

Empowering Embedded Systems

µC/OS-II

µC/Probe

and the NXP LPC2103 (Using the IAR LPC2103-02-SK Kickstart Kit)

> Application Note AN-1074

> > www.Micrium.com

About Micriµm

Micriµm provides high-quality embedded software components in the industry by way of engineer-friendly source code, unsurpassed documentation, and customer support. The company's world-renowned real-time operating system, the Micriµm μ C/OS-II, features the highest-quality source code available for today's embedded market. Micriµm delivers to the embedded marketplace a full portfolio of embedded software components that complement μ C/OS-II. A TCP/IP stack, USB stack, CAN stack, File System (FS), Graphical User Interface (GUI), as well as many other high quality embedded components. Micriµm's products consistently shorten time-to-market throughout all product development cycles. For additional information on Micriµm, please visit www.micrium.com.

About **µC/OS-II**

μC/OS-II is a preemptive, real-time, multitasking kernel. **μC/OS-II** has been ported to over 45 different CPU architectures.

µC/OS-II is small yet provides all the services you'd expect from an RTOS: task management, time and timer management, semaphore and mutex, message mailboxes and queues, event flags an much more.

You will find that **µC/OS-II** delivers on all your expectations and you will be pleased by its ease of use.

Licensing

 μ C/OS-II is provided in source form for FREE evaluation, for educational use or for peaceful research. If you plan on using μ C/OS-II in a commercial product you need to contact Micriµm to properly license its use in your product. We provide ALL the source code with this application note for your convenience and to help you experience μ C/OS-II. The fact that the source is provided DOES NOT mean that you can use it without paying a licensing fee. Please help us continue to provide the Embedded community with the finest software available. Your honesty is greatly appreciated.

About **µC/Probe** Demo Version

 μ C/Probe is a Windows application that allows a user to display and change the value (at run-time) of virtually any variable or memory location on a connected embedded target. The user simply populates μ C/Probe's graphical environment with gauges, tables, graphs, and other components, and associates each of these with a variable or memory location. Once the application is loaded onto the target, the user can begin μ C/Probe's data collection, which will update the screen with variable values fetched from the target.

µC/Probe retrieves the values of global variables from a connected embedded target and displays the values in an engineer-friendly format. The supported data-types are: booleans, integers, floats and ASCII strings.

µC/Probe can have any number of 'data screens' where these variables are displayed. This allows to logically grouping different 'views' into a product.

This **µC/Probe** demo version can only retrieve information from <u>RS-232C</u> or <u>J-LINK</u> interfaces and is limited up to 15 symbols.

The demo version of $\mu C/Probe$ is available on the Micriµm website:

http://www.micrium.com/products/probe/probe.html

About **µC/Probe** Full Version

The full version of μ C/Probe allows you to use a TCP/IP is a Windows application that allows a user to display and change the value (at run-time) of virtually any variable or memory location on a connected embedded target. The user simply populates μ C/Probe's graphical environment with gauges, tables, graphs, and other components, and associates each of these with a variable or memory location. Once the application is loaded onto the target, the user can begin μ C/Probe's data collection, which will update the screen with variable values fetched from the target.

Manual Version

If you find any errors in this document, please inform us and we will make the appropriate corrections for future releases.

Version	Date	Ву	Description
V 1.00	2008/19/19	FT	Initial revision.

Software Versions

This document may or may not have been downloaded as part of an executable file, *Micrium-NXP-uCOS-II-LPC2103-02-SK.exe* containing the code and projects described here. If so, then the versions of the Micriµm software modules in the table below would be included. In either case, the software port described in this document uses the module versions in the table below

Module	Version	Comment
μC/OS-II	V2.86	
µC/Probe	V2.20	

Document Conventions

Numbers and Number Bases

- Hexadecimal numbers are preceded by the "0x" prefix and displayed in a monospaced font. Example: 0xFF886633.
- Binary numbers are followed by the suffix "b"; for longer numbers, groups of four digits are separated with a space. These are also displayed in a monospaced font. Example: 0101 1010 0011 1100b.
- Other numbers in the document are decimal. These are displayed in the proportional font prevailing where the number is used.

Typographical Conventions

- Hexadecimal and binary numbers are displayed in a monospaced font.
- Code excerpts, variable names, and function names are displayed in a monospaced font. Functions names are always followed by empty parentheses (e.g., OS_Start()). Array names are always followed by empty square brackets (e.g., BSP Vector Array[]).
- File and directory names are always displayed in an italicized serif font. Example: /Micrium/Sofware/uCOS-II/Source/.
- A bold style may be layered on any of the preceding conventions—or in ordinary text—to more strongly emphasize a particular detail.
- Any other text is displayed in a sans-serif font.

Micriµm µC/OS-II and µC/Probe for the NXP LPC2103 CPU

Table of Contents

Table of Co	ntents	6
1.	Introduction	7
2. 2.01 2.02 2.03 2.03.01 2.04	Getting Started Setting up the Hardware Directory Tree Using the IAR Projects µC/OS-II Kernel Awareness Example Applications	9 9 11 11 12
3.	Directories and Files	13
4. 4.01 4.02	Application Code app.c os_cfg.h	16 16 18
5. 5.01 5.02	Board Support Package (BSP) BSP, <i>bsp.c</i> and <i>bsp.h</i> Processor Initialization Functions	19 19 20
6.	µC/Probe	22
Licensing		25
References		25
Contacts		25

1. Introduction

This document, *AN-1074*, explains example code for using μ C/OS-II and μ C/Probe with the NXP LPC2103 processor on the IAR LPC2103-02-SK evaluation board, shown in Figure 1.

The LPC2103 microcontroller is based on a 16-bit/32-bit ARM7TDMI-S CPU with real-time emulation that combines the microcontroller with 32 kB of embedded high speed flash memory. A blend of serial communications interfaces, ranging from multiple UARTS, SPI, and SSP to two I2Cs, and on-chip SRAM of 8 kB make these devices very well suited for communication gateways and protocol converters.

The IAR LPC2103-02-SK board includes :

- NXP LPC2103 MCU
- J-LINK on-chip with USB cable
- One serial port
- Reset button
- In-system programming (ISP) button
- Two user-defined buttons
- 16 fully configurable LEDs
- Power-on LED
- Lithium back-up battery and holder
- Breakout headers for all pins (suitable for mounting daughter boards)
- Small array of plated holes for prototyping



Figure 1-1. IAR LCP2103-02-SK

If this appnote was downloaded in a packaged executable zip file, then it should have been found in the directory /*Micrium/Appnotes/AN1xxx-RTOS/AN1074-uCOS-II-NXP-LPC2103-02-SK* and the code files referred to herein are located in the directory structure displayed in Section 2.02; these files are described in Section 3.

The executable zip also includes example workspaces for μ C/Probe. μ C/Probe is a Windows program which retrieves the value of variables form a connected embedded target and displays the values in an engineer-friendly format as shown in Figure 1-2. It interfaces with the IAR LPC2103-02-SK via RS-232C. For more information, including instructions for downloading a trial and the demo version of the program, please refer to Section 6.

