

SDK文档_FlexCAN模块配置及应用（三）

1. 前言

YTM32单片机中，CAN FD模式下不能使用legacy FIFO，只能使用enhanced FIFO，经典CAN和CAN FD模式下的CAN功能可用性的具体信息如下图所示：

Feature	Classical CAN	CAN FD
Legacy RX FIFO	YES	NO
Legacy RX FIFO DMA	YES	NO
Enhanced RX FIFO	YES	YES
Enhanced RX FIFO DMA	YES	YES
Pretended Networking	YES	NO

对于需要用到CAN FD，又要使用FIFO接收的场景，就要用到enhanced FIFO，基于此，本文介绍了基于YCT的CAN enhanced FIFO的配置及使用方法。



只有M系列和H系列支持CAN enhanced FIFO，L系列不支持CAN enhanced FIFO。

2. 芯片型号及工具版本信息

- 型号：YTM32B1ME0MLQT
- 配置工具版本：2.2.0
- SDK版本：1_1_1

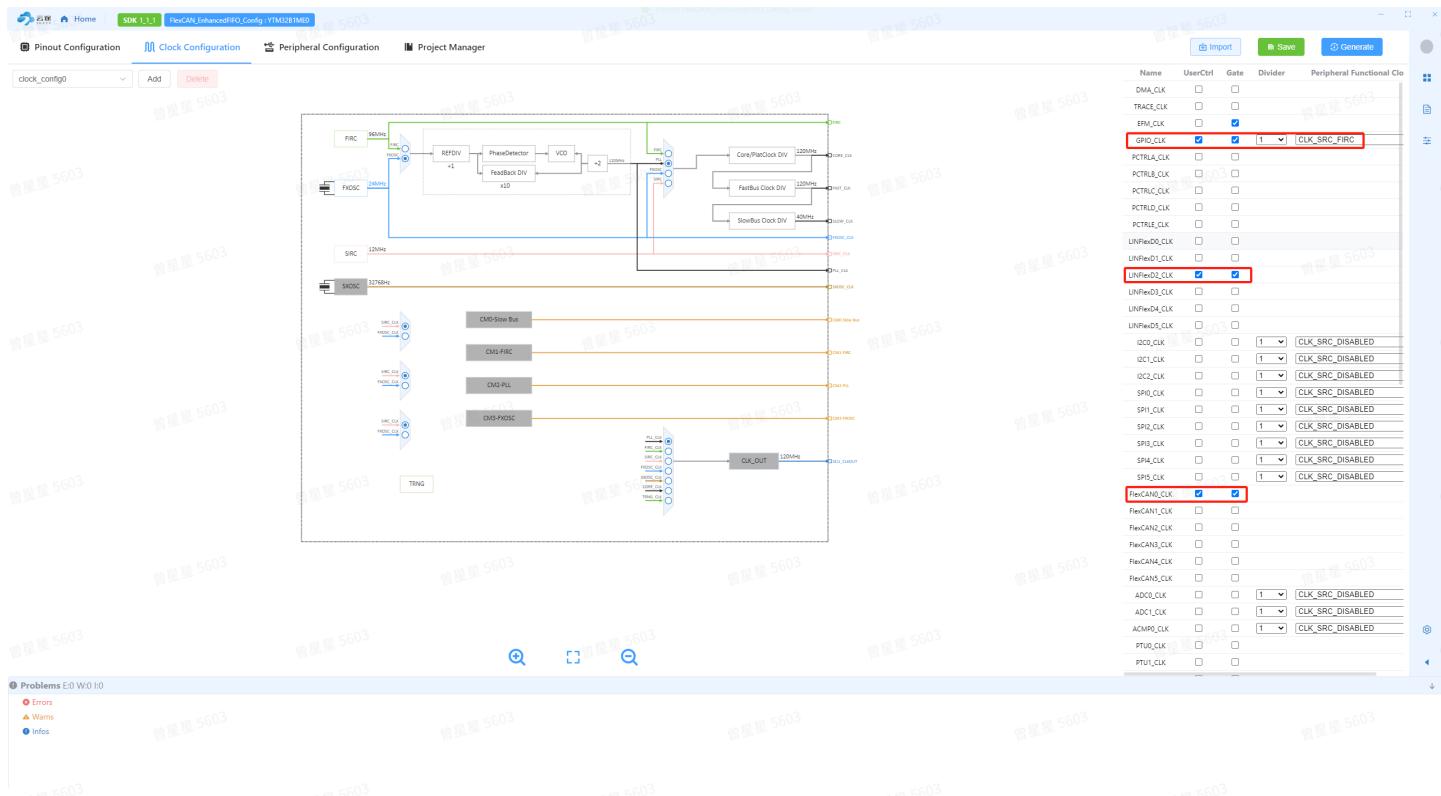
The screenshot shows the 'New Project' interface. In the 'MCU/MPU Selector' section, the YTM32B1M0D part is selected. The 'SDK Version' field, which contains '1.1.1', is highlighted with a red box. The 'Check for config tool and embedded soft...' button is also highlighted with a red box.

3. 配置步骤

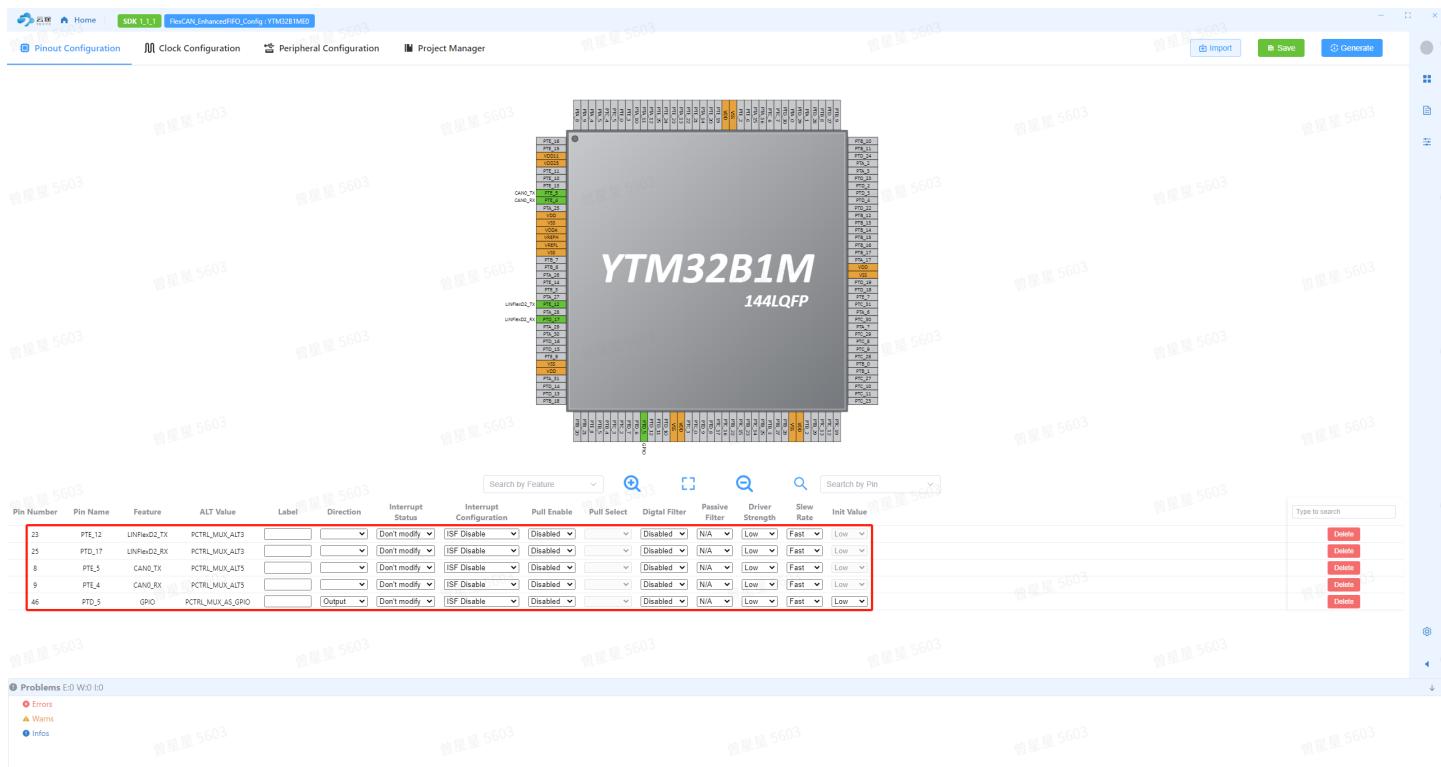
3.1 建立工程

The screenshot shows the 'Project Manager' interface with the 'Clock Configuration' tab selected. The 'Clock' section is expanded, showing various clock settings: FIRC, FXOSC, CM2-PLL, and CM0-Slow Bus. The 'Problems' section at the bottom shows one error related to clock configuration.

