GOAL V2.21.1

CC-Link IE Field Basic
Reference Manual
port GmbH

port GmbH
Regensbuger Str. 7
D-06132 Halle/Saale

Disclaimer

This manual represents the current state of the product. Please check with port.de for the latest version as the document may have a newer version since errors may be corrected or changes for a newer version of the product may be incorporated. Port.de assumes no responsibility for errors in this document. Qualified feedback is appreciated at service@port.de.

This document is the Intellectual Property of port.de and is intended to be used with the described product only. It may be forwarded and/or copied in the original and unmodified format. All rights reserved.

The product enables to use technologies such as PROFINET, EtherNet/IP and/or EtherCAT and others. These technologies are promoted by trade organizations, such as PNO (profibus.org), ODVA (odva.org) or ETG (ethercat.org). These trade organizations as well maintain the specification and care about legal issues. We strongly recommend to become a member of these organisations. Most technologies are making use of patented or otherwise copyrighted technologies, approaches or other intellectual property. The membership usually automatically entitles the member for use of most of the technology-inherent copyrighted or otherwise protected Intellectual Property of the corresponding trade organization and most 3rd parties. Otherwise the user will need to obtain licenses for many patented technologies separately.

Further we suggest to you to subscribe to the corresponding Conformance Test Tool of these trade organizations. For instance the ODVA only accepts conformance test applications from companies who have a valid membership and have a valid subscription to the recent Conformance Test Tool. We as port are members in all corresponding organizations and are holding a subscription to these tools - however you as a customer need to have an own membership and an own subscription to the tool.

All rights reserved

The programs, boards and documentations supplied by port GmbH are created with due diligence, checked carefully and tested on several applications. Nevertheless, port GmbH cannot guarantee and nor assume liability that the program, the hardware board or the documentation are error-free or appropriate to serve a specific customer purpose. In particular performance characteristics and technical data given in this document may not be interpreted to be guaranteed product features in any legal sense.

For consequential damages, every legal responsibility or liability is excluded. port has the right to modify the products described or their documentation at any time without prior warning, as long as these changes are made for reasons of reliability or technical improvement. All rights of this documentation are with port. Unless expressively granted - the transfer of rights to third parties or duplication of this document in any form, whole or in part, is subject to written approval by port. Copies of this document may however be made exclusively for the use of the user and his engineers. The user is thereby responsible that third parties do not obtain access to these copies. The soft- and hardware designations used are mostly registered and are subject to copyright.

Copyright
© 2020port GmbH
Regensburger Straße 7
D-06132 Halle
Tel. +49 345 - 777 55 0
Fax. +49 345 - 777 55 20
E-Mail service@port.de
www.port.de
www.port-automation.com

Contents

1 GOAL CC-Link IE Field Basic Reference Manual 4

1.1 Introduction 4

2 Data Structure Documentation 4

2.1 CCLIEFB_CFG_T Struct Reference 4

2.1.1 Detailed Description 4

2.2 CCLIEFB_CORE_T Struct Reference 4

2.2.1 Detailed Description 5

2.3 CCLIEFB_CYCDATAREQ_T Struct Reference 5

2.3.1 Detailed Description 6

2.4 CCLIEFB_INSTANCE_T Struct Reference 6

2.4.1 Detailed Description 6

2.5 CCLIEFB_NET_T Struct Reference 6

2.5.1 Detailed Description 7

2.6 CCLIEFB_PDM_T Struct Reference 7

2.6.1 Detailed Description 7

2.7 CCLIEFB_SLMP_T Struct Reference 7

2.7.1 Detailed Description 8

2.8 GOAL_CCLIEFB_CB_DATA_T Union Reference 8

2.8.1 Detailed Description 8

3 File Documentation 8

3.1 ccliefb_api.c File Reference 8

3.1.1 Detailed Description 9

3.1.2 Function Documentation 9

3.2 ccliefb_core.c File Reference 15

3.2.1 Detailed Description 16

3.2.2 Function Documentation 16

3.2.3 Variable Documentation 18

3.3 ccliefb_core.h File Reference 18

3.3.1 Detailed Description 19

3.3.2 Function Documentation 19

3.3.3 Variable Documentation 21

3.4 ccliefb_includes.h File Reference 21

3.4.1 Detailed Description 21

3.5 ccliefb_net.c File Reference 21

3.5.1 Detailed Description 23

3.5.2 Function Documentation . . . 23
3.6 ccliefb_net.h File Reference . . . 25
3.6.1 Detailed Description . . . 26
3.6.2 Function Documentation . . . 26
3.7 ccliefb_pdm.c File Reference . . . 28
3.7.1 Detailed Description . . . 29
3.7.2 Function Documentation . . . 29
3.8 ccliefb_pdm.h File Reference . . . 32
3.8.1 Detailed Description . . . 33
3.8.2 Function Documentation . . . 33
3.9 ccliefb_slmp.c File Reference . . . 37
3.9.1 Detailed Description . . . 38
3.9.2 Function Documentation . . . 39
3.10 ccliefb_slmp.h File Reference . . . 39
3.10.1 Detailed Description . . . 40
3.10.2 Function Documentation . . . 40
3.11 ccliefb_types.h File Reference . . . 41
3.11.1 Detailed Description . . . 42
3.11.2 Enumeration Type Documentation . . . 42
3.12 goal_ccl_ie_fb.h File Reference . . . 42
3.12.1 Detailed Description . . . 44
3.12.2 Enumeration Type Documentation . . . 44
3.12.3 Function Documentation . . . 45

Index **51**

1 GOAL CC-Link IE Field Basic Reference Manual

1.1 Introduction

The GOAL CC-Link IE Field Basic Protocol stack can be used to implement a Slave Station of the CC-Link IE Field Basic protocol.

This Reference Manual has been generated from the source code. It documents all elements of the source code.

Further details, a description of the API and code samples can be found in the User Manual.

2 Data Structure Documentation

2.1 CCLIEFB_CFG_T Struct Reference

CCLIEFB config data.

Data Fields

- uint16_t **vendorCode**

  vendor code
- uint32_t **modelCode**

  model code
- uint16_t **deviceVersion**

  device version
- uint8_t **numStations**

  number of stations in device

2.1.1 Detailed Description

CCLIEFB config data.

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

- **ccliefb_types.h**

2.2 CCLIEFB_CORE_T Struct Reference

CCLIEFB core instance data.

Data Fields

- uint32_t **stationId**

  host station Id
- GOAL_BOOL_T **appStop**

  application stopped
- uint16_t **slaveErrCode**

  Slave Error Code.

- uint32_t **localManagementInfo**

  Slave Error Code details.
- GOAL_TIMER_T ∗ **pTimer**

  timeout timer
- uint32_t **masterId**

  id of controlling master
- uint8_t **groupId**

  group Id assigned to slave
- GOAL_BOOL_T **masterAppStopped**

  master application is stopped
- GOAL_BOOL_T **cycDataInvalid**

  Cyclic Data from MAster is invalid.

2.2.1 Detailed Description

CCLIEFB core instance data.

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

- **ccliefb_types.h**

2.3 CCLIEFB_CYCDATAREQ_T Struct Reference

cyclicData request data

Data Fields

- **CCLIEFB_MASTER_APP_STATE_T** **appState**

  master application status
- uint32_t **masterId**

  Master's IP address.
- uint8_t **groupId**

  slave's group number
- uint16_t **seqNum**

  frame sequence number
- uint32_t **timeOut**

  timeout in ms for next reception
- uint16_t **paramNum**

  parameter number
- GOAL_BOOL_T **txStateOn**

  cyclic tranmission enabled
- uint8_t **numStations**

  number of device's stations addressed in request
- uint8_t ∗ **pDataRww**

  RWw Ouptut data for station.
- uint8_t ∗ **pDataRy**

  Ry Output data for station.

2.3.1 Detailed Description

cyclicData request data

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

- **ccliefb_types.h**

2.4 CCLIEFB_INSTANCE_T Struct Reference

CCLIEFB instance handle.

Public Member Functions

- **GOAL_INSTANCE_HEADER** (**CCLIEFB_INSTANCE_T**)
  instance header

Data Fields

- **GOAL_CCLIEFB_FUNC_CB_T pAppCb**
  application callback
- **CCLIEFB_CORE_T core**
  core instance data
- **CCLIEFB_CFG_T cfg**
  stack configuration
- **CCLIEFB_NET_T net**
  CCLIEFB net instance data.
- **CCLIEFB_CYCDATAREQ_T cycDataReq**
  cyclicData request data
- **CCLIEFB_SLMP_T slmp**
  SLMP instance data.
- **CCLIEFB_PDM_T pdm**
  process data memory handler

2.4.1 Detailed Description

CCLIEFB instance handle.

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

- **ccliefb_types.h**

2.5 CCLIEFB_NET_T Struct Reference

CCLIEFB net instance data.

