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：

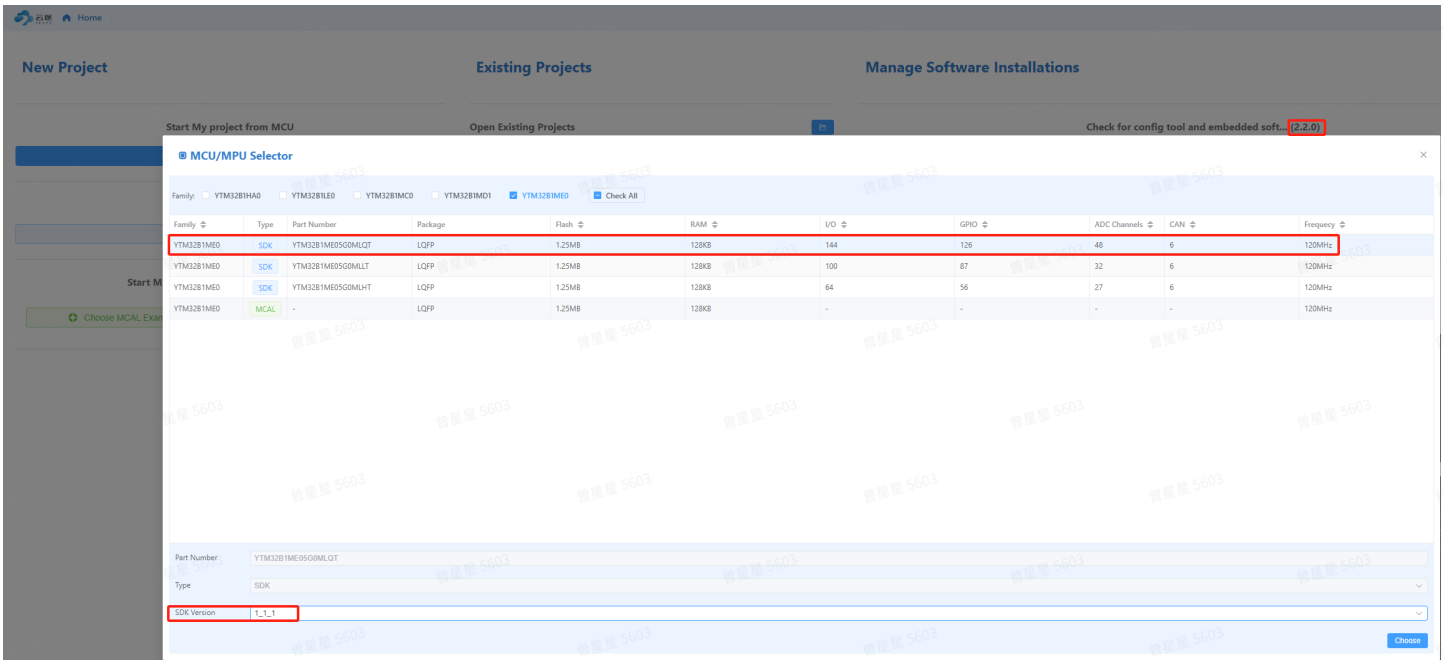
Feature	Classical CAN	CAN FD
Legacy RX FIFO	YES	NO
Legacy RX FIFO DMA	YES	NO
Enhanced RX FIFO	YES	YES
Enhanced RX FIFO DMA	YES	YES
Pretended Networking	YES	NO

