**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**



**BÁO CÁO THỰC TẬP**

**NGHIÊN CỨU CÁC GIẢI PHÁP, SẢN PHẨM VIỄN THÔNG DỰA TRÊN CÔNG NGHỆ VOIP – VOICE OVER INTERNET PROTOCOL**

GVHD: Ngô Hiếu Trường

SV thực hiện: Trần Trọng Huy

MSSV: 20520555

Lớp: MTIO2020

*Tp.Hồ Chí Minh, tháng 07 năm 2024*

LỜI CẢM ƠN

Trong suốt khoảng thời gian 3 tháng thực tập tại Công ty TMA Solutions, từ ngày 08 tháng 04 năm 2024 đến ngày 05 tháng 07 năm 2024, dù thời gian không quá dài nhưng em đã có cơ hội được học hỏi, rèn luyện và tích lũy nhiều kinh nghiệm cũng như nhiều kiến thức quý báu từ các anh chị trong công ty. Đặc biệt là tác phong làm việc chuyên nghiệp và cách giao tiếp với mọi người, cách làm việc nhóm hiệu quả, giúp em vận dụng những kiến thức đã học ở trường vào thực tế.

Em xin gửi lời cảm ơn sâu sắc đến toàn thể quý thầy cô tại trường Đại học Công nghệ Thông tin - Đại học Quốc gia Hồ Chí Minh, đặc biệt là các thầy cô trong Khoa Kỹ thuật Máy tính đã giảng dạy và hỗ trợ em trong suốt quá trình học tập. Em cũng xin cảm ơn thầy Ngô Hiếu Trường đã hướng dẫn, tao điều kiện và hỗ trợ em hoàn thành tốt kỳ thực tập này. Những kiến thức quý báu từ thầy cô không chỉ hữu ích trong thời điểm hiện tại mà còn là nền tảng vững chắc cho tương lai của em.

Em chân thành cảm ơn Công ty TMA Solutions đã tạo điều kiện cho em tham gia kỳ thực tập, giúp em áp dụng kiến thức học được vào thực tế một cách hiệu quả. Đặc biệt, em cảm ơn các anh chị trong team dự án đã nhiệt tình hỗ trợ và chỉ dẫn, giúp em hoàn thành tốt nhiệm vụ được giao và trang bị cho em những kỹ năng cần thiết trong công việc.

Do còn hạn chế về kiến thức và kinh nghiệm, lại lần đầu làm việc trong môi trường công nghiệp, em không tránh khỏi những sai sót. Kính mong thầy cô và anh chị trong công ty góp ý để em có thể hoàn thiện hơn bản thân mình.

Một lần nữa, em xin chân thành cảm ơn.

MỤC LỤC

[CHƯƠNG 1 GIỚI THIỆU VỀ CÔNG TY 1](#_Toc75822559)

[1.1 Tập đoàn Công nghệ TMA (TMA Tech Group) 1](#_Toc75822560)

[1.2 Công ty TMA Solutions 2](#_Toc75822561)

[1.3 Bộ phận thực tập TMA Telecom 3](#_Toc75822562)

[1.4 Cá nhân tham gia thực tập 3](#_Toc75822563)

[1.5 Lịch làm việc nơi thực tập 3](#_Toc75822564)

[1.5.1 Thời gian thực tập 3](#_Toc75822565)

[1.5.2 Công việc được giao 4](#_Toc75822566)

[1.6 Chính sách bảo mật 4](#_Toc75822567)

[CHƯƠNG 2 Nội dung thực tập 5](#_Toc75822568)

[2.1 Công nghệ VoIP 5](#_Toc75822569)

[2.1.1 Tổng quan về VoIP 5](#_Toc75822570)

[2.1.2 Phương pháp kết nối trong VoIP 6](#_Toc75822571)

[2.1.3 Các thành phần của VoIP 7](#_Toc75822572)

[2.1.4 Cách thức hoạt động của VoIP 7](#_Toc75822572)

[2.1.5 Các giao thức trong VoIP 7](#_Toc75822572)

[2.1.5.1 Giao thức H323 7](#_Toc75822572)

[2.1.5.1 Giao thức SIP 7](#_Toc75822572)

[2.2 Mô hình Verification and Validation 10](#_Toc75822573)

[2.2.1 Khái niệm mô hình V-model 10](#_Toc75822574)

[2.2.2 Verification Phases 10](#_Toc75822575)

[2.2.3 Validation Phases 11](#_Toc75822576)

[2.3 Cách thức kiểm thử UT, IT 12](#_Toc75822577)

[2.3.1 Tổng quan về Unit Test 12](#_Toc75822578)

[2.3.2 Tổng quan về Integration Test 14](#_Toc75822579)

[2.3.3 Sử dụng Cantata để chạy UT và IT 15](#_Toc75822580)

[CHƯƠNG 3 KẾT QUẢ 19](#_Toc75822581)

[3.1 Về kiến thức 19](#_Toc75822582)

[3.2 Về kỹ năng 19](#_Toc75822583)

[3.3 Kinh nghiệm tích lũy 19](#_Toc75822584)

[CHƯƠNG 4 KẾT LUẬN 20](#_Toc75822585)

[4.1 Kết luận quá trình thực tập của sinh viên 20](#_Toc75822586)

[4.2 Nhận xét, đóng góp ý kiến về quy trình đào tạo của khoa sau khi sinh viên thực tập tại doanh nghiệp 20](#_Toc75822587)

[TÀI LIỆU THAM KHẢO 21](#_Toc75822588)

DANH MỤC HÌNH ẢNH

[Hình 1.1: Trụ sở tập đoàn FPT tại thành phố Hà Nội 1](#_Toc76072024)

[Hình 1.2: Tòa nhà F-Town 1 và F-Town 2 tại Quận 9, TP. Hồ Chí Minh 2](#_Toc76072025)

[Hình 2.1: Logo của chuẩn Autosar 5](#_Toc76072026)

[Hình 2.2:Tầm nhìn của Autosar 5](#_Toc76072027)

[Hình 2.3: AUTOSAR layer 6](#_Toc76072028)

[Hình 2.4:Chi tiết các tầng của AUTOSAR 7](#_Toc76072029)

[Hình 2.5:AUTOSAR MCAL layer 8](#_Toc76072030)

[Hình 2.6: ECU Abstraction layer 8](#_Toc76072031)

[Hình 2.7: Autosar MCAL lớp services 9](#_Toc76072032)

[Hình 2.8: Autosar Layer lớp Complex Drivers 9](#_Toc76072033)

[Hình 2.9: Verification and Validation model 10](#_Toc76072034)

[Hình 2.10: Các tầng kiểm thử của một hệ thống 12](#_Toc76072035)

[Hình 2.11: Logo Cantata 15](#_Toc76072036)