Data Fields

- GOAL_NET_CHAN_T * **pChan**

  CyclicData UDP channel.
- GOAL_NET_ADDR_T **addr**

  sender address

2.5.1 Detailed Description

CCLIEFB net instance data.

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

- **ccliefb_types.h**

2.6 CCLIEFB_PDM_T Struct Reference

Process data memory handler.

Data Fields

- uint8_t * **pMem** [**GOAL_CCLIEFB_LINK_DEV_END**]

  process data memory
- uint8_t * **pMemCache** [**GOAL_CCLIEFB_LINK_DEV_END**]

  process data memory cache used by application
- uint16_t **size** [**GOAL_CCLIEFB_LINK_DEV_END**]

  size of process data memory
- GOAL_LOCK_T * **pLock**

  access lock
- GOAL_BOOL_T **bFlgAppAccess**

  application has access flag

2.6.1 Detailed Description

Process data memory handler.

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

- **ccliefb_types.h**

2.7 CCLIEFB_SLMP_T Struct Reference

SLMP instance data.

Data Fields

- GOAL_NET_CHAN_T ∗ **pChan**
  SLMP UDP channel.
- uint16_t **serialNo**
  serial number in request
- uint16_t **cmd**
  command in request
- uint16_t **subCmd**
  subcommand in request

2.7.1 Detailed Description

SLMP instance data.

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

- **ccliefb_types.h**

2.8 GOAL_CCLIEFB_CB_DATA_T Union Reference

application callback data

Data Fields

- uint32_t ∗ **pNewMasterId**
  Id of new master station.
- void ∗ **pRawData**
  raw data

2.8.1 Detailed Description

application callback data

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

- **goal_ccl_ie_fb.h**

3 File Documentation

3.1 ccliefb_api.c File Reference

CC-Link IE Field Basic API.

Functions

- GOAL_STATUS_T **goal_ccIeFbInit** (void)

  Register CC-Link IE Field Basic stack.
- GOAL_STATUS_T **goal_ccIeFbNew** (**GOAL_CCLIEFB_HANDLE_T** **ppCfb, const uint32_t id, **GOAL_CCLIEFB_FUNC_CB_T** pFunc)

  Create a new instance of a CC-Link IE Field Basic Slave Station.
- GOAL_STATUS_T **goal_ccIeFbVersionGet** (**GOAL_CCLIEFB_HANDLE_T** *pCfb, const char **pp↩ Version)

  Get the version of the CC-Link IE Field Basic stack.
- GOAL_STATUS_T **goal_ccIeFbCfgDeviceVersionSet** (uint16_t version)

  Set the Device Version of this station.
- GOAL_STATUS_T **goal_ccIeFbCfgDeviceVendorCodeSet** (uint16_t vendorCode)

  Set the Device Vendor Code of this station.
- GOAL_STATUS_T **goal_ccIeFbCfgDeviceModelCodeSet** (uint32_t productId)

  Set the Model Code of this station.
- GOAL_STATUS_T **goal_ccIeFbCfgNumStationsSet** (uint8_t numStations)

  Set the number of occupied Stations of this device.
- GOAL_STATUS_T **goal_ccIeFbAppStopSet** (**GOAL_CCLIEFB_HANDLE_T** *pCfb, GOAL_BOO↩ L_T stop)

  Set the Stop status of the application.
- GOAL_STATUS_T **goal_ccIeFbAppErrorCodeSet** (**GOAL_CCLIEFB_HANDLE_T** *pCfb, uint16_t errCode, uint32_t errDetails)

  Set an application specific error code.
- GOAL_STATUS_T **goal_ccIeFbIoAccessStart** (**GOAL_CCLIEFB_HANDLE_T** *pCfb)

  Get access to process data.
- GOAL_STATUS_T **goal_ccIeFbIoAccessEnd** (**GOAL_CCLIEFB_HANDLE_T** *pCfb)

  Release access to process data.
- GOAL_STATUS_T **goal_ccIeFbOutputGet** (**GOAL_CCLIEFB_HANDLE_T** *pCfb, **GOAL_CCL↩ IEFB_LINK_DEV_ID_T** devId, uint8_t *pBuf, uint16_t bufLen, uint16_t offset)

  Get data from an Output Link Device (data received from Master)
- GOAL_STATUS_T **goal_ccIeFbInputSet** (**GOAL_CCLIEFB_HANDLE_T** *pCfb, **GOAL_CCLIE↩ FB_LINK_DEV_ID_T** devId, uint8_t *pBuf, uint16_t bufLen, uint16_t offset)

  Set data of an Input Link Device (data sent to Master)

3.1.1 Detailed Description

CC-Link IE Field Basic API.

This module provides the Application Programming Interface of the CCLIEFB TSN stack.

Copyright

3.1.2 Function Documentation

3.1.2.1 goal_cclIeFbAppErrorCodeSet()

```
GOAL_STATUS_T goal_cclIeFbAppErrorCodeSet (
             GOAL_CCLIEFB_HANDLE_T * pCfb,
            uint16_t errCode,
            uint32_t errDetails )
```

Set an application specific error code.

This error code will be part of the CyclicData response

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	GOAL CCLIEFB handle
errCode	application error code
errDetails	additional information

3.1.2.2 goal_cclIeFbAppStopSet()

```
GOAL_STATUS_T goal_cclIeFbAppStopSet (
             GOAL_CCLIEFB_HANDLE_T * pCfb,
            GOAL_BOOL_T stop )
```

Set the Stop status of the application.

This function is used to indicate the stop status to the Master station.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	GOAL CCLIEFB handle
stop	application is stopped

3.1.2.3 goal_cclIeFbCfgDeviceModelCodeSet()

```
GOAL_STATUS_T goal_cclIeFbCfgDeviceModelCodeSet (
            uint32_t productId )
```

Set the Model Code of this station.

This function must be called before **goal_cclIeFbNew()** (p. 14).

Return values

GOAL_OK	successful
other	failed

Parameters

product↩ Id	new product Id

3.1.2.4 goal_cclIeFbCfgDeviceVendorCodeSet()

```
GOAL_STATUS_T goal_cclIeFbCfgDeviceVendorCodeSet (
            uint16_t vendorCode )
```

Set the Device Vendor Code of this station.

This function must be called before **goal_cclIeFbNew()** (p. 14).

Return values

GOAL_OK	successful
other	failed

Parameters

vendorCode	new vendor code

3.1.2.5 goal_cclIeFbCfgDeviceVersionSet()

```
GOAL_STATUS_T goal_cclIeFbCfgDeviceVersionSet (
            uint16_t version )
```

Set the Device Version of this station.

This function must be called before **goal_cclIeFbNew()** (p. 14).

Return values

GOAL_OK	successful
other	failed

Parameters

version	new device version

3.1.2.6 goal_ccIIeFbCfgNumStationsSet()

```
GOAL_STATUS_T goal_cclIeFbCfgNumStationsSet (
            uint8_t numStations )
```

Set the number of occupied Stations of this device.

This function must be called before **goal_ccIIeFbNew()** (p. 14).

Return values

GOAL_OK	successful
other	failed

Parameters

numStations	number of occupied stations

3.1.2.7 goal_ccIIeFbInit()

```
GOAL_STATUS_T goal_cclIeFbInit (
            void  )
```

Register CC-Link IE Field Basic stack.

Return values

GOAL_OK	success
other	failure

3.1.2.8 goal_ccIIeFbInputSet()

```
GOAL_STATUS_T goal_cclIeFbInputSet (
             GOAL_CCLIEFB_HANDLE_T * pCfb,
             GOAL_CCLIEFB_LINK_DEV_ID_T devId,
            uint8_t * pBuf,
            uint16_t bufLen,
            uint16_t offset )
```

Set data of an Input Link Device (data sent to Master)

Return values

GOAL_OK	successful
other	failed

Parameters

	pCfb	GOAL CCLIEFB handle
	devId	link device Id
in	*pBuf*	new data
	bufLen	data length
	offset	offset within link device

3.1.2.9 goal_cclIeFbIoAccessEnd()

```
GOAL_STATUS_T goal_cclIeFbIoAccessEnd (
             GOAL_CCLIEFB_HANDLE_T * pCfb )
```

Release access to process data.

It unblocks update of process data allowing receive and send of current data. This function must be called after all read and write actions which shall be applied at once. Before all read and write actions goal_cclIeFbIoAccessStart must be called to block process data update.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	GOAL CCLIEFB handle

3.1.2.10 goal_cclIeFbIoAccessStart()

```
GOAL_STATUS_T goal_cclIeFbIoAccessStart (
             GOAL_CCLIEFB_HANDLE_T * pCfb )
```

Get access to process data.

