

HAL IO library

Version
12/9/2009 10:49:00 AM

Table of Contents

Main Page	1
File Index	1
File Documentation	1
haldig.c	1
haldig.h	4
halpwr.c	8
halpwr.h	11
halusb.c	14
halusb.h	16
io-errors.h	17
Index	18

Main Page

Library libhalio.so contains following usefull user-space interfaces:

- [haldig.h](#)
 - [halpwr.h](#)
 - [halusb.h](#)
-

File Index

File List

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

haldig.c (Digital IO interface file of the IO HAL)	1
haldig.h (Cross HAL definitions for DIG I/O routines)	4
halpwr.c (PWR IO interface file of the IO HAL)	8
halpwr.h (Cross HAL definitions for Power I/O routines)	11
halusb.c (USB IO interface file of the IO HAL)	14
halusb.h (Cross HAL definitions for USB I/O routines)	16
io-errors.h (Cross HAL error definitions)	17
typedefs.h	Error! Bookmark not defined.

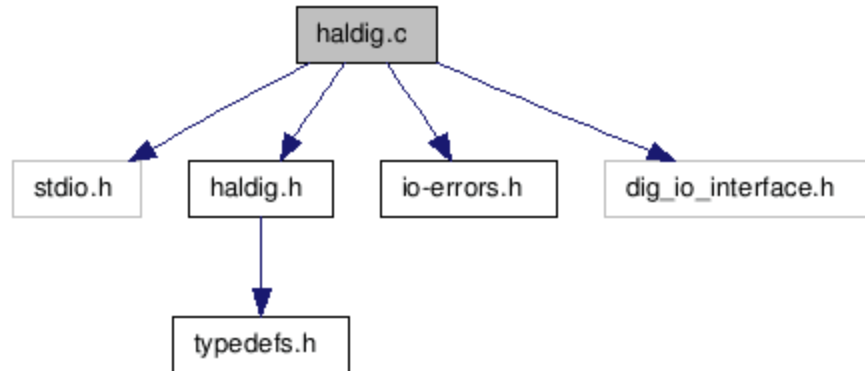
File Documentation

haldig.c File Reference

Digital IO interface file of the IO HAL.

```
#include <stdio.h>
#include "haldig.h"
#include "io-errors.h"
#include "dig_io_interface.h"
```

Include dependency graph for haldig.c:



Functions

- int [HAL_IO_DIGinit](#) (void)
a function for opening DIG i/o-interface.
- int [HAL_IO_DIGuninit](#) (void)
a function for closing DIG i/o-interface.
- int [HAL_IO_DIGgetHardwareVersion](#) (void)
a function for getting hardware version
- int [HAL_IO_DIGget](#) (uint8 channel)
a function for reading digital input value
- int [HAL_IO_DIGgetAll](#) ()
a function for reading digital input values
- int [HAL_IO_DIGset](#) (uint8 channel, uint8 value)
a function for setting digital output value
- int [HAL_IO_DIGledEnable](#) ([INFO_LEDS](#) led, [LED_COLORS](#) color)
a function for setting LED
- int [HAL_IO_DIGledDisable](#) ([INFO_LEDS](#) led)
a function for disabling LED
- int [HAL_IO_DIGledBlink](#) ([INFO_LEDS](#) led, [LED_COLORS](#) color, uint8 hz)
a function for blinking LED

Detailed Description

Digital IO interface file of the IO HAL.

Author:

Teemu Keskinarkaus / CC Systems Oy
Definition in file [haldig.c](#).

Function Documentation

int HAL_IO_DIGget (uint8 *channel*)

a function for reading digital input value

Parameters:

channel which line to read

Returns:

The result if enable was succeeded
Definition at line 45 of file haldig.c.

int HAL_IO_DIGgetAll ()

a function for reading digital input values

Returns:

The result bitmask if enable was succeeded
Definition at line 56 of file haldig.c.

int HAL_IO_DIGgetHardwareVersion (void)

a function for getting hardware version

Returns:

Hardware version or error code
Definition at line 34 of file haldig.c.

int HAL_IO_DIGinit (void)

a function for opening DIG i/o-interface.

This function must be called before using any other DIG interface functions.

