TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

**KHOA KỸ THUẬT MÁY TÍNH**

--------------------------------



**BÁO CÁO THỰC TẬP**

**CÔNG TY TNHH PHẦN MỀM FPT**

**BÁO CÁO THỰC TẬP DOANH NGHIỆP**

Nhóm SV thực hiện:

|  |  |
| --- | --- |
| Họ và tên | Mssv |
| Trần Thanh Duy | 16520308 |
| Phạm Kim Thành | 16521129 |
| Lương Quốc Hải | 16520327 |
| Nguyễn Tấn Phát | 16521619 |
| Lã Tuấn Vinh | 16521439 |

Lớp: CE501.K11.MTCL

GVHD: Trần Đại Dương

*Thành phố Hồ Chí Minh – Năm 2019*

# **Lời cảm ơn**

Để đợt thực tập này đạt kết quả tốt đẹp, chúng em đã nhận được sự hỗ trợ, giúp đỡ của nhiều tổ chức, cá nhân. Với tình cảm sâu sắc, chân thành, cho phép em được bài tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện giúp đỡ em trong quá trình thực tập này.

Trước hết em xin gởi tới các thầy cô khoa Kỹ Thuật Máy Tính của trường Đại học Công Nghệ Thông Tin lời chào trân trọng, lời chúc sức khỏe và cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình đã cung cấp cho chúng em nền tảng, kiến thức để em có thể hoàn thành công việc được giao trong đợt thực tập một cách tốt nhất.

Xin gởi lời cám ơn tới các anh chị thuộc đơn vị FGA đã hỗ trợ, giúp đỡ, cải thiện những thiếu sót của chúng em trong suốt quá trình thực tập qua.

Và cuối cùng, em xin gởi lời cảm ơn tới công ty FPT Software đã phối hợp với khoa Kỹ Thuật Máy Tính mở phòng LAB Nghiên cứu và phát triển Automotive ngay tại trường Đại học Công Nghệ Thông Tin để có một môi trường nghiên cứu và thực tập thuận tiện cho cả việc học tập của chúng em.

Vì thời gian có hạn cũng như kinh nghiệm còn thiếu nên trong quá trình thực tập có thể không tránh khỏi những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến từ thầy cô để có thể hoàn thiện bản thân mình hơn.

Một lần nửa em xin chân thành cảm ơn!

Tp. Hồ Chí Minh, 27/12/2019

**Mục Lục**

[**Lời cảm ơn** 2](#_Toc28411842)

[**Mục Lục** 3](#_Toc28411843)

[**1.** **Lịch làm việc tại phòng Lab Automotive của FSoft.** 4](#_Toc28411844)

[**2.** **Thông tin về Công ty TNHH phần mềm FPT.** 4](#_Toc28411845)

[**3.** **Nội dung và nhiệm vụ chính.** 6](#_Toc28411846)

[**3.1.** **Sơ lược về nội dung** 6](#_Toc28411847)

[**3.2.** **Nhiệm vụ chính:** 8](#_Toc28411848)

[**a.** **Thực hành trên board STM8L để nắm rõ cách hoạt động, lập trình cho vi điều khiển:** 8](#_Toc28411849)

[**b.** **Tìm hiểu về AutoSar và viết driver cho các module liên quan:** 10](#_Toc28411850)

[**4.** **Nội dung nhiệm vụ của từng thành viên.** 19](#_Toc28411851)

[**a.** **Trần Thanh Duy** 20](#_Toc28411852)

[**b.** **Lương Quốc Hải** 20](#_Toc28411853)

[**c.** **Lã Tuấn Vinh** 21](#_Toc28411854)

[**d.** **Phạm Kim Thành** 22](#_Toc28411855)

[**e.** **Nguyễn Tấn Phát** 23](#_Toc28411856)

[**5.** **Kết quả đạt được qua đợt thực tập.** 23](#_Toc28411857)

[**a.** **Trần Thanh Duy** 23](#_Toc28411858)

[**b.** **Phạm Kim Thành** 24](#_Toc28411859)

[**c.** **Nguyễn Tấn Phát** 25](#_Toc28411860)

[**d.** **Lương Quốc Hải** 26](#_Toc28411861)

[**e.** **Lã Tuấn Vinh** 26](#_Toc28411862)

[**6.** **Nhận xét, góp ý về Chương trình Đào tạo của Khoa Kỹ Thuật Máy Tính.** 26](#_Toc28411863)

1. **Lịch làm việc tại phòng Lab Automotive của FSoft.**

Thời gian thực tập từ 6/6/2019 đến ngày 5/9/2019.

Thời gian làm việc từ 8h30 đến 16h30, từ thứ 2 đến thứ 6.

1. **Thông tin về Công ty TNHH phần mềm FPT.**



*Hình 1: Cận cảnh trụ sở FPT Software tại TP HCM. Nguồn : Chungta.vn*

FPT Software thành lập năm 1999, là công ty thành viên của FPT, Tập đoàn công nghệ hàng đầu của Việt Nam. Sau 17 năm thành lập, FPT Software hiện đang là công ty phần mềm lớn nhất của Việt Nam và đứng Top 100 nhà cung cấp dịch vụ Outsourcing toàn cầu do International Asociation of Outsourcing Professionals (IAOP) đánh giá.

FPT Software có nhiều văn phòng và chi nhánh trực thuộc trong khắp cả nước và cả trên thế giới bao gồm: 6 chi nhánh ở nước ngoài là Nhật Bản, Pháp, Mỹ, Úc, Singapore, Malaysia, 2 văn phòng đại diện tại Thái Lan và Philippines. Fsoft có hơn 3500 kỹ sư phần mềm và các nhân viên hoạt động chuyên sâu trong các lĩnh vực: gia công và kiểm thử phần mềm. Trên 90% khách hàng của FPT Software là các công ty nước ngoài, trong đó có nhiều tập đoàn lớn như Hitachi, Canon, Pepsi, IBM, Panasonic… Năm 2006, FPT Software là công ty đầu tiên của Việt Nam đạt chứng chỉ CMMI5, chứng chỉ cao nhất về độ trưởng thành trong sản xuất phần mềm. Về danh thu, năm 2009 công ty đạt mức 750 tỷ VND, năm 2010 ước tính doanh thu tăng trưởng 34%, đạt mức trên 1000 tỷ VNĐ.

Là công ty chuyên xuất khẩu dịch vụ phần mềm, FPT Software cung cấp các dịch vụ phát triển phần mềm và bảo trì, triển khai ERP, QA, chuyển đổi ứng dụng, hệ thống nhúng, điện toán di động, điện toán đám mây… trong nhiều lĩnh vực như: Tài chính ngân hàng, Viễn thông, Y tế, Chế tạo, Công nghiệp xe hơi, Dịch vụ công.... Hiện tại Fsoft đang nghiên cứu và phát triển dịch vụ CNTT dựa trên những nền tảng công nghệ mới như IoT, S.M.A.C, cho các lĩnh vực sản xuất máy bay, sản xuất ô tô, ngân hàng, truyền hình vệ tinh, viễn thông trên phạm vi toàn cầu. Công ty hiện đã và đang cung cấp dịch vụ cho khoảng 450 khách hàng là các tập đoàn lớn trên thế giới, trong đó có 43 khách hàng năm trong danh sách Forturn Global 500.