Task Stack Information Name Stack Pointer Stack Maximum Current Current Starts © Ends © uC/05-II Stat 0x20006000 80/512 72/512 0x2000648 0x2000648 uC/05-II Thit 0x20006100 228/512 12/512 0x2000648 0x2000648 uC/05-II Thir 0x20006100 228/512 12/512 0x2000648 0x2000648 Start 0x2000510 12/2/121 12/512 0x2000648 0x2000648 Start 0x2000510 228/512 12/512 0x2000648 0x2000648 Probe R5-232 0x20005618 92/512 96/512 0x2000648 0x2000648 Probe OS Plugin 0x20005618 124/512 96/512 0x2000648 0x2000648 SCP1000 Sensor 0x20005181 124/512 96/512 0x2000648 0x2000648 Probe USB 0x20005181 124/1024 0x2000548 0x2000648 0x2000648 Probe USB 0x2000510 124/1024 0x2000548 0x2000548 0x20005488	Task Stack Information Name Stack Pointer Stack Usage Maximum Current Stack © State © Ends © Ends © uC/05-II Tide 0x20006C00 80/512 72/512 0x20006A8 0x20006A8 uC/05-II Timr 0x20006DD 112/512 88/512 0x20006A8 0x20006A8 uC/05-II Timr 0x20006DD 128/512 0x20006E48 0x20006A8 Probe R5-232 0x20006FB 92/512 0x20006F8 0x20006E48 Probe R5-232 0x20006FB 92/512 0x20006E48 0x20006E48 Probe R5-232 0x20006FB 91/512 112/512 0x20006E48 0x20006E48 Probe SF-Upilin 0x20006FB 91/512 112/512 0x20006E48 0x20006E48 SCP1000 Sensor 0x200065B 122/1024 12/512 0x20005A88 0x20003E48 Probe USB 0x200035B 126/1024 0x20005A88 0x20003E88 0x20003E88 Probe USB 0x200035B 126/1024 0x20005A88 0x20003E88 0x20003E88 MSD Task 0x20004160 376/1024 168/1024 0x20003E88 0x20003E88 </th <th>Stack Information Name Stack Waximum Current Stack Waximum Stack Waximum uC/OS-11 Idle 0.20006000 80/512 72/512 0x20006448 0x20006448 uC/OS-11 Stat 0.20006000 104/512 112/512 88/512 0x20006448 0x20006468 uC/OS-11 Two 0.20006000 104/512 12/512 0x2000648 0x20006488 start 0.20006768 104/512 12/512 0x20006488 0x20006488 SCP100 Sensor 0.20006788 92/512 96/512 10/200448 0x20006488 SCP100 Sensor 0.20006788 104/2048 104/2048 0x20006488 SCP100 Sensor 0.2000588 102/0124 10/20048 0x20005488 SCP100 Sensor 0.2000588 104/2048 0x20005488 0x20005488 MSD Task 0.2000518 102/0124 0x20005488 0x20005488 MSD Task 0.2000518 102/0124 0x20005488 0x20005488 MSD Task 0.2000518 106/1024 0x20005488</th>	Stack Information Name Stack Waximum Current Stack Waximum Stack Waximum uC/OS-11 Idle 0.20006000 80/512 72/512 0x20006448 0x20006448 uC/OS-11 Stat 0.20006000 104/512 112/512 88/512 0x20006448 0x20006468 uC/OS-11 Two 0.20006000 104/512 12/512 0x2000648 0x20006488 start 0.20006768 104/512 12/512 0x20006488 0x20006488 SCP100 Sensor 0.20006788 92/512 96/512 10/200448 0x20006488 SCP100 Sensor 0.20006788 104/2048 104/2048 0x20006488 SCP100 Sensor 0.2000588 102/0124 10/20048 0x20005488 SCP100 Sensor 0.2000588 104/2048 0x20005488 0x20005488 MSD Task 0.2000518 102/0124 0x20005488 0x20005488 MSD Task 0.2000518 102/0124 0x20005488 0x20005488 MSD Task 0.2000518 106/1024 0x20005488
Name Stack Stack Maximum Current Starts @ Ends @ uC/OS-II Idle 0x20006000 80/512 72/512 0x2000648 0x2000648 uC/OS-II Stat 0x20006000 112/512 88/512 0x2000648 0x2000648 uC/OS-II Timr 0x20006000 104/512 112/512 0x2000648 0x20006648 Start 0x20005100 122/512 10/512 112/512 0x20006648 Probe RS-232 0x20006788 92/512 96/512 0x20006648 0x20006648 Probe DS-Pulgin 0x20006788 92/512 96/512 0x2000648 0x2000648 Probe DS-Pulgin 0x20006580 96/512 112/512 0x2000648 0x2000648 SCP1000 Sensor 0x2000580 169/512 0x2000200 0x2000648 0x2000648 Probe DS 0x2000580 120/1024 0x2000200 0x2000648 0x2000648 SCP1000 Sensor 0x2000580 120/1024 0x2000408 0x2000548 0x2000548 Probe USB <td< th=""><th>Name Stack Maximum Curret Stack Ends @ uC/05-II Ide 0x20006C00 80/512 72/512 0x20006C48 0x20006A48 uC/05-II Tur 0x20006D08 10/512 12/512 0x20006E48 0x20006C48 uC/05-II Tur 0x20006D08 0x201026E48 0x20006E48 0x20006C48 Start 0x20006F08 0x20006E48 0x20006E48 0x20006C48 Start 0x20006F08 0x20006E48 0x20006E48 0x20006E48 Start 0x20006F08 12/512 0x20006E48 0x20006E48 Start 0x20006F08 12/512 0x20006E48 0x20006E48 Start 0x20006F08 108/512 112/512 0x20006E48 0x20006E48 SCP1000 Sensor 0x20002898 18/4/2048 10/4/2048 0x200021200 0x20002400 Probe Str 0x2000386 12/512 96/512 12/01024 0x2000348 0x2000348 JUS B Task 0x20003188 68/04096 12/04/248 0x20003488 0x20003488 0x20003488</th><th>Name Stack Pointer Hazimum (UCYOS-II Idle Ox20006C00 80/512 72/512 0x20006648 0x20006508 0x20006508 0x20006508 0x20006508 0x20006648 0x20006420 0x20006420</th></td<>	Name Stack Maximum Curret Stack Ends @ uC/05-II Ide 0x20006C00 80/512 72/512 0x20006C48 0x20006A48 uC/05-II Tur 0x20006D08 10/512 12/512 0x20006E48 0x20006C48 uC/05-II Tur 0x20006D08 0x201026E48 0x20006E48 0x20006C48 Start 0x20006F08 0x20006E48 0x20006E48 0x20006C48 Start 0x20006F08 0x20006E48 0x20006E48 0x20006E48 Start 0x20006F08 12/512 0x20006E48 0x20006E48 Start 0x20006F08 12/512 0x20006E48 0x20006E48 Start 0x20006F08 108/512 112/512 0x20006E48 0x20006E48 SCP1000 Sensor 0x20002898 18/4/2048 10/4/2048 0x200021200 0x20002400 Probe Str 0x2000386 12/512 96/512 12/01024 0x2000348 0x2000348 JUS B Task 0x20003188 68/04096 12/04/248 0x20003488 0x20003488 0x20003488	Name Stack Pointer Hazimum (UCYOS-II Idle Ox20006C00 80/512 72/512 0x20006648 0x20006508 0x20006508 0x20006508 0x20006508 0x20006648 0x20006420
uC/OS-II Idle 0x20006C00 80/512 72/512 0x20006C48 0x20006A48 uC/OS-II Stat 0x20006DDB 104/512 112/512 88/512 0x20006E48 0x20006E48 uC/OS-II Tmr 0x20006DDB 104/512 112/512 0x20006E48 0x20006E48 0x20006E48 Stat 0x20005DD 12/512 12/512 0x20005E48 0x20006E48 Probe S-232 0x200067E8 92/512 96/512 0x20006E48 0x20006E48 Probe S-19ugIn 0x200067B 92/512 96/512 0x20006E48 0x20006E48 Probe S-19ugIn 0x200067B 92/512 112/512 0x20006E48 0x20006E48 SCP1000 Sensor 0x20005BB 184/2048 104/2048 0x20002400 0x20002400 Probe SF 0x20005B1 122/112 102/1024 0x20006488 0x20002400 Probe SF 0x20005B1 122/112 102/1024 0x20002400 0x20002400 Probe USB 0x20005B1 120/1024 0x20005A88 0x20003488 0x20003488	uC/OS-II Idle 0x20006000 80/512 72/512 0x20006A48 0x20006A48 uC/OS-II Stat 0x20006DD8 104/512 112/512 0x20006E48 0x20006A48 uC/OS-II Tmr 0x20006DD8 104/512 112/512 0x20006A48 0x20006A48 Stat 0x20006DD8 104/512 112/512 0x20006248 0x20006A48 Probe RS-232 0x200067B 92/512 96/512 0x20007048 0x20006548 Probe OS-PlugIn 0x200063D8 96/512 112/512 0x20006448 0x2000648 SCP1000 Sensor 0x200063D8 96/512 112/512 0x20006448 0x20006448 Probe USB 0x200058B 122/512 0x20006448 0x20006448 0x20006448 Probe USB 0x2000510 122/512 0x20006448 0x20006448 0x20003A88 Probe USB 0x2000510 122/512 0x20006448 0x20003488 0x20003488 Probe USB 0x2000510 120/1024 120/1024 0x20004288 0x20003488 MSD Task <t< th=""><th>UC/OS-II Tidle 0x20006C00 80/512 72/512 0x20006748 0x2000648 UC/OS-II Stat 0x20005970 112/512 88/512 0x2000648 0x2000648 VUC/OS-II Tim 0x20005780 102/512 12/512 0x2000648 0x2000648 Probe RS-232 0x20006788 92/512 12/512 0x2000648 0x2000648 Probe DS Plugin 0x20006788 92/512 96/512 0x2000648 0x2000648 Probe DS Plugin 0x20006788 92/512 112/512 0x2000648 0x2000648 SCP1000 Sensor 0x20003508 96/512 112/0124 0x2000648 0x2000648 VUSB Task 0x20003508 12/4/212 0x2000648 0x2000648 0x2000648 VUSD Task 0x20003508 12/2/1024 120/1024 0x2000648 0x2000648 VUSD Task 0x20003508 12/2/1024 120/1024 0x20003688 0x200036048 VUC/OS-II Take 0x20003508 0x20003688 0x20003688 0x20003688 MSD Task 0x20004100</th></t<>	UC/OS-II Tidle 0x20006C00 80/512 72/512 0x20006748 0x2000648 UC/OS-II Stat 0x20005970 112/512 88/512 0x2000648 0x2000648 VUC/OS-II Tim 0x20005780 102/512 12/512 0x2000648 0x2000648 Probe RS-232 0x20006788 92/512 12/512 0x2000648 0x2000648 Probe DS Plugin 0x20006788 92/512 96/512 0x2000648 0x2000648 Probe DS Plugin 0x20006788 92/512 112/512 0x2000648 0x2000648 SCP1000 Sensor 0x20003508 96/512 112/0124 0x2000648 0x2000648 VUSB Task 0x20003508 12/4/212 0x2000648 0x2000648 0x2000648 VUSD Task 0x20003508 12/2/1024 120/1024 0x2000648 0x2000648 VUSD Task 0x20003508 12/2/1024 120/1024 0x20003688 0x200036048 VUC/OS-II Take 0x20003508 0x20003688 0x20003688 0x20003688 MSD Task 0x20004100
uC/05-II Statt 0x20006D0B 112/512 89/512 0x2000648 0x20006028 uC/05-II Tmr 0x20006D0B 104/512 112/512 0x2000648 0x20006048 Statt 0x20005100 228/512 120/512 0x20006248 0x20006048 Probe RS-232 0x20005610 112/1024 120/1024 0x20005688 0x2000648 Probe RS-1232 0x20005618 92/512 96/512 0x2000648 0x2000648 Probe USP lugIn 0x2000578 92/512 96/512 0x2000748 0x2000648 Probe DS-10gIn 0x2000588 96/512 112/512 0x2000648 0x2000648 SCP1000 Sensor 0x2000588 142/2048 0x2000648 0x2000648 0x2000648 Probe USP 0x2000588 120/1024 120/1024 0x2000648 0x2000388 Probe USB 0x2000510 120/1024 120/1024 0x2000400 0x20003400 Probe USB 0x2000510 176/1024 120/4096 0x2000588 0x2000388 MSD Task 0x20004	uC/05-II Stat uC/05-II Tmr 0x20006D0 112/512 88/512 0x2000648 0x2000648 Stat 0x20006D0 104/512 112/512 0x20006248 0x2000648 Probe R5-232 0x20005D0 112/512 12/512 0x2000548 0x2000648 Probe R5-232 0x20005E10 112/1024 120/1024 0x20005688 0x2000548 Probe D5 Pulgin 0x2000650 108/512 112/512 0x2000748 0x2000648 Probe D5 Pulgin 0x20006308 96/512 112/512 0x2000648 0x2000648 SCP1000 Sensor 0x2000588 124/512 96/512 0x2000648 0x2000648 Probe ST 0x2000588 124/512 96/512 10/204 0x2000648 Probe ST 0x20005908 124/512 96/512 0x2000648 0x2000648 Probe ST 0x20005908 124/512 96/512 0x2000400 0x2000340 Probe USB 0x20004100 0x2000408 0x2000420 0x2000420 Probe USB 0x20004100 376/1024	UC/05-II Stat 0x200069F0 112/S12 88/S12 0x2000648 0x2000648 UC/05-II Tmr 0x20006100 228/S12 112/S12 0x2000648 0x2000648 Stat 0x20006768 22/S12 12/S12 0x2000528 0x2000544 0x2000648 Probe S232 0x20006788 92/S12 112/S12 0x2000648 0x2000648 0x2000648 Probe S1 0x20006788 138/S12 112/S12 0x2000648 0x2000648 0x2000678 Probe S1 0x20005288 138/S12 12/S12 0x2000648 0x2000648 0x2000648 S7 Task 0x20005288 132/S12 12/S12 0x2000648 0x2000648 0x2000648 Probe Str 0x2000510 124/S12 96/S12 0x2000388 0x2000388 0x2000388 MSD Task 0x2000518 176/1024 176/1024 0x2000388 0x20003688 0x2000388 MSD Task 0x20004100 376/1024 186/1024 0x2000428 0x2000388 0x20005688 0x20005688 MSD Task 0x20004100 376/1024 186/1024 0x200757 0x2016
