

MCU

Generated by Doxygen 1.10.0

1 Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

[pBCCIM_Interface](#) ??

2 File Index

2.1 File List

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

Firmware/Source/Controller/[Controller.h](#)
Файл [Controller.h](#) ??

Firmware/Source/Platform/[DeviceObjectDictionary.h](#)
Device object dictionary ??

/home/pumukun/Avocado_Test/hw-STM32-lib/Interface/[BCCIMaster.c](#)
Библиотека интерфейса Master BCCI ??

/home/pumukun/Avocado_Test/hw-STM32-lib/Interface/[BCCIMaster.h](#)
Библиотека интерфейса Master BCCI ??

3 Data Structure Documentation

3.1 pBCCIM_Interface Struct Reference

```
#include <BCCIMaster.h>
```

Data Fields

- `pBCCI_IOConfig IOConfig`
Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
- `Int32U TimeoutValueTicks`
Величина таймаута при обмене сообщениями по CAN, в миллисекундах
- `volatile Int64U * pTimerCounter`
Указатель на системный таймер, в миллисекундах

3.1.1 Detailed Description

BCCI instance state

The documentation for this struct was generated from the following file:

- /home/pumukun/Avocado_Test/hw-STM32-lib/Interface/[BCCIMaster.h](#)

4 File Documentation

4.1 Firmware/Source/Controller/Controller.h File Reference

Файл [Controller.h](#).

```
#include "stdinc.h"
```

Functions

- void CONTROL_Init ()
Функция инициализации

4.1.1 Detailed Description

Файл [Controller.h](#).

Детали:

4.2 Firmware/Source/Platform/DeviceObjectDictionary.h File Reference

Device object dictionary.

Macros

- #define ACT_DIAG_INT_FAN 10
Проверка вентилятора
- #define ACT_DIAG_GREEN_LED 11
Проверка зеленого индикатора
- #define ACT_DIAG_RED_LED 12
Проверка красного индикатора
- #define ACT_DIAG_PC_SWITCH 13
Включение ПК
- #define ACT_SAVE_TO_ROM 200
Сохранение пользовательских данных во FLASH процессора
- #define ACT_RESTORE_FROM_ROM 201
Восстановление данных из FLASH.
- #define ACT_RESET_TO_DEFAULT 202
Сброс DataTable в состояние по умолчанию
- #define ACT_BOOT_LOADER_REQUEST 320
Request reboot to bootloader.
- #define REG_LAMP_GREEN 128
Управление зелёным индикатором
- #define REG_LAMP_RED 129
Управление красным индикатором
- #define REG_COMPATIBILITY_1 130
(Не используется, для совместимости)

- `#define REG_INT_FAN 131`
Управление вентилятором
- `#define REG_DEV_STATE 192`
Device state (Не используется, для совместимости)
- `#define REG_FAULT_REASON 193`
Fault reason in the case DeviceState -> FAULT (Не используется, для совместимости)
- `#define REG_DISABLE_REASON 194`
Fault reason in the case DeviceState -> DISABLED (Не используется, для совместимости)
- `#define REG_WARNING 195`
Warning if present (Не используется, для совместимости)
- `#define REG_SENSOR_2 197`
(Не используется, для совместимости)
- `#define REG_SENSOR_3 198`
(Не используется, для совместимости)
- `#define REG_SENSOR_4 199`
(Не используется, для совместимости)
- `#define REG_EXT_BUTTON 200`
Состояние внешней кнопки
- `#define REG_OVERLAP_COUNT_REQ 210`
Overlapping requests via SCCI interface.
- `#define REG_OVERLAP_COUNT_RESP 211`
Overlapping responses via SCCI interface.
- `#define REG_MME_CODE 250`
MME code number.
- `#define REG_FWINFO_SLAVE_NID 256`
Device CAN slave node ID.
- `#define REG_FWINFO_MASTER_NID 257`
Device CAN master node ID (if presented)
- `#define REG_FWINFO_STR_LEN 260`
Length of the information string record.
- `#define REG_FWINFO_STR_BEGIN 261`
Beginning of the information string record.

4.2.1 Detailed Description

Device object dictionary.

