

I'm generating a waveform with PWM in TIM3 Ch1, the data from the array is moved with DMA, this part works properly. But now I want to change the array who sources the DMA, and I can't, someone knows why? here is the main:

[code] <--- FIXME: how I make this forum show the next paragraph like code???

```

/* Includes -----*/
#include "main.h"
#include "stm32f4xx_hal.h"
#include <string.h>
#include <math.h>
/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private variables -----*/
TIM_HandleTypeDef htim3;
DMA_HandleTypeDef hdma_tim3_ch1_trig;

/* USER CODE BEGIN PV */
/* Private variables -----*/
#define PI 3.14159
#define ASR 0.08789 //360 / 8192 = 0.04394
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_TIM3_Init(void);
void HAL_TIM_MspPostInit(TIM_HandleTypeDef *htim);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----*/
uint16_t seg0 [8192];
uint16_t seg1 [8192];
uint8_t act_seg = 0;
/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

int main(void)
{
    float angle;

    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

```

```

/* Configure the system clock */
SystemClock_Config();

/* USER CODE BEGIN SysInit */

/* USER CODE END SysInit */

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_DMA_Init();
MX_TIM3_Init();

/* USER CODE BEGIN 2 */

for (uint16_t i = 0; i < 8192; i++) {
    angle = ASR*(float)i;
    seg0 [i] = (uint16_t) rint(4096 + 4095*sinf(angle*(PI/180)));
    seg1 [i] = i;
}
// HAL_TIM_PWM_Start_DMA(&htim3, TIM_CHANNEL_1, (uint32_t *)seg [act_seg], 8192);

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    HAL_Delay(500);
    if (act_seg == 0) {
        act_seg = 1;
        HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, act_seg);
        HAL_TIM_PWM_Stop_DMA (&htim3, TIM_CHANNEL_1);
        HAL_TIM_PWM_Start_DMA(&htim3, TIM_CHANNEL_1, (uint32_t *) seg0, 8192);
    }
    else {
        act_seg = 0;
        HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, act_seg);
        HAL_TIM_PWM_Stop_DMA (&htim3, TIM_CHANNEL_1);
        HAL_TIM_PWM_Start_DMA(&htim3, TIM_CHANNEL_1, (uint32_t *) seg1, 8192);
    }
}
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */

}
/* USER CODE END 3 */

}

/** System Clock Configuration
*/
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct;
    RCC_ClkInitTypeDef RCC_ClkInitStruct;

```

```

/**Configure the main internal regulator output voltage
*/
__HAL_RCC_PWR_CLK_ENABLE();

__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);

/**Initializes the CPU, AHB and APB busses clocks
*/
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
RCC_OscInitStruct.HSIState = RCC_HSI_ON;
RCC_OscInitStruct.HSICalibrationValue = 16;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
RCC_OscInitStruct.PLL.PLLM = 16;
RCC_OscInitStruct.PLL.PLLN = 336;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV4;
RCC_OscInitStruct.PLL.PLLQ = 7;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}

/**Initializes the CPU, AHB and APB busses clocks
*/
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}

/**Configure the SysTick interrupt time
*/
HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

/**Configure the SysTick
*/
HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

/* SysTick_IRQn interrupt configuration */
HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

/* TIM3 init function */
static void MX_TIM3_Init(void)
{
    TIM_ClockConfigTypeDef sClockSourceConfig;
    TIM_MasterConfigTypeDef sMasterConfig;
    TIM_OC_InitTypeDef sConfigOC;

    htim3.Instance = TIM3;
    htim3.Init.Prescaler = 0;
    htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim3.Init.Period = 8192;

```

```

htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}

sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
if (HAL_TIM_ConfigClockSource(&htim3, &sClockSourceConfig) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}

if (HAL_TIM_PWM_Init(&htim3) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}

sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}

sConfigOC.OCMode = TIM_OCMODE_PWM1;
sConfigOC.Pulse = 100;
sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}

HAL_TIM_MspPostInit(&htim3);

}

/**
 * Enable DMA controller clock
 */
static void MX_DMA_Init(void)
{
    /* DMA controller clock enable */
    __HAL_RCC_DMA1_CLK_ENABLE();

    /* DMA interrupt init */
    /* DMA1_Stream4_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Stream4_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Stream4_IRQn);
}

/** Configure pins as
    * Analog
    * Input
    * Output
    * EVENT_OUT
    * EXTI
    PA2 -----> USART2_TX
    PA3 -----> USART2_RX

