School of Science Engineering and Technology

**EEET-2481: Embedded System Design and Implementation**

**Lecturer:** Thanh Pham

1. Name: Huynh Ngoc Tai Student ID: s3978680

2. Name: Tran Quang Minh Student ID: s3988776

**Group: 12**

**Submission Due Date: 15/01/2025**

"I declare that in submitting all work for this assessment I have read, understood and agree to the content and expectations of the [Assessment declaration](https://www.rmit.edu.au/students/my-course/assessment-results/assessment)

**TABLE OF CONTENTS**

**Topic Page**

Table of Contents 2

**I. Introduction 3**

**II. Assessment Problem 3**

**III. Testing 4**

**IV. Group Contribution 9**

**V. Conclusion 9**

**I. Introduction**

This project report documents our work in embedded system design and implementation course in RMIT using the Keil MDK platform. The exercises completed during this project provided hands-on experience with key embedded programming concepts, including UART communication, timer interrupts.

Through these tasks, we explored the integration of software and hardware in real-world microcontroller-based systems, strengthening our understanding of embedded system design.

**II. Assessment Problem**

The assessment focuses on the design and implementation of two embedded system applications on the NUC140 microcontroller using Keil MDK.

Exercise 1 – UART Communication:

Develop a program to establish bidirectional data communication between the NUC140 board and a laptop using UART0. The program configures UART settings such as clock frequency (22.1184 MHz) and baud rate (115200 bps) to send and receive a message.

Then the system verifies successful communication by echoing the transmitted message back to the laptop.

Exercise 2 – Stopwatch System:

Design and implement a stopwatch system with precise time measurement using Timer0 interrupts. The system features multiple operational modes (Idle, Count, Pause, Check) controlled via push buttons (K1, K9, K5) on the board.

Moreover, we will use LEDs and a 7-segment display for real-time feedback, allows for recording and displaying up to 5 lap times, and includes functionalities for pausing, resetting, and reviewing recorded laps.

**III. Testing**

Exercise 1 – UART Communication:

In the UART project, after coding in Keil, we launched the Terminal application, uploaded the program to the NUC140, and connected it to the appropriate port. Once we sent the message, the board successfully received and echoed it back, displaying the output on our screen.

