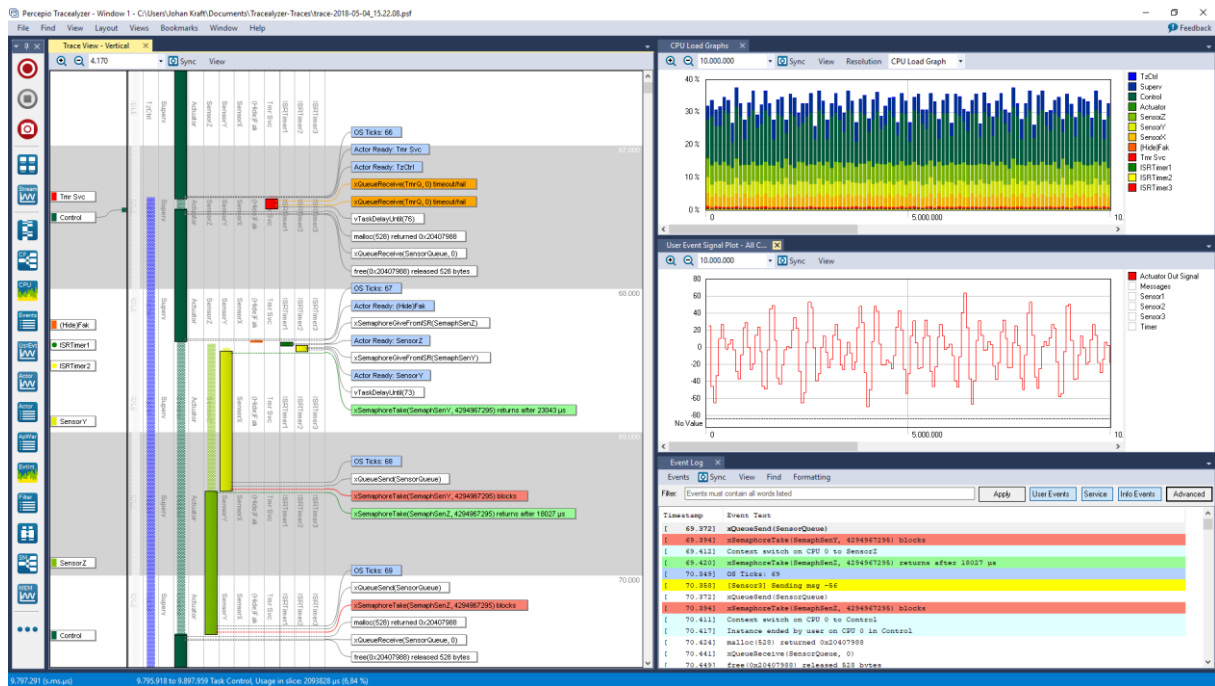


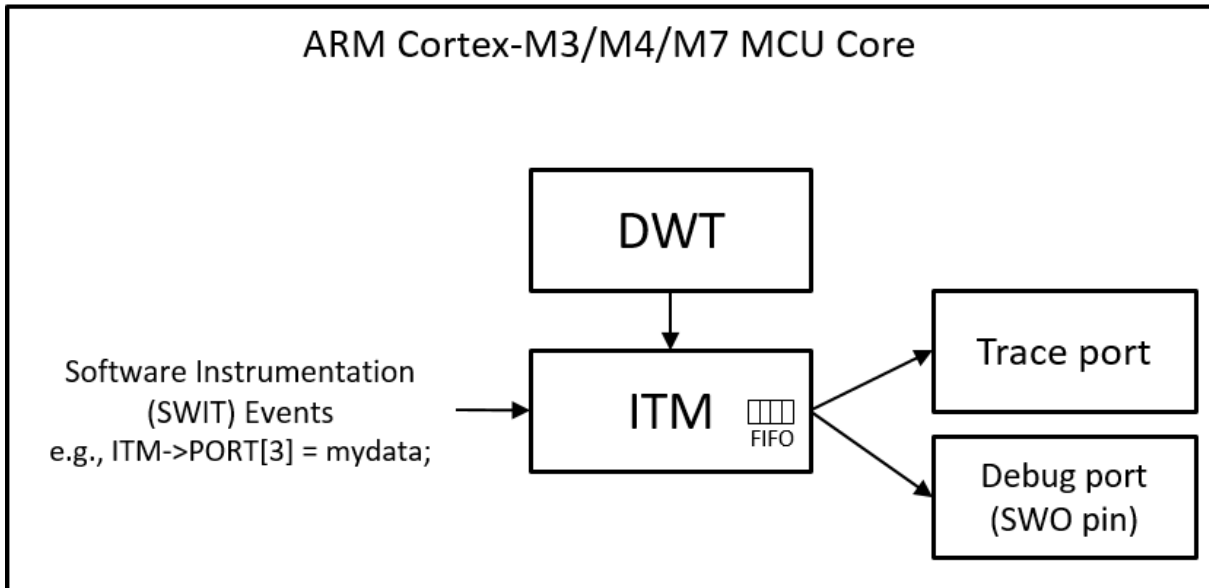
Tracealyzer streaming over ARM ITM using ULINKpro/ULINKplus

