```c
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "adc.h"
#include "dac.h"
#include "usart.h"
#include "usb_otg.h"
#include "gpio.h"
#include "pds.h"

#define N_MUESTRAS      128
#define FREQ_MUESTREO   10000
#define BITS 10

/* Header added to the stream */
struct header_struct {
    char     head[4];
    uint32_t id;
    uint16_t N;
    uint16_t fs ;
    uint32_t maxIndex;
    uint32_t minIndex;
    q15_t    maxValue;
    q15_t    minValue;
    q15_t    rms;
    char     tail[4];
} header={"head",0,N_MUESTRAS,FREQ_MUESTREO,0,0,0,0,0,"tail"};


uint32_t tick   = 0   ;
uint16_t tone   = 440 ;
uint16_t B      = 4000;
uint16_t sweept = 10;

void SystemClock_Config(void);

int main(void)
{
  /* System Initialization */
  HAL_Init();
  SystemClock_Config();
  MX_GPIO_Init();
  MX_USART3_UART_Init();
  MX_USB_OTG_FS_PCD_Init();
  MX_ADC1_Init();
  MX_USART2_UART_Init();
  MX_DAC_Init();

  uint16_t sample = 0;
  DBG_CyclesCounterInit(CLOCK_SPEED); // Enable the cycle counter
  int16_t adc [N_MUESTRAS];
  uint16_t tone_value = 0;
  float t = 0;
```

```
  while (1)
  {
    /* Reset cycle counter to 0 */
    DBG_CyclesCounterReset();

    /* Get the ADC sample */
     adc[sample] = ((((int16_t )ADC_Read(0)-512))>>(10-BITS))<<(6+(10-BITS));

    /* Send the sample in an Array */
     uartWriteByteArray(&huart2, (uint8_t* )&adc[sample], sizeof(adc[0]));

    t = tick/(float)header.fs;
    tick++;

    /* Calculate the the tone value. The nucleo board has a DAC that can work in
12 or 8 bits.*/
    tone_value = 2048*arm_sin_f32 (t*tone*2*PI)+2048;
    DAC_Write( &hdac, tone_value);

    /* Increment the sample counter and check if we are in the last sample */
    if ( ++sample==header.N )
    {
        /* Send the max value */
        DAC_Write( &hdac, 2048);

        /* Blinks at fs/N frequency */
        gpioToggle (GPIOB,LD1_Pin);

        /* Calculate max, min and rms */
        arm_max_q15 ( adc, header.N, &header.maxValue,&header.maxIndex );
        arm_min_q15 ( adc, header.N, &header.minValue,&header.minIndex );
        arm_rms_q15 ( adc, header.N, &header.rms                       );

        /* Increment id */
        header.id++;

        /* Send the header in an Array */
        uartWriteByteArray (&huart2, (uint8_t*)&header, sizeof(header));

        /* Reset the samples */
        sample = 0;
    }
    /* Blinks at fs/2 frequency */
    gpioToggle (GPIOB,LD3_Pin);

    /* Wait until it completes the Cycles. 168.000.000/10.000 = 16.800 cycles */
    while(DBG_CyclesCounterRead() < CLOCK_SPEED/header.fs);
  }
}
```