



Release Notes

V1.28.00

Micrium

For the Way Engineers Work

Revision History

Version	Date	Description
V1.28.00	2011 Feb	Re-released μ C/CPU V1.28 core files as V1.28.00 & port files as V1.28.00.00
V1.28	2010 Dec	Bug fixes and improvements
V1.27	2010 Oct	New features & improvements
V1.26	2010 Apr	Improvements
V1.25.01	2010 Apr	Port updates only—NO changes to core files
V1.25	2010 Jan	Bug fixes and improvements
V1.24	2009 Dec	New features, bug fixes, & improvements
V1.23	2009 Jul	CPU timestamp, timer, & time measurement features First version with release history & user's manual

900-uC-CPU-004

Required Modules

Version 1.28.00

µC/LIB version 1.34

New Features

Version 1.28.00

N/A

Version 1.27

V1.27-001

Added `CPU_SW_EXCEPTION()` / `CPU_SW_Exception()` to trap on unrecoverable exceptions, primarily NULL pointers to return errors (a condition which cannot be returned via the NULL return pointer). See also 'Improvements V1.27-001a'.

Version 1.26

N/A

Version 1.25.01

N/A

Version 1.25

N/A

Version 1.24

V1.24-001

Added `CPU_STK_SIZE` data type definition to each `cpu.h`.

V1.24-002a

Added (optional) CPU timestamp's timer frequency, `CPU_TS_TmrFreq_Hz`.

V1.24-002b

Added new CPU timestamp timer functions:

<code>CPU_TS_TmrFreqGet()</code>	gets the CPU timestamp's timer frequency (in Hertz)
<code>CPU_TS_TmrFreqSet()</code>	sets the CPU timestamp's timer frequency (in Hertz)

See also 'New Features V1.23-001c'.

Version 1.23

V1.23-001

Added new CPU timestamp, timer, and time measurement features. (Note that an application must call `CPU_Init()` to initialize CPU timestamp or time measurement features prior to any other calls to CPU time functions.)

V1.23-001a

Added `CPU_CFG_TS_EN` in `cpu_cfg.h` to enable/disable CPU timestamps:

<code>CPU_TS_Get()</code>	gets the current, real-time value of 64-bit CPU timestamp, returned via two 32-bit values
<code>CPU_TS_GetLo()</code>	gets only the lower 32-bits of 64-bit timestamp
<code>CPU_TS_Update()</code>	updates the real-time value of 64-bit CPU timestamp [see 'New Features V1.23-001c <code>CPU_TS_TmrRd()</code> ']

See also 'Changes V1.25-001a1 & V1.25-001c'.

V1.23-001b

Added `CPU_CFG_INT_DIS_MEAS_EN` & `CPU_CFG_INT_DIS_MEAS_OVRHD_NBR` in `cpu_cfg.h` to enable/disable measuring interrupts disabled times:

<code>CPU_IntDisMeasMaxGet()</code>	gets the maximum time interrupts are disabled, returned via a 32-bit timestamp value; this maximum value is non-resetable
<code>CPU_IntDisMeasMaxCurGet()</code>	gets the current maximum time interrupts are disabled, returned via a 32-bit timestamp value; this maximum value is resetable
<code>CPU_IntDisMeasMaxCurReset()</code>	resets the current maximum time interrupts are disabled

See also 'Changes V1.25-002'.

V1.23-001c

The following timer functions must be implemented in an application if either CPU timestamps *or* interrupts disabled time measurements are enabled:

<code>CPU_TS_TmrInit()</code>	initializes & starts a hardware (or software) timer to update CPU timestamps & time measurements
<code>CPU_TS_TmrRd()</code>	gets current hardware (or software) timer value to update CPU timestamps or time measurements
<code>CPU_TS_to_uSec()</code>	convert (up to) 64 bits of a CPU timestamp value into microseconds, returned via two 32-bit values

See also 'Changes V1.25-001d & V1.25-001e' & 'New Features V1.24-002b'.

Improvements

Version 1.28.00

V1.28.00-001

Updated μ C/CPU's CERT-C and MISRA-C compliance:

V1.28.00-001a

Removed 'u' qualifier from certain integer constants. This reverts a previously implemented improvement only for certain integer constants that may be used in both signed and unsigned expressions. See also 'Improvements V1.24-001a1'.

V1.28.00-001b

Added `const` modifier to all appropriate API function pointer arguments. See also 'Changes V1.28-001'.

Version 1.27

V1.27-001

Updated μ C/CPU's CERT-C and MISRA-C compliance:

V1.27-001a

Added `CPU_SW_EXCEPTION()` / `CPU_SW_Exception()` to trap on unrecoverable exceptions, primarily `NULL` pointers to return errors (a condition which cannot be returned via the `NULL` return pointer).

V1.27-001a1

Modified functions to trap `NULL` 'p_err' pointers.

Version 1.26

V1.26-001

Updated μ C/CPU's CERT-C and MISRA-C compliance:

V1.26-001a

Added argument names to function pointer data types.