Application Note PA021, 2018-05-03



As of version 4.1, Tracealyzer supports [ITM tracing](#) for ARM Cortex-M3, M4 and M7 MCUs. This is initially available for FreeRTOS and Micrium μ C/OS-III, but support is planned for SaferRTOS and ThreadX soon. An example screenshot of Tracealyzer using this setup is shown above.

The ITM interface (short for Instrumentation Trace Macrocell) allows the Tracealyzer recorder library to send event data via the debug probe, either using the SWO pin (Serial Wire Output), or over the ETM trace pins if available. SWO is found on all [ARM Cortex Debug connectors](#) and is supported by most debug probes.



With a fast debug probe like Keil [ULINKpro](#) or [ULINKplus](#), ITM allows for high data rates (over 2 MB/s) and the target-side overhead is minimal. In our experiments using a ULINKpro, writing 16 bytes of data took just 55-60 clock cycles, despite this requiring four separate writes. Moreover, unlike some other streaming solutions we support, ITM tracing does not require any RAM buffer.

Requirements

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

- **Target system:** Any ARM processor with ITM support.
 - o Supported by practically all Arm Cortex-M3, M4 or M7 MCUs.
 - o Not supported by ARM Cortex-M0 or M0+ MCUs.

- **RTOS:** An RTOS supported by Percepio Tracealyzer using the Percepio recorder library
 - o FreeRTOS (v7.3 or newer)
 - o Micrium μ C/OS-III (v3.04 or newer)

- **Development tools**
 - o Keil μ Vision - any version since November 2013 should work.
 - o Keil ULINKpro or ULINKplus
 - o Percepio Tracealyzer v4.1, with a license matching your RTOS.

Target-side setup

Follow the getting started guide for streaming mode, as provided in the Tracealyzer User Manual (“Integrating the recorder” and “Setting up streaming”). Make sure to include the “stream port” found in “TraceRecorder/streamports/ARM_ITM”. This is a generic solution for ITM tracing that works with any IDE and debug probe, assuming they have Trace/SWO support and an ability to output the ITM data to a binary file.

When calling `vTraceEnable` to initialize the recorder, we recommend using the parameter “TRC_START”, which starts the tracing directly.

