

Embedded System Design Techniques™

Debugging Real-time Embedded Software – Hands-on

Session 4: Utilizing System Viewers and Trace Tools to Debug Firmware

July 14th, 2016
Jacob Beningo, CSDP

Course Overview

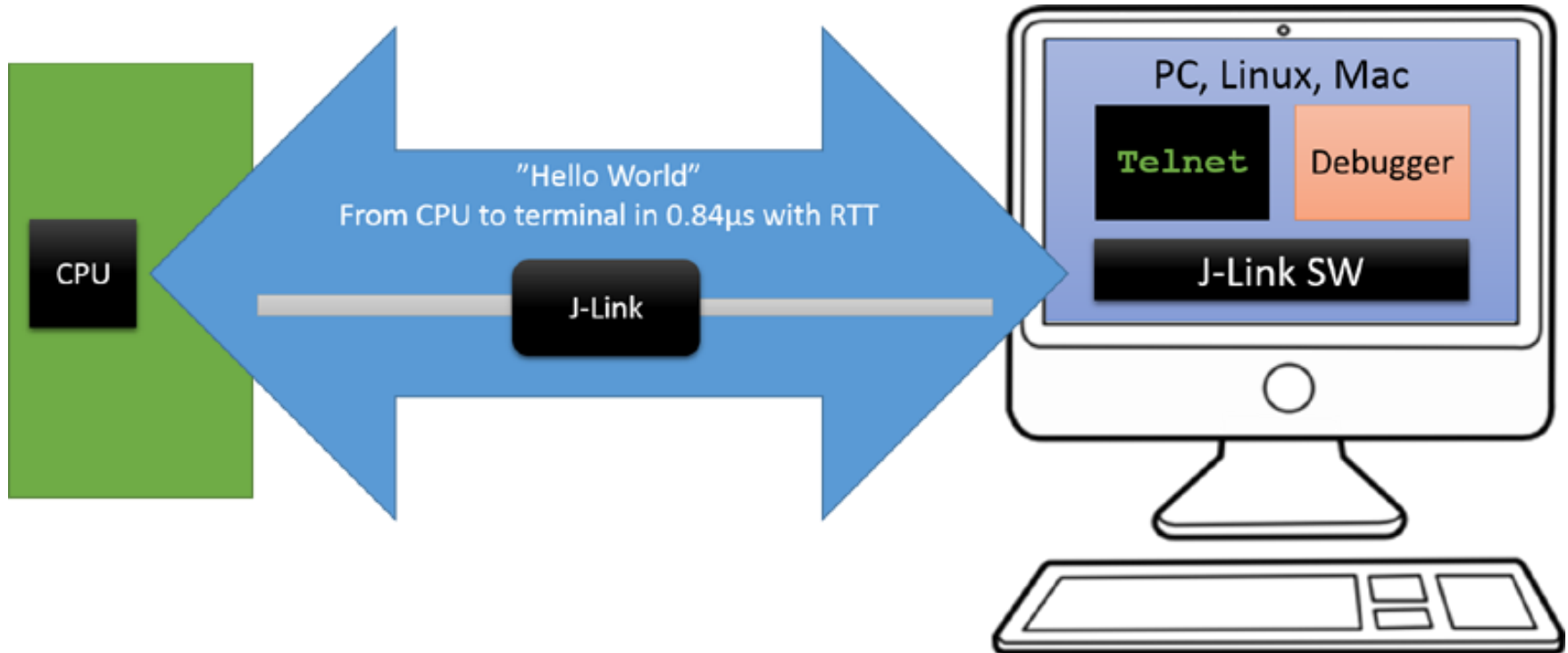
- Introduction to Debugging Real-time Embedded Systems
- Foundational Debugging Techniques
- Debugging the ARM Cortex-M Microcontroller
- **Utilizing Systems Viewers and Trace tools to Debug Firmware**
- Tips and Tricks for Debugging Embedded Systems

Session Overview

- Overview of RTT
- SystemView Setup
- SystemViewer Setup
- Analyzing the trace
- Percepio Tracealyzer



Overview of RTT



Source: https://www.segger.com/jlink-rtt.html#RTT_What_is_RTT

SystemView Setup

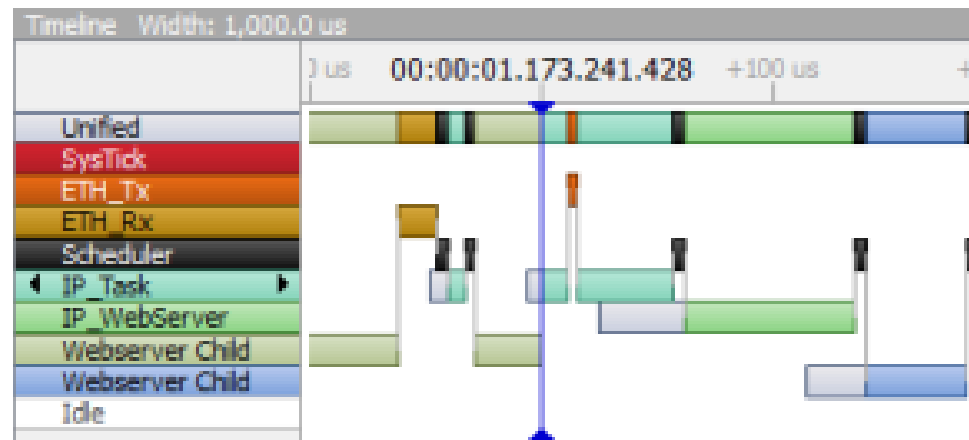
1 Download and Install the Tools

Segger SystemView

- MCUonEclipse Plugin
 - Download from Source Forge
- Raw source from Segger website