3.2 时钟配置

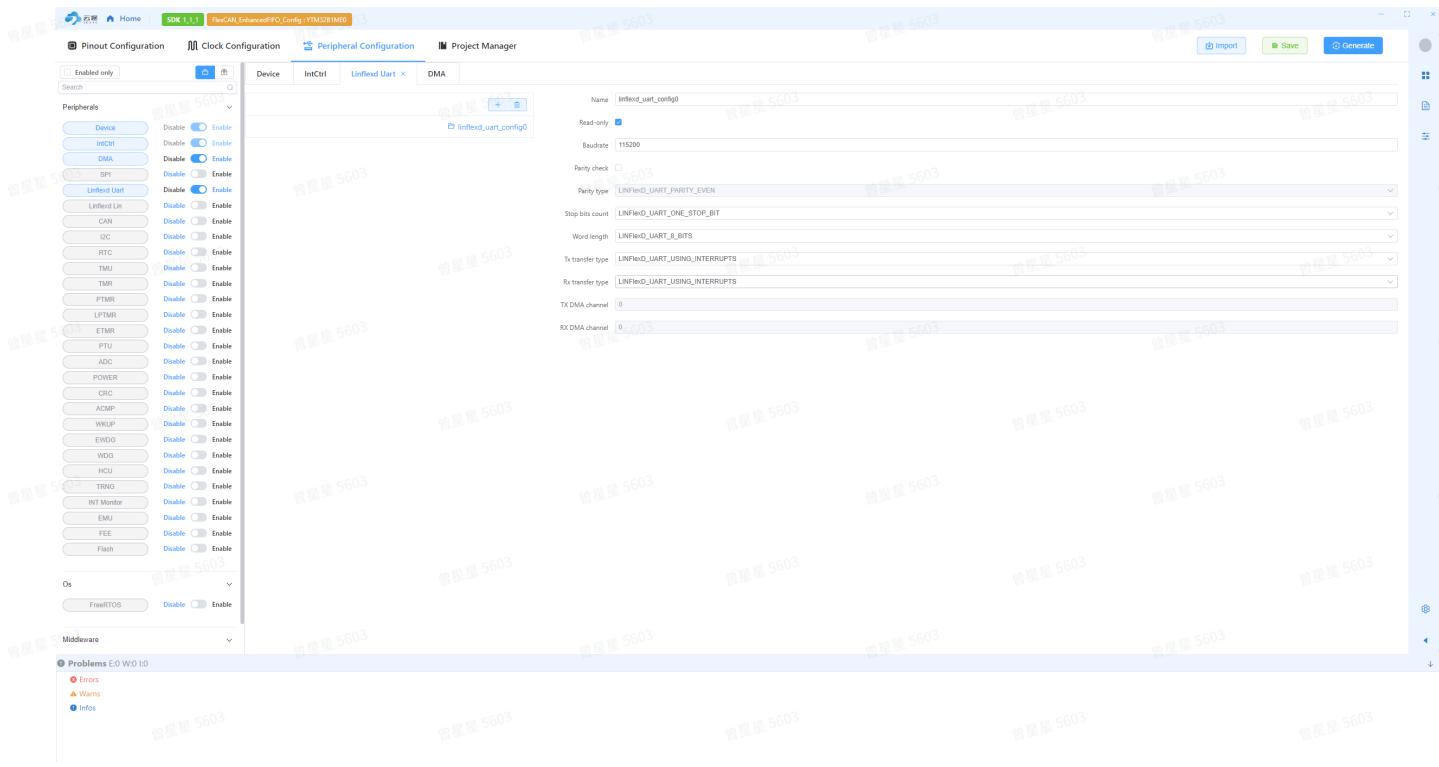


3.3 引脚配置

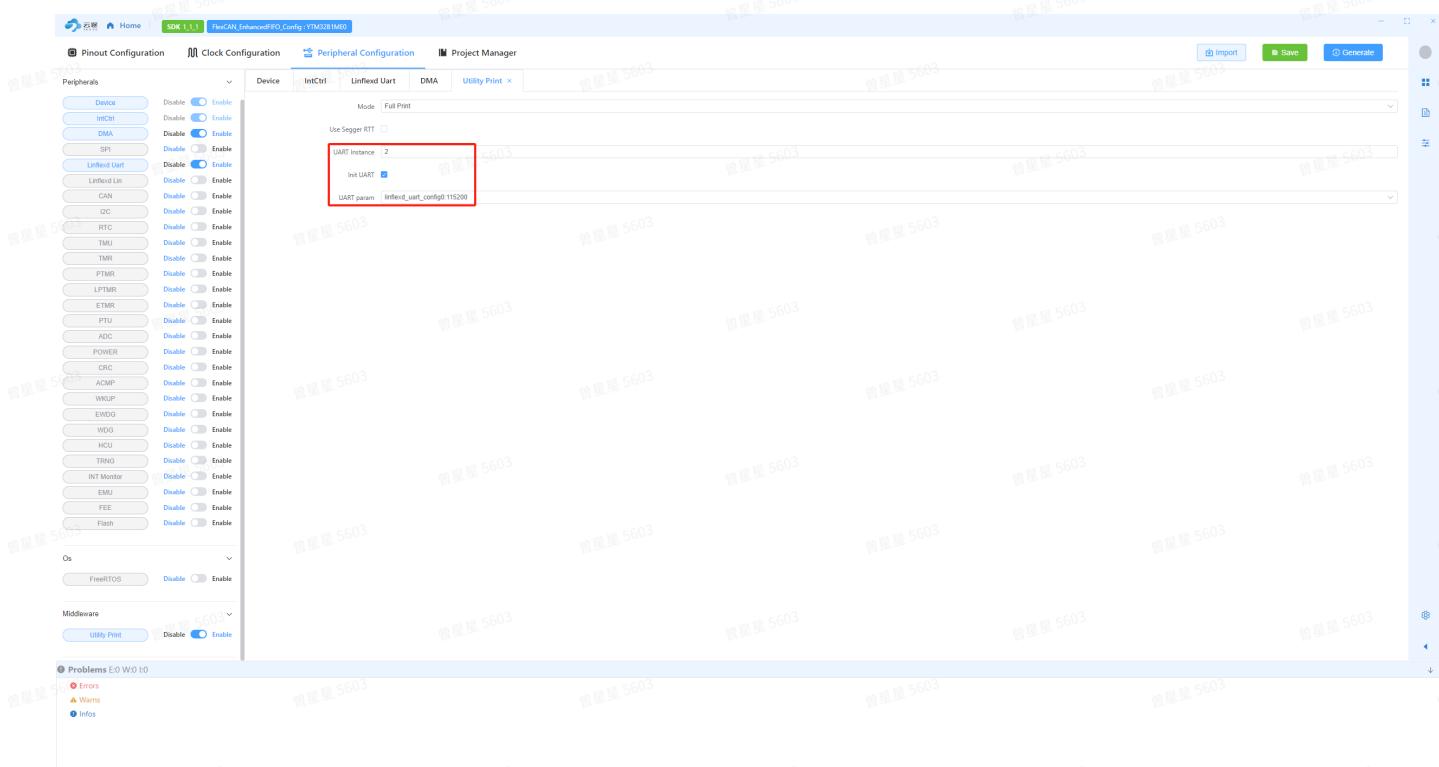


3.4 模块功能配置

3.4.1 UART功能配置



3.4.2 添加打印函数



3.4.3 CAN功能配置

使用enhanced FIFO时，必须要先使能FIFO（勾选下图中Use Rx FIFO配置项），再使能enhanced FIFO，否则SDK中不会打开enhanced FIFO使能控制位。FIFO传输类型设置为中断方式，使能CAN FD模式，设置邮箱负载为64字节，最大邮箱数设置为7，其他参数保持默认配置。

FlexCAN Phase

- Propagation segment: 5
- Phase segment 1: 8
- Phase segment 2: 2
- Prescaler division factor: 3
- Resync jump width: 2
- Brate (Kbps): 500
- Sampling point (%): 87.5

FlexCAN CBT Phase

- Propagation segment: 6
- Phase segment 1: 3
- Phase segment 2: 2
- Prescaler division factor: 1
- Resync jump width: 2
- Brate (Kbps): 2000
- Sampling point (%): 83.333

上图中的The number of enhanced ID filter elements和The number of enhanced extended ID filter elements分别代表过滤器的总数量和扩展帧过滤器的数量，这两个值分别对应下面的寄存器控制位：

Offset: c0ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERFE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 18.69: CAN ERFCR Register Description

Field	Function
22 - 16 NEXIF	Number of Extended ID Filter Elements - Defines the number of extended ID filter elements used during the Enhanced Rx FIFO matching process. NEXIF must be less than or equal to NFE + 1. The number of standard ID filter elements can be determined by the following equation: Number of Standard ID Filter Elements = 2 x (NFE - NEXIF +1) "CAN NEXIF value and number of extended or standard ID filter elements" shows the number of extended ID filters and standard ID filters available for Enhanced Rx FIFO if all the filter elements are used. NEXIF can be written only in Freeze mode because it is blocked by hardware in other modes.
13 - 8 NFE	Number of Enhanced Rx FIFO Filter Elements - Defines the total number of filter elements used during the Enhanced Rx FIFO matching process, as shown in Table 68. NFE can be written only in Freeze mode because it is blocked by hardware in other modes.

NFE的值加1就是实际的滤波器数量，在配置时按照实际的数量配置即可，其他运算会在SDK中进行处理。

CAN FD模式的配置方式以及其他各个基础配置项的介绍请参考“SDK应用_FlexCAN模块配置及应用（一）”。



上面配置中的The number of enhanced Rx FIFO watermarks/The number of Message Buffers Transferred by DMA项必须大于0，否则初始化会报错，进而导致通信失败。

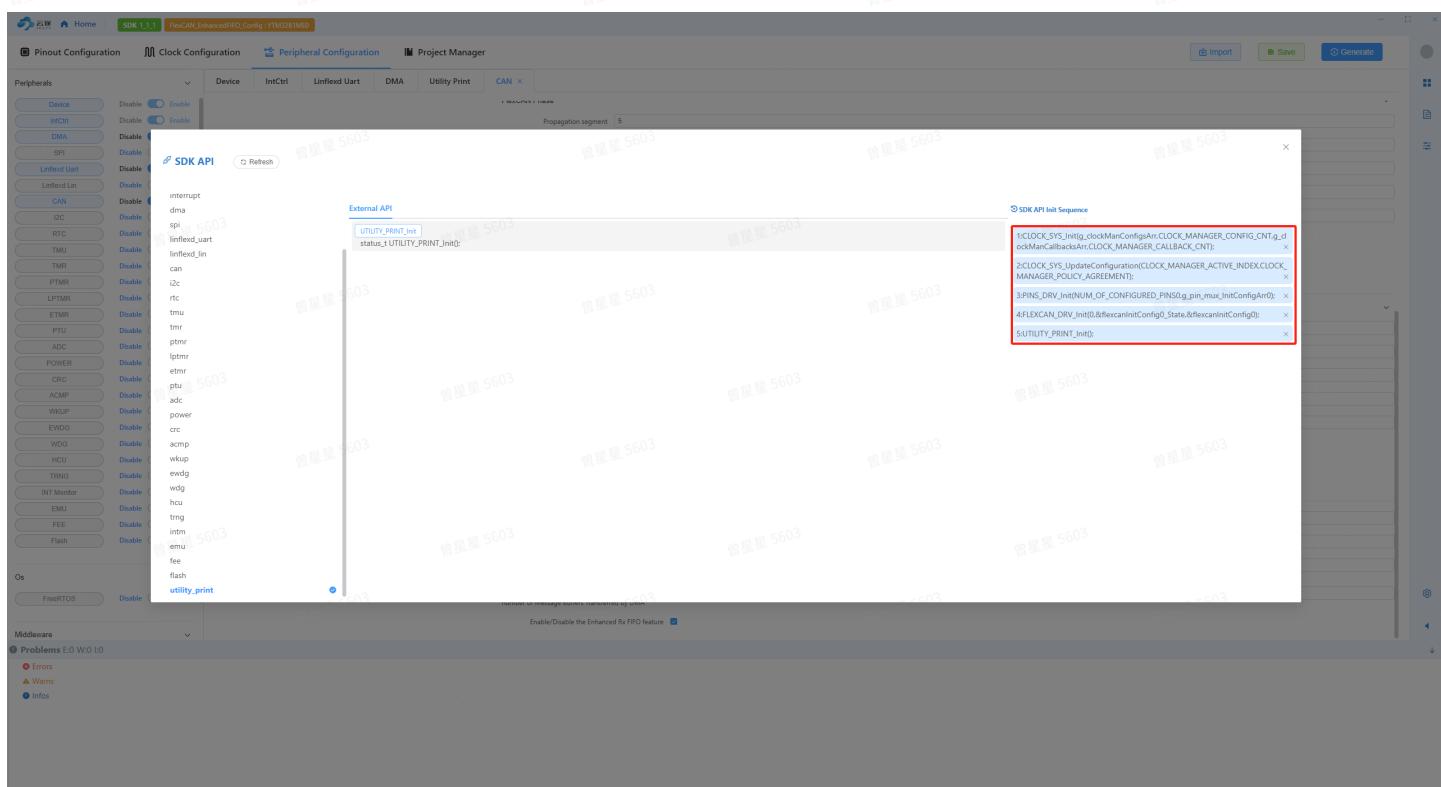


一个滤波器可以设置一个扩展帧ID，或者两个标准帧ID（设置了两个标准帧ID时，The number of enhanced ID filter elements配置项设置1即可）。