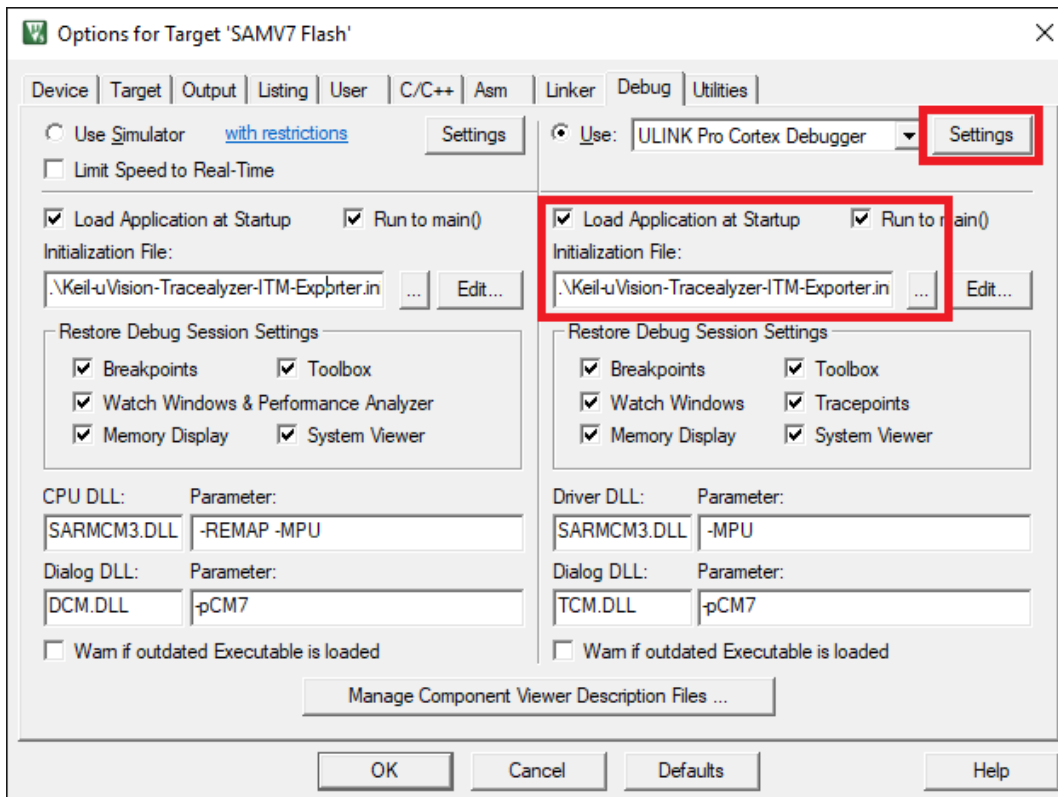
Host-side setup

To make Keil μ Vision output the ITM data, there is an excellent command called ITMLOG. This is described in Arm Keil Application Note 240, [Redirecting ITM Output to third-party application](#). You don’t need to learn the details of this command, as we provide a μ Vision script file that automates this for you, *Keil-uVision-Tracealyzer-ITM-Exporter.ini*, found in the TraceRecorder/streamports/ARM_ITM directory. The central part of the script is this line:

```
exec("ITMLOG 1 > .\\tracealyzer.psf");
```

This makes μ Vision output all data on ITM port 1 to the “tracealyzer.psf” file, created in the project working directory. Moreover, the script adds two new buttons to Keil μ Vision, “Start Recording” and “Stop Recording”, shown when starting a debug session. If they don't appear automatically, select View -> Toolbox Window. By default, the script starts the tracing directly, just like if clicking on “Start Recording”, but you can modify that if you like (there are some comments in the .ini file).

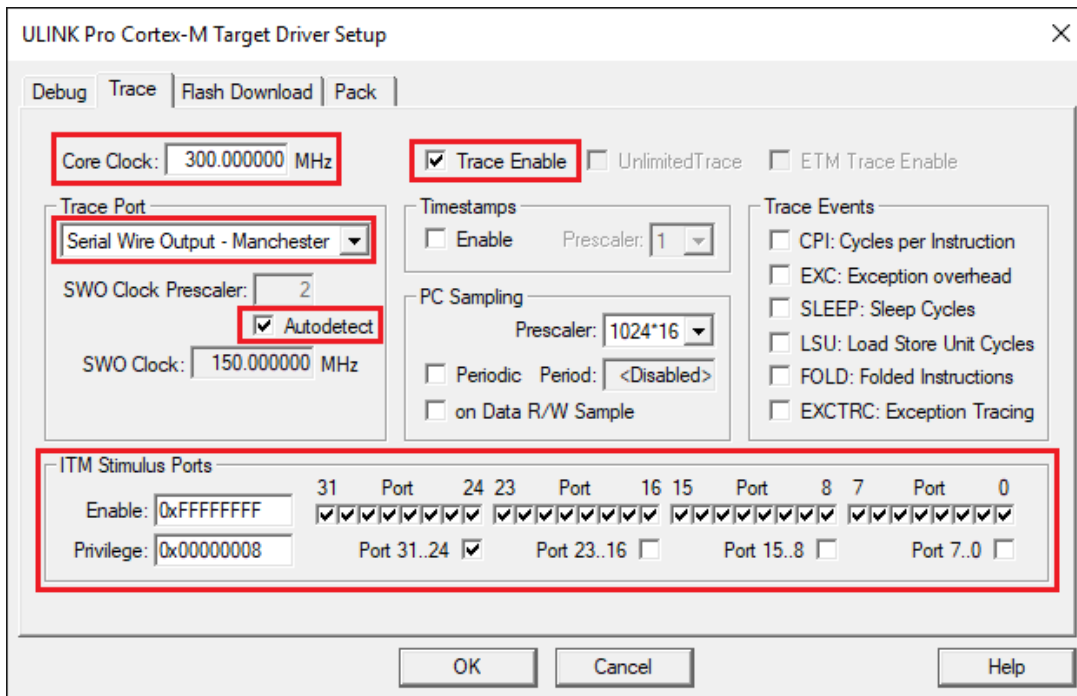
The first step is to register this script file to execute every time you start a debug session. Open Project -> Options for Target... and specify Keil-uVision-Tracealyzer-ITM-Exporter.ini as “Initialization File”, as seen below, found in Tracealyzer 4/TraceRecorder/streamports/ARM_ITM.