SystemViewer Utility

- Segger website

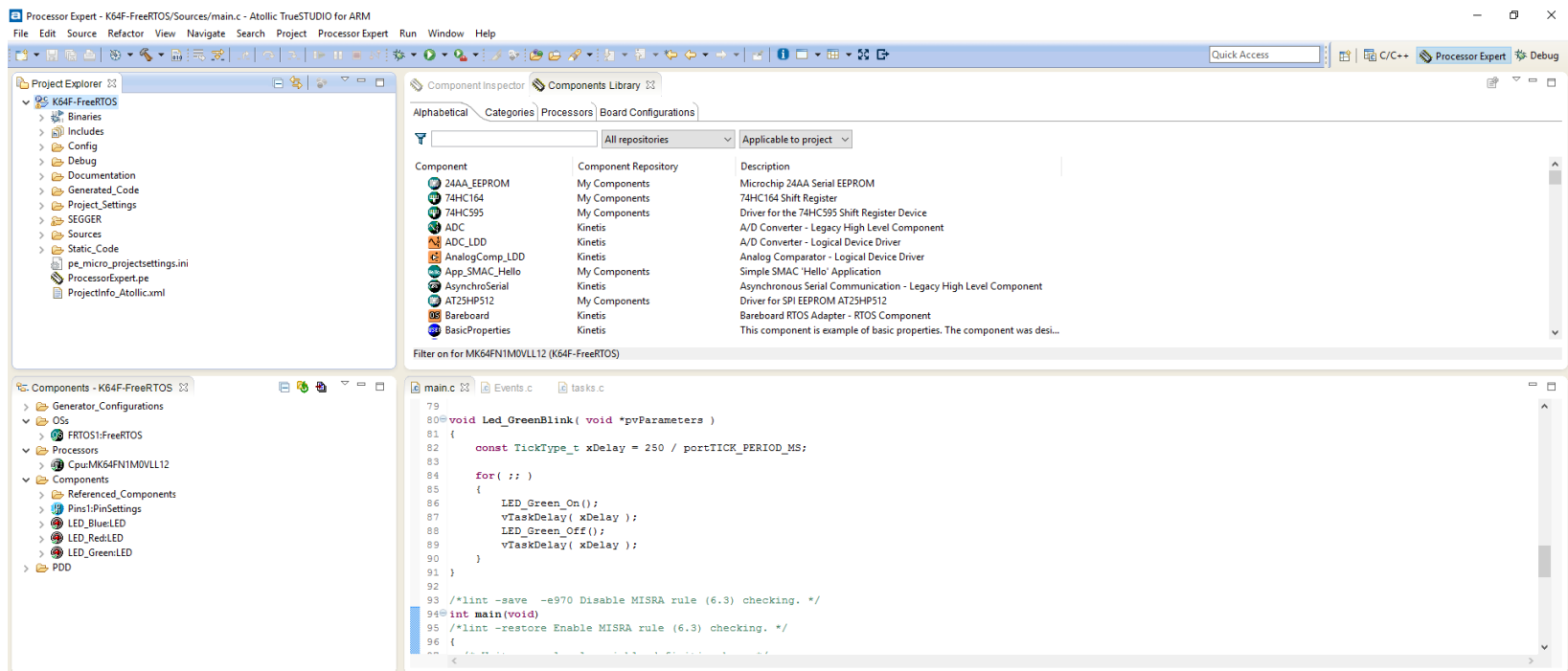


SystemView Setup

2 Create a Project

Project Requirements

- RTOS based
- Multiple Threads
- Use current project as a base



Presented by:

SystemView Setup

3 Add example code

```
void Led_BlueBlink( void *pvParameters )
{
    const TickType_t xDelay = 500 / portTICK_PERIOD_MS;

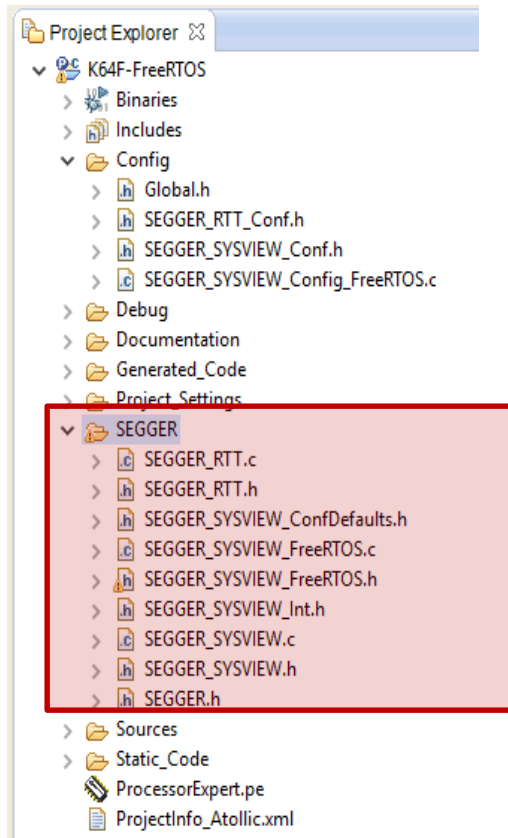
    for( ;; )
    {
        LED_Blue_On();
        vTaskDelay( xDelay );
        LED_Blue_Off();
        vTaskDelay( xDelay );
    }
}

void Led_GreenBlink( void *pvParameters )
{
    const TickType_t xDelay = 250 / portTICK_PERIOD_MS;

    for( ;; )
    {
        LED_Green_On();
        vTaskDelay( xDelay );
        LED_Green_Off();
        vTaskDelay( xDelay );
    }
}
```

SystemView Setup

4 Add SystemView Source code



Project should look like this!

