Embedded System Design Techniques™

Debugging Real-time Embedded Software – Hands-on

Session 5: Tips and Tricks for Debugging Embedded Systems

July 15th, 2016

Jacob Beningo, CSDP

DesignNews

CEC CONTINUING EDUCATION CENTER

Presented by: Digi-Key ELECTRONICS

Course Overview

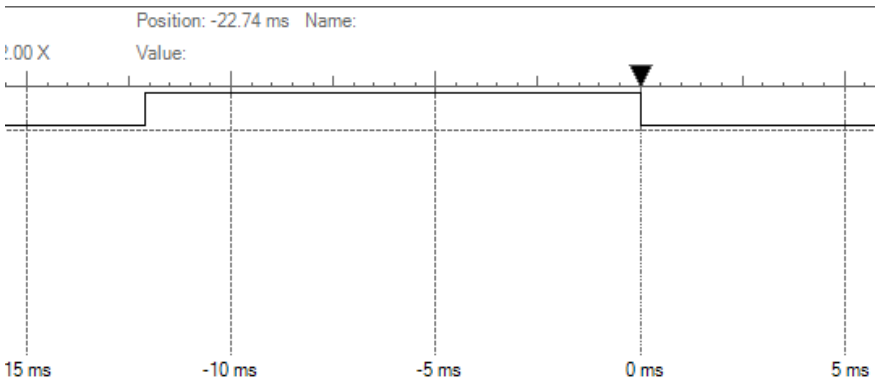
- Introduction to Debugging Real-time Embedded Systems
- Foundational Debugging Techniques
- Debugging the ARM Cortex-M Microcontroller
- Utilizing Systems Viewers and Trace tools to Debug Firmware
- **Tips and Tricks for Debugging Embedded Systems**

Session Overview

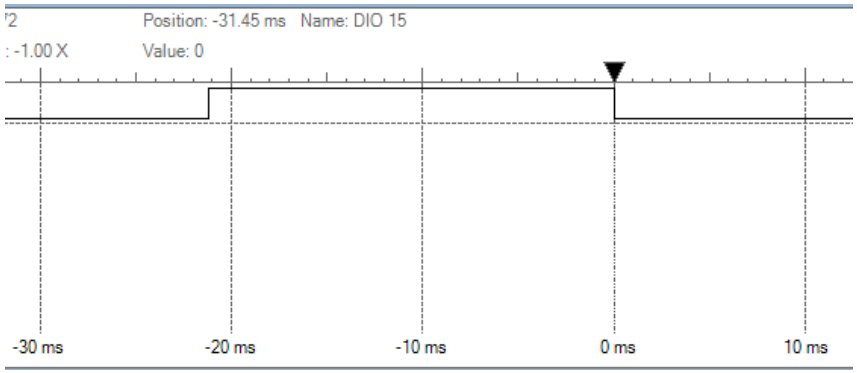
- Tips for Debugging Embedded Systems
  - printf with ITM
  - printf with RTT
  - Data Tracing
  - Analysis
- Course Review

