LAB-1 REPORT

Subject: **Embedded Hardware Design**

Subject Code: **EL203**

Members: 1) Shantanu Tyagi **(201801015)**
          2) Nikhil Mehta **(201801030)**
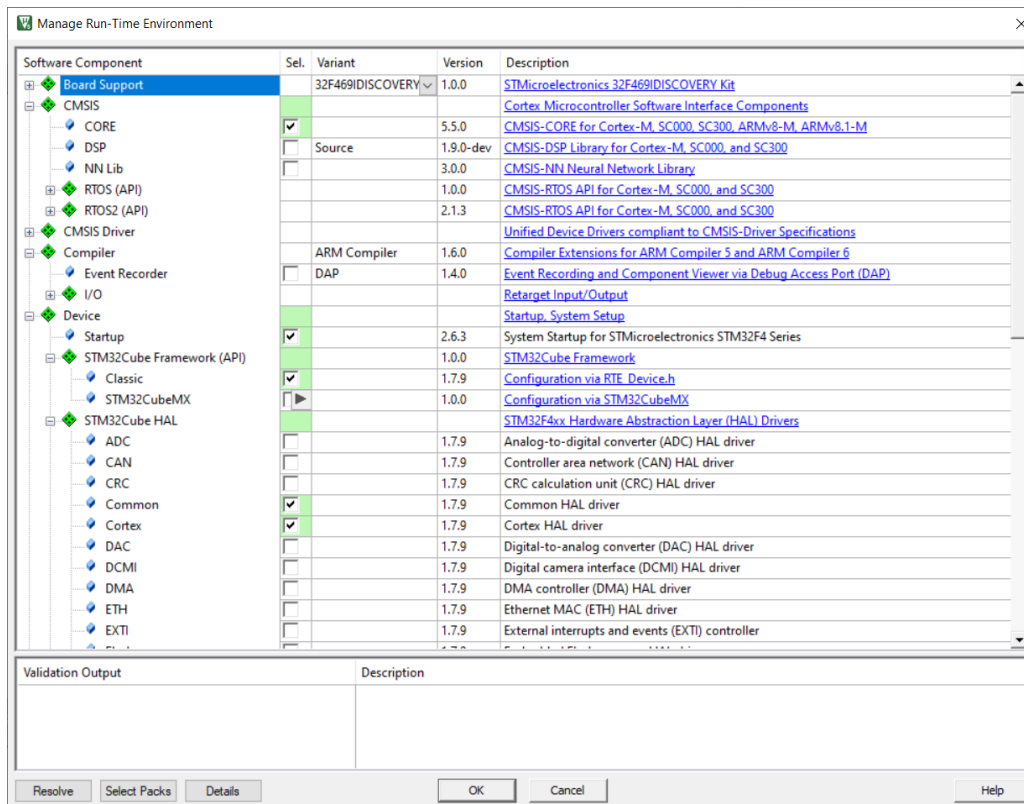   3) Sudhanshu Mishra **(201801114)**

Steps:

Installation of the Software

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

Environment Setup

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

2 Create a new Project and rename it using the appropriate nomenclature.

3 Make a new group within the project.

4 Modify the options for the target device:

4.1 Under the device header, select STM32F407VGTX as the target device.

4.2 Select the Simulator radio button and load the KEIL_STM.ini file as an
initialization file under the Debug header.

5 Under the manager's Run-time environment settings section, make the
following changes:

Software Component	Sel.	Variant	Version	Description
⊞ ◆ Board Support		32F469IDISCOVERY ⌄	1.0.0	STMicroelectronics 32F469IDISCOVERY Kit
⊟ ◆ CMSIS				Cortex Microcontroller Software Interface Components
— ● CORE	☑		5.5.0	CMSIS-CORE for Cortex-M, SC000, SC300, ARMv8-M, ARMv8.1-M
— ● DSP	☐	Source	1.9.0-dev	CMSIS-DSP Library for Cortex-M, SC000, and SC300
— ● NN Lib	☐		3.0.0	CMSIS-NN Neural Network Library
⊞ ◆ RTOS (API)			1.0.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
⊞ ◆ RTOS2 (API)			2.1.3	CMSIS-RTOS API for Cortex-M, SC000, and SC300
⊞ ◆ CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifications
⊟ ◆ Compiler		ARM Compiler	1.6.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
— ● Event Recorder	☐	DAP	1.4.0	Event Recording and Component Viewer via Debug Access Port (DAP)
⊞ ◆ I/O				Retarget Input/Output
⊟ ◆ Device				Startup, System Setup
— ● Startup	☑		2.6.3	System Startup for STMicroelectronics STM32F4 Series
⊟ ◆ STM32Cube Framework (API)			1.0.0	STM32Cube Framework
— ● Classic	☑		1.7.9	Configuration via RTE_Device.h
— ● STM32CubeMX	☐▶		1.0.0	Configuration via STM32CubeMX
⊟ ◆ STM32Cube HAL				STM32F4xx Hardware Abstraction Layer (HAL) Drivers
— ● ADC	☐		1.7.9	Analog-to-digital converter (ADC) HAL driver
— ● CAN	☐		1.7.9	Controller area network (CAN) HAL driver
— ● CRC	☐		1.7.9	CRC calculation unit (CRC) HAL driver
— ● Common	☑		1.7.9	Common HAL driver
— ● Cortex	☑		1.7.9	Cortex HAL driver
— ● DAC	☐		1.7.9	Digital-to-analog converter (DAC) HAL driver
— ● DCMI	☐		1.7.9	Digital camera interface (DCMI) HAL driver
— ● DMA	☐		1.7.9	DMA controller (DMA) HAL driver
— ● ETH	☐		1.7.9	Ethernet MAC (ETH) HAL driver
— ● EXTI	☐		1.7.9	External interrupts and events (EXTI) controller

6 Add the main.c and clock.c files to the project's newly established group, along
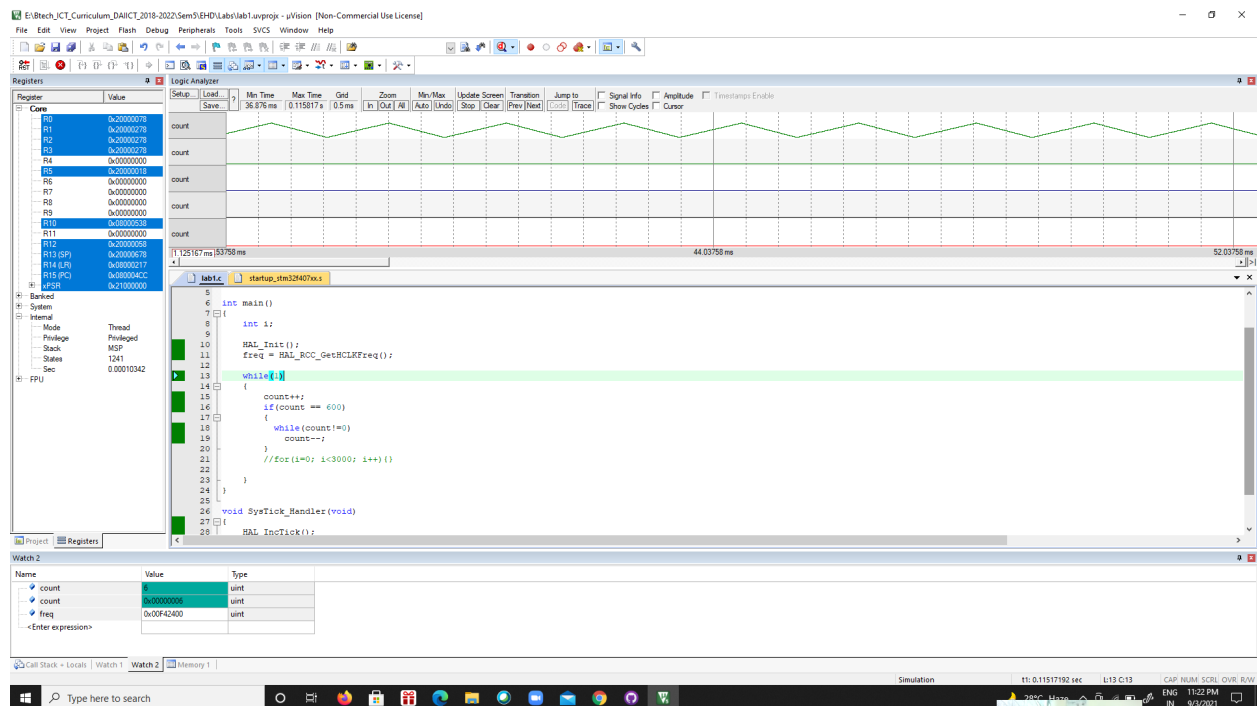with the appropriate header file (#include "stm32f4xx hal.h").

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.

Simulation and Graphs:

count=600

count=1000