[Hình 2.12: Giao diện tạo một project mới trên Cantata 16](#_Toc76072037)

[Hình 2.13: Test case được chạy thành công 16](#_Toc76072038)

[Hình 2.14: Test case chạy bị lỗi 17](#_Toc76072039)

[Hình 2.15: Minh họa cho test spec UT 18](#_Toc76072040)

[Hình 2.16: Minh họa cho test spec IT 18](#_Toc76072041)

DANH MỤC BẢNG

[Bảng 1.1: Lịch làm việc, học tập 3](#_Toc75810990)

[Bảng 1.2: Nội dung công việc trong quá trình thực tập 4](#_Toc75810991)

[Bảng 2.1: Bảng mô tả điều kiện trong MC/DC 14](#_Toc75810992)

DANH MỤC TỪ VIẾT TẮT

UT - Unit Test

IT - Integration Test

BSW - Basic Software layer

OEM - Original Equipment Manufacturer

API - Application Programing Interface

ECU - Electronic Control Unit

MCAL - Microcontroller Abstraction Layer

RTE - Runtime Environment layer

MCU - Microcontroller Unit

E/E - Exchangeability / Exchangeability

SDLC - Software Development Life Cycle

HLD - High Level Design

SWS - Software Specification

SRS - Software Requirement Specification

GAM - Global Automotive & Manufacturing

TÓM TẮT

“Báo cáo thực tập” là tài liệu tổng kết về quá trình thực tập trong suốt 13 tuần tại Công ty TMA Solutions, từ ngày 08 tháng 04 năm 2024 đến ngày 05 tháng 07 năm 2024.

Nội dung báo cáo trình bày các nhiệm vụ được giao và kết quả hoàn thành, đồng thời đánh giá khả năng của sinh viên trong quá trình thực tập. Báo cáo cũng nêu rõ cách sinh viên đã áp dụng những kiến thức học được từ trường vào công việc thực tế.

Báo cáo được chia thành 04 chương với các nội dung như sau:

Chương 1: Thông tin tổng quan về đơn vị thực tập.

Chương 2: Mô tả chi tiết các nhiệm vụ và nội dung thực tập.

Chương 3: Kết quả đạt được và những kinh nghiệm học hỏi sau khi hoàn thành công việc.

Chương 4: Kết luận về quá trình thực tập tại công ty.

# GIỚI THIỆU VỀ CÔNG TY

## Tập đoàn Công nghệ TMA (TMA Tech Group)

Tập đoàn Công nghệ TMA (hay còn gọi là TMA Tech Group), với tiền thân là Doanh nghiệp tư nhân dịch vụ Tường Minh được thành lập vào năm 1997 tại TP. Hồ Chí Minh, là một trong những công ty công nghệ hàng đầu cung cấp dịch vụ phần mềm, công nghệ thông tin, và giải pháp công nghệ. Vào năm 2002, TMA thuộc nhóm 15 công ty gia công phần mềm toàn cầu tốt nhất, đồng thời cũng là công ty gia công phần mềm có quy mô lớn nhất Thành phố Hồ Chí Minh. Với 27 năm phát triển cùng với đội ngũ hơn 4,000 kỹ sư và chuyên gia, TMA đã thực hiện hàng ngàn dự án quốc tế và nhận được nhiều giải thưởng uy tín.

TMA Tech Group bao gồm:

* TMA Solutions: Công ty giải pháp phần mềm chuyên về dịch vụ phát triển phần mềm, kiểm thử phần mềm và tư vấn giải pháp công nghệ.
* TMA Solutions Bình Định: Trung tâm công nghệ hàng đầu miền Trung Việt Nam
* TMA Innovation: Cung cấp sản phẩm và giải pháp công nghệ tiên tiến cho nhiều lĩnh vực

TMA Tech Group hiện có 6 chi nhánh tại Việt Nam (5 tại TP Hồ Chí Minh và 1 ở TP Quy Nhơn) cùng với 6 văn phòng ở nước ngoài (Mỹ, Canada, Đức, Nhật, Úc, Singapore). Công ty giúp khách hàng thành công bằng cách cung cấp các nguồn lực phù hợp vào đúng thời điểm, đáp ứng các yêu cầu chất lượng nghiêm ngặt cho các công ty hàng đầu. Ngoài ra, TMA giúp các thương hiệu toàn cầu, doanh nghiệp lớn, doanh nghiệp quy mô vừa và các công ty khởi nghiệp bằng các giải pháp sáng tạo.

## Công ty TMA Solutions

TMA Solutions là công ty thành viên đầu tiên thuộc Tâp đoàn Công nghệ TMA, một trong những công ty tiên phong trong lĩnh vực gia công phần mềm tại Việt Nam, với sứ mệnh "Đưa tên tuổi Việt Nam lên bản đồ gia công phần mềm thế giới."

Lịch sử phát triển của TMA Solutions:

* 1997: Thành lập tại TP. Hồ Chí Minh, với văn phòng đầu tiên đặt tại phòng khách nhà bà Bùi Ngọc Anh và chỉ 6 nhân viên đầu tiên là những sinh viên mới ra trường.
* 1999: Chuyển sang trụ sở mới tại số 111 Nguyễn Đình Chính, quận Phú Nhuận, khi có trên 30 nhân viên.
* 2005: Phát triển lên 4 văn phòng tại quận Phú Nhuận với con số 500 nhân viên.
* 2009: Khánh thành và đi vào hoạt động TMA Tower tại Công viên phần mềm Quang Trung, với diện tích 24,000m², đáp ứng cho khoảng 3,000 kỹ sư làm việc.
* 2020: Giữa đại dịch Covid-19, TMA Solutions tiếp tục vững vàng và đạt được nhiều đột phá ấn tượng.

TMA Solutions chuyên cung cấp các dịch vụ gia công phần mềm chất lượng cao, bao gồm: Phát triển phần mềm tùy chỉnh, Kiểm thử phần mềm, Chuyển đổi số, Tư vấn công nghệ. TMA Solutions không chỉ tập trung vào việc đem lại lợi ích cho đối tác và khách hàng mà còn chú trọng đến việc phát triển nguồn nhân lực, tạo ra môi trường làm việc tích cực và thân thiện, đồng thời tích cực tham gia vào các hoạt động cộng đồng và từ thiện.

## Bộ phận thực tập TMA Telecom

Giới thiệu chung: TMA Telecom là đơn vị chuyên cung cấp các dịch vụ và giải pháp công nghệ thông tin viễn thông (ICT) với hơn 27 năm kinh nghiệm trong ngành. Công ty đã phát triển các ứng dụng viễn thông hàng đầu và cung cấp dịch vụ từ đầu đến cuối cho khách hàng trên toàn thế giới.