Tip #1 – Use printf through ITM

- printf("Hello World!");
  - 12.5 ms

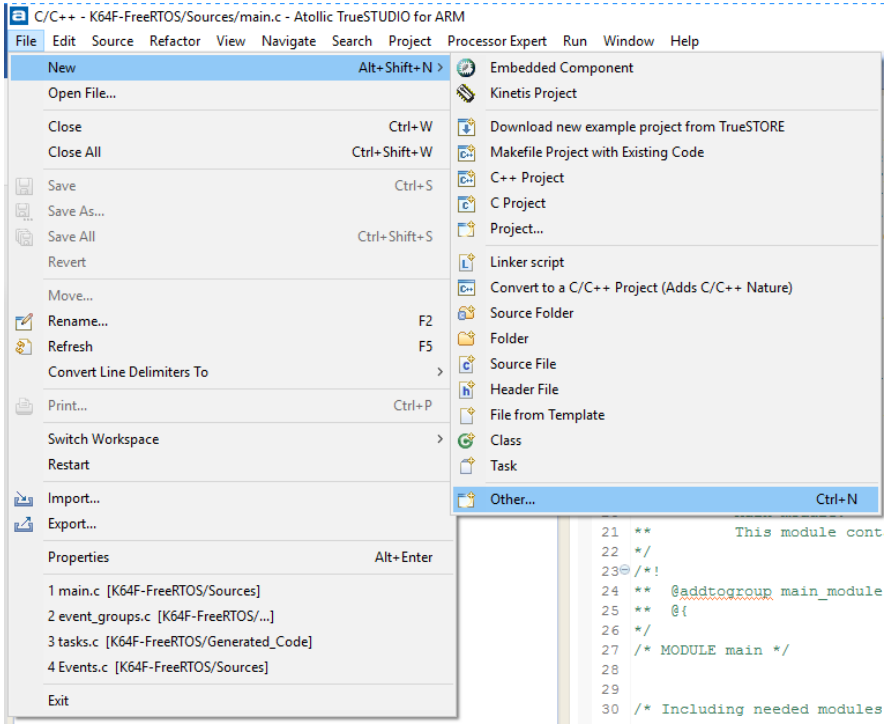


- printf("The System state is %d", State);
  - 21 ms

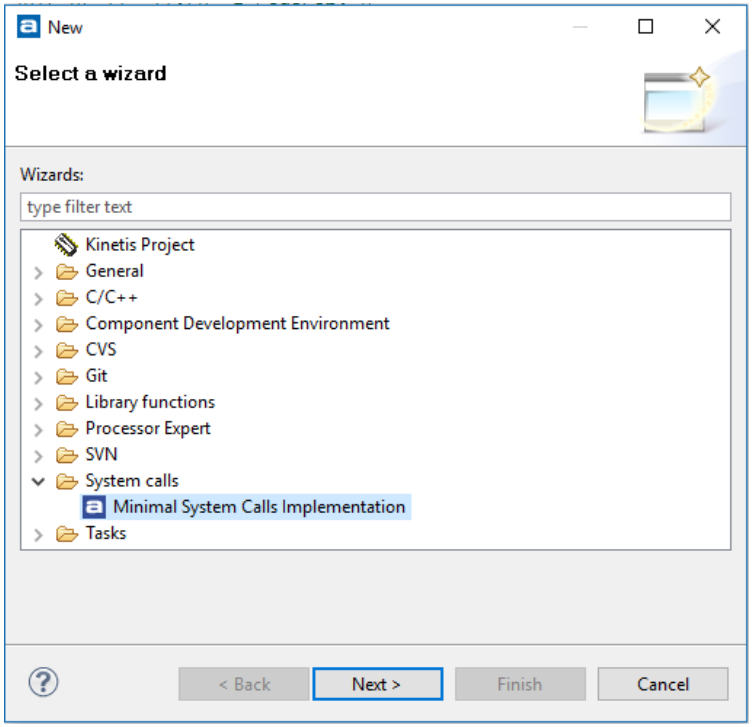


Tip #1 – Use printf through ITM

1 File->New->Other

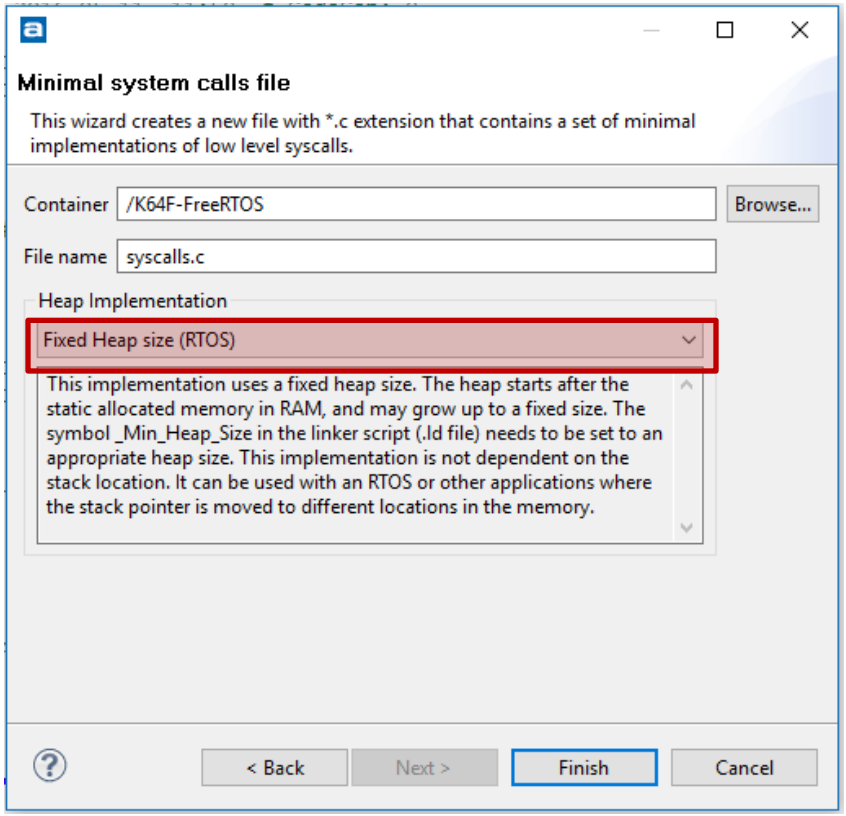


2 System Calls -> Minimal

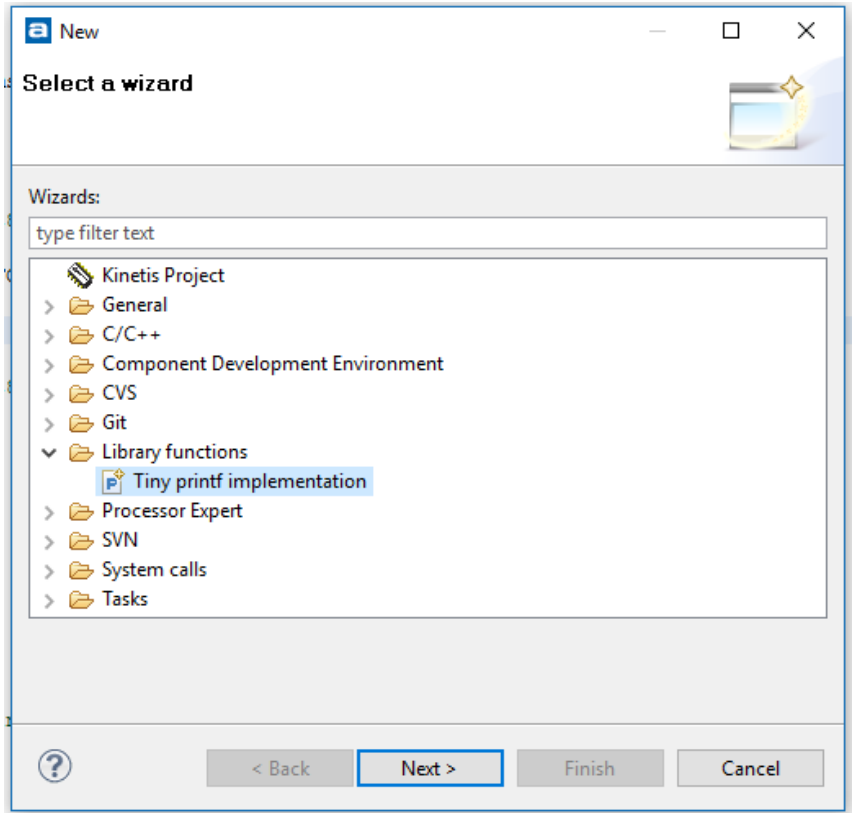


Tip #1 – Use printf through ITM

3 Fixed Heap



4 Tiny printf



Tip #1 – Use printf through ITM

5 Redirect printf

```
257 int iprintf(const char *fmt, ...)
258 {
259     int length = 0;
260     va_list va;
