

# **FINTEK**

## **F81601A**

### **FD and CAN 2.0 Series**

## **Windows Software Programming Guide**

V1.04

July 23, 2024

7/23/2024

## Contents

1.	Fintek CAN FD DLL Control APIs .....	3
1.1	Support Fintek CAN FD IC .....	3
1.2	Function List .....	3
1.2.1	FintekCanbusFd_Open .....	4
1.2.2	FintekCanbusFd_Close .....	5
1.2.3	FintekCanbusFd_SetBaudRate .....	5
1.2.4	FintekCanbusFd_GetBaudRate .....	6
1.2.5	FintekCanbusFd_SetId .....	7
1.2.6	FintekCanbusFd_SetFilter .....	7
1.2.7	FintekCanbusFd_ClearFilter .....	8
1.2.8	FintekCanbusFd_SetAcrAmrFilter .....	8
1.2.9	FintekCanbusFd_SetErrorFilter .....	12
1.2.10	FintekCanbusFd_GetErrorFilter .....	12
1.2.11	FintekCanbusFd_Setler .....	13
1.2.12	FintekCanbusFd_Getler .....	13
1.2.13	FintekCanbusFd_GetErrorCode .....	14
1.2.14	FintekCanbusFd_Send .....	14
1.2.15	FintekCanbusFd_Receive .....	15
1.2.16	FintekCanbusFd_ReceiveEx .....	16
1.2.17	FintekCanbusFd_GetIdVal .....	17
1.2.18	FintekCanbusFd_SetMode .....	17
1.2.19	FintekCanbusFd_GetMode .....	17
1.2.20	FintekCanbusFd_Reset .....	18
1.2.21	FintekCanbusFd_Create .....	18
1.2.22	FintekCanbusFd_Delete .....	19
1.2.23	FintekCanbusFd_Start .....	19
1.2.24	FintekCanbusFd_Stop .....	19
2.	Fintek CAN2.0 DLL Control APIs .....	20
2.1	Support Fintek CANBus IC .....	20
2.2	Function List .....	20
2.2.1	FintekCanbus_Open .....	21
2.2.2	FintekCanbus_Close .....	22
2.2.3	FintekCanbus_SetBaudRate .....	22
2.2.4	FintekCanbus_GetBaudRate .....	23
2.2.5	FintekCanbus_SetId .....	23
2.2.6	FintekCanbus_SetFilter .....	24
2.2.7	FintekCanbus_ClearFilter .....	24
2.2.8	FintekCanbus_SetAcrAmrFilter .....	24
2.2.9	FintekCanbus_SetErrorFilter .....	29
2.2.10	FintekCanbus_GetErrorFilter .....	29
2.2.11	FintekCanbus_Setler .....	30
2.2.12	FintekCanbus_Getler .....	30
2.2.13	FintekCanbus_GetErrorCode .....	31
2.2.14	FintekCanbus_Send .....	31

2.2.15	FintekCanbus_Receive .....	32
2.2.16	FintekCanbus_ReceiveEx .....	33
2.2.17	FintekCanbus_SetMode .....	33
2.2.18	FintekCanbus_GetMode .....	34
2.2.19	FintekCanbus_Reset .....	34
2.2.20	FintekCanbus_Create .....	35
2.2.21	FintekCanbus_Delete .....	35
2.2.22	FintekCanbus_Start .....	35
2.2.23	FintekCanbus_Stop .....	36
3.	Error Codes .....	37

# 1. Fintek CAN FD DLL Control APIs

## 1.1 Support Fintek CAN FD IC

- F81601a

## 1.2 Function List

This section provides the specifications of all Fintek CAN FD functions and structures. All APIs use the naming convention FintekCanbusFd\_xxx specific to below table:

<i>ID</i>	<i>Function Name</i>
1.2.1	FintekCanbusFd_Open
1.2.2	FintekCanbusFd_Close
1.2.3	FintekCanbusFd_SetBaudRate
1.2.4	FintekCanbusFd_GetBaudRate
1.2.5	FintekCanbusFd_SetId
1.2.6	FintekCanbusFd_SetFilter
1.2.7	FintekCanbusFd_ClearFilter
1.2.8	FintekCanbusFd_SetAcrAmrFilter
1.2.9	FintekCanbusFd_SetErrorFilter
1.2.10	FintekCanbusFd_GetErrorFilter
1.2.11	FintekCanbusFd_Setler
1.2.12	FintekCanbusFd_Getler
1.2.13	FintekCanbusFd_GetErrorCode
1.2.14	FintekCanbusFd_Send
1.2.15	FintekCanbusFd_Receive
1.2.16	FintekCanbusFd_ReceiveEx
1.2.17	FintekCanbusFd_GetIdVal
1.2.18	FintekCanbusFd_SetMode
1.2.19	FintekCanbusFd_GetMode
1.2.20	FintekCanbusFd_Reset
1.2.21	FintekCanbusFd_Create
1.2.22	FintekCanbusFd_Delete

1.2.23	FintekCanbusFd_Start
1.2.24	FintekCanbusFd_Stop

## 1.2.1 FintekCanbusFd\_Open

**long FintekCanbusFd\_Open(IN char\* sComPortNumber, CanFdInitial\* config);**

**Function:** Open Can FD virtual COM Port.

**Parameters:**

sComPortNumber: COM port number.

config: The CAN FD frame initialized value.

```
struct CanFdInitial {
    BOOL fd_en;
    BOOL brs_en;
    BOOL iso_en;
    BOOL ssp_en;
    FLOAT nominal_samplepoint;
    FLOAT data_samplepoint;
    BYTE nominal_sjw;
    BYTE data_sjw;
    DWORD nominal_brp;
    BYTE data_brp;
    BYTE baudrate_prediv;
    BYTE ssp_offset;
};
```

- Please refer to the table below for the nominal and data sample point and BRP settings.
- Set SJW to a value between 0 and 3.
- The baudrate\_prediv is fixed at 1.
- The ssp\_offset can be adjusted after setting ssp\_en to 1.

F81601a nominal baudrate		
CAN baudrate	SamplePoint value (hex)	BRP
1000	75%: 0x1D4C	0x09
800	75%: 0x1D4C	0x04
500	75%: 0x1D4C	0x09

250	75%: 0x1D4C	0x04
125	87.5%: 0x222E	0x27
100	85%: 0x2134	0x27
50	85%: 0x2134	0x4F
20	85%: 0x2134	0xC7
10	85%: 0x2134	0x18F

F81601a data baudrate		
CAN baudrate	SamplePoint value (hex)	BRP
5000	75%: 0x1D4C	0x00
4000	80%: 0x1F40	0x01
2000	80%: 0x1F40	0x01

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.2 FintekCanbusFd\_Close

**long FintekCanbusFd\_Close(IN char\* sComPortNumber);**

**Function:** Close Can FD virtual COM Port.

#### Parameters:

sComPortNumber: COM port number.

