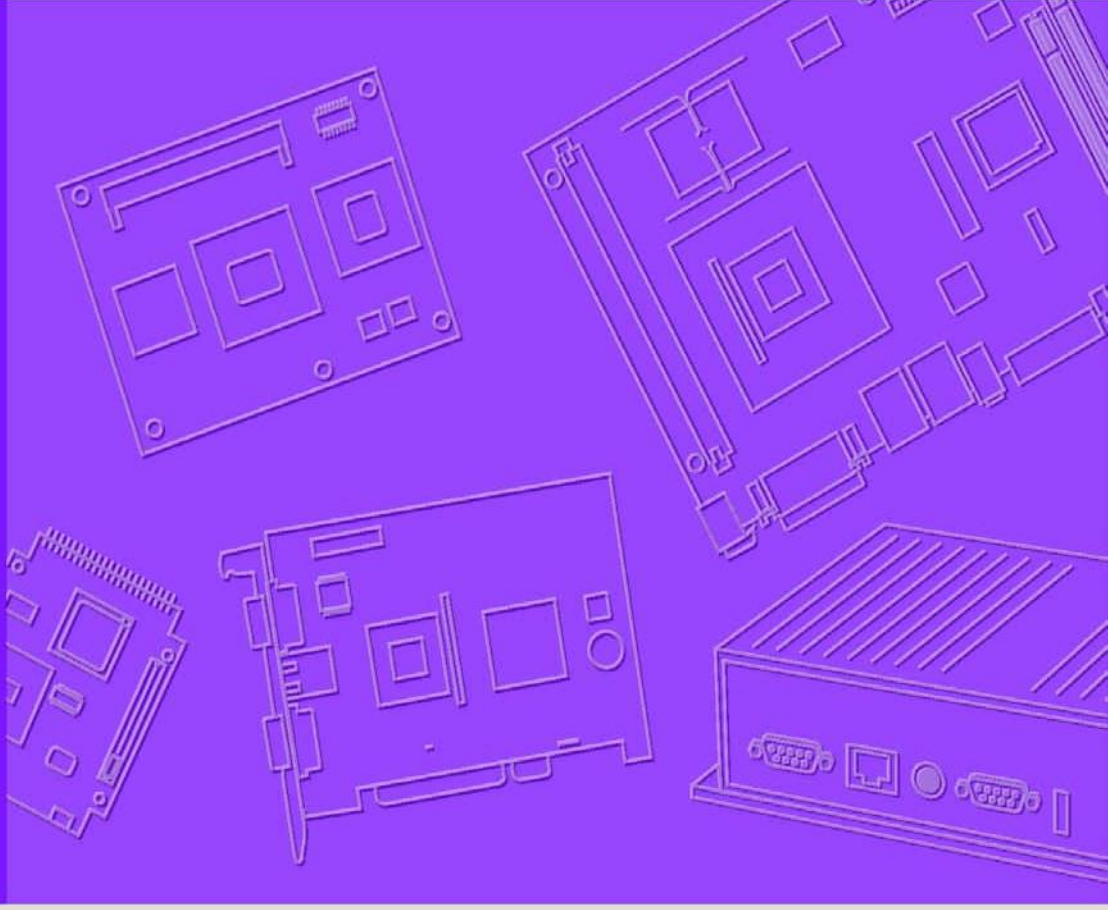

User Manual

EMCB-200U serial

EMCB-200U-MP01E
Software APIs

ADVANTECH EmbCore

Edition 1.03 Part. No. **EMCB-200U**
Nov 05 2013 Printed in Taiwan

Contents

ADVANTECH EmbCore 1

CONTENTS **3**

LIST OF TABLES **4**

1 INTRODUCTION **5**

1.1 BENEFITS 5

1.2 ENVIRONMENT REQUIREMENTS 5

1.2.1 Operating Systems *5*

2 EMCB-200U DEFINITION **6**

2.1 STATUS CODES 6

2.2 ITEM ID 8

3 EMCB-200U API **9**

3.1 INITIALIZATION FUNCTIONS 9

3.1.1 EMCBLibInitialize *9*

3.1.2 EMCBLibUninitialize *9*

3.2 INFORMATION FUNCTIONS 11

3.2.1 EMCBDeviceGetValue *11*

3.3 CAN MESSAGE FUNCTIONS 12

3.3.1 EMCBMsgTx *14*

3.3.2 EMCBMsgRx *15*

3.4 DEVICE SETTING FUNCTIONS 17

3.4.1 EMCBSetChanneValue *18*

3.4.2 EMCBGetChanneValue *20*

3.4.3 EMCBSetMask *21*

3.4.4 EMCBSetFilter *21*

3.4.5 EMCBReset *22*

List of Tables

Table 1 Device information capabilities item ID 11
Table 2 Device ID type 15
Table 3 Message type 15
Table 4 Operation of ID version. 17
Table 5 Example for operation of filter and mask (1). 17
Table 6 Example for operation of filter and mask (2). 17
Table 7 Channel capabilities item ID 18
Table 8 Terminal resister status of EMCB_IID_TERMINAL_RES 19
Table 9 Bus status definition of EMCB_IID_BUS_STATUS 19
Table 10 Baud rate of bus 19
Table 11 Bus mode of operation 19
Table 12 Rx buffer mode of operation 20

1 Introduction

1.1 Benefits

1.2 Environment Requirements

1.2.1 Operating Systems

Windows XP Embedded

Windows XP Pro or Home Edition 32-bit

Windows 7 (x86 / x64)

WES7 (x86 / x64)

Windows 8 Desktop (x86 / x64)

Others (Project based)

2 EMCB-200U Definition

EMCB.h file includes the API declaration, constants and flags that are required for programming.

2.1 Status Codes

All EMCB-200U API functions immediately return a status code from a common list of possible errors. Each function may return any of the defined status codes.

#define EMCB_STATUS_NOT_INITIALIZED 0xFFFFFFFF

Description

The EMCB-200U API library is not yet or unsuccessfully initialized. EMCBLibInitialize needs to be called prior to the first access of any other EMCB-200U API function.

Actions

Call EMCBLibInitialize.

#define EMCB_STATUS_INITIALIZED 0xFFFFFFFE

