

Embedded System Design Techniques™

Debugging Real-time Embedded Software – Hands-on

Session 1: Introduction to Debugging Real-time Embedded Systems

July 11th, 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

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

Jacobs CEC Courses

CEC 2013 – 2015

Fundamentals of Embedded Software (2013)

Mastering the Software Design Cycle (2014)

Python for Embedded Systems(2014)

Software Architecture Design (2014)

Baremetal C (2015)

Mastering the ARM Cortex-M Processor (2015)

Writing Portable and Robust Firmware in C (2015)

Design Patterns and the Internet (2015)

CEC 2016

Bootloader Design for MCUs
January 2016

Rapid Prototyping w/
Micro Python
May 2016

Debugging
July 2016

Professional Firmware
October 2016

Side Topics 2016

Real-Time Software
using Micro Python

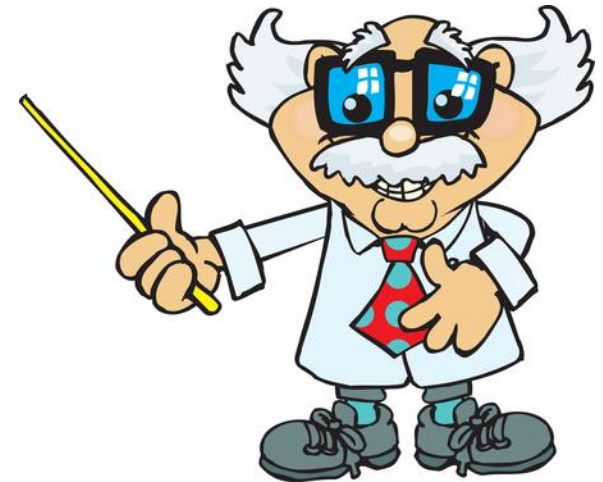
PROMO-PYTHON

Embedded Bytes
Newsletter

<http://bit.ly/1BAHYXm>

Session Overview

- The Greatest Development Challenge
- Overview of Debugging Techniques
- The K64F Freedom Board
- Test Bench Setup



The Greatest Development Challenge



The Greatest Development Challenge

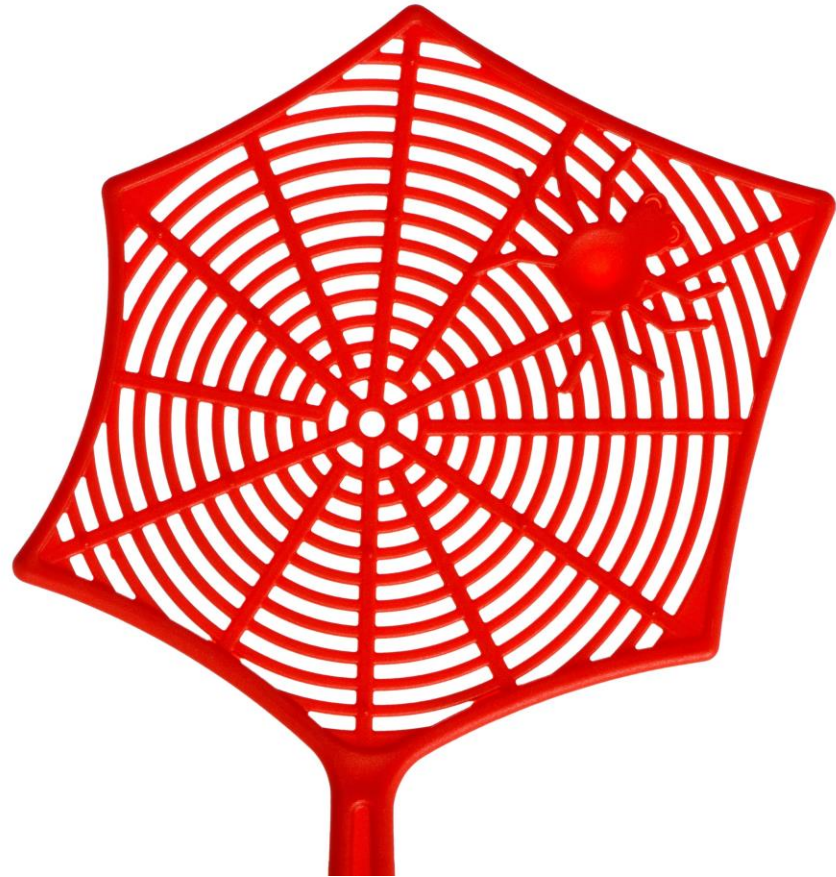
7 Tips to Squash Bugs

- Develop a software architecture
- Use coding standards such as MISRA-C
- Use a style guide (improve readability)
- Read the C standard
- Develop a disciplined approach
- Use regression testing
- Perform code reviews frequently

