

การเชื่อมต่ออินเทอร์เน็ตเข้ากับสภาพแวดล้อมรอบตัว:
ระบบแสดงผลการเฝ้าดูและแจ้งเตือนของระบบดูแลและการจัดการเรียนพยาบาล

นางสาวธนพร เดชวิจิตรเลิศ

นางสาวธัญรดา เครือรัตน์ชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2557

Internet of Things: Monitoring and Alert System of
Nurse Call & Care Management Systems

Ms. Tanaporn Dechavichitlert

Ms. Thanrada Kuerattanachai

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF BACHELOR OF COMPUTER ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK
ACADEMIC YEAR 2014

ปริญญานิพนธ์เรื่อง : การเชื่อมต่ออินเทอร์เน็ตเข้ากับสภาพแวดล้อมรอบตัว:
ระบบแสดงผลการเฝ้าดูและแจ้งเตือนของระบบดูแลและการจัดการ
เรียกพยาบาล
ชื่อ : นางสาวชนพร เดชาวิชิตเลิศ
นางสาวธัญรดา เครือรัตน์ชัย
สาขาวิชา : วิศวกรรมคอมพิวเตอร์
ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะ : วิศวกรรมศาสตร์
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร. วรา วราวิทย์
ปีการศึกษา : 2557

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ อนุมัติให้
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์

..... หัวหน้าภาควิชาวิศวกรรมไฟฟ้า
(ผู้ช่วยศาสตราจารย์ ดร. นกมล วิวัชรโกเศศ) และคอมพิวเตอร์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร. วรา วราวิทย์)

..... กรรมการ
(รองศาสตราจารย์ ดร. ณชล ไชยรัตน์นะ)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. วรรณ วนษ์เสรี)

ลิขสิทธิ์ของภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

Project Report Title : Internet of Things: Monitoring and Alert System of
Nurse Call & Care Management Systems

Name : Ms. Tanaporn Dechavichitlert
Ms. Thanrada Kuerattanachai

Major Field : Computer Engineering


Department : Electrical and Computer Engineering

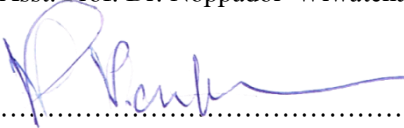
Faculty : Engineering


Project Advisor(s) : Assoc. Prof. Dr. Vara Varavithya


Academic Year : 2014

Accepted by the Faculty of Engineering, King Mongkut's University of Technology North
Bangkok in Partial Fulfillment of the Requirements for the Degree of Bachelor of Computer
Engineering


..... Chairperson of Department of Electrical
(Asst. Prof. Dr. Noppadol Wiwatcharagoses) and Computer Engineering


..... Chairperson
(Assoc. Prof. Dr. Vara Varavithya)


..... Member
(Assoc. Prof. Dr. Nachol Chaiyaratana)


..... Member
(Asst. Prof. Dr. Waranyu Wongseree)

Copyright of the Department of Electrical and Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology North Bangkok

บทคัดย่อ

สถานพยาบาลมีระบบที่สำคัญระบบหนึ่ง ได้แก่ ระบบเรียกพยาบาล ในห้องผู้ป่วยจะมีสวิตช์ตามตำแหน่งต่าง ๆ ที่เหมาะสม จากนั้นเมื่อมีการเรียกพยาบาล สัญญาณเรียกพยาบาลจะมาปรากฏบนเครื่องเรียกพยาบาลที่สถานีเรียกพยาบาลที่ประจำตามหอผู้ป่วย ซึ่งระบบลักษณะนี้เป็นระบบธรรมดาที่ใช้อยู่ทั่วไป ในปัจจุบันการเรียกพยาบาลนั้นได้ทำผ่านระบบสวิตช์ ระบบโทรศัพท์และไฟแจ้งเตือน ซึ่งการแจ้งเตือนในลักษณะนี้อาจทำให้เกิดความล่าช้าในการดูแล ผู้ใช้จึงมีความต้องการจะให้มีการแจ้งเตือนผ่านจอต่างๆ ที่มีราคาถูกลงและมีการใช้งานอย่างแพร่หลายในปัจจุบัน เพื่อให้เกิดความสะดวกมากยิ่งขึ้น ในงานก่อนหน้านี้นี้มีการพัฒนาต่อเชื่อมกับระบบเรียกพยาบาลแล้ว โดยผ่านทางพอร์ต RS485 อย่างไรก็ตามการเชื่อมต่อระบบเรียกพยาบาลผ่านพอร์ต RS485 ยังมีข้อจำกัดในการทำงาน คือ ไม่สามารถรองรับการเรียกพยาบาลพร้อมกันจากหลายๆห้องได้ และนอกจากนี้ระบบที่ได้มีการพัฒนาไว้ก่อนหน้านี้ได้มีการใช้ Web Service ซึ่งจะมีข้อจำกัดเรื่องเวลาในการตอบสนองค่อนข้างช้า และมีข้อจำกัดในการพัฒนาต่อยอด เนื่องจากมีช่องการเชื่อมต่ออีกแบบหนึ่ง ในโครงการนี้เราจึงได้พัฒนาต่อยอดเป็นรูปแบบการเชื่อมต่อผ่านพอร์ต RS232 ซึ่งสามารถรองรับการเรียกพยาบาลจากหลายๆห้องได้พร้อมกัน และได้พัฒนาต่อยอดระบบการเชื่อมต่อสำหรับแสดงผลแบบใหม่ คือ WAMP (The Web Application Messaging Protocol) ที่สามารถประหยัดการใช้ทรัพยากรของตัวประมวลผล ทำงานได้แบบ Real time และยังสามารถรองรับการพัฒนาต่อยอดได้อย่างกว้างขวาง

Abstract

In hospital operations, the nurse call system is commonly deployed in patient wards. A patient ward consists of a set of rooms and a nurse station. There are several occasions that a patient needs care or medical assistance. Therefore, quite a few types of switches is installed to each room. The patient can activate the switch to notify the nurse where a central call management machine at the nurse station create alarm. Examples of notifying devices include switches, telephone, and notified light. With the commodity TV display, we can develop a digital signage to improve the notification services in which the ward map and associated alarms are displayed on the screen. In the past project, the digital signage for nurse call was developed using RS485 communication on the BCRC nurse call system. However, the information from RS485 do not support multiple calls from different rooms. In this project, we extends the previous works with additional functionality and architecture changes. First, the RS232 interface was used to support multiple call functionality. The associated software subsystem for RS232 was developed and a user interface was revamped. Secondly, we refactored the architecture by adopting the Web Application Messaging Protocol to serve as information exchanges between the processing subsystems. The crossbar I/O was responsible for routing events. The prototype was created, tested, and delivered to the industrial partner for further development.

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จเรียบร้อยได้ด้วยรับความอนุเคราะห์บุคคลหลายท่าน ซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด

ผู้มีพระคุณท่านแรกที่คุณศึกษาใคร่ขอกราบพระคุณคือ รองศาสตราจารย์ ดร. วรา วราวิทย์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้สั่งสอนความรู้ ให้คำปรึกษา คำแนะนำ ตลอดจนช่วยเหลือไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่เป็นอย่างดี

ขอขอบพระคุณบริษัท กวินไคนามิค จำกัด และ นายวัฒนา หินไชยศรี ที่กรุณาเอื้อเฟื้ออำนวยความสะดวกในด้านอุปกรณ์สำหรับการทดลอง พร้อมทั้งให้คำแนะนำ ช่วยเหลือ จนปริญญานิพนธ์ชิ้นนี้เสร็จสมบูรณ์ได้

ขอขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ได้ประสิทธิ์ประสาทวิชาความรู้ต่าง ๆ เพื่อนำมาประยุกต์ใช้ในการคิด จัดทำวิทยานิพนธ์นี้จนเสร็จสิ้น รวมไปถึงสถาบันการศึกษา มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ที่เอื้อเฟื้ออุปกรณ์และสถานที่ ให้มีโอกาสดูแลศึกษา หาความรู้ ปฏิบัติทดลอง รวมไปถึงพบเจอประสบการณ์ต่าง ๆ ที่มีค่ามากมาย

สุดท้ายนี้คณะผู้จัดทำกราบขอบพระคุณบิดา มารดา และครอบครัว ตลอดจนซาบซึ้งกำลังใจ ความช่วยเหลือต่าง ๆ ที่ได้รับจากพี่ ๆ เพื่อน ๆ น้อง ๆ ทุกคน ซึ่งเป็นส่วนหนึ่งที่ช่วยผลักดันให้ปริญญานิพนธ์ชิ้นนี้สำเร็จลุล่วงได้ด้วยดี

ธนพร เดชาวิจิตรเลิศ

ธัญรดา เกื้อรัตนชัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	จ
บทคัดย่อภาษาอังกฤษ	ฉ
กิตติกรรมประกาศ	ช
สารบัญภาพ	ฌ
บทที่ 1. บทนำ	1
บทที่ 2. ระบบเรียกพยาบาล	5
2.1. ระบบเรียกพยาบาล	5
2.2. บอร์ด Raspberry Pi	10
2.3. การสื่อสารข้อมูลแบบ UART	11
2.4. ระบบบัส RS232	12
2.5. ซอฟต์แวร์ที่ใช้พัฒนาระบบ	13
บทที่ 3. ระบบแสดงผลแจ้งเตือนระบบเรียกพยาบาล	22
3.1. สถาปัตยกรรมของระบบ	23
3.2. ติดตั้งซอฟต์แวร์สำหรับงานบนบอร์ด Raspberry Pi ซึ่งประกอบด้วย	31
3.3. จัดทำระบบ/เขียนโปรแกรมสำหรับทำงาน	32
บทที่ 4. ผลการดำเนินงาน	34
4.1. Master Node	35
4.2. Display Node	37
4.3. ภาพแสดงผลการดำเนินงาน ที่ได้จากการพัฒนาระบบ	39
บทที่ 5. สรุป	44
เอกสารอ้างอิง	46
ประวัติผู้แต่ง	48

สารบัญภาพ

ภาพที่	หน้า
2-1 ชุดควบคุม (Master Station)	6
2-2 ชุดเรียกพยาบาลติดตั้งที่หัวเตียง	6
2-3 สายกดเรียกพยาบาล (Call Cord)	7
2-4 สวิตช์ดึงเรียกพยาบาลในห้องน้ำ (Emergency Switch)	7
2-5 โคมไฟหน้าห้อง (Corridor Light)	8
2-6 เครื่อง BCRC	9
2-7 ระบบของเครื่อง BCRC	9
2-8 Patient's Call Report Function	10
2-9 ภาพแสดงการส่งข้อมูลแบบ UART	12
2-10 ระดับสัญญาณของ RS232	13
2-11 ภาพแสดงการทำงานแบบ Remote procedure calls ในแบบของ Client-Server model	14
2-12 ภาพแสดงการส่งข้อมูลจาก Publisher และ Subscriber ผ่าน Broker ("topic")	15
2-13 ภาพการส่งข้อมูล RPC ในรูปแบบ Dealer model	16
2-14 เราต์เตอร์คือการรวมกันระหว่าง Broker กับ Dealer	16
2-15 การ register	17
2-16 แสดงถึงการที่ component หนึ่งทำการ call ไปหาอีก component หนึ่งผ่าน crossbar.io และ component ที่ register นั้นทำการส่งข้อมูลกลับไป	18
2-17 แสดงถึงการ subscribe ไปที่ topic A (สมมติ)	18
2-18 แสดงภาพการทำงานขณะที่ component publish topic A แล้ว component ที่ทำการ subscribe ที่ topic A ไว้ก็จะได้รับ event (ข้อมูล) จาก crossbar.io	19
3-1 โครงสร้างของระบบ	23
3-2 สถาปัตยกรรมซอฟต์แวร์ของ Master Node	24
3-3 สถาปัตยกรรมซอฟต์แวร์ของ Display Node	24

