

POK

Generated by Doxygen 1.8.1.2

Fri Jan 25 2013 11:46:26



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List	3
<b>3</b>	<b>Class Documentation</b>	<b>11</b>
3.1	<a href="#">__attribute__ Struct Reference</a>	11
3.1.1	<a href="#">Detailed Description</a>	11
3.2	<a href="#">ARINC_ATTRIBUTE Struct Reference</a>	11
3.2.1	<a href="#">Detailed Description</a>	12
3.3	<a href="#">bf_key_st Struct Reference</a>	12
3.3.1	<a href="#">Detailed Description</a>	12
3.4	<a href="#">BLACKBOARD_STATUS_TYPE Struct Reference</a>	12
3.4.1	<a href="#">Detailed Description</a>	12
3.5	<a href="#">BUFFER_STATUS_TYPE Struct Reference</a>	12
3.5.1	<a href="#">Detailed Description</a>	13
3.6	<a href="#">DES_ks Struct Reference</a>	13
3.6.1	<a href="#">Detailed Description</a>	13
3.7	<a href="#">ERROR_STATUS_TYPE Struct Reference</a>	13
3.7.1	<a href="#">Detailed Description</a>	13
3.8	<a href="#">EVENT_STATUS_TYPE Struct Reference</a>	13
3.8.1	<a href="#">Detailed Description</a>	14
3.9	<a href="#">exception Struct Reference</a>	14
3.9.1	<a href="#">Detailed Description</a>	14
3.10	<a href="#">ieee_double_shape_type Union Reference</a>	14
3.10.1	<a href="#">Detailed Description</a>	14
3.11	<a href="#">ieee_float_shape_type Union Reference</a>	14
3.11.1	<a href="#">Detailed Description</a>	15
3.12	<a href="#">PARTITION_STATUS_TYPE Struct Reference</a>	15
3.12.1	<a href="#">Detailed Description</a>	15
3.13	<a href="#">pok_allocator_space_t Struct Reference</a>	15

3.13.1 Detailed Description . . . . .	15
3.14 pok_arinc653_event_layer_t Struct Reference . . . . .	15
3.14.1 Detailed Description . . . . .	16
3.15 pok_arinc653_semaphore_layer_t Struct Reference . . . . .	16
3.15.1 Detailed Description . . . . .	16
3.16 pok_blackboard_status_t Struct Reference . . . . .	16
3.16.1 Detailed Description . . . . .	16
3.17 pok_blackboard_t Struct Reference . . . . .	16
3.17.1 Detailed Description . . . . .	17
3.18 pok_buffer_status_t Struct Reference . . . . .	17
3.18.1 Detailed Description . . . . .	17
3.19 pok_buffer_t Struct Reference . . . . .	17
3.19.1 Detailed Description . . . . .	17
3.20 pok_error_report_t Struct Reference . . . . .	18
3.20.1 Detailed Description . . . . .	18
3.21 pok_error_status_t Struct Reference . . . . .	18
3.21.1 Detailed Description . . . . .	18
3.22 pok_lockobj_attr_t Struct Reference . . . . .	18
3.22.1 Detailed Description . . . . .	18
3.23 pok_lockobj_lockattr_t Struct Reference . . . . .	19
3.23.1 Detailed Description . . . . .	19
3.24 pok_mutex_attr_t Struct Reference . . . . .	19
3.24.1 Detailed Description . . . . .	19
3.25 pok_port_queueing_status_t Struct Reference . . . . .	19
3.25.1 Detailed Description . . . . .	19
3.26 pok_port_sampling_status_t Struct Reference . . . . .	20
3.26.1 Detailed Description . . . . .	20
3.27 pok_queue_t Struct Reference . . . . .	20
3.27.1 Detailed Description . . . . .	20
3.28 pok_syscall_args_t Struct Reference . . . . .	20
3.28.1 Detailed Description . . . . .	20
3.29 pok_thread_attr_t Struct Reference . . . . .	21
3.29.1 Detailed Description . . . . .	21
3.30 PROCESS_ATTRIBUTE_TYPE Struct Reference . . . . .	21
3.30.1 Detailed Description . . . . .	21
3.31 PROCESS_STATUS_TYPE Struct Reference . . . . .	21
3.31.1 Detailed Description . . . . .	22
3.32 QUEUING_PORT_STATUS_TYPE Struct Reference . . . . .	22
3.32.1 Detailed Description . . . . .	22
3.33 s_file Struct Reference . . . . .	22

3.33.1 Detailed Description . . . . .	22
3.34 s_format Struct Reference . . . . .	22
3.34.1 Detailed Description . . . . .	23
3.35 s_ne2000_dev Struct Reference . . . . .	23
3.35.1 Detailed Description . . . . .	23
3.36 s_pci_device Struct Reference . . . . .	23
3.36.1 Detailed Description . . . . .	23
3.37 SAMPLING_PORT_STATUS_TYPE Struct Reference . . . . .	24
3.37.1 Detailed Description . . . . .	24
3.38 SEMAPHORE_STATUS_TYPE Struct Reference . . . . .	24
3.38.1 Detailed Description . . . . .	24
3.39 u_arg Union Reference . . . . .	24
3.39.1 Detailed Description . . . . .	24
<b>4 File Documentation . . . . .</b>	<b>25</b>
4.1 /home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/semaphore.c File Reference . . . . .	25
4.1.1 Detailed Description . . . . .	25
4.2 /home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/drivers/rtl8029.c File Reference . . . . .	26
4.2.1 Detailed Description . . . . .	26
4.2.2 Function Documentation . . . . .	26
4.2.2.1 rtl8029_init . . . . .	26
4.2.2.2 rtl8029_polling . . . . .	27
4.2.2.3 rtl8029_read . . . . .	29
4.2.2.4 rtl8029_write . . . . .	29
4.3 /home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/protocols/cesar.h File Reference . . . . .	30
4.3.1 Detailed Description . . . . .	31
4.3.2 Function Documentation . . . . .	31
4.3.2.1 pok_protocols_cesar_marshall . . . . .	31
4.3.2.2 pok_protocols_cesar_unmarshall . . . . .	31
4.4 /home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/des/des.c File Reference . . . . .	32
4.4.1 Detailed Description . . . . .	32
4.4.2 Function Documentation . . . . .	32
4.4.2.1 pok_protocols_des_marshall . . . . .	32
4.4.2.2 pok_protocols_des_unmarshall . . . . .	33



