PIETRO LOREFICE
Embedded Developer

Firmware debugging

Develer TPS

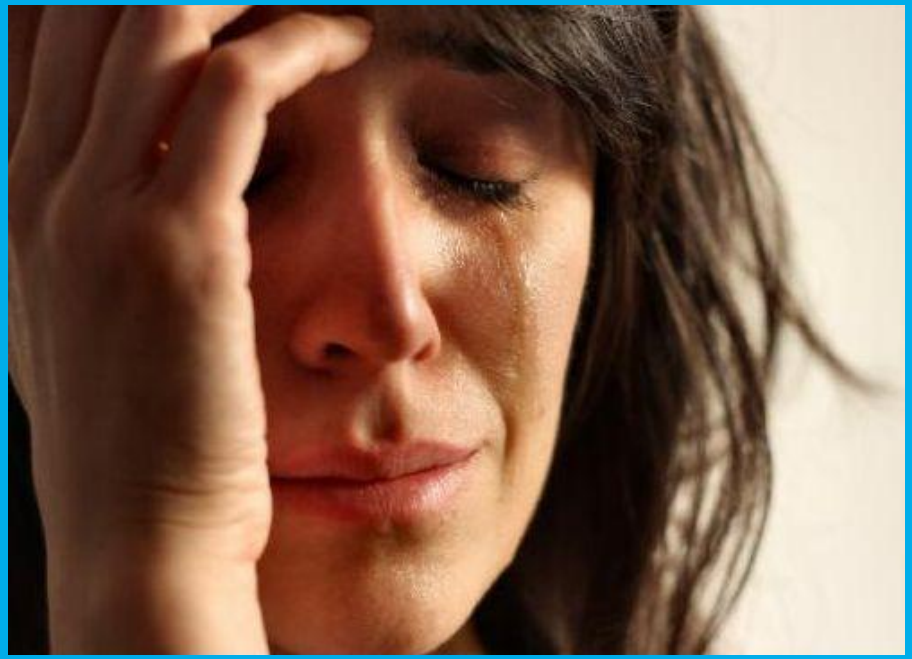
30/09/2020

develer

“DEBUGGARE IL SOFTWARE E’ DIFFICILE”

“Debuggare il firmware
è peggio.”

— *Qualcuno*



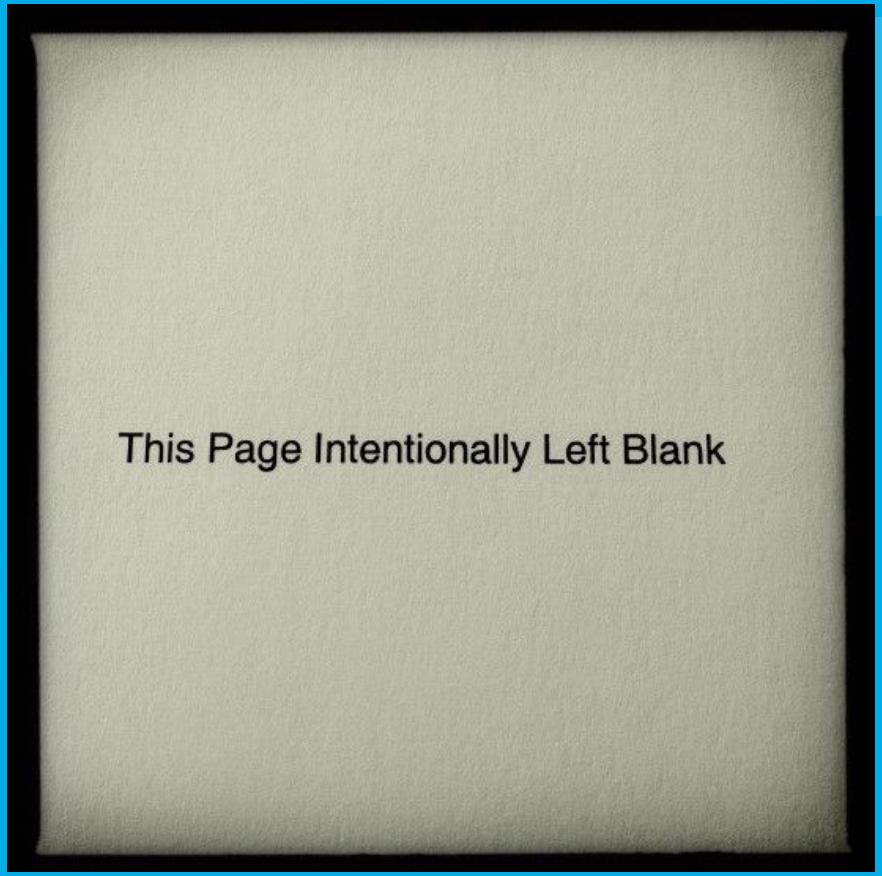
SCENE DI VITA VISSUTA

- Clock che non clockano
- Interrupt che non interrompono
- Interrupt che interrompono troppo
- Stack overflow silenti
- Reset casuali