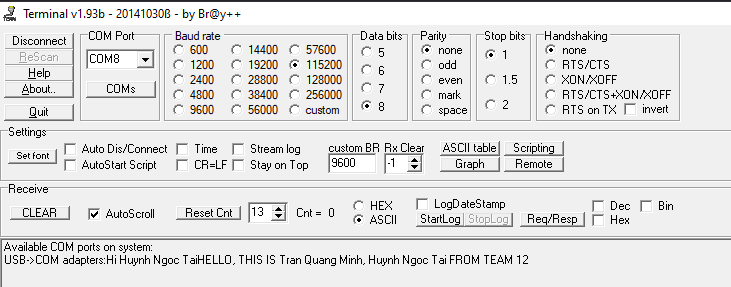


Figure 1.1: The output that we got after sending it

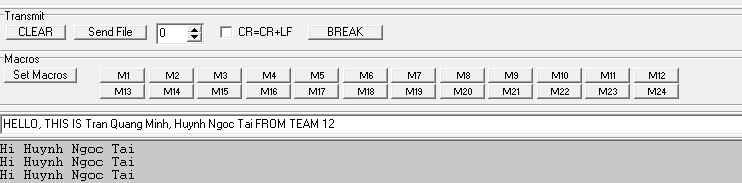
Figure 1.2: The input that we sent

Figure 1.3. The board and the UART that we plugged in



Figure 1.4. The board with the laptop that has our result

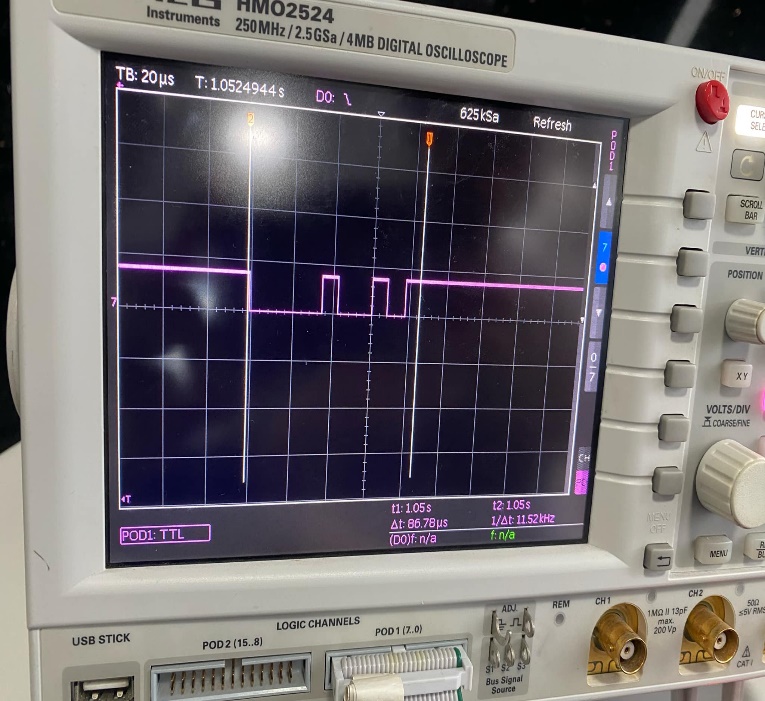
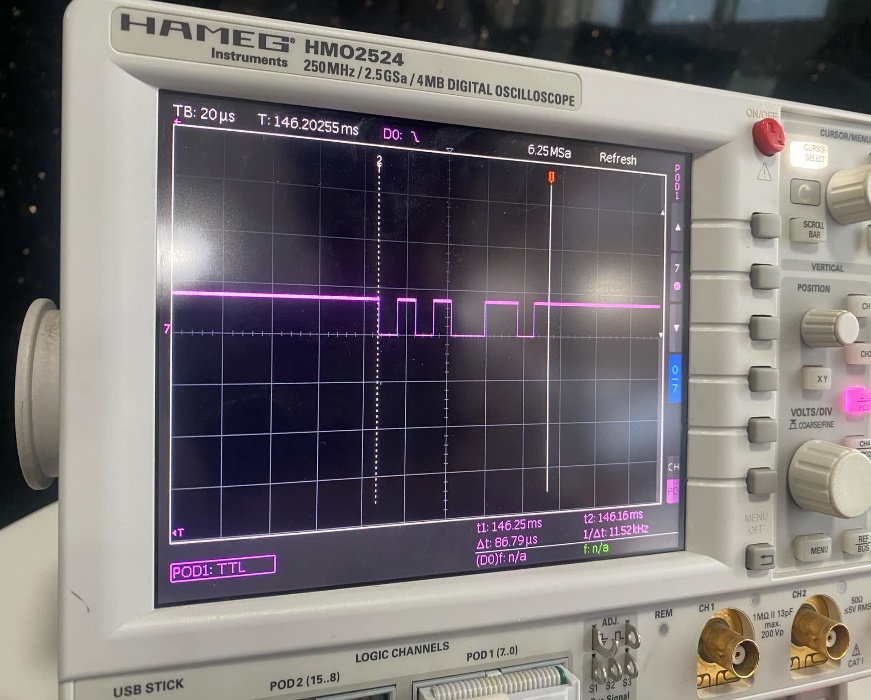


Figure 1.5 – 1.6: Waveform of the word H (01001000) and E (01100101)

