Using Serial Wire Viewer (SWV) to Confirm CPU Speed

With Cortex®-M Processors and Keil™ MDK 5 toolkit.

Spring 2018 V 1.2 bob.boys@arm.com

The latest version of this document is here: www.keil.com/appnotes/docs/apnt_297.asp

Introduction:

This note describes how to take advantage of the UART communications utilized in the CoreSight™ Serial Wire Viewer to determine the CPU frequency. SWV Data trace frames are output on the Serial Wire Output (SWO) at a bit rate that is a ratio of the CPU core clock. A divider is adjustable to configure the bit rate of the UART in the range of the debug adapter.

The debugger (such as Keil µVision®) sets the divider in the processor and also configures the debug adapter (such as a Keil ULINK™2) to the same frequency of the SWO output. When these frequencies are the same, valid data trace frames will be visible in the Trace Records window. If the frequencies are different, there may be no frames or invalid (garbage) frames displayed in the Trace Records window. A signal will be output on the SWO pin that can be measured with an oscilloscope.

From this, we can calculate the CPU frequency knowing the divider ratio and the measured output period on the SWO pin.

Serial Wire Viewer (SWV) is a data trace feature found on many ARM Cortex-M3, M4, M7, M23, and M33 processors. Coretx-M0 and Cortex-M0+ do not have SWV. The SWV frames can be sent out either the one pin SWO (Serial Wire Output) pin or the 4 bit Trace Port. We will use this SWO pin in UART mode. It is timed by the CPU clock.

The UART baud rate must be specified and if this frequency is correct as evidenced by proper operation, the CPU frequency is easily confirmed. SWO Manchester mode or the 4 bit Trace Port cannot be used as the frequency is embedded in the signal.

Limitation: This method will give a rough approximation to the CPU frequency. The divider has a large granularity and small measurement is not possible. Recall a UART will still work properly if the two frequencies are slightly different.

TIP: If you want to display your CPU (Core Clock) frequency in a graphical and real-time format: consider using a ULINK*plus* and the new Keil System Analyzer window. www.keil.com/ulinkplus/.

Debug Adapters:

This method works with a Keil ULINK2, ULINK*plus*, ST-Link V2 or a Segger J-Link connected to your target processor. All these use the UART mode of SWV. ST-Link V2 is installed on many ST evaluation boards and a stand-alone version is available. The J-Link can be on or off board. Any J-Link V6 or later including J-Link Lite will work. The Atmel SAM-ICE (blue case) works with Atmel Cortex-M processors. Atmel is now Microchip.

Since the ULINK*pro* uses either Manchester mode or the 4 bit Trace Port (both have significant SWV advantages), it cannot be used with this technique. A CMSIS-DAP adapter will not work but it will with the new version found in CMSIS 5. For CMSIS 5 source code see https://github.com/ARM-software/CMSIS_5.

Software:

The free evaluation version of Keil MDK (MDK-Lite) can be used. This note assumes you have MDK and the Software Pack for your processor is installed. See the Getting Started Guide: www.keil.com/gsg

Serial Wire Viewer (SWV) must be configured. Instructions are on the next page if you do not already have this working.

µVision®, can connect to a target in either JTAG or SWD (Serial Wire Debug) mode. Since the SWO pin is multiplexed with the JTAG TDO pin, JTAG mode cannot be used or a conflict occurs. SWD (SW in µVision) must be selected. SWD is a two pin alternative to JTAG. They are functionally the same.

Frequency Measurement (Using an oscillosope):

An oscilloscope is used to measure the frequency (or actually the period) of the UART output. Since the SWO pin is usually less than 10 MHz, an expensive scope is not needed. A modest USB scope will do. The SWO waveform will have pulses of varying width and you need to measure the time of the narrowest pulse. A frequency counter is probably not useful.

SystemCoreClock Global Variable:

Programs that are CMSIS compatible will have a file system_*CPUname*.c This file often contains PLL code. A global variable SystemCoreClock is created in this file which contains the CPU frequency. The function SystemCoreClockUpdate() is often used for configure purposes. You can enter SystemCoreClock in a Watch window to display it. SystemCoreClock is a calculated value and not measured. If there is a bug in system_*CPUname*.c or the PLL code, this value can be wrong. You can use SWV as described in this note to confirm SystemCoreClock is correct.

