

I INDEX

A

- abstract base class of a thread [4-6](#)
- abstracting hardware implementation [4-30](#)
- allocating memory with malloc/new [4-5](#), [4-7](#)
- allocating system resources for threads [4-5](#)
- Analog products [1-2](#)
- API
 - assembly macros [6-59](#)
 - format of library reference [6-9](#)
 - header (vdk.h) [6-2](#)
 - linking library functions [6-2](#)
 - list of functions [6-5](#)
 - naming conventions [6-4](#)
 - passing function parameters [6-3](#)
 - working with library headers [6-2](#)
- application programming interface (API) [A-1](#)
- architecture of ISRs [4-26](#)

B

- behavior of device flags [4-24](#)
- behavior of events [4-20](#)
- behavior of semaphores [4-16](#)
- Bitfield data type [5-3](#)

- blocking on semaphores [4-17](#), [4-18](#)
- boot thread [4-5](#)

C

- C and assembly threads [4-7](#)
- C++ threads [4-6](#)
- calling library functions [6-2](#)
- ClearEventBit() [4-21](#), [4-23](#), [6-10](#)
- ClearThreadError() [6-11](#)
- CloseDevice() [4-33](#), [4-34](#), [6-12](#)
- code reuse [2-3](#)
- Communication Manager [4-30](#)
- configuring a project [3-2](#)
- configuring a VDK project [3-2](#)
- constructor of a thread [4-4](#), [4-5](#)
- context switch [4-12](#), [4-18](#), [4-23](#), [4-26](#), [4-29](#), [A-1](#)
- conventions of this manual [1-6](#)
- cooperative multitasking [4-10](#)
- cooperative scheduling [4-10](#)
- create function of a thread [4-4](#)
- CreateThread() [4-4](#), [6-13](#)
- critical region [4-9](#), [4-25](#), [4-26](#), [4-27](#), [4-31](#), [4-33](#), [4-39](#), [A-1](#)
- customer support [1-2](#)

INDEX

D

data type

- Bitfield [5-3](#)
 - DeviceDescriptor [5-4](#)
 - DeviceDispatchID [5-5](#)
 - DeviceDispatchUnion [5-6](#)
 - DeviceFlagID [5-8](#)
 - EventBitID [5-9](#)
 - EventData [5-10](#)
 - EventID [5-9](#)
 - Priority [5-11](#)
 - SemaphoreID [5-12](#)
 - SystemError [5-16](#)
 - ThreadID [5-12](#)
 - ThreadStatus [5-13](#)
 - ThreadType [5-14](#)
 - Ticks [5-15](#)
 - VersionStruct [5-18](#)
- debugged control structures [2-2](#)
- debugging a VDK project [3-44](#)
- general tips [3-50](#)
- description format of API reference [6-9](#)
- DestroyThread() [4-5](#), [6-14](#)
- destructor function of a thread [4-5](#)
- device driver [2-12](#), [4-30](#)
- activate function [4-37](#)
 - dispatch function [4-29](#), [4-30](#), [4-31](#)
 - execution [4-30](#)
 - general notes [4-39](#)
 - init function [4-33](#)
 - initialization [4-30](#)
 - interaction with ISRs [4-30](#)

- open/close by a thread [4-33](#)
- device flag [4-24](#), [4-37](#)
- posting [4-24](#)
- DeviceDescriptor data type [5-4](#)
- DeviceDispatchID data type [5-5](#)
- DeviceDispatchUnion data type [5-6](#)
- DeviceFlagID data type [5-8](#)
- DeviceID [4-36](#)
- DeviceIOctl() [4-35](#), [6-15](#)
- disabling interrupts [2-9](#), [4-26](#)
- disabling scheduling [2-9](#), [4-11](#)
- dispatch function of a device driver [4-31](#)
- DispatchThreadError() [4-4](#), [6-16](#)
- DispatchUnion [4-32](#)
- documented library functions [6-5](#)
- domain
- interrupt domain [4-18](#)
 - thread domain [4-18](#)
- dynamic thread creation [4-4](#)