Description

Library has been initialized.

Actions

none.

#define EMCB_STATUS_INVALID_PARAMETER 0xFFFFFEFF

Description

One or more of the function call parameters are out of the defined range.

Actions

Verify Function Parameters.

#define EMCB_STATUS_UNSUPPORTED 0xFFFFFCFF

Description

This function or channel is not supported at the actual hardware environment.

Actions

none.

#define EMCB_STATUS_NOT_FOUND 0xFFFFFBFF

Description

Selected device was not found

Actions

none.

#define EMCB_STATUS_TIMEOUT 0xFFFFFBFE

Description

Device has no response.

Actions

none.

#define EMCB_STATUS_READ_ERROR 0xFFFFFAFF

Description

An error was detected during a read operation.

Actions

Retry.

#define EMCB_STATUS_WRITE_ERROR 0xFFFFFAFE

Description

An error was detected during a write operation.

Actions

Retry.

#define EMCB_STATUS_WRITE_BUSY 0xFFFFFAFD

Description

Device is busy now.

Actions

Retry.

#define EMCB_STATUS_MORE_DATA 0xFFFFF9FF

Description

The amount of available data exceeds the buffer size. Storage buffer overflow was prevented. Read count was larger than the defined buffer length.

Actions

Either increase the buffer size or reduce the block length.

#define EMCB_STATUS_MORE_DATA 0xFFFFF9FE

Description

The operation has no data.

Actions

Wait a while and try again.

#define EMCB_STATUS_ERROR 0xFFFFF0FF

Description

Generic error message. No further error details are available.

Actions

none.

#define EMCB_STATUS_SUCCESS 0

Description

The operation was successful.

Actions

none.

2.2 Item ID

#define EMCB_IID_FW_VER 0x00000000
#define EMCB_IID_LIB_VER 0x00000001
#define EMCB_IID_CHANNEL_NUM 0x00000010

Description

Use in EMCBDeviceGetValue.

#define EMCB_IID_TERMINAL_RES 0x00000000
#define EMCB_IID_BUS_STATUS 0x00010000
#define EMCB_IID_BUS_SPEED 0x00010001
#define EMCB_IID_BUS_MODE 0x00010002
#define EMCB_IID_RXB0_MODE 0x00020000
#define EMCB_IID_RXB1_MODE 0x00020001

Description

Use in EMCBDeviceGetValue or EMCBDeviceGetValue

3 EMCB-200U API

The EMCB-200U API provides the functions to control ADVANTECH platforms. EMCB-200U API functions are based on a dynamic library. EMCB-200U API can be implemented in various other programming languages.

