

SIO4/8

Four/Eight Channel High Speed Serial I/O

**All SIO4 and SIO8 Models
All Form Factors
All Standard SYNC Versions**

SYNC Protocol Library Reference Manual

**Manual Revision: May 17, 2014
Library Release Version: 1.8**

**General Standards Corporation
8302A Whitesburg Drive
Huntsville, AL 35802
Phone: (256) 880-8787
Fax: (256) 880-8788
URL: <http://www.generalstandards.com>
E-mail: sales@generalstandards.com
E-mail: support@generalstandards.com**

Preface

Copyright © 2012-2014, **General Standards Corporation**

Additional copies of this manual or other literature may be obtained from:

General Standards Corporation
8302A Whitesburg Dr.
Huntsville, Alabama 35802
Phone: (256) 880-8787
FAX: (256) 880-8788
URL: <http://www.generalstandards.com/>
E-mail: sales@generalstandards.com

General Standards Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Although extensive editing and reviews are performed before release to ECO control, **General Standards Corporation** assumes no responsibility for any errors that may exist in this document. No commitment is made to update or keep current the information contained in this document.

General Standards Corporation does not assume any liability arising out of the application or use of any product or circuit described herein, nor is any license conveyed under any patent rights or any rights of others.

General Standards Corporation assumes no responsibility for any consequences resulting from omissions or errors in this manual or from the use of information contained herein.

General Standards Corporation reserves the right to make any changes, without notice, to this product to improve reliability, performance, function, or design.

ALL RIGHTS RESERVED.

The Purchaser of this software may use or modify in source form the subject software, but not to re-market or distribute it to outside agencies or separate internal company divisions. The software, however, may be embedded in the Purchaser's distributed software. In the event the Purchaser's customers require the software source code, then they would have to purchase their own copy of the software.

General Standards Corporation makes no warranty of any kind with regard to this software, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose and makes this software available solely on an "as-is" basis. **General Standards Corporation** reserves the right to make changes in this software without reservation and without notification to its users.

The information in this document is subject to change without notice. This document may be copied or reproduced provided it is in support of products from **General Standards Corporation**. For any other use, no part of this document may be copied or reproduced in any form or by any means without prior written consent of **General Standards Corporation**.

GSC is a trademark of **General Standards Corporation**.

PLX and PLX Technology are trademarks of PLX Technology, Inc.

Zilog and Z16C30 are trademarks of Zilog, Inc.

Table of Contents

1. Introduction.....	4
1.1. Purpose	4
1.2. Reference Material.....	4
2. The SYNC Protocol Library	5
2.1. Data Structures	5
2.1.1. sio4_sync_t.....	5
2.1.2. sio4_sync_rx_t.....	5
2.1.3. sio4_sync_tx_t.....	7
2.2. Functions.....	10
2.2.1. sio4_sync_version().....	10
2.2.2. sio4_sync_get().....	11
2.2.3. sio4_sync_set()	12
2.2.4. sio4_sync_gpio_rx()	13
2.2.5. sio4_sync_gpio_tx()	13
2.2.6. sio4_sync_rx_get()	14
2.2.7. sio4_sync_rx_set().....	15
2.2.8. sio4_sync_tx_get()	16
2.2.9. sio4_sync_tx_set().....	17
Document History	19

1. Introduction

This document provides information on the SYNC Protocol Library, which is a library designed to facilitate use of the SYNC model SIO4 and SIO8 model boards. This reference manual applies to library release version 1.8. This library version is designed to work with the 2.x series driver released on the same day as this version of the library.

NOTE: The device models listed on the front cover are those that are specifically supported by this release of the SYNC Protocol Library. Other models may be supported, though the level of support may vary.

1.1. Purpose

The purpose of this document is to describe the SYNC Protocol Library for the SIO4 Device Driver. The library is provided in source form for use as a statically linked library.

1.2. Reference Material

The following reference material may be of particular benefit in using the SYNC Protocol Library and the SIO4. The specifications provide the information necessary for an in depth understanding of the specialized features implemented on this board.

- The *SIO4/SIO8 Driver Reference Manual* from General Standards Corporation.
- The applicable *SIO4/SIO8 User Manual* from General Standards Corporation.
- The *PCI Bus Master Interface Chip* data handbook for the PCI9056/9080 from PLX Technology, Inc.

PLX Technology Inc.
870 Maude Avenue
Sunnyvale, California 94085 USA
Phone: 1-800-759-3735
WEB: <http://www.plxtech.com/>

- The *Z16C30 USC User's Manual* from Zilog. *
- The *Z16C30 Electronic Programmer's Manual* from Zilog (Zilog part number ZEPMDC00001). *

* The Zilog material is available from:

Zilog, Inc.
910 E Hamilton Ave
CAMPBELL, CA 95008 USA
Phone: 1-408-558-8500
WEB: <http://www.zilog.com/>

2. The SYNC Protocol Library

The driver release is accompanied by a number of items besides the documentation. One of the items included is this SYNC Protocol Library, which is described below. The library includes a small set of functions and corresponding data structures. The functions are defined via the header file `sio4\sync\lib\sio4_sync.h`. The static library is file `sio4\sync\lib\sio4_sync`. (The exact location and file extension depends on the OS in use.)

