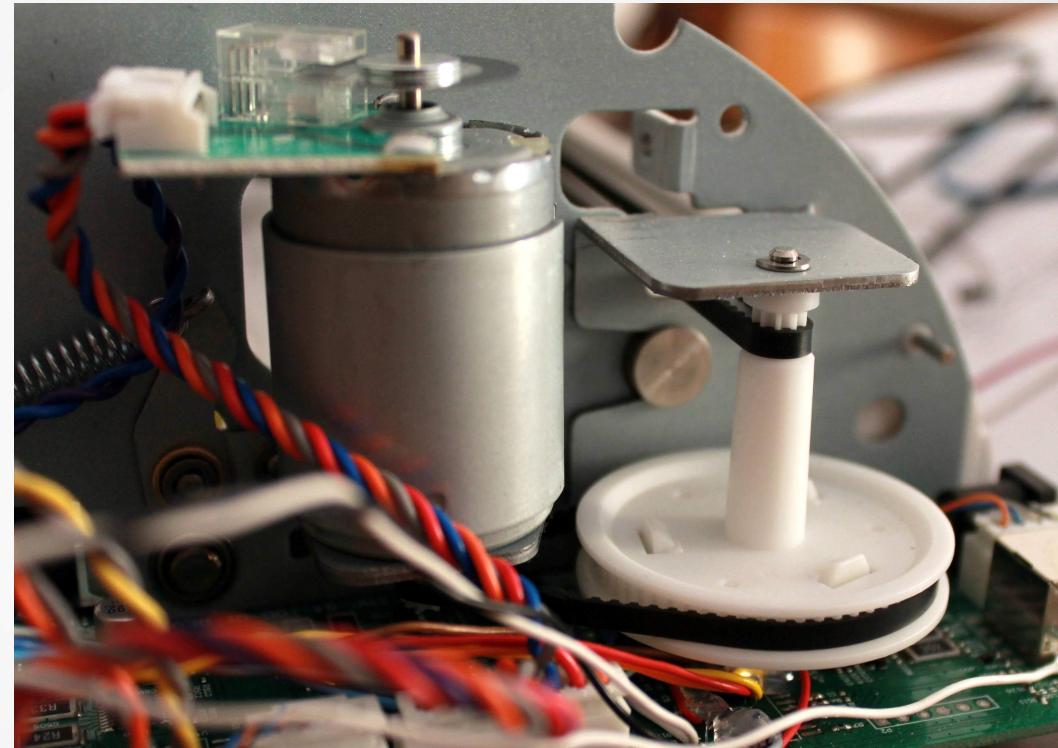
- Due motori DC 12V (asse X e asse Y)
- Controllo con PWM: modulando duty cycle si varia la velocità (la tensione media)
- Direzione rotazione dipende dalla polarità con cui i motori vengono alimentati
 - Pin a cui si applica PWM determina la direzione



Cricut, FREEcut!

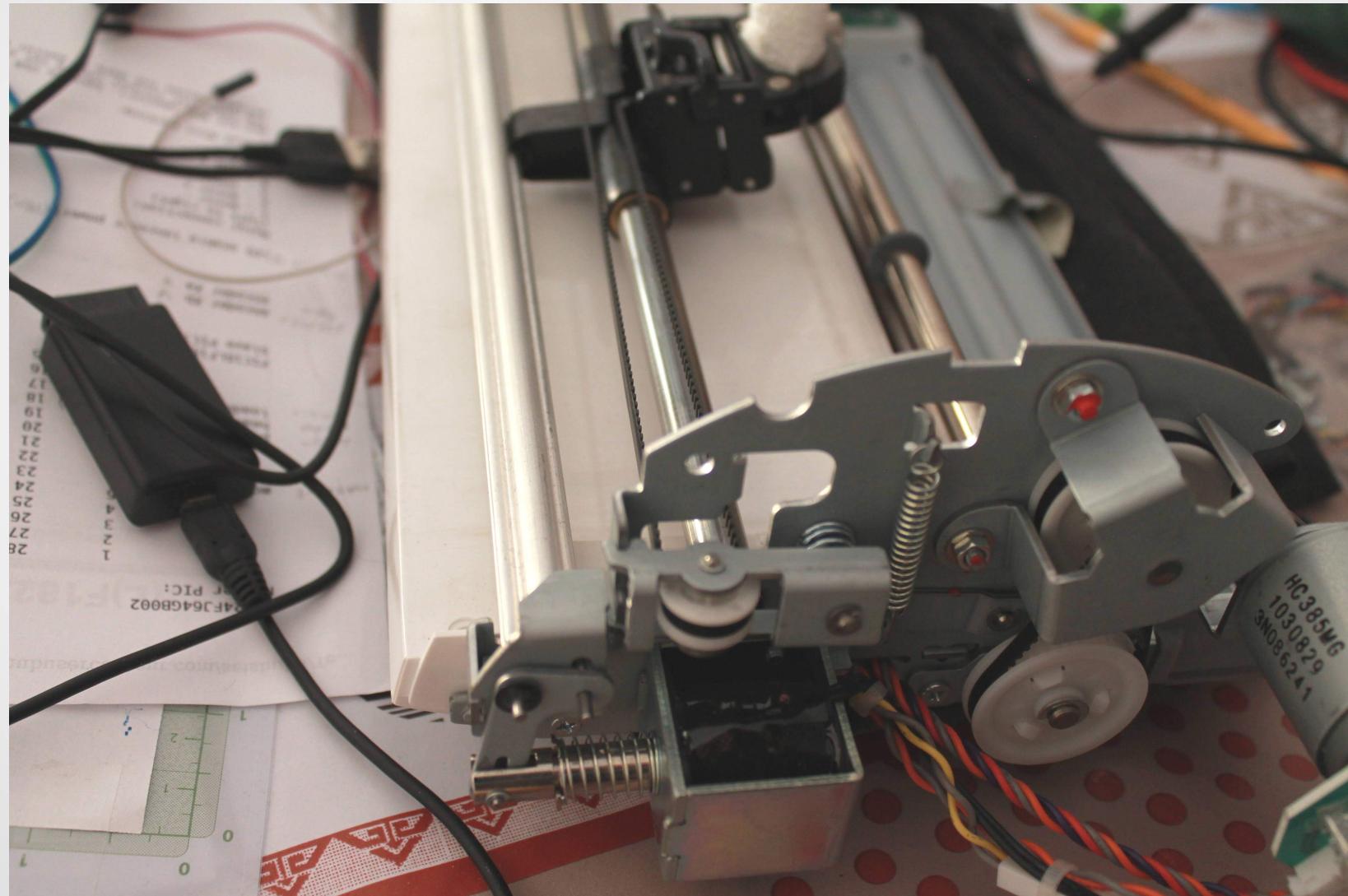
Analisi hardware - Motori

- Pinout:
 - Motore +/- (12V)
 - Motore -/+ (12V)
 - GND
 - Fotodiodo A
 - Anodo Led (+)
 - Fotodiodo B
- Pilotaggio potenza su main board



Cricut, FREEcut!