Next, click the “Settings” button for your ULINKpro or ULINKplus debug probe, in the top right corner. This opens another settings window specific for your debug probe, in my case “ULINK Pro Cortex-M Target Driver Setup” (should be similar for ULINKplus).

Make sure you have the following settings:

1. Debug tab
 - a. Port: SW
 - b. Max Clock: 50 MHz / highest setting
2. Trace tab
 - a. Core Clock: The clock frequency of your processor
 - b. Trace Enable: Checked
 - c. Trace Port: Serial Wire Output – Manchester
 - d. SWO Clock Prescaler: Autodetect
 - e. ITM Stimulus Ports – At minimum Port 1 enabled (default setting)
 - f. Other settings: Unchecked



The ITM port used is defined in `trcStreamingPort.h` and is by default Port 1. This can be changed, assuming you adjust the host-side settings accordingly - the `ITMLOG` command in `Keil-uVision-Tracealyzer-ITM-Exporter.ini` and the ITM Stimulus Ports enabled.

Recording and viewing traces in Tracealyzer – Basic Approach

Once you have integrated the trace recorder in your project, configured it for streaming using the `ARM_ITM` stream port, and configured Keil μ Vision as instructed, you can start recording traces and view them in Tracealyzer. The most basic approach is:

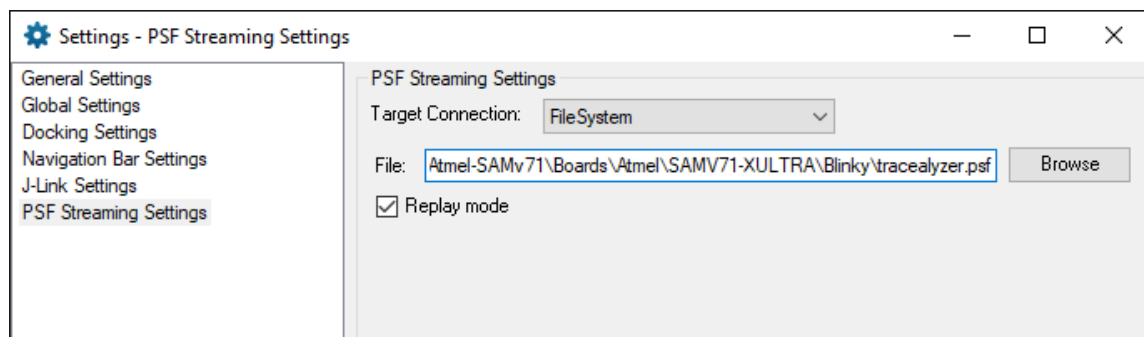
1. Start a new debug session in Keil μ Vision and let your system run. The tracing starts directly by default, so you don't need to use the "Start Recording" button in Keil μ Vision.
2. When you have recorded your test session, click the "Stop Recording" button in Keil μ Vision (or simply close the debug session) to make sure the trace file is properly closed.
3. In the μ Vision project directory, you should now have a trace file, "tracealyzer.psf". Open this file in Tracealyzer using File -> Open, or simply just drag the file into Tracealyzer.

