TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

**KHOA KỸ THUẬT MÁY TÍNH**

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



**HỆ THỐNG NHÚNG TRONG AUTOSAR**

**BÁO CÁO THỰC TẬP DOANH NGHIỆP**

SV thực hiện: Đặng Đình Khang

MSSV: 15520336

Lớp: KTMT2015

GVHD: Nguyễn Mạnh Thảo

*Thành phố Hồ Chí Minh – Năm 2019-2020.*

**Mục lục**

Chương I. Giới thiệu về tổ chức của nơi thực tập4

1.1 FGA - Đơn vị làm phần mềm ô tô lớn nhất tại Việt Nam4

1.2 Phòng Nghiên cứu và Phát triển Automotive (FSOFT - UIT Automotive R&D LAB)5

Chương II. Lịch làm việc tại nơi thực tập6

Chương III. Nội dung nhiệm vụ chính được giao10

Chương IV. Nội dung các công việc và kết quả đạt được12

4.1 Tổng quan về AUTOSAR12

4.2 Phân lớp trong Basic Software14

4.2.1 MCAL – Microcontroller Abstraction Layer14

4.2.2 ECU Abstraction Layer15

4.2.3 Complex Drivers15

4.2.4 Service Layer16

4.3 Phân tích các tài liệu của lớp MCAL16

4.3.1 Requirement - SRS16

4.3.2 Specification - SWS17

4.4 Testing19

Chương V. Kết quả đạt được qua đợt thực tập21

5.1 Những kiến thức lý thuyết đã được củng cố21

5.2 Những kỹ năng thực hành đã được học thêm21

5.3 Những kinh nghiệm thực tiễn đã tích lũy được22

# Chương VI. Nhận xét góp ý về Chương trình Đào tạo của Khoa22

Tài liệu tham khảo23

**Danh mục bảng**

Bảng 1: Nội dung chương trình thực tập6

Bảng 2: Nội dung các công việc đã thực hiện trong đợt thực tập10

**Danh mục hình**

Hình 1: FPT Software Hồ Chí Minh4

Hình 2: Đại diện FSOFT, công ty AISIN và các thầy cô trường Đại học Công nghệ Thông tin cùng chụp ảnh lưu niệm tại phòng nghiên cứu5

Hình 3: AUTOSAR tạo sự liên kết và không lệ thuộc giữa các hardware và phần mềm13

Hình 4: Các thành viên phát triển AUTOSAR14

Hình 5: Các phân lớp chính trong AUTOSAR15

Hình 6: Các module trong lớp MCAL15

Hình 7: Phần mở đầu của tài liệu SRS16

Hình 8: Phần mở đầu của tài liệu SWS17

Hình 9: Yêu cầu của API Port\_Init trong module PORT18

Hình 10: Mô hình V-model19

**Lời cảm ơn**

Lời cám ơn đầu tiên em xin gửi đến Ban giám hiệu trường Đại học Công Nghệ Thông Tin, quý thầy cô khoa Kỹ Thuật Máy Tính đã đưa phòng Lab FSOFT – UIT Automotive về đến khoa. Từ đó tạo điều kiện thuận lợi để em và các bạn sinh viên có được môi trường học, thực tập tốt và tiếp cận được những kiến thức mới, xu hướng công nghệ của một doanh nghiệp trong suốt những tháng hè năm 2019.

Em cũng xin gửi lời cám ơn chân thành đến ban lãnh đạo FSOFT, các anh chị trainer, người đã tận tình hướng dẫn em các kiến thức, công việc trong suốt quá trình thực tập tại phòng Lab.

Vì thời gian và kiến thức còn hạn hẹp nên bài báo cáo không thể tránh khỏi những thiếu sót, rất mong sự góp ý quý thầy cô và các bạn, để em rút kinh nghiệm và hoàn thành tốt hơn. Em xin chân thành cám ơn!

# **Chương I. Giới thiệu về tổ chức của nơi thực tập**

FPT Software là một thành viên thuộc Tập đoàn FPT thành lập ngày 13/1/1999, hoạt động trong lĩnh vực gia công xuất khẩu phần mềm của Việt nam. Với các quy trình chuẩn của thế giới về sản xuất phần mềm, quản lý chất lượng và bảo mật thông tin như CMMI-5, ISO 9001:2000, ISO 27001:2005, FPT Software là một Công ty phần mềm có các quy trình chuẩn thế giới hàng đầu tại Việt nam. Hiện tại, FPT Software có 7 công ty thành viên tại Nhật bản, Singapore, Pháp, Mỹ, Malaysia, Úc, Việt nam và 3 chi nhánh tại Hà Nội, Tp. HCM và Đà Nẵng.



