BlueParrot XCP Documentation

Release 1.0

Christoph Schueler

Jun 19, 2023

Contents:

1 Installation **1**

2 Tutorial **3**

3 User API **5**

4 Configuration Options **7**
- 4.1 General Options . . . 7
- 4.2 Resource Protection Options . . . 9
- 4.3 DAQ Options . . . 9
- 4.4 Optional Services . . . 10
  - 4.4.1 Optional Standard Services . . . 11
  - 4.4.2 Optional Calibration Services . . . 11
  - 4.4.3 Optional Paging Services . . . 11
  - 4.4.4 Optional DAQ Services . . . 11
  - 4.4.5 Optional Programming Services . . . 12
- 4.5 Transport-Layer specific options . . . 12
- 4.6 Customization options . . . 12
- 4.7 Platform specific options . . . 12
- 4.8 Section to cross-reference . . . 12

5 Hook Functions **13**

6 Indices and tables **15**

Index **17**

CHAPTER 1

Installation

CHAPTER 2

Tutorial

CHAPTER 3

User API

See Doxygen API doc.

CHAPTER 4

Configuration Options

Configuration options need to be supplied in a file named **xcp_config.h**. Typically one uses a file from the examples directory as a starting point.

Note:

- **bool** is either **XCP_ON** or **XCP_OFF** .
- Options starting with **XCP_ENABLE_** are of type **bool** to enable/disable some functionality.

4.1 General Options

XCP_GET_ID_0
A short description of your node. This is the return value of **GET_ID(0)** .

XCP_GET_ID_1
A2L filename without path and extension. This is the return value of **GET_ID(1)** .

XCP_BUILD_TYPE
Build type is either **XCP_DEBUG_BUILD** or **XCP_RELEASE_BUILD**. Most useful if you need to debug your application or your're a contributor to **BlueParrotXCP**.

XCP_ENABLE_EXTERN_C_GUARDS
If enabled, include-files are frame with:

```
#if defined(__cplusplus)
extern "C"
{
#endif  /* __cplusplus *

/*
...
...
```

(continues on next page)

(continued from previous page)

```
*/

#if defined(__cplusplus)
}
#endif  /* __cplusplus *
```

Required for C++ linkage.

XCP_ENABLE_SLAVE_BLOCKMODE
If enabled, slave may use block transfer mode.

XCP_ENABLE_MASTER_BLOCKMODE
If enabled, master may use block transfer mode. In this case, options *XCP_MAX_BS* and *XCP_MIN_ST* apply.

XCP_ENABLE_STIM ****bool****
STIM is currently not **implemented.**

XCP_CHECKSUM_METHOD

Choose one of:

- XCP_CHECKSUM_METHOD_XCP_ADD_11
- XCP_CHECKSUM_METHOD_XCP_ADD_12
- XCP_CHECKSUM_METHOD_XCP_ADD_14
- XCP_CHECKSUM_METHOD_XCP_ADD_22
- XCP_CHECKSUM_METHOD_XCP_ADD_24
- XCP_CHECKSUM_METHOD_XCP_ADD_44
- XCP_CHECKSUM_METHOD_XCP_CRC_16
- XCP_CHECKSUM_METHOD_XCP_CRC_16_CITT
- XCP_CHECKSUM_METHOD_XCP_CRC_32

XCP_USER_DEFINED not supported yet.

XCP_CHECKSUM_CHUNKED_CALCULATION ****bool****
If **XCP_FALSE** checksums are completely calculated for requested memory blocks, otherwise chunked in a periodically called MainFunction.

XCP_CHECKSUM_CHUNK_SIZE
Chunk size in bytes. s. `XCP_CHECKSUM_CHUNKED_CALCULATION`

XCP_CHECKSUM_MAXIMUM_BLOCK_SIZE
You may want to limit maximum checksum block size (in bytes), **0** means unlimited (4294967295 to be exact).

XCP_BYTE_ORDER
Byteorder / endianess of your platform, choose either **XCP_BYTE_ORDER_INTEL** or **XCP_BYTE_ORDER_MOTOROLA**

XCP_ADDRESS_GRANULARITY
Choose **XCP_ADDRESS_GRANULARITY_BYTE**, **XCP_ADDRESS_GRANULARITY_WORD** and **XCP_ADDRESS_GRANULARITY_DWORD** are not supported yet.

XCP_ENABLE_STATISTICS
If enabled collect some statistics like traffic and so on.