DEBUG DI UN APPLICATIVO DESKTOP

- *printf()*
- Rettangoli colorati
- Debug passo-passo
- Memory analyzers
- Core dumps
- Testing

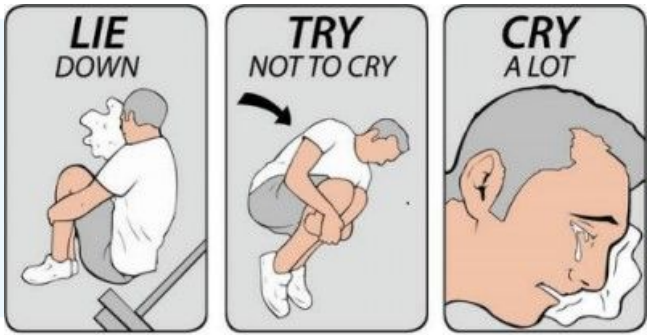
DEBUG DI UN FIRMWARE



PROBLEMI NEL DEBUG DEL FIRMWARE

- I/O testuale limitato o assente
- Tempistiche hardware da rispettare
- No crash log o coredump
- Testing spesso trascurato

E QUINDI CHE SI FA?



- Svincolarsi dalla *printf()*
- Utilizzare debugger on-chip
- Sapere quando fare debug passo-passo
- Prediligere debugging “asincrono”

LIMITAZIONI PRINTF SU UART

- Non utilizzabile in alcuni contesti (eg. ISR)
- CPU-intensive e *bloated*
- Legata alla disponibilità hardware
- Tradizionalmente lenta

UNA POSSIBILE ALTERNATIVA: SWO

- *Serial Wire Output*
- Presente su ARM Cortex-M
- 📈 Veloce (fino a 100 Mbps) e asincrono rispetto al core
- 📈 32 canali su un solo filo
- 📉 Richiede HW e SW dedicati

DEBUG PASSO-PASSO

- Utile nel debug della business logic
- Difficile da usare su problemi legati all'I/O
  - Un core stoppato non serve gli interrupt
  - L'hardware spesso non gradisce

UNA POSSIBILE ALTERNATIVA: TRACING

- Supportato da varie architetture/MCU
- Registrazione delle ultime *N* istruzioni eseguite
- Riproduzione offline del trace
- Ideale per debuggare eccezioni, fault e reset

TRACING SU ARM: ETM

- Non impegna la CPU
- Supporta diversi trigger
  - Breakpoint
  - Watchpoint
  - Interrupt
  - Reset
  - Trigger esterni
- Richiede HW e SW dedicati

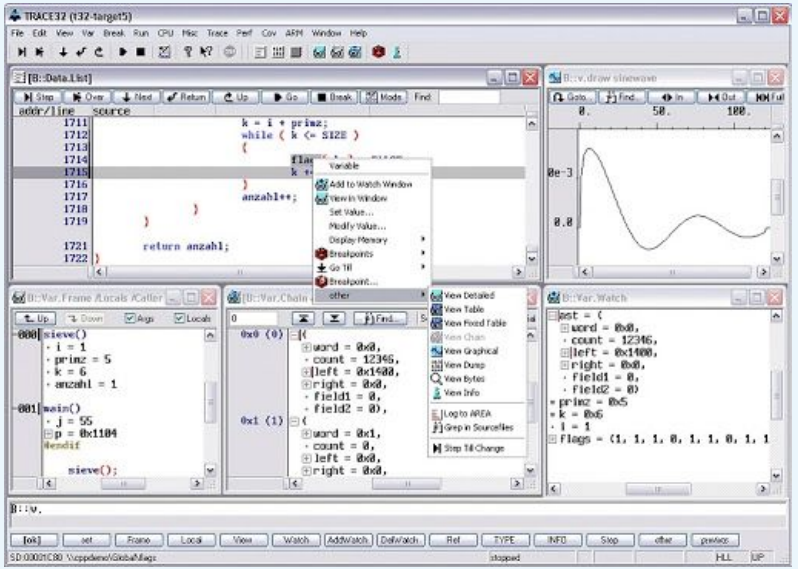
Tutto bello, ma come uso questa roba?