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

__attribute__	11
ARINC_ATTRIBUTE	11
bf_key_st	12
BLACKBOARD_STATUS_TYPE	12
BUFFER_STATUS_TYPE	12
DES_ks	13
ERROR_STATUS_TYPE	13
EVENT_STATUS_TYPE	13
exception	14
ieee_double_shape_type	14
ieee_float_shape_type	14
PARTITION_STATUS_TYPE	15
pok_allocator_space_t	15
pok_arinc653_event_layer_t	15
pok_arinc653_semaphore_layer_t	16
pok_blackboard_status_t	16
pok_blackboard_t	16
pok_buffer_status_t	17
pok_buffer_t	17
pok_error_report_t	18
pok_error_status_t	18
pok_lockobj_attr_t	18
pok_lockobj_lockattr_t	19
pok_mutex_attr_t	19
pok_port_queueing_status_t	19
pok_port_sampling_status_t	20
pok_queue_t	20
pok_syscall_args_t	20
pok_thread_attr_t	21
PROCESS_ATTRIBUTE_TYPE	21
PROCESS_STATUS_TYPE	21
QUEUEING_PORT_STATUS_TYPE	22
s_file	22
s_format	22
s_ne2000_dev	23
s_pci_device	23
SAMPLING_PORT_STATUS_TYPE	24
SEMAPHORE_STATUS_TYPE	24

[u\\_arg](#) ..... 24



# Chapter 2

## File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ <b>arch.h</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-blackboards.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-buffers.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-events.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-health_monitoring.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-module_schedules.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-partitions.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-processes.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-queuing_ports.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-sampling_ports.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-semaphores.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex-timing.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/ada/arinc653/ <b>apex.ads</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arch/ppc/ <b>arch.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arch/sparc/ <b>arch.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arch/x86/ <b>arch.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arch/x86/ <b>ioports.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arch/x86/ <b>pci.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arch/x86/ <b>syscall.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>arincutils.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>blackboard.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>buffer.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>error.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>event.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>partition.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>process.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>queueing.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>sampling.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>semaphore.c</b> . . . . .	??
Provides ARINC653 API fonctionnalities for semaphore management . . . . .	25
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/arinc653/ <b>time.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/ <b>allocator.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/ <b>errno.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/ <b>errorconfirm.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/ <b>errorget.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/ <b>errorhandlercreate.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/ <b>errorhandlersetready.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/ <b>errorhandlerworker.c</b> . . . . .	??

/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/errorignore.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/errorlog.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/errorraise.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/eventbroadcast.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/eventcreate.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/eventlock.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/eventsignal.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/eventunlock.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/eventwait.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/main.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/mutexcreate.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/mutexlock.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/mutextrylock.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/mutexunlock.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/semcreate.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/semsignal.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/semstatus.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/semwait.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/threadattrinit.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/threadcreate.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/threaddelayedstart.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/threadid.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/threadperiod.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/threadpriority.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/threadresume.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/threadsleep.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/threadstatus.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/timecomputedeadline.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/core/timeget.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/drivers/rtl8029.c	
RTL8029 driver	26
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/drivers/rtl8029.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arch.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/assert.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/errno.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/libm.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/stdio.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/stdlib.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/string.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/types.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arch/x86/ioports.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arch/x86/pci.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arch/x86/types.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/arincutils.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/blackboard.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/buffer.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/error.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/event.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/partition.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/process.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/queueing.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/sampling.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/semaphore.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/time.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/types.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/allocator.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/dependencies.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/error.h	??

/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/event.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/lockobj.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/mutex.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/partition.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/semaphore.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/syscall.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/thread.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/core/time.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/libc/stdio.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/libc/stdlib.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/libc/string.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/middleware/blackboard.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/middleware/buffer.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/middleware/port.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/middleware/queue.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/protocols/blowfish.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/protocols/ceasar.h	??
Ceasar crypto protocol	30
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/protocols/des.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/protocols/protocols.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/protocols/ssl.h	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/misc/__udivdi3.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/stdio/printf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/stdlib/calloc.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/stdlib/free.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/stdlib/malloc.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/stdlib/rand.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/string/itoa.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/string/memcmp.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/string/memcpy.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/string/memset.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/string/strcmp.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/string/strcpy.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/string/streq.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/string/strlen.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/string/x86/memcpy.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/string/x86/strlen.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/acosh.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/acoshf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/asin.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/asinh.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/atan.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/atan2.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/atanf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/atanh.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/atanhf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/cbrt.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/cbrtf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/ceil.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/ceilf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/copysign.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/copysignf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/lib/cos.c	??

/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/cosf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/cosh.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/coshf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/drem.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/dremf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_acos.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_acosf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_acosh.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_acoshf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_asin.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_asinf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_atan2.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_atan2f.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_atanh.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_atanhf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_cosh.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_coshf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_exp.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_expf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_fmod.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_fmodf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_hypot.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_hypotf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_j0.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_j0f.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_j1.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_j1f.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_jn.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_jnf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_lgamma_r.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_lgammaf_r.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_log.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_log10.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_log10f.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_log2.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_log2f.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_logf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_pow.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_powf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_rem_pio2.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_rem_pio2f.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_remainder.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_remainderf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_scalb.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_scalbf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_sinh.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_sinhf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_sqrt.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/e_sqrtf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/erf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/erff.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/exp.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/expf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/expm1.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/expm1f.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/fabs.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/fabsf.c	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/finite.c	??

/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>finitef.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>floor.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>floorf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>fmod.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>fmodf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>fpclassify.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>frexp.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>frexpf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>gamma.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>gamma_r.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>gammaf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>gammaf_r.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>hypot.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>hypotf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>ilogb.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>ilogbf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>infinity.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>isinf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>isinf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>isnan.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>isnanf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>j0.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>j0f.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>j1.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>j1f.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>jn.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>jnf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>k_cos.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>k_cosf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>k_rem_pio2.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>k_rem_pio2f.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>k_sin.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>k_sinf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>k_standard.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>k_tan.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>k_tanf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>ldexp.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>ldexpf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>lgamma.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>lgamma_r.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>lgammaf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>lgammaf_r.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>log.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>log10.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>log10f.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>log1p.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>log1pf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>log2.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>log2f.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>logb.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>logbf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>logf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>math_private.h</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>matherr.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>modf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>modff.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>namespace.h</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>nextafter.c</b>	??