E

- effect of unscheduled regions on
- event calculation [4-22](#)
- entering the scheduler from API calls [4-12](#)
- entering the scheduler from
- interrupts [4-12](#)
- enumeration
- device driver's values [4-31](#)
 - error codes [4-8](#)
 - error function [4-4](#)
 - priority of a thread [4-3](#)
- error codes [4-8](#)

error function of a thread [4-4](#)

error handling facilities [4-8](#)

ErrorFunction() [4-8](#)

event [A-2](#)

calculating value of [4-20](#)

calculation [4-21](#)

combination of event bits [4-19](#)

loading new event data [4-24](#)

mask flag [4-20](#)

matchAll flag [4-20](#)

recalculation [4-23](#), [4-24](#)

setting up [4-20](#)

state (true of false) [4-19](#)

state of (true or false) [4-19](#)

triggering [4-24](#)

values flag [4-20](#)

VDK_EventData [4-20](#)

event bits [4-19](#)

changing status from interrupt

domain [4-23](#)

changing status from thread

domain [4-23](#)

eventbit [A-2](#)

EventBitID data type [5-9](#)

EventData data type [5-10](#)

EventID data type [5-9](#)

EXTERN assembler directive [4-7](#)

F

FreeDestroyedThreads() [6-17](#)

freeing memory with free/delete [4-5](#)

freeing thread's resources [4-5](#)

G

GetEventBitValue() [6-18](#)

GetEventData() [4-24](#), [6-19](#)

GetEventValue() [6-20](#)

GetLastThreadError() [4-8](#), [6-21](#)

GetLastThreadErrorValue() [4-8](#),
[6-22](#)

GetPriority() [6-23](#)

GetSemaphoreValue() [6-24](#)

GetThreadHandle() [4-7](#), [6-25](#)

GetThreadID() [4-2](#), [6-26](#)

GetThreadStatus() [6-27](#)

GetThreadType() [6-28](#)

GetUptime() [6-29](#)

GetVersion() [6-30](#)

global data [4-27](#)

global state of event bits [4-20](#)

global variables [4-7](#)

GLOBAL, assembler directive [4-7](#)

H

hardware abstraction [2-4](#)

header files [3-3](#)

header files for C/asm written

threads [4-7](#)

history buffer [3-44](#)

history log [3-45](#)

I

idle thread [3-48](#), [4-5](#), [4-14](#)

In This Chapter [5-1](#), [6-1](#)

inDestroyNow [4-5](#)

init function of a device driver [4-33](#)

init function of a thread [4-4](#)

INDEX

- init function/constructor [4-5](#)
- InitFunction() (C/asm threads) [4-4](#), [4-5](#)
- instrumented build info [3-44](#)
- interrupt [A-3](#)
- interrupt domain [4-18](#)
- interrupt domain with hardware scheduling [2-12](#)
- interrupt handler [4-27](#)
- interrupt handling [4-26](#)
- interrupt latency [4-26](#), [4-27](#)
- interrupt nesting [4-27](#)
- Interrupt Service Routine (ISR) [A-3](#)
- Interrupt Service Routine (see ISR)
- interrupt vector table [4-27](#)
- invoking scheduler
 - from APIs [4-12](#)
 - from ISRs [4-12](#)
- IOCntl_t struct [4-35](#)
 - command [4-36](#)
 - dataH [4-36](#)
 - parameters [4-36](#)
 - timeout [4-36](#)
- ISR
 - activate a device driver [4-36](#)
 - architecture of [4-26](#)
 - assembly-implemented [4-26](#)
 - communication with thread domain [4-28](#)
 - disabling [4-26](#)
 - entering the scheduler [4-12](#)
 - re-enabling [4-26](#)

L

- LDF (linker description file) [3-2](#)
- library format [6-9](#)
- library linking at build time
 - with defined events [4-19](#)
 - without defined events [4-19](#)
- linker description file [3-2](#)
- linking library functions [6-2](#)
- linking with thread-safe standard libraries [3-2](#)
- LoadEvent() [4-24](#), [6-32](#)
- LogHistoryEvent() [6-31](#)
- lowest priority interrupt [4-27](#), [4-29](#), [4-36](#)

