**ECEN 5803**

**Mastering Embedded Systems Architecture**

**Project Report**

**Submitted By**

**Swati Kadivar**

**Abijith Ananda Krishnan**

**Guided By**

**Prof. Timothy Scherr**

## 

## 

## 

## 0) Index or Table of Contents

1. Executive Summary 2
2. Problem Statement 2
   1. Objectives 2
3. Approach and methodology of evaluation 3

### Block diagram or Semi-Schematic diagram 3

### Hardware evaluation 3

1. Module test results 4
   1. Module 1 results
   2. Module 2 results
   3. Module 3 results
   4. Module 4 results
2. List of project deliverables 12
3. Recommendations: GO/NoGO 12
4. References 13
5. Project Staffing 13

## 

## 1) Executive Summary

In accordance with a request for Services issued by Keithley Corporation, the suitability of the STM32F401RE microcontroller was evaluated for the design of their Signal Analyzer product. The NUCLEO-F401RE evaluation board is used with analog and digital peripherals to test hardware performance of it. Software performance was tested based on the ability of the MCU to handle interrupt-driven, parallel and processing intensive threads.

Below are the test results found from the proposed project - evaluation of STM32F401RE MCU based product:

* The STM32F401RE MCU interfaced with the serial terminal successfully, interface handling and programming is easy.
* also was found to run at ca. **130 DMIPS**, which agrees with the specified requirements for the Signal Analyzer application.
* Audio frequency synthesis (20 Hz — 20 kHz) was done using STM32F401RE, along with adjustable frequency and amplitude by means of potentiometer.
* Using SPDT button switches, controlled the blinking of LEDs based on wave parameters.
* Measured Real time performance by judging the ability of the STM32F401RE MCU to respond to interrupt-driven threads. Implemented multiple threads and executed them in parallel - concurrently in a cyclic executive manner to serve peripheral interrupts. Without any latency issues, MCU executed all threads very efficiently.
* Provided samples of an audio signal(floating point) with 1004 Hz frequency externally to the STM32F401RE microcontroller. We did harmonic analysis and estimated the input audio frequency with an accuracy of 0.4%.

Based on these test results, the design evaluation team concludes that the STM32F401RE satisfies all the basic required functionalities, plus meeting all of the critical performance criteria listed in the RFS, and hence we give it a **GO** for the Signal Analyzer application.

## 2) Problem Statement

To evaluate ARM Cortex M4 32-bit STM32F401RET6 microcontroller embedded in NUCLEO-F401RE board by means of various modules like ADC, DAC, interfacing LCD - measure temperature using external DS1631 temperature sensor and display it on LCD, interfacing buttons - programming with and without interrupt, developing rtos threads and executing all threads with mutex protection etc.

### **Objectives:**

**Module1:** To learn how to interface and program Nucleo board by means of blinking onboard LED at specific frequency.

**Module2:** To control the LEDs blinking using SPDT buttons - without interrupt implementation and with interrupt based implementation and gauge how STM32F401RE is able to handle interrupt on peripherals.

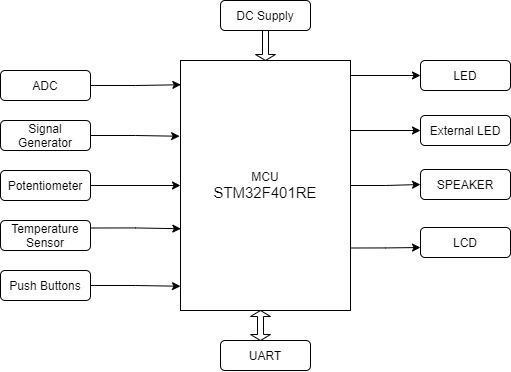
**Module3:** Interface LCD, measure temperature using DS1631 temperature sensor and display it on LCD in a multithreaded environment with mutex protection. To evaluate STM32F401RE for a multithreading environment - rtos implementation.

**Module4:** Evaluate STM32F401RE performance as signal analyzer by means of implementing ADC and DAC - performing harmonic analysis.

## 3) Approach and methodology of evaluation

Tests and data analyses were performed using the NUCLEO-F401RE evaluation board, which provides STMicroelectronics Morpho extension pin headers for full access to all STM32 I/O pins. Figure 1. illustrates the functional block diagram of the NUCLEO-F401RE board. Connectivity to the platform was established via the Mini USB port and a Windows host machine.

