

OS Awareness Manual uiPLUS

TRACE32 Online Help

TRACE32 Directory

TRACE32 Index

TRACE32 Documents	
OS Awareness Manuals	
OS Awareness Manual uiPLUS	1
History	3
Overview	3
Brief Overview of Documents for New Users	3
Supported Versions	4
Configuration	5
Hooks & Internals of μ iPLUS	5
Features	6
Display of Kernel Resources	6
Task Runtime Statistics	6
Task State Analysis	7
Function Runtime Statistics	7
Task Stack Coverage	7
μ iPLUS specific Menu	7
μ iPLUS Commands	8
TASK.UIDTQ	Display data queues 8
TASK.UIFLG	Display event flags 8
TASK.UIMBX	Display mailboxes 9
TASK.UIMPF	Display fixed memory pools 9
TASK.UIMPL	Display variable memory pools 9
TASK.UISEM	Display semaphores 10
TASK.UITSK	Display tasks 10
μ iPLUS PRACTICE Functions	12

History

- | | |
|-----------|--|
| 28-Aug-18 | The title of the manual was changed from “RTOS Debugger for <x>” to “OS Awareness Manual <x>”. |
|-----------|--|

Overview

The OS Awareness for μ iPLUS contains special extensions to the TRACE32 Debugger. This manual describes the additional features, such as additional commands and statistic evaluations.

Brief Overview of Documents for New Users

Architecture-independent information:

- **“Training Basic Debugging”** (training_debugger.pdf): Get familiar with the basic features of a TRACE32 debugger.
- **“T32Start”** (app_t32start.pdf): T32Start assists you in starting TRACE32 PowerView instances for different configurations of the debugger. T32Start is only available for Windows.
- **“General Commands”** (general_ref_<x>.pdf): Alphabetic list of debug commands.

Architecture-specific information:

- **“Processor Architecture Manuals”**: These manuals describe commands that are specific for the processor architecture supported by your Debug Cable. To access the manual for your processor architecture, proceed as follows:
 - Choose **Help** menu > **Processor Architecture Manual**.
- **“OS Awareness Manuals”** (rtos_<os>.pdf): TRACE32 PowerView can be extended for operating system-aware debugging. The appropriate OS Awareness manual informs you how to enable the OS-aware debugging.

Supported Versions

Currently μ iPLUS is supported for the following versions:

- ARM processors.

Configuration

µiPLUS is a µiTRON interface layer upon Nucleus PLUS. The OS Awareness for µiPLUS is part of the OS Awareness for Nucleus PLUS. Thus, configure the OS Awareness for Nucleus PLUS and you will automatically get all µiPLUS features as well. See “[OS Awareness Manual Nucleus PLUS](#)”.

The **TASK.CONFIG** command loads an extension definition file called “nucleus.t32” (directory “~/demo/<processor>/kernel/nucleus”). It contains all necessary µiPLUS extensions.

Automatic configuration tries to locate the µiPLUS internals automatically. For this purpose all symbol tables must be loaded and accessible at any time the OS Awareness is used.

If you want to have dual port access for the display functions (display “On The Fly”), you have to map emulation or shadow memory to the address space of all used system tables.

For automatic configuration it is necessary that all system internal symbols are loaded and accessible at any time, the OS Awareness is used. Use the automatic configuration of the OS Awareness for Nucleus PLUS:

Format:

TASK.CONFIG nucleus

Hooks & Internals of µiPLUS

The OS Awareness for µiPLUS uses kernel symbols and kernel structures to find the relevant information. Be sure that the µiPLUS library is built with symbol information.

Additionally symbols and structures of the Nucleus PLUS kernel are used.

Features

The OS Awareness for μ iPLUS supports the following features.

Display of Kernel Resources

The extension defines new commands to display various kernel resources. Information on the following μ iPLUS components can be displayed:

TASK.UITSK	Tasks
TASK.UIMBX	Mailboxes
TASK.UIDTQ	Data queues
TASK.UISEM	Semaphores
TASK.UIFLG	Event flags
TASK.UIMPF	Fixed sized memory pools
TASK.UIMPL	Variable sized memory pools

For a description of the commands, refer to chapter “ **μ iPLUS Commands**”.

When working with emulation memory or shadow memory, these resources can be displayed “On The Fly”, i.e. while the target application is running, without any intrusion to the application. If using this dual port memory feature, be sure that emulation memory is mapped to all places, where μ iPLUS holds its tables.

When working only with target memory, the information will only be displayed if the target application is stopped.

Task Runtime Statistics

The time spent in a task can be evaluated statistically and displayed graphically.

This is provided by the standard Nucleus PLUS awareness.

Task State Analysis

The time different tasks are in a certain state (running, ready, suspended or waiting) can be evaluated statistically or displayed graphically.

This is provided by the standard Nucleus PLUS awareness.

Function Runtime Statistics

All function related statistic and time chart evaluations can be used with task specific information.

This is provided by the standard Nucleus PLUS awareness.