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3-4 ตัวอย่างหน้าจอการแสดงผล	25
3-5 ตัวอย่างหน้าจอการแสดงผลเมื่อมี General Call เข้ามา	26
3-6 ตัวอย่างหน้าจอการแสดงผลเมื่อมี General Call เข้ามา 2 ห้อง	26
3-7 ตัวอย่างหน้าจอการแสดงผลเมื่อมี Emergency Call และ General call	27
3-8 ตารางเก็บข้อมูลของตึก	27
3-9 ตารางเก็บข้อมูลเตียง	28
3-10 ตารางเก็บข้อมูลห้อง	28
3-11 ตารางคนไข้	29
3-12 ตารางสำหรับเก็บข้อมูลคิที่เกิดขึ้นจากการ call จากเครื่อง BCRC	29
3-13 ตารางสำหรับเก็บข้อมูลในการทำ report	30
3-14 ตารางเก็บข้อมูลในเวิร์คพยาบาล	30
3-15 ตารางเก็บข้อมูลสำหรับใช้แสดงผล	30
4-1 ภาพตัวอย่างการทำงาน เมื่อเริ่มให้โปรแกรมเริ่มทำงาน	35
4-2 ภาพตัวอย่างเมื่อเปิดเครื่อง BCRC	35
4-3 ภาพตัวอย่างเมื่อมีการกดเรียกพยาบาลจากข้างเตียงผู้ป่วย (General Call) จากห้อง 005	35
4-4 ภาพตัวอย่างเมื่อมีการกดรีเซ็ตการเรียกพยาบาลจากข้างเตียงผู้ป่วยห้อง 005	36
4-5 ภาพตัวอย่างเมื่อมีการกดการเรียกพยาบาลจากห้องน้ำ (Emergency Call) และมีการกดรีเซ็ตจากห้อง 005	36
4-6 ภาพตัวอย่างเมื่อมีการกดการเรียกพยาบาลจากเตียงผู้ป่วย (General Call) และมีการขหนูเพื่อคุยกับผู้ป่วยจากห้อง 005	36
4-7 ภาพตัวอย่างผลการส่งข้อมูลไปลง database	36
4-8 แสดงตัวอย่างโค้ดอย่างง่ายที่ใช้ในการเขียน Kivy	38
4-9 ภาพขณะเปิดหน้าจอครั้งแรก โดยยังไม่ได้ทำการต่อเชื่อมรับข้อมูลจาก Master node	39

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4-10 ภาพตัวอย่างหน้าจอขณะรอการเชื่อมต่อกับ Master node	39
4-11 ภาพหลังจากเชื่อมต่อกับ Master node สำเร็จ	40
4-12 ภาพตัวอย่างที่มีการเรียก General Call จากห้อง 005	40
4-13 ภาพตัวอย่างที่มีการเรียก General Call จากห้อง 005 และ	41
4-14 ภาพตัวอย่างที่มีการเรียก Emergency Call จากห้อง 001	41
4-15 ภาพตัวอย่างที่มีการเรียก Emergency Call จากห้อง 001 และ	42
4-16 ภาพตัวอย่างที่มีการเรียก General Call จากห้อง 005 และ	42
4-17 ภาพหลังจากหลุดจากการเชื่อมต่อกับ Master node	43

บทที่ 1

บทนำ

ปัจจุบันเทคโนโลยีมีความเกี่ยวข้องใกล้ชิดกับชีวิตของคนเรามากขึ้นในหลาย ๆ แง่มุม เช่น การเชื่อมต่อกันระหว่างอุปกรณ์กับเครื่องใช้ไฟฟ้า การชำระสินค้าผ่านบัตรเครดิตหรือบัตรเงินสดแบบแตะสัมผัสกับเครื่องอ่านบัตร ระบบรักษาความปลอดภัย หรือแม้แต่รถยนต์ที่สามารถช่วยในการนำทางเมื่อผู้ใช้งานไม่ทราบทิศทางที่จะไปยังจุดหมาย ดังนั้นจะเห็นได้ว่าสิ่งหรืออุปกรณ์เหล่านี้มีความฉลาดมากขึ้นเรื่อย ๆ เพราะนอกจากจะสามารถอำนวยความสะดวกเร็วในการทำงานหรือเป็นตัวช่วยทุ่นเวลาในชีวิตประจำวันตามความสามารถทางกายภาพของตัวเองแล้ว ยังสามารถเชื่อมต่อกับอินเทอร์เน็ต (Internet) กับระบบภายนอก สามารถคิดคำนวณหรือคาดการณ์เรื่องต่าง ๆ โดยใช้ข้อมูลที่เก็บได้จากสิ่งแวดล้อมและพฤติกรรมของผู้ใช้งานอุปกรณ์นั้น ๆ และทำการตัดสินใจ หรือแนะนำการกระทำที่เหมาะสมให้กับผู้ใช้งานได้อย่างมีประสิทธิภาพ

การนำเทคโนโลยีและระบบสารสนเทศ มาประยุกต์ใช้ในงานต่าง ๆ ของโรงพยาบาล ทำให้สามารถรวบรวมและจัดเก็บข้อมูลจากแหล่งข้อมูลต่าง ๆ ทั้งภายในและภายนอกโรงพยาบาลอย่างมีประสิทธิภาพ ตามกฎระเบียบ ข้อบังคับและมาตรฐานของระบบรับรองคุณภาพต่าง ๆ จึงมีความสำคัญเป็นอย่างยิ่ง [2] โดยมีวัตถุประสงค์นำมาประกอบผลและจัดรูปแบบให้ได้สารสนเทศที่ช่วยสนับสนุนการทำงาน และการตัดสินใจในด้านต่าง ๆ ของผู้บริหาร เพื่อให้การดำเนินงานของโรงพยาบาลเป็นไปอย่างมีประสิทธิภาพ และถูกต้องเที่ยงตรงในทุกขั้นตอนก่อนให้เกิดการใช้ทรัพยากรอย่างเกิดประโยชน์สูงสุด โดยการทำงานของระบบจะเป็นการทำงานแบบ Interactive คือ จัดการข้อมูลที่เกิดขึ้นจากแต่ละหน่วยงาน เช่น แผนกผู้ป่วยนอก ห้องผ่าตัด ฯลฯ และสามารถรวบรวมข้อมูลต่าง ๆ ที่เกิดขึ้นทันทีจากแต่ละหน่วยงานมาใช้ประมวลผลได้ทันทีทันใด ซึ่งจะทำให้บุคลากรปฏิบัติงานได้สะดวกและรวดเร็ว ทำให้มีเวลาในการให้บริการแก่ผู้ป่วยมากขึ้น มีเวลาพัฒนาคุณภาพบริการให้ดีขึ้น

ระบบสารสนเทศที่ใช้ในโรงพยาบาลสามารถจำแนกตามลักษณะงาน [3] จำแนกได้เป็น 2 แบบ คือ 1. ระบบงานบริการผู้ป่วย (Front Office) และ 2. ระบบงานบริหารจัดการที่ไม่เกี่ยวกับงานบริการ (Back Office)

ตัวอย่างของระบบงานบริการผู้ป่วย (Front Office) ได้แก่ ระบบเวชระเบียนอิเล็กทรอนิกส์ (Electronic Medical Records หรือ Electronic Health Records), ระบบสารสนเทศโรงพยาบาล (Hospital Information System) หรือระบบสารสนเทศทางคลินิก (Clinical Information System) และยังรวมไปถึงระบบงานย่อย ๆ ของหน่วยบริการภายในโรงพยาบาล เป็นต้น

ในส่วน of ระบบงานบริหารจัดการที่ไม่เกี่ยวกับงานบริการ (Back Office) ได้แก่ ระบบสารสนเทศเพื่อการบริหาร (Management Information Systems หรือ MIS), ระบบบริหารจัดการทรัพยากรองค์กร (Enterprise Resource Planning หรือ ERP), ระบบสารสนเทศการวิจัยและการศึกษา, เว็บไซต์ และอินทราเน็ตภายในองค์กร และระบบงานสารบรรณ (การเวียนเอกสาร) เป็นต้น

ระบบงานพยาบาลยังมีการนำสารสนเทศมาใช้ด้วยเช่นกัน ซึ่งในที่นี้จะมุ่งเน้นการพัฒนา ระบบเรียกพยาบาลเพื่อช่วยพัฒนาและปรับปรุงประสิทธิภาพการดูแล อำนาจความสะดวกให้แก่ ผู้ป่วยและผู้ที่เกี่ยวข้องกับผู้ป่วย ระบบพยาบาลเดิมที่เป็นสิ่งอำนวยความสะดวกของผู้ป่วยในกรณีฉุกเฉิน โดยทั่วไปเมื่อผู้ป่วยมีความต้องการในสิ่งใด สามารถกดที่ปุ่มขอความช่วยเหลือซึ่งจะส่ง สัญญาณหรือกริ่งไปที่ตัวควบคุม ที่ติดตั้งอยู่ที่เคาน์เตอร์พยาบาลหรือห้องพักพยาบาล เพื่อแจ้ง พยาบาลในการดำเนินการช่วยเหลืออย่างรวดเร็ว หลักการทำงานของระบบเรียกพยาบาลนั้น ผู้ป่วย จะต้องกดสวิทช์ที่ข้างเตียงผู้ป่วย หรือสวิทช์ในห้องน้ำจากนั้นระบบเรียกพยาบาล จะมีการแสดงผล สัญญาณที่ห้องพักพยาบาล โดยมีลักษณะเป็นสัญญาณไฟประจำห้องและสัญญาณเสียงเตือน การ แสดงผลลักษณะนี้ทำให้เจ้าหน้าที่พยาบาลจำเป็นต้องไปยังห้องพักหรือเตียงผู้ป่วยเพื่อตรวจสอบว่า มีเหตุการณ์อะไรเกิดขึ้น แล้วจึงทำการช่วยเหลือ ระบบเรียกพยาบาลใน โรงพยาบาลที่มีใช้กันอยู่ ทั่วไปมีราคาที่สูงหลายขึ้นอยู่กับการประสิทธิภาพหรือรูปแบบการใช้งาน และความต้องการของ โรงพยาบาลนั้น ๆ [4]

การเพิ่มประสิทธิภาพในการให้บริการแก่ผู้ป่วยที่พักรักษาตัวอยู่หอพักผู้ป่วย ระบบเรียก พยาบาลที่นำมาใช้พัฒนาในปริญญานิพนธ์ (เครื่อง BCRC) จะนำมาต่อยอดในเรื่องของการ

แสดงผลออกทางหน้าจอเพื่ออำนวยความสะดวกแก่พยาบาลผู้ปฏิบัติงาน โดยจะแสดงจุดตำแหน่งของห้องผู้ป่วยที่มีการเรียกสายเข้ามา และข้อมูลเบื้องต้นของผู้ป่วย

ระบบการเรียกพยาบาลที่ถูกพัฒนาไว้แล้วนั้น มีการใช้งานชุดควบคุมกับหน่วยประมวลผล โดยสื่อสารกันผ่านระบบบัส RS 485 (Half-Duplex) [5] ซึ่งจะมีข้อจำกัดที่สำคัญในแง่ของการให้บริการอย่างรวดเร็ว เช่น ในกรณีที่มิสสายเรียกเข้าพร้อมกันแล้วไม่สามารถทราบได้ว่ามีสายซ้อนเรียกเข้ามาด้วย ในปัจจุบันระบบได้มีแนวคิดที่จะพัฒนาการใช้การสื่อสารแบบพอร์ตอนุกรม RS 232 (Full-Duplex) ด้วยเช่นกัน เพื่อให้เกิดการรองรับสายเรียกเข้าที่มีเข้ามามากกว่าหนึ่งสายในเวลาเดียวกัน

ผู้จัดทำวิทยานิพนธ์มีความตั้งใจที่จะพัฒนาต่อขอระบบเรียกพยาบาล ให้สามารถทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น โดยจะเน้นไปที่การนำสารสนเทศที่มีอยู่เดิมในระบบเรียกพยาบาลมาพัฒนาในส่วนการแสดงผลผ่านหน้าจอแสดงผล โดยจะใช้หน่วยประมวลผลบอร์ดราสป์เบอรี่พาย (Raspberry Pi) ในการดำเนินการ เพื่อติดต่อกับชุดควบคุมผ่านการส่งผ่านข้อมูลแบบอนุกรม (Serial) แบบพอร์ต RS232 ในการนำข้อมูลที่ได้รับจากชุดควบคุม ซึ่งเป็นบันทึกสายเรียกเข้าจากห้องพักรักษาผู้ป่วยมาเก็บไว้ในฐานข้อมูล พร้อมกันนั้นจะมีส่วนของการแสดงผลบนหน้าจอเพื่อการเฝ้าดูผู้ป่วยในหอพักรักษาผู้ป่วย และแจ้งเตือนทันทีที่มีการเรียกสายบนหน้าจอ แสดงตำแหน่งของห้องที่มีการโทรเข้าและข้อมูลเบื้องต้นของคนไข้ นั้น ๆ เพื่อให้สามารถช่วยเหลือ ให้บริการได้อย่างรวดเร็วที่สุด

ระบบเรียกพยาบาล รุ่น BCRC (BCRC-AE20) ที่นำเข้าโดยบริษัท แครีคอม จำกัด สามารถรองรับการจัดการห้องพักรักษาผู้ป่วยได้สูงสุด 40 ห้องต่อหนึ่งเครื่อง เป็นเครื่องที่จะนำมาใช้ทดสอบการพัฒนาการแสดงผลทางหน้าจอของระบบเรียกพยาบาล ส่วนการประมวลผลจะใช้บอร์ดราสป์เบอรี่พาย โมเดล บี (Raspberry Pi Model B) ที่ใช้ระบบปฏิบัติการราสป์เบียน (Raspbian) เพื่อทำการควบคุมการทำงานของซอฟต์แวร์และจัดการข้อมูลที่ได้รับจากชุดควบคุมในแต่ละโหนด ซึ่งจะมีการนำมาตรวจสอบกับข้อมูลของผู้ป่วยจากฐานข้อมูลในระบบ โดยเลือกการส่งผ่านข้อมูลจะใช้ระบบบัส RS232 ในการติดต่อสื่อสารระหว่างอุปกรณ์กับหน่วยประมวลผล โดยการแสดงผลกำหนดอัตราส่วนการแสดงผลบนหน้าจอเป็น 16:9 เชื่อมต่อผ่านพอร์ต HDMI มาใช้แสดงผลสารสนเทศของโรงพยาบาล