*Hình 1: FPT Software Hồ Chí Minh*

## **1.1 FGA - Đơn vị làm phần mềm ô tô lớn nhất tại Việt Nam**

FGA (FPT Global Automotive) là đơn vị trực thuộc FPT Software chuyên về lĩnh vực Automotive, hướng đến các sản phẩm và giải pháp được ứng dụng rộng rãi trong xe hơi tương lai. FGA phát triển ở bốn mảng chính: Ứng dụng trong ô tô, Thiết kế, IT/ERP và Kiểm thử phần mềm ô tô.

Trong số 7 lĩnh vực được dự đoán tăng trưởng mạnh của Automotive, FGA tập trung phát triển ứng dụng trong ô tô (automotive applications) tại 3 lĩnh vực chính là xe tự hành, hệ thống an toàn và hệ thống thông tin giải trí.

Hiện FGA là đối tác của hàng chục hãng xe hơi nổi tiếng nhất thế giới đến từ Nhật Bản, Hàn Quốc, EU… với đội ngũ chuyWên gia nhiều năm kinh nghiệm trong lĩnh vực CAD/CAM/CAE và công nghệ ô tô. Tính đến nay, FGA đang tham gia khoảng 150 dự án trong tất cả các lĩnh vực Automotive.

**1.2 Phòng Nghiên cứu và Phát triển Automotive (FSOFT - UIT Automotive R&D LAB)**

Ngày 06/6/2019, trường Đại học Công nghệ Thông tin ĐHQG-HCM phối hợp cùng tập đoàn FPT tổ chức lễ khánh thành Phòng Nghiên cứu và Phát triển Automotive (FSOFT - UIT Automotive R&D LAB). Đây là phòng nghiên cứu và phát triển chuyên về lĩnh vực xe tự hành - một lĩnh vực đang được FSOFT đẩy mạnh đầu tư phát triển trong những năm gần đây, trong khi đó lĩnh vực này cũng là mũi nhọn nghiên cứu của thầy và trò Khoa Kỹ thuật Máy tính khi nội dung lập trình cho xe tự hành được đưa vào cuộc thi Lập trình xe tự động - UIT Car Racing lần thứ VIII do trường Đại học Công nghệ Thông tin tổ chức vào từ tháng 3 đến tháng 5 vừa qua. TS. Nguyễn Minh Sơn - Trưởng Khoa Kỹ thuật Máy tính được giao nhiệm vụ Trưởng phòng nghiên cứu.Phòng nghiên cứu đã tạo ra môi trường tốt cho sinh viên thực tập và làm việc,tiếp cận trực tiếp với công việc thực tế của môi trường doanh nghiệp vận dụng được những kiến thức đã học trong trường,củng cố kiến thức bản thân đồng thời tiếp cận với những công nghệ mới trong quá trình thực tập.



*Hình 2: Đại diện FSOFT, công ty AISIN và các thầy cô trường Đại học Công nghệ Thông tin cùng chụp ảnh lưu niệm tại phòng nghiên cứu*

# **Chương II. Lịch làm việc tại nơi thực tập**

- Thời gian thực tập: 3 tháng (6/6/2019 – 5/9/2019)

- Thời gian làm việc: 40 giờ/tuần, từ thứ 2 đến thứ 6.

- Vắng mặt: phải thông báo trước.

- Chương trình thực tập do FPT Software đề xuất:

