

Embedded Systems

CS 397

TRIMESTER 3, AY 2021/22

Assignments 4

Additional Information

Dr. LIAW Hwee Choo

Department of Electrical and Computer Engineering

DigiPen Institute of Technology Singapore

HweeChoo.Liaw@DigiPen.edu

Using DMA for ADC: DMA must be initialized before ADC

The screenshot shows the STM32CubeMX interface. The left sidebar has tabs for Project, Code Generator, and Advanced Settings. The main area is divided into two panels: Driver Selector and Generated Function Calls.

Driver Selector: A search bar and a list of peripherals. The list includes RCC, GPIO, USART, DMA, ADC, DAC, CAN, and CORTEX_M7. The 'ADC' and 'DMA' items are expanded.

Generated Function Calls: A table with columns: Generate Code, Rank, Function Name, Peripheral Instance, Do Not Generate Function Call, and Visibility (Static).

Generate Code	Rank	Function Name	Peripheral Instance	Do Not Generate Function Call	Visibility (Static)
<input checked="" type="checkbox"/>	1	SystemClock_...	RCC	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	2	MX_GPIO_Init	GPIO	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	3	MX_USART3_U...	USART3	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	4	MX_DMA_Init	DMA	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	5	MX_ADC1_Init	ADC1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	6	MX_ADC3_Init	ADC3	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	7	MX_DAC_Init	DAC	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	8	MX_CAN1_Init	CAN1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	9	MX_CAN2_Init	CAN2	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Code Window: A code editor showing the initialization functions. A red arrow points to the `MX_DMA_Init();` line.

```

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USART3_UART_Init();
MX_DMA_Init();
MX_ADC1_Init();
MX_ADC3_Init();
MX_DAC_Init();
MX_CAN1_Init();
MX_CAN2_Init();
  
```

Annotations: Red arrows point from the text 'Move ADC below DMA' to the `MX_DMA_Init();` line in the code and the `MX_ADC1_Init` and `MX_ADC3_Init` entries in the table.

Move ADC below DMA

Pinout & Configuration Clock Configuration Project Manager

Project

Code Generator

Advanced Settings

Driver Selector

Search (Ctrl+F)

RCC HAL

GPIO HAL

> USART HAL

DMA HAL

> ADC HAL

DAC HAL

> CAN HAL

CORTEX_M7 HAL

Generated Function Calls

Generate Code	Rank	Function Name	Peripheral Insta...	Do Not Generate Function Ca	Visibility (Static)
<input checked="" type="checkbox"/>	1	SystemClock_...	RCC	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	2	MX_GPIO_Init	GPIO	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	3	MX_USART3_U...	USART3	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	4	MX_DMA_Init	DMA	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	5	MX_ADC1_Init	ADC1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	6	MX_ADC3_Init	ADC3	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	7	MX_DAC_Init	DAC	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	8	MX_CAN1_Init	CAN1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	9	MX_CAN2_Init	CAN2	<input type="checkbox"/>	<input checked="" type="checkbox"/>

* (indicated by a red arrow pointing to the 'MX_ADC1_Init' row)

For example, select the line by click on "ADC1", then apply the change *

```

/* main.c */
/* Private define */
/* USER CODE BEGIN PD */
#define LBYTE( x ) (uint8_t) ( x )
#define HBYTE( x ) (uint8_t) (( x ) >> 8)
/* USER CODE END PD */

/* USER CODE BEGIN PV */

// 3.3 * 2050/4096 = 1.6516
// 2050 -> 1.65V

uint32_t dac[2] = {0, 2050};
uint32_t adc[4] = {0}, buffer[3];

uint8_t TxData1[8] = {0};
uint32_t TxMailbox1;

/* USER CODE END PV */

```

```

/* USER CODE BEGIN WHILE */

.
.
.
TxHeader1.DLC = 8;
TxData1[0] = LBYTE( dac[0] );
TxData1[1] = HBYTE( dac[0] );
TxData1[2] = LBYTE( adc[0] );
TxData1[3] = HBYTE( adc[0] );
.
.
.

/* USER CODE END WHILE */

