

Lauterbach Trace32: Preguntas Frecuentes

rev. Abril/2018

[Lauterbach: ficha de empresa](#)

Los equipos de depuración y traza de Lauterbach son los mejores del mercado. Con ellos se puede hacer todo lo que se quiera (o casi). A continuación se resumen algunas preguntas habituales acerca de Trace32 y sus respectivas respuestas.

1. [Equipo general](#)
2. [Entorno](#)
3. [Traza](#)
4. [Emulación](#)
5. [Soporte técnico](#)

1. EQUIPO GENERAL

Q. ¿Necesito mucho tiempo para aprender a usar el equipo?

A. Sistema de ayudas sensible al contexto. **Tutoriales** y videos. Desde Capel ofrecemos además cursos para una rápida adaptación ([ver Cursos](#))

Q. ¿Tienes que cambiar de **sistema operativo** en tu host para depurar?

A. Debugger instalable en cualquier sistema operativo (dos, win32, win64, linux, unix, VM, OSX...).

Q1, Q2. ¿Cuánto tardas en cargar el objeto absoluto en tu tarjeta?

A. Hardware de **conexión** veloz (usb3, ethernet, gigabit ethernet). Basado en micros de 64bits.

[Video 1](#)

Q. ¿Tienes que comprar otro emulador cuando cambias de arquitectura o familia?

A. Los equipos son genéricos (fácilmente **personalizables** a otra cpu/arquitectura, sólo comprando una licencia o una pinza).

Q. ¿Puedes añadir un analizador lógico a tu emulador, o un rom monitor?

A. Sistema **modular** y ampliable: traza, analizador lógico, medida potencia (necesario PowerIntegrator o CombiProbe o uTrace), generador estímulos, emulador rom.

Q. ¿Puedes **sincronizar** con otros equipos? Analizadores lógicos, osciloscopios.

A. Sí, mediante salida de pin de trigger programable.

Q. ¿Puedes depurar tarjetas con **multiprocesador (SMP y AMP)**?

A. Si tienen un JTAG común, para SMP, sólo tienes que comprar la licencia de software, en caso de que quieras depurar en dos instancias. Se sincroniza el arranque y la parada de ambas instancias del core.

Q. ¿Cuánto tardas en **cargar** el código en tu tarjeta cada vez que haces un cambio?

A. Alta velocidad de carga (modo diferencias y dualport).

Q. ¿Necesitas **grabar** otros dispositivos soldados en la tarjeta o sd card?

A. Grabación de distintos tipos de memorias: flash interna del microprocesador y flash externa (NOR, NAND, serial, MMC). Detección CFI. Protocolos de control de errores ECC.

Q. ¿Puedes ver el contenido de una NAND y explorar la tabla ECC?

A. Visualización de datos en memorias NAND-Flash: contenido y control de errores.

Q. ¿Debes usar siempre el mismo compilador, porque tu emulador no acepta otros formatos?

A. Cargadores de código y símbolos para **cualquier formato de compilador/enlazador** (C/C++, Java, ADA...).

Q. ¿Puedes hacer pruebas antes de tener la tarjeta en funcionamiento, o si quieres cambiar de familia, o si quieres analizar una traza guardada (post-mortem)?

A. **Simuladores** disponibles. Evaluación flexible por tiempo limitado.

Q. ¿Puedes comprobar con tu emulador el circuito impreso y los chips (entradas y salidas)?

A. Comandos **BoundaryScan**. Permite controlar los pines de todos los chips en la cadena JTAG. Usado en test de tarjetas. También puede usarse BSDL para programar Flash externas.

2. ENTORNO

Q. ¿Puedo usar el entorno en test y producción?

A. No tienes por qué usar el entorno de depuración suministrado por Lauterbach. Puedes hacerte uno propio, fácil de usar, que llame a funciones del **API de T32** remotamente. Control remoto del equipo, para aplicaciones de test. API para conexión a programas de terceros o para llamadas a T32 en modo batch lenguaje de script completo con lenguaje (**PRACTICE**) de programación para automatizar funciones.

