

POK

Generated by Doxygen 1.8.3.1

Sat May 25 2013 18:39:20

Contents

- 1 Class Index** **1**
- 1.1 Class List 1

- 2 File Index** **3**
- 2.1 File List 3

- 3 Class Documentation** **11**
- 3.1 [__attribute__ Struct Reference](#) 11
- 3.1.1 Detailed Description 11
- 3.2 [ARINC_ATTRIBUTE Struct Reference](#) 11
- 3.2.1 Detailed Description 12
- 3.3 [bf_key_st Struct Reference](#) 12
- 3.3.1 Detailed Description 12
- 3.4 [BLACKBOARD_STATUS_TYPE Struct Reference](#) 12
- 3.4.1 Detailed Description 12
- 3.5 [BUFFER_STATUS_TYPE Struct Reference](#) 12
- 3.5.1 Detailed Description 13
- 3.6 [DES_ks Struct Reference](#) 13
- 3.6.1 Detailed Description 13
- 3.7 [ERROR_STATUS_TYPE Struct Reference](#) 13
- 3.7.1 Detailed Description 13
- 3.8 [EVENT_STATUS_TYPE Struct Reference](#) 13
- 3.8.1 Detailed Description 14
- 3.9 [exception Struct Reference](#) 14
- 3.9.1 Detailed Description 14
- 3.10 [ieee_double_shape_type Union Reference](#) 14
- 3.10.1 Detailed Description 14
- 3.11 [ieee_float_shape_type Union Reference](#) 14
- 3.11.1 Detailed Description 15
- 3.12 [PARTITION_STATUS_TYPE Struct Reference](#) 15
- 3.12.1 Detailed Description 15
- 3.13 [pok_allocator_space_t Struct Reference](#) 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
4 File Documentation	25
4.1 /home/hipse/git-repos/pok/trunk/libpok/arinc653/semaphore.c File Reference	25
4.1.1 Detailed Description	25
4.2 /home/hipse/git-repos/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/hipse/git-repos/pok/trunk/libpok/include/protocols/cesar.h File Reference	30
4.3.1 Detailed Description	30
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/hipse/git-repos/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
Index	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/hiphse/git-repos/pok/trunk/libpok/ arch.h	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-blackboards.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-buffers.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-events.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-health_monitoring.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-module_schedules.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-partitions.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-processes.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-queuing_ports.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-sampling_ports.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-semaphores.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex-timing.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/ada/arinc653/ apex.ads	??
/home/hiphse/git-repos/pok/trunk/libpok/arch/ppc/ arch.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arch/sparc/ arch.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arch/x86/ arch.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arch/x86/ ioports.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arch/x86/ pci.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arch/x86/ syscall.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ arincutils.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ blackboard.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ buffer.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ error.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ event.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ partition.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ process.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ queueing.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ sampling.c	??
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ semaphore.c Provides ARINC653 API functionalities for semaphore management	25
/home/hiphse/git-repos/pok/trunk/libpok/arinc653/ time.c	??
/home/hiphse/git-repos/pok/trunk/libpok/core/ allocator.c	??
/home/hiphse/git-repos/pok/trunk/libpok/core/ errno.c	??
/home/hiphse/git-repos/pok/trunk/libpok/core/ errorconfirm.c	??
/home/hiphse/git-repos/pok/trunk/libpok/core/ errorget.c	??
/home/hiphse/git-repos/pok/trunk/libpok/core/ errorhandlercreate.c	??
/home/hiphse/git-repos/pok/trunk/libpok/core/ errorhandlersetready.c	??
/home/hiphse/git-repos/pok/trunk/libpok/core/ errorhandlerworker.c	??

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

/home/hipse/git-repos/pok/trunk/libpok/include/core/event.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/core/lockobj.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/core/mutex.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/core/partition.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/core/semaphore.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/core/syscall.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/core/thread.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/core/time.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/libc/stdio.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/libc/stdlib.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/libc/string.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/middleware/blackboard.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/middleware/buffer.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/middleware/port.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/middleware/queue.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/protocols/blowfish.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/protocols/ceasar.h	??
Ceasar crypto protocol	30
/home/hipse/git-repos/pok/trunk/libpok/include/protocols/des.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/protocols/protocols.h	??
/home/hipse/git-repos/pok/trunk/libpok/include/protocols/ssl.h	??
/home/hipse/git-repos/pok/trunk/libpok/libc/misc/_udivdi3.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/stdio/printf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/stdlib/calloc.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/stdlib/free.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/stdlib/malloc.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/stdlib/rand.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/string/itoa.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/string/memcmp.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/string/memcpy.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/string/memset.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/string/strcmp.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/string/strcpy.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/string/strreq.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/string/strlen.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/string/x86/memcpy.c	??
/home/hipse/git-repos/pok/trunk/libpok/libc/string/x86/strlen.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/acos.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/acosf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/acosh.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/acoshf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/asin.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/asinf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/asinh.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/asinhf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/atan.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/atan2.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/atan2f.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/atanf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/atanh.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/atanhf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/cbrt.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/cbrtf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ceil.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ceilf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/copysign.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/copysignf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/cos.c	??

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

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

/home/hipse/git-repos/pok/trunk/libpok/libm/ nextafterf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ pow.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ powf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ remainder.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ remainderf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ rint.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ rintf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ round.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ roundf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ scalb.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ scalbf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ scalbn.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ scalbnf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ signgam.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ significantd.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ significantdf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ sin.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ sinf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ sinh.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ sinhf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ sqrt.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ sqrtf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ tan.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ tanf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ tanh.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ tanhf.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ trunc.c	??
/home/hipse/git-repos/pok/trunk/libpok/libm/ truncf.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ blackboardclear.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ blackboardcreate.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ blackboarddisplay.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ blackboardid.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ blackboardinit.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ blackboardread.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ blackboardstatus.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ buffercreate.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ bufferid.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ bufferinit.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ bufferreceive.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ bufferstatus.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ portqueueingcreate.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ portqueueingreceive.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ portqueueingsend.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ portsamplingcreate.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ portsamplingread.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ portsamplingwrite.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ portvirtualcreate.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ portvirtualdestination.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ portvirtualgetglobal.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ portvirtualnbdestinations.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ queueinit.c	??
/home/hipse/git-repos/pok/trunk/libpok/middleware/ resources.c	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/blowfish/ bf_enc.c	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/blowfish/ bf_locl.h	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/blowfish/ bf_pi.h	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/blowfish/ bf_key.c	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/blowfish/ blowfish.c	??

/home/hipse/git-repos/pok/trunk/libpok/protocols/blowfish/ blowfish.h	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/ceasar/ ceasar.c	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/des/ des.c	
DES crypto protocol	32
/home/hipse/git-repos/pok/trunk/libpok/protocols/des/ des.h	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/des/ des_enc.c	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/des/ des_locl.h	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/des/ ncbc_enc.c	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/des/ set_key.c	??
/home/hipse/git-repos/pok/trunk/libpok/protocols/des/ spr.h	??

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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hiphse/git-repos/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/hipse/git-repos/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/hipse/git-repos/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/hipse/git-repos/pok/trunk/libpok/libc/stdio/printf.c`

Chapter 4

File Documentation

4.1 /home/hipse/git-repos/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/hiphse/git-repos/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](#).

```

383 {
384     dev.pci.vendorid = 0x10ec;
385     dev.pci.deviceid = 0x8029;
386     dev.pci.io_range = 0x10;
387
388     if (pci_register(&(dev.pci)) != 0)
389     {
390         printf("rtl8029: PCI init failed!\n");
391         return;
392     }
393
394     dev.addr = dev.pci.bar[0] & (~0x1F);
395
396     unsigned char i = 0;
397     unsigned char buf[6 * 2]; // used for MAC address
398
399     NE2000_SELECT_PAGE(&dev, 0);
400
401     /* This bit is the STOP command. When it is set, no packets will be
402        received or transmitted. POWER UP=1. */
403     outb(NE2000_CR_STP, dev.addr + NE2000_CR);
404
405     // Sets several options... Read the datasheet!
406     outb(0x00, dev.addr + NE2000_TCR);

```



```

407 outb(NE2000_RCR_AB, dev.addr + NE2000_RCR);
408 outb(NE2000_DCR_LS | NE2000_DCR_FT1, dev.addr + NE2000_DCR);
409
410 /* The Page Start register sets the start page address
411    of the receive buffer ring. */
412 outb(NE2000_RXBUF, dev.addr + NE2000_PSTART);
413 /* The Page Stop register sets the stop page address
414    of the receive buffer ring. */
415 outb(NE2000_MEMSZ, dev.addr + NE2000_PSTOP);
416 /* This register is used to prevent overwrite of the receive buffer ring.
417    It is typically used as a pointer indicating the last receive buffer
418    page the host has read. */
419 outb(NE2000_RXBUF, dev.addr + NE2000_BNRY);
420
421 /* These two registers set the data byte counts of remote DMA. */
422 outb(0, dev.addr + NE2000_RBCR0);
423 outb(0, dev.addr + NE2000_RBCR1);
424
425 NE2000_SELECT_PAGE(&dev, 1);
426
427 /* This register points to the page address of the first receive buffer
428    page to be used for a packet reception. */
429 outb(NE2000_RXBUF + 1, dev.addr + NE2000_CURR);
430
431 // Init mac address
432 /* Here's something I do not understand... Section 6.2.2 of the datasheet
433    says bytes 00H-05H of the PROM corresponds to the Ethernet ID. But it
434    looks like each byte of the MAC address is written twice...
435    Therefore I read 2 * sizeof(mac) and select one of the two bytes
436    corresponding to the MAC... Weird... Really... */
437 ne2000_read(&dev, buf, 6 * 2, 0);
438 for (i = 0; i < 6; i++)
439     dev.mac[i] = buf[i * 2];
440
441 /* These registers contain my Ethernet node address and are used to compare
442    the destination address of incoming packets for acceptance or rejection.*/
443 outb(dev.mac[0], dev.addr + NE2000_PAR0);
444 outb(dev.mac[1], dev.addr + NE2000_PAR1);
445 outb(dev.mac[2], dev.addr + NE2000_PAR2);
446 outb(dev.mac[3], dev.addr + NE2000_PAR3);
447 outb(dev.mac[4], dev.addr + NE2000_PAR4);
448 outb(dev.mac[5], dev.addr + NE2000_PAR5);
449
450 NE2000_SELECT_PAGE(&dev, 0);
451
452 // Start command
453 outb(NE2000_CR_STA, dev.addr + NE2000_CR);
454
455 // Reactivating interrupts
456 /* ISR register must be cleared after power up. */
457 outb(0xFF, dev.addr + NE2000_ISR);
458 /* All bits correspond to the bits in the ISR register. POWER UP=all 0s.
459    Setting individual bits will enable the corresponding interrupts. */
460 /* Since POK use polling, ALL interrupts are disabled */
461 outb(0x00, dev.addr + NE2000_IMR);
462
463 for (i = 0; i < 20; i++) /* TODO: random constant */
464 {
465     dev.recv_buf[i].len = 0;
466     dev.recv_buf[i].off = 0;
467 }
468
469 return;
470 }

```

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.

```

280 {
281     unsigned char state; // ISR state
282
283     NE2000_SELECT_PAGE(&dev, 0);
284
285     while (1)
286     {
287         // do we have an interrupt flag set?
288         if ((state = pok_inb(dev.addr + NE2000_ISR)) == 0)

```

```

289     continue;
290
291     if (state & NE2000_ISR_PRX)
292     {
293         if ((pok_inb(dev.addr + NE2000_RSR) & NE2000_RSR_PRX) == 0)
294         {
295             // error
296         }
297
298         printf("[*]\n");
299
300         /* no errors */
301         s_ne2000_header    ne2000_hdr;    // ne2000 packet header
302         unsigned short    offset;        // dma offset
303         unsigned char     start, end;    // pointers for the ring buffer
304         pok_packet_t      recv_packet;
305
306         while (1)
307         {
308
309             /* This register is used to prevent overwrite of the receive buffer ring.
310              It is typically used as a pointer indicating the last receive buffer
311              page the host has read.*/
312             start = pok_inb(dev.addr + NE2000_BNRY) + 1;
313
314             /* This register points to the page address of the first receive
315              buffer page to be used for a packet reception. */
316             NE2000_SELECT_PAGE(&dev, 1);
317             end = pok_inb(dev.addr + NE2000_CURR);
318             NE2000_SELECT_PAGE(&dev, 0);
319
320             if ((end % NE2000_MEMSZ) == (start % NE2000_MEMSZ) + 1)
321             {
322                 break;
323             }
324
325             /* et on decapsule! */
326             offset = start << 8;
327             // ne2000 header
328             offset += ne2000_read(&dev, &ne2000_hdr, sizeof(s_ne2000_header),
329                                 offset);
330
331             ne2000_read(&dev, &recv_packet,
332                        ne2000_hdr.size - sizeof(s_ne2000_header), offset);
333             rtl8029_enqueue(&recv_packet);
334
335             // update the BNRY register... almost forgot that
336             outb(ne2000_hdr.next > NE2000_MEMSZ ?
337                NE2000_RXBUF - 1 : ne2000_hdr.next - 1, dev.addr + NE2000_BNRY);
338         }
339
340         outb(NE2000_ISR_PRX, dev.addr + NE2000_ISR); // Clear PRX flag
341     }
342
343     if (state & NE2000_ISR_PTX)
344     {
345         outb(NE2000_ISR_PTX, dev.addr + NE2000_ISR); // Clear PTX flag
346     }
347
348     if (state & NE2000_ISR_RXE)
349     {
350         outb(NE2000_ISR_RXE, dev.addr + NE2000_ISR); // Clear RXE flag
351     }
352
353     if (state & NE2000_ISR_TXE)
354     {
355         outb(NE2000_ISR_TXE, dev.addr + NE2000_ISR); // Clear TXE flag
356     }
357
358     if (state & NE2000_ISR_OVW)
359     {
360         outb(NE2000_ISR_OVW, dev.addr + NE2000_ISR); // Clear OVW flag
361     }
362
363     if (state & NE2000_ISR_CNT)
364     {
365         outb(NE2000_ISR_CNT, dev.addr + NE2000_ISR); // Clear CNT flag
366     }
367
368     if (state & NE2000_ISR_RST)
369     {
370         outb(NE2000_ISR_RST, dev.addr + NE2000_ISR); // Clear RST bit
371     }
372 }
373 }
374 }

```

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.

```

147 {
148     pok_port_id_t global;
149     pok_ret_t     ret;
150
151     ret = pok_port_virtual_get_global (port_id, &global);
152
153     if (ret == POK_ERRNO_OK)
154     {
155         char          *dest = data;
156         pok_queue_t* queue = dev.recv_buf + global;
157         uint32_t      size = len < queue->len ? len : queue->len;
158         uint32_t      copied = 0;
159
160         printf ("[RTL8029] READ DATA FROM LOCAL PORT %d "
161              "GLOBAL_PORT=%d), size=%d\n", port_id, global, len);
162
163         /* is there something to read ? */
164         if (queue->len == 0)
165         {
166             printf("rtl8029_read: error: empty read ring buffer %d!\n", port_id);
167             return;
168         }
169
170         /* copy from the queue to the buffer */
171         for (copied = 0; copied < size; copied++)
172         {
173             dest[copied % RECV_BUF_SZ] = queue->data[queue->off];
174             queue->off = (queue->off + 1) % RECV_BUF_SZ;
175         }
176
177         /* updating data length in this queue */
178         queue->len -= size;
179     }
180 }

```

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.

```

188 {
189     uint32_t      nbdest;
190     uint32_t      tmp;
191     uint32_t      dest;
192     pok_ret_t     ret;
193     char          node2[6] = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };
194     pok_packet_t  packet;
195     const char*   d;
196     size_t        cpylen = 0;
197     size_t        sndlen = 0;
198     unsigned char state; // ISR state
199
200     ret = pok_port_virtual_nb_destinations (port_id, &nbdest);
201     if (ret != POK_ERRNO_OK)
202     {
203         return;
204     }
205
206     for (tmp = 0 ; tmp < nbdest ; tmp++)
207     {
208         ret = pok_port_virtual_destination (port_id, tmp, &dest);
209         if (ret == POK_ERRNO_OK)
210         {
211             printf ("[RTL8029] SEND DATA THROUGH NETWORK FROM LOCAL PORT %d "
212                  "TO GLOBAL PORT %d, size=%d\n", port_id, dest, len);
213
214             memcpy(packet.eth.src, dev.mac, ETH_MAC_LEN);
215             memcpy(packet.eth.dst, node2, ETH_MAC_LEN);

```

```

216     packet.eth.ethertype = 0x4242;
217     packet.udp.src = port_id;
218     packet.udp.dst = dest;
219
220     for (d = data; len != 0; len -= cpylen, data += cpylen)
221     {
222         // too short; let's cut
223         if (len <= NET_DATA_MINLEN)
224         {
225             cpylen = len;
226             sndlen = ETH_DATA_MINLEN + sizeof(eth_hdr_t);
227         }
228         else
229         {
230             // too big; let's pad
231             if (len >= NET_DATA_MAXLEN)
232             {
233                 cpylen = NET_DATA_MAXLEN;
234                 sndlen = ETH_DATA_MAXLEN + sizeof(eth_hdr_t);
235             }
236             // normal
237             else
238             {
239                 cpylen = len;
240                 sndlen = sizeof(eth_hdr_t) + sizeof(udp_hdr_t) + cpylen;
241             }
242         }
243
244         packet.udp.len = cpylen;
245         memcpy(&(packet.data), data, cpylen);
246
247         ne2000_write(&dev, &packet, sndlen, NE2000_TXBUF * 256);
248
249         do
250         {
251             state = pok_inb(dev.addr + NE2000_ISR);
252         }
253         while ((state & NE2000_ISR_RDC) != NE2000_ISR_RDC);
254
255         /* This register sets the start page address of
256            the packet to be transmitted. */
257         outb(NE2000_TXBUF, dev.addr + NE2000_TPSR); /*?
258
259         /* These two registers set the byte counts of
260            the packet to be transmitted. */
261         outb(sndlen, dev.addr + NE2000_TBCR0);
262         outb(sndlen >> 8, dev.addr + NE2000_TBCR1);
263
264         /* This bit must be set to transmit a packet. */
265         outb(pok_inb(dev.addr + NE2000_CR) | NE2000_CR_TXP,
266             dev.addr + NE2000_CR);
267
268         outb(NE2000_ISR_RDC, dev.addr + NE2000_ISR); // Clear RDC bit
269     }
270 }
271 }
272 }

```

4.3 /home/hipse/git-repos/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 uncrypted_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

We don't provide an associated marshalling type for the Ceasar 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 Ceasar 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.

```
35 {
36     uint8_t* uncrypted;
37     uint8_t* crypted;
38     size_t tmp;
39
40     uncrypted = (uint8_t*) uncrypted_data;
41     crypted = (uint8_t*) crypted_data;
42
43     for (tmp = 0 ; tmp < uncrypted_size ; tmp++)
44     {
45         crypted[tmp] = (uncrypted[tmp] + 4) % 255;
46     }
47
48     *crypted_size = uncrypted_size;
49 }
```