ระบบที่ผู้จัดทำปริญญานิพนธ์มีความตั้งใจที่จะพัฒนาต่อยอนั้น ประกอบไปด้วย 2 ส่วน คือ 1. ระบบการจัดการข้อมูล 2. ส่วนติดต่อประสานงานผู้ใช้ ในระบบการจัดการข้อมูลจะสร้างระบบให้บริการ (Web Service) เพื่อติดต่อแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ ด้วย WAMP โดยเลือกใช้ Autobahn และ Crossbar.io ร่วมกันพัฒนาระบบ และในส่วนติดต่อประสานงานผู้ใช้ (User Interface) มีการแสดงหน้าต่างเตือนพร้อมข้อมูลของผู้ป่วยที่กดปุ่มเรียกจากห้องพักผู้ป่วยนั้น จะใช้ Kivy ซึ่งเป็นไลบรารีหนึ่งของภาษาไพธอนที่พัฒนาต่อมาจาก PyGame

ข้อกำหนดของระบบในปริญญานิพนธ์ ระบบเรียกพยาบาลจะประกอบด้วยประมวลผลที่เป็น Raspberry Pi เชื่อมต่อสูงสุดได้ 3 เครื่อง เพื่อให้ระบบยังสามารถทำงานได้เสถียร โดยมีโหนดมาสเตอร์ (Master Node) มีหน้าที่ควบคุมกำหนดเส้นทางการเชื่อมต่อเพื่อสื่อสารระหว่างแต่ละโหนด โหนดสเลฟ (Slave Node หรือ Display Node) ทำหน้าที่ควบคุมการแสดงผลและเป็นเซิร์ฟเวอร์ของฐานข้อมูล อาจใช้ Raspberry Pi หนึ่งตัวทำหน้าที่มากกว่าหนึ่งโหนดเพื่อเป็นการลดจำนวนฮาร์ดแวร์ที่ใช้งานได้ตามความเหมาะสม ซึ่ง Raspberry Pi แต่ละเครื่องจะเชื่อมต่อกันด้วยสายแลน (LAN) และกำหนด IP Address แบบกำหนดค่าตายตัว (Static IP) เพื่อให้สะดวกต่อการตั้งค่า

เนื้อหาในปริญญานิพนธ์จะกล่าวถึงทฤษฎีความรู้ที่ผู้สนใจโครงการควรมีและสามารถทำความเข้าใจได้ พร้อมขั้นตอน แนวคิดในการพัฒนาและทดสอบระบบ แสดงวิธีการและกล่าวถึงอุปกรณ์ สภาพแวดล้อมที่ใช้ในการทดลองการทำงานของระบบ และฟังก์ชันเชิงการทำงานของระบบ ท้ายที่สุดมีกล่าวสรุปถึงผลการทดสอบจริงที่เกิดขึ้นว่าผู้จัดทำได้พัฒนาระบบไปในทิศทางใด และมีความคาดหวังว่าระบบนั้นจะได้มีการต่อยอดพัฒนาให้ดียิ่งขึ้นได้อย่างไร ดังนั้นการจัดทำปริญญานิพนธ์ชิ้นนี้ ผู้จัดทำมีความคาดหวังว่าจะได้รับความรู้จากการปฏิบัติงานกับระบบที่ใช้จริงและสามารถนำความรู้ที่ได้ไปประยุกต์ใช้เพื่อแก้ไขปัญหาที่อาจเกิดขึ้นภายในอนาคต มีความสามารถจัดการวางแผนอย่างเป็นระบบและการทำงานร่วมกับผู้อื่นได้เป็นอย่างดี อีกทั้งได้นำความรู้ทฤษฎีที่เคยเรียนรู้มาใช้ประกอบการพัฒนาระบบ และได้มีโอกาสเรียนรู้เพิ่มเติมเกี่ยวกับเทคโนโลยีที่เกี่ยวข้องกับการเชื่อมต่ออินเทอร์เน็ตกับสภาพแวดล้อมรอบตัวเพิ่มมากขึ้น

บทที่ 2

ระบบเรียกพยาบาล

ระบบเรียกพยาบาล [6] เป็นอุปกรณ์ที่สามารถพบโดยทั่วไปในสถานพยาบาล โดยมีการริเริ่มมาจาก ฟลอเรนซ์ นิงเกิลผู้บุกเบิกด้านพยาบาลคิดวิธีการอำนวยความสะดวกให้กับทหารที่ป่วยหรือบาดเจ็บจากสงครามให้สามารถขอความช่วยเหลือจากพยาบาลได้ ด้วยการแขวนกริ่งไว้เหนือเตียงเพื่อให้ผู้ป่วยสามารถดึงกริ่งเรียกพยาบาล จากนั้นการเรียกพยาบาลได้มีการพัฒนาต่อยอดกลายมาเป็นระบบเรียกพยาบาลที่มีความสามารถมากยิ่งขึ้น โดยทั่วไปสวิตช์ที่ใช้สำหรับกดกริ่งเรียกพยาบาลจะติดตั้งที่ข้างเตียงผู้ป่วย หรือห้องน้ำ เมื่อผู้ป่วยกดสวิตช์จากตำแหน่งที่ติดตั้งไว้ระบบเรียกพยาบาลจะมีการแสดงผลสัญญาณที่ห้องพักรักษา โดยมัลักษณะเป็นสัญญาณไฟประจำห้องและเสียงสัญญาณเตือน จากการแสดงผลนี้ทำให้เจ้าหน้าที่พยาบาลสามารถเข้าไปยังห้องพักรักษาหรือเตียงผู้ป่วยเพื่อตรวจสอบและให้การช่วยเหลือได้ทันที โดยทั่วไประบบเรียกพยาบาลนี้มีราคาที่หลากหลายขึ้นอยู่กับประสิทธิภาพหรือรูปแบบ การใช้งาน และความต้องการของโรงพยาบาลนั้น

2.1 ระบบเรียกพยาบาล

2.1.1 ระบบเรียกพยาบาล แบ่งออกเป็น 3 แบบ ได้แก่

2.1.1.1 ระบบเรียกพยาบาลแบบสื่อสารทางเดียว เรียกพยาบาลได้อย่างเดียว ไม่สามารถโต้ตอบได้ การเรียกผู้ป่วยจะแสดงที่สถานีพยาบาลในลักษณะเสียงสัญญาณพร้อมไฟแสดงตำแหน่งเป็นระบบเรียกพยาบาลขั้นพื้นฐาน ส่วนใหญ่ใช้กับห้องเตียงรวม ราคาประหยัด

2.1.1.2 ระบบเรียกพยาบาลที่สามารถโต้ตอบได้ (Intercom) แบ่งออกเป็น 2 แบบ คือ 1. แบบที่โต้ตอบกัน ได้แต่ต้องผลัดกันพูดเหมือนวิทยุสื่อสาร (Simplex) 2. สามารถสนทนาโต้ตอบได้เหมือนโทรศัพท์ (Full Duplex) ส่วนใหญ่ระบบเรียกพยาบาลลักษณะนี้จะใช้ในห้องพักผู้ป่วยพิเศษ

2.1.1.3 ระบบเรียกพยาบาลแบบไร้สาย ทำงานโดยอาศัยคลื่นความถี่ มีขีดจำกัดเรื่องระยะทางจึงเหมาะกับงานที่มีจำนวนจุดเรียกไม่มากนัก เช่น ในบ้านพักที่มีผู้สูงอายุ

2.1.2 ส่วนประกอบของระบบเรียกพยาบาล

2.1.2.1 ชุดควบคุม (Master Station) เป็นอุปกรณ์ที่ใช้สื่อสารระหว่างผู้ป่วยกับพยาบาล เมื่อมีการเรียกพยาบาลจะมีสัญญาณเสียงและไฟแสดงหมายเลขห้องหรือเตียงผู้ป่วย ในกรณีที่ชุดเรียกพยาบาลเป็นรุ่นโต้ตอบได้ ชุดควบคุมจะมีลำโพง และไมโครโฟนภายในตัวเครื่อง ชุดควบคุมนี้อาจมีช่องทางการเชื่อมกับคอมพิวเตอร์สำหรับเก็บข้อมูลหรือนำไปพัฒนาต่อยอดเพื่อการแสดงผลที่มากขึ้นได้



ภาพที่ 2-1 ชุดควบคุม (Master Station)

2.1.2.2 ชุดเรียกพยาบาลใช้สำหรับเรียกพยาบาล ติดตั้งที่หัวเตียง ใช้งานร่วมกับสายกดเรียกพยาบาล และมีสวิทช์ยกเลิกการเรียกพยาบาล ชุดเรียกพยาบาลมีทั้งรุ่นที่สามารถโต้ตอบกับชุดควบคุมได้ และไม่สามารถโต้ตอบกับชุดควบคุมได้



ภาพที่ 2-2 ชุดเรียกพยาบาลติดตั้งที่หัวเตียง

2.1.2.3 สายกดเรียกพยาบาล (Call Cord) เป็นอุปกรณ์ที่เชื่อมต่อจากชุดเรียกพยาบาล ถูกออกแบบมาให้กระชับมือ มีลักษณะเป็นสาย ทำให้สามารถกดปุ่มได้สะดวก มักจะวางอยู่ข้างตัวผู้ป่วย



ภาพที่ 2-3 สายกดเรียกพยาบาล (Call Cord)

2.1.2.4 สวิตช์ดึงเรียกพยาบาลในห้องน้ำ (Emergency Switch) คือสวิตช์เรียกพยาบาลที่ติดตั้งในห้องน้ำ เป็นชนิดกดหรือดึงเรียกพยาบาล ประกอบด้วย สวิตช์กดเรียก พร้อมเชือกสำหรับดึง เนื่องจากเมื่อมีการลื่นหกล้มในห้องน้ำ ต้องสามารถเรียกพยาบาลเพื่อขอความช่วยเหลือได้ทันที



ภาพที่ 2-4 สวิตช์ดึงเรียกพยาบาลในห้องน้ำ (Emergency Switch)

2.1.2.5 โคมไฟน้ำห้อง (Corridor Light) ติดตั้งหน้าห้องพักคนไข้ เป็นหลอดไฟหรืออาจเป็น LED ครอบด้วยโคมพลาสติกสีขาว สัญญาณไฟจะติดเมื่อมีการกดสวิตช์ที่หัวเตียงหรือดึงสวิตช์ห้องน้ำ เพื่อให้พยาบาลสามารถทราบได้ว่ามีคนไข้เรียกมาจากห้องไหน และสัญญาณไฟจะดับเมื่อมีการกดสวิตช์ที่ชุดควบคุม (Master Station) หรือที่สวิตช์ยกเลิกของชุดเรียกพยาบาล (Sub Station)



ภาพที่ 2-5 โคมไฟหน้าห้อง (Corridor Light)

2.1.3 ระบบเรียกพยาบาล รุ่น BCRC (BCRC-AE20) [7]

เครื่องระบบเรียกพยาบาล รุ่น BCRC ใช้สำหรับให้ผู้ป่วยทำการเรียกพยาบาลผ่านเครื่องโทรศัพท์ ซึ่งสามารถทำการเรียกได้ 2 แบบ คือ แบบปกติ (General Call) ซึ่งปุ่มกดเรียกจะติดตั้งอยู่ด้านข้างเตียงผู้ป่วย และแบบฉุกเฉิน (Emergency Call) โดยจะติดตั้งปุ่มกดเรียกในห้องน้ำ ระบบเรียกพยาบาลรุ่นนี้จะสามารถรองรับการจัดการสายเรียกเข้าได้ทั้งหมด 20 เสียง

ในหนึ่งชุดจะประกอบไปด้วยชุดควบคุม โคมไฟหน้าห้อง สวิตช์ดึงเรียกพยาบาลในห้องน้ำ ปุ่มกดรีเซ็ต (Reset) ชุดเรียกพยาบาลซึ่งเป็นส่วนที่ใช้พูดคุยกับพยาบาล และสายกดเรียกพยาบาลที่เชื่อมต่อมาจากชุดเรียกพยาบาลอีกทีหนึ่ง