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。

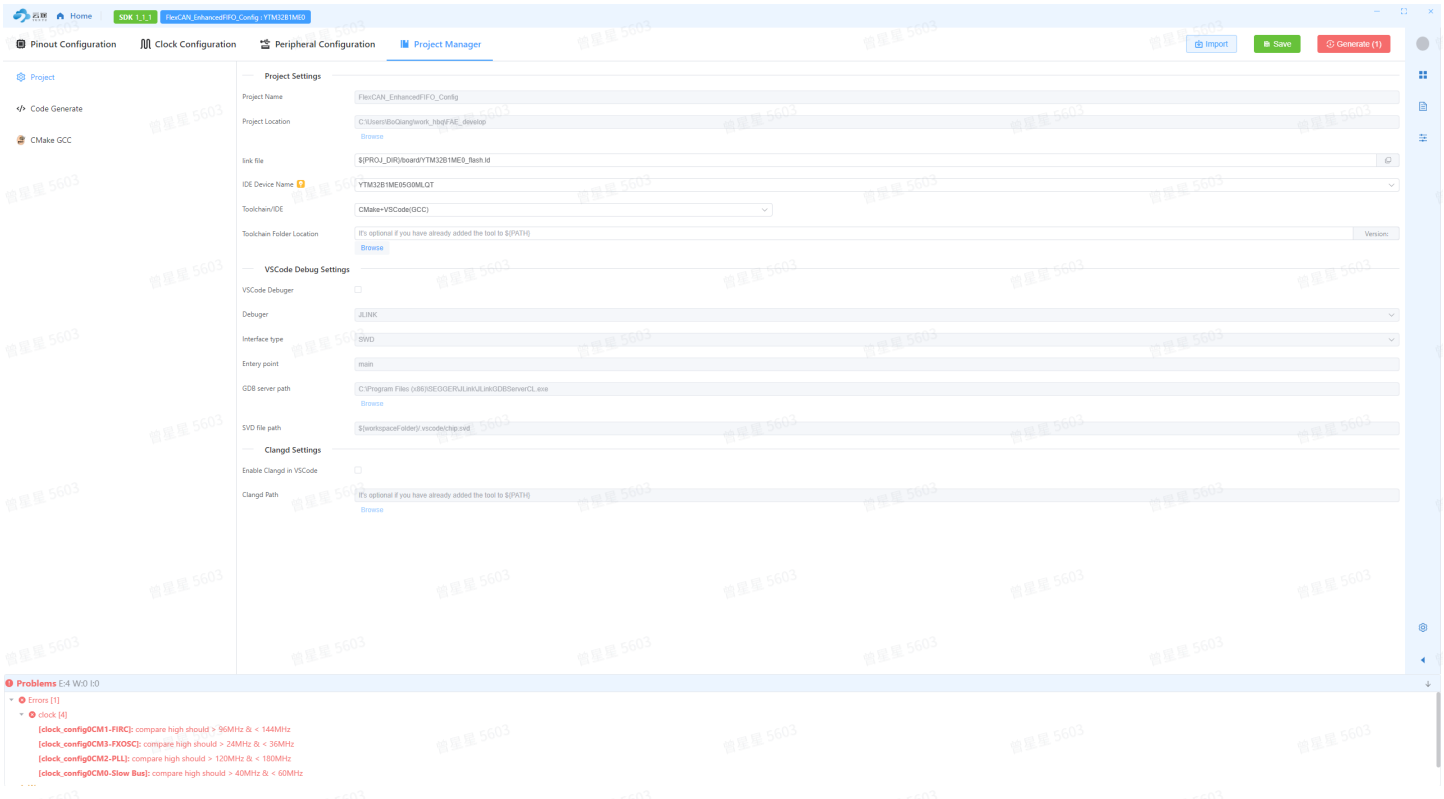
2. 芯片型号及工具版本信息

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

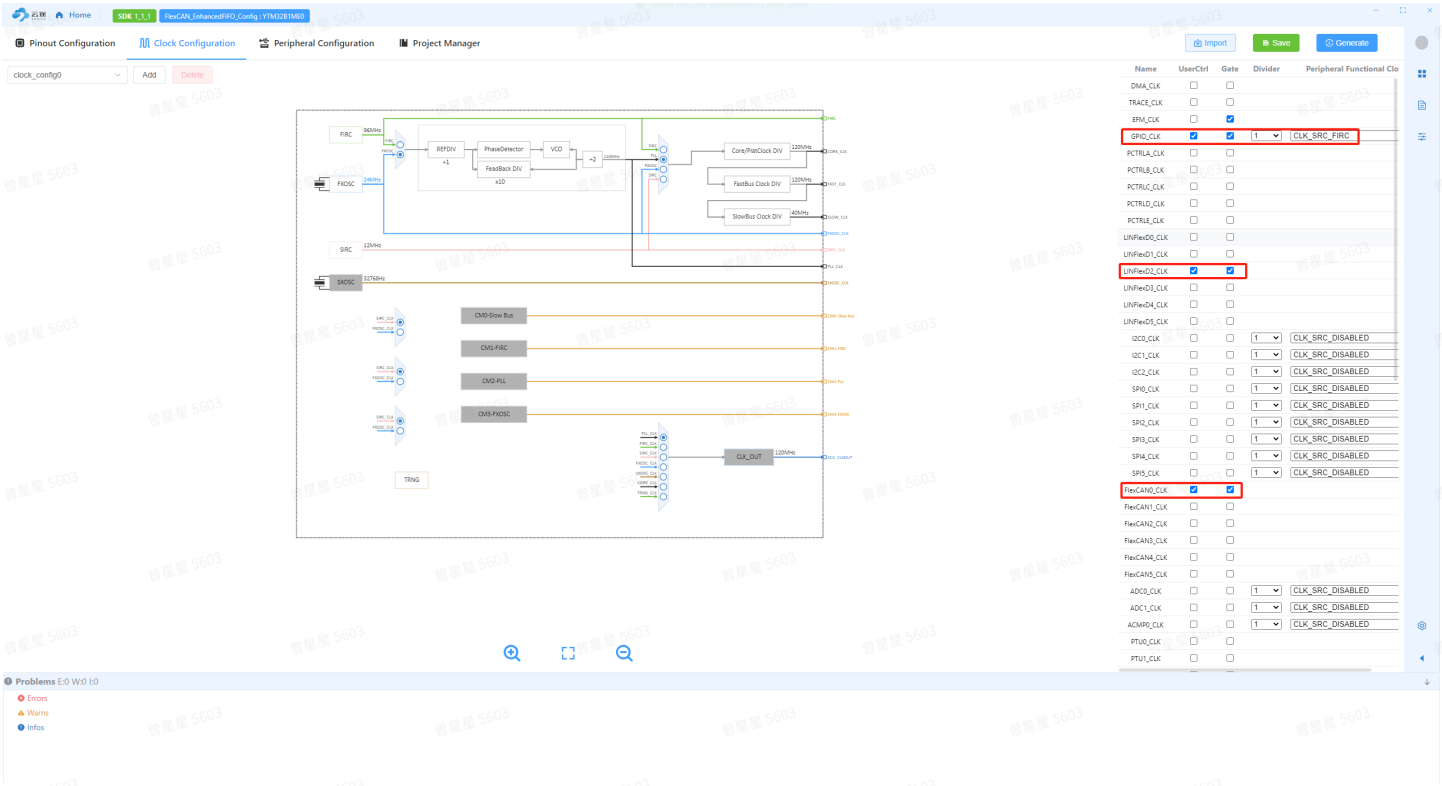


3. 配置步骤

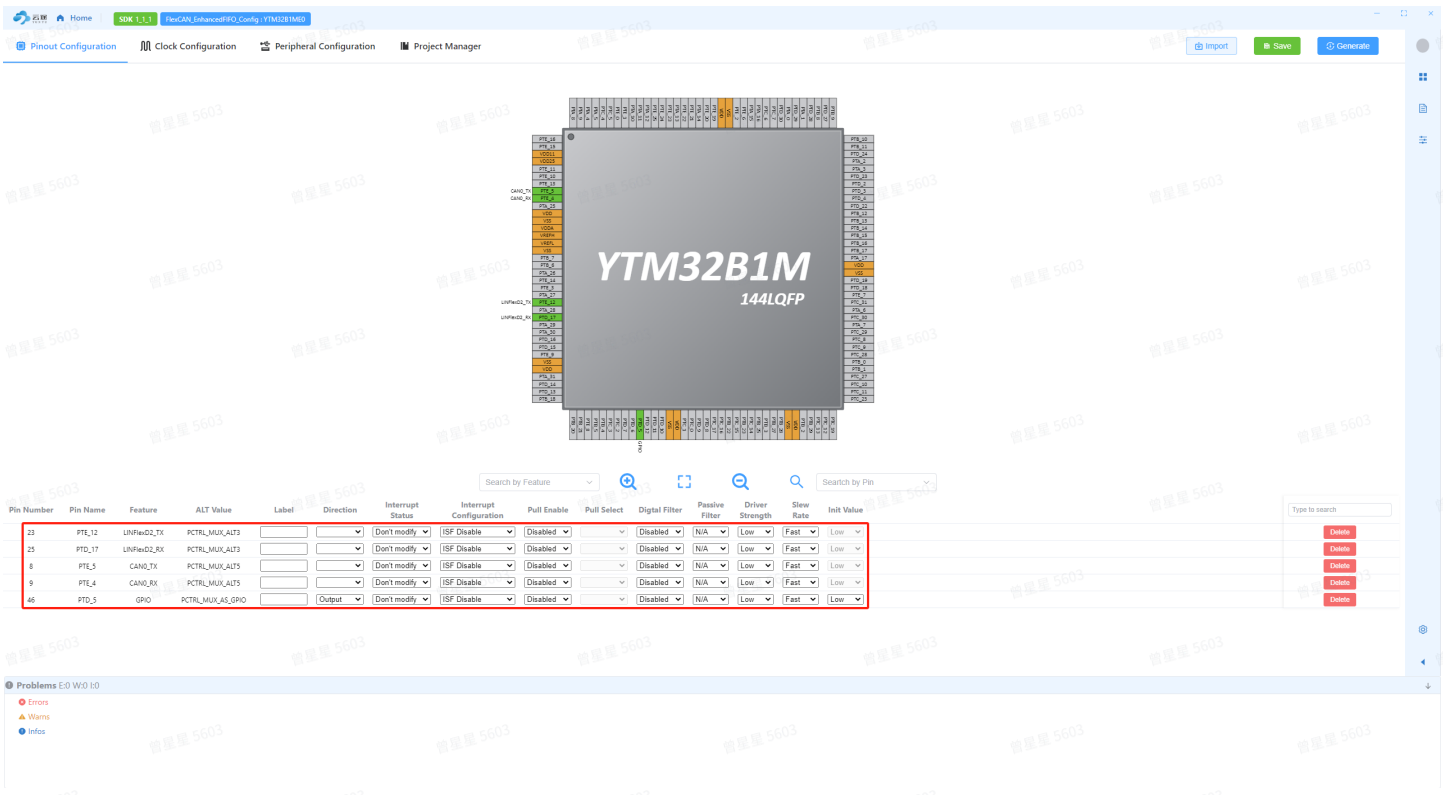
3.1 建立工程



3.2 时钟配置

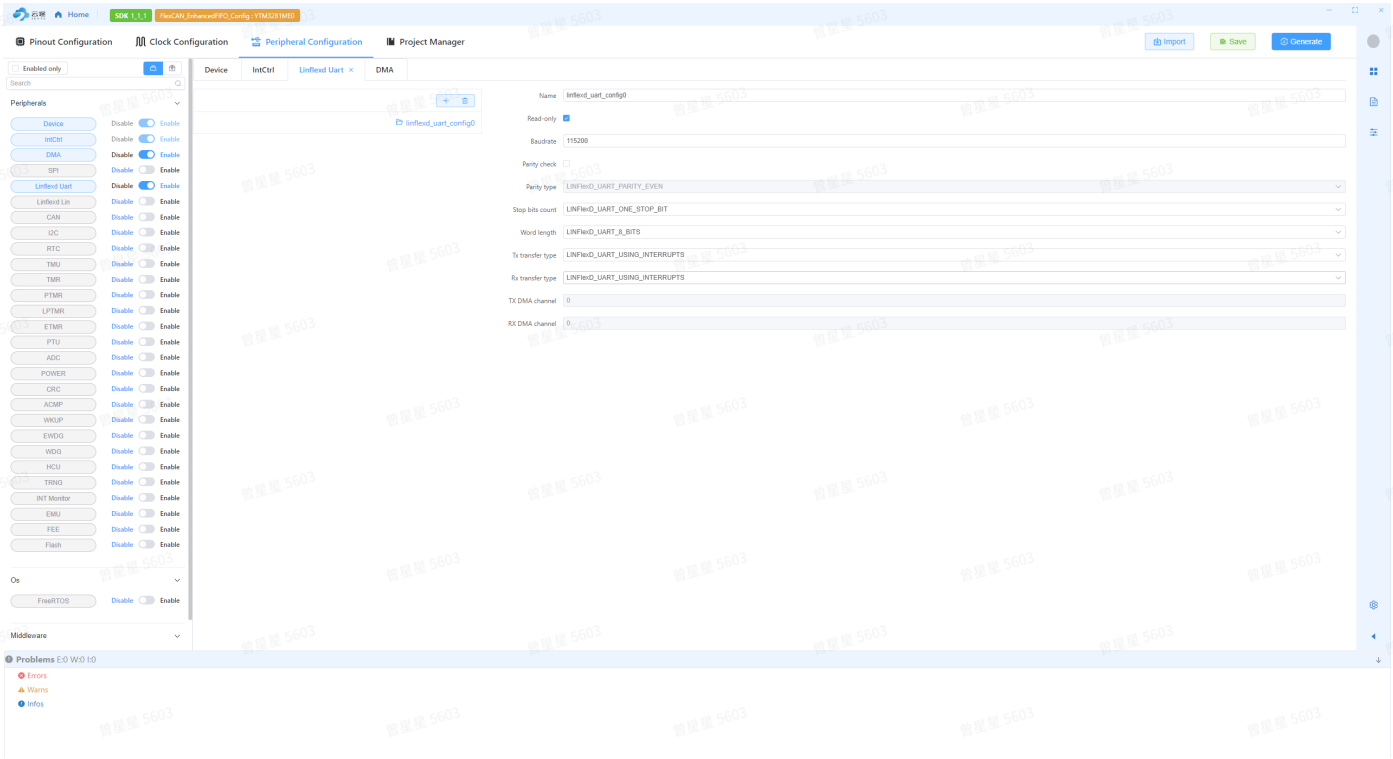


3.3 引脚配置

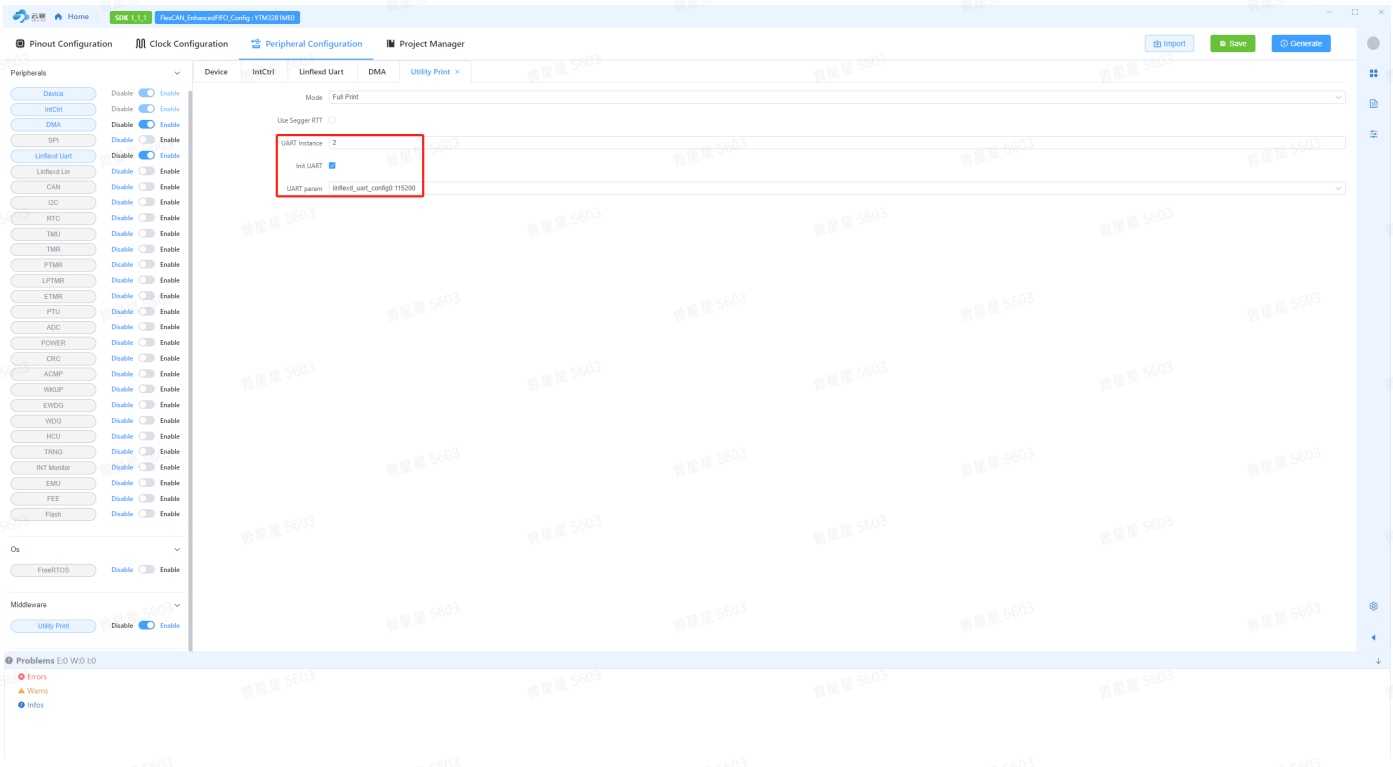


3.4 模块功能配置

3.4.1 UART功能配置

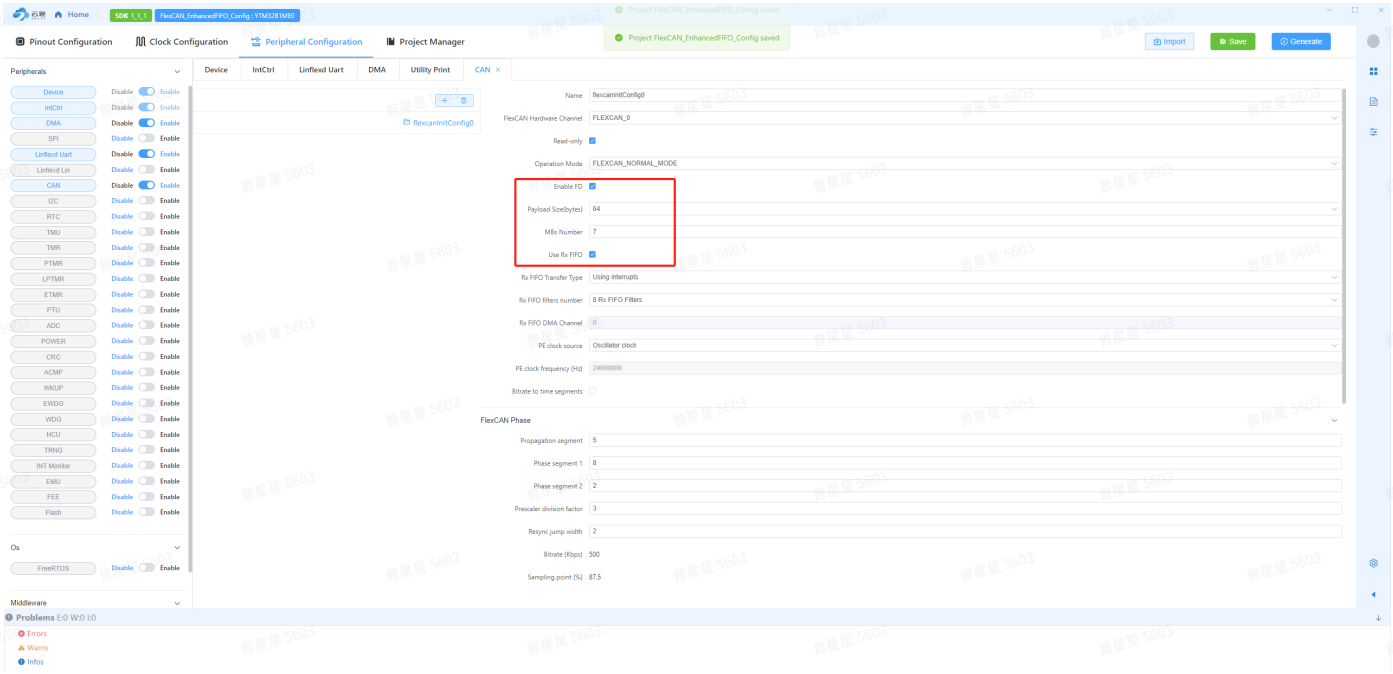


