**LAB-2 REPORT**

Subject: **Embedded Hardware Design**

Subject Code: **EL203**

Members: 1) Shantanu Tyagi **(201801015)**

2) Nikhil Mehta **(201801030)**

3) Sudhanshu Mishra **(201801114)**

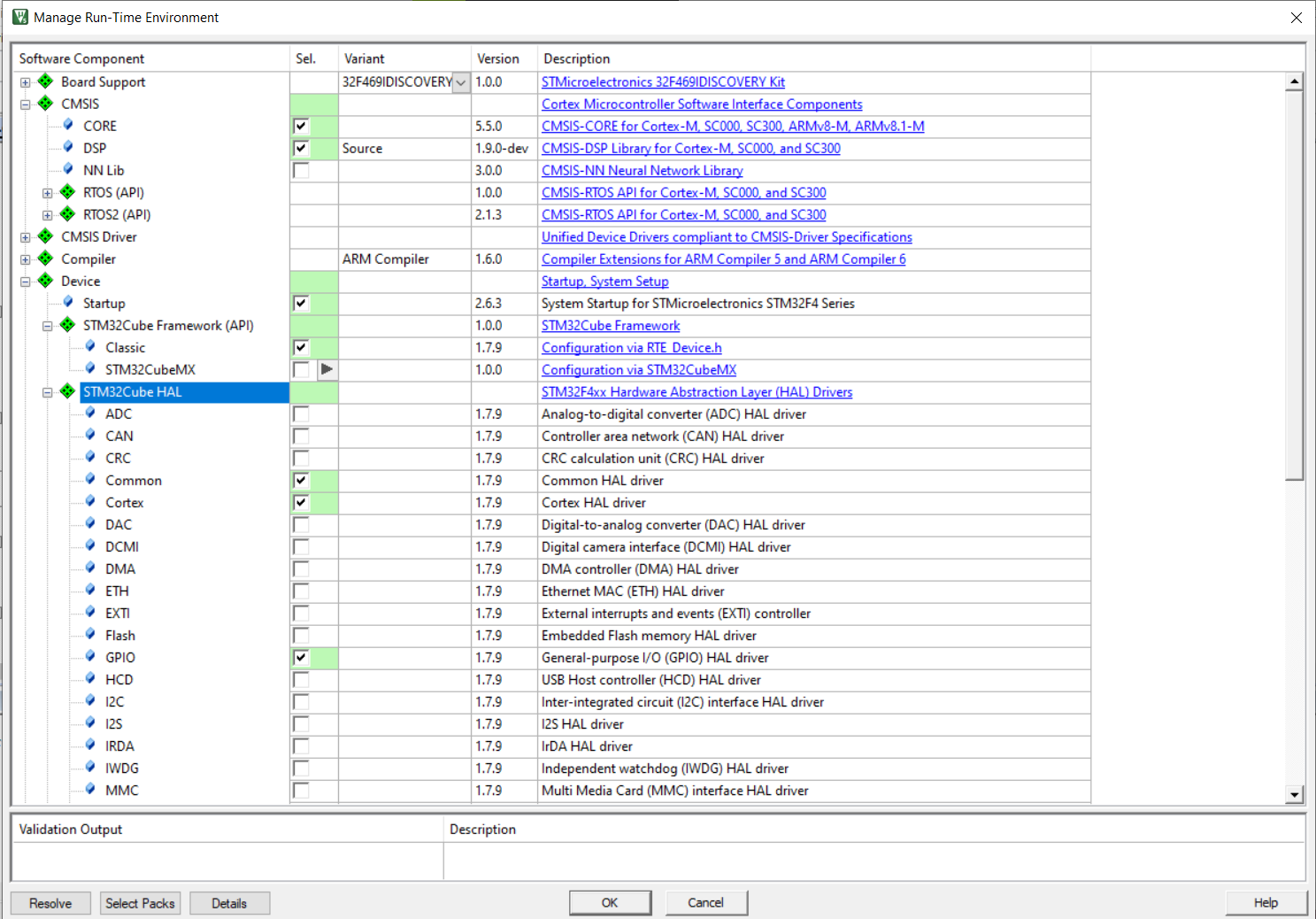
4) Bhavana Kolli **(201701082)**

**Installation of the Software**

Download Keil Microcontroller Development Kit from the link as provided <https://www2.keil.com/mdk5>

**Environment Setup**

In the IDE, install the Keil::STM32F4xx DFP Package from Pack Manager.