Note that the next trace session will overwrite the `tracealyzer.psf` file, so if you would like to save your trace you need to move or rename this file.

Recording and viewing traces in Tracealyzer – File Streaming and Live Visualization

Tracealyzer 4 supports live visualization in several views. You may view ITM traces collected by Keil μ Vision live, by configuring Tracealyzer to stream the data from the trace file produced by Keil μ Vision. Tracealyzer can read and show the trace file “live” while it is being recorded.

Open the Tracealyzer settings (File -> Settings, or “Recorder Settings” on the start screen) and select PSF streaming settings. Set Target Connection to “FileSystem” and select “Browse” to specify your tracealyzer.psf file. Note that Tracealyzer regards this as a temporary file, just a “data source”, and still saves the trace data in a separate permanent file. So you don’t need to make copies of tracealyzer.psf manually.



The “Replay mode” decides how Tracealyzer reads the data, which affects the smoothness and latency of the display.

- **Replay mode enabled:** This mode limits the speed of Tracealyzer to match the target system timestamps. Assuming some data already in the file when the Tracealyzer begins to read, it typically won’t run out of data to display. This way you get smooth live views, despite Keil μ Vision writing the ITM data in big chunks due to internal buffering. However, this adds some additional latency between the target system and the Tracealyzer views.
- **Replay mode disabled:** Displays the data as fast as possible, in the rate it arrives in the file. When using this for Keil ITM streaming, this minimizes the latency between target system and the Tracealyzer views, but the updates of the live views will not be as smooth as in the replay mode.

To view the trace “live” in Tracealyzer, follow these steps:

1. Launch a new debug session and start it. The tracealyzer.psf file is now accumulating data.
2. Select “Start Recording” in Tracealyzer. Tracealyzer now goes into live mode and begins displaying the trace. Note that some views and features are not available in live mode.
3. When satisfied, select “Stop Recording” to end the live mode and enable all views and features in Tracealyzer.

Additional Notes

- During tracing, you may stop on break points, single-step and resume execution without interfering with the resulting trace. This is because the event timestamps are set based on the target system cycle counter, so halting the system “stops the time” for the recorder library. However, like always, halting the system may interfere with the target system as a whole, e.g. if this is controlling a physical system with real-time requirements.
- If you run code in user mode (i.e. with memory protection) you should also check that the ITM privilege flags is set accordingly. Otherwise you are not allowed to write to the ITM port from restricted user mode code, e.g. using vTracePrintF.
- In the Keil trace setup, we don’t enable “Trace Events” and “Timestamps” as these hardware-generated events are not included in the Keil ITMLOG output. The Tracealyzer recorder library provides timestamps as part of the event data and allows for Exception (ISR) tracing via explicit logging calls in the ISR handlers (vTraceStoreISRBegin and vTraceStoreISREnd, described in trcRecorder.h and in the Tracealyzer User Manual).
- If you do multiple traced runs within the same debug session, they will be appended to the same tracealyzer.psf unless this file is reset between the trace sessions. Appending multiple traces to the same file is not suitable in replay mode, as the file is replayed from the beginning when starting a new Tracealyzer session.
 - You may avoid appending multiple traces sessions to the same trace file by restarting the µVision debug session after each trace (i.e. exiting debug mode).
 - You may also make multiple (separate) traces within the same debug session, by following these steps to reset the trace file:

- Reset the target system and make sure the system is halted. Also make sure to stop Tracealyzer if still in live streaming mode.
- Select “Stop Recording” in Keil μ Vision.
- If you were viewing the trace live in Tracealyzer, a copy of the trace file is stored automatically. Otherwise, move/rename “tracealyzer.psf” to keep the previous trace.
- Select “Start Recording” in Keil μ Vision and run the debug session.
- Start recording in Tracealyzer.

Learning more

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

- Tracealyzer User Manual (Help -> User Manual)
- Tracealyzer “Getting Started” portal - <https://perceprio.com/gettingstarted>
- About ITM trace - <https://perceprio.com/2016/06/09/arm-itm/>
- About the recorder and custom streaming, <http://perceprio.com/2016/10/05/rtos-tracing>

For questions, please contact support@perceprio.com