*Hình 2: Phòng Nghiên cứu và phát triển Automovie. Nguồn: vnuhcm.edu.vn*

FPT Software cũng thành lập thêm nhiều Trung tâm Phần mềm chiến lược để nhằm đáp ứng mục tiêu tăng trưởng và phát triển của công ty. Đặc biệt là vào tháng 6/2016, Fsoft đã hợp tác cùng trường Đại học Công Nghệ Thông Tin mở Phòng Nghiên cứu và phát triển Automovie ngay tại tòa nhà E của trường để sinh viên có thể tiện lợi cho việc nghiên cứu và học tập.

Website của FPT Software: https://www.fpt-software.com/

1. **Nội dung và nhiệm vụ chính.**
   1. **Sơ lược về nội dung**

Khóa thực tập của nhóm chúng em kéo dài trong vòng 3 tháng. Với mục tiêu chính là tìm hiểu và thực hành căn bản về mảng lập trình nhúng sau đó áp dụng vào để tìm hiểu về AUTOSAR (Automotive Open System Architecture).

Để đạt được mục tiêu đó thì chương trình đào tạo của đơn vị FGA Ebedded Tranining đã chia nội dung thành 4 phần chính với mỗi phần bao gồm các lession như sau:

**Section 1: General & Tool.**

***Lession 1:***

Introduce about FGA.AIS

Introduction about Embedded trainning course

***Lession 2:***

Source code management SVN and GIT

Practice about SVN/GIT

***Lession 3:***

Embedded Overview

***Self study:***

Embedded Overview and Environment

**Section 2: Embedded Trainning Basic Embedded.**

***Lession 1:***

STM8L Board Overview

***Lession 2:***

I/O interface: General-Purpose Input/Output (GPIO)

***Lession 3:***

Peripherals Clock distribution & Multipurpose Clock Generator

***Lession 4:***

I/O Interface, Interrupts

***Lession 5:***

Communication Stack: SPI, I2C, UART

***Lession 6:***

Timer

***Lession 7:***

Linker: Ethernet, USB

***Lession 8:***

Realtime Operating System

***Solf Skill:***

Apply management method in task control.  
Task schedule & adjust effort in working

***Mock Project:***

I/O Interrupt, UART.

**Section 3: Embedded Trainning Special AutoSar.**

***Lession 1:*** Autosar Overview

***Lession 2:*** Microcontroler Overview: MCU, WDT, GPT

***Lession 3:*** I/O Overview: DIO,PORT, PWM

***Lession 4:*** ADC

***Lession 5:***

Memory driver Overview: Flash

Communication Modules: CAN, LIN

Communication: SPI

Communication: ETH

***Mock Project:***

Summary and Assigment Project

***Solf Skill:***

Estimation and planning  
Traceability  
Impact analysis  
Review guideline

**Section 4: Testing in Embedded.**

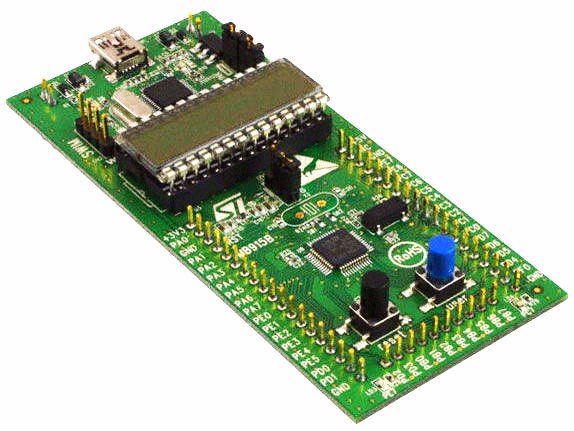
Testing in Embedded

Practice Unit Test on source code

MOCK

* 1. **Nhiệm vụ chính:**

1. **Thực hành trên board STM8L để nắm rõ cách hoạt động, lập trình cho vi điều khiển:**



*Hình 3: Board STM8L*

STM8L là dòng vi điều kheiern 8bits của STMicroelectronics ở phân khúc giá rẻ. Hiện tại được sử dụng rất rộng rãi trong nhiều dự án vì tính tiện dụng của nó.

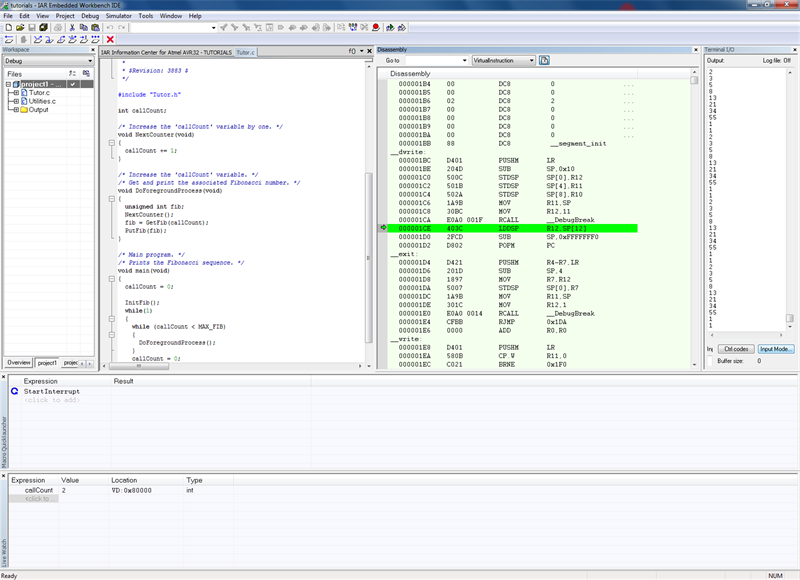
Kit đã được tích hợp đầy đủ mạch nạp, Debug, ra chân điều khiển, cảm ứng điện dung.

**Các tính năng chính:**

* Bộ vi điều khiển STM8L152C6T6, 32 KB Flash, RAM 2 KB, 1 KB EEPROM trong 48-pin LQFP
* ST-Link trên bo mạch với công tắc chế độ chọn để sử dụng bộ làm ST-Link độc lập (với đầu nối SWIM để lập trình và gỡ lỗi)
* Hai đèn LED màu đỏ; LD1 cho giao tiếp USB, LD2 cho 3.3 V bật nguồn.
* Được thiết kế để được cấp nguồn bằng USB hoặc nguồn cung cấp bên ngoài 5 V hoặc 3,3 V
* Có thể cung cấp ứng dụng đích với 5 V và 3 V
* Hai đèn LED của người dùng, LD3 và LD4 (xanh lá cây và xanh dương)
* Hai nút nhấn (User và Reset)
* Tôi DD  đo lường hiện tại
* LCD 28-pin DIP (24 phân đoạn, 4 commons)
* Đầu đề mở rộng cho tất cả các QFP48 I / Os cho kết nối nhanh chóng để tạo mẫu bảng để dễ dàng thăm dò

**Công cụ lập trình, Nạp code cho chip STM8L:**

IAR Systems: Được dịch từ tiếng Anh-IAR Systems là một công ty phần mềm máy tính của Thụy Điển cung cấp các công cụ phát triển cho các hệ thống nhúng. IAR Systems được thành lập vào năm 1983 và được liệt kê trên NASDAQ OMX ở Stockholm. "IAR" là tên viết tắt của Ingenjörsfirman Anders Rundgren, có nghĩa là Công ty Kỹ thuật Anders Rundgren.