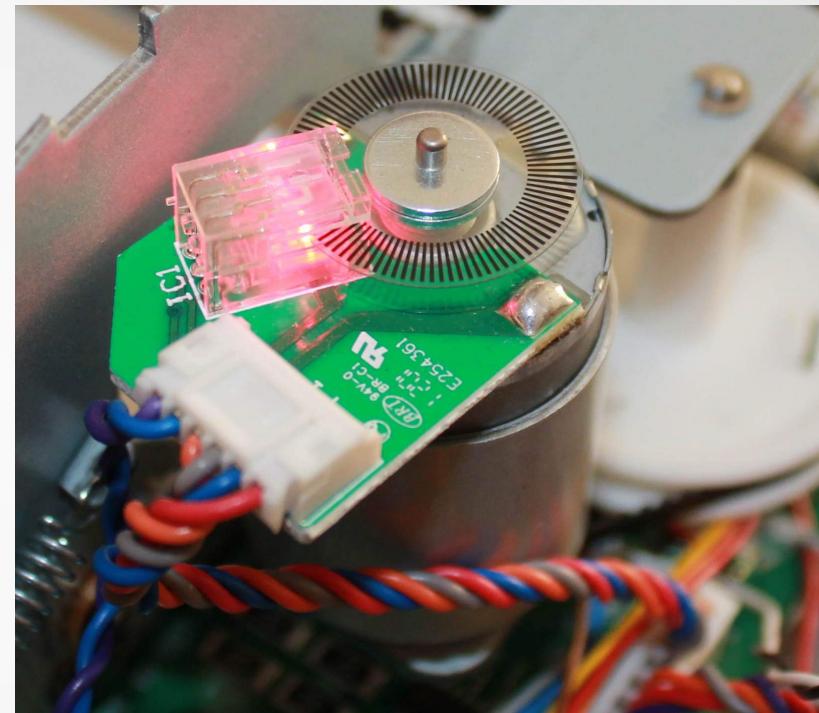
Analisi hardware - Solenoide



Cricut, FREEcut!

Analisi hardware - Encoder

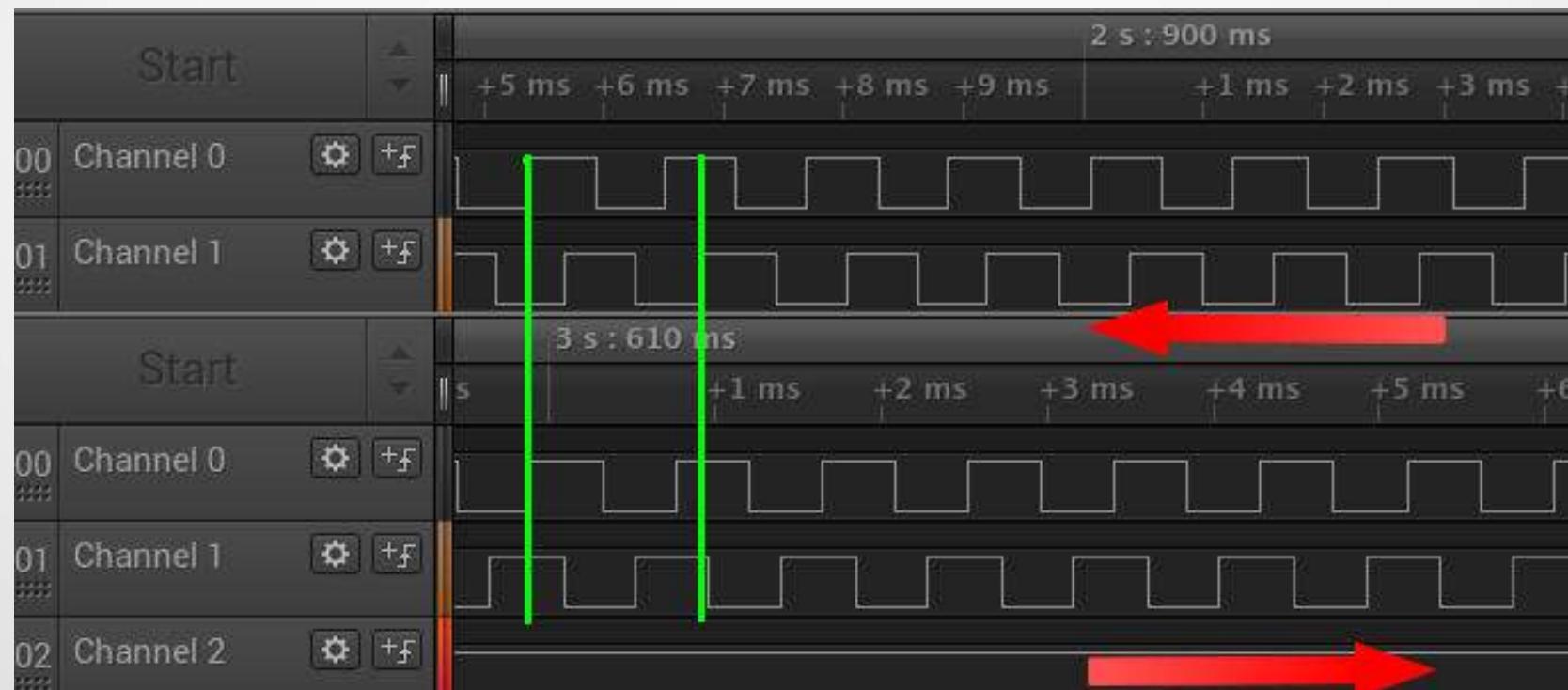
- Quadrature encoder
- Led + due fotodiodi
- Impulsi che danno informazioni dirette su
 - Posizione
 - Direzione



Cricut, FREEcut!

Analisi hardware - Encoder

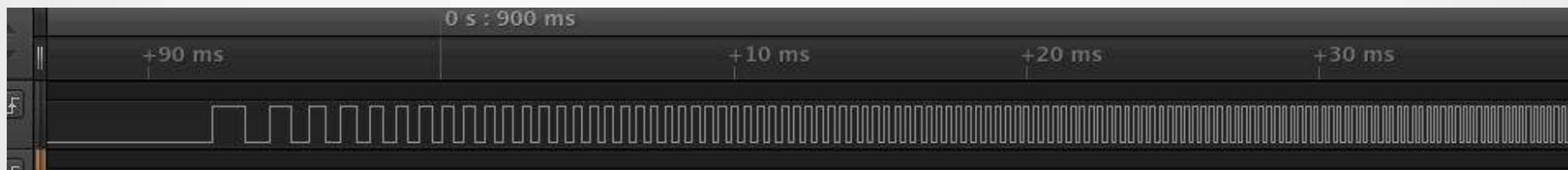
- Come funzionano
 - La fase tra i due canali stabilisce la direzione



Cricut, FREEcut!