/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>nextafterf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>pow.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>powf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>remainder.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>remainderf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>rint.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>rintf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>round.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>roundf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>scalb.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>scalbf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>scalbn.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>scalbnf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>signgam.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>significand.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>significandf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>sin.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>sinf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>sinh.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>sinhf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>sqrt.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>sqrtf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>tan.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>tanf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>tanh.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>tanhf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>trunc.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libm/ <b>truncf.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>blackboardclear.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>blackboardcreate.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>blackboarddisplay.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>blackboardid.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>blackboardinit.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>blackboardread.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>blackboardstatus.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>buffercreate.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>bufferid.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>bufferinit.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>bufferreceive.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>bufferstatus.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>portqueueingcreate.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>portqueueingreceive.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>portqueueingsend.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>portsamplingcreate.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>portsamplingread.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>portsamplingwrite.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>portvirtualcreate.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>portvirtualdestination.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>portvirtualgetglobal.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>portvirtualnbdestinations.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>queueinit.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/middleware/ <b>resources.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/blowfish/ <b>bf_enc.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/blowfish/ <b>bf_locl.h</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/blowfish/ <b>bf_pi.h</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/blowfish/ <b>bf_skey.c</b>	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/blowfish/ <b>blowfish.c</b>	??

---

/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/blowfish/ <b>blowfish.h</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/ceasar/ <b>ceasar.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/des/ <b>des.c</b>	
DES crypto protocol . . . . .	32
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/des/ <b>des.h</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/des/ <b>des_enc.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/des/ <b>des_locl.h</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/des/ <b>ncbc_enc.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/des/ <b>set_key.c</b> . . . . .	??
/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/protocols/des/ <b>spr.h</b> . . . . .	??





## Chapter 3

# Class Documentation

### 3.1 `__attribute__` Struct Reference

#### Public Attributes

- char **dst** [ETH\_MAC\_LEN]
- char **src** [ETH\_MAC\_LEN]
- unsigned short **ethertype**
- unsigned short **src**
- unsigned short **dst**
- unsigned short **len**
- unsigned short **chk**
- eth\_hdr\_t **eth**
- udp\_hdr\_t **udp**
- char **data** [NET\_DATA\_MAXLEN]
- uint32\_t **len**
- uint32\_t **off**
- unsigned char **status**
- unsigned char **next**
- unsigned short **size**

#### 3.1.1 Detailed Description

Definition at line 34 of file rtl8029.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projects/parsec/pok\_orig/pok/trunk/libpok/drivers/rtl8029.h

### 3.2 `ARINC_ATTRIBUTE` Struct Reference

```
#include <arincutils.h>
```

#### Public Attributes

- PROCESS\_NAME\_TYPE **NAME**
- PRIORITY\_TYPE **BASE\_PRIORITY**
- STACK\_SIZE\_TYPE **STACK\_SIZE**

### 3.2.1 Detailed Description

Struct for save data NAME => Not use by pok BASE\_PRIORITY => This value, in pok, is modified. Here we save the base value

Definition at line 26 of file arincutils.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/arinc653/arincutils.h

## 3.3 bf\_key\_st Struct Reference

### Public Attributes

- BF\_LONG **P** [BF\_ROUNDS+2]
- BF\_LONG **S** [4 \*256]

### 3.3.1 Detailed Description

Definition at line 100 of file blowfish.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/protocols/blowfish/blowfish.h

## 3.4 BLACKBOARD\_STATUS\_TYPE Struct Reference

### Public Attributes

- EMPTY\_INDICATOR\_TYPE **EMPTY\_INDICATOR**
- MESSAGE\_SIZE\_TYPE **MAX\_MESSAGE\_SIZE**
- WAITING\_RANGE\_TYPE **WAITING\_PROCESSES**

### 3.4.1 Detailed Description

Definition at line 44 of file blackboard.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/arinc653/blackboard.h

## 3.5 BUFFER\_STATUS\_TYPE Struct Reference

### Public Attributes

- MESSAGE\_RANGE\_TYPE **NB\_MESSAGE**
- MESSAGE\_RANGE\_TYPE **MAX\_NB\_MESSAGE**
- MESSAGE\_SIZE\_TYPE **MAX\_MESSAGE\_SIZE**
- WAITING\_RANGE\_TYPE **WAITING\_PROCESSES**

### 3.5.1 Detailed Description

Definition at line 43 of file buffer.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/arinc653/buffer.h

## 3.6 DES\_ks Struct Reference

### Public Attributes

- union {
  - DES\_cblock **cblock**
  - DES\_LONG **deslong** [2]
- } **ks** [16]

### 3.6.1 Detailed Description

Definition at line 97 of file des.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/protocols/des/des.h

## 3.7 ERROR\_STATUS\_TYPE Struct Reference

### Public Attributes

- ERROR\_CODE\_TYPE **ERROR\_CODE**
- MESSAGE\_SIZE\_TYPE **LENGTH**
- PROCESS\_ID\_TYPE **FAILED\_PROCESS\_ID**
- SYSTEM\_ADDRESS\_TYPE **FAILED\_ADDRESS**
- ERROR\_MESSAGE\_TYPE **MESSAGE**

### 3.7.1 Detailed Description

Definition at line 51 of file error.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/arinc653/error.h

## 3.8 EVENT\_STATUS\_TYPE Struct Reference

### Public Attributes

- EVENT\_STATE\_TYPE **EVENT\_STATE**
- WAITING\_RANGE\_TYPE **WAITING\_PROCESSES**

### 3.8.1 Detailed Description

Definition at line 64 of file event.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/arinc653/event.h

## 3.9 exception Struct Reference

### Public Attributes

- int **type**
- char \* **name**
- double **arg1**
- double **arg2**
- double **retval**

### 3.9.1 Detailed Description

Definition at line 26 of file libm.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/libm.h

## 3.10 ieee\_double\_shape\_type Union Reference

### Public Attributes

- double **value**
- struct {
  - uint32\_t **lsw**
  - uint32\_t **msw**
- } **parts**

### 3.10.1 Detailed Description

Definition at line 87 of file math\_private.h.

The documentation for this union was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/libm/math\_private.h

## 3.11 ieee\_float\_shape\_type Union Reference

### Public Attributes

- float **value**
- uint32\_t **word**

### 3.11.1 Detailed Description

Definition at line 159 of file math\_private.h.

The documentation for this union was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/libm/math\_private.h

## 3.12 PARTITION\_STATUS\_TYPE Struct Reference

### Public Attributes

- SYSTEM\_TIME\_TYPE **PERIOD**
- SYSTEM\_TIME\_TYPE **DURATION**
- PARTITION\_ID\_TYPE **IDENTIFIER**
- LOCK\_LEVEL\_TYPE **LOCK\_LEVEL**
- OPERATING\_MODE\_TYPE **OPERATING\_MODE**
- START\_CONDITION\_TYPE **START\_CONDITION**

### 3.12.1 Detailed Description

Definition at line 43 of file partition.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/arinc653/partition.h

## 3.13 pok\_allocator\_space\_t Struct Reference

### Public Attributes

- size\_t **start**
- size\_t **size**
- bool\_t **allocated**

### 3.13.1 Detailed Description

Definition at line 60 of file allocator.c.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/core/allocator.c

## 3.14 pok\_arinc653\_event\_layer\_t Struct Reference

### Public Attributes

- pok\_event\_id\_t **core\_id**
- pok\_bool\_t **ready**

### 3.14.1 Detailed Description

Definition at line 43 of file event.c.

