



How to debug with Serial Wire Viewer tracing on STM32MP15

How to debug with Serial Wire Viewer tracing on STM32MP15



Contents

1. How to debug with Serial Wire Viewer tracing on STM32MP15	3
2. STM32MP15 resources	5



This article does not intend to cover all STM32CubeIDE Serial Wire Viewer (SWV) capabilities; complete information is available in (UM2609), chapter 4. It only provides some setup information for STM32MP15 serie, debugging Cortex-M in *Production Mode*.

In that mode, the available console on the board (UART4) is used by Cortex-A Linux. The clock tree is managed by Linux and 'Trace clock' needed to setup SWO is available from Linux console with command:

```
awk 'ck_trace/{print $5}' /sys/kernel/debug/clk/clk_summary
```

In order to test, let's modify main.c file from a generated project with a looping variable 'i', as depicted hereafter.

The screenshot shows the STM32CubeIDE interface. The Project Explorer on the left shows the project structure for 'dk2-mx', with 'main.c' selected under the 'Src' folder. The main editor displays the code for 'main.c', which includes a loop that increments a variable 'i' and calls 'HAL_Delay(300);' every 35 iterations. The console at the bottom shows the execution of the command 'awk 'ck_trace/{print \$5}' /sys/kernel/debug/clk/clk_summary' on the target device, resulting in the output '133250000'.

```

workspace_0.0.0.0-gerrit186909-p1 - dk2-mx_CM4/Core/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug Project Explorer main.c
100
101 /* USER CODE BEGIN SysInit */
102
103 /* USER CODE END SysInit */
104
105 /* Initialize all configured peripherals */
106 MX_GPIO_Init();
107 /* USER CODE BEGIN 2 */
108
109 /* USER CODE END 2 */
110
111 /* Infinite loop */
112 /* USER CODE BEGIN WHILE */
113 while (1)
114 {
115     /* USER CODE END WHILE */
116     i++;
117     i++;
118     i++;
119     i++;
120     HAL_Delay(300);
121     if (i>35)
122         i=0;
123     /* USER CODE BEGIN 3 */
124 }
125 /* USER CODE END 3 */
126 }
127
128 /**
129  * @brief System Clock Configuration
130  * @retval None
131  */
  
```

```

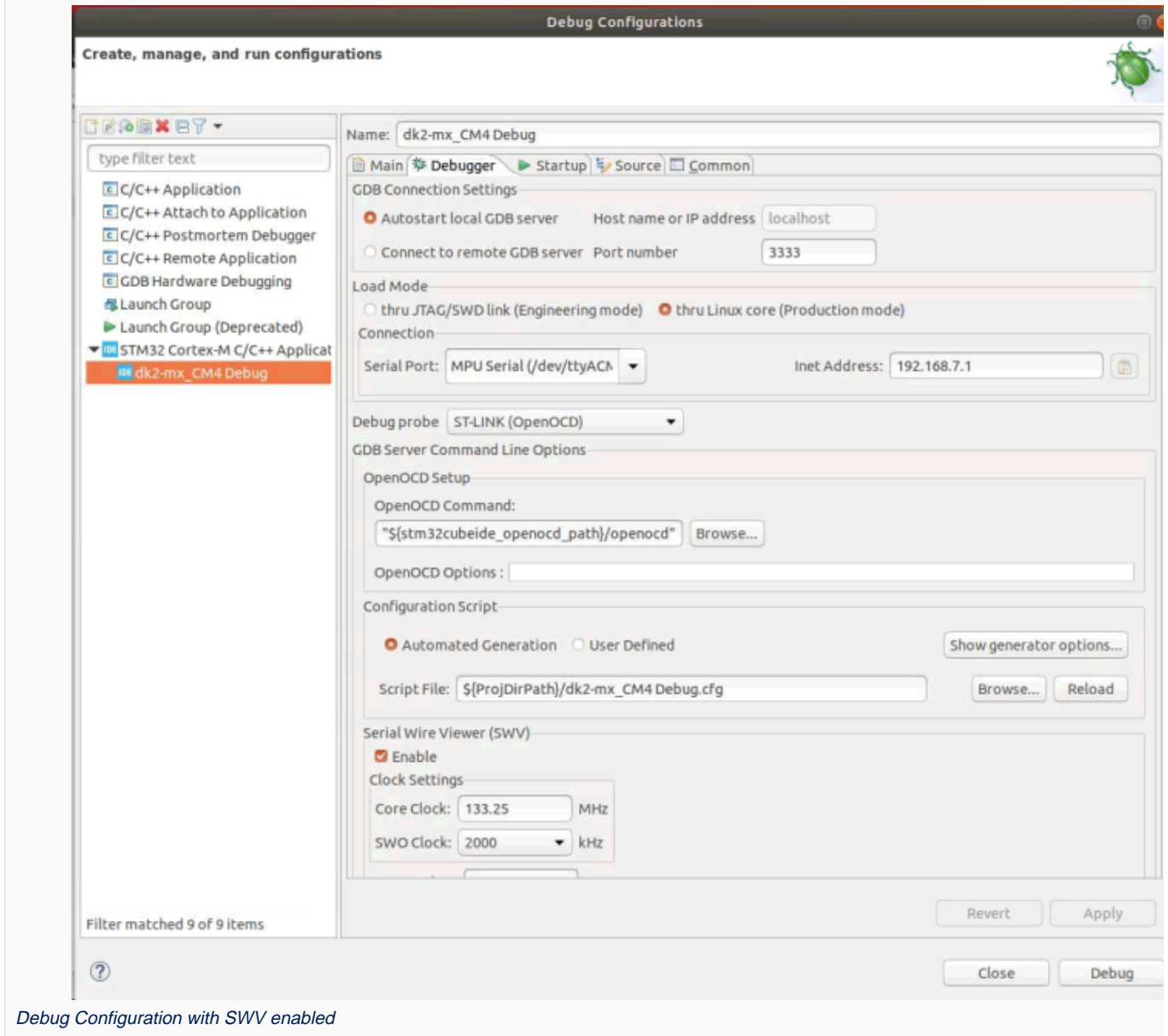
MPU Serial (CONNECTED)
root@stm32mp1:~# awk 'ck_trace/{print $5}' /sys/kernel/debug/clk/clk_summary
133250000
root@stm32mp1:~#
  
