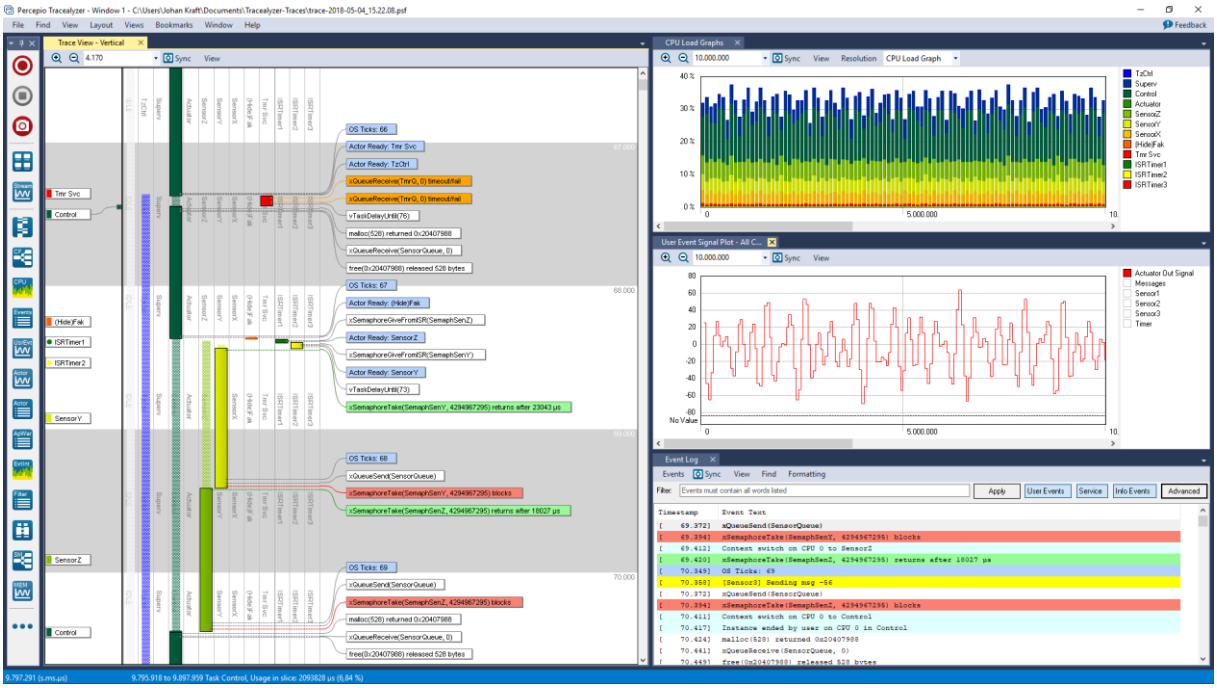
Tracealyzer streaming over Arm ITM using IAR Embedded Workbench

Application Note PA023 (v2), 2025-08-27.



For Arm Cortex-M devices, Tracealyzer supports trace streaming over Arm's debug interface ITM. This is well supported via IAR Embedded Workbench and I-Jet debug probes, offering high performance and a convenient workflow. This works for all real-time operating systems supported by Percepio's TraceRecorder library including Zephyr and FreeRTOS.

The ITM interface (short for Instrumentation Trace Macrocell) allows the Tracealyzer recorder library to transmit software-generated trace data via the debug probe using the SWO pin (Serial Wire Output) which is present on the standard 9/10-pin Cortex debug connector and supported by the standard IAR I-Jet probe. With an IAR I-Jet, ITM over SWO allows for high data rates (20+ MHz if using Manchester mode) resulting in minimal target-side overhead, as can be seen the Performance Measurement section at the end of this document.

Requirements

To follow the instructions in this document, you need the following:

- Target system: Any Arm Cortex-M processor with ITM support and a trace-enabled debug connector. A full ETM trace port can be used but is not required. SWO is used in this guide.

  - ITM is supported by practically all Arm Cortex-M except for M0, M0+ and M23 devices. It is typically available in M4, M7 and M33 MCUs.
- A target software system: Using an RTOS supported by Percepio TraceRecorder (e.g. Zephyr or FreeRTOS), alternatively a bare-metal system with TraceRecorder added.
- Development tools
  - IAR Embedded Workbench for ARM
  - IAR I-Jet or I-Jet Trace
  - Percepio Tracealyzer v4.x or later.