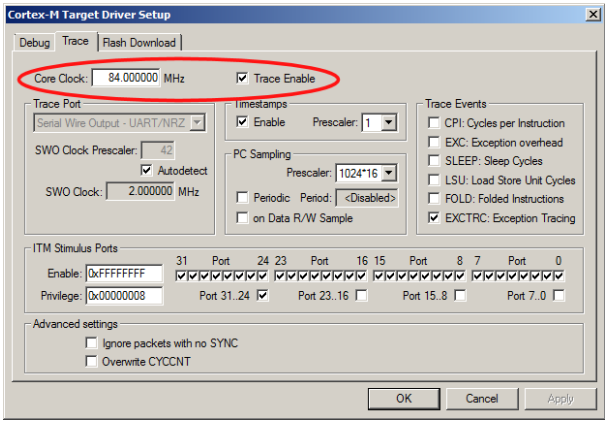
2) Configuring Serial Wire Viewer (SWV): *If you have SWV working, skip this page.*

Note: SWV must be configured and at least one type of SWV frame must be output on the SWO pin for measurement.

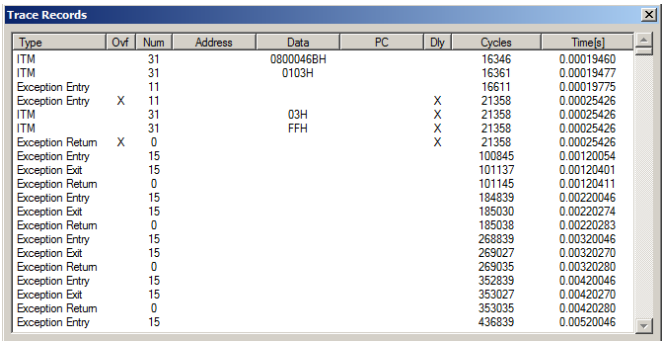
Important Things to Know about SWV:

1. You must know the exact CPU frequency to successfully configure the SWO pin in UART mode. This note helps you find this CPU frequency even if SWV is not working correctly.

Configure the SWV Trace:



1. Stop the processor and exit Debug mode.
2. Select Target Options or ALT-F7 and select the Debug tab. Select Settings: on the right side of this window. Confirm SW is selected. SW selection is mandatory for SWV. JTAG will not work.
3. Select the Trace tab.
4. In the Trace tab, select Trace Enable. Set Core Clock: to the frequency the CPU is running. Make your best estimate if this is unknown.
5. If your program is not using any interrupts or you are unsure, select Periodic to activate PC Samples. Enabling Periodic guarantees there will be frames on the SWO pin.
6. You can optionally unselect ITM 31 and 0.
7. Leave everything else at default.
8. Click OK twice to return to the main µVision menu.
9. Enter Debug mode. No need to recompile.
10. Click RUN The program will run as before.



Confirm SWV is working:

1. Select View/Trace/Records or click on the arrow beside the Trace icon and select Records.
2. The Trace Records window will open up: (if using a J-Link, you must stop the program to update the Trace window)
3. If you see Exception 15, ITM 31 or PC Samples (if selected) frames, then SWV is working correctly. You might not see all of these frames depending on your program. See the Trace Records window above.
4. If you see valid trace frames then ***the Core Clock: box probably contains the correct CPU speed.***

TIP: You can double-click in the Trace Records window to clear it.

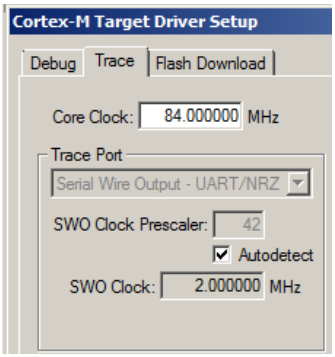
5. If you see nothing or any ITM other than 0 or 31, the most probable cause is the Core Clock: is likely incorrect. Check SystemCoreClock and enter this value into Core clock:. Try a simple multiple of it in Core Clock:
6. If you can see a waveform on the SWO pin, this can be used to calculate the CPU frequency. See step 5 below.