```

```

*/
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOH_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin : B1_Pin */
    GPIO_InitStructure.Pin = B1_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_IT_FALLING;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(B1_GPIO_Port, &GPIO_InitStructure);

    /*Configure GPIO pins : USART_TX_Pin USART_RX_Pin */
    GPIO_InitStructure.Pin = USART_TX_Pin|USART_RX_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    GPIO_InitStructure.Alternate = GPIO_AF7_USART2;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);

    /*Configure GPIO pin : LD2_Pin */
    GPIO_InitStructure.Pin = LD2_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(LD2_GPIO_Port, &GPIO_InitStructure);
}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @param None
 * @retval None
 */
void _Error_Handler(char * file, int line)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */
    while(1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT

```

```

/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t* file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line number,
  ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */

}

#endif

/**
 * @}
 */

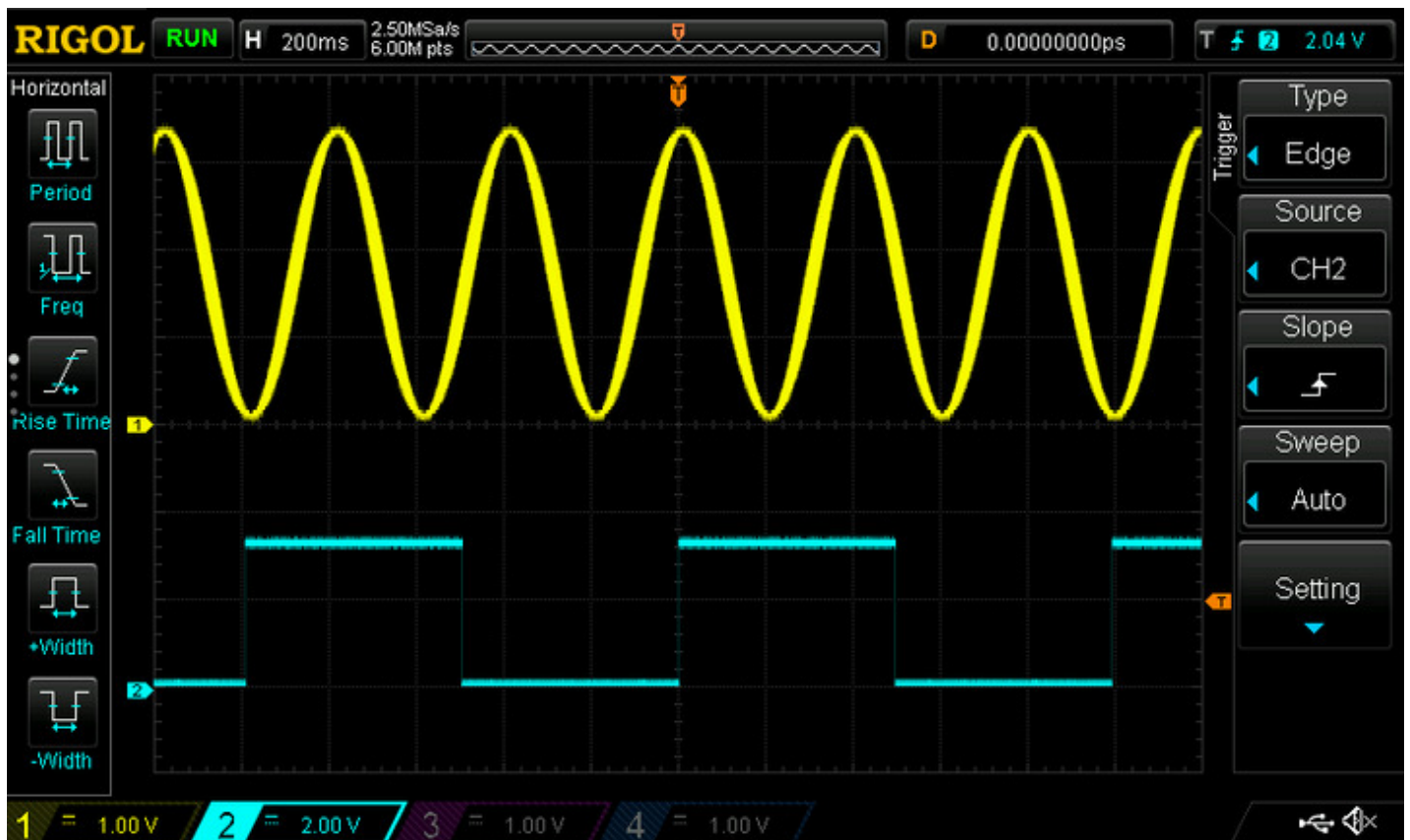
/**
 * @}
 */

/***** (C) COPYRIGHT STMicroelectronics *****/

```

[/code]

Functions were generated by CubeMX for a STM32F401RE on a nucleo board. I'm attaching the other two files. Here is the picture of the output in my scope:



any ideas on how I should make it to change the source array?