Đội ngũ nhân sự: TMA Telecom sở hữu một đội ngũ kỹ sư đông đảo và có chuyên môn cao, luôn sẵn sàng đáp ứng các nhu cầu phức tạp của khách hàng.

Trung tâm nghiên cứu và phát triển (R&D): Công ty có các trung tâm nghiên cứu và phát triển (R&D) dành cho các công ty hàng đầu, nhằm nghiên cứu và triển khai các công nghệ mới nhất.

Thành tựu và dự án tiêu biểu: TMA Telecom đã khẳng định được vị thế của mình trong lĩnh vực viễn thông, phục vụ các khách hàng lớn trên toàn cầu bao gồm các nhà cung cấp dịch vụ viễn thông hàng đầu, các công ty công nghệ, và các tổ chức chính phủ. Một số thành tựu và dự án tiêu biểu của TMA Telecom gồm:

• Dự án mạng 5G

• Hệ thống quản lý dịch vụ viễn thông

Với những thành tựu này, TMA Telecom tiếp tục duy trì và phát triển vị thế của mình trên thị trường quốc tế, mang đến các giải pháp công nghệ viễn thông tiên tiến và đáp ứng tốt nhất nhu cầu của khách hàng.

## Cá nhân tham gia thực tập

* Họ và tên: Trần Trọng Huy
* MSSV: 20520555
* Công ty thực tập: TMA Solutions
* Địa điểm thực tập: Lab 6Tòa nhà TMA, Công viên phần mềm Quang Trung, P. Tân Chánh Hiệp, Quận 12
* Chức vụ: Thực tập sinh

## Lịch làm việc nơi thực tập

### **Thời gian thực tập**

|  |  |  |
| --- | --- | --- |
| **Thời gian** | **Ngày** | **Giờ làm việc** |
| 08/04/2024 - 05/07/2024 | Thứ 2, 3 và 6 | 8h30 - 18h00 |
| Người quản lý: Nguyễn Thị Huyền Trang  Mail: [nthtrang1@tma.com.vn](mailto:nthtrang1@tma.com.vn)  Số điện thoại: [+84-8 3997 8000](callto:+84-8%203997%208000) | | |

Bảng 1.1 Lịch thực tập tại công ty

### **Công việc được giao**

|  |  |
| --- | --- |
| **Thời gian** | **Công việc** |
| Tuần 1  (08/04 - 12/04) | * Tham gia buổi giới thiệu về công ty, nhận thực tâp và thiết bị * Hoàn thành khóa học về Security Awareness (online course + test), * Tìm hiểu các tài liệu và luyện tập sử dụng thành thạo Linux/Red Hat * Tìm hiểu các tài liêu về VMware, bao gồm cách cài đặt, sử dụng, SSH... |
| Tuần 2 – 5  (15/04 - 07/05) | * Luyện tâp lập trình C trên Linux (bao gồm Cấu trúc dữ liệu và Giải thuật). Viết các chương trình một cách tối ưu và hiệu quả. * Tìm hiểu, đọc tài liệu về Socket, Multi-thread và Multi-process * Tìm hiểu vê IPC và Shared Memorry * Tìm hiểu về Shell Script trên Linux * Tham gia khóa học về Software Development Process |
| Tuần 6  (10/05 - 20/05) | * Tìm hiểu về các công cụ phục vụ cho công việc như SIPP, SVN, Redmine, MobaXterm, WinSCP * Tìm hiểu, đọc các tài liệu về Valgrind và GDB. Thực hành sử dụng 2 công cụ trên để nắm được cách sử dụng và cách sửa lỗi * Tham gia khóa học về Team Work Skill |
| Tuần 7 - 9  (21/05 - 10/06) | * Tìm hiểu, đọc các tài liệu về VoIP và NGN * Tìm hiểu về các giao thức TCP/IP và SIP. Đặc biệt là mô hình hoạt động của giao thức SIP. * Tìm hiểu về các giao thức SDP, ISUP và Diameter * Thực hiện dự án được giao với yêu cầu xây dựng Chat Application vận dụng các kiến thức đã được học. * Tham gia khóa học về Email writing & Communication Skills |
| Tuần 10 – 12  (11/06 - 28/06) | * Tìm hiểu về SBC và IMS * Cấu hình và thực hiện cuộc gọi cơ bản thông qua một số module SBC |
| Tuần 13  (01/07 - 05/07) | * Viết báo cáo thực tập * Thuyết trình và demo dự án được giao với mentor |

Bảng 1.2: Nội dung công việc trong quá trình thực tập

## Chính sách bảo mật

Vì là một công ty hoạt động trong lĩnh vực công nghệ thông tin cho nên bảo mật là yếu tố vô cùng quan trọng. Mọi thông tin và tài liệu liên quan đến công ty đều không được tự ý mang về nhà hoặc truyền tải ra ngoài dưới mọi hình thức. Và cũng như tuyệt đối không được tiết lộ cho người ngoài công ty.

Theo chính sách bảo mật của công ty, sinh viên chỉ được phép mô tả sơ lược về các công việc đã thực hiện mà không đi sâu vào chi tiết cụ thể.

# Nội dung thực tập

## Công nghệ VoIP

### **Tổng quan về VoIP**

VoIP, viết tắt của Voice over Internet Protocol, nghĩa là truyền giọng nói qua giao thức Internet, là công nghệ truyền âm thanh qua mạng máy tính sử dụng giao thức TCP/IP. Phương pháp này sử dụng các gói dữ liệu IP trên LAN/WAN/Internet với thông tin được truyền tải dưới dạng mã hóa của âm thanh.

Bản chất của công nghệ VoIP là dựa trên chuyển mạch gói, thay thế phương pháp truyền thống là chuyển mạch kênh. Điều này giúp nén nhiều kênh thoại trên một đường truyền tín hiệu và truyền qua mạng Internet.

VoIP cho phép tạo cuộc gọi sử dụng kết nối băng thông thay vì đường truyền dây điện thoại analog. Công nghệ này cho phép bạn thực hiện cuộc gọi với các số điện thoại nội bộ, đường dài, di động, và quốc tế với chi phí rất thấp. Nguyên tắc hoạt động của VoIP bao gồm mã hóa tín hiệu giọng nói, nén tín hiệu, chia nhỏ các gói tín hiệu, và truyền qua mạng.

Để thực hiện được việc này, các điện thoại IP thường được tích hợp sẵn các giao thức báo hiệu chuẩn như SIP hoặc H.323, kết nối với tổng đài IP (IP PBX) của doanh nghiệp hoặc của nhà cung cấp dịch vụ.