M

- MakePeriodic() [6-33](#)
- manual
 - contents description [1-4](#)
 - conventions [1-6](#)
 - intended audience [1-3](#)
 - new in this edition [1-5](#)
 - related documents [1-5](#)
- memory allocation
 - malloc or new [4-5](#)
- methodologies of scheduling [4-9](#)
- motivation [2-2](#)
- multitasking [4-10](#)

N

- namespace
 - VDK [4-6](#), [6-4](#)
- notation conventions
 - of API library [6-4](#)

of this manual [1-6](#)
 number of events in a system [4-19](#)

O

open/close functions of a thread
[4-33](#)
 OpenClose_t struct
[4-34](#)
 dataH [4-34](#)
 flags [4-34](#)
 OpenDevice() [4-33](#), [4-34](#), [6-35](#)
 operations in thread's windows [3-47](#)

P

partitioning an application [2-5](#)
 passing function parameters [6-3](#)
 PendDeviceFlag() [4-24](#), [6-36](#)
 PendEvent() [4-23](#), [6-37](#)
 pending on a device flag [4-37](#)
 pending on a semaphore [4-17](#)
 PendSemaphore() [4-17](#), [6-39](#)
 periodic semaphores [4-17](#), [4-18](#)
 PopCriticalRegion() [4-26](#), [6-41](#)
 PopNestedCriticalRegions() [6-42](#)
 PopNestedUnscheduledRegions()
[4-12](#), [6-43](#)
 PopUnscheduledRegion() [4-21](#),
[6-44](#)
 PostDeviceFlag() [4-24](#), [6-45](#)
 posting a device flag [4-38](#)
 posting a semaphore [4-18](#)
 PostSemaphore() [4-18](#), [4-19](#), [6-46](#)
 preemption [2-8](#)
 pre-emptive kernel [A-4](#)

pre-emptive scheduling [4-10](#)
 Priority data type [5-11](#)
 priority of a thread [2-8](#), [3-50](#), [4-3](#)
 changing dynamically [4-3](#)
 default priority [4-3](#)
 highest priority [4-3](#)
 project (see VDK project)
 protected regions [2-9](#), [4-7](#), [4-9](#), [4-35](#)
 nested [4-11](#)
 notes on [4-39](#)
 PushCriticalRegion() [4-26](#), [6-47](#)
 PushUnscheduledRegion() [6-48](#)

Q

queue of ready threads (see ready
 queue)
 queue of wait threads [4-3](#)

R

rapid application development [2-2](#)
 ReadWrite_t struct [4-34](#)
 data [4-35](#)
 dataH [4-35](#)
 dataSize [4-35](#)
 timeout [4-35](#)
 ready queue [2-6](#), [4-9](#), [4-10](#), [4-13](#),
[4-18](#), [4-19](#), [4-24](#), [6-37](#)
 Real-Time Operating System
 (RTOS) [A-3](#)
 recalculating an event [4-20](#)
 re-enabling interrupts [4-26](#)
 reentrancy [4-7](#)
 related documents [1-5](#)
 RemovePeriodic() [6-50](#)

INDEX

- reschedule ISR [4-29](#)
- ResetPriority() [4-3](#), [6-51](#)
- round-robin scheduling [4-10](#)
- RTOS kernel [A-4](#)
- run function of a thread [4-3](#)
- Run() (C++ run function of a thread) [4-3](#)
- RunFunction() (C/asm run function of a thread) [4-3](#)