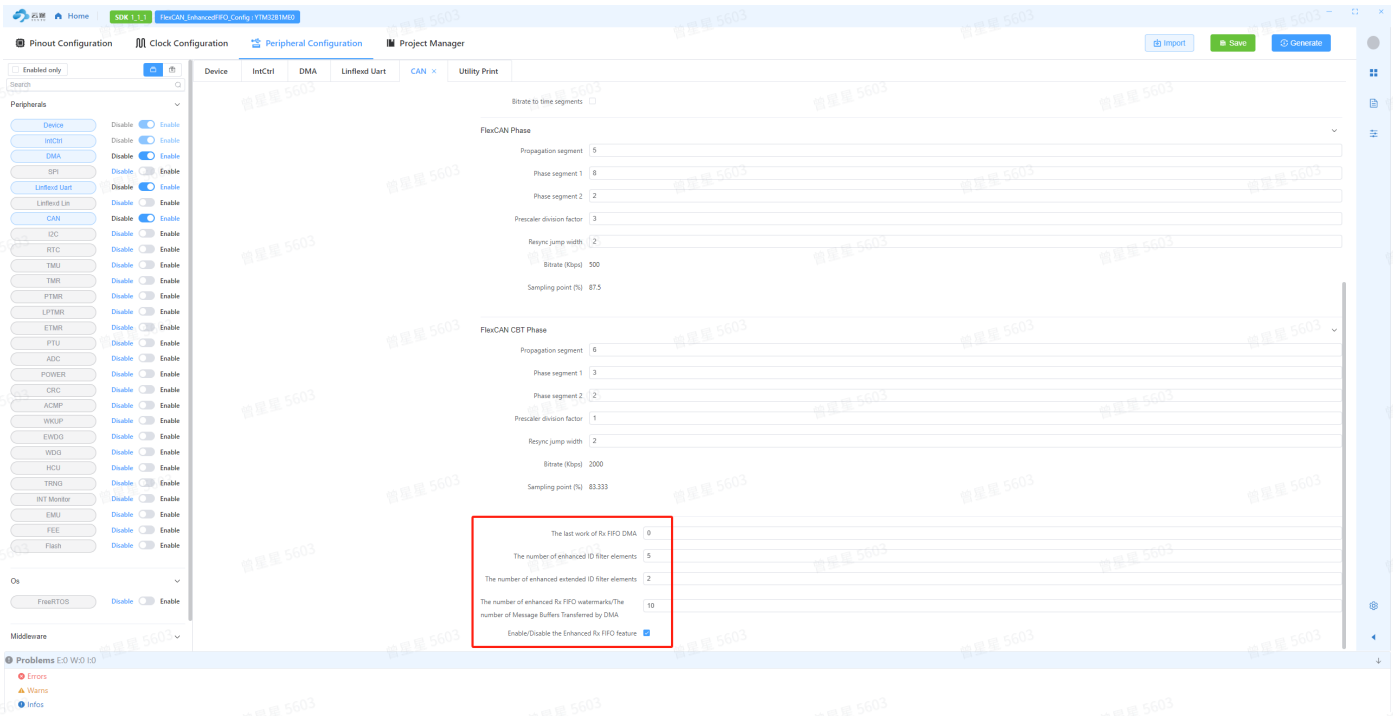
3.4.2 添加打印函数



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。





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

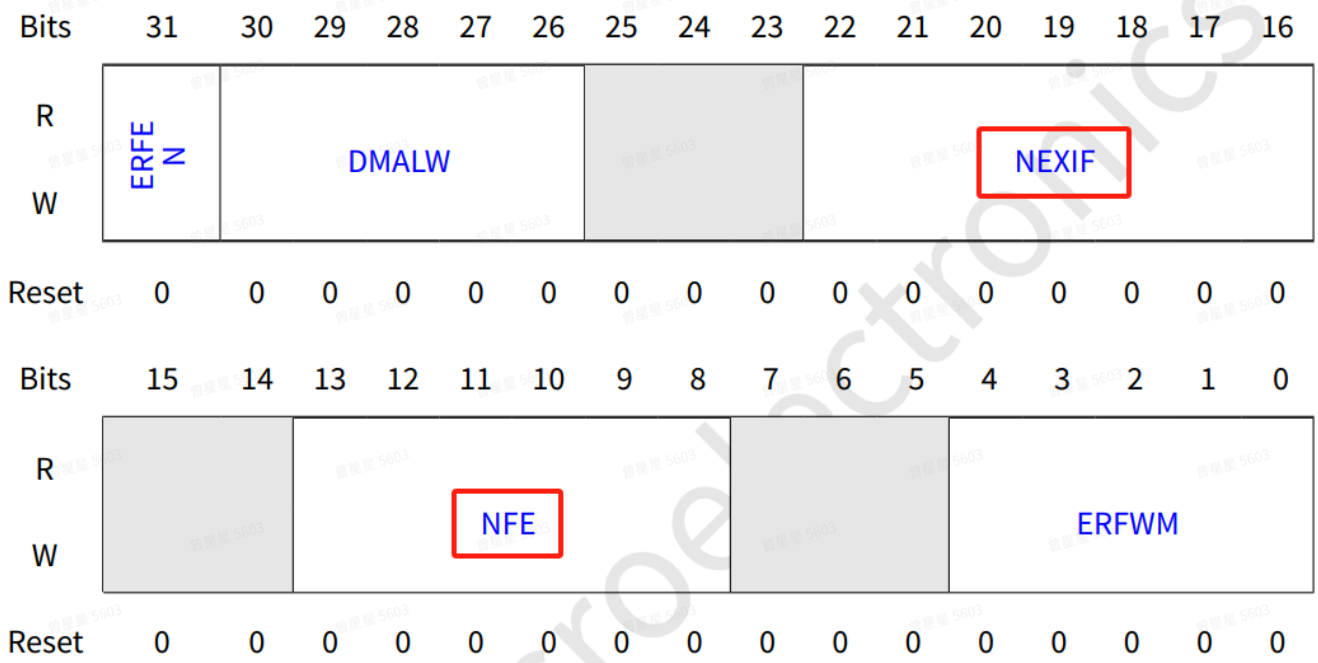


Table 18.69: CAN ERFCR Register Description

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）”。

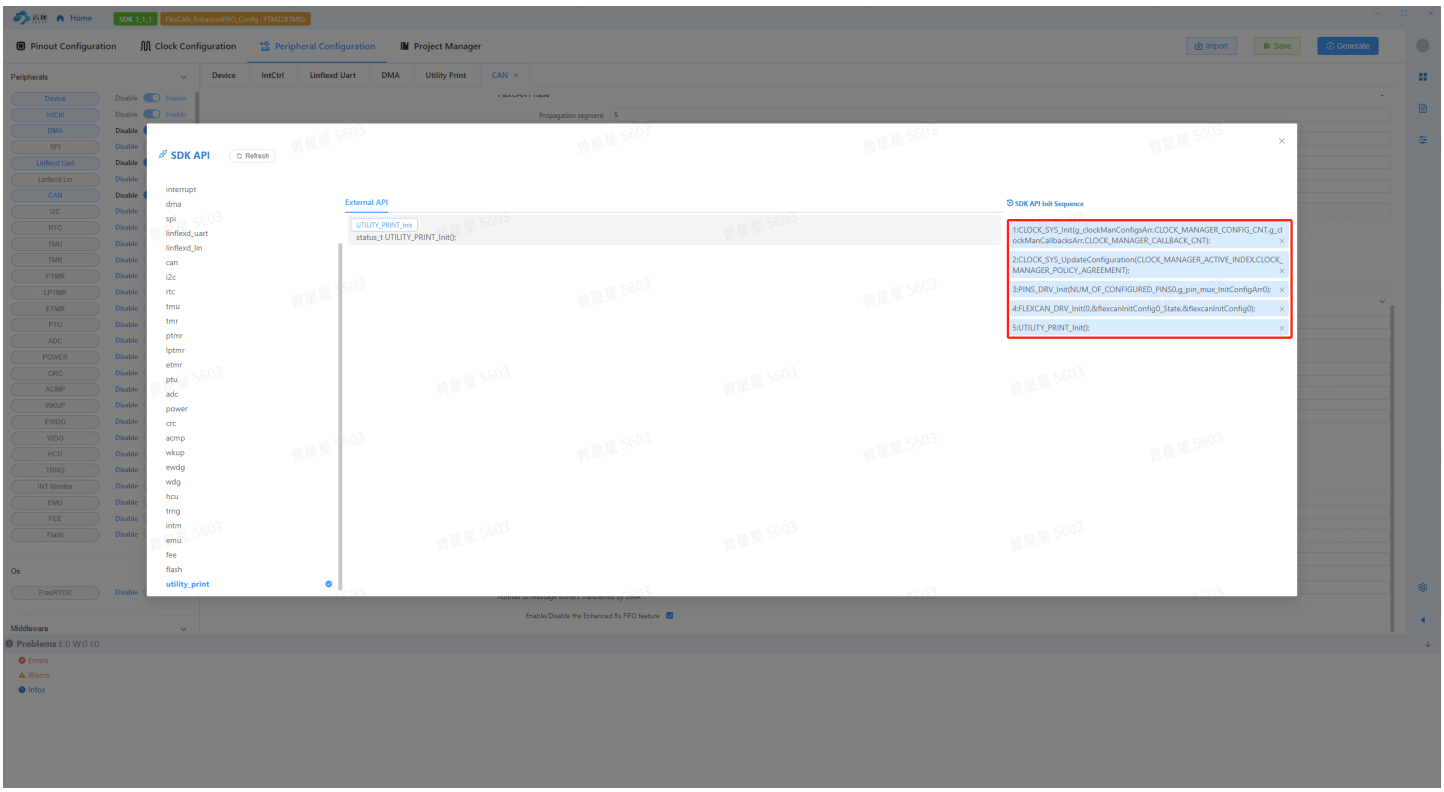
💡 上面配置中的The number of enhanced Rx FIFO watermarks/The number of Message Buffers Transferred by DMA项必须大于0，否则初始化会报错，进而导致通信失败。