Lauterbach PowerDebug USB 3

- 80+ architetture supportate
- Interfaccia JTAG ed SWD
- Programmazione e debug
- Bare-metal e OS-aware
- Debug userspace e kernel
- Supporto per multicore
- Costo complessivo: svariati euri

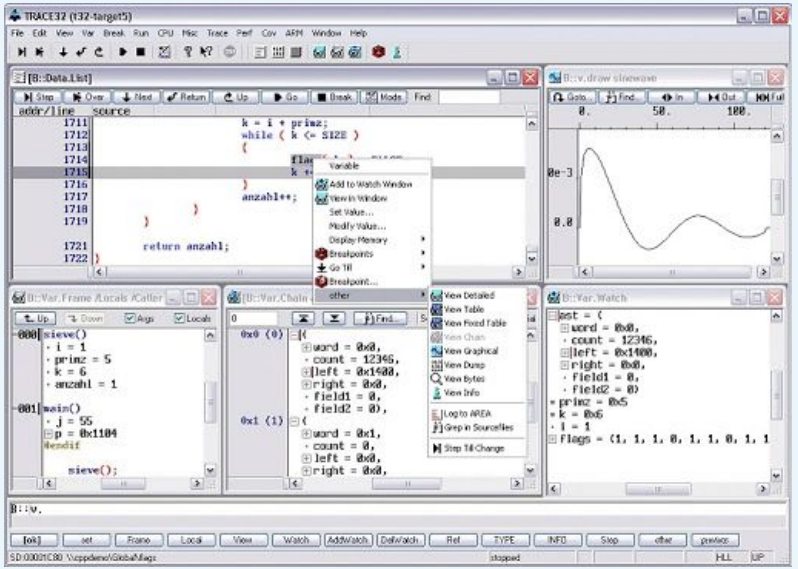
Lauterbach TRACE32

- Interfaccia verso PowerProbe
- Unico per tutte le architetture
- Completamente scriptabile
- Multipiattaforma
- UI moderna e user-friendly



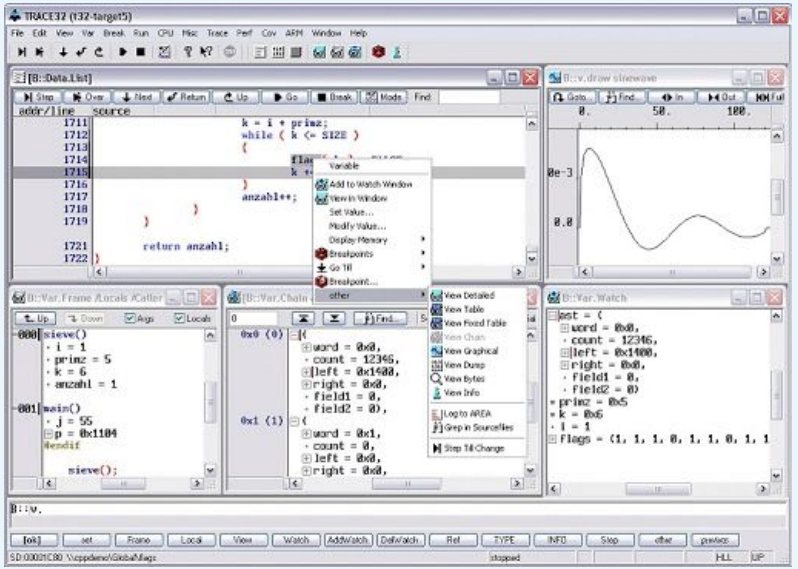
Lauterbach TRACE32 - Funzionalità bare-metal