Overview of Debugging Techniques

Available Techniques:

- Basic Breakpoint
- Advanced Breakpoint
- Variable watch
- expressions
- printf
- assert
- Data watch
- Serial Wire Viewer
 - Statistical profiling
 - Data profiling
- System Trace
 - Task and data tracing
 - Instruction tracing
 - Branch detection

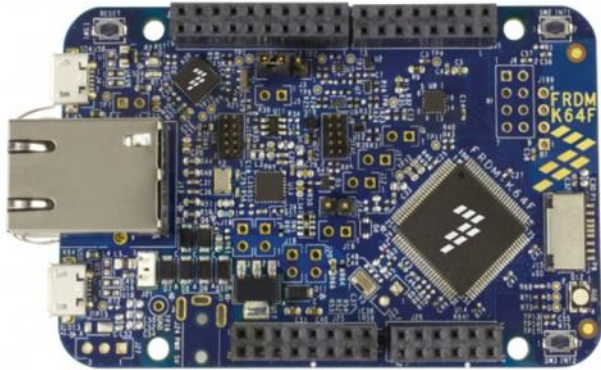


Overview of Debugging Techniques

7 Tips to Debug Software

- Change one line of code at a time
- Make a record of the attempted fixes
- Increase assertion density
- Avoid breakpoint debugging
- Discuss the problem out loud
- Go for a walk
- Sleep on the problem