Analisi hardware - Encoder

- Come funzionano
 - La frequenza degli impulsi determina la velocità di rotazione.
 - La variazione di quest'ultima determina l'accelerazione

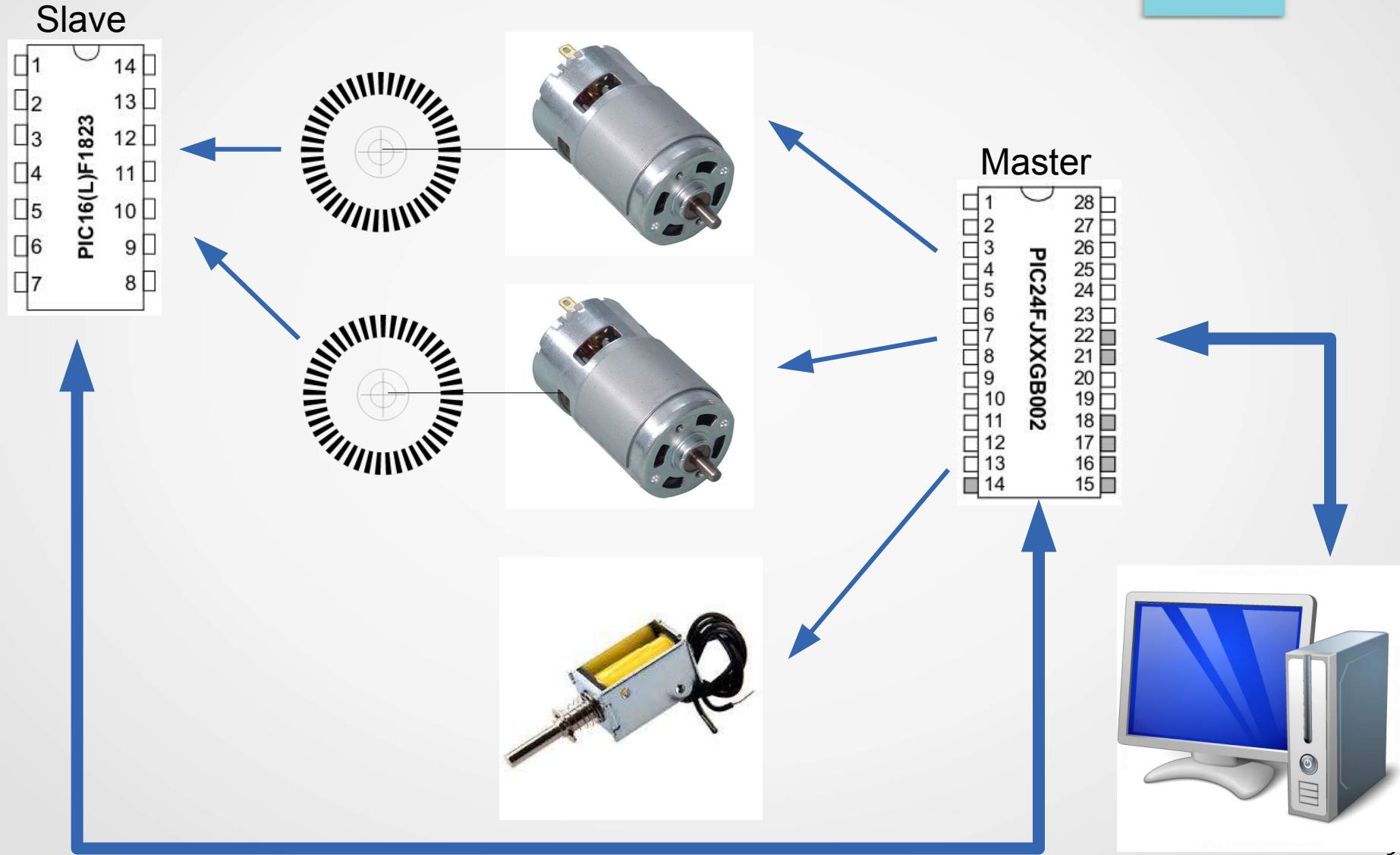


- Il numero di impulsi determina la posizione relativa

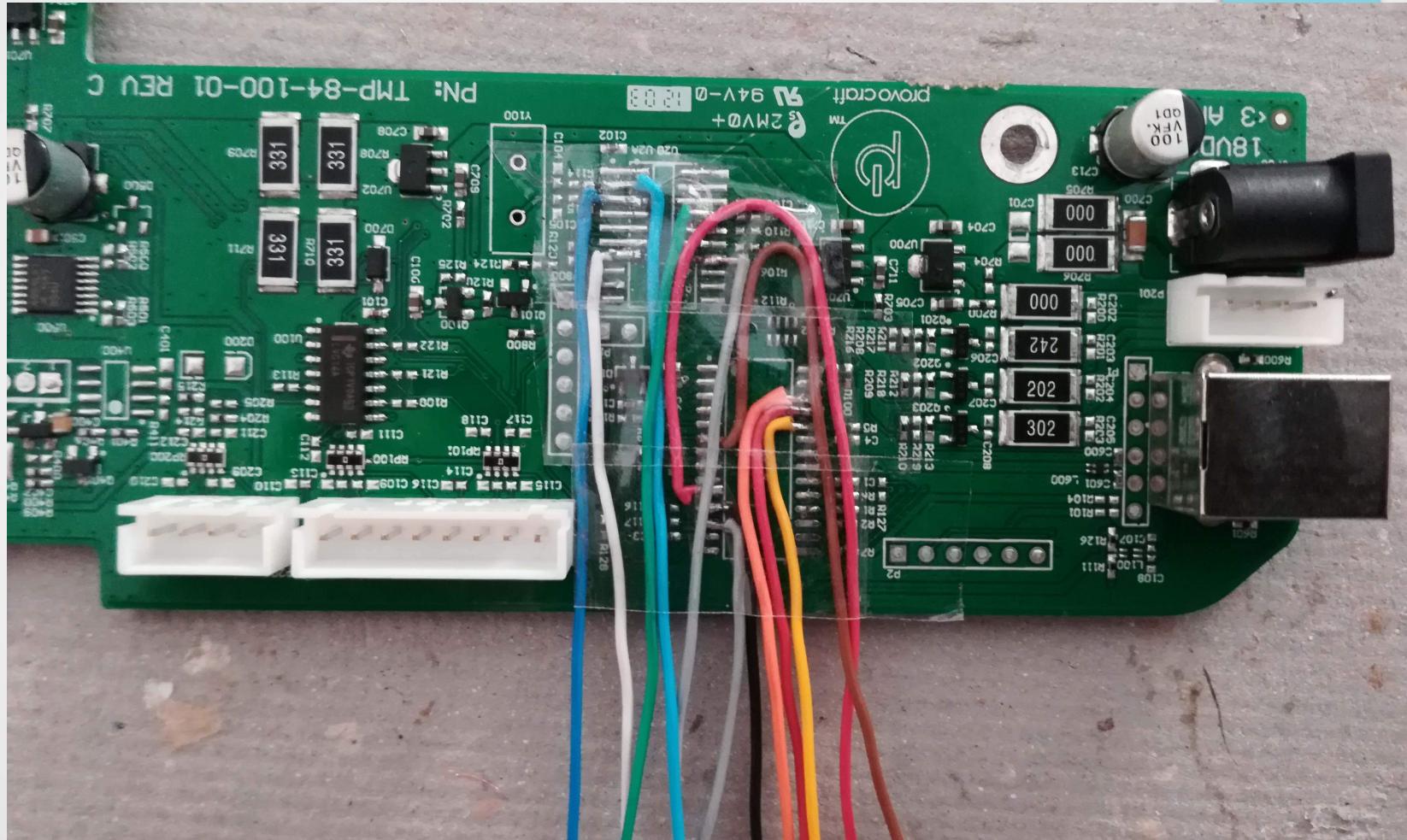
- Interfaccia segnali
 - PIC24FJ64GB002 (Master)
 - Pin 6: Motore Y, direzione A
 - Pin 10: Power button
 - Pin 11: Solenoide
 - Pin 12: Feed button
 - Pin 24: Motore X, direzione A
 - Pin 25: Motore X, direzione B
 - Pin 26: Motore Y, direzione B