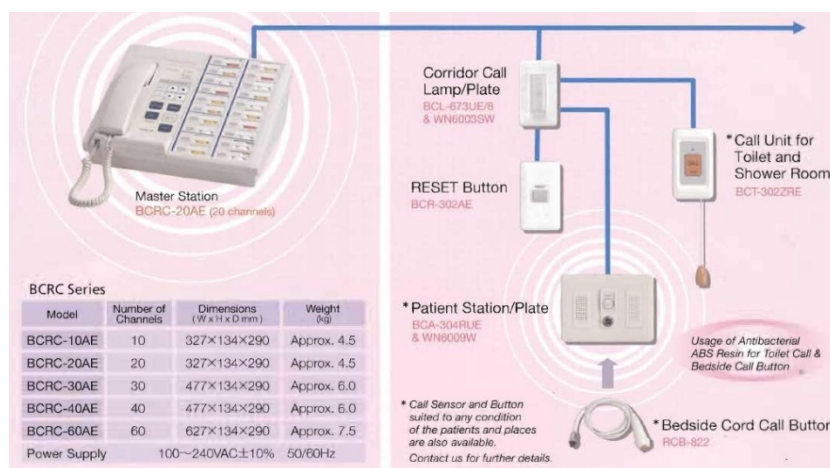
เมื่อผู้ป่วยทำการเรียกแบบปกติจะมีแจ้งเตือนไปยังเครื่อง BCRC ที่อยู่ในความดูแลของพยาบาล โดยเสียงการแจ้งเตือนของเครื่อง BCRC จะหยุด เมื่อมีการรับโทรศัพท์ที่เครื่อง BCRC หรือการเดินไปกดรีเซ็ตในห้องผู้ป่วย แต่ถ้าผู้ป่วยทำการเรียกแบบฉุกเฉิน พยาบาลจะต้องไปกดปุ่มรีเซ็ตในห้องที่มีการเรียกเข้ามาเพื่อหยุดการแจ้งเตือนเท่านั้น

โครงการนี้จะกล่าวถึงการสร้างระบบแสดงผลการแจ้งเตือนของเครื่อง BCRC ผ่านจอแสดงผล เพื่อพัฒนาระบบแสดงผลการแจ้งเตือนในโรงพยาบาลให้มีประสิทธิภาพ สามารถติดตั้งและใช้งานได้หลายตำแหน่งทดแทนการดูผลจากเครื่อง BCRC เพียงเครื่องเดียว



ภาพที่ 2-6 เครื่อง BCRC

เครื่อง BCRC สามารถตั้งค่า Volume Control Function ปรับระดับเสียงได้ 5 ระดับ สามารถเลือกเสียงเรียกเข้าได้แตกต่างกัน 10 แบบ ทั้งแบบเสียงเพลงและเสียงระฆังเตือน โดยเครื่อง BCRC นั้นรองรับการเรียก 2 แบบคือ Emergency Call เป็นการเรียกแบบฉุกเฉิน (ในห้องน้ำและห้องอาบน้ำ) และ General Call เป็นการเรียกแบบปกติ (ด้านข้างเตียงผู้ป่วย) และเครื่อง BCRC จะมีการคืนค่าฟังก์ชันอัตโนมัติเมื่อวางหูโทรศัพท์ที่ไม่ตรง โดยจะทำการคืนค่าเมื่อหูโทรศัพท์ที่อยู่นอกที่วางมากกว่า 3 นาที เพื่อป้องกันการวางหูไม่สนิท



ภาพที่ 2-7 ระบบของเครื่อง BCRC

2.2.1 คุณสมบัติทางเทคนิคของบอร์ด Raspberry Pi

หน่วยประมวลผล	Broadcom BCM2835 ARM11 700MHz
หน่วยความจำชั่วคราว (RAM)	512 MB*
หน่วยความจำหลัก	การ์ด SD
ช่องเชื่อมต่อเครือข่าย LAN	10/100 Mbps
แสดงผลวิดีโอ	พอร์ต HDMI
พอร์ต USB 2.0	2 พอร์ต
ช่องเชื่อมต่ออินพุตเอาต์พุต (GPIO)	17 ช่อง
ช่องต่อสัญญาณเสียง	3.5 mm
ไฟเลี้ยง	5 VDC
ขนาดบอร์ด	85.0 x 56.0 mm

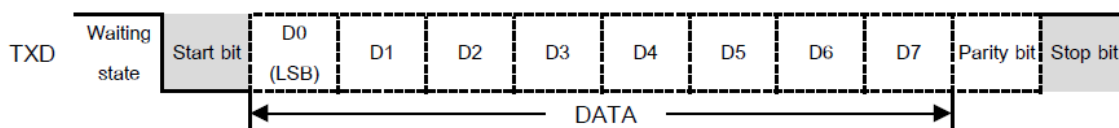
*รุ่น Model B ในรุ่นก่อน ๆ มีหน่วยความจำ 256 MB

2.3 การสื่อสารข้อมูลแบบ UART

UART (Universal Asynchronous Receiver Transmitter) [9] เป็นการรับ-ส่งข้อมูลบิตอนุกรมแบบอะซิงโครนัส (Asynchronous Transmission) หรือก็คือ การรับส่งข้อมูลที่ไม่จำเป็นต้องอาศัยสัญญาณนาฬิกา (Clock) เป็นตัวกำหนดจังหวะ ใช้การกำหนดรูปแบบการรับส่งสัญญาณแทน ซึ่งรูปแบบและความเร็วในการรับ-ส่งข้อมูลจะต้องเหมือนกันทั้งฝั่งส่งและฝั่งรับ โดยการรับส่งข้อมูลแบบ UART นี้ จะเป็นการสื่อสารแบบ Full Duplex คือ สามารถรับส่งข้อมูลได้พร้อมกัน

รูปแบบการส่งข้อมูลแบบอะซิงโครนัส (Asynchronous Transmission) ประกอบด้วย 4 ส่วน ดังนี้

- บิตเริ่มต้น (Start Bit) : บิตที่บอกว่าได้มีการเริ่มต้นการส่งข้อมูลแล้ว จำนวน 1 บิต
- บิตข้อมูล (Data Bits) : ข้อมูลที่ต้องการส่ง จำนวน 5-8 บิต
- พาริตีบิต (Parity Bit) : บิตที่เอาไว้ตรวจสอบว่าข้อมูลที่ได้รับมีความถูกต้องหรือไม่ มีหรือไม่ก็ได้ โดยปกติแล้วจะมีจำนวน 1 บิต
- บิตสิ้นสุด (Stop Bit) : บิตที่บอกว่าสิ้นสุดการส่งข้อมูล โดยปกติมีจำนวน 1 บิต หรือ 2 บิต



ภาพที่ 2-9 ภาพแสดงการส่งข้อมูลแบบ UART

ภาพที่ 2-9 เมื่อไม่มีการส่งข้อมูล ขาสัญญาณจะมีสถานะเป็นลอจิก “1” หรือสถานะรอ (Waiting stage) เมื่อเริ่มต้นการส่งข้อมูลจะส่งบิตเริ่มต้นเพื่อบอกว่าเริ่มทำการส่งข้อมูลแล้ว โดยมีลอจิก “0” หลังจากนั้นจึงทำการส่งข้อมูล เป็นบิตข้อมูลเป็นจำนวน 8 บิต โดยส่งบิตที่มีค่าต่ำสุดไปก่อน (LSB) หลังจากนั้นจึงส่งพาริตีบิต (จะมีหรือไม่มีก็ได้) แล้วจึงส่งบิตสิ้นสุดเป็นลอจิก “1” อย่างน้อย 1 บิต เพื่อบอกว่าสิ้นสุดการส่งข้อมูล

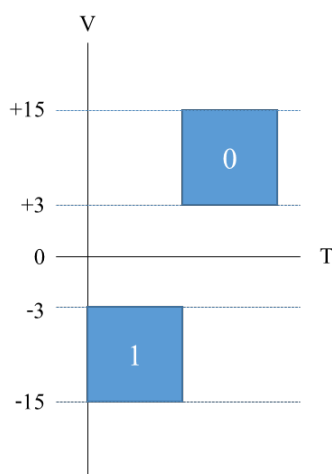
อัตราการส่งข้อมูล (Baud rate) คือความเร็วของการรับ-ส่งข้อมูล มีหน่วยเป็นจำนวนบิตต่อวินาทีเช่น 300, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000 เป็นต้น การเลือกอัตราการส่งข้อมูลขึ้นอยู่กับชนิดของสายสัญญาณ ระยะทาง และปริมาณสัญญาณรบกวน

2.4 ระบบบัส RS232

RS232 [10] เป็นมาตรฐานแบบหนึ่งในการรับส่งข้อมูลบิตอนุกรมแบบอะซิงโครนัส (Asynchronous Serial Communication) ใช้วงจรที่เรียกว่า UART (Universal Asynchronous Receiver Transmitter) ในการส่งข้อมูล

ในระบบบัส RS232 จะมีขาสัญญาณ 2 ขาคือ

- Tx (Transmit) เป็นขาที่ใช้ส่งข้อมูล
- Rx (Receive) เป็นขาที่ใช้รับข้อมูล



ภาพที่ 2-10 ระดับสัญญาณของ RS232

โปรโตคอล RS232 ถูกออกแบบมาให้มีสองลักษณะเด่นคือ สามารถจัดการสัญญาณรบกวนได้ และ ส่งได้ไกล จึงเกิดการออกแบบให้ logic level 1 มีค่าอยู่ในช่วง -3 ถึง -15 v และ logic 0 มีค่าอยู่ในช่วง +3 ถึง +15 V ตามภาพที่ 2-10

2.5 ซอฟต์แวร์ที่ใช้พัฒนาระบบ

2.5.1 pySerial [11] ทำหน้าที่เชื่อมโยงระหว่าง Python และพอร์ตอนุกรม ให้สามารถรับส่งข้อมูลผ่านโปรแกรม Python ได้ โดยโมดูล pySerial นี้จะทำงานเป็น Background ของโปรแกรม

2.5.1.1 คุณสมบัติที่สำคัญของ pySerial

- รองรับหลายแพลตฟอร์ม
- รองรับการตั้งค่าพอร์ตอนุกรมที่หลากหลาย เช่น ขนาด, จำนวน stop bits, parity และ flow control
- API รองรับการทำงานเช่นเดียวกับไฟล์ เช่นกับการเขียนและการอ่านข้อมูล

2.5.1.2 ความต้องการของระบบ

- Python version 2.3 ขึ้นไป (Python 3.x มีโมดูลนี้รวมอยู่แล้ว)
- ต้องมี ctypes extension ของ Python อยู่แล้ว (ใน Python version 2.6 ขึ้นไป มีอยู่แล้ว)

2.5.1.3 การติดตั้ง pySerial

การติดตั้งบนระบบปฏิบัติการ Windows ให้เราเลือกติดตั้งโดยให้เลือกใช้แบบทุก user หากเป็นการติดตั้งบนระบบปฏิบัติการ Linux ให้ติดตั้งโดย root user หลังจากติดตั้ง pySerial เรียบร้อยแล้วแล้ว เราจะต้อง import serial เข้ามาในโปรแกรม

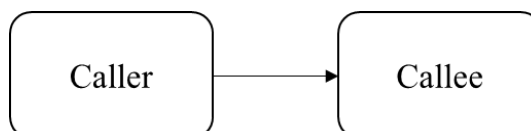
2.5.2 Kivy [12] เป็นซอฟต์แวร์โอเพนซอร์สอีกตัวหนึ่งพัฒนามาจากไลบรารีของภาษาไพธอน (Python) เช่น pygame, gstreamer, PIL, Cairo และอื่น ๆ อีกมากมาย โดยสร้างขึ้นมาเพื่อพัฒนาแอปพลิเคชันในส่วนของการติดต่อกับผู้ใช้ (User Interface) ซึ่งสามารถทำงานบนระบบปฏิบัติการได้หลากหลาย ไม่ว่าจะเป็น Linux, Windows, Android และ iOS

2.5.3 WAMP (The Web Application Messaging Protocol) [13] เป็นโปรโตคอลการสื่อสารชนิดหนึ่ง ซึ่งช่วยกระจายการติดต่อสื่อสารระหว่างแอปพลิเคชัน เราสามารถเปรียบ WAMP เสมือนเป็นเราเตอร์ (Router) ที่จัดการเส้นทางการสื่อสารระหว่างแอปพลิเคชัน WAMP ยังสามารถใช้รับส่งข้อมูลที่ใช้นบนเทคโนโลยีที่แตกต่างกัน (เช่น โทรศัพท์มือถือสมาร์ทโฟน ระบบสมองกลฝังตัว คอมพิวเตอร์) ติดต่อสื่อสารและทำงานได้แบบเรียลไทม์ (Real time)

นอกจากนี้ WAMP ยังสามารถนำไปพัฒนาซอฟต์แวร์ได้หลายภาษา โดยการส่งข้อมูลผ่าน WAMP จะมีทั้งหมด 2 แบบ คือ