Target-side setup

The trace recorder library needed by Tracealyzer is included with the application and found in the application directory, under “Tracealyzer 4/<RTOS Name>/TraceRecorder”. To integrate this in your IAR project in streaming mode, follow the getting started guide provided in the Tracealyzer User Manual (see “Integrating the recorder” and “Setting up streaming”).

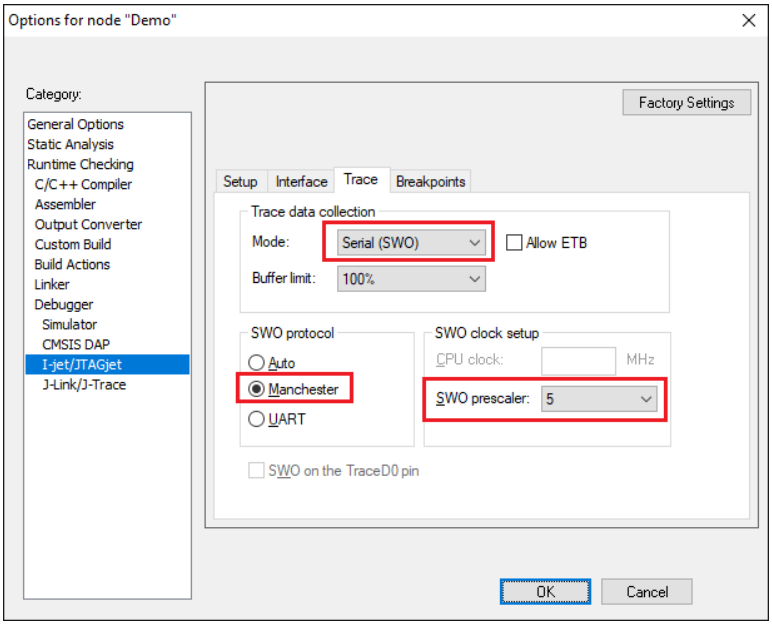
Make sure to include the “streamport” module for ITM, found in streamports/ARM_ITM. This works with any IDE and debug probe, assuming they have ITM support and an ability to output the ITM data to a binary file, although the ITM throughout depends on the debug probe used.

When calling xTraceEnable to initialize the recorder, use the parameter “TRC_START” which starts the tracing directly. It is not yet possible to control the recording from the Tracealyzer application, as there is only one-way communication when using ITM. However, since you can trace over practically unlimited periods in the streaming mode, you can record your entire debug sessions.

Host-side setup – IAR Embedded Workbench

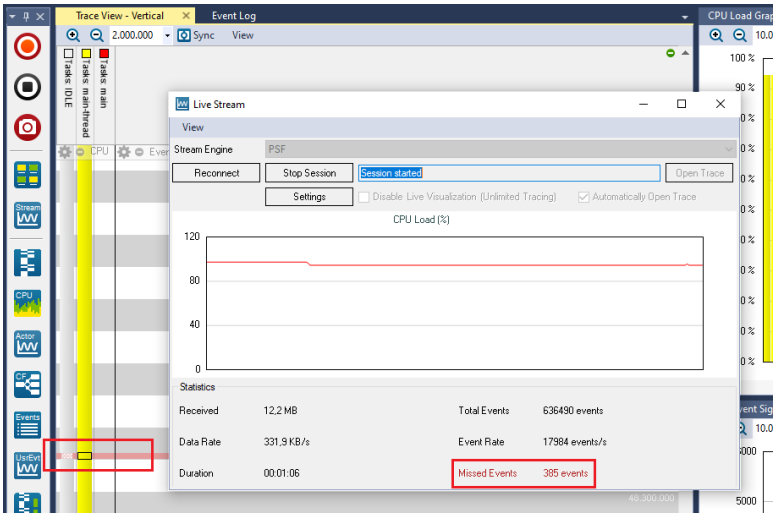
In IAR Embedded Workbench, start by opening Options… for your project and check the following settings:

- Debugger -> Setup -> Driver: I-Jet/JTAGjet
- Debugger -> I-Jet/JTAGjet -> Interface: SWD
- Debugger -> I-Jet/JTAGjet -> Trace -> Trace data collection -> Mode: SWO
- Debugger -> I-Jet/JTAGjet -> Trace -> SWO protocol: Manchester
- Debugger -> I-Jet/JTAGjet -> Trace -> SWO clock setup -> SWO prescaler: (see below)



For the prescaler value, note that the “auto” setting for the SWO prescaler can result in a slightly too high SWO frequency and occasional transmission errors. These are seen in the Tracealyzer Live Stream as a “Missed Events” counter, as seen below. This is only displayed if there are missed events. In that case, increase the prescaler value to reduce the SWO frequency.