💡 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）。

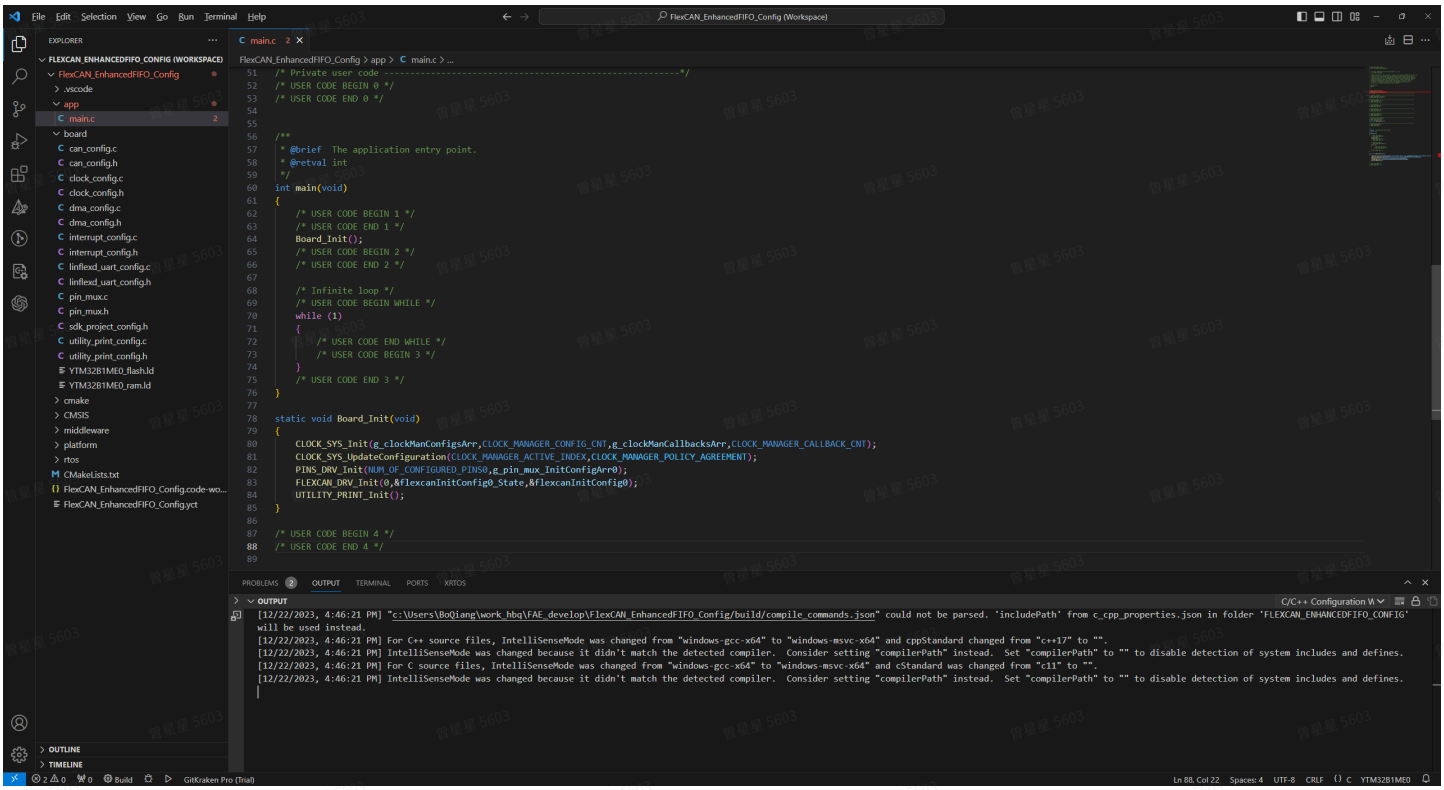
下表展示了NFE的值与可设置的扩展ID或标准ID的对应关系。

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

3.5 添加API



4. 生成工程展示



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：

a. 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：

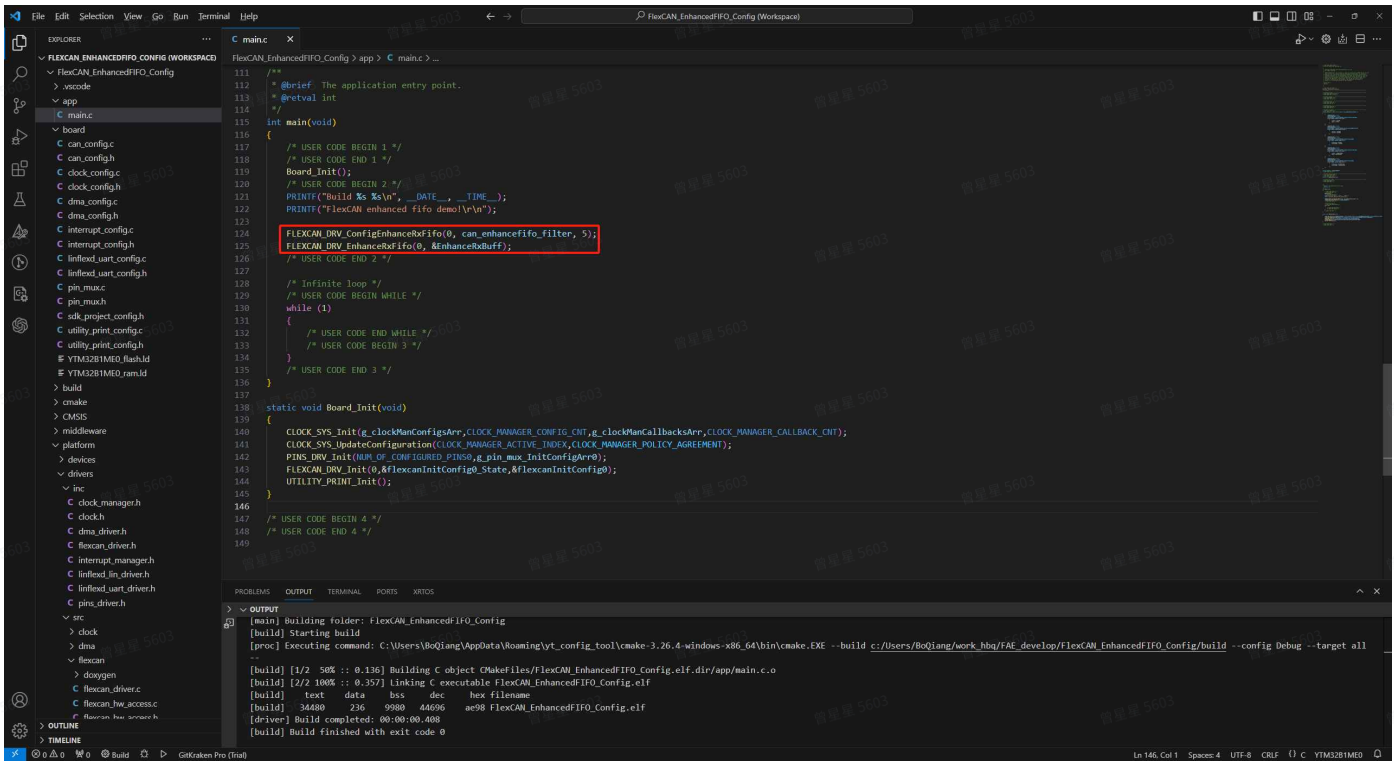
- isRemoteFrame，该帧是否为远程帧（bool型变量）
- isExtendeFrame，该帧是否为扩展帧（bool型变量）
- filterScheme，滤波器格式选择
- filter_table，这是一个联合体，可根据上面配置的滤波器格式，设置相应过滤ID和掩码

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控制器只能够接收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）”。



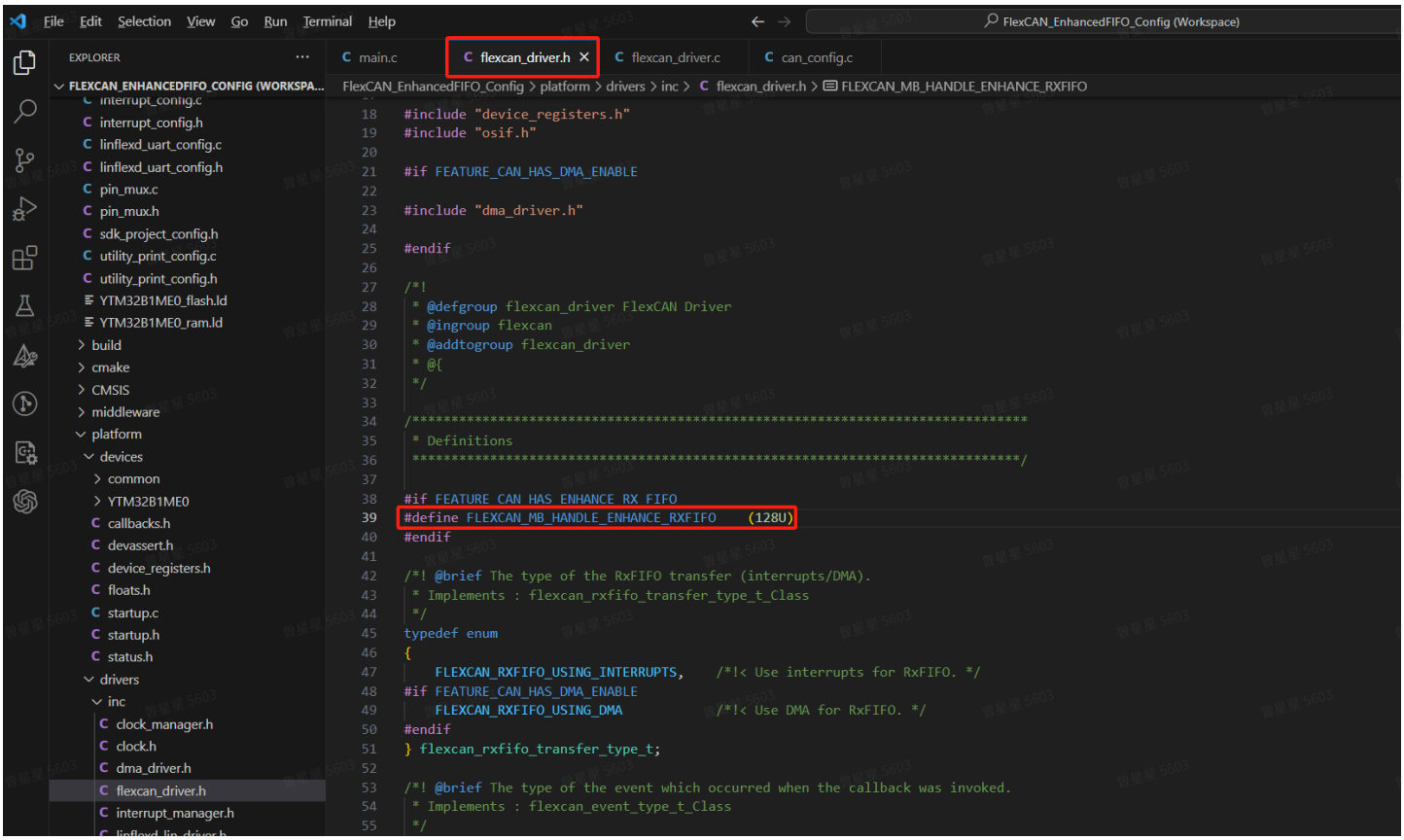
5.2 配置数据的轮询发送和接收，周期100ms

```c
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：



5.3 配置数据的中断发送和接收

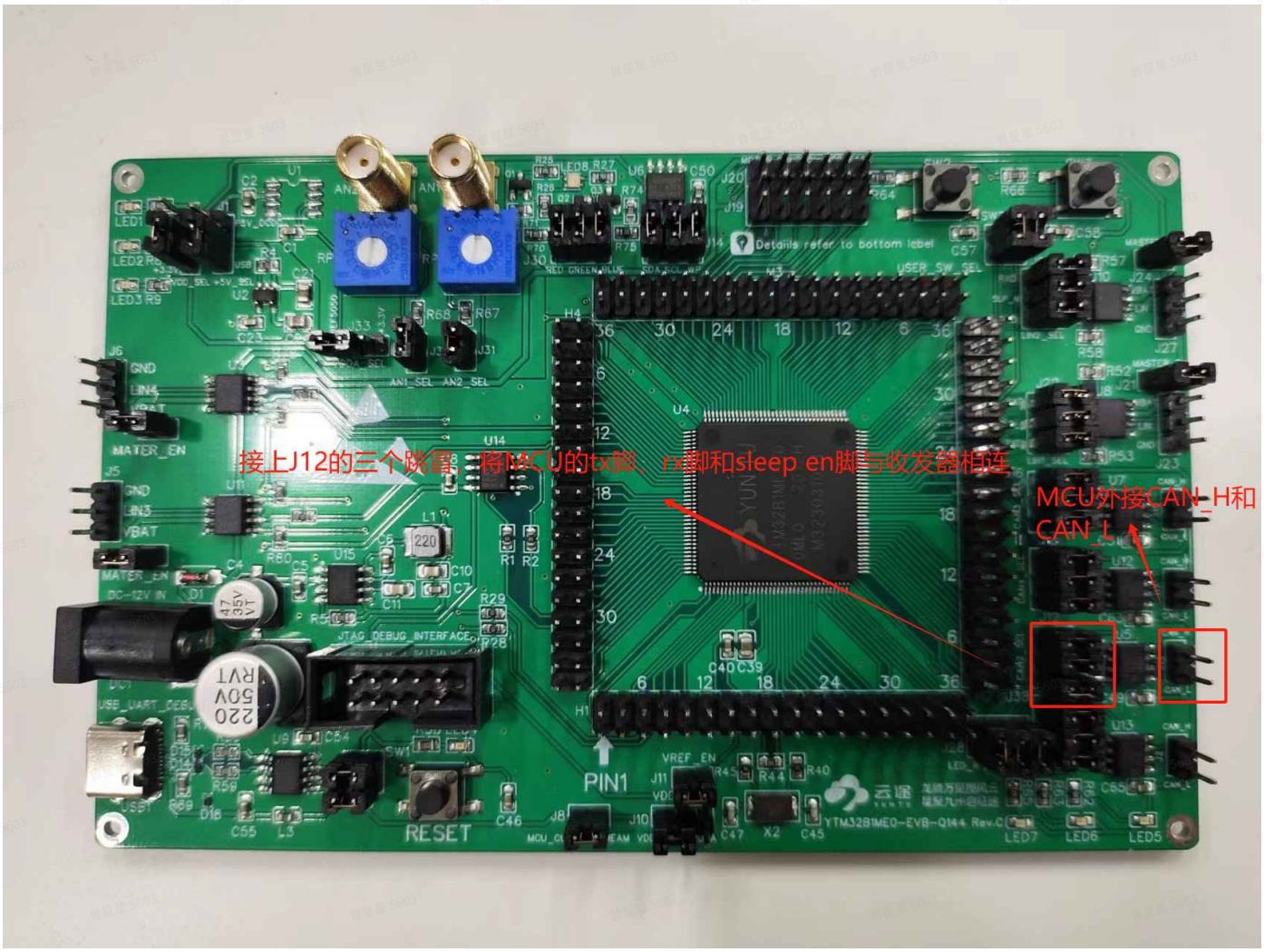
```c
void can_callback(uint8_t instance, flexcan_event_type_t eventType, \
                  uint32_t buffIdx, flexcan_state_t *flexcanState)
{
    (void)instance;
    (void)flexcanState;
    switch (eventType)
    {
    case FLEXCAN_EVENT_TX_COMPLETE:
        FLEXCAN_DRV_Send(0, 0, &txInfo, TX_MSG_ID,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