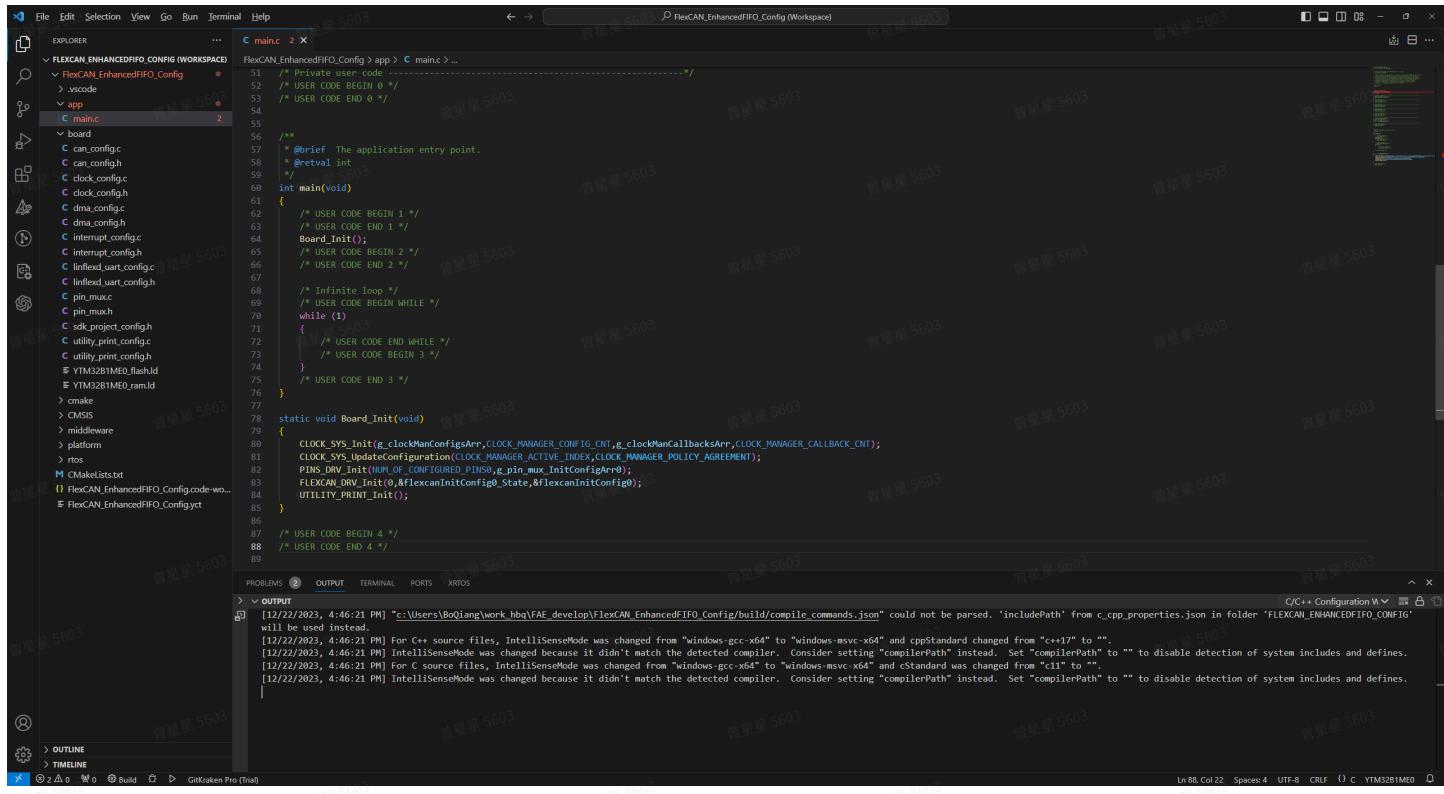
下表展示了NFE的值与可设置的扩展ID或标准ID的对应关系。

NFE值	扩展ID滤波参数	标准ID滤波参数
0	1	2
1	2	4
...
62	63	126
63	64	128

3.5 添加API



4. 生成工程展示



```
FlexCAN_EnhancedFIFO_Config > app > C main.c ...
```

```
/* Private user code */
/* USER CODE BEGIN 0 */
/* USER CODE END 0 */

int main(void)
{
    /* @brief  The application entry point.
    * @retval int
    */
    int main(void)
    {
        /* USER CODE BEGIN 1 */
        /* USER CODE END 1 */
        Board_Init();
        /* USER CODE BEGIN 2 */
        /* USER CODE END 2 */
        /* Infinite loop */
        /* USER CODE BEGIN WHILE */
        while (1)
        {
            /* USER CODE END WHILE */
            /* USER CODE BEGIN 3 */
        }
        /* USER CODE END 3 */
    }
}

static void Board_Init(void)
{
    CLOCK_SYS_Init(g_clockManConfigsArr, CLOCK_MANAGER_CONFIG_CNT, g_clockManCallbacksArr, CLOCK_MANAGER_CALLBACK_CNT);
    CLOCK_SYS_UpdateConfiguration(CLOCK_MANAGER_ACTIVE_INDEX, CLOCK_MANAGER_POLICY_AGREEMENT);
    PINS_DRV_Init(0, 0, PINS0_g_pin_mux_InitConfigArr0);
    FLEXCAN_DRV_Init(0, &flexcanInitConfig0.State, &flexcanInitConfig0);
    UTILITY_PRINT_init();
}

/* USER CODE BEGIN 4 */
/* USER CODE END 4 */

```

PROBLEMS OUTPUT TERMINAL PORTS XRTOS

```
[12/22/2023, 4:46:21 PM] "c:\Users\BoQiang\work\hbg\FAE\develop\FlexCAN_EnhancedFIFO_Config\build\compile_commands.json" could not be parsed. 'includePath' from c_cpp_properties.json in folder 'FLEXCAN_ENHANCEDFIFO_CONFIG' will be used instead.
[12/22/2023, 4:46:21 PM] For C++ source files, IntelliSenseMode was changed from "windows-gcc-x64" to "windows-msvc-x64" and cppStandard changed from "c++17" to "".
[12/22/2023, 4:46:21 PM] IntelliSenseMode was changed because it didn't match the detected compiler. Consider setting "compilerPath" instead. Set "compilerPath" to "" to disable detection of system includes and defines.
[12/22/2023, 4:46:21 PM] For C source files, IntelliSenseMode was changed from "windows-gcc-x64" to "windows-msvc-x64" and cStandard was changed from "c11" to "".
[12/22/2023, 4:46:21 PM] IntelliSenseMode was changed because it didn't match the detected compiler. Consider setting "compilerPath" instead. Set "compilerPath" to "" to disable detection of system includes and defines.
```

Ln 88, Col 22 Spaces: 4 UTF-8 CRLF C YTM32B1ME0

5. 应用实例

本例使用了FlexCAN的enhanced FIFO功能，配置了5个滤波ID，包括3个标准帧ID，2个扩展帧ID。主程序中利用发送邮箱（编号为0）发送ID为0x10FF1369的扩展帧，周期为100ms，目标接收标准帧报文ID为0x310、0x420和0x530，扩展帧报文ID为0x0D314450和0x10FF1063。

5.1 通用配置

YT Config Tool不支持直接配置CAN enhanced FIFO模式的过滤器，需要用户手动添加，添加方式如下：

- 设置FIFO滤波器数组。由于前面功能配置中设置了filter参数为5，因此这里需要定义一个长度为5的滤波器数组。

```
flexcan_enhance_rx_fifo_filter_table_t can_enhancefifo_filter[] =  
{  
    {  
        .isRemoteFrame = false,  
        .isExtendedFrame = false,  
        .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,  
        .filter_table.scheme_filter_mask =  
        {  
            .filter = 0x310U,  
            .mask = 0x7FFU,  
        },  
    },  
    {  
        .isRemoteFrame = false,  
        .isExtendedFrame = false,  
        .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,  
        .filter_table.scheme_filter_mask =  
        {  
            .filter = 0x420U,  
            .mask = 0x7FFU,  
        },  
    },  
    {  
        .isRemoteFrame = false,  
        .isExtendedFrame = false,  
        .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,  
        .filter_table.scheme_filter_mask =  
        {  
            .filter = 0x530U,  
            .mask = 0x7FFU,  
        },  
    },  
    {  
        .isRemoteFrame = false,  
        .isExtendedFrame = true,  
        .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,  
        .filter_table.scheme_filter_mask =  
        {  
            .filter = 0x0D314450U,  
            .mask = 0x1FFFFFFFU,  
        },  
    },  
    {  
        .isRemoteFrame = false,  
        .isExtendedFrame = true,  
        .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,  
        .filter_table.scheme_filter_mask =  
        {  
            .filter = 0x10FF1063U,  
            .mask = 0x1FFFFFFFU,  
        },  
    },  
};
```

一个滤波器数组元素需要设置以下几个参数：

- isRemoteFrame, 该帧是否为远程帧 (bool型变量)
 - isExtendeFrame, 该帧是否为扩展帧 (bool型变量)
 - filterScheme, 滤波器格式选择
 - filter_table, 这是一个联合体, 可根据上面配置的滤波器格式, 设置相应过滤ID和掩码