- S
- scheduled region [4-23](#)
- scheduler [4-18](#), [4-23](#), [4-24](#), [A-4](#)
- scheduling [2-6](#), [4-9](#)
 - disable [4-11](#)
 - enable [4-18](#)
 - non-deterministic operations [4-19](#)
 - when a thread pends on a semaphore [4-18](#), [4-20](#)
 - when a thread pends on an event [4-20](#)
- scheduling methods/modes [4-9](#)
 - cooperative [4-10](#)
 - pre-emptive [4-10](#)
 - round-robin [4-10](#), [A-4](#)
- scheduling of periodic semaphores [4-18](#)
- scope of a thread's data members [4-6](#)
- semaphore [4-16](#), [A-4](#)
 - pending on [4-17](#)
 - periodic [4-17](#)
 - posting of [4-17](#)
- semaphore posting from
 - interrupt domain [4-18](#)
 - thread domain [4-18](#)
- SemaphoreID data type [5-12](#)
- SetEventBit() [4-20](#), [4-23](#), [6-53](#)
- SetPriority() [4-3](#), [6-54](#)
- SetThreadError() [6-55](#)
- signal.h [4-26](#)
- signals [4-16](#)
 - device flags [4-24](#)
 - events and event bits [4-19](#)
 - semaphores [4-16](#)
- Sleep() [6-52](#)
- software interrupt [4-18](#)
- source templates [4-4](#)
- stack overflows [4-3](#)
- stack size of a thread [4-3](#)
- standard C/C++ libraries,
 - thread-safe versions [3-2](#)
- state of the system [4-19](#)
- storing event bits' state in a global variable [4-20](#)
- synchronization of threads [4-16](#)
 - device flags [4-24](#)
 - events and event bits [4-19](#)
 - semaphores [4-16](#)
- SyncRead() [4-34](#), [6-56](#)
- SyncWrite() [4-34](#), [6-57](#)
- System Configurator [A-5](#)
- System State History window [3-44](#)
- SystemError data type [5-16](#)

- T
- Target Load window [3-48](#)
- technical support [1-2](#)

- thread 4-2, A-5
 - blocked 2-8
 - blocking 4-24
 - C, C++, or asm-implemented 4-6
 - controlling a device's parameters 4-35
 - create function 4-4
 - default priority 4-3
 - destructor function 4-5
 - dynamic creation with
 - malloc/new 4-5
 - entry point 4-3
 - error function 4-3, 4-17
 - header file 4-4
 - interaction with device flags 4-24
 - interaction with semaphores 4-17
 - lowest priority (idle) 3-48
 - open/close a device driver 4-34
 - pending on a device flag 4-24
 - pending on events 4-19, 4-23
 - priority of 3-50
 - read/write to/from a device driver 4-34
 - required functionality 4-3
 - run() function 4-3
 - scope of a thread 4-5
 - scope of data members 4-6
 - setting/clearing of event bits 4-23
 - specific parameters 4-2
 - stack size 4-3
 - state and status data 3-49
 - status of 3-47
 - swapping threads 4-12
 - thread and hardware interaction 2-11
 - thread domain 4-18, 4-28
 - calling dispatch function from 4-32
 - thread domain with software
 - scheduling 2-11
 - thread parameters 4-2
 - thread template
 - C++ 4-6
 - thread types 4-2
 - thread's interaction with events 4-22
 - thread's interaction with
 - semaphores 4-17
 - ThreadId 4-2
 - ThreadId data type 5-12
 - threads implemented in
 - C and assembly 4-7
 - C++ 4-6
 - thread-safe libraries 3-2
 - ThreadStatus data type 5-13
 - ThreadType data type 5-14
 - Ticks data type 5-15
 - time slicing 4-10, A-4
 - timed system events 4-29
 - timeout 4-17
 - timer ISR 4-29
 - timer interrupt 4-27
 - triggering multiple events 4-22

INDEX

U

unscheduled region [4-11](#), [4-13](#),
[4-22](#), [4-23](#), [4-26](#), [4-33](#), [4-39](#),
[A-5](#)
unscheduled regions [4-9](#)
using keyword [4-6](#)
using the compiler's built-in
functions [6-2](#)