Task Stack Coverage

This is provided by the standard Nucleus PLUS awareness.

µiPLUS specific Menu

The menu file “nucleus.men” contains a menu with µiPLUS specific menu items. Load this menu with the **MENU.ReProgram** command. It is necessary to load the application symbols *before* executing the **MENU.ReProgram** command; otherwise this command will not find the µiPLUS symbols and will not create the µiPLUS menu.

You will find a new menu called **µiPLUS**.

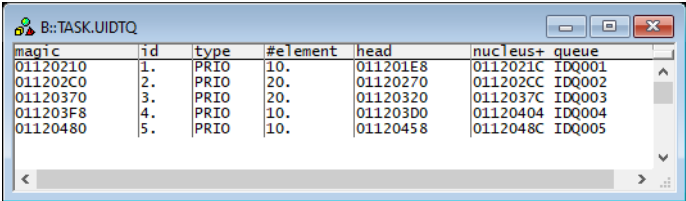
- The **Display** menu items launch the kernel resource display windows.

TASK.UIDTQ

Display data queues

Format: **TASK.UIDTQ** <data_queue>

Displays a table with the µiPLUS data queues. Specifying a queue magic number or ID will show you the queue pointers, queue message contents and the waiting tasks of that queue.



magic	id	type	#element	head	nucleus+	queue
01120210	1.	PRIQ	10.	011201E8	0112021C	IDQ001
011202C0	2.	PRIQ	20.	01120270	011202CC	IDQ002
01120370	3.	PRIQ	20.	01120320	0112037C	IDQ003
011203F8	4.	PRIQ	10.	011203D0	01120404	IDQ004
01120480	5.	PRIQ	10.	01120458	0112048C	IDQ005

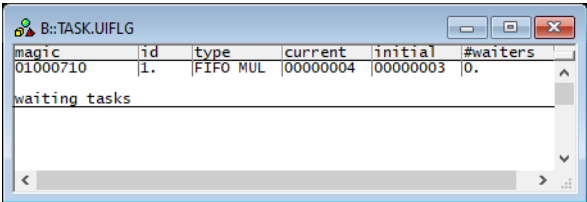
Double-click on “magic” to get more information about a specific data queue.

TASK.UIFLG

Display event flags

Format: **TASK.UIFLG** <event_flag>

Displays a table with the µiPLUS event flags. Specifying an event flag magic number or ID will show you the waiting tasks of that event flag.



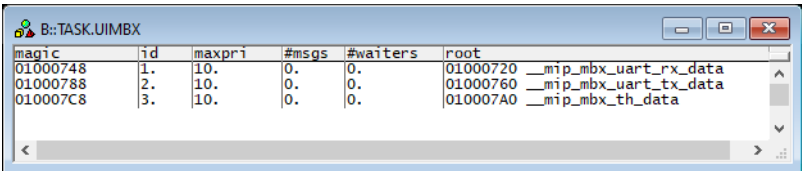
magic	id	type	current	initial	#waiters
01000710	1.	FIFO MUL	00000004	00000003	0.

waiting tasks

Double-click on “magic” to get more information about a specific event flag.

Format: **TASK.UIMBX** <mailbox>

Displays a table with the μ iPLUS mailboxes. Specifying a mailbox magic number or ID will show you the message content and the waiting tasks of that mailbox.



magic	id	maxpri	#msgs	#waiters	root
01000748	1.	10.	0.	0.	01000720 __mip_mbx_uart_rx_data
01000788	2.	10.	0.	0.	01000760 __mip_mbx_uart_tx_data
010007C8	3.	10.	0.	0.	010007A0 __mip_mbx_th_data

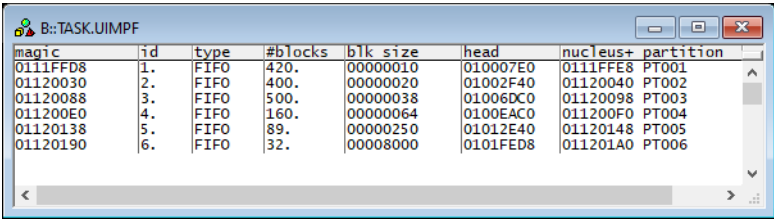
Double-click on “magic” to get more information about a specific mailbox.

TASK.UIMPF

Display fixed memory pools

Format: **TASK.UIMPF**

Displays a table with the μ iPLUS fixed sized memory pools.



magic	id	type	#blocks	blk size	head	nucleus+	partition
0111FFD8	1.	FIFO	420.	00000010	010007E0	0111FFE8	PT001
01120030	2.	FIFO	400.	00000020	01002F40	01120040	PT002
01120088	3.	FIFO	500.	00000038	01006DC0	01120098	PT003
011200E0	4.	FIFO	160.	00000064	0100EAC0	011200F0	PT004
01120138	5.	FIFO	89.	00000250	01012E40	01120148	PT005
01120190	6.	FIFO	32.	00008000	0101FED8	011201A0	PT006

Double-click on the address of the Nucleus+ partition to get more information about a specific pool.