- Interfaccia segnali
 - PIC16LF1823 (Slave)
 - Pin 2: Encoder X, canale B
 - Pin 3: Encoder X, canale A
 - Pin 7: Led enable
 - Pin 8: Encoder Y, canale A
 - Pin 11: Encoder Y, canale B

Cricut, FREEcut! Analisi hardware – Architettura



Cricut, FREEcut! Hacking hardware



- Attenzione a non rovinare piste

Cricut, FREEcut!

Ora?

Cricut, FREEcut! Obiettivi - specifiche

- Obiettivi
 - Ascoltare comandi di controllo standard da PC (gcode)
 - Non dover dipendere da una piattaforma o da un software specifico
 - Flessibile
 - Tradurre i comandi in spostamenti fisici
 - Disegnare su un foglio di carta (per il momento...)

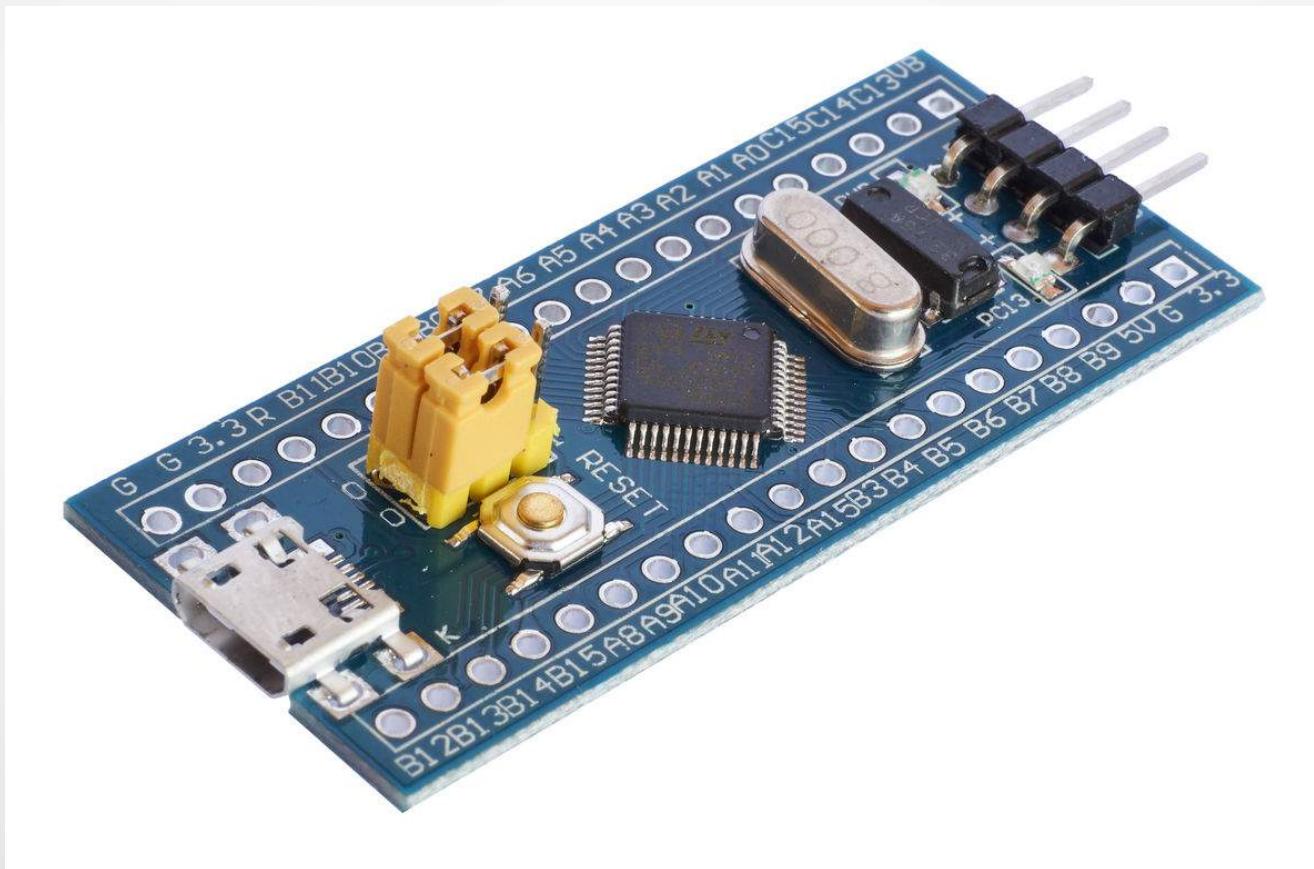
Cricut, FREEcut! Obiettivi - specifiche

- Scelta del microcontrollore
- Di cosa abbiamo bisogno?
 - 4 canali pwm
 - 4 ingressi digitali per encoder (timer?, interrupt?)
 - 2 ingressi digitali per pulsanti
 - 1 uscita digitale per solenoide
 - 1 uscita digitale per LED encoder
 - Seriale per debug
 - Seriale (usb?) per interfaccia verso pc
- Piccolo ed economico

- Cosa deve fare?
 - Catturare gli impulsi degli encoder per avere un riferimento spaziale
 - Variare il duty cycle dei pwm (velocità dei motori) in base alla “distanza” o “errore” rispetto alla posizione voluta. Quando l’errore è “zero” allora siamo arrivati a destinazione
 - Azionare la testa per disegnare

Cricut, FREEcut! STM32 BluePill

- STM32 BluePill



Cricut, FREEcut!