**Bảng 1: Nội dung chương trình thực tập**

|  |  |  |
| --- | --- | --- |
| **Tông quan và công cụ** | | |
| Test đầu vào | Bài test ngôn ngữ C Phỏng vấn | 1. Bài test ngôn ngữ C 2. Phỏng vấn |
| Buổi 1 | Giới thiệu tổ chức FGA.AIS Giới thiệu tổ chức khóa training lập trình nhúng | 1. Giới thiệu tổ chức FGA.AIS 2. Giới thiệu tổ chức khóa training lập trình nhúng |
| Buổi 2 | Công cụ quản lý source code | 1. SVN 2. GIT  3. Luyên tập sử dụng SVN, GIT |
| Buổi 3 | Tổng quan Embedded | 1. Tổng quan về hệ thống nhúng  2. Cài đặt môi trường, công cụ  3. Tự nghiên cứu |
| **Training Lập trình nhúng cơ bản** | | |
| Buổi 1 | Tổng quan về board STM8L Discovery | 1. Giới thiệu 2. Thanh ghi, phép toán logic, dịch/clear bit  3. Bài tập thực hành |
| Buổi 2 | GPIO | 1. Cấu hình, thực hiện chức năng Input, Output trên các chân  2. Bài tập thực hành |
| Buổi 3 | Clock | 1. Cấu hình bộ tạo xung và cấp xung cho các ngoại vi  2. Bài tập thực hành |
| Kỹ năng mềm | Làm việc nhóm Viết email Viết báo cáo, tài liệu | 1. Làm việc nhóm + Giao tiếp giữa các thành viên  + Phân chia công việc 2. Viết email 3. Viết báo cáo, tài liệu |
| Buổi 4 | Ngắt | 1. Cấu hình ngắt ngoài trên các chân IO  2. Bài tập thực hành |
| Buổi 5 | Communication Stack | 1. Giao thức SPI, I2C, UART  2. Bài tập thực hành |
| Buổi 6 | Timer | 1. Timer  2. Bài tập thực hành |
| Buổi 7 | Linker | 1. Linker  2. Bài tập thực hành |
| Buổi 8 | Realtime Operating System | 1. Tổng quan RTOS  2. Bài tập thực hành |
| Kỹ năng mềm | Quản lý thời gian  Quản lý công việc | Áp dụng các phương pháp quản lý công việc, lên kế hoạch và điều chỉnh công sức bỏ ra. |
| Bài tập lớn | Đề tài 1 | RTOS |
| Đề tài 2 | I/O Interrupt |
| Đề tài 3 | Timer |
| Đề tài 4 | I2C |
| Đề tài 5 | SPI |
| Đề tài 6 | UART |
| **AUTOSAR** | | |
| Buổi 1 | Tổng quan về AUTOSAR | 1. Tổng quan về tiêu chuẩn AUTOSAR 2. Cách nghiên cứu yêu cầu của Module  3. Cách nghiên cứu tài liệu AUTOSAR 4. Cơ bản về lớp MCAL |
| Buổi 2 | Tổng quan về nhóm Microcontroler: MCU, WDT, GPT | 1. Cách nghiên cứu đặc điểm của Module (AUTOSAR SWS) 2. Hiểu các hoạt động của module, API, yêu cầu về chức năng của module  3. Hiểu các cấu hình của module 4. Hiểu cách test module |
| Buổi 3 | Tổng quan về nhóm IO: DIO, PORT, PWM | 1. Cách nghiên cứu đặc điểm của Module (AUTOSAR SWS) 2. Hiểu các hoạt động của module, API, yêu cầu về chức năng của module  3. Hiểu các cấu hình của module 4. Hiểu cách test module |
| Buổi 4 | Module ADC | 1. Cách nghiên cứu đặc điểm của Module (AUTOSAR SWS) 2. Hiểu các hoạt động của module, API, yêu cầu về chức năng của module  3. Hiểu các cấu hình của module 4. Hiểu cách test module |
| Buổi 5 | Memory driver: Flash | 1. Cách nghiên cứu đặc điểm của Module (AUTOSAR SWS) 2. Hiểu các hoạt động của module, API, yêu cầu về chức năng của module  3. Hiểu các cấu hình của module 4. Hiểu cách test module |
| Communication Modules: CAN, LIN | 1. Cách nghiên cứu đặc điểm của Module (AUTOSAR SWS) 2. Hiểu các hoạt động của module, API, yêu cầu về chức năng của module  3. Hiểu các cấu hình của module 4. Hiểu cách test module |
| Communication: SPI | 1. Cách nghiên cứu đặc điểm của Module (AUTOSAR SWS) 2. Hiểu các hoạt động của module, API, yêu cầu về chức năng của module  3. Hiểu các cấu hình của module 4. Hiểu cách test module |
| Communication: ETH | 1. Cách nghiên cứu đặc điểm của Module (AUTOSAR SWS) 2. Hiểu các hoạt động của module, API, yêu cầu về chức năng của module  3. Hiểu các cấu hình của module 4. Hiểu cách test module |
| Bài tập lớn | Tổng kết và bài tập lớn |  |
| Kỹ năng mềm | Làm việc trong dự án | 1. Ước lượng và lên kế hoạch 2. Truy xuất nguồn gốc 3. Phân tích các tác động 4. Hướng dẫn review |
| Horenso | Làm việc với Horenso: 1. Horenso - Hokoku 2. Horenso - Renraku 3. Horenso - Sodan |
| **Kiểm thử trong lập trình nhúng** | | |
| Buổi 1 | Testing in Embedded | 1. Mục đích 2. Hiểu các loại kiểm thử UI, IT, ST 3. Các cách kiểm thử phần mềm |
| Luyện tập Unit Test | Thực hiện Unit Test trên souce code có sẵn |
| Bài tập |  |

# **Chương III. Nội dung nhiệm vụ chính được giao**

Quá trình thực tập được chia thành các Buổi và Section được thể hiện ở lịch làm việc do công ty phân bổ trong phần II. Kết thúc mỗi Section đều có những bài test để củng cố kiến thức được học.