It blocks update of process data preventing read or write inconsistent data. This function must be called before all read and write actions which shall be applied at once. After all read and write actions goal_cclIeFbIoAccessEnd must be called to unblock process data update.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	GOAL CCLIEFB handle

3.1.2.11 goal_ccIIeFbNew()

```
GOAL_STATUS_T goal_cclIeFbNew (
             GOAL_CCLIEFB_HANDLE_T ** ppCfb,
            const uint32_t id,
             GOAL_CCLIEFB_FUNC_CB_T pFunc )
```

Create a new instance of a CC-Link IE Field Basic Slave Station.

Create a new instance with the given ID and register a callback.

Return values

GOAL_OK	successful
other	failed

Parameters

ppCfb	GOAL CCLIEFB instance ref
id	instance id
pFunc	GOAL CCLIEFB callback function

3.1.2.12 goal_ccIIeFbOutputGet()

```
GOAL_STATUS_T goal_cclIeFbOutputGet (
             GOAL_CCLIEFB_HANDLE_T * pCfb,
             GOAL_CCLIEFB_LINK_DEV_ID_T devId,
            uint8_t * pBuf,
            uint16_t bufLen,
            uint16_t offset )
```

Get data from an Output Link Device (data received from Master)

Return values

GOAL_OK	successful
other	failed

Parameters

	pCfb	GOAL CCLIEFB handle
	devId	link device Id
out	*pBuf*	write buffer

Parameters

	bufLen	write buffer length
	offset	offset within link device

3.1.2.13 goal_cclIeFbVersionGet()

```
GOAL_STATUS_T goal_cclIeFbVersionGet (
             GOAL_CCLIEFB_HANDLE_T * pCfb,
            const char ** ppVersion )
```

Get the version of the CC-Link IE Field Basic stack.

Return values

GOAL_OK	success
other	failure

Parameters

pCfb	GOAL CCLIEFB handle
ppVersion	version string buffer reference

3.2 cclíefb_core.c File Reference

CCLIEFB Core module.

Macros

- #define **CCLIEFB_CORE_MASTER_ID_UNKNOWN** 0x00000000

  Controlling Master is undetermined.
- #define **CCLIEFB_CORE_GROUP_ID_UNKNOWN** 0xFF

  Group ID is undetermined.

Functions

- GOAL_STATUS_T **cclieфb_coreInit** (void)

  Initialize the CCLIEFB stack.
- GOAL_STATUS_T **cclieфb_coreNew** (**CCLIEFB_INSTANCE_T** *pCfb)

  Create a new instance of the CCLIEFB stack.
- GOAL_STATUS_T **cclieфb_coreTimeoutTimerStop** (**CCLIEFB_INSTANCE_T** *pCfb)

  Stop the timeout timer.
- GOAL_STATUS_T **cclieфb_coreTimeoutTimerStart** (**CCLIEFB_INSTANCE_T** *pCfb)

  Start the timeout timer.
- void **cclieфb_coreCycDataProcess** (**CCLIEFB_INSTANCE_T** *pCfb)

  Process the data of a received CyclicData request.

Variables

- **CCLIEFB_CFG_T ccliefbCfg**
  current configuration of CCLIETSN stack

3.2.1 Detailed Description

CCLIEFB Core module.

This module implements the behavior of the CC-Link IE Field Basic stack.

Copyright

3.2.2 Function Documentation

3.2.2.1 ccliefb_coreCycDataProcess()

```
void ccliefb_coreCycDataProcess (
             CCLIEFB_INSTANCE_T * pCfb )
```

Process the data of a received CyclicData request.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.2.2.2 ccliefb_coreInit()

```
GOAL_STATUS_T ccliefb_coreInit (
            void  )
```

Initialize the CCLIEFB stack.

Register IP set callback for staion IDs and shutdown stage.

Return values

GOAL_OK	- success
other	- failed

3.2.2.3 ccliefb_coreNew()

```
GOAL_STATUS_T ccliefb_coreNew (
              CCLIEFB_INSTANCE_T * pCfb )
```

Create a new instance of the CCLIEFB stack.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.2.2.4 ccliefb_coreTimeoutTimerStart()

```
GOAL_STATUS_T ccliefb_coreTimeoutTimerStart (
              CCLIEFB_INSTANCE_T * pCfb )
```

Start the timeout timer.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.2.2.5 ccliefb_coreTimeoutTimerStop()

```
GOAL_STATUS_T ccliefb_coreTimeoutTimerStop (
              CCLIEFB_INSTANCE_T * pCfb )
```

Stop the timeout timer.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.2.3 Variable Documentation

3.2.3.1 ccliefbCfg

```
CCLIEFB_CFG_T ccliefbCfg
```

Initial value:

```
= {
    CCLIEFB_CFG_DFLT_VENDOR_CODE,
    CCLIEFB_CFG_DFLT_MODEL_CODE,
    CCLIEFB_CFG_DFLT_VERSION,
    CCLIEFB_CFG_DFLT_NUM_STASTIONS,
}
```

current configuration of CCLIETSN stack

current configuration of CCLIEFB stack

3.3 ccliefb_core.h File Reference

CCLIEFB Core module.

Macros

- #define **CCLIEFB_CORE_PROTOCOL_VERSION** 2

  protocol version implemented by this device

Functions

- GOAL_STATUS_T **ccliefb_coreInit** (void)

  Initialize the CCLIEFB stack.
- GOAL_STATUS_T **ccliefb_coreNew** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Create a new instance of the CCLIEFB stack.
- GOAL_STATUS_T **ccliefb_coreTimeoutTimerStop** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Stop the timeout timer.
- GOAL_STATUS_T **ccliefb_coreTimeoutTimerStart** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Start the timeout timer.
- void **ccliefb_coreCycDataProcess** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Process the data of a received CyclicData request.

Variables

- **CCLIEFB_CFG_T ccliefbCfg**

  current configuration of CCLIEFB stack

3.3.1 Detailed Description

CCLIEFB Core module.

This module implements the behavior of the CC-Link IE Field Basic stack.

Copyright

3.3.2 Function Documentation

3.3.2.1 ccliefb_coreCycDataProcess()

```
void ccliefb_coreCycDataProcess (
             CCLIEFB_INSTANCE_T * pCfb )
```

Process the data of a received CyclicData request.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.3.2.2 ccliefb_coreInit()

```
GOAL_STATUS_T ccliefb_coreInit (
            void  )
```

Initialize the CCLIEFB stack.

Register IP set callback for staion IDs and shutdown stage.

Return values

GOAL_OK	- success
other	- failed

3.3.2.3 ccliefb_coreNew()

```
GOAL_STATUS_T ccliefb_coreNew (
             CCLIEFB_INSTANCE_T * pCfb )
```

Create a new instance of the CCLIEFB stack.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.3.2.4 ccliefb_coreTimeoutTimerStart()

```
GOAL_STATUS_T ccliefb_coreTimeoutTimerStart (
             CCLIEFB_INSTANCE_T * pCfb )
```

Start the timeout timer.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.3.2.5 ccliefb_coreTimeoutTimerStop()

```
GOAL_STATUS_T ccliefb_coreTimeoutTimerStop (
             CCLIEFB_INSTANCE_T * pCfb )
```

Stop the timeout timer.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.3.3 Variable Documentation

3.3.3.1 ccliefbCfg

```
CCLIEFB_CFG_T ccliefbCfg
```

current configuration of CCLIEFB stack

current configuration of CCLIEFB stack

3.4 ccliefb_includes.h File Reference

CC-Link IE Field Basic stack.

3.4.1 Detailed Description

CC-Link IE Field Basic stack.

This header file must be included by all files that are part of the CC-Link IE Field Basic stack.

Copyright

3.5 ccliefb_net.c File Reference

CCLIEFB Net module.