SystemView Setup

5 Configure the project

main.c

```
52 /* User includes (#include below this line is not maintained by Processor Expert) */
53 #include "SEGGER_SYSVIEW.h"

103 /* Write your code here */
104 /* For example: for(;;) { } */
105 SEGGER_SYSVIEW_Conf(); /* Configure and initialize SystemView */
106 xTaskCreate(Led_BlueBlink, (const char* const)"led_blue", configMINIMAL_STACK_SIZE, 0, 1, 0);
107 xTaskCreate(Led_RedBlink, (const char* const)"led_red", configMINIMAL_STACK_SIZE, 0, 1, 0);
108 xTaskCreate(Led_GreenBlink, (const char* const)"led_green", configMINIMAL_STACK_SIZE, 0, 1, 0);
```

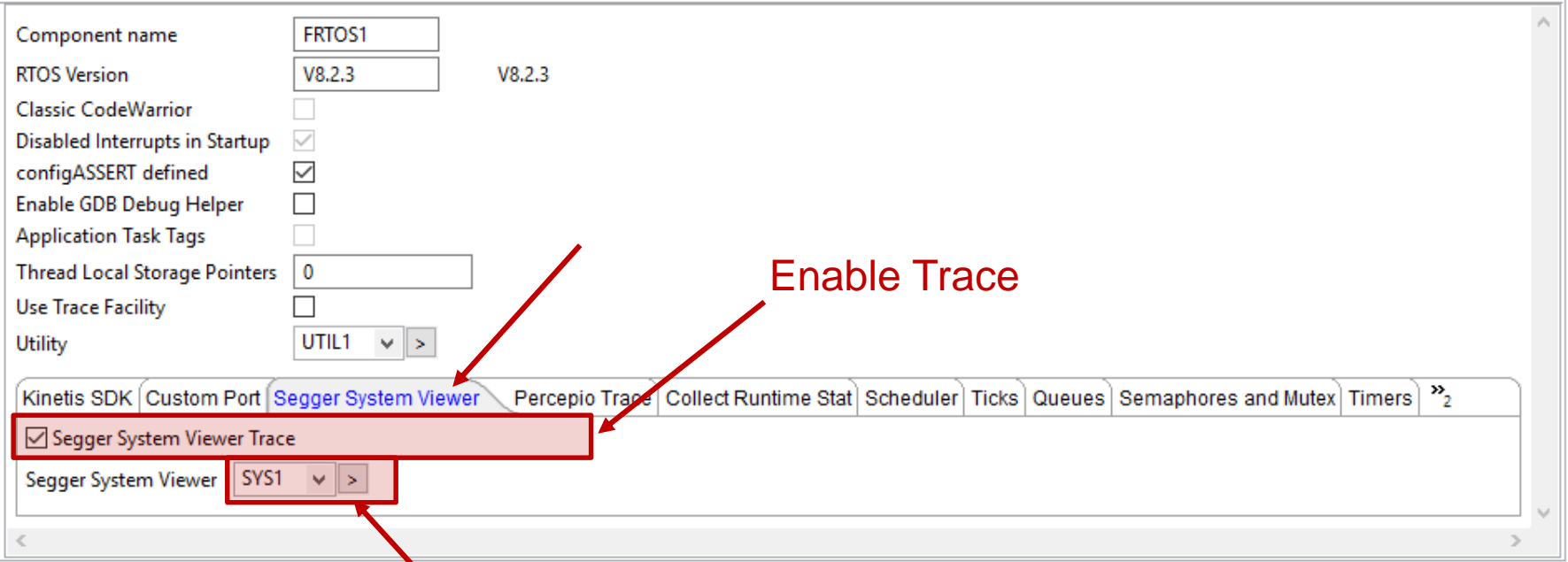
FreeRTOSConfig.h

```
165 #define configUSE_SEGGER_SYSTEM_VIEWER_HOOKS 1

250 #include "SEGGER_SYSVIEW_FreeRTOS.h"
```