### **a) Block diagram or Semi-Schematic diagram**

Integration and testing of peripherals such as SPDT button switches, speakers, potentiometer, LCD and LEDs was accomplished using the Analog, PWM, Digital - input/outputs, and ADC/DAC interfaces. These peripherals were mounted on a bread-board and connected with the evaluation board using the pin headers. Switch/button interfaces were tested using LED on/off and blinking functionalities. For testing audio signal synthesis and modulation, potentiometer was used along with a speaker module. A sawtooth waveform was generated by outputting software generated samples over one of the PWM pins. Potentiometer was used to modulate this audio signal by altering the frequency and the duty-cycle - amplitude, and then results were evaluated accordingly. 

Display peripherals such as the LCD, and the serial terminal of the host machine were tested using the SPI, and USART interfaces, respectively. An incrementing counter, and temperature values read by a temperature sensor were displayed on the LCD with the help of a shift register and the SPI interface. The USART interface was configured to communicate with the host machine to display the same information on the UART terminal.

Evaluated multitasking capabilities of the STM32F401RE MCU using RTOS threads, created for a subset of the tasks detailed above, and then spawned at regular intervals within a cyclical executive.

The MCU was provided with floating-point samples of an audio wave at 1004 Hz, which were then analyzed using a peak-detection algorithm to yield an estimate of the input frequency. This was done to test harmonic analyzer functionality.

Dhrystone benchmarking (VAX DMIPS) was used to quantify the processing performance of the MCU. The mBed online compiler was used to develop and compile

test code, and the Keil MDK5 IDE was the debugging environment employed.

### **b) Hardware evaluation**

Hardware evaluation is done based on whether the required peripherals can be implemented on the given MCU for Keithley 2017 Signal Analyzer product.

**Input Requirements:**

1. **Temperature Sensor:** As there is no temperature sensor on the board, we used an external temperature DS1631 which is a high precision temperature sensor, providing 9 to 12-bit temperature readings over -55℃ to +125℃. To implement DS1631 with the MCU we used SDA and SCL serial communication ports.
2. **Potentiometer:** We used two P103 high precision potentiometers as input to the microcontroller. The potentiometers are used to adjust the volume and frequency of the audio signal at PWM output. These potentiometers are of 10K ohms specification with ±10% tolerance.
3. **Switches:** Four SPDT type button switches were used to toggle LED states.
4. **ADC:** An ADC is provided on the board. It is a 12-bit 16 channel ADC. The output of the signal generator is given as input to ADC to perform further analysis. The output of the temperature sensor is given to the ADC which then using SPI is displayed on the LCD.
5. **DAC:** There is no DAC provided on the board. But MCU has analog input pins to connect to an external DAC. Microchip MCP4921 is used in this system. This is a 12-bit low cost DAC appropriate for this system. DAC is used to create a 1004 Hz tone to perform further analysis. In this analysis, we used a software DAC to generate the 1004 Hz tone.

**Output Requirements:**

1. **LCD:** STM32F401RE has no on-board LCD. We use the NEWHAVEN NHD-0216HZ LCD module instead. This LCD is 2 lines x 16 characters with a required power supply of 3.3V. LCD is interfaced with the processor using a shift register and SPI module.
2. **LED:** The board has an internal LED LD2 which is a GREEN LED. We also use an external green/red colored LED. The external LED is connected to the digital output port on STM32F401RE. The brightness of the LED is controlled using the potentiometer.
3. **PWM:** In STM32F401RE, one 16-bit timer can be used as PWM time for motor control. The PWM output port is used to drive the speaker module. By adjusting the period and pulse width of the PWM output, volume and pitch of the speaker can be adjusted. The brightness of LED is also controlled using a potentiometer and PWM.
4. **Speaker:** An analog speaker directly driven by the MCU AnalogOut pin was used to sense changes made to the frequency and volume of an audio signal.

**Result:** The STM32F401RE MCU can be used to implement all the input and output subsystems given in the requirements in the RFS. The features required are Digital I/O, Analog I/O, ADC, DAC, SPI, UART and PWM output.