```

/dk2-mx_CM4/Core/Src/main.c Addr: 192.168.7.1

Trace clock from Linux console



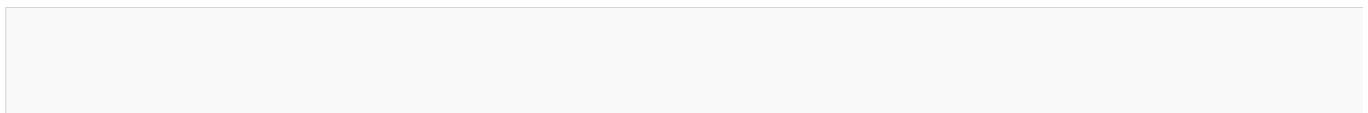
Then, setup debug configuration, enabling SWV and setting the clock: 133.25MHz here.

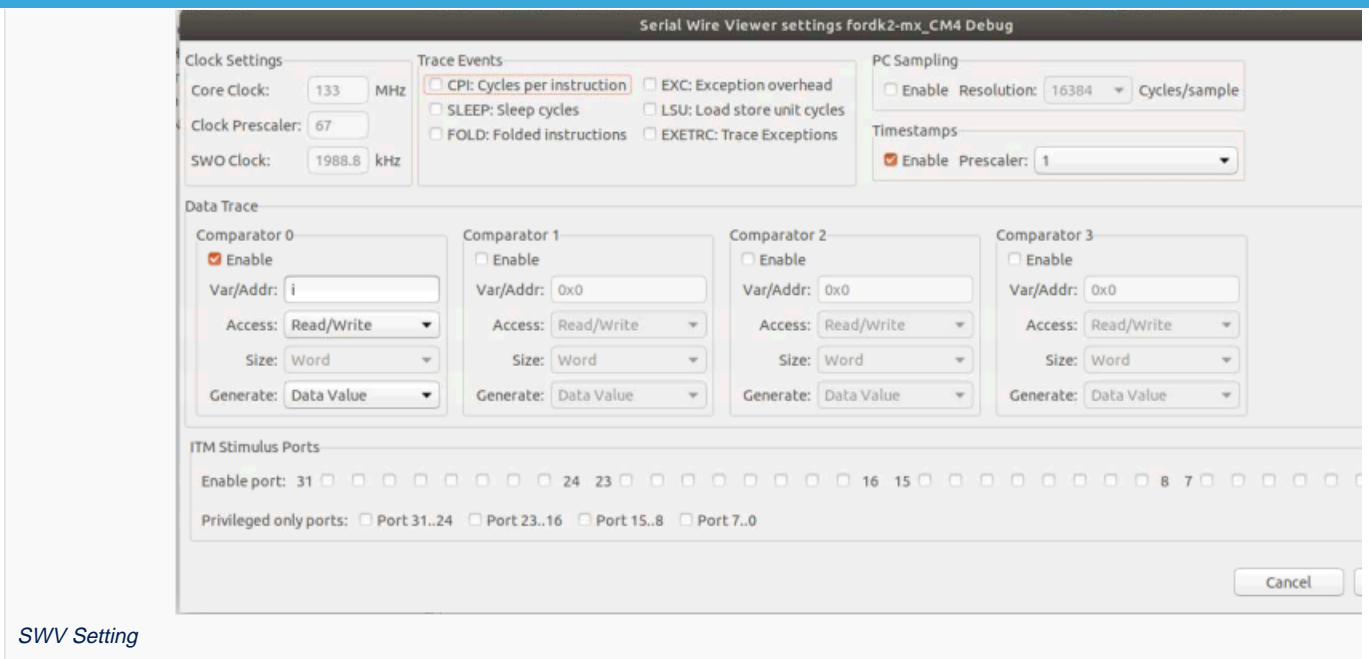


Debug Configuration with SWV enabled

Stopping Debug session, open Serial Wire Views: *Window > Show View > Other... > SWV > SWV Trace Log & SWV Data Trace.*