1. Publish & Subscribe (PubSub)
2. Remote Procedure Calls (RPC)

โลกของ Client-Server นั้น Remote Procedure Call จะส่งข้อมูลจาก Caller ไปยัง Callee โดยตรงดังภาพที่ 2-11



ภาพที่ 2-11 ภาพแสดงการทำงานแบบ Remote procedure calls ในแบบของ Client-Server model

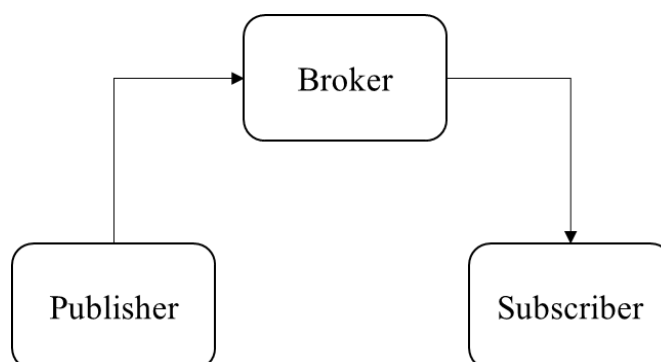
การส่งแบบ Client-server model นั้น Caller จะต้องรู้ว่า Callee อยู่ที่ไหน และจะส่งข้อมูลไปยัง Callee ได้อย่างไร ซึ่งแสดงให้เห็นการเชื่อมต่อแบบเจาะจงระหว่าง Caller และ Callee

ซึ่งถ้าระบบมีความซับซ้อนมากขึ้น อาจส่งผลให้การบำรุงรักษาระบบมีความยุ่งยากมากขึ้น ซึ่ง WAMP สามารถแก้ปัญหาในจุดนี้ได้

ปัญหาการเชื่อมต่อระหว่าง Applications นี้ถูกค้นพบมานานมาแล้ว และนั่นเป็นสาเหตุทำให้ผู้พัฒนาทำการพัฒนา publish-subscribe model นี้ขึ้นมา

2.5.3.1 Publish & Subscribe (PubSub)

การส่งข้อมูลแบบ Publish-subscribe model นั้น Publisher จะประกาศหัวข้อที่มีการส่งข้อมูล (topic) และ Subscriber จะทำการประกาศหัวข้อที่ตนเองสนใจจะรับข้อมูล เมื่อ Publisher ส่งข้อมูลมาหัวข้อเดียวกับที่ Subscriber ประกาศไว้ ก็จะมีการรับส่งข้อมูลขึ้น ซึ่งการรับส่งข้อมูลในลักษณะนี้ Publisher และ Subscriber ไม่ได้รับส่งข้อมูลกันโดยตรง แต่รับส่งข้อมูลกันผ่านทาง Broker



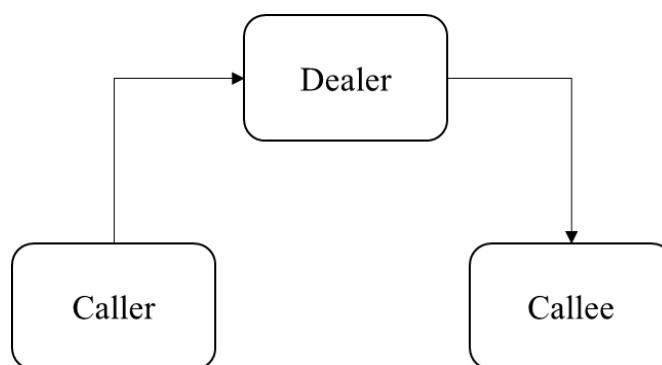
ภาพที่ 2-12 ภาพแสดงการส่งข้อมูลจาก Publisher และ Subscriber ผ่าน Broker (“topic”)

Broker จะเป็นตัวที่คอยทำการตรวจสอบว่า ใคร subscribe บน topic ไหน เมื่อ Publisher ทำการ publish ข้อมูลใน topic นั้น ๆ Broker จะกำหนดจำนวน Subscriber ทั้งหมดใน topic ที่มีการ Publish หลังจากนั้นข้อมูลจะถูกส่งไปยัง subscriber ทั้งหมดใน topic นั้น ๆ

การกำหนดผู้รับที่จะได้รับข้อมูลทั้งหมด และการส่งข้อมูล (forward) นั้น ๆ ไปยังผู้รับ เรียกว่า routing

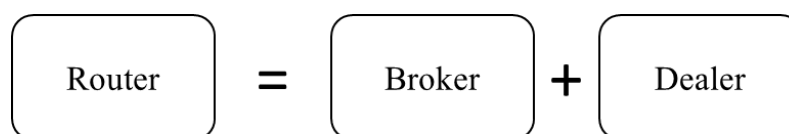
2.5.3.2 Remote Procedure Calls (RPC)

การส่งข้อมูลแบบ RPC จะแตกต่างกับการส่งข้อมูลแบบ client-server model ตรงที่ Caller จะส่งข้อมูลไปยัง Callee ผ่าน Dealer



ภาพที่ 2-13 ภาพการส่งข้อมูล RPC ในรูปแบบ Dealer model

หน้าที่ของดีลเลอร์ (Dealer) จะเหมือนกับ Broker คือทำการ routing ข้อมูลจาก Caller ไปยัง Callee และหาเส้นทางจาก Callee ไปยัง Caller ดีลเลอร์จะเป็นตัวเก็บข้อมูลว่าจะต้องส่งข้อมูลไปที่ใด และส่งอย่างไร



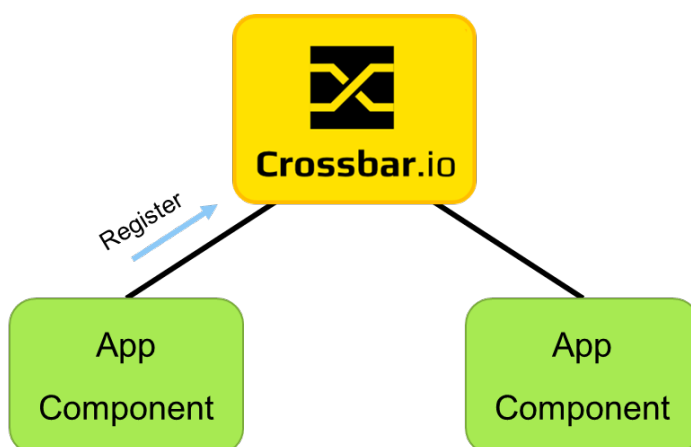
ภาพที่ 2-14 เราท์เตอร์คือการรวมกันระหว่าง Broker กับ Dealer

การนำ Broker และ Dealer มาใช้งานร่วมกัน ก็จะทำงานเสมือนเราท์เตอร์ที่ทำการส่งข้อมูล หรือที่เรียกว่า WAMP สำหรับการส่งข้อมูลในโครงการจะขอกว่าถึงการใช้งาน crossbar.io เป็นสำคัญ

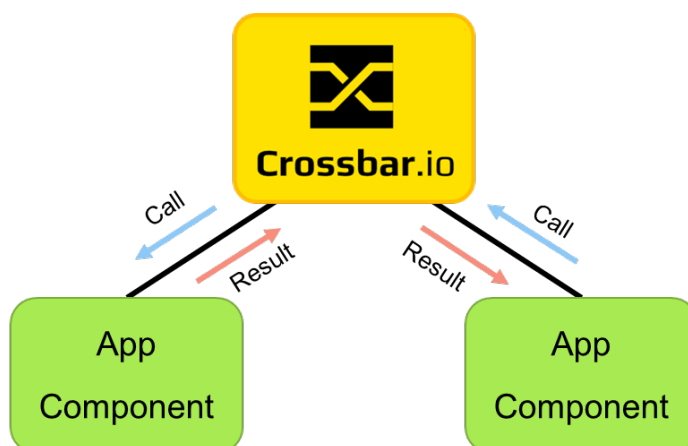
2.5.4 Crossbar.io [14] ถ้าเปรียบ WAMP เป็นเราเตอร์ Crossbar.io ก็เปรียบเสมือนหนึ่งในยี่ห้อของเราเตอร์ที่มีให้เลือกใช้งาน ดังนั้น Crossbar.io จึงมีหลักการทำงานเหมือนกับ WAMP ทุกประการ ซึ่งจะมี Publish & Subscribe (PubSub) และ Remote Procedure Calls (RPC) โดยในที่นี้จะแสดงตัวอย่างการใช้งาน PubSub และ RPC เพื่ออธิบายความให้เข้าใจมากยิ่งขึ้น

2.5.4.1 Remote Procedure Calls (RPC)

การใช้งาน RPC นั้น ไม่ว่าจะเป็น component ใดก็สามารถลงทะเบียน (Register) และเรียกหรือร้องขอ (Call) ได้ โดย Register จะทำหน้าที่เสมือนผู้ที่เก็บข้อมูลการลงทะเบียน เมื่อ Register เปิดให้ลงทะเบียนได้ component อื่นจะสามารถเรียกหรือร้องขอ component ที่ลงทะเบียนไว้ก่อนหน้าทำงานได้ โดย component ที่ได้ลงทะเบียนไว้จะสามารถส่งข้อมูลกลับไปยัง component ที่มีการเรียกหรือร้องขอมา



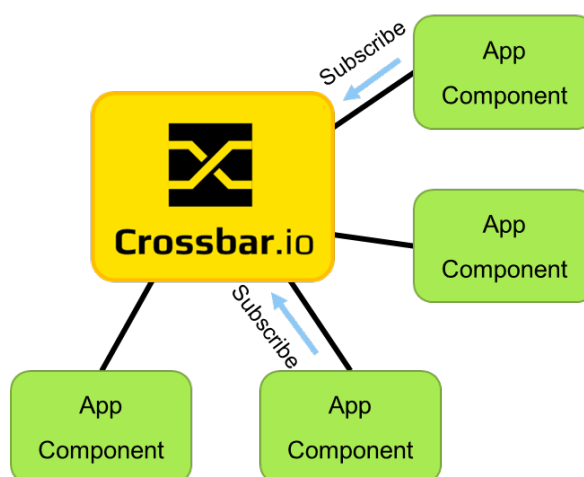
ภาพที่ 2-15 การ register



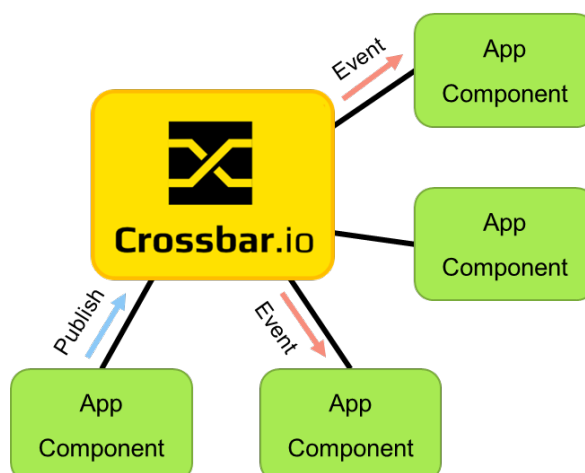
ภาพที่ 2-16 แสดงถึงการที่ component หนึ่งทำการ call ไปหาอีก component หนึ่งผ่าน crossbar.io และ component ที่ register นั้นทำการส่งข้อมูลกลับไป

2.5.4.2 Publish & Subscribe (PubSub)

component ใด subscribe ไปที่ topic เดียวกันกับที่ publisher ประกาศไว้ crossbar.io จะทำหน้าที่เป็นเราเตอร์ส่งข้อมูลให้ component นั้น ๆ ทุกครั้งที่เกิดเหตุการณ์ที่ publisher มีการเปลี่ยนแปลง



ภาพที่ 2-17 แสดงถึงการ subscribe ไปที่ topic A (สมมติ)



ภาพที่ 2-18 แสดงภาพการทำงานขณะที่ component publish topic A แล้ว component ที่ทำการ subscribe ที่ topic A ไว้ก็จะได้รับ event (ข้อมูล) จาก crossbar.io

2.5.5 Autobahn [15] เป็นโอเพนซอร์สที่จัดทำระบบในเรื่องของการติดต่อแบบโปรโตคอล WebSocket และ Web Application Messaging Protocol (WAMP)

WebSocket อนุญาตให้มีการเชื่อมต่อแบบสองทางแบบเรียลไทม์ (Real time) ระหว่าง Web และ WAMP ได้ และยังเพิ่มความสามารถ Remote Procedure Calls และ Publish & Subscribe แบบอะซิงโครนัส (Asynchronous) ในส่วนของ WebSocket WAMP นั้นเป็นแนวคิดการกระจายระหว่าง client หลาย ๆ ตัวกับ Server Application เช่น Multi-user database app, Sensors network, IM หรือ เกมแนว MMORPG

สรุปคือ WAMP ประกอบด้วยสองส่วน คือ ส่วนที่เปรียบเสมือนเราเตอร์ข้อมูลก็คือ crossbar.io เป็นตัวจัดการในการส่ง สวิตซ์ข้อมูล เหมือนเราเตอร์ข้อมูล และส่วนที่เป็นไลบรารี (Library) ก็คือ Autobahn|Python เป็นไลบรารีสำหรับการนำไปใช้เขียนโปรแกรม

2.5.5.1 Connection Lifecycle ประกอบด้วย

- onConnect() – Opening Handshake เมื่อใดที่ client ตัวใหม่ connect กับ server จะสร้างโปรโตคอลใหม่โดยทันที และฟังก์ชัน onConnect() จะถูกเรียกโดยทันทีที่ WebSocket handshake เริ่มต้นกับ client

- onOpen() – onOpen() จะถูกเรียกทันทีที่ WebSocket opening handshake สำเร็จอย่างสมบูรณ์ หลังจากนั้นจะสามารถส่งและรับข้อมูลผ่านการเชื่อมต่อนี้ได้
- sendClose() – จะ initiate WebSocket closing handshake และหลังจากทำการ close WebSocket connection แล้วจะไม่สามารถส่ง message ได้อีก และสุดท้าย onClose() จะถูกเรียก
- onClose() – เมื่อ WebSocket connection ถูกปิดเรียบร้อยแล้ว จะเรียกฟังก์ชันนี้

2.5.6 MySQL (มายเอสคิวแอล) [16] เป็นระบบจัดการฐานข้อมูลแบบโอเพนซอร์ส ใช้ภาษา SQL ในการจัดการฐานข้อมูล MySQL ถูกพัฒนาขึ้นโดยชาวสวีเดน 2 คน และชาวฟินแลนด์ ชื่อ David Axmark, Allan Larsson และ Michael “Monty” Widenius.

