EE 113D LAB 1B
Lab 1

Khyle Calpe
405016683

EE113D: Digital Signal Processing Design

Lab 1: Waveform Generation and Measurement

Objective:

The objective of Lab 1 is to introduce the hardware and software functionality of the STM32 Nucleo board through the generation of sinusoidal waves in C. More specifically, the lab focuses on the use of the debugger and the trace function for data verification.

STEP 1:

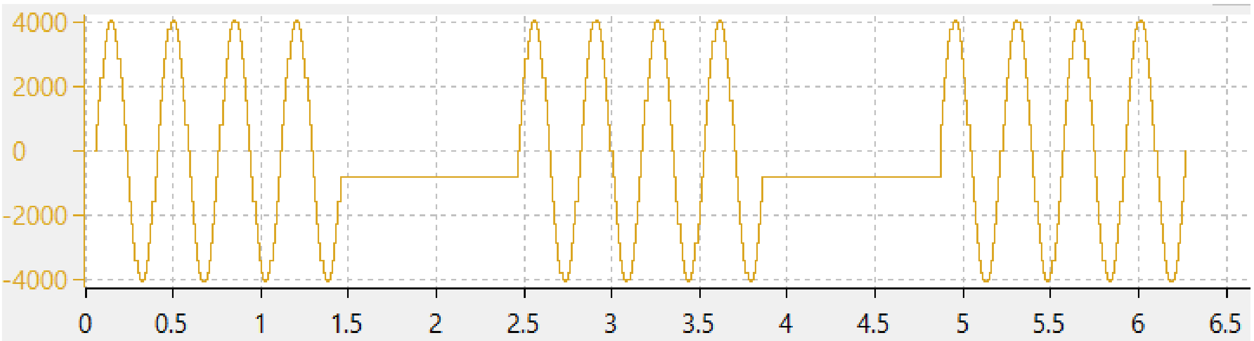
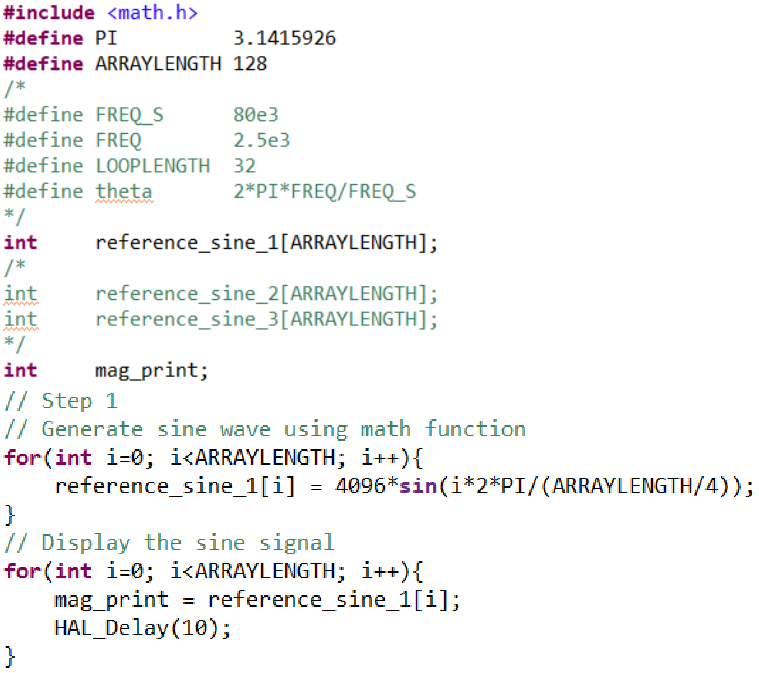


Fig. 1: Figure of the 3 generated sine waves. The first wave is generated through the math library.

CODE:



```
#include <math.h>
#define PI          3.1415926
#define ARRAYLENGTH 128
/*
#define FREQ_S      80e3
#define FREQ        2.5e3
#define LOOPLENGTH  32
#define theta       2*PI*FREQ/FREQ_S
*/
int     reference_sine_1[ARRAYLENGTH];
/*
int     reference_sine_2[ARRAYLENGTH];
int     reference_sine_3[ARRAYLENGTH];
*/
int     mag_print;
// Step 1
// Generate sine wave using math function
for(int i=0; i<ARRAYLENGTH; i++){
    reference_sine_1[i] = 4096*sin(i*2*PI/(ARRAYLENGTH/4));
}
// Display the sine signal
for(int i=0; i<ARRAYLENGTH; i++){
    mag_print = reference_sine_1[i];
    HAL_Delay(10);
}
```