The documentation for this struct was generated from the following file:

- [/home/rosen/projets/parsec/pok\\_orig/pok/trunk/libpok/arinc653/event.c](#)

## 3.15 pok\_arinc653\_semaphore\_layer\_t Struct Reference

### Public Attributes

- pok\_bool\_t **ready**
- pok\_sem\_id\_t **core\_id**

### 3.15.1 Detailed Description

Definition at line 42 of file semaphore.c.

The documentation for this struct was generated from the following file:

- [/home/rosen/projets/parsec/pok\\_orig/pok/trunk/libpok/arinc653/semaphore.c](#)

## 3.16 pok\_blackboard\_status\_t Struct Reference

### Public Attributes

- pok\_port\_size\_t **msg\_size**
- pok\_bool\_t **empty**
- pok\_range\_t **waiting\_processes**

### 3.16.1 Detailed Description

Definition at line 37 of file blackboard.h.

The documentation for this struct was generated from the following file:

- [/home/rosen/projets/parsec/pok\\_orig/pok/trunk/libpok/include/middleware/blackboard.h](#)

## 3.17 pok\_blackboard\_t Struct Reference

### Public Attributes

- pok\_size\_t **size**
- pok\_bool\_t **empty**
- pok\_range\_t **waiting\_processes**
- pok\_size\_t **index**
- pok\_bool\_t **ready**
- pok\_event\_id\_t **lock**

### 3.17.1 Detailed Description

Definition at line 27 of file blackboard.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/middleware/blackboard.h

## 3.18 pok\_buffer\_status\_t Struct Reference

### Public Attributes

- pok\_range\_t **nb\_messages**
- pok\_range\_t **max\_messages**
- pok\_size\_t **message\_size**
- pok\_range\_t **waiting\_processes**

### 3.18.1 Detailed Description

Definition at line 47 of file buffer.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/middleware/buffer.h

## 3.19 pok\_buffer\_t Struct Reference

### Public Attributes

- pok\_bool\_t **ready**
- pok\_bool\_t **empty**
- pok\_bool\_t **full**
- pok\_size\_t **size**
- pok\_size\_t **index**
- pok\_port\_size\_t **off\_b**
- pok\_port\_size\_t **off\_e**
- pok\_port\_size\_t **msgsize**
- pok\_range\_t **waiting\_processes**
- pok\_queueing\_discipline\_t **discipline**
- pok\_event\_id\_t **lock**

### 3.19.1 Detailed Description

Definition at line 32 of file buffer.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/middleware/buffer.h

## 3.20 pok\_error\_report\_t Struct Reference

### Public Attributes

- uint32\_t **thread**
- uint32\_t **error**
- pok\_time\_t **when**

### 3.20.1 Detailed Description

Definition at line 37 of file error.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/core/error.h

## 3.21 pok\_error\_status\_t Struct Reference

### Public Attributes

- uint8\_t **error\_kind**
- uint32\_t **failed\_thread**
- uint32\_t **failed\_addr**
- char \* **msg**
- uint32\_t **msg\_size**

### 3.21.1 Detailed Description

Definition at line 27 of file error.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/core/error.h

## 3.22 pok\_lockobj\_attr\_t Struct Reference

### Public Attributes

- pok\_lockobj\_kind\_t **kind**
- pok\_locking\_policy\_t **locking\_policy**
- pok\_queueing\_discipline\_t **queueing\_policy**
- pok\_sem\_value\_t **initial\_value**
- pok\_sem\_value\_t **max\_value**

### 3.22.1 Detailed Description

Definition at line 38 of file lockobj.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/core/lockobj.h



## 3.23 pok\_lockobj\_lockattr\_t Struct Reference

### Public Attributes

- pok\_lockobj\_operation\_t **operation**
- pok\_lockobj\_kind\_t **obj\_kind**
- pok\_lockobj\_lock\_kind\_t **lock\_kind**
- uint64\_t **time**

### 3.23.1 Detailed Description

Definition at line 62 of file lockobj.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/core/lockobj.h

## 3.24 pok\_mutex\_attr\_t Struct Reference

### Public Attributes

- pok\_mutex\_policy\_t **policy**

### 3.24.1 Detailed Description

Definition at line 31 of file mutex.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/core/mutex.h

## 3.25 pok\_port\_queueing\_status\_t Struct Reference

### Public Attributes

- pok\_port\_size\_t **size**
- pok\_port\_direction\_t **direction**
- uint8\_t **nb\_messages**
- uint8\_t **waiting\_processes**

### 3.25.1 Detailed Description

Definition at line 60 of file port.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/middleware/port.h

## 3.26 pok\_port\_sampling\_status\_t Struct Reference

### Public Attributes

- pok\_port\_size\_t **size**
- pok\_port\_direction\_t **direction**
- uint64\_t **refresh**
- bool\_t **validity**

### 3.26.1 Detailed Description

Definition at line 107 of file port.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/middleware/port.h

## 3.27 pok\_queue\_t Struct Reference

### Public Attributes

- char \* **data**
- uint8\_t **size**
- uint8\_t **available\_size**

### 3.27.1 Detailed Description

Definition at line 25 of file queue.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/middleware/queue.h

## 3.28 pok\_syscall\_args\_t Struct Reference

### Public Attributes

- uint32\_t **nargs**
- uint32\_t **arg1**
- uint32\_t **arg2**
- uint32\_t **arg3**
- uint32\_t **arg4**
- uint32\_t **arg5**

### 3.28.1 Detailed Description

Definition at line 92 of file syscall.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/core/syscall.h

## 3.29 pok\_thread\_attr\_t Struct Reference

### Public Attributes

- uint8\_t **priority**
- void \* **entry**
- uint64\_t **period**
- uint64\_t **deadline**
- uint64\_t **time\_capacity**
- uint32\_t **stack\_size**
- uint32\_t **state**

### 3.29.1 Detailed Description

Definition at line 32 of file thread.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/core/thread.h

## 3.30 PROCESS\_ATTRIBUTE\_TYPE Struct Reference

### Public Attributes

- SYSTEM\_TIME\_TYPE **PERIOD**
- SYSTEM\_TIME\_TYPE **TIME\_CAPACITY**
- SYSTEM\_ADDRESS\_TYPE **ENTRY\_POINT**
- STACK\_SIZE\_TYPE **STACK\_SIZE**
- PRIORITY\_TYPE **BASE\_PRIORITY**
- DEADLINE\_TYPE **DEADLINE**
- PROCESS\_NAME\_TYPE **NAME**

### 3.30.1 Detailed Description

Definition at line 55 of file process.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/arinc653/process.h

## 3.31 PROCESS\_STATUS\_TYPE Struct Reference

### Public Attributes

- SYSTEM\_TIME\_TYPE **DEADLINE\_TIME**
- PRIORITY\_TYPE **CURRENT\_PRIORITY**
- PROCESS\_STATE\_TYPE **PROCESS\_STATE**
- [PROCESS\\_ATTRIBUTE\\_TYPE](#) **ATTRIBUTES**

### 3.31.1 Detailed Description

Definition at line 65 of file process.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/arinc653/process.h

## 3.32 QUEUING\_PORT\_STATUS\_TYPE Struct Reference

### Public Attributes

- MESSAGE\_RANGE\_TYPE **NB\_MESSAGE**
- MESSAGE\_RANGE\_TYPE **MAX\_NB\_MESSAGE**
- MESSAGE\_SIZE\_TYPE **MAX\_MESSAGE\_SIZE**
- PORT\_DIRECTION\_TYPE **PORT\_DIRECTION**
- WAITING\_RANGE\_TYPE **WAITING\_PROCESSES**

### 3.32.1 Detailed Description

Definition at line 32 of file queueing.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/arinc653/queueing.h

## 3.33 s\_file Struct Reference

### Public Attributes

- char **buff** [MY\_BUF\_SIZE]
- size\_t **pos**

### 3.33.1 Detailed Description

Definition at line 38 of file printf.c.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/libc/stdio/printf.c

## 3.34 s\_format Struct Reference

### Public Attributes

- char **ch**
- t\_fmtfun **fun**
- int **flags**

### 3.34.1 Detailed Description

Definition at line 55 of file printf.c.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/libc/stdio/printf.c

## 3.35 s\_ne2000\_dev Struct Reference

### Public Attributes

- [s\\_pci\\_device](#) **pci**
- unsigned int **addr**
- char **mac** [6]
- [pok\\_queue\\_t](#) **recv\_buf** [20]

### 3.35.1 Detailed Description

Definition at line 66 of file rtl8029.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/drivers/rtl8029.h

## 3.36 s\_pci\_device Struct Reference

### Public Attributes

- uint16\_t **bus**
- uint16\_t **dev**
- uint16\_t **fun**
- uint16\_t **vendorid**
- uint16\_t **deviceid**
- uint16\_t **irq\_line**
- uint16\_t **io\_range**
- uint32\_t **bar** [6]
- uint32\_t **addr**
- void \* **irq\_handler**

### 3.36.1 Detailed Description

Definition at line 28 of file pci.h.

The documentation for this struct was generated from the following file:

- /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/arch/x86/pci.h

### 3.37 SAMPLING\_PORT\_STATUS\_TYPE Struct Reference

#### Public Attributes

- SYSTEM\_TIME\_TYPE **REFRESH\_PERIOD**
- MESSAGE\_SIZE\_TYPE **MAX\_MESSAGE\_SIZE**
- PORT\_DIRECTION\_TYPE **PORT\_DIRECTION**
- VALIDITY\_TYPE **LAST\_MSG\_VALIDITY**

#### 3.37.1 Detailed Description

Definition at line 39 of file `sampling.h`.

The documentation for this struct was generated from the following file:

- `/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/sampling.h`

### 3.38 SEMAPHORE\_STATUS\_TYPE Struct Reference

#### Public Attributes

- SEMAPHORE\_VALUE\_TYPE **CURRENT\_VALUE**
- SEMAPHORE\_VALUE\_TYPE **MAXIMUM\_VALUE**
- WAITING\_RANGE\_TYPE **WAITING\_PROCESSES**

#### 3.38.1 Detailed Description

Definition at line 64 of file `semaphore.h`.

The documentation for this struct was generated from the following file:

- `/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/include/arinc653/semaphore.h`

### 3.39 u\_arg Union Reference

#### Public Attributes

- `uint32_t` **value**
- `uint32_t` **uint**
- `int` **sint**
- `double` **vdouble**
- `void *` **ptr**

#### 3.39.1 Detailed Description

Definition at line 44 of file `printf.c`.

The documentation for this union was generated from the following file:

- `/home/rosen/projets/parsec/pok_orig/pok/trunk/libpok/libc/stdio/printf.c`

# Chapter 4

## File Documentation

### 4.1 /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/arinc653/semaphore.c File Reference

Provides ARINC653 API fonctionnalités for semaphore management.

#### Classes

- struct [pok\\_arinc653\\_semaphore\\_layer\\_t](#)

#### Functions

- void **CREATE\_SEMAPHORE** (SEMAPHORE\_NAME\_TYPE SEMAPHORE\_NAME, SEMAPHORE\_VALUE\_TYPE CURRENT\_VALUE, SEMAPHORE\_VALUE\_TYPE MAXIMUM\_VALUE, QUEUING\_DISCIPLINE\_TYPE QUEUING\_DISCIPLINE, SEMAPHORE\_ID\_TYPE \*SEMAPHORE\_ID, RETURN\_CODE\_TYPE \*RETURN\_CODE)
- void **WAIT\_SEMAPHORE** (SEMAPHORE\_ID\_TYPE SEMAPHORE\_ID, SYSTEM\_TIME\_TYPE TIME\_OUT, RETURN\_CODE\_TYPE \*RETURN\_CODE)
- void **SIGNAL\_SEMAPHORE** (SEMAPHORE\_ID\_TYPE SEMAPHORE\_ID, RETURN\_CODE\_TYPE \*RETURN\_CODE)
- void **GET\_SEMAPHORE\_ID** (SEMAPHORE\_NAME\_TYPE SEMAPHORE\_NAME, SEMAPHORE\_ID\_TYPE \*SEMAPHORE\_ID, RETURN\_CODE\_TYPE \*RETURN\_CODE)
- void **GET\_SEMAPHORE\_STATUS** (SEMAPHORE\_ID\_TYPE SEMAPHORE\_ID, [SEMAPHORE\\_STATUS\\_TYPE](#) \*SEMAPHORE\_STATUS, RETURN\_CODE\_TYPE \*RETURN\_CODE)

#### Variables

- pok\_bool\_t **pok\_arinc653\_semaphores\_initialized** = 0
- char \* **pok\_arinc653\_semaphores\_names** [POK\_CONFIG\_ARINC653\_NB\_SEMAPHORES]
- [pok\\_arinc653\\_semaphore\\_layer\\_t](#) **pok\_arinc653\_semaphores\_layers** [POK\_CONFIG\_ARINC653\_NB\_SEMAPHORES]

#### 4.1.1 Detailed Description

Provides ARINC653 API fonctionnalités for semaphore management.

Definition in file [semaphore.c](#).

## 4.2 /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/drivers/rtl8029.c File Reference

RTL8029 driver.

### Functions

- void [rtl8029\\_read](#) (pok\_port\_id\_t port\_id, void \*data, uint32\_t len)  
*Reads data from the corresponding network stack.*
- void [rtl8029\\_write](#) (pok\_port\_id\_t port\_id, const void \*data, uint32\_t len)  
*Send data to the interface.*
- void [rtl8029\\_polling](#) ()  
*Polls rtl8029 device.*
- void [rtl8029\\_init](#) ()  
*Initializes rtl8029 device.*

### 4.2.1 Detailed Description

RTL8029 driver.

#### Author

Laurent

#### Date

PFE GISTR 2010

Definition in file [rtl8029.c](#).

### 4.2.2 Function Documentation

#### 4.2.2.1 void rtl8029\_init ( )

Initializes rtl8029 device.

Seeks and registers PCI interface, set configuration and fills the dev structure.

Definition at line 382 of file rtl8029.c.

```
{
dev.pci.vendorid = 0x10ec;
dev.pci.deviceid = 0x8029;
dev.pci.io_range = 0x10;

if (pci_register(&(dev.pci)) != 0)
{
printf("rtl8029: PCI init failed!\n");
return;
}

dev.addr = dev.pci.bar[0] & (~0x1F);

unsigned char i = 0;
unsigned char buf[6 * 2]; // used for MAC address

NE2000_SELECT_PAGE(&dev, 0);

/* This bit is the STOP command. When it is set, no packets will be
received or transmitted. POWER UP=1. */
outb(NE2000_CR_STP, dev.addr + NE2000_CR);
```



```

// Sets several options... Read the datasheet!
outb(0x00, dev.addr + NE2000_TCR);
outb(NE2000_RCR_AB, dev.addr + NE2000_RCR);
outb(NE2000_DCR_LS | NE2000_DCR_FT1, dev.addr + NE2000_DCR);

/* The Page Start register sets the start page address
   of the receive buffer ring. */
outb(NE2000_RXBUF, dev.addr + NE2000_PSTART);
/* The Page Stop register sets the stop page address
   of the receive buffer ring. */
outb(NE2000_MEMSZ, dev.addr + NE2000_PSTOP);
/* This register is used to prevent overwrite of the receive buffer ring.
   It is typically used as a pointer indicating the last receive buffer
   page the host has read. */
outb(NE2000_RXBUF, dev.addr + NE2000_BNRY);

/* These two registers set the data byte counts of remote DMA. */
outb(0, dev.addr + NE2000_RBCR0);
outb(0, dev.addr + NE2000_RBCR1);

NE2000_SELECT_PAGE(&dev, 1);

/* This register points to the page address of the first receive buffer
   page to be used for a packet reception. */
outb(NE2000_RXBUF + 1, dev.addr + NE2000_CURR);

// Init mac address
/* Here's something I do not understand... Section 6.2.2 of the datasheet
   says bytes 00H-05H of the PROM corresponds to the Ethernet ID. But it
   looks like each byte of the MAC address is written twice...
   Therefore I read 2 * sizeof(mac) and select one of the two bytes
   corresponding to the MAC... Weird... Really... */
ne2000_read(&dev, buf, 6 * 2, 0);
for (i = 0; i < 6; i++)
    dev.mac[i] = buf[i * 2];

/* These registers contain my Ethernet node address and are used to compare
   the destination address of incoming packets for acceptance or rejection.*/
outb(dev.mac[0], dev.addr + NE2000_PAR0);
outb(dev.mac[1], dev.addr + NE2000_PAR1);
outb(dev.mac[2], dev.addr + NE2000_PAR2);
outb(dev.mac[3], dev.addr + NE2000_PAR3);
outb(dev.mac[4], dev.addr + NE2000_PAR4);
outb(dev.mac[5], dev.addr + NE2000_PAR5);

NE2000_SELECT_PAGE(&dev, 0);

// Start command
outb(NE2000_CR_STA, dev.addr + NE2000_CR);

// Reactivating interrupts
/* ISR register must be cleared after power up. */
outb(0xFF, dev.addr + NE2000_ISR);
/* All bits correspond to the bits in the ISR register. POWER UP=all 0s.
   Setting individual bits will enable the corresponding interrupts. */
/* Since POK use polling, ALL interrupts are disabled */
outb(0x00, dev.addr + NE2000_IMR);

for (i = 0; i < 20; i++) /* TODO: random constant */
{
    dev.recv_buf[i].len = 0;
    dev.recv_buf[i].off = 0;
}

return;
}

```

#### 4.2.2.2 void rtl8029\_polling ( )

Polls rtl8029 device.

Watches for events, typically for receiving queued packets.

Definition at line 279 of file rtl8029.c.

```

{
    unsigned char state; // ISR state

    NE2000_SELECT_PAGE(&dev, 0);

    while (1)
    {

```

```

// do we have an interrupt flag set?
if ((state = pok_inb(dev.addr + NE2000_ISR)) == 0)
    continue;

if (state & NE2000_ISR_PRX)
{
    if ((pok_inb(dev.addr + NE2000_RSR) & NE2000_RSR_PRX) == 0)
    {
        // error
    }

    printf("[*]\n");

    /* no errors */
    s_ne2000_header    ne2000_hdr;    // ne2000 packet header
    unsigned short    offset;        // dma offset
    unsigned char     start, end;    // pointers for the ring buffer
    pok_packet_t      recv_packet;

    while (1)
    {
        /* This register is used to prevent overwrite of the receive buffer
        ring.
        It is typically used as a pointer indicating the last receive buffer
        page the host has read.*/
        start = pok_inb(dev.addr + NE2000_BNRY) + 1;

        /* This register points to the page address of the first receive
        buffer page to be used for a packet reception. */
        NE2000_SELECT_PAGE(&dev, 1);
        end = pok_inb(dev.addr + NE2000_CURR);
        NE2000_SELECT_PAGE(&dev, 0);

        if ((end % NE2000_MEMSZ) == (start % NE2000_MEMSZ) + 1)
        {
            break;
        }

        /* et on decapsule! */
        offset = start << 8;
        // ne2000 header
        offset += ne2000_read(&dev, &ne2000_hdr, sizeof(s_ne2000_header),
                             offset);

        ne2000_read(&dev, &recv_packet,
                    ne2000_hdr.size - sizeof(s_ne2000_header), offset);
        rtl8029_enqueue(&recv_packet);

        // update the BNRY register... almost forgot that
        outb(ne2000_hdr.next > NE2000_MEMSZ ?
             NE2000_RXBUF - 1 : ne2000_hdr.next - 1, dev.addr + NE2000_BNRY);
    }

    outb(NE2000_ISR_PRX, dev.addr + NE2000_ISR); // Clear PRX flag
}

if (state & NE2000_ISR_PTX)
{
    outb(NE2000_ISR_PTX, dev.addr + NE2000_ISR); // Clear PTX flag
}

if (state & NE2000_ISR_RXE)
{
    outb(NE2000_ISR_RXE, dev.addr + NE2000_ISR); // Clear RXE flag
}

if (state & NE2000_ISR_TXE)
{
    outb(NE2000_ISR_TXE, dev.addr + NE2000_ISR); // Clear TXE flag
}

if (state & NE2000_ISR_OVW)
{
    outb(NE2000_ISR_OVW, dev.addr + NE2000_ISR); // Clear OVW flag
}

if (state & NE2000_ISR_CNT)
{
    outb(NE2000_ISR_CNT, dev.addr + NE2000_ISR); // Clear CNT flag
}

if (state & NE2000_ISR_RST)
{
    outb(NE2000_ISR_RST, dev.addr + NE2000_ISR); // Clear RST bit
}