*Hình 4: Giao diện lập trình của IAR*

**Tìm hiểu về dòng chip STM8L151:**

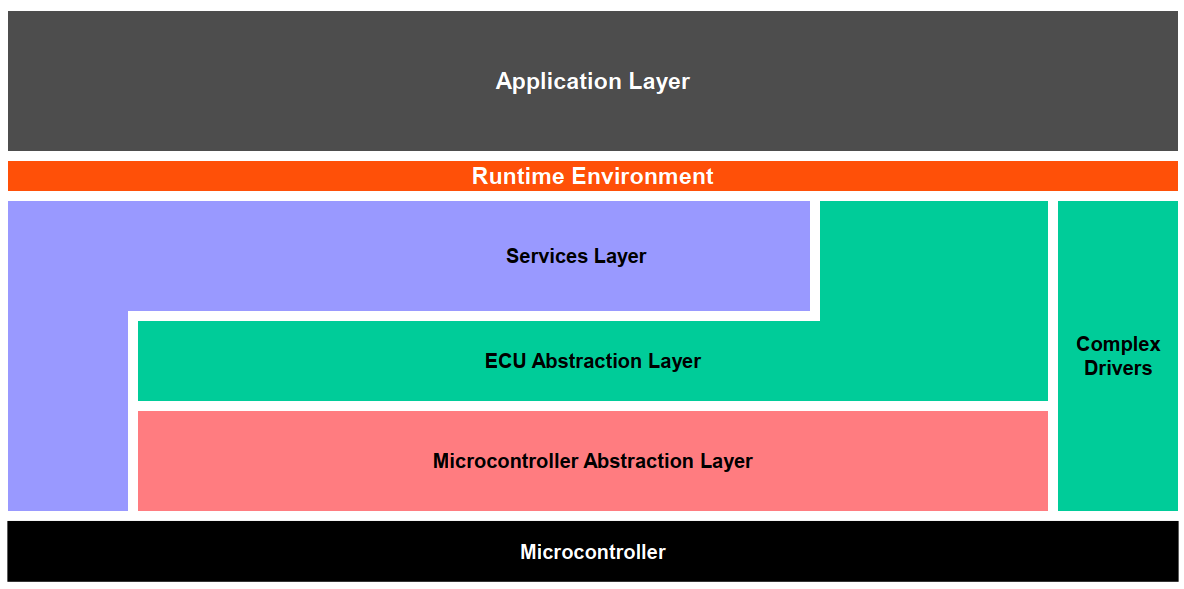
Là dòng vi điều khiển tiết kiệm năng lượng, thích hợp cho các dứng dụng dùng PIN, IoT..

Thông số cơ bản:

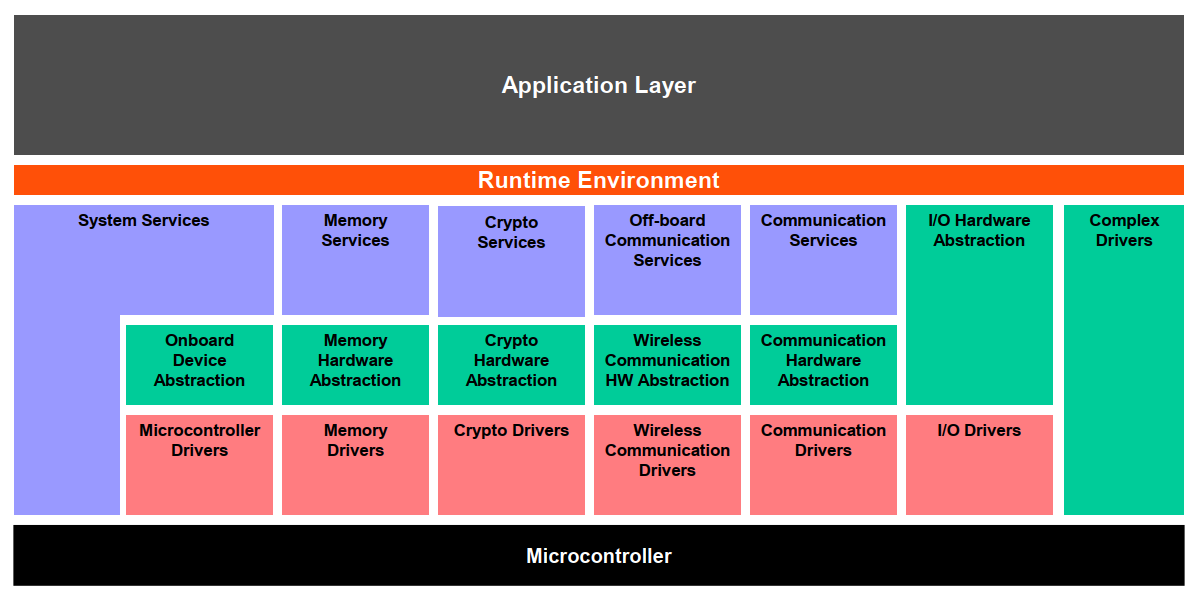
* Sử dụng kiến trúc Harvard
* Tần số tối đa 16MHz, 16 CISC MIPS peak
* Hỗ trợ đến 40 nguồn ngắt ngoài
* Xung clock hỗ trợ từ 1 đến 16MHz
* 32Hz xung thạch anh nội
* Hỗ trợ 2 thanh ghi 16-bit timers với 2 kênh
* Hỗ trợ các chuẩn kết nối như SPI, I2C, UART

1. **Tìm hiểu về AutoSar và viết driver cho các module liên quan:**

**AUTOSAR (AUTomotive Open System Architecture)** là viết tắt của Kiến trúc hệ thống mở ô tô, dưới sự hợp tác của nhiều nhà cung cấp hang hóa, dịch vụ, các công ty, …. Để thiết lập một chuẩn code, giao tiếp dùng chung cho các hệ thống đặc biệt như ô tô