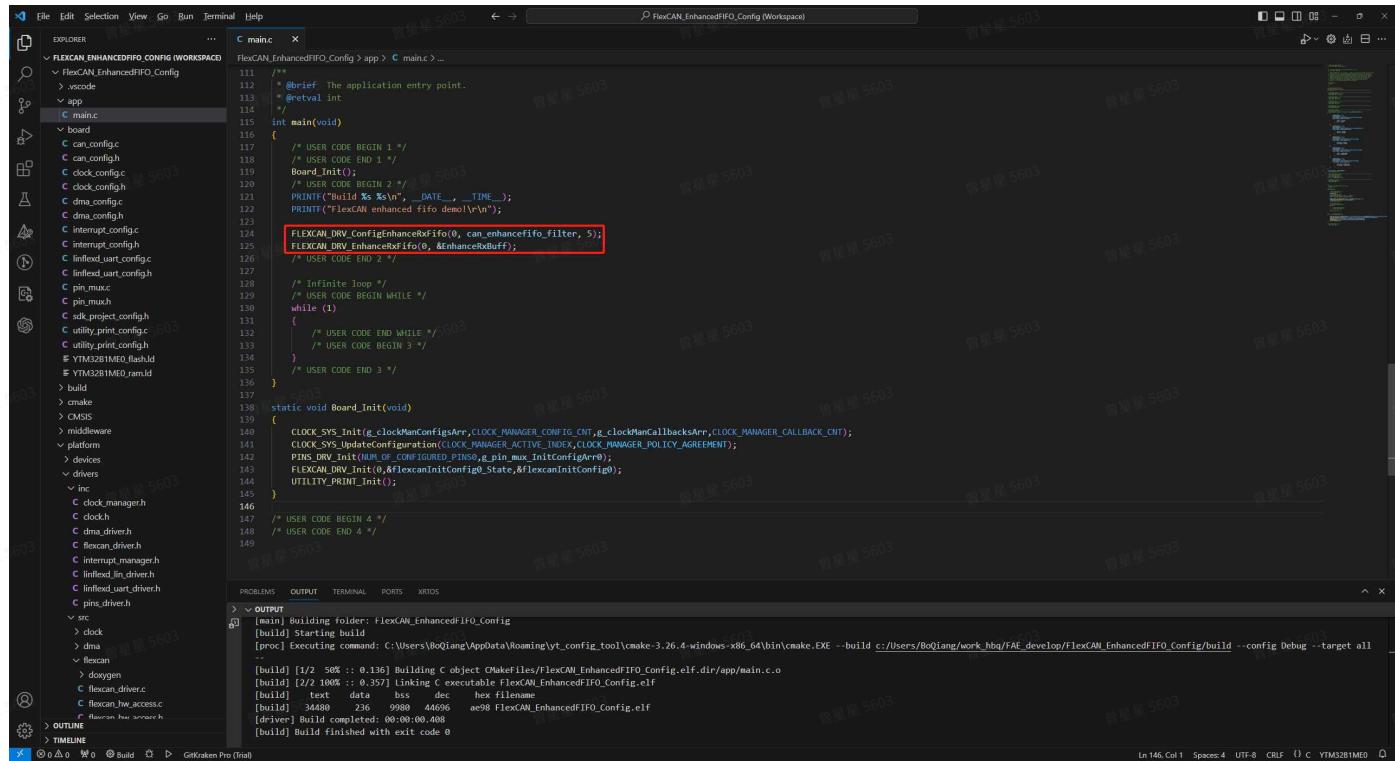
上图中的filterScheme项可以配置滤波器的格式，有以下三种方式可选：

```
typedef enum{
    FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,
    FLEXCAN_ENAHNCE_RXFIFO_FSCH_RANGE,
    FLEXCAN_ENAHNCE_RXFIFO_FSCH_DOUBLE_FILTER,
}flexcan_enchance_rx_fifo_filter_scheme_t;
```

FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK表示过滤ID加掩码方式，这时需要设置一个过滤ID和这个ID的独立掩码（filter和mask）；FLEXCAN_ENAHNCE_RXFIFO_FSCH_RANGE表示范围过滤方式，这时需要分别设置一个低值和一个高值（filterLow和filterHigh），CAN控制器只能接收ID在这两个值之间的报文，其他ID的报文都会被过滤掉；

FLEXCAN_ENAHNCE_RXFIFO_FSCH_DOUBLE_FILTER表示双滤波器模式，此时需要设置两个过滤ID（filter1和filter2），这两个ID同时生效。

b. 配置FIFO滤波器以及开启数据接收。这里与legacy FIFO的配置类似，详细介绍请参考“SDK文档 - FlexCAN模块配置及应用（二）”。



5.2 配置数据的轮询发送和接收，周期100ms

```

int main(void)
{
    /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */
    Board_Init();
    /* USER CODE BEGIN 2 */
    PRINTF("Build %s %s\n", __DATE__, __TIME__);
    PRINTF("FlexCAN enhanced fifo demo!\r\n");

    FLEXCAN_DRV_ConfigEnhanceRxFifo(0, can_enhancefifo_filter, 5);
    FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);

    for (uint8_t i =0; i < 64; i++)
    {
        TxMsgBuff.data[i] = i;
    }

    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, 0))
        {
            FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
        }
        if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, (uint8_t)FLEXCAN_MB_HANDLE_ENHANCE_RXFIFO))
        {
            FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
        }
        OSIF_TimeDelay(100);
    }
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
}

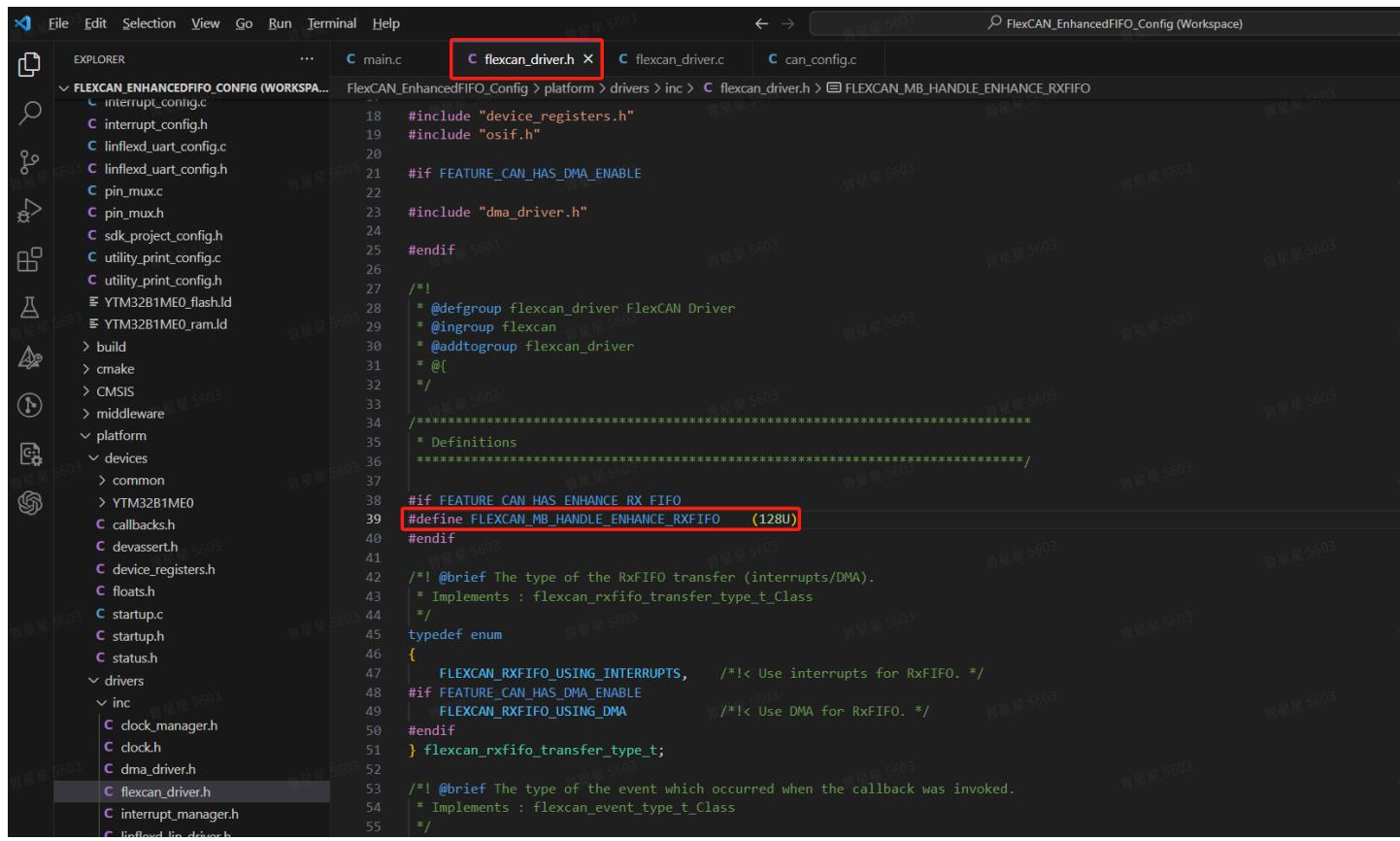
/* USER CODE END 3 */

```

设置要发送的数据

循环调用数据发送和增强型FIFO接收函数

上图中的宏定义FLEXCAN_MB_HANDLE_ENHANCE_RXFIFO是enhanced rx FIFO的邮箱软件编号，其定义在flexcan_driver.h头文件中，如下图：



```
File Edit Selection View Go Run Terminal Help
EXPLORER ... C main.c C flexcan_driver.h x C flexcan_driver.c C can_config.c
FLEXCAN_ENHANCEDFIFO_CONFIG (Workspace)
interrupt_config.c
interrupt_config.h
linflexd_uart_config.c
linflexd_uart_config.h
pin_mux.c
pin_mux.h
sdk_project_config.h
utility_print_config.c
utility_print_config.h
YTM32B1ME0_flash.ld
YTM32B1ME0_ram.ld
> build
> cmake
> CMSIS
> middleware
platform
> devices
> common
> YTM32B1ME0
callbacks.h
devassert.h
device_registers.h
floats.h
startup.c
startup.h
status.h
> drivers
> inc
clock_manager.h
clock.h
dma_driver.h
flexcan_driver.h
interrupt_manager.h
linflexd_uart_driver.h
FlexCAN_EnhancedFIFO_Config > platform > drivers > inc > flexcan_driver.h > FLEXCAN_MB_HANDLE_ENHANCE_RXFIFO
18 #include "device_registers.h"
19 #include "osif.h"
20
21 #if FEATURE_CAN_HAS_DMA_ENABLE
22 #include "dma_driver.h"
23
24 #endif
25
26
27 */
28 * @defgroup flexcan_driver FlexCAN Driver
29 * @ingroup flexcan
30 * @addtogroup flexcan_driver
31 * @{
32 */
33 ****
34 * Definitions
35 ****
36
37 #if FEATURE_CAN_HAS_ENHANCE_RX_FIFO
38 #define FLEXCAN_MB_HANDLE_ENHANCE_RXFIFO (128U)
39
40 #endif
41
42 /*! @brief The type of the RxFIFO transfer (interrupts/DMA).
43 * Implements : flexcan_rxfifo_transfer_type_t_Class
44 */
45
46 typedef enum
47 {
48     FLEXCAN_RXFIFO_USING_INTERRUPTS, /*!< Use interrupts for RxFIFO. */
49     #if FEATURE_CAN_HAS_DMA_ENABLE
50     FLEXCAN_RXFIFO_USING_DMA /*!< Use DMA for RxFIFO. */
51     #endif
52 } flexcan_rxfifo_transfer_type_t;
53
54 /*! @brief The type of the event which occurred when the callback was invoked.
55 * Implements : flexcan_event_type_t_Class
56 */
57
```

5.3 配置数据的中断发送和接收

```
void can_callback(uint8_t instance, flexcan_event_type_t eventType, \
                  uint32_t buffIdx, flexcan_state_t *flexcanState)
{
    (void)instance;
    (void)flexcanState;
    switch (eventType)
    {
        case FLEXCAN_EVENT_TX_COMPLETE:
            FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
            break;
        case FLEXCAN_EVENT_ENHANCE_RXFIFO_AVAILABLEDATA:
            FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
        default:
            break;
    }
}
```

定义中断回调函数

```
/* USER CODE END PFDC */
static void Board_Init(void);
```

```
/* Private user code ----- */
/* USER CODE BEGIN 0 */
/* USER CODE END 0 */
```

```
/***
 * @brief The application entry point.
 * @retval int
 */
```

```
int main(void)
```

```
{
    /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */
    Board_Init();
    /* USER CODE BEGIN 2 */
```

```
    PRINTF("Build %s %s\n", __DATE__, __TIME__);
    PRINTF("FlexCAN enhanced fifo demo!\r\n");
```

```
#if USE_INTERRUPT
```

```
    FLEXCAN_DRV_InstallEventCallback(0, can_callback, NULL);
```

```
#endif
```