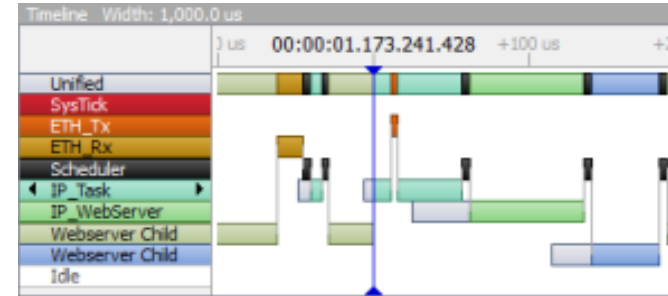
Course Tools



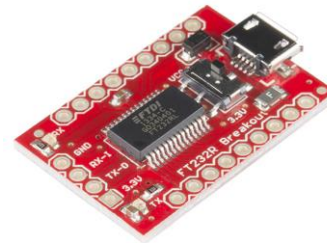
NXP K64F Freedom Board



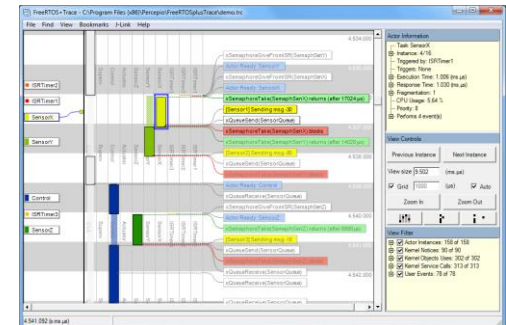
J-Link



Segger SystemViewer



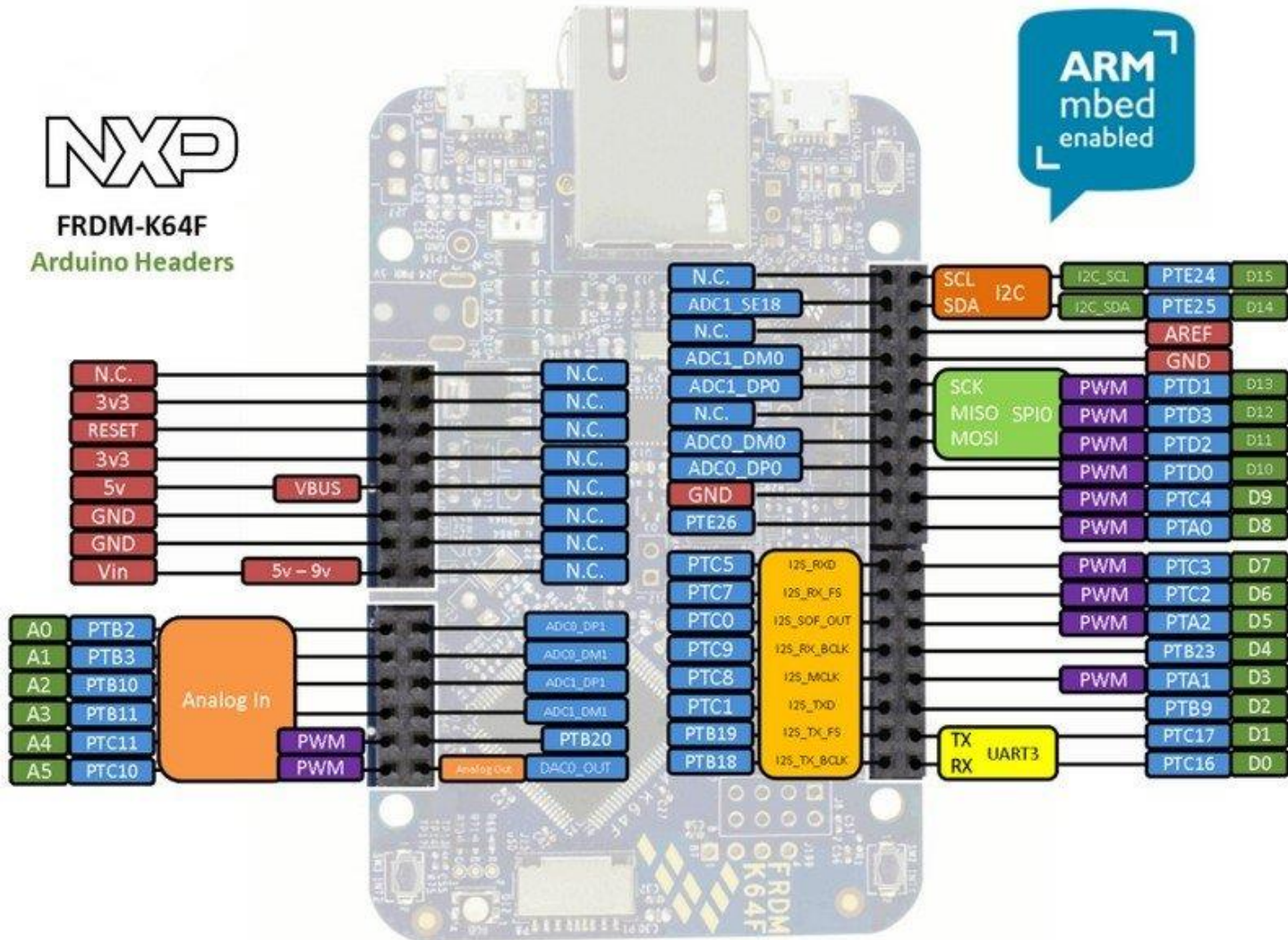
Uart-to-USB



Percepio Tracealyzer™

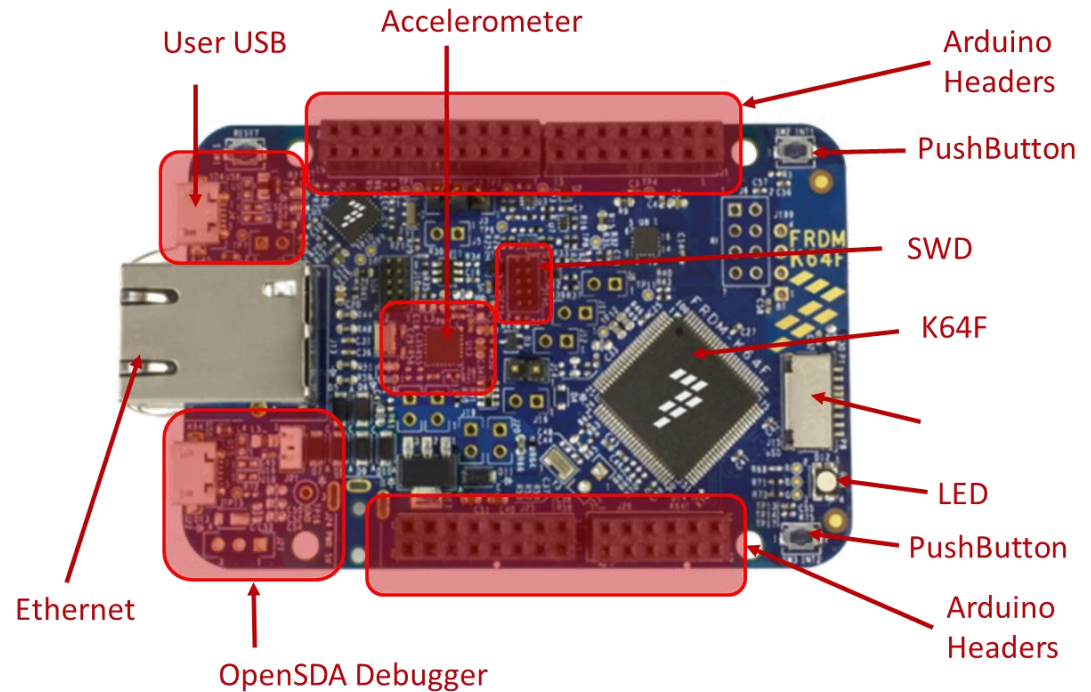
Presented by:

The K64F Freedom Board



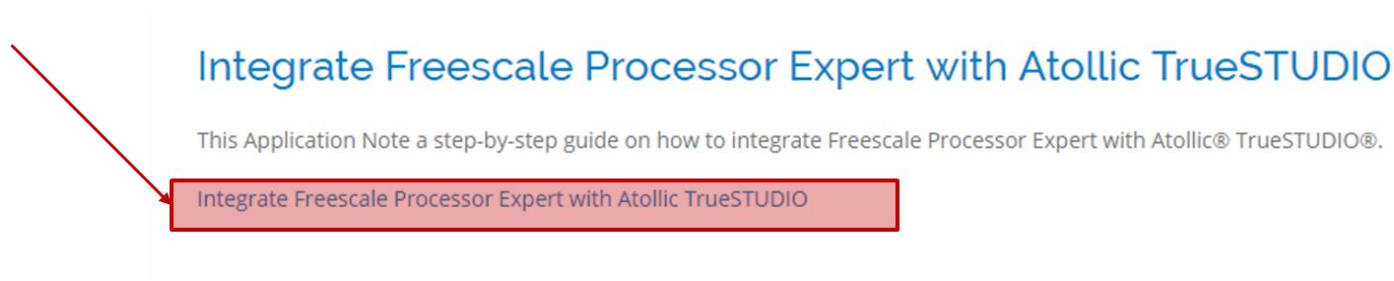
The K64F Freedom Board

- NXP K64F Freedom Board
 - ARM Cortex-M4
 - 120 MHz
 - 1 MB Flash
 - 256 KB RAM
 - Tri-Color LED
 - Accelerometer
 - Built-in Debugger
 - Ethernet
 - \$35



Test Bench Setup

- Install Atollic TrueSTUDIO
- Install **PEx plug-in** or SDK
 - Timor.atollic.com/resources/application-notes/