定义中断回调函数

注册中断回调函数

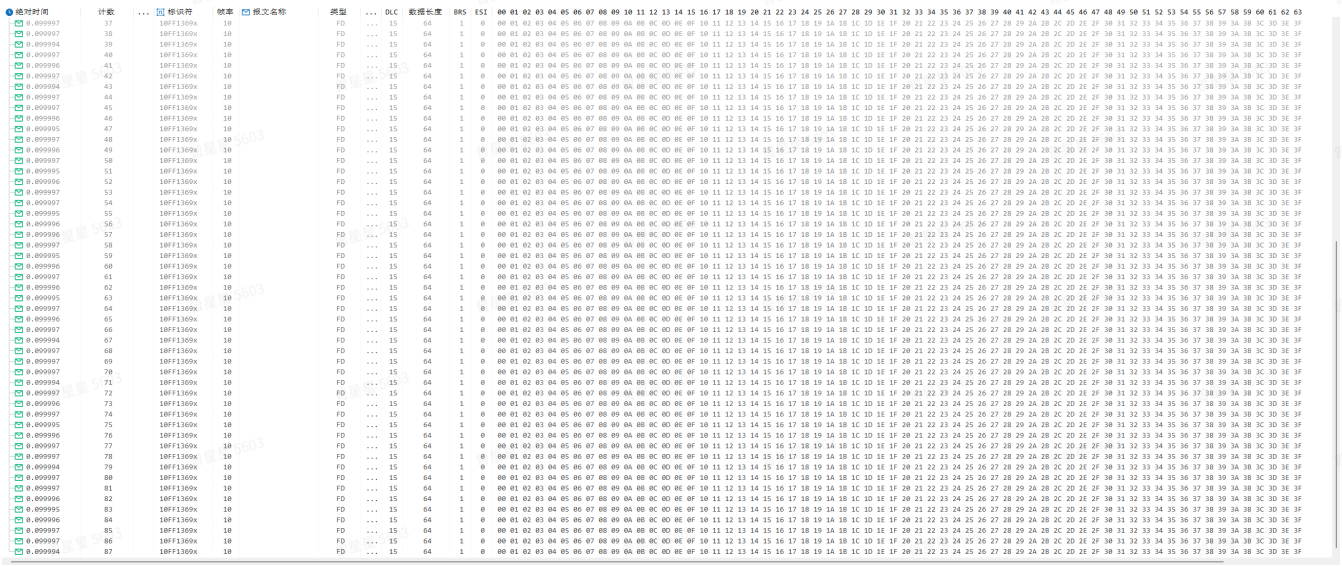
5.4 通信结果展示

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：



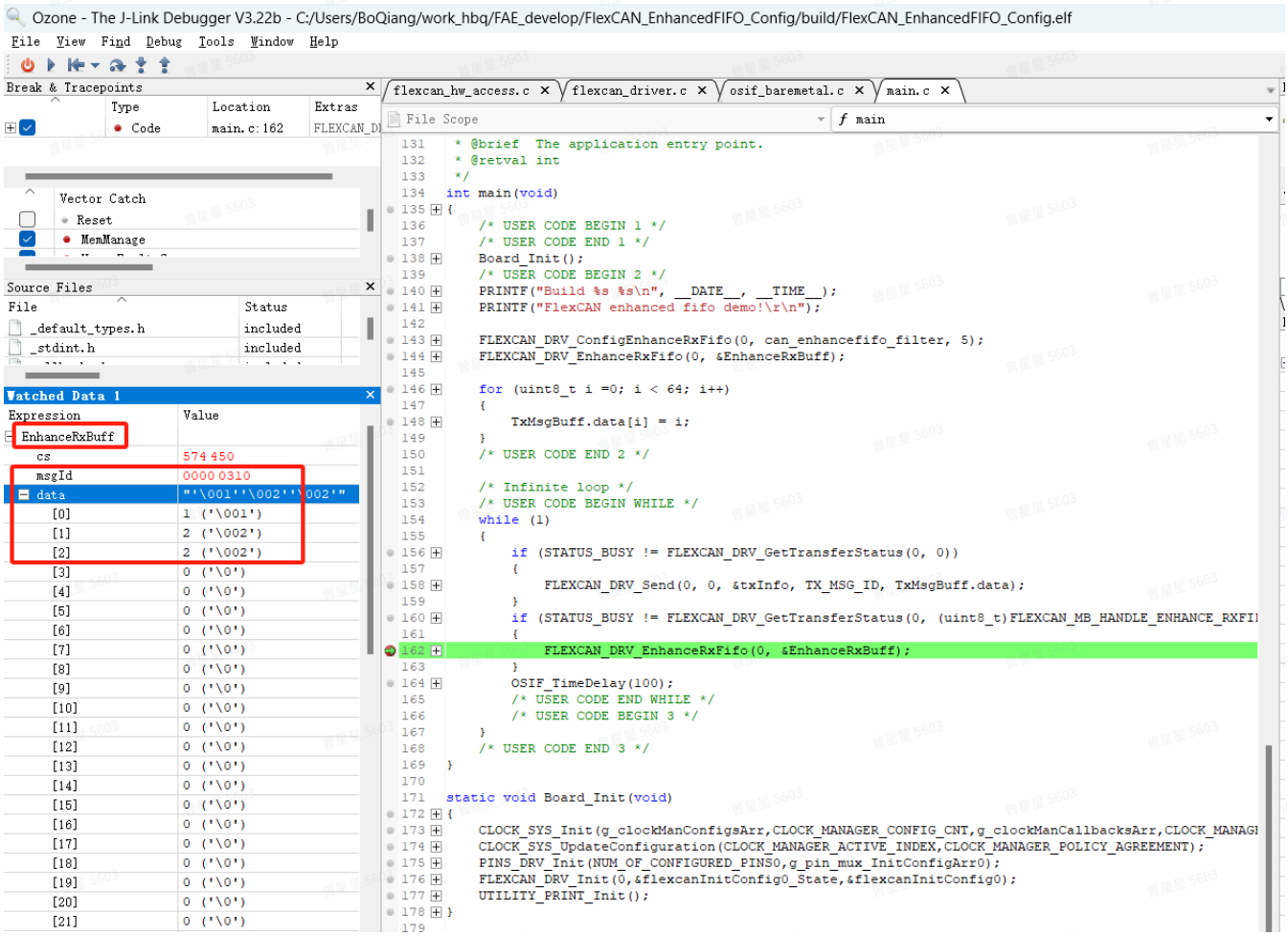
5.4.1 轮询方式

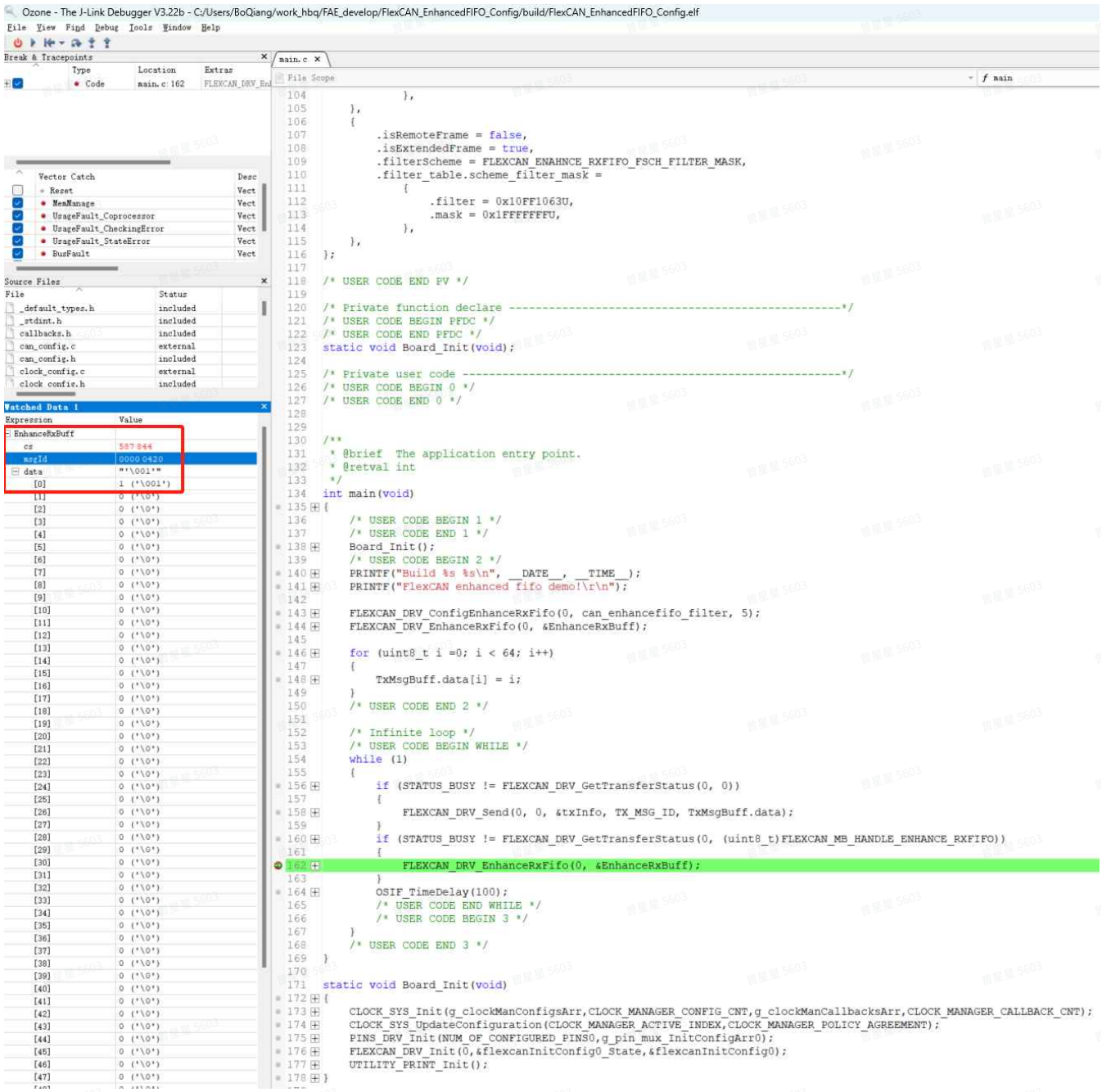
i. 上位机收到ID为0x10FF1369的报文，数据长度为64，周期100ms。