If you have trouble:

1. Make sure the Core Clock: is entered with a value as close as possible to its potential actual vaue.
2. A few boards need a solder bridge or a jumper connected to get SWV working. This connects the SWO pin.
3. A few boards might need an initialization file to configure I/O pins. This file is provided with the µVision examples.
4. Cycle the power to the board and restart µVision.
5. Check for a signal on the SWO pin with an oscilloscope – this pin is found on the JTAG/SWD connector. See www.keil.com/coresight/coresight-connectors/ In order for this method to work, a waveform must be present on the SWO pin even if no trace frames are visible in the Trace Data window. Calculate the frequency on the next page.
6. If your program is not using exceptions, try enabling Periodic (PC Samples) in the trace setup window and try again.

Using SWV to Determine and/or Confirm CPU Speed:

A waveform must be present on the SWO pin containing SWV frames in UART mode. The SWO output comes from a UART in the processor when used with a ULINK2, ULINK-ME, ST-LINK/V2 or a J-Link. The SWO Clock: frequency is the UART speed. This method provides accuracy expected when using a UART.



How it Works:

1. Select Target Options and select the Debug tab. Select Settings:
2. Select the Trace tab as shown here:
1. Enter what you believe is the correct CPU frequency in Core Clock:
2. µVision takes this frequency and calculates a SWO Clock Prescaler value to create a SWO Clock frequency within the range of your selected debug adapter.
3. In this case, µVision calculated a Prescaler of 42 that is appropriate and the resulting SWO Clock: frequency will be 2 MHz.
   a. The SWO Clock Prescaler is a CoreSight register in the processor.
   b. The actual SWO Clock can be measured with an oscilloscope on the SWO pin.
   c. **SWO Clock = Core Clock: / SWO Clock prescaler**. In this example: 84/42= 2 MHz.
   d. This information plus the measured SWO pin frequency, you can calculate what Core Clock: actually is.

Steps to take:

1. Connect an oscilloscope to the SWO pin and obtain a display of the SWO signal. Run the program.
2. Measure the period of the narrowest pulse and invert it. This gives the actual SWO Clock:
3. Calculate CPU frequency as shown with this equation: **SWO frequency * Prescaler = actual Core Clock:**
4. Stop the program and open the Trace tab as shown above right.
5. Enter your calculated Core Clock: value in the Core Clock: box.
6. Close these windows and enter Debug mode and run the program again.
7. If SWV now works correctly by displaying only valid frames in the Trace Records window as shown below, then the CPU frequency entered in Core Clock: is correct. You can probably assume Core Clock: is the CPU frequency.
8. Note the only ITM frames are 31. These (and 0) are the only valid types. The others are not used by µVision. In this example, Periodic is turned on: this is why PC Samples are displayed. The "X" in the Ovf and Dly columns are from SWV overflow. Sometimes the Cycles and Time(s) are the same from SWO pin overload. Ignore them here.

Unusual Situations: A very few processors have the SWO clock divided by2: **CPU clock / 2**. This information is usually not clearly mentioned in the device datasheet. If you have problems with this technique, try entering CPU/2 into the Core Clock: box. In a few rare cases, devices could have SWO output driven by a fixed clock source. In this case, the method described here will not work to determine the CPU frequency. You might be able to correlate this from the SWO speed.

Conclusions:

1. If you have valid and stable trace frames in the Trace Records window, the frequency in the Core Clock: is valid.
2. If you have no or erroneous frames in the Trace Records window, the Core Clock: is probably incorrect.
3. Erroneous frames include any ITM Ports other than 0 or 31, exceptions known not to be active or other garbage.
4. See an example on the next page.

TIP: It is possible to unselect Autodetect and set the SWO Clock Prescaler manually.