In *SWV Trace Log > Configure Trace* menu, setup *Comparator 0* in order to spy variable 'i'.



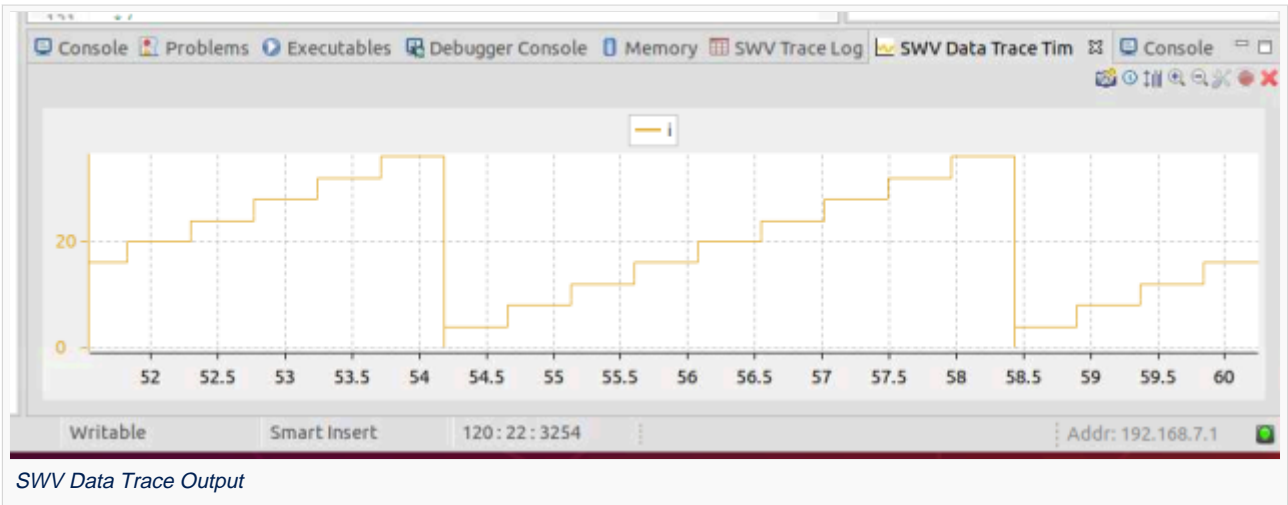


SWV Setting

Then start the trace



Resuming debug session gives inside SWV Data Trace view the corresponding graphic.



SWV Data Trace Output

Cortex®

Linux® is a registered trademark of Linus Torvalds.
 Stable: 17.11.2020 - 17:06 / Revision: 10.11.2020 - 07:49

Invalid target: no reviewed revision corresponds to the given ID.

Return to STM32MP15 resources.