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.3 FintekCanbusFd\_SetBaudRate

**long FintekCanbusFd\_SetBaudRate(IN char\* sComPortNumber, IN DWORD BaudRate);**

**Function:** Set Can FD port Baud Rate. According to the FintekCanbusFd\_Open API, specify the nominal and data baud rates based on the BRS settings. Set the BaudRate parameter using hexadecimal values.

#### Parameters:

sComPortNumber: COM port number.

BaudRate[31:16]: Nominal bit rate

5M:5000, set bit[31:16] to 0x1388

4M:4000

2M:2000

1M:1000

BaudRate[15:0]: Data bit rate

1M:1000 set bit[31:16] to 0x03E8

800K:800

500K:500

250K:250

100K:100

50K:50

20K:20

10K:10

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.4 FintekCanbusFd\_GetBaudRate

**long FintekCanbusFd\_GetBaudRate(IN char\* sComPortNumber, OUT DWORD\* BaudRate);**

**Function:** Get Can FD port Baud Rate.

#### Parameters:

sComPortNumber: COM port number.

\*BaudRate[31:16]:

5M:5000

4M:4000

2M:2000

1M:1000

\*BaudRate[15:0]:

1M:1000

800K:800

500K:500

250K:250

100K:100

50K:50

20K:20

10K:10

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.5 FintekCanbusFd\_SetId

**long FintekCanbusFd\_SetId(IN char\* sComPortNumber, IN CanFrameFormat type, IN DWORD id);****Function:** Set Can FD id.**Parameters:**

sComPortNumber: COM port number.

type: An CanFrameFormat enumerated type specifying the CAN protocol with 11bit or 29bit.

```
enum class CanFrameFormat {  
    CP_29Bit,  
    CP_11Bit  
};
```

id: Specified the id.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.6 FintekCanbusFd\_SetFilter

**long FintekCanbusFd\_SetFilter(IN char\* sComPortNumber, IN DWORD pattern, IN DWORD mask);****Function:** Set Can FD port Filter. Default: pattern:0; mask:0, all frame will be received**Parameters:**

sComPortNumber: COM port number.

pattern: Specified CAN ID pattern to filter.

mask: Specified the mask for filter.

**Return Value:**

If the function succeeds, the return value is filter total count - 1 (0-14, MAX: 15). If the function fails, the return value is error code generated by the API. You can find more information about error codes at the section 3 of the document.



## 1.2.7 FintekCanbusFd\_ClearFilter

**long FintekCanbusFd\_ClearFilter(IN char\* sComPortNumber);**

**Function:** Clear Can FD port Filter.

**Parameters:**

sComPortNumber: COM port number.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.8 FintekCanbusFd\_SetAcrAmrFilter

**long FintekCanbusFd\_SetAcrAmrFilter(IN char\* sComPortNumber, IN char cFilterMode, IN DWORD acr, IN DWORD amr)**

**Function:** Set Can FD port ACR/AMR HW Filter.

**Parameters:**

sComPortNumber: COM port number.

cFilterMode: 1: Single Filter, 0: Dual Filter.

acr: Specified CAN ID ACR (Acceptance Code Register).

amr: Specified the AMR (Acceptance Mask Register).

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

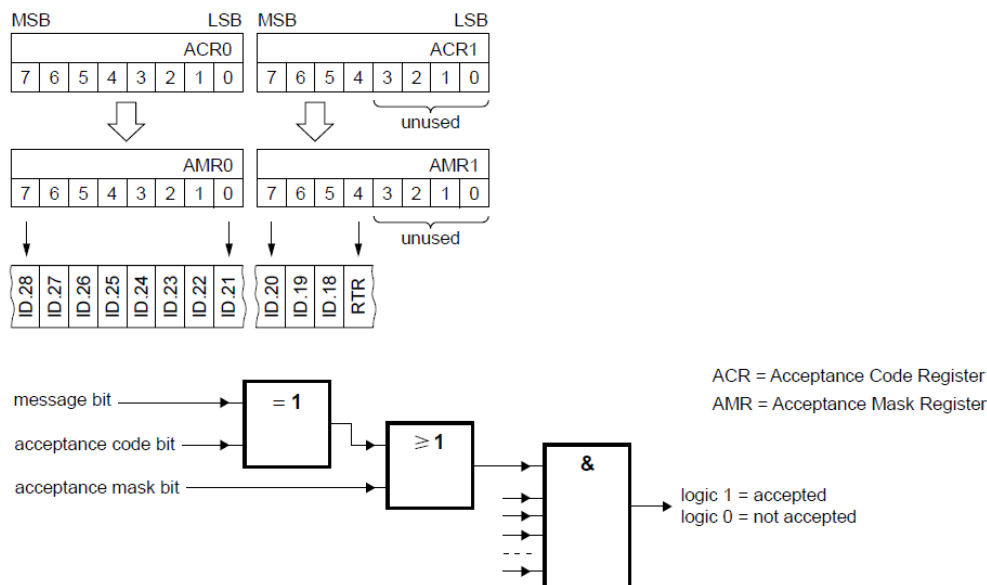
**Notice:**

With the help of the acceptance filter the CAN controller is able to allow passing of received messages to the RXFIFO only when the identifier bits of the received message are equal to the predefined ones within the acceptance filter registers. The acceptance filter is defined by the Acceptance Code Registers (ACR) and the Acceptance Mask Registers (AMR). The bit patterns of messages to be received are defined within the acceptance code registers.

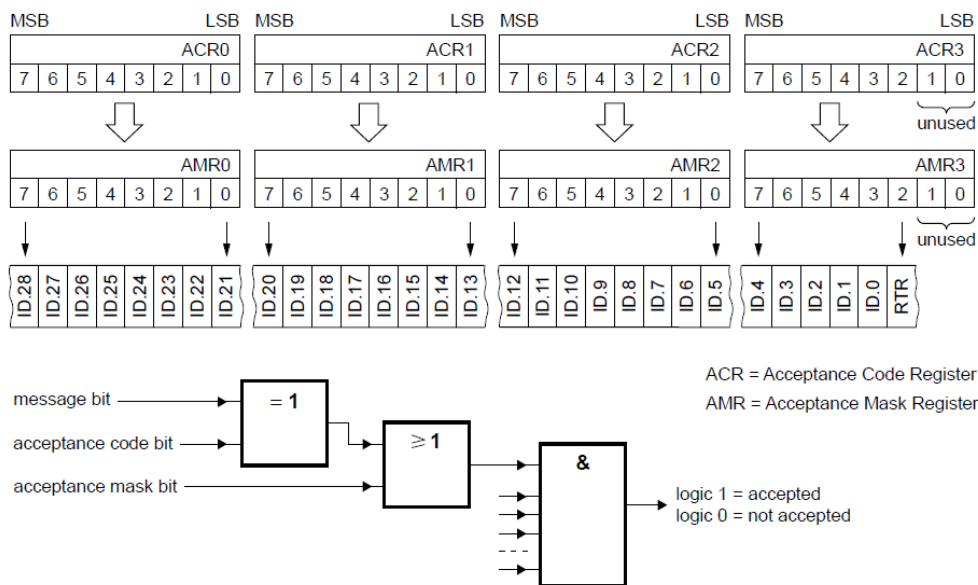
The corresponding acceptance mask registers allow to define certain bit positions to be 'don't care'.