[Video 2](#)

Q. ¿Puedo ver estado de mi sistema operativo en alto nivel, tareas, semáforos, stack...?

A. Soporte **RTOS incluido**. Tasks breakpoint, task performance, display elementos RTOS, consumo de stack, RTOS awareness incluido para los más conocidos: Linux, uCLinux, Android, CMX, ECOS, QNX, pSOS, RTX, SYmbian, uC/OS-II, WinCE, VxWorks, FreeRTOS, Keil,ORTI...

LINUX / android: front-end para gdb y UndoDB . Para depuración de aplicaciones de Linux (modo RUN). No es necesario hardware.

LINUX /android: Modo RUN/STOP. Se necesita emulador, permite depurar Boot, kernel y drivers.

[Video 3](#)

Q. ¿Puedo depurar sistemas con múltiples sistemas operativos y máquinas virtuales?

A. T32 dispone de un sistema de gestión de hypervisor, que permite ampliar el rango de las direcciones a las que accede el microcontrolador y detectar el sistema, la máquina virtual y el sistema operativo en que está corriendo.

Q. ¿Puedo trabajar en mi IDE o tengo que moverme al entorno de Trace32?

A. **Interfaces** con IDE de desarrollo (FastDataXchange): Eclipse CDT o TCF, EasyCase, VBasic, CodeWright, DA-C, Rhapsody, LabView, SimuLink, PIL, WindRiver WorkBench

Por ejemplo, usando PIL puedes usar el hardware real conectado al debugger Trace32 como target del test. Trace32 permite pasar parámetros a las funciones y ejecutarlas, obteniendo un resultado que se devuelve al test preparado por PIL.

3. TRAZA

Q. ¿Soporta todas las conexiones de traza?

A. Máxima velocidad y tamaño de traza (Hasta 640Mbytes/s; 4Gigabit serie); para soportar las últimas tecnologías.

Modo **Streaming** y RTS (realtime). Análisis de la traza en tiempo real, sin pararla (necesita PowerTrace II o CombiProbe o uTrace).

Autofocus en algunas arquitecturas (ARM, PPC), permite sincronizar los tiempos y niveles de muestreo del bus de traza automáticamente.

Sincronismo de tiempos con las ventanas de fuente y desensamblado y otras vistas de la traza.

[Video 4](#)

Q. Code coverage. ¿Cómo saber si se ejecuta todo el código escrito (normativa de seguridad)?

A.

Data coverage: saber si las variables se han escrito/leído

Shadow memory: permite ver variables en ejecución

[Video 5](#)

Q. **Análisis estadísticos de las funciones.**

A. Tiempo min, máx, medio. Número de veces que entran y salen.

Q. CTS

A. Ejecución del programa **hacia atrás** con los datos de la traza visualizando registros y variables.

Q. Traza instrumentada ITM.

A. Algunos chips (Cortex) tienen distintos registros para escribir en ellos desde el código de la aplicación. Dicha traza se graba en el buffer de la unidad de traza, o se manda al pc en tiempo real (pipe, stream) para ser analizada. Uno de los filtros sobre estos datos es la interpretación ascii.

[Video 6](#)

Q. Traza de datos CoreSight

A. Además de ver la evolución de una variable global, pueden capturarse por la traza accesos a dicha variable, junto con otro contador del procesador. T32 analiza la traza capturada.

[Video 7](#)

Q. Contadores BMC

A. Algunos micros disponen de **contadores** internos. La traza puede programarse para leer esos contadores en tiempo real. De ese modo tenemos un perfil del trabajo de la cpu en cada momento.

[Video 8](#)

4. EMULACION

Q. ¿Puedes ver la evolución de las variables y poner breaks sin parar la emulación?

A. Aprovechamiento del modo de acceso a la cpu y memoria. **Dual-port** (poner breaks mientras está en ejecución, trazas de variables en tiempo real) y modo SNOOPER de acceso a datos (para leer o modificar mientras está en ejecución).

Q. Mi micro no tiene posibilidad de traza. ¿Puedo calcular tiempos en mis funciones?

A. Efectivamente, sin traza las medidas no son exactas pero tienes el modo **snooper** y el módulo RT para capturar periódicamente el contador de programa. A partir de esta pseudo-traza tienes disponible toda la estadística de lo capturado.