### **c) Additional Evaluations:**

We did further evaluation in which **a benchmark test** was done to check the MCU’s capability and a result of **129.7 DMIPS** was obtained which satisfies the requirements mentioned in RFS. (Please find the screenshot in the Appendix)

**BOM:**

|  |  |  |
| --- | --- | --- |
| **Component** | **Cost per Unit** | **Quantity** |
| STM32F401RE | $4.22(if ordering 1000 units) | 1 |
| LCD NHD0216HZ | $5.40 | 1 |
| DS1631 Temp Sensor | $2.2 | 1 |
| Switches | $0.4 | 4 |
| Potentiometer P103 | $1 | 2 |
| DAC Microchip MCP4921 | $2.58 | 1 |
| Speaker | $1.1 | 1 |
| LED | $0.2 | 1 |
|  | TOTAL=$19.3 |  |

**Our total cost was 19.3$ - this satisfies the requirement of <$20. Our MCU cost is $4.22 which was available at $2.98 earlier. RFS is a bit older and now MCU is not available at <$4.**

## **4) Module test results**

### **4.1) Module 1 results:**

2. Where (at what address) does the Reset handler begin in the memory map?

Answer: **The reset handler begins at 0x08000304 -** please refer to Appendix I for a screenshot.

3.How much memory is used by the code (Led blinking code for the homework)?

Answer: **22kB flash memory is used by the code.** - please refer to Appendix II for a screenshot.



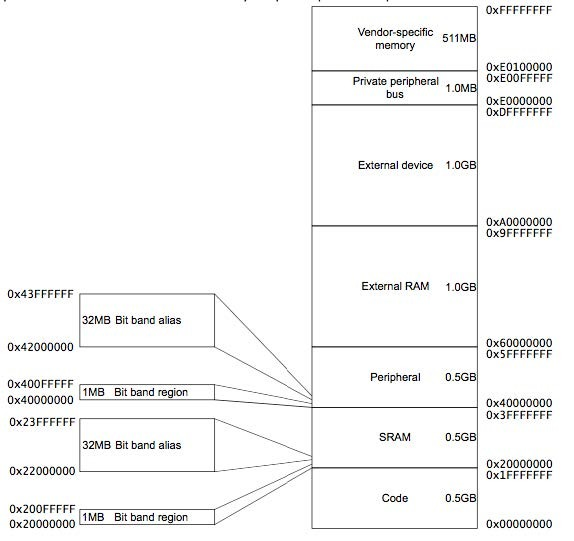
4. Run the mBed Nucleo Example (display\_time). Set the time to the current time, and combine this with your mBed Nucleo Example (printf) to print the current time to a terminal window on your PC. Capture a screenshot of the terminal window. How much memory is used by this code?

Answer: **15.2 kB flash memory is used by the code. -** Please refer to Appendix III for a screenshot.

**Output screenshot:**

5.Explain the memory model of ARM Cortex-M4 with respect to the code memory, data memory, IRQ handlers and peripherals. Explain with the help of a diagram where required.

**Answer:** The below diagram is the memory model of the ARM Cortex M4. It has **4GB** of addressable memory.



* As shown in the figure above the Cortex-M4 has a predefined memory map, and the memory can be accessed by simple memory instructions by other built in peripherals.
* The Cortex-M4 has a fixed 4GB memory map, out of which 0.5GB is code memory which can be used to store program code,this region can also be used for data memory.
* The SRAM is 0.5GB which is used to store data which includes the stack. It can also be used to store program memory
* The peripheral memory is another 0.5GB and can be used to store peripherals and data memory.
* The RAM is of 1GB and can be used as a continuous memory block to store data.
* The Internal Private peripheral bus memory space is 1MB in size and is allocated to store NVIC and other debug components.

6. As a separate project, run either the Dhrystone or the Whetstone benchmark program on your target processor using the code provided, which may need to be modified. If running the Dhrystone, calculate the number of VAX DMIPS.

Answer: We ran Dhrystone benchmark program on the target MCU and found **129.7DMIPS** as output. Please refer to the Appendix IV for a screenshot.