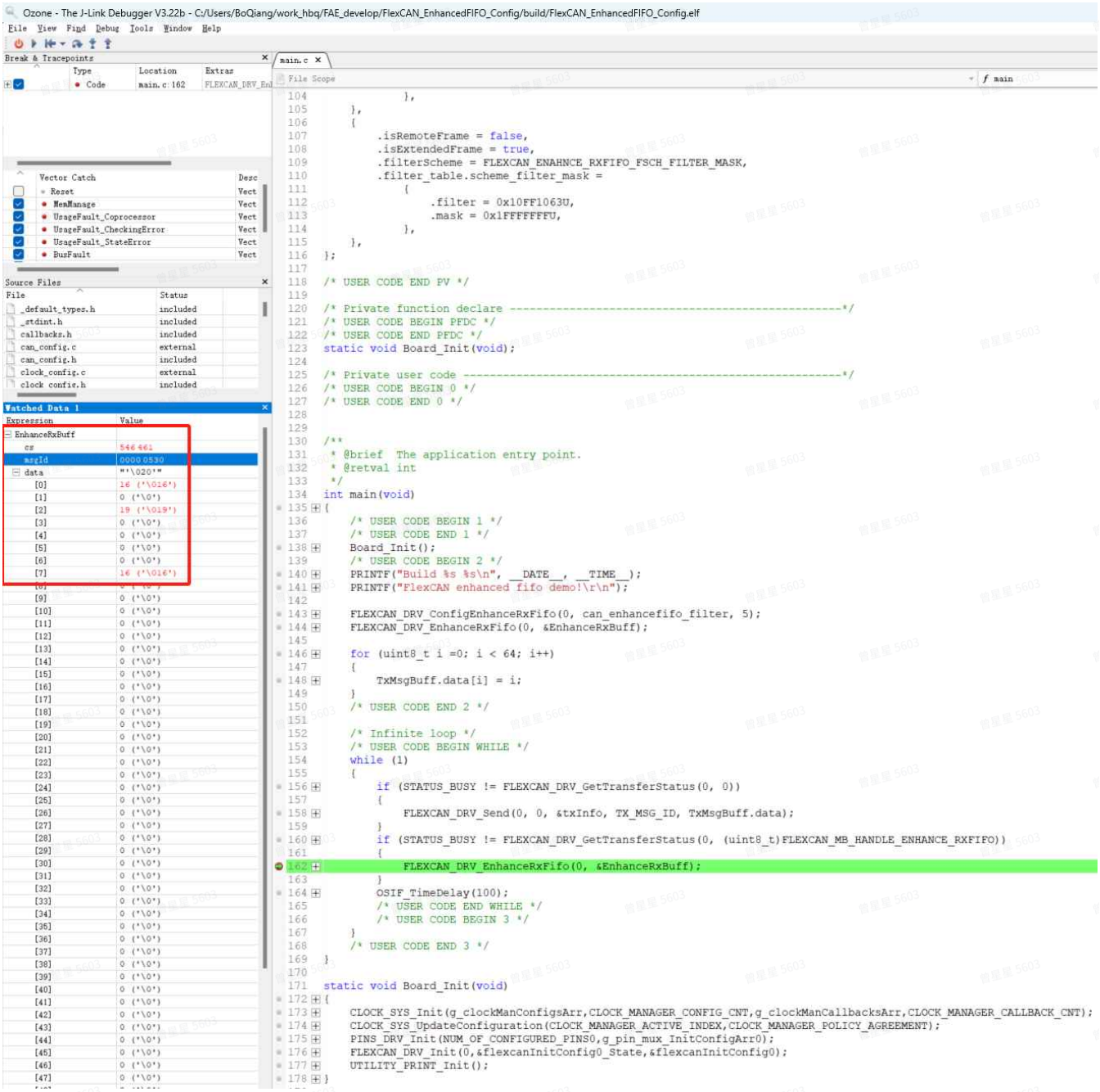


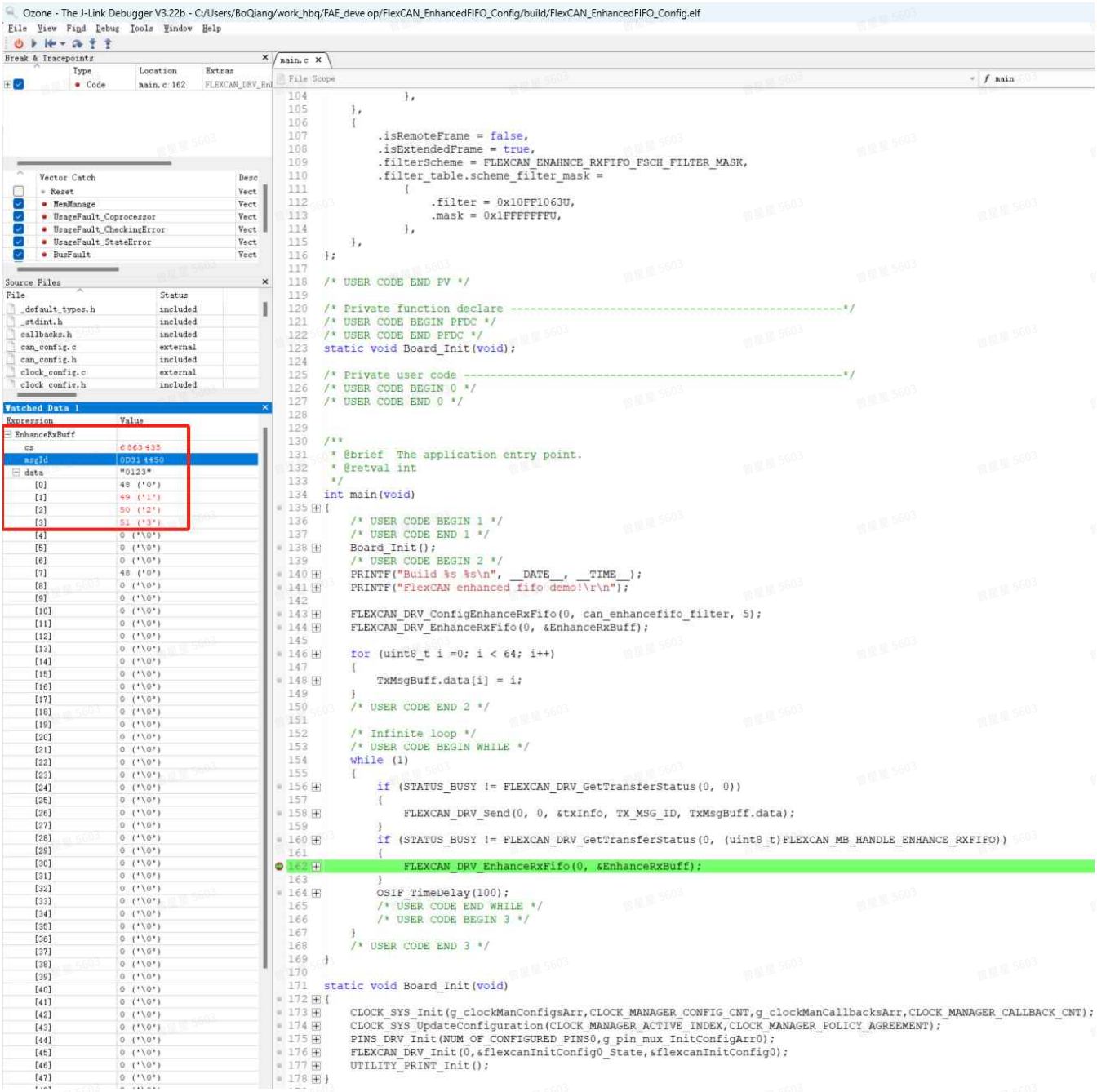
ii. demo板能成功接收到滤波表中设置的所有报文。

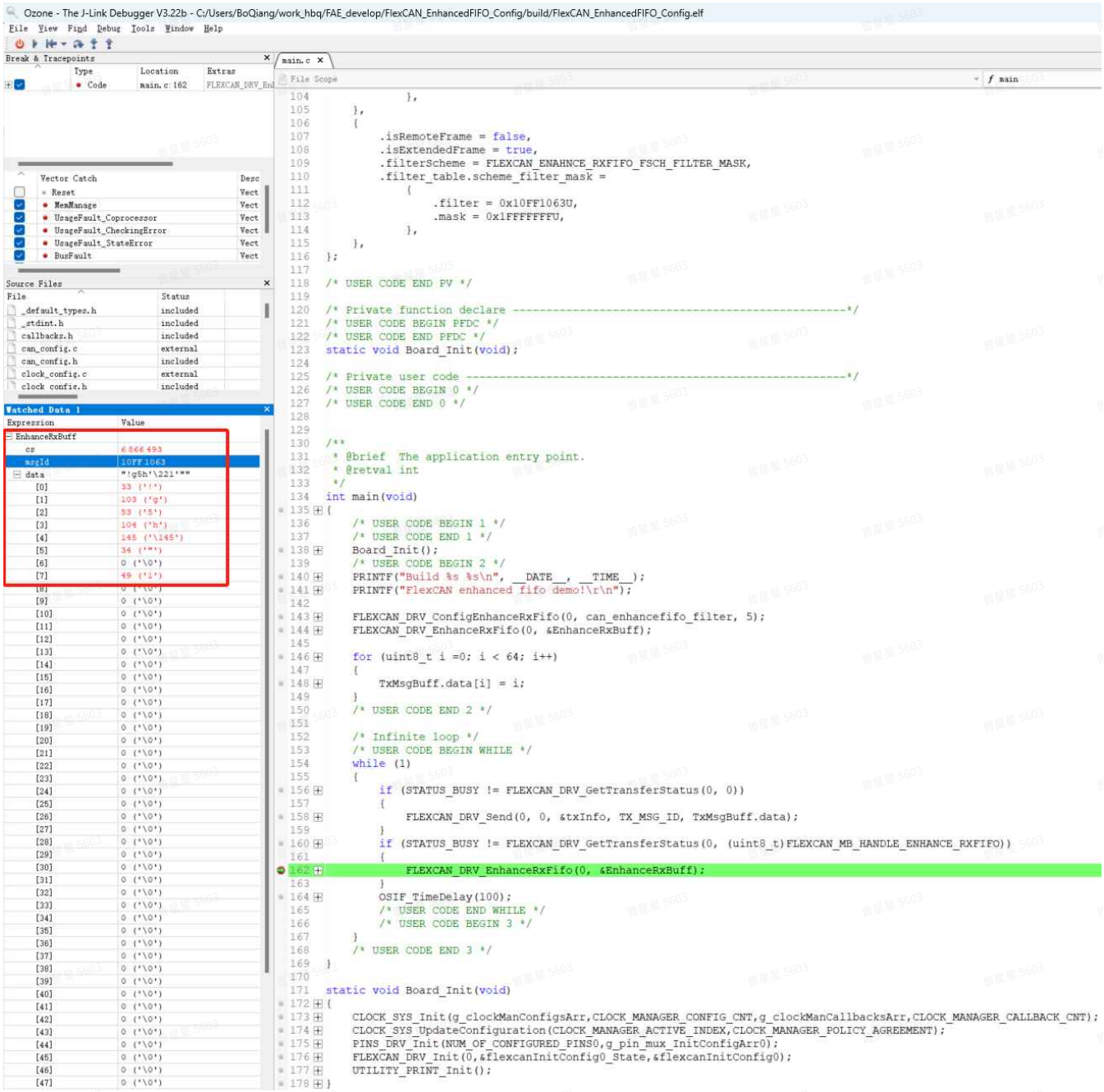
行	发送	触发	报文名称	标识符	通道	类型	DLC	BRS	D0	D1	D2	D3	D4	D5	D6	D7	注释
1	▶	手动	NewMsg	530	1	标准数据帧	8	☐	10	00	13	00	00	00	00	10	
2	▶	手动	NewMsg	0D314450	1	扩展数据帧	8	☐	30	31	32	33	00	00	00	30	
3	▶	手动	NewMsg	10FF1063	1	扩展数据帧	8	☐	21	67	35	68	91	22	00	31	
4	▶	手动	NewMsg	310	1	标准数据帧	8	☐	00	02	00	00	00	00	00	00	
5	▶	手动	NewMsg	420	1	标准数据帧	8	☐	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 Location Extras
Code main.c: 162 FLEXCAN_DRV_Enh
Vector Catch
Reset
MemManage
UsageFault_Coprocessor
UsageFault_CheckingError
UsageFault_StateError
BusFault
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
cs 587 844
msgId 0000 0420
data "'\001'"
[0] 1 ('\001')
main.c
File Scope
f main
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 PFDC */
122 /* USER CODE END 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 /**
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_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 }