V1.26-001b

Encapsulated all macros defined as code blocks within `do...while(0)` conditions.

Version 1.25.01

N/A

Version 1.25

V1.25-001a

Improved CPU timestamp API & performance. See also 'Changes V1.25-001'.

V1.25-002a

Refactored `CPU_CntLeadZeros()` to improve performance.

V1.25-002b

Added 64-bit support to `CPU_CntLeadZeros()`.

V1.25-003

Added 64-bit data types to most `cpu.h`'s.

Version 1.24

V1.24-001

Updated μ C/CPU's CERT-C and MISRA-C compliance:

V1.24-001a1

Appended unsigned 'u' qualifier to all unsigned integer constants.

V1.24-001a2

Removed redundant 'L' qualifier from all long integer constants.

V1.24-001b

Replaced all calls to unbounded μ C/LIB string library functions [e.g. `Str_Copy()`] with calls to bounded functions [e.g. `Str_Copy_N()`].

Version 1.23

V1.23-001

Added `CPU_CFG_MODULE_PRESENT` header guard to ensure `cpu_cfg.h` is processed only once, regardless if `#include'd` by multiple source or header files.

Changes

Version 1.28.00

V1.28.00-001

Added const modifier to all appropriate pointer arguments in the following functions:

CPU_NameSet ()

Version 1.27

N/A

Version 1.26

N/A

Version 1.25.01

V1.25.01-001a

Renamed `\Micrium\Software\uC-CPU\Win32\Microsoft` directory to `\Micrium\Software\uC-CPU\Win32\Visual Studio`.

V1.25.01-001b

Refactored `\Micrium\Software\uC-CPU\Win32\Visual Studio` port files' critical section initialization & implementation.

Version 1.25

V1.25-001

Refactored CPU timestamps configuration, API, & implementation to improve performance (see also 'uC/CPU's User's Manual Section 3.03'):

V1.25-001a1

Replaced `cpu_cfg.h` configuration constant `CPU_CFG_TS_EN` with new configuration constants:

<code>CPU_CFG_TS_32_EN</code>	enables 32-bit CPU timestamps
<code>CPU_CFG_TS_64_EN</code>	enables 64-bit CPU timestamps

V1.25-001a2

Added `cpu_cfg.h` configuration constant `CPU_CFG_TS_TMR_SIZE` to configure the word size of the CPU timestamp's hardware (or software) timer.

V1.25-001b1

Replaced CPU_TS data type with new CPU timestamp data types:

CPU_TS32	handles 32-bit CPU timestamps
CPU_TS64	handles 64-bit CPU timestamps

V1.25-001b2

Added CPU_TS_TMR data type to handle CPU timestamp timer values instead of CPU_TS.

V1.25-001c

Replaced CPU_TS_Get () & CPU_TS_GetLo () with new CPU timestamp functions:

CPU_TS_Get32 ()	gets 32-bit CPU timestamp
CPU_TS_Get64 ()	gets 64-bit CPU timestamp

V1.25-001d

Modified developer-defined CPU timestamp timer function prototypes:

```
void          CPU_TS_TmrInit(void);  
CPU_TS_TMR   CPU_TS_TmrRd   (void);
```

V1.25-001e

Replaced (optional) developer-defined CPU_TS_to_uSec () with new CPU timestamp functions:

CPU_TS32_to_uSec ()	converts 32-bit CPU timestamp to microseconds
CPU_TS64_to_uSec ()	converts 64-bit CPU timestamp to microseconds

V1.25-002

Modified CPU interrupts disabled time measurement function prototypes:

```
CPU_TS_TMR   CPU_IntDisMeasMaxCurReset(void);  
CPU_TS_TMR   CPU_IntDisMeasMaxCurGet   (void);  
CPU_TS_TMR   CPU_IntDisMeasMaxGet       (void);
```

Version 1.24

N/A

Version 1.23

V1.23-001a

Moved CPU_ERR data type definition from each cpu_cfg.h to cpu_core.h.

Corrections

Version 1.28

N/A

Version 1.27

N/A

Version 1.26

N/A

Version 1.25.01

N/A

Version 1.25

V1.25-001

Previous `CPU_TS_Get()` failed to re-entrantly calculate the current CPU timestamp since the current CPU timestamp timer was read [via a call to `CPU_TS_TmrRd()`] with interrupts enabled but saved for the next timestamp calculation with interrupts disabled. Fixed in `CPU_TS_Get32()` & `CPU_TS_Get64()` [see 'Changes V1.25-001c'] by calling `CPU_TS_TmrRd()` with interrupts disabled.

Version 1.24

N/A

Version 1.23

N/A

Known Problems

Version 1.28.00

Version 1.27

Version 1.26

Version 1.25.01

Version 1.25

Version 1.24

Version 1.23

N/A

Limitations

001

Support for 64-bit address/data not available for some CPUs

Contacts

Micrium

1290 Weston Road, Suite 306
Weston, FL 33326
USA

Phone: +1 954 217 2036

Fax: +1 954 217 2037

E-mail: Licensing@Micrium.com

Web: www.Micrium.com