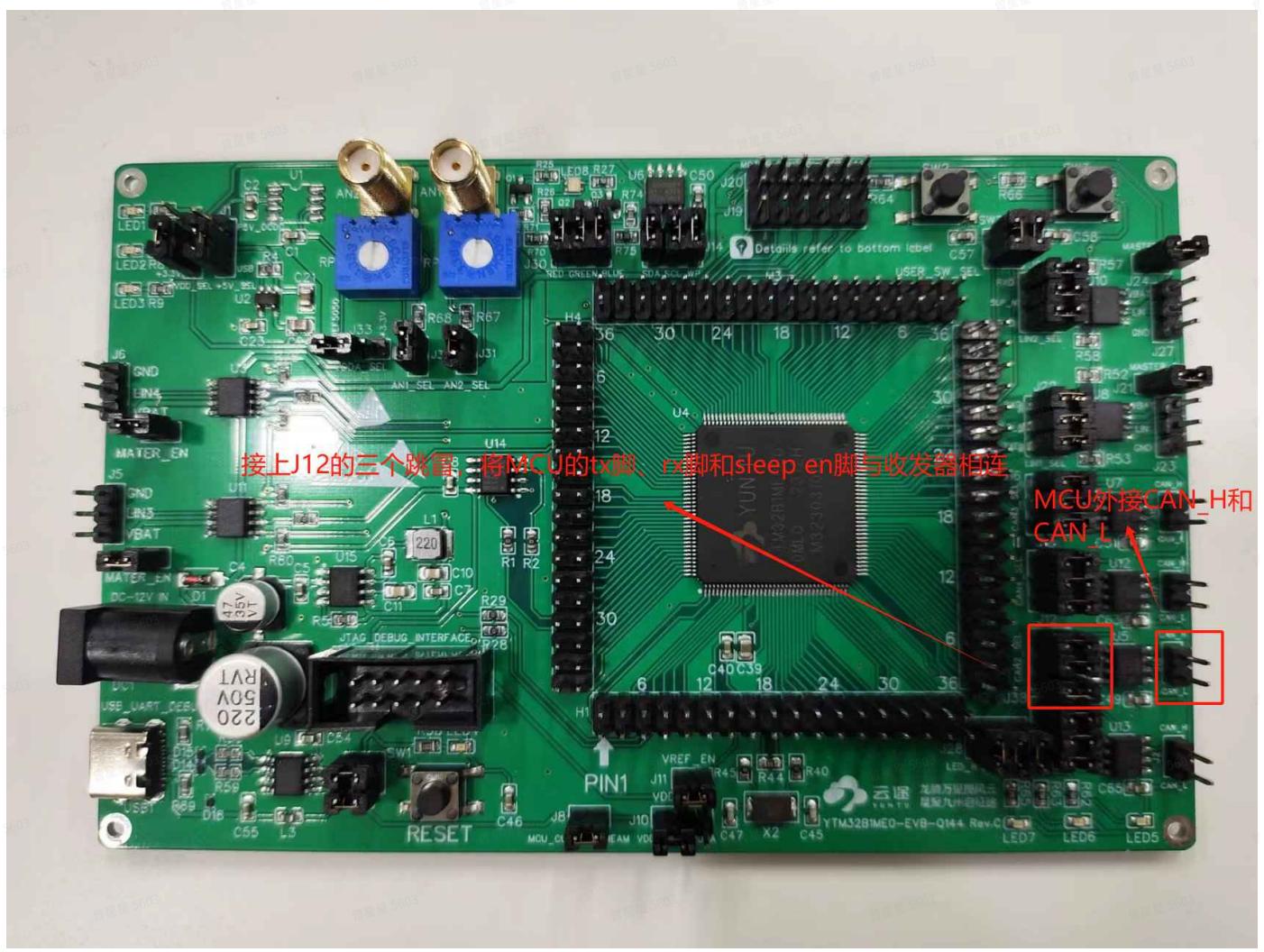
```
    FLEXCAN_DRV_ConfigEnhanceRxFifo(0, can_enhancefifo_filter, 4);
```

```
    FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
```

注册中断回调函数

5.4 通信结果展示

以ME0的demo板YTM32B1M-EVB-Q144为例进行演示，CAN分析仪使用TSMaster，TSMaster的CAN_H、CAN_L分别连接到demo板上J13的CAN_H和CAN_L，如下图中箭头所指位置，上面为CAN_H，下面为CAN_L：



5.4.1 轮询方式

- i. 上位机收到ID为0x10FF1369的报文，数据长度为64，周期100ms。

- ii. demo板能成功接收到滤波表中设置的所有报文。

行	发送	触发	报文名称	标识符	通道	类型	DLC	BRS	D0	D1	D2	D3	D4	D5	D6	D7	注释
1	▶	手动	NewMsg	530	1	标准数据帧	8	0	10 00 13 00 00 00 00 10								
2	▶	手动	NewMsg	0D314450	1	扩展数据帧	8	0	30 31 32 33 00 00 00 30								
3	▶	手动	NewMsg	10FF1063	1	扩展数据帧	8	0	21 67 35 68 91 22 00 31								
4	▶	手动	NewMsg	310	1	标准数据帧	8	0	00 02 00 00 00 00 00 00								
5	▶	手动	NewMsg	420	1	标准数据帧	8	0	01 00 00 00 00 00 00 00								

Ozone - The J-Link Debugger V3.22b - C:/Users/BoQiang/work_hbq/FAE_develop/FlexCAN_EnhancedFIFO_Config/build/FlexCAN_EnhancedFIFO_Config.elf

File View Find Debug Tools Window Help

Break & Tracepoints

Type: Code Location: main.c:162 Extras: FLEXCAN_DRV

File Scope: f main

```

131 * @brief The application entry point.
132 * @retval int
133 */
134 int main(void)
135 {
136     /* USER CODE BEGIN 1 */
137     /* USER CODE END 1 */
138     Board_Init();
139     /* USER CODE BEGIN 2 */
140     PRINTF("Build %s %s\n", __DATE__, __TIME__);
141     PRINTF("FlexCAN enhanced fifo demo!\r\n");
142
143     FLEXCAN_DRV_ConfigEnhanceRxFifo(0, can_enhancefifo_filter, 5);
144     FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
145
146     for (uint8_t i = 0; i < 64; i++)
147     {
148         TxMsgBuff.data[i] = i;
149     }
150     /* USER CODE END 2 */
151
152     /* Infinite loop */
153     /* USER CODE BEGIN WHILE */
154     while (1)
155     {
156         if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, 0))
157         {
158             FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
159         }
160         if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, (uint8_t)FLEXCAN_MB_HANDLE_ENHANCE_RXFIFO))
161         {
162             FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
163         }
164         OSIF_TimeDelay(100);
165         /* USER CODE END WHILE */
166         /* USER CODE BEGIN 3 */
167     }
168     /* USER CODE END 3 */
169 }
170
171 static void Board_Init(void)
172 {
173     CLOCK_Sys_Init(g_clockManConfigsArr, CLOCK_MANAGER_CONFIG_CNT, g_clockManCallbacksArr, CLOCK_MANAGER_CONFIG_CALLBACKS_CNT);
174     CLOCK_Sys_UpdateConfiguration(CLOCK_MANAGER_ACTIVE_INDEX, CLOCK_MANAGER_POLICY AGREEMENT);
175     PINS_DRV_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_initConfigArr0);
176     FLEXCAN_DRV_Init(0, &flexcanInitConfig0_State, &flexcanInitConfig0);
177     UTILITY_PRINT_Init();
178 }

```

Watched Data 1

Expression	Value
EnhanceRxBuff	574 450
msgId	0000 0310
data	"\"001"\002"\002"
[0]	1 ('001')
[1]	2 ('002')
[2]	2 ('002')
[3]	0 ('0')
[4]	0 ('0')
[5]	0 ('0')
[6]	0 ('0')
[7]	0 ('0')
[8]	0 ('0')
[9]	0 ('0')
[10]	0 ('0')
[11]	0 ('0')
[12]	0 ('0')
[13]	0 ('0')
[14]	0 ('0')
[15]	0 ('0')
[16]	0 ('0')
[17]	0 ('0')
[18]	0 ('0')
[19]	0 ('0')
[20]	0 ('0')
[21]	0 ('0')

Ozone - The J-Link Debugger V3.22b - C:/Users/BoQiang/work_hbq/FAE_develop/FlexCAN_EnhancedFIFO_Config/build/FlexCAN_EnhancedFIFO_Config.elf

File View Find Debug Tools Window Help

Break & Tracepoints

Type Location Extras

Code main.c:162 FLEXCAN_DRV_End

File Scope

104 },
105 },
106 {
107 .isRemoteFrame = false,
108 .isExtendedFrame = true,
109 .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCN_FILTER_MASK,
110 .filter_table.scheme_filter_mask =
111 {
112 .filter = 0x10FF1063U,
113 .mask = 0x1FFFFFFU,
114 },
115 },
116 };
117 /* USER CODE END PV */
118 /* Private function declare ----- */
119 /* USER CODE BEGIN FFDC */
120 /* USER CODE END FFDC */
121 static void Board_Init(void);
122 /* Private user code ----- */
123 /* USER CODE BEGIN 0 */
124 /* USER CODE END 0 */
125 /* USER CODE END PV */
126 /* USER CODE BEGIN 1 */
127 /* USER CODE END 1 */
128 /*
129 * @brief The application entry point.
130 * @retval int
131 */
132 int main(void)
133 {
134 /* USER CODE BEGIN 1 */
135 /* USER CODE END 1 */
136 Board_Init();
137 /* USER CODE BEGIN 2 */
138 /* FLEXCAN_Drv_EnhanceRxFifo(0, &EnhanceRxBuff); */
139 for (uint8_t i = 0; i < 64; i++)
140 {
141 TxMsgBuff.data[i] = i;
142 }
143 /* USER CODE END 2 */
144 /* Infinite loop */
145 /* USER CODE BEGIN WHILE */
146 while (1)
147 {
148 if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, 0))
149 {
150 FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
151 }
152 if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, (uint8_t)FLEXCAN_MB_HANDLE_ENHANCE_RXFIFO))
153 {
154 FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
155 }
156 }
157 /* USER CODE END WHILE */
158 /* USER CODE BEGIN 3 */
159 /* FLEXCAN_Drv_EnhanceRxFifo(0, &EnhanceRxBuff); */
160 /* Infinite loop */
161 /* USER CODE END 3 */
162 /* USER CODE BEGIN 4 */
163 /* OSIF_Timedelay(100); */
164 /* USER CODE END 4 */
165 /* USER CODE BEGIN WHILE */
166 /* USER CODE END WHILE */
167 /* USER CODE BEGIN 5 */
168 /* Infinite loop */
169 /* USER CODE END 5 */
170 /* USER CODE BEGIN 6 */
171 static void Board_Init(void)
172 {
173 CLOCK_SYS_Init(g_clockManConfigsArr, CLOCK_MANAGER_CONFIG_CNT, g_clockManCallbacksArr, CLOCK_MANAGER_CALLBACK_CNT);
174 CLOCK_SYS_UpdateConfiguration(CLOCK_MANAGER_ACTIVE_INDEX, CLOCK_MANAGER_POLICY AGREEMENT);
175 PINS_DRV_Init(NUm_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0);
176 FLEXCAN_DRV_Init(0, &flexcanInitConfig0_State, &flexcanInitConfig0);
177 UTILITY_PRINT_Init();
178 }
179 }