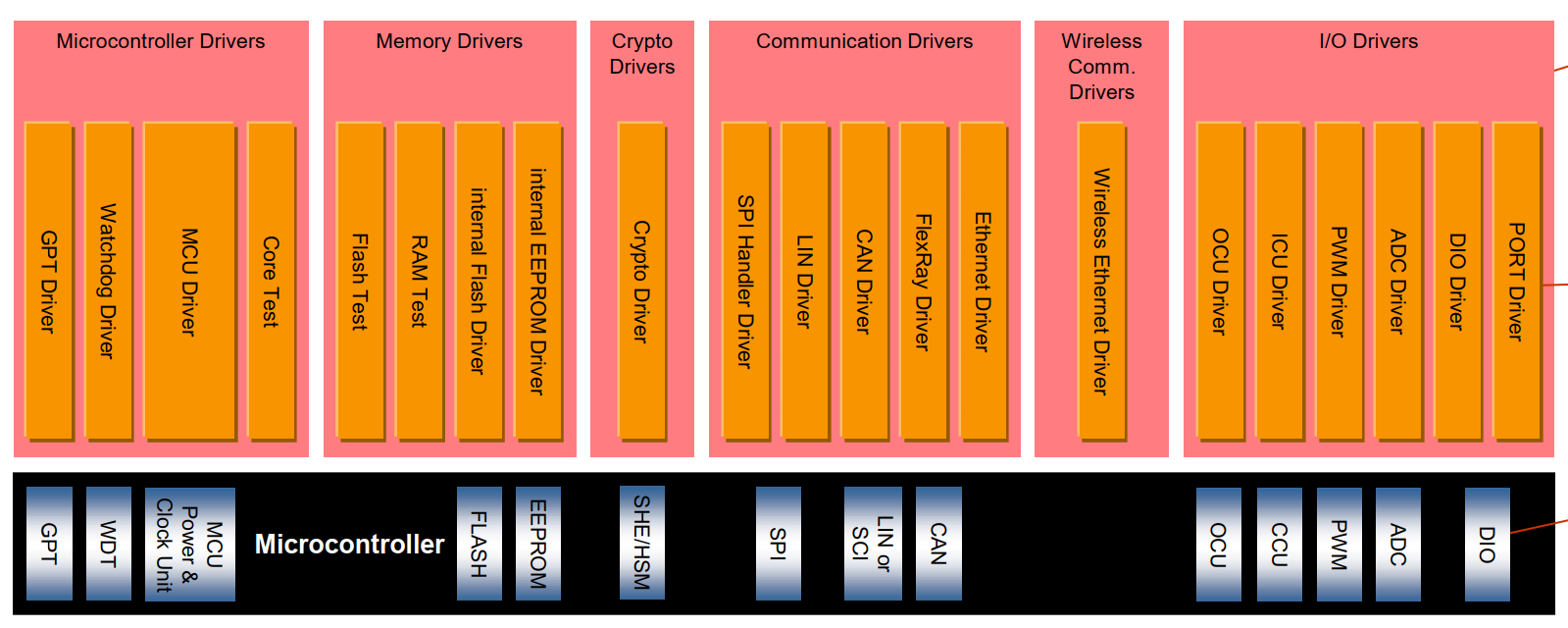


*Hình 5: Nền tảng AutoSar được xây dựng và chia ra thành 4 tầng chính*

*Hình 6: Mỗi Layer được chia thành nhiều nhóm chức năng khác nhau*

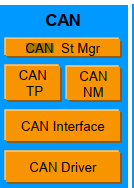
Trong kì thực tập này nhóm sẽ được tìm hiểu về **Microcontroller Abstraction Layer (MAL)** bao gồm nhiều Driver liên quan để hoàn thiện việc giao tiếp giữa các khối với nhau:

Các Driver chính bao gồm

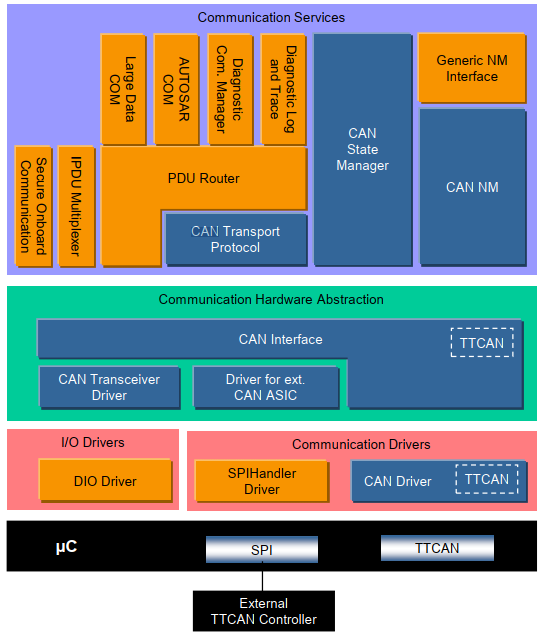
* Microcontroller Drivers
* **Communication Drivers**
* Memory Drivers
* I/O Drivers
* Cryto Drivers
* Wireless Communication Driver

*Hình 7: Các Driver chính*

Nhiệm vụ chính trong kì thực tập lần này của nhóm về Autosar là tìm hiểu về giao tiếp **Controller Area Network (CAN)**

Theo AutoSar, kiến trúc CAN có thể được xây dựng như sau:

*Hình 8: Kiến trúc CAN*



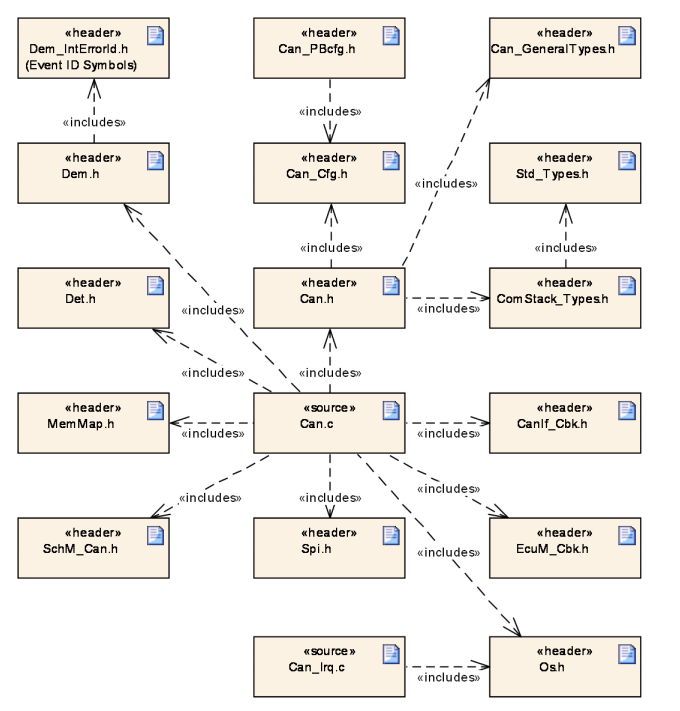
*Hình 9: Sơ đồ kiến trúc*

Trong đó, các mục tiêu xây dụng quan trọng là:

* CAN state manager
* CAN network manager
* CAN transport protocol
* CAN interface
* CAN driver

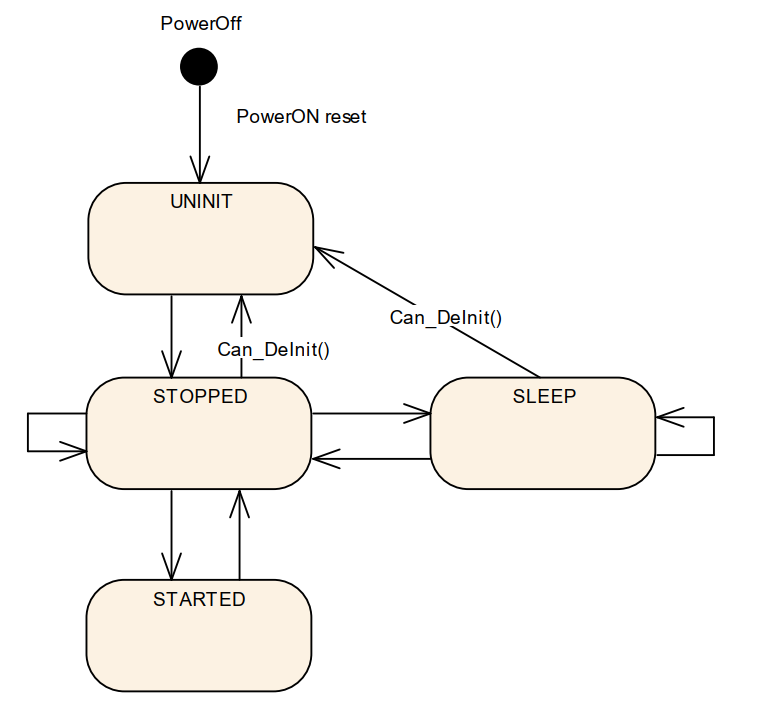
***Driver CAN:***

Để xây dụng giao tiếp CAN thì đầu tiên Autosar quy định cách tổ chức file Header như sau:



*Hình 10: Tổ chức của Header file*

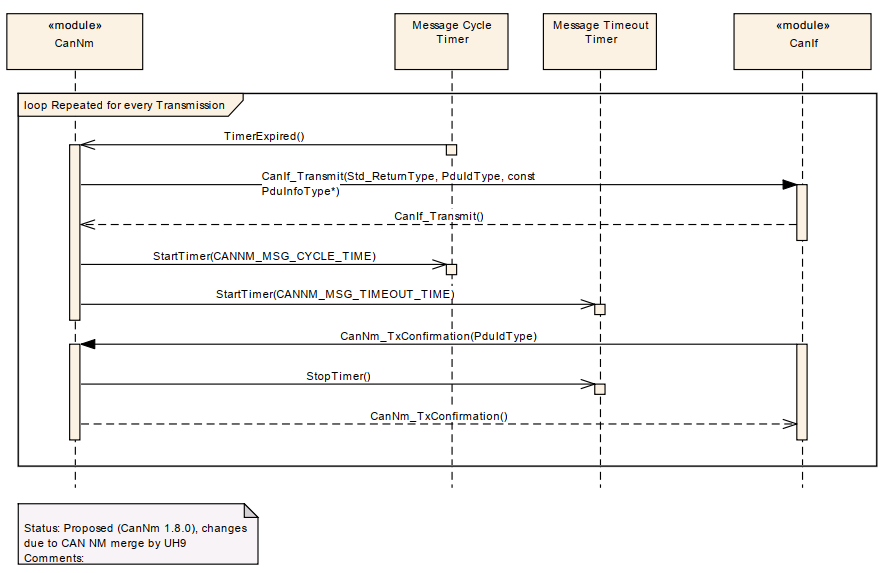
* CAN State Manager: bao gồm 4 State
* UNINIT: hủy khởi tạo CAN, các thanh ghi thuộc về CAN quay trở về trạng thái reset, ngoài ra Interrupt bị ngắt và CAN controller sẽ không còn trực tiếp can thiệp vào CAN BUS
* STOPPED: CAN controller được khởi tạo nhưng tạm thời bị ngừng và không can thiệp vào CAN bus
* STARTED: CAN hoạt động 1 cách binh thường.
* SLEEP: CAN sẽ vào trạng thái ngủ (cần phải wake-up)

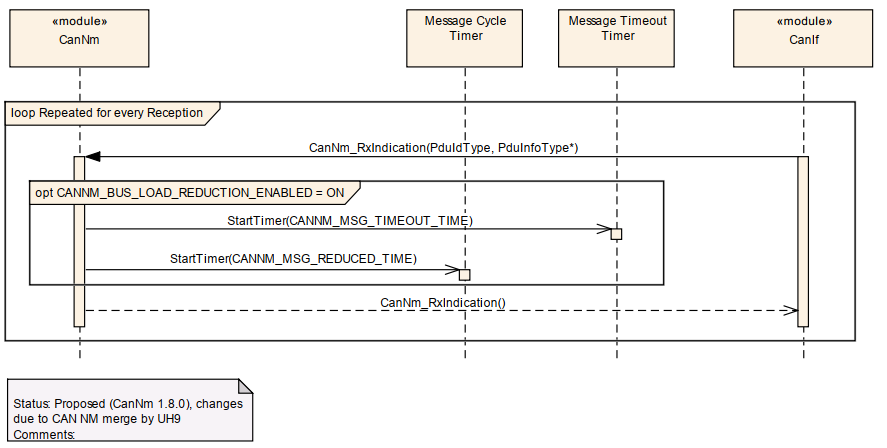
Sự tương tác tổng quát giữa các state với nhau:

*Hình 11: Tương tác giữa các state*

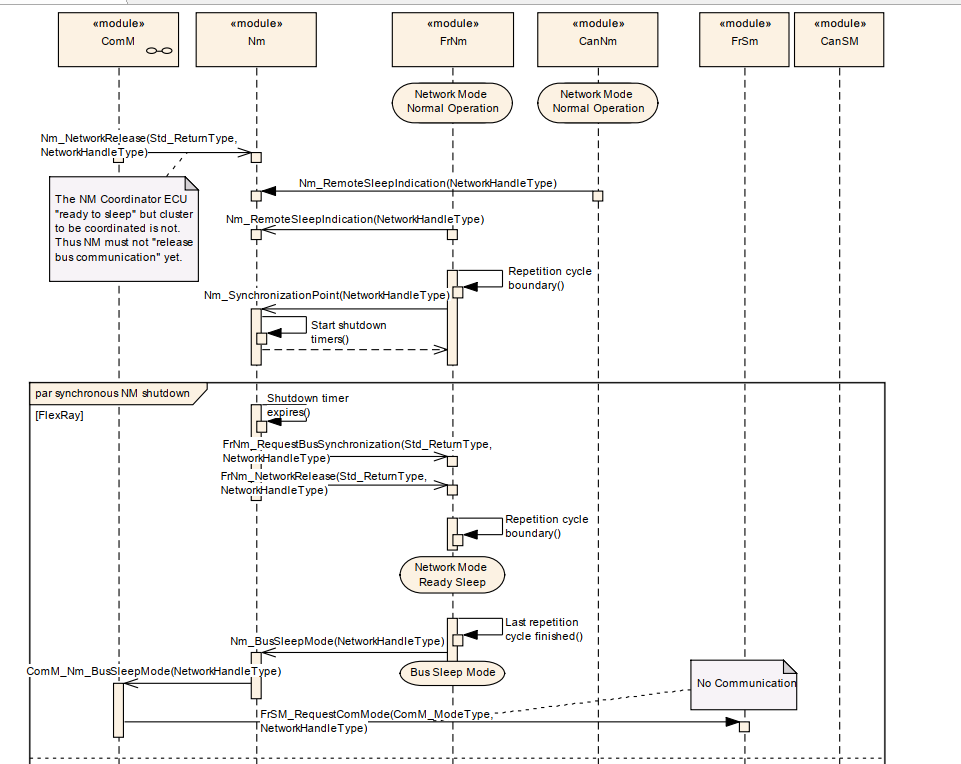
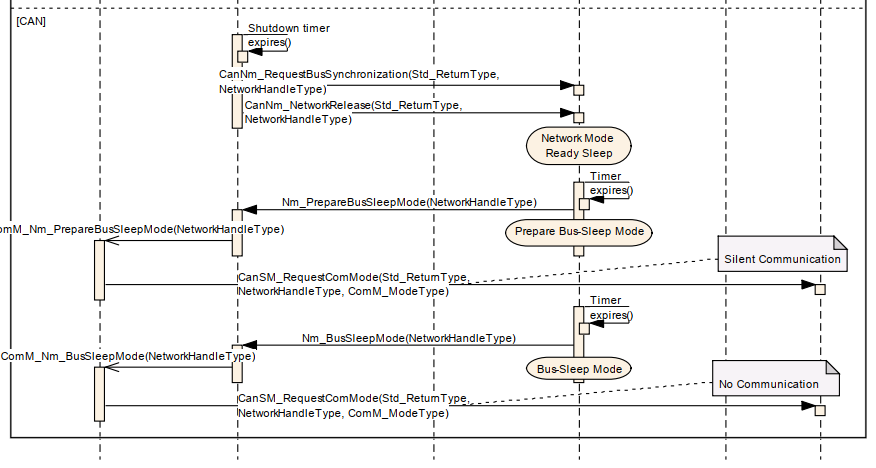
* CAN Network Manager: gồm 3 phần chính:

CANNm Transmission:

* Hình 12: CAN Transmission*



*Hình 13: CANNm Reception*

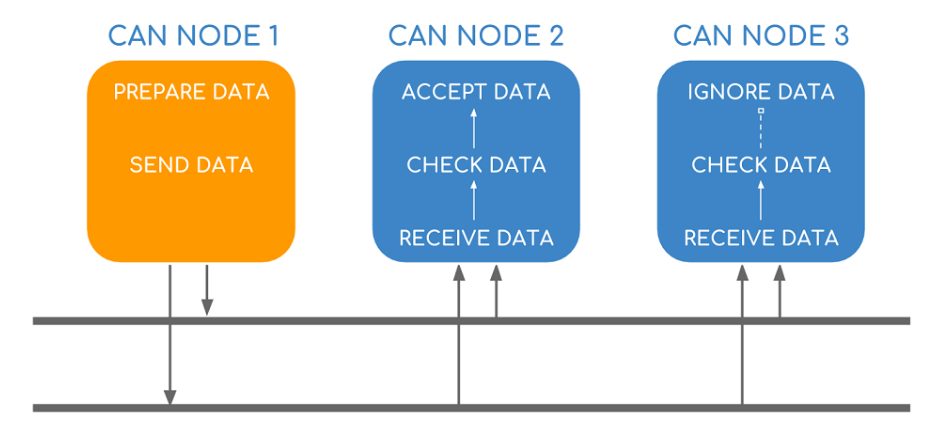


*Hình 14: NM Coordination*

* CAN Transport Protocol:

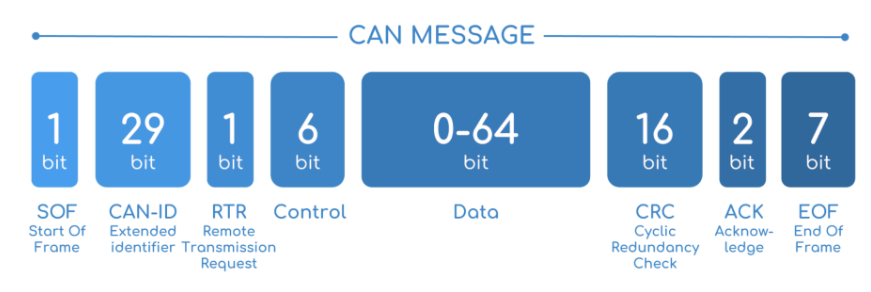
Trong các thiết bị chứa nhiều ECU, đặc biệt là xe hơi, … để các ECU giao tiếp được với nhau thì phương thức giao tiếp phổ biến nhất là CAN

* Cách thức hoạt động truyền của CAN

CAN hoạt động dưới hình thức truyền nhận thông qua 2 dây chính:

*Hình 15: Hoạt động truyền thông của CAN*

Mỗi Frame truyền của CAN được định dạng như sau:

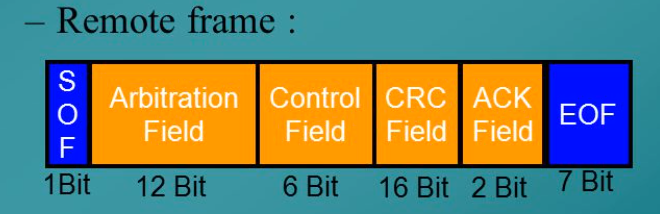


*Hình 16: các Frame truyền của CAN*

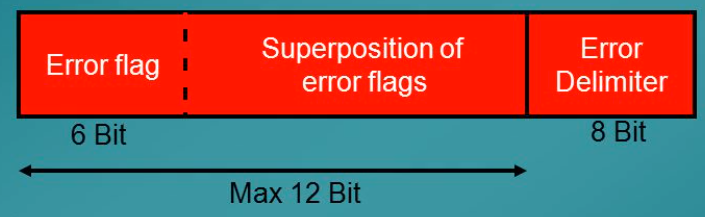
Ban đầu Node sẽ gửi đi một Frame như trên:

* SOF: báo hiệu bắt đầu một frame truyền.
* CAN-ID: sau đó từng node sẽ so sánh địa chỉ ID nếu đúng sẽ đọc bit RTR
* RTR: bit yêu cầu truyền nhận data, nếu node này đang nhận dữ liệu hoặc đang xử ra tranh chấp thì dữ liệu của node được ưu tiên cao nhất sẽ được giữ lại/chờ đợi đến khi được đọc, nhưng data của những node yêu tiên thấp hơn sẽ bị xóa bỏ
* Control: 6 bit xác định mục tiêu xử lý của data
* Data
* CRC: Bit check đúng sai
* ACK: Bit check truyền nhận được hay chưa
* EOF: báo hiệu kết thúc 1 frame

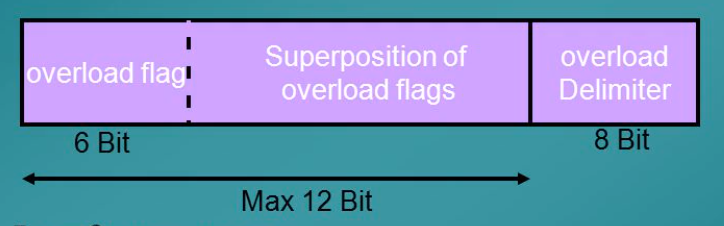
Ngoài ra CAN còn có các Frame truyền khác như: Remote Frame, Error Frame, Overload frame, …



*Hình 17: Remote frame*



*Hình 18: Error Flag*



*Hình 19: Overload frame*

1. **Nội dung nhiệm vụ của từng thành viên.**

Nhóm chúng em gồm 5 thành viên, với mỗi thành viên có nhiệm vụ và vai trò riêng biệt. Từ Section 2 trở đi thì có các công việc được phân công cụ thể như sau:

Thành viên:

1. **Trần Thanh Duy**

***Section 2:***

* Tìm hiểu về Board STM8L, giao tiếp I/O interface và các thanh ghi liên quan.
* Tìm hiểu và ghi chú các thanh ghi quan trọng, cần thiết của chức năng Interrupt.
* Kiểm tra và sửa lỗi cho chức năng UART
* Tìm hiểu và ghi chú về các thanh ghi chức năng của Timer
* Viết thư viện cho LCD trên Board hoạt động
* Tìm hiểu về cách hoạt động của Realtime Operating System