Macros

- #define **CCLIEFB_NET_UDP_PORT** 61450
  UDP port number for CyclicData messages.
- #define **CCLIEFB_NET_NUM_GOAL_BUF_FIXED** 2
  GOAL buffers reserved for CCLIEFB.
- #define **CCLIEFB_NET_NUM_GOAL_BUF_TMP** 0
  additioanl GOAL buffers
- #define **CCLIEFB_NET_LEN_CYC_DATA_HDR** 15
  cyclic data header length
- #define **CCLIEFB_NET_HDR_FTYPE_REQ** 0x5000
  fixed value of request frame type
- #define **CCLIEFB_NET_HDR_FTYPE_RES** 0xD000

fixed value of response frame type

- #define **CCLIEFB_NET_HDR_NETNO** 0x00

  fixed value of network number
- #define **CCLIEFB_NET_HDR_NODENO** 0xFF

  fixed value of node number
- #define **CCLIEFB_NET_HDR_DSTPROCNO** 0x03FF

  fixed value of destination processor number
- #define **CCLIEFB_NET_HDR_RESERVED1** 0x00

  fixed value of reserved1
- #define **CCLIEFB_NET_HDR_RESERVED2** 0x0000

  fixed value of reserved2
- #define **CCLIEFB_NET_HDR_COMMAND** 0x0E70

  fixed value of command
- #define **CCLIEFB_NET_HDR_SUBCOMMAND** 0x0000

  fixed value of subcommand
- #define **CCLIEFB_NET_LEN_CYC_DATA_FIXED** 52

  length of fixed CyclicData request fields
- #define **CCLIEFB_NET_LEN_OFFSET_ADDR_RSVD** 14

  length of offsetAddrInfo.reserved field
- #define **CCLIEFB_NET_TIMEOUT_VAL_DFLT** 500

  default timeout value in ms
- #define **CCLIEFB_NET_TIMEOUT_FACTOR_DFLT** 3

  default timeout factor
- #define **CCLIEFB_NET_LEN_CYC_DATA_VAR** 76

  length of variable CyclicData request fields for one station
- #define **CCLIEFB_NET_STATION_ID_ADDITIONAL** 0xFFFFFFFF

  cyclic data also covers additional station of device
- #define **CCLIEFB_NET_CYCINFO_OFST_KNOWN** 36

  known offset of cyclicInfo field in request
- #define **CCLIEFB_NET_CYCINFO_OFST_RES** 40

  offset of cyclicInfo field in response
- #define **CCLIEFB_NET_APP_STATUS_STOP** 0x0000

  device's application stopped
- #define **CCLIEFB_NET_APP_STATUS_RUN** 0x0001

  device's application running
- #define **CCLIEFB_NET_LEN_CYC_DATA_RES_PRE_DL** 9

  length of CyclicData response header before DL fields measurement starts
- #define **CCLIEFB_NET_MASTER_INFO_RUN** (1<<0)

  Master App is running.
- #define **CCLIEFB_NET_MASTER_INFO_STOP_USER** (1<<1)

  Master App stopped by user.
- #define **CCLIEFB_NET_STATION_NUM_MIN** 1

  minimum value for slaveTotalOccupiedStationCount
- #define **CCLIEFB_NET_STATION_NUM_MAX** 16

  minimum value for slaveTotalOccupiedStationCount

Functions

- GOAL_STATUS_T **ccliefb_netInit** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Initialize the net module.
- GOAL_STATUS_T **ccliefb_netShutdown** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Shut down the net module.
- GOAL_STATUS_T **ccliefb_netCycDataResTx** (**CCLIEFB_INSTANCE_T** ∗pCfb, uint16_t endCode)

  Send a CyclicData response.

3.5.1 Detailed Description

CCLIEFB Net module.

This module binds the CC-Link IE Field Basic stack to the GOAL Net module.

Copyright

3.5.2 Function Documentation

3.5.2.1 ccliefb_netCycDataResTx()

```
GOAL_STATUS_T ccliefb_netCycDataResTx (
            CCLIEFB_INSTANCE_T ∗ pCfb,
           uint16_t endCode )
```

Send a CyclicData response.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle
endCode	end code of response

3.5.2.2 ccliefb_netInit()

```
GOAL_STATUS_T ccliefb_netInit (
            CCLIEFB_INSTANCE_T ∗ pCfb )
```

Initialize the net module.

This functions opens a UDP socket that receives CyclicData frames.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.5.2.3 ccliefb_netShutdown()

```
GOAL_STATUS_T ccliefb_netShutdown (
            CCLIEFB_INSTANCE_T * pCfb )
```

Shut down the net module.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.6 ccliefb_net.h File Reference

CCLIEFB Net module.

Macros

- #define **CCLIEFB_NET_END_CODE_OK** 0x0000

  success
- #define **CCLIEFB_NET_END_CODE_ERR_MASTER_DUPL** 0xCFE0

  Master Station duplication.
- #define **CCLIEFB_NET_END_CODE_ERR_NUM_STATIONS** 0xCFE1

  Slave Station cannot handle number of occupied stations.
- #define **CCLIEFB_NET_END_CODE_ERR_OTHER** 0xCFF0

  internal error
- #define **CCLIEFB_NET_END_CODE_ERR_DISCON** 0xCFFF

  Slave Station disconnected itself.

Functions

- GOAL_STATUS_T **ccliefb_netInit** (**CCLIEFB_INSTANCE_T** ∗pCfb)
  Initialize the net module.
- GOAL_STATUS_T **ccliefb_netShutdown** (**CCLIEFB_INSTANCE_T** ∗pCfb)
  Shut down the net module.
- GOAL_STATUS_T **ccliefb_netCycDataResTx** (**CCLIEFB_INSTANCE_T** ∗pCfb, uint16_t endCode)
  Send a CyclicData response.

3.6.1 Detailed Description

CCLIEFB Net module.

This module binds the CC-Link IE Field Basic stack to the GOAL Net module.

Copyright

3.6.2 Function Documentation

3.6.2.1 ccliefb_netCycDataResTx()

```
GOAL_STATUS_T ccliefb_netCycDataResTx (
            CCLIEFB_INSTANCE_T * pCfb,
           uint16_t endCode )
```

Send a CyclicData response.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle
endCode	end code of response

3.6.2.2 ccliefb_netInit()

```
GOAL_STATUS_T ccliefb_netInit (
            CCLIEFB_INSTANCE_T * pCfb )
```

Initialize the net module.

This functions opens a UDP socket that receives CyclicData frames.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.6.2.3 ccliefb_netShutdown()

```
GOAL_STATUS_T ccliefb_netShutdown (
            CCLIEFB_INSTANCE_T * pCfb )
```

Shut down the net module.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.7 ccliefb_pdm.c File Reference

CCLIEFB Process Data Handler.

Functions

- GOAL_STATUS_T **ccliefb_pdMemInit** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Initialize the Process data memory.
- GOAL_STATUS_T **ccliefb_pdMemShutdown** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Shutdown the Process data memory.
- GOAL_STATUS_T **ccliefb_pdLockGet** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Get lock of process data buffer.
- void **ccliefb_pdLockPut** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Put lock of process data buffer.
- GOAL_STATUS_T **ccliefb_pdAppAccessStart** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Get access to process data from application side.
- GOAL_STATUS_T **ccliefb_pdAppAccessEnd** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Release access to process data from application side.
- GOAL_STATUS_T **ccliefb_pdRead** (**CCLIEFB_INSTANCE_T** ∗pCfb, **GOAL_CCLIEFB_LINK_↩ DEV_ID_T** linkDevId, uint8_t ∗pBuf, uint16_t dataLen, uint16_t offset, GOAL_BOOL_T flgIsApp)

Read data from a Link device.

- GOAL_STATUS_T **ccliefb_pdWrite** (**CCLIEFB_INSTANCE_T** ∗pCfb, **GOAL_CCLIEFB_LINK↩_DEV_ID_T** linkDevId, uint8_t ∗pBuf, uint16_t dataLen, uint16_t offset, GOAL_BOOL_T flgIsApp)

  Write data to a Link device.

3.7.1 Detailed Description

CCLIEFB Process Data Handler.

This module provides functions for accessing the process data memory.

Copyright

3.7.2 Function Documentation

3.7.2.1 ccliefb_pdAppAccessEnd()

```
GOAL_STATUS_T ccliefb_pdAppAccessEnd (
             CCLIEFB_INSTANCE_T * pCfb )
```

Release access to process data from application side.

The cache Input data are copied to Input buffer which allowed the application to write consistent data using multiple write operations.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.7.2.2 ccliefb_pdAppAccessStart()

```
GOAL_STATUS_T ccliefb_pdAppAccessStart (
             CCLIEFB_INSTANCE_T * pCfb )
```

Get access to process data from application side.

The current Output data are copied to cache allowing the application to read consistent data using multiple read operations.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.7.2.3 ccliefb_pdLockGet()

```
GOAL_STATUS_T ccliefb_pdLockGet (
                CCLIEFB_INSTANCE_T * pCfb )
```

Get lock of process data buffer.

This allows to do more than one read or write operation at once while process data stay consistent.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.7.2.4 ccliefb_pdLockPut()

```
void ccliefb_pdLockPut (
                CCLIEFB_INSTANCE_T * pCfb )
```

Put lock of process data buffer.

This allows to do more than one read or write operation at once while process data stay consistent.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.7.2.5 ccliefb_pdMemInit()

```
GOAL_STATUS_T ccliefb_pdMemInit (
             CCLIEFB_INSTANCE_T * pCfb )
```

Initialize the Process data memory.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.7.2.6 ccliefb_pdMemShutdown()

```
GOAL_STATUS_T ccliefb_pdMemShutdown (
             CCLIEFB_INSTANCE_T * pCfb )
```

Shutdown the Process data memory.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.7.2.7 ccliefb_pdRead()

```
GOAL_STATUS_T ccliefb_pdRead (
             CCLIEFB_INSTANCE_T * pCfb,
             GOAL_CCLIEFB_LINK_DEV_ID_T linkDevId,
            uint8_t * pBuf,
            uint16_t dataLen,
            uint16_t offset,
            GOAL_BOOL_T flgIsApp )
```

Read data from a Link device.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle
link↩ DevId	link device id
pBuf	read data
dataLen	read data length
offset	offset in link device
flgIsApp	called by application flag

3.7.2.8 ccliefb_pdWrite()

```
GOAL_STATUS_T ccliefb_pdWrite (
             CCLIEFB_INSTANCE_T * pCfb,
             GOAL_CCLIEFB_LINK_DEV_ID_T linkDevId,
            uint8_t * pBuf,
            uint16_t dataLen,
            uint16_t offset,
            GOAL_BOOL_T flgIsApp )
```

Write data to a Link device.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle
link↩ DevId	link device id
pBuf	write data
dataLen	write data length
offset	offset in link device
flgIsApp	called by application flag

3.8 ccliefb_pdm.h File Reference

CCLIEFB Process Data Handler.