Watched Data 1

Expression	Value
EnhanceRxBuff	587 844
cs	0000 0420
wrid	"'\001'"
data	1 ("'\001")
[0]	0 ("'\0")
[1]	0 ("'\0")
[2]	0 ("'\0")
[3]	0 ("'\0")
[4]	0 ("'\0")
[5]	0 ("'\0")
[6]	0 ("'\0")
[7]	0 ("'\0")
[8]	0 ("'\0")
[9]	0 ("'\0")
[10]	0 ("'\0")
[11]	0 ("'\0")
[12]	0 ("'\0")
[13]	0 ("'\0")
[14]	0 ("'\0")
[15]	0 ("'\0")
[16]	0 ("'\0")
[17]	0 ("'\0")
[18]	0 ("'\0")
[19]	0 ("'\0")
[20]	0 ("'\0")
[21]	0 ("'\0")
[22]	0 ("'\0")
[23]	0 ("'\0")
[24]	0 ("'\0")
[25]	0 ("'\0")
[26]	0 ("'\0")
[27]	0 ("'\0")
[28]	0 ("'\0")
[29]	0 ("'\0")
[30]	0 ("'\0")
[31]	0 ("'\0")
[32]	0 ("'\0")
[33]	0 ("'\0")
[34]	0 ("'\0")
[35]	0 ("'\0")
[36]	0 ("'\0")
[37]	0 ("'\0")
[38]	0 ("'\0")
[39]	0 ("'\0")
[40]	0 ("'\0")
[41]	0 ("'\0")
[42]	0 ("'\0")
[43]	0 ("'\0")
[44]	0 ("'\0")
[45]	0 ("'\0")
[46]	0 ("'\0")
[47]	0 ("'\0")

Ozone - The J-Link Debugger V3.22b - C:/Users/BoQiang/work_hbq/FAE_develop/FlexCAN_EnhancedFIFO_Config/build/FlexCAN_EnhancedFIFO_Config.elf

File View Find Debug Tools Window Help

Break & Tracepoints main.c x

Type Location Extras

Code main.c:162 FLEXCAN_DRV_End

```

104     },
105     },
106     {
107         .isRemoteFrame = false,
108         .isExtendedFrame = true,
109         .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCHE_FILTER_MASK,
110         .filter_table.scheme_filter_mask =
111             {
112                 .filter = 0x10FF1063U,
113                 .mask = 0x1FFFFFFFU,
114             },
115     },
116 };
117 /* USER CODE END PV */
118
119 /* Private function declare */
120 /* USER CODE BEGIN FFDC */
121 /* USER CODE END FFDC */
122 /* USER CODE BEGIN PFDC */
123 static void Board_Init(void);
124
125 /* Private user code */
126 /* USER CODE BEGIN 0 */
127 /* USER CODE END 0 */
128
129 /**
130  * @brief The application entry point.
131  * @retval int
132  */
133 int main(void)
134 {
135     /* USER CODE BEGIN 1 */
136     /* USER CODE END 1 */
137     Board_Init();
138     /* USER CODE BEGIN 2 */
139     PRINTF("Build %s %s\n", __DATE__, __TIME__);
140     PRINTF("FlexCAN enhanced fifo demo!\r\n");
141
142     FLEXCAN_DRV_ConfigEnhanceRxFifo(0, can_enhancefifo_filter, 5);
143     FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
144
145     for (uint8_t i = 0; i < 64; i++)
146     {
147         TxMsgBuff.data[i] = i;
148     }
149     /* USER CODE END 2 */
150
151     /* Infinite loop */
152     /* USER CODE BEGIN WHILE */
153     while (1)
154     {
155         if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, 0))
156         {
157             FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
158         }
159         if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, (uint8_t)FLEXCAN_MB_HANDLE_ENHANCE_RXFIFO))
160         {
161             FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
162         }
163     }
164     OSIF_TimeDelay(100);
165     /* USER CODE END WHILE */
166     /* USER CODE BEGIN 3 */
167 }
168 /* USER CODE END 3 */
169
170
171 static void Board_Init(void)
172 {
173     CLOCK_SYS_Init(g_clockManConfigsArr, CLOCK_MANAGER_CONFIG_CNT, g_clockManCallbacksArr, CLOCK_MANAGER_CALLBACK_CNT);
174     CLOCK_SYS_UpdateConfiguration(CLOCK_MANAGER_ACTIVE_INDEX, CLOCK_MANAGER_POLICY AGREEMENT);
175     PINS_DRV_Init(NUm_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0);
176     FLEXCAN_DRV_Init(0, &flexcanInitConfig0_State, &flexcanInitConfig0);
177     UTILITY_PRINT_Init();
178 }

```

Source Files

File Status

- _default_types.h included
- _stdint.h included
- callbacks.h included
- can_config.c external
- can_config.h included
- clock_config.c external
- clock config.h included

Watched Data 1

Expression	Value
EnhanceRxBuff	544 461
cs	0000 0530
msgId	0000 0530
data	0000 0200
[0]	16 ("\\01E")
[1]	0 ("\\0")
[2]	19 ("\\019")
[3]	0 ("\\0")
[4]	0 ("\\0")
[5]	0 ("\\0")
[6]	0 ("\\0")
[7]	16 ("\\01E")
[8]	0 ("\\0")
[9]	0 ("\\0")
[10]	0 ("\\0")
[11]	0 ("\\0")
[12]	0 ("\\0")
[13]	0 ("\\0")
[14]	0 ("\\0")
[15]	0 ("\\0")
[16]	0 ("\\0")
[17]	0 ("\\0")
[18]	0000 0000
[19]	0 ("\\0")
[20]	0 ("\\0")
[21]	0 ("\\0")
[22]	0 ("\\0")
[23]	0 ("\\0")
[24]	0 ("\\0")
[25]	0 ("\\0")
[26]	0 ("\\0")
[27]	0 ("\\0")
[28]	0 ("\\0")
[29]	0 ("\\0")
[30]	0 ("\\0")
[31]	0 ("\\0")
[32]	0 ("\\0")
[33]	0 ("\\0")
[34]	0 ("\\0")
[35]	0 ("\\0")
[36]	0 ("\\0")
[37]	0 ("\\0")
[38]	0 ("\\0")
[39]	0 ("\\0")
[40]	0 ("\\0")
[41]	0 ("\\0")
[42]	0 ("\\0")
[43]	0 ("\\0")
[44]	0 ("\\0")
[45]	0 ("\\0")
[46]	0 ("\\0")
[47]	0 ("\\0")

Ozone - The J-Link Debugger V3.22b - C:/Users/BoQiang/work_hbq/FAE_develop/FlexCAN_EnhancedFIFO_Config/build/FlexCAN_EnhancedFIFO_Config.elf

File View Find Debug Tools Window Help

Break & Tracepoints

Type Location Extras

Code main.c 162 FLEXCAN_DRV_End

main.c

104 },
105 },
106 {
107 .isRemoteFrame = false,
108 .isExtendedFrame = true,
109 .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,
110 .filter_table.scheme_filter_mask =
111 {
112 .filter = 0x10FF1063U,
113 .mask = 0x1FFFFFFFU,
114 },
115 },
116 };
117
118 /* USER CODE END PV */
119
120 /* Private function declare */
121 /* USER CODE BEGIN FFDC */
122 /* USER CODE END FFDC */
123 static void Board_Init(void);
124
125 /* Private user code */
126 /* USER CODE BEGIN 0 */
127 /* USER CODE END 0 */
128
129 /**
130 * @brief The application entry point.
131 * @retval int
132 */
133 int main(void)
134 {
135 /* USER CODE BEGIN 1 */
136 /* USER CODE END 1 */
137 Board_Init();
138 /* USER CODE BEGIN 2 */
139 PRINTF("Build %s %s\n", __DATE__, __TIME__);
140 PRINTF("FlexCAN enhanced fifo demo!\r\n");
141
142 FLEXCAN_DRV_ConfigEnhanceRxFifo(0, can_enhancefifo_filter, 5);
143 FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
144
145 for (uint8_t i = 0; i < 64; i++)
146 {
147 TxMsgBuff.data[i] = i;
148 }
149 /* USER CODE END 2 */
150
151 /* Infinite loop */
152 /* USER CODE BEGIN WHILE */
153 while (1)
154 {
155 if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, 0))
156 {
157 FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
158 }
159 if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, (uint8_t)FLEXCAN_MB_HANDLE_ENHANCE_RXFIFO))
160 {
161 /* USER CODE END 3 */
162 FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
163 }
164 OSIF_TimeDelay(100);
165 /* USER CODE END WHILE */
166 /* USER CODE BEGIN 3 */
167 }
168 /* USER CODE END 3 */
169 }
170
171 static void Board_Init(void)
172 {
173 CLOCK_SYS_Init(g_clockManConfigsArr, CLOCK_MANAGER_CONFIG_CNT, g_clockManCallbacksArr, CLOCK_MANAGER_CALLBACK_CNT);
174 CLOCK_SYS_UpdateConfiguration(CLOCK_MANAGER_ACTIVE_INDEX, CLOCK_MANAGER_POLICY AGREEMENT);
175 PINS_DRV_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0);
176 FLEXCAN_DRV_Init(0, &flexcanInitConfig0_State, &flexcanInitConfig0);
177 UTILITY_PRINT_Init();
178 }

Watched Data 1

Expression	Value
EnhanceRxBuff	6863435
cz	6863435
msgId	0D014450
data	0x1234
[0]	48 ('0')
[1]	49 ('1')
[2]	50 ('2')
[3]	51 ('3')
[4]	0 ('0')
[5]	0 ('0')
[6]	0 ('0')
[7]	48 ('0')
[8]	0 ('0')
[9]	0 ('0')
[10]	0 ('0')
[11]	0 ('0')
[12]	0 ('0')
[13]	0 ('0')
[14]	0 ('0')
[15]	0 ('0')
[16]	0 ('0')
[17]	0 ('0')
[18]	0 ('0')
[19]	0 ('0')
[20]	0 ('0')
[21]	0 ('0')
[22]	0 ('0')
[23]	0 ('0')
[24]	0 ('0')
[25]	0 ('0')
[26]	0 ('0')
[27]	0 ('0')
[28]	0 ('0')
[29]	0 ('0')
[30]	0 ('0')
[31]	0 ('0')
[32]	0 ('0')
[33]	0 ('0')
[34]	0 ('0')
[35]	0 ('0')
[36]	0 ('0')
[37]	0 ('0')
[38]	0 ('0')
[39]	0 ('0')
[40]	0 ('0')
[41]	0 ('0')
[42]	0 ('0')
[43]	0 ('0')
[44]	0 ('0')
[45]	0 ('0')
[46]	0 ('0')
[47]	0 ('0')