NOTE: Contact General Standards Corporation if additional driver functionality is required.

2.1. Data Structures

The library defines the following data structures.

2.1.1. `sio4_sync_t`

This structure is used to define SYNC specific configuration items that are independent of the transmitter and the receiver.

Definition

```
typedef struct
{
    int      dce_enable;    // 0 = No, !0 = Yes (Precedence over DTE.)
    int      dte_enable;    // 0 = No, !0 = Yes (If DCE disabled.)

    struct
    {
        int  enable;        // 0 = No (disble), !0 = Yes (enable)
        int  internal;      // 0 = No (Extern), !0 = Yes (Intern)
    } lb;
} sio4_sync_t;
```

Fields	Description
<code>dce_enable</code>	If non-zero this enables DCE cable interface operation. *
<code>dte_enable</code>	If non-zero this enables DTE cable interface operation. *
<code>lb</code>	This structure contains loop back settings.
<code>lb.enable</code>	If non-zero this enables loop back operation.
<code>lb.internal</code>	If loopback is enabled this selects between internal and external loopback. If non-zero, the internal loopback is selected. If zero, then external loopback is selected.

* If both are selected, then DCE is enabled. If neither is selected, then legacy operation is presumed.

2.1.2. `sio4_sync_rx_t`

This structure is used to define receiver specific SYNC configuration items.

Definition

```
typedef struct
{
    int      enable;        // SIO4_IOCTL_SYNC_RX_ENABLE
    int      bit_order;     // SIO4_IOCTL_SYNC_RX_BIT_ORDER
    int      reset;         // 0 = no error/reset, !0 = error/reset
    ul6      word_size;     // SIO4_IOCTL_SYNC_RX_BIT_COUNT, READ-ONLY
```

```

struct
{
    s32  cfg;           // SIO4_IOCTL_SYNC_RXC_CFG
    s32  pol;           // SIO4_IOCTL_SYNC_RXC_POL
} clock;

struct
{
    s32  cfg;           // SIO4_IOCTL_SYNC_RXE_CFG
    s32  pol;           // SIO4_IOCTL_SYNC_RXE_POL
} env;

struct
{
    s32  cfg;           // SIO4_IOCTL_SYNC_RXD_CFG
    s32  legacy;        // SIO4_IOCTL_SYNC_LEG_RXD_CFG
} data;
} sio4_sync_rx_t;

```

Fields	Description						
enable	<p>This field specifies the receiver enable state. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_ENABLE_NO</td><td>The receiver is disabled.</td></tr> <tr> <td>SIO4_SYNC_ENABLE_YES</td><td>The receiver is enabled.</td></tr> </table>	Options	Description	SIO4_SYNC_ENABLE_NO	The receiver is disabled.	SIO4_SYNC_ENABLE_YES	The receiver is enabled.
Options	Description						
SIO4_SYNC_ENABLE_NO	The receiver is disabled.						
SIO4_SYNC_ENABLE_YES	The receiver is enabled.						
bit_order	<p>This field specifies the bit receive order. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_BIT_ORDER_LSB</td><td>Receive the least significant bit first.</td></tr> <tr> <td>SIO4_SYNC_BIT_ORDER_MSB</td><td>Receive the most significant bit first.</td></tr> </table>	Options	Description	SIO4_SYNC_BIT_ORDER_LSB	Receive the least significant bit first.	SIO4_SYNC_BIT_ORDER_MSB	Receive the most significant bit first.
Options	Description						
SIO4_SYNC_BIT_ORDER_LSB	Receive the least significant bit first.						
SIO4_SYNC_BIT_ORDER_MSB	Receive the most significant bit first.						
reset	<p>SET: If non-zero, then any receive bit count error is cleared. If zero, then any receiver bit error is not cleared.</p> <p>GET: If zero, then no bit error occurred. If non-zero, then a bit error occurred.</p>						
word_size	When the settings are retrieved, this is the number of bits in the most recently received data. The value will be from zero to 0xFFFF.						
clock	This structure contains Rx Clock settings.						
clock. cfg	<p>This field specifies the basic clock signal configuration. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_RXC_CFG_FALL_EDGE</td><td>The receiver clocks data in on the clock's falling edge.</td></tr> <tr> <td>SIO4_SYNC_RXC_CFG_RISE_EDGE</td><td>The receiver clocks data in on the clock's rising edge.</td></tr> </table>	Options	Description	SIO4_SYNC_RXC_CFG_FALL_EDGE	The receiver clocks data in on the clock's falling edge.	SIO4_SYNC_RXC_CFG_RISE_EDGE	The receiver clocks data in on the clock's rising edge.
Options	Description						
SIO4_SYNC_RXC_CFG_FALL_EDGE	The receiver clocks data in on the clock's falling edge.						
SIO4_SYNC_RXC_CFG_RISE_EDGE	The receiver clocks data in on the clock's rising edge.						
clock. pol	<p>This field specifies the clock's polarity. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_RXC_CFG_FALL_EDGE</td><td>The receiver clocks data in on the clock's falling edge.</td></tr> <tr> <td>SIO4_SYNC_RXC_CFG_RISE_EDGE</td><td>The receiver clocks data in on the clock's rising edge.</td></tr> </table>	Options	Description	SIO4_SYNC_RXC_CFG_FALL_EDGE	The receiver clocks data in on the clock's falling edge.	SIO4_SYNC_RXC_CFG_RISE_EDGE	The receiver clocks data in on the clock's rising edge.
Options	Description						
SIO4_SYNC_RXC_CFG_FALL_EDGE	The receiver clocks data in on the clock's falling edge.						
SIO4_SYNC_RXC_CFG_RISE_EDGE	The receiver clocks data in on the clock's rising edge.						
env	This structure contains Rx Envelope settings.						