**Bảng 2: Nội dung các công việc đã thực hiện trong đợt thực tập**

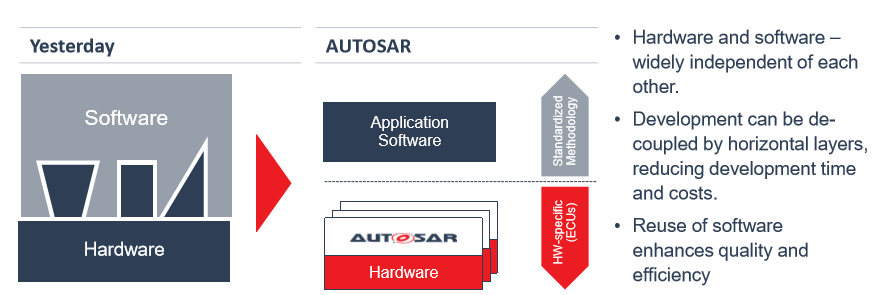
|  |  |  |
| --- | --- | --- |
| **STT** | **Nội dung** | **Chi tiết** |
| 1 | Tổng quan | - Công cụ quản lý source code GIT và SVN  - Tổng quan về hệ thống nhúng, các khái niệm, ứng dụng. |
| 2 | Hệ thống nhúng cơ bản | - Luyện tập, củng cố các kỹ thuật lập trình nhúng với ngôn ngữ C trên board STM8L Discovery:  + Kỹ thuật lập trình với thanh ghi, toán tử bitwise, ôn tập ngôn ngữ C.  + GPIO: cấu hình và điều khiển các chân chức năng, xuất tín hiệu output, đọc input.  + Clock generator: cấu hình bộ tạo xung của Mcu, thực hiện cấp clock cho các peripheral.  + Timer: cấu hình bộ đếm timer, thực hiện đếm số trên LCD.  + Giao thức communication: UART, SPI. Cấu hình và thực hiện truyền nhận giữa 2 Board.  + Real Time Operating System: khái niệm, demo Atomthread trên Board STM8L.  - Bài tập lớn (Mô phỏng trò chơi thi nhấn phím nhanh giữa hai board STM8L bằng giao thức SPI). |
| 3 | AUTOSAR | - Tổng quan về AUTOSAR  - Cách nghiên cứu tài liệu về AUTOSAR  - Tổng quan về lớp MCAL (Microcontroller Abstraction Layer) trong AUTOSAR.  - Tổng quan và phân tích các chức năng của các module trong lớp MCAL:  + Module Mcu, Gpt.  + Module Dio, Port  + Module Can  + Module Spi  - Bài tập lớn: phân tích, đưa ra định hướng cài đặt cho các module trong AUTOSAR đựa trên datasheet của hardware (module Port, Spi). |
| 4 | Testing | - Tổng quan về kiểm thử phần mềm.  - Một số qui trình kiểm thử mềm.  - Unit Test  - Intergration Test, System Test  - Bài tập lớn: thực hiện System Test trên bài tập lớn ở phần 1. |
| 5 | Kỹ năng mềm | - Cách viết email, tài liệu, báo cáo  - Làm việc nhóm  - Quản lý thời gian |

# **Chương IV. Nội dung các công việc và kết quả đạt được**

## **4.1 Tổng quan về AUTOSAR**

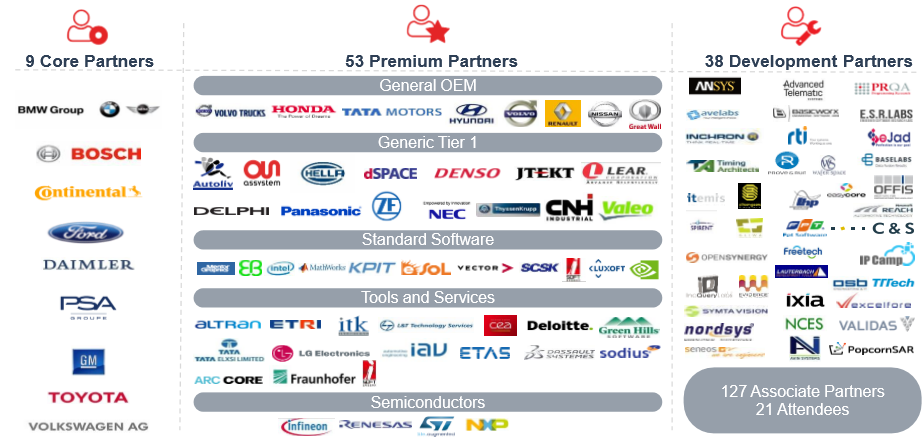
Với sự phát triển mạnh mẽ của công nghệ, ô tô ngày nay đều được tích hợp và được hỗ trợ bởi các hệ thống điện tử, các phần mềm. Công nghệ được tích hợp ở mọi khía cạnh trên chiếc ô tô từ động cơ, các hệ thống an toàn, các hệ thống điều khiển … cho đến hệ thống thông tin, giải trí ở trên xe. AUTOSAR ra đời với mục đích tạo ra một kiến trúc phần mềm tiêu chuẩn giúp các hãng xe đựa vào đó mà giảm chi phí nghiên cứu, xây dựng hệ thống phần mềm trên ô tô.