uC/OS-II Tmr 0x200061D0 104/512 112/512 0x20006148 0x20006100 228/512 12/512 0x2000648 0x2000648 0x20006501 112/12/12 12/512 0x2000648 0x20005288 0x20005788 0x20005788 0x20005788 0x20005788 0x20005788 0x20005788 0x2000648	uC/05-II Tmr 0x20006DB 104/512 112/512 0x20006248 0x20006048 Start 0x2000510 12/612 12/0124 0x20005688 0x20005288 Probe R5-232 0x200067B8 92/512 112/512 0x20005648 0x20006248 Probe OS Plugin 0x20006508 96/512 112/512 0x2000648 0x20006248 SCP1000 Sensor 0x20005058 124/512 92/0124 0x2000648 0x2000648 SCP1000 Sensor 0x20006508 164/2048 104/2048 0x2000648 0x2000648 SCP1000 Sensor 0x20003508 124/512 96/512 102/1024 0x2000548 SCP1000 Sensor 0x20003508 124/512 96/512 0x2000348 0x2000548 SCP1000 Sensor 0x20003108 220/1024 120/1024 0x20003488 0x2000348 FS Task 0x20003188 680/4096 120/4096 0x20004288 0x20005688 MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003588 UC/05-II Idle 65534 30 Belay Maiting On Message Switches<	UC/OS-II Tmr 0.20006D08 104/512 112/512 0x20006C48 0x20006C48 Start 0x2000510 121/1024 120/1024 0x20006648 0x20005648 Probe RS-322 0x20006F08 92/512 112/512 0x20006448 0x20006648 Probe DS PlugIn 0x20006F08 108/512 112/512 0x20006448 0x20006648 Probe ST 0x20006508 164/2048 104/2048 0x2000200 0x20002400 Probe ST 0x20005618 124/212 104/2048 0x2000100 0x20000448 SCP1000 Sensor 0x20005868 124/2048 104/2048 0x2000100 0x20000448 Probe ST stak 0x2000598 176/1024 120/1024 0x2000100 0x20000400 Probe USB 0x2000598 176/1024 102/408 0x20003E88 0x20003E88 0x20003E88 MSD Task 0x2000598 176/1024 102/4024 0x20004288 0x20003E88 0x20003E88 UC/OS-II Idle 0553 31 Ready ============== Setter Setter Setter Setter UC/OS-II Idle 0553
Start 0x20005100 228/512 120/512 0x20005648 0x20005648 0x20005648 Probe RS-232 0x200057E8 92/512 95/512 0x20005648 0x20005648 Probe OS PlugIn 0x200057B8 92/512 112/512 0x2000548 0x20006648 Probe OS PlugIn 0x200057B8 92/512 112/512 0x20006448 0x20006648 SCP1000 Sensor 0x20002898 144/2048 0x20006448 0x20002400 Probe Str 0x20005568 124/512 96/512 0x2000648 0x20006448 USB Task 0x20005568 124/512 96/512 0x20003200 0x20002400 Probe Str 0x20003588 124/512 96/512 0x20003488 0x2000448 USB Task 0x20003588 0x20001400 0x20003400 0x20003400 0x2000400 Probe USB 0x200035908 176/1024 176/1024 0x20005488 0x20003588 MSD Task 0x20004160 376/1024 168/1024 0x20003588 0x20003588 Mame ID </td <td>Start 0x20005100 228/512 120/512 0x2000548 0x2000548 Probe RS-232 0x20005610 112/1024 120/1024 0x20005688 0x20005648 KSD LED Task 0x20006708 92/512 96/512 0x20005648 0x20006648 Probe OS PlugIn 0x20006708 92/512 112/512 0x20006448 0x20006248 SCP1000 Sensor 0x20005618 124/2048 104/2048 0x20000200 0x200002400 Probe Str 0x20003518 124/512 96/512 0x20003488 0x200003488 USB Task 0x20003518 124/512 96/512 0x20003488 0x20000440 Probe Str 0x20003508 176/1024 120/4096 0x20001400 0x20000400 Probe USB 0x2000188 680/4096 120/4096 0x20004288 0x20003588 MSD Task 0x200014100 0x20004288 0x20003588 0x20003588 UC/05-II Task Information Information Information Information UC/05-II Tast 65534 30</td> <td>Start 0x200061D0 228/512 120/512 0x2000648 0x2000648 Probe RS-232 0x20005510 112/1024 120/1024 0x2000568 0x20005648 KSD LED Task 0x20006768 92/512 96/512 0x2000648 0x20006648 Probe OS PlugIn 0x20006518 108/512 112/512 0x20006448 0x20006448 SCP1000 Sensor 0x20006518 124/212 0x20006448 0x20006448 0x20006448 USB Task 0x2000510 122/102/1024 120/1024 0x20005488 0x20005448 Probe USB 0x2000590 176/1024 126/1024 0x2000588 0x2000388 MSD Task 0x2000590 176/1024 168/1024 0x2000588 0x2000588 UC/OS-II Tald 65535 31 Ready 0S 30952 84.57% UC/OS-II Tald 65534 30 Delay 56 720 0.69% UC/OS-II Tald 65534 30 Delay 56 720</td>	Start 0x20005100 228/512 120/512 0x2000548 0x2000548 Probe RS-232 0x20005610 112/1024 120/1024 0x20005688 0x20005648 KSD LED Task 0x20006708 92/512 96/512 0x20005648 0x20006648 Probe OS PlugIn 0x20006708 92/512 112/512 0x20006448 0x20006248 SCP1000 Sensor 0x20005618 124/2048 104/2048 0x20000200 0x200002400 Probe Str 0x20003518 124/512 96/512 0x20003488 0x200003488 USB Task 0x20003518 124/512 96/512 0x20003488 0x20000440 Probe Str 0x20003508 176/1024 120/4096 0x20001400 0x20000400 Probe USB 0x2000188 680/4096 120/4096 0x20004288 0x20003588 MSD Task 0x200014100 0x20004288 0x20003588 0x20003588 UC/05-II Task Information Information Information Information UC/05-II Tast 65534 30	Start 0x200061D0 228/512 120/512 0x2000648 0x2000648 Probe RS-232 0x20005510 112/1024 120/1024 0x2000568 0x20005648 KSD LED Task 0x20006768 92/512 96/512 0x2000648 0x20006648 Probe OS PlugIn 0x20006518 108/512 112/512 0x20006448 0x20006448 SCP1000 Sensor 0x20006518 124/212 0x20006448 0x20006448 0x20006448 USB Task 0x2000510 122/102/1024 120/1024 0x20005488 0x20005448 Probe USB 0x2000590 176/1024 126/1024 0x2000588 0x2000388 MSD Task 0x2000590 176/1024 168/1024 0x2000588 0x2000588 UC/OS-II Tald 65535 31 Ready 0S 30952 84.57% UC/OS-II Tald 65534 30 Delay 56 720 0.69% UC/OS-II Tald 65534 30 Delay 56 720
Probe RSD LED Data L2/1024 12/1024 102/1024 0x2000588 0x2000588 0x2000648 Probe OS Plugin 0x20006FB 92/512 96/512 0x2000648 0x20006648 Push Buttons 0x20002B98 184/2048 104/2048 0x20002C00 0x20002400 Probe SCP1000 Sensor 0x200055E8 124/512 96/512 0x20002400 0x20002400 Probe Str 0x200055E8 124/512 96/512 0x20002400 0x20006448 USB Task 0x200055E8 124/512 96/512 0x20003648 0x20006448 Probe Str 0x200055E8 124/512 96/512 0x20003648 0x20004648 USB Task 0x20005581 124/512 120/1024 0x20003588 0x2000400 Probe USB 0x20005908 166/1024 168/1024 0x2000588 0x20003588 MSD Task 0x20004160 376/1024 168/1024 0x20004288 0x20003588 Sceneral Task Informatio	Probe R5-232 0x200057E8 92/512 96/512 0x200067E8 0x200067E8 92/512 112/512 0x200067E8 0x200067E8 92/512 112/512 0x200067E8 0x200067E8 0x200067E8 0x200067E8 0x20002700 0x20002700 0x200067E8 0x200067E8 0x20002700 0x2	Probe RS-232 0x20005610 112/1024 120/1024 0x20005688 0x20005288 Probe OS PlugIn 0x200067B8 108/512 112/512 0x20006484 0x20006648 Probe OS PlugIn 0x200067B8 96/512 112/512 0x20006480 0x20006648 Probe OS PlugIn 0x200065B8 162/1024 0x2000648 0x20006480 0x20006480 SCP1000 Sensor 0x20002898 142/512 96/512 102/1024 0x20002100 0x20002400 Probe USB 0x20003886 122/1024 120/1024 0x20003180 0x20000400 Probe USB 0x20003808 176/1024 176/1024 0x20004288 0x20003888 MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003888 VC/OS-II Ide 65535 31 Ready 30952 84.57% UC/OS-II Ide 65534 30 Delay 56 30952 84.57% Stat 2 Mallbox 251 Kbd Mbox 0 295 0.01% Probe RS-232 13 13 Semaphore </td
Name ID Priority Task Status Context Current Name ID Priority Status Context Current	Name ID Priority State Delay Notice State	Note D Priority Task Ox2000648 Ox2000648 Push Buttons 0x20006588 96/512 112/512 0x20007046 0x20006648 Push Buttons 0x20006588 142/512 05/512 0x20006648 0x20006448 SCP1000 Sensor 0x20006588 124/512 05/512 0x20006648 0x20006448 USB Task 0x20003508 680/4096 120/4096 0x200004648 0x20003688 Probe USB 0x20003508 680/4096 120/4096 0x200005488 0x20003688 MSD Task 0x200041E0 376/1024 168/1024 0x20003588 0x20003688 MSD Task 0x200041E0 376/1024 168/1024 0x20003588 0x20003688 UC/OS-II Idle 65535 31 Ready 30952 84.57% UC/OS-II Irm 65533 29 Semaphore 720 0.69% UC/OS-II Irm 65534 30 Delay 56 720 0.69% UC/OS-II
Push Butrons SCP1000 Sensor 0x2000308 0x2000308 06/512 96/512 112/512 12/12 0x20006448 0x20006448 0x20006248 0x2000648 Probe Str Probe Str SCP1000 Sensor 0x2000510 220/1024 102/1024 0x20006448 0x20006448 Probe Str Probe Str Probe USB 0x2000310 220/1024 120/1024 0x20004648 0x20003408 Probe USB 0x20005908 176/1024 166/1024 0x2000488 0x2000588 MSD Task 0x20004160 376/1024 168/1024 0x20004288 0x20003688 Seneral Task Information Task Status Context Current	Name ID Priority Task Status Context 0x2000548 Context 0x2000548 SCP1000 Sensor 0x20005508 184/2048 104/2048 0x20002648 0x2000240 Probe Str 0x20005508 124/512 96/512 102/002648 0x20002448 USB Task 0x20001508 220/1024 122/1024 0x20003488 0x20003488 FS Task 0x20001388 680/4096 120/1024 0x20003488 0x20003488 Probe USB 0x20001388 680/4096 120/1024 0x20003488 0x20003488 MSD Task 0x200014E0 376/1024 176/1024 0x20004288 0x20003588 MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003588 UC/OS-II Idle 65533 31 Ready 30952 84.57% UC/OS-II That 65534 30 Delay 56 30952 84.57% UC/OS-II Thar 65533 29 Semaphore OS-TmrSig 30	None 10 Priority Table 12 0x20000548 0x20006448 0x20006448 VBS Buttons 0x200005205 96/512 122/512 0x20006448 0x200006448 Probe Str 0x20000510 122/0124 120/1024 0x200005288 0x20003288 USB Task 0x20000510 122/01024 120/1024 0x20003288 0x20003488 Probe USB 0x20000510 120/1024 120/1024 0x20005288 0x20003568 MSD Task 0x20000510 376/1024 168/1024 0x20003288 0x20003888 MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003888 UC/OS-II Idle 65535 31 Ready 30952 84.57% UC/OS-II Stat 65533 29 Semaphore 0S-TmrSig 720 0.69% Statt 2 2 Mallbox 251 Kod Mbox 0 295 0.01% Probe RS-232 13 3 Semaphore 0S-TmrSig 720 0.69% 0.01% 1295
SCP1000 Sensor 0x20002898 184/2048 104/2048 0x2000200 0x20002400 Probe Str 0x20000588 124/512 96/512 0x20006648 0x2000648 USB Task 0x2000388 680/4096 120/1024 0x2000400 0x20003488 Probe USB 0x2000388 680/4096 120/4096 0x20000400 0x20000400 Probe USB 0x200041E0 376/1024 176/1024 0x2000588 0x2000388 MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003E88 General Task Information Task Status Context Current Name ID Priority Task Status Context CUrrent	SCP1000 Sensor Probe Str USB Task 0x20002B98 0x20005E8 184/2048 124/512 104/2048 96/512 0x20002C00 0x20003E88 0x20002400 0x20003B88 FS Task Probe USB 0x20003E10 220/1024 120/1024 0x20003B88 0x20003A88 MSD Task 0x20005908 176/1024 120/4096 0x20005A88 0x20005688 MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003E88 State Delay Waiting On Vaiting On UC/05-II Stat Proverside State Delay Waiting On Vaiting On State Message Context Switches Current CPU Usage UC/05-II Tast 65534 30 Beady 55 30952 84.57% UC/05-II Timr 65533 29 Semaphore 0S-TmrSig 720 0.69% UC/05-II Timr 2 Mailbox 251 Kbd Mbox 0 295 0.01%	SCP1000 Sensor 0x20002898 184/2048 104/2048 0x20002C00 0x20002400 Probe Str 0x20006810 124/512 96/512 0x20006486 0x20006486 0x20003488 USB Task 0x20003180 680/4096 120/4096 0x20003488 0x20003488 Probe USB 0x2000510 120/4096 0x20001400 0x20005688 0x20005688 MSD Task 0x20001E0 376/1024 176/1024 0x20005488 0x20003688 VC/OS-Task Dreinty Task Delay Waiting On Message Switches VC/OS-TI Idle 65535 31 Ready 30952 84.57% UC/OS-TI Tidle 65533 29 Semaphore 0S-TmrSig Start 2 2 Mailbox 251 Kob Mbox 0 295 0.01% Probe RS-232 13 Semaphore OS-TmrSig 588 0.04% String T X #00012 String T X #00013 String T X
