

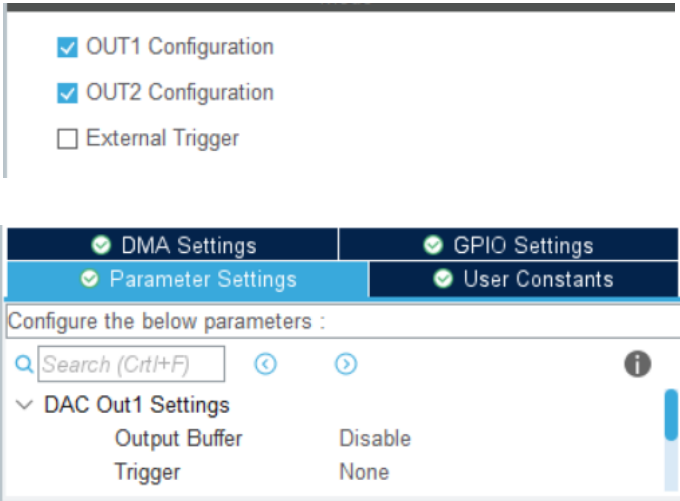
关于智能水箱嵌入式软件DAC模块的说明

作者：薛荣坤

DAC模块

DAC模块主要的作用就是用来驱动继电器，来实现对于电机的控制。代码相对简单，CUBEmx图形化设计软件生成代码可用，只用使能DAC，并填入寄存器即可。

CUBEMX配置



- 因为我们在程序中DAC可能控制的就是继电器，所以不需要采用定时器，中断，但是在电子设计大赛中，应当掌握如何使用DAC TIMER DMA输出可变频率的正弦波。
- Output Buffer 我们改为Disable;这意味着我们取消了DAC的输出缓冲，因此不应该直接将DAC直接接在用电器上，电流驱动能力减弱。

代码源码

DAC.c

```
/* Includes -----  
-----*/
```

```

#include "dac.h"

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

DAC_HandleTypeDef hdac;

/* DAC init function */
void MX_DAC_Init(void)
{

    /* USER CODE BEGIN DAC_Init 0 */

    /* USER CODE END DAC_Init 0 */

    DAC_ChannelConfTypeDef sConfig = {0};

    /* USER CODE BEGIN DAC_Init 1 */

    /* USER CODE END DAC_Init 1 */
    /** DAC Initialization
    */
    hdac.Instance = DAC;
    if (HAL_DAC_Init(&hdac) != HAL_OK)
    {
        Error_Handler();
    }
    /** DAC channel OUT1 config
    */
    sConfig.DAC_Trigger = DAC_TRIGGER_NONE;
    sConfig.DAC_OutputBuffer = DAC_OUTPUTBUFFER_DISABLE;
    if (HAL_DAC_ConfigChannel(&hdac, &sConfig, DAC_CHANNEL_1) !=
    HAL_OK)
    {
        Error_Handler();
    }
    /** DAC channel OUT2 config
    */
    sConfig.DAC_OutputBuffer = DAC_OUTPUTBUFFER_ENABLE;
    if (HAL_DAC_ConfigChannel(&hdac, &sConfig, DAC_CHANNEL_2) !=
    HAL_OK)
    {

```

```

    Error_Handler();
}

/* USER CODE BEGIN DAC_Init 2 */
/* USER CODE END DAC_Init 2 */

}

void HAL_DAC_MspInit(DAC_HandleTypeDef* dacHandle)
{

    GPIO_InitTypeDef GPIO_InitStruct = {0};
    if(dacHandle->Instance==DAC)
    {
        /* USER CODE BEGIN DAC_MspInit 0 */

        /* USER CODE END DAC_MspInit 0 */
        /* DAC clock enable */
        __HAL_RCC_DAC_CLK_ENABLE();

        __HAL_RCC_GPIOA_CLK_ENABLE();
        /**DAC GPIO Configuration
        PA4      -----> DAC_OUT1
        PA5      -----> DAC_OUT2
        */
        GPIO_InitStruct.Pin = GPIO_PIN_4|GPIO_PIN_5; //初始化2个DAC
        GPIO_InitStruct.Mode = GPIO_MODE_ANALOG; //模拟输入
        HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

        /* USER CODE BEGIN DAC_MspInit 1 */

        /* USER CODE END DAC_MspInit 1 */
    }
}

void HAL_DAC_MspDeInit(DAC_HandleTypeDef* dacHandle)
{

    if(dacHandle->Instance==DAC)
    {
        /* USER CODE BEGIN DAC_MspDeInit 0 */

        /* USER CODE END DAC_MspDeInit 0 */
        /* Peripheral clock disable */
    }
}

```

```

__HAL_RCC_DAC_CLK_DISABLE();

/**DAC GPIO Configuration
PA4      -----> DAC_OUT1
PA5      -----> DAC_OUT2
*/
HAL_GPIO_DeInit(GPIOA, GPIO_PIN_4|GPIO_PIN_5);

/* USER CODE BEGIN DAC_MspDeInit 1 */

/* USER CODE END DAC_MspDeInit 1 */
}
}

/* USER CODE BEGIN 1 */
void DAC_Set_vol(u16 vol)
{
    double temp=vol;
    temp/=1000;
    temp=temp*4096/3.3;
    HAL_DAC_SetValue(&hdac,DAC_CHANNEL_1,DAC_ALIGN_12B_R,temp);// 12
    位右对齐数据格式设置DAC值
    HAL_DAC_SetValue(&hdac,DAC_CHANNEL_2,DAC_ALIGN_12B_R,temp);//
    12位右对齐数据格式设置DAC值
    HAL_DAC_Start(&hdac,DAC_CHANNEL_1);
    HAL_DAC_Start(&hdac,DAC_CHANNEL_2);
}

/* USER CODE END 1 */

```

需要注意的部分

- 向DAC写入输出值时，应当采用右对齐，笔者因为这个错误，DEbug了许久。只有采用右对齐时，数据才是完全和输入值一样。
- ```

HAL_DAC_Start(&hdac,DAC_CHANNEL_1);
//一定要注意使能DAC，HAL库不会生成这句话，一定要加上

```

## Main.c

```
DAC_Set_Vol(1500); //输出1.5伏
```

## 硬件连接部分

1. 在硬件连接中，一定要注意DAC和硬件驱动装置中设计一个用于补充电流的电路（三极管即可），因为DAC很难长时间提供高强度电流，这对于芯片会产生极大的不稳定性。（笔者曾经因为这个问题烧坏了两个STM32F429（高性能32芯片））