SystemView Setup

5 Configure the project – The easy way

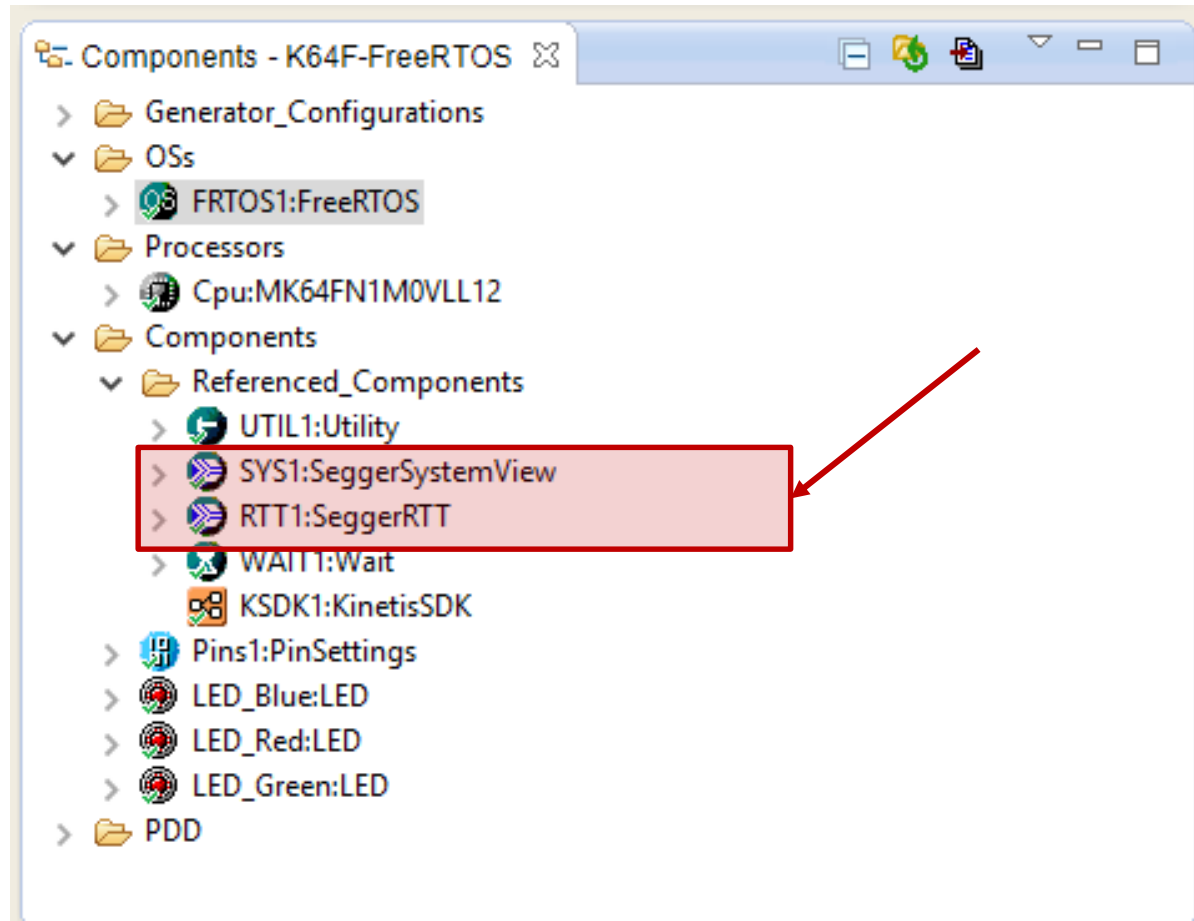


The screenshot shows the SystemView configuration window. The 'Component name' is set to 'FRTOS1' and the 'RTOS Version' is 'V8.2.3'. The 'Segger System Viewer Trace' checkbox is checked and highlighted with a red box. The 'Segger System Viewer' dropdown menu is also highlighted with a red box and set to 'SYS1'. Red arrows point to these elements with the following annotations:

- Enable Trace**: Points to the 'Segger System Viewer Trace' checkbox.
- Create a new component**: Points to the 'Segger System Viewer' dropdown menu.

Other visible options include: Classic CodeWarrior, Disabled Interrupts in Startup, configASSERT defined, Enable GDB Debug Helper, Application Task Tags, Thread Local Storage Pointers, Use Trace Facility, and Utility (UTIL1).

SystemView Setup



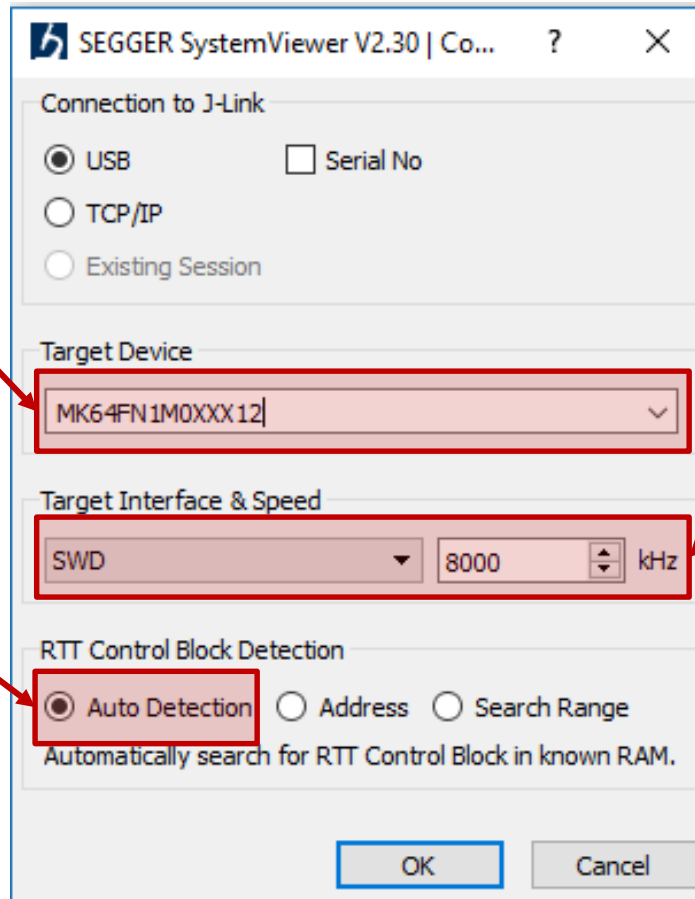
SystemView Setup

6 Run the SystemViewer

Must know target!