XCP_MAX_BS
Indicates the maximum allowed block size as the number of consecutive command packets (**DOWNLOAD_NEXT**) in a block sequence.

XCP_MIN_ST
Indicates the required minimum separation time between the packets of a block transfer from the master device to the slave device in units of 100 microseconds.

XCP_QUEUE_SIZE
Applies to **INTERLEAVED_MODE**, which is currently not supported.

4.2 Resource Protection Options

These option determine the initial per-session resource protection state.

XCP_PROTECT_CAL **bool**

XCP_PROTECT_PAG **bool**

XCP_PROTECT_DAQ **bool**

XCP_PROTECT_STIM **bool**

XCP_PROTECT_PGM **bool**

4.3 DAQ Options

XCP_DAQ_CONFIG_TYPE

- **XCP_DAQ_CONFIG_TYPE_NONE** No DAQ lists at all.
- **XCP_DAQ_CONFIG_TYPE_STATIC** Only static DAQ lists.
- **XCP_DAQ_CONFIG_TYPE_DYNAMIC** Only dynamic DAQ lists.

XCP_DAQ_DTO_BUFFER_SIZE
Size of DTO message buffer (in bytes).

XCP_DAQ_ENABLE_PREDEFINED_LISTS
Enable support for predefined DAQ lists.

XCP_DAQ_TIMESTAMP_UNIT
Choose:

- XCP_DAQ_TIMESTAMP_UNIT_1NS
- XCP_DAQ_TIMESTAMP_UNIT_10NS
- XCP_DAQ_TIMESTAMP_UNIT_100NS
- XCP_DAQ_TIMESTAMP_UNIT_1US
- XCP_DAQ_TIMESTAMP_UNIT_10US
- XCP_DAQ_TIMESTAMP_UNIT_100US
- XCP_DAQ_TIMESTAMP_UNIT_1MS
- XCP_DAQ_TIMESTAMP_UNIT_10MS
- XCP_DAQ_TIMESTAMP_UNIT_100MS

- XCP_DAQ_TIMESTAMP_UNIT_1S
- XCP_DAQ_TIMESTAMP_UNIT_1PS
- XCP_DAQ_TIMESTAMP_UNIT_10PS
- XCP_DAQ_TIMESTAMP_UNIT_100PS

XCP_DAQ_TIMESTAMP_SIZE
Timestamps could be either 1, 2, or 4 bytes in size:

- XCP_DAQ_TIMESTAMP_SIZE_1
- XCP_DAQ_TIMESTAMP_SIZE_2
- XCP_DAQ_TIMESTAMP_SIZE_4

XCP_DAQ_ENABLE_PRESCALER ****bool****
DAQ list prescaling is currently not supported.

XCP_DAQ_ENABLE_ADDR_EXT ****bool****
Measurement quantities can't have an address extension yet.

XCP_DAQ_ENABLE_BIT_OFFSET ****bool****
Bit offsets are currently not supported.

XCP_DAQ_ENABLE_PRIORITIZATION ****bool****
DAQ list prioritization not supported yet.

XCP_DAQ_ENABLE_ALTERNATING ****bool****
Alternating display mode not supported yet.

XCP_DAQ_ENABLE_WRITE_THROUGH ****bool****
XCP_OFF: Disable internal buffering of **DTO** messages, in this case buffering must be handled by your network/socket stack.

XCP_DAQ_MAX_DYNAMIC_ENTITIES
The maximum number of allocatable DAQ entities – DAQ lists, ODTs, and ODT entries. Multiply by sizeof(*XcpDaq_EntityType*) on your platform to get memory usage.

XCP_DAQ_MAX_EVENT_CHANNEL
Number of available event channels.

XCP_DAQ_ENABLE_MULTIPLE_DAQ_LISTS_PER_EVENT ****bool****
Enable/disable support for multiple DAQ list per event.

XCP_DAQ_ENABLE_RESET_DYN_DAQ_CONFIG_ON_SEQUENCE_ERROR ****bool****
Expert option: If **XCP_ON**, re-initialize dynamic DAQ structures after an **ERR_SEQUENCE**.

XCP_DAQ_LIST_TYPE **uint8_t**
Choose: **uint8_t**, **uint16_t**, or **uint32_t**.

XCP_DAQ_ODT_TYPE **uint8_t**
Choose: **uint8_t**, **uint16_t**, or **uint32_t**.