1. Create a new Project and rename it using the appropriate nomenclature.
   * + 1. Set STM32F407VGTx as the target device under the device header
       2. Modify the options for the target device:
2. 
   1. Create a new group under the project
   2. Make modifications for the following additional options for the target device:
      1. Set ARM compiler version 6 under the Target -> Code Generation header
      2. Select the Simulator radio button and load the KEIL\_STM.ini file as an initialization file under the Debug header
   3. Add the main.c and sinewave.c file under the created group in the project with the relevant header file (#include "stm32f4xx\_hal.h",#include "arm\_math.h")

**Running the Simulation**

**1** To run the code, first we have to compile the main.c file. This is done by clicking on the build button in Keil IDE.

**2** Once the build is complete, we are prompted with errors and we need to fix them.

**3** After fixing the bugs we move to the debug section and run the code to get the results.

**4** To see the results, we can either see the result value or check out the graph for the simulation.

**5** We can see the values of the variables directly by clicking on the variable and then on ‘add to watch’.

**6** In order to get the graph, we select the variable and click on ‘analyze’.

**7** After checking the output, we will have to close the debug session to make further changes in the code. Then follow the steps mentioned above again to see the new output.

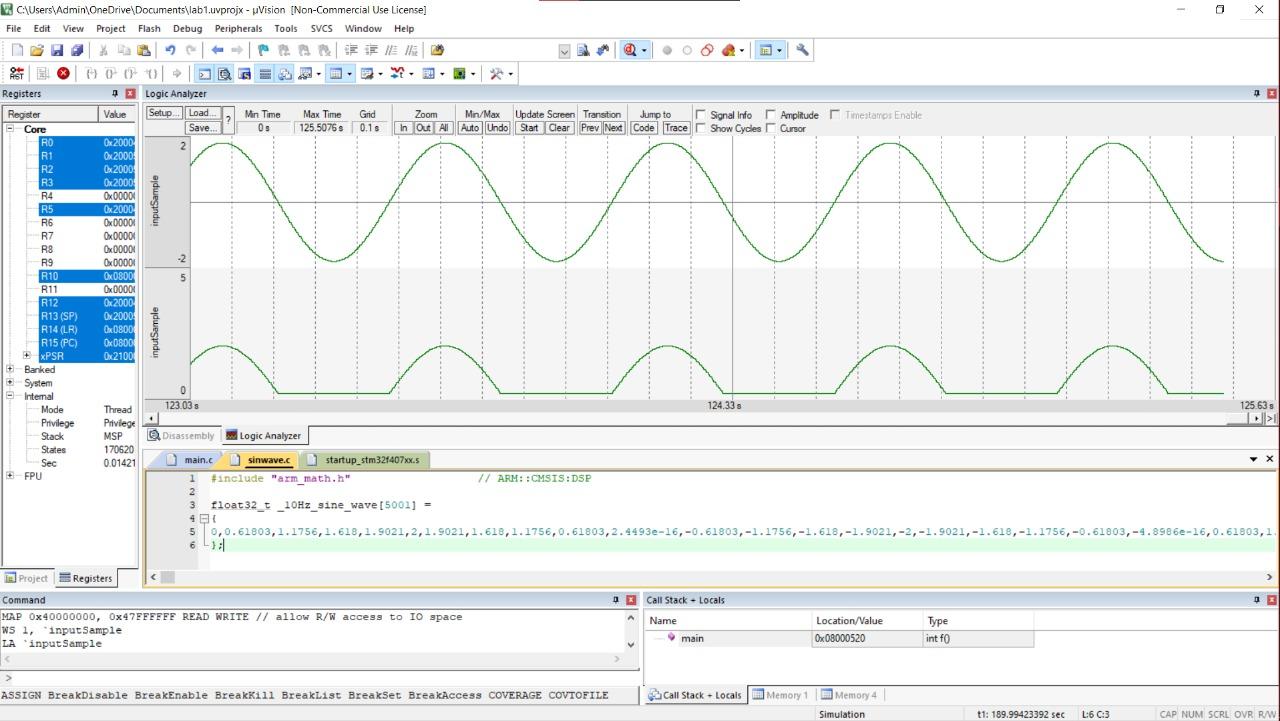
**CODE:**

| **% MATLAB**  **clear all;  freq = 5; Amp = 2; ts = 0.01;**  **T = 5;  t=0:ts:T; y = Amp\*sin(2\*pi\*freq\*t); plot(t,y) csvwrite("\_5Hz\_sine\_wave.txt",y)** |
| --- |