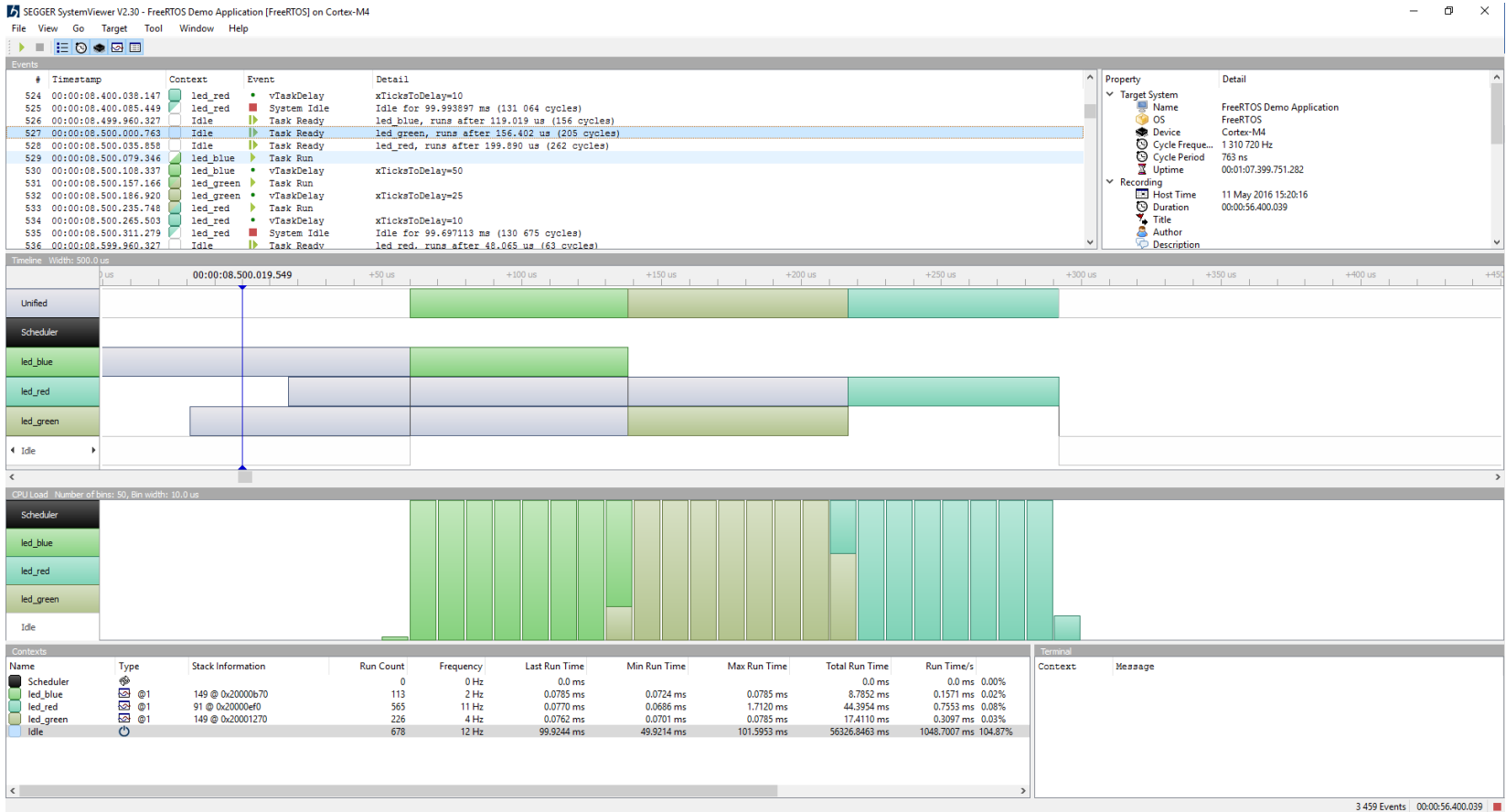
Easiest option

Match the SWD



SystemView Setup

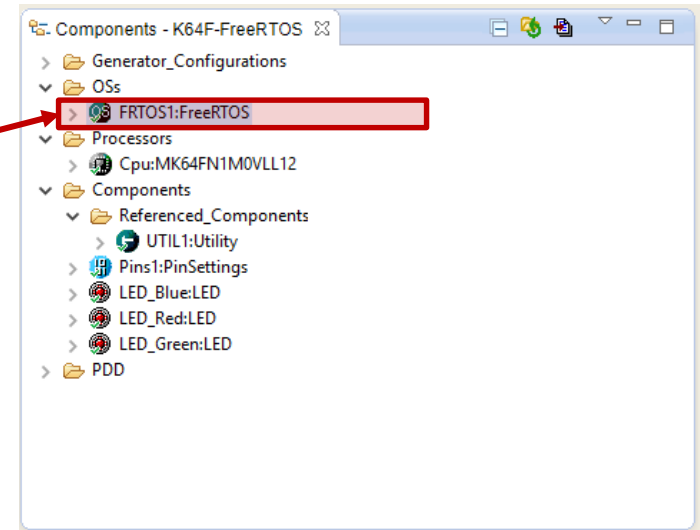
7 Review the results



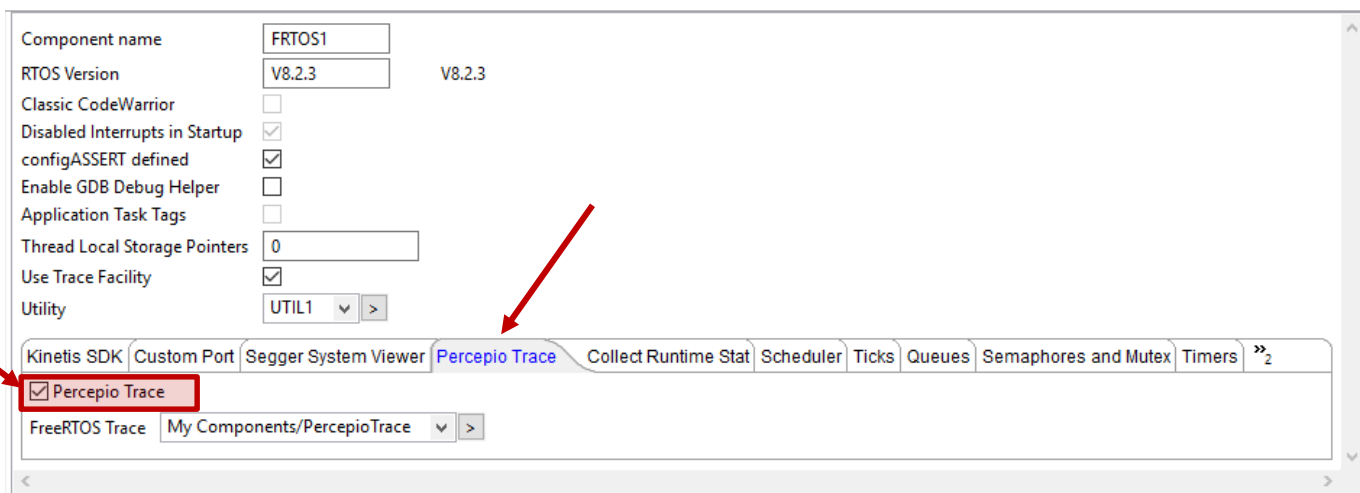
Presented by:

Percepio Tracealyzer

1 Setup Source



Enable



Percepio Tracealyzer

1 Setup Source

Enable

The screenshot shows the 'Streaming with RTT' configuration window in Percepio Tracealyzer. The 'Streaming with RTT' checkbox is checked and highlighted with a red box. A red arrow points from the word 'Enable' above to this checkbox. Below the checkbox, various configuration options are listed in a table-like format. The 'Segger RTT' dropdown menu is open, showing 'RTT1' selected and a red box around the right-pointing arrow button. A red arrow points from the text 'Configure RTT' to this button. The other configuration options are: Streaming Recorder Version (V3.0.2), Symbol Table Slots (30), Symbol Max Length (24), Object Data Slots (20), Ctrl Task Priority (1), Ctrl Task Stack Size (configMINIMAL_STACK_SIZE), Up Buffer Index (0), Up Buffer Size (1024), Down Buffer Index (0), and Down Buffer Size (32).

Streaming Recorder Version	V3.0.2	V3.0.2
Symbol Table Slots	30	
Symbol Max Length	24	
Object Data Slots	20	
Ctrl Task Priority	1	
Ctrl Task Stack Size	configMINIMAL_STACK_SIZE	
Segger RTT	RTT1	>
Up Buffer Index	0	
Up Buffer Size	1024	
Down Buffer Index	0	
Down Buffer Size	32	

Percepio Tracealyzer

1 Setup Source

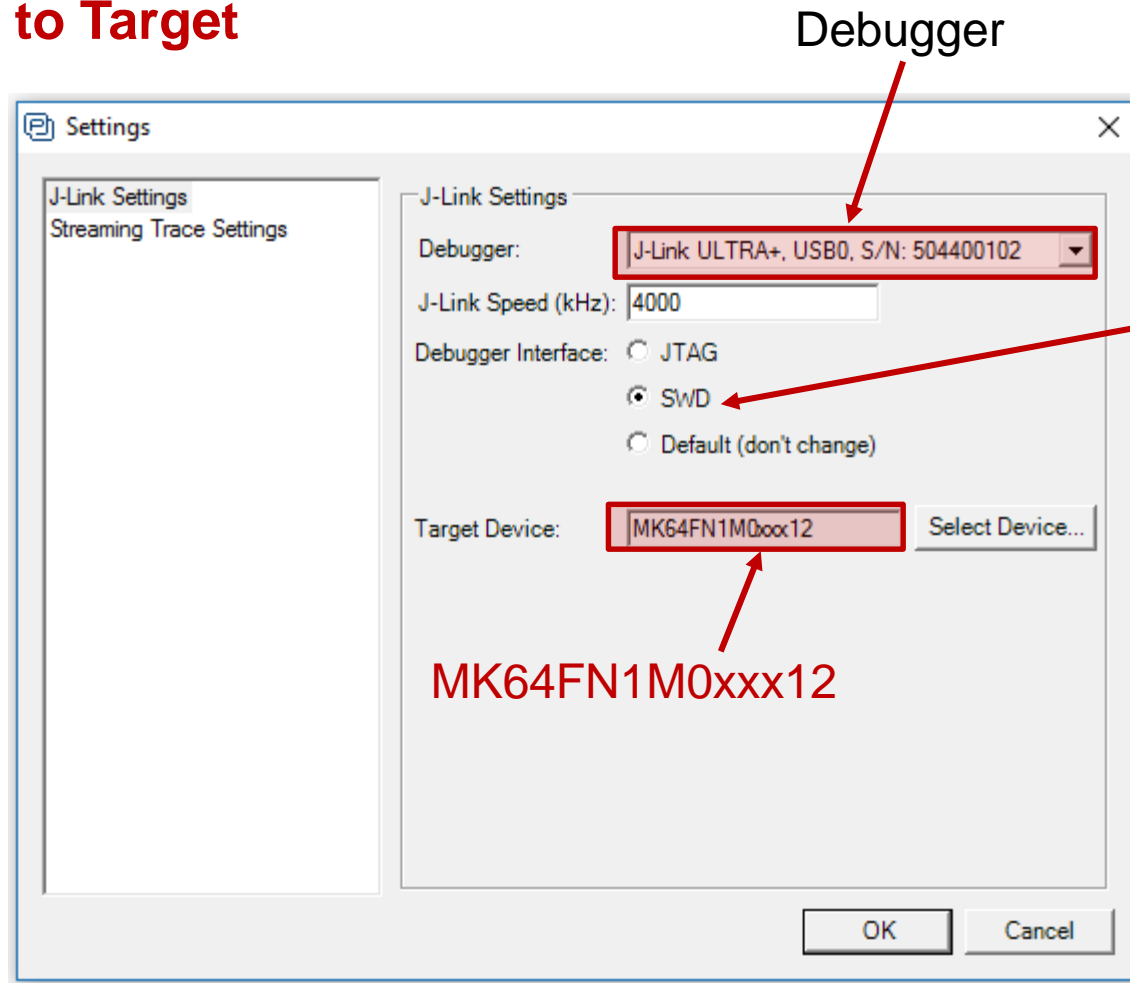
Component name	<input type="text" value="RTT1"/>
Version	<input type="text" value="V2.20a"/> V2.20a
Number of Up Channels	<input type="text" value="2"/>
Number of Down Channels	<input type="text" value="1"/>
Terminal (Channel 0)	
Up Buffer Size (Tx)	<input type="text" value="1024"/>
Up Buffer Mode	<input type="text" value="Skip (Default)"/> ▾
Down Buffer Size (Rx)	<input type="text" value="64"/>
Down Buffer Mode	<input type="text" value="Skip (Default)"/> ▾
Printf Buffer Size	<input type="text" value="64"/>
<input type="checkbox"/> Blocking Send	
Timeout (ms)	<input type="text" value="5"/>
Wait	<input type="text" value="WAIT1"/> ▾ >
Wait Time (ms)	<input type="text" value="1"/>
Kinetis SDK	<input type="text" value="KSDK1"/> ▾ >

Disable!