4.3 /home/pumukun/Avocado_Test/hw-STM32-libs/Interface/BCCIMaster.c File Reference

Библиотека интерфейса Master BCCI.

```
#include "BCCIMaster.h"
#include "BCCIxParams.h"
#include "CRC16.h"
#include "SysConfig.h"
#include "DeviceProfile.h"
#include "ZwIWDG.h"
```

Functions

- void [BCCIM_SendFrame](#) (pBCCIM_Interface Interface, Int16U Mailbox, pCANMessage Message, Int32U Node)
Отправка CAN сообщения.
- Int16U [BCCIM_WaitResponse](#) (pBCCIM_Interface Interface, Int16U Mailbox)
Ожидание ответа от узла, полученного по мейлбоксу Mailbox.
- Boolean [BCCIM_HandleReadBlock16](#) (pBCCIM_Interface Interface)
Обработка получения блока данных из CAN сети.
- Boolean [BCCIM_HandleReadBlockFloat](#) (pBCCIM_Interface Interface)
Обработка получения блока(массива) данных с значениями типа float из CAN сети.
- void [BCCIM_Init](#) (pBCCIM_Interface Interface, pBCCI_IOConfig IOConfig, Int32U Message←TimeoutTicks, volatile Int64U *pTimer)
Инициализация Master BCCI интерфейса - сетап Mailbox для записи и чтения, запись Mailbox в IOConfig.
- void [BCCIM_InitWithNodeID](#) (pBCCIM_Interface Interface, pBCCI_IOConfig IOConfig, Int32U MessageTimeoutTicks, volatile Int64U *pTimer, Int16U NodeID)
Инициализация Master BCCI интерфейса для конкретного узла(NodeID)
- Int16U [BCCIM_Read16](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Register, pInt16U Data)
Чтение 16-битного значения из регистра узла в CAN сети.
- Int16U [BCCIM_ReadFloat](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Register, float *Data)
Чтение 32-битного float значения из регистра узла в CAN сети.
- Int16U [BCCIM_ReadLimitFloat](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Register, Boolean ReadHighLimit, float *Data)
Чтение 32-битного float значения из регистра узла в CAN сети.
- Int16U [BCCIM_Write16](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Register, Int16U Data)
Запись 16-битного значения в регистр узла в CAN сети.
- Int16U [BCCIM_WriteFloat](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Register, float Data)
Запись 32-битного float значения в регистр узла в CAN сети.
- Int16U [BCCIM_Call](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Action)
Вызов функции узла в CAN сети.
- Int16U [BCCIM_ReadBlock16](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Endpoint)
Чтение блока(массива) данных из узла в CAN сети.
- Int16U [BCCIM_WriteBlock16](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Endpoint, p←Int16U Data, Int16U DataLength)
Запись блока(массива) данных в узел в CAN сети.
- Int16U [BCCIM_ReadBlockFloat](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Endpoint)
Чтение блока(массива) данных со значениями типа данных float из регистра узла в CAN сети.

4.3.1 Detailed Description

Библиотека интерфейса Master BCCI.

4.3.2 Function Documentation

BCCIM_Call()

```
Int16U BCCIM_Call (  
    pBCCIM_Interface Interface,  
    Int16U Node,  
    Int16U Action )
```

Вызов функции узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Action	- ID вызываемой функции

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймута.

BCCIM_HandleReadBlock16()

```
Boolean BCCIM_HandleReadBlock16 (  
    pBCCIM_Interface Interface )
```

Обработка получения блока данных из CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
-----------	---

Returns

TRUE, если блок данных получен, FALSE в противном случае.

BCCIM_HandleReadBlockFloat()

```
Boolean BCCIM_HandleReadBlockFloat (  
    pBCCIM_Interface Interface )
```

Обработка получения блока(массива) данных с значениями типа float из CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
-----------	---

Returns

TRUE, если блок данных получен, FALSE в противном случае.