env. cfg	<p>This field specifies the basic envelope signal configuration. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_RXE_CFG_ACTIVE_HI</td><td>The envelope is high when asserted.</td></tr> <tr> <td>SIO4_SYNC_RXE_CFG_ACTIVE_LO</td><td>The envelope is low when asserted.</td></tr> <tr> <td>SIO4_SYNC_RXE_CFG_DISABLE</td><td>The envelope is disabled.</td></tr> </table>	Options	Description	SIO4_SYNC_RXE_CFG_ACTIVE_HI	The envelope is high when asserted.	SIO4_SYNC_RXE_CFG_ACTIVE_LO	The envelope is low when asserted.	SIO4_SYNC_RXE_CFG_DISABLE	The envelope is disabled.
Options	Description								
SIO4_SYNC_RXE_CFG_ACTIVE_HI	The envelope is high when asserted.								
SIO4_SYNC_RXE_CFG_ACTIVE_LO	The envelope is low when asserted.								
SIO4_SYNC_RXE_CFG_DISABLE	The envelope is disabled.								
env. pol	<p>This field specifies the polarity of the envelope signal. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_ENV_POL_ACTIVE_HI</td><td>The envelope is high when asserted.</td></tr> <tr> <td>SIO4_SYNC_ENV_POL_ACTIVE_LO</td><td>The envelope is low when asserted.</td></tr> </table>	Options	Description	SIO4_SYNC_ENV_POL_ACTIVE_HI	The envelope is high when asserted.	SIO4_SYNC_ENV_POL_ACTIVE_LO	The envelope is low when asserted.		
Options	Description								
SIO4_SYNC_ENV_POL_ACTIVE_HI	The envelope is high when asserted.								
SIO4_SYNC_ENV_POL_ACTIVE_LO	The envelope is low when asserted.								
data	This structure contains Rx Data settings.								
data. cfg	<p>This field specifies the basic data signal configuration. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_RXD_CFG_ACTIVE_HI</td><td>The data signal uses active high logic.</td></tr> <tr> <td>SIO4_SYNC_RXD_CFG_ACTIVE_LO</td><td>The data signal uses active low logic.</td></tr> </table>	Options	Description	SIO4_SYNC_RXD_CFG_ACTIVE_HI	The data signal uses active high logic.	SIO4_SYNC_RXD_CFG_ACTIVE_LO	The data signal uses active low logic.		
Options	Description								
SIO4_SYNC_RXD_CFG_ACTIVE_HI	The data signal uses active high logic.								
SIO4_SYNC_RXD_CFG_ACTIVE_LO	The data signal uses active low logic.								
data. legacy	<p>This field specifies the basic data signal configuration for legacy cable configurations. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_LEG_RXD_CFG_LOW</td><td>The data signal is routed to the lower cable signal pins.</td></tr> <tr> <td>SIO4_SYNC_LEG_RXD_CFG_TRI</td><td>The data signal is disabled.</td></tr> <tr> <td>SIO4_SYNC_LEG_RXD_CFG_UP</td><td>The data signal is routed to the upper cable signal pins.</td></tr> </table>	Options	Description	SIO4_SYNC_LEG_RXD_CFG_LOW	The data signal is routed to the lower cable signal pins.	SIO4_SYNC_LEG_RXD_CFG_TRI	The data signal is disabled.	SIO4_SYNC_LEG_RXD_CFG_UP	The data signal is routed to the upper cable signal pins.
Options	Description								
SIO4_SYNC_LEG_RXD_CFG_LOW	The data signal is routed to the lower cable signal pins.								
SIO4_SYNC_LEG_RXD_CFG_TRI	The data signal is disabled.								
SIO4_SYNC_LEG_RXD_CFG_UP	The data signal is routed to the upper cable signal pins.								