3.1 Initialization Functions

3.1.1 EMCBLibInitialize

UINT32 EMCB_API EMCBLibInitialize(void)

Description:

This is a general initialization function of the EMCB-200U API. Prior to calling any EMCB-200U API function the library needs to be initialized by calling this function. The status code for all EMCB-200U API functions will be EMCB_STATUS_NOT_INITIALIZED unless this function called.

Parameters:

None

Return Status Code:

Condition	Return Value
Success	EMCB_STATUS_SUCCESS
Library Already initialized	EMCB_STATUS_INITIALIZED
Else	EMCB_STATUS_ERROR

3.1.2 EMCBLibUninitialize

UINT32 EMCB_API EMCBLibUninitialize(void)

Description:

This is a function to uninitialize the EMCB-200U API library that should be called before program exits. In a dynamic library environment, this function is not expected to replace the native uninitialized routines. It is expected that in this environments this function has no functionality.

Parameters:

None

Return Status Code:

Condition	Return Value
Success	EMCB_STATUS_SUCCESS
Library Uninitialized	EMCB_STATUS_NOT_INITIALIZED
Else	EMCB_STATUS_ERROR

3.2 Information Functions

3.2.1 EMCBDeviceGetValue

UINT32 EMCB_API EMCBDeviceGetValue(UINT32 ItemID, UINT32 *pVal)

Description:

Get information about the firmware or AP platform in value format.

Parameters:

ItemID

Select target of item IDs. See **Table 1**.

pVal

A pointer to the value of item's information.

Return Status Code:

Condition	Return Value
Success	EMCB_STATUS_SUCCESS
Library Uninitialized	EMCB_STATUS_NOT_INITIALIZED
Unknown Item ID	EMCB_STATUS_UNSUPPORTED
pVal == NULL	EMCB_STATUS_INVALID_PARAMETER
Get Failed	EMCB_STATUS_READ_ERROR
Get Timeout	EMCB_STATUS_TIMEOUT
Else	EMCB_STATUS_ERROR

Table 1 Device information capabilities item ID

Item ID	Description
EMCB_IID_FW_VER	FW Specification Version
EMCB_IID_LIB_VER	AP Specification Version
EMCB_IID_CHANNEL_NUM	Number of channel in device

3.3 CAN Message Functions

This function sub set is about CAN Bus message commands of transmission and reception. There are five message frames: Standard Data Frame Message and Extended Data Frame Message are known as 2.0A and 2.0B respectively. The ID length in 2.0A version is 11 bits. On the other hand, in 2.0B version, the ID length is 29 bits. See Figure 1 and Figure 2. Remote Frame Message is with the setting up value of Remote Transmission Request (RTR) bit and no data. See Figure 3. Error Frame is a data frame or remote frame adds an error frame. See Figure 4. Overload Frame can only be generated during an interframe space. See Figure 5.