2.5.6.1 การใช้งาน

MySQL นิยมใช้กับฐานข้อมูลสำหรับเว็บไซต์ โดย MySQL นี้นิยมใช้กับโปรแกรมภาษา PHP แต่ทั้งนี้ทั้งนั้น MySQL ยังสามารถใช้ได้ร่วมกับหลายภาษา เช่น ภาษาซี ซีพลัสพลัส ปาสคาล ซีชาร์ป ภาษาจาวา ภาษาเพิร์ล พีเอชพี ไพทอน รูบี และภาษาอื่น ๆ

2.5.6.2 โปรแกรมช่วยในการจัดการฐานข้อมูล และทำงานกับฐานข้อมูล

การจัดการฐานข้อมูล MySQL โดยปกติแล้วกระทำผ่านโปรแกรมแบบ command-line โดยใช้คำสั่ง mysql และ mysqladmin หรือสามารถดาวน์โหลดโปรแกรมจัดการฐานข้อมูลแบบ GUI มาใช้ในการจัดการฐานข้อมูลได้เช่นเดียวกัน เช่น MySQL Administrator, MySQL Query Browser และ phpMyAdmin ซึ่งจะทำให้เกิดความสะดวกสบายในการใช้งานฐานข้อมูลมากขึ้น มี interface ที่รองรับการเชื่อมต่อกับโปรแกรมที่พัฒนาจากภาษาได้หลากหลาย เช่น ODBC, JDBC, MyODBC ดังนั้น MySQL จึงเหมาะแก่การนำมาประยุกต์ใช้งานเพื่อพัฒนากับภาษาต่าง ๆ เช่น Python

MySQL มี API ที่สนับสนุนในการเชื่อมต่อกับภาษาต่าง ๆ ดังนี้

- DBI สำหรับการเชื่อมต่อกับภาษา perl
- Ruby สำหรับการเชื่อมต่อกับภาษา ruby
- Python สำหรับการเชื่อมต่อกับภาษา python
- .NET สำหรับการเชื่อมกับภาษา .NET framework

- MySQL++ สำหรับเชื่อมต่อกับภาษา C++
- Ch สำหรับการเชื่อมต่อกับ Ch (C/C++ interpreter)

อุปกรณ์และเทคโนโลยีที่เกี่ยวข้อง ไม่ว่าจะเป็นความรู้เบื้องต้นเพื่อทำความเข้าใจกับเครื่องเรียกพยาบาล บอร์ดที่ใช้ในการประมวลผล วิธีการติดต่อสื่อสารหรือการส่งข้อมูลกันระหว่างอุปกรณ์ มีความจำเป็นอย่างยิ่งที่จะต้องศึกษาให้เกิดความรู้ความเข้าใจ และซอฟต์แวร์ที่จำเป็นต่าง ๆ จะต้องมีการค้นคว้าหาวิธีการเพื่อนำมาติดตั้งให้สามารถเข้ากันกับระบบและอุปกรณ์ได้เป็นอย่างดี จึงจะสามารถพัฒนาระบบเรียกพยาบาลให้เป็นไปอย่างราบรื่น

บทที่ 3

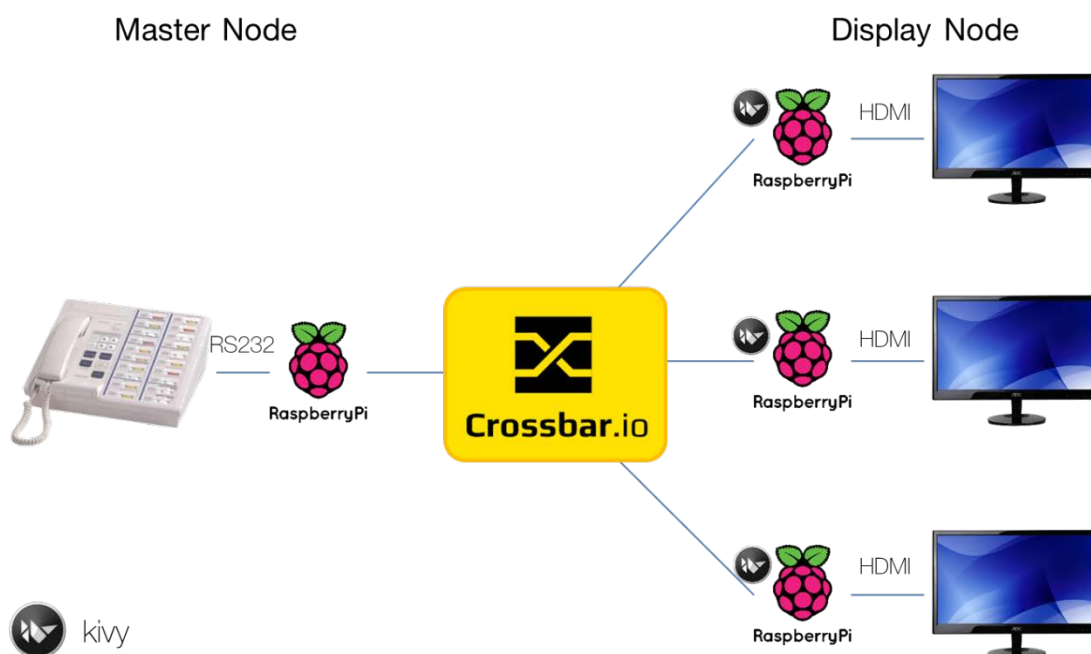
ระบบแสดงผลแจ้งเตือนระบบเรียกพยาบาล

ขั้นตอนการดำเนินการสร้างระบบแสดงผลการแจ้งเตือนของเครื่อง BCRC จำเป็นต้องมีการติดตั้งซอฟต์แวร์ต่าง ๆ บน Raspberry Pi เพื่อให้บอร์ด Raspberry Pi สามารถรองรับการทำงานในการรับส่งข้อมูลกับเครื่อง BCRC ไปเก็บไว้ในฐานข้อมูลและแสดงผลบนจอแสดงผล โดยมีขั้นตอนการดำเนินงานโดยรวมดังต่อไปนี้

1. ออกแบบระบบ
 - โครงสร้างของระบบ
 - การออกแบบหน้าจอการแสดงผล
 - การออกแบบการจัดเก็บความเชื่อมโยงของฐานข้อมูล
2. ติดตั้งซอฟต์แวร์สำหรับงานบนบอร์ด Raspberry Pi ซึ่งประกอบด้วย
 - PySerial 2.7
 - autobahn.ws & crossbar.io
 - kivy
 - โปรแกรมหรือไลบรารีอื่น ๆ ที่จำเป็นต้องติดตั้งสำหรับโปรแกรมหลักข้างต้น
3. จัดทำระบบ/เขียนโปรแกรมสำหรับการทำงาน
4. ทดสอบการทำงานว่าอุปกรณ์และหน่วยประมวลผลทำงานได้ถูกต้องหรือไม่
5. ปรับปรุงแก้ไขปัญหาที่เกิดขึ้น

3.1 สถาปัตยกรรมของระบบ

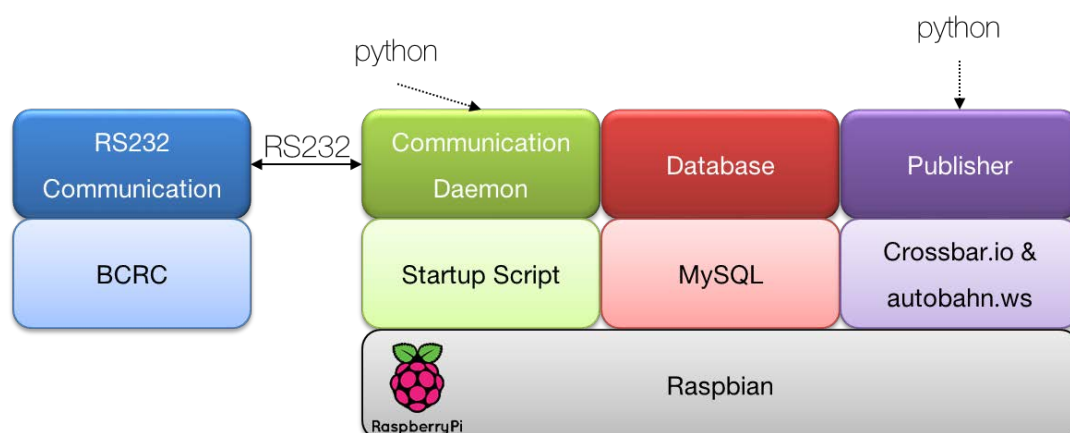
3.1.1 โครงสร้างของระบบ



ภาพที่ 3-1 โครงสร้างของระบบ

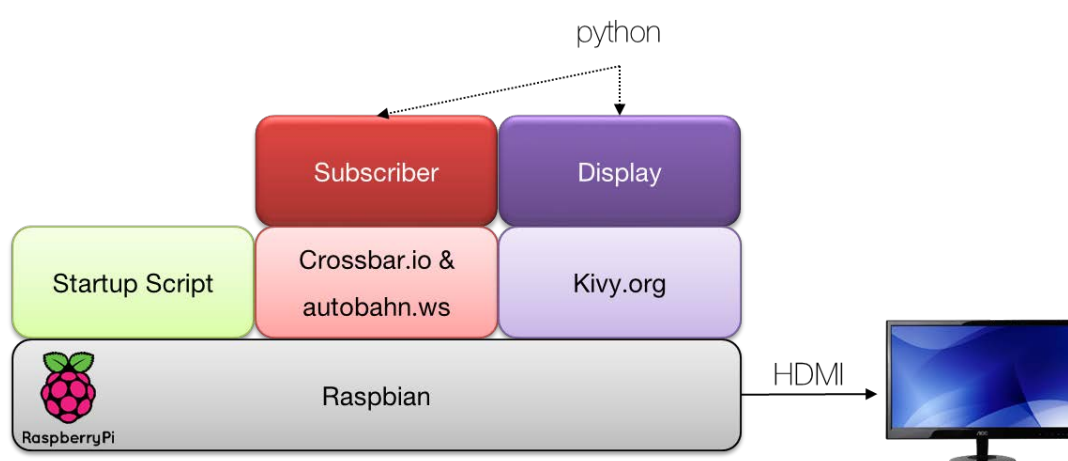
เครื่องเรียกพยาบาล BCRC จะทำการติดต่อสื่อสารกับ Raspberry Pi ผ่าน USB-to-serial RS232 Raspberry Pi จะนำค่าที่ BCRC ส่งมาไปประมวลผลและส่งข้อมูลต่อไปยัง Raspberry Pi หรือคอมพิวเตอร์เครื่องอื่น ๆ ต่อไป โดยผ่าน crossbar.io ซึ่งเป็นตัวกระจายเปรียบเสมือน router การใช้ crossbar.io นั้นจะใช้ร่วมกับ autobahn.ws (ซึ่งเป็น open-source software สำหรับการทำให้ implementations ของ The WebSocket Protocol กับ The Web Application Messaging Protocol (WAMP) ซึ่งในระบบนี้จะใช้ WAMP เป็นหลัก) หลังจาก Raspberry Pi (ฝั่งด้านขวาของภาพที่ 3-1) ได้รับข้อมูลเรียบร้อยแล้ว จึงจะนำข้อมูลที่ได้มาประมวลผลและแสดงผลผ่าน kivy (open source Python library สำหรับการทำให้ user interface) และแสดงผลผ่านหน้าจอโดยใช้สาย HDMI