2.1.3. sio4_sync_tx_t

This structure is used to define transmitter specific SYNC configuration items.

Definition

```
typedef struct
{
    s32    enable;           // SIO4_IOCTL_SYNC_TX_ENABLE
    s32    empty_cfg;       // SIO4_IOCTL_TX_FIFO_EMPTY_CFG
    s32    bit_order;       // SIO4_IOCTL_SYNC_TX_BIT_ORDER
    u16    word_size;       // SIO4_IOCTL_SYNC_TX_WORD_SIZE
    u16    gap_size;        // SIO4_IOCTL_SYNC_TX_GAP_SIZE

    struct
    {
        s32  cfg;           // SIO4_IOCTL_SYNC_TXC_CFG
        s32  pol;           // SIO4_IOCTL_SYNC_TXC_POL
        s32  src;           // SIO4_IOCTL_SYNC_TXC_SRC
        s32  idle;          // SIO4_IOCTL_SYNC_TXC_IDLE
        s32  idle_cfg;      // SIO4_IOCTL_SYNC_TXC_IDLE_CFG
    } clock;

    struct
    {
        s32  cfg;           // SIO4_IOCTL_SYNC_TXE_CFG
        s32  pol;           // SIO4_IOCTL_SYNC_TXE_POL
    }
}
```

```

} env;

struct
{
    s32  cfg;           // SIO4_IOCTL_SYNC_TXD_CFG
    s32  idle;          // SIO4_IOCTL_SYNC_TXD_IDLE_CFG
    s32  legacy;        // SIO4_IOCTL_SYNC_LEG_TXD_CFG
} data;

s32      aux_clock;    // SIO4_IOCTL_SYNC_TXAUXC_CFG
s32      spare;        // SIO4_IOCTL_SYNC_TXSP_CFG
} sio4_sync_tx_t;

```

Fields	Description						
enable	<p>This field specifies the transmitter enable state. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_ENABLE_NO</td><td>The transmitter is disabled.</td></tr> <tr> <td>SIO4_SYNC_ENABLE_YES</td><td>The transmitter is enabled.</td></tr> </table>	Options	Description	SIO4_SYNC_ENABLE_NO	The transmitter is disabled.	SIO4_SYNC_ENABLE_YES	The transmitter is enabled.
Options	Description						
SIO4_SYNC_ENABLE_NO	The transmitter is disabled.						
SIO4_SYNC_ENABLE_YES	The transmitter is enabled.						
empty_cfg	<p>This field specifies the transmitter response to the Tx FIFO going empty. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_TX_FIFO_EMPTY_CFG_IGNORE</td><td>No actions takes place</td></tr> <tr> <td>SIO4_TX_FIFO_EMPTY_CFG_TX_OFF</td><td>The transmitter is disabled.</td></tr> </table>	Options	Description	SIO4_TX_FIFO_EMPTY_CFG_IGNORE	No actions takes place	SIO4_TX_FIFO_EMPTY_CFG_TX_OFF	The transmitter is disabled.
Options	Description						
SIO4_TX_FIFO_EMPTY_CFG_IGNORE	No actions takes place						
SIO4_TX_FIFO_EMPTY_CFG_TX_OFF	The transmitter is disabled.						
bit_order	<p>This field specifies the bit transmit order. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_BIT_ORDER_LSB</td><td>Transmit the least significant bit first.</td></tr> <tr> <td>SIO4_SYNC_BIT_ORDER_MSB</td><td>Transmit the most significant bit first.</td></tr> </table>	Options	Description	SIO4_SYNC_BIT_ORDER_LSB	Transmit the least significant bit first.	SIO4_SYNC_BIT_ORDER_MSB	Transmit the most significant bit first.
Options	Description						
SIO4_SYNC_BIT_ORDER_LSB	Transmit the least significant bit first.						
SIO4_SYNC_BIT_ORDER_MSB	Transmit the most significant bit first.						
word_size	This specifies the size, in bits, of each transmitted data word. The value eight (8) is most common. Valid values are from one to 0xFFFF.						
gap_size	This specifies the size of the gap, in single bit periods, that is inserted between consecutive data words. Valid values are from zero to 0xFFFF.						
clock	This structure contains Tx Clock settings.						
clock. cfg	<p>This field specifies the basic clock signal configuration. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_TXC_CFG_EXT</td><td>The transmitter is configured to use an external clock.</td></tr> <tr> <td>SIO4_SYNC_TXC_CFG_INT</td><td>The transmitter is configured to use an internal clock.</td></tr> </table>	Options	Description	SIO4_SYNC_TXC_CFG_EXT	The transmitter is configured to use an external clock.	SIO4_SYNC_TXC_CFG_INT	The transmitter is configured to use an internal clock.
Options	Description						
SIO4_SYNC_TXC_CFG_EXT	The transmitter is configured to use an external clock.						
SIO4_SYNC_TXC_CFG_INT	The transmitter is configured to use an internal clock.						
clock. pol	<p>This field specifies the clock's polarity. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_CLOCK_POL_FALL</td><td>The transmitter clocks data out on the clock's falling edge.</td></tr> <tr> <td>SIO4_SYNC_CLOCK_POL_RISE</td><td>The transmitter clocks data out on the clock's rising edge.</td></tr> </table>	Options	Description	SIO4_SYNC_CLOCK_POL_FALL	The transmitter clocks data out on the clock's falling edge.	SIO4_SYNC_CLOCK_POL_RISE	The transmitter clocks data out on the clock's rising edge.
Options	Description						
SIO4_SYNC_CLOCK_POL_FALL	The transmitter clocks data out on the clock's falling edge.						
SIO4_SYNC_CLOCK_POL_RISE	The transmitter clocks data out on the clock's rising edge.						

