

FINTEK

F81601A / F81605

CAN FD and CAN 2.0 Series

Windows Software Programming Guide

V1.05

December 17, 2025

12/17/2025

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_SetFilter	7
1.2.6	FintekCanbusFd_ClearFilter	7
1.2.7	FintekCanbusFd_SetAcrAmrFilter	7
1.2.8	FintekCanbusFd_SetErrorFilter	12
1.2.9	FintekCanbusFd_GetErrorFilter	12
1.2.10	FintekCanbusFd_Setler	13
1.2.11	FintekCanbusFd_Getler	13
1.2.12	FintekCanbusFd_GetErrorCode	14
1.2.13	FintekCanbusFd_Send	14
1.2.14	FintekCanbusFd_Receive	15
1.2.15	FintekCanbusFd_ReceiveEx	16
1.2.16	FintekCanbusFd_GetIdVal	17
1.2.17	FintekCanbusFd_SetMode	17
1.2.18	FintekCanbusFd_GetMode	18
1.2.19	FintekCanbusFd_Reset	18
1.2.20	FintekCanbusFd_Create	18
1.2.21	FintekCanbusFd_Delete	19
1.2.22	FintekCanbusFd_Start	19
1.2.23	FintekCanbusFd_Stop	20
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_SetFilter	23
2.2.6	FintekCanbus_ClearFilter	24
2.2.7	FintekCanbus_SetAcrAmrFilter	24
2.2.8	FintekCanbus_SetErrorFilter	28
2.2.9	FintekCanbus_GetErrorFilter	28
2.2.10	FintekCanbus_Setler	29
2.2.11	FintekCanbus_Getler	29
2.2.12	FintekCanbus_GetErrorCode	30
2.2.13	FintekCanbus_Send	30
2.2.14	FintekCanbus_Receive	31
2.2.15	FintekCanbus_ReceiveEx	32

2.2.16	FintekCanbus_SetMode	32
2.2.17	FintekCanbus_GetMode	33
2.2.18	FintekCanbus_Reset	33
2.2.19	FintekCanbus_Create	34
2.2.20	FintekCanbus_Delete	34
2.2.21	FintekCanbus_Start	34
2.2.22	FintekCanbus_Stop	35
3.	Error Codes	36
4.	CAN API Configuration Procedure.....	37

1. Fintek CAN FD DLL Control APIs

1.1 Support Fintek CAN FD IC

- F81601a / F81605

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_SetFilter
1.2.6	FintekCanbusFd_ClearFilter
1.2.7	FintekCanbusFd_SetAcrAmrFilter
1.2.8	FintekCanbusFd_SetErrorFilter
1.2.9	FintekCanbusFd_GetErrorFilter
1.2.10	FintekCanbusFd_Setler
1.2.11	FintekCanbusFd_Getler
1.2.12	FintekCanbusFd_GetErrorCode
1.2.13	FintekCanbusFd_Send
1.2.14	FintekCanbusFd_Receive
1.2.15	FintekCanbusFd_ReceiveEx
1.2.16	FintekCanbusFd_GetIdVal
1.2.17	FintekCanbusFd_SetMode
1.2.18	FintekCanbusFd_GetMode
1.2.19	FintekCanbusFd_Reset
1.2.20	FintekCanbusFd_Create
1.2.21	FintekCanbusFd_Delete
1.2.22	FintekCanbusFd_Start

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/F81605 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/F81605 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

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_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.

1.2.6 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.

1.2.7 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.

