CAN API

for Sital CAN/CAN FD IP Core

Programmer and Reference Guide

Rev 1.18

July 2020

Current documentation version does not include Returned Error

Table of Contents

1 Introduction

1.1 Scope

This document is the Programmer and Reference Guide for programming with the CAN API for the the following protocols: CAN / CAN FD.

1.2 Audience

This document assumes that the reader is familiar with the above specified protocols.

1.3 Related Documentation

SI3111 CAN BUS IP Core HSID User Manual

1.4 Support

If you have any question or require further assistance, use any of the following methods to contact Sital customer support:

- By Email: support@sitaltech.com
- By Phone: +972-9-7633300
- By Fax: +972-9-7663394

2 Concept & High Level Workflow

The CAN toolbox of API functions are intended to a SW developer who wishes to perform serial automotive bus communications using Sital Technology's CAN/CAN FD IP Core in one of its hardware implementations, i.e. through AXI, PCI, PCIe and other buses.

The target of the SW development is to transmit and receive messages on CAN bus.
The API functions let you manage multiple devices.

At minimum:

1. Initialize a device.
2. Transmit and receive messages simulteniously on a device.

It is advised to start the coding from examples which are provided.

3 CAN API Reference

3.1 sitalCan_DeviceInit

```
S16BIT sitalCan_DeviceInit      (
                                  U16BIT       deviceId
                                  U32BIT       baseAddress
                                  U16BIT       IRQnumber
                                  U16BIT       baseRate
                                  U16BIT       dataRate
                                )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
baseAddress	Base Address in hardware project
IRQnumber	Interrupt request - IRQ number in hardware project
baseRate	Base rate for CAN or arbitration rate for CAN FD protocol in Kbit/sec
dataRate	Data rate for CAN FD protocol in Kbit/sec.

Description

Since this is a “configurations and settings” function, it stops the device activities and data transfer. It is recommended to use ‘sitalCan_Close’ fuction before using this function in order to avoid data loss.

3.2 sitalCan_OpenDevices

S16BIT sitalCan_MapDevices	**(**		
		U16BIT*	***numberOfDevices***
	)		

Parameters

Outputs

numberOfDevices	Number of devices in the system 0, 1 - (sitalMaximum_DEVICES)

Description

Open all initialized devices.

If Number of Devices is zero – no devices in the system, if Number of Devices is one – there is one device in the system etc.

Filters are disabled, but prepared not to allow any CAN Message to go through.

Sequencers are disabled.

3.3 sitalCan_MapDevices

```
S16BIT sitalCan_MapDevices    (
                               U16BIT*          numberOfDevices
                              )
```

Parameters

Outputs

numberOfDevices Number of devices in the system 0, 1 - (sitalMaximum_DEVICES)

Description

Open initialized device, if the device is not opened yet.

If Number of Devices is zero – no devices in the system, if Number of Devices is one – there is one device in the system etc.

3.4 sitalCan_GetNextMsg

```
S16BIT sitalCan_GetNextMsg   (
                               U16BIT          deviceId
                               BOOL *          pIsReceived
                               RecMsgInfo *    rx
                             )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
Outputs	
pIsReceived	FALSE – no messages in FIFO, no message received
rx	A pointer to RecMsgInfo structure

Description

Read the next CAN message from receive FIFO to the provided RecMsgInfo structure.

3.5 sitalCan_SendMsg

```
S16BIT sitalCan_SendMsg          (
                                   U16BIT                deviceId
                                   U16BIT                protocol
                                   BOOL *                pFifoFull
                                   TxMsgInfo  *          tx
                                 )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
protocol	User Code option for sending in a protocol - Protocol_CAN or Protocol_CANFD (ISO)
tx	A pointer to TxMsgInfo structure

Outputs

pFifoFull	TRUE – transmit FIFO is full, the message is not transmitted

Description

Write the message from provided TxMsgInfo structure to transmit FIFO.

3.6 sitalCan_EnableFilters

```
S16BIT sitalCan_EnableFilters   (
                                  U16BIT      deviceId
                                  BOOL        filtersEnable
                                )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
filtersEnable	Hardware filters enable, TRUE – enable, FALSE - disable

Description

Receive hardware (IP Core) filters are enabled or disabled.
If SnS calibration or real-time processing is started - all hardware filters must be disabled.