Ozone - The J-Link Debugger V3.22b - C:/Users/BoQiang/work_hbq/FAE_develop/FlexCAN_EnhancedFIFO_Config/build/FlexCAN_EnhancedFIFO_Config.elf

File View Find Debug Tools Window Help

Break & Tracepoints x main.c x

Type Location Extras

Code main.c:162 FLEXCAN_DRV_EnhanceRxFifo()

File Scope

104 },
105 },
106 {
107 .isRemoteFrame = **false**,
108 .isExtendedFrame = **true**,
109 .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,
110 .filter_table.scheme_filter_mask =
111 {
112 .filter = 0x10FF1063U,
113 .mask = 0xFFFFFFFFU,
114 },
115 },
116 };
117 /* USER CODE END PV */
118 /* Private function declare */
119 /* USER CODE BEGIN FDPC */
120 /* USER CODE END FDPC */
121 /* USER CODE BEGIN 0 */
122 /* USER CODE BEGIN 0 */
123 static void Board_Init(void);
124 /* Private user code */
125 /* USER CODE BEGIN 0 */
126 /* USER CODE END 0 */
127 /* USER CODE END 0 */
128
129
130 /**
131 * @brief The application entry point.
132 * @retval int
133 */
134 int main(void)
135 {
136 /* USER CODE BEGIN 1 */
137 /* USER CODE END 1 */
138 Board_Init();
139 /* USER CODE BEGIN 2 */
140 PRINTF("Build %s %s\n", __DATE__, __TIME__);
141 PRINTF("FlexCAN enhanced fifo demo!\r\n");
142
143 FLEXCAN_DRV_ConfigEnhanceRxFifo(0, can_enhancefifo_filter, 5);
144 FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
145
146 for (uint8_t i = 0; i < 64; i++)
147 {
148 TxMsgBuff.data[i] = i;
149 }
150 /* USER CODE END 2 */
151
152 /* Infinite loop */
153 /* USER CODE BEGIN WHILE */
154 while (1)
155 {
156 if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, 0))
157 {
158 FLEXCAN_DRV_Send(0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
159 }
160 if (STATUS_BUSY != FLEXCAN_DRV_GetTransferStatus(0, (uint8_t)FLEXCAN_MB_HANDLE_ENHANCE_RXFIFO))
161 {
162 FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
163 }
164 OSIF_TimeDelay(100);
165 /* USER CODE END WHILE */
166 /* USER CODE BEGIN 3 */
167 }
168 /* USER CODE END 3 */
169 }
170
171 static void Board_Init(void)
172 {
173 CLOCK_SYS_Init(g_clockManConfigsArr, CLOCK_MANAGER_CONFIG_CNT, g_clockManCallbacksArr, CLOCK_MANAGER_CALLBACK_CNT);
174 CLOCK_SYS_UpdateConfiguration(CLOCK_MANAGER_ACTIVE_INDEX, CLOCK_MANAGER_POLICY AGREEMENT);
175 PINS_DRV_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0);
176 FLEXCAN_DRV_Init(0, &flexcanInitConfig0_State, &flexcanInitConfig0);
177 UTILITY_PRINT_Init();
178 }

Vector Catch

• Reset Vect

• MemManage Vect

• UsageFault_Coprocessor Vect

• UsageFault_CheckingError Vect

• UsageFault_StateError Vect

• BusFault Vect

Source Files

File Status

_default_types.h included

_stdint.h included

callbacks.h included

can_config.c external

can_config.h included

clock_config.c external

clock_config.h included

Watched Data 1

Expression Value

EnhanceRxBuff

cg 6866493

mid 10FF1063

data *105h\`221**

[0] 33 ('1')
[1] 103 ('M')
[2] 53 ('5')
[3] 104 ('M')
[4] 148 ('145')
[5] 34 ('**')
[6] 0 ('0')
[7] 49 ('1')
[8] 0 ('0')
[9] 0 ('0')
[10] 0 ('0')
[11] 0 ('0')
[12] 0 ('0')
[13] 0 ('0')
[14] 0 ('0')
[15] 0 ('0')
[16] 0 ('0')
[17] 0 ('0')
[18] 0 ('0')
[19] 0 ('0')
[20] 0 ('0')
[21] 0 ('0')
[22] 0 ('0')
[23] 0 ('0')
[24] 0 ('0')
[25] 0 ('0')
[26] 0 ('0')
[27] 0 ('0')
[28] 0 ('0')
[29] 0 ('0')
[30] 0 ('0')
[31] 0 ('0')
[32] 0 ('0')
[33] 0 ('0')
[34] 0 ('0')
[35] 0 ('0')
[36] 0 ('0')
[37] 0 ('0')
[38] 0 ('0')
[39] 0 ('0')
[40] 0 ('0')
[41] 0 ('0')
[42] 0 ('0')
[43] 0 ('0')
[44] 0 ('0')
[45] 0 ('0')
[46] 0 ('0')
[47] 0 ('0')

5.4.2 中断方式

a. 上位机收到ID为0x10FF1369的报文，数据长度为64，周期约为400us。

b. demo板能成功接收到滤波表中设置的所有报文。

13	▶	手动	NewMsg	310	1	标准数据帧	8	01	02	03	04	05	06	07	08	
14	▶	手动	NewMsg	420	1	标准数据帧	8	01	02	04	03	05	06	08	07	
15	▶	手动	NewMsg	530	1	标准数据帧	8	04	03	02	01	08	07	06	05	
16	▶	手动	NewMsg	0D314450	1	扩展数据帧	8	08	07	06	05	04	03	02	01	
17	▶	手动	NewMsg	10FF1063	1	扩展数据帧	8	07	08	05	06	03	04	01	02	

Ozone - The J-Link Debugger V3.22b - C:/Users/BoQiang/work_hbq/FAE_develop/FlexCAN_EnhancedFIFO_Config/build/FlexCAN_EnhancedFIFO_Config.elf

File View Find Debug Tools Window Help

Break & Tracepoints x main.c x

Type Location Extras

+ Code main.c:243 FLEXCAN_DRV_Enhanc

Vector Catch Description

Reset Vector

MemManage Vector

Source Files x

File Status :

_default_types.h included

_stdint.h included

callbacks.h included

can_config.c external

can_config.h included

clock_config.c external

clock_config.h included

clock_YTM32B1Mx.c compiled 2

clock_ytm32b1mx.h included

cmsis_gcc.h included

core_cm33.h compiled

dma_config.h included

Watched Data 1 x

Expression Value

EnhanceRxBuff

cs 544 957

msgId 0000 0310

data "001"\002"\003"\004"\005"\006"\007"\008"

[0] 1 ('001')

[1] 2 ('002')

[2] 3 ('003')

[3] 4 ('004')

[4] 5 ('005')

[5] 6 ('006')

[6] 7 ('007')

[7] 8 ('008')

[8] 0 ('0')

[9] 0 ('0')

[10] 0 ('0')

[11] 0 ('0')

[12] 0 ('0')

[13] 0 ('0')

[14] 0 ('0')

[15] 0 ('0')

[16] 0 ('0')

File Scope f can_callback

```

219     .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,
220     .filter_table.scheme_filter_mask =
221     {
222         .filter = 0x10FF1063U,
223         .mask = 0x1FFFFFFU,
224     },
225 };
226 #endif
227 /* USER CODE END PV */
228 /* Private function declare -----
229 /* USER CODE BEGIN PFDC */
230 void can_callback(uint8_t instance, flexcan_event_types_t eventType, \
231                     uint32_t buffIdx, flexcan_state_t *flexcanState)
232 {
233     (void)instance;
234     (void)flexcanState;
235     switch (eventType)
236     {
237         case FLEXCAN_EVENT_TX_COMPLETE:
238             FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
239             break;
240         case FLEXCAN_EVENT_ENHANCE_RXFIFO_AVAILABLEDATA:
241             FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
242             break;
243         default:
244             break;
245     }
246 }
247 /* USER CODE END PFDC */
248 static void Board_Init(void);
249 /* Private user code -----
250 /* USER CODE BEGIN 0 */
251 /* USER CODE END 0 */
252 /**
253 * @brief The application entry point.
254 */
255 int main(void)
256 {
257     /* @retval int
258 */
259     int main()
260     {
261         /* USER CODE BEGIN 1 */
262         /* USER CODE END 1 */
263         Board_Init();
264         /* USER CODE BEGIN 2 */
265         PRINTF("Build %s\n", __DATE__, __TIME__);
266         PRINTF("FlexCAN enhanced fifo demo!\r\n");
267         /* USER CODE END 2 */
268         #if USE_INTERRUPT
269         FLEXCAN_DRV_InstallEventCallback(0, can_callback, NULL);
270         #endif
271     }
272 }

```

Ozone - The J-Link Debugger V3.22b - C:/Users/BoQiang/work_hbq/FAE_develop/FlexCAN_EnhancedFIFO_Config/build/FlexCAN_EnhancedFIFO_Config.elf