Two different filter modes are selectable within the mode:

1. Single filter mode (cFilterMode is 1).

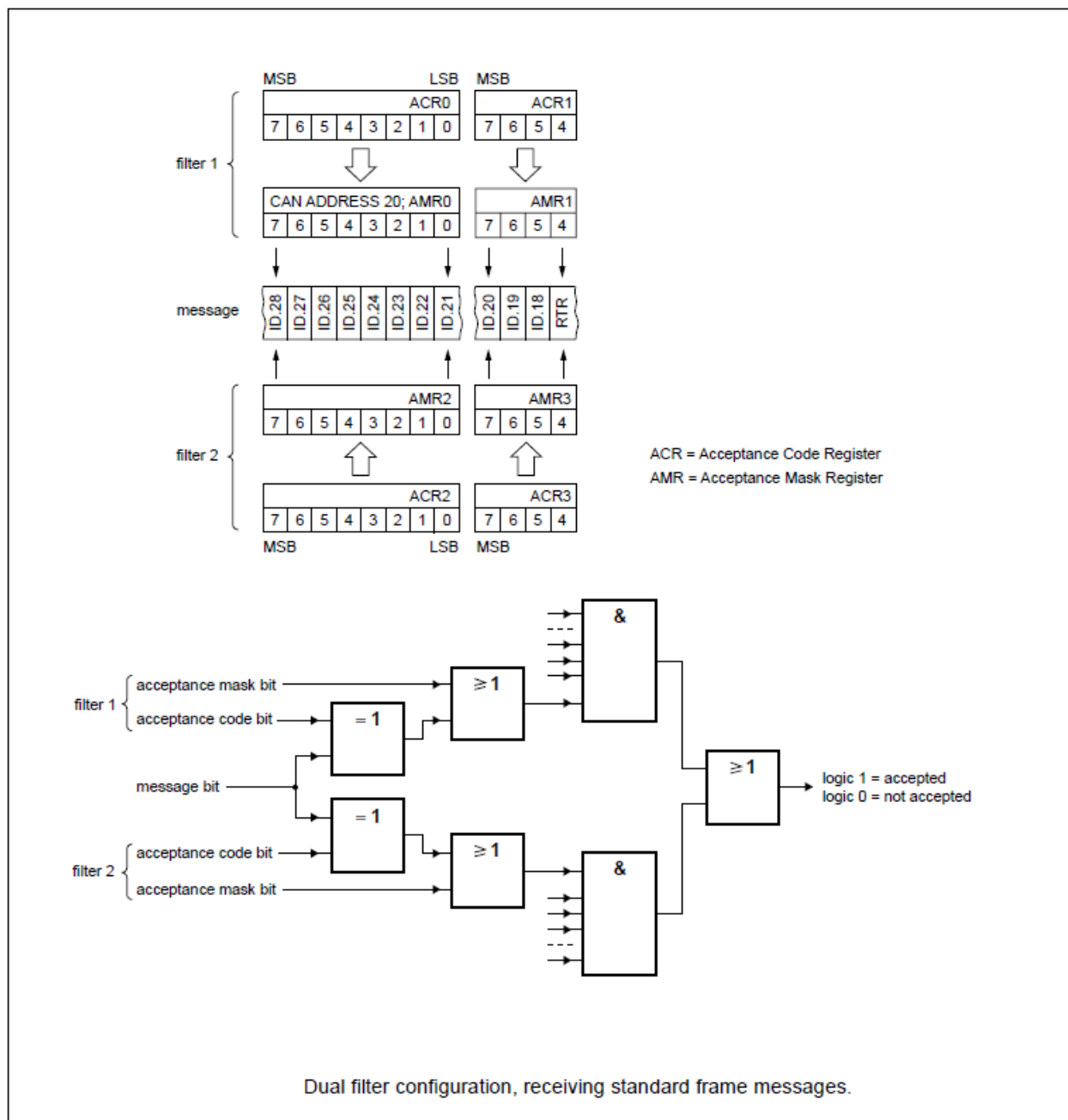


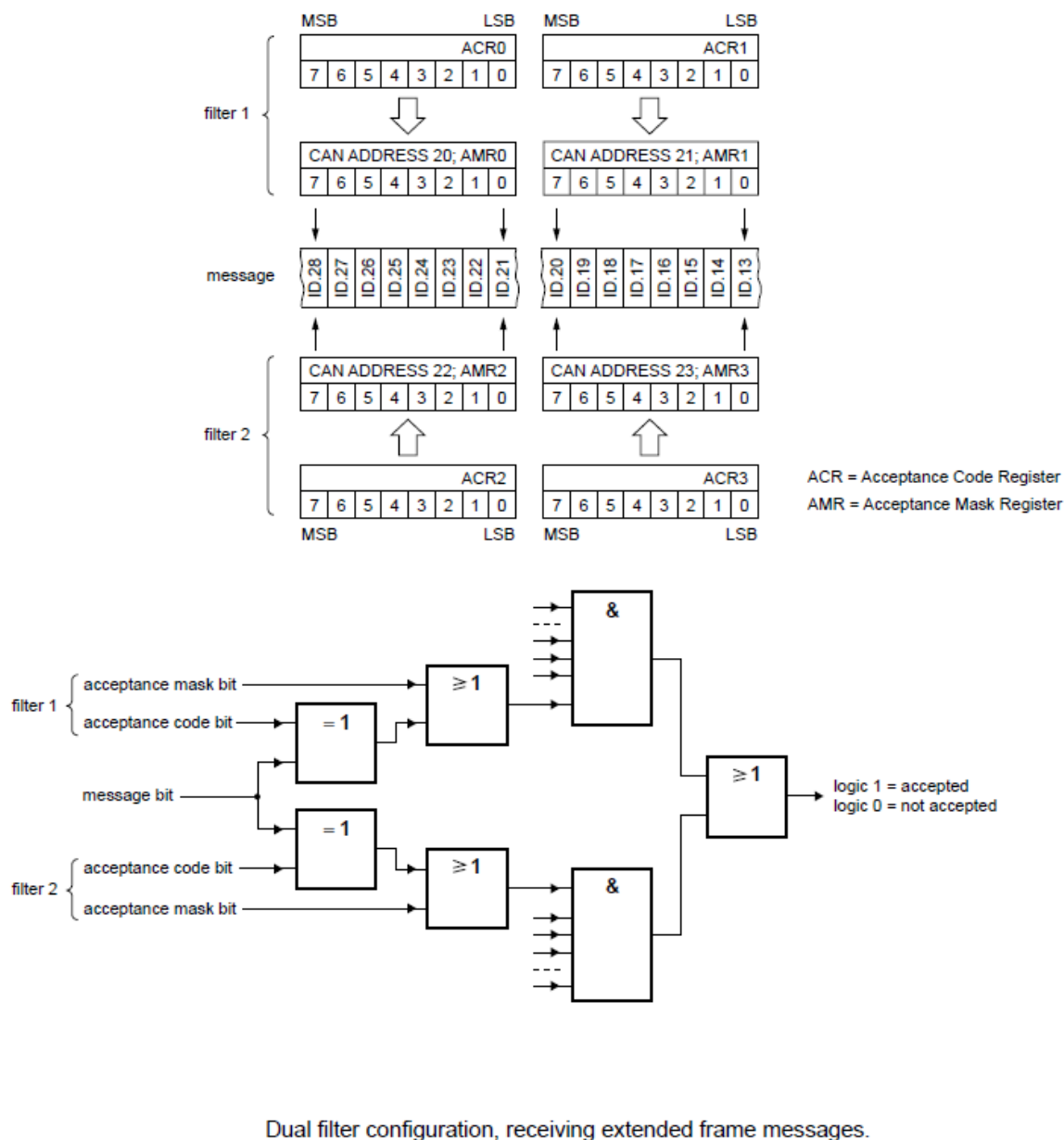
Single filter configuration, receiving standard frame messages.



