

The program on the following page sends the variable tmp to the D/A converter. Each time through the for loop the variable is incremented by 200 so tmp goes from 0 to 4000. The result is a ramp function which has a delay determined by timer 3 between samples. The timer is set for 1 millisecond.

Part 1:

Modify the program so that instead of sending out a ramp function it sends out 20 values stored in an array of ints which vary sinusoidally between 0 and 4000. MATLAB[®] can be used to generate the values:

```
%SineOut
%Produces sine wave data
t = 0:.05:.95;
y = fix(2000*(sin(2*pi*t) + 1));
figure(1);clf;
plot(t, y);
fprintf(1, '%d ', y);
```

This program prints:

2000, 2618, 3175, 3618, 3902, 4000, 3902, 3618, 3175, 2618, 2000, 1381, 824, 381, 97, 0, 97,
381, 824, 1381

Part 2:

Modify the timer value so that the frequency of the sinusoid is 1000 Hz.

```

/*ArraySineOut
*/
#include "stm32f407vg.h"

int main()
{int i, tmp;
  RCC_AHB1ENR |= 1;           //Bit 0 is GPIOA clock enable bit
  RCC_APB1ENR |= (1 << 29); //Bit 29 is DAC clock enable bit
  RCC_APB1ENR |= 2;           //Enable peripheral timer for timer 3 (bit 1)

  GPIOA_MODER |= 0x4000;     //Bits 15-14 = 01 for digital output on PA7
  //OTYPER register resets to 0 so it is push/pull by default
  GPIOA_OSPEEDER |= 0xC000; //Bits 15-14 = 11 for high speed on PA7
  //PUPDR defaults to no pull up no pull down
  GPIOA_MODER |= 0xF00;      //PA4-PA5 are analog
  GPIOA_PUPDR &= 0xFFFF0FF; //Pins PA4 PA5 are no pull up and no pull down
  DAC_CR |= 0x3E;            //Bits 3, 4, 5 = 111 for software trigger ch1
                               //Bit 2 = 1 for Ch 1 trigger enabled
                               //Bit 1 = 1 for Ch 1 output buffer enabled
  DAC_CR |= 1;               //Bit 0 = 1 for Ch 1 enabled
  //Timer 3 bits
  TIM3_CR1 |= (1 << 7);      //Auto reload is buffered
  TIM3_CR1 |= (1 << 3);      //One pulse mode is on. ADD THIS LINE
  TIM3_PSC = 0;              //Don't use prescaling
  TIM3_ARR = 16000;          //16 MHz/16000 = 1000 Hz
  TIM3_CR1 |= 1;            //Enable Timer
  while(1)
  {tmp = 0;
   for(i=0;i<20;i++)
   {DAC_DHR12R1 = tmp;
    DAC_SWTRIGR |= 0x1;      //Start the D/A conversion
    tmp += 4000/20;
    while((TIM3_CR1 & 1) != 0); //Wait here until timer runs out
    TIM3_CR1 |= 1;          //Restart timer
   }
  }
}

```