Functions

- GOAL_STATUS_T **ccliefb_pdMemInit** (**CCLIEFB_INSTANCE_T** *pCfb)

  Initialize the Process data memory.
- GOAL_STATUS_T **ccliefb_pdMemShutdown** (**CCLIEFB_INSTANCE_T** *pCfb)

  Shutdown the Process data memory.
- GOAL_STATUS_T **ccliefb_pdLockGet** (**CCLIEFB_INSTANCE_T** *pCfb)

  Get lock of process data buffer.
- void **ccliefb_pdLockPut** (**CCLIEFB_INSTANCE_T** *pCfb)

  Put lock of process data buffer.
- GOAL_STATUS_T **ccliefb_pdAppAccessStart** (**CCLIEFB_INSTANCE_T** *pCfb)

  Get access to process data from application side.
- GOAL_STATUS_T **ccliefb_pdAppAccessEnd** (**CCLIEFB_INSTANCE_T** *pCfb)

  Release access to process data from application side.
- GOAL_STATUS_T **ccliefb_pdRead** (**CCLIEFB_INSTANCE_T** *pCfb, **GOAL_CCLIEFB_LINK_DEV_ID_T** linkDevId, uint8_t *pBuf, uint16_t dataLen, uint16_t offset, GOAL_BOOL_T flgIsApp)

  Read data from a Link device.
- GOAL_STATUS_T **ccliefb_pdWrite** (**CCLIEFB_INSTANCE_T** *pCfb, **GOAL_CCLIEFB_LINK_DEV_ID_T** linkDevId, uint8_t *pBuf, uint16_t dataLen, uint16_t offset, GOAL_BOOL_T flgIsApp)

  Write data to a Link device.

3.8.1 Detailed Description

CCLIEFB Process Data Handler.

This module provides functions for accessing the process data memory.

Copyright

3.8.2 Function Documentation

3.8.2.1 ccliefb_pdAppAccessEnd()

```
GOAL_STATUS_T ccliefb_pdAppAccessEnd (
            CCLIEFB_INSTANCE_T * pCfb )
```

Release access to process data from application side.

The cache Input data are copied to Input buffer which allowed the application to write consistent data using multiple write operations.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.8.2.2 ccliefb_pdAppAccessStart()

```
GOAL_STATUS_T ccliefb_pdAppAccessStart (
            CCLIEFB_INSTANCE_T * pCfb )
```

Get access to process data from application side.

The current Output data are copied to cache allowing the application to read consistent data using multiple read operations.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.8.2.3 ccliefb_pdLockGet()

```
GOAL_STATUS_T ccliefb_pdLockGet (
            CCLIEFB_INSTANCE_T * pCfb )
```

Get lock of process data buffer.

This allows to do more than one read or write operation at once while process data stay consistent.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.8.2.4 ccliefb_pdLockPut()

```
void ccliefb_pdLockPut (
            CCLIEFB_INSTANCE_T * pCfb )
```

Put lock of process data buffer.

This allows to do more than one read or write operation at once while process data stay consistent.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.8.2.5 ccliefb_pdMemInit()

```
GOAL_STATUS_T ccliefb_pdMemInit (
            CCLIEFB_INSTANCE_T * pCfb )
```

Initialize the Process data memory.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.8.2.6 ccliefb_pdMemShutdown()

```
GOAL_STATUS_T ccliefb_pdMemShutdown (
            CCLIEFB_INSTANCE_T * pCfb )
```

Shutdown the Process data memory.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

3.8.2.7 ccliefb_pdRead()

```
GOAL_STATUS_T ccliefb_pdRead (
             CCLIEFB_INSTANCE_T * pCfb,
             GOAL_CCLIEFB_LINK_DEV_ID_T linkDevId,
            uint8_t * pBuf,
            uint16_t dataLen,
            uint16_t offset,
            GOAL_BOOL_T flgIsApp )
```

Read data from a Link device.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle
link↩DevId	link device id
pBuf	read data
dataLen	read data length
offset	offset in link device
flgIsApp	called by application flag

3.8.2.8 ccliefb_pdWrite()

```
GOAL_STATUS_T ccliefb_pdWrite (
             CCLIEFB_INSTANCE_T * pCfb,
             GOAL_CCLIEFB_LINK_DEV_ID_T linkDevId,
            uint8_t * pBuf,
            uint16_t dataLen,
            uint16_t offset,
            GOAL_BOOL_T flgIsApp )
```

Write data to a Link device.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	CCLIEFB instance handle

Parameters

link↩ DevId	link device id
pBuf	write data
dataLen	write data length
offset	offset in link device
flgIsApp	called by application flag

3.9 ccliefb_slmp.c File Reference

CCLIEFB SLMP module.

Macros

- #define **CCLIEFB_SLMP_UDP_PORT** 61451

  UDP port number for SLMP messages.
- #define **CCLIEFB_SLMP_NUM_GOAL_BUF_FIXED** 0

  GOAL buffers reserved for CCLIEFB.
- #define **CCLIEFB_SLMP_NUM_GOAL_BUF_TMP** 2

  additioanl GOAL buffers
- #define **CCLIEFB_SLMP_LEN_MT_HDR** 19

  length of header of a SLMP MT request
- #define **CCLIEFB_SLMP_LEN_NODESEARCH_REQ** 11

  length of NodeSearch request data
- #define **CCLIEFB_SLMP_LEN_NODESEARCH_RES** 51

  length of NodeSearch response data
- #define **CCLIEFB_SLMP_LEN_IPADDRSET_REQ_MIN** 26

  minimum length of IPAddressSet request data
- #define **CCLIEFB_SLMP_LEN_IPADDRSET_RES** 6

  length of IPAddressSet response data
- #define **CCLIEFB_SLMP_HDR_FTYPE_MT_REQ** 0x0054

  MT type request.
- #define **CCLIEFB_SLMP_HDR_FTYPE_MT_RES** 0x00D4

  MT type request.
- #define **CCLIEFB_SLMP_HDR_RESERVED1** 0x0000

  reserved field
- #define **CCLIEFB_SLMP_HDR_NETNO** 0x00

  fixed value of network number
- #define **CCLIEFB_SLMP_HDR_NODENO** 0xFF

  fixed value of node number
- #define **CCLIEFB_SLMP_HDR_DSTPROCNO** 0x03FF

  fixed value of destination processor number
- #define **CCLIEFB_SLMP_HDR_RESERVED2** 0x00

  reserved field
- #define **CCLIEFB_SLMP_DL_RES_OFFSET** 2

difference of DL value and actual response data size
- #define **CCLIEFB_SLMP_DL_ERRRES_OFFSET** 11
  difference of DL value and actual error response data size
- #define **CCLIEFB_SLMP_CMD_NODESEARCH** 0x0E30
  SLMP command NodeSearch.
- #define **CCLIEFB_SLMP_CMD_IPADDRESSSET** 0x0E31
  SLMP command IpAddressSet.
- #define **CCLIEFB_SLMP_SUBCMD_DEFAULT** 0x0000
  default SLMP subcommand
- #define **CCLIEFB_SLMP_END_CODE_OK** 0x0000
  success
- #define **CCLIEFB_SLMP_END_CODE_ERR_CMD** 0xC059
  invalid command or subcommand
- #define **CCLIEFB_SLMP_END_CODE_ERR_REQ** 0xC05C
  error in request data
- #define **CCLIEFB_SLMP_END_CODE_ERR_LEN** 0xC061
  request length does not match data
- #define **CCLIEFB_SLMP_END_CODE_ERR_BUSY** 0xCEE0
  request cannot be processed, server busy
- #define **CCLIEFB_SLMP_NS_RES_IP_ADDR_SIZE** 4
  size of the IP address
- #define **CCLIEFB_SLMP_NS_RES_GW_ADDR** 0xFFFFFFFF
  Gateway IP address.
- #define **CCLIEFB_SLMP_NS_RES_HOSTNAME_SIZE** 0
  Host name size.
- #define **CCLIEFB_SLMP_NS_RES_TU_ADDR** 0xFFFFFFFF
  target unit IP address
- #define **CCLIEFB_SLMP_NS_RES_TU_PORT** 0xFFFF
  target unit port number
- #define **CCLIEFB_SLMP_NS_RES_PROTO_UDP** 0x01
  server protocol: UDP
- #define **CCLIEFB_SLMP_IPS_REQ_IP_ADDR_SIZE** 4
  size of the IP address

Functions

- GOAL_STATUS_T **ccliefb_slmpInit** (**CCLIEFB_INSTANCE_T** ∗pCfb)
  Initialize the net module.
- GOAL_STATUS_T **ccliefb_slmpShutdown** (**CCLIEFB_INSTANCE_T** ∗pCfb)
  Shut down the SLMP module.