STM32 BluePill - Caratteristiche

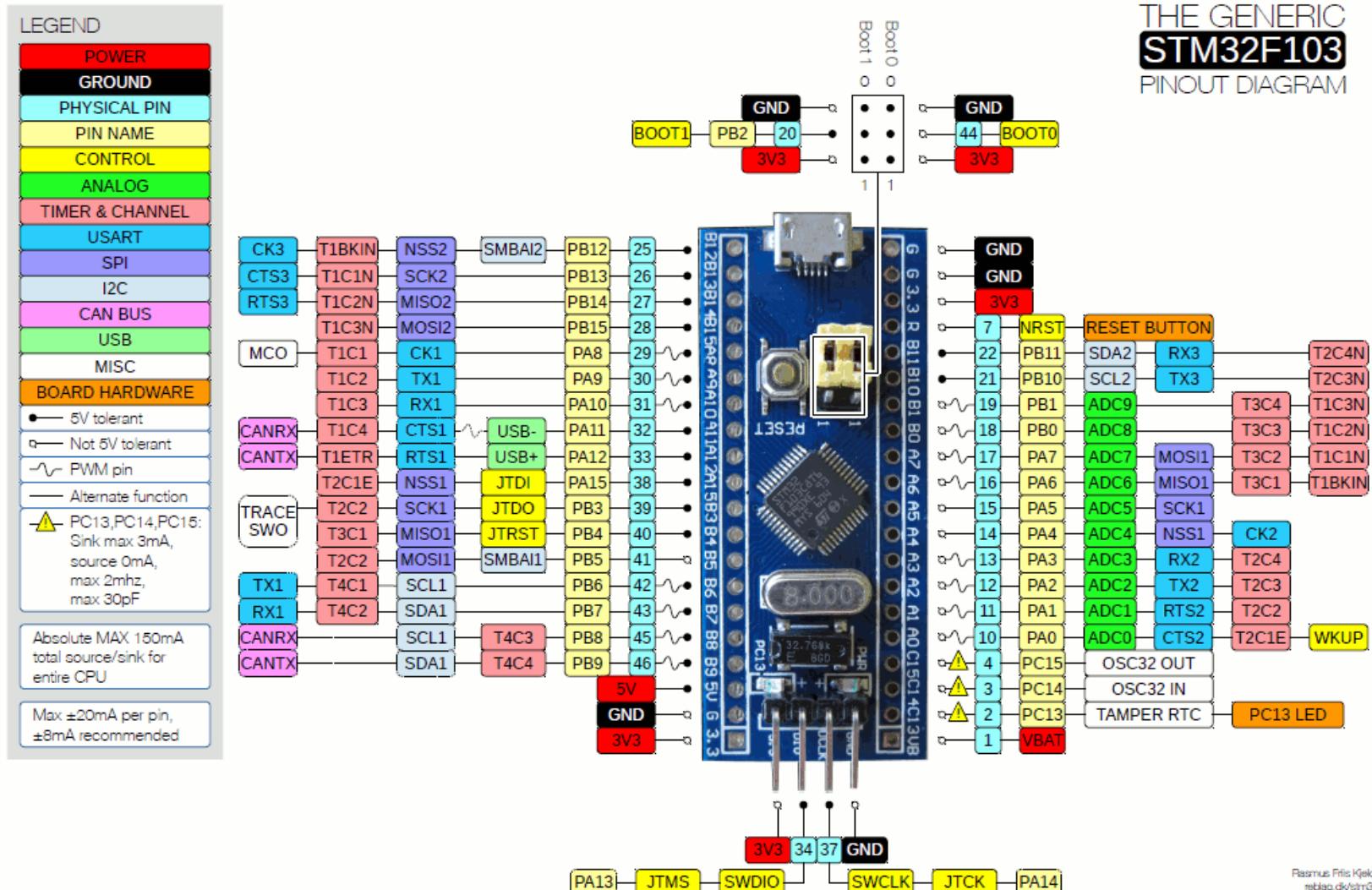
- Basata su STM32F103C8
- Cortex M3 @ 72MHz
- 128KB Flash
- 20KB Ram
- Buon punto di partenza:
https://wiki.stm32duino.com/index.php?title=Blue_Pill

Cricut, FREEcut!

STM32 BluePill – Periferiche

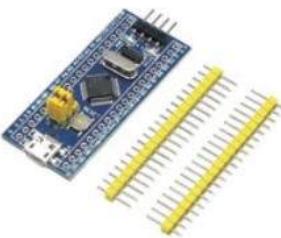
- 2x SPI
- 3x USART
- 1x USB (Device)
- 2x I2C
- 1x ADC (10 canali)
- 1x DAC
- 7x TIMER
- 1x CAN BUS

Cricut, FREEcut! STM32 BluePill – Periferiche



Cricut, FREEcut! STM32 BluePill

- Perchè questa board e non un'altra?
 - Economica ~ 2€
 - Ne possedevo una
 - Supportata da NuttX



STM32F103C8T6 BRACCIO STM32 Minimi di Sistema Scheda di Sviluppo Modulo per arduino KIT FAI DA TE

US \$1.62 / parte
Spedizione gratuita

★★★★★ Resoconto (654) | Ordini (1774)

♥ Aggiungi alla Lista dei Desideri

- Alternative?
 - Infinite (Arduino, Teensy...)