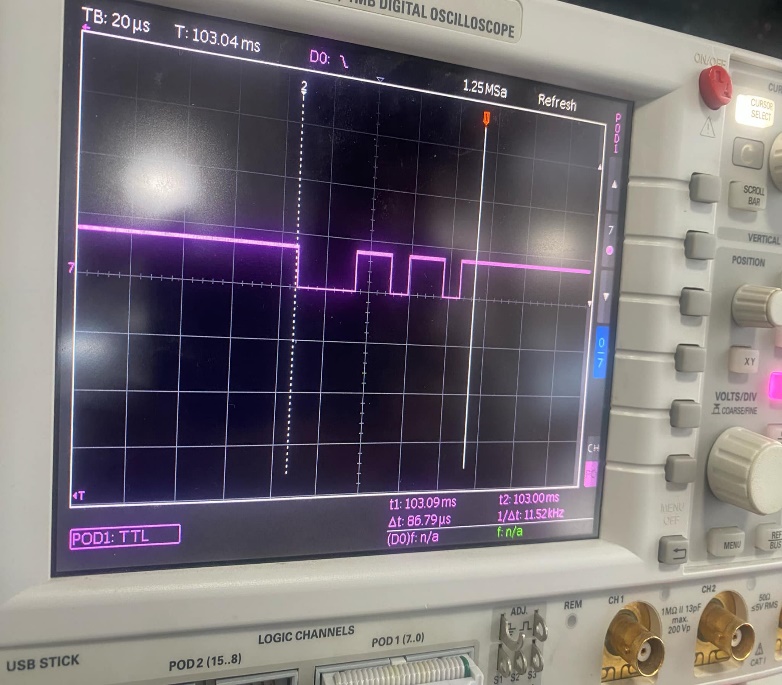
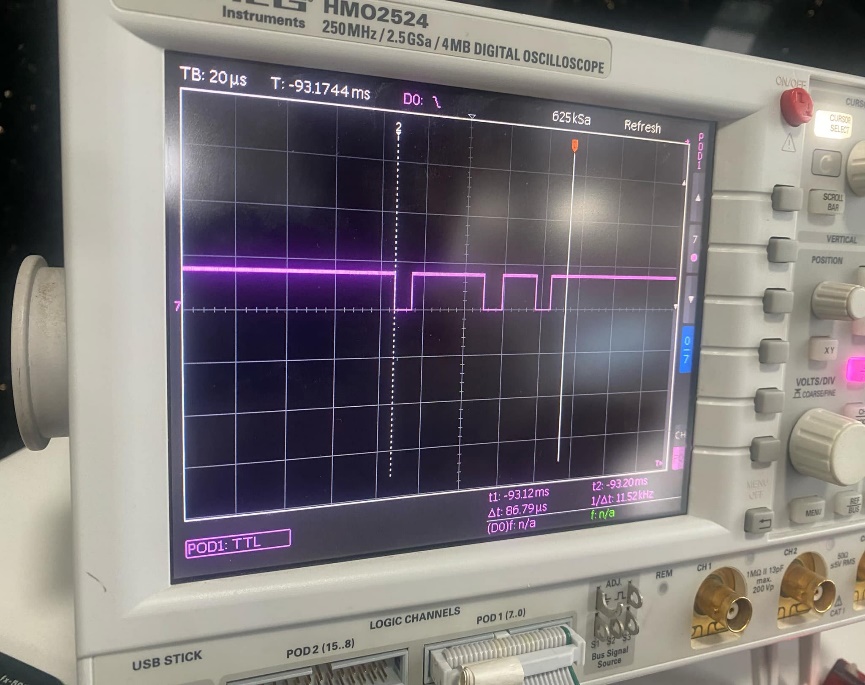
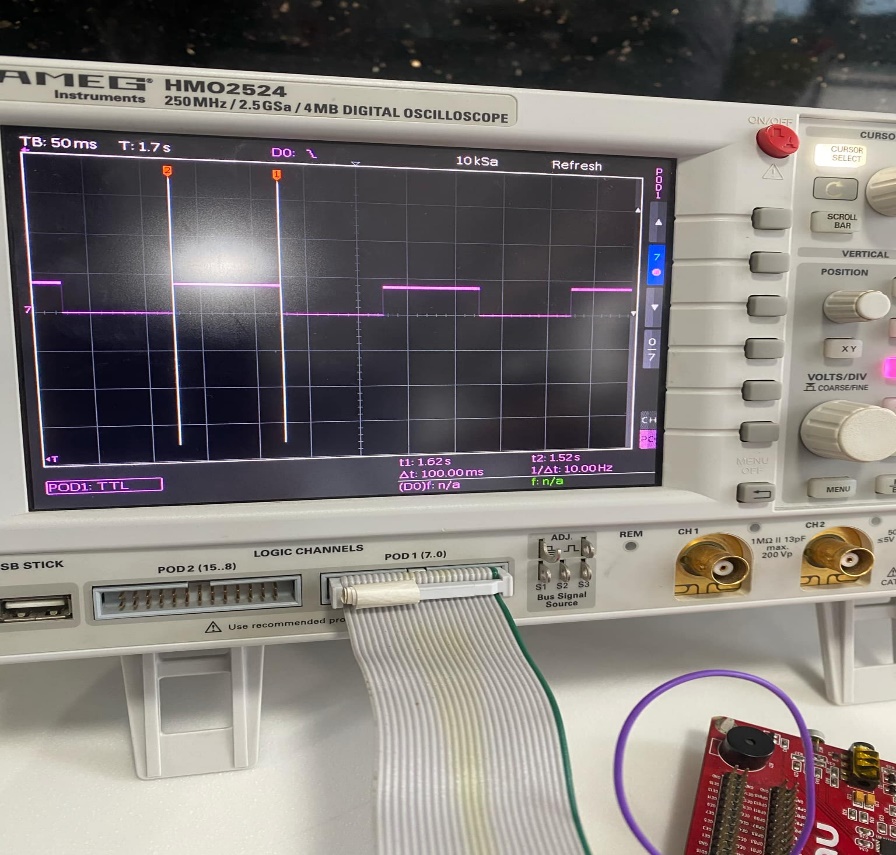