261     va_start(va, fmt);
262     length = ts_formatlength(fmt, va);
263     va_end(va);
264     {
265         char buf[length];
266         va_start(va, fmt);
267         length = ts_formatstring(buf, fmt, va);
268         length = _write(1, buf, length);
269         va_end(va);
270     }
271     return length;
272 }
```

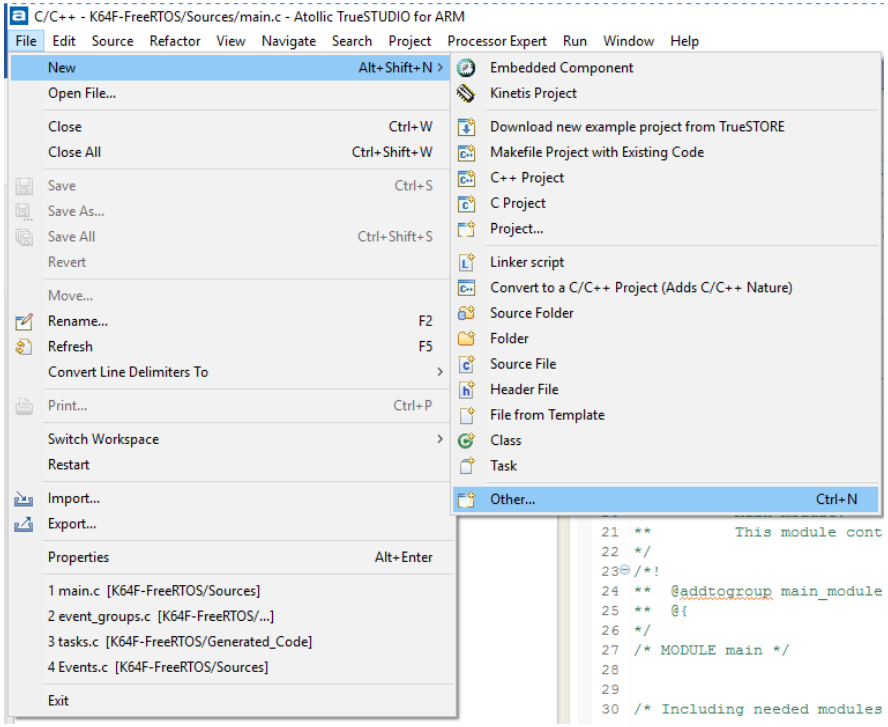
```
int _write(int32_t file, uint8_t *ptr, int32_t len)
{
  /* Implement your write code here, this is used by puts and printf for example */
  int i=0;
  for(i=0 ; i<len ; i++)
  {
    ITM_SendChar((*ptr++));
  }

  return len;
}
```