***Section 3:***

* Tìm hiểu và viết báo cáo sơ lược về Autosar
* Viết báo cáo về chức năng cơ bản của PORT và PWM của Autosar
* Tìm hiểu về CAN Network Management trên Autosar
* Thuyết trình về CAN Network Management trên Autosar

***Section 4:***

* Tìm hiểu về phần Test planning và Test execution
* Áp dụng kiến thức đã được trainning để test lại từng chức năng của bài Mock UART ở Section 2

1. **Lương Quốc Hải**

***Section 2:***

* Tìm hiểu Board STM8L, các chân I/O, các thanh ghi.
* Cách đọc tài liệu và tìm kiếm các chức năng tương ứng của các thanh ghi.
* Đầu tiên học về I/O interface: GPIO, cách cấu hình các chân không dùng thư viện.
* Tiếp theo tìm hiểu về Clock và Interrupts 🡪 Viết Interrupts Button, GPIO.init.
* Tìm hiểu về cách truyền dữ liệu dùng: SPI, I2C, UART. Cấu hình tiêu chuẩn cho các phương thức trên. Ở bài Midterm Mock thì nhóm em làm về UART 🡪 Cấu hình UART (Init)
* Tìm hiểu về Timer, các loại Timer và chức năng cho từng yêu cầu (Timer1)
* Viết font chữ cho thư viện LCD trên Board.
* Tìm hiểu về RTOS và cách hoạt động của nó.

***Section 3:***

* Giới thiệu tổng quát về Autosar, hiểu được tầm quan trọng của Autosar.
* Viết báo cáo về chức năng cơ bản của PORT và PWM của Autosar
* Tổng quát về Microcontroler: MCU, WDT, GPT
* Tỉm hiểu về CAN Network Management trên Autosar và tạo Init (CAN.INIT)

***Section 4:***

* Tổng quát về Testing, Unit Test
* Từ bài Midterm Mock áp dụng Unit Test để kiểm tra hệ thống và ghi nhận kết quả.

1. **Lã Tuấn Vinh**

***Section 2:***

* Đọc datasheet và chức năng các thanh ghi GPIO của STM8L.
* Tìm hiểu trên mạng cách lập trình GPIO cho STM8L không cần thư viện.
* Lập trình button, set clock, Interrupts, GPIO cơ bản.
* Đọc reference manual và user manual để lập trình LCD trên kit STM8L.
* Đọc chức năng thanh ghi và lập trình SPI , I2C trên STM8L.
* Đọc qua lý thuyết về RTOS.
* Test , đối chiếu code, sửa lỗi và thảo luận với các bạn trong nhóm.

***Section 3:***

* Đọc lý thuyết và viết báo cáo về chuẩn Autosar.
* Đọc lý thuyết và viết báo cáo về giao thức CAN.
* Viết và thuyết trình về thuật toán để giải quyết yêu cầu đề bài của công ty FPT đưa ra.

***Section 4:***

* Tìm hiểu về Testing và Unit Test.
* Xem lại các phần trước để làm Mock Project.

1. **Phạm Kim Thành**

***Section 2:***

* Tìm hiểu chức năng thanh ghi GPIO, cách xem địa chỉ các thanh ghi cơ bản.
* Đọc lý thuyết UART, Interrupt, Timer, SPI, I2C.
* Đọc reference manual để lập trình LCD cho STM8L.
* Lập trình các chức năng và giao tiếp cơ bản của STM8L(GPIO, Interrupt, Set Clock, Timer, UART, I2C, SPI).
* Thảo luận và fix bug code với các thành viên trong nhóm.
* Làm slide và thuyết trình về UART.
* Tìm hiểu và cách sử dụng thư viện RTOS.

***Section 3:***

* Đọc lý thuyết và làm slide về Autosar.
* Đọc lý thuyết, lập trình test thử và thuyết trình về CAN.
* Viết và sửa lỗi code với các thành viên trong nhóm.
* Vẽ sơ đồ giải thuật theo đề bài SPI mà công ty đưa ra.
* Đọc và viết báo cáo các chức năng của PWM.
* Đọc lý thuyết về MCU, WDP, GPT.

***Section 4:***

* Tìm hiểu về Test và Unit Test.
* Vận dụng những thứ đã học để làm Mock Project.
* Làm slide phân chia công việc và tổng hợp những thứ nhóm đã làm trong kỳ thực tập.
* Thuyết trình Mock Project.

1. **Nguyễn Tấn Phát**

***Section 2:***

* Tìm hiểu về Board STM8L, giao tiếp I/O interface và các thanh ghi liên quan.
* Tạo thư viện cơ bản cho kit STM8L
* Giúp các bạn trong nhóm hiểu về thư viện cơ bản dựa vào đó để hoàn thành các bài tập được giao
* Phân tích và phân chia công việc của các bài tập được giao
* Hướng dẫn và fix lỗi code của từng thành viên
* Tìm hiểu về cách hoạt động của RTOS

***Section 3:***

* Phân tích và phân chia công việc
* Tìm hiểu và viết báo cáo sơ lược về Autosar
* Đọc lý thuyết và viết báo cáo về giao thức CAN.
* Viết và thuyết trình về thuật toán để giải quyết yêu cầu đề bài của công ty FPT đưa ra.
* Giám sát quá trình test và viết báo cáo
* Tìm hiểu về phần Test planning và Test execution
* Áp dụng kiến thức đã được trainning để test lại từng chức năng của bài Mock UART ở Section 2

1. **Kết quả đạt được qua đợt thực tập.**

Trải qua đợt thực tập, mỗi thành viên của nhóm đạt được những kết quả cụ thể như sau:

1. **Trần Thanh Duy**

*Qua việc tìm hiểu và thực hành trực tiếp trên kit STM8L, em đã có cơ hội vận dụng về kiến thức lập trình nhúng và vi điều khiển đã được học. Từ đó củng cố lại kiến thức cũng như bổ sung thêm những thiếu sót của bản thân qua từng section như:*

***Section 2:***

* Hiểu rõ cách đọc tài liệu về vi điều khiển
* Học được cách cấu hình cơ bản về các thanh ghi chức năng của STM8L
* Hiểu về các loại xung và cấu hình xung nhịp cho STM8L hoạt động
* Củng cố kiến thức và thực hành nhuần nhuyễn hơn về Interrupt
* Biết cách debug về UART, cách để 2 board giao tiếp với nhau tốt nhất
* Biết cách sử dụng Timer cho các chức năng khác đặc biệt là PWM
* Hiểu nguyên lý hoạt động và có thể tự viết thư viện cho module LCD
* Biết cách chạy RTOS cho các dòng vi điều khiển nhỏ

***Section 3:***

* Lần đầu biết đến Autosar
* Đã tìm hiểu và ở mức có thể đọc được tài liệu về module PWM và PORT
* Hiểu sơ về nguyên lý hoạt động của CAN Network Management trên Autosar

***Section 4:***

* Biết cách vận dụng, phân tích các trường hợp để test một module hoặc một sản phẩm đã hoàn thiện