clock. src	<p>This field specifies the clock's source. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_TXC_SRC_0</td><td>The clock input is a logic low.</td></tr> <tr> <td>SIO4_SYNC_TXC_SRC_1</td><td>The clock input is a logic high.</td></tr> <tr> <td>SIO4_SYNC_TXC_SRC_EXT_FALL</td><td>Data is transmitted on the falling edge of the external clock.</td></tr> <tr> <td>SIO4_SYNC_TXC_SRC_EXT_RISE</td><td>Data is transmitted on the rising edge of the external clock.</td></tr> <tr> <td>SIO4_SYNC_TXC_SRC_OSC_HALF_FALL</td><td>Data is transmitted on the falling edge of the internal clock, which is divided by two.</td></tr> <tr> <td>SIO4_SYNC_TXC_SRC_OSC_HALF_RISE</td><td>Data is transmitted on the rising edge of the internal clock, which is divided by two.</td></tr> </table>	Options	Description	SIO4_SYNC_TXC_SRC_0	The clock input is a logic low.	SIO4_SYNC_TXC_SRC_1	The clock input is a logic high.	SIO4_SYNC_TXC_SRC_EXT_FALL	Data is transmitted on the falling edge of the external clock.	SIO4_SYNC_TXC_SRC_EXT_RISE	Data is transmitted on the rising edge of the external clock.	SIO4_SYNC_TXC_SRC_OSC_HALF_FALL	Data is transmitted on the falling edge of the internal clock, which is divided by two.	SIO4_SYNC_TXC_SRC_OSC_HALF_RISE	Data is transmitted on the rising edge of the internal clock, which is divided by two.
Options	Description														
SIO4_SYNC_TXC_SRC_0	The clock input is a logic low.														
SIO4_SYNC_TXC_SRC_1	The clock input is a logic high.														
SIO4_SYNC_TXC_SRC_EXT_FALL	Data is transmitted on the falling edge of the external clock.														
SIO4_SYNC_TXC_SRC_EXT_RISE	Data is transmitted on the rising edge of the external clock.														
SIO4_SYNC_TXC_SRC_OSC_HALF_FALL	Data is transmitted on the falling edge of the internal clock, which is divided by two.														
SIO4_SYNC_TXC_SRC_OSC_HALF_RISE	Data is transmitted on the rising edge of the internal clock, which is divided by two.														
clock. idle	<p>This field specifies the clock's operation when data is not be transmitted. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_TXC_IDLE_NO</td><td>The clock is idle.</td></tr> <tr> <td>SIO4_SYNC_TXC_IDLE_YES</td><td>The clock continues to run.</td></tr> </table>	Options	Description	SIO4_SYNC_TXC_IDLE_NO	The clock is idle.	SIO4_SYNC_TXC_IDLE_YES	The clock continues to run.								
Options	Description														
SIO4_SYNC_TXC_IDLE_NO	The clock is idle.														
SIO4_SYNC_TXC_IDLE_YES	The clock continues to run.														
clock. idle_cfg	<p>This field specifies the clock's operation when data is not be transmitted. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_TXC_IDLE_CFG_ACTIVE</td><td>The clock continues to run.</td></tr> <tr> <td>SIO4_SYNC_TXC_IDLE_CFG_IDLE_0</td><td>The signal is held low.</td></tr> <tr> <td>SIO4_SYNC_TXC_IDLE_CFG_IDLE_1</td><td>The signal is held high.</td></tr> </table>	Options	Description	SIO4_SYNC_TXC_IDLE_CFG_ACTIVE	The clock continues to run.	SIO4_SYNC_TXC_IDLE_CFG_IDLE_0	The signal is held low.	SIO4_SYNC_TXC_IDLE_CFG_IDLE_1	The signal is held high.						
Options	Description														
SIO4_SYNC_TXC_IDLE_CFG_ACTIVE	The clock continues to run.														
SIO4_SYNC_TXC_IDLE_CFG_IDLE_0	The signal is held low.														
SIO4_SYNC_TXC_IDLE_CFG_IDLE_1	The signal is held high.														
env	This structure contains Tx Envelope settings.														
env. cfg	<p>This field specifies the basic envelope signal configuration. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_TXE_CFG_ACTIVE_LO</td><td>The envelope is low when asserted.</td></tr> <tr> <td>SIO4_SYNC_TXE_CFG_ACTIVE_HI</td><td>The envelope is high when asserted.</td></tr> <tr> <td>SIO4_SYNC_TXE_CFG_OUT_0</td><td>The signal is driven low.</td></tr> <tr> <td>SIO4_SYNC_TXE_CFG_OUT_1</td><td>The signal is driven high.</td></tr> </table>	Options	Description	SIO4_SYNC_TXE_CFG_ACTIVE_LO	The envelope is low when asserted.	SIO4_SYNC_TXE_CFG_ACTIVE_HI	The envelope is high when asserted.	SIO4_SYNC_TXE_CFG_OUT_0	The signal is driven low.	SIO4_SYNC_TXE_CFG_OUT_1	The signal is driven high.				
Options	Description														
SIO4_SYNC_TXE_CFG_ACTIVE_LO	The envelope is low when asserted.														
SIO4_SYNC_TXE_CFG_ACTIVE_HI	The envelope is high when asserted.														
SIO4_SYNC_TXE_CFG_OUT_0	The signal is driven low.														
SIO4_SYNC_TXE_CFG_OUT_1	The signal is driven high.														
env. pol	<p>This field specifies the polarity of the envelope signal. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_ENV_POL_ACTIVE_HI</td><td>The envelope is high when asserted.</td></tr> <tr> <td>SIO4_SYNC_ENV_POL_ACTIVE_LO</td><td>The envelope is low when asserted.</td></tr> </table>	Options	Description	SIO4_SYNC_ENV_POL_ACTIVE_HI	The envelope is high when asserted.	SIO4_SYNC_ENV_POL_ACTIVE_LO	The envelope is low when asserted.								
Options	Description														
SIO4_SYNC_ENV_POL_ACTIVE_HI	The envelope is high when asserted.														
SIO4_SYNC_ENV_POL_ACTIVE_LO	The envelope is low when asserted.														
data	This structure contains Tx Data settings.														
data. cfg	<p>This field specifies the basic data signal configuration. Valid options are as given below.</p> <table> <tr> <th>Options</th><th>Description</th></tr> <tr> <td>SIO4_SYNC_TXD_CFG_ACTIVE_HI</td><td>The data signal uses active high logic.</td></tr> <tr> <td>SIO4_SYNC_TXD_CFG_ACTIVE_LO</td><td>The data signal uses active low logic.</td></tr> <tr> <td>SIO4_SYNC_TXD_CFG_OUT_0</td><td>The data signal is driven low.</td></tr> <tr> <td>SIO4_SYNC_TXD_CFG_OUT_1</td><td>The data signal is driven high.</td></tr> </table>	Options	Description	SIO4_SYNC_TXD_CFG_ACTIVE_HI	The data signal uses active high logic.	SIO4_SYNC_TXD_CFG_ACTIVE_LO	The data signal uses active low logic.	SIO4_SYNC_TXD_CFG_OUT_0	The data signal is driven low.	SIO4_SYNC_TXD_CFG_OUT_1	The data signal is driven high.				
Options	Description														
SIO4_SYNC_TXD_CFG_ACTIVE_HI	The data signal uses active high logic.														
SIO4_SYNC_TXD_CFG_ACTIVE_LO	The data signal uses active low logic.														
SIO4_SYNC_TXD_CFG_OUT_0	The data signal is driven low.														
SIO4_SYNC_TXD_CFG_OUT_1	The data signal is driven high.														
data. legacy	<p>This field specifies the basic data signal configuration for legacy cable configurations. Valid options are as given below.</p>														