3.7 sitalCan_SetFilter

```
S16BIT sitalCan_SetFilter      (
                                 U16BIT        deviceId
                                 U16BIT        filterIndex
                                 BOOL          extendedId
                                 canid_t       msgId
                                 U8BIT*        firstTwoDataBytes
                               )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
filterIndex	Unique Filter Index 0 - (sitalMaximum_FILTERS - 1)
extendedId	FALSE – normal CAN message ID format, TRUE – extended format
msgId	CAN message ID
firstTwoDataBytes	A pointer to first two data bytes of CAN message

Description

Set filter for received CAN message. The filter is working with associated mask (with the same *filterIndex*).

3.8 sitalCan_GetFilter

```
S16BIT sitalCan_GetFilter      (
                                U16BIT          deviceId
                                U16BIT          filterIndex
                                BOOL *          pExtendedId
                                canid_t *       pMsgId
                                U8BIT *         pFirstTwoDataBytes
                               )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
filterIndex	Unique Filter Index 0 - (sitalMaximum_FILTERS - 1)
Outputs	
pExtendedId	A pointer to extended ID: FALSE – normal CAN message ID format, TRUE – extended format
pMsgId	A pointer to CAN message ID
pFirstTwoDataBytes	A pointer to first two data bytes of CAN message

Description

Get received CAN message filter using provided *filterIndex*. The filter is working with associated mask (with the same *filterIndex*).

3.9 sitalCan_SetFilterMask

```
S16BIT sitalCan_SetFilterMask     (
                                    U16BIT          deviceId
                                    U16BIT          maskIndex
                                    BOOL            extendedId
                                    canid_t         msgId
                                    U8BIT*          firstTwoDataBytes
                                  )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
maskIndex	Unique Mask Index 0 - (sitalMaximum_FILTERS - 1)
extendedId	FALSE – normal CAN message ID format, TRUE – extended format
msgId	CAN message ID mask
firstTwoDataByte	A pointer to first two data bytes of CAN message mask

Description

Set mask for associated received CAN message filter (with the same index *maskIndex = filterIndex*). The filter bit is taken care only when appropriative mask bit is set.

3.10 sitalCan_GetFilterMask

```
S16BIT sitalCan_GetFilterMask       (
                                      U16BIT              deviceId
                                      U16BIT              maskIndex
                                      BOOL *              pExtendedId
                                      canid_t *           pMsgId
                                      U8BIT *             pFirstTwoDataByte
                                    )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
maskIdx	Unique Mask IDx 0 - (sitalMaximum_FILTERS - 1)
Outputs	
pExtendedId	A pointer to extended ID: FALSE – normal CAN message ID format, TRUE – extended format
pMsgId	A pointer to CAN message ID mask
pFirstTwoDataByte	A pointer to first two data bytes of CAN message mask

Description

Get mask using provided *maskIndex*. This mask is associated with received CAN message filter (the same Index - *maskIndex = filterIndex*). The filter bit is taken care only when appropriative mask bit is set.

3.11 sitalCan_RemoveFilter

```
S16BIT sitalCan_RemoveFilter    (
                                  U16BIT        deviceId
                                  U16BIT        filterIndex
                                )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
filterIndex	Unique Filter Index 0 - (sitalMaximum_FILTERS - 1)

Description

Receive filter is removed and no longer is filtering any CAN Message.

3.12 sitalCan_EnableSequencers

```
S16BIT sitalCan_EnableSequencers  (
                                     U16BIT          deviceId
                                     BOOL            sequncersEnable
                                  )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
sequncersEnable	Sequencers enable, TRUE – enable, FALSE - disable

Description

Transmit hardware (IP Core) sequencers are enabled or disabled.

