

AC784xx_DFP CRC

6.1.0

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	2
2.1	File List	2
3	Class Documentation	3
3.1	Crc_ConfigType Struct Reference	3
3.1.1	Detailed Description	3
3.1.2	Member Data Documentation	3
3.1.2.1	FinalXOR	3
3.1.2.2	Poly	4
3.1.2.3	Protocol	4
3.1.2.4	ReadTranspose	4
3.1.2.5	Seed	4
3.1.2.6	WriteTranspose	4
4	File Documentation	5
4.1	AC784xx_API_Reference_Manual_CRC.pdf File Reference	5
4.2	AC784xx_Crc_Reg.h File Reference	5
4.2.1	Detailed Description	6
4.2.2	Function Documentation	6
4.2.2.1	Crc_Reg_GetData()	6
4.2.2.2	Crc_Reg_GetPoly()	6
4.2.2.3	Crc_Reg_GetProtocolType()	7
4.2.2.4	Crc_Reg_GetReadTranspose()	7

4.2.2.5	Crc_Reg_GetResultXorMode()	8
4.2.2.6	Crc_Reg_GetWriteTranspose()	8
4.2.2.7	Crc_Reg_SetCtrl()	9
4.2.2.8	Crc_Reg_SetData()	9
4.2.2.9	Crc_Reg_SetDataL()	10
4.2.2.10	Crc_Reg_SetDataLL()	10
4.2.2.11	Crc_Reg_SetPoly()	11
4.2.2.12	Crc_Reg_SetProtocolType()	11
4.2.2.13	Crc_Reg_SetReadTranspose()	12
4.2.2.14	Crc_Reg_SetResultXorMode()	12
4.2.2.15	Crc_Reg_SetSeedOrDataMode()	13
4.2.2.16	Crc_Reg_SetWriteTranspose()	13
4.3	Crc_Hal.c File Reference	13
4.3.1	Detailed Description	15
4.3.2	Macro Definition Documentation	15
4.3.2.1	CRC_POLYNOMIAL16	15
4.3.2.2	CRC_POLYNOMIAL16ARC	15
4.3.2.3	CRC_POLYNOMIAL32	16
4.3.2.4	CRC_POLYNOMIAL32P4	16
4.3.2.5	CRC_POLYNOMIAL64	16
4.3.2.6	CRC_POLYNOMIAL8	17
4.3.2.7	CRC_POLYNOMIAL8H2F	17
4.3.2.8	CRC_TABLE16_SIZE	17
4.3.2.9	CRC_TABLE256_SIZE	17
4.3.3	Function Documentation	17
4.3.3.1	Crc_Hal_CalculateCRC()	17
4.3.3.2	Crc_Hal_CalculateCRC16()	18
4.3.3.3	Crc_Hal_CalculateCRC16ARC()	18
4.3.3.4	Crc_Hal_CalculateCRC32()	19
4.3.3.5	Crc_Hal_CalculateCRC32P4()	20
4.3.3.6	Crc_Hal_CalculateCRC64()	20

4.3.3.7	Crc_Hal_CalculateCRC8()	21
4.3.3.8	Crc_Hal_CalculateCRC8H2F()	22
4.3.3.9	Crc_Hal_Deinit()	22
4.3.3.10	Crc_Hal_DmaCalculateCRC()	23
4.3.3.11	Crc_Hal_GetConfig()	23
4.3.3.12	Crc_Hal_GetCRCResult()	24
4.3.3.13	Crc_Hal_Init()	24
4.3.3.14	Crc_Hal_SetSeed()	25
4.4	Crc_Hal.h File Reference	25
4.4.1	Detailed Description	26
4.4.2	Function Documentation	26
4.4.2.1	Crc_Hal_CalculateCRC()	26
4.4.2.2	Crc_Hal_CalculateCRC16()	27
4.4.2.3	Crc_Hal_CalculateCRC16ARC()	27
4.4.2.4	Crc_Hal_CalculateCRC32()	28
4.4.2.5	Crc_Hal_CalculateCRC32P4()	29
4.4.2.6	Crc_Hal_CalculateCRC64()	29
4.4.2.7	Crc_Hal_CalculateCRC8()	30
4.4.2.8	Crc_Hal_CalculateCRC8H2F()	30
4.4.2.9	Crc_Hal_Deinit()	31
4.4.2.10	Crc_Hal_DmaCalculateCRC()	31
4.4.2.11	Crc_Hal_GetConfig()	32
4.4.2.12	Crc_Hal_GetCRCResult()	32
4.4.2.13	Crc_Hal_Init()	33
4.4.2.14	Crc_Hal_SetSeed()	33
4.5	Crc_Hal_Types.h File Reference	34
4.5.1	Detailed Description	35
4.5.2	Macro Definition Documentation	35
4.5.2.1	CRC_DATA_IS_DATA	35
4.5.2.2	CRC_DATA_IS_SEED	35
4.5.2.3	CRC_HARDWARE_MODE	35
4.5.2.4	CRC_INITIAL_VALUE16	35
4.5.2.5	CRC_INITIAL_VALUE16ARC	36
4.5.2.6	CRC_INITIAL_VALUE32	36
4.5.2.7	CRC_INITIAL_VALUE64	36
4.5.2.8	CRC_INITIAL_VALUE8	36
4.5.2.9	CRC_INITIAL_VALUE8H2F	36
4.5.2.10	CRC_RUNTIME_MODE	37
4.5.2.11	CRC_TABLE_16_BYTE_MODE	37
4.5.2.12	CRC_TABLE_256_BYTE_MODE	37
4.5.3	Enumeration Type Documentation	37
4.5.3.1	Crc_ProtocolType	37
4.5.3.2	Crc_TransposeType	37

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Crc_ConfigType	
CRC configuration structure	3

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_CRC.pdf	5
AC784xx_Crc_Reg.h	
This file provides extern Crc Reg API	5
Crc_Hal.c	
This file provides extern Crc Hal API implement	13
Crc_Hal.h	
This file provides extern Crc Hal API	25
Crc_Hal_Types.h	
This file provides extern Crc macro enum and structure info for hal/mcal	34

Chapter 3

Class Documentation

3.1 Crc_ConfigType Struct Reference

CRC configuration structure.

```
#include <Crc_Hal_Types.h>
```

Public Attributes

- [Crc_ProtocolType Protocol](#)
- [Crc_TransposeType WriteTranspose](#)
- [Crc_TransposeType ReadTranspose](#)
- boolean [FinalXOR](#)
- uint32 [Poly](#)
- uint32 [Seed](#)

3.1.1 Detailed Description

CRC configuration structure.

This structure holds the configuration settings for the crc

Definition at line 112 of file Crc_Hal_Types.h.

3.1.2 Member Data Documentation

3.1.2.1 FinalXOR

```
boolean Crc_ConfigType::FinalXOR
```

Enable/disable result XOR

Definition at line 117 of file Crc_Hal_Types.h.

3.1.2.2 Poly

```
uint32 Crc_ConfigType::Poly
```

CRC polynomial

Definition at line 118 of file Crc_Hal_Types.h.

3.1.2.3 Protocol

```
Crc_ProtocolType Crc_ConfigType::Protocol
```

CRC 16/32 protocol type

Definition at line 114 of file Crc_Hal_Types.h.