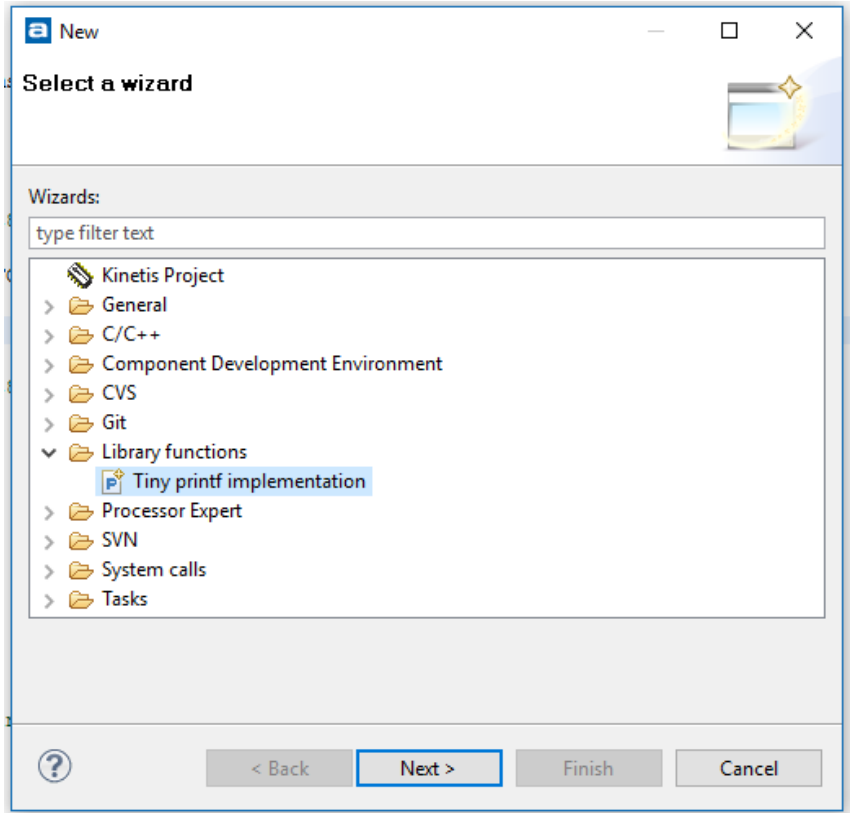
CMSIS ITM call

Tip #2 – Use printf through RTT

1 File->New->Other



2 Tiny printf



Tip #2 – Use printf through RTT

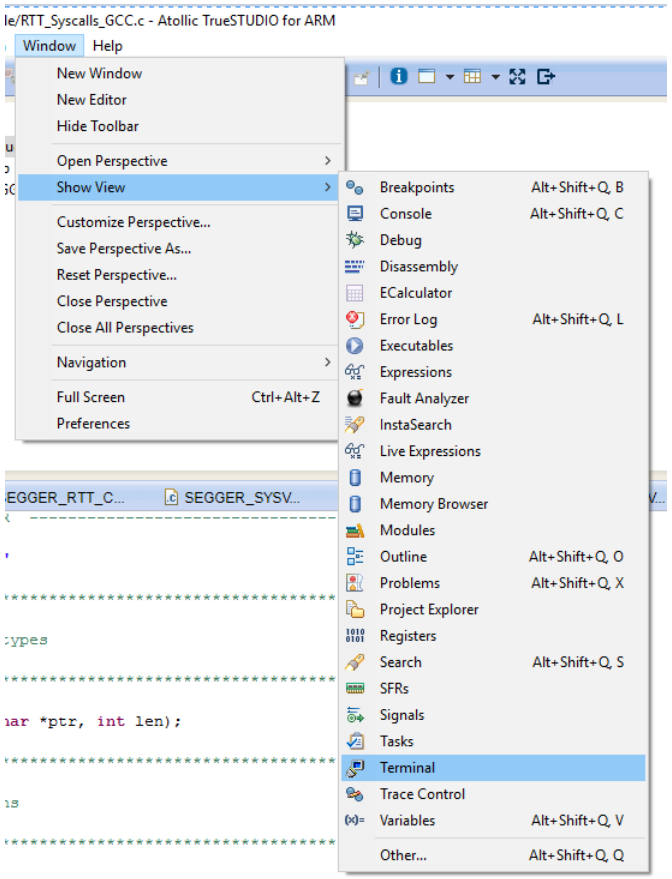
3 Redirect to RTT

main.c | tiny_printf.c | SEGGER

```
45  #include <stdarg.h>
46  #include <stdio.h>
47  #include <string.h>
48  #include "SEGGER_RTT.h"
```

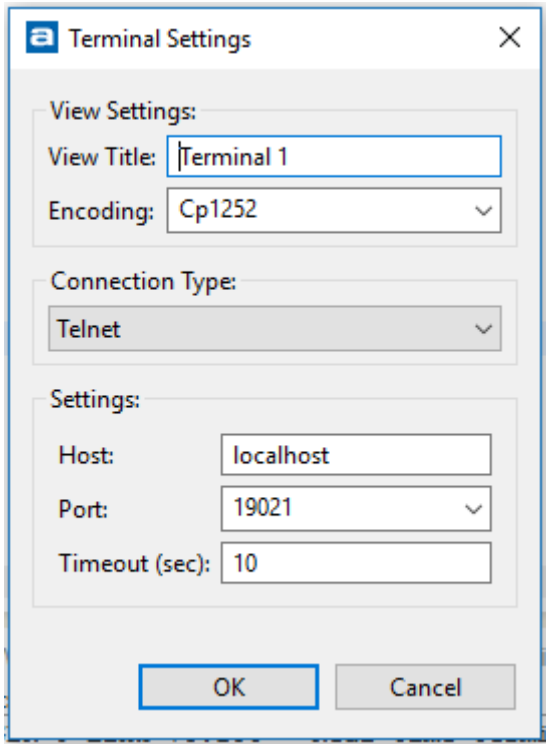
```
/*********************************************************************
*
*       _write()
*
* Function description
*   Low-level write function.
*   libc subroutines will use this system routine for output to all files,
*   including stdout.
*   Write data via RTT.
*/
int _write(int file, char *ptr, int len) {
  (void) file;  /* Not used, avoid warning */
  SEGGER_RTT_Write(0, ptr, len);
  return len;
}
```