#### **Ưu điểm:**

* Các cuộc gọi sẽ hoàn toàn miễn phí khi sử dụng cùng thiết bị VoIP, cùng dịch vụ hoặc cùng tổng đài VoIP hay còn gọi cách khác là các cuộc gọi nội mạng.
* Tiết kiệm được chi phí đầu tư khi có thể tích hợp được mạng thoại, số liệu và báo hiệu trên cùng một IP
* Việc mở rộng cũng rất dễ dàng, bạn chỉ cần mua thiết bị mới sau đó tiến hành kết nối mạng là có thể sử dụng được.
* Có thể tiến hành chia sẻ dữ liệu hoặc thực hiện cuộc gọi video call.
* Đảm bảo thông tin liên lạc không bị gián đoạn, sử dụng ổn định và ít gặp sự cố.

#### **Nhược điểm:**

* Vì sử dụng qua hệ thống mạng Internet nên việc bảo mật hết sức quan trọng. Nhưng không phải tổ chức nào cũng có thể đáp ựng được vấn đề này.

Kết nối mạng cũng phải đủ mạnh, băng thông lớn giúp duy trì sự ổn định trong việc sử dụng.

### **Phương pháp kết nối trong VoIP**

**Máy tính đến máy tính:** Dịch vụ này cung cấp cuộc gọi qua Internet miễn phí sử dụng phần mềm điện thoại như Skype, Messenger, Zalo,… Đây là dịch vụ VoIP dựa trên phần mềm và cả người gọi lẫn người nhận phải sử dụng máy tính để thực hiện cuộc gọi. Yêu cầu bao gồm: phần mềm điện thoại, card âm thanh và kết nối Internet tốt. Với dịch vụ này, người dùng không thể gọi đến điện thoại cố định hoặc di động, và người nhận phải đang trực tuyến.

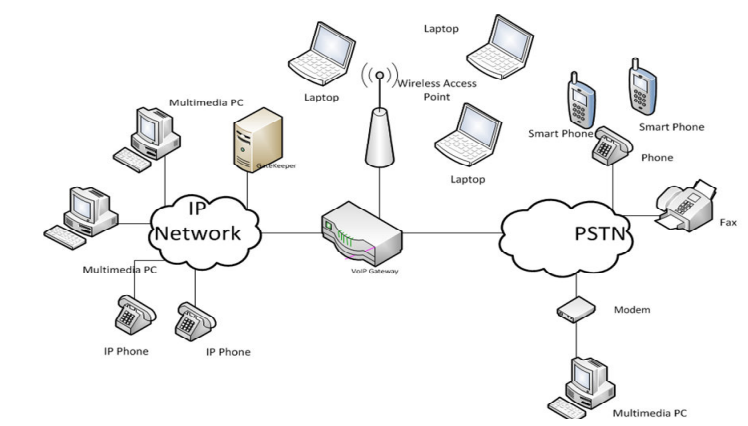
**Máy tính đến điện thoại và ngược lại:** Đây là dịch vụ dựa trên cả phần mềm và phần cứng. Phần mềm điện thoại định tuyến cuộc gọi qua giao thức Internet và chuyển tiếp tới mạng điện thoại truyền thống. Để sử dụng dịch vụ, người dùng cần đăng ký và phải trả phí thấp. Ví dụ như Skype, MSN và Google Talk cho phép khách hàng gọi đến điện thoại cố định từ máy tính. Yêu cầu gồm có: điện thoại và máy tính có kết nối Internet, đăng ký dịch vụ VoIP, modem và bộ chuyển đổi tín hiệu tương tự sang số và ngược lại. Dịch vụ này không cho phép gọi khẩn cấp và cần có máy tính kết nối Internet.

**Điện thoại đến điện thoại:** Đây là dịch vụ dựa trên phần cứng cho phép người gọi và người nhận gọi nhau qua Internet. Nhiều công ty điện thoại sử dụng dịch vụ này để xử lý các cuộc gọi đường dài. VoIP chuyển đổi âm thanh thành các gói dữ liệu và truyền chúng qua Internet. Dịch vụ này cho phép gọi khẩn cấp và không cần mạng PSTN để bắt đầu và kết thúc cuộc gọi.

### **Các thành phần của VoIP**

Các thành phần của VoIP

* Gateway: là thiết bị giúp có thể chuyển đổi tín hiệu analog qua tín hiệu số.
* VoIP gateway: được xem như cầu nối giữa mạng điện thoại PSTN và mạng VoIP
* VoIP sever: là thiết bị định tuyến, bảo mật cho các cuộc gọi voip. Trong mạng H.323 được gọi là gatekeeper.
* Thiết bị đầu cuối (end user equipment): softphone và máy tính gồm headphone, phần mềm, thiết bị kết nối internet.
* Điện thoại truyền thông với IP adapter: để sử dụng dịch vụ VoIP thì điện thoại phải được gắn với thiết bị IP adapter để có thể kết nối được với VoIP server. Adapter là thiết bị có ít nhất một cổng kết nối RJ11- gắn với điện thoại, RJ45 – gắn với đường truyền internet hay PSTN.
* Điện thoại IP: là các dòng điện thoại dùng riêng cho mạng VoIP



### **Cách thức hoạt động của VoIP**

Khi nói vào micro điện thoại, giọng nói sẽ tạo ra tín hiệu điện tử được gọi là tín hiệu analog. Sau đó, những tín hiệu này sẽ được mã hoá thành các tín hiệu số. Đa phần những thiết bị khác nhau sẽ có những thuật toán chuyển đổi khác nhau.

Các bước để thực hiện cuộc gọi trong VoIP:

* Xác định địa điểm như mã quốc gia, mã tỉnh.
* Dữ liệu sẽ được chuyển đổi thông qua kết nối được thiết lập lúc đầu.
* Dữ liệu chứa âm thanh mà bạn nói sẽ được chuyển hoá trở thành âm thanh mà người nghe có thể hiểu được

### **Các giao thức trong VoIP**

#### **Giao thức H323**

Giao thức H.323 được phát triển bởi Liên minh Viễn thông Quốc tế (ITU-T) vào năm 1997, dựa trên giao thức Kiểm soát Vận chuyển Thời gian Thực (RTCP) để truyền tải giọng nói, video và dữ liệu qua mạng dựa trên IP. H.323 cung cấp hội nghị đa phương tiện trên Mạng Cục Bộ (LAN) và kết hợp giao tiếp điểm-điểm và hội nghị đa điểm. Giao thức này được sử dụng rộng rãi vì tính đáng tin cậy và dễ bảo trì.