Các hệ thống điện tử, phần mềm trên ô tô trước đây khá rời rạc, chúng thường được chia thành các module, chỉ thực hiện từng chức năng riêng rẽ, không có tính kết nối với nhau. Kiến trúc của AUTOSAR cho phép các hệ thống điện tử trên xe tạo thành một network riêng và liên kết tới cả hệ thống phần mềm chính trên xe. Điều đó giúp cho bản thân các module trên xe có thể giao tiếp với nhau, việc sửa chữa cũng trở nên dễ dàng hơn khi thông tin của tất cả module trên xe đều có thể thu thập được thông qua hệ thống phần mềm chính trên ô tô.



*Hình 3: AUTOSAR tạo sự liên kết và không lệ thuộc giữa các hardware và phần mềm*

Các thành viên phát triển AUTOSAR:



*Hình 4: Các thành viên phát triển AUTOSAR*

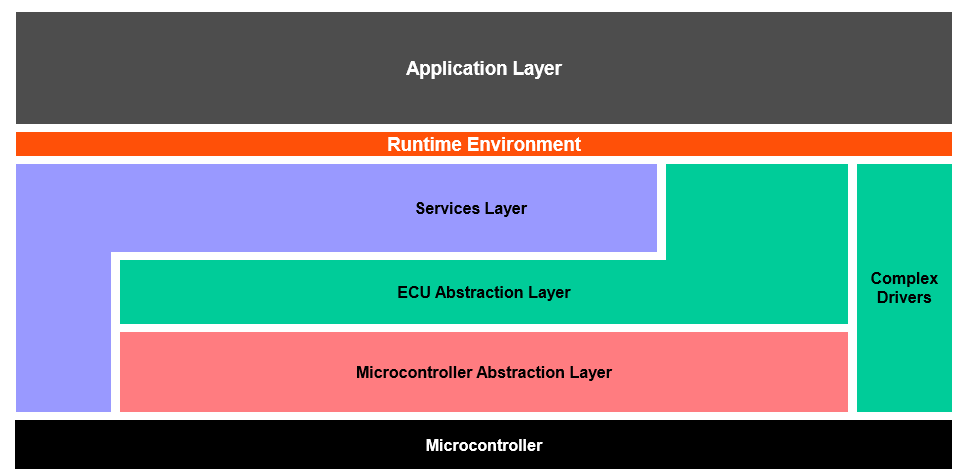
Kiến trúc của AUTOSAR:

+ Appilcation Layer.

+ Runtime Envirionment (RTE) là lớp liên kết giữa Application và Basic Software. Cung cấp các service giúp cho lớp application có thể giao tiếp với các hệ thống điện tử trong xe.

+ Basic Software (BSW): là lớp phụ trách điều khiển hardware, kết nối các hardware và liên kết tới application.

FPT Software hiện đang tập trung vào lớp Basic Software và chủ yếu làm vào lớp MCAL. FPT Software cũng đang mở rộng lên các lớp trên MCAL trong BSW.



*Hình 5: Các phân lớp chính trong AUTOSAR*

Tại mỗi lớp, AUTOSAR đều đưa ra tiêu chuẩn và yêu rõ ràng, các thành viên phát triển bắt buộc phải tuân theo các tiêu chuẩn đó. Các tiêu chuẩn và yêu cầu của tất cả module trong AUTOSAR được qui định trong hai tài liệu chính là yêu cầu (SRS, Requirement) và các chức năng (SWS, Specification).

## **4.2 Phân lớp trong Basic Software**

### 4.2.1 MCAL – Microcontroller Abstraction Layer

Đây là lớp thấp nhất trong Basic Software, thực hiện cài đặt các chức năng của Microcontroller và tạo ra một lớp API giúp người phát triển ở lớp trên là ECU Abstraction có thể sử dụng và tương tác dễ dàng với Microcontroller khi chỉ cần dựa theo các chức năng mà AUTOSAR đã qui định cho module đó.

Các module trong lớp MCAL:

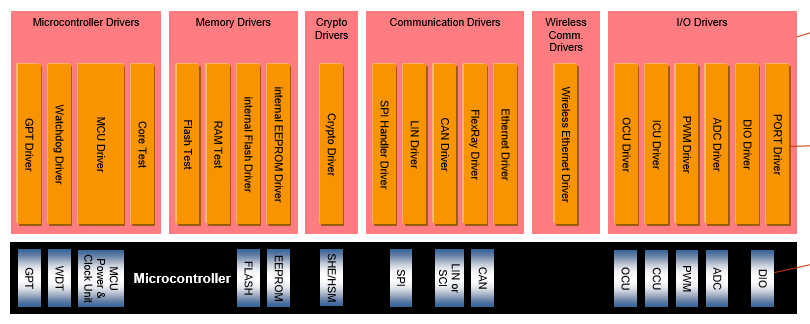
+ Microcontroller Drivers: là các module sử dụng các peripheral nội bộ trong Microtroller.

