

DAC

DAC的功能是将数字量转化成模拟量

DAC 可以通过软件或者硬件触发转换。

硬件触发D/A转换：可以通过定时器或者外部中断触发D/A执行D/A转换

软件触发：在程序中写入D/A触发函数，实现软件触发

HAL库中D/A转换的函数：

```
//使能启动DAC指定通道转换
HAL_StatusTypeDef HAL_DAC_Start(DAC_HandleTypeDef *hdac, uint32_t
Channel);
//配置DAC通道输出值
HAL_StatusTypeDef HAL_DAC_SetValue(DAC_HandleTypeDef *hdac,
uint32_t Channel, uint32_t Alignment, uint32_t Data);
//获取所选DAC通道最后一个输出值
uint32_t HAL_DAC_GetValue(DAC_HandleTypeDef *hdac, uint32_t
Channel);
//启动DAC使用DMA方式
HAL_StatusTypeDef HAL_DAC_Start_DMA(DAC_HandleTypeDef *hdac,
uint32_t Channel, uint32_t *pData, uint32_t Length, uint32_t
Alignment);
//停止DAC的DMA方式
HAL_StatusTypeDef HAL_DAC_Stop_DMA(DAC_HandleTypeDef *hdac,
uint32_t Channel);
```

使用DAC+DMA输出信号

DMA是直接存储器控制，DAC可以发送DMA请求请求DMA传输，这样做的好处是传输的速率相较于定时器中断传输要快。

实现方式是在一个定时器的控制下周期性的对外部实行DMA传输。

使用方式代码：

```
//启动DAC以DMA的方式传输数据
HAL_StatusTypeDef HAL_DAC_Start_DMA(DAC_HandleTypeDef *hdac,
uint32_t Channel, uint32_t *pData, uint32_t Length, uint32_t
Alignment);
//停止DAC的DMA传输
HAL_StatusTypeDef HAL_DAC_Stop_DMA(DAC_HandleTypeDef *hdac,
uint32_t Channel);
```

DAC+DMA+定时器的实现方式是每当定时器加一计数后，DMA向外输出一个值。例如当前定时器设置频率为30K，DMA设置为向外输出100个值，那么输出波形的频率为 $30K/100=300HZ$

常见用法：

- 定时器中断里面使用DAC输出

如果想要周期性的输出DAC，我们可以在一个定时器中断里面使用DAC输出。这种方法适用于低速的场景，这也是由于定时器中断耗费太多时间

- **使用定时器作为触发源，并且DMA传输**

我们可以设置定时器作为触发源，定时器每跳变一下，输出一次DAC，同时我们可以通过DMA的方式实现存储器到外设的数据传输，更加节省了cpu时间，提高了效率。