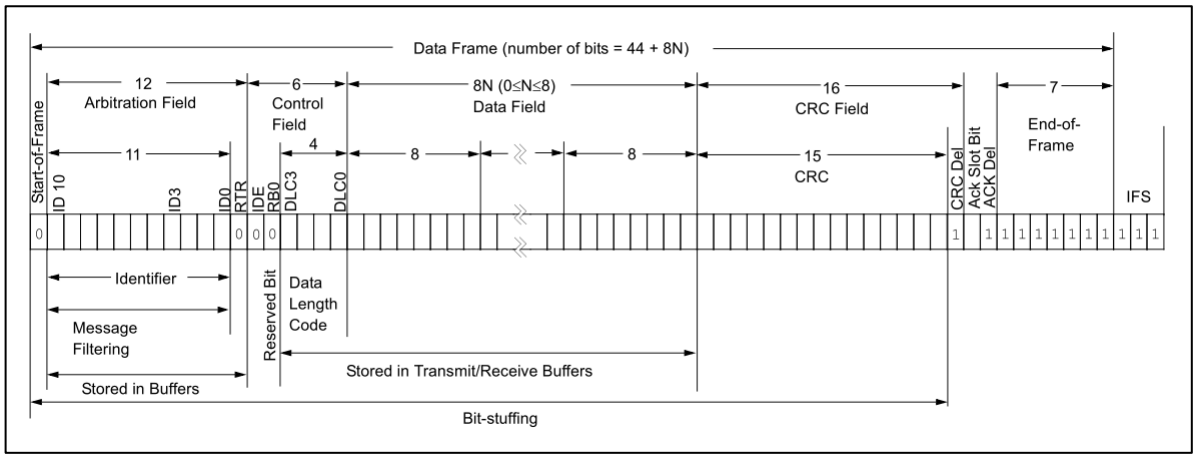

Figure 1 Standard Data Frame

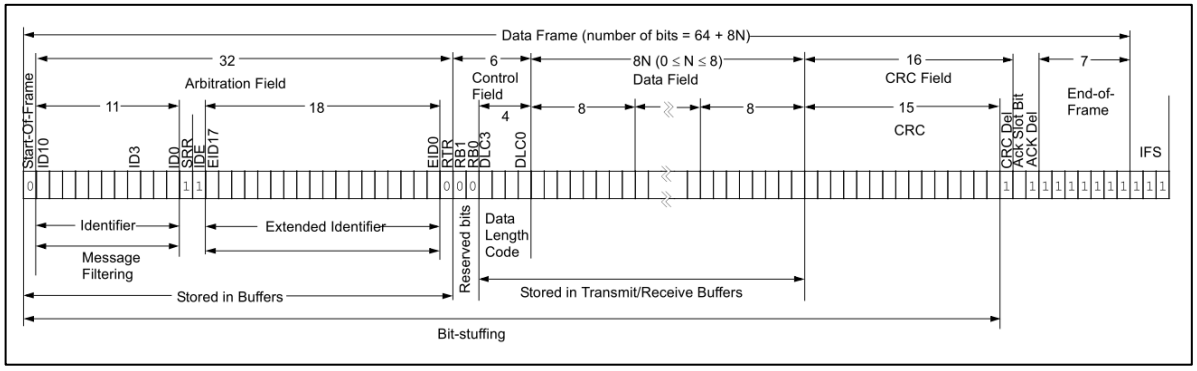

Figure 2 Extended Data Frame

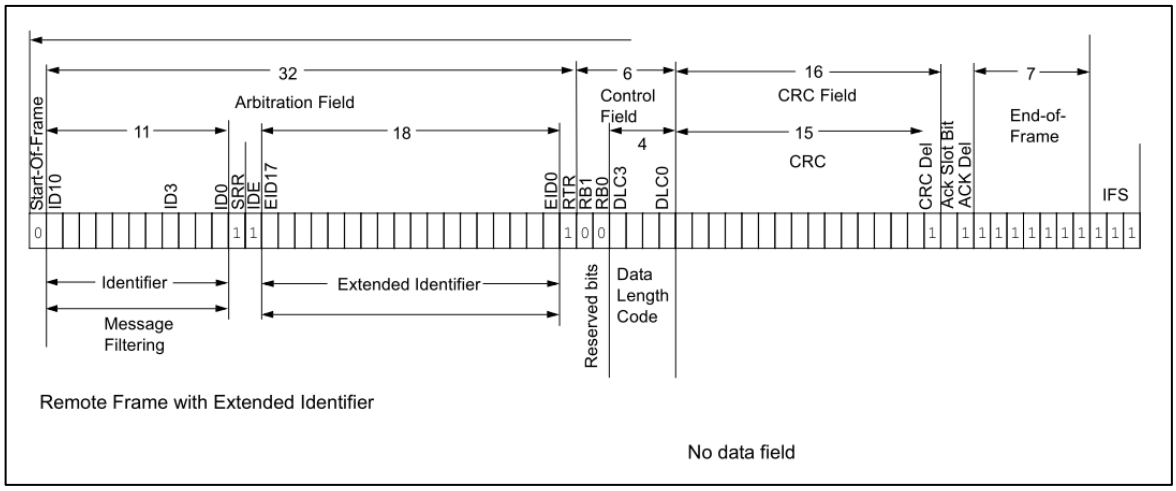

Figure 3 Remote Frame

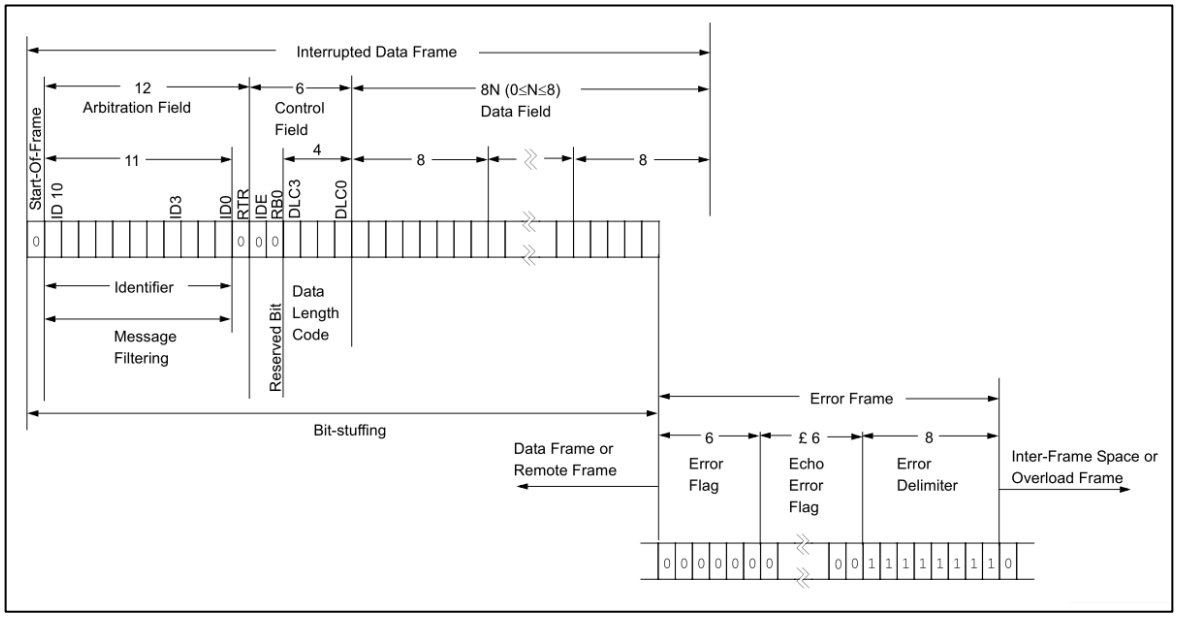

Figure 4 Error Frame

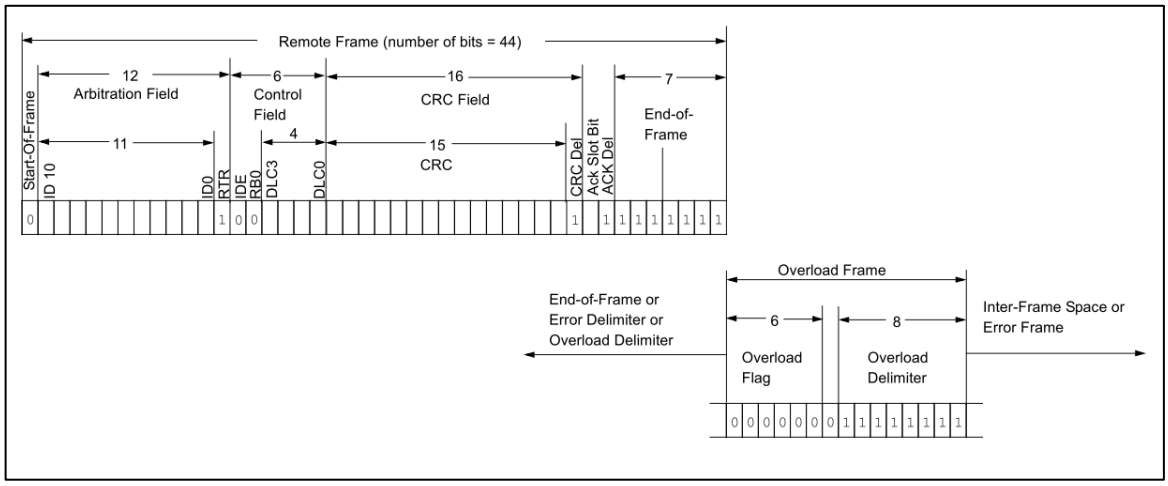

Figure 5 Overload Frame

3.3.1 EMCBMsgTx

UINT32 EMCB_API EMCBMsgTx(BYTE ch, UINT32 ID, BYTE EID, BYTE RTR, BYTE *pData, BYTE Len)

Description:

Send a CAN message with non-blocking mode, if CANBus device is busy that will return status EMCB_STATUS_TIMEOUT after 100ms. The main message to transmit is Standard Data Frame and Extended Data Frame that this function supports both 2.0A and 2.0B version. This function also supports Remote Frame.