Trace Records

Type	Ovf	Num	Address	Data	PC	Dly	Cycles	Time[s]
ITM		31		0102H		X	27235	0.00037826
ITM		31		080003A7H		X	27235	0.00037826
ITM		31		0103H		X	27235	0.00037826
ITM		31		02H		X	27235	0.00037826
ITM		31		03H		X	27235	0.00037826
ITM		31		FFH		X	27235	0.00037826
Exception Return	X	0				X	27235	0.00037826
PC Sample					08000716H	X	27235	0.00037826
PC Sample					08000716H		36190	0.00050264
PC Sample					08000716H		52574	0.00073019
PC Sample					08000716H		68958	0.00095775
PC Sample					08000716H		85342	0.00118531
Exception Entry		15					86073	0.00119546
Exception Return	X	0				X	88429	0.00122818
PC Sample					08000716H		101726	0.00141286
PC Sample					08000716H		118110	0.00164042
PC Sample					08000716H		134494	0.00186797
PC Sample					08000716H		150878	0.00209553
Exception Entry		15					158072	0.00219544
Exception Exit		15					158364	0.00219950

Example:

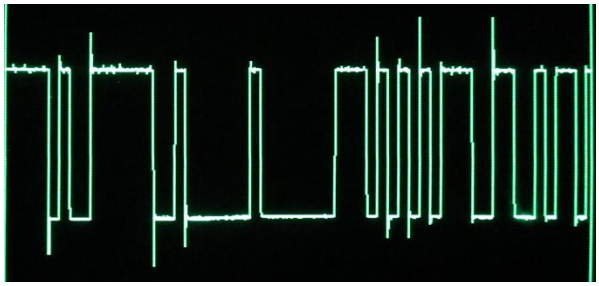
1. With a µVision project, Core Clock: was set to 60 MHz. The actual CPU speed was 84 MHz.
2. The resulting Trace Records window is shown below: Note the ITM 24, 4, 20 frames and so on are bogus. The Data Write data has a bogus value in the Data column. The Cycles and Time(s) columns are mostly all the same value. It is safe to assume the Core Clock: setting is probably incorrect.
3. Connect an oscilloscope to the SWO pin on the JTAG/SWD connector and as carefully as possible measure the period of the shortest pulse in the SWO signal.

Trace Records

Type	Ovf	Num	Address	Data	PC	Dly	Cycles	Time[s]
ITM		24		FFH		X	1759577039	22.06499194
ITM		4		FFC1H		X	1759577039	22.06499194
ITM		4		FFC1H		X	1759577039	22.06499194
ITM		4		BFC1H		X	1759577039	22.06499194
ITM		4		BFC1H		X	1759577039	22.06499194
Data Write			20000000H	25B02008H		X	1759577039	22.06499194
ITM		20		0DA0H			1759577042	22.06499199
ITM		23		BABFH			1759577042	22.06499199
ITM		23		A8B9H			1759577042	22.06499199
ITM		23		BAB8H			1759577042	22.06499199
ITM		7		FEH			1759577042	22.06499199

Calculations:

1. Period of shortest pulse = 353 nsec.
2. Prescaler value is set to 30 by µVision as shown in the SWO Clock Prescaler window in the Trace tab.
3. (1/(353 nsec)) * 30 = 84.98 MHz.
4. Round to 85 MHz and enter this in Core Clock: and the Trace Records now works normally. An example Trace Records window is shown below:
5. Displayed in the Trace Records below are Systick Timer (15) and PC Sample frames. There are no ITM frames in this case. This is because they unselected in this example. No bogus ITM frames are created since the Core Clock: value is now correct.
6. With this example, you can assume the CPU clock frequency is 85 MHz.
7. Shown here is the SWO signal on the oscilloscope:
   It is approximately 3.3 VP-P.



TIP: For CoreSight debug connector specifications see:
www.keil.com/coresight/coresight-connectors/

Trace Records

Type	Ovf	Num	Address	Data	PC	Dly	Cycles	Time[s]
Exception Return		0				X	2159901452	29.99863128
PC Sample					08000716H		2159912922	29.99879058
PC Sample					08000716H		2159929306	29.99901814
PC Sample					08000716H		2159945690	29.99924569
PC Sample					08000716H		2159962074	29.99947325
Exception Entry		15					2159970354	29.99958825
Exception Exit		15					2159970643	29.99959226
Exception Return		0				X	2159972194	29.99961381
PC Sample					08000716H		2159978458	29.99970081
PC Sample					08000716H		2159994842	29.99992836
PC Sample					08000716H		2160011226	30.00015592
PC Sample					08000716H		2160027610	30.00038347
Exception Entry		15					2160042354	30.00058825
Exception Exit		15					2160042643	30.00059226
Exception Return		0				X	2160047276	30.00065661
PC Sample					08000716H	X	2160047276	30.00065661
PC Sample					08000716H		2160060378	30.00083858
PC Sample					08000716H		2160076762	30.00106614
PC Sample					08000716H		2160093146	30.00129369
PC Sample					08000716H		2160109530	30.00152125