BCCIM_Init()

```
void BCCIM_Init (  
    pBCCIM_Interface Interface,
```

```
pBCCI_IOConfig IOConfig,
Int32U MessageTimeoutTicks,
volatile Int64U * pTimer )
```

Инициализация Master BCCI интерфейса - сетеп Mailbox для записи и чтения, запись Mailbox в IOConfig.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
IOConfig	- Указатель на структуру, содержащую функции обратного вызова. Эти функции передаются в интерфейс.
MessageTimeoutTicks	- Величина таймаута при обмене сообщениями по CAN, в миллисекундах
pTimer	- Указатель на системный таймер, в миллисекундах

BCCIM_InitWithNodeID()

```
void BCCIM_InitWithNodeID (
    pBCCIM_Interface Interface,
    pBCCI_IOConfig IOConfig,
    Int32U MessageTimeoutTicks,
    volatile Int64U * pTimer,
    Int16U NodeID )
```

Инициализация Master BCCI интерфейса для конкретного узла(NodeID)

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
IOConfig	- Указатель на структуру, содержащую функции обратного вызова. Эти функции передаются в интерфейс.
MessageTimeoutTicks	- Величина таймаута при обмене сообщениями по CAN, в миллисекундах
pTimer	- Указатель на системный таймер, в миллисекундах
NodeID	- ID узла в CAN сети

BCCIM_Read16()

```
Int16U BCCIM_Read16 (
    pBCCIM_Interface Interface,
    Int16U Node,
    Int16U Register,
    pInt16U Data )
```

Чтение 16-битного значения из регистра узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Register	- Регистр блока из которого будет читаться значение
Data	- Указатель на буфер, в который будет записан результат

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_ReadBlock16()

```
Int16U BCCIM_ReadBlock16 (  
    pBCCIM_Interface Interface,  
    Int16U Node,  
    Int16U Endpoint )
```

Чтение блока(массива) данных из узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Endpoint	- Регистр начала массива

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_ReadBlockFloat()

```
Int16U BCCIM_ReadBlockFloat (  
    pBCCIM_Interface Interface,  
    Int16U Node,  
    Int16U Endpoint )
```

Чтение блока(массива) данных со значениями типа данных float из регистра узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Endpoint	- Регистр начала массива

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_ReadFloat()

```
Int16U BCCIM_ReadFloat (  
    pBCCIM_Interface Interface,  
    Int16U Node,  
    Int16U Register,  
    float * Data )
```

Чтение 32-битного float значения из регистра узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Register	- Регистр блока из которого будет читаться значение
Data	- Указатель на буфер, в котором будет записан результат

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_ReadLimitFloat()

```
BCCIM_ReadLimitFloat (  
    pBCCIM_Interface Interface,  
    Int16U Node,  
    Int16U Register,  
    Boolean ReadHighLimit,  
    float * Data )
```

Чтение 32-битного float значения из регистра узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Register	- Регистр блока из которого будет читаться значение
ReadHighLimit	- Если true - читается верхняя граница, если false - нижняя
Data	- Указатель на буфер, в котором будет записан результат

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_SendFrame()

```
void BCCIM_SendFrame (  
    pBCCIM_Interface Interface,  
    Int16U Mailbox,  
    pCANMessage Message,  
    Int32U Node )
```

Отправка CAN сообщения.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Mailbox	- ID мейлбокса внутри узла в который будет отправлено сообщение.
Message	- Указатель на структуру CAN сообщения.
Node	- ID узла в CAN сети.

BCCIM_WaitResponse()

```
Int16U BCCIM_WaitResponse (  
    pBCCIM_Interface Interface,  
    Int16U Mailbox )
```

Ожидание ответа от узла, полученного по мейлбоксу Mailbox.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Mailbox	- ID мейлбокса внутри узла, с которым собираемся общаться, в который будет отправлено сообщение.

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймута.

Return values

ERR_NO_ERROR	- Ответ получен.
ERR_TIMEOUT	- Тайм-аут

BCCIM_Write16()

```
Int16U BCCIM_Write16 (  
    pBCCIM_Interface Interface,  
    Int16U Node,
```

Int16U Register,
Int16U Data)

Запись 16-битного значения в регистр узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Register	- Регистр блока в который будет записано значение
Data	- Значение, которое будет записано в регистр

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймута.

BCCIM_WriteBlock16()

```
Int16U BCCIM_WriteBlock16 (
    pBCCIM_Interface Interface,
    Int16U Node,
    Int16U Endpoint,
    pInt16U Data,
    Int16U DataLength )
```

Запись блока(массива) данных в узел в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Endpoint	- Регистр начала массива
Data	- Указатель на массив, который будет записан в узел
DataLength	- Размер записываемого массива

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймута.