Parameters:

ch

Select the CAN channel.

ID

Select a target device's ID that intends to send to.

EID

Select the version of the target device's ID. See Table 2.

RTR

Select this transmits message type. See Table 3.

pData

Pointer to a buffer that the data will be sent. When RTR = 1 (a Remote Frame Message) or Len = 0, pData will be ignored.

Len

The number of the byte of data length that will be sent. The maximum value is eight.

Return Status Code:

Condition	Return Value
Success	EMCB_STATUS_SUCCESS
Library Uninitialized	EMCB_STATUS_NOT_INITIALIZED
pData == NULL	EMCB_STATUS_INVALID_PARAMETER
(Len > 8) || (EID > 1) || (RTR > 1)	EMCB_STATUS_INVALID_PARAMETER
Unknown Channel	EMCB_STATUS_UNSUPPORTED
Tx Failed	EMCB_STATUS_WRITE_ERROR
Tx busy	EMCB_STATUS_WRITE_BUSY
Tx Timeout	EMCB_STATUS_TIMEOUT
Else	EMCB_STATUS_ERROR

Table 2 Device ID type

EID	Description
0	2.0A version
1	2.0B version

Table 3 Message type

RTR	Description
0	The message is a Data Frame
1	The message is a Remote Transmit Request

3.3.2 EMCBMsgRx

UINT32 EMCB_API EMCBMsgRx(BYTE ch, UINT32 *pID, BYTE *pEID, BYTE *pRTR, BYTE *pData, BYTE *pLen)

Description:

Receive a CAN message with non-blocking mode and return status immediately. This supports all frames of CAN message. The main action is to get the information of the message that be received. The information can help us to know: “from Identifier”, “type frame”, and “data”.

Parameters:

ch

Select the CAN channel.

pID

Pointer to two bytes that the ID of device which receive from.

pEID

Pointer to a byte that the ID version of device which receive from. See Table 2.

pRTR

Pointer to a byte that the received message type.

pData

Pointer to a buffer that the received data.

pLen

Pointer to a byte that the number of the received data bytes.

Return Status Code:

Condition	Return Value
Success	EMCB_STATUS_SUCCESS
Library Uninitialized	EMCB_STATUS_NOT_INITIALIZED

pID==NULL || pEID==NULL	EMCB_STATUS_INVALID_PARAMETER
pRTR==NULL	EMCB_STATUS_INVALID_PARAMETER
pData==NULL || pLen==NULL	EMCB_STATUS_INVALID_PARAMETER
*pLen < Receive length	EMCB_STATUS_MORE_DATA
No data in buffers	EMCB_STATUS_NO_DATA
Unknown Channel	EMCB_STATUS_UNSUPPORTED
RX Failed	EMCB_STATUS_READ_ERROR
RX Timeout	EMCB_STATUS_TIMEOUT
Else	EMCB_STATUS_ERROR

3.4 Device Setting Functions

These setting functions are about CAN Bus transmission and reception. These functions must be set firstly before call the function EMCBMsgTx or EMCBMsgRx. In this section, we provide two functions to set masks and filters. Masks and filters are work as following:

For ID version filtration, the versions of filter ID and mask ID must be the same, or it may result in undesirable behavior. All conditions please see Table 4. For continually checking ID operation, in general, holding two examples in 2.0A version. The first example is as Table 5, ID_Receive is the received ID, ID_Filter is filter ID, and ID_Mask is mask ID. If ID_2 = 0, then the system will receive this message. Otherwise, the system will reject this message. In the result of Table 5, the system will reject this message. In the example (2) of Table 6, which is only mask ID different with example (1), ID_2 is equal to zero, so the system will receive this message.

Table 4 Operation of ID version.