CoreSight Definitions: It is useful to have a basic understanding of these terms:

Cortex-M0 and Cortex-M0+ may have only features 2) and 4) plus 11), 12) and 13) implemented. Cortex-M3, Cortex-M4 and Cortex-M7 can have all features listed implemented. MTB is normally found on Cortex-M0+. It is possible some processors have all features except ETM Instruction trace and the trace port. Consult your specific datasheet.

1. **JTAG:** Provides access to the CoreSight debugging module located on the Cortex processor. It uses 4 to 5 pins.
2. **SWD:** Serial Wire Debug is a two pin alternative to JTAG and has about the same capabilities except Boundary Scan is not possible. SWD is referenced as SW in the µVision Cortex-M Target Driver Setup.
   The SWJ box must be selected in ULINK2/ME or ULINK*pro*. Serial Wire Viewer (SWV) must use SWD because the JTAG signal TDO shares the same pin as SWO. The SWV data normally comes out the SWO pin or Trace Port.
3. JTAG and SWD are functionally equivalent. The signals and protocols are not directly compatible.
4. **DAP:** Debug Access Port. This is a component of the ARM CoreSight debugging module that is accessed via the JTAG or SWD port. One of the features of the DAP are the memory read and write accesses which provide on-the-fly memory accesses without the need for processor core intervention. µVision uses the DAP to update Memory, Watch, Peripheral and RTOS kernel awareness windows while the processor is running. You can also modify variable values on the fly. No CPU cycles are used, the program can be running and no code stubs are needed. You do not need to configure or activate DAP. µVision configures DAP when you select a function that uses it. Do not confuse this with CMSIS_DAP which is an ARM on-board debug adapter standard.
5. **SWV:** Serial Wire Viewer: A trace capability providing display of reads, writes, exceptions, PC Samples and printf.
6. **SWO:** Serial Wire Output: SWV frames usually come out this one pin output. It shares the JTAG signal TDO.
7. **Trace Port:** A 4 bit port that ULINK*pro* uses to collect ETM frames and optionally SWV (rather than SWO pin).
8. **ITM:** Instrumentation Trace Macrocell: As used by µVision, ITM is thirty-two 32 bit memory addresses (Port 0 through 31) that when written to, will be output on either the SWO or Trace Port. This is useful for printf type operations. µVision uses Port 0 for printf and Port 31 for the RTOS Event Viewer. The data can be saved to a file.
9. **ETM:** Embedded Trace Macrocell: Displays all the executed instructions. The ULINK*pro* provides ETM. ETM requires a special 20 pin CoreSight connector. ETM also provides Code Coverage and Performance Analysis. ETM is output on the Trace Port or accessible in the ETB (ETB has no Code Coverage or Performance Analysis).
10. **ETB:** Embedded Trace Buffer: A small amount of internal RAM used as an ETM trace buffer. This trace does not need a specialized debug adapter such as a ULINK*pro*. ETB runs as fast as the processor and is especially useful for very fast Cortex-A processors. Not all processors have ETB. See your specific datasheet.
11. **MTB:** Micro Trace Buffer. A portion of the device internal user RAM is used for an instruction trace buffer. Only on Cortex-M0+ processors. Cortex-M3/M4 and Cortex-M7 processors provide ETM trace instead.
12. **Hardware Breakpoints:** The Cortex-M0+ has 2 breakpoints. The Cortex-M3, M4 and M7 usually have 6. These can be set/unset on-the-fly without stopping the processor. They are no skid: they do not execute the instruction they are set on when a match occurs. The CPU is halted before the instruction is executed.
13. **Watchpoints:** Both the Cortex-M0, M0+, Cortex-M3, Cortex-M4 and Cortex-M7 can have 2 Watchpoints. These are conditional breakpoints. They stop the program when a specified value is read and/or written to a specified address or variable. There also referred to as Access Breaks in Keil documentation.