ระบบเรียกพยาบาลที่คณะผู้จัดทำปริญญาโทพัฒนาขึ้นนั้น ได้แบ่งออกเป็น 2 ส่วน คือ 1. Master node ทำหน้าที่ควบคุมกำหนดเส้นทางและการติดต่อสื่อสารระหว่างชุดควบคุมหรือเครื่อง BCRC และ 2. Display node ทำหน้าที่กำหนดการแสดงผลการแจ้งเตือนบนหน้าจอ



ภาพที่ 3-2 สถาปัตยกรรมซอฟต์แวร์ของ Master Node

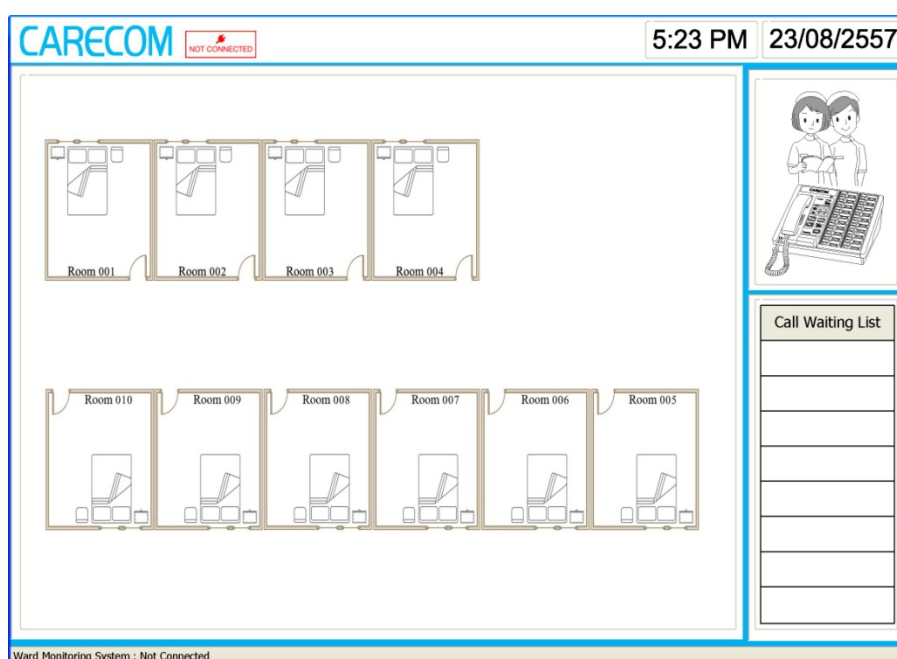
Master Node จะมีการติดต่อสื่อสารกับเครื่อง BCRC ผ่าน RS232 ซึ่งส่วนที่ทำการติดต่อสื่อสารผ่านระบบบัส RS232 นั้นจะพัฒนาด้วยภาษา Python จากนั้นจึงนำข้อมูลที่ได้มาประมวลผลและเก็บค่าลงฐานข้อมูล (Database) โดยในระบบนี้จะใช้ฐานข้อมูล MySQL และในขณะเดียวกันจะทำการ publish ข้อมูลเพื่อให้คอมพิวเตอร์ (Raspberry Pi) เครื่องอื่นนำข้อมูลนี้ไปแสดงผลต่อไป ซึ่งการ publish ข้อมูลนี้จะใช้ Crossbar.io และ autobahn.ws ซึ่งเป็น open source ในการ publish



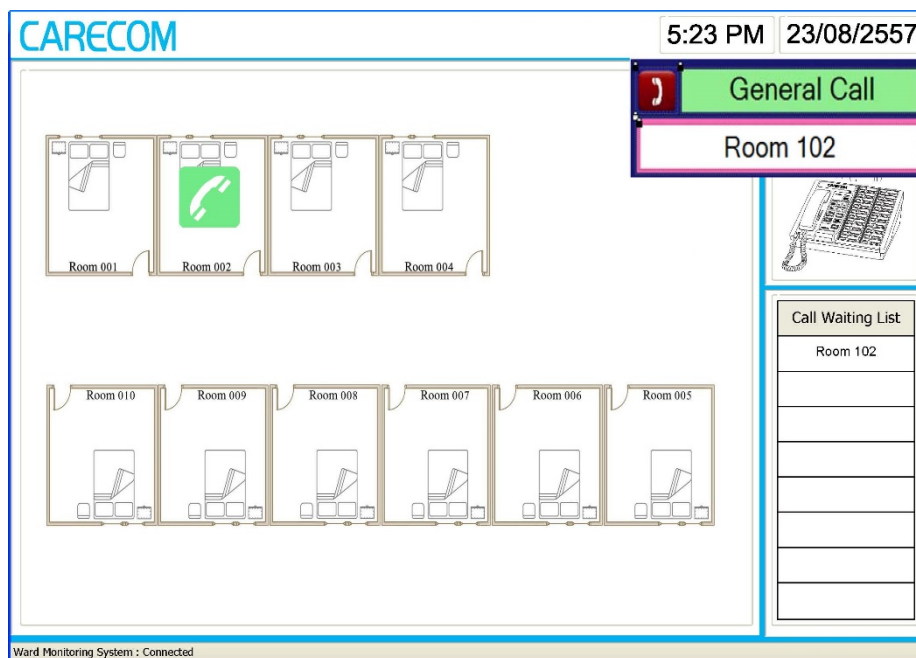
ภาพที่ 3-3 สถาปัตยกรรมซอฟต์แวร์ของ Display Node

หลังจากที่ Master node ทำการ publish ข้อมูลออกมา จากนั้นทางฝั่ง Display Node จะทำ
 ใ้ได้รับข้อมูลตาม topic ที่ Display Node subscribe ไว้ (ใช้ crossbar.io และ autobahn.ws โดยพัฒนา
 บน Python ซึ่งฝั่ง Display Node นี้ได้มีการติดตั้งและตั้งค่า crossbar.io และ autobahn.ws ไว้
 เรียบร้อยแล้ว) จากนั้นจะนำค่าที่ได้รับจากการ publish & subscribe มาประมวลผล และใช้ kivy ใน
 การแสดงผล โดย kivy พัฒนบน Python และต่อสาย HDMI เพื่อแสดงผลผ่านทางหน้าจอ

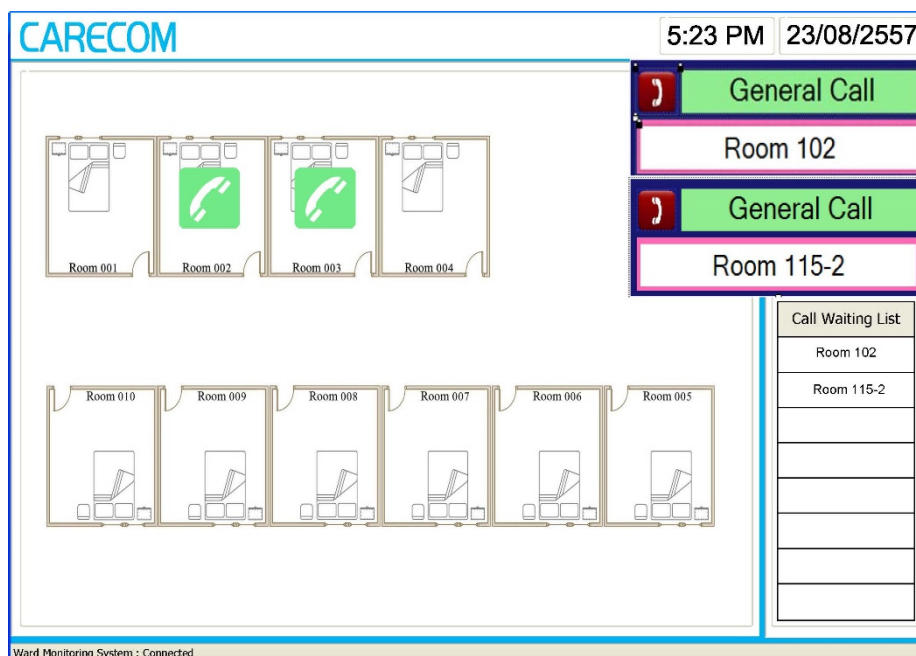
3.1.2 การออกแบบหน้าจอการแสดงผล



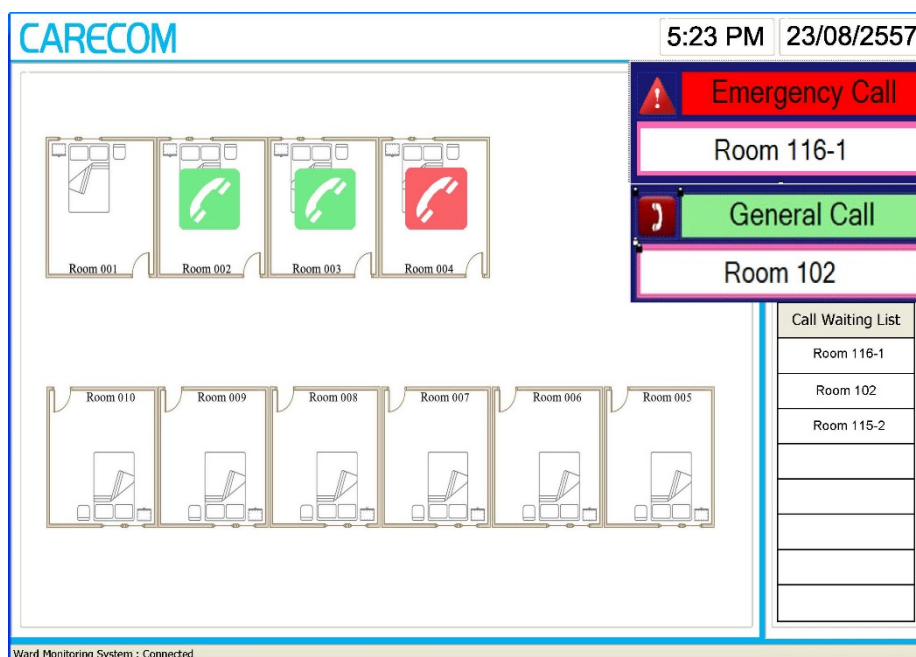
ภาพที่ 3-4 ตัวอย่างหน้าจอการแสดงผล



ภาพที่ 3-5 ตัวอย่างหน้าจอการแสดงผลเมื่อมี General Call เข้ามา



ภาพที่ 3-6 ตัวอย่างหน้าจอการแสดงผลเมื่อมี General Call เข้ามา 2 ห้อง

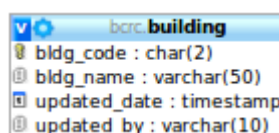


ภาพที่ 3-7 ตัวอย่างหน้าจอการแสดงผลเมื่อมี Emergency Call และ General call

ภาพตัวอย่างหน้าจอการแสดงผลทั้ง 4 ภาพ จะมีส่วนของ Call waiting list สำหรับแสดงผลว่า ณ เวลานั้น ๆ มีการ Call เข้ามาก็ห้อง ห้องไหนบ้าง โดยจะแสดงผล pop-up ตรงมุมขวาบนเพียง 2 ห้องเท่านั้น โดยจะเรียงลำดับจากความสำคัญคือ

1. Emergency Call มีความสำคัญมากกว่า General Call และ
2. ถ้า Call มีความสำคัญเท่ากันจะยึดความสำคัญตามเวลาที่มีการ Call เข้ามาก่อน

3.1.3 การออกแบบ Database



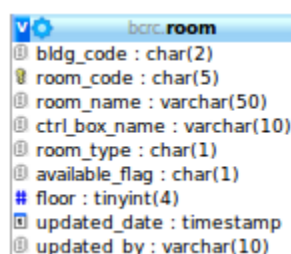
ภาพที่ 3-8 ตารางเก็บข้อมูลของตึก

ภาพที่ 3-8 ตารางเก็บข้อมูลของตึกนี้จะประกอบไปด้วย รหัสตึก ชื่อตึก วันที่อัปเดตและคนที่ทำการอัปเดตข้อมูล