Received ID	Filter ID	Mask ID	Result
2.0A version	2.0A version	2.0A version	checking ID
2.0A version	2.0B version	2.0B version	rejection
2.0B version	2.0A version	2.0A version	rejection
2.0B version	2.0B version	2.0B version	checking ID
Don't care	2.0A version	2.0B version	undesirable behavior
Don't care	2.0B version	2.0A version	undesirable behavior

Table 5 Example for operation of filter and mask (1).

Value of ID (Bin)	Description
00110110010	ID_Receive, EID = 0
00110111111	ID_Filter, EID = 0
11111111111	ID_Mask, EID = 0
00000001101	ID_1 = ID_Receive XOR ID_Filter
00000001101	ID_2 = ID_1 AND ID_Mask1

Table 6 Example for operation of filter and mask (2).

Value of ID (Bin)	Description
00110110010	ID_Receive, EID = 0
00110111111	ID_Filter, EID = 0
11111110000	ID_Mask, EID = 0
00000001101	ID_1 = ID_Receive XOR ID_Filter
00000000000	ID_2 = ID_1 AND ID_Mask1

3.4.1 EMCBSetChanneValue

UINT32 EMCB_API EMCBSetChannel(BYTE ch, UINT32 ItemID, UINT32 Val)

Description:

Set CAN Bus values select by ItemID. For example, speed of bus, operation mode …etc.

Parameters:

ch

Select the CAN channel.

ItemID

Select target of item IDs. See **Table 7**.

Val

Select the CAN channel.

Return Status Code:

Condition	Return Value
Success	EMCB_STATUS_SUCCESS
Library Uninitialized	EMCB_STATUS_NOT_INITIALIZED
Invalid Value	EMCB_STATUS_INVALID_PARAMETER
Unknown ItemID	EMCB_STATUS_UNSUPPORTED
Unknown Channel	EMCB_STATUS_UNSUPPORTED
Set Failed	EMCB_STATUS_WRITE_ERROR
Set Timeout	EMCB_STATUS_TIMEOUT
Else	EMCB_STATUS_ERROR

Table 7 Channel capabilities item ID

Item ID	Description
EMCB_IID_TERMINAL_RES	Enable or disable 120 ohm terminal resistor, see Table 8.
EMCB_IID_BUS_STATUS	Bus status, see Table 9.
EMCB_IID_BUS_SPEED	Bus baud rate, see Table 10.
EMCB_IID_BUS_MODE	Bus mode of operation, see Table 11
EMCB_IID_RXB0_MODE	Rx Buffer 0 mode of operation, see Table 12.
EMCB_IID_RXB1_MODE	Rx Buffer 1 mode of operation, see Table 12.

Table 8 Terminal resister status of EMCB_IID_TERMINAL_RES

Status	Description	Value
EMCB_TR_ENALBE	Enable 120 ohm resistor	1
EMCB_TR_DISABLE	Disable 120 ohm resistor	0

Table 9 Bus status definition of EMCB_IID_BUS_STATUS

Status Flags	Description	Value
EMCB_BUS_STATUS_NORMAL	No Error	0
EMCB_BUS_STATUS_RX1OVR	Receive Buffer 1 Overflow	(1 << 7)
EMCB_BUS_STATUS_RX0OVR	Receive Buffer 0 Overflow	(1 << 6)
EMCB_BUS_STATUS_TXBO	Bus-Off error	(1 << 5)
EMCB_BUS_STATUS_TXEP	Transmit Error-Passive	(1 << 4)
EMCB_BUS_STATUS_RXEP	Receive Error-Passive	(1 << 3)
EMCB_BUS_STATUS_TXWAR	Transmit Error-Warning	(1 << 2)
EMCB_BUS_STATUS_RXWAR	Receive Error-Warning	(1 << 1)
EMCB_BUS_STATUS_EWARN	Error-Warning	(1 << 0)

Table 10 Baud rate of bus

Value (Speed)	Description
1000	Baud rate is 1 Mbps
800	Baud rate is 800 kbps
500	Baud rate is 500 kbps
250	Baud rate is 250 kbps
125	Baud rate is 125 kbps
50	Baud rate is 50 kbps
20	Baud rate is 20 kbps
10	Baud rate is 10 kbps
else	Invalid Value of Speed

Table 11 Bus mode of operation

Mode	Description	Value
EMCB_MODE_NORMAL	Normal mode	0
EMCB_MODE_LISTENONLY	Listen-Only mode	3

