```c
/*-------------------------------------------------------------------------
 * Name:    ArbitoryFunc.c
 * Purpose: Code to ouput a specified resolution of a wave table to the DAC, using
 *          DMA requests to free up CPU time.
 * Note(s): adapted from example code found at
 *          http://00xnor.blogspot.co.uk/2014/01/6-stm32-f4-dac-dma-waveform-generator.html
 *-------------------------------------------------------------------------
 *
 *-------------------------------------------------------------------------*/

#include "STM32F4xx.h"
#include "main_2.h"

#define   DAC_DHR12R1_ADDR  0x40007408
#define   OUT_FREQ          5000                                  // Output waveform frequency
#define   WAVE_RES          128                                    // Waveform resolution
#define   CNT_FREQ          84000000                              // TIM6 counter clock (prescaled APB1)
#define   TIM_PERIOD        ((CNT_FREQ)/((WAVE_RES)*(OUT_FREQ))) // Autoreload reg value

// Sinc fucntion
const uint16_t waveForm[WAVE_RES] = { 3995, 3987, 3964, 3925, 3872, 3805, 3725, 3633, 3531, 3419,
                                      3300, 3176, 3047, 2915, 2784, 2653, 2524, 2400, 2282, 2171,
                                      2068, 1975, 1891, 1819, 1758, 1708, 1670, 1644, 1629, 1624,
                                      1630, 1646, 1669, 1700, 1738, 1780, 1827, 1876, 1926, 1977,
                                      2027, 2075, 2120, 2161, 2198, 2229, 2255, 2275, 2289, 2296,
                                      2297, 2293, 2282, 2267, 2247, 2223, 2195, 2165, 2134, 2101,
                                      2068, 2036, 2005, 1976, 1950, 1927, 1907, 1891, 1880, 1873,
                                      1870, 1871, 1877, 1886, 1899, 1916, 1935, 1956, 1979, 2003,
                                      2027, 2051, 2075, 2097, 2118, 2136, 2152, 2165, 2175, 2182,
                                      2185, 2185, 2182, 2175, 2166, 2154, 2140, 2124, 2106, 2087,
                                      2068, 2049, 2030, 2011, 1994, 1979, 1965, 1954, 1945, 1939,
                                      1935, 1935, 1937, 1941, 1948, 1957, 1969, 1981, 1996, 2011,
                                      2027, 2043, 2059, 2074, 2089, 2102, 2114, 212 };

void TIM5_Config(void)
{
  TIM_TimeBaseInitTypeDef TIM5_TimeBase;

  /* TIM5 Periph clock enable */
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM5, ENABLE);

  /* Time base configuration */
  TIM_TimeBaseStructInit(&TIM5_TimeBase);
  TIM5_TimeBase.TIM_Period        = (uint16_t)TIM_PERIOD;
  TIM5_TimeBase.TIM_Prescaler     = 0;
  TIM5_TimeBase.TIM_ClockDivision = 0;
  TIM5_TimeBase.TIM_CounterMode   = TIM_CounterMode_Up;
  TIM_TimeBaseInit(TIM6, &TIM5_TimeBase);

  /* TIM5 TRGO selection */
  TIM_SelectOutputTrigger(TIM5, TIM_TRGOSource_Update);

  /* TIM5 enable counter */
  TIM_Cmd(TIM5, ENABLE);
}

void DAC_Ch1_ArbitoryConfig(void)
{
  DAC_InitTypeDef DAC_INIT;
  DMA_InitTypeDef DMA_INIT;

  /* DAC channel1 Configuration */
  DAC_INIT.DAC_Trigger        = DAC_Trigger_T5_TRGO;
  DAC_INIT.DAC_WaveGeneration = DAC_WaveGeneration_None;
  DAC_INIT.DAC_OutputBuffer   = DAC_OutputBuffer_Enable;
  DAC_Init(DAC_Channel_1, &DAC_INIT);

  /* DMA1_Stream5 channel7 configuration **************************************/
  DMA_DeInit(DMA1_Stream5);
  DMA_INIT.DMA_Channel            = DMA_Channel_7;
  DMA_INIT.DMA_PeripheralBaseAddr = (uint32_t)DAC_DHR12R1_ADDR;
  DMA_INIT.DMA_Memory0BaseAddr    = (uint32_t)&waveForm;
  DMA_INIT.DMA_DIR                = DMA_DIR_MemoryToPeripheral;
  DMA_INIT.DMA_BufferSize         = WAVE_RES;
  DMA_INIT.DMA_PeripheralInc      = DMA_PeripheralInc_Disable;
  DMA_INIT.DMA_MemoryInc          = DMA_MemoryInc_Enable;
  DMA_INIT.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
```

```
78      DMA_INIT.DMA_MemoryDataSize     = DMA_MemoryDataSize_HalfWord;
79      DMA_INIT.DMA_Mode               = DMA_Mode_Circular;
80      DMA_INIT.DMA_Priority           = DMA_Priority_High;
81      DMA_INIT.DMA_FIFOMode           = DMA_FIFOMode_Disable;
82      DMA_INIT.DMA_FIFOThreshold      = DMA_FIFOThreshold_HalfFull;
83      DMA_INIT.DMA_MemoryBurst        = DMA_MemoryBurst_Single;
84      DMA_INIT.DMA_PeripheralBurst    = DMA_PeripheralBurst_Single;
85      DMA_Init(DMA1_Stream5, &DMA_INIT);
86
87      /* Enable DMA1_Stream5 */
88      DMA_Cmd(DMA1_Stream5, ENABLE);
89
90      /* Enable DAC Channel1 */
91      DAC_Cmd(DAC_Channel_1, ENABLE);
92
93      /* Enable DMA for DAC Channel1 */
94      DAC_DMACmd(DAC_Channel_1, ENABLE);
95    }
96
97    void DAC_Arbitory_On(void)
98    {
99      /* Enable DAC Channel1 */
100     DAC_Cmd(DAC_Channel_1, ENABLE);
101   }
102
103   void DAC_Arbitory_Off(void)
104   {
105     /* Disable DAC Channel1 */
106     DAC_Cmd(DAC_Channel_1, DISABLE);
107   }
108
```