3.9.1 Detailed Description

CCLIEFB SLMP module.

This module handles SLMP requests.

Copyright

3.9.2 Function Documentation

3.9.2.1 ccliefb_slmpInit()

```
GOAL_STATUS_T ccliefb_slmpInit (
             CCLIEFB_INSTANCE_T * pCfb )
```

Initialize the net module.

This functions opens a UDP socket that receives CyclicData frames.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.9.2.2 ccliefb_slmpShutdown()

```
GOAL_STATUS_T ccliefb_slmpShutdown (
             CCLIEFB_INSTANCE_T * pCfb )
```

Shut down the SLMP module.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.10 ccliefb_slmp.h File Reference

CCLIEFB SLMP module.

Functions

- GOAL_STATUS_T **ccliefb_slmpInit** (**CCLIEFB_INSTANCE_T** *pCfb)
  Initialize the net module.

- GOAL_STATUS_T **ccliefb_slmpShutdown** (**CCLIEFB_INSTANCE_T** ∗pCfb)

  Shut down the SLMP module.

3.10.1 Detailed Description

CCLIEFB SLMP module.

This module handles SLMP requests.

Copyright

3.10.2 Function Documentation

3.10.2.1 ccliefb_slmpInit()

```
GOAL_STATUS_T ccliefb_slmpInit (
            CCLIEFB_INSTANCE_T * pCfb )
```

Initialize the net module.

This functions opens a UDP socket that receives CyclicData frames.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.10.2.2 ccliefb_slmpShutdown()

```
GOAL_STATUS_T ccliefb_slmpShutdown (
            CCLIEFB_INSTANCE_T * pCfb )
```

Shut down the SLMP module.

Return values

GOAL_OK	- success
other	- failed

Parameters

pCfb	CCLIEFB instance handle

3.11 ccliefb_types.h File Reference

CCLIEFB data types.

Data Structures

- struct **CCLIEFB_CFG_T**

  CCLIEFB config data.
- struct **CCLIEFB_CORE_T**

  CCLIEFB core instance data.
- struct **CCLIEFB_NET_T**

  CCLIEFB net instance data.
- struct **CCLIEFB_CYCDATAREQ_T**

  cyclicData request data
- struct **CCLIEFB_SLMP_T**

  SLMP instance data.
- struct **CCLIEFB_PDM_T**

  Process data memory handler.
- struct **CCLIEFB_INSTANCE_T**

  CCLIEFB instance handle.

Macros

- #define **CCLIEFB_CFG_DFLT_VENDOR_CODE** 0x0E13

  default value of device vendor code
- #define **CCLIEFB_CFG_DFLT_MODEL_CODE** 0x00000001

  default value of model code
- #define **CCLIEFB_CFG_DFLT_VERSION** 0x0101

  default value of device version
- #define **CCLIEFB_CFG_DFLT_NUM_STASTIONS** 4

  default value of number of stations

Enumerations

- enum **CCLIEFB_MASTER_APP_STATE_T** { **CCLIEFB_MASTER_APP_RUN**, **CCLIEFB_MASTER_APP_STOP_UNKNOWN**, **CCLIEFB_MASTER_APP_STOP_ERROR**, **CCLIEFB_MASTER_APP_STOP_USER** }

  master application status

3.11.1 Detailed Description

CCLIEFB data types.

This module defines dat types used by the CC-Link IE Field Basic stack.

Copyright

> Copyright 2010-2020 port GmbH Halle/Saale. This software is protected Intellectual Property and may only be used according to the license agreement.

3.11.2 Enumeration Type Documentation

3.11.2.1 CCLIEFB_MASTER_APP_STATE_T

`enum CCLIEFB_MASTER_APP_STATE_T`

master application status

Enumerator

CCLIEFB_MASTER_APP_RUN	master application is running
CCLIEFB_MASTER_APP_STOP_UNKNOWN	master application stopped
CCLIEFB_MASTER_APP_STOP_ERROR	master application stopped by error
CCLIEFB_MASTER_APP_STOP_USER	master application stopped by user

3.12 goal_ccl_ie_fb.h File Reference

CC-Link IE Field Basic API.

Data Structures

- union **GOAL_CCLIEFB_CB_DATA_T**

  application callback data

Macros

- #define **GOAL_CCLIEFB_VERSION** STR(GOAL_VER_MAJ) "." STR(GOAL_VER_MIN) "." STR(G↩ OAL_VER_SUB)

  CCLIEFB version.
- #define **GOAL_CCLIEFB_INSTANCE_DEFAULT** 0

  default instance Id
- #define **GOAL_CCLIEFB_CYCLIC_LEN_RX** 8

  length of RX data per station
- #define **GOAL_CCLIEFB_CYCLIC_LEN_RWR** 64

  length of RWr data per station

- #define **GOAL_CCLIEFB_CYCLIC_LEN_RY** 8

  length of RY data per station
- #define **GOAL_CCLIEFB_CYCLIC_LEN_RWW** 64

  length of RWw data per station

Typedefs

- typedef void **GOAL_CCLIEFB_HANDLE_T**

  GOAL CCLIEFB instance handle.
- typedef GOAL_STATUS_T(∗ **GOAL_CCLIEFB_FUNC_CB_T**) (**GOAL_CCLIEFB_HANDLE_T** ∗pCfb, **GOAL_CCLIEFB_CB_ID_T** cbId, **GOAL_CCLIEFB_CB_DATA_T** ∗pCbData)

  application callback

Enumerations

- enum **GOAL_CCLIEFB_CB_ID_T** { **GOAL_CCLIEFB_CB_ID_TIMEOUT**, **GOAL_CCLIEFB_CB_ID_CYC_STOP**, **GOAL_CCLIEFB_CB_ID_NEW_MASTER**, **GOAL_CCLIEFB_CB_ID_MASTER_STOP_ERR**, **GOAL_CCLIEFB_CB_ID_MASTER_STOP_USER**, **GOAL_CCLIEFB_CB_ID_MASTER_STOP_UNKNOWN**, **GOAL_CCLIEFB_CB_ID_MASTER_RUN**, **GOAL_CCLIEFB_CB_ID_CYC_DATA_VALID**, **GOAL_CCLIEFB_CB_ID_CYC_DATA_INVALID**, **GOAL_CCLIEFB_CB_ID_DUPL_MASTER**, **GOAL_CCLIEFB_CB_ID_DUPL_SLAVE** }

  application callback IDs
- enum **GOAL_CCLIEFB_LINK_DEV_ID_T** { **GOAL_CCLIEFB_LINK_DEV_RX** = 0, **GOAL_CCLIEFB_LINK_DEV_RWR** = 1, **GOAL_CCLIEFB_LINK_DEV_RY** = 2, **GOAL_CCLIEFB_LINK_DEV_RWW** = 3, **GOAL_CCLIEFB_LINK_DEV_END** }

  Link device type Ids.

Functions

- GOAL_STATUS_T **goal_cclIeFbInit** (void)

  Register CC-Link IE Field Basic stack.
- GOAL_STATUS_T **goal_cclIeFbNew** (**GOAL_CCLIEFB_HANDLE_T** ∗∗ppCfb, const uint32_t id, **GOAL_CCLIEFB_FUNC_CB_T** pFunc)

  Create a new instance of a CC-Link IE Field Basic Slave Station.
- GOAL_STATUS_T **goal_cclIeFbVersionGet** (**GOAL_CCLIEFB_HANDLE_T** ∗pCfb, const char ∗∗ppVersion)

  Get the version of the CC-Link IE Field Basic stack.
- GOAL_STATUS_T **goal_cclIeFbCfgDeviceVersionSet** (uint16_t version)

  Set the Device Version of this station.
- GOAL_STATUS_T **goal_cclIeFbCfgDeviceVendorCodeSet** (uint16_t vendorCode)

  Set the Device Vendor Code of this station.
- GOAL_STATUS_T **goal_cclIeFbCfgDeviceModelCodeSet** (uint32_t productId)

  Set the Model Code of this station.
- GOAL_STATUS_T **goal_cclIeFbCfgNumStationsSet** (uint8_t numStations)

  Set the number of occupied Stations of this device.