Table 12 Rx buffer mode of operation

Mode	Description	Value
EMCB_BUF_MODE_FILTER_MASK_OFF	Turn mask/filters off; receive any message	3
EMCB_BUF_MODE_EX_ONLY	Receive only valid message with extended identifiers that meet filter criteria.	2
EMCB_BUF_MODE_STD_ONLY	Receive only valid message with standard identifiers that meet filter criteria.	1
EMCB_BUF_MODE_BOTH	Receive all valid message using either standard or extended identifiers that meet filter criteria.	0

3.4.2 EMCBGetChanneValue

UINT32 EMCB_API EMCBGetChannel(BYTE ch, UINT32 ItemID, UINT32 *pVal)

Description:

Get CAN Bus values select by ItemID.

Parameters:

ch

Select the CAN channel.

ItemID

Select target of item IDs. See **Table 7**.

pVal

A pointer to get bus values or capabilities.

Return Status Code:

Condition	Return Value
Success	EMCB_STATUS_SUCCESS
Library Uninitialized	EMCB_STATUS_NOT_INITIALIZED
pVal==NULL	EMCB_STATUS_INVALID_PARAMETER
Unknown ItemID	EMCB_STATUS_UNSUPPORTED
Unknown Channel	EMCB_STATUS_UNSUPPORTED
Get Failed	EMCB_STATUS_READ_ERROR
Get Timeout	EMCB_STATUS_TIMEOUT
Else	EMCB_STATUS_ERROR

3.4.3 EMCBSetMask

UINT32 EMCB_API EMCBSetMask(BYTE ch, BYTE Mask, UINT32 ID, BYTE EID)

Description:

Set a CAN mask. Work with CAN filters (Please see EMCBSetFilter).

Parameters:

ch

Select the CAN channel.

Mask

Select an index of masks. There are two RX buffer in chip. Mask = 0 is for RX0 buffer, and Mask = 1 is for RX1 buffer.

ID

Pointer to a buffer that mask value.

EID

Pointer to a byte that the ID version of mask. See Table 2.

Return Status Code:

Condition	Return Value
Success	EMCB_STATUS_SUCCESS
Library Uninitialized	EMCB_STATUS_NOT_INITIALIZED
Invalid EID || Mask	EMCB_STATUS_INVALID_PARAMETER
Unknown Channel	EMCB_STATUS_UNSUPPORTED
Set Failed	EMCB_STATUS_WRITE_ERROR
Set Timeout	EMCB_STATUS_TIMEOUT
Else	EMCB_STATUS_ERROR

3.4.4 EMCBSetFilter

UINT32 EMCB_API EMCBSetFilter(BYTE ch, BYTE Filter, UINT32 ID, BYTE EID)

Description:

Set CAN filter. Work with CAN masks (Please see EMCBSetMask).

Parameters:

ch

Select the CAN channel.

Filter

Select an index of filters. There are two RX buffer in chip. Filter = 0 or 1 is for RX0 buffer, work with Mask 0. Filter is 2 to 5 are for RX1 buffer, work with Mask 1.

ID

Pointer to a buffer that the filter value.

EID

Pointer to a byte that the ID version of filter. See Table 2.

Return Status Code:

Condition	Return Value
Success	EMCB_STATUS_SUCCESS
Library Uninitialized	EMCB_STATUS_NOT_INITIALIZED
Invalid EID || Filter	EMCB_STATUS_INVALID_PARAMETER
Unknown Channel	EMCB_STATUS_UNSUPPORTED
Set Failed	EMCB_STATUS_WRITE_ERROR
Set Timeout	EMCB_STATUS_TIMEOUT
Else	EMCB_STATUS_ERROR

3.4.5 EMCBReset

UINT32 EMCB_API EMCBReset(BYTE ch)

Description:

Reset CANBus device.

Parameters:

ch

Select the CAN channel.

Return Status Code:

Condition	Return Value
Success	EMCB_STATUS_SUCCESS
Library Uninitialized	EMCB_STATUS_NOT_INITIALIZED
Unknown Channel	EMCB_STATUS_UNSUPPORTED
Reset Failed	EMCB_STATUS_WRITE_ERROR
Reset Timeout	EMCB_STATUS_TIMEOUT
Else	EMCB_STATUS_ERROR