STEP 2:

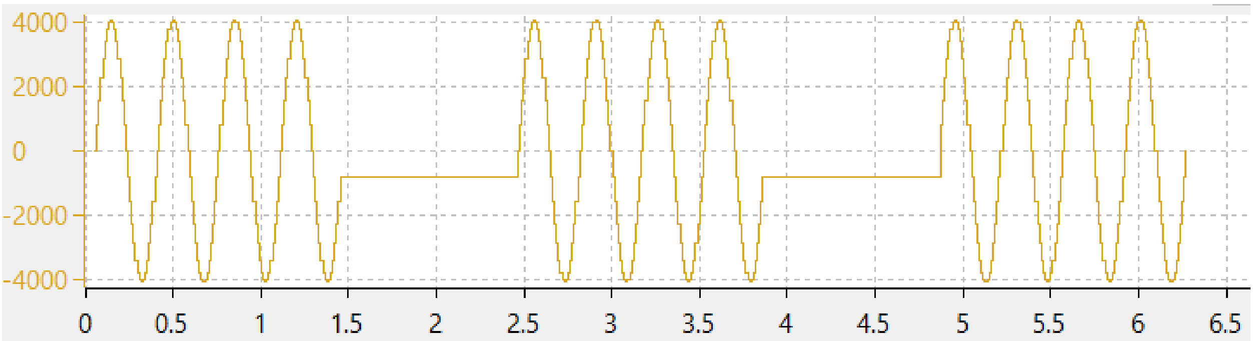
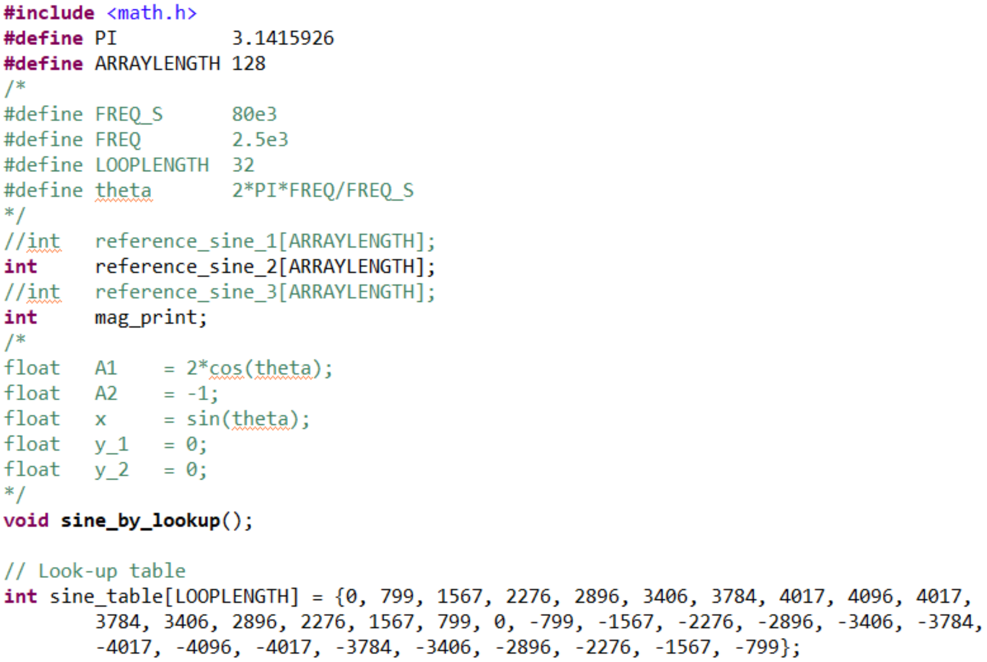


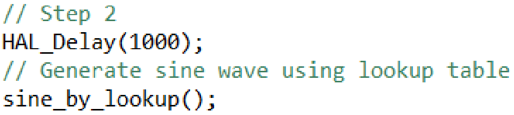
Fig. 2: Figure of the 3 generated sine waves. The second wave is generated through a look-up table.