3.1.2.4 ReadTranspose

```
Crc_TransposeType Crc_ConfigType::ReadTranspose
```

CRC read out transpose type

Definition at line 116 of file Crc_Hal_Types.h.

3.1.2.5 Seed

```
uint32 Crc_ConfigType::Seed
```

CRC Seed

Definition at line 119 of file Crc_Hal_Types.h.

3.1.2.6 WriteTranspose

```
Crc_TransposeType Crc_ConfigType::WriteTranspose
```

CRC write in transpose type

Definition at line 115 of file Crc_Hal_Types.h.

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

- [Crc_Hal_Types.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_CRC.pdf File Reference

4.2 AC784xx_Crc_Reg.h File Reference

This file provides extern Crc Reg API.

```
#include "Device_Register.h"
```

Functions

- LOCAL_INLINE void [Crc_Reg_SetCtrl](#) (uint8 Instance, uint32 Value)
Config CRC CTRL register.
- LOCAL_INLINE void [Crc_Reg_SetPoly](#) (uint8 Instance, uint32 Value)
Set CRC Poly register.
- LOCAL_INLINE uint32 [Crc_Reg_GetPoly](#) (uint8 Instance)
Get CRC Poly register Value.
- LOCAL_INLINE void [Crc_Reg_SetData](#) (uint8 Instance, uint32 Value)
Sets the 32 bits of CRC data register.
- LOCAL_INLINE uint32 [Crc_Reg_GetData](#) (uint8 Instance)
Get CRC Data register Value.
- LOCAL_INLINE void [Crc_Reg_SetDataL](#) (uint8 Instance, uint16 Value)
Sets the low 16 bits of CRC data register.
- LOCAL_INLINE void [Crc_Reg_SetDataLL](#) (uint8 Instance, uint8 Value)
Sets the low 8 bits of CRC data register.
- LOCAL_INLINE void [Crc_Reg_SetProtocolType](#) (uint8 Instance, uint8 Type)
Select crc protocol mode between crc16 or crc32.
- LOCAL_INLINE uint8 [Crc_Reg_GetProtocolType](#) (uint8 Instance)
Get crc protocol mode between crc16 or crc32.
- LOCAL_INLINE void [Crc_Reg_SetSeedOrDataMode](#) (uint8 Instance, uint8 Mode)
Select Seed or data mode.
- LOCAL_INLINE void [Crc_Reg_SetWriteTranspose](#) (uint8 Instance, uint8 TransType)
Set CRC write transpose mode.
- LOCAL_INLINE uint8 [Crc_Reg_GetWriteTranspose](#) (uint8 Instance)
Get CRC write transpose mode.

- LOCAL_INLINE void [Crc_Reg_SetReadTranspose](#) (uint8 Instance, uint8 TransType)
Set CRC read transpose mode.
- LOCAL_INLINE uint8 [Crc_Reg_GetReadTranspose](#) (uint8 Instance)
Get CRC read transpose mode.
- LOCAL_INLINE void [Crc_Reg_SetResultXorMode](#) (uint8 Instance, boolean En)
Set CRC result xor mode.
- LOCAL_INLINE boolean [Crc_Reg_GetResultXorMode](#) (uint8 Instance)
Get CRC read xor mode.

4.2.1 Detailed Description

This file provides extern Crc Reg API.

4.2.2 Function Documentation

4.2.2.1 Crc_Reg_GetData()

```
LOCAL_INLINE uint32 Crc_Reg_GetData (  
    uint8 Instance )
```

Get CRC Data register Value.

Note

Function ID : DES_CRC_API_305

Parameters

in	Instance	CRC Hardware Device instance
----	----------	------------------------------

Returns

data Value

Definition at line 122 of file AC784xx_Crc_Reg.h.

4.2.2.2 Crc_Reg_GetPoly()

```
LOCAL_INLINE uint32 Crc_Reg_GetPoly (  
    uint8 Instance )
```

Get CRC Poly register Value.

Note

Function ID : DES_CRC_API_303

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
----	-----------------	------------------------------

Returns

polynomial Value

Definition at line 99 of file AC784xx_Crc_Reg.h.

4.2.2.3 Crc_Reg_GetProtocolType()

```
LOCAL_INLINE uint8 Crc_Reg_GetProtocolType (  
    uint8 Instance )
```

Get crc protocol mode between crc16 or crc32.

Note

Function ID : DES_CRC_API_309

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
----	-----------------	------------------------------

Returns

crc protocol mode

Definition at line 169 of file AC784xx_Crc_Reg.h.

4.2.2.4 Crc_Reg_GetReadTranspose()

```
LOCAL_INLINE uint8 Crc_Reg_GetReadTranspose (  
    uint8 Instance )
```

Get CRC read transpose mode.

Note

Function ID : DES_CRC_API_314

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
----	-----------------	------------------------------

Returns

read transpose mode

Definition at line 227 of file AC784xx_Crc_Reg.h.

4.2.2.5 Crc_Reg_GetResultXorMode()

```
LOCAL_INLINE boolean Crc_Reg_GetResultXorMode (  
    uint8 Instance )
```

Get CRC read xor mode.

Note

Function ID : DES_CRC_API_314

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
----	-----------------	------------------------------

Returns

read xor mode

Definition at line 250 of file AC784xx_Crc_Reg.h.

4.2.2.6 Crc_Reg_GetWriteTranspose()

```
LOCAL_INLINE uint8 Crc_Reg_GetWriteTranspose (  
    uint8 Instance )
```

Get CRC write transpose mode.

Note

Function ID : DES_CRC_API_312

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
----	-----------------	------------------------------

Returns

write transpose mode

Definition at line 204 of file AC784xx_Crc_Reg.h.

4.2.2.7 Crc_Reg_SetCtrl()

```
LOCAL_INLINE void Crc_Reg_SetCtrl (
    uint8 Instance,
    uint32 Value )
```

Config CRC CTRL register.

Note

Function ID : DES_CRC_API_301

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
in	<i>Value</i>	config Value

Returns

void

Definition at line 76 of file AC784xx_Crc_Reg.h.

4.2.2.8 Crc_Reg_SetData()

```
LOCAL_INLINE void Crc_Reg_SetData (
    uint8 Instance,
    uint32 Value )
```

Sets the 32 bits of CRC data register.

Note

Function ID : DES_CRC_API_304

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
in	<i>Value</i>	data Value

Returns

void

Definition at line 111 of file AC784xx_Crc_Reg.h.

4.2.2.9 Crc_Reg_SetDataL()

```
LOCAL_INLINE void Crc_Reg_SetDataL (
    uint8 Instance,
    uint16 Value )
```

Sets the low 16 bits of CRC data register.

Note

Function ID : DES_CRC_API_306

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
in	<i>Value</i>	data Value

Returns

void

Definition at line 134 of file AC784xx_Crc_Reg.h.

4.2.2.10 Crc_Reg_SetDataLL()

```
LOCAL_INLINE void Crc_Reg_SetDataLL (
    uint8 Instance,
    uint8 Value )
```

Sets the low 8 bits of CRC data register.

Note

Function ID : DES_CRC_API_307

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
in	<i>Value</i>	data Value

Returns

void

Definition at line 146 of file AC784xx_Crc_Reg.h.