Khi làm việc với H.323, đây không phải là một giao thức đơn lẻ mà là tập hợp của một nhóm giao thức. Các giao thức riêng trong H.323 bao gồm:

* H.255 – báo hiệu cuộc gọi
* H.245 – điều khiển đa phương tiện
* H.235 – bảo mật và chứng thực
* Q.391 – sử dụng cho tín hiệu cuộc gọi
* T.120 – chia sẻ tệp tin, dữ liệu
* RTP – truyền tải đa phương tiện

Các thành phần cơ bản trong hệ thống mạng H.323 được quy định như sau: thiết bị đầu cuối, cổng kết nối, các thiết bị điều khiển cổng kết nối (gatekeeper) và khối điều khiển đa điểm MCU.

* Terminal thường được xem là phần mềm hoặc phần cứng điện thoại VoIP
* Gateway là một thiết bị cho phép một thông tin giao tiếp hai chiều với các thiết bị trong mạng viễn thông khác.
* MCU là một thiết bị được dùng cho cuộc gọi hội thoại nhiều người. Là thiết bị chịu trách nhiệm cho việc trộn các kênh âm thanh, video trong các cuộc hội thoại.

Terminal, gateway, mcu được gọi chung là các thiết bị đầu cuối. H.323 có thêm một thành phần thứ 4 là gatekeeper. Thiết bị gatekeeper đóng vai trò như một bộ điều khiển trung tâm trong mạng với nhiệm vụ là đăng ký thiết bị đầu cuối gọi vào.

Ảnh có chứa biểu đồ, ảnh chụp màn hình, hàng, thiết kế

Mô tả được tạo tự động

Các giai đoạn trong kết nối âm thanh giữa terminal, gateway và gatekeeper trong giao thức H.323 bao gồm:

1. Đăng ký của terminal và Gatekeeper (H.255-RAS).
2. Định tuyến cuộc gọi giữa terminal và gatekeeper (H.225-RAS và H.225-Q.931).
3. Giao tiếp ban đầu và trao đổi khả năng.
4. Thiết lập giao tiếp âm thanh (mở kênh logic) (H.245).
5. Truyền âm thanh (RTP/RTCP).

H.323 tuân theo các bước sau để thiết lập kết nối giữa các máy khách (Mona &, Nirmala, 2002). Hai điểm cuối sẽ đăng ký với gatekeeper.

1. Terminal A sẽ bắt đầu gửi thông báo trạng thái đăng ký thông qua việc khởi tạo cuộc gọi tới gatekeeper.
2. Thông tin được cung cấp bởi gatekeeper được sử dụng bởi Terminal A để liên lạc với B.
3. Terminal A gửi một thông báo SETUP đến Terminal B.
4. Terminal B gửi một thông báo "Call proceeding" và yêu cầu quyền từ gatekeeper.
5. Terminal B gửi một thông báo "Alerting" và kết nối.
6. Xác định terminal chủ/sẵn sàng và mở các kênh logic bởi Terminal A và B bằng cách sử dụng trao đổi thông điệp H.245.
7. Thiết lập các đường dẫn giao thời thời thực (RTP).

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, máy tính

Mô tả được tạo tự động

#### **Giao thức SIP**

Trước đây, tiêu chuẩn quốc tế H.323 thường được nhắc đến khi nói về VoIP. Nhưng thời gian gần đây thì giao thức SIP được xem là sự thay thế của H.323. Session Initiation Protocol (SIP) là một giao thức được phát triển bởi Internet Engineering Task Force (IETF) nhằm khởi tạo và kết thúc các phiên VoIP với một hoặc nhiều người tham gia. Đây là một giao thức ứng dụng dựa trên ASCII để khởi tạo, thay đổi, tạo ra và kết thúc các phiên truyền thông đa phương tiện tương tác giữa người dùng. SIP tương tự như giao thức truyền tải siêu văn bản (http), bao gồm mô hình client-server. Khách hàng gửi một yêu cầu đến máy chủ và máy chủ xử lý yêu cầu và gửi lại cho khách hàng trong một quá trình gọi là giao dịch. Sử dụng giao thức mô tả người dùng (SDP) để thực hiện thương thảo để xác định codec cho phép hỗ trợ di động người dùng thông qua máy chủ proxy và chuyển hướng đến vị trí hiện tại của người dùng.

SIP bao gồm hai thành phần chính là User Agent (UA) và Network Server. User Agent là điểm kết thúc hành động thay mặt cho người dùng và có thể là máy khách hoặc máy chủ. Máy khách được gọi là user agent client và giúp khởi tạo yêu cầu SIP trong khi máy chủ được gọi là user agent server nhận yêu cầu, xử lý và trả lời yêu cầu thay mặt cho người dùng. Các máy chủ mạng bao gồm máy chủ đăng ký, máy chủ proxy và máy chuyển hướng. Máy chủ đăng ký được sử dụng để tải lên vị trí hiện tại của người dùng, máy chủ proxy nhận yêu cầu và chuyển tiếp nó đến điểm tiếp theo trong khi máy chuyển hướng khi nhận yêu cầu, xác định điểm tiếp theo và trả lại địa chỉ của máy chủ tiếp theo cho khách hàng thay vì chuyển tiếp yêu cầu.

Ảnh có chứa biểu đồ, ảnh chụp màn hình, bản phác thảo, thiết kế

Mô tả được tạo tự động

Luồng cuộc gọi SIP bao gồm sáu bước như sau:

* Đăng ký, khởi tạo và xác định vị trí của người dùng.
* Xác định phương tiện truyền thông.
* Xác định sự sẵn lòng chấp nhận hoặc từ chối cuộc gọi.
* Thiết lập cuộc gọi.
* Sửa đổi và xử lý cuộc gọi.
* Kết thúc cuộc gọi (kết thúc cuộc gọi).

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

SIP định nghĩa một số tin nhắn để giao tiếp với máy khách và máy chủ SIP. Một số tin nhắn được liệt kê như sau:

* INVITE: được sử dụng để khởi tạo một cuộc gọi bằng cách mời người dùng tham gia cuộc gọi phiên SIP.
* ACK: để trao đổi tin nhắn mời một cách đáng tin cậy.
* CANCEL: để hủy một yêu cầu đang chờ của máy khách.
* BYE: cho biết việc kết thúc cuộc gọi đã được khởi tạo.
* REGISTER: cung cấp thông tin về vị trí của người dùng cho máy chủ SIP.
* OPTION: được sử dụng để lấy thông tin về khả năng cuộc gọi.
* INFO: xử lý thông tin ngoài phạm vi.

#### Một số giao thức SIP:

* DUP (User Datagram Protocol): được dùng để vận chuyển bản tin SIP vì đơn giản và thích hợp với các ứng dụng trong thời gian thực.
* TCP (Transmission Control Protocol): có thể vận chuyển nhiều gói tin bất kỳ. Ngoài ra, là giao thức ở tầng vận chuyển có điều khiển tắc nghẽn.
* SDP (Session Description Protocol): dùng để mô tả thông số media, cuộc gọi, các thông số này là thông tin về video, audio, chất lượng băng thông.