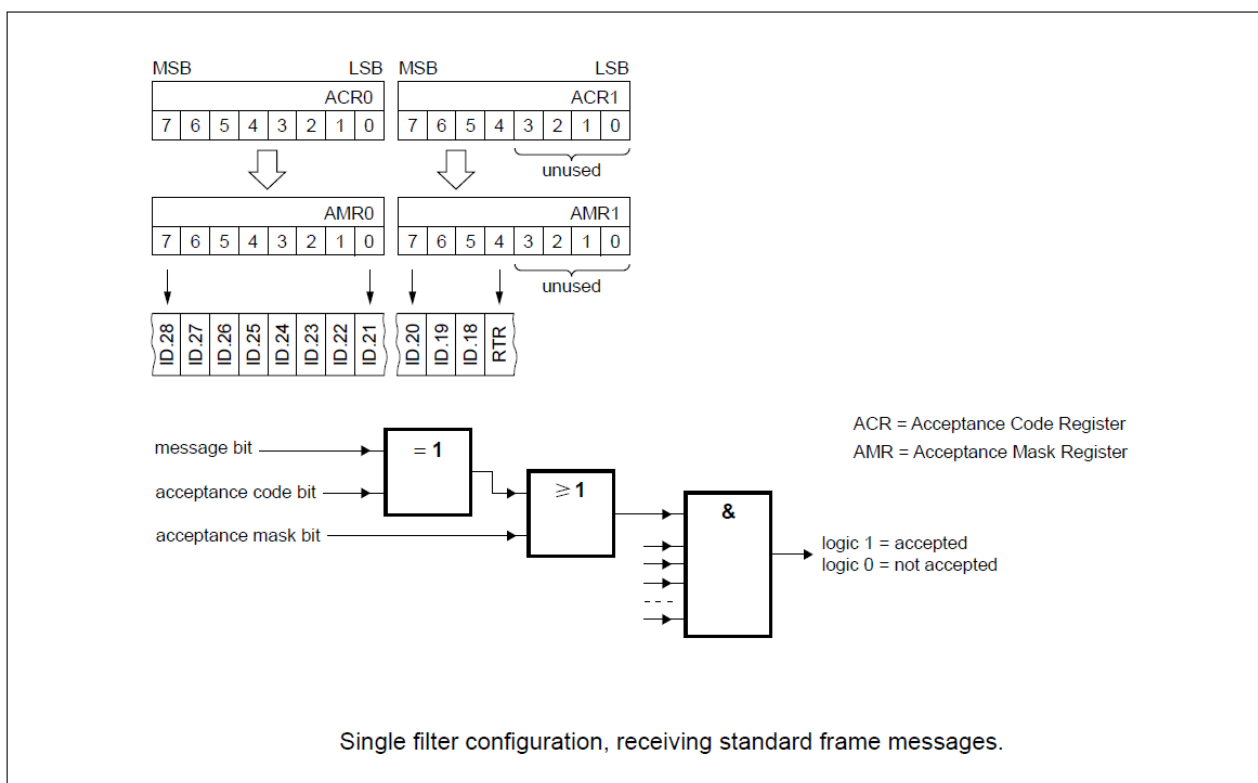
Notice:

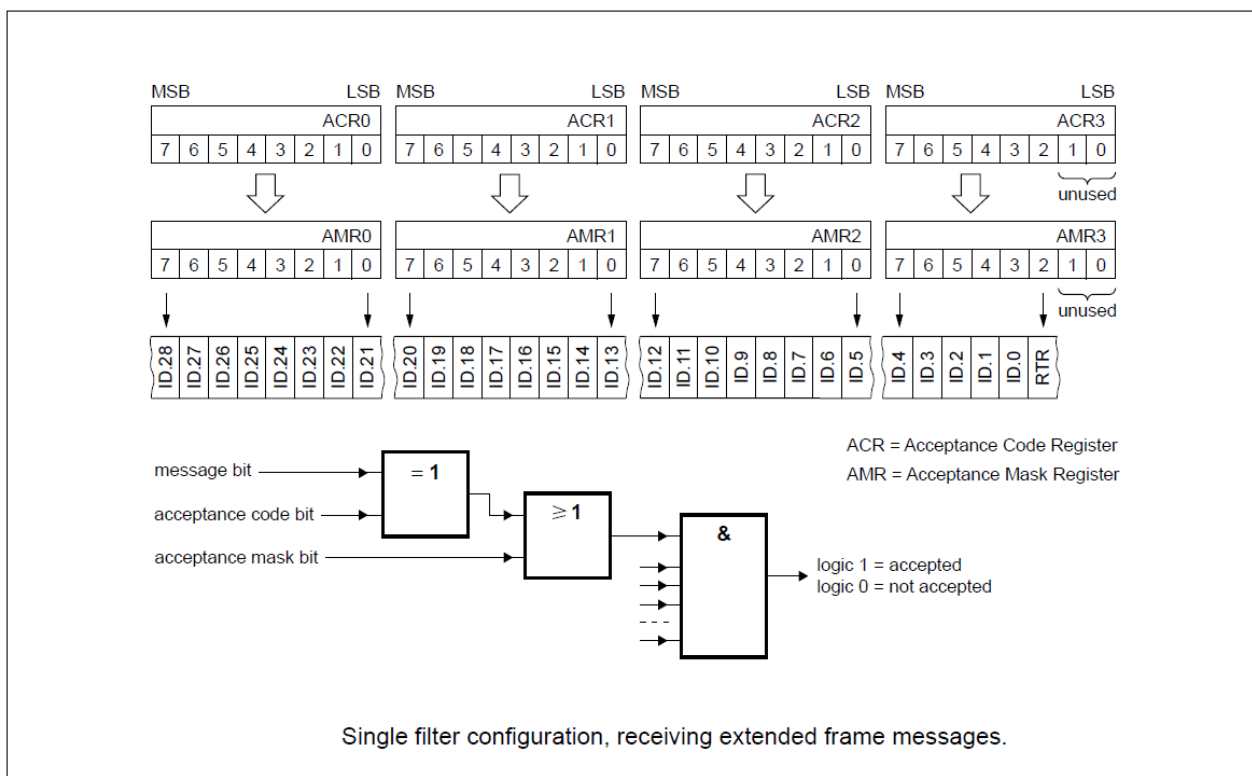
With the help of the acceptance filter the CAN controller is able to allow passing of received messages to the RXFIFO only when the identifier bits of the received message are equal to the predefined ones within the acceptance filter registers. The acceptance filter is defined by the Acceptance Code 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'.