File View Find Debug Tools Window Help

Break & Tracepoints Code main.c:243 FLEXCAN_DRV_Enhanc

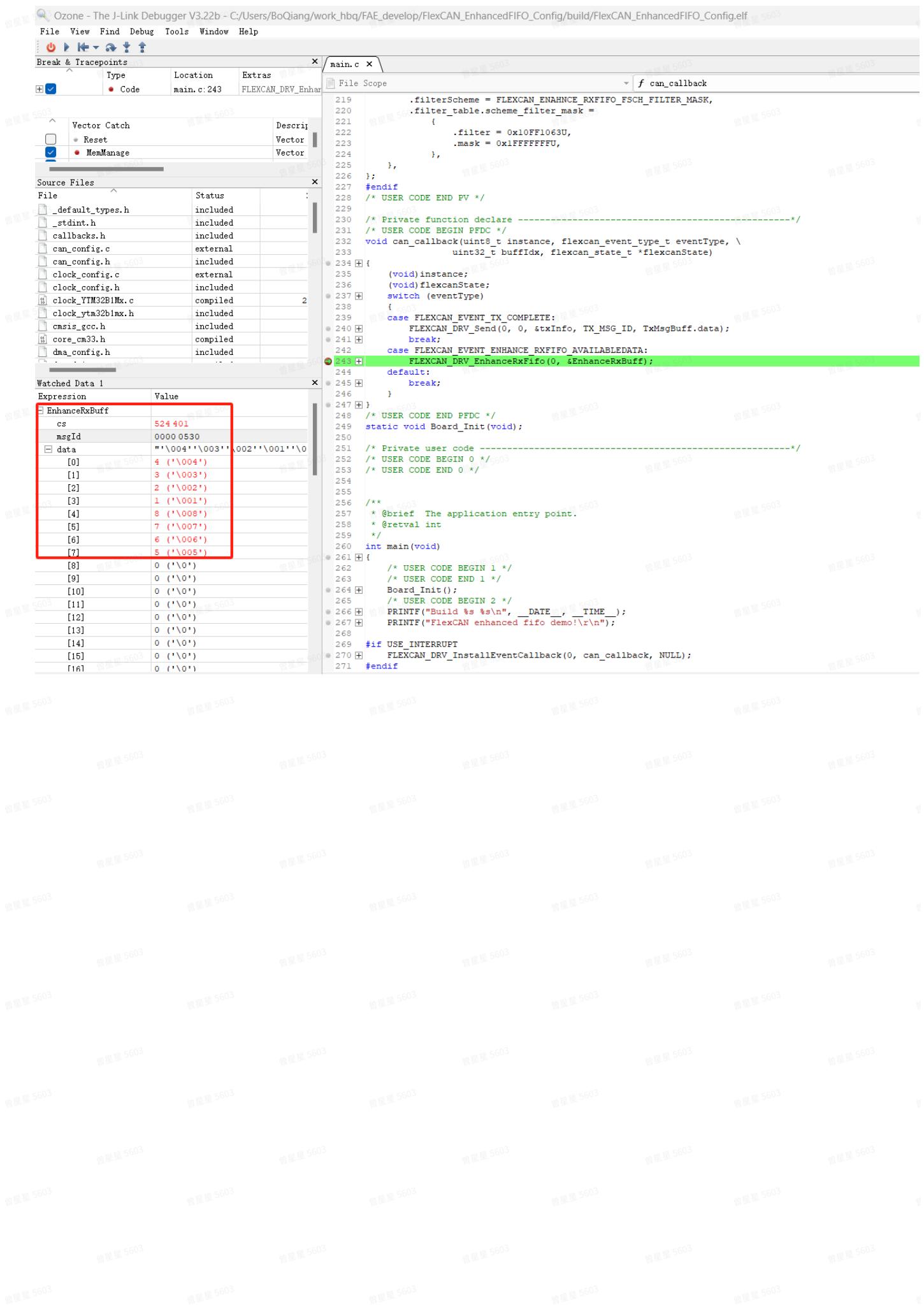
Source Files File Status : _default_types.h included _stdint.h included callbacks.h included can_config.c external can_config.h included clock_config.c external clock_config.h included clock_YTM32B1Mx.c compiled 2 clock_ytm32b1mx.h included cmsis_gcc.h included core_cm33.h compiled dma_config.h included

Watched Data 1 Expression Value

EnhanceFxBuff	564 630
cs	564 630
msgId	0000 0420
data	"'\001'\002'\004'\003'\005'\006'\007'
[0]	1 (''\001'')
[1]	2 (''\002'')
[2]	4 (''\004'')
[3]	3 (''\003'')
[4]	5 (''\005'')
[5]	6 (''\006'')
[6]	8 (''\008'')
[7]	7 (''\007'')
[8]	0 (''\01'')
[9]	0 (''\01'')
[10]	0 (''\01'')
[11]	0 (''\01'')
[12]	0 (''\01'')
[13]	0 (''\01'')
[14]	0 (''\01'')
[15]	0 (''\01'')
[16]	0 (''\01'')

File Scope f can_callback

```
219     .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,
220     .filter_table.scheme_filter_mask =
221     {
222         .filter = 0x10FF1063U,
223         .mask = 0x1FFFFFFFU,
224     },
225 };
226 #endif
227 /* USER CODE END PV */
228 /* Private function declare */
229 /* USER CODE BEGIN PFDC */
230 void can_callback(uint8_t instance, flexcan_event_type_t eventType, \
231                     uint32_t buffIdx, flexcan_state_t*flexcanState)
232 {
233     (void)instance;
234     (void)flexcanState;
235     switch (eventType)
236     {
237         case FLEXCAN_EVENT_TX_COMPLETE:
238             FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
239             break;
240         case FLEXCAN_EVENT_ENHANCE_RXFIFO_AVAILABLEDATA:
241             FLEXCAN_DRV_EnhanceRxBuff(0, &EnhanceRxBuff);
242             break;
243         default:
244             break;
245     }
246 }
247 /* USER CODE END PFDC */
248 static void Board_Init(void);
249 /* Private user code */
250 /* USER CODE BEGIN 0 */
251 /* USER CODE END 0 */
252 /**
253 * @brief The application entry point.
254 * @retval int
255 */
256 /**
257 * @brief The application entry point.
258 * @retval int
259 */
260 int main(void)
261 {
262     /* USER CODE BEGIN 1 */
263     /* USER CODE END 1 */
264     Board_Init();
265     /* USER CODE BEGIN 2 */
266     PRINTF("Build %s %s\n", __DATE__, __TIME__);
267     PRINTF("FlexCAN enhanced fifo demo!\r\n");
268     #if USE_INTERRUPT
269     FLEXCAN_DRV_InstallEventCallback(0, can_callback, NULL);
270     #endif
271 }
```



Ozone - The J-Link Debugger V3.22b - C:/Users/BoQiang/work_hbq/FAE_develop/FlexCAN_EnhancedFIFO_Config/build/FlexCAN_EnhancedFIFO_Config.elf

File View Find Debug Tools Window Help

Break & Tracepoints

Type	Location	Extras
Code	main.c:243	FLEXCAN_DRV_Enhanc...
Vector Catch		Vector
Reset		Vector
MemManage		Vector

Source Files

File	Status
_default_types.h	included
_stdint.h	included
callbacks.h	included
can_config.c	external
can_config.h	included
clock_config.c	external
clock_config.h	included
clock_YTM32B1Mx.c	compiled 2
clock_ytm32b1mx.h	included
cmsis_gcc.h	included
core_cm33.h	compiled
dma_config.h	included

Watched Data 1

Expression	Value
EnhanceRxBuff	<pre>cs 6 874 826 msgId 0D31 4450 data "\010'\007'\005'\005'\00 [0] 8 ('008') [1] 7 ('007') [2] 6 ('006') [3] 5 ('005') [4] 4 ('004') [5] 3 ('003') [6] 2 ('002') [7] 1 ('001') [8] 0 ('0') [9] 0 ('0') [10] 0 ('0') [11] 0 ('0') [12] 0 ('0') [13] 0 ('0') [14] 0 ('0') [15] 0 ('0') [16] 0 ('0')</pre>

File Scope

```

219     .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_F SCH_FILTER_MASK,
220     .filter_table.scheme_filter_mask =
221     {
222         .filter = 0x10FF1063U,
223         .mask = 0xFFFFFFFFU,
224     },
225 },
226 };
227 #endif
228 /* USER CODE END PV */
229
230 /* Private function declare -----
231 /* USER CODE BEGIN PFDC */
232 void can_callback(uint8_t instance, flexcan_event_type_t eventType, \
233                     uint32_t buffIdx, flexcan_state_t *flexcanState)
234 {
235     (void)instance;
236     (void)flexcanState;
237     switch (eventType)
238     {
239     case FLEXCAN_EVENT_TX_COMPLETE:
240         FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
241         break;
242     case FLEXCAN_EVENT_ENHANCE_RXFIFO_AVAILABLEDATA:
243         FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
244     default:
245         break;
246     }
247 }
248 /* USER CODE END PFDC */
249 static void Board_Init(void);
250
251 /* Private user code -----
252 /* USER CODE BEGIN 0 */
253 /* USER CODE END 0 */
254
255
256 /**
257  * @brief The application entry point.
258  * @retval int
259 */
260 int main(void)
261 {
262     /* USER CODE BEGIN 1 */
263     /* USER CODE END 1 */
264     Board_Init();
265     /* USER CODE BEGIN 2 */
266     PRINTF("Build %s %s\n", __DATE__, __TIME__);
267     PRINTF("FlexCAN enhanced fifo demo!\r\n");
268
269     #if USE_INTERRUPT
270     FLEXCAN_DRV_InstallEventCallback(0, can_callback, NULL);
271     #endif

```

The screenshot shows the Ozone debugger interface with the main.c file open. The code is as follows:

```

219     .filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,
220     .filter_table.scheme_filter_mask =
221     {
222         .filter = 0x10FF1063U,
223         .mask = 0x1FFFFFFFU,
224     },
225 },
226 #endif
227 /* USER CODE END PV */
228
229 /* Private function declare -----
230 /* USER CODE BEGIN PFDC */
231 void can_callback(uint8_t instance, flexcan_event_type_t eventType,
232                   uint32_t buffIdx, flexcan_state_t *flexcanState)
233 {
234     (void)instance;
235     (void)flexcanState;
236     switch (eventType)
237     {
238         case FLEXCAN_EVENT_TX_COMPLETE:
239             FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID, TxMsgBuff.data);
240             break;
241         case FLEXCAN_EVENT_ENHANCE_RXFIFO_AVAILABLEDATA:
242             FLEXCAN_DRV_EnhanceRxFifo(0, &EnhanceRxBuff);
243             break;
244         default:
245             break;
246     }
247 } /* USER CODE END PFDC */
248 static void Board_Init(void);
249
250 /* Private user code -----
251 /* USER CODE BEGIN 0 */
252 /* USER CODE BEGIN 0 */
253 /* USER CODE END 0 */
254
255 /**
256 * @brief The application entry point.
257 * @retval int
258 */
259
260 int main(void)
261 {
262     /* USER CODE BEGIN 1 */
263     /* USER CODE END 1 */
264     Board_Init();
265     /* USER CODE BEGIN 2 */
266     PRINTF("Build %s %s\n", __DATE__, __TIME__);
267     PRINTF("FlexCAN enhanced fifo demo!\r\n");
268
269     #if USE_INTERRUPT
270         FLEXCAN_DRV_InstallEventCallback(0, can_callback, NULL);
271     #endif
272 }

```

The 'EnhanceRxBuff' variable is highlighted in the 'Watched Data 1' window. Its structure is:

Expression	Value
EnhanceRxBuff	
cs	6 019 354
msgId	10FF 1063
data	"\007"\010"\005"\006"\00
[0]	7 ('\\007')
[1]	8 ('\\008')
[2]	5 ('\\005')
[3]	6 ('\\006')
[4]	3 ('\\003')
[5]	4 ('\\004')
[6]	1 ('\\001')
[7]	2 ('\\002')
[8]	0 ('\\0')
[9]	0 ('\\0')
[10]	0 ('\\0')
[11]	0 ('\\0')
[12]	0 ('\\0')
[13]	0 ('\\0')
[14]	0 ('\\0')
[15]	0 ('\\0')
[16]	0 ('\\0')

6. 主要API函数介绍

1. 配置enhanced FIFO滤波表：

```
1 void FLEXCAN_DRV_ConfigEnhanceRxFifo(uint8_t instance, const
                                         flexcan_enhance_rx_fifo_filter_table_t *id_filter_table, uint8_t
                                         id_filter_table_length)
```

id_filter_table: 滤波器配置数组地址；

id_filter_table_length: 滤波表数组长度。

2. enhanced FIFO数据接收函数：

```
1 status_t FLEXCAN_DRV_EnhanceRxFifo(uint8_t instance, flexcan_msghbuff_t *data)
```

data: 装填接收数据的软件缓冲区。

该函数用于开启enhanced FIFO的数据接收，与邮箱接收数据一样，调用一次只能接收一次，需要不断调用该函数实现连续通信。

其他API介绍请参考"SDK应用_FlexCAN模块配置及应用（一）"。

7. 文档历史

版本号	日期	修订记录
V1.0	2024.01.16	初始版本