3.13 sitalCan_SetSeqEntryMsg

```
S16BIT sitalCan_SetSeqEntryMsg    (
                                    U16BIT          deviceId
                                    U16BIT          protocol
                                    U16BIT          seqEntryIndex
                                    U16BIT          rate
                                    U16BIT          skew
                                    TxMsgInfo*      tx
                                  )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
protocol	User Code option for sending in a protocol - Protocol_CAN or Protocol_CANFD (ISO)
seqEntryIndex	Unique Sequencer Entry Index 0 - (sitalMaximum_SEQ_ETRIES - 1)
rate	A Rate value in milliseconds
skew	A Skew value in millisecond
tx	A pointer to TxMsgInfo structure

Description

Set Sequencer Entry on CAN message with rate and skew.

3.14 sitalCan_UpdateSeqEntryMsgData

```
S16BIT
sitalCan_UpdateSeqEntryMsgData   (
                                   U16BIT               deviceId
                                   U16BIT               seqEntryIndex
                                   TxMsgInfo  *         tx
                                 )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
seqEntryIndex	Unique Sequencer Entry Index 0 - (sitalMaximum_SEQ_ETRIES - 1)
tx	A pointer to TxMsgInfo structure

Description

Update Sequencer Entry CAN message with new message Data. The whole TxMsgInfo structure information is needed for check the Message Id.

3.15 sitalCan_RemoveSeqEntryMsg

```
S16BIT
sitalCan_RemoveSeqEntryMsg      (
                                  U16BIT                    deviceId
                                  U16BIT                    seqEntryIndex
                                )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
seqEntryIndex	Unique Sequencer Entry Index 0 - (sitalMaximum_SEQ_ETRIES - 1)

Description

Disable of Sequencer Entry CAN message.

4 Service Functions

4.1 sitalCan_Read

```
S16BIT sitalCan_Read                    (
                                          U16BIT          deviceId
                                          U16BIT          address
                                          U16BIT          bufferSize
                                          U8BIT*          buffer
                                        )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
address	Address of device Memory to read data from.
bufferSize	Size of buffer to read.
Outputs	
buffer	A pointer to the buffer that returns the data read.

Description

This service function returns (in the Data pointer) the data available according to the specified address related to the device.
It is advised to use this function for debug and print-outs purposes.

4.2 sitalCan_Write

```
S16BIT sitalCan_Write          (
                                 U16BIT          deviceId
                                 U16BIT          address
                                 U16BIT          bufferSize
                                 U8BIT*          buffer
                               )
```

Parameters

deviceId	Unique Device ID 0 - (sitalMaximum_DEVICES - 1)
address	Address to write data to.
bufferSize	Size of buffer to write.
Buffer	A pointer to the buffer to write.

Description

This service function writes the data in buffer to the specified address related to the device.
It is advised to use this function for debug and print-outs purposes.

5 Code Samples

5.1 Send CAN message

```
while(1)
{
        swResult = sitalCan_SendMsg(deviceId0, &FifoFull, &tx_msg);
        if (sitalReturnCode_SUCCESS != swResult)
        {
                xil_printf("\r\n  get swResult = %d \r\n", swResult);
                return swResult;
        }
        if (!FifoFull) break;
}
```

5.2 Receive CAN message

```
            while(1)
            {
                    swResult = sitalCan_GetNextMsg(deviceId0, &IsReceived,
&rx_msg);
                    if (sitalReturnCode_SUCCESS != swResult)
                    {
                            xil_printf("\r\n  get swResult = %d \r\n", swResult);
                            return swResult;
                    }
                    if (IsReceived) break;
            }
```

17 Atir Yeda St., Kfar-Saba, ISRAEL 44643

Email: info@sitaltech.com

Website: http://www.sitaltech.com