Probe Str USB Task 0x200055E8 124/512 96/512 0x20006488 0x20006488 0x20003648 FS Task 0x20001388 680/4096 120/1024 120/1024 0x20001400 0x20000408 Probe USB 0x20005908 176/1024 176/1024 0x2000488 0x2000588 MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003E88 General Task Information Task Status Context Current Name ID Priority Task Status Context CUrrent	Probe Str USB Task 0x20005E8 0x20003B8 124/512 0x2001288 96/512 0x20003E88 0x20006448 0x20003A88 0x20003A88 Probe USB MSD Task 0x20005P08 0x20005P08 120/1024 120/1024 0x20005A88 0x20003A88 MSD Task 0x20005P08 376/1024 168/1024 0x20004288 0x20003E88 Sceneral Task Information Information Information Information Information VC/05-II Idle 65535 31 Ready Information Information <thinformation< th=""></thinformation<>	Probe Str USB Task 0x20005EB 124/512 96/512 0x2000648 0x2000648 0x20003EB 0x20003B8 680/4096 120/1024 0x20003EB 0x20003EB 0x20003EB 0x200056B 0x20056B 0x20056B 0x20056B 0x20056B 0x20056B 0x20056B 0x20056B 0x20056B 0x20
USB Task 0x20003E10 220/1024 120/1024 0x20003888 0x20003A88 P5 Task 0x20003888 680/4096 120/4096 0x20001400 0x20000588 Probe USB 0x20005908 176/1024 176/1024 0x2000588 0x2000588 MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003E88 General Task Information Task Status Context Current Name ID Priority Status Switches Cill Usage	USB Task 0x20003E10 220/1024 120/1024 0x20003E88 0x20003A88 Probe USB 0x20005908 176/1024 120/1094 0x20005588 0x20003408 MSD Task 0x200041E0 376/1024 168/1024 0x20005488 0x20003488 MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003E88 Schereal Task Infinity Task Statt Context Current CPU Usage uC/05-II Ide 65534 30 Delay Waiting On Message Statt 220 0.69% uC/05-II Timr 65533 29 Semaphore 0S-TmrSig 720 0.69% uC/05-II Timr 5533 29 Semaphore 0S-TmrSig 720 0.69% uC/05-II Timr 65533 29 Semaphore 0S-TmrSig 720 0.69% uC/05-II Timr 5533 29 Mailbox 25 Viblox 0 295 0.01%	USB Task 0x20003E10 220/1024 120/1024 0x20003E88 0x20003A88 0x20003A88 Probe USB 0x2000359D8 176/1024 176/1024 0x2/10124 0x2/0005888 0x20003E88 0x20003E88 MSD Task 0x2000359D8 176/1024 176/1024 0x2/10124 0x2/0005888 0x2/0003E88 0x2/0003E88 MsD Task Dx200041E0 X76/1024 168/1024 0x2/0004288 0x2/0003E88 0x2/0003E88 VC/0S-II Idle D Priority Task Delay Waiting on Message Switches Current UC/OS-II Idle 65535 31 Ready OS-TmrSig 720 0.69% UC/OS-II Thir 65533 29 Semaphore OS-TmrSig 720 0.69% Statt 2 2 Mallbox 21 Voite RS-232 13 13 Semaphore OS-TmrSig 720 0.69% 0.01% String T # 11 11 Delay 12 0S-TmrSig 720
FS Task Probe USB MSD Task 0x20001988 0x20005908 680/4096 176/1024 0x20001400 176/1024 0x2000408 0x20005888 General Task Information Task Status Context Switches Current CPU Usore	Probe USB MSD Task 0x20001398 0x200059D8 0x200041E0 680/4096 176/1024 0x2000400 176/1024 0x2000400 0x2000588 0x20004288 Seneral Task Information Task Friority State Task Task Context CPU Usage Waiting On UC/05-II Stat Context CSS3 Current CPU Usage Context UC/05-II Tark 65533 30 Ready Delay State 0S-TmrSig Delay 30952 State 84.57% 0.05% Start 2 2 Mailbox 251 Kbd Mbox 0S-TmrSig Denaphore 588 0.04%	Frask Probe USB MSD Task 0x20001388 0x20005908 680/4096 176/1024 0x2000400 176/1024 0x2000588 0x2000428 0x20005688 0x20003688 MSD Task 0x200041E0 0x7/1024 0x2001428 0x20003688 0x20003688 MSD Task Information 0x2001400 0x20004288 0x2000428 0x20003688 Mame ID Priority Task Context Current Switches Current CPU Usage uC/OS-II Idle 65534 31 Ready 30952 84.57% uC/OS-II Timr 65533 29 Semaphore OS-TmrSig 720 0.69% uC/OS-II Timr 65533 29 Semaphore OS-TmrSig 720 0.69% uC/OS-II Timr 65534 10 Delay 10 720 0.69% 0.01% String Tx 11 Delay 12 588 0.04% 0.01% Probe RS-232 13 3 Semaphore Str
Probe USB MSD Task 0x20005908 0x200041E0 176/1024 376/1024 176/1024 168/1024 0x20005688 0x20004288 0x20005688 0x20003E88 General Task Information Task Status Context Current Current Name ID Priority Task Status Context Current	Probe USB MSD Task 0x200059D8 0x200041E0 176/1024 376/1024 176/1024 168/1024 0x20005688 0x20004288 0x20005688 0x20003E88 Seneral Task Information UC/05-II Idle ID Priority Task Task Context State Current Switches Current CPU Usage uC/05-II Idle 6553 31 Ready 30952 84.57% uC/05-II Tast 65534 30 Delay 56 720 0.69% uC/05-II Tast 65533 29 Semaphore 0S-TmrSig 720 0.69% uC/05-II Tast 2 Mallbox 251 Kbd Mbox 0 295 0.01%	Probe USB MSD Task 0x20005908 0x200041E0 176/1024 376/1024 176/1024 168/1024 0x2000588 0x20004288 0x20005688 0x20003688 Beneral Task Information Task Context Delay Context Waiting On UC/05-II Idle Context 05533 Current CPU Usage uC/05-II Idle 65535 31 Ready 05-TmrSig 30952 84.57% uC/05-II Idle 65533 29 Semaphore 05-TmrSig 720 0.69% uC/05-II Timr 65533 29 Semaphore 05-TmrSig 588 0.04% start 2 2 Mallox 251 Kbd Mbox 0 295 0.01% Probe RS-232 13 Semaphore 05-TmrSig 588 0.04% String Tx 400012 55 51 Delay 12
MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003E88 General Task Information Task Status Context Current Name ID Priority Task Status Context Current	MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003E88 General Task Information Task Status Context Current Name ID Priority State Delay Waiting On Message Current uC/05-II Idle 65535 31 Ready 720 0.69% uC/05-II Tmr 65533 29 Semaphore 0S-TmrSig 720 0.69% uC/05-II Tmr 65533 29 Semaphore Probe R5-232 13 13 Semaphore 720 0.69% 0.01%	MSD Task 0x200041E0 376/1024 168/1024 0x20004288 0x20003E88 eneral Task Information Information Task Status Context Current Switches Current CPU Usage uC/05-II Idle 65535 31 Ready 0 30952 84.57% uC/05-II Stat 65533 29 Semaphore 0S-TmrSig start 2 2 Mallox 251 Kod Mbox 0 295 0.01% Start 2 2 Mallox 251 Kod Mbox 0 295 0.01% Probe RS-232 13 Semaphore Probe RS-232 String Tx #00012 Probe OS PlugIn 11 11 Delay 12 String Tx #00013 String Tx 5 Delay 814
General Task Information	Name ID Priority Task State Context Current UC/05-II Idle 65535 31 Ready 30952 84.57% UC/05-II Thr 65533 29 Semaphore 05-TmrSig 720 0.69% Start 2 2 Mailbox 251 Kbd Mbox 0 295 0.01%	Name ID Priority Task Valing On Message Switches Current Switches UC/05-II Idle 65535 31 Ready 0 Waiting On Message 84,57% UC/05-II Idle 65534 30 Delay 56 05-TmrSig 720 0.69% UC/05-II Thr 65533 29 Semaphore 05-TmrSig 720 0.69% Start 2 2 Mallbox 25 KbM Mbox 0 295 0.01% Probe R5-232 13 13 Semaphore 05-TmrSig 588 0.04% Probe R5-232 13 13 Semaphore 05-TmrSig 588 0.04% Probe Str<232
Seneral Task Information	Seneral Task Information Name ID Priority Task Delay Waiting On Message Context Switches Current CPU Usage uC/05-II Idle 65534 30 Ready 30952 84.57% uC/05-II That 65534 30 Delay 56 720 0.69% uC/05-II Timr 65533 29 Semaphore 05-TmrSig 588 0.04% Start 2 2 Mailbox 251 Kbd Mbox 0 295 0.01%	Seneration Name ID Priority Task Context Current Switches Current CPU Usage uC/OS-II Idie 65535 31 Ready
Name ID Priority Task Status Context Current	Name ID Priority State Delay Waiting On Message Context Switches Current CPU Usage uC/05-II Idle 65535 31 Ready 30952 84.57% uC/05-II Stat 65534 30 Delay 56 720 0.69% uC/05-II Thrr 65533 29 Semaphore 0S-TmrSig 588 0.04% Start 2 2 Mailbox 251 Kbd Mbox 0 295 0.01% Probe RS-232 13 13 Semaphore Probe RS-232 0.01%	Name ID Priority Task Status Context Switches Current CPU Usage uC/OS-II Idle 65535 31 Ready 30952 84.57% uC/OS-II Stat 65534 30 Delay 56 720 0.69% uC/OS-II Thr 65533 29 Semaphore OS-TmrSig 588 0.04% Start 2 Mallbox 251 Kbd Mbox 0 295 0.01% Probe RS-232 13 13 Semaphore 0Frobe RS-232 0.01% Probe RS-232 13 3 Semaphore 0S-TmrSig KSD LED Task 12 12 Delay 10 String Tx #00012 Probe OS PlugIn 11 11 Delay 12 String Tx #00013 SCP1000 Sensor 16 16 Delay 814
State Delay Waiting (in Messade Switches Croosage)	uC/OS-II Idle 65535 31 Ready Ready 30952 84.57% uC/OS-II Stat 65534 30 Delay 56 720 0.69% uC/OS-II Tmr 65533 29 Semaphore 0S-TmrSig 588 0.04% Start 2 2 Mailbox 251 Kbd Mbox 0 295 0.01% Probe RS-232 13 13 Semaphore Probe RS-232 Frage	UC/05-II Idle 65535 31 Ready belay Finite of 56 Finite of 57 Finit of 57 Finit of 57 Fi
uC/OS-II Idle 65535 31 Ready 30952 84.57%	uC/OS-II Stat 65534 30 Delay 56 720 0.69% uC/OS-II Tmr 65533 29 Semaphore 0S-TmrSig 588 0.04% Start 2 2 Mailbox 251 Kbd Mbox 0 295 0.01% Probe RS-232 13 13 Semaphore Probe RS-232 From RS-232	uC/OS-II Statu 65534 30 Delay 56
uC/OS-II Stat 65534 30 Delay 56 720 0.69%	uC/OS-II Tmr 65533 29 Semaphore OS-TmrSig 588 0.04% Start 2 2 Mailbox 251 Kbd Mbox 0 295 0.01% Probe RS-232 13 13 Semaphore Probe RS-232	uC/OS-II Tmr 6553 29 Semaphore 0S-TmrSig 588 0.04% Start 2 2 Mailbox 25 Kbd Mbox 0 295 0.01% Probe RS-232 13 13 Semaphore Probe RS-232 Target Out> KSD LED Task 12 12 Delay 10 Probe RS-232 String Tx #00012 Probe OS PlugIn 11 11 Delay 12 String Tx #00013 SCP1000 Sensor 16 16 Delay 814 String Tx #00015 Probe Str 5 Delay 814
uC/OS-II Tmr 65533 29 Semaphore OS-TmrSig 588 0.04%	Start 2 2 Mailbox 251 Kbd Mbox 0 295 0.01% Probe RS-232 13 13 Semaphore Probe RS-232 Prob	Start 2 2 Mailbox 251 Kbd Mbox 0 295 0.01% Probe RS-232 13 Semaphore Probe RS-232 Fraget Output KSD LED Task 12 12 Delay 10 String TX #00012 Probe OS PlugIn 11 11 Delay 12 String TX #00013 Push Buttons 3 3 Delay 19 String TX #00014 SCP100 Sensor 16 16 Delay 814 String TX #00015 Probe Str 5 5 Delay 814 String TX #00016 USB Task 7 7 Delay 224