- Come programmarla
 - Embedded bootloader su seriale (adattatore USB Seriale TTL 3.3V)
 - Programmatore SWD (JTAG/serial wire debugging), ad esempio STLink

Cricut, FREEcut! STM32 BluePill - STLink

- Programmatore STLink
 - Debug
 - Breakpoint
 - Memory viewer
 - Pause, Halt, Reset CPU
 - ...
 - Economico (< 2€)
 - Ci risparmiamo un bootloader di secondo livello



ST-Link V2 **stlink** mini
STM8STM32 **STLINK** simulator
scaricare programmazione Con
La Copertura

US \$1.83 / parte
Spedizione gratuita

★★★★★ (48) | Ordini (97)

Advanced Tech

Cricut, FREEcut! STM32 BluePill – Ambiente di sviluppo

- STLink tools (<https://github.com/texane/stlink>)
 - st-flash
 - st-util (gdb server)
 - st-info
 - stlink-gui
- Per flashare:
 - st-flash --reset write nuttx.bin 0x08000000

Cricut, FREEcut! STM32 BluePill – Ambiente di sviluppo

- Eclipse con plugin
- GNU ARM Eclipse Plugin
 - direttamente dal marketplace
 - oppure <https://gnu-mcu-eclipse.github.io>

- Analizzatore di stati logici
 - L'equivalente dell'oscilloscopio per segnali digitali
 - Debug (seriale è troppo lenta in alcuni contesti)
 - Multicanale
 - Si può trovare a prezzi molto bassi (< 5€)
 - Tramite software di controllo permette:
 - Analizzare e decodificare protocolli di comunicazione
 - Statistiche su segnali digitali (frequenza, duty cycle, periodo minimo/massimo)

Cricut, FREEcut!

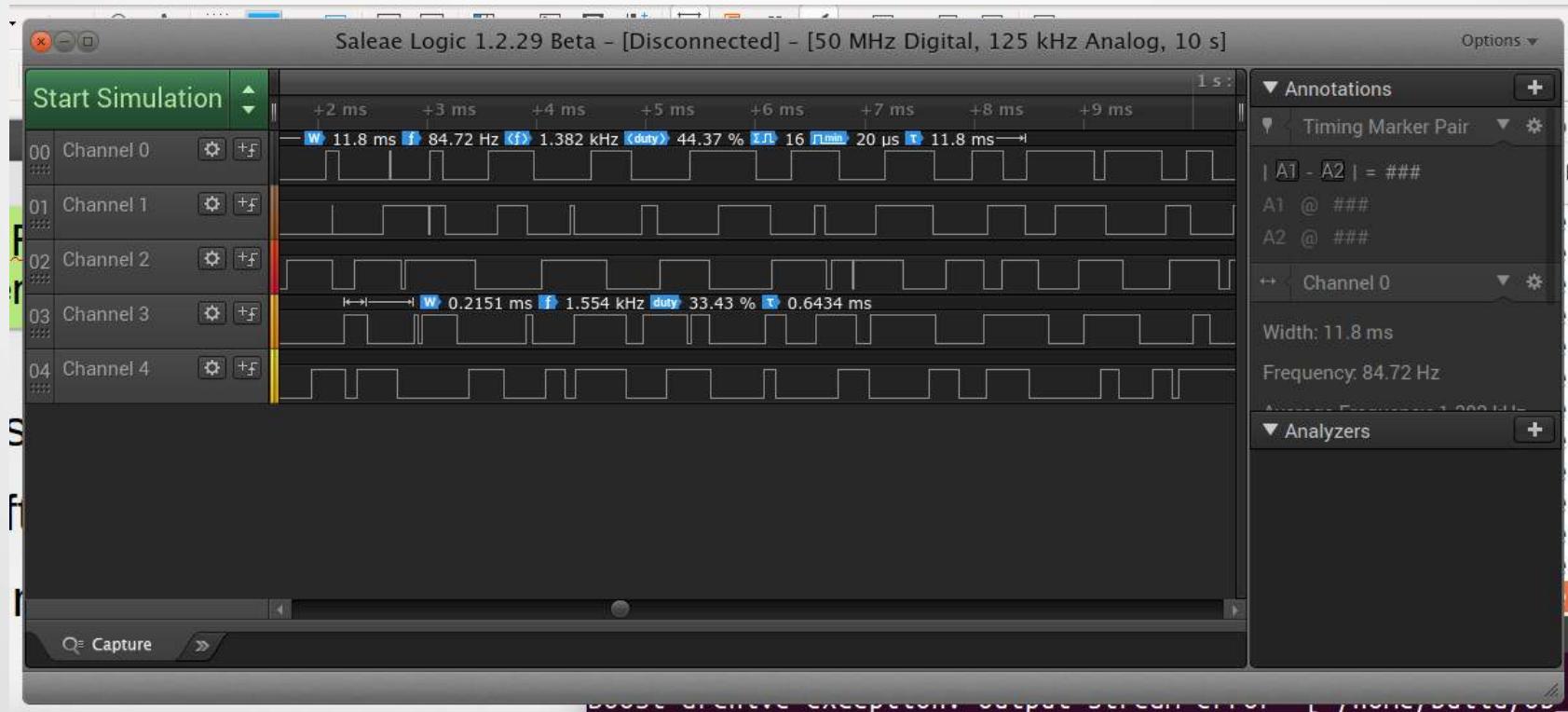
Strumenti – Analizzatore di stati logici

- 8 canali, frequenza di sampling 24MHz



Cricut, FREEcut! Strumenti – Analizzatore di stati logici

- Basato su chip Cypress FX2 (driver open fx2lafw)
- Software di controllo Saleae Logic (Win/Mac/Linux)



- Adattatore USB Seriale
- Economico (~2€ con spese)
- Basta un emulatore di terminale (e.g. minicom)
- Su Linux non sono richiesti driver aggiuntivi



E il firmware?

Cricut, FREEcut! Firmware

- Due strade:
 - Bare-metal
 - Firmware senza livelli di astrazione
 - Solo l'applicativo utente “gira” sulla cpu
 - Sistema operativo real-time
 - E' un sistema operativo!

Cricut, FREEcut! Firmware

- Bare-metal
 - Pro:
 - Prestazioni (minore overhead)
 - Minore occupazione ram / flash
 - ...
 - Contro:
 - Difficile (datasheet alla mano)
 - Parti da “zero” o quasi (no driver periferiche, a meno di utilizzare BSP)
 - Può richiedere più tempo per lo sviluppo
 - ...

Cricut, FREEcut! Firmware

- Sistema operativo real-time
 - Pro:
 - Scheduler (preemptive, cooperative,...)
 - Orientato a portabilità e riuso
 - Middleware (stack usb, ethernet, ...)
 - ...
 - Contro:
 - Può essere difficile il debug (diversi livelli di astrazione)
 - Overhead
 - ...