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
Vector Catch
Reset
MemManage
UsageFault_Coprocessor
UsageFault_CheckingError
UsageFault_StateError
BusFault
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
cs 546 461
msgId 0000 0530
data "'\020'"
[0] 16 ('\016')
[1] 0 ('\0')
[2] 19 ('\019')
[3] 0 ('\0')
[4] 0 ('\0')
[5] 0 ('\0')
[6] 0 ('\0')
[7] 16 ('\016')
[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')
[36] 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')
main.c
File Scope
main
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 PFDC */
122 /* USER CODE END 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 /**
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_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 }


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_Enh
Vector Catch Desc
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
cs 6863435
msgId 0D314450
data "0123"
[0] 48 ('0')
[1] 49 ('1')
[2] 50 ('2')
[3] 51 ('3')
main.c
File Scope
main
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 PFDC */
122 /* USER CODE END 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 /**
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_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 }

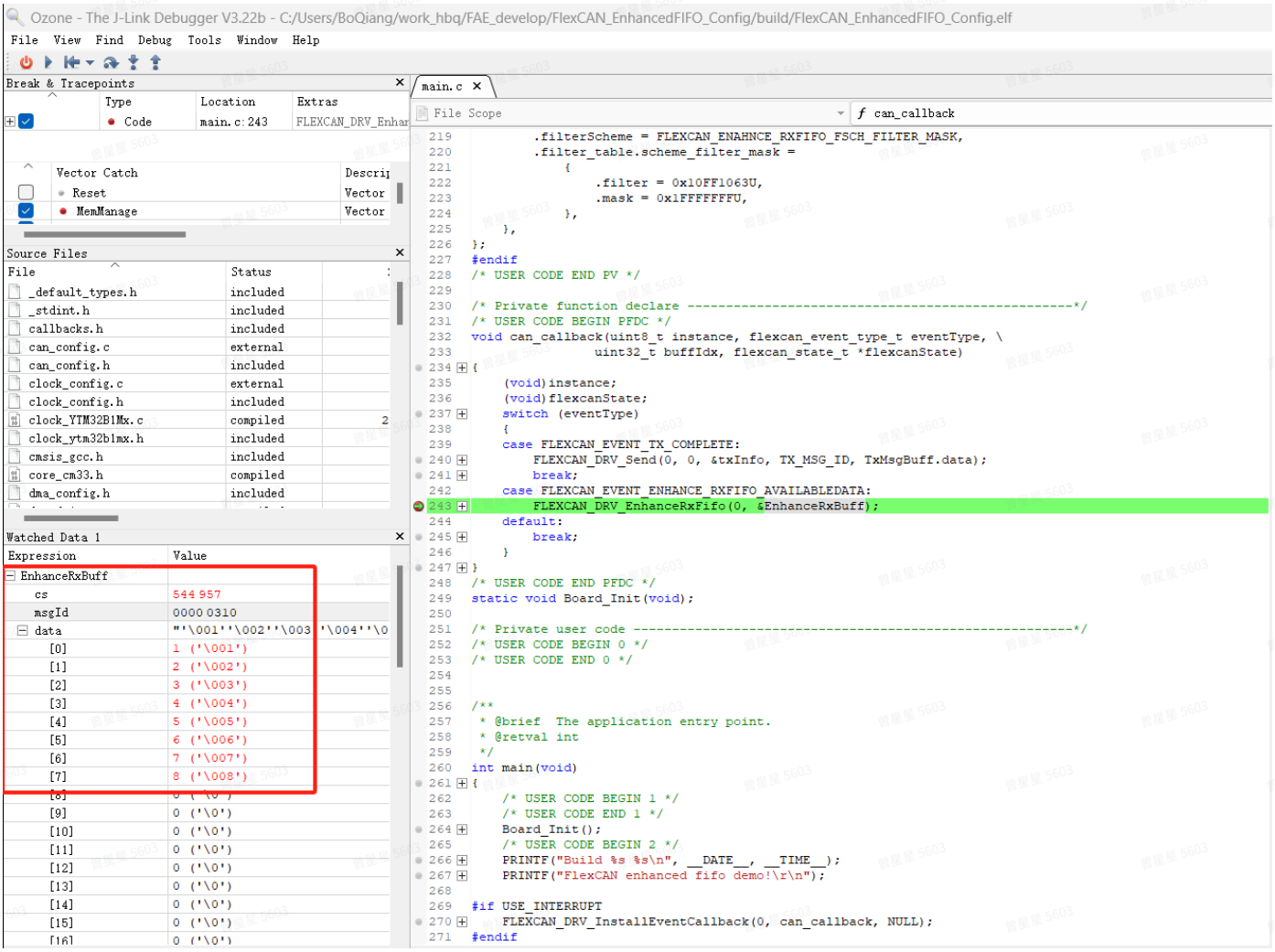


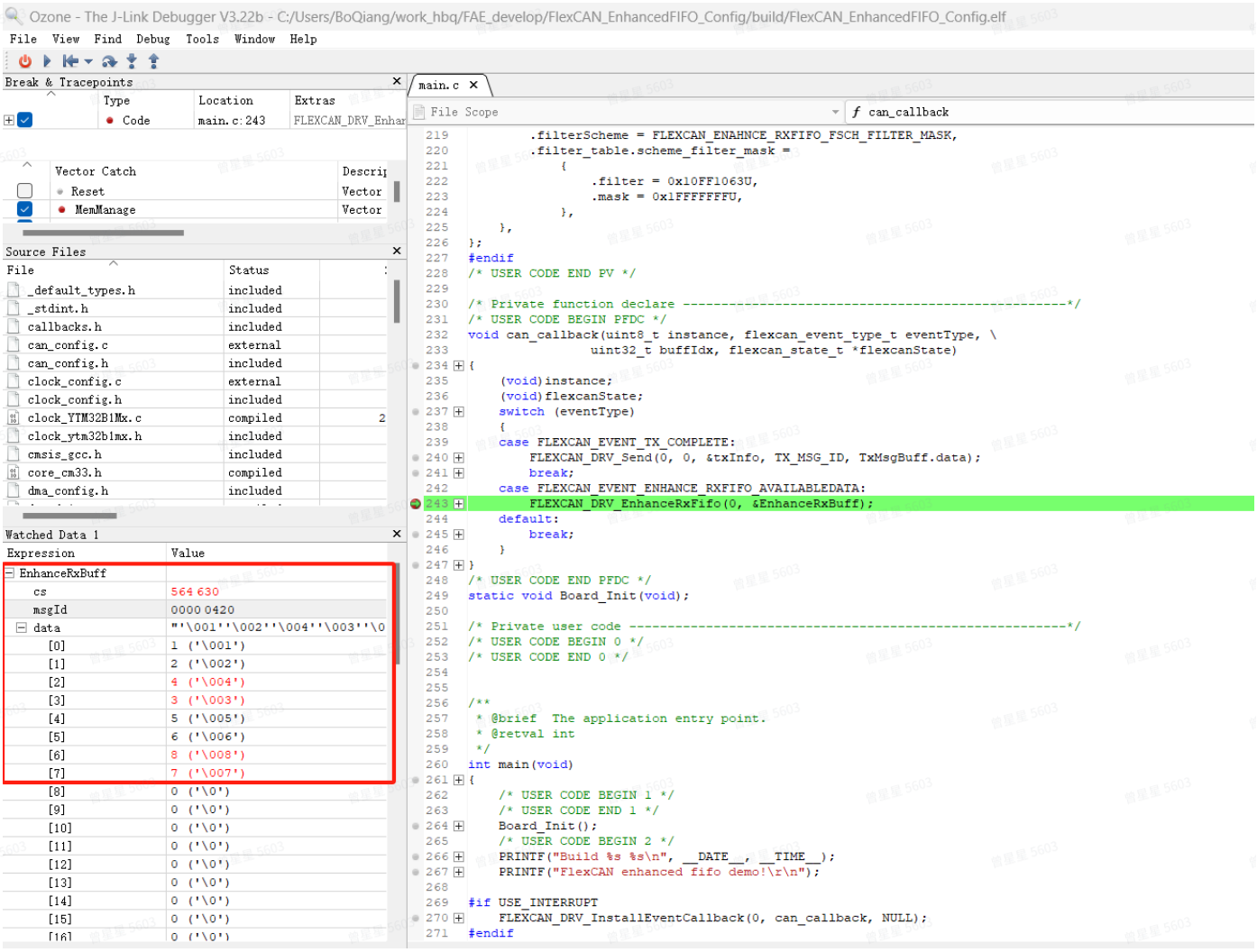
5.4.2 中断方式

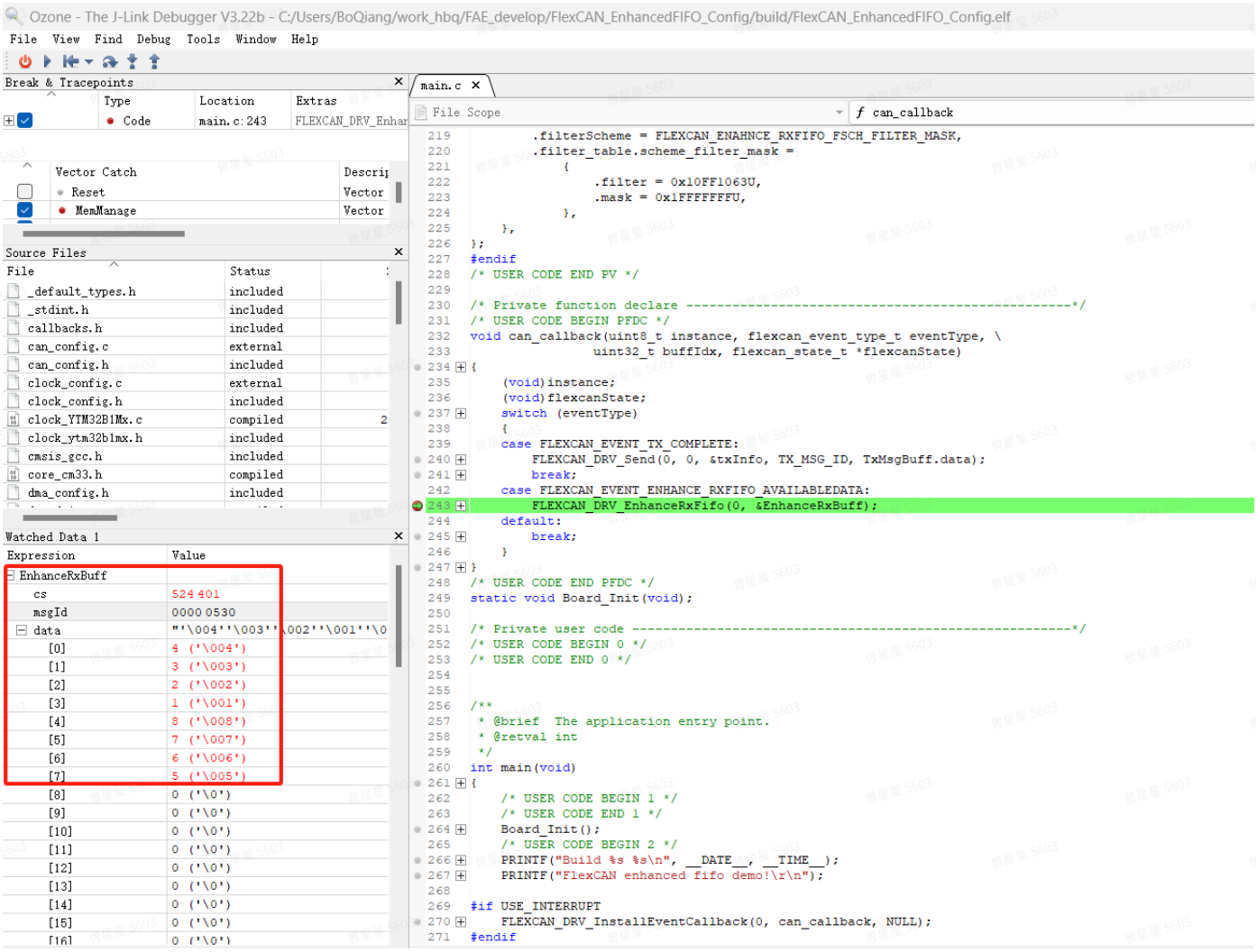
a. 上位机收到ID为0x10FF1369的报文，数据长度为64，周期约为400us。