	Options	Description
	SIO4_SYNC_LEG_TXD_CFG_LOW	The data signal is routed to the lower cable signal pins.
	SIO4_SYNC_LEG_TXD_CFG_TRI	The data signal is tri-stated.
	SIO4_SYNC_LEG_TXD_CFG_UP	The data signal is routed to the upper cable signal pins.
	SIO4_SYNC_LEG_TXD_CFG_UP_LOW	The data signal is routed to both the upper and the upper cable signal pins.
aux_clock	This field specifies the basic data signal configuration for Tx Auxiliary Clock output signal. Valid options are as given below.	
	Options	Description
	SIO4_SYNC_TXAUXC_CFG_TRI	The signal is tri-stated.
	SIO4_SYNC_TXAUXC_CFG_OSC_HALF	The data signal is driven by the on-board oscillator, which is divided in half.
	SIO4_SYNC_TXAUXC_CFG_OUT_0	The output signal is driven low.
	SIO4_SYNC_TXAUXC_CFG_OUT_1	The output signal is driven high.
spare	This field specifies the basic data signal configuration for Tx Auxiliary Clock output signal. Valid options are as given below.	
	Options	Description
	SIO4_SYNC_TXSP_CFG_DISABLE	The signal is disabled.
	SIO4_SYNC_TXSP_CFG_INPUT	The signal is configured as an input.
	SIO4_SYNC_TXSP_CFG_OUT_0	The output signal is driven low.
	SIO4_SYNC_TXSP_CFG_OUT_1	The output signal is driven high.