[Video 9](#)

Q. Mi micro no tiene traza de instrucciones o no tengo los pines conectados al conector. ¿Puedo ver la evolución de mi programa?

A. Tienes **distintos modos de traza** dependiendo del hardware, aún sin tener traza física. Snoop, stop/go, logger (traza en estructuras de ram de usuario, intrusivo).

[Video 10](#)

Q. ¿Cuántos breaks puedes poner en tu código?

A. **Breaks ilimitados** en ram/flash: algoritmo de diferencias en Flash.

[Video 11](#)

[Tutorial Breakpoints con Trace32](#)

Q. ¿Puedes detectar cuando una determinada función escribe un byte de un dato y dejar el resultado en un archivo?

A. **Breaks avanzados:** ejecución de acciones y cuentas, con calificación del tipo de acceso.

[Video 12](#)

Q. ¿Puedes desconectar los breaks de excepciones?

A. Control individual de cada excepción, para que produzca o no un break.

Q. ¿Puedes parar los **periféricos** cuando estás en un break, por ejemplo, para que un timer no siga descontando?

A. Depende del microprocesador. Trace32 tiene acceso a todos los registros del core y del SOC, donde puedes programar el modo de depuración.

Q. ¿Puedo parar la ejecución con señales externas de mi hardware?

A. Señales de **trigger out** desde eventos de emulación. Señales de trigger in desde el exterior.

Q. ¿Puedo ver las tablas de **MMU** y el estado de **CACHE**?

A. Visualización de MMU y analizador de CACHE, hits, miss...

Q. Tengo que homologar mi software. ¿Tengo informes de code coverage?

A. Tienes un **code coverage** limitado sin traza, completo con traza. Plantillas editables XML para los informes de traza y código, usada en homologaciones.

Q. ¿Tengo que descriptar el dump del stack para ver su contenido?

A. Presentación legible del **stack frame**, con símbolos, variables locales y llamadas en cada función.

Q. ¿Cómo estudio el llenado del stack?

A. Coloreado de zonas de memoria, análisis de memoria: malloc/free, traza de allocation, display gráfico de uso de memoria. Análisis de CACHE.

Q. Tengo que leerme todo el manual para interpretar qué bits controlar y qué función en los periféricos del microcontrolador?

A. Visualización de **periféricos** externos o internos estructurada y legible: bits, words, modos. Bits agrupados por función, visualización textual de la función.

Q. ¿Puedo hacer "performance analysis" ?

A. Sí, puedes obtener los datos a partir de la traza o mediante SNOOPER. Análisis de tiempos, hits de funciones, anidamiento de las funciones.

[Video 13](#)

[Introducción a Performance Analyzer](#)

Q. ¿Cómo controlo la ejecución?

A. Control de go/step/step over/step diverge sobre código y acceso a variables (parar cuando una variable o un registro tenga un valor determinado).

Q. ¿Cómo visualizas el contenido de variables, datos, arrays, estructuras?

A. Además de los normales watch y dump de memoria, puedes **procesar la variable**: (tablas, estructuras, punteros indirectos y dinámicos, imágenes, variación en el tiempo, FFT...).

[Video 14](#)

[Visualización y Edición de Variables](#)

Q. Cuando conecto mi equipo a una CPU de 2 núcleos SMP (caso Zynq-7000 por ejemplo), el kernel awareness se aplica a los dos núcleos si hay dos aplicaciones distintas ejecutándose en cada core. Se depura bien pero el kernel awareness queda confundido. ¿Solución?

A. En el caso de micros ARM Cortex-Av7 se soluciona con añadir una licencia multicore y ejecutar dos instancias, así se puede cargar el awareness que quieras en cada instancia. En el caso de disponer de varias licencias de una misma familia (ARM en este caso) en un mismo cable se incluye por defecto la licencia multicore. En el caso de ARM Cortex-Av8 la licencia multicore se incluye por defecto.

5. SOPORTE TÉCNICO

Directo de los diseñadores. <http://www.lauterbach.com/tsupport.html> search “arm” o “icd”.

O de Capel a través de tecnico@captura-el.com