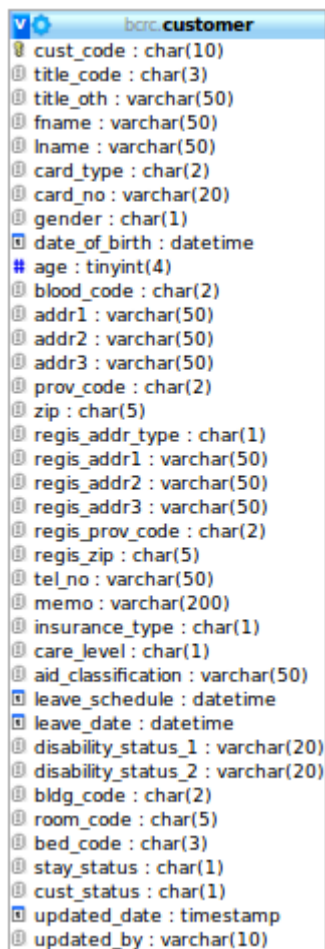
ภาพที่ 3-9 ตารางเก็บข้อมูลเตียง

ตารางเก็บข้อมูลเตียงประกอบด้วย รหัสตึก รหัสห้อง รหัสเตียง ชื่อเตียง channel_no และอื่น ๆ ซึ่ง channel_no นี้จะเป็นตัวบอกว่า call (General call & Emergency call) ที่เกิดขึ้นมาจากห้องไหน (เลขห้องที่ติดหน้าห้อง) นอกจากนี้ตารางนี้ยังเก็บข้อมูลตำแหน่ง x, y สำหรับนำไปใช้ในการแสดงผลบนหน้าจอ User Interface



ภาพที่ 3-10 ตารางเก็บข้อมูลห้อง

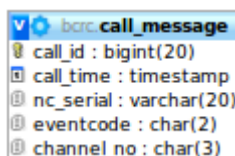
ภาพที่ 3-10 ตารางเก็บข้อมูลห้องประกอบด้วย รหัสตึก รหัสห้อง ชื่อห้อง และอื่น ๆ



bcrc.customer	
cust_code	: char(10)
title_code	: char(3)
title_oth	: varchar(50)
fname	: varchar(50)
lname	: varchar(50)
card_type	: char(2)
card_no	: varchar(20)
gender	: char(1)
date_of_birth	: datetime
age	: tinyint(4)
blood_code	: char(2)
addr1	: varchar(50)
addr2	: varchar(50)
addr3	: varchar(50)
prov_code	: char(2)
zip	: char(5)
regis_addr_type	: char(1)
regis_addr1	: varchar(50)
regis_addr2	: varchar(50)
regis_addr3	: varchar(50)
regis_prov_code	: char(2)
regis_zip	: char(5)
tel_no	: varchar(50)
memo	: varchar(200)
insurance_type	: char(1)
care_level	: char(1)
aid_classification	: varchar(50)
leave_schedule	: datetime
leave_date	: datetime
disability_status_1	: varchar(20)
disability_status_2	: varchar(20)
bldg_code	: char(2)
room_code	: char(5)
bed_code	: char(3)
stay_status	: char(1)
cust_status	: char(1)
updated_date	: timestamp
updated_by	: varchar(10)

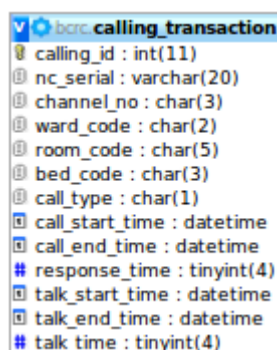
ภาพที่ 3-11 ตารางคนไข้

ตารางคนไข้ เก็บข้อมูลต่าง ๆ ที่เกี่ยวกับคนไข้ อาทิเช่น ชื่อ ที่อยู่ วันเดือนปีเกิด รหัสบัตรประชาชน เบอร์โทรศัพท์ อาการ รหัสตึก รหัสห้อง รหัสเตียง และอื่น ๆ



bcrc.call_message	
call_id	: bigint(20)
call_time	: timestamp
nc_serial	: varchar(20)
eventcode	: char(2)
channel_no	: char(3)

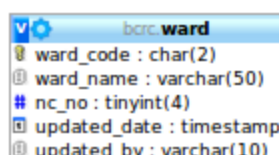
ภาพที่ 3-12 ตารางสำหรับเก็บข้อมูลดิบที่เกิดจากการ call จากเครื่อง BCRC



brc.calling_transaction	
calling_id	int(11)
nc_serial	varchar(20)
channel_no	char(3)
ward_code	char(2)
room_code	char(5)
bed_code	char(3)
call_type	char(1)
call_start_time	datetime
call_end_time	datetime
response_time	tinyint(4)
talk_start_time	datetime
talk_end_time	datetime
talk_time	tinyint(4)

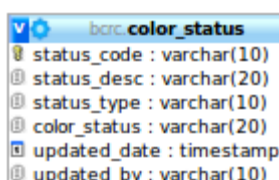
ภาพที่ 3-13 ตารางสำหรับเก็บข้อมูลในการทำ report

ตารางสำหรับนำข้อมูลดิบมาเก็บเป็นข้อมูลเพื่อนำไปใช้สำหรับการทำ report ต่อไป ซึ่งประกอบด้วยข้อมูล channel_no รหัสห้อง รหัสเตียง ประเภทการ call (General call หรือ Emergency call) เวลาที่เริ่มมีการ call และเวลาที่มีการตอบสนอง ระยะเวลาในการตอบสนอง เวลาที่เริ่มคุย เวลาที่วาง และระยะเวลาในการคุย เป็นต้น



brc.ward	
ward_code	char(2)
ward_name	varchar(50)
nc_no	tinyint(4)
updated_date	timestamp
updated_by	varchar(10)

ภาพที่ 3-14 ตารางเก็บข้อมูลในเวิร์ดพยาบาล



brc.color_status	
status_code	varchar(10)
status_desc	varchar(20)
status_type	varchar(10)
color_status	varchar(20)
updated_date	timestamp
updated_by	varchar(10)

ภาพที่ 3-15 ตารางเก็บข้อมูลสำหรับใช้แสดงผล

การเชื่อมโยงของฐานข้อมูล

- tb_bed จะมีการดึงข้อมูล bldg_code จาก tb_building และ room_code จาก tb_room เพื่อบอกตำแหน่งห้องและเตียงที่ผู้ป่วยพัก
- tb_room จะมีการดึงข้อมูล bldg_code จาก tb_building เพื่อบอกถึงเตียงที่ผู้ป่วยพัก

- tb_customer จะมีการดึงข้อมูล bldg_code จาก tb_building, room_code จาก tb_room และ bed_code จาก tb_bed เพื่อบอกเตียง หมายเลขห้องพัก และเตียงที่ผู้ป่วยพักอยู่

3.2 ติดตั้งซอฟต์แวร์สำหรับงานบนบอร์ด Raspberry Pi ซึ่งประกอบด้วย

3.2.1 PySerial 2.7

```
sudo apt-get install python-serial
```

3.2.2 autobahn.ws & crossbar.io (<http://crossbar.io/docs/Installation-on-the-RaspberryPi/>)

1. อัปเดต (Update) ระบบปฏิบัติการ

```
sudo apt-get update
sudo apt-get -y dist-upgrade
```

2. Install prerequisites

```
sudo apt-get install -y build-essential libssl-dev libffi-dev
python-dev
```

ติดตั้ง pip

```
wget https://bootstrap.pypa.io/get-pip.py
sudo python get-pip.py
```

3. Install Crossbar.io

```
sudo pip install crossbar
```

4. Test installation

```
crossbar version
```

5. Installing Autobahn

```
pip install autobahn
```

6. check installation

```
>>> from autobahn import __version__
>>> print(__version__)
0.9.1
```

3.2.3 kivy

การติดตั้งบนระบบปฏิบัติการ Raspbian มีขั้นตอนดังนี้

1. เพิ่ม APT sources สำหรับ Gstreamer 1.0 ใน directory `/etc/apt/sources.list` ดังนี้

```
deb http://vontaene.de/raspbian-updates/ . main
```

2. เพิ่ม APT key สำหรับ vontaene.de ดังนี้

```
apt-get install debian-keyring
```

```
gpg --keyserver pgp.mit.edu --recv-keys 0C667A3E
gpg --armor --export 0C667A3E | apt-key add -
```

3. อัปเดตระบบปฏิบัติการและติดตั้งส่วนเสริม

```
sudo apt-get update
sudo apt-get install pkg-config libgl1-mesa-dev libgles2-mesa-
dev \ python-pygame python-setuptools libgstreamer1.0-dev git-
core \ gstreamer1.0-plugins-{bad,base,good,ugly} \
gstreamer1.0-{omx,alsa} python-dev sudo apt-get update
```

4. ติดตั้ง pip

```
wget https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py
sudo python get-pip.py
```

5. ติดตั้ง Cython (กรณี debian package เป็นเวอร์ชันเก่า)

```
sudo pip install cython==version0.20
```

6. ถัดลอกไฟล์และ compile Kivy

```
git clone https://github.com/kivy/kivy
cd kivy
```

7. ติดตั้ง Kivy เวอร์ชัน globally

```
python setup.py build
sudo python setup.py install
```

3.3 จัดทำระบบ/เขียนโปรแกรมสำหรับทำงาน

ระบบจะแบ่งเป็น 2 ส่วนหลัก คือ

1. ส่วน Master Node ทำหน้าที่ควบคุมกำหนดเส้นทางและการติดต่อสื่อสารระหว่างชุดควบคุมหรือเครื่อง BCRC

2. ส่วน Display Node ทำหน้าที่กำหนดการแสดงผลการแจ้งเตือนบนหน้าจอ

ระบบทั้งสองจะถูกพัฒนาด้วย คอมพิวเตอร์ที่รันด้วยระบบปฏิบัติการ Ubuntu 12.04 โดยทั้งสองส่วนพัฒนาร่วมกับเครื่อง BCRC เพื่อคอยตรวจสอบความถูกต้อง จากนั้นจึงติดตั้งและเตรียมส่วนต่าง ๆ ที่จำเป็นบนบอร์ด Raspberry Pi เพื่อนำระบบที่พัฒนาจนสามารถทำงานได้ในระดับหนึ่ง นำมาทดสอบการใช้งานต่อร่วมกับเครื่อง BCRC อีกครั้งและแก้ไขปัญหาที่เกิดขึ้นต่อไป

การทดสอบระบบด้วยบอร์ด Raspberry Pi โมเดล B ในการทดสอบระบบ โดยใช้ระบบปฏิบัติการ Raspbian ทำการติดตั้ง Autobahn, Crossbar.io และ kivy เพื่อให้สามารถรันระบบได้เหมือนที่จำลองบนคอมพิวเตอร์

การวางแผนขั้นตอนวิธีการต่าง ๆ นั้นเพื่อเป็นแนวทางในเบื้องต้นและช่วยให้การดำเนินงานเป็นได้โดยสะดวกมากขึ้น การแบ่งงานเป็นส่วนย่อย ๆ จะทำให้สามารถคาดคะเนระยะเวลาที่ใช้สำหรับแต่ละส่วนงานได้ และสามารถแก้ไขปัญหาที่เกิดขึ้นแต่ละจุด โดยเฉพาะส่วนหลักของงานชิ้นนี้จำเป็นต้องมีความเข้าใจเกี่ยวกับโครงสร้างสถาปัตยกรรมของระบบ เข้าใจแนวคิดวิธีการหน้าที่ของแต่ละส่วนว่าควรจะทำงานอย่างไร เพื่อให้สามารถคิดหาแนวทางในการพัฒนาระบบต่อไปได้อย่างเหมาะสม

บทที่ 4

ผลการดำเนินงาน

ขั้นตอนการดำเนินในบทที่ผ่านมากล่าวถึงการสร้างระบบแสดงผลการแจ้งเตือนของเครื่อง BCRC ทั้งการดำเนินการติดตั้งซอฟต์แวร์ต่าง ๆ บน Raspberry Pi เพื่อให้บอร์ด Raspberry Pi สามารถรองรับการทำงานในการรับส่งข้อมูลกับเครื่อง BCRC และสามารถดึงข้อมูลมาแสดงผลบนจอแสดงผล ได้ตรงตามความต้องการของระบบที่ได้ออกแบบกำหนดไว้ มีรายละเอียดผลการดำเนินงาน แบ่งเป็น 2 ส่วนหลักดังนี้

1. ส่วนประมวลผลที่หนึ่งซึ่งทำหน้าที่ติดต่อและแปลงข้อมูลที่ได้รับจากเครื่อง BCRC (Master Node)
2. ส่วนประมวลผลที่สองทำหน้าที่ประมวลการแสดงผลข้อมูลออกทางหน้าจอแสดงผล (Display Node)

การทดลองระบบแสดงผลเรียกพยาบาลจัดเตรียมการทดสอบในห้องปฏิบัติการเครือข่ายในคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ซึ่งใช้เพื่อจำลองแทนสถานพยาบาลจริง โดยคาดว่าจะมีสภาพแวดล้อมติดตั้งอยู่ในห้องที่มีการติดตั้งเครื่องปรับอากาศ หน่วยประมวลผลเชื่อมต่อเข้ากับเครือข่ายในวงแลน (LAN) เดียวกัน และเครือข่ายนั้นสามารถเชื่อมต่อถึงกันได้อย่างราบรื่นไม่มีปัญหา เครื่องเรียกