NAME

own, **own_send_command**, **own_command_wait**, **own_self_command**, **own_acquire_bus**, **own crc**, **own_release_bus**, **OWN_ACQUIRE_BUS**, **OWN_CRC**, **OWN_RELEASE_BUS**, **OWN_SEND_COMMAND -** Dallas Semiconductor 1-Wire Network and Transport Interface

SYNOPSIS

#include <sys/bus.h>
#include <dev/ow/own.h>

int
own_send_command(*device_t pdev*, *struct ow_cmd *cmd*);

int
own_command_wait(*device_t pdev*, *struct ow_cmd *cmd*);

int
own_self_command(*device_t pdev*, *struct ow_cmd *cmd*, *uint8_t xpt_cmd*);

int
own_acquire_bus(*device_t pdev*, *int how*);

int
own_release_bus(*device_t pdev*);

int
own_crc(*device_t pdev*, *uint8_t *buffer*, *size_t len*);

int
OWN_SEND_COMMAND(*device_t ndev*, *device_t pdev*, *struct ow_cmd *cmd*);

int
OWN_ACQUIRE_BUS(*device_t ndev*, *device_t pdev*, *int how*);

void
OWN_RELEASE_BUS(*device_t ndev*, *device_t pdev*);

uint8_t
OWN_CRC(*device_t ndev*, *device_t pdev*, *uint8_t *buffer*, *size_t len*);

DESCRIPTION

The **own** interface defines three sets of functions for interacting with 1-Wire devices: sending commands, reserving the bus, and ensuring data integrity. Wrappers are provided for the raw **OWN** kobj(9) interfaces and should be used for improved safety over the kobj(9) ones.

Bus Commands

The 1-Wire bus defines different layers of access to the devices on the bus. The **own** functions provide access to the network and transport layers. The network layer designates the next command as being either for all devices on the bus, or for a specific device. The network layer also specifies the speed used by the link layer.

struct ow_cmd encapsulates network access, speed, and timing information. It specifies the commands to send and whether or not to read data. Its members are:

flags Flags controlling the interpretation of the structure. These flags are defined in <*dev/ow/ow.h*>:

OW_FLAG_OVERDRIVE
Send *xpt_cmd* bytes and read *xpt_read* bytes at overdrive speed.

OW_FLAG_READ_BIT
Interpret *xpt_read_len* to be in bits to be read after *xpt_cmd* rather than bytes.

rom_cmd
ROM command bytes to send.

rom_len
Number of ROM command bytes to send.

rom_read_len
Number of bytes to read after sending the ROM command.

rom_read
Buffer for bytes read after the ROM command.

xpt_cmd
Transport command to send.

xpt_len
Length of the transport command bytes to send. Specify 0 for no transport command.

xpt_read_len

Number of bytes to read after *xpt_cmd* bytes are sent. If the OW_FLAG_READ_BIT bit is set in *flags*, then it is the number of bits to read. Bits read are packed into bytes.

xpt_read

Buffer for data read.

own_command_wait() acquires the 1-Wire bus, waiting if necessary, sends the command, and then releases the bus. **own_send_command**() sends the command without bus reservation. *pdev* is the client device (the presentation layer device) sending the command. The *cmd* argument describes the transaction to send to the 1-Wire bus.

own_self_command() fills in *cmd* with a MATCH_ROM ROM command, the ROM address of *pdev* and the *xpt_cmd* as a convenient way to create directed commands.

Bus Reservation

The 1-Wire system includes an advisory lock for the bus that presentation layer devices can use to coordinate access. Locking is purely advisory at this time.

own_acquire_bus() reserves the bus. It waits indefinitely if the *how* argument is OWN_WAIT and returns the error EWOULDBLOCK if passed OWN_DONTWAIT when the bus is owned by another client.

own_release_bus() releases the bus.

Data Integrity

own_crc() computes the 1-Wire standard CRC function over the data passed in *buffer* and *len* and returns the result.

Notes

The 1-Wire standard (Maxim AN937) defines layers that are akin to ISO networking layers. The lowest relevant layer, the link layer, defines the polling windows and the timing of the signaling of different modes. The network layer is built on top of the link layer and is used to address devices in a unicast or multicast manner. The transport layer defines commands and responses from the devices. The presentation layer is composed of the device specific commands and actions used to control the specific 1-Wire devices on bus.

These interfaces are implemented by the ow(4) device. Presentation layer devices (children of the newbus ow(4) device) should only call the functions described here. The functionality provided by the owc(4) device (specifically the owll(9) interface) should only be called by the ow(4) driver.

SEE ALSO

ow(4), owc(4), owll(9) *https://pdfserv.maximintegrated.com/en/an/AN937.pdf*

LEGAL

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

HISTORY

The **own** driver first appeared in FreeBSD 11.0.

AUTHORS

The **own** device driver and this manual page were written by Warner Losh.