Returns:

The result if initializing was succeeded
Definition at line 12 of file haldig.c.

int HAL_IO_DIGledBlink ([INFO_LEDS](#) *led*, [LED_COLORS](#) *color*, uint8 *hz*)

a function for blinking LED

Parameters:

led Led that is controlled

color Led color to blink

hz Blink frequency in Hertz

Returns:

The result if enable was succeeded
Definition at line 100 of file haldig.c.

int HAL_IO_DIGledDisable ([INFO_LEDS](#) *led*)

a function for disabling LED

Parameters:

led Led that is controlled

Returns:

The result if disable was succeeded

Definition at line 89 of file haldig.c.

int HAL_IO_DIGledEnable ([INFO_LEDS](#) *led*, [LED_COLORS](#) *color*)

a function for setting LED

Parameters:

led Led that is controlled

color Led color to enable

Returns:

The result if enable was succeeded

Definition at line 78 of file haldig.c.

int HAL_IO_DIGset (uint8 *channel*, uint8 *value*)

a function for setting digital output value

Parameters:

channel which line to read

value into which set

Returns:

The result if enable was succeeded

Definition at line 67 of file haldig.c.

int HAL_IO_DIGuninit (void)

a function for closing DIG i/o-interface.

This function must be called when DIG i/o-interface is no longer needed.

Returns:

The result if closing was succeeded

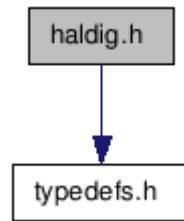
Definition at line 23 of file haldig.c.

haldig.h File Reference

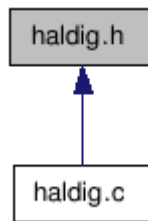
Cross HAL definitions for DIG I/O routines.

```
#include "typedefs.h"
```

Include dependency graph for haldig.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum [INFO_LEDS](#) { [STATUS_LED](#) = 0, [CAN0_LED](#), [CAN1_LED](#) }
Enums for info LEDs.
- enum [LED_COLORS](#) { [LED_RED](#) = 0, [LED_GREEN](#), [LED_AMBER](#) }
Enums for LED colors.

Functions

- int [HAL_IO_DIGinit](#) (void)
a function for opening DIG i/o-interface.
- int [HAL_IO_DIGuninit](#) (void)
a function for closing DIG i/o-interface.
- int [HAL_IO_DIGgetHardwareVersion](#) (void)
a function for getting hardware version
- int [HAL_IO_DIGget](#) (uint8 channel)
a function for reading digital input value
- int [HAL_IO_DIGgetAll](#) ()
a function for reading digital input values
- int [HAL_IO_DIGset](#) (uint8 channel, uint8 value)
a function for setting digital output value
- int [HAL_IO_DIGledEnable](#) ([INFO_LEDS](#) led, [LED_COLORS](#) color)
a function for setting LED
- int [HAL_IO_DIGledDisable](#) ([INFO_LEDS](#) led)
a function for disabling LED
- int [HAL_IO_DIGledBlink](#) ([INFO_LEDS](#) led, [LED_COLORS](#) color, uint8 hz)
a function for blinking LED

Detailed Description

Cross HAL definitions for DIG I/O routines.

Author:

Teemu Keskinarkaus / CC Systems Oy
Definition in file [haldig.h](#).

Enumeration Type Documentation

enum [INFO_LEDS](#)

Enums for info LEDS.

Enumerator:

STATUS_LED Status LED aka. frontled
CAN0_LED LED for CAN0
CAN1_LED LED for CAN1

Definition at line 16 of file haldig.h.

enum [LED_COLORS](#)

Enums for LED colors.

Enumerator:

LED_RED RED LED on
LED_GREEN Green LED on
LED_AMBER Both LEDs on

Definition at line 27 of file haldig.h.

Function Documentation

int HAL_IO_DIGget (uint8 *channel*)

a function for reading digital input value

Parameters:

channel which line to read

Returns:

The result if enable was succeeded
Definition at line 45 of file haldig.c.

int HAL_IO_DIGgetAll ()

a function for reading digital input values

Returns:

The result bitmask if enable was succeeded
Definition at line 56 of file haldig.c.

int HAL_IO_DIGgetHardwareVersion (void)

a function for getting hardware version

Returns:

Hardware version or error code
Definition at line 34 of file haldig.c.

int HAL_IO_DIGinit (void)

a function for opening DIG i/o-interface.

This function must be called before using any other DIG interface functions.