CODE:

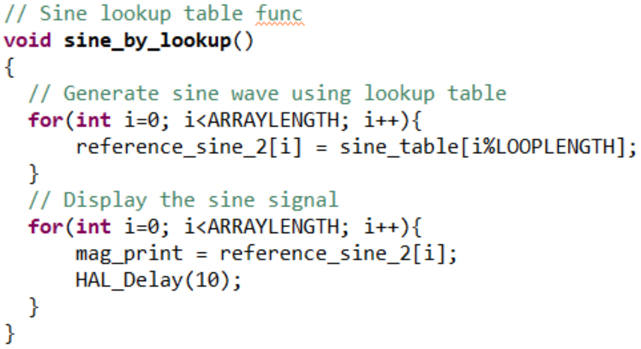


```
#include <math.h>
#define PI          3.1415926
#define ARRAYLENGTH 128
/*
#define FREQ_S      80e3
#define FREQ        2.5e3
#define LOOPLENGTH  32
#define theta       2*PI*FREQ/FREQ_S
*/
//int   reference_sine_1[ARRAYLENGTH];
int     reference_sine_2[ARRAYLENGTH];
//int   reference_sine_3[ARRAYLENGTH];
int     mag_print;
/*
float   A1    = 2*cos(theta);
float   A2    = -1;
float   x     = sin(theta);
float   y_1   = 0;
float   y_2   = 0;
*/
void sine_by_lookup();

// Look-up table
int sine_table[LOOPLENGTH] = {0, 799, 1567, 2276, 2896, 3406, 3784, 4017, 4096, 4017,
        3784, 3406, 2896, 2276, 1567, 799, 0, -799, -1567, -2276, -2896, -3406, -3784,
        -4017, -4096, -4017, -3784, -3406, -2896, -2276, -1567, -799};
```



```
// Step 2
HAL_Delay(1000);
// Generate sine wave using lookup table
sine_by_lookup();
```



```
// Sine lookup table func
void sine_by_lookup()
{
  // Generate sine wave using lookup table
  for(int i=0; i<ARRAYLENGTH; i++){
      reference_sine_2[i] = sine_table[i%LOOPLENGTH];
  }
  // Display the sine signal
  for(int i=0; i<ARRAYLENGTH; i++){
      mag_print = reference_sine_2[i];
      HAL_Delay(10);
  }
}
```

STEP 3:

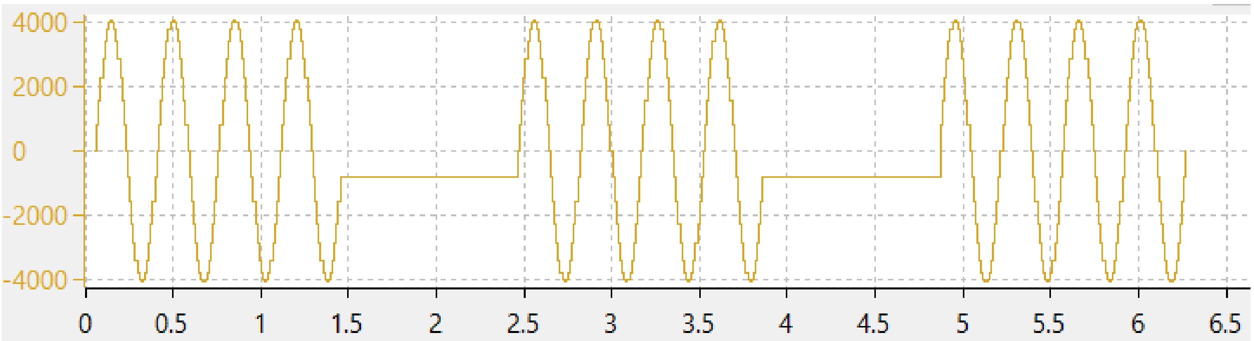


Fig. 3: Figure of the 3 generated sine waves. The third wave is generated through the difference equation:

$$y(n) = x(n) + 2\cos(\theta)\, y(n-1) - y(n-2)$$

CODE:

```
#include <math.h>
#define PI          3.1415926
#define ARRAYLENGTH 128
#define FREQ_S      80e3
#define FREQ        2.5e3
#define LOOPLENGTH  32
#define theta       2*PI*FREQ/FREQ_S

int     reference_sine_1[ARRAYLENGTH];
int     reference_sine_2[ARRAYLENGTH];
int     reference_sine_3[ARRAYLENGTH];
int     mag_print;

float   A1    = 2*cos(theta);
float   A2    = -1;
float   x     = sin(theta);
float   y_1   = 0;
float   y_2   = 0;


// Step 3
HAL_Delay(1000);
// Generate sine wave using difference equation
float y_n = x + A1*y_1 + A2*y_2;
for(int i=0; i<ARRAYLENGTH; i++){
    reference_sine_3[i] = 4096*y_n;
    y_2 = y_1;
    y_1 = y_n;
    y_n = 0 + A1*y_1 + A2*y_2;
}
// Display the sine signal
  for(int i=0; i<ARRAYLENGTH; i++){
    mag_print = reference_sine_3[i];
    HAL_Delay(10);
  }
```