Figure 1.7 – 1.8: Waveform of the word L (01101100) and O (01101111)

Exercise 2 – Stopwatch System:

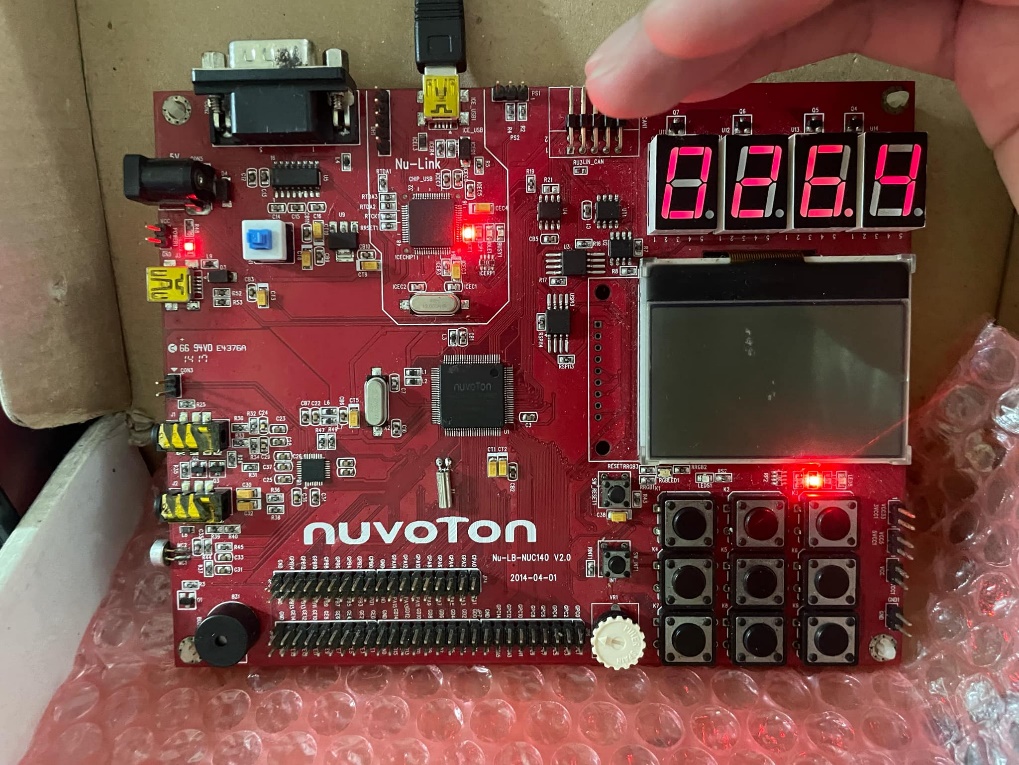
Figure 2.1 Timer Count Waveform that fluctuate whenever a 1/10 second is reached.