4.2.2.11 Crc_Reg_SetPoly()

```
LOCAL_INLINE void Crc_Reg_SetPoly (
    uint8 Instance,
    uint32 Value )
```

Set CRC Poly register.

Note

Function ID : DES_CRC_API_302

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
in	<i>Value</i>	polynomial Value

Returns

void

Definition at line 88 of file AC784xx_Crc_Reg.h.

4.2.2.12 Crc_Reg_SetProtocolType()

```
LOCAL_INLINE void Crc_Reg_SetProtocolType (
    uint8 Instance,
    uint8 Type )
```

Select crc protocol mode between crc16 or crc32.

Note

Function ID : DES_CRC_API_308

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
in	<i>Type</i>	crc check mode select

Returns

void

Definition at line 158 of file AC784xx_Crc_Reg.h.

4.2.2.13 Crc_Reg_SetReadTranspose()

```
LOCAL_INLINE void Crc_Reg_SetReadTranspose (
    uint8 Instance,
    uint8 TransType )
```

Set CRC read transpose mode.

Note

Function ID : DES_CRC_API_313

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
in	<i>TransType</i>	read transpose mode

Returns

void

Definition at line 216 of file AC784xx_Crc_Reg.h.

4.2.2.14 Crc_Reg_SetResultXorMode()

```
LOCAL_INLINE void Crc_Reg_SetResultXorMode (
    uint8 Instance,
    boolean En )
```

Set CRC result xor mode.

Note

Function ID : DES_CRC_API_315

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
in	<i>En</i>	result xor mode enable/disable

Returns

void

Definition at line 239 of file AC784xx_Crc_Reg.h.

4.2.2.15 Crc_Reg_SetSeedOrDataMode()

```
LOCAL_INLINE void Crc_Reg_SetSeedOrDataMode (
    uint8 Instance,
    uint8 Mode )
```

Select Seed or data mode.

Note

Function ID : DES_CRC_API_310

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
in	<i>Mode</i>	mode select

Returns

void

Definition at line 181 of file AC784xx_Crc_Reg.h.

4.2.2.16 Crc_Reg_SetWriteTranspose()

```
LOCAL_INLINE void Crc_Reg_SetWriteTranspose (
    uint8 Instance,
    uint8 TransType )
```

Set CRC write transpose mode.

Note

Function ID : DES_CRC_API_311

Parameters

in	<i>Instance</i>	CRC Hardware Device instance
in	<i>TransType</i>	write transpose mode

Returns

void

Definition at line 193 of file AC784xx_Crc_Reg.h.

4.3 Crc_Hal.c File Reference

This file provides extern Crc Hal API implement.

```
#include "Ckgen_Hal.h"
#include "Dma_Hal.h"
#include "Crc_Hal.h"
#include "AC784xx_Crc_Reg.h"
```

Macros

- #define [CRC_TABLE16_SIZE](#) (16U)
CRC calculate table size 16.
- #define [CRC_TABLE256_SIZE](#) (256U)
CRC calculate table size 256.
- #define [CRC_POLYNOMIAL8](#) 0x1DU
SAE J1850 CRC8 polynomial.
- #define [CRC_POLYNOMIAL8H2F](#) 0x2FU
CRC8 0x2F polynomial.
- #define [CRC_POLYNOMIAL16](#) 0x1021U
definition of key width CRC16 polynomial [SWS_Crc_00002]
- #define [CRC_POLYNOMIAL16ARC](#) 0xA001U
definition of key width CRC16ARC polynomial [SWS_Crc_00002] In there, the polynomial 0x8005U is specified to be used.
- #define [CRC_POLYNOMIAL32](#) 0xEDB88320U
definition of key width CRC32 polynomial [SWS_Crc_00002]
- #define [CRC_POLYNOMIAL32P4](#) 0xC8DF352FU
definition of key width CRC32P4 polynomial [SWS_Crc_00002]
- #define [CRC_POLYNOMIAL64](#) 0xC96C5795D7870F42ULL
definition of key width CRC64 polynomial [SWS_Crc_00062]

Functions

- void [Crc_Hal_Init](#) (uint8 Instance, const [Crc_ConfigType](#) *ConfigPtr)
This function initializes the driver.
- void [Crc_Hal_Deinit](#) (void)
This function deinitializes the driver.
- uint32 [Crc_Hal_CalculateCRC](#) (uint8 Instance, const uint8 *DataPtr, uint32 Length)
Appends a block of bytes to the current CRC calculation.
- Hal_StatusType [Crc_Hal_DmaCalculateCRC](#) (uint8 Channel, const uint8 *DataPtr, uint32 Length, const void *Params)
Appends a block of bytes to the current CRC calculation using dma.
- uint32 [Crc_Hal_GetCRCResult](#) (uint8 Instance)
Gets the current result of the CRC32/CRC16 calculation.
- void [Crc_Hal_SetSeed](#) (uint8 Instance, uint32 Seed)
Sets seed value for CRC module.
- Hal_StatusType [Crc_Hal_GetConfig](#) (uint8 Instance, [Crc_ConfigType](#) *ConfigPtr)
Gets the configuration structure of the CRC module currently.
- uint8 [Crc_Hal_CalculateCRC8](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint8 Crc_StartValue8, boolean Crc_IsFirstCall, uint8 Mode)
CRC8 caculate function with software.
- uint8 [Crc_Hal_CalculateCRC8H2F](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint8 Crc_StartValue8H2F, boolean Crc_IsFirstCall, uint8 Mode)
CRC8H2F caculate function with software.
- uint16 [Crc_Hal_CalculateCRC16](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint16 Crc_StartValue16, boolean Crc_IsFirstCall, uint8 Mode)

CRC16 caculate function with software.

- uint16 [Crc_Hal_CalculateCRC16ARC](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint16 Crc_StartValue16, boolean Crc_IsFirstCall, uint8 Mode)

CRC16ARC caculate function with software.

- uint32 [Crc_Hal_CalculateCRC32](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint32 Crc_StartValue32, boolean Crc_IsFirstCall, uint8 Mode)

CRC32 caculate function with software.

- uint32 [Crc_Hal_CalculateCRC32P4](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint32 Crc_StartValue32, boolean Crc_IsFirstCall, uint8 Mode)

CRC32P4 caculate function with software.

- uint64 [Crc_Hal_CalculateCRC64](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint64 Crc_StartValue64, boolean Crc_IsFirstCall, uint8 Mode)

CRC64 caculate function with software.

4.3.1 Detailed Description

This file provides extern Crc Hal API implement.

4.3.2 Macro Definition Documentation

4.3.2.1 CRC_POLYNOMIAL16

```
#define CRC_POLYNOMIAL16 0x1021U
```

definition of key width CRC16 polynomial [SWS_Crc_00002]

Definition at line 65 of file Crc_Hal.c.

4.3.2.2 CRC_POLYNOMIAL16ARC

```
#define CRC_POLYNOMIAL16ARC 0xA001U
```

definition of key width CRC16ARC polynomial [SWS_Crc_00002] In there, the polynomial 0x8005U is specified to be used.

In that standard, the reflection of all input bytes is specified. We use an optimized algorithm where we do not reflect the input but the polynomial. So the polynomial 0xA001 specified below is the reflected polynomial of the polynomial 0x8005U.