XCP_DAQ_ODT_ENTRY_TYPE **uint8_t**
Choose: **uint8_t**, **uint16_t**, or **uint32_t**.

4.4 Optional Services

Enable/disable optional XCP service categories/services. These options are rather self-explanatory.

4.4.1 Optional Standard Services

```
    XCP_ENABLE_GET_COMM_MODE_INFO
    XCP_ENABLE_GET_ID
    XCP_ENABLE_SET_REQUEST
    XCP_ENABLE_GET_SEED
    XCP_ENABLE_UNLOCK
    XCP_ENABLE_SET_MTA
    XCP_ENABLE_UPLOAD
    XCP_ENABLE_SHORT_UPLOAD
    XCP_ENABLE_BUILD_CHECKSUM
    XCP_ENABLE_TRANSPORT_LAYER_CMD
    XCP_ENABLE_USER_CMD
XCP_ENABLE_CAL_COMMANDS
```

4.4.2 Optional Calibration Services

```
    XCP_ENABLE_DOWNLOAD_NEXT
    XCP_ENABLE_DOWNLOAD_MAX
    XCP_ENABLE_SHORT_DOWNLOAD
    XCP_ENABLE_MODIFY_BITS
XCP_ENABLE_PAG_COMMANDS
```

4.4.3 Optional Paging Services

```
    XCP_ENABLE_GET_PAG_PROCESSOR_INFO
    XCP_ENABLE_GET_SEGMENT_INFO
    XCP_ENABLE_GET_PAGE_INFO
    XCP_ENABLE_SET_SEGMENT_MODE
    XCP_ENABLE_GET_SEGMENT_MODE
    XCP_ENABLE_COPY_CAL_PAGE
XCP_ENABLE_DAQ_COMMANDS
```

4.4.4 Optional DAQ Services

```
    XCP_ENABLE_GET_DAQ_CLOCK
    XCP_ENABLE_READ_DAQ
    XCP_ENABLE_GET_DAQ_PROCESSOR_INFO
    XCP_ENABLE_GET_DAQ_RESOLUTION_INFO
```

```
    XCP_ENABLE_GET_DAQ_LIST_INFO
    XCP_ENABLE_GET_DAQ_EVENT_INFO
    XCP_ENABLE_FREE_DAQ
    XCP_ENABLE_ALLOC_DAQ
    XCP_ENABLE_ALLOC_ODT
    XCP_ENABLE_ALLOC_ODT_ENTRY
    XCP_ENABLE_WRITE_DAQ_MULTIPLE
XCP_ENABLE_PGM_COMMANDS
```

4.4.5 Optional Programming Services

```
    XCP_ENABLE_GET_PGM_PROCESSOR_INFO
    XCP_ENABLE_GET_SECTOR_INFO
    XCP_ENABLE_PROGRAM_PREPARE
    XCP_ENABLE_PROGRAM_FORMAT
    XCP_ENABLE_PROGRAM_NEXT
    XCP_ENABLE_PROGRAM_MAX
    XCP_ENABLE_PROGRAM_VERIFY
```

4.5 Transport-Layer specific options

4.6 Customization options

4.7 Platform specific options

4.8 Section to cross-reference

This is the text of the section. It refers to the section itself, see *Section to cross-reference*.

CHAPTER 5

Hook Functions

Hook functions are used to customize the behaviour of **Blueparrot XCP**.

bool **`Xcp_HookFunction_GetSeed`** (*uint8_t resource*, Xcp_1DArrayType * *result*)
hello

bool **`Xcp_HookFunction_Unlock`** (*uint8_t resource*, Xcp_1DArrayType const * *key*)
world