Returns:

The result if initializing was succeeded
Definition at line 12 of file haldig.c.

int HAL_IO_DIGledBlink ([INFO LEDS](#) led, [LED COLORS](#) color, uint8 hz)

a function for blinking LED

Parameters:

led Led that is controlled
color Led color to blink
hz Blink frequency in Hertz

Returns:

The result if enable was succeeded
Definition at line 100 of file haldig.c.

int HAL_IO_DIGledDisable ([INFO LEDS](#) led)

a function for disabling LED

Parameters:

led Led that is controlled

Returns:

The result if disable was succeeded
Definition at line 89 of file haldig.c.

int HAL_IO_DIGledEnable ([INFO LEDS](#) led, [LED COLORS](#) color)

a function for setting LED

Parameters:

led Led that is controlled
color Led color to enable

Returns:

The result if enable was succeeded
Definition at line 78 of file haldig.c.

int HAL_IO_DIGset (uint8 *channel*, uint8 *value*)

a function for setting digital output value

Parameters:

channel which line to read
value into which set

Returns:

The result if enable was succeeded
Definition at line 67 of file haldig.c.

int HAL_IO_DIGuninit (void)

a function for closing DIG i/o-interface.

This function must be called when DIG i/o-interface is no longer needed.

Returns:

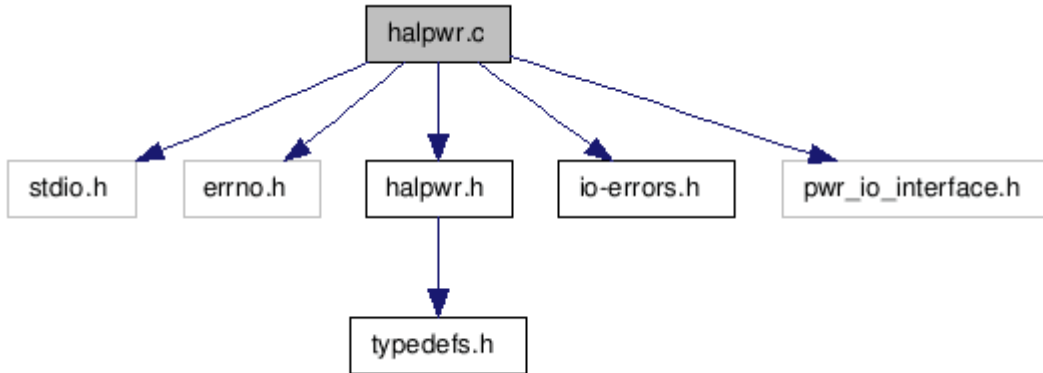
The result if closing was succeeded
Definition at line 23 of file haldig.c.

halpwr.c File Reference

PWR IO interface file of the IO HAL.

```
#include <stdio.h>
#include <errno.h>
#include "halpwr.h"
#include "io-errors.h"
#include "pwr_io_interface.h"
```

Include dependency graph for halpwr.c:



Functions

- int [HAL_IO_PWRinit](#) (void)
a function for opening PWR i/o-interface.
- int [HAL_IO_PWRuninit](#) (void)
a function for closing PWR i/o-interface.
- int [HAL_IO_PWRgetPowerDown](#) (void)
a function for getting power down signal state
- int [HAL_IO_PWRgetOvervoltage](#) (void)
a function for getting overvoltage signal state
- int [HAL_IO_PWRregisterPowerEventHandler](#) (void(*function)(int))
a function for registering event handler for power events.
- int [HAL_IO_PWRsetPowerState](#) ([PWR_STATE_PINS](#) pin, uint8 state)
a function for setting power state for peripheral
- int [HAL_IO_PWRgetPowerState](#) ([PWR_STATE_PINS](#) pin)
a function for getting power state of the peripheral

Detailed Description

PWR IO interface file of the IO HAL.

Author:

Teemu Keskinarkaus / CC Systems Oy

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

Function Documentation

int HAL_IO_PWRgetOvervoltage (void)

a function for getting overvoltage signal state

Returns:

The result of the signal or error code
Definition at line 50 of file halpwr.c.

int HAL_IO_PWRgetPowerDown (void)

a function for getting power down signal state

Returns:

The result of the signal or error code
Definition at line 39 of file halpwr.c.

int HAL_IO_PWRgetPowerState ([PWR_STATE_PINS](#) *pin*)

a function for getting power state of the peripheral

Parameters:

pin Power pin to get

Returns:

The result of the get or error code
Definition at line 91 of file halpwr.c.

int HAL_IO_PWRinit (void)

a function for opening PWR i/o-interface.