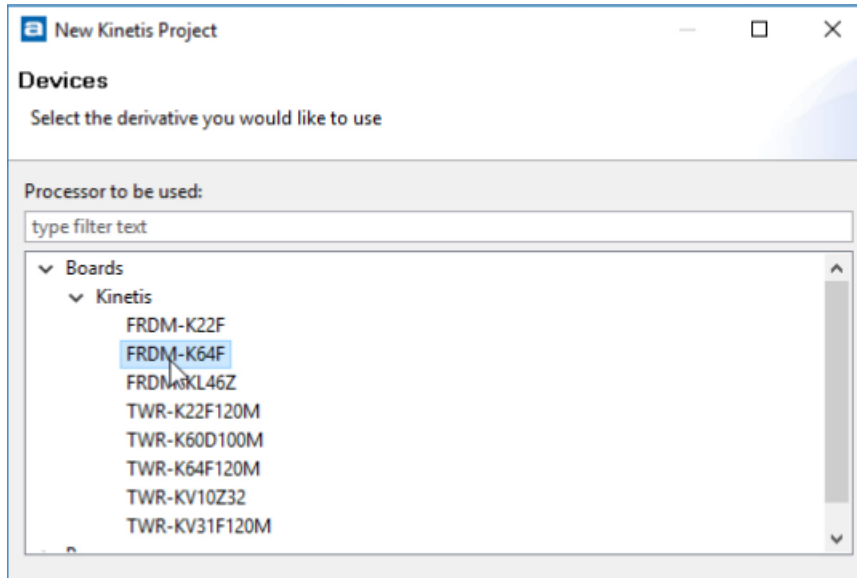


- Install MCUonEclipse Processor Expert Components

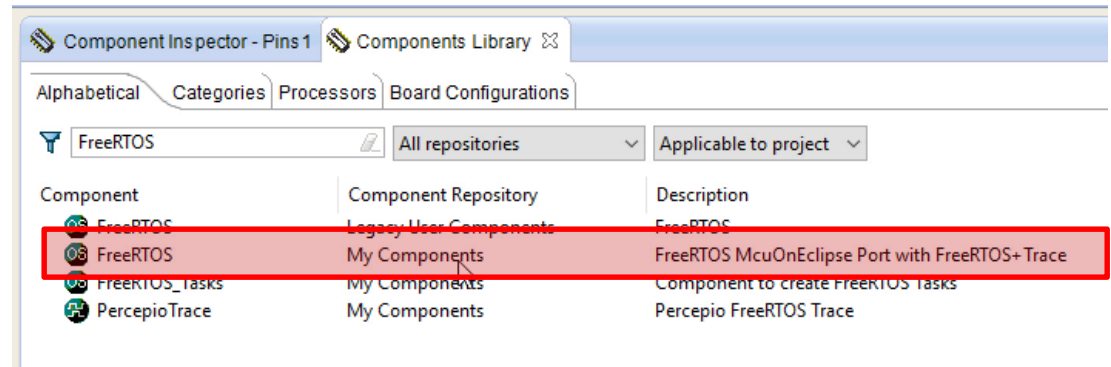
We are ready to create a project, setup FreeRTOS and blink some LEDs!

Test Bench Setup

1



2

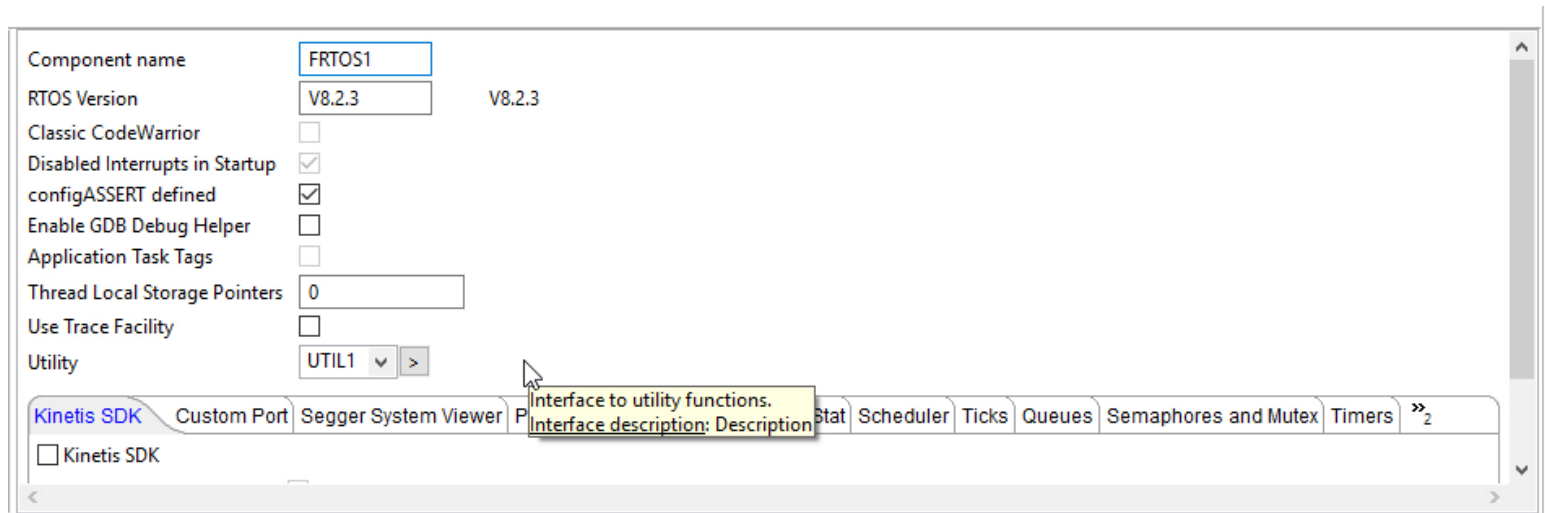


Test Bench Setup

3



4



Test Bench Setup

5

```
43 /* User includes (#include below this line is not maintained by Processor Expert) */
44 void Led_BlueBlink(void *pvParameters)
45 {
46     const TickType_t xDelay = 500 / portTICK_PERIOD_MS;
47     for(;;)
48     {
49         LED1_On();
50         vTaskDelay(xDelay);
51         LED1_Off();
52         vTaskDelay(xDelay);
53     }
54 }
55 }
```

6

```
59 /*lint -save -e970 Disable MISRA rule (6.3) checking. */
60 int main(void)
61 /*lint -restore Enable MISRA rule (6.3) checking. */
62 {
63     /* Write your local variable definition here */
64
65     /*** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! ***/
66     PE_low_level_init();
67     /*** End of Processor Expert internal initialization. ***/
68
69     /* Write your code here */
70     /* For example: for(;;) { } */
71     xTaskCreate(Led_BlueBlink, (const char* const)"led_blue", configMINIMAL_STACK_SIZE, 0, 1, 0);
```

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 Systems

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