1. **Phạm Kim Thành**

***Section 2:***

* Nắm bắt được cách sử dụng cách loại tài liệu liên quan đến STM8L và áp dụng cho các kit tương tự
* Học được cách cấu hình cơ bản về các thanh ghi chức năng của STM8L
* Hiểu được cách viết 1 thư viện cơ bản để chạy những chức năng thiết yếu cho kit
* Nắm vững kiến thức về giao tiếp: UART, I2C, SPI, …
* Hiểu được nguyên lý hoạt động của LCD
* Biết giao tiếp giữa 2 board và cùng team hoàn thiện 1 bài toán lớn (MockTest)

***Section 3:***

* Hiểu được mục tiêu và hướng đi của AutoSar
* Nguyên cứu về nền tảng Autosar mà rõ nhất là giao tiếp CAN
* Hiểu được cách quy định/chuẩn mực code của Autosar

***Section 4:***

* Nắm bắt được kiến thức về test
* Thực hành cách kiểm tra sản phẩm do chính mình làm ra
* Phân biệt được giũa các loại hình kiểm tra sản phẩm

1. **Nguyễn Tấn Phát**

*Qua đợt thực tập em có được cơ hội trải nghiệm về phong cách làm việc chuyên nghiệp của công ty Fsoft và bổ sung rất nhiều kiến thức về ngành nhúng nói chung và CNTT nói riêng.*

***Section 2:***

* Biết được cách đọc các tài liệu liên quan đến vi điều khiển: Datasheet, References Manual, User Manual, …
* Có cái nhìn tổng quan hơn về lập trình thư viện cho 1 vi điều khiển: Thanh ghi, địa chỉ, …
* Hiểu kiến thức cơ bản về RTOS

***Section 3:***

* Được trải nghiệm về công nghệ AUTOSAR

***Section 4:***

* Được tham gia trải nghiệm về quy trình test chuyên nghiệp của doanh nghiệp

1. **Lương Quốc Hải**

***Section 2:***

* Hiểu biết rõ hơn về vi điều khiển và các linh kiện ngoại vi khác
* Tăng cao khả năng đọc tài liệu về các linh kiện
* Hiểu và có thể cấu hình GPIO cho vi điều khiển
* Hiểu hơn về các giao thức truyền thông SPI, I2C, UART
* Biết cách viết thư viện cho một module ngoại vi
* Hiểu được cách hoạt động của RTOS

***Section 3:***

* Biết thêm kiến thức về Autosar
* Nắm được kiến thức tổng quát về Microcontroler MCU, WDT, GPT
* Nắm được cấu hình cơ bản cho Can Network Management trên Autosar

***Section 4:***

* Nắm được cơ bản về quy trình Test của doanh nghiệp

1. **Lã Tuấn Vinh**

***Section 2:***

* Vận dụng được kiến thức đã học để lập trình cho vi điều khiển
* Hiểu rõ hơn về cấu trúc, đặc biệt là thanh ghi của vi điều kheiern
* Nắm được kiến thức cơ bản về RTOS

***Section 3:***

* Lần đầu được tiếp cận về Autosar

***Section 4:***

* Hiểu thêm về cách Test một sản phẩm

1. **Nhận xét, góp ý về Chương trình Đào tạo của Khoa Kỹ Thuật Máy Tính.**

***Nhận xét:***

* Các môn học cơ sở ngành của khoa như Lý thuyết mạch điện, Vi xử lý – vi điều khiển, Hệ thống nhúng… đã cung cấp cho nhóm khối kiên thức cơ bản về việc lập trình nhúng, mạch điện tử để dễ dàng đi vào tìm hiểu và hoàn thành công việc được giao ở đợt thực tập.
* Cùng với đó thì các môn học như Thiết kế luận lý số, Xử lý tín hiệu số đã rèn luyện và nâng cao khả năng tư duy, logic và giải quyết vấn đề hơn.
* Câu lạc bộ CEEC cung cấp phòng LAB, linh kiện hỗ trợ cho quá trình học tập.
* Tuy nhiên còn một số hạn chế như: môn Vi xử lý – vi điều khiển học về dòng 8051 và 8061 khá cũ làm việc tìm kiếm linh kiện học tập, thực hành khó khăn.

***Góp ý:***

* Hướng dẫn thêm cho sinh viên về việc lựa chọn tài liệu học tập: Chẳng hạn như một MCU, module… có rất nhiều nhà sản xuất và nhiều mẫu mã khác nhau nhưng tài liệu trên mạng lại rất nhiều. Muốn hiểu và lập trình cho một module hoạt động được thì phải đọc qua nhiều kiểu tài liệu như tài liệu như datasheet, tài liệu hướng dẫn người sử dụng của nhà sản xuất và đôi khi phải đọc cả tài liệu hướng dẫn cho thư viện. Việc có quá nhiều loại tài liệu như vậy sẽ dễ làm sinh viên lựa chọn nhầm lẫn và khó khăn cho việc bắt đầu hơn.
* Nâng cấp một số linh kiện cũng như phòng thực hành đã cũ: Ở một số môn học thực hành cơ sở ngành như Thiết kế luận lý số thì phòng thực hành và thiết bị đã quá cũ (Phòng C302), linh kiện bị hỏng nhiều. Có lúc phải đến 2 nhóm dùng chung 1 kit để thực hành, máy đo tín hiệu thì khởi động lâu và chập chờn rất khó đo được kết quả đúng. Máy tính rất hay tự khởi động lại, đôi lúc làm sinh viên mất cả bài tập đang làm dang dở trong cả tiết học.
* Hỗ trợ thêm và kiểm tra lại về linh kiện thực hành: Ví dụ như môn học hiện hành là Lập trình nhúng căn bản. Đôi lúc sinh viên phải liên tục code đi code lại vì nghĩ mình sai và ngồi sửa lỗi ở phần code nhưng không biết rằng một module trên bản mạch đã hỏng, không hoạt động được chứ không phải do khâu lập trình. Gây mất thời gian không cần thiết.
* Tăng thêm tiết thực hành: Lý thuyết thì sinh viên nắm khá tốt nhưng còn chưa biết cách vận dụng ở một trường hợp cụ thể. Điều này làm khó củng cố kiến thức và quên ngay kiến thức môn học từ 1 đến 2 kì học sau đó.
* Phát huy tinh thần làm việc nhóm cho sinh viên: Khi báo cáo một bài tập nhóm hoặc báo cáo một đồ án, giảng viên nên yêu cầu bảng phân chia công việc. Từng thành viên chịu trách nhiệm cụ thể cho phần nào, và viết cụ thể nội dung kiến thức mà từng cá nhân đã học được sau đồ án/ bài tập đó là gì.
* Nâng cao điểm phần trăm đồ án hơn là điểm thi: Bản thân em thấy việc chú trọng vào làm một đồ án kéo dài cả kì quan trọng hơn là điểm của một ngày thi. Vì sau khi thực hiện xong một đồ án với thời gian dài, sinh viên sẽ rút ra được nhiều kiến thức về môn học hơn và nhớ lâu hơn. Việc nâng cao phần trăm đồ án cho sinh viên thấy tầm quan trọng của môn học nên sẽ học và thực hành trong xuyên suốt cả học kì thay vì chỉ chú tâm vào ôn thi vài ngày rồi lại quên kiến thức môn đó ở các kì học kế tiếp.