### **Module 2 results:**

**Q1 . Try to issue an interrupt on different signal edges (rising edge or falling edge). What changes?**

Circuit diagram**:**

* The below circuit diagram illustrates connections between the NUCLEOF401RE evaluation board and peripherals including SPDT button switches and LEDs.
* When the interrupt is triggered on the falling edge, the LED is toggled as soon as the button is pressed. On the other hand, When the interrupt is triggered on the rising edge, the LED is toggled once the button is released after it is pressed.
* The amount by which the variable ‘i’ is incremented, the period of the resulting waveform changes, thereby causing a difference in the frequency (pitch) of the audio signal.Diagram, schematic

  Description automatically generated
* Variable i is used to determine the duty cycle of PWM, thus determining the volume of the speaker. That is, when we increase the value of i, the volume of the speaker would be turned up. On the other hand, when we decrease the value of i, the volume of the speaker would be turned down.

**Video Link:** [**https://drive.google.com/file/d/1-JXUxWAlfPy2usv6ye-AZaJEFzUhx\_PW/view?usp=sharing**](https://drive.google.com/file/d/1-JXUxWAlfPy2usv6ye-AZaJEFzUhx_PW/view?usp=sharing)

### **Module 3 results:**

We created 4 Threads to:-

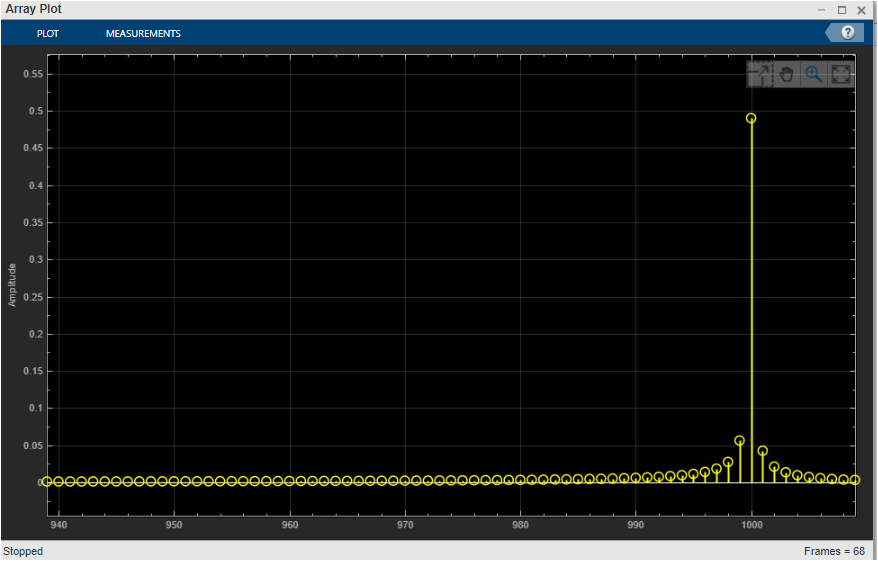
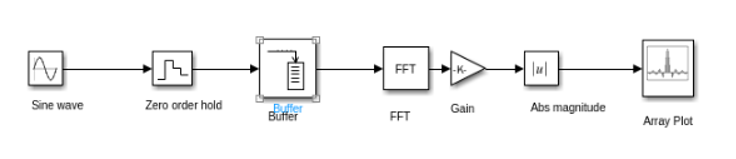
* Display the temperature on the LCD
* Adjust the brightness of the external LED using a potentiometer
* Display an incrementing counter on the LCD.
* Blink an LED

These threads were created using rtos mbed-os library. The threads used for incrementing the counter on the LCD and displaying temperature on LCD required a mutex lock as they are writing to the same shared resource.The mutex lock prevents both the threads to use the resource at the same time, i.e when one thread is writing on the LCD the other thread is waiting for the mutex to be unlocked.

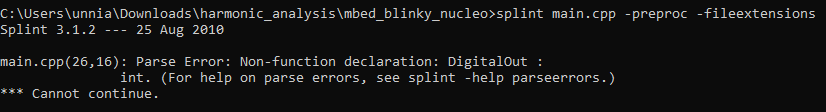
**Video Link:** [**https://drive.google.com/file/d/1-orwDTHySH3WLe6NVP7vlrD2F7z3-sU6/view?usp=sharing**](https://drive.google.com/file/d/1-orwDTHySH3WLe6NVP7vlrD2F7z3-sU6/view?usp=sharing)

### **Module 4 results:**

We evaluated STM32F401RE MCU in Module 4 with respect to how it functions as a signal analyzer and to estimate the frequency of an analog input signal. We saved samples of an audio signal in floating point number format. The audio signal was of 1004 Hz frequency, sampled with a 10 us resolution. We used an autocorrelation based peak detection algorithm to estimate the input signal frequency, which was found to be 1000 Hz. Same is illustrated with the Simulink model as shown in the figure below.

We observed that the error in frequency estimation is just 0.4% with respect to the sampled 1004 Hz signal. But still this error is much larger than the 5 ppm as listed in the RFS. The frequency is computed using the inverse of time difference between peaks in the autocorrelation function and so the error persists. There is a lower limit on the achievable frequency resolution because of limitation in the autocorrelation function in time resolution by the sampling period of 10us. At last, We added a LED program to blink it at a rate proportional to the estimated frequency of 1000 Hz.

**2.Run Splint or an equivalent code checker (like CppCheck or Cpplint) on your code in 1 of this module. Resolve any errors, explain the warnings.**



The above error is a parse error. DigitalOut is a C++ class. myled is declared as an object of DigitalOut using pin LED. “LED2” has been defined previously which represents the specific pin where LED2 is located.

We referred to split FAQs and found that this error is a limitation of current splint release and will be fixed in future releases.

**4. Estimate the processor load in % of CPU cycles.**

With a measured elapsed time of approximately 191 ms the CPU utilization is 19.1% per second of CPU utilization.

## 5) List of project deliverables

Below are the files attached in the submitted zip.

|  |  |
| --- | --- |
| Detailed Technical Report | Executive Summary Report |
| Bill of Materials(BOM) | Module 1-4 design files |
| Dhrystone Benchmark test run result | Simulink data file |
| Doxygen output files for all the Modules |  |

## 6) Recommendations: GO/NoGO

Given the requirements and budget for the project by Keithley Corporation this STM32F401RE is **strongly recommended-GO** **as a signal analyzer product**. Though the product doesn’t have inbuilt DAC and temperature sensor, the cost of external components together doesn’t exceed the total budget of $20. While testing the ability to interface the LEDs and Switches, the MCU has good response for interrupt driven code and real time functionalities.The controller can also be used to generate PWM in range of audio signal. Harmonic analysis capabilities were evaluated by implementing the peak detection algorithm.Based on these test results we can give the STM32F401RE a go , and can confirm it is capable of working as a signal analyzer.

Further analysis was done and budget constraints are also reached as the entire project budget would be $17 which is less than $20. The software development platform was mbed online compiler which was easy to code in and available as open source. The required DMIPS was 100 and after running the dhrystone benchmark, we found out it was 130 DMIPS which satisfies the requirement.

## 7) References

* STM32F401RE Reference Manual
* STM32F401RE Datasheet
* STM32 Nucleo-64 user manual
* STM32 HAL library user guide
* <https://os.mbed.com/handbook/Homepage>
* <https://www.sciencedirect.com/topics/engineering/memory-space>
* <https://www.mathworks.com/matlabcentral/answers/780027-vector-scope-simulink-r2020b>
* <https://www.mathworks.com/help/dsp/ug/transform-time-domain-data-into-the-frequency-domain-using-the-fft-block.html>
* <https://www.youtube.com/watch?v=HxErwlrDIms&ab_channel=ENDOChannel>

## 8) Project Staffing

**Instructor:**

**Prof. Timothy Scherr**

Teaching Associate Professor

EMBEDDED SYSTEMS ENGINEERING

University of Colorado Boulder

Email: [Timothy.Scherr@colorado.edu](mailto:Timothy.Scherr@colorado.edu)