+ Memory Drivers: là các module tương tác với bộ nhớ.

+ Crypto Drivers: tương tác thiết bị Crypto.

+ Commnuication Driver: là các module thực hiện các chuẩn giao tiếp.

+ I/O Driver: là các module chức năng analog, digital trên các chân I/O.



*Hình 6: Các module trong lớp MCAL*

### 4.2.2 ECU Abstraction Layer

Lớp ECU abstraction layer có nhiệm vụ cài đặt driver và cung cấp API giúp cho lớp trên có thể sử dụng và truy cập được tới các ngoại vi và thiết bị bất kể nó nằm trong hoặc ngoài, bất kể nó là hardware unit hoặc port pin nào của Microcontroller. Ví dụ: Loa là một thiết bị nằm trong một hệ thống điện tử. Lớp trên chỉ cần thông qua các API của ECU Abstraction Layer là có thể tương tác được với thiết bị loa đó.

### 4.2.3 Complex Drivers

Complex Drivers là các module có chức năng đặc biệt và chức năng đó không nằm trong các Layer hoặc tiêu chuẩn của AUTOSAR. Complex Drivers trực tiếp điều khiển Microcontroller và sử dụng trong các chức năng mà yêu cầu của chúng không phù hợp với AUTOSAR. Ví dụ như hệ thống phun xăng điện tử, van điện tử … các chức năng này yêu cầu về thời gian, tốc độ xử lý khá khắt khe.

### 4.2.4 Service Layer

Service Layer giúp application sử dụng được các chức năng được cài đặt bởi ECU Abstraction Layer và cung cấp các chức năng:

+ Hệ điều hành.

+ Mạng lưới liên kết và quản lý các hardware.

+ Memory Service (quản lý NVRAM).

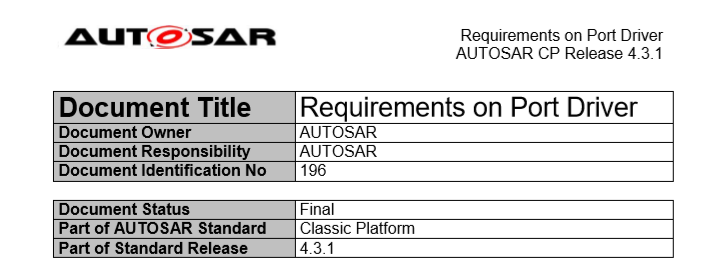
+ Các Service phát hiện lỗi (thông qua UDS Communication và lỗi bộ nhớ).

+ Quản lý trạng thái của ECU

## **4.3 Phân tích các tài liệu của lớp MCAL**

Tất cả các tài liệu về các module, các lớp của AUTOSAR đều được tìm thấy trên [https://www.AUTOSAR.org/standards/classic-platform/](https://www.autosar.org/standards/classic-platform/), có hai loại tài liệu chính là Requirement – SRS và Specification SWS.

### 4.3.1 Requirement – SRS



*Hình 7: Phần mở đầu của tài liệu SRS*

Tài liệu SRS đưa ra các yêu cầu chung, ngắn gọn cho một module cần phải đáp ứng. Mỗi requirement đều có một mã định danh (ví dụ: SRS\_Port\_12001) giúp cho việc truy xuất, theo dõi các yêu cầu dễ dàng hơn.

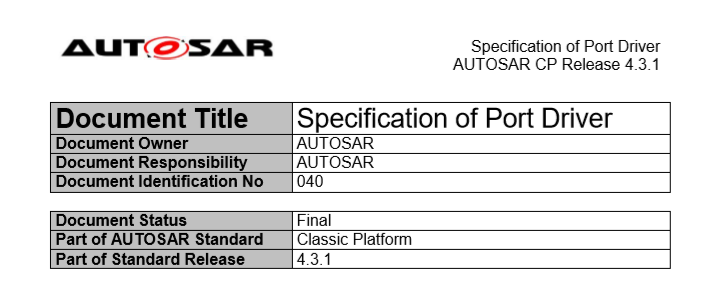
Các phần quan trọng của tài liệu SRS:

+ Chương 3: Acronyms and abbreviations – giải thích các từ viết tắt

+ Chương 4: Functional Overview – tổng quan chức năng của module

+ Chương 5: Requirement Specification – các yêu cầu về tham số cấu hình, các API mà module phải thực hiện.

### 4.3.2 Specification – SWS



*Hình 8: Phần mở đầu của tài liệu SWS*

Tài liệu SWS cung cấp chi tiết về thông tin chức năng, cấu trúc file, các kiểu dữ liệu, tên và chức năng của từng API, thông tin các tham số cấu hình để thỏa mãn được các yêu cầu trong SRS của module đó. Việc cài đặt phải bám sát theo SWS để đảm bảo được chức năng và yêu cầu của AUTOSAR.