## NGN

### **Khái niệm mô hình V-model**

* Software Development Life Cycle (SDLC) là một quá trình theo sau cho một dự án phần mềm, trong một tổ chức phần mềm bao gồm một kế hoạch chi tiết mô tả làm thế nào để phát triển, duy trì, thay đổi hoặc nâng cấp phần mềm.



Hình 2.9: Verification and Validation model

* V-model là một SDLC trong đó việc thực thi các quy trình diễn ra tuần tự theo hình chữ V được biết đến với tên gọi là **Verification and Validation model**.
* V-model dựa trên sự liên kết của mỗi giai đoạn thử nghiệm (testing stage) tương ứng cho mỗi giai đoạn phát triển (development stage). Quá phát triển một hệ thống được thể hiện chi tiết trong hình 2.9.

### **Verification Phases**

* **Business Requirement Analysis:** là giai đoạn đầu tiên trong quy trình phát triển phần mềm. Ở giai đoạn này người phát triển (development) sẽ phải nghiên cứu yêu cầu khách hàng. Từ đó lập ra bản thảo chi tiết các công việc sẽ làm, kế hoạch đặt ra để thực hiện theo đúng yêu cầu.
* **System Design:** Khi đã nắm rõ yêu cầu thì sẽ tiến hành đến quy trình thiết kế hệ thống. Giai đoạn này sẽ dựa vào yêu cầu của khách hàng đồng thời cũng phải dựa vào đặc tính của phần cứng hỗ trợ mà định nghĩa ra các thành phần khác nhau trong hệ thống.
* **Architectural Design:** Ở giai đoạn này chúng ta sẽ phải định nghĩa được tên các hàm, kiểu dữ liệu, cấu trúc, cấu tạo và tổ chức files. Giai đoạn này còn được biết đến với tên gọi High Level Design (HLD).
* **Module design (Detailed Design):** Giai đoạn này quyết định cách thức làm việc của coding vì nó chứa các flowchart quy định luồng xử lý. Đồng thời ở giai đoạn này còn quy định về giá trị trả về, phương thức thông báo có lỗi, …
* **Coding:** là giai đoạn cuối của việc tạo ra sản phẩm (chưa qua kiểm duyệt). Người phát triển sẽ phải dựa vào Detailed Design mà viết ra các firmware, APIs tương ứng tuân thủ các flowchart trước đó đã thiết kế.

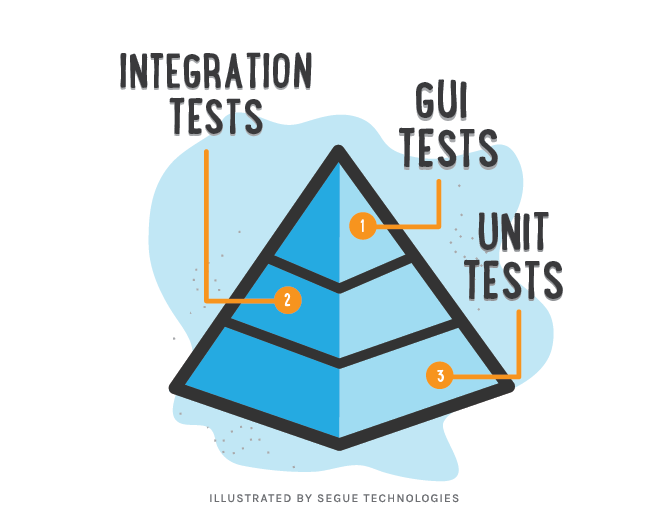
### **Validation Phases**

* **Unit testing:** là đánh giá mức độ bao phủ các trường hợp của source code. Một source code hoàn chỉnh phải có độ phủ gần như 100% cho tất cả các nhánh. Một input phải tương ứng với output mong đợi. Đây là functional testing, chỉ tập trung vào kiểm tra khả năng của hàm trong các trường hợp lý tưởng.
* **Integration testing:** Ở bước này tester phải kiểm tra chức năng các APIs. Người kiểm duyệt phải hiểu được input và output của các APIs từ đó viết ra các test case kiểm tra kết quả. Ở bước này cho ra đánh giá cụ thể khả năng đáp ứng của module và khả năng đáp ứng các yêu cầu đề ra trong SRS của module.
* **System testing:** Ở bước này sẽ kết hợp nhiều module lại với nhau để đảm bảo rằng các resource không bị tranh chấp. Bên cạnh đó yếu tố performance và memory resource cũng sẽ được kiểm tra đánh giá.
* **Acceptance testing:** đây là bước kiểm tra cuối cùng trước khi giao sản phẩm cho khách hàng. Ở bước này cần kiểm tra chức năng của sản phẩm có đáp ứng được yêu cầu khách hàng hay không. Độ tin cậy của sản phẩm được đánh giá dựa vào các test case có được kiểm thử với tất cả trường hợp có thể xảy ra hay không. Bước này được thực hiện với những trường hợp cụ thể do bên khách hàng kiểm thử và báo kết quả.

## Cách thức kiểm thử UT, IT

### **Tổng quan về Unit Test**

* Unit Testing (UT) là một loại kiểm thử phần mềm trong đó các đơn vị hay thành phần riêng lẻ của phần mềm được kiểm thử. Kiểm thử đơn vị được thực hiện trong quá trình phát triển ứng dụng. Mục tiêu của kiểm thử đơn vị là cô lập một phần code và xác minh tính chính xác của đơn vị đó.