2.2. Functions

This library includes the following functions.

2.2.1. sio4_sync_version()

This function retrieves the library version number as well as its build data and time string.

Prototype

```
void sio4_sync_version(const char** version, const char** built);
```

Argument	Description
version	A pointer to the library's version number is recorded here. The string referenced is a static buffer.
built	A pointer to the library's build data and time is recorded here. The string referenced is a static buffer.

Return Value	Description
None	N/A

Example

```
#include <stdio.h>

#include "sio4_sync_dsl.h"
```

```

void sio4_sync_version_get(
    const char**    version,
    const char**    built,
    int             verbose)
{
    const char* blt;
    const char* ver;

    sio4_sync_version(&ver, &blt);

    if (verbose)
    {
        printf("SIO4-SYNC Library Version:\n");
        printf("  Version: %s\n", ver );
        printf("  Built:   %s\n", blt);
    }

    if (version)
        version[0] = ver;

    if (built)
        built[0]   = blt;
}

```

2.2.2. sio4_sync_get()

This function retrieves the SYNC settings that are independent of both the transmitter and the receiver.

Prototype

```
int sio4_sync_get(int fd, sio4_sync_t* sync);
```

Argument	Description
fd	This is the file descriptor used to access the device.
sync	The settings requested are recorded here.

Return Value	Description
-1	An error occurred. Consult <code>errno</code> .
0	The operation succeeded.

Example

```

#include <errno.h>
#include <stdio.h>

#include "sio4_sync_dsl.h"

int sio4_sync_dte_get(int fd, int* enable)
{
    int          sts;
    sio4_sync_t sync;

```

```

    sts          = sio4_sync_get(fd, &sync);
    enable[0]    = sync.dte_enable;

    if (sts == -1)
        printf("sio4_sync_get() failure, errno = %d\n", errno);

    return(sts);
}

```

2.2.3. sio4_sync_set()

This function applies the SYNC settings that are independent of both the transmitter and the receiver.

Prototype

```
int sio4_sync_set(int fd, const sio4_sync_t* sync);
```

Argument	Description
fd	This is the file descriptor used to access the device.
sync	The settings to apply are recorded here.

Return Value	Description
-1	An error occurred. Consult errno.
0	The operation succeeded.

Example

```

#include <errno.h>
#include <stdio.h>

#include "sio4_sync_dsl.h"

int sio4_sync_dce_set(int fd, int enable)
{
    const char* str = "sio4_sync_get";
    int sts;
    sio4_sync_t sync;

    sts = sio4_sync_get(fd, &sync);

    if (sts == 0)
    {
        sync.dce_enable= enable;
        sts          = sio4_sync_set(fd, &sync);
        str          = "sio4_sync_set";
    }

    if (sts == -1)
        printf("%s() failure, errno = %d\n", str, errno);
}

```

```

    return(sts);
}

```

2.2.4. sio4_sync_gpio_rx()

This function retrieves the cable signal levels for those signals configured for GPIO operation.

Prototype

```
int sio4_sync_gpio_rx(int fd, u32* value);
```

Argument	Description
fd	This is the file descriptor used to access the device.
value	The value read is recorded here.

Return Value	Description
-1	An error occurred. Consult <code>errno</code> .
0	The operation succeeded.

Example

```

#include <errno.h>
#include <stdio.h>

#include "sio4_sync_dsl.h"

int sio4_sync_gpio_read_tx(int fd, u32* value)
{
    int sts;

    sts      = sio4_sync_gpio_rx(fd, value);
    value[0] &= 0x2F0;

    if (sts == -1)
    {
        printf("sio4_sync_gpio_read() failure, errno = %d\n",
               errno);
    }

    return(sts);
}

```

2.2.5. sio4_sync_gpio_tx()

This function applies an update to the cable signal levels for those signals configured for GPIO operation.

Prototype

```
int sio4_sync_gpio_tx(int fd, u32 value);
```

Argument	Description
fd	This is the file descriptor used to access the device.
value	This is the value to apply.

Return Value	Description
-1	An error occurred. Consult <code>errno</code> .
0	The operation succeeded.

Example

```
#include <errno.h>
#include <stdio.h>

#include "sio4_sync_dsl.h"

int sio4_sync_gpio_mod(int fd, u32 value, u32 mask)
{
    const char* str = "sio4_sync_gpio_rx";
    int        sts;
    u32        v;

    sts = sio4_sync_gpio_rx(fd, &v);

    if (sts == 0)
    {
        value = (value & mask) | (v & ~mask);
        sts   = sio4_sync_gpio_tx(fd, value);
        str    = "sio4_sync_gpio_tx";
    }

    if (sts == -1)
        printf("%s() failure, errno = %d\n", str, errno);

    return(sts);
}
```

2.2.6. sio4_sync_rx_get()

This function retrieves the receiver specific SYNC setting.