This function must be called before using any other PWR interface functions.

Returns:

The result if initializing was succeeded
Definition at line 13 of file halpwr.c.

int HAL_IO_PWRregisterPowerEventHandler (void*)(int) *function*)

a function for registering event handler for power events.

Parameters:

function call-back function for powerdown/overvoltage event

Returns:

The result if registering was succeeded
Definition at line 61 of file halpwr.c.

int HAL_IO_PWRsetPowerState ([PWR_STATE_PINS](#) *pin*, uint8 *state*)

a function for setting power state for peripheral

Parameters:

pin Power pin to set
state State for the pin

Returns:

The result of the set or error code
 Definition at line 72 of file halpwr.c.

int HAL_IO_PWRinit (void)

a function for closing PWR i/o-interface.

This function must be called when PWR i/o-interface is no longer needed.

Returns:

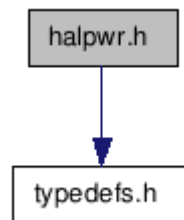
The result if closing was succeeded
 Definition at line 28 of file halpwr.c.

halpwr.h File Reference

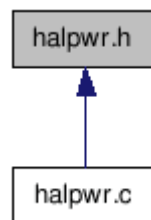
Cross HAL definitions for Power I/O routines.

```
#include "typedefs.h"
```

Include dependency graph for halpwr.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum [PWR_STATE_PINS](#) { [PWR_PIN_ETH](#) = 0, [PWR_PIN_WLAN](#), [PWR_PIN_ADDON](#), [PWR_PIN_CAN](#), [PWR_PIN_USB](#), [PWR_PIN_GPRS](#), [NUMBER_OF_PWR_PINS](#) }
Power state pins.

Functions

- int [HAL_IO_PWRinit](#) (void)
a function for opening PWR i/o-interface.

- int [HAL_IO_PWRinit](#) (void)
a function for closing PWR i/o-interface.
 - int [HAL_IO_PWRgetPowerDown](#) (void)
a function for getting power down signal state
 - int [HAL_IO_PWRgetOvervoltage](#) (void)
a function for getting overvoltage signal state
 - int [HAL_IO_PWRsetPowerState](#) ([PWR_STATE_PINS](#) pin, uint8 state)
a function for setting power state for peripheral
 - int [HAL_IO_PWRgetPowerState](#) ([PWR_STATE_PINS](#) pin)
a function for getting power state of the peripheral
 - int [HAL_IO_PWRregisterPowerEventHandler](#) (void(*function)(int))
a function for registering event handler for power events.
-

Detailed Description

Cross HAL definitions for Power I/O routines.

Author:

Teemu Keskinarkaus / CC Systems Oy

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

Enumeration Type Documentation

enum [PWR_STATE_PINS](#)

Power state pins.

Enumerator:

PWR_PIN_ETH Ethernet.

PWR_PIN_WLAN WLAN.

PWR_PIN_ADDON Addon card.

PWR_PIN_CAN CAN.

PWR_PIN_USB USB.

PWR_PIN_GPRS GSM/GPRS.

NUMBER_OF_PWR_PINS last counter

Definition at line 13 of file [halpwr.h](#).

Function Documentation

int HAL_IO_PWRgetOvervoltage (void)

a function for getting overvoltage signal state

Returns:

The result of the signal or error code
Definition at line 50 of file halpwr.c.

int HAL_IO_PWRgetPowerDown (void)

a function for getting power down signal state

Returns:

The result of the signal or error code
Definition at line 39 of file halpwr.c.

int HAL_IO_PWRgetPowerState ([PWR_STATE_PINS](#) pin)

a function for getting power state of the peripheral

Parameters:

pin Power pin to get

Returns:

The result of the get or error code
Definition at line 91 of file halpwr.c.

int HAL_IO_PWRinit (void)

a function for opening PWR i/o-interface.

This function must be called before using any other PWR interface functions.

Returns:

The result if initializing was succeeded
Definition at line 13 of file halpwr.c.

int HAL_IO_PWRregisterPowerEventHandler (void*)(int) *function*)

a function for registering event handler for power events.