Cricut, FREEcut! NuttX

- NuttX RTOS (<http://nuttx.org>)
- Perchè?
 - Open source
 - Licenza BSD (basta citare l'autore)
 - Lo conosco
 - Posix (interfacce verso OS standard)
 - Ottimo supporto per STM32
 - Non ho tempo per scrivere il firmware bare-metal
 - Ben documentato
 - Progetto attivo (nuovi commit ogni giorno)



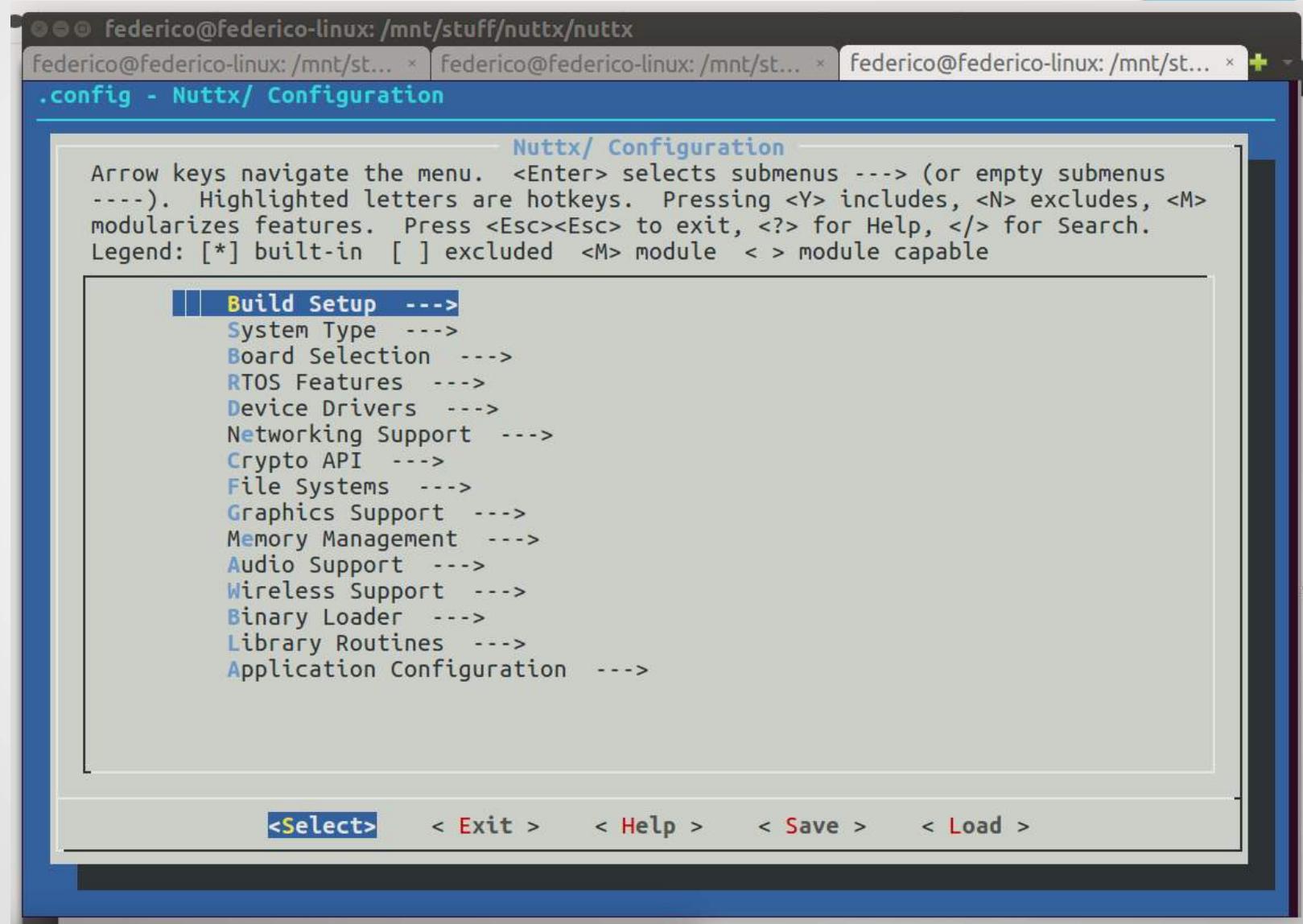
Cricut, FREEcut! NuttX

- Caratteristiche principali:
 - Segue i principali standard
 - Design modulare (driver bottom/upper half)
 - Alto grado di configurabilità (tramite kconfig)
 - Realtime
 - Interfacce POSIX/ANSI (task control, named message queue, semafori/mutex, segnali, thread, filesystem,...)
 - BSD socket
 - “Separazione” kernel / lato applicativo (dipende da configurazione memoria)
 - System logging
 - Misura carico cpu per-thread
 - Filesystems (device a blocchi, mountpoint,...)

Cricut, FREEcut! NuttX

- Driver:
 - VFS (device a caratteri e a blocchi)
 - Network
 - USB (Host/Device)
 - Dispositivi input/output (tastiere, mouse, touch, lcd, display grafici,...)
 - Fifo / pipes, loop device,...
 - Audio/Video codecs
 - Memory technology devices
 - Periferiche più comuni su MCU (adc/dac, pwm, timer, ...)
 - Networking (TCP/UDP, IP, ...)

Cricut, FREEcut! NuttX



- Alternative RTOS:
 - FreeRTOS (<https://www.freertos.org>)
 - Zephyr (<https://www.zephyrproject.org>)
 - Mbed (<https://www.mbed.com>)
 - ...

Cricut, FREEcut! Firmware – Implementazione

Dettagli implementazione

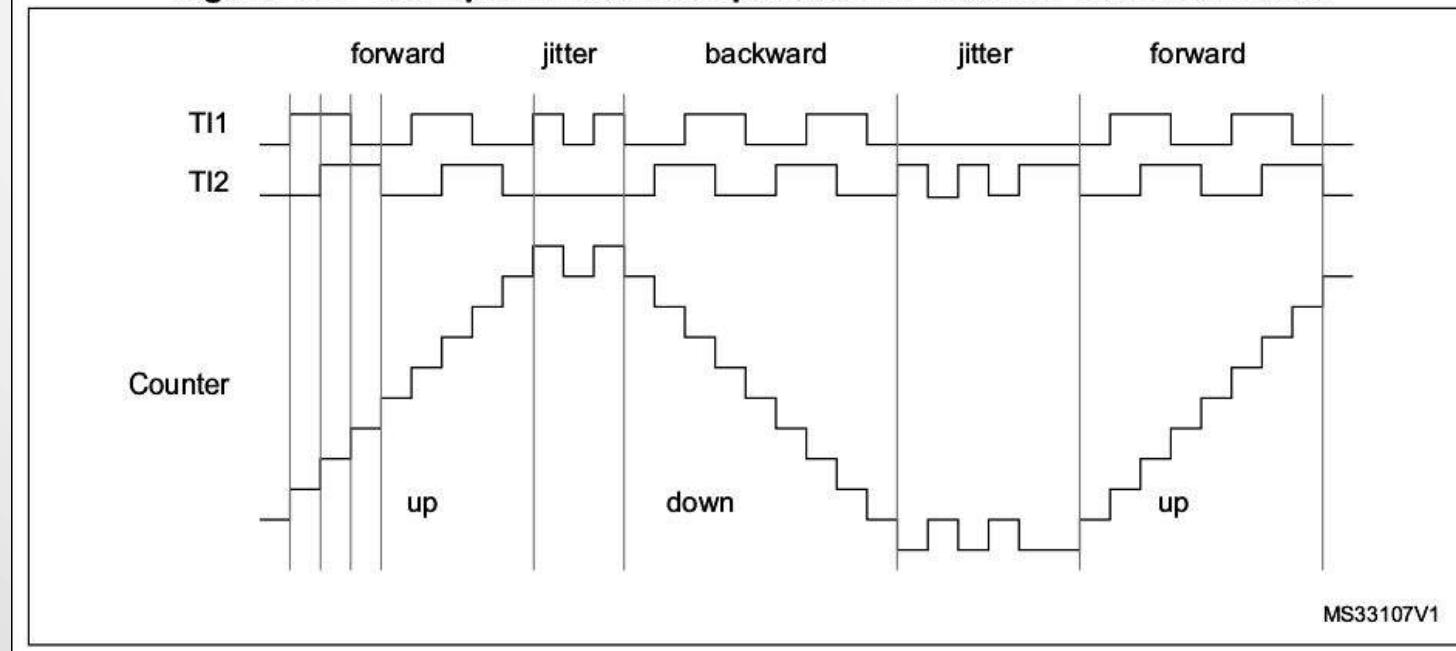
Cricut, FREEcut! Firmware – Implementazione