BCCIM_WriteFloat()

```
Int16U BCCIM_WriteFloat (
    pBCCIM_Interface Interface,
    Int16U Node,
    Int16U Register,
    float Data )
```

Запись 32-битного float значения в регистр узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Register	- Регистр блока в котором будет записано значение
Data	- Значение, которое будет записано в регистр

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

4.4 /home/pumukun/Avocado_Test/hw-STM32-libs/Interface/BCCIMaster.h File Reference

Библиотека интерфейса Master BCCI.

```
#include "stdinc.h"
#include "BCCITypes.h"
```

Data Structures

- struct [pBCCIM_Interface](#)

Functions

- void [BCCIM_Init](#) (pBCCIM_Interface Interface, pBCCI_IOConfig IOConfig, Int32U Message← TimeoutTicks, volatile Int64U *pTimer)
Инициализация Master BCCI интерфейса - сетеп Mailbox для записи и чтения, запись Mailbox в IOConfig.
- void [BCCIM_InitWithNodeID](#) (pBCCIM_Interface Interface, pBCCI_IOConfig IOConfig, Int32U MessageTimeoutTicks, volatile Int64U *pTimer, Int16U NodeID)
Инициализация Master BCCI интерфейса для конкретного узла(NodeID)
- Int16U [BCCIM_Read16](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Register, pInt16U Data)
Чтение 16-битного значения из регистра узла в CAN сети.
- Int16U [BCCIM_ReadFloat](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Register, float *Data)
Чтение 32-битного float значения из регистра узла в CAN сети.
- Int16U [BCCIM_ReadLimitFloat](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Register, Boolean ReadHighLimit, float *Data)
Чтение 32-битного float значения из регистра узла в CAN сети.
- Int16U [BCCIM_Write16](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Register, Int16U Data)
Запись 16-битного значения в регистр узла в CAN сети.
- Int16U [BCCIM_WriteFloat](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Register, float Data)
Запись 32-битного float значения в регистр узла в CAN сети.
- Int16U [BCCIM_Call](#) (pBCCIM_Interface Interface, Int16U Node, Int16U Action)

Вызов функции узла в CAN сети.

- `Int16U BCCIM_ReadBlock16` (`pBCCIM_Interface Interface`, `Int16U Node`, `Int16U Endpoint`)
Чтение блока(массива) данных из узла в CAN сети.
- `Int16U BCCIM_WriteBlock16` (`pBCCIM_Interface Interface`, `Int16U Node`, `Int16U Endpoint`, `pInt16U Data`, `Int16U DataLength`)
Запись блока(массива) данных в узел в CAN сети.
- `Int16U BCCIM_ReadBlockFloat` (`pBCCIM_Interface Interface`, `Int16U Node`, `Int16U Endpoint`)
Чтение блока(массива) данных со значениями типа данных float из регистра узла в CAN сети.
- `Int16U BCCIM_WriteBlockFloat` (`pBCCIM_Interface Interface`, `Int16U Node`, `Int16U Endpoint`, `pFloat Data`, `Int16U DataLength`)
Запись блока(массива) данных со значениями типа данных float в регистр узла в CAN сети.

4.4.1 Detailed Description

Библиотека интерфейса Master BCCI.

4.4.2 Function Documentation

BCCIM_Call()

```
Int16U BCCIM_Call (
    pBCCIM_Interface Interface,
    Int16U Node,
    Int16U Action )
```

Вызов функции узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Action	- ID вызываемой функции

Returns

`ERR_NO_ERROR` или `ERR_TIMEOUT`, если произошла ошибка таймаута.

BCCIM_Init()

```
void BCCIM_Init (
    pBCCIM_Interface Interface,
    pBCCI_IOConfig IOConfig,
    Int32U MessageTimeoutTicks,
    volatile Int64U * pTimer )
```

Инициализация Master BCCI интерфейса - сетап Mailbox для записи и чтения, запись Mailbox в IOConfig.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
IOConfig	- Указатель на структуру, содержащую функции обратного вызова. Эти функции передаются в интерфейс.
MessageTimeoutTicks	- Величина таймаута при обмене сообщениями по CAN, в миллисекундах
pTimer	- Указатель на системный таймер, в миллисекундах