- optional if…
- required if …

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Index

I

implemented. (*C macro*), 8

U

uint8_t (*C macro*), 10

X

XCP_ADDRESS_GRANULARITY (*C macro*), 8
XCP_BUILD_TYPE (*C macro*), 7
XCP_BYTE_ORDER (*C macro*), 8
XCP_CHECKSUM_CHUNK_SIZE (*C macro*), 8
XCP_CHECKSUM_MAXIMUM_BLOCK_SIZE (*C macro*), 8
XCP_CHECKSUM_METHOD (*C macro*), 8
XCP_DAQ_CONFIG_TYPE (*C macro*), 9
XCP_DAQ_DTO_BUFFER_SIZE (*C macro*), 9
XCP_DAQ_ENABLE_PREDEFINED_LISTS (*C macro*), 9
XCP_DAQ_MAX_DYNAMIC_ENTITIES (*C macro*), 10
XCP_DAQ_MAX_EVENT_CHANNEL (*C macro*), 10
XCP_DAQ_TIMESTAMP_SIZE (*C macro*), 10
XCP_DAQ_TIMESTAMP_UNIT (*C macro*), 9
XCP_ENABLE_ALLOC_DAQ (*C macro*), 12
XCP_ENABLE_ALLOC_ODT (*C macro*), 12
XCP_ENABLE_ALLOC_ODT_ENTRY (*C macro*), 12
XCP_ENABLE_BUILD_CHECKSUM (*C macro*), 11
XCP_ENABLE_CAL_COMMANDS (*C macro*), 11
XCP_ENABLE_COPY_CAL_PAGE (*C macro*), 11
XCP_ENABLE_DAQ_COMMANDS (*C macro*), 11
XCP_ENABLE_DOWNLOAD_MAX (*C macro*), 11
XCP_ENABLE_DOWNLOAD_NEXT (*C macro*), 11
XCP_ENABLE_EXTERN_C_GUARDS (*C macro*), 7
XCP_ENABLE_FREE_DAQ (*C macro*), 12
XCP_ENABLE_GET_COMM_MODE_INFO (*C macro*), 11
XCP_ENABLE_GET_DAQ_CLOCK (*C macro*), 11
XCP_ENABLE_GET_DAQ_EVENT_INFO (*C macro*), 12
XCP_ENABLE_GET_DAQ_LIST_INFO (*C macro*), 11
XCP_ENABLE_GET_DAQ_PROCESSOR_INFO (*C macro*), 11
XCP_ENABLE_GET_DAQ_RESOLUTION_INFO (*C macro*), 11
XCP_ENABLE_GET_ID (*C macro*), 11
XCP_ENABLE_GET_PAG_PROCESSOR_INFO (*C macro*), 11
XCP_ENABLE_GET_PAGE_INFO (*C macro*), 11
XCP_ENABLE_GET_PGM_PROCESSOR_INFO (*C macro*), 12
XCP_ENABLE_GET_SECTOR_INFO (*C macro*), 12
XCP_ENABLE_GET_SEED (*C macro*), 11
XCP_ENABLE_GET_SEGMENT_INFO (*C macro*), 11
XCP_ENABLE_GET_SEGMENT_MODE (*C macro*), 11
XCP_ENABLE_MASTER_BLOCKMODE (*C macro*), 8
XCP_ENABLE_MODIFY_BITS (*C macro*), 11
XCP_ENABLE_PAG_COMMANDS (*C macro*), 11
XCP_ENABLE_PGM_COMMANDS (*C macro*), 12
XCP_ENABLE_PROGRAM_FORMAT (*C macro*), 12
XCP_ENABLE_PROGRAM_MAX (*C macro*), 12
XCP_ENABLE_PROGRAM_NEXT (*C macro*), 12
XCP_ENABLE_PROGRAM_PREPARE (*C macro*), 12
XCP_ENABLE_PROGRAM_VERIFY (*C macro*), 12
XCP_ENABLE_READ_DAQ (*C macro*), 11
XCP_ENABLE_SET_MTA (*C macro*), 11
XCP_ENABLE_SET_REQUEST (*C macro*), 11
XCP_ENABLE_SET_SEGMENT_MODE (*C macro*), 11
XCP_ENABLE_SHORT_DOWNLOAD (*C macro*), 11
XCP_ENABLE_SHORT_UPLOAD (*C macro*), 11
XCP_ENABLE_SLAVE_BLOCKMODE (*C macro*), 8
XCP_ENABLE_STATISTICS (*C macro*), 8
XCP_ENABLE_TRANSPORT_LAYER_CMD (*C macro*), 11
XCP_ENABLE_UNLOCK (*C macro*), 11
XCP_ENABLE_UPLOAD (*C macro*), 11
XCP_ENABLE_USER_CMD (*C macro*), 11
XCP_ENABLE_WRITE_DAQ_MULTIPLE (*C macro*), 12
XCP_GET_ID_0 (*C macro*), 7
XCP_GET_ID_1 (*C macro*), 7

`Xcp_HookFunction_GetSeed` (*C function*), 13
`Xcp_HookFunction_Unlock` (*C function*), 13
`XCP_MAX_BS` (*C macro*), 8
`XCP_MIN_ST` (*C macro*), 9
`XCP_QUEUE_SIZE` (*C macro*), 9