Definition at line 76 of file Crc_Hal.c.

4.3.2.3 CRC_POLYNOMIAL32

```
#define CRC_POLYNOMIAL32 0xEDB88320U
```

definition of key width CRC32 polynomial [SWS_Crc_00002]

The CRC32 routine is based on IEEE-802.3 CRC32 Ethernet standard. In there, the polynomial 0x04C11DB7 is specified to be used.

In that standard, the reflection of all input bytes is specified. We use an optimized algorithm where we do not reflect the input but the polynomial. So the polynomial 0xEDB88320 specified below is the reflected polynomial of the polynomial 0x04C11DB7.

See the "A Painless Guide to CRC Error Detection Algorithms", R. Williams, 1993.

Definition at line 91 of file Crc_Hal.c.

4.3.2.4 CRC_POLYNOMIAL32P4

```
#define CRC_POLYNOMIAL32P4 0xC8DF352FU
```

definition of key width CRC32P4 polynomial [SWS_Crc_00002]

In there, the polynomial 0xF4ACFB13 is specified to be used.

In that standard, the reflection of all input bytes is specified. We use an optimized algorithm where we do not reflect the input but the polynomial. So the polynomial 0xC8DF352F specified below is the reflected polynomial of the polynomial 0xF4ACFB13.

See the "A Painless Guide to CRC Error Detection Algorithms", R. Williams, 1993.

Definition at line 105 of file Crc_Hal.c.

4.3.2.5 CRC_POLYNOMIAL64

```
#define CRC_POLYNOMIAL64 0xC96C5795D7870F42ULL
```

definition of key width CRC64 polynomial [SWS_Crc_00062]

The CRC64 routine is based on CRC-64-ECMA standard. In there, the polynomial 0x42F0E1EBA9EA3693 is specified to be used.

In that standard, the reflection of all input bytes is specified. We use an optimized algorithm where we do not reflect the input but the polynomial. So the polynomial 0xC96C5795D7870F42 specified below is the reflected polynomial of the polynomial 0x42F0E1EBA9EA3693.

See the "A Painless Guide to CRC Error Detection Algorithms", R. Williams, 1993.

Definition at line 120 of file Crc_Hal.c.

4.3.2.6 CRC_POLYNOMIAL8

```
#define CRC_POLYNOMIAL8 0x1DU
```

SAE J1850 CRC8 polynomial.

According to AUTOSAR R4.0 CRC SWS SWS_Crc_00030

Definition at line 59 of file Crc_Hal.c.

4.3.2.7 CRC_POLYNOMIAL8H2F

```
#define CRC_POLYNOMIAL8H2F 0x2FU
```

CRC8 0x2F polynomial.

Definition at line 62 of file Crc_Hal.c.

4.3.2.8 CRC_TABLE16_SIZE

```
#define CRC_TABLE16_SIZE (16U)
```

CRC calculate table size 16.

Definition at line 52 of file Crc_Hal.c.

4.3.2.9 CRC_TABLE256_SIZE

```
#define CRC_TABLE256_SIZE (256U)
```

CRC calculate table size 256.

Definition at line 54 of file Crc_Hal.c.

4.3.3 Function Documentation

4.3.3.1 Crc_Hal_CalculateCRC()

```
uint32 Crc_Hal_CalculateCRC (  
    uint8 Instance,  
    const uint8 * DataPtr,  
    uint32 Length )
```

Appends a block of bytes to the current CRC calculation.

Note

Function ID : DES_CRC_API_202

Parameters

in	<i>Instance</i>	CRC Hardware Device instance.
in	<i>DataPtr</i>	The Pointer to the data array
in	<i>Length</i>	Number of the data array

Returns

CRC calculate result.

Definition at line 529 of file Crc_Hal.c.

4.3.3.2 Crc_Hal_CalculateCRC16()

```
uint16 Crc_Hal_CalculateCRC16 (  
    const uint8 * Crc_DataPtr,  
    uint32 Crc_Length,  
    uint16 Crc_StartValue16,  
    boolean Crc_IsFirstCall,  
    uint8 Mode )
```

CRC16 caculate function with software.

Note

Function ID : DES_CRC_API_209

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue16</i>	The CRC16 start value.
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

CRC16 result.

Definition at line 904 of file Crc_Hal.c.

4.3.3.3 Crc_Hal_CalculateCRC16ARC()

```
uint16 Crc_Hal_CalculateCRC16ARC (  
    const uint8 * Crc_DataPtr,  
    uint32 Crc_Length,
```

```
uint16 Crc_StartValue16,
boolean Crc_IsFirstCall,
uint8 Mode )
```

CRC16ARC caculate function with software.

Note

Function ID : DES_CRC_API_210

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue16</i>	The CRC16ARC start value.
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

CRC16ARC result.

Definition at line 998 of file Crc_Hal.c.

4.3.3.4 Crc_Hal_CalculateCRC32()

```
uint32 Crc_Hal_CalculateCRC32 (
    const uint8 * Crc_DataPtr,
    uint32 Crc_Length,
    uint32 Crc_StartValue32,
    boolean Crc_IsFirstCall,
    uint8 Mode )
```

CRC32 caculate function with software.

Note

Function ID : DES_CRC_API_211

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue32</i>	The CRC32 start value.
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

CRC32 result.

Definition at line 1090 of file Crc_Hal.c.

4.3.3.5 Crc_Hal_CalculateCRC32P4()

```
uint32 Crc_Hal_CalculateCRC32P4 (
    const uint8 * Crc_DataPtr,
    uint32 Crc_Length,
    uint32 Crc_StartValue32,
    boolean Crc_IsFirstCall,
    uint8 Mode )
```

CRC32P4 caculate function with software.

Note

Function ID : DES_CRC_API_212

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue32</i>	The CRC32P4 start value.
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

CRC32P4 result.

Definition at line 1200 of file Crc_Hal.c.

4.3.3.6 Crc_Hal_CalculateCRC64()

```
uint64 Crc_Hal_CalculateCRC64 (
    const uint8 * Crc_DataPtr,
    uint32 Crc_Length,
    uint64 Crc_StartValue64,
    boolean Crc_IsFirstCall,
    uint8 Mode )
```

CRC64 caculate function with software.

Note

Function ID : DES_CRC_API_213

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue64</i>	The CRC64 start value.
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

CRC64 result.

Definition at line 1307 of file Crc_Hal.c.

4.3.3.7 Crc_Hal_CalculateCRC8()

```
uint8 Crc_Hal_CalculateCRC8 (
    const uint8 * Crc_DataPtr,
    uint32 Crc_Length,
    uint8 Crc_StartValue8,
    boolean Crc_IsFirstCall,
    uint8 Mode )
```

CRC8 caculate function with software.

Note

Function ID : DES_CRC_API_207

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue8</i>	The CRC8 start value
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

Crc8 result.

Definition at line 711 of file Crc_Hal.c.

4.3.3.8 Crc_Hal_CalculateCRC8H2F()

```
uint8 Crc_Hal_CalculateCRC8H2F (
    const uint8 * Crc_DataPtr,
    uint32 Crc_Length,
    uint8 Crc_StartValue8H2F,
    boolean Crc_IsFirstCall,
    uint8 Mode )
```

CRC8H2F caculate function with software.