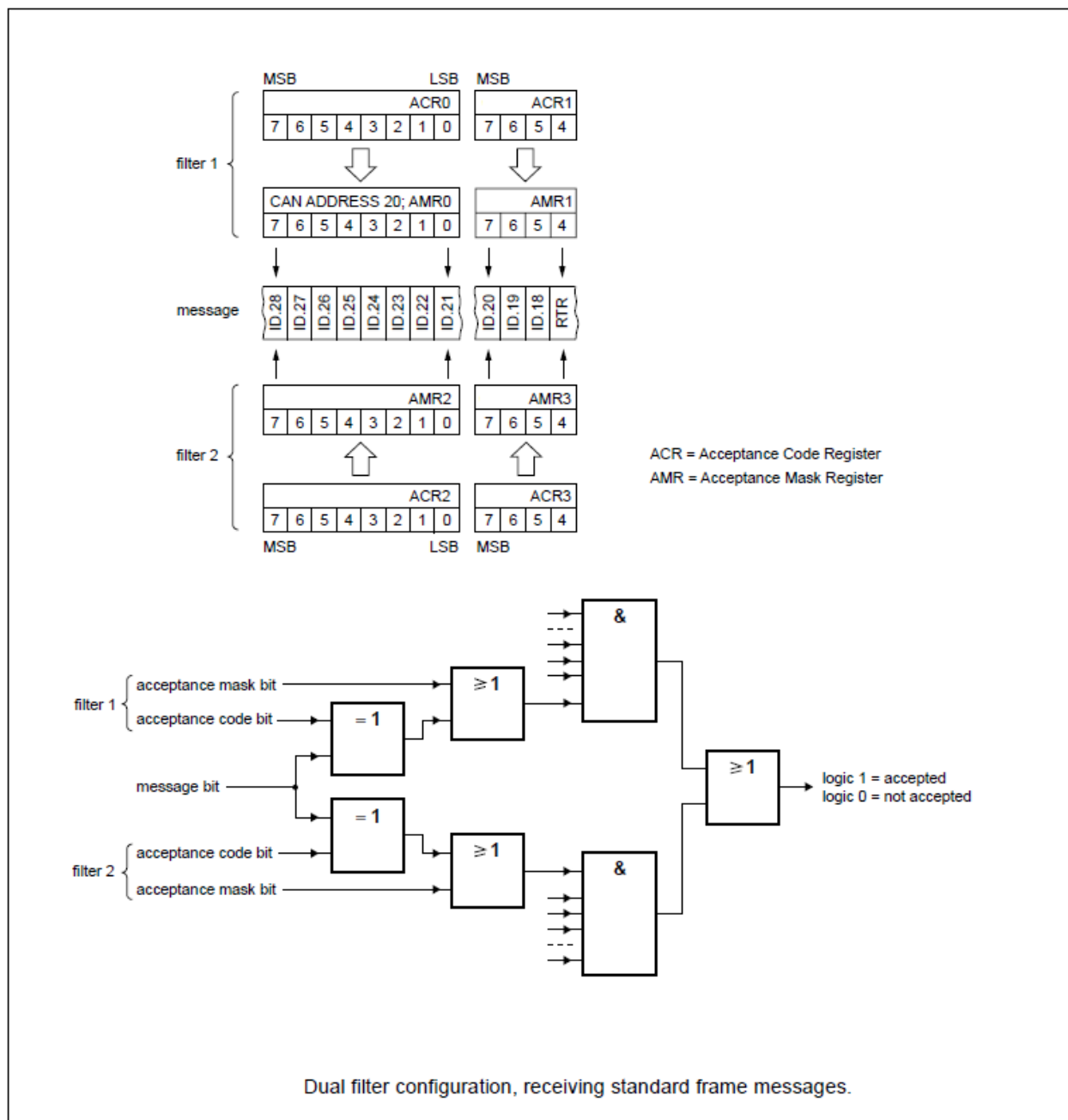
Two different filter modes are selectable within the mode:

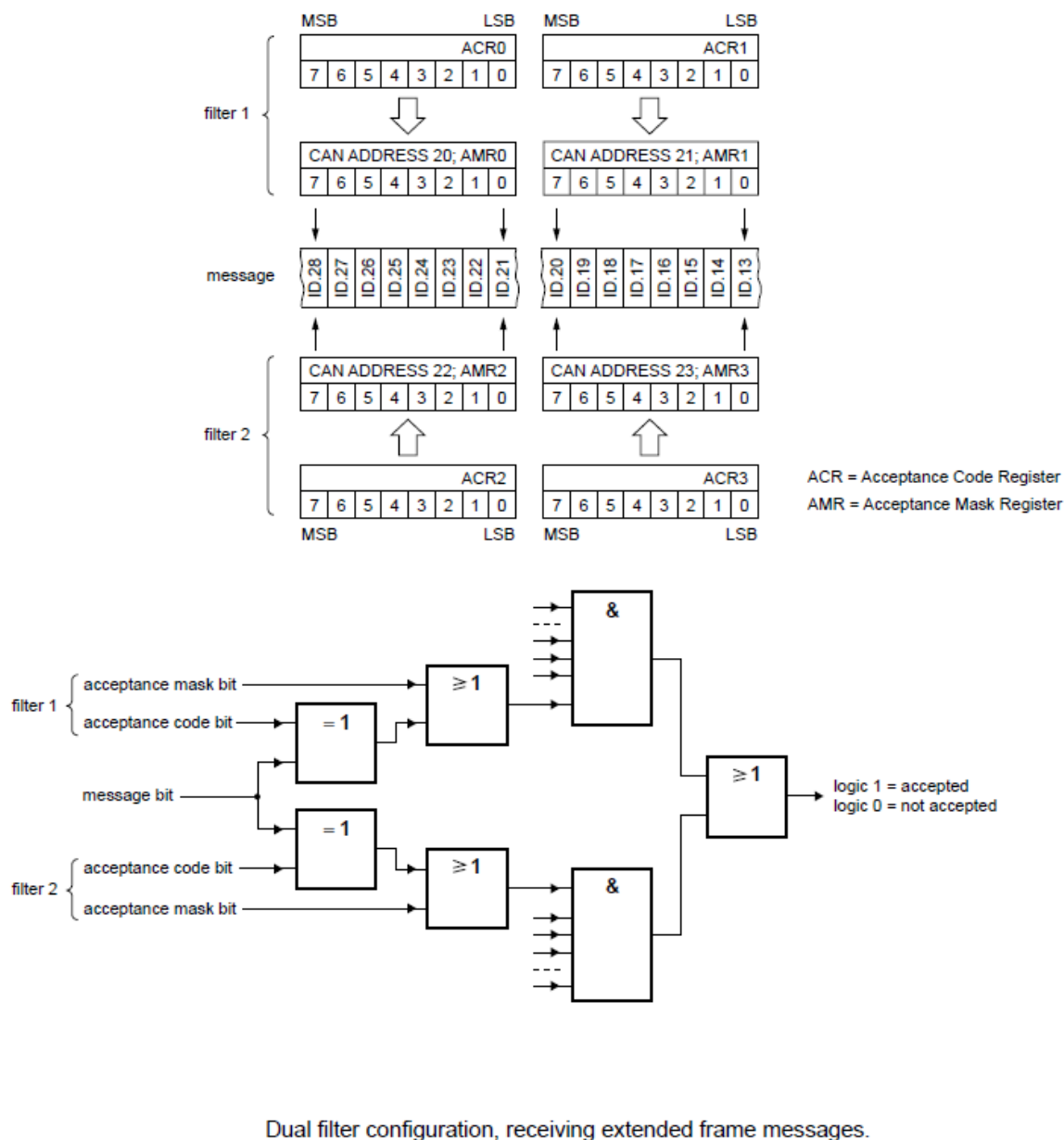
1. Single filter mode (cFilterMode is 1).





2. Dual filter mode (cFilterMode is 0).





Dual filter configuration, receiving extended frame messages.

1.2.8 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.

1.2.9 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.

1.2.10 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.

1.2.11 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.

1.2.12 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.

1.2.13 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 CanFdFrameFormat {
    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.

1.2.14 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 CanFdFrameFormat {
        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.

1.2.15 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 CanFdFrameFormat {
        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_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.

1.2.17 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.

1.2.18 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.

1.2.19 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.

1.2.20 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.

1.2.21 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.

1.2.22 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.

1.2.23 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 / F81605

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_SetFilter
2.2.6	FintekCanbus_ClearFilter
2.2.7	FintekCanbus_SetAcrAmrFilter
2.2.8	FintekCanbus_SetErrorFilter

2.2.9	FintekCanbus_GetErrorFilter
2.2.10	FintekCanbus_Setler
2.2.11	FintekCanbus_Getler
2.2.12	FintekCanbus_GetErrorCode
2.2.13	FintekCanbus_Send
2.2.14	FintekCanbus_Receive
2.2.15	FintekCanbus_ReceiveEx
2.2.16	FintekCanbus_SetMode
2.2.17	FintekCanbus_GetMode
2.2.18	FintekCanbus_Reset
2.2.19	FintekCanbus_Create
2.2.20	FintekCanbus_Delete
2.2.21	FintekCanbus_Start
2.2.22	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_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.

2.2.6 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.

2.2.7 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.