Hình 2.10: Các tầng kiểm thử của một hệ thống

* Một Unit là một thành phần PM nhỏ nhất mà ta có thể kiểm tra được như các hàm (Function), thủ tục (Procedure), lớp (Class), hoặc các phương thức (Method).
* Công việc viết Unit Test có thể đem lại những lợi ích sau [2][3]:
* Đảm bảo chất lượng của source code.
* Tìm kiếm và phát hiện các vấn đề tìm ẩn có thể gây ảnh hưởng không tốt đến chương trình
* Giảm chi phí sửa chữa và chi phí kiểm soát chất lượng dự án
* Quá trình phát triển một sản phẩm phần mềm luôn đi cùng với các lỗi phát sinh, nhiệm vụ của các nhà phát triển là phải hạn chế số lượng lỗi và hạn chế chi phí sửa lỗi, lỗi được phát hiện càng sớm thì chi phí sửa lỗi càng ít và ngược lại.
* Test coverage được định nghĩa là một kỹ thuật xác định xem các trường hợp thử nghiệm có thực sự bao trùm mã ứng dụng hay không và bao nhiêu mã được thực hiện khi chạy các trường hợp thử nghiệm đó, có các kỹ thuật Test Coverage phổ biến như sau [4]:
* **C0**: Tỉ lệ bao phủ câu lệnh (Command Coverage Ratio hay Statement Coverage) đảm bảo rằng tất cả các dòng lệnh trong mã nguồn đã được kiểm tra ít nhất một lần. Nó cung cấp các chi tiết của cả hai khối mã được thực thi và thất bại trong tổng số các khối mã.
* **C1**: Tỉ lệ bao phủ nhánh (Branch Coverage). Các nhà phát triển không thể viết mã trong một chế độ liên tục, tại bất kỳ điểm nào họ cần phân nhánh mã để đáp ứng các yêu cầu chức năng. Sự phân nhánh trong mã thực sự là một bước nhảy từ điểm quyết định này sang điểm khác. Branch Coverage kiểm tra mọi đường dẫn có thể hoặc chi nhánh trong mã được kiểm thử.
* **C2**: Tỉ lệ bao phủ điều kiện (Condition Coverage) tập trung vào điều kiện trong nhánh điều kiện, bao phủ test toàn bộ các kết quả điều kiện. Điểm khác với bao phủ nhánh (C1) là: Cho dù nhiều điều kiện bị ràng buộc với nhau bằng AND hay OR, thì vẫn sẽ được coi như các điều kiện độc lập với nhau.
* **MC/DC** (Modified Condition Decision Coverage) là chuẩn đo source về độ bao phủ. Kết quả trả về sau khi kiểm tra xong là True hoặc False, cùng với 1 thông số quan trọng nhất là % độ bao phủ của hàm đó.
* Transaction Level Modeling (TLM) là một phần của SystemC. TLM được sử dụng để mô hình hóa mức Transaction, tập trung vào việc mô hình cách thông tin giữa các process thông qua việc gọi và sử dụng các chức năng có sẵn. Dưới góc nhìn phần cứng, TLM sẽ được sử dụng để mô hình các trao đổi thông tin giữa các khối (block) hoặc module thay vì dùng cách kết nối từng tín hiệu.
* Có ít nhất một test case cho ra kết quả là TRUE nhờ điều kiện TRUE.
* Có ít nhất một test case cho ra kết quả là FALSE nhờ điều kiện FALSE.
* Cách tạo test case MC/DC dựa theo bảng 2.1:
* Xác định giá trị trung lập và giá trị quyết định kết quả.
* Giá trị quyết định kết quả là giá trị sẽ quyết định kết quả là true hay false, nếu nó là true thì kết quả ra là True, nếu nó là false kết quả ra là False.
* Giá trị trung lập là giá trị không ảnh hưởng đến kết quả.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | (A & B) | C |
| 1 | T | T | T | T |
| 2 | T | T | F | T |
| 3 | T | F | T | T |
| 4 | T | F | F | F |
| 5 | F | T | T | T |
| 6 | F | T | F | F |
| 7 | F | F | T | T |
| 8 | F | F | F | F |

Bảng 2.1: Bảng mô tả điều kiện trong MC/DC

### **Tổng quan về Integration Test**

* Kiểm thử tích hợp (Integration testing) [5] là một giai đoạn trong kiểm thử phần mềm. Mỗi module phần mềm riêng biệt được kết hợp lại và kiểm thử theo nhóm.
* Kiểm thử tích hợp xảy ra sau kiểm thử đơn vị (Unit Test) và trước kiểm thử xác nhận. Kiểm thử tích hợp nhận các module đầu vào đã được kiểm thử đơn vị, nhóm chúng vào các tập hợp lớn hơn, áp dụng các ca kiểm thử đã được định nghĩa trong kế hoạch kiểm thử tích hợp vào tập hợp đó, và cung cấp đầu ra cho hệ thống tích hợp.
* Mặc dù mỗi module đều được kiểm thử đơn vị (Unit test) nhưng vẫn có thể tồn tại lỗi do các nguyên nhân sau:
* Một Module nói chung được thiết kế bởi một lập trình viên có hiểu biết và logic lập trình có thể khác với các lập trình viên khác. Kiểm thử tích hợp là cần thiết để đảm bảo tính hợp nhất của phần mềm.
* Tại thời điểm phát triển module vẫn có thể có thay đổi trong spec của khách hàng, những thay đổi này có thể không được kiểm tra ở giai đoạn unit test trước đó.
* Giao diện và cơ sở dữ liệu của các module có thể chưa hoàn chỉnh khi được ghép lại.
* Khi tích hợp hệ thống các module có thể không tương thích với cấu hình chung của hệ thống.
* Thiếu các xử lý ngoại lệ có thể xảy ra.
* Các bước tạo test case IT:
* Chuẩn bị Integration Test Plan dựa trên SRS.
* Thiết kế các kịch bản thử nghiệm, trường hợp, và script.
* Thực hiện kiểm tra theo test case đã viết.
* Theo dõi và tái kiểm tra các lỗi ở trên.

### **Sử dụng Cantata để chạy UT và IT**

* Cantata (hình 2.11) là một đơn vị và công cụ kiểm tra tích hợp, cho phép các nhà phát triển xác minh đoạn mã tiêu chuẩn hoặc đoạn mã quan trọng của doanh nghiệp trên các nền tảng gốc và nhúng của máy chủ.



Hình 2.11: Logo Cantata

* Quá trình testing bao gồm các bước sau [6]:
* Phân tích diagram design và điền giá trị trả về mong muốn và giá trị truyền vào vào trong test spec.
* Diagram design là sơ đồ từ detailed spec để coder có thể dựa vào và viết các dòng lệnh, kiểm thử xem các đoạn lệnh có giống như mô tả hay không.
* Từ test spec sử dụng công cụ do đội ngũ nhân viên công ty viết ra để xuất ra file chứa các test case.
* Tất cả các test case để kiểm thử cho cùng một đoạn lệnh sẽ được xuất ra trong một file để dễ dàng quản lý cũng như nhập vào Cantata.
* Khởi tạo một project mới trên Cantata (hình 2.12), truyền mã cần test cùng với các test case. Chạy các đoạn mã cùng test case, Cantata sẽ trả về kết quả test case đó được thông qua hay có bất kỳ lỗi nào không.

Graphical user interface, application

Description automatically generated

Hình 2.12: Giao diện tạo một project mới trên Cantata

* Nếu các test case được chạy thành công (hình 2.13), tester tổng hợp các config lại với nhau và sử dụng công cụ để tạo báo cáo.

Table

Description automatically generated

Hình 2.13: Test case được chạy thành công

* Nếu các test case bị lỗi (hình 2.14) hoặc trong báo cáo trả về thiếu trường hợp kiểm tra, tester quay lại một trong các bước trên để sửa lỗi hoặc thêm test case mới.