Additional Serial Wire Viewer (SWV) Information:

Serial Wire Viewer data (frames) are output on the 1 pin SWO. This pin is located on 13 on the 20 pin legacy connector (the BIG one). It is also available on the 10 or 20 pin CoreSight Debug connectors. SWO is on pin 6 on the 10 pin and pin 14 on the 20 pin. SWO is multiplexed with JTAG TDO pin. This means SWD (Serial Wire Debug) must be used and not JTAG mode. This is easily set in µVision. SWD = SW in µVision.

SWO is only one pin and it can be challenging to send a large amount of SWV data through it. A ULINK*pro* using Manchester mode on the SWO pin is more efficient. For even more throughput, ULINK*pro* can output SWV on the 4 bit Trace Port. This port is available on most Cortex-M3, M4 and M7. Cortex-M0 does not have SWV nor the Trace Port but has DAP read/write.

Document Resources: See www.keil.com

Books:

1. ***NEW!*** **Getting Started with MDK 5:** Obtain this free book here: www.keil.com/mdk5/
2. There is a good selection of books available on ARM: www.arm.com/support/resources/arm-books
3. µVision contains a window titled Books. Many documents including data sheets are located there.
4. **The Definitive Guide to the ARM Cortex-M0/M0+** by Joseph Yiu. Search the web for retailers.
5. **The Definitive Guide to the ARM Cortex-M3/M4** by Joseph Yiu. Search the web for retailers.
6. **Embedded Systems: Introduction to Arm Cortex-M Microcontrollers** (3 volumes) by Jonathan Valvano
7. MOOC: Massive Open Online Class: University of Texas: http://users.ece.utexas.edu/~valvano/

Application Notes:

1. ***NEW!*** ARM Compiler Qualification Kit: Compiler Safety Certification: www.keil.com/safety
2. Using Cortex-M3 and Cortex-M4 Fault Exceptions www.keil.com/appnotes/files/apnt209.pdf
3. CAN Primer using Keil MCB170: www.keil.com/appnotes/files/apnt_247.pdf
4. Segger emWin GUIBuilder with µVision™ www.keil.com/appnotes/files/apnt_234.pdf
5. Porting an mbed™ Project to Keil MDK™ www.keil.com/appnotes/docs/apnt_207.asp
6. MDK-ARM™ Compiler Optimizations www.keil.com/appnotes/docs/apnt_202.asp
7. GNU tools (GCC) for use with µVision https://launchpad.net/gcc-arm-embedded
8. Barrier Instructions http://infocenter.arm.com/help/topic/com.arm.doc.dai0321a/index.html
9. Cortex-M Processors for Beginners: http://community.arm.com/docs/DOC-8587
10. Lazy Stacking on the Cortex-M4: www.arm.com and search for DAI0298A
11. Cortex Debug Connectors: www.keil.com/coresight/coresight-connectors
12. FlexMemory configuration using MDK www.keil.com/appnotes/files/apnt220.pdf
13. Sending ITM printf to external Windows applications: www.keil.com/appnotes/docs/apnt_240.asp
14. ***NEW!*** Migrating Cortex-M3/M4 to Cortex-M7 processors: www.keil.com/appnotes/docs/apnt_270.asp
15. ***NEW!*** ARMv8-M Architecture Technical Overview www.keil.com/appnotes/files/apnt_291.pdf
16. ***NEW!*** Determining Cortex-M CPU Frequency using SWV www.keil.com/appnotes/docs/apnt_297.asp

Useful ARM Websites:

1. ***NEW!*** CMSIS 5 Standards: https://github.com/ARM-software/CMSIS_5 and www.keil.com/cmsis/
2. ARM and Keil Community Forums: www.keil.com/forum and http://community.arm.com/groups/tools/content
3. ARM University Program**:** www.arm.com/university. Email: university@arm.com
4. mbed™: http://mbed.org **Keil Training and Events:** www.keil.com/events

For more information: www.keil.com/mdk5/selector

Sales In Americas: sales.us@keil.com or 800-348-8051. **Europe/Asia:** sales.intl@keil.com +49 89/456040-20

Keil Technical Support in USA: support.us@keil.com or 800-348-8051. Outside the US: support.intl@keil.com.

Global Inside Sales Contact Point: Inside-Sales@arm.com ARM **Keil World Distributors:** www.keil.com/distis