Parameters:

function call-back function for powerdown/overvoltage event

Returns:

The result if registering was succeeded
Definition at line 61 of file halpwr.c.

int HAL_IO_PWRsetPowerState ([PWR_STATE_PINS](#) pin, uint8 state)

a function for setting power state for peripheral

Parameters:

pin Power pin to set
state State for the pin

Returns:

The result of the set or error code
Definition at line 72 of file halpwr.c.

int HAL_IO_PWRuninit (void)

a function for closing PWR i/o-interface.

This function must be called when PWR i/o-interface is no longer needed.

Returns:

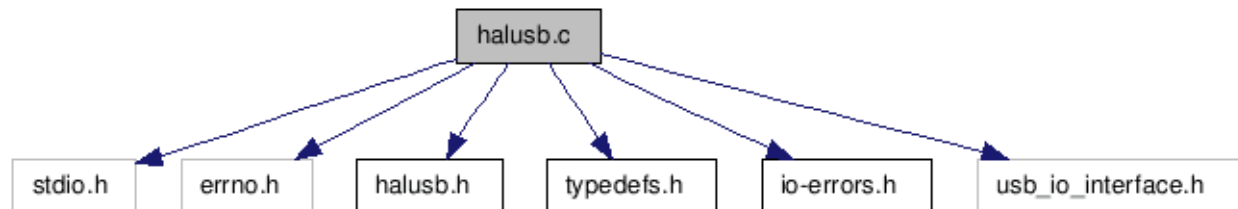
The result if closing was succeeded
Definition at line 28 of file halpwr.c.

halusb.c File Reference

USB IO interface file of the IO HAL.

```
#include <stdio.h>
#include <errno.h>
#include "halusb.h"
#include "typedefs.h"
#include "io-errors.h"
#include "usb_io_interface.h"
```

Include dependency graph for halusb.c:



Functions

- int [HAL_IO_USBinit](#) (void)
a function for opening USB i/o-interface.
- int [HAL_IO_USBuninit](#) (void)
a function for closing USB i/o-interface.
- int [HAL_IO_USBgetPlugged](#) (void)
a function for getting usb plugged signal state
- int [HAL_IO_USBregisterPluggedEventHandler](#) (void(*function)(int))
a function for registering event handler for usb plugged events.

Detailed Description

USB IO interface file of the IO HAL.

Author:

Teemu Keskinarkaus / CC Systems Oy

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

Function Documentation

int HAL_IO_USBgetPlugged (void)

a function for getting usb plugged signal state

Returns:

The result of the signal or error code

Definition at line 40 of file halusb.c.

int HAL_IO_USBinit (void)

a function for opening USB i/o-interface.

This function must be called before using any other USB interface functions.

Returns:

The result if initializing was succeeded

Definition at line 14 of file halusb.c.

int HAL_IO_USBregisterPluggedEventHandler (void*)(int) *function*

a function for registering event handler for usb plugged events.

Parameters:

function call-back function for usb plugged/unplugged event

Returns:

The result if registering was succeeded

Definition at line 51 of file halusb.c.

int HAL_IO_USBuninit (void)

a function for closing USB i/o-interface.

This function must be called when USB i/o-interface is no longer needed.

Returns:

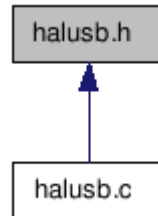
The result if closing was succeeded

Definition at line 29 of file halusb.c.

halusb.h File Reference

Cross HAL definitions for USB I/O routines.

This graph shows which files directly or indirectly include this file:



Functions

- int [HAL_IO_USBinit](#) (void)
a function for opening USB i/o-interface.
- int [HAL_IO_USBuninit](#) (void)
a function for closing USB i/o-interface.
- int [HAL_IO_USBgetPlugged](#) (void)
a function for getting usb plugged signal state
- int [HAL_IO_USBregisterPluggedEventHandler](#) (void(*function)(int))
a function for registering event handler for usb plugged events.

Detailed Description

Cross HAL definitions for USB I/O routines.

Author:

Teemu Keskinarkaus / CC Systems Oy

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

Function Documentation

int HAL_IO_USBgetPlugged (void)

a function for getting usb plugged signal state

Returns:

The result of the signal or error code

Definition at line 40 of file halusb.c.

int HAL_IO_USBinit (void)

a function for opening USB i/o-interface.

This function must be called before using any other USB interface functions.

Returns:

The result if initializing was succeeded
Definition at line 14 of file halusb.c.

int HAL_IO_USBregisterPluggedEventHandler (void*)(int) *function*

a function for registering event handler for usb plugged events.

Parameters:

function call-back function for usb plugged/unplugged event

Returns:

The result if registering was succeeded
Definition at line 51 of file halusb.c.

int HAL_IO_USBuninit (void)

a function for closing USB i/o-interface.

This function must be called when USB i/o-interface is no longer needed.

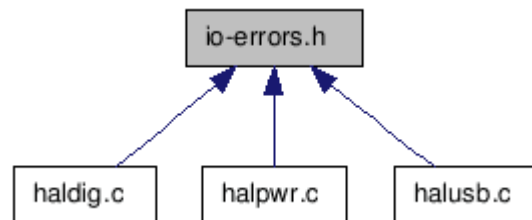
Returns:

The result if closing was succeeded
Definition at line 29 of file halusb.c.

io-errors.h File Reference

Cross HAL error definitions.

This graph shows which files directly or indirectly include this file:



Enumerations

- enum [IO_ERRORS](#) { [IO_OK](#) = 0, [IO_ERROR_NOT_INITIALIZED](#) = -1, [IO_ERROR_ALREADY_INITIALIZED](#) = -2, [IO_ERROR_INVALID_PARAM](#) = -3, [IO_ERROR_INVALID_COMMAND](#) = -4, [IO_ERROR_FAILED_TO_OPEN_INTERFACE](#) = -5 }
- Error values used in addition to other standard libc errors.*
-

Detailed Description

Cross HAL error definitions.

Author:

Teemu Keskinarkaus / CC Systems Oy

Definition in file [io-errors.h](#).

Enumeration Type Documentation

enum [IO_ERRORS](#)

Error values used in addition to other standard libc errors.

Enumerator:

IO_OK OK.

IO_ERROR_NOT_INITIALIZED Not initialized.

IO_ERROR_ALREADY_INITIALIZED Already initialized.

IO_ERROR_INVALID_PARAM Invalid param.

IO_ERROR_INVALID_COMMAND Invalid command.

IO_ERROR_FAILED_TO_OPEN_INTERFACE Failed to open.

Definition at line 11 of file io-errors.h.

Index

CAN0_LED	haldig.c, 4
haldig.h, 6	haldig.h, 7
CAN1_LED	HAL_IO_DIGledEnable
haldig.h, 6	haldig.c, 4
HAL_IO_DIGget	haldig.h, 7
haldig.c, 3	HAL_IO_DIGset
haldig.h, 6	haldig.c, 4
HAL_IO_DIGgetAll	haldig.h, 8
haldig.c, 3	HAL_IO_DIGunit
haldig.h, 7	haldig.c, 4
HAL_IO_DIGgetHardwareVersion	haldig.h, 8
haldig.c, 3	HAL_IO_PWRgetOvervoltage
haldig.h, 7	halpwr.c, 9
HAL_IO_DIGinit	halpwr.h, 13
haldig.c, 3	HAL_IO_PWRgetPowerDown
haldig.h, 7	halpwr.c, 10
HAL_IO_DIGledBlink	halpwr.h, 13
haldig.c, 3	HAL_IO_PWRgetPowerState
haldig.h, 7	halpwr.c, 10
HAL_IO_DIGledDisable	halpwr.h, 13