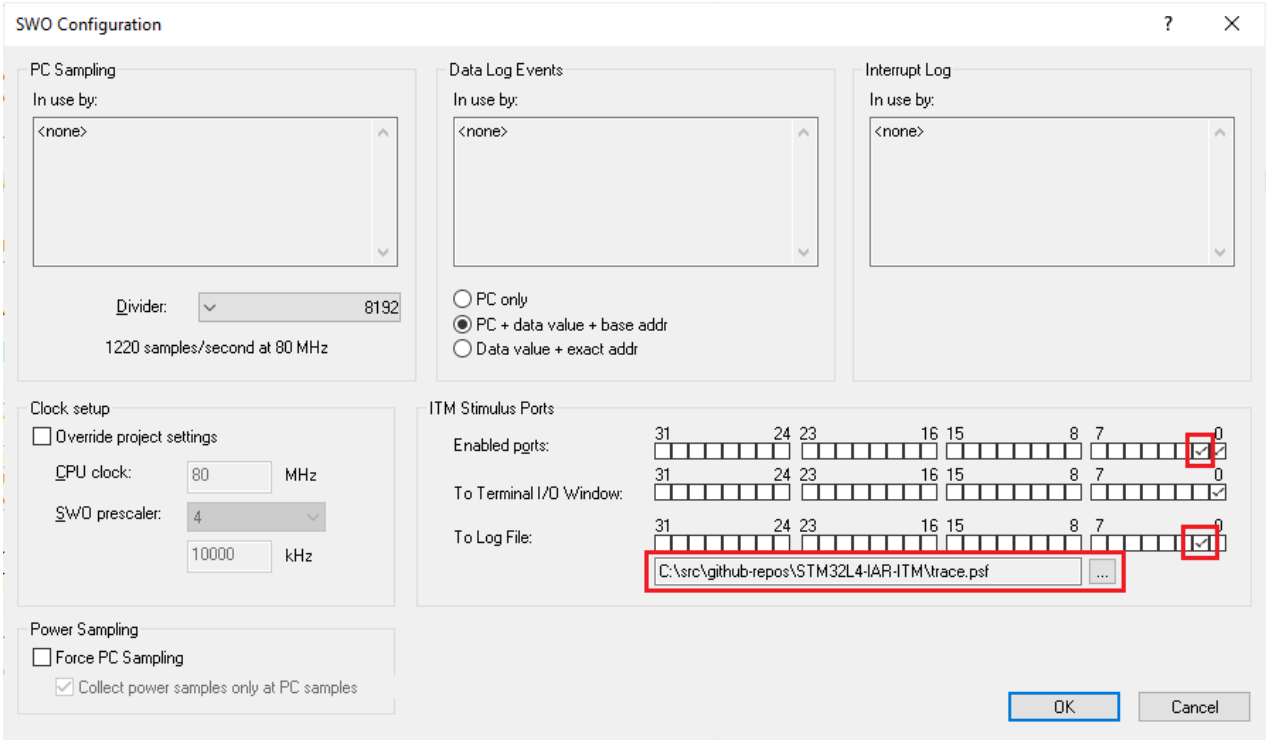
As rule of thumb for the prescaler value, start with a value corresponding to the processor core clock frequency divided by 40-50 and round up. For example, at 240 MHz, a prescaler of 5 or 6 should be suitable. For an 80 MHz MCU, a prescaler of 2 should be sufficient.



It is recommended to try different prescaler values to find the highest SWO frequency that doesn’t cause any missed events.

https://percepio.com

Start a debug session and open the SWO Configuration dialog, found in the I-Jet menu.



Under *ITM Stimulus Ports*, check the second checkbox from the right to enable Port 1, as shown above. Also enable port 1 under “To Log File” and specify the output file name as “trace.psf”.