- GOAL_STATUS_T **goal_cclIeFbAppStopSet** (**GOAL_CCLIEFB_HANDLE_T** ∗pCfb, GOAL_BOOL_T stop)

  Set the Stop status of the application.
- GOAL_STATUS_T **goal_cclIeFbAppErrorCodeSet** (**GOAL_CCLIEFB_HANDLE_T** ∗pCfb, uint16_t errCode, uint32_t errDetails)

  Set an application specific error code.
- GOAL_STATUS_T **goal_cclIeFbIoAccessStart** (**GOAL_CCLIEFB_HANDLE_T** ∗pCfb)

  Get access to process data.
- GOAL_STATUS_T **goal_cclIeFbIoAccessEnd** (**GOAL_CCLIEFB_HANDLE_T** ∗pCfb)

  Release access to process data.
- GOAL_STATUS_T **goal_cclIeFbOutputGet** (**GOAL_CCLIEFB_HANDLE_T** ∗pCfb, **GOAL_CCLIEFB_LINK_DEV_ID_T** devId, uint8_t ∗pBuf, uint16_t bufLen, uint16_t offset)

  Get data from an Output Link Device (data received from Master)
- GOAL_STATUS_T **goal_cclIeFbInputSet** (**GOAL_CCLIEFB_HANDLE_T** ∗pCfb, **GOAL_CCLIEFB_LINK_DEV_ID_T** devId, uint8_t ∗pBuf, uint16_t bufLen, uint16_t offset)

  Set data of an Input Link Device (data sent to Master)

3.12.1 Detailed Description

CC-Link IE Field Basic API.

This module provides the Application Programming Interface of the CCLIEFB TSN stack.

Copyright

> Copyright 2020 port GmbH Halle/Saale. This software is protected Intellectual Property and may only be used according to the license agreement.

3.12.2 Enumeration Type Documentation

3.12.2.1 GOAL_CCLIEFB_CB_ID_T

`enum GOAL_CCLIEFB_CB_ID_T`

application callback IDs

Enumerator

GOAL_CCLIEFB_CB_ID_TIMEOUT	cyclic connection to master timed out
GOAL_CCLIEFB_CB_ID_CYC_STOP	cyclic stop instruction from master
GOAL_CCLIEFB_CB_ID_NEW_MASTER	new master station
GOAL_CCLIEFB_CB_ID_MASTER_STOP_ERR	Master application stopped by error.
GOAL_CCLIEFB_CB_ID_MASTER_STOP_USER	Master application stopped by application.
GOAL_CCLIEFB_CB_ID_MASTER_STOP_UNKNOWN	Master application stopped by unknown reason.
GOAL_CCLIEFB_CB_ID_MASTER_RUN	Master application running.
GOAL_CCLIEFB_CB_ID_CYC_DATA_VALID	cyclic data from Master is valid
GOAL_CCLIEFB_CB_ID_CYC_DATA_INVALID	cyclic data from Master is invalid
GOAL_CCLIEFB_CB_ID_DUPL_MASTER	Master station duplication detected.
GOAL_CCLIEFB_CB_ID_DUPL_SLAVE	Slave station ID duplication detected.

3.12.2.2 GOAL_CCLIEFB_LINK_DEV_ID_T

```
enum GOAL_CCLIEFB_LINK_DEV_ID_T
```

Link device type Ids.

Enumerator

GOAL_CCLIEFB_LINK_DEV_RX	Link device type: RX (Bit Input, SM)
GOAL_CCLIEFB_LINK_DEV_RWR	Link device type: RWr (Word Input, SM)
GOAL_CCLIEFB_LINK_DEV_RY	Link device type: RY (Bit Output, MS)
GOAL_CCLIEFB_LINK_DEV_RWW	Link device type: RWw (Word Output, MS)
GOAL_CCLIEFB_LINK_DEV_END	end marker (internal use only)

3.12.3 Function Documentation

3.12.3.1 goal_cclIeFbAppErrorCodeSet()

```
GOAL_STATUS_T goal_cclIeFbAppErrorCodeSet (
             GOAL_CCLIEFB_HANDLE_T * pCfb,
            uint16_t errCode,
            uint32_t errDetails )
```

Set an application specific error code.

This error code will be part of the CyclicData response

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	GOAL CCLIEFB handle
errCode	application error code
errDetails	additional information

3.12.3.2 goal_cclIeFbAppStopSet()

```
GOAL_STATUS_T goal_cclIeFbAppStopSet (
             GOAL_CCLIEFB_HANDLE_T * pCfb,
            GOAL_BOOL_T stop )
```

Set the Stop status of the application.

This function is used to indicate the stop status to the Master station.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	GOAL CCLIEFB handle
stop	application is stopped

3.12.3.3 goal_cclIeFbCfgDeviceModelCodeSet()

```
GOAL_STATUS_T goal_cclIeFbCfgDeviceModelCodeSet (
            uint32_t productId )
```

Set the Model Code of this station.

This function must be called before **goal_cclIeFbNew()** (p. 14).

Return values

GOAL_OK	successful
other	failed

Parameters

product↩ Id	new product Id

3.12.3.4 goal_cclIeFbCfgDeviceVendorCodeSet()

```
GOAL_STATUS_T goal_cclIeFbCfgDeviceVendorCodeSet (
            uint16_t vendorCode )
```

Set the Device Vendor Code of this station.

This function must be called before **goal_cclIeFbNew()** (p. 14).

Return values

GOAL_OK	successful
other	failed

Parameters

vendorCode	new vendor code

3.12.3.5 goal_ccIeFbCfgDeviceVersionSet()

```
GOAL_STATUS_T goal_cclIeFbCfgDeviceVersionSet (
            uint16_t version )
```

Set the Device Version of this station.

This function must be called before **goal_ccIeFbNew()** (p. 14).

Return values

GOAL_OK	successful
other	failed

Parameters

version	new device version

3.12.3.6 goal_ccIeFbCfgNumStationsSet()

```
GOAL_STATUS_T goal_cclIeFbCfgNumStationsSet (
            uint8_t numStations )
```

Set the number of occupied Stations of this device.

This function must be called before **goal_ccIeFbNew()** (p. 14).

Return values

GOAL_OK	successful
other	failed

Parameters

numStations	number of occupied stations

3.12.3.7 goal_ccIeFbInit()

```
GOAL_STATUS_T goal_cclIeFbInit (
            void )
```

Register CC-Link IE Field Basic stack.

Return values

GOAL_OK	success
other	failure

3.12.3.8 goal_cclIeFbInputSet()

```
GOAL_STATUS_T goal_cclIeFbInputSet (
             GOAL_CCLIEFB_HANDLE_T * pCfb,
             GOAL_CCLIEFB_LINK_DEV_ID_T devId,
            uint8_t * pBuf,
            uint16_t bufLen,
            uint16_t offset )
```

Set data of an Input Link Device (data sent to Master)

Return values

GOAL_OK	successful
other	failed

Parameters

	pCfb	GOAL CCLIEFB handle
	devId	link device Id
in	*pBuf*	new data
	bufLen	data length
	offset	offset within link device

3.12.3.9 goal_cclIeFbIoAccessEnd()

```
GOAL_STATUS_T goal_cclIeFbIoAccessEnd (
             GOAL_CCLIEFB_HANDLE_T * pCfb )
```

Release access to process data.

It unblocks update of process data allowing receive and send of current data. This function must be called after all read and write actions which shall be applied at once. Before all read and write actions goal_cclIeFbIoAccessStart must be called to block process data update.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	GOAL CCLIEFB handle

3.12.3.10 goal_ccIIeFbIoAccessStart()

```
GOAL_STATUS_T goal_ccIIeFbIoAccessStart (
             GOAL_CCLIEFB_HANDLE_T * pCfb )
```

Get access to process data.

It blocks update of process data preventing read or write inconsistent data. This function must be called before all read and write actions which shall be applied at once. After all read and write actions goal_ccIIeFbIoAccessEnd must be called to unblock process data update.

Return values

GOAL_OK	successful
other	failed

Parameters

pCfb	GOAL CCLIEFB handle

3.12.3.11 goal_ccIIeFbNew()

```
GOAL_STATUS_T goal_ccIIeFbNew (
             GOAL_CCLIEFB_HANDLE_T ** ppCfb,
            const uint32_t id,
             GOAL_CCLIEFB_FUNC_CB_T pFunc )
```

Create a new instance of a CC-Link IE Field Basic Slave Station.

Create a new instance with the given ID and register a callback.