Start 2 2 Mailbox 251 Kbd Mbox 0 295 0.01%	Probe RS-232 13 13 Semaphore Probe RS-232	Probe RS-232 13 Semaphore Probe RS-232 Target Output KSD LED Task 12 Delay 10 String Tx #00012 Probe OS PlugIn 11 11 Delay 12 String Tx #00013 Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 VDB Task 7 7 Delay 224 String Tx #00016 VSB Task 6 6 Semaphore 407 App FS Lock String Tx #00017
Probe RS-232 13 13 Semaphore Probe RS-232 Probe RS-232	- KCD LED Tarle 10 E Target Output	KSD HED Task 12 12 12 12 11 Delay 10 The second s
Probe OS PlugIn 11 11 Delay 12 String Tx #00012	NOULEU Idok 12 12 Delay 10	Push Buttons 3 3 Delay 11 String Tx #00013 SCP1000 Sensor 16 16 Delay 19
	Probe OS Plugin 11 11 Delay 12 String Tx #00012	SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00015 USB Task 7 7 Delay 224 String Tx #00016 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013	Probe OS PlugIn 11 11 Delay 12 String Tx #00012 Push Buttons 3 3 Delay 19 String Tx #00013	Probe Str 5 5 Delay 814
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014	Probe OS PlugIn 11 11 Delay 12 3 String Tx #00012 Push Buttons 3 3 Delay 19	USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016	Probe OS Plugin 11 11 Delay 12 String Tx #00012 Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 Delay 1173 String Tx #00014 Probe Str 5 Delay 814 String Tx #00015	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12 String Tx #00012 Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00015 USB Task 7 7 Delay 224 String Tx #00016	
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018	Probe OS Plugin 11 11 Delay 12 String Tx #00012 Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 Delay 814 String Tx #00015 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 6 Semaphore 407 App FS Lock	Probe USB 15 15 Ready String Ty #00019
Push Buttons 3 3 Delay 19	Probe OS PlugIn 11 11 Delay 12 String Tx #00012 Push Buttons 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 D6 Delay 117 Probe Str 5 Delay 814 String Tx #00014 USB Task 7 Delay 224 String Tx #00017 FS Task 6 6 Senaphore 407 Probe USB 15 15 Ready String Tx #00018	String ix #00015
Push Buttons 3 3 Delay 19	Probe OS PlugIn 11 11 Delay 12 January String Tx #00012 Push Buttons 3 Delay 19 String Tx #00013 String Tx #00014 SCP1000 Sensor 16 Delay 117 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00015 USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 Semaphore 407 App FS Lock String Tx #00018 Probe USB 15 15 Ready String Tx #00019 MSD Task 8 8 Semaphore String Tx #00019	MSD Task 8 8 Semaphore uC/USB-Device String Tx #00019
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00015 USB Task 7 7 Delay 224 String Tx #00016 Probe USB 15 15 Ready String Tx #00019 MSD Task 8 8 Semaphore uC/USB-Device String Tx #00020 String Tx #00021 String Tx #00020 String Tx #00021	Probe OS Plugin 11 11 Delay 12 January String Tx #00012 Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00015 USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 Semaphore 407 App FS Lock String Tx #00018 Probe USB 15 15 Ready String Tx #00019 MSD Task 8 8 Semaphore uC/USB-Device String Tx #00021 String Tx #00021 String Tx #00021 String Tx #00021	MSD Task 8 8 Semaphore uC/USB-Device String Tx #00020 String Tx #00020 String Tx #00021
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00016 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018 Probe USB 15 15 Ready uC/USB-Device String Tx #00019 MSD Task 8 8 Semaphore uC/USB-Device String Tx #00020 String Tx #00021 String Tx #00021 String Tx #00023 String Tx #00023 String Tx #00024	Probe OS Plugin 11 11 Delay 12	MSD Task 8 8 Semaphore uC/USB-Device String Tx #00020 String Tx #00021 String Tx #00021 String Tx #00023
Push Buttons 3 3 Delay 19	Probe OS PlugIn Push Buttons 11 11 Delay Delay 12 String Tx #00012 SCP1000 Sensor 16 16 Delay 19 String Tx #00013 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 814 String Tx #00016 Probe USB 15 15 Ready UC/USB-Device String Tx #00019 MSD Task 8 8 Semaphore UC/USB-Device String Tx #00021 String Tx #00022 String Tx #00023 String Tx #00024	MSD Task 8 8 Semaphore uC/USB-Device String Tx #00020 String Tx #00020 String Tx #00021 String Tx #00023 String Tx #00024
Push Buttons 3 3 Delay 19	Probe OS PlugIn 11 11 Delay 12	MSD Task 8 8 Semaphore uC/USB-Device String Tx #00021 String Tx #00021 String Tx #00021 String Tx #00023
Probe OS PlugIn 11 11 Delay 12 String IX #00012	String Ty #00012	Push Buttons 3 3 Delay 19 String Tx #00014 SCP1000 Sensor 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
	Probe OS PlugIn 11 11 Delay 12 String Tx #00012	SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00015 USB Task 7 7 Delay 224 String Tx #00016 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
String Tx #00013	Probe OS PlugIn 11 11 Delay 12 String Tx #00012 String Tx #00013	SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00015 USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013	Probe OS PlugIn 11 11 Delay 12 String Tx #00012 Push Buttons 3 3 Delay 19 String Tx #00013	Probe Str 5 5 Delay 814
Push Buttons 3 3 Delay 19 String Tx #00013 String Tx #00014 String Tx #00014	Probe OS PlugIn 11 11 Delay 12 String Tx #00012 Push Buttons 3 3 Delay 19 String Tx #00013	Probe Str 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014	Probe OS PlugIn 11 11 Delay 12 String Tx #00012 Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015	USB Task 7 7 Delay 224
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Brobe Str 5 5 Delay 814 String Tx #00015	Probe OS PlugIn 11 11 Delay 12	USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016	Probe OS PlugIn 11 11 Delay 12 The second	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00017
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00016
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Tark 7 7 7 Point 77	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 UISB Task 7 7 Delay 224 String Tx #00016	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00016	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 Delay 814 String Tx #00016 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00016	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	PS lask 0 0 Semaphore 40/ App PS LOCK String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 824 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	PS lask 0 0 Semaphore 40/ App PS LOCK String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 824 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 824 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	rs lask 0 0 Semaphore 407 App rs Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	PS lask 0 0 Semaphore 40/ App PS LOCK String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 Delay 814 String Tx #00016 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USD Text 7 7 7 Point 77	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock SULING 1X #00017
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00017 String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016	Probe OS PlugIn 11 11 Delay 12 String Tx #00012 Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016	FS Task 6 6 6 Semaphore 407 App FS Lock String TX #00017
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USD Text 7 7 7 Point 77	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Tack 7 7 Delay 224 String Tx #00016	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String TX #00016
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 Delay 814 String Tx #00016 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	PS lask 0 Semaphore 407 App rs Lock String Tx #00018
Push Buttons 3 3 Delay 19	Probe OS PlugIn 11 11 Delay 12	String IX #UUU18
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	String IX #00016
Push Buttons 3 3 Delay 19	Probe OS PlugIn 11 11 Delay 12	String IX #00018
Push Buttons 3 3 Delay 19	Probe OS PlugIn 11 11 Delay 12	String TX #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS Plugin 11 11 Delay 12	FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00016	Probe OS Plugin 11 11 Delay 12	PS lask 0 0 Semaphore 40/ App PS LOCK String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	String TX #00018
Push Buttons 3 3 Delay 19	Probe OS PlugIn 11 11 Delay 12	String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	PS lask 0 Semaphore 407 App rs Lock String Tx #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	PS lask 0 Semaphore 407 App rs Lock String Tx #00018
Push Buttons 3 3 Delay 19	Probe OS PlugIn 11 11 Delay 12	String TX #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	String TX #00018
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00014 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017	Probe OS PlugIn 11 11 Delay 12	String IX #00016
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017 ES Tark 6 6 Senaphone 400 ES Low String Tx #00017	Probe OS PlugIn 11 11 Delay 12	
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018	Probe OS PlugIn 11 11 Delay 12	
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00016 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018	Probe OS PlugIn 11 11 Delay 12	
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 Delay 814 String Tx #00016 USB Task 7 7 Delay 224 String Tx #00017 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018	Probe OS PlugIn 11 11 Delay 12	Desha LICD 1E Dearty
Push Buttons 3 3 Delay 19 String Tx #00013 SCP1000 Sensor 16 16 Delay 1173 String Tx #00015 Probe Str 5 Delay 814 String Tx #00015 String Tax 7 7 Delay 224 FS Task 6 6 Semaphore 407 App FS Lock String Tx #00018	Probe OS Plugin 11 11 Delay 12	Probe USB 15 15 Ready and Array Magazine Transform