Note

Function ID : DES_CRC_API_208

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue8H2F</i>	The CRC8H2F start value
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

Crc8H2F result.

Definition at line 809 of file Crc_Hal.c.

4.3.3.9 Crc_Hal_Deinit()

```
void Crc_Hal_Deinit (
    void )
```

This function deinitializes the driver.

Note

Function ID : DES_CRC_API_206

Returns

void

Definition at line 514 of file Crc_Hal.c.

4.3.3.10 Crc_Hal_DmaCalculateCRC()

```
Hal_StatusType Crc_Hal_DmaCalculateCRC (
    uint8 Channel,
    const uint8 * DataPtr,
    uint32 Length,
    const void * Params )
```

Appends a block of bytes to the current CRC calculation using dma.

Note

Function ID : DES_CRC_API_203

Parameters

in	<i>Channel</i>	Dma Channel Id
in	<i>DataPtr</i>	The Pointer to the data array
in	<i>Length</i>	Number of the data array
in	<i>Params</i>	Dma callback parametes delive to callback (not used currently)

Returns

call dma success or error.

Definition at line 581 of file Crc_Hal.c.

4.3.3.11 Crc_Hal_GetConfig()

```
Hal_StatusType Crc_Hal_GetConfig (
    uint8 Instance,
    Crc_ConfigType * ConfigPtr )
```

Gets the configuration structure of the CRC module currently.

Note

Function ID : DES_CRC_API_205

Parameters

in	<i>Instance</i>	The CRC instance number
out	<i>ConfigPtr</i>	Pointer to structure of CRC configuration

Returns

The result of execution

- STATUS_SUCCESS: Operation was successful
- STATUS_ERROR: Operation was successful

Definition at line 661 of file Crc_Hal.c.

4.3.3.12 Crc_Hal_GetCRCResult()

```
uint32 Crc_Hal_GetCRCResult (
    uint8 Instance )
```

Gets the current result of the CRC32/CRC16 calculation.

Note

Function ID : DES_CRC_API_204

Parameters

in	<i>Instance</i>	The CRC instance number
----	-----------------	-------------------------

Returns

Result of CRC32/CRC16 calculation

Definition at line 621 of file Crc_Hal.c.

4.3.3.13 Crc_Hal_Init()

```
void Crc_Hal_Init (
    uint8 Instance,
    const Crc_ConfigType * ConfigPtr )
```

This function initializes the driver.

Note

Function ID : DES_CRC_API_201

Parameters

in	<i>Instance</i>	CRC Hardware Device instance.
in	<i>ConfigPtr</i>	Pointer to a selected configuration structure

Returns

void.

Definition at line 478 of file Crc_Hal.c.

4.3.3.14 Crc_Hal_SetSeed()

```
void Crc_Hal_SetSeed (
    uint8 Instance,
    uint32 Seed )
```

Sets seed value for CRC module.

Note

Function ID : DES_CRC_API_214

Parameters

in	<i>Instance</i>	The CRC instance number
in	<i>Seed</i>	New seed data for CRC module

Returns

void

Definition at line 642 of file Crc_Hal.c.

4.4 Crc_Hal.h File Reference

This file provides extern Crc Hal API.

```
#include "Crc_Hal_Types.h"
```

Functions

- void [Crc_Hal_Init](#) (uint8 Instance, const [Crc_ConfigType](#) *ConfigPtr)
This function initializes the driver.
- void [Crc_Hal_Deinit](#) (void)
This function deinitializes the driver.
- uint32 [Crc_Hal_CalculateCRC](#) (uint8 Instance, const uint8 *DataPtr, uint32 Length)
Appends a block of bytes to the current CRC calculation.
- Hal_StatusType [Crc_Hal_DmaCalculateCRC](#) (uint8 Channel, const uint8 *DataPtr, uint32 Length, const void *Params)
Appends a block of bytes to the current CRC calculation using dma.
- uint32 [Crc_Hal_GetCRCResult](#) (uint8 Instance)
Gets the current result of the CRC32/CRC16 calculation.
- void [Crc_Hal_SetSeed](#) (uint8 Instance, uint32 Seed)
Sets seed value for CRC module.
- Hal_StatusType [Crc_Hal_GetConfig](#) (uint8 Instance, [Crc_ConfigType](#) *ConfigPtr)
Gets the configuration structure of the CRC module currently.
- uint8 [Crc_Hal_CalculateCRC8](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint8 Crc_StartValue8, boolean Crc_IsFirstCall, uint8 Mode)
CRC8 caculate function with software.

- uint8 [Crc_Hal_CalculateCRC8H2F](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint8 Crc_StartValue8H2F, boolean Crc_IsFirstCall, uint8 Mode)
CRC8H2F caculate function with software.
- uint16 [Crc_Hal_CalculateCRC16](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint16 Crc_StartValue16, boolean Crc_IsFirstCall, uint8 Mode)
CRC16 caculate function with software.
- uint16 [Crc_Hal_CalculateCRC16ARC](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint16 Crc_StartValue16, boolean Crc_IsFirstCall, uint8 Mode)
CRC16ARC caculate function with software.
- uint32 [Crc_Hal_CalculateCRC32](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint32 Crc_StartValue32, boolean Crc_IsFirstCall, uint8 Mode)
CRC32 caculate function with software.
- uint32 [Crc_Hal_CalculateCRC32P4](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint32 Crc_StartValue32, boolean Crc_IsFirstCall, uint8 Mode)
CRC32P4 caculate function with software.
- uint64 [Crc_Hal_CalculateCRC64](#) (const uint8 *Crc_DataPtr, uint32 Crc_Length, uint64 Crc_StartValue64, boolean Crc_IsFirstCall, uint8 Mode)
CRC64 caculate function with software.

4.4.1 Detailed Description

This file provides extern Crc Hal API.

4.4.2 Function Documentation

4.4.2.1 Crc_Hal_CalculateCRC()

```
uint32 Crc_Hal_CalculateCRC (
    uint8 Instance,
    const uint8 * DataPtr,
    uint32 Length )
```

Appends a block of bytes to the current CRC calculation.

Note

Function ID : DES_CRC_API_202

Parameters

in	<i>Instance</i>	CRC Hardware Device instance.
in	<i>DataPtr</i>	The Pointer to the data array
in	<i>Length</i>	Number of the data array

Returns

CRC calculate result.

Definition at line 529 of file Crc_Hal.c.

4.4.2.2 Crc_Hal_CalculateCRC16()

```
uint16 Crc_Hal_CalculateCRC16 (  
    const uint8 * Crc_DataPtr,  
    uint32 Crc_Length,  
    uint16 Crc_StartValue16,  
    boolean Crc_IsFirstCall,  
    uint8 Mode )
```

CRC16 caculate function with software.

Note

Function ID : DES_CRC_API_209

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue16</i>	The CRC16 start value.
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

CRC16 result.

Definition at line 904 of file Crc_Hal.c.

4.4.2.3 Crc_Hal_CalculateCRC16ARC()

```
uint16 Crc_Hal_CalculateCRC16ARC (  
    const uint8 * Crc_DataPtr,  
    uint32 Crc_Length,  
    uint16 Crc_StartValue16,  
    boolean Crc_IsFirstCall,  
    uint8 Mode )
```

CRC16ARC caculate function with software.