Return values

GOAL_OK	successful
other	failed

Parameters

ppCfb	GOAL CCLIEFB instance ref
id	instance id
pFunc	GOAL CCLIEFB callback function

3.12.3.12 goal_cclIeFbOutputGet()

```
GOAL_STATUS_T goal_cclIeFbOutputGet (
             GOAL_CCLIEFB_HANDLE_T * pCfb,
             GOAL_CCLIEFB_LINK_DEV_ID_T devId,
            uint8_t * pBuf,
            uint16_t bufLen,
            uint16_t offset )
```

Get data from an Output Link Device (data received from Master)

Return values

GOAL_OK	successful
other	failed

Parameters

	pCfb	GOAL CCLIEFB handle
	devId	link device Id
out	*pBuf*	write buffer
	bufLen	write buffer length
	offset	offset within link device

3.12.3.13 goal_cclIeFbVersionGet()

```
GOAL_STATUS_T goal_cclIeFbVersionGet (
             GOAL_CCLIEFB_HANDLE_T * pCfb,
            const char ** ppVersion )
```

Get the version of the CC-Link IE Field Basic stack.

Return values

GOAL_OK	success
other	failure

Parameters

pCfb	GOAL CCLIEFB handle
ppVersion	version string buffer reference

Index

CCLIEFB_CFG_T, 4
CCLIEFB_CORE_T, 4
CCLIEFB_CYCDATAREQ_T, 5
CCLIEFB_INSTANCE_T, 6
CCLIEFB_MASTER_APP_STATE_T
  cclief b_types.h, 42
CCLIEFB_NET_T, 6
CCLIEFB_PDM_T, 7
CCLIEFB_SLMP_T, 7
ccliefb_api.c, 8
  goal_ccIeFbAppErrorCodeSet, 9
  goal_ccIeFbAppStopSet, 10
  goal_ccIeFbCfgDeviceModelCodeSet, 10
  goal_ccIeFbCfgDeviceVendorCodeSet, 11
  goal_ccIeFbCfgDeviceVersionSet, 11
  goal_ccIeFbCfgNumStationsSet, 12
  goal_ccIeFbInit, 12
  goal_ccIeFbInputSet, 12
  goal_ccIeFbIoAccessEnd, 13
  goal_ccIeFbIoAccessStart, 13
  goal_ccIeFbNew, 14
  goal_ccIeFbOutputGet, 14
  goal_ccIeFbVersionGet, 15
ccliefb_core.c, 15
  ccliefb_coreCycDataProcess, 16
  ccliefb_coreInit, 16
  ccliefb_coreNew, 17
  ccliefb_coreTimeoutTimerStart, 17
  ccliefb_coreTimeoutTimerStop, 17
  ccliefbCfg, 18
ccliefb_core.h, 18
  ccliefb_coreCycDataProcess, 19
  ccliefb_coreInit, 19
  ccliefb_coreNew, 20
  ccliefb_coreTimeoutTimerStart, 20
  ccliefb_coreTimeoutTimerStop, 20
  ccliefbCfg, 21
ccliefb_coreCycDataProcess
  ccliefb_core.c, 16
  ccliefb_core.h, 19
ccliefb_coreInit
  ccliefb_core.c, 16
  ccliefb_core.h, 19
ccliefb_coreNew
  ccliefb_core.c, 17
  ccliefb_core.h, 20
ccliefb_coreTimeoutTimerStart
  ccliefb_core.c, 17
  ccliefb_core.h, 20
ccliefb_coreTimeoutTimerStop
  ccliefb_core.c, 17
  ccliefb_core.h, 20
ccliefb_includes.h, 21
ccliefb_net.c, 21
  ccliefb_netCycDataResTx, 23
  ccliefb_netInit, 23
  ccliefb_netShutdown, 25
ccliefb_net.h, 25
  ccliefb_netCycDataResTx, 26
  ccliefb_netInit, 26
  ccliefb_netShutdown, 28
ccliefb_netCycDataResTx
  ccliefb_net.c, 23
  ccliefb_net.h, 26
ccliefb_netInit
  ccliefb_net.c, 23
  ccliefb_net.h, 26
ccliefb_netShutdown
  ccliefb_net.c, 25
  ccliefb_net.h, 28
ccliefb_pdAppAccessEnd
  ccliefb_pdm.c, 29
  ccliefb_pdm.h, 33
ccliefb_pdAppAccessStart
  ccliefb_pdm.c, 29
  ccliefb_pdm.h, 34
ccliefb_pdLockGet
  ccliefb_pdm.c, 30
  ccliefb_pdm.h, 34
ccliefb_pdLockPut
  ccliefb_pdm.c, 30
  ccliefb_pdm.h, 34
ccliefb_pdMemInit
  ccliefb_pdm.c, 31
  ccliefb_pdm.h, 35
ccliefb_pdMemShutdown
  ccliefb_pdm.c, 31
  ccliefb_pdm.h, 35
ccliefb_pdRead
  ccliefb_pdm.c, 31
  ccliefb_pdm.h, 36
ccliefb_pdWrite
  ccliefb_pdm.c, 32
  ccliefb_pdm.h, 36
ccliefb_pdm.c, 28
  ccliefb_pdAppAccessEnd, 29
  ccliefb_pdAppAccessStart, 29
  ccliefb_pdLockGet, 30
  ccliefb_pdLockPut, 30
  ccliefb_pdMemInit, 31
  ccliefb_pdMemShutdown, 31
  ccliefb_pdRead, 31
  ccliefb_pdWrite, 32
ccliefb_pdm.h, 32
  ccliefb_pdAppAccessEnd, 33
  ccliefb_pdAppAccessStart, 34

- ccliefb_pdLockGet, 34
- ccliefb_pdLockPut, 34
- ccliefb_pdMemInit, 35
- ccliefb_pdMemShutdown, 35
- ccliefb_pdRead, 36
- ccliefb_pdWrite, 36

ccliefb_slmp.c, 37
- ccliefb_slmpInit, 39
- ccliefb_slmpShutdown, 39

ccliefb_slmp.h, 39
- ccliefb_slmpInit, 40
- ccliefb_slmpShutdown, 40

ccliefb_slmpInit
- ccliefb_slmp.c, 39
- ccliefb_slmp.h, 40

ccliefb_slmpShutdown
- ccliefb_slmp.c, 39
- ccliefb_slmp.h, 40

ccliefb_types.h, 41
- CCLIEFB_MASTER_APP_STATE_T, 42

ccliefbCfg
- ccliefb_core.c, 18
- ccliefb_core.h, 21

GOAL_CCLIEFB_CB_DATA_T, 8

GOAL_CCLIEFB_CB_ID_T
- goal_ccl_ie_fb.h, 44

GOAL_CCLIEFB_LINK_DEV_ID_T
- goal_ccl_ie_fb.h, 45

goal_ccl_ie_fb.h, 42
- GOAL_CCLIEFB_CB_ID_T, 44
- GOAL_CCLIEFB_LINK_DEV_ID_T, 45
- goal_cclIeFbAppErrorCodeSet, 45
- goal_cclIeFbAppStopSet, 45
- goal_cclIeFbCfgDeviceModelCodeSet, 46
- goal_cclIeFbCfgDeviceVendorCodeSet, 46
- goal_cclIeFbCfgDeviceVersionSet, 47
- goal_cclIeFbCfgNumStationsSet, 47
- goal_cclIeFbInit, 47
- goal_cclIeFbInputSet, 48
- goal_cclIeFbIoAccessEnd, 48
- goal_cclIeFbIoAccessStart, 49
- goal_cclIeFbNew, 49
- goal_cclIeFbOutputGet, 50
- goal_cclIeFbVersionGet, 50

goal_cclIeFbAppErrorCodeSet
- ccliefb_api.c, 9
- goal_ccl_ie_fb.h, 45

goal_cclIeFbAppStopSet
- ccliefb_api.c, 10
- goal_ccl_ie_fb.h, 45

goal_cclIeFbCfgDeviceModelCodeSet
- ccliefb_api.c, 10
- goal_ccl_ie_fb.h, 46

goal_cclIeFbCfgDeviceVendorCodeSet
- ccliefb_api.c, 11
- goal_ccl_ie_fb.h, 46

goal_cclIeFbCfgDeviceVersionSet
- ccliefb_api.c, 11
- goal_ccl_ie_fb.h, 47

goal_cclIeFbCfgNumStationsSet
- ccliefb_api.c, 12
- goal_ccl_ie_fb.h, 47

goal_cclIeFbInit
- ccliefb_api.c, 12
- goal_ccl_ie_fb.h, 47

goal_cclIeFbInputSet
- ccliefb_api.c, 12
- goal_ccl_ie_fb.h, 48

goal_cclIeFbIoAccessEnd
- ccliefb_api.c, 13
- goal_ccl_ie_fb.h, 48

goal_cclIeFbIoAccessStart
- ccliefb_api.c, 13
- goal_ccl_ie_fb.h, 49

goal_cclIeFbNew
- ccliefb_api.c, 14
- goal_ccl_ie_fb.h, 49

goal_cclIeFbOutputGet
- ccliefb_api.c, 14
- goal_ccl_ie_fb.h, 50

goal_cclIeFbVersionGet
- ccliefb_api.c, 15
- goal_ccl_ie_fb.h, 50