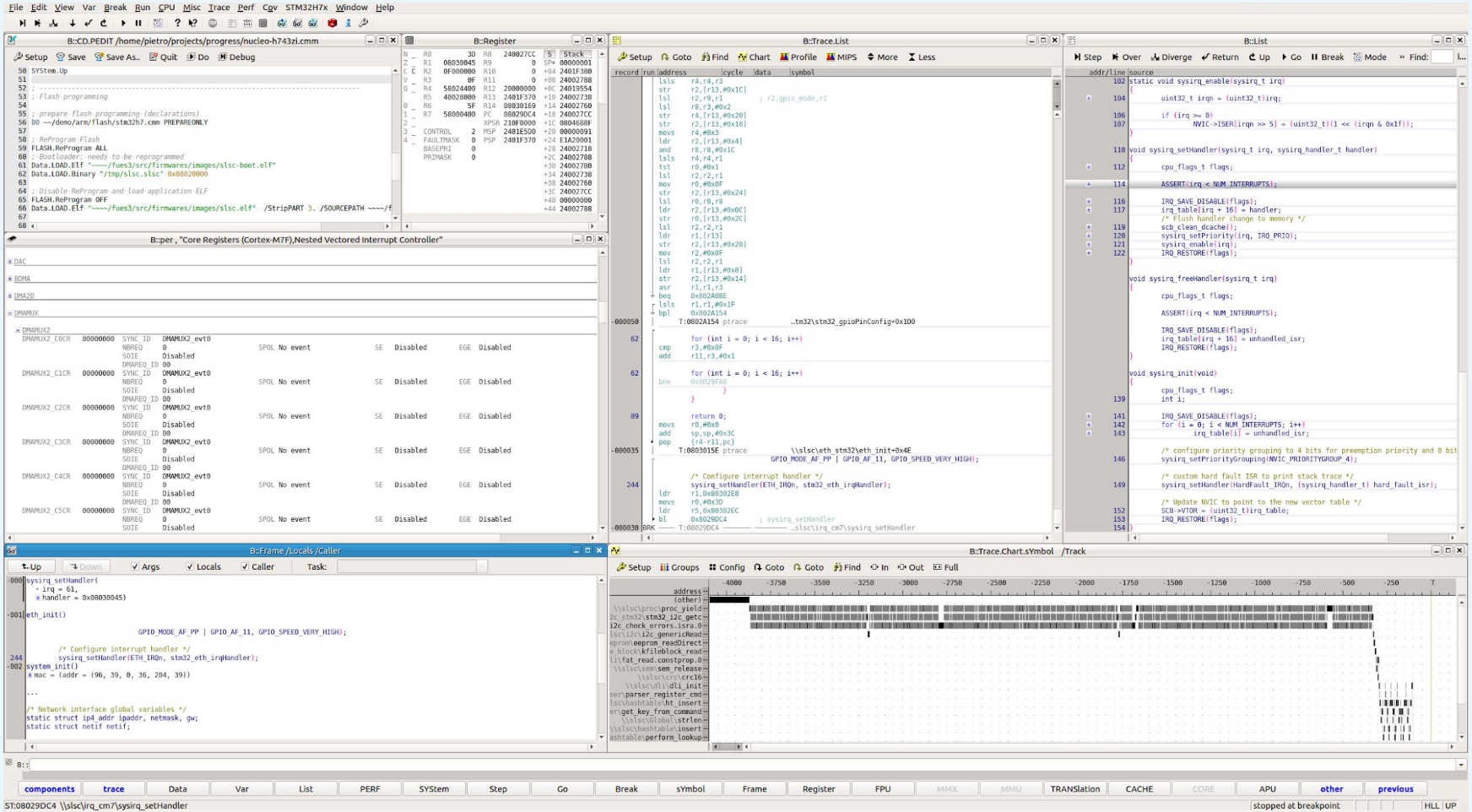
- Debug passo-passo
- Watchpoint su variabili o registri
- Accesso registri CPU e MMIO
- Breakpoint su reset o fault
- Plot di variabili
- Trace visuale



Lauterbach TRACE32 - Funzionalità OS-aware

- Supporto MMU
- Debug processi e kernelspace
- Analisi risorse del kernel
- Loading e debug di moduli




File Edit View Var Break Run CPU Misc Trace Perf Cov STM32H7x Window Help
B::CD.PEDIT /home/pietro/projects/progress/nucleo-h743zi.cmm
B::Register
B::Trace.List
B::List
B::per , "Core Registers (Cortex-M7F),Nested Vectored Interrupt Controller"
B::Frame /Locals /Caller
B::Trace.Chart.sYmbol /Track
components
trace
Data
Var
List
PERF
SYStem
Step
Go
Break
sYmbol
Frame
Register
FPU
MMX
MMU
TRANSlation
CACHE
CORE
APU
other
previous
ST:08029DC4 \\slsc\irq_cm7\sysirq_setHandler
stopped at breakpoint

“BREVI” STORIE TRISTI

- Inizializzazione dei clock
  - Mancava il quarzo 😇
- Crash su context switch
  - Errore nel settaggio delle priorità
- Crash su alcuni IRQ
  - Bug su configurazione D-Cache e MPU

Pietro Lorefice

pietro@develer.com

Vuoi rimanere aggiornato sugli eventi Develer?
Seguici nei nostri canali social:

develer

www.develer.com