Note

Function ID : DES_CRC_API_210

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue16</i>	The CRC16ARC start value.
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

CRC16ARC result.

Definition at line 998 of file Crc_Hal.c.

4.4.2.4 Crc_Hal_CalculateCRC32()

```
uint32 Crc_Hal_CalculateCRC32 (
    const uint8 * Crc_DataPtr,
    uint32 Crc_Length,
    uint32 Crc_StartValue32,
    boolean Crc_IsFirstCall,
    uint8 Mode )
```

CRC32 caculate function with software.

Note

Function ID : DES_CRC_API_211

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue32</i>	The CRC32 start value.
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

CRC32 result.

Definition at line 1090 of file Crc_Hal.c.

4.4.2.5 Crc_Hal_CalculateCRC32P4()

```
uint32 Crc_Hal_CalculateCRC32P4 (
    const uint8 * Crc_DataPtr,
    uint32 Crc_Length,
    uint32 Crc_StartValue32,
    boolean Crc_IsFirstCall,
    uint8 Mode )
```

CRC32P4 caculate function with software.

Note

Function ID : DES_CRC_API_212

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue32</i>	The CRC32P4 start value.
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

CRC32P4 result.

Definition at line 1200 of file Crc_Hal.c.

4.4.2.6 Crc_Hal_CalculateCRC64()

```
uint64 Crc_Hal_CalculateCRC64 (
    const uint8 * Crc_DataPtr,
    uint32 Crc_Length,
    uint64 Crc_StartValue64,
    boolean Crc_IsFirstCall,
    uint8 Mode )
```

CRC64 caculate function with software.

Note

Function ID : DES_CRC_API_213

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue64</i>	The CRC64 start value.
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

CRC64 result.

Definition at line 1307 of file Crc_Hal.c.

4.4.2.7 Crc_Hal_CalculateCRC8()

```
uint8 Crc_Hal_CalculateCRC8 (
    const uint8 * Crc_DataPtr,
    uint32 Crc_Length,
    uint8 Crc_StartValue8,
    boolean Crc_IsFirstCall,
    uint8 Mode )
```

CRC8 caculate function with software.

Note

Function ID : DES_CRC_API_207

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	caculate crc data length
in	<i>Crc_StartValue8</i>	The CRC8 start value
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

Crc8 result.

Definition at line 711 of file Crc_Hal.c.

4.4.2.8 Crc_Hal_CalculateCRC8H2F()

```
uint8 Crc_Hal_CalculateCRC8H2F (
    const uint8 * Crc_DataPtr,
    uint32 Crc_Length,
    uint8 Crc_StartValue8H2F,
    boolean Crc_IsFirstCall,
    uint8 Mode )
```

CRC8H2F caculate function with software.

Note

Function ID : DES_CRC_API_208

Parameters

in	<i>Crc_DataPtr</i>	The pointer to data block.
in	<i>Crc_Length</i>	calculate crc data length
in	<i>Crc_StartValue8H2F</i>	The CRC8H2F start value
in	<i>Crc_IsFirstCall</i>	check it is first call or not.
in	<i>Mode</i>	Crc Calculate Mode -CRC_TABLE_16_BYTE_MODE -CRC_TABLE_256_BYTE_MODE -CRC_RUNTIME_MODE

Returns

Crc8H2F result.

Definition at line 809 of file Crc_Hal.c.

4.4.2.9 Crc_Hal_Deinit()

```
void Crc_Hal_Deinit (
    void )
```

This function deinitializes the driver.

Note

Function ID : DES_CRC_API_206

Returns

void

Definition at line 514 of file Crc_Hal.c.

4.4.2.10 Crc_Hal_DmaCalculateCRC()

```
Hal_StatusType Crc_Hal_DmaCalculateCRC (
    uint8 Channel,
    const uint8 * DataPtr,
    uint32 Length,
    const void * Params )
```

Appends a block of bytes to the current CRC calculation using dma.

Note

Function ID : DES_CRC_API_203

Parameters

in	<i>Channel</i>	Dma Channel Id
in	<i>DataPtr</i>	The Pointer to the data array
in	<i>Length</i>	Number of the data array
in	<i>Params</i>	Dma callback parameters delivered to callback (not used currently)

Returns

call dma success or error.

Definition at line 581 of file Crc_Hal.c.

4.4.2.11 Crc_Hal_GetConfig()

```
Hal_StatusType Crc_Hal_GetConfig (  
    uint8 Instance,  
    Crc_ConfigType * ConfigPtr )
```

Gets the configuration structure of the CRC module currently.

Note

Function ID : DES_CRC_API_205

Parameters

in	<i>Instance</i>	The CRC instance number
out	<i>ConfigPtr</i>	Pointer to structure of CRC configuration

Returns

The result of execution

- STATUS_SUCCESS: Operation was successful
- STATUS_ERROR: Operation was successful

Definition at line 661 of file Crc_Hal.c.

4.4.2.12 Crc_Hal_GetCRCResult()

```
uint32 Crc_Hal_GetCRCResult (  
    uint8 Instance )
```

Gets the current result of the CRC32/CRC16 calculation.

Note

Function ID : DES_CRC_API_204

Parameters

in	<i>Instance</i>	The CRC instance number
----	-----------------	-------------------------

Returns

Result of CRC32/CRC16 calculation

Definition at line 621 of file Crc_Hal.c.

4.4.2.13 Crc_Hal_Init()

```
void Crc_Hal_Init (
    uint8 Instance,
    const Crc_ConfigType * ConfigPtr )
```

This function initializes the driver.

Note

Function ID : DES_CRC_API_201

Parameters

in	<i>Instance</i>	CRC Hardware Device instance.
in	<i>ConfigPtr</i>	Pointer to a selected configuration structure

Returns

void.

Definition at line 478 of file Crc_Hal.c.

4.4.2.14 Crc_Hal_SetSeed()

```
void Crc_Hal_SetSeed (
    uint8 Instance,
    uint32 Seed )
```

Sets seed value for CRC module.

Note

Function ID : DES_CRC_API_214

Parameters

in	<i>Instance</i>	The CRC instance number
in	<i>Seed</i>	New seed data for CRC module

Returns

void

Definition at line 642 of file Crc_Hal.c.

4.5 Crc_Hal_Types.h File Reference

This file provides extern Crc macro enum and structure info for hal/mcal.

```
#include "Device_Register.h"
```

Classes

- struct [Crc_ConfigType](#)
CRC configuration structure.

Macros

- #define [CRC_TABLE_16_BYTE_MODE](#) 0x0U
Support for caculate method. caculate method by define.
- #define [CRC_TABLE_256_BYTE_MODE](#) 0x1U
- #define [CRC_RUNTIME_MODE](#) 0x2U
- #define [CRC_HARDWARE_MODE](#) 0x3U
- #define [CRC_INITIAL_VALUE8](#) 0xFFU
Definition of the initial value of the SAE J1850 CRC8.
- #define [CRC_INITIAL_VALUE8H2F](#) 0xFFU
Definition of the initial value of the CRC8 on polynom 0x2F.
- #define [CRC_INITIAL_VALUE16](#) 0xFFFFU
Definition of the initial value of crc16.
- #define [CRC_INITIAL_VALUE16ARC](#) (0x0U)
Definition of the initial value of crc16.
- #define [CRC_INITIAL_VALUE32](#) 0xFFFFFFFFU
Definition of the initial value of crc32.
- #define [CRC_INITIAL_VALUE64](#) 0xFFFFFFFFFFFFFFFFFULL
Definition of the initial value of crc64.
- #define [CRC_DATA_IS_DATA](#) (0x0U)
DATA register is data.
- #define [CRC_DATA_IS_SEED](#) (0x1U)
DATA register is Seed.