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.

```
57 {
58     uint8_t* uncrypted;
59     uint8_t* crypted;
60     size_t tmp;
61
62     uncrypted = (uint8_t*) uncrypted_data;
63     crypted = (uint8_t*) crypted_data;
64
65     for (tmp = 0 ; tmp < crypted_size ; tmp++)
66     {
67         uncrypted[tmp] = (crypted[tmp] - 4) % 255;
68     }
69
70     *uncrypted_size = crypted_size;
71 }
```

4.4 /home/hipse/git-repos/pok/trunk/libpok/protocols/des/des.c File Reference

DES crypto protocol.

Functions

- void **pok_protocols_des_init** ()
- void **pok_protocols_des_marshall** (void *uncrypted_data, pok_size_t uncrpyted_size, void *crypted_data, size_t *crypted_size)
- void **pok_protocols_des_unmarshall** (void *crypted_data, pok_size_t crypted_size, void *uncrypted_data, size_t *uncrypted_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 * *uncrypted_data*, pok_size_t *uncrypted_size*, void * *crypted_data*, size_t * *crypted_size*)

Function that crypts data.

Definition at line 54 of file des.c.

```
55 {
56     DES_cblock ivec;
57     DES_key_schedule schedule;
58
59     DES_set_key_checked (&cbc_key, &schedule);
60
61     memcpy (ivec, initVector, sizeof (initVector));
62
63     DES_ncbc_encrypt (uncrypted_data, crypted_data, uncrpyted_size, &schedule, &ivec, DES_ENCRYPT);
64     *crypted_size = 8;
65 }
```

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.

```
69 {
70     DES_cblock ivec;
71     DES_key_schedule schedule;
72
73     DES_set_key_checked (&cbc_key, &schedule);
74
75     memcpy(ivec, initVector, sizeof(initVector));
76     DES_ncbc_encrypt(crypted_data, uncrpyted_data, crypted_size, &schedule, &ivec, DES_DECRYPT);
77
78     *uncrypted_size = 8;
79 }
```

Index

/home/hipse/git-repos/pok/trunk/libpok/arinc653/semaphore.- des.c, [32](#)
c, [25](#) pok_protocols_des_unmarshall

/home/hipse/git-repos/pok/trunk/libpok/drivers/rtl8029.- des.c, [32](#)
c, [26](#) pok_queue_t, [20](#)

/home/hipse/git-repos/pok/trunk/libpok/include/protocols/cesar.h, [30](#) pok_syscall_args_t, [20](#)
h, [30](#) pok_thread_attr_t, [21](#)

/home/hipse/git-repos/pok/trunk/libpok/protocols/des/des.-
c, [32](#) rtl8029.c
__attribute__, [11](#) rtl8029_init, [26](#)
rtl8029_polling, [27](#)
rtl8029_read, [28](#)
rtl8029_write, [29](#)

ARINC_ATTRIBUTE, [11](#) rtl8029_init
rtl8029.c, [26](#)

BUFFER_STATUS_TYPE, [12](#) rtl8029_polling
rtl8029.c, [27](#)

bf_key_st, [12](#) rtl8029_read
rtl8029.c, [28](#)

cesar.h rtl8029_write
rtl8029.c, [29](#)

 pok_protocols_cesar_marshall, [31](#)

 pok_protocols_cesar_unmarshall, [31](#)

DES_ks, [13](#)

des.c s_file, [22](#)
s_format, [22](#)
s_ne2000_dev, [23](#)
s_pci_device, [23](#)

 pok_protocols_des_marshall, [32](#)

 pok_protocols_des_unmarshall, [32](#)

ERROR_STATUS_TYPE, [13](#)

EVENT_STATUS_TYPE, [13](#)

exception, [14](#) u_arg, [24](#)

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_cesar_marshall
cesar.h, [31](#)

pok_protocols_cesar_unmarshall
cesar.h, [31](#)

pok_protocols_des_marshall