Các phần quan trọng SWS, khi cài đặt module cần phải đọc hiểu:

+ Chương 1: Introduction and functional overview – tổng quan chức năng của module.

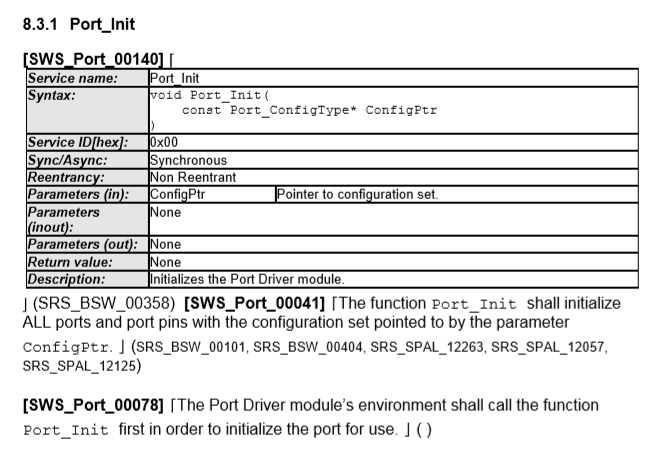
+ Chương 2: Acronyms and abbreviations – giải thích các từ viết tắt.

+ Chương 4: Constraints and assumptions – các lưu ý và một số giới hạn về chức năng của module.

+ Chương 5: Dependencies to other modules – cấu trúc file và mô tả về các module phụ thuộc.

+ Chương 7: Functional specification – chi tiết các chức năng của module.

+ Chương 8: API specification – mô tả các kiểu dữ liệu, cấu trúc các API của module cùng với các yêu cầu cụ thể của chúng. Module phải đảm bảo có đầy đủ các kiểu dữ liệu, API phải theo cú pháp và chức năng phải hoạt động đúng như trong mô tả của từng API. Ví dụ về API Port\_Init



*Hình 9: Yêu cầu của API Port\_Init trong module PORT*

+ Chương 9: Sequence diagrams – hướng dẫn các bước để thực hiện một chức năng của module.

+ Chương 10: Configuration specification – người phát triển ở lớp trên sẽ được cung cấp một công cụ để cấu hình các module trong MCAL mà không phải quan tâm về các thanh ghi, cơ chế cụ thể của Microcontroller. Chương 10 sẽ đưa ra các tham số cần phải có của công cụ đó. Ví dụ: Module PORTcó chức năng khởi tạo các chế độ của chân là GPIO hoặc ADC, PWM … Developer ở lớp trên chỉ cần chọn chức năng mà mình muốn trên chân đó và module PORT phải thực hiện khởi tạo như Developer đã cấu hình.

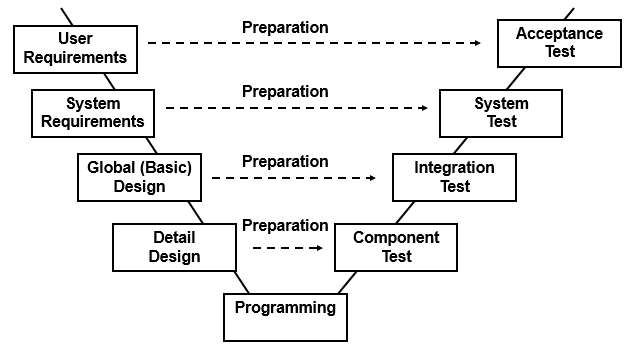
Tương tự như SRS, các yêu cầu trong SWS đều có một mã định danh (SWS\_Port\_00130).

Trong một số trường hợp chức năng của Microcontroller không phù hợp hoặc không hỗ trợ theo chuẩn AUTOSAR thì một số yêu cầu có thể không tuân thủ. Tuy nhiên tất cả phải có sự xác nhận của người sử dụng hoặc Developer ở lớp cao hơn.

## **4.4 Testing**

Testing là giai đoạn bắt buộc trong bất kỳ qui trình phát triển phần mềm nào. Testing nhằm mục đích phát hiện lỗi của phần mềm, xác minh tính năng của phần mềm phải phù hợp với yêu cầu đề ra.

Qui trình test V-model: việc testing sẽ thực hiện song song từ quá trình phân tích yêu cầu đến thiết kế và coding. Qui trình này giúp phát hiện lỗi ngay từ những giai đoạn đầu tiên từ đó giảm được chi phí sửa lỗi, thời gian. Ví dụ như việc phát hiện lỗi design ngay từ đầu sẽ giúp việc coding và testing ít tốn thời gian. Ở mô hình truyền thống như Waterfall, nếu design lỗi sẽ dẫn đến việc coding và test lại từ đầu.