Enumerations

- enum `Crc_ProtocolType` { `CRC_PROTOCOL_16BIT` = 0x00U, `CRC_PROTOCOL_32BIT` }
CRC protocol enum.
- enum `Crc_TransposeType` { `CRC_TRANSPOSE_NONE` = 0x00U, `CRC_TRANSPOSE_BITS`, `CRC_TRANSPOSE_BYTES`, `CRC_TRANSPOSE_BITS_BYTES`, `CRC_TRANSPOSE_BYTES` }
CRC transpose enum.

4.5.1 Detailed Description

This file provides extern Crc macro enum and structure info for hal/mcal.

4.5.2 Macro Definition Documentation

4.5.2.1 CRC_DATA_IS_DATA

```
#define CRC_DATA_IS_DATA (0x0U)
```

DATA register is data.

Definition at line 86 of file Crc_Hal_Types.h.

4.5.2.2 CRC_DATA_IS_SEED

```
#define CRC_DATA_IS_SEED (0x1U)
```

DATA register is Seed.

Definition at line 88 of file Crc_Hal_Types.h.

4.5.2.3 CRC_HARDWARE_MODE

```
#define CRC_HARDWARE_MODE 0x3U
```

Definition at line 64 of file Crc_Hal_Types.h.

4.5.2.4 CRC_INITIAL_VALUE16

```
#define CRC_INITIAL_VALUE16 0xFFFFU
```

Definition of the initial value of crc16.

Definition at line 74 of file Crc_Hal_Types.h.

4.5.2.5 CRC_INITIAL_VALUE16ARC

```
#define CRC_INITIAL_VALUE16ARC (0x0U)
```

Definition of the initial value of crc16.

Definition at line 77 of file Crc_Hal_Types.h.

4.5.2.6 CRC_INITIAL_VALUE32

```
#define CRC_INITIAL_VALUE32 0xFFFFFFFFU
```

Definition of the initial value of crc32.

Definition at line 80 of file Crc_Hal_Types.h.

4.5.2.7 CRC_INITIAL_VALUE64

```
#define CRC_INITIAL_VALUE64 0xFFFFFFFFFFFFFFFFULL
```

Definition of the initial value of crc64.

Definition at line 83 of file Crc_Hal_Types.h.

4.5.2.8 CRC_INITIAL_VALUE8

```
#define CRC_INITIAL_VALUE8 0xFFU
```

Definition of the initial value of the SAE J1850 CRC8.

Definition at line 68 of file Crc_Hal_Types.h.

4.5.2.9 CRC_INITIAL_VALUE8H2F

```
#define CRC_INITIAL_VALUE8H2F 0xFFU
```

Definition of the initial value of the CRC8 on polynom 0x2F.

Definition at line 71 of file Crc_Hal_Types.h.

4.5.2.10 CRC_RUNTIME_MODE

```
#define CRC_RUNTIME_MODE 0x2U
```

Definition at line 63 of file Crc_Hal_Types.h.

4.5.2.11 CRC_TABLE_16_BYTE_MODE

```
#define CRC_TABLE_16_BYTE_MODE 0x0U
```

Support for caculate method. caculate method by define.

Definition at line 61 of file Crc_Hal_Types.h.

4.5.2.12 CRC_TABLE_256_BYTE_MODE

```
#define CRC_TABLE_256_BYTE_MODE 0x1U
```

Definition at line 62 of file Crc_Hal_Types.h.

4.5.3 Enumeration Type Documentation

4.5.3.1 Crc_ProtocolType

```
enum Crc_ProtocolType
```

CRC protocol enum.

Enumerator

CRC_PROTOCOL_16BIT	CRC Protocol 16bit mode
CRC_PROTOCOL_32BIT	CRC Protocol 32bit mode

Definition at line 91 of file Crc_Hal_Types.h.

4.5.3.2 Crc_TransposeType

```
enum Crc_TransposeType
```

CRC transpose enum.

Enumerator

CRC_TRANSPOSE_NONE	CRC write in without tranpose
CRC_TRANSPOSE_BITS	CRC write in with bits transpose
CRC_TRANSPOSE_BITS_BYTES	CRC write in with bits and bytes transpose
CRC_TRANSPOSE_BYTES	CRC write in with bytes transpose

Definition at line 98 of file Crc_Hal_Types.h.

Index

AC784xx_API_Reference_Manual_CRC.pdf, [5](#)

AC784xx_Crc_Reg.h, [5](#)

[Crc_Reg_GetData, 6](#)

[Crc_Reg_GetPoly, 6](#)

[Crc_Reg_GetProtocolType, 7](#)

[Crc_Reg_GetReadTranspose, 7](#)

[Crc_Reg_GetResultXorMode, 8](#)

[Crc_Reg_GetWriteTranspose, 8](#)

[Crc_Reg_SetCtrl, 8](#)

[Crc_Reg_SetData, 9](#)

[Crc_Reg_SetDataLL, 10](#)

[Crc_Reg_SetDataL, 9](#)

[Crc_Reg_SetPoly, 10](#)

[Crc_Reg_SetProtocolType, 11](#)

[Crc_Reg_SetReadTranspose, 11](#)

[Crc_Reg_SetResultXorMode, 12](#)

[Crc_Reg_SetSeedOrDataMode, 12](#)

[Crc_Reg_SetWriteTranspose, 13](#)

CRC_DATA_IS_DATA

[Crc_Hal_Types.h, 35](#)

CRC_DATA_IS_SEED

[Crc_Hal_Types.h, 35](#)

CRC_HARDWARE_MODE

[Crc_Hal_Types.h, 35](#)

CRC_INITIAL_VALUE16

[Crc_Hal_Types.h, 35](#)

CRC_INITIAL_VALUE16ARC

[Crc_Hal_Types.h, 35](#)

CRC_INITIAL_VALUE32

[Crc_Hal_Types.h, 36](#)

CRC_INITIAL_VALUE64

[Crc_Hal_Types.h, 36](#)

CRC_INITIAL_VALUE8

[Crc_Hal_Types.h, 36](#)

CRC_INITIAL_VALUE8H2F

[Crc_Hal_Types.h, 36](#)

CRC_POLYNOMIAL16

[Crc_Hal.c, 15](#)

CRC_POLYNOMIAL16ARC

[Crc_Hal.c, 15](#)

CRC_POLYNOMIAL32

[Crc_Hal.c, 15](#)

CRC_POLYNOMIAL32P4

[Crc_Hal.c, 16](#)

CRC_POLYNOMIAL64

[Crc_Hal.c, 16](#)

CRC_POLYNOMIAL8

[Crc_Hal.c, 16](#)

CRC_POLYNOMIAL8H2F

[Crc_Hal.c, 17](#)