Ensure that nothing else uses ITM port 1 and that no other ITM ports are logged to the file. If you want to change the ITM port used, you need to update this setting also in the TraceRecorder library. The setting is called TRC_CFG_STREAM_PORT_ITM_PORT and found in streamports/ARM_ITM/config/trcStreamPortConfig.h.

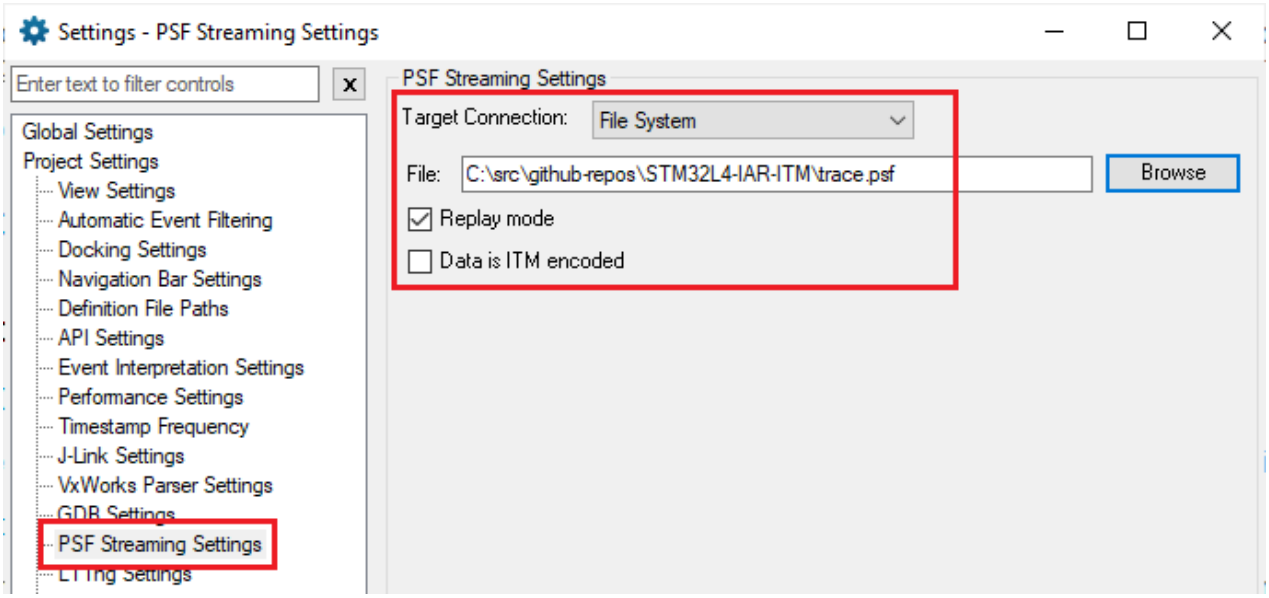
IAR Embedded Workbench should now be correctly configured for Tracealyzer streaming.

Host-side Setup - Tracealyzer

To simply view your recorded trace file, no settings are required in Tracealyzer. Simply select File -> Open -> Open File and select your trace.psf file. Note that file will be overwritten each time, so you need to save a copy manually if you want to keep the trace.

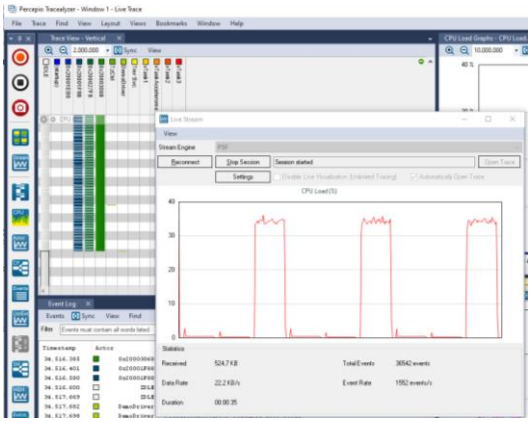
Tracealyzer also supports live visualization using a similar approach. In this case, trace.psf is used as a channel between IAR and Tracealyzer and a copy is saved automatically after each session. This requires the following configuration in Tracealyzer:

Select File -> Settings -> PSF Streaming Settings and set the “Target Connection” setting to “File System” and as File specify your trace.psf, like shown below.



To record a trace, begin in Tracealyzer by selecting Start Recording (the big red button in the main navigation bar), or in the main menu select Trace -> Open Live Stream Tool, Click Connect and Start Session.