Table

Description automatically generated

Hình 2.14: Test case chạy bị lỗi

* Để biết xảy ra lỗi gì thì tester mở tệp tin mà Cantata đã xuất ra, sau đó đối chiếu các giá trị trong test case.
* Để biết cần tạo mới test case nào thì tester cần đọc hiểu được lệnh của hàm đang test và theo dõi luồng chạy của chúng để xem bị thiếu ở đâu.
* Sau khi thêm test case để hoàn tất kiểm thử đoạn lệnh, tester nên thêm vào test spec ngay để những người sau có thể nắm được toàn bộ quá trình và dễ dàng đánh giá lại kết quả.
* Sau khi hoàn tất việc chạy các test case, tester sử dụng công cụ để tạo báo cáo, báo cáo này thể hiện độ bao phủ của việc kiểm thử, nếu không hoàn thiện thì cần xem lại để xác định nguyên nhân và sửa chữa hoặc ghi nhận lại.

Graphical user interface, text, application, email

Description automatically generated

Hình 2.15: Minh họa cho test spec UT

Graphical user interface, text, application, email

Description automatically generated

Hình 2.16: Minh họa cho test spec IT

Mỗi loại test khác nhau yêu cầu các test spec khác nhau, ta có thể thấy mẫu test spec cho UT như hình 2.15 và IT như hình 2.16.

# KẾT QUẢ

## Về kiến thức

* Cải thiện khả năng lập trình C/C++.
* Quy trình để phát triển một phần mềm.
* Công cụ hỗ trợ kiểm tra và sửa lỗi (Valgrind, GDB)
* Các thành phần, chức năng và mô hình hoạt động của VoIP.

## Về kỹ năng

* Nâng cao khả năng đọc và viết bằng tiếng Anh: Vì tất cả văn bản như tài liệu kỹ thuật, báo cáo viết hằng ngày, thông báo từ công ty, email, ... đều viết bằng tiếng Anh. Vì vậy, điều đó đòi hỏi nhân viên phải rèn luyện kỹ năng đọc và viết thật tốt.
* Bên cạnh đó cũng cải thiện được kỹ năng nghe, nói tiếng Anh thông qua các buổi thuyết trình, chia sẻ của mỗi nhân viên về những kiến thức thú vị mà họ có được trong quá trình làm việc và trong cuộc sống.
* Học kỹ năng viết báo cáo, viết email: do tính chất công việc thì mỗi ngày nhân viên đều phải thực hiện công việc này. Từ đó, sinh viên học được cách viết sao cho ngắn gọn, súc tích nhưng vẫn truyền đạt đầy đủ ý.

## Kinh nghiệm tích lũy

* Làm quen và dần thích nghi với môi trường làm việc chuyên nghiệp ở công ty và tác phong làm việc như một kỹ sư thực thụ.
* Cải thiện khả năng làm việc nhóm lẫn khả năng làm việc độc lập.
* Quản lý thời gian: sắp xếp thời gian làm việc một cách hợp lý để luôn đảm bảo kiểm soát được tiến độ công việc và hoàn thành nhiệm vụ đúng thời hạn.

# KẾT LUẬN

## Kết luận quá trình thực tập của sinh viên

Quá trình thực tập ở công ty đã giúp sinh viên áp dụng những kiến thức học được trong công việc, biết được nhiều kiến thức chuyên ngành khác có ích cho công việc sau này. Đồng thời, tác phong làm việc và những yêu cầu dành cho bản thân khi làm việc trong một môi trường chuyên nghiệp là những điều quan trọng khác mà sinh viên đã nhận thức được.

Do thời gian thực hiện báo cáo có hạn cùng quy định về việc tiết lộ thông tin trong hợp đồng với công ty, sinh viên chi giới thiệu các nội dung cơ bản, một số nội dung chi tiết về hoạt động của công ty đã không được trình bày trong bài viết này.

## Nhận xét, đóng góp ý kiến về quy trình đào tạo của khoa sau khi sinh viên thực tập tại doanh nghiệp

Qua 13 tuần thực tập tại công ty TNHH phần mềm FPT sinh viên nhận thấy các môn học được giảng dạy ở Khoa là phù hợp với nhu cầu kiến thức cần có trong việc làm của doanh nghiệp, cách dạy các môn song song giữa 02 chuyên ngành “Thiết kế hệ thống nhúng và Robotics” và “Thiết kế vi mạch” đảm bảo cho sinh viên có đủ kiến thức trong ngành Kỹ thuật Máy tính về tư duy thiết kế các mạch điện tử và lập trình điều khiển tự động các thiết bị.

Sinh viên cũng mong rằng Khoa trang bị thêm kiến thức về việc viết các công cụ phần mềm đơn giản mà hiệu quả để thiết kế và kiểm thử mạch điện tử và hệ thống nhúng nhanh hơn, hiệu quả hơn cũng như giúp sinh viên có thêm kiến thức thực tế bám sát hơn với doanh nghiệp.

Chương trình đào tạo của Khoa là phù hợp với nhu cầu của công ty. Việc thực hiện đồ án tương ứng với mỗi môn học giúp cho sinh viên sẽ không quá bỡ ngỡ với môi trường làm việc thực tế tại công ty. Tuy nhiên sinh viên nhận thấy vẫn còn nhiều bạn bị bỡ ngỡ, chưa có định hướng về việc giải quyết các bài toán đặt ra trong mỗi đồ án được giao. Vì vậy sinh viên có ý kiến là Khoa nên thường xuyên tổ chức những buổi seminar định kỳ để truyền đạt cho những bạn sinh viên năm nhất các kỹ năng phân tích, tìm hướng giải quyết những vấn đề gặp phải trong quá trình làm đồ án của mình.

# TÀI LIỆU THAM KHẢO

1. AUTOSAR, “Classic platform”, Nov 30,2020. [Online]. Available: <https://www.autosar.org/standards/classic-platform/>
2. Sergey Kolodiy, “Unit Tests, How to Write Testable Code and Why it Matters”, 2016. [Online]. Available: <https://www.toptal.com/qa/how-to-write-testable-code-and-why-it-matters>
3. Guru99, “Unit Testing Tutorial: What is, Types, Tools & Test EXAMPLE”, Unknown. [Online]. Available: https://www.guru99.com/unit-testing-guide.html
4. Segue Technologies, “How Much Test Coverage Is Enough For Your Testing Strategy?”, June 13, 2014. [Online]. Available: <https://www.seguetech.com/how-much-test-coverage-enough-testing-strategy/>
5. Guru99, “Integration Testing: What is, Types, Top Down & Bottom Up Example”, Unknown. [Online]. Available: https://www.guru99.com/integration-testing.html
6. QA Systems, “Automated unit & integration testing for C & C++”, Unknown. [Online]. Available: <https://www.qa-systems.com/tools/cantata/>