Don't forget to regenerate and recompile!

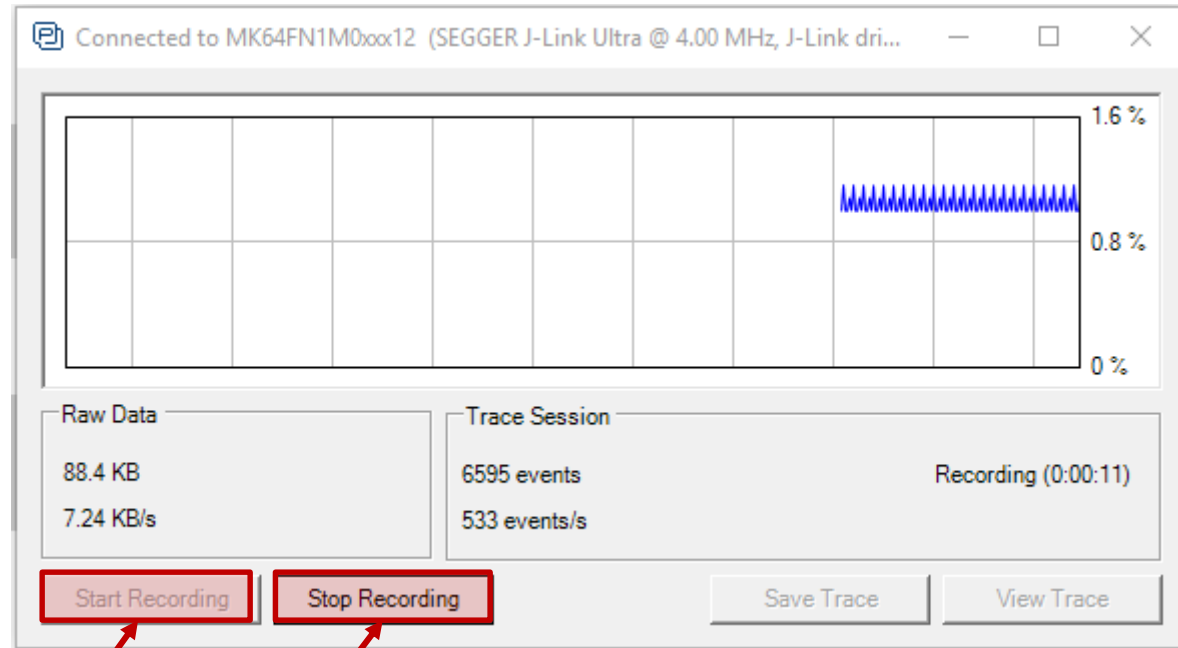
Percepio Tracealyzer

2 Connect to Target



Percepio Tracealyzer

3 Start Acquiring Trace

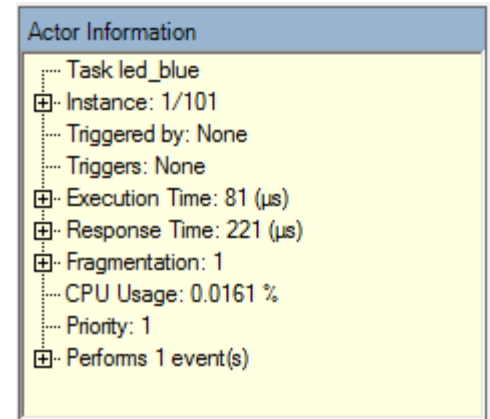
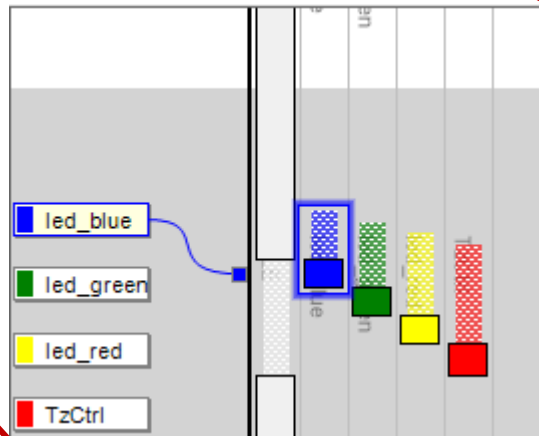