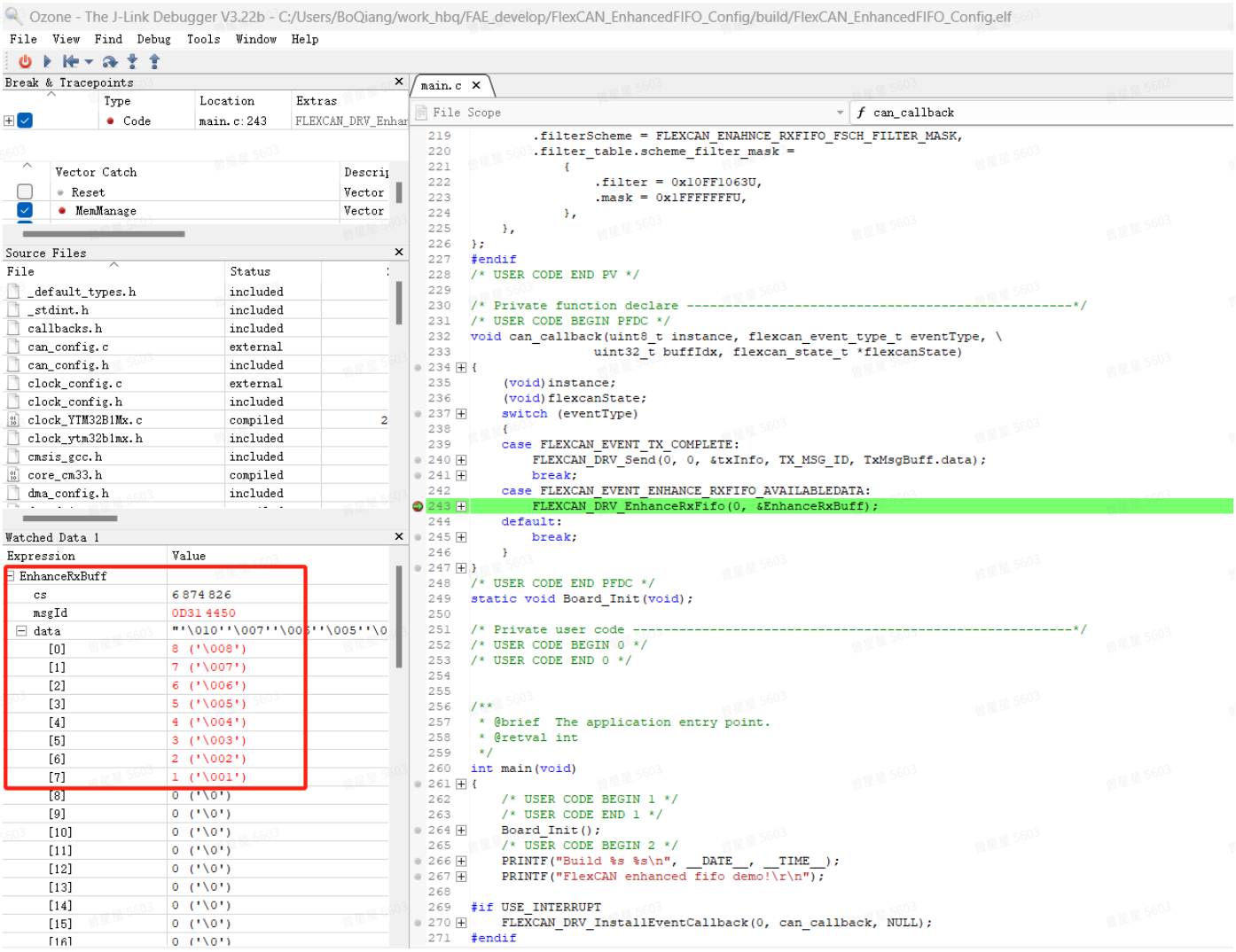
b. demo板能成功接收到滤波表中设置的所有报文。

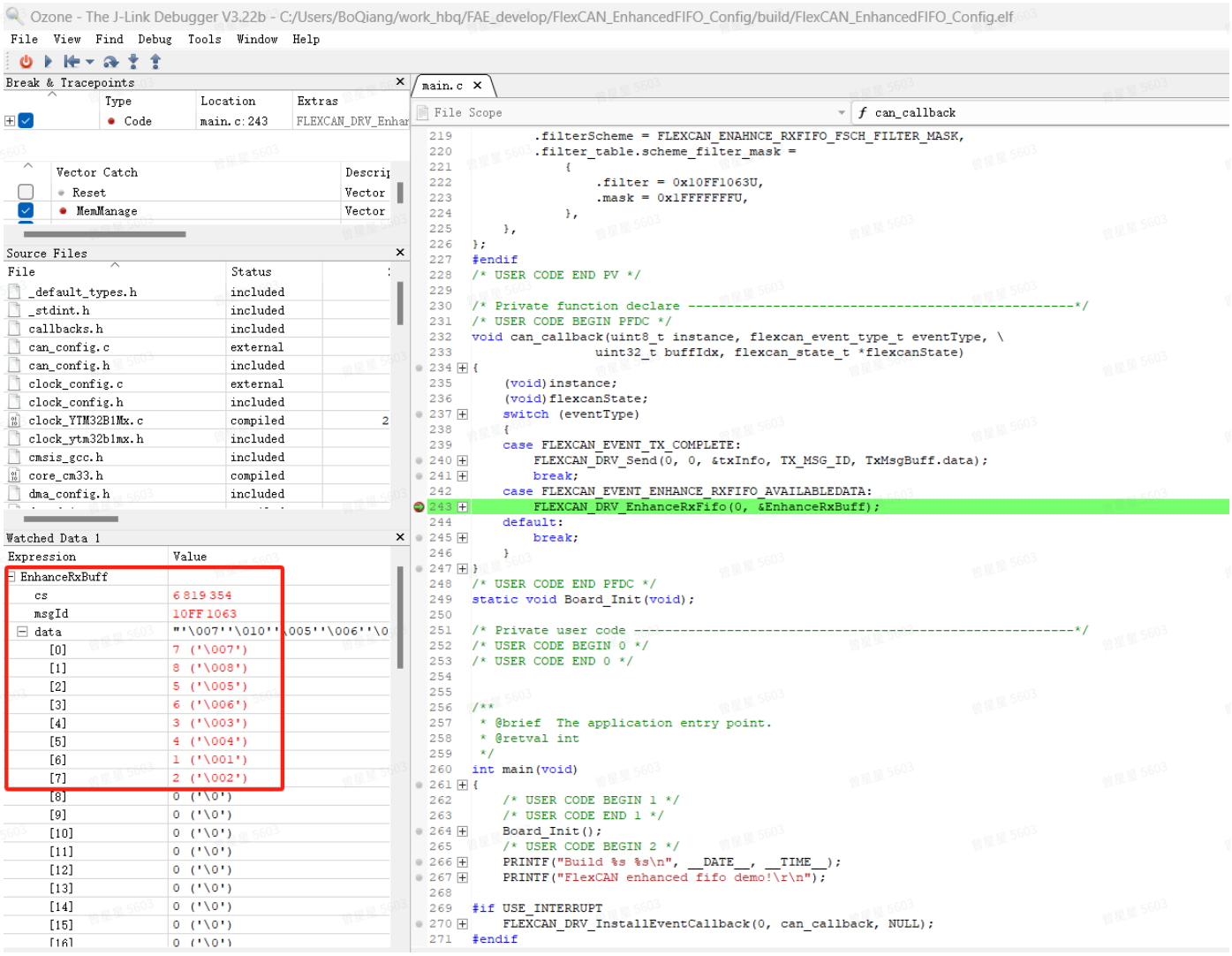
13	▶	手动	NewMsg	310	1	标准数据帧	8	☐	01	02	03	04	05	06	07	08
14	▶	手动	NewMsg	420	1	标准数据帧	8	☐	01	02	04	03	05	06	08	07
15	▶	手动	NewMsg	530	1	标准数据帧	8	☐	04	03	02	01	08	07	06	05
16	▶	手动	NewMsg	0D314450	1	扩展数据帧	8	☐	08	07	06	05	04	03	02	01
17	▶	手动	NewMsg	10FF1063	1	扩展数据帧	8	☐	07	08	05	06	03	04	01	02




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
Type Location Extras
Code main.c:243 FLEXCAN_DRV_Enhar
Vector Catch Descri
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
cs 564 630
msgId 0000 0420
data "'\001''\002''\004''\003''\0
[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 ('\0')
[9] 0 ('\0')
[10] 0 ('\0')
[11] 0 ('\0')
[12] 0 ('\0')
[13] 0 ('\0')
[14] 0 ('\0')
[15] 0 ('\0')
main.c
File Scope
can_callback
.filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,
.filter_table.scheme_filter_mask =
{
.filter = 0x10FF1063U,
.mask = 0x1FFFFFFFU,
},
},
};
#endif
/* USER CODE END PV */
/* Private function declare ----------*/
/* USER CODE BEGIN PFDC */
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;
}
}
/* 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


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_Enhar
Vector Catch
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
clock_ytm32b1mx.h included
cmsis_gcc.h included
core_cm33.h compiled
dma_config.h included
Watched Data 1
Expression Value
EnhanceRxBuff
cs 524 401
msgId 0000 0530
data "'\004''\003''\002''\001''\0
[0] 4 ('\004')
[1] 3 ('\003')
[2] 2 ('\002')
[3] 1 ('\001')
[4] 8 ('\008')
[5] 7 ('\007')
[6] 6 ('\006')
[7] 5 ('\005')
[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')
main.c
File Scope
can_callback
.filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,
.filter_table.scheme_filter_mask =
{
.filter = 0x10FF1063U,
.mask = 0x1FFFFFFFU,
},
};
#endif
/* USER CODE END PV */
/* Private function declare ----------*/
/* USER CODE BEGIN PFDC */
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;
}
}
/* 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


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_Enhar
Vector Catch
Reset
MemManage
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
clock_ytm32b1mx.h included
cmsis_gcc.h included
core_cm33.h compiled
dma_config.h included
Watched Data 1
Expression Value
EnhanceRxBuff
cs 6 874 826
msgId 0D31 4450
data
[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')
main.c
File Scope
can_callback
.filterScheme = FLEXCAN_ENAHNCE_RXFIFO_FSCH_FILTER_MASK,
.filter_table.scheme_filter_mask =
{
.filter = 0x10FF1063U,
.mask = 0x1FFFFFFFU,
},
},
};
#endif
/* USER CODE END PV */
/* Private function declare ----------*/
/* USER CODE BEGIN PFDC */
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;
}
}
/* 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



6. 主要API函数介绍

1. 配置enhanced FIFO滤波表：

```
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：

```
status_t FLEXCAN_DRV_EnhanceRxFifo(uint8_t instance, flexcan_msgbuff_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