Next, start a debug session in IAR Embedded Workbench and run the application. Tracealyzer should now receive and display the data live. Stop the recording to enable all 30+ views in Tracealyzer.



During tracing, you may stop on break points, single-step and resume execution without interfering with the resulting trace. This since the event timestamps are set based on the target system cycle counter, so halting the system “stops the time” for the recorder library.

You don’t need to have the Event Log or SWO Trace views open in IAR Embedded Workbench, the data is forwarded to Tracealyzer anyway, as long as a debug session is running with the “To Event Log option” enabled´.

Troubleshooting

If Tracealyzer doesn't show any data, you may check the following:

Step 1. To ensure that the trace recorder library has been properly integrated in your target system, put a breakpoint in prvTraceItmWrite(), found in trcStreamingPort.c, and verify that it is called from the trace macros inside the RTOS functions. If not, consult the general troubleshooting guide in the Tracealyzer User Manual.

Step 2. To ensure the data is transferred correctly between the target system and IAR Embedded Workbench, make a simple test program like shown in the screenshot below. This writes a known byte pattern to the ITM port, so you can check the integrity of the data. Use the function ITM_EVENT32 (defined in "arm_itm.h"), where the first parameter is the ITM port.

```
#include "arm_itm.h"

void test_itm_transfer(void)
{
  while(1)
  {
    ITM_EVENT32(1, 0xAABBCCDD);
    ITM_EVENT32(1, 0x11223344);
  }
}

void main(void)
{
  test_itm_transfer();
}
```

Start a debug session and open the Event Log in IAR Embedded Workbench, found in the I-Jet menu. Then run the test program for a while and stop. Check carefully that you get the right data, without any random anomalies. If the data looks good, you can try decreasing the prescaler and repeat the analysis. If you get errors in the data, increase the prescaler and try again.

Step 3. If you don't get any data at all, check the debug cable between the IAR I-Jet and the debug connector on the board. Make sure the cable is in good shape and that the connectors are fully inserted.

Step 4. Try setting the CPU clock manually in SWO Configuration, under Clock Setup. The clock frequency of your target system can be found by inspecting the global variable SystemCoreClock, once the low-level setup has completed. This variable is defined in Arm's CMSIS library, which is often available by default.

Performance Measurement

We have measured the performance of ITM streaming on a Renesas S7G2 with an Arm Cortex-M7 core running at 240 MHz, IAR Embedded Workbench v7.71.1 (Renesas Synergy version) and an IAR I-Jet configured at 24 MHz SWO frequency.

With this setup and the compiler optimization level set to “high”, sending 16 bytes of data (4 writes of 32 bit each) took only 36 clock cycles (0.15 µs). With the compiler optimization level set to “low” the write time was 88 clock cycles, but this is still quite fast (0.37 µs).

Achieving this performance however requires that the ITM FIFO has room for the data, otherwise the write operation will block while waiting for the previous data to be transmitted. We measured this by writing the 16-byte bursts without any delay in order to saturate the SWO transfer, and the write time then increased to 3 266 clock cycles (13,6 µs) due to this blocking.

This also revealed the maximum throughput of the setup. We managed to transmit over 73 400 events per second of 16 byte each, meaning that the throughput was 1 175 KB/s. This is a lot more than typically needed for RTOS tracing with Tracealyzer.

However, in real applications where the events are usually separated in time by application code, there is typically very little blocking as the ITM FIFO has time to flush out the data in between the ITM writes. To test this, we added a delay of 3300 clock cycles in between the 16-byte bursts to simulate application code running between traced events. This resulted in constant ITM write times (36 resp. 88 clock cycles, depending on optimization level) which confirmed that no blocking occurred.

Note that the bottleneck here is the trace port, i.e. the 24 MHz SWO frequency. This is not affected by the compiler optimization level or the relatively high clock frequency of the Renesas S7G2 MCU. Similar throughput is possible also on slower MCUs by lowering the SWO prescaler.

Learning more

To learn more about Tracealyzer and ARM ITM, we recommend the following resources:

- Download: https://percepio.com/tracealyzer/download-tracealyzer
- Tracealyzer User Manual (Help -> User Manual)
- Tracealyzer “Getting Started” portal - https://percepio.com/gettingstarted

For technical support and other questions, please contact at https://percepio.com/contact-us.