Figure 1-2. **µC/Probe** (with Target Output Window)

2. Getting Started

The following sections step through the prerequisites for using the demonstration application described in this document, *AN-1074* First, the setup of the hardware will be outlined. Second, the use and setup of the IAR Embedded Workbench project will be described. Thirdly, the steps to build the projects and load the application onto the board through a JTAG will be described. Lastly, instructions will be provided for using the example application.

2.01 Setting up the Hardware

The evaluation board can only be powered through the USB connector on the board. The processor on the evaluation board can be programmed and debugged through the USB J-LINK port.

To use μ C/Probe with the IAR-LPC2103-02-SK, download and install the trial version of the program from the Micriµm website as discussed in Section 6. After programming your target with one of the included example projects, connect a RS-232 cable between your PC and the evaluation board, configure the RS-232 options (also covered in Section 6), and start running the program. The open data screens should update. The IAR LPC2103-02-SK example application is configured to use UARTO, the RS-232C connector labeled "RS-232 for μ C/Probe" in Figure 1-1.

2.02 Directory Tree

If this file were downloaded as part of an executable zip file (which should have been named *Micrium-NXP-uCOS-II-LPC2103-02-SK.exe*, then the code files referred to herein are located in the directory structure shown in Figure 2-2.

Micriµm

μC/OS-II and μC/Probe for the NXP LPC2103 CPU



Figure 2-1. Directory Structure

2.03 Using the IAR Projects

One IAR project is located in the directory marked "IAR Example Project " in Figure 2-1:

/Micrium/Software/EvalBoards/NXP/LPC2103-02-SK/IAR/OS-Probe

The example project, *LPC2103-02-SK-OS-Probe-v5-2.ewp*, is intended for EWARM v5.2x. To view this example, start an instance of IAR EWARM v5.2x, and open the workspace file *LPC2103-02-SK-OS-Probe-v5-2.eww*. To do this, select the "Open" menu command under the "File" menu, select the "Workspace..." submenu command and select the workspace file after navigating to the project directory. The project tree shown in Figure 2-2 should appear. (In addition, the workspace should be openable by double-clicking on the file itself in a Windows Explorer window.)

2.03.01 µC/OS-II Kernel Awareness

When running the IAR C-Spy debugger, the μ C/OS-II Kernel Awareness Plug-In can be used to provide useful information about the status of μ C/OS-II objects and tasks. If the μ C/OS-II Kernel Awareness Plug-In is currently enabled, then a " μ C/OS-II" menu should be displayed while debugging. Otherwise, the plug-in can be enabled. Stop the debugger (if it is currently active) and select the "Options" menu item from the "Project" menu. Select the "Debugger" entry in the list box and then select the "Plugins" tab pane. Find the μ C/OS-II entry in the list and select the check box beside the entry, as shown in Figure 2-4.

When the code is reloaded onto the evaluation board, the " μ C/OS-II" menu should appear. Options are included to display lists of kernel objects such as semaphores, queues, and mailboxes, including for each entry the state of the object. Additionally, a list of the current tasks may be displayed, including for each task pertinent information such as used stack space, task status, and task priority, in addition to showing the actively executing task. An example task list for this project is shown in Figure 2-5.

Category:		Factory Settings
Seneral Options C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel GDB Server IAR ROM-monitor Dai Ink/Croce	Setup Downloa Select plugins to SEGGER em OSE Epsilon Power Pac R ThreadX Ø µC/OS-II Code Covera OBTI RTOS	d Extra Options Plugins load: bOS A TOS ge
LMI FTDI Macraigor RDI Third-Party Driver	Description: Location: Originator: Version:	RTDS awareness for CMX. C:\Program Files\IAR Systems\Embedded Workbench 5.2\ IAR Systems 5.20.0.50993

Figure 2-4. Enabling the **µC/OS-II** Kernel Awareness Plug-In

Micrium µC/OS-II and µC/Probe for the NXP LPC2103 CPU

Task List														×
Name	Ref	Prio	State	Dly	Waiting On	Msg Ctx Sw	Stk Ptr	Max%	Cur%	Max	Cur	Size	Starts @	Ends @
Start Task	3	2	Dly	12		81	04002CC0	40%	20%	208	104	512	04002D28	04002B28
User I/F	7	3	Mbox	80	?	79	04002EB0	46%	23%	240	120	512	04002F28	04002D28
Keyboard	8	4	Dly	30		157	040030B0	34%	23%	176	120	512	04003128	04002F28
Probe Str	9	5	Dly	12		17	04003278	45%	34%	232	176	512	04003328	04003128
Probe OS PlugIn	4	6	Dly	30		157	04001370	9%	7%	200	144	2048	04001400	04000C00
KSD LED Task	6	7	Dly	10		780	040034A0	37%	26%	192	136	512	04003528	04003328
Probe RS-232	5	11	Sem	0	Probe RS-232	1	04002768	17%	12%	184	128	1024	040027E8	040023E8
uC/OS-II Tmr	2	61	Sem	0	OS-TmrSig	82	04003AA8	35%	25%	184	128	512	04003B28	04003928
uC/OS-II Stat	1	62	Dly	81		79	040036C0	31%	20%	160	104	512	04003728	04003528
> uC/OS-II Idle	0	63	Ready	0		934	040038C0	20%	20%	104	104	512	04003928	04003728

Figure 2-5. µC/OS-II Task List.

2.04 Example Applications

The example projects include a basic demonstration of the μ C/OS-II and μ C/Probe. After you load the evaluation board the sample project, the LEDs will start blinkging

Stack Out of Range Notification

While debugging this project (or any other **µC/OS-II** project), IAR may log a SVC stack pointer out-of-range notification in the "Debug Log" window. This is actually normal behavior and does **NOT** indicate an error. IAR EWARM does not understand that the SVC stack pointer points to the stack for the current task stack.

3. Directories and Files

Application Notes

/Micrium/AppNotes/AN1xxx-RTOS\AN1014-uCOS-II-ARM

This directory contains AN-1014.pdf, the application note describing the ARM port for $\mu C/OS-II$, and AN-1014.PPT.pdf, a supplement to AN-1014.pdf.

MicriumAppNotesAN1xxx-RTOSAN1074-uCOS-II-NXP-LPC2103-02-SK This directory contains this application note, *AN-1074pdf*.

*Micrium**AppNotes**AN9xxx-MULT**AN-9913-IAR-Probe-Demo* This directory contains this application note, *AN-9913.pdf*.

Licensing Information

\Micrium\Licensing

Licensing agreements are located in this directory. Any source code accompanying this appnote is provided for evaluation purposes only. If you choose to use $\mu C/OS-II$ in a commercial product, you must contact Micriµm regarding the necessary licensing.

µC/OS-II Files

\Micrium\Software\uCOS-II\Doc

This directory contains documentation for **µC/OS-II**.

|Micrium|Software|uCOS-II|Ports|ARM|Generic|IAR

This directory contains the standard processor-specific files for the generic μ C/OS-II ARM port assuming the IAR toolchain. These files could easily be modified to work with other toolchains (i.e., compiler/assembler/linker/locator/debugger); however, the modified files should be placed into a different directory. The following files are in this directory:

- os_cpu.h
- os_cpu_a.asm
- os_cpu_c.c
- os_dcc.c
- os_dbg.c

With this port, μ C/OS-II can be used in either ARM or Thumb mode. Thumb mode, which drastically reduces the size of the code, was used in this example, but compiler settings may be switched (as discussed in Section 2.30) to generate ARM-mode code without needing to change either the port or the application code. The ARM/Thumb port is described in application note *AN*-1014 which is available from the Micrium web site.

|Micrium|Software|uCOS-II|Source

This directory contains the processor-independent source code for $\mu C/OS-II$.

µC/Probe Files

|Micrium|Software|uC-Probe|Communication|Generic|

This directory contains the μ C/Probe generic communication module, the target-side code responsible for responding to requests from the μ C/Probe Windows application (including requests over RS-232).

\Micrium\Software\uC-Probe\Communication\Generic\Source

This directory contains *probe_com.c* and *probe_com.h*, the source code for the generic communication module.

\Micrium\Software\uC-Probe\Communication\Generic\OS\uCOS-II

This directory contains *probe_com_os.c*, which is the $\mu C/OS-II$ port for the $\mu C/Probe$ generic communication module.

|Micrium| Software| uC-Probe| Communication| Generic| Source| RS-232

This directory contains the RS-232 specific code for μ C/Probe generic communication module, the target-side code responsible for responding to requests from the μ C/Probe Windows application over RS-232

\Micrium\Software\uC-Probe\Communication\Generic\Source\RS-232\Source

This directory contains *probe_rs232.c* and *probe_rs232.h*, the source code for the generic communication module RS-232 code.

Micrium\Software\uC-Probe\Communication\Generic\Source\RS-232\Ports\NXP\LPC21xx This directory contains *probe_rs232c.c* and *probe_rs232c.h*, the LPC21xx port for the RS-232 communications.

\Micrium\Software\uC-Probe\Communication\Generic\Source\RS-232\OS\uCOS-II

This directory contains *probe_rs232_os.c*, which is the μ C/OS-II port for the μ C/Probe RS-232 communication module.

\Micrium\Software\uC-Probe\Target\Demo\KSD\Source

This directory contains *ksd.c* and *ksd.h*, the source code for the IAR Kickstart kits demo example for the demo version of $\mu C/Probe$.

\Micrium\Software\uC-Probe\Target\Demo\KSD\Workspace

This directory contains *OS-Probe-Kickstart-Demo-Workspace.wsp* which is the generic **µC/Probe** workspace for the IAR Kickstart kits.

µC/CPU Files

\Micrium\Software\uC-CPU

This directory contains *cpu_def.h*, which declares #define constants for CPU alignment, endianness, and other generic CPU properties.

\Micrium\Software\uC-CPU\ARM\IAR

This directory contains *cpu.h* and *cpu_a.s. cpu.h* defines the Micriµm portable data types for 8-, 16-, and 32-bit signed and unsigned integers (such as CPU_INT16U, a 16-bit unsigned integer). These allow code to be independent of processor and compiler word size definitions. *cpu_a.s* contains generic assembly code for ARM7 and ARM9 processors which is used to enable and disable interrupts within the operating system. This code is called from C with OS ENTER CRITICAL() and OS EXIT CRITICAL().

µC/LIB Files

\Micrium\Software\uC-LIB

This directory contains *lib_def.h*, which provides #defines for useful constants (like DEF_TRUE and DEF DISABLED) and macros.

\Micrium\Software\uC-LIB\Doc

This directory contains the documentation for μ C/LIB.

Application Code

\Micrium\Software\EvalBoards\NXP\LPC2103-02-SK \IAR\OS-Probe

This directory contains the soruce code the μ C/OS-II and μ C/Probe example application:

- app.c contains the test code for the example application including calls to the functions that start multitasking within µC/OS-II, register tasks with the kernel, and update the user interface (the LEDs and the push buttons)
- *app_cfg.h* is a configuration file specifying stack sizes and priorities for all user tasks and #defines for important global application constants.
- *includes.h* is the master include file used by the application.
- *os_cfg.h* is the **µC/OS-II** configuration file.
- *LPC2103-02-SK-OS-Probe Workspace.wsp* is an example **µC/Probe** workspace.
- LPC2103-02-SK-OS-Probe-v5-2.* are the IAR EWARM v5.2x project files.

\Micrium\Software\EvalBoards\NXP\LPC2103-02-SK \IAR\BSP

This directory contains the Board Support Package for the IAR LPC2103-02-SK evaluation board:

- *bsp.c* contains the board support package functions which initialize critical processor functions (e.g., the PLL) and provide support for peripherals such as the push buttons and LEDs.
- *bsp.h* contains prototypes for functions that may be called by the user.
- *cstartup.s* is the IAR EWARM v5.2x startup file. This file performs critical processor initialization (such as the initialization of task stacks), readying the platform to enter main().
- *LPC2103_Flash.icf* is a IAR EWARM v5.2x linker file which contains information about the placement of data and code segments in the processor's memory map.
- *LPC2103_Flash.mac* contains instructions that are executed prior to loading code onto the processor.