Start

Stop

Percepio Tracealyzer

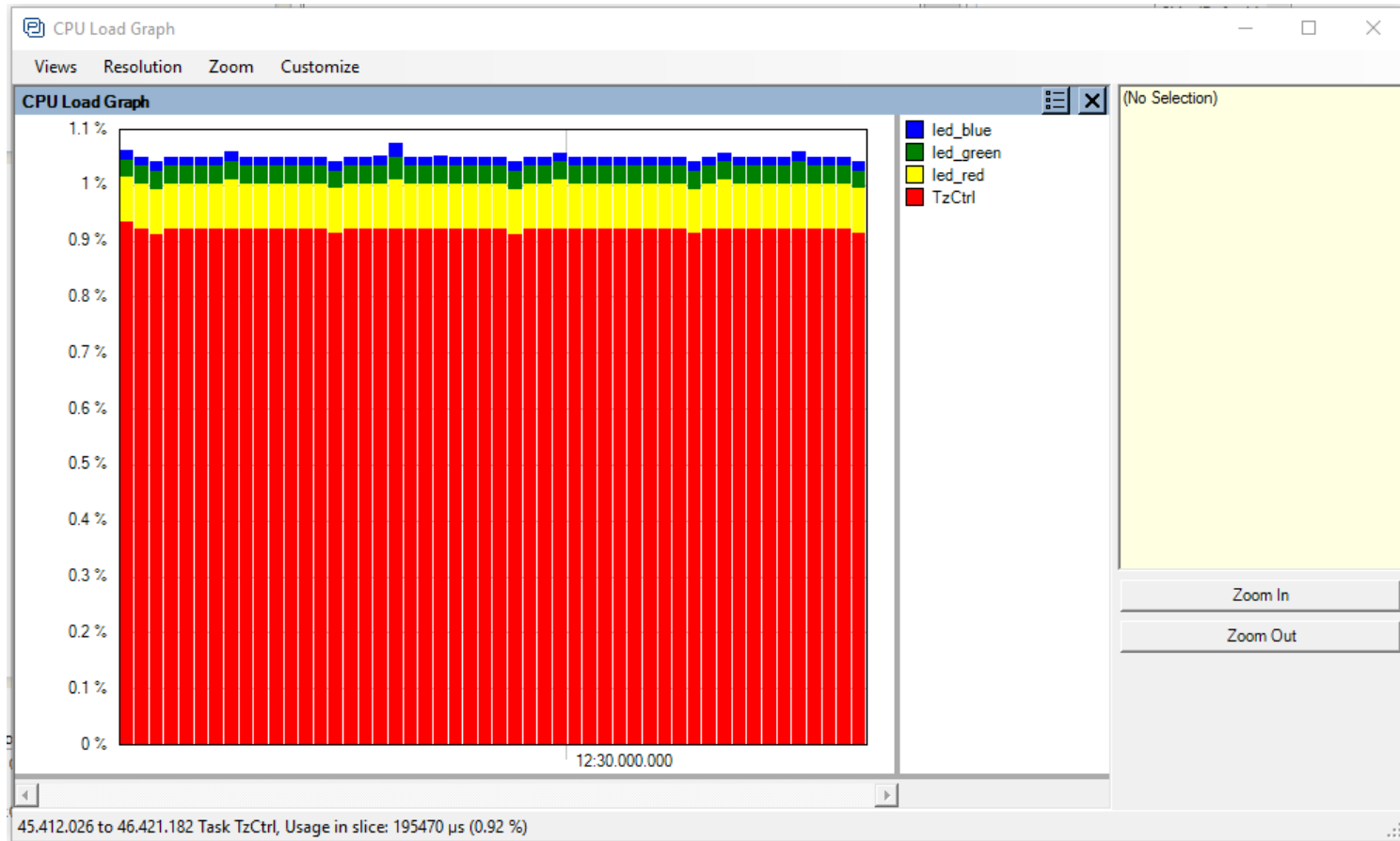
4 Analyze the Trace



Presented by:

Percepio Tracealyzer

4 Analyze Trace

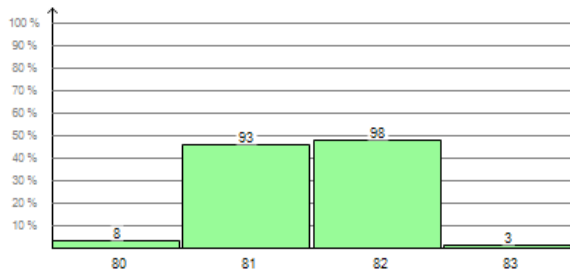


Percepio Tracealyzer

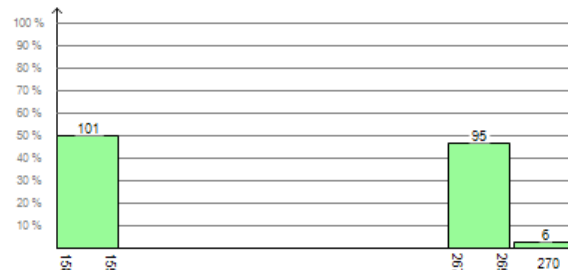
4 Analyze Trace

Actor	Priority		Count	CPU Usage %	Execution Time			Response Time			Periodicity			Separation			Fragmentation			
	Min	Max			Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	
Task IDLE	0	0	1	98.948	-	-	-	-	-	-	-	-	-	-	-	-	-	5048	5048	5048
Task led_blue	1	1	101	0.016	80	81	82	220	221	222	499.998	500.000	500.001	499.777	499.778	499.780	1	1	1	
Task led_green	1	1	202	0.033	80	81	83	158	213	270	249.965	250.000	250.034	249.697	249.786	249.876	1	1	1	
Task led_red	1	1	504	0.081	80	81	83	156	189	318	99.933	100.000	100.067	99.616	99.811	99.909	1	1	1	
Task TzCtrl	1	1	5047	0.923	0	92	111	43	148	378	7.810	10.000	10.100	7.699	9.851	9.962	1	1	1	

Task led_green - Execution Time



Task led_green - Response Time



Task led_green - Periodicity



Troubleshooting Trace

- Clean the project
- Rebuild the Processor Expert Components
- Build the entire project
- May need to modify includes in the generated components
- When changing PE Component, click on a new box, save all, then rebuild
- Review PE dependencies

Additional Resources

- Download Course Material for
 - Updated C Doxygen Templates (Sept 2015)
 - Example source code
 - Templates
 - YouTube Videos
- Microcontroller API Standard
- EDN Embedded Basics Articles
- Embedded Bytes Newsletter
 - <http://bit.ly/1BAHYXm>



From www.beningo.com under






- Blog > Debugging Realtime Embedded Software

The Lecturer – Jacob Beningo



Jacob Beningo
Principal Consultant

Social Media / Contact

-  : jacob@beningo.com
-  : 248-719-6850
-  : Jacob_Beningo
-  : Beningo Engineering
-  : JacobBeningo

EDN : Embedded Basics

CONSULTING

- Secure Bootloaders
- Code Reviews
- Architecture Design
- Real-time Software
- Expert Firmware Analysis

EMBEDDED TRAINING



www.beningo.com