```

stm32f7xx_hal_can.h (Partial)

```

/** @defgroup CAN_identifier_type CAN Identifier Type
    * @{ */
#define CAN_ID_STD          (0x00000000U) /*!< Standard Id */
#define CAN_ID_EXT          (0x00000004U) /*!< Extended Id */
/**    * @} */

/** @defgroup CAN_remote_transmission_request CAN Remote Transmission Request
    * @{ */
#define CAN_RTR_DATA        (0x00000000U) /*!< Data frame */
#define CAN_RTR_REMOTE      (0x00000002U) /*!< Remote frame */
/**    * @} */

/** @defgroup CAN_receive_FIFO_number CAN Receive FIFO Number
    * @{ */
#define CAN_RX_FIFO0        (0x00000000U) /*!< CAN receive FIFO 0 */
#define CAN_RX_FIFO1        (0x00000001U) /*!< CAN receive FIFO 1 */
/**    * @} */

/** @defgroup CAN_Tx_Mailboxes CAN Tx Mailboxes
    * @{ */
#define CAN_TX_MAILBOX0     (0x00000001U) /*!< Tx Mailbox 0 */
#define CAN_TX_MAILBOX1     (0x00000002U) /*!< Tx Mailbox 1 */
#define CAN_TX_MAILBOX2     (0x00000004U) /*!< Tx Mailbox 2 */
/**    * @} */

```

Filter bank number
x = i = 0..27

Filter Bank Configuration & Register Organization

CAN_FiR1
Filter Num.

FBMi = 0

FSCi = 1

FBMi = 1

FSCi = 0

One 32-Bit Filter - Identifier Mask

ID	CAN_FxR1[31:24]	CAN_FxR1[23:16]	CAN_FxR1[15:8]	CAN_FxR1[7:0]	n						
Mask	CAN_FxR2[31:24]	CAN_FxR2[23:16]	CAN_FxR2[15:8]	CAN_FxR2[7:0]							
Std ID	STID[10:3]		STID[2:0]								
Ext ID	EXTID[28:21]		EXID[20:13]		EXID[12:5]		EXID[4:0]		IDE	RTR	0

Two 32-Bit Filters - Identifier List

ID	CAN_FxR1[31:24]	CAN_FxR1[23:16]	CAN_FxR1[15:8]	CAN_FxR1[7:0]	n
ID	CAN_FxR2[31:24]	CAN_FxR2[23:16]	CAN_FxR2[15:8]	CAN_FxR2[7:0]	n+1
Mapping STD ID	STID[10:3]	STID[2:0]			
Mapping Ext ID	EXTID[28:21]	EXID[20:13]	EXID[12:5]	EXID[4:0]	IDE RTR 0

Two 16-Bit Filters - Identifier Mask

ID	CAN_FxR1[15:8]	CAN_FxR1[7:0]				n
Mask	CAN_FxR1[31:24]	CAN_FxR1[23:16]				
ID	CAN_FxR2[15:8]	CAN_FxR2[7:0]				n+1
Mask	CAN_FxR2[31:24]	CAN_FxR2[23:16]				
Mapping	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]	

Four 16-Bit Filters - Identifier List

ID	CAN_FxR1[15:8]	CAN_FxR1[7:0]				n
ID	CAN_FxR1[31:24]	CAN_FxR1[23:16]				n+1
ID	CAN_FxR2[15:8]	CAN_FxR2[7:0]				n+2
ID	CAN_FxR2[31:24]	CAN_FxR2[23:16]				n+3
Mapping	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]	

FM1R = Filter Mode Register
FS1R = Filter Scale Register

ID = identifier

2. In CAN_FM1R, FBMi = (0/1) for (Identifier Mask / Identifier List) mode

1. In CAN_FS1R, FSCi = (0/1) for (dual 16-bit / single 32-bit) scale configuration

```

void CAN2_Config(void)
{
    CAN_FilterTypeDef  sFilterConfig2;

    /*##-1- Configure the CAN Filter ##*/
    sFilterConfig2.FilterBank = 14;
    .
    .
    .

    sFilterConfig2.FilterBank = 15;
    sFilterConfig2.FilterMode = CAN_FILTERMODE_IDLIST;
    sFilterConfig2.FilterScale = CAN_FILTERSCALE_32BIT;
    // sn = FF, 16-bit high
    sFilterConfig2.FilterIdHigh = ((0x00397FF5<<3) & 0xFFFF0000)>>16;
    // 16-bit low
    sFilterConfig2.FilterIdLow  = ((0x00397FF5<<3)|CAN_ID_EXT|CAN_RTR_DATA) & 0xFFFF;
    sFilterConfig2.FilterMaskIdHigh = ((0x00397FFA<<3) & 0xFFFF0000)>>16;
    sFilterConfig2.FilterMaskIdLow  = ((0x00397FFA<<3)|CAN_ID_EXT|CAN_RTR_DATA) & 0xFFFF;
    sFilterConfig2.FilterFIFOAssignment = CAN_RX_FIFO1;
    sFilterConfig2.FilterActivation = ENABLE;
    // sFilterConfig2.SlaveStartFilterBank = 14;
    if (HAL_CAN_ConfigFilter(&hcan2, &sFilterConfig2) != HAL_OK)
    {
        /* Filter configuration Error */
        Error_Handler();
    }
}

```