- Sorgenti
 - <https://bitbucket.org/fbraghioli/>
 - <https://bitbucket.org/fbraghioli/nuttx-apps/src/cricut/>
 - <https://bitbucket.org/fbraghioli/nuttx/src/cricut/>

Cricut, FREEcut! Firmware – Implementazione

- Quadrature encoder
 - Supporto in hardware: equivale ad avere un timer con un clock esterno (encoder) più un registro con l'informazione sulla direzione

Figure 134. Example of counter operation in encoder interface mode



Cricut, FREEcut! Firmware – Implementazione

- Quadrature encoder
 - Non è necessario andare in polling o appoggiarsi ad interrupt per tenere aggiornata la posizione
 - Supporto NuttX presente
 - Per conoscere la direzione e posizione corrente basta fare una ioctl() al driver (QEIOC_POSITION)
 - Funziona
 - Encoder Y assegnato al Timer 2 (PA0, PA1)
 - Encoder X assegnato al Timer 3 (PA7, PA6)

Cricut, FREEcut! Firmware – Implementazione

- PWM Multichannel
 - 4 segnali pwm: X (destra, sinistra), Y (avanti, indietro)
 - 2 saranno utilizzabili contemporaneamente (X e Y)
 - Utilizzo di soli due timer ognuno con la sua frequenza
 - Ogni timer ha un registro prescaler, autoreload e counter
 - Al canale di ogni timer associo la direzione:
 - Ogni canale ha un compare register proprio
 - Ciascun canale può generare duty cycle in maniera indipendente (ma con la stessa frequenza del timer)

Cricut, FREEcut! Firmware – Implementazione

- PWM Multichannel
 - Supporto NuttX presente
 - Per impostare il duty cycle basta una ioctl() al driver (PWMIOC_SETCHARACTERISTICS)
 - Essendo timer a 16bit, il duty cycle può variare da 0 a 65535
 - Controllo motore X assegnato al timer 1
 - Canale 1 (PA8) → destra
 - Canale 2 (PA9) → sinistra
 - Controllo motore Y assegnato al timer 4
 - Canale 1 (PB6) → avanti
 - Canale 2 (PB7) → indietro

Cricut, FREEcut! Firmware – Implementazione

- Controllo motori
- Quale frequenza scegliere?
 - Si potrebbe utilizzare una frequenza alta per evitare che sia udibile ($> 20\text{kHz}$)
 - Con frequenze di quell'ordine il motore della cricut non si muove (non conosciamo le caratteristiche elettriche del motore)
 - Dopo vari esperimenti, $f = 500\text{Hz}$
 - Si riesce ad avere un rapporto duty cycle proporzionale alla velocità del motore (analizzatore stati logici + set frequenza e duty manualmente)
- Da fermo il motore richiede un duty minimo per partire (~25%)
 - Attrito

Cricut, FREEcut! Firmware – Implementazione

- Calibrazione asse X
- L'idea è quella di muovere la testa verso sinistra finchè non arriva a fine corsa
- A quel punto azzero il contatore del timer legato all'encoder dell'asse X ottenendo il punto 0
- Problema: Cricut non ha il fine corsa
- Soluzione:
 - muovo lentamente la testa a sinistra
 - calcolo periodicamente la velocità
 - quando la velocità è zero sono arrivato a fine corsa
 - (in sostanza faccio schiantare la testa in modo controllato)
- Quanto lentamente?
 - Duty: 30% (~5cm/s)

Cricut, FREEcut! Firmware – Implementazione

- Implementazione del task di controllo in un driver (cmini_cnc)
 - Dispositivo a caratteri (open / write / ioctl, ...)
 - Interfaccia verso user-space con ioctl:
 - CNCIOC_ENABLE
 - CNCIOC_DISABLE
 - CNCIOC_GETSTATUS
 - CNCIOC_SETRAWPOS
 - CNCIOC_SETWAITRAWPOS
 - CNCIOC_STARTCAL
 - CNCIOC_STARTDRAW
 - CNCIOC_STOPDRAW

Cricut, FREEcut! Firmware – Implementazione

- Thread sincronizzato da timer con periodo a 1ms:
 - Legge posizione X, Y
 - Calcola velocità X, Y
 - Controlla che la testa non sia troppo vicino al bordo
 - Esegue controllo PID
 - Errore: differenza tra posizione target e posizione corrente
 - Variabile di controllo (uscita): duty cycle
 - Se la posizione corrente è “vicina” al target, ho fatto “hit”

Cricut, FREEcut! Firmware – Implementazione

- Tramite semafori, segnalo in spazio utente “l’arrivo” alla posizione richiesta
- In spazio utente tramite la ioctl()
CNCIOC_SETWAITRAWPOS posso impostare un target e rimanere “bloccato” finché il driver non mi segnala l’arrivo alla posizione target
- In questo modo l’applicazione potrà tenersi una coda di posizioni da inviare sequenzialmente
- Per ora l’applicazione in spazio utente calcola e invia le coordinate per disegnare un cerchio (come demo)

Cricut, FREEcut! Firmware – TODO

- TODO (software):
 - Migliorare il loop di controllo (risoluzione, velocità)
 - Interfaccia seriale su USB
 - Supporto a comandi Gcode
 - Standard per macchine a controllo numerico
 - Stampanti 3D
 - Multipiattaforma
- TODO (hardware)
 - Utilizzare la porta usb saldata sulla main board
 - Alimentare STM32 dalla main board
 - Richiuderla