V

variable access, ISR and device
driver [4-39](#)
Variables [4-39](#)
variables [4-39](#)
VDK
Communication Manager [4-30](#)
Status window [3-49](#)
System State History Window
[3-45](#)
Target Load graph [3-48](#)
Thread Status and Thread Event
windows [3-46](#)
VDK library
assembly macros [6-59](#)
functions, list of [6-5](#)
header (vdk.h) [6-2](#)
linking library functions [6-2](#)
naming conventions [6-4](#)
reference format [6-9](#)
working with vdk library header
[6-2](#)
VDK namespace [4-6](#), [6-4](#)
VDK project
adding a boot thread [3-14](#)

adding a device driver [3-38](#)
adding a new boot thread [3-14](#)
adding a new thread type [3-10](#)
adding a round-robin priority
[3-18](#)
adding a semaphore [3-22](#)
adding a thread type [3-10](#)
adding an event [3-28](#)
adding an event bit [3-25](#)
adding an interrupt [3-34](#)
adding device flags to a device
driver [3-42](#)
adding event bits to an event [3-31](#)
changing the order of boot
threads' creation [3-17](#)
creating a VDK-enabled project
[3-4](#)
deleting a boot thread [3-16](#)
deleting a device driver [3-41](#)
deleting a round-robin priority
[3-21](#)
deleting a semaphore [3-24](#)
deleting a thread type [3-13](#)
deleting an event [3-30](#)
deleting an event bit [3-27](#)
deleting an Interrupt [3-37](#)
linker description file [3-2](#)
modifying a boot thread's
properties [3-15](#)
modifying a dependent event bit's
condition [3-33](#)
modifying a device driver's
properties [3-40](#)
modifying a round-robin's

- priorities [3-20](#)
- modifying a round-robin's
 - priority [3-21](#)
- modifying a semaphore's
 - properties [3-23](#)
- modifying a thread type's
 - properties [3-12](#)
- modifying an event bit's
 - properties [3-26](#)
- modifying an event's properties [3-29](#)
- modifying an interrupt's
 - properties [3-36](#)
- modifying system parameters [3-8](#)
- renaming a boot thread [3-16](#)
- renaming a semaphore [3-24](#)
- renaming an event [3-30](#)
- renaming an event bit [3-27](#)
- starting a VDK-enabled project [3-4](#)
- VDK.h header file [3-3](#)
- VDK_DispatchUnion [4-34](#), [4-35](#)
- VDK_EventData [4-20](#), [4-22](#), [4-24](#)
- VDK_ISR_ACTIVATE_DEVICE
 - _DRIVER_ [6-60](#)
- VDK_ISR_ACTIVATE_DEVICE
 - _DRIVER_() [4-36](#)
- VDK_ISR_CLEAR_EVENTBIT_
 - [6-61](#)
- VDK_ISR_CLEAR_EVENTBIT_
 - () [4-23](#)
- VDK_ISR_LOG_HISTORY_EVENT
 - NT_ [6-62](#)

- VDK_ISR_POST_SEMAPHORE
 - _ [6-63](#)
- VDK_ISR_POST_SEMAPHORE
 - _() [4-18](#), [4-19](#), [4-28](#)
- VDK_ISR_SET_EVENTBIT_
 - [6-64](#)
- VDK_ISR_SET_EVENTBIT_()
 - [4-23](#)
- VDK_kDDActivate [4-36](#)
- VDK_kDDInit [4-33](#)
- VDK_kDDIOCntl [4-35](#)
- VDK_kDDOpen [4-33](#)
- VDK_kDDSyncRead [4-34](#)
- VDK_kDDSyncWrite [4-34](#)
- vector table [4-27](#)
- VersionStruct data type [5-18](#)
- VisualDSP++
 - State History graph [3-44](#)
 - System State History window [3-45](#)
 - Target Load graph [3-48](#)
 - Thread History window [4-15](#)
 - Thread Status and Thread Event windows [3-46](#)
 - Thread Status window [3-49](#)
 - VDK Status window [3-49](#)
- VisualDSP++ Kernel (VDK) [A-5](#)

W

- working with library headers [6-2](#)
- writing threads in different languages [4-6](#)

INDEX

Y

Yield() [6-58](#)