4. Application Code

The example application described in this appnote, *AN-1073*, is a simple demonstration of $\mu C/OS-II$ and $\mu C/Probe$ for the NXP's LPC2103 processor on the IAR LPC2103-02-SK evaluation board. The basic procedure for setting up and using each of these can be gleaned from an inspection of the application code contained in *app.c*, which should serve as a beginning template for further use of these software modules. Being but a basic demonstration of software and hardware functionality, this code will make evident the power and convenience of $\mu C/OS-II$ "The Real-Time Kernel" used on the NXP's LPC2103 processor without the clutter or confusion of a more complex example.

4.01 *app.c*

Five functions of interest are located in *app.c*:

- 1. main() is the entry point for the application, as it is with most C programs. This function initializes the operating system, creates the primary application task, AppTaskStart(), begins multitasking, and exits.
- 2. App_TaskStart(), after creating the user interface tasks, enters an infinite loop in which it blinks the LEDs on the board.

voi	d main (void)			/* Note 1 */
i	CPU INTO8U err;			
	<pre>BSP_IntDisAll();</pre>			/* Note 2 */
	OSInit();			/* Note 3 */
	OSTaskCreateExt((void (*) (void	<pre>*)) App_TaskStart, *) 0.</pre>	/* Note 4 */
		(OS_STK	*) & AppTaskStartStk[APP_CFG_TAS]	K_START_STK_SIZE - 1],
		(INT80 (INT16U) APP CFG TASK START PRIO,) APP CFG TASK START PRIO,	
		(OS_STK (INT32U	*) & AppTaskStartStk[0],) APP CFG TASK START STK SIZE	
		(void	*) 0,	
		(INT16U) (OS_TASK_OPT_STK_CHK OS_TA	SK_OPT_STK_CLR));
#if	OS_TASK_NAME_SIZ	E > 13		/* Note 5 */
	OSTaskNameSet (AP	P_CFG_TASK_STAR	T_PRIO, "Start Task", &err);	
#end	dıt			
	OSStart();			/* Note 6 */
1				

Listing 4-1, main()

Listing 4-1, Note 1: As with most C applications, the code starts in main().

- Listing 4-1, Note 2: All interrupts are disabled to make sure the application does not get interrupted until it is fully initialized.
- Listing 4-1, Note 3: OSInit() must be called before creating a task or any other kernel object, as must be done with all µC/OS-II applications.

- Listing 4-1, Note 4: At least one task must be created (in this case, using OSTaskCreateExt() to obtain additional information about the task). In addition, µC/OS-II creates either one or two internal tasks in OSInit(). µC/OS-II always creates an idle task, OS_TaskIdle(), and will create a statistic task, OS_TaskStat() if you set OS_TASK_STAT_EN to 1 in *os_cfg.h*.
- Listing 4-1, Note 5: As of V2.6x, you can now name µC/OS-II tasks (and other kernel objects) and display task names at run-time or with a debugger. In this case, the App_TaskStart() is given the name "Start Task". Because C-Spy can work with the Kernel Awareness Plug-In available from Micriµm, task names can be displayed during debugging.
- Listing 4-1, Note 6: Finally multitasking under µC/OS-II is started by calling <code>OSSTart(). µC/OS-II</code> will then begin executing <code>App_TaskStart()</code> since that is the highest-priority task created (both <code>OS_TaskStat()</code> and <code>OS_TaskIdle()</code> having lower priorities).

```
static void App TaskStart (void *p arg)
    CPU INTO8U i;
    (void)p arg;
   BSP Init();
                                                                  /* Note 1 */
#if (OS TASK STAT EN > 0)
                                                                  /* Note 2 */
   OSStatInit();
 #endif
#if (APP CFG PROBE COM EN
                              == DEF ENABLED) || \setminus
    (APP CFG PROBE OS PLUGIN EN == DEF ENABLED)
   App_ProbeInit();
                                                                  /* Note 3 */
#endif
   App TaskCreate();
                                                                  /* Note 4 */
   App EventCreate();
   BSP LED Off(0);
    while (DEF_TRUE) {
                                                                  /* Note 5 */
        for (i = 1; i <= 16; i++) {
             BSP LED Toggle(i);
             OSTimeDlyHMSM(0, 0, 0, 100);
        }
        for (i = 1; i <= 10; i++) {
             BSP LED Toggle(0);
             OSTimeDlyHMSM(0, 0, 0, 50);
        }
    }
```

Listing 4-2, App_TaskStart()

- Listing 4-2, Note 1: BSP_Init() initializes the Board Support Package—the I/Os, tick interrupt, etc. See Section 5 for details.
- Listing 4-2, Note 2: OSStatInit() initializes µC/OS-II's statistic task. This only occurs if you enable the statistic task by setting OS_TASK_STAT_EN to 1 in *os_cfg.h*. The statistic task measures overall CPU usage (expressed as a percentage) and performs stack checking for all the tasks that have been created with OSTaskCreateExt() with the stack checking option set.

- Listing 4-2, Note 3: App_ProbeInit() initialize µC/Probe. This function calls OSProbe_Init()
 which initializes the µC/Probe plug-in for µC/OS-II, which maintains CPU usage statistics for
 each task. ProbeCom_Init() which initializes the µC/Probe generic communication module,
 ProbeRS232_Init() which initializes the RS-232 communication module and KSD_Init()
 which initializes the IAR Kickstart kit demo (KSD) for the demo version of µC/Probe. (see AN9913). After these have been initialized, the µC/Probe Windows program will be able to
 download data from the processor. For more information, see Section 6.
- Listing 4-2, Note 4: App_TaskCreate(), App_EventCreate() Creates all the application tasks and events (respectively).
- Listing 4-2, Note 9: Any task managed by $\mu C/OS-II$ must either enter an infinite loop 'waiting' for some event to occur or terminate itself. This task enters an infinite loop in which the LEDs are toggled.

4.02 os_cfg.h

The file $os_cfg.h$ is used to configure $\mu C/OS-II$ and defines the maximum number of tasks that your application can have, which services will be enabled (semaphores, mailboxes, queues, etc.), the size of the idle and statistic task and more. In all, there are about 60 or so #define that you can set in this file. Each entry is commented and additional information about the purpose of each #define can be found in Jean Labrosse's book, $\mu C/OS-II$, The Real-Time Kernel, 2nd Edition. $os_cfg.h$ assumes you have $\mu C/OS-II$ V2.83 or higher but also works with previous versions of $\mu C/OS-II$.

- OS_APP_HOOKS_EN is set to 1 so that the cycle counters in the OS_TCBs will be maintained.
- Task sizes for the Idle (OS_TASK_IDLE_STK_SIZE), statistics OS_TASK_STAT_STK_SIZE) and timer (OS_TASK_TMR_STK_SIZE) task are set to 128 OS_STK elements (each is 4 bytes) and thus each task stack is 512 bytes. If you add code to the examples make sure you account for additional stack usage.
- OS_DEBUG_EN is set to 1 to provide valuable information about µC/OS-II objects to IAR's C-Spy through the Kernel Awareness plug-in. Setting OS_DEBUG_EN to 0 should some code space (though it will not save much).
- OS LOWEST PRIO is set to 31, allowing up to 64 total tasks.
- OS_MAX_TASKS determines the number of "application" tasks and is currently set to 10.
- OS_TICKS_PER_SEC is set to 1000 Hz. This value can be changed as needed and the proper tick rate will be adjusted in *bsp.c* if you change this value. You would typically set the tick rate betweek 10 and 1000 Hz. The higher the tick rate, the more overhead µC/OS-II will impose on the application. However, you will have better tick granularity with a higher tick rate.

Micriµm

μC/OS-II and μC/Probe for the NXP LPC2103 CPU

5. Board Support Package (BSP)

The Board Support Package (BSP) provides functions to encapsulate common I/O access functions and make porting your application code easier. Essentially, these files are the interface between the application and the IAR LPC2103-02-SK evaluation board. Though one file, *bsp.c*, contains some functions which are intended to be called directly by the user (all of which are prototyped in *bsp.h*), the other files serve the compiler (as with *cstartup.*).

5.01 BSP, *bsp.c* and *bsp.h*

The file *bsp.c* implements several global functions, each providing some important service, such as the initialization of processor functions for $\mu C/OS-II$ to operate or the toggling of an LED. Several local functions are defined as well to perform some atomic duty, initializing the I/O for the LED or initialize the $\mu C/OS-II$ tick timer. The discussion of the BSP will be limited to the discussion of the global functions that might be called from user code (and may be called from the example application).

The global functions defined in *bsp.c* (and prototyped in *bsp.h*) may be roughly divided into two categories: critical processor initialization and user interface services. Three functions constitute the former:

- BSP_Init() is called by the application code to initialize critical processor features (particularly the µC/OS-II tick interrupt) after multitasking has started (i.e., OS_Start() has been called). This function should be called before any other BSP functions are used. See Listing 5-1 for more details.
- **BSP_IntDis()** is called to disable an specific interrupt source.
- **BSP_IntDisAll()** is called to disable all interrupts, thereby preventing any interrupts until the processor is ready to handle them.
- **BSP** IntEn() is called to enable a specific interrupt souce.
- **BSP IntVectSet()** is called to assign a ISR handler to a specific interrupt source.
- **BSP_CPU_ClkFreq()** returns the clock frequency in Hz.

Several functions provide access to user interface components:

- BSP_LED_Toggle(), BSP_LED_On() and BSP_LED_Off() will toggle, turn on, and turn off (respectively) the LED corresponding to the ID passed as the argument If an argument of 0 is provided, the appropriate action will be performed on all LEDs. Valid IDs are 1, 2, 3 and 4 (inclusive).
- **BSP_PB_GetStatus** () returns the status of the board's push buttons corresponding the ID passed as the argument.

5.02 **Processor Initialization Functions**

/* Note 1 */
/* Note 2 */
/* Note 3 */
/* Note 4 */
/* Note 5 */

Listing 5-1, BSP Init()

- Listing 5-1, Note 1: MEMMAP register is set to User Flash Mode , in this mode the interrupt vectors are not re-mapped and reside in Flash.
- Listing 5-1, Note 2: The PLL is initialized.
- Listing 5-1, Note 3: The vectored interrupt controller is initialized.

Listing 5-1, Note 4: The I/Os for the board's peripherals are initialized (LEDs and Push buttons).

Listing 5-1, Note 4: The µC/OS-II tick interrupt source is initialized.

Listings 5-2 and 5-3 give the μ C/OS-II timer tick initialization function, BSP_Tmr_TickInit(), the tick ISR handler, BSP_Tmr_TickISR_Handler(). These may serve as examples for initializing an interrupt and servicing that interrupt.

```
static void BSP_Tmr_TickInit (void)
{
    CPU INT32U pclk freq;
CPU_INT32U tmr_reload;
    BSP IntVectSet((CPU INT08U
                                   )BSP INT ID TMR0,
                                                                                   /* Note 1 */
                     (CPU INT08U )0,
(CPU INT08U )BSP INT TYPE NO VEC IRQ,
                    (CPU INT08U
                    (CPU FNCT VOID) BSP Tmr TickISR Handler);
    BSP IntEn(BSP INT ID TMR0);
    pclk freq = BSP CPU PclkFreq();
                                                                                   /* Note 2 */
    tmr_reload = pclk_freq / OS_TICKS_PER_SEC;
    TOTCR
                = 0;
                                                                                   /* Note 3 */
    TOPC
                = 0;
    TOMRO
               = tmr reload;
               = DEF BIT 00 | DEF BIT 01;
    TOMCR
               = 0;
    TOCCR
               = 0;
    TOEMR
                = DEF BIT 00;
    TOTCR
```

Listing 5-2, BSP_Tmr_TickInit()

Listing 5-2, Note 1: The tick ISR handler is programmed into the vectored interrupt controller.