4 Add terminal to debug

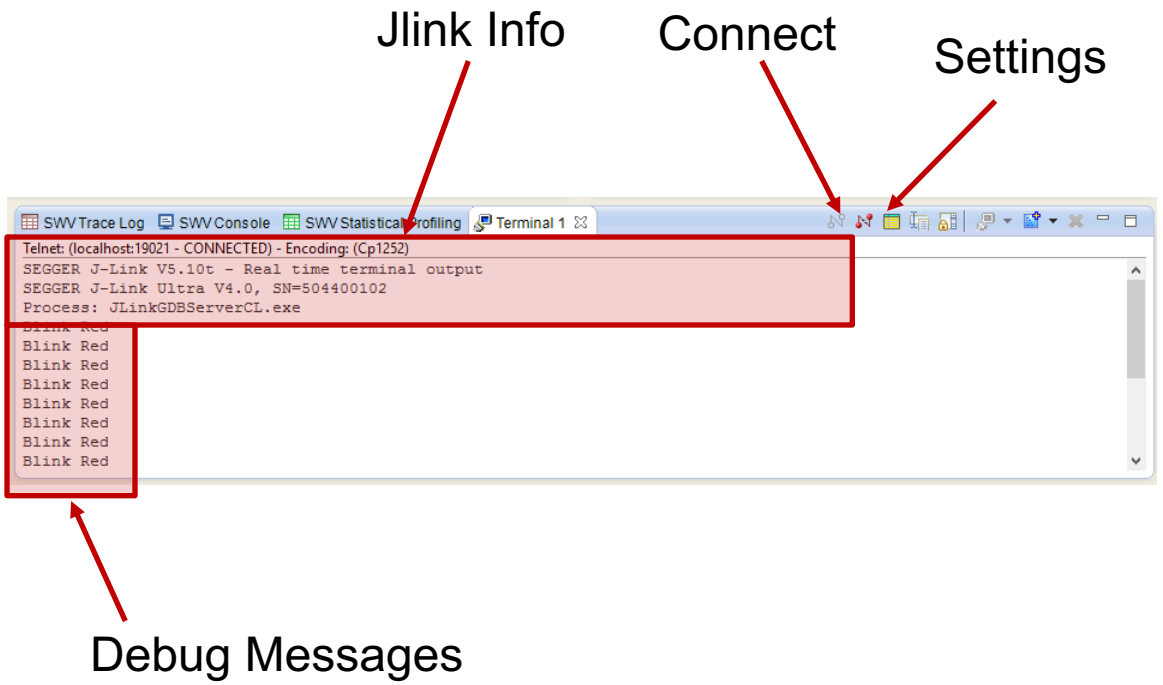


Tip #2 – Use printf through RTT

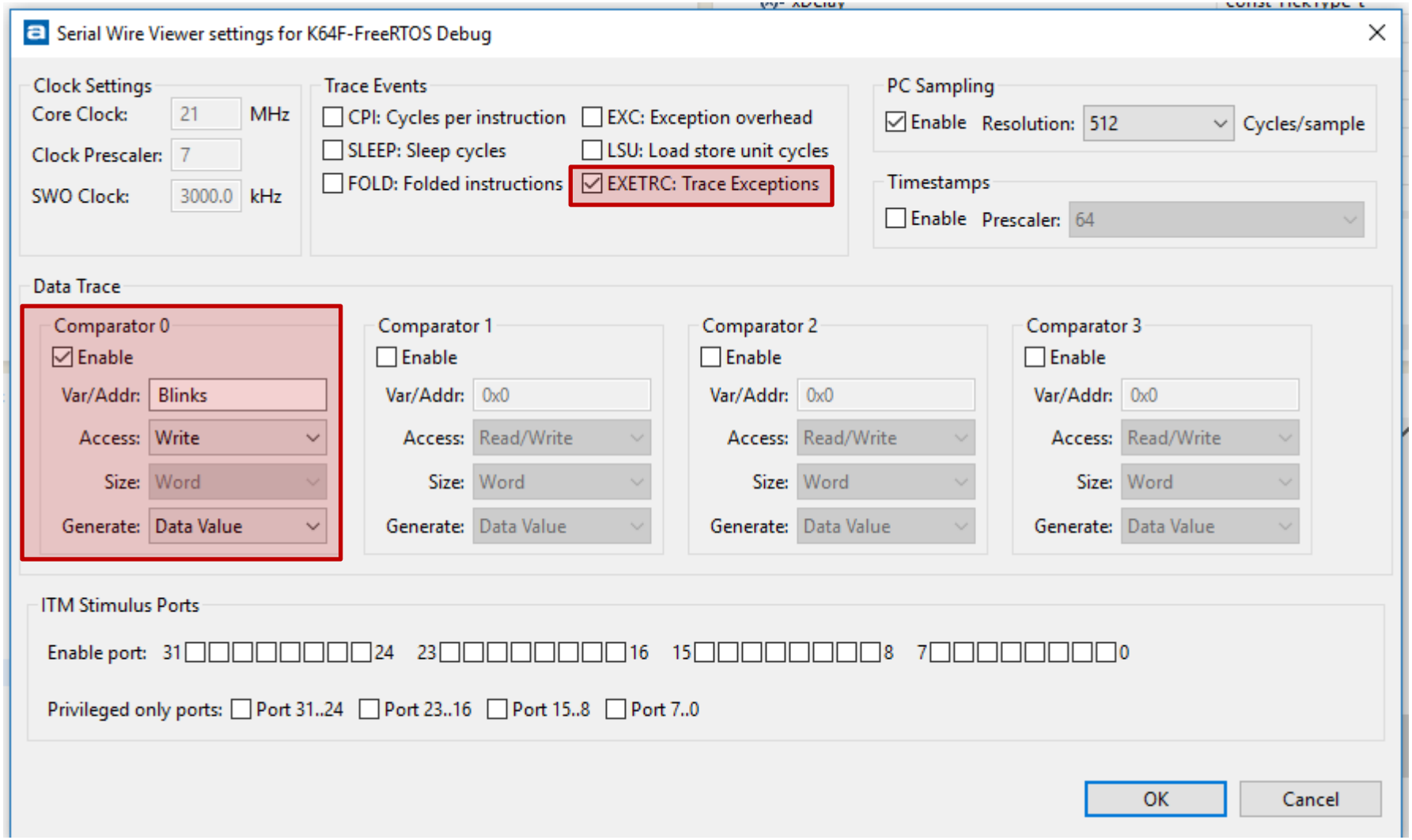
5 Connect to terminal



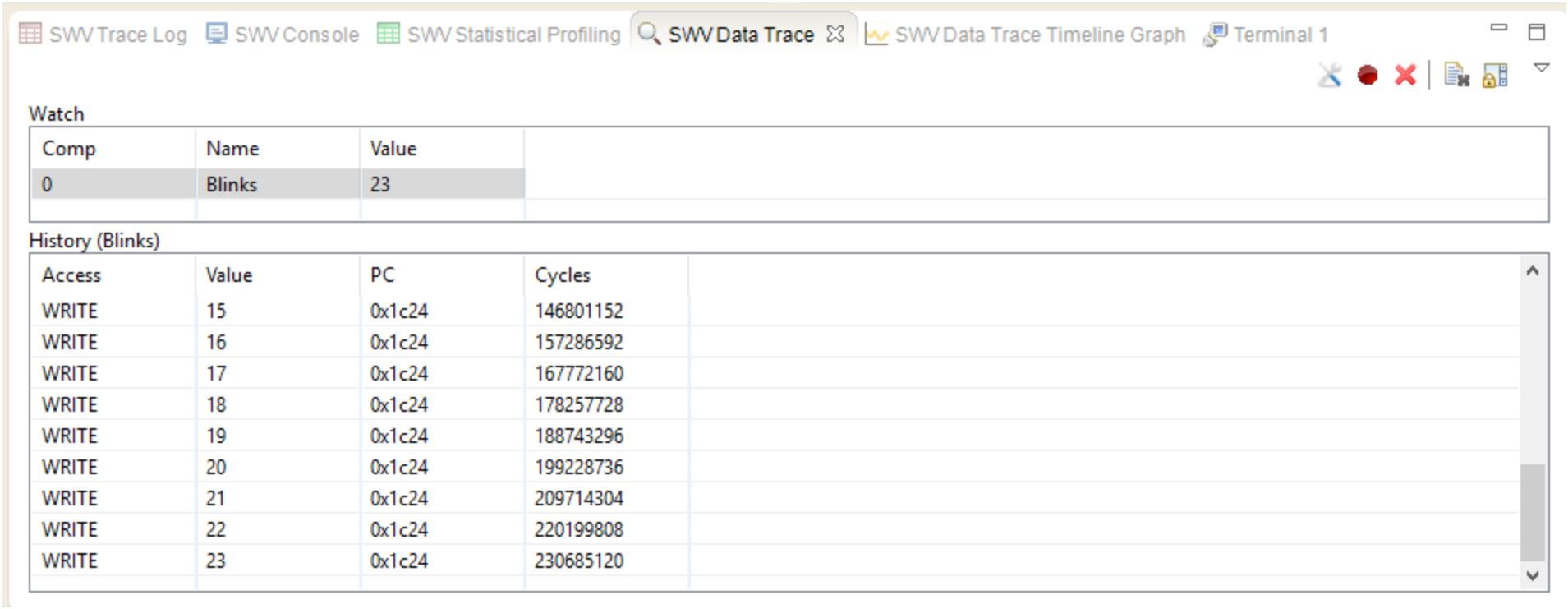