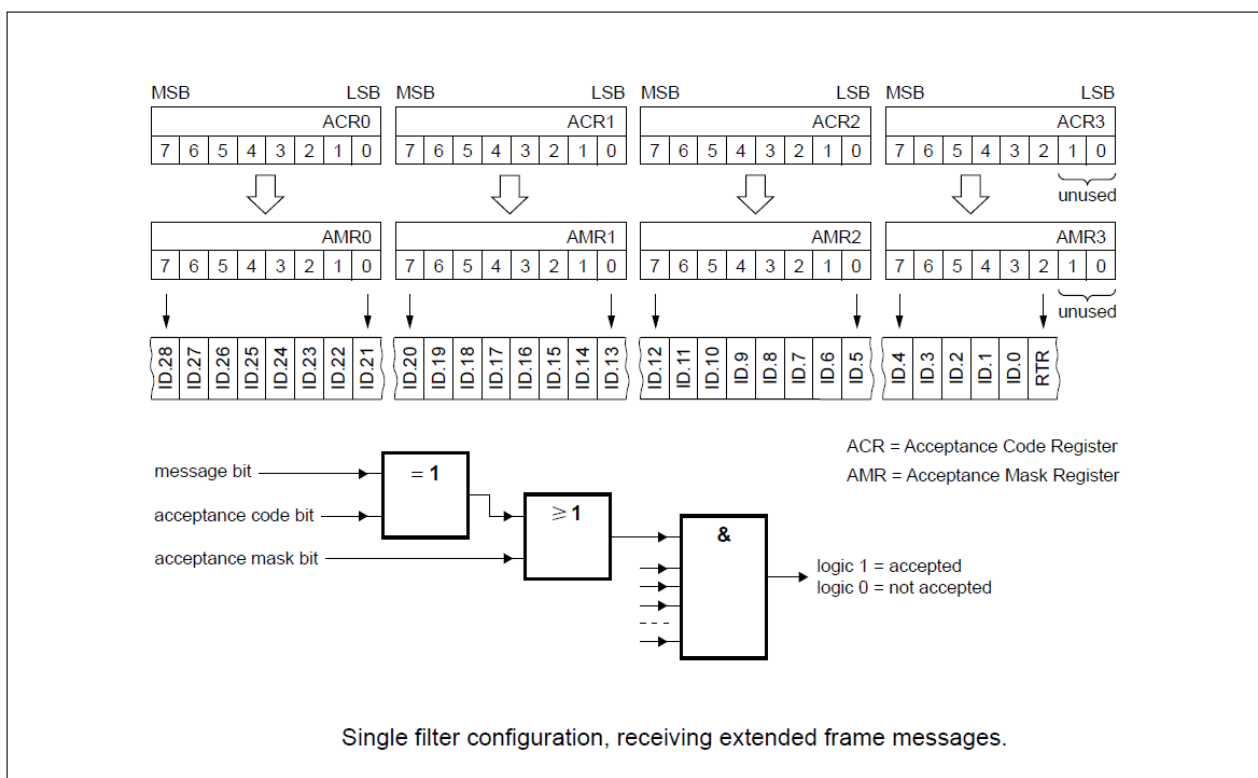
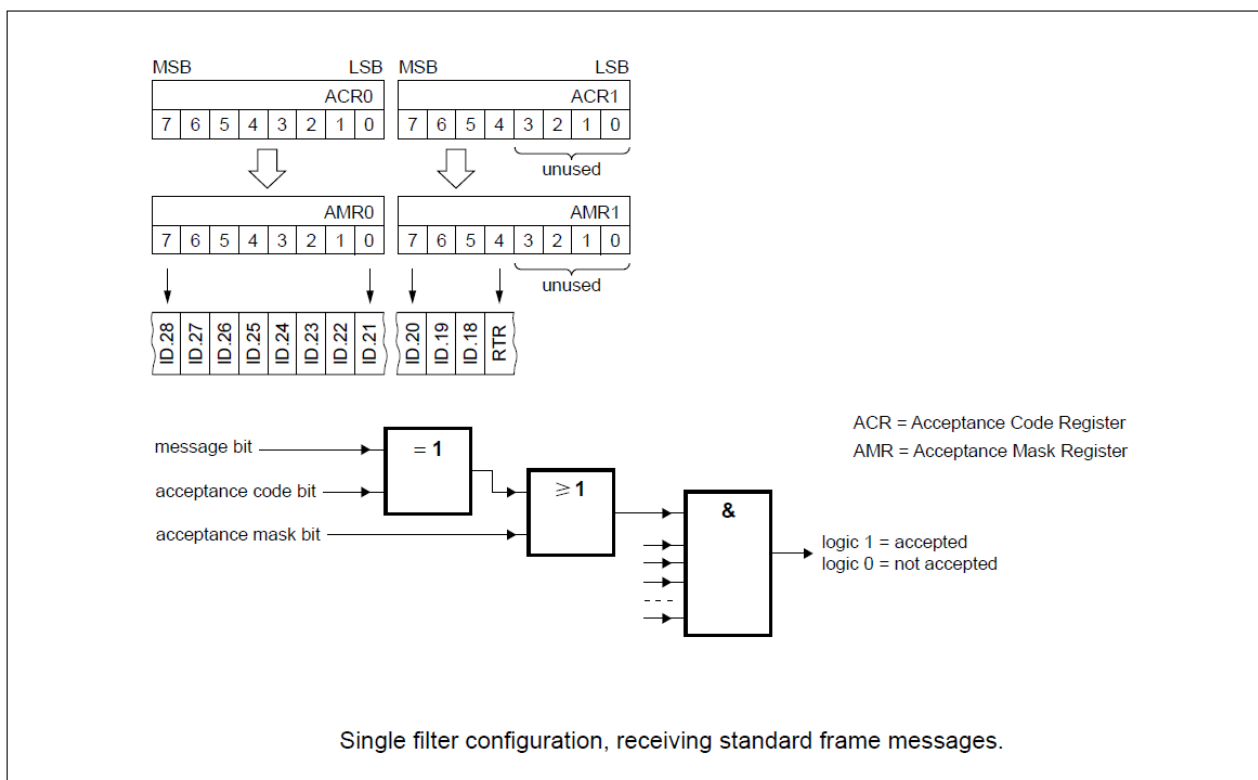
Notice:

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.

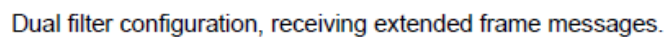
The corresponding acceptance mask registers allow to define certain bit positions to be 'don't care'.

Two different filter modes are selectable within the mode:

1. Single filter mode (cFilterMode is 1).



[illegible]



2.2.8 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.

2.2.9 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.

2.2.10 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.

2.2.11 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.

2.2.12 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.

2.2.13 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.

2.2.14 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.

2.2.15 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.

2.2.16 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.

2.2.17 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.

2.2.18 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.

2.2.19 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.

2.2.20 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.

2.2.21 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.

2.2.22 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

4. CAN API Configuration Procedure

The CAN-FD configuration can be performed in two ways:

1.
 - Use `FintekCanbusFd_Open` to set the CAN-FD initialization parameters.
 - Use `FintekCanbusFd_SetBaudRate` to configure the CAN-FD baud rate.
 - Use `FintekCanbusFd_SetFilter` to apply software-level frame filtering.
 - Use `FintekCanbusFd_SetAcrAmrFilter` to apply hardware-level frame filtering.
 - Use `FintekCanbusFd_Send` to transmit CAN frames.
 - Use `FintekCanbusFd_Receive` to receive CAN frames.
2. The difference between **`FintekCanbusFd_Open`** and **`FintekCanbusFd_Create` / `FintekCanbusFd_Start`** is that `FintekCanbusFd_Open` performs the combined operations of `FintekCanbusFd_Create` and `FintekCanbusFd_Start` sequentially. `FintekCanbusFd_Create` performs CAN FD initialization and CAN communication (transmit/receive) can only operate normally after `FintekCanbusFd_Start` is called. Conversely, `FintekCanbusFd_Stop` must be called before `FintekCanbusFd_Delete`. `FintekCanbusFd_Close` performs the combined operations of `FintekCanbusFd_Stop` and `FintekCanbusFd_Delete` sequentially.

The API call flow for CAN 2.0 is similar to that of CAN FD, The main difference is whether CAN FD initialization is performed.