Figure 2.2 NUC140 with timewatch code embedded and is counting

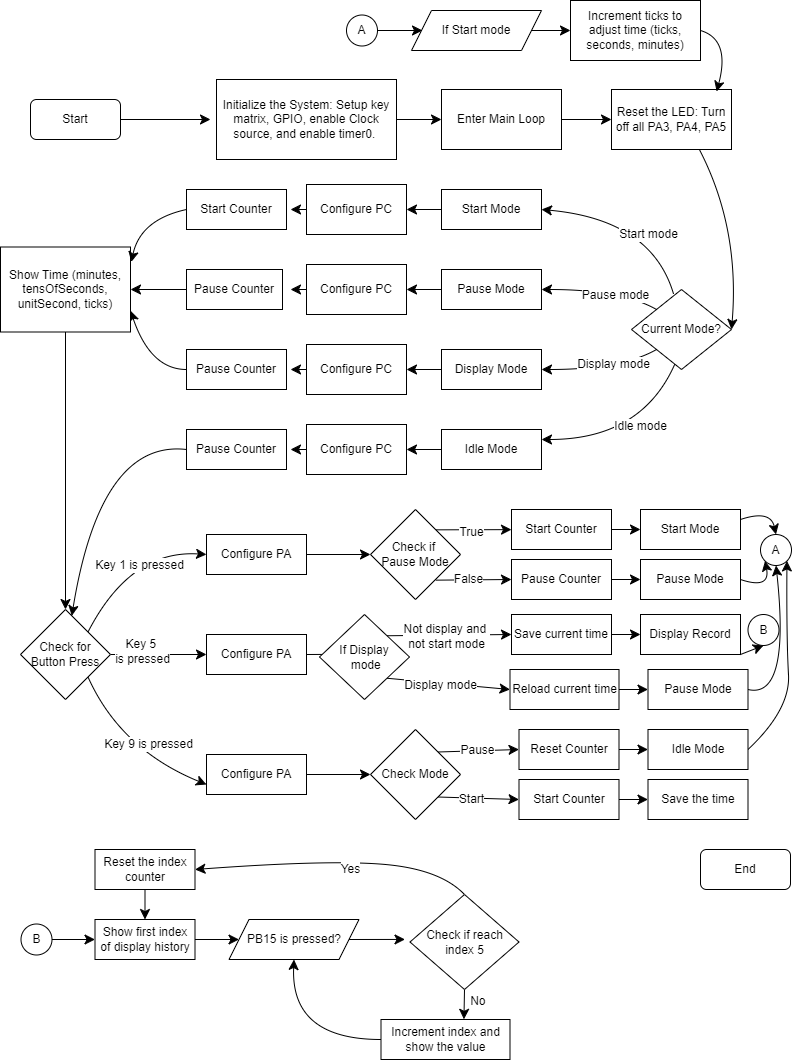
In our stopwatch system, after finishing the coding part, we loaded our code into the NUC140 and it satisfied pretty much every requirements given (operates in multiple modes, pressing key to start/ pause/ reset the timer, also the program can record the timelap and display it back to the user through LED segment)

Figure 2.3 Flowchart of the program

**IV. Group Contribution**

|  |  |  |
| --- | --- | --- |
|  | Huynh Ngoc Tai | Tran Quang Minh |
| Code Part 1 |  | 50% |
| Code Part 2 |  |  |
| Total Contribution |  |  |

**VI. Conclusion**

This assessment allowed us to explore key concepts in embedded system design using the NUC140 microcontroller. By working on UART communication and real-time timer-based functionality, we gained practical experience in configuring hardware, writing embedded code, and troubleshooting system behavior. While most aspects of the implementation, were successful, there were minor areas for improvement in system responsiveness and handling edge cases.

Despite these challenges, the project strengthened our understanding of core embedded programming concepts, including peripheral configuration, interrupt handling, and real-time event management. This experience has enhanced our problem-solving skills and prepared us to take on more complex embedded system design tasks in the future.