6 Observe debug messages



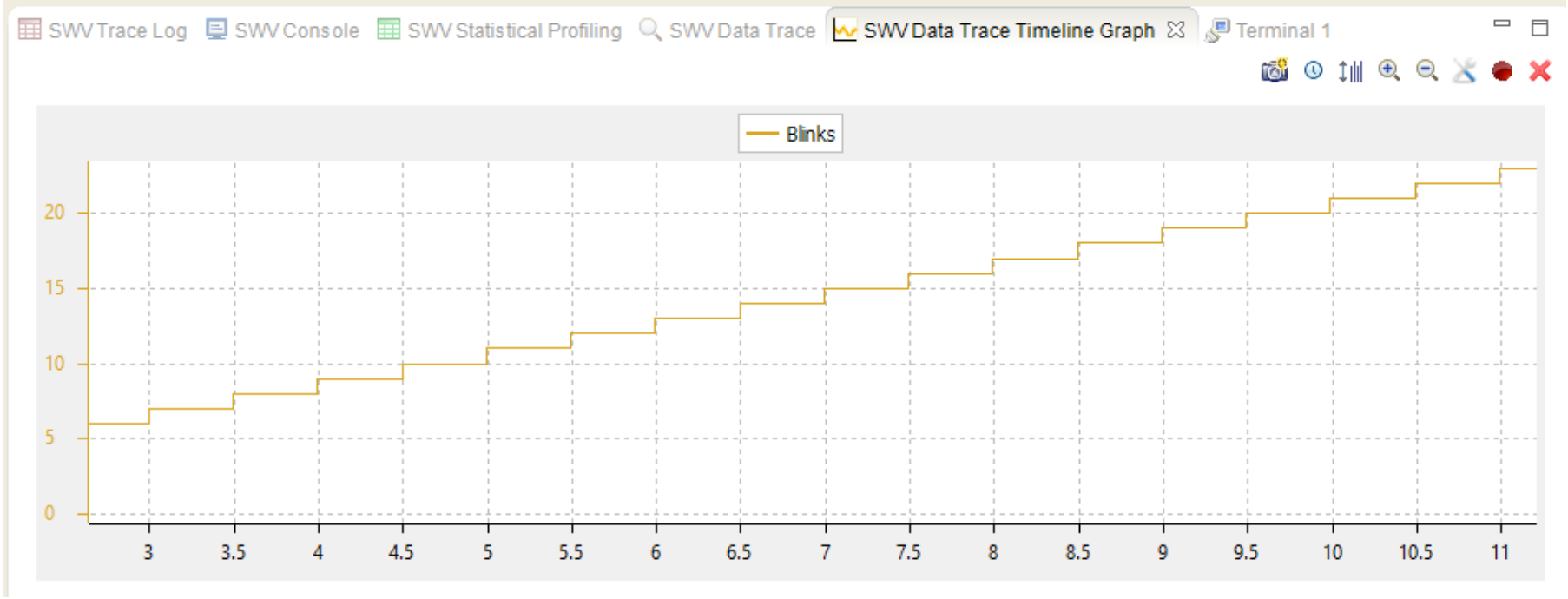
Tip #3 - Use DataWatch



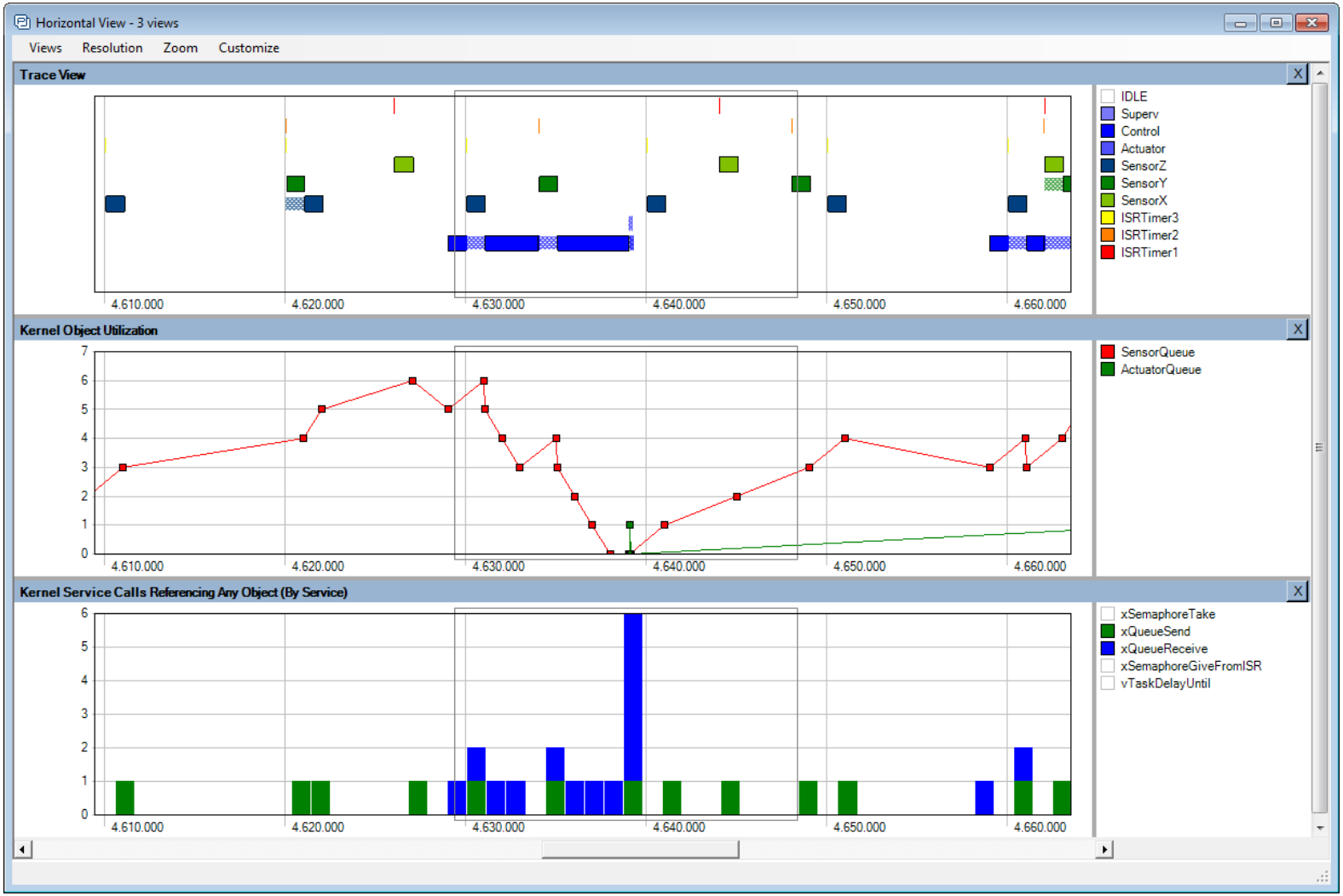
Tip #3 – Use Data Watch



Tip #4 – Use Data Trace Graphing



Tip #5 – Use Multiple Views