```

```

}
}

```

#### 4.2.2.3 void rtl8029\_read ( pok\_port\_id\_t port\_id, void \* data, uint32\_t len )

Reads data from the corresponding network stack.

Reads enqueued data in the stack partition.

Definition at line 146 of file rtl8029.c.

```

{
    pok_port_id_t global;
    pok_ret_t      ret;

    ret = pok_port_virtual_get_global (port_id, &global);

    if (ret == POK_ERRNO_OK)
    {
        char          *dest = data;
        pok_queue_t* queue = dev.recv_buf + global;
        uint32_t      size = len < queue->len ? len : queue->len;
        uint32_t      copied = 0;

        printf ("[RTL8029] READ DATA FROM LOCAL PORT %d "
                "GLOBAL_PORT=%d), size=%d\n", port_id, global, len);

        /* is there something to read ? */
        if (queue->len == 0)
        {
            printf("rtl8029_read: error: empty read ring buffer %d!\n", port_id);
            return;
        }

        /* copy from the queue to the buffer */
        for (copied = 0; copied < size; copied++)
        {
            dest[copied % RECV_BUF_SZ] = queue->data[queue->off];
            queue->off = (queue->off + 1) % RECV_BUF_SZ;
        }

        /* updating data length in this queue */
        queue->len -= size;
    }
}

```

#### 4.2.2.4 void rtl8029\_write ( pok\_port\_id\_t port\_id, const void \* data, uint32\_t len )

Send data to the interface.

Writes data to be sent to network.

Definition at line 187 of file rtl8029.c.

```

{
    uint32_t      nbdest;
    uint32_t      tmp;
    uint32_t      dest;
    pok_ret_t      ret;
    char          node2[6] = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };
    pok_packet_t  packet;
    const char*   d;
    size_t        cpylen = 0;
    size_t        sndlen = 0;
    unsigned char state; // ISR state

    ret = pok_port_virtual_nb_destinations (port_id, &nbdest);
    if (ret != POK_ERRNO_OK)
    {
        return;
    }

    for (tmp = 0 ; tmp < nbdest ; tmp++)
    {
        ret = pok_port_virtual_destination (port_id, tmp, &dest);
        if (ret == POK_ERRNO_OK)

```

```

{
    printf (" [RTL8029] SEND DATA THROUGH NETWORK FROM LOCAL PORT %d "
           "TO GLOBAL PORT %d, size=%d\n", port_id, dest, len);

    memcpy(packet.eth.src, dev.mac, ETH_MAC_LEN);
    memcpy(packet.eth.dst, node2, ETH_MAC_LEN);
    packet.eth.ethertype = 0x4242;
    packet.udp.src = port_id;
    packet.udp.dst = dest;

    for (d = data; len != 0; len -= cpylen, data += cpylen)
    {
        // too short; let's cut
        if (len <= NET_DATA_MINLEN)
        {
            cpylen = len;
            sndlen = ETH_DATA_MINLEN + sizeof(eth_hdr_t);
        }
        else
        {
            // too big; let's pad
            if (len >= NET_DATA_MAXLEN)
            {
                cpylen = NET_DATA_MAXLEN;
                sndlen = ETH_DATA_MAXLEN + sizeof(eth_hdr_t);
            }
            // normal
            else
            {
                cpylen = len;
                sndlen = sizeof(eth_hdr_t) + sizeof(udp_hdr_t) + cpylen;
            }
        }

        packet.udp.len = cpylen;
        memcpy(&(packet.data), data, cpylen);

        ne2000_write(&dev, &packet, sndlen, NE2000_TXBUF * 256);

        do
        {
            state = pok_inb(dev.addr + NE2000_ISR);
        }
        while ((state & NE2000_ISR_RDC) != NE2000_ISR_RDC);

        /* This register sets the start page address of
           the packet to be transmitted. */
        outb(NE2000_TXBUF, dev.addr + NE2000_TPSR); ???

        /* These two registers set the byte counts of
           the packet to be transmitted. */
        outb(sndlen, dev.addr + NE2000_TBCR0);
        outb(sndlen >> 8, dev.addr + NE2000_TBCR1);

        /* This bit must be set to transmit a packet. */
        outb(pok_inb(dev.addr + NE2000_CR) | NE2000_CR_TXP,
            dev.addr + NE2000_CR);

        outb(NE2000_ISR_RDC, dev.addr + NE2000_ISR); // Clear RDC bit
    }
}
}
}

```

### 4.3 /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/protocols/cesar.h File Reference

Cesar crypto protocol.

#### Functions

- void [pok\\_protocols\\_cesar\\_unmarshall](#) (void \*crypted\_data, pok\_size\_t crypted\_size, void \*uncrypted\_data, size\_t \*uncrypted\_size)
- void [pok\\_protocols\\_cesar\\_marshall](#) (void \*uncrypted\_data, pok\_size\_t uncryptd\_size, void \*crypted\_data, size\_t \*crypted\_size)

### 4.3.1 Detailed Description

Cesar crypto protocol.

#### Author

Julien Delange

#### Date

2009 This is a very basic crypto protocol that just change the order of bytes in data. There is no public/private key, the algorithm is known by the attacker so that it's a very weak crypto protocol. Interested people can gather more information about this protocol on: [http://en.wikipedia.org/wiki/Caesar\\_cipher](http://en.wikipedia.org/wiki/Caesar_cipher)

We don't provide an associated marshalling type for the Cesar protocol since the crypted size is the same than the uncrypted size.

