Returns

IO_ErrorType

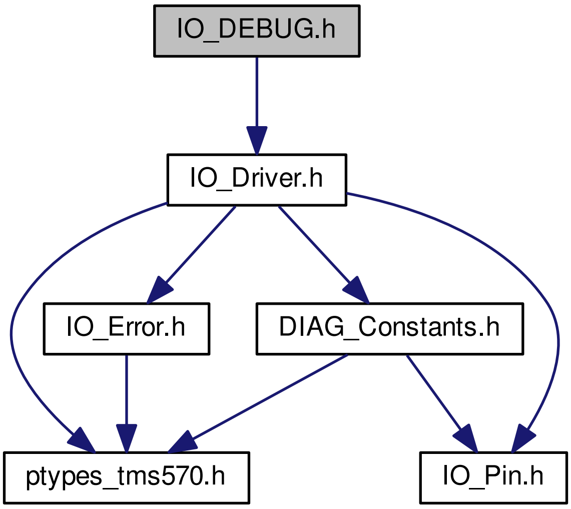
Return values

IO_E_OK	everything fine
IO_E_BUSY	message object busy - no data has been transmitted
IO_E_CAN_WRONG_HANDLE	invalid handle has been passed
IO_E_NULL_POINTER	null pointer has been passed
IO_E_INVALID_PARAMETER	an invalid message has been passed
IO_E_CHANNEL_NOT_CONFIGURED	the given handle has not been configured
IO_E_CAN_TIMEOUT	the CAN node reported a timeout
IO_E_INTERNAL_CSM	an internal error occurred
IO_E_UNKNOWN	an unknown error occurred

7.6 IO_DEBUG.h File Reference

IO Driver functions for DEBUG utilities.

Include dependency graph for IO_DEBUG.h:



Macros

- #define IO_DEBUG_OUTPUT_PIN_0 0U

- #define IO_DEBUG_OUTPUT_PIN_1 1U
- #define IO_DEBUG_OUTPUT_PIN_2 2U
- #define IO_DEBUG_WD_NORMAL 0U
- #define IO_DEBUG_WD_STATE_ACTIVE 2U
- #define IO_DEBUG_WD_STATE_PREPARED 1U
- #define IO_DEBUG_WD_STATE_SAFE 3U
- #define IO_DEBUG_WD_STATE_UNKNOWN 4U

Functions

- IO_ErrorType IO_DEBUG_GetTxStatus (void)
  Checks if the stdout transmission buffers are empty.
- IO_ErrorType IO_DEBUG_GetWatchdogState (ubyte1 ∗const wd_state)
  Returns the current debug state of the watchdog module.
- IO_ErrorType IO_DEBUG_SetOutputPin (ubyte1 pin, bool value)
  Sets a debug output pin (LED).
- IO_ErrorType IO_DEBUG_StdioDeInit (void)
  Deinitializes the UART interface for standard IO usage.
- IO_ErrorType IO_DEBUG_StdioInit (void)
  Initializes the UART interface for standard IO usage.

7.6.1 Detailed Description

IO Driver functions for DEBUG utilities.

Provides functions to set debug output pins (LEDs), to interface the UART in an Stdio like manner (e.g. printf) and to obtain the watchdog's state when running in debug mode.

The pins (LEDs) are only accessible via the JTAG adapter board and thus only with open devices and connected JTAG adapter board. These pins are only suitable for debugging purposes, like function runtime measurements, or debug status outputs.

DEBUG-API Usage:

- Examples for DEBUG API functions

Note

> The standard output line buffered. This means that strings which do not end with a line break character (`'\n'`) and are less than 80 characters long will stay in the stdout buffer. This buffer can be flushed immediately with `fflush(stdout);`

7.6.2 DEBUG Code Examples

Examples for using the DEBUG API

7.6.2.1 Example for DEBUG pin setting

```
// set debug pin 0 (LED 0) to high
IO_DEBUG_SetOutputPin(IO_DEBUG_OUTPUT_PIN_0, //pin 0
                      TRUE);                 //set high
```