Listing 5-2, Note 2: The number of counts per tick is calculated

Listing 5-2, Note 2: The calculated re-load value is programmed into the Timer 0, the timer interrupt is enabled and the timer is started

Listing 5-3, BSP_Tmr_TickISR_Handler()

Listing 5-3, Note 1: The timer 0 interrupt is cleared.

Listing 5-3, Note 2: OSTimeTick () informs µC/OS-II of the tick interrupt.

6. **µC/Probe**

 μ C/Probe is a Windows program which retrieves the values of global variables from a connected embedded target and displays the values in a engineer-friendly format. To accomplish this, an ELF file, created by the user's compiler and containing the names and addresses of all the global symbols on the target, is monitored by μ C/Probe. The user places components (such as gauges, labels, and charts) into a Data Screen in a μ C/Probe workspace and assigns each one of these a variable from the Symbol Browser, which lists all symbols from the ELF file. The symbols associated with components placed on an open Data Screen will be updated after the user presses the start button (assuming the user's PC is connected to the target).

A small section of code resident on the target receives commands from the Windows application and responds to those commands. The commands ask for a certain number of bytes located at a certain address, for example, "Send 16 bytes beginning at 0x0040102C". The Windows application, upon receiving the response, updates the appropriate component(s) on the screens with the new values.





To use $\mu C/Probe$ with the example project (or your application), do the following:

 Download and Install µC/Probe. A trial version of µC/Probe can be downloaded from the Micriµm website at

http://www.micrium.com/products/probe/probe.html

IAR Kickstart Kits Users

If this development board is part of the IAR Kickstart Kit a demo version of **µC/Probe** is already included in the installation CD. Please refer to the application note **AN-9913** for more details in how to use the demo version of **µC/Probe** with the IAR Kickstart kits.

Open µC/Probe. If the trail version was installed, open the example µC/Probe workspace for µC/OS-II, named OS-Probe-Workspace.wsp, which should be located in your installation directory at

/Program Files//Micrium/uC-Probe/Target/Plugins/uCOS-II/Workspace

If the demo version was installed open the example workspace for the IAR Kickstarts kits named *OS-Probe-Kickstart-Demo-Workspace.wsp*, which should be located in your installation directory at

/Micrium/Software/uC-Probe/Target/Demo/KSD/Workspace

- 3. **Connect Target to PC**. Currently, **µC/Probe** can use RS-232 to retrieve information from the target. You should connect a RS-232 cable between your target and computer.
- 4. Load Your ELF File. The example projects included with this application note are already configured to output an ELF file. (If you are using your own project, please refer to Appendix A of the µC/Probe user manual for directions for generating an ELF file with your compiler.) This file should be in

/<Project Directory>/<Configuration Name>/exe/

where *<Project Directory>* is the directory in which the IAR EWARM project is located (extension *.ewp) and *<Configuration Name>* is the name of the configuration in that project which was built to generate the ELF file and which will be loaded onto the target. The ELF file will be named

<*Project Name>.elf*

in EWARM v4.4x and

<Project Name>.out

in EWARM v5.xx unless you specify otherwise. To load this ELF file, right-click on the symbol browser and choose "Add Symbols".

5. Configure the RS-232 Options. In µC/Probe, choose the "Options" menu item on the "Tools" menu. A dialog box as shown in Figure 6-2 (left) should appear. Choose the "RS-232" radio button. Next, select the "RS-232" item in the options tree, and choose the appropriate COM port and baud rate. The baud rate for the projects accompanying this appnote is 115200.



Start Running. You should now be ready to run µC/Probe. Just press the run button ([▶]) to see the variables in the open data screens update. Figure 6-3 displays the µC/OS-II workspace which displays detailed information about each task's state.

Options	2	Options	
Communication FS-232 J-Link TCP/IP USB Environment General Screen Target	Settings Image: Seting	Communication J-Link TCP/IP USB Environment General Screen Target	Settings COM Port: COM7 V Baud Rate: 115200 V
	OK Cancel Apply		OK Cancel Apply

Figure 6.2. µC/Probe Options

0S: General Info	5' ask n		0. 1001010	, ougo	oo. rain otaan	Jougo Co	. ooringaration (a	orioraly
Task Stack	Infor	mati	ion					
Name	Stac	k er N	Stack	Usage Current	Starts @	ack Fnds @]	
uC/OS-II Idle	0x00201	DB0	80/512	72/512	0x00201DF8	0x00201BF8		
uC/OS-II Stat	0x00201	BA0	132/512	88/512	0x00201BF8	0x002019F8		
Start Task	0x00201	1390	196/512	104/512	0x002013F8	0x002011F8		
Probe OS PlugIn	0x00202	2188	156/512	112/512	0x002021F8	0x00201FF8		
RSD LED Task	0x00201	1998	140/512	96/512	0x002019F8	0x002017F8		
Keyboard	0x00201	590	148/512	104/512	0x002011F8	0x00200DF8		
Probe Str	0x00201	798	172/512	96/512	0x002017F8	0x002015F8		
General Tas	k Info	orm	ation					
General Tas	k Info	Priori	ation	T ite Dela	ask Status ay Waiting (On Message	Context Switches	Curren CPU Usa
General Tas Name uC/OS-II Idle	k Info	Priori 31	ation ity Sta	ite Dela idy	ask Status ay Waiting (Dn Message	Context Switches 118860	Curren CPU Usa 90.51%
General Tas Name uC/OS-II Idle uC/OS-II Stat	ID 65535 65534	Priori 31 30	ation ity Sta Rea Del	ite Dela idy ay 9	ask Status ay Waiting (Dn Message	Context Switches 118860 10116	Curren CPU Usa 90.51% 1.34%
General Tas Name uC/05-II Idle uC/05-II Stat Start Task	ID 65535 65534 5	Priori 31 30 5	ation ity Sta Del Del	tte Dell Idy Iay 9 Iay 1	ask Status ay Waiting (Dn Message	Context Switches 118860 10116 20062	Curren CPU Usa 90.51% 1.34% 0.24%
Seneral Tas Name uC/05-II Idle uC/05-II Stat Start Task Probe OS PlugIn KSD IED Tast	ID 65535 65534 5 7 8	Priori 31 30 5 7	ation ity Sta Del Del Del	tte Dell Idy Iay 9 Iay 1 Iay 1	ask Status ay Waiting (Dn Message	Context Switches 118860 10116 20062 20060 100292	Curren CPU Usa 90.51% 1.34% 0.24% 2.72% 0.96%
General Tas Name UC/05-II Idle UC/05-II Stat Start Task Probe 05 PlugIn KSD LED Task Probe 52-32	ID 65535 65534 5 7 8 9	Priori 31 30 5 7 8 9	ation ity State Del Del Bea	tte Del. dy	ask Status ay Waiting (Dn Message	Context Switches 118860 10116 20062 20060 100298 19656	Curren CPU Usa 90.51% 1.34% 0.24% 2.72% 0.96% 4.16%
Seneral Tas Name uC/05-11 Idle uC/05-11 Idle Start Task Probe OS PlugIn KSD LED Task Probe R5-232 Keyboard	ID 65535 65534 5 7 8 9 4	Priori 31 30 5 7 8 9 4	ation ity Sta Del Del Del Del Del Del Del Del	tte Dela Idy	ask Status ay Waiting (-	Dn Message	Context Switches 118860 10116 20062 20060 100298 19656 10030	Curren CPU Usa 90.51% 1.34% 2.72% 0.96% 4.16% 0.12%
Seneral Tas Name uc/05-II Idle uc/05-II Idle uc/05-II Idle Start Tas Start Tas Probe 05 PlugIn KSD LED Task Probe N5-232 Reyboard Probe Str	ID 65535 65534 5 7 8 9 4 6	Priori 31 30 5 7 8 9 4 6	ation ity State Del Del Del Del Del Del Del Del Del De	tte Dela tdy	ask Status ay Waiting (Dn Message	Context Switches 118860 10116 20062 20060 100298 19656 10030 5150	Curren CPU Usa 90.51% 1.34% 0.24% 2.72% 0.96% 4.16% 0.12% 0.03%
General Tas Name uc/OS-II die uc/OS-II Stat Stat Task Probe OS PlugIn KSD LED Task Probe KS-232 Keyboord Probe Str	k Info 10 65535 65534 5 7 8 9 4 6	Priori 31 30 5 7 8 9 4 6	ation ity State Del Del Del Del Del	tte Dela day 9 ay 1 ay 1 ay 1 ay 1 ay 3 ay 3 ay 76	ask Status ay Waiting (Dn Message	Context Switches 118860 10116 20062 20060 100298 19556 10030 5150	Curren CPU Usa 90.51% 1.34% 0.24% 2.72% 0.96% 4.16% 0.12% 0.03%
General Tas Name uc/05-11 Ide uc/05-11 Stat Stat Task Probe 05 PlugTak Keyboard Probe Str	k Info ID 65535 65534 5 7 8 9 4 6	Priori 31 30 5 7 8 9 4 6	ation ity Sta Del Del Del Del Del Del Del Del	tte Dela dy	ask Status y Walting (Dn Message	Context Switches 118860 10116 20060 20060 100298 19656 10030 5150	Curren CPU Usa 90.51% 1.34% 0.24% 2.72% 0.96% 4.16% 0.12% 0.03%
General Tas Name uc/OS-II Ide uc/OS-II Istat Start Task Probe OS PlugIn KSD LED Task Probe NS-22 Keyboard Probe Str	65535 65535 65535 65535 9 4 6	Priori 31 30 5 7 8 9 4 6	ation ity sta Del Del Rez Del Del Del Del	tte Dei ddy	ask Status y Waiting (0n Message	Context Switches 118860 10116 20062 20060 100298 19656 10030 5150	Curren CPU Usa 90.51% 1.34% 0.24% 2.72% 0.96% 4.16% 0.12% 0.03%
General Tas Name uc/OS-II die uc/OS-II Stat Stat Task Probe OS PlugIn KSD LED Task Probe KS-232 Keyboard Probe Str	ID 65535 65534 5 7 8 9 4 6	Priori 31 30 5 7 8 9 4 6	ation ity sta Del Del Rea Del Del Del	te Dela dy	ašk Status ay Waiting (-	Dn Message	Context Switches 118860 10116 20062 20060 100295 19656 10030 5150	Curren CPU Usa 90.51% 1.34% 0.24% 2.724% 0.96% 4.16% 0.12% 0.03%

Figure 6-3. **µC/Probe** Run-Time: **µC/OS-II** Task Information

Licensing

 μ C/OS-II is provided in source form for FREE evaluation, for educational use or for peaceful research. If you plan on using μ C/OS-II in a commercial product you need to contact Micriµm to properly license its use in your product. We provide ALL the source code with this application note for your convenience and to help you experience μ C/OS-II. The fact that the source is provided does NOT mean that you can use it without paying a licensing fee. Please help us continue to provide the Embedded community with the finest software available. Your honesty is greatly appreciated.

References

μ**C/OS-II, The Real-Time Kernel, 2nd Edition** Jean J. Labrosse R&D Technical Books, 2002 ISBN 1-57820-103-9

Embedded Systems Building Blocks Jean J. Labrosse R&D Technical Books, 2000 ISBN 0-87930-604-1

Contacts

IAR Systems

Century Plaza 1065 E. Hillsdale Blvd Foster City, CA 94404 USA

+1 650 287 4250 +1 650 287 4253 (FAX)

e-mail: Info@IAR.com WEB: http://www.IAR.com

Micriµm 949 Crestview Circle Weston, FL 33327 USA

+1 954 217 2036 +1 954 217 2037 (FAX)

e-mail: Jean.Labrosse@Micrium.com WEB: http://www.Micrium.com