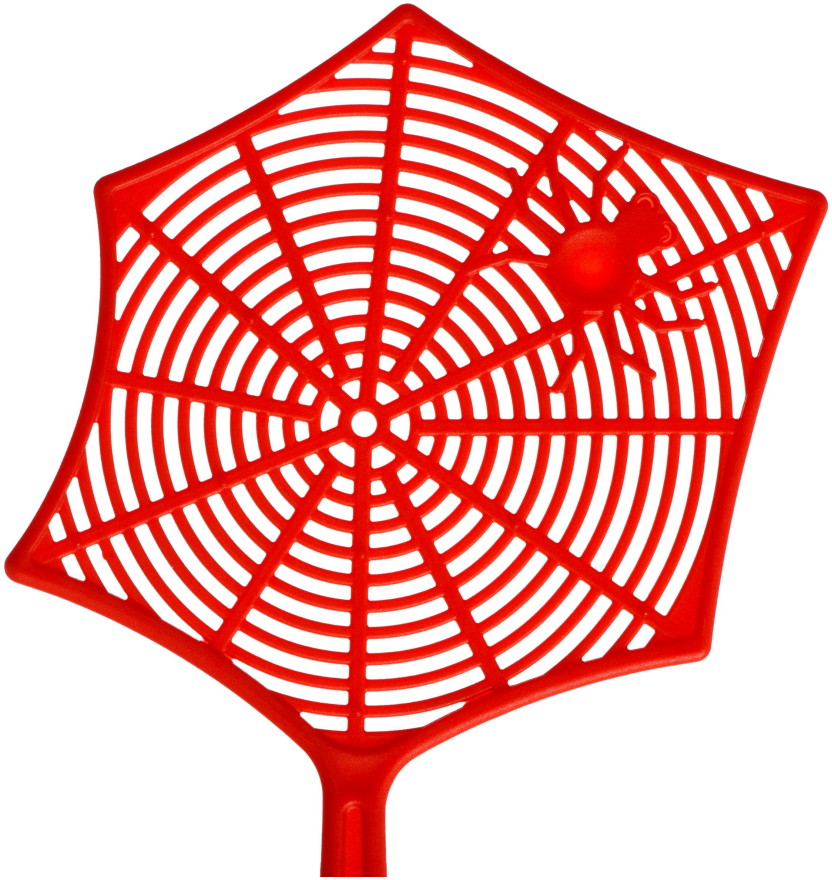


Tip #6 – Master Analytics

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



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

- E : jacob@beningo.com
- T : 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

DOULOS CERTIFIED TRAINING PARTNER

BENINGO EMBEDDED GROUP

www.beningo.com

CEC CONTINUING EDUCATION CENTER

Presented by: Digi-Key ELECTRONICS