HAL_IO_PWRinit
 halpwr.c, 10
 halpwr.h, 13
 HAL_IO_PWRregisterPowerEventHandler
 halpwr.c, 10
 halpwr.h, 13
 HAL_IO_PWRsetPowerState
 halpwr.c, 10
 halpwr.h, 13
 HAL_IO_PWRuninit
 halpwr.c, 11
 halpwr.h, 14
 HAL_IO_USBgetPlugged
 halusb.c, 15
 halusb.h, 16
 HAL_IO_USBinit
 halusb.c, 15
 halusb.h, 16
 HAL_IO_USBregisterPluggedEventHandler
 halusb.c, 15
 halusb.h, 17
 HAL_IO_USBuninit
 halusb.c, 15
 halusb.h, 17
 haldig.c, 1
 HAL_IO_DIGget, 3
 HAL_IO_DIGgetAll, 3
 HAL_IO_DIGgetHardwareVersion, 3
 HAL_IO_DIGinit, 3
 HAL_IO_DIGledBlink, 3
 HAL_IO_DIGledDisable, 4
 HAL_IO_DIGledEnable, 4
 HAL_IO_DIGset, 4
 HAL_IO_DIGuninit, 4
 haldig.h, 4
 CAN0_LED, 6
 CAN1_LED, 6
 HAL_IO_DIGget, 6
 HAL_IO_DIGgetAll, 7
 HAL_IO_DIGgetHardwareVersion, 7
 HAL_IO_DIGinit, 7
 HAL_IO_DIGledBlink, 7
 HAL_IO_DIGledDisable, 7
 HAL_IO_DIGledEnable, 7
 HAL_IO_DIGset, 8
 HAL_IO_DIGuninit, 8
 INFO_LEDS, 6
 LED_AMBER, 6
 LED_COLORS, 6
 LED_GREEN, 6
 LED_RED, 6
 STATUS_LED, 6
 halpwr.c, 8
 HAL_IO_PWRgetOvervoltage, 9
 HAL_IO_PWRgetPowerDown, 10
 HAL_IO_PWRgetPowerState, 10
 HAL_IO_PWRinit, 10
 HAL_IO_PWRregisterPowerEventHandler, 10
 HAL_IO_PWRsetPowerState, 10
 HAL_IO_PWRuninit, 11
 halpwr.h, 11
 HAL_IO_PWRgetOvervoltage, 13
 HAL_IO_PWRgetPowerDown, 13
 HAL_IO_PWRgetPowerState, 13
 HAL_IO_PWRinit, 13
 HAL_IO_PWRregisterPowerEventHandler, 13
 HAL_IO_PWRsetPowerState, 13
 HAL_IO_PWRuninit, 14
 NUMBER_OF_PWR_PINS, 12
 PWR_PIN_ADDON, 12
 PWR_PIN_CAN, 12
 PWR_PIN_ETH, 12
 PWR_PIN_GPRS, 12
 PWR_PIN_USB, 12
 PWR_PIN_WLAN, 12
 PWR_STATE_PINS, 12
 halusb.c, 14
 HAL_IO_USBgetPlugged, 15
 HAL_IO_USBinit, 15
 HAL_IO_USBregisterPluggedEventHandler, 15
 HAL_IO_USBuninit, 15
 halusb.h, 16
 HAL_IO_USBgetPlugged, 16
 HAL_IO_USBinit, 16
 HAL_IO_USBregisterPluggedEventHandler, 17
 HAL_IO_USBuninit, 17
 INFO_LEDS
 haldig.h, 6
 IO_ERROR_ALREADY_INITIALIZED
 io-errors.h, 18
 IO_ERROR_FAILED_TO_OPEN_INTERFACE
 io-errors.h, 18
 IO_ERROR_INVALID_COMMAND
 io-errors.h, 18
 IO_ERROR_INVALID_PARAM
 io-errors.h, 18
 IO_ERROR_NOT_INITIALIZED
 io-errors.h, 18
 IO_ERRORS
 io-errors.h, 18
 IO_OK
 io-errors.h, 18
 io-errors.h, 17
 IO_ERROR_ALREADY_INITIALIZED, 18
 IO_ERROR_FAILED_TO_OPEN_INTERFACE,
 18
 IO_ERROR_INVALID_COMMAND, 18
 IO_ERROR_INVALID_PARAM, 18
 IO_ERROR_NOT_INITIALIZED, 18
 IO_ERRORS, 18
 IO_OK, 18
 LED_AMBER

haldig.h, 6
LED_COLORS
haldig.h, 6
LED_GREEN
haldig.h, 6
LED_RED
haldig.h, 6
NUMBER_OF_PWR_PINS
halpwr.h, 12
PWR_PIN_ADDON
halpwr.h, 12
PWR_PIN_CAN
halpwr.h, 12

PWR_PIN_ETH
halpwr.h, 12
PWR_PIN_GPRS
halpwr.h, 12
PWR_PIN_USB
halpwr.h, 12
PWR_PIN_WLAN
halpwr.h, 12
PWR_STATE_PINS
halpwr.h, 12
STATUS_LED
haldig.h, 6