```c
// Link this source code with his .h file.
#include "dac.h"


void config_DAC(void)
{
  /*
    config_DAC :: void -> void

    Enables the DAC, the output
    pin is P0.26.
  */

  LPC_PINCON->PINSEL1  |= (2<<20);                    // Configure the pin P0.26 function as AOUT.
  LPC_PINCON->PINMODE1 |= (2<<20);                    // Pull-up-pull-down not enabled.
  LPC_DAC->DACCTRL      = 0;                          // DMA not enabled
}

void config_timer_dac(void)
{
  /*
    config_timer_dac :: void -> void

    Configure the Timer1 to
    change the sample of
    the DAC.
  */

  LPC_SC->PCONP |=(1<<1);                             // Configure the power supply.
  LPC_TIM1->PR   = 0;                                 // No prescale -> 25MHz.
  LPC_TIM1->MCR |= 3;                                 // When the time counter reaches MR0 interrupts and reset the Timer Counter.
  NVIC_EnableIRQ(TIMER1_IRQn);                        // Enables the interruption.
}

void generate_samples(void)
{
  /*
    generate_samples :: void -> void

    Generate the samples of
    the sinusoidal signal.
  */

  int t;
  for(t=0; t < N_SAMPLES; t++)
    samples[t]= (uint16_t)(1023 *                     // Calculate the corresponding sample of the sine
        (0.5 + 0.5 * sin(2*PI*t/N_SAMPLES)));
}

void TIMER1_IRQHandler(void)
{
  /*
    TIMER1_IRQHandler :: void -> void

    Handles the interruption that is
    generated when the Timer1 matchs
    the sample period. This handles
    changes the value of the DAC.
  */
  static char index = 0;
  LPC_TIM1->IR|=(1<<0);                               // Clear the interruption flag of the timer.
  LPC_DAC->DACR=samples[index++]<<6;                  // Change the value of the DAC.

  if(index == N_SAMPLES -1 )                          // If we go through all the samples
    index = 0;                                        // Restart from the begining
}

```