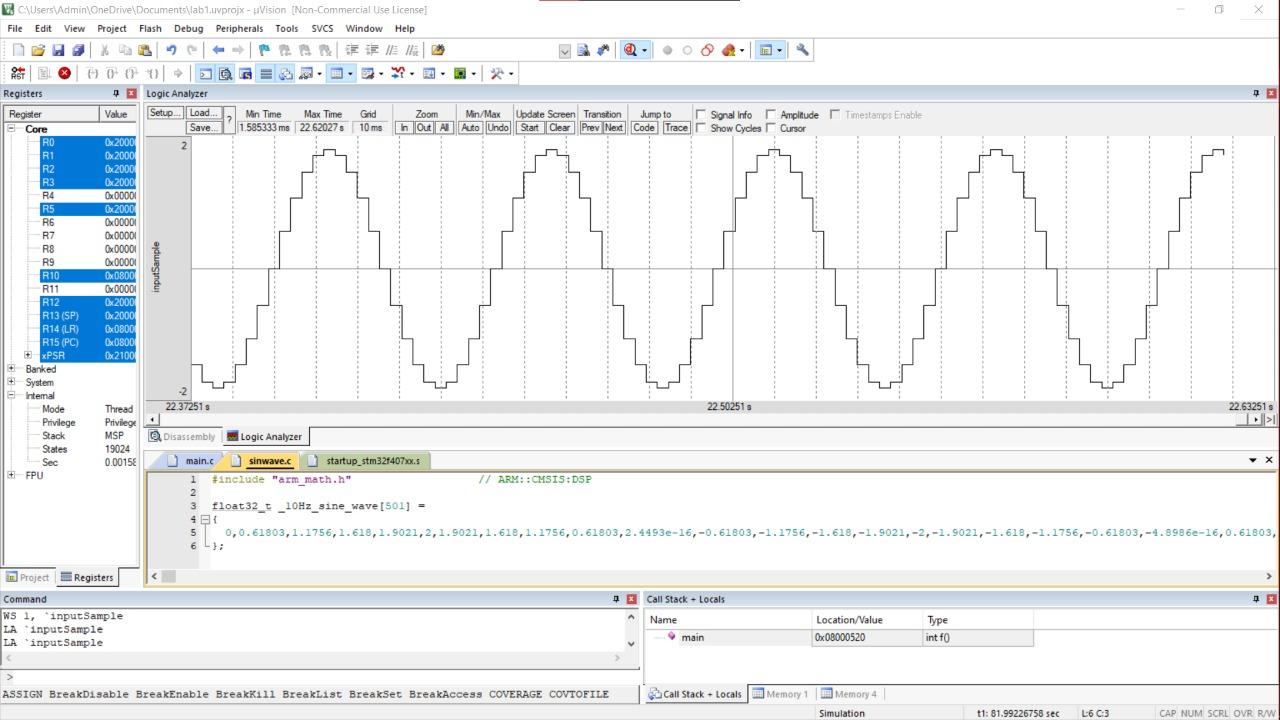
| **// main.c**  **# include "stm32f4xx\_hal.h" // Keil::Device:STM32Cube HAL:Common # include "arm\_math\_f16.h" // ARM::CMSIS:DSP  # define SIG\_LENGTH 5001  uint32\_t freq; extern float32\_t \_10Hz\_sine\_wave[SIG\_LENGTH]; //void SysTick\_Handler(void); void plot\_sine\_wave(void);  float32\_t inputSample;  int main() {  HAL\_Init();  freq = HAL\_RCC\_GetHCLKFreq();  plot\_sine\_wave();  while(1)  {  plot\_sine\_wave();  } } void plot\_sine\_wave(void) {  int i,j;  for(i = 0;i<SIG\_LENGTH;i++)  {  inputSample = \_10Hz\_sine\_wave[i];  HAL\_Delay(1);  //for(j=0;j<3000;j++){}  } } void SysTick\_Handler(void) {  HAL\_IncTick();  HAL\_SYSTICK\_IRQHandler(); }** |
| --- |

| **// sinewave.c #include "arm\_math.h" // ARM::CMSIS:DSP  float32\_t \_10Hz\_sine\_wave[501] = {  // enter 501 values of sine wave generated from matlab. }** |
| --- |

**SIG\_LENGTH=5001**

****

**SIG\_LENGTH=501**

****

**Conclusion:**

1. As we increase the size of the array containing sine wave values, it tends closer to the actual sinewave and interpolation error reduces.
2. This happens when we reduce the time step value in MATLAB giving us higher accuracy.