Definition in file [cesar.h](#).

### 4.3.2 Function Documentation

#### 4.3.2.1 void pok\_protocols\_cesar\_marshall ( void \* *uncrypted\_data*, pok\_size\_t *uncrypted\_size*, void \* *crypted\_data*, size\_t \* *crypted\_size* )

Function that encrypts data

\file libpok/protocols/cesar.c \brief Function to crypt/uncrypt data using the Cesar cipher. \author Julien Delange  
 \brief Marshall data, the crypted size has the same size than uncrypted data.

Definition at line 34 of file cesar.c.

```
{
    uint8_t* uncrypted;
    uint8_t* crypted;
    size_t tmp;

    uncrypted = (uint8_t*) uncrypted_data;
    crypted = (uint8_t*) crypted_data;

    for (tmp = 0 ; tmp < uncrypted_size ; tmp++)
    {
        crypted[tmp] = (uncrypted[tmp] + 4) % 255;
    }

    *crypted_size = uncrypted_size;
}
```

#### 4.3.2.2 void pok\_protocols\_cesar\_unmarshall ( void \* *crypted\_data*, pok\_size\_t *crypted\_size*, void \* *uncrypted\_data*, size\_t \* *uncrypted\_size* )

Function that uncrypts data

\brief Unmarshall data, the crypted size has the same size than uncrypted data.

Definition at line 56 of file cesar.c.

```
{
    uint8_t* uncrypted;
    uint8_t* crypted;
    size_t tmp;

    uncrypted = (uint8_t*) uncrypted_data;
    crypted = (uint8_t*) crypted_data;

    for (tmp = 0 ; tmp < crypted_size ; tmp++)
    {
```

```

        unencrypted[tmp] = (crypted[tmp] - 4) % 255;
    }
    *unencrypted_size = crypted_size;
}

```

## 4.4 /home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/protocols/des/des.c File Reference

DES crypto protocol.

### Functions

- void **pok\_protocols\_des\_init** ()
- void **pok\_protocols\_des\_marshall** (void \*unencrypted\_data, pok\_size\_t unencrypted\_size, void \*crypted\_data, size\_t \*crypted\_size)
- void **pok\_protocols\_des\_unmarshall** (void \*crypted\_data, pok\_size\_t crypted\_size, void \*unencrypted\_data, size\_t \*unencrypted\_size)

### Variables

- unsigned char **initVector** [8] = POK\_PROTOCOLS\_DES\_INIT
- int **pok\_protocols\_des\_is\_init** = 0

#### 4.4.1 Detailed Description

DES crypto protocol.

##### Author

Julien Delange

##### Date

2009 This file is a wrapper that interfaces with OpenSSL functions. It sets the crypto key, initialisation string and calls OpenSSL function to crypts data.

Definition in file [des.c](#).

#### 4.4.2 Function Documentation

##### 4.4.2.1 void pok\_protocols\_des\_marshall ( void \* *unencrypted\_data*, pok\_size\_t *unencrypted\_size*, void \* *crypted\_data*, size\_t \* *crypted\_size* )

Function that crypts data.

Definition at line 54 of file des.c.

```

{
    DES_cblock ivec;
    DES_key_schedule schedule;

    DES_set_key_checked (&cbc_key, &schedule);

    memcpy(ivec, initVector, sizeof(initVector));

    DES_ncbc_encrypt(unencrypted_data, crypted_data, unencrypted_size, &schedule, &
        ivec, DES_ENCRYPT);
    *crypted_size = 8;
}

```

4.4.2.2 void pok\_protocols\_des\_unmarshall ( void \* *crypted\_data*, pok\_size\_t *crypted\_size*, void \* *uncrypted\_data*, size\_t \* *uncrypted\_size* )

Function that uncrypts data.

Definition at line 68 of file des.c.

```
{
    DES_cblock ivec;
    DES_key_schedule schedule;

    DES_set_key_checked (&cbc_key, &schedule);

    memcpy(ivec, initVector, sizeof(initVector));
    DES_ncbc_encrypt(crypted_data, uncrpyted_data, crypted_size, &schedule, &
        ivec, DES_DECRYPT);

    *uncrpyted_size = 8;
}
```

# Index

/home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/arinc653/semaphore.c, 25

/home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/drivers/rtl8029.c, 26

/home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/include/pok\_protocols/ceasar.h, 30

/home/rosen/projets/parsec/pok\_orig/pok/trunk/libpok/protocols/des/des.c, 32

\_\_attribute\_\_, 11

ARINC\_ATTRIBUTE, 11

BUFFER\_STATUS\_TYPE, 12

bf\_key\_st, 12

ceasar.h

- pok\_protocols\_ceasar\_marshall, 31
- pok\_protocols\_ceasar\_unmarshall, 31

DES\_ks, 13

des.c

- pok\_protocols\_des\_marshall, 32
- pok\_protocols\_des\_unmarshall, 32

ERROR\_STATUS\_TYPE, 13

EVENT\_STATUS\_TYPE, 13

exception, 14

ieee\_double\_shape\_type, 14

ieee\_float\_shape\_type, 14

PROCESS\_STATUS\_TYPE, 21

pok\_allocator\_space\_t, 15

pok\_arinc653\_event\_layer\_t, 15

pok\_arinc653\_semaphore\_layer\_t, 16

pok\_blackboard\_status\_t, 16

pok\_blackboard\_t, 16

pok\_buffer\_status\_t, 17

pok\_buffer\_t, 17

pok\_error\_report\_t, 18

pok\_error\_status\_t, 18

pok\_lockobj\_attr\_t, 18

pok\_lockobj\_lockattr\_t, 19

pok\_mutex\_attr\_t, 19

pok\_port\_queueing\_status\_t, 19

pok\_port\_sampling\_status\_t, 20

pok\_protocols\_ceasar\_marshall

- ceasar.h, 31

pok\_protocols\_ceasar\_unmarshall

- ceasar.h, 31

pok\_protocols\_des\_marshall

- ceasar.h, 31
- pok\_protocols\_des\_unmarshall, 32

pok\_queue\_t, 20

pok\_thread\_attr\_t, 21

rtl8029.c

- rtl8029\_init, 26
- rtl8029\_polling, 27
- rtl8029\_read, 29
- rtl8029\_write, 29

rtl8029\_init

- rtl8029.c, 26

rtl8029\_polling

- rtl8029.c, 27

rtl8029\_read

- rtl8029.c, 29

rtl8029\_write

- rtl8029.c, 29

s\_file, 22

s\_format, 22

s\_ne2000\_dev, 23

s\_pci\_device, 23

u\_arg, 24