**Students:**

**Swati Kadivar**

Graduate Student - University of Colorado Boulder

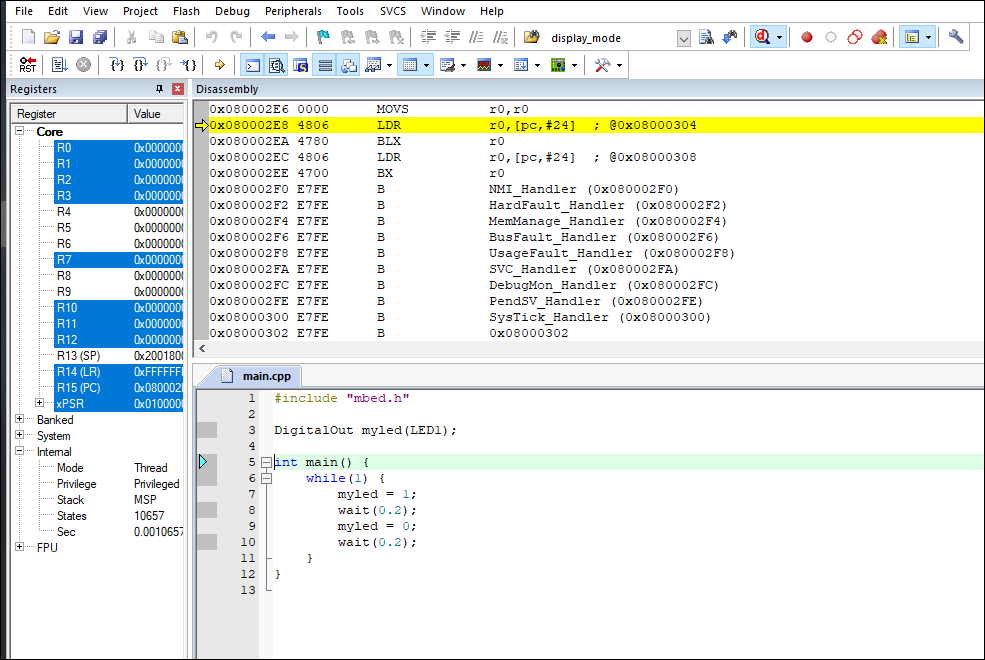
Email: [swati.kadivar@colorado.edu](mailto:swati.kadivar@colorado.edu)

**Abijith Ananda Krishnan**

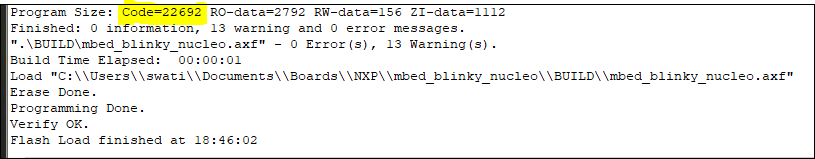
Graduate Student - University of Colorado Boulder

Email: [abaa3023@colorado.edu](mailto:abaa3023@colorado.edu)

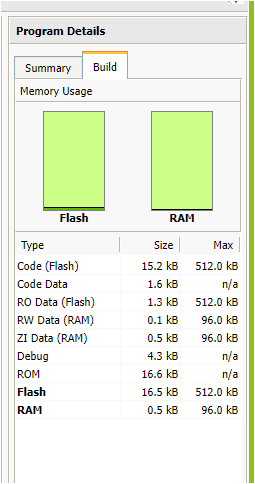
**Appendix I:**

****

**Appendix II:**

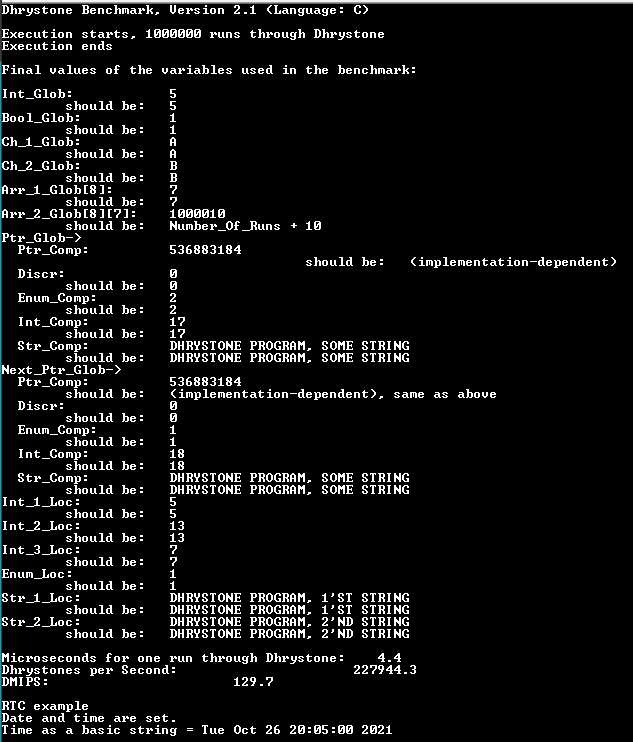


**Appendix III:**



**Appendix IV:**

**Evaluation of STM32F401RE with Dhrystone:**

****