CRC_RUNTIME_MODE

[Crc_Hal_Types.h, 36](#)

CRC_TABLE16_SIZE

[Crc_Hal.c, 17](#)

CRC_TABLE256_SIZE

[Crc_Hal.c, 17](#)

CRC_TABLE_16_BYTE_MODE

[Crc_Hal_Types.h, 37](#)

CRC_TABLE_256_BYTE_MODE

[Crc_Hal_Types.h, 37](#)

Crc_ConfigType, [3](#)

[FinalXOR, 3](#)

[Poly, 3](#)

[Protocol, 4](#)

[ReadTranspose, 4](#)

[Seed, 4](#)

[WriteTranspose, 4](#)

Crc_Hal.c, [13](#)

[CRC_POLYNOMIAL16, 15](#)

[CRC_POLYNOMIAL16ARC, 15](#)

[CRC_POLYNOMIAL32, 15](#)

[CRC_POLYNOMIAL32P4, 16](#)

[CRC_POLYNOMIAL64, 16](#)

[CRC_POLYNOMIAL8, 16](#)

[CRC_POLYNOMIAL8H2F, 17](#)

[CRC_TABLE16_SIZE, 17](#)

[CRC_TABLE256_SIZE, 17](#)

[Crc_Hal_CalculateCRC16, 18](#)

[Crc_Hal_CalculateCRC16ARC, 18](#)

[Crc_Hal_CalculateCRC32, 19](#)

[Crc_Hal_CalculateCRC32P4, 20](#)

[Crc_Hal_CalculateCRC64, 20](#)

[Crc_Hal_CalculateCRC8, 21](#)

[Crc_Hal_CalculateCRC8H2F, 21](#)

[Crc_Hal_CalculateCRC, 17](#)

[Crc_Hal_Deinit, 22](#)

[Crc_Hal_DmaCalculateCRC, 22](#)

[Crc_Hal_GetCRCResult, 24](#)

[Crc_Hal_GetConfig, 23](#)

[Crc_Hal_Init, 24](#)

[Crc_Hal_SetSeed, 24](#)

Crc_Hal.h, [25](#)

[Crc_Hal_CalculateCRC16, 27](#)

[Crc_Hal_CalculateCRC16ARC, 27](#)

[Crc_Hal_CalculateCRC32, 28](#)

[Crc_Hal_CalculateCRC32P4, 28](#)

[Crc_Hal_CalculateCRC64, 29](#)

[Crc_Hal_CalculateCRC8, 30](#)

[Crc_Hal_CalculateCRC8H2F, 30](#)

[Crc_Hal_CalculateCRC, 26](#)

[Crc_Hal_Deinit, 31](#)

[Crc_Hal_DmaCalculateCRC, 31](#)

[Crc_Hal_GetCRCResult, 32](#)

- Crc_Hal_GetConfig, 32
- Crc_Hal_Init, 33
- Crc_Hal_SetSeed, 33
- Crc_Hal_CalculateCRC16
 - Crc_Hal.c, 18
 - Crc_Hal.h, 27
- Crc_Hal_CalculateCRC16ARC
 - Crc_Hal.c, 18
 - Crc_Hal.h, 27
- Crc_Hal_CalculateCRC32
 - Crc_Hal.c, 19
 - Crc_Hal.h, 28
- Crc_Hal_CalculateCRC32P4
 - Crc_Hal.c, 20
 - Crc_Hal.h, 28
- Crc_Hal_CalculateCRC64
 - Crc_Hal.c, 20
 - Crc_Hal.h, 29
- Crc_Hal_CalculateCRC8
 - Crc_Hal.c, 21
 - Crc_Hal.h, 30
- Crc_Hal_CalculateCRC8H2F
 - Crc_Hal.c, 21
 - Crc_Hal.h, 30
- Crc_Hal_CalculateCRC
 - Crc_Hal.c, 17
 - Crc_Hal.h, 26
- Crc_Hal_Deinit
 - Crc_Hal.c, 22
 - Crc_Hal.h, 31
- Crc_Hal_DmaCalculateCRC
 - Crc_Hal.c, 22
 - Crc_Hal.h, 31
- Crc_Hal_GetCRCResult
 - Crc_Hal.c, 24
 - Crc_Hal.h, 32
- Crc_Hal_GetConfig
 - Crc_Hal.c, 23
 - Crc_Hal.h, 32
- Crc_Hal_Init
 - Crc_Hal.c, 24
 - Crc_Hal.h, 33
- Crc_Hal_SetSeed
 - Crc_Hal.c, 24
 - Crc_Hal.h, 33
- Crc_Hal_Types.h, 34
 - CRC_DATA_IS_DATA, 35
 - CRC_DATA_IS_SEED, 35
 - CRC_HARDWARE_MODE, 35
 - CRC_INITIAL_VALUE16, 35
 - CRC_INITIAL_VALUE16ARC, 35
 - CRC_INITIAL_VALUE32, 36
 - CRC_INITIAL_VALUE64, 36
 - CRC_INITIAL_VALUE8, 36
 - CRC_INITIAL_VALUE8H2F, 36
 - CRC_RUNTIME_MODE, 36
 - CRC_TABLE_16_BYTE_MODE, 37
 - CRC_TABLE_256_BYTE_MODE, 37
 - Crc_ProtocolType, 37
 - Crc_TransposeType, 37
- Crc_ProtocolType
- Crc_Reg_GetData
 - AC784xx_Crc_Reg.h, 6
- Crc_Reg_GetPoly
 - AC784xx_Crc_Reg.h, 6
- Crc_Reg_GetProtocolType
 - AC784xx_Crc_Reg.h, 7
- Crc_Reg_GetReadTranspose
 - AC784xx_Crc_Reg.h, 7
- Crc_Reg_GetResultXorMode
 - AC784xx_Crc_Reg.h, 8
- Crc_Reg_GetWriteTranspose
 - AC784xx_Crc_Reg.h, 8
- Crc_Reg_SetCtrl
 - AC784xx_Crc_Reg.h, 8
- Crc_Reg_SetData
 - AC784xx_Crc_Reg.h, 9
- Crc_Reg_SetDataLL
 - AC784xx_Crc_Reg.h, 10
- Crc_Reg_SetDataL
 - AC784xx_Crc_Reg.h, 9
- Crc_Reg_SetPoly
 - AC784xx_Crc_Reg.h, 10
- Crc_Reg_SetProtocolType
 - AC784xx_Crc_Reg.h, 11
- Crc_Reg_SetReadTranspose
 - AC784xx_Crc_Reg.h, 11
- Crc_Reg_SetResultXorMode
 - AC784xx_Crc_Reg.h, 12
- Crc_Reg_SetSeedOrDataMode
 - AC784xx_Crc_Reg.h, 12
- Crc_Reg_SetWriteTranspose
 - AC784xx_Crc_Reg.h, 13
- Crc_TransposeType
 - Crc_Hal_Types.h, 37
- FinalXOR
 - Crc_ConfigType, 3
- Poly
 - Crc_ConfigType, 3
- Protocol
 - Crc_ConfigType, 4
- ReadTranspose
 - Crc_ConfigType, 4
- Seed
 - Crc_ConfigType, 4
- WriteTranspose
 - Crc_ConfigType, 4