Prototype

```
int sio4_sync_rx_get(int fd, sio4_sync_rx_t* rx);
```

Argument	Description
fd	This is the file descriptor used to access the device.
rx	The settings are retrieved and are recorded here.

Return Value	Description
-1	An error occurred. Consult <code>errno</code> .
0	The operation succeeded.

Example

```
#include <errno.h>
#include <stdio.h>

#include "sio4_sync_dsl.h"

int sio4_sync_rx_data_get(int fd, int* cfg, int* legacy)
{
    sio4_sync_rx_t rx;
    int sts;

    sts = sio4_sync_rx_get(fd, &rx);
    cfg[0] = rx.data.cfg;
    legacy[0] = rx.data.legacy;

    if (sts == -1)
        printf("sio4_sync_rx_get() failure, errno = %d\n", errno);

    return(sts);
}
```

2.2.7. sio4_sync_rx_set()

This function applies receiver specific SYNC setting.

Prototype

```
int sio4_sync_rx_set(int fd, const sio4_sync_rx_t* rx);
```

Argument	Description
fd	This is the file descriptor used to access the device.
rx	The settings to apply are recorded here.

Return Value	Description
-1	An error occurred. Consult <code>errno</code> .
0	The operation succeeded.

Example

```
#include <errno.h>
#include <stdio.h>

#include "sio4_sync_dsl.h"

int sio4_sync_rx_env_set(int fd, int cfg, int pol)
```

```

{
    sio4_sync_rx_t rx;
    const char* str = "sio4_sync_rx_get";
    int sts;

    sts = sio4_sync_rx_get(fd, &rx);

    if (sts == 0)
    {
        rx.env.cfg = cfg;
        rx.env.pol = pol;
        sts = sio4_sync_rx_set(fd, &rx);
        str = "sio4_sync_rx_set";
    }

    if (sts == -1)
        printf("%s() failure, errno = %d\n", str, errno);

    return(sts);
}

```

2.2.8. sio4_sync_tx_get()

This function retrieves the transmitter specific SYNC setting.

Prototype

```
int sio4_sync_tx_get(int fd, sio4_sync_tx_t* tx);
```

Argument	Description
fd	This is the file descriptor used to access the device.
tx	The settings are retrieved and are recorded here.

Return Value	Description
-1	An error occurred. Consult errno.
0	The operation succeeded.

Example

```

#include <errno.h>
#include <stdio.h>

#include "sio4_sync_dsl.h"

int sio4_sync_tx_env_get(int fd, int* cfg, int* pol)
{
    int sts;
    sio4_sync_tx_t tx;

    sts = sio4_sync_tx_get(fd, &tx);
    cfg[0] = tx.env.cfg;

```



```

    pol[0] = tx.env.pol;

    if (sts == -1)
        printf("sio4_sync_tx_get() failure, errno = %d\n", errno);

    return(sts);
}

```

2.2.9. sio4_sync_tx_set()

This function applies transmitter specific SYNC setting.

Prototype

```
int sio4_sync_tx_set(int fd, const sio4_sync_tx_t* tx);
```

Argument	Description
fd	This is the file descriptor used to access the device.
tx	The settings to apply are recorded here.

Return Value	Description
-1	An error occurred. Consult errno.
0	The operation succeeded.

Example

```

#include <errno.h>
#include <stdio.h>

#include "sio4_sync_dsl.h"

int sio4_sync_tx_data_set(int fd, int cfg, int idle, int legacy)
{
    const char* str = "sio4_sync_tx_get";
    int sts;
    sio4_sync_tx_t tx;

    sts = sio4_sync_tx_get(fd, &tx);

    if (sts == 0)
    {
        tx.data.cfg = cfg;
        tx.data.idle = idle;
        tx.data.legacy = legacy;
        sts = sio4_sync_tx_set(fd, &tx);
        str = "sio4_sync_tx_set";
    }

    if (sts == -1)
        printf("%s() failure, errno = %d\n", str, errno);
}

```

```
    return (sts);  
}
```

Document History

Revision	Description
May 17, 2014	Update the release date.
April 16, 2014	Update to library release version 1.8.
October 22, 2013	Update to library release version 1.7.
October 15, 2013	Update to library release version 1.6.
August 27, 2013	Update to library release version 1.5. Renamed library base name to <code>sio4_sync</code> . Rewrote library. Updated structure field names and valid value sets.
October 11, 2012	Update to driver release version 1.4. Renamed library base name to <code>sio4_sync</code> .
September 9, 2012	Initial library release, version 1.3, for the 2.x series driver.