TASK.UIMPL

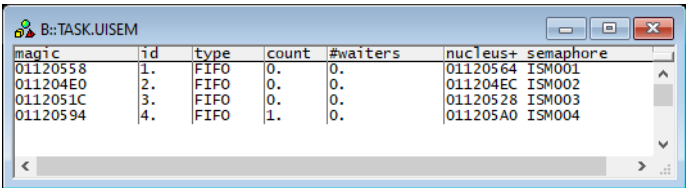
Display variable memory pools

Format: **TASK.UIMPL**

Displays a table with the μ iPLUS variable sized memory pools. Double click on the address of the Nucleus+ memory pool to get more information about a specific pool.

Format: **TASK.UISEM** <sema>

Displays a table with the µiPLUS semaphores. Specifying a semaphore magic number or ID will show you the waiting tasks of that semaphore.



magic	id	type	count	#waiters	nucleus+ semaphore
01120558	1.	FIFO	0.	0.	01120564 ISM001
011204E0	2.	FIFO	0.	0.	011204EC ISM002
0112051C	3.	FIFO	0.	0.	01120528 ISM003
01120594	4.	FIFO	1.	0.	011205A0 ISM004

Double click on “magic” to get more information about a specific semaphore.

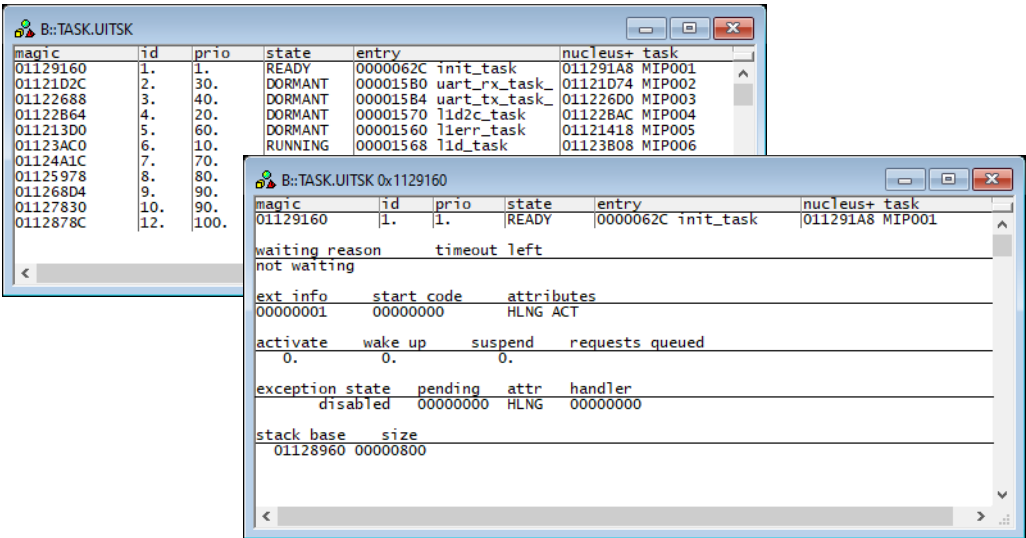
TASK.UITSK

Display tasks

Format: **TASK.UITSK** <task>

Displays the task table of µiPLUS or detailed information about one specific task.

Without any arguments, a table with all created tasks will be shown.
Specify a task ID or magic number to display detailed information on that task.



magic	id	prio	state	entry	nucleus+ task
01129160	1.	1.	READY	0000062C init_task	011291A8 MIP001
01121D2C	2.	30.	DORMANT	00001580 uart_rx_task	01121D74 MIP002
01122688	3.	40.	DORMANT	00001584 uart_tx_task	011226D0 MIP003
01122864	4.	20.	DORMANT	00001570 lid2c_task	011228AC MIP004
011213D0	5.	60.	DORMANT	00001560 lierr_task	01121418 MIP005
01123AC0	6.	10.	RUNNING	00001568 lid_task	01123B08 MIP006
01124A1C	7.	70.			
01125978	8.	80.			
011268D4	9.	90.			
01127830	10.	90.			
0112878C	12.	100.			

magic	id	prio	state	entry	nucleus+ task
01129160	1.	1.	READY	0000062C init_task	011291A8 MIP001

waiting reason timeout left
not waiting

ext info	start code	attributes
00000001	00000000	HLNG ACT

activate	wake up	suspend	requests queued
0.	0.	0.	

exception state	pending	attr	handler
disabled	00000000	HLNG	00000000

stack base	size
01128960	00000800

“magic” is a unique ID, used by the OS Awareness to identify a specific task (address of the CTSK).

The fields “magic”, “entry”, “nucleus+ task”, “stackbase” and “handler” are mouse sensitive, double clicking on them opens appropriate windows. Right-clicking a value in the **magic** column will show a local menu.

µiPLUS PRACTICE Functions

Currently there are no special PRACTICE functions for µiPLUS.