BCCIM_InitWithNodeID()

```
void BCCIM_InitWithNodeID (
    pBCCIM_Interface Interface,
    pBCCI_IOConfig IOConfig,
    Int32U MessageTimeoutTicks,
    volatile Int64U * pTimer,
    Int16U NodeID )
```

Инициализация Master BCCI интерфейса для конкретного узла(NodeID)

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
IOConfig	- Указатель на структуру, содержащую функции обратного вызова. Эти функции передаются в интерфейс.
MessageTimeoutTicks	- Величина таймаута при обмене сообщениями по CAN, в миллисекундах
pTimer	- Указатель на системный таймер, в миллисекундах
NodeID	- ID узла в CAN сети

BCCIM_Read16()

```
Int16U BCCIM_Read16 (
    pBCCIM_Interface Interface,
    Int16U Node,
    Int16U Register,
    pInt16U Data )
```

Чтение 16-битного значения из регистра узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Register	- Регистр блока из которого будет читаться значение
Data	- Указатель на буфер, в который будет записан результат

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_ReadBlock16()

```
Int16U BCCIM_ReadBlock16 (  
    pBCCIM_Interface Interface,  
    Int16U Node,  
    Int16U Endpoint )
```

Чтение блока(массива) данных из узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Endpoint	- Регистр начала массива

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_ReadBlockFloat()

```
Int16U BCCIM_ReadBlockFloat (  
    pBCCIM_Interface Interface,  
    Int16U Node,  
    Int16U Endpoint )
```

Чтение блока(массива) данных со значениями типа данных float из регистра узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Endpoint	- Регистр начала массива

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_ReadFloat()

```
Int16U BCCIM_ReadFloat (  
    pBCCIM_Interface Interface,
```



```

Int16U Node,
Int16U Register,
float * Data )

```

Чтение 32-битного float значения из регистра узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Register	- Регистр блока из которого будет читаться значение
Data	- Указатель на буфер, в котором будет записан результат

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_ReadLimitFloat()

```

Int16U BCCIM_ReadLimitFloat (
    pBCCIM_Interface Interface,
    Int16U Node,
    Int16U Register,
    Boolean ReadHighLimit,
    float * Data )

```

Чтение 32-битного float значения из регистра узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Register	- Регистр блока из которого будет читаться значение
ReadHighLimit	- Если true - читается верхняя граница, если false - нижняя
Data	- Указатель на буфер, в котором будет записан результат

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_Write16()

```

Int16U BCCIM_Write16 (
    pBCCIM_Interface Interface,
    Int16U Node,
    Int16U Register,
    Int16U Data )

```

Запись 16-битного значения в регистр узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Register	- Регистр блока в который будет записано значение
Data	- Значение, которое будет записано в регистр

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_WriteBlock16()

```
Int16U BCCIM_WriteBlock16 (  
    pBCCIM_Interface Interface,  
    Int16U Node,  
    Int16U Endpoint,  
    pInt16U Data,  
    Int16U DataLength )
```

Запись блока(массива) данных в узел в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Endpoint	- Регистр начала массива
Data	- Указатель на массив, который будет записан в узел
DataLength	- Размер записываемого массива

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_WriteBlockFloat()

```
Int16U BCCIM_WriteBlockFloat (  
    pBCCIM_Interface Interface,  
    Int16U Node,  
    Int16U Endpoint,  
    pFloat Data,  
    Int16U DataLength )
```

Запись блока(массива) данных со значениями типа данных float в регистр узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Endpoint	- Регистр начала массива
Data	- Указатель на массив, который будет записан в узел
DataLength	- Размер записываемого массива

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.

BCCIM_WriteFloat()

```
Int16U BCCIM_WriteFloat (  
    pBCCIM_Interface Interface,  
    Int16U Node,  
    Int16U Register,  
    float Data )
```

Запись 32-битного float значения в регистр узла в CAN сети.

Parameters

Interface	- Указатель на структуру, хранящую параметры CAN-интерфейса (таймаут и функции обратного вызова).
Node	- ID узла в CAN сети
Register	- Регистр блока в котором будет записано значение
Data	- Значение, которое будет записано в регистр

Returns

ERR_NO_ERROR или ERR_TIMEOUT, если произошла ошибка таймаута.