*Hình 10: Mô hình V-model*

Các loại testing:

+ Unit Test (Component Test): Kiểm thử tài nguyên sử dụng, độ bao phủ code cho nội bộ function.

+ Intergration Test: Kiểm thử chức năng của function. Việc testing sẽ không quan tâm đến cấu trúc bên trong của function như thế nào mà chỉ quan tâm input, output. chức năng của nó thực hiện như thế nào.

+ System Test: Tích hợp các function thành một hệ thống hoàn chỉnh và thực hiện test cho hệ thống lớn.

+ Acceptance Test: Giống như System Test nhưng người testing sẽ là khách hàng hoặc người đưa ra yêu cầu cho phần mềm. Ngoài ra Acceptance Test còn bao gồm việc review các document, tất cả các sản phẩm của người phát triển.

Trong AUTOSAR, việc testing cũng bao gồm 4 loại testing trên. Chiến lược test, test case đều được đưa ra dựa theo các định danh (ID) của các yêu cầu SRS và SWS. Ngoại trừ các SRS, SWS ID được chấp thuận có thể không tuân thủ, thì mọi yêu cầu trong AUTOSAR đều phải được test. Khi lập test case, phải chỉ rõ rằng test case đó test cho SRS và SWS ID nào. Với một số SRS và SWS ID không test được bằng Unit Test, Intergration Test, System Test thì phải đảm bảo thông qua review.

Truy xuất nguồn gốc (Traceability) là việc gán các SRS, SWS ID vào từng phần thiết kế, coding và testing để đảm bảo rằng yêu cầu đó đã được cài đặt và testing. Việc đưa các SRS, SWS ID vào trong các test case là một phần của công việc truy xuất nguồn gốc (Traceablility).

# **Chương V. Kết quả đạt được qua đợt thực tập**

## **5.1 Những kiến thức lý thuyết đã được củng cố**

Củng cổ kĩ năng đọc hiểu về Datasheet, Hardware Manual và User Manual của Kit Board thông qua KIT STM8L-Discovery.

Kĩ năng lập trình Hardware, hiểu sâu về GPIO, Timer, Communication (UART, SPI), Interrupt.

Lập trình nhúng với C.

**5.2 Những kỹ năng thực hành đã được học thêm**

Sử dụng SVN và Git.

Tiếp cận với cách lập trình C mới, có tổ chức chương trình, nguyên tắc rõ ràng, đặt tên biến, macro.

Lập trình hệ thống nhúng với board STM8L-Discovery qua những tác vụ với:

+ General-Purpose Input/Output (GPIO).

+ Interrupt.

+ Timer và LCD (Liquid Crystal Display).

+ Hiểu thêm về cách cấu hình bộ tạo xung.

+ Giao thức giao tiếp: SPI (Serial Peripheral Interface), UART (Universal asynchronous receiver-transmitter).

Hệ thống nhúng trong AUTOSAR:

+ Tổng quan về AUTOSAR.

+ Hiểu và khai thác các tài liệu: Specification of Module Driver (AUTOSAR SWS), Requirement of Module Driver (AUTOSAR SRS).

+ Đọc và hiểu về các API (Application programming interface) của các module và chức năng của các hàm trong API

Hiểu về một số quy trình testing.

Hiểu được khái niệm Unit test, Intergration test, System test.

## **5.3 Những kinh nghiệm thực tiễn đã tích lũy được**

Fresher phải chủ động trong việc tìm hiểu và thắc mắc, đặt câu hỏi để hiểu và giải quyết vấn đề.

Các công việc cần được phân chia cụ thể trong nhóm, có thời gian, deadline.

Cách viết Email.

Kỹ năng tự sắp xếp công việc của bản thân.

Ý thức hơn về trách nhiệm của bản thân trong việc hoàn thành công việc.

# **Chương VI. Nhận xét góp ý về Chương trình Đào tạo của Khoa**

Chương trình đào tạo của khoa giúp sinh viên nắm được kiến thức nền tảng, kiến thức lập trình nhúng, các kiến thức về Microcontroller đồng thời giúp sinh viên tiếp cận được với các xu hướng công nghệ mới như IOT, AI, xử lý ảnh …

Chương trình đào tạo của khoa nên mở rộng giúp sinh viên tiếp cận các hệ thống nhúng lớn hơn như các hệ thống IOT, các hệ thống nhúng tích hợp hệ điều hành Linux, RTOS hoặc các hệ thống điều khiển phức tạp, wireless … từ đó giúp sinh viên có hiểu biết để tiếp cận nhanh hơn các hệ thống lớn trong các dự án của doanh nghiệp.

**Tài liệu tham khảo**

[1] AUTOSAR, “Layered Software Architechture”, tháng 11-2016.

[2] AUTOSAR, “Requirements on Port Driver”, tháng 11-2016.

[3] AUTOSAR, “Specification of Port Driver”, tháng 11-2016.