Single filter configuration, receiving extended frame messages.

2. Dual filter mode (cFilterMode is 0).





## 1.2.9 FintekCanbusFd\_SetErrorFilter

**long FintekCanbusFd\_SetErrorFilter (IN char\* sComPortNumber, IN UCHAR errorfilter);**

**Function:** Set Can FD port error or warning report filter. Parameter errorfilter allow to define certain bit positions to be 'don't care'.

**Parameters:**

sComPortNumber: COM port number.

errorfilter: Set 0 to ignore the Error/Warning Bit report.

BIT7: Bus Error

BIT6: Arbitration Lost

BIT5: Error Passive

BIT4: Wake-Up

BIT3: Data Overrun

BIT2: Error Warning

BIT0-1: Reserved. Must to be 0

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.10 FintekCanbusFd\_GetErrorFilter

**long FintekCanbusFd\_GetErrorFilter (IN char\* sComPortNumber, OUT UCHAR\* errorfilter);**

**Function:** Get Can FD port error or warning report filter value.

**Parameters:**

sComPortNumber: COM port number.

\*errorfilter: Error/Warning Bit report value, 0 is ignore.

BIT7: Bus Error

BIT6: Arbitration Lost

BIT5: Error Passive

BIT4: Wake-Up

BIT3: Data Overrun

BIT2: Error Warning

BIT0-1: Reserved. Must to be 0

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.11 FintekCanbusFd\_Setler

**long FintekCanbusFd\_Setler(IN char\* sComPortNumber, IN UCHAR ier);**

**Function:** Set Can FD port interrupt enable register (IER). The function allows to enable/disable different types of interrupt sources

**Parameters:**

sComPortNumber: COM port number.

ier: Set 1 to enable the interrupt Bit report.

BIT7: Bus Error interrupt

BIT6: Arbitration Lost interrupt

BIT5: Error Passive interrupt

BIT4: Wake-Up interrupt

BIT3: Data Overrun interrupt

BIT2: Error Warning interrupt

BIT0-1: Reserved. Must to be 1

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.12 FintekCanbusFd\_Getler

**long FintekCanbusFd\_Getler(IN char\* sComPortNumber, OUT UCHAR\* ier);**

**Function:** Get Can FD port interrupt enable register (IER).

**Parameters:**

sComPortNumber: COM port number.

\*ier: The interrupt Bit report value, 0 is disable, 1 is enable.

BIT7: Bus Error interrupt

BIT6: Arbitration Lost interrupt

BIT5: Error Passive interrupt

BIT4: Wake-Up interrupt

BIT3: Data Overrun interrupt  
 BIT2: Error Warning interrupt  
 BIT0-1: Reserved. Must to be 1

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.13 FintekCanbusFd\_GetErrorCode

**long FintekCanbusFd\_GetErrorCode(IN char\* sComPortNumber, OUT DWORD\* ErrorCode);**

**Function:** Get Can FD port error code value.

**Parameters:**

sComPortNumber: COM port number.

\*ErrorCode:

bit [31:24]: Receive Error Counter

bit [23:16]: Transmit Error Counter

bit[15:8]: Error Code Capture

bit[7]: Bus Error

bit[6]: Arbitration Lost

bit[5]: Error Passive

bit[3]: Data Overrun

bit[2]: Error Warning

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.14 FintekCanbusFd\_Send

**long FintekCanbusFd\_Send(IN char\* sComPortNumber, CanFdFrameInfor\* msg);**

**Function:** Send Can FD frame to virtual COM Port.

**Parameters:**

sComPortNumber: COM port number.

msg: The send frame, include frame format, frame id, frame length and frame data.

```

struct CanFdFrameInfor {
    CanFdFrameFormat type;
    BYTE    fd;
    BYTE    brs;
    BYTE    rtr;
    BYTE    esi;
    DWORD   id;
    BYTE    data_len;
    BYTE    data[CANBUSFD_MAX_DATA_SIZE];
}; enum class CanFrameFormat {
    CP_29Bit,
    CP_11Bit
};

```

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.15 FintekCanbusFd\_Receive

**long FintekCanbusFd\_Receive(IN char\* sComPortNumber, IN CanFdFrameInforProc callback);**

**Function:** Receive Can FD frame to virtual COM Port.

#### Parameters:

sComPortNumber: COM port number.

callback:

typedef void(CALLBACK \*CanFdFrameInforProc)(long error, CanFdFrameInfor\* msg);

```

struct CanFdFrameInfor {
    CanFdFrameFormat type;
    BYTE    fd;
    BYTE    brs;
    BYTE    rtr;
    BYTE    esi;
    DWORD   id;
    BYTE    data_len;
    BYTE    data[CANBUSFD_MAX_DATA_SIZE];
};

```



```
}; enum class CanFrameFormat {
    CP_29Bit,
    CP_11Bit
};
```

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the callback parameter “error” is nonzero and the error code generated by it. You can find more information about error codes at the section 3 of the document.

## 1.2.16 FintekCanbusFd\_ReceiveEx

**long FintekCanbusFD\_ReceiveEx(IN char\* sComPortNumber, IN CanFdFrameInforProc callback);**

**Function:** Receive Can FD frame to virtual COM Port.

#### Parameters:

sComPortNumber: COM port number.

callback:

```
typedef void(CALLBACK *CanFdFrameInforProcEx)(long error, CanFdFrameInforEx* msg);
```

```
struct CanFdFrameInforEx {
    CanFdFrameFormat type;
    INT com_number;
    BYTE fd;
    BYTE brs;
    BYTE rtr;
    BYTE esi;
    DWORD id;
    BYTE data_len;
    BYTE data[CANBUSFD_MAX_DATA_SIZE];
};

enum class CanFrameFormat {
    CP_29Bit,
    CP_11Bit
};
```

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the callback parameter “error” is nonzero

and the error code generated by it. You can find more information about error codes at the section 3 of the document.

## 1.2.17 FintekCanbusFd\_GetIdVal

**long FintekCanbusFd\_GetIdVal(IN char\* sComPortNumber, OUT UCHAR\* IdVal);**

**Function:** Get F81601a id value from ID pins, this API is only support for F81601a.

**Parameters:**

sComPortNumber: COM port number.

\* IdVal: 0~7.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.18 FintekCanbusFd\_SetMode

**long FintekCanbusFd\_SetMode(IN char\* sComPortNumber, IN UCHAR mode);**

**Function:** Set Can FD Mode. The function allows to change the behaviour of the CAN controller.

**Parameters:**

sComPortNumber: COM port number.

mode: Set 1 to enable the Canbus Mode.

BIT3: Acceptance Filter Mode: 1: Single Filter, 0: Dual Filter.

BIT1: Listen Only Mode: 1 is enable, 0 is disable.

BIT0, 2, 4-7: Reserved.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.19 FintekCanbusFd\_GetMode

**long FintekCanbusFd\_GetMode(IN char\* sComPortNumber, OUT UCHAR\* mode);**

**Function:** Get Can FD Mode.

**Parameters:**

sComPortNumber: COM port number.

\* mode: The Mode Bit report value.

BIT3: Acceptance Filter Mode: 1: Single Filter, 0: Dual Filter.

BIT1: Listen Only Mode: 1 is enable, 0 is disable.

BIT0, 2, 4-7: Reserved.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.20 FintekCanbusFd\_Reset

**long FintekCanbusFd\_Reset(IN char\* sComPortNumber, IN UCHAR mode);**

**Function:** Reset can FD port.

**Parameters:**

sComPortNumber: COM port number.

mode: Reserved. Must to be 0

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.21 FintekCanbusFd\_Create

**long FintekCanbusFd\_Create(IN char\* sComPortNumber);**

**Function:** Initializes (create) can FD port but does not start it. Please use function FintekCanbusFd\_Start to startup. The difference from FintekCanbusFd\_Open function is that includes initialization and startup.

**FintekCanbusFd\_Open = FintekCanbusFd\_Create + FintekCanbusFd\_Start**

**Parameters:**

sComPortNumber: COM port number.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.22 FintekCanbusFd\_Delete

**long FintekCanbusFd\_Delete(IN char\* sComPortNumber);**

**Function:** UnInitializes (Delete) can FD port. This call is a reciprocal to FintekCanbusFd\_Create.

**Parameters:**

sComPortNumber: COM port number.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.23 FintekCanbusFd\_Start

**long FintekCanbusFd\_Start(IN char\* sComPortNumber);**

**Function:** Start can FD port. You must initialize can FD before using this function. Please refer to the function FintekCanbusFd\_Create description.

**Parameters:**

sComPortNumber: COM port number.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 1.2.24 FintekCanbusFd\_Stop

**long FintekCanbusFd\_Stop(IN char\* sComPortNumber);**

**Function:** Stop can FD port. This call is a reciprocal to FintekCanbusFd\_Start

**Parameters:**

sComPortNumber: COM port number.

### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2. Fintek CAN2.0 DLL Control APIs

### 2.1 Support Fintek CANBus IC

- F81601a

### 2.2 Function List

This section provides the specifications of all Fintek CAN2.0 functions and structures. All APIs use the naming convention FintekCanbus\_xxx specific to below table:

<i>ID</i>	<i>Function Name</i>
2.2.1	FintekCanbus_Open
2.2.2	FintekCanbus_Close
2.2.3	FintekCanbus_SetBaudRate
2.2.4	FintekCanbus_GetBaudRate
2.2.5	FintekCanbus_SetId
2.2.6	FintekCanbus_SetFilter
2.2.7	FintekCanbus_ClearFilter
2.2.8	FintekCanbus_SetAcrAmrFilter
2.2.9	FintekCanbus_SetErrorFilter
2.2.10	FintekCanbus_GetErrorFilter
2.2.11	FintekCanbus_Setler
2.2.12	FintekCanbus_Getler
2.2.13	FintekCanbus_GetErrorCode
2.2.14	FintekCanbus_Send
2.2.15	FintekCanbus_Receive
2.2.16	FintekCanbus_ReceiveEx

2.2.17	FintekCanbus_GetIdVal (F81601 only)
2.2.18	FintekCanbus_SetCtrlVal (F81604 only)
2.2.19	FintekCanbus_GetCtrlVal (F81604 only)
2.2.20	FintekCanbus_SetMode
2.2.21	FintekCanbus_GetMode
2.2.22	FintekCanbus_Reset
2.2.23	FintekCanbus_Create
2.2.24	FintekCanbus_Delete
2.2.25	FintekCanbus_Start
2.2.26	FintekCanbus_Stop

## 2.2.1 FintekCanbus\_Open

**long FintekCanbus\_Open(IN char\* sComPortNumber, IN ULONG ulSamplePoint, IN UCHAR ulSJW, IN ULONG ulBRP);**

**Function:** Open Canbus virtual COM Port.

Parameters:

sComPortNumber: COM port number.

ulSamplePoint: Please refer to the table below for the baud rate comparison

ulSJW: Synchronization Jump Width, recommended to set the value to 0x03.

ulBRP: Baud Rate Prescaler, Please refer to the table below for the baud rate comparison

F81601a CAN2.0		
CAN baudrate	SamplePoint value (hex)	BRP
1000	75%: 0x1D4C	0x09
800	75%: 0x1D4C	0x04
500	75%: 0x1D4C	0x09
250	75%: 0x1D4C	0x04
125	87.5%: 0x222E	0x27
100	85%: 0x2134	0x27
50	85%: 0x2134	0x4F
20	85%: 0x2134	0xC7
10	85%: 0x2134	0x18F

### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error

code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.2 FintekCanbus\_Close

**long FintekCanbus\_Close(IN char\* sComPortNumber);**

**Function:** Close Canbus virtual COM Port.

**Parameters:**

sComPortNumber: COM port number.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.3 FintekCanbus\_SetBaudRate

**long FintekCanbus\_SetBaudRate(IN char\* sComPortNumber, IN DWORD BaudRate);**

**Function:** Set Canbus port Baud Rate.

**Parameters:**

sComPortNumber: COM port number.

BaudRate:

1M:1000000

800K:800000

500K:500000

250K:250000

100K:100000

50K:50000

20K:20000

10K:10000

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.4 FintekCanbus\_GetBaudRate

**long FintekCanbus\_GetBaudRate(IN char\* sComPortNumber, OUT DWORD\* BaudRate);**

**Function:** Get Canbus port Baud Rate.

**Parameters:**

sComPortNumber: COM port number.

\*BaudRate:

1M:1000000

800K:800000

500K:500000

250K:250000

100K:100000

50K:50000

20K:20000

10K:10000

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.5 FintekCanbus\_SetId

**long FintekCanbus\_SetId(IN char\* sComPortNumber, IN CanFrameFormat type, IN DWORD id);**

**Function:** Set Canbus id.

**Parameters:**

sComPortNumber: COM port number.

type: An CanFrameFormat enumerated type specifying the CAN protocol with 11bit or 29bit.

```
enum class CanFrameFormat {
```

```
    CP_29Bit,
```

```
    CP_11Bit
```

```
};
```

id: Specified the id.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.



## 2.2.6 FintekCanbus\_SetFilter

**long FintekCanbus\_SetFilter(IN char\* sComPortNumber, IN DWORD pattern, IN DWORD mask);**

**Function:** Set Canbus port Filter. Default: pattern:0; mask:0, all frame will be received

**Parameters:**

sComPortNumber: COM port number.

pattern: Specified CAN ID pattern to filter.

mask: Specified the mask for filter.

**Return Value:**

If the function succeeds, the return value is filter total count - 1 (0-14, MAX: 15). If the function fails, the return value is error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.7 FintekCanbus\_ClearFilter

**long FintekCanbus\_ClearFilter(IN char\* sComPortNumber);**

**Function:** Clear Canbus port Filter.

**Parameters:**

sComPortNumber: COM port number.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.8 FintekCanbus\_SetAcrAmrFilter

**long FintekCanbus\_SetAcrAmrFilter(IN char\* sComPortNumber, IN char cFilterMode, IN DWORD acr, IN DWORD amr)**

**Function:** Set Canbus port ACR/AMR HW Filter.

**Parameters:**

sComPortNumber: COM port number.

cFilterMode: 1: Single Filter, 0: Dual Filter.

acr: Specified CAN ID ACR (Acceptance Code Register).

amr: Specified the AMR (Acceptance Mask Register).

### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

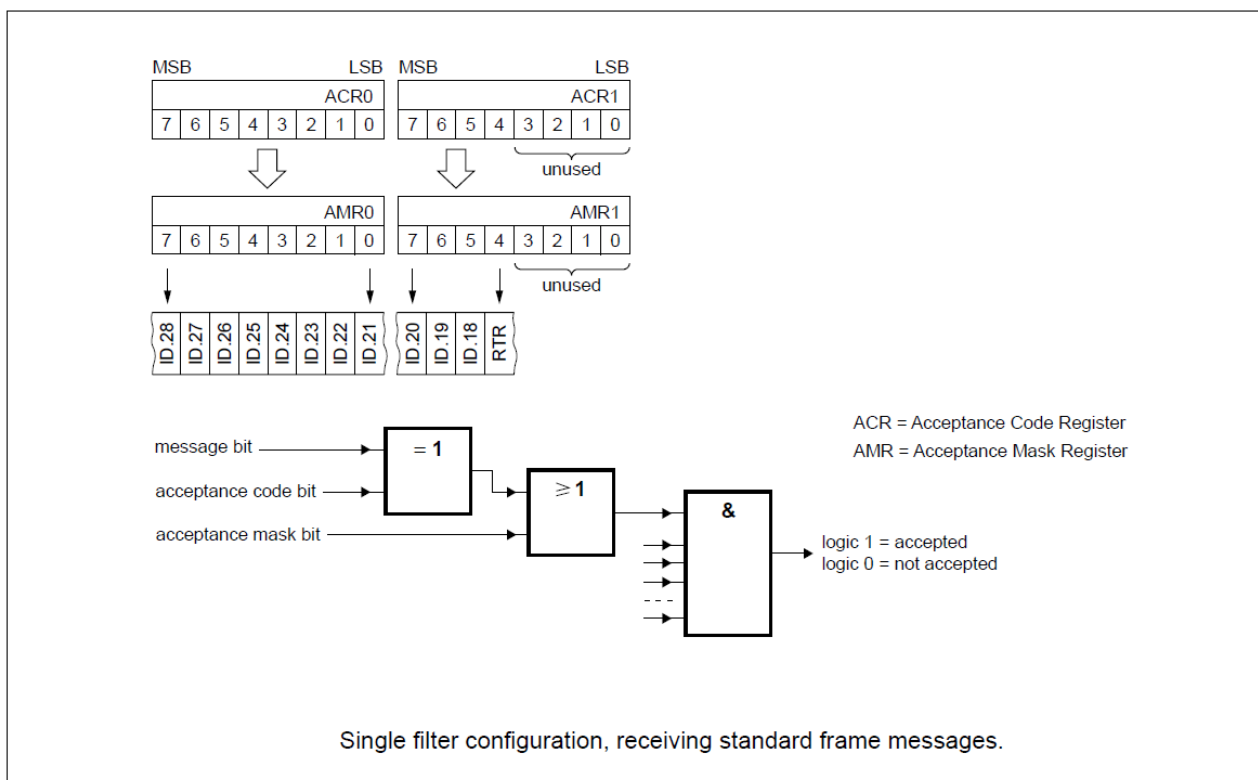
**Notice:**

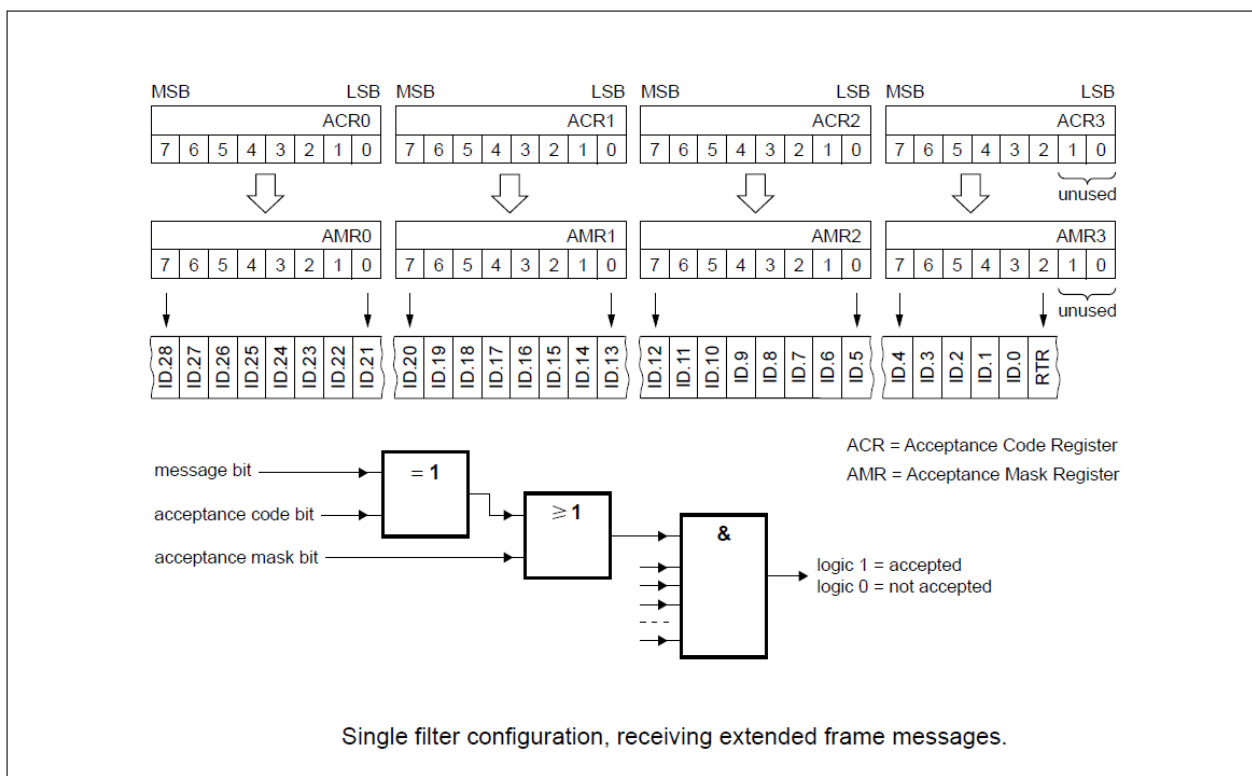
With the help of the acceptance filter the CAN controller is able to allow passing of received messages to the RXFIFO only when the identifier bits of the received message are equal to the predefined ones within the acceptance filter registers. The acceptance filter is defined by the Acceptance Code Registers (ACR) and the Acceptance Mask Registers (AMR). The bit patterns of messages to be received are defined within the acceptance code registers.

The corresponding acceptance mask registers allow to define certain bit positions to be 'don't care'.

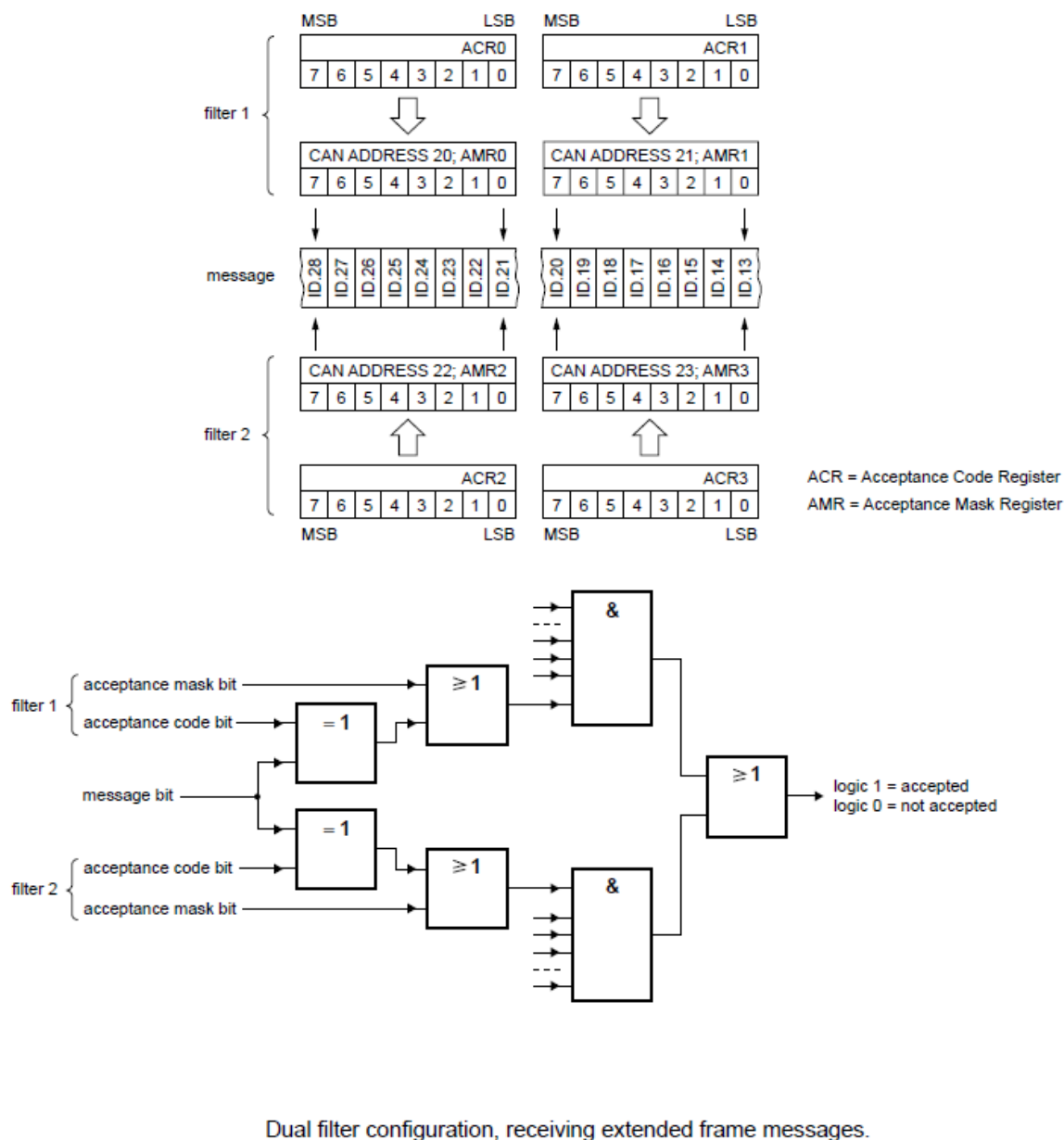
Two different filter modes are selectable within the mode:

1. Single filter mode (cFilterMode is 1).





[illegible]



## 2.2.9 FintekCanbus\_SetErrorFilter

**long FintekCanbus\_SetErrorFilter (IN char\* sComPortNumber, IN UCHAR errorfilter);**

**Function:** Set Canbus port error or warning report filter. Parameter errorfilter allow to define certain bit positions to be 'don't care'.

**Parameters:**

sComPortNumber: COM port number.

errorfilter: Set 0 to ignore the Error/Warning Bit report.

BIT7: Bus Error

BIT6: Arbitration Lost

BIT5: Error Passive

BIT4: Wake-Up

BIT3: Data Overrun

BIT2: Error Warning

BIT0-1: Reserved. Must to be 0

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.10 FintekCanbus\_GetErrorFilter

**long FintekCanbus\_GetErrorFilter (IN char\* sComPortNumber, OUT UCHAR\* errorfilter);**

**Function:** Get Canbus port error or warning report filter value.

**Parameters:**

sComPortNumber: COM port number.

\*errorfilter: Error/Warning Bit report value, 0 is ignore.

BIT7: Bus Error

BIT6: Arbitration Lost

BIT5: Error Passive

BIT4: Wake-Up

BIT3: Data Overrun

BIT2: Error Warning

BIT0-1: Reserved. Must to be 0

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.11 FintekCanbus\_Setler

**long FintekCanbus\_Setler(IN char\* sComPortNumber, IN UCHAR ier);**

**Function:** Set Canbus port interrupt enable register (IER). The function allows to enable/disable different types of interrupt sources

**Parameters:**

sComPortNumber: COM port number.

ier: Set 1 to enable the interrupt Bit report.

BIT7: Bus Error interrupt

BIT6: Arbitration Lost interrupt

BIT5: Error Passive interrupt

BIT4: Wake-Up interrupt

BIT3: Data Overrun interrupt

BIT2: Error Warning interrupt

BIT0-1: Reserved. Must to be 1

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.12 FintekCanbus\_Getler

**long FintekCanbus\_Getler(IN char\* sComPortNumber, OUT UCHAR\* ier);**

**Function:** Get Canbus port interrupt enable register (IER).

**Parameters:**

sComPortNumber: COM port number.

\*ier: The interrupt Bit report value, 0 is disable, 1 is enable.

BIT7: Bus Error interrupt

BIT6: Arbitration Lost interrupt

BIT5: Error Passive interrupt

BIT4: Wake-Up interrupt

BIT3: Data Overrun interrupt

BIT2: Error Warning interrupt

BIT0-1: Reserved. Must to be 1

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.13 FintekCanbus\_GetErrorCode

**long FintekCanbus\_GetErrorCode(IN char\* sComPortNumber, OUT DWORD\* ErrorCode);**

**Function:** Get Canbus port error code value.

#### Parameters:

sComPortNumber: COM port number.

\*ErrorCode:

bit [31:24]: Receive Error Counter

bit [23:16]: Transmit Error Counter

bit[15:8]: Error Code Capture

bit[7]: Bus Error

bit[6]: Arbitration Lost

bit[5]: Error Passive

bit[3]: Data Overrun

bit[2]: Error Warning

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.14 FintekCanbus\_Send

**long FintekCanbus\_Send(IN char\* sComPortNumber, CanFrameInfor\* msg);**

**Function:** Send Canbus frame to virtual COM Port.

#### Parameters:

sComPortNumber: COM port number.

msg: The send frame, include frame format, frame id, frame length and frame data.

```
struct CanFrameInfor {
    CanFrameFormat type;
```



```

        BYTE    rtr;
        DWORD   id;
        BYTE    data_len;
        BYTE    data[CANBUS_MAX_DATA_SIZE];
    };

    enum class CanFrameFormat {
        CP_29Bit,
        CP_11Bit
    };

```

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.15 FintekCanbus\_Receive

**long FintekCanbus\_Receive(IN char\* sComPortNumber, IN CanFrameInforProc callback);**

**Function:** Receive Canbus frame to virtual COM Port.

#### Parameters:

sComPortNumber: COM port number.

callback:

```
typedef void(CALLBACK *CanFrameInforProc)(long error, CanFrameInfor* msg);
```

```

    struct CanFrameInfor {
        CanFrameFormat type;
        BYTE    rtr;
        DWORD   id;
        BYTE    data_len;
        BYTE    data[CANBUS_MAX_DATA_SIZE];
    };

    enum class CanFrameFormat {
        CP_29Bit,
        CP_11Bit
    };

```

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the callback parameter “error” is nonzero and the error code generated by it. You can find more information about error codes at the section 3 of the document.

## 2.2.16 FintekCanbus\_ReceiveEx

**long FintekCanbus\_ReceiveEx(IN char\* sComPortNumber, IN CanFrameInforProc callback);**

**Function:** Receive Canbus frame to virtual COM Port.

**Parameters:**

sComPortNumber: COM port number.

callback:

```
typedef void(CALLBACK *CanFrameInforProcEx)(long error, CanFrameInforEx* msg);
```

```
struct CanFrameInforEx {
    CanFrameFormat type;
    INT      com_number;
    BYTE     rtr;
    DWORD    id;
    BYTE     data_len;
    BYTE     data[CANBUS_MAX_DATA_SIZE];
};

enum class CanFrameFormat {
    CP_29Bit,
    CP_11Bit
};
```

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the callback parameter “error” is nonzero and the error code generated by it. You can find more information about error codes at the section 3 of the document.

## 2.2.17 FintekCanbus\_SetMode

**long FintekCanbus\_SetMode(IN char\* sComPortNumber, IN UCHAR mode);**

**Function:** Set Canbus Mode. The function allows to change the behaviour of the CAN controller.

**Parameters:**

sComPortNumber: COM port number.

mode: Set 1 to enable the Canbus Mode.

BIT3: Acceptance Filter Mode: 1: Single Filter, 0: Dual Filter.

BIT1: Listen Only Mode: 1 is enable, 0 is disable.

BIT0, 2, 4-7: Reserved.

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.18 FintekCanbus\_GetMode

**long FintekCanbus\_GetMode(IN char\* sComPortNumber, OUT UCHAR\* mode);**

**Function:** Get Canbus Mode.

#### Parameters:

sComPortNumber: COM port number.

\* mode: The Mode Bit report value.

BIT3: Acceptance Filter Mode: 1: Single Filter, 0: Dual Filter.

BIT1: Listen Only Mode: 1 is enable, 0 is disable.

BIT0, 2, 4-7: Reserved.

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.19 FintekCanbus\_Reset

**long FintekCanbus\_Reset(IN char\* sComPortNumber, IN UCHAR mode);**

**Function:** Reset canbus port.

#### Parameters:

sComPortNumber: COM port number.

mode: Reserved. Must to be 0

#### Return Value:

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error

code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.20 FintekCanbus\_Create

**long FintekCanbus\_Create(IN char\* sComPortNumber);**

**Function:** Initializes (create) canbus port but does not start it. Please use function FintekCanbus\_Start to startup. The difference from FintekCanbus\_Open function is that includes initialization and startup.

**FintekCanbus\_Open = FintekCanbus\_Create + FintekCanbus\_Start**

**Parameters:**

sComPortNumber: COM port number.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.21 FintekCanbus\_Delete

**long FintekCanbus\_Delete(IN char\* sComPortNumber);**

**Function:** UnInitializes (Delete) canbus port. This call is a reciprocal to FintekCanbus\_Create.

**Parameters:**

sComPortNumber: COM port number.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.22 FintekCanbus\_Start

**long FintekCanbus\_Start(IN char\* sComPortNumber);**

**Function:** Start canbus port. You must initialize canbus before using this function. Please refer to the function FintekCanbus\_Create description.

**Parameters:**

sComPortNumber: COM port number.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

## 2.2.23 FintekCanbus\_Stop

**long FintekCanbus\_Stop(IN char\* sComPortNumber);**

**Function:** Stop canbus port. This call is a reciprocal to FintekCanbus\_Start

**Parameters:**

sComPortNumber: COM port number.

**Return Value:**

If the function succeeds, the return value is zero. If the function fails, the return value is nonzero and the error code generated by the API. You can find more information about error codes at the section 3 of the document.

# 3. Error Codes

The below table lists errors that Fintek CANBus functions API returns in response to calls.

Error Code	Description
0x80000005	CAN number ERROR
0x80000007	Thread ERROR
0x8005FFFF	F8160X FATAL ERROR(bus off)
0x80050FFF	CAN BUFFER FULL
0x80060002	INVALID HANDLE VALUE
0x80060003	No response from device
0x80060008	CAN function called fail
0x80060101	CAN communication fail
0x80060201	Write fail
0x80060202	Read fail
other	CAN ERROR: bit [31:24]: Receive Error Counter bit [23:16]: Transmit Error Counter bit[15:8]: Error Code Capture bit[7]: Bus Error bit[6]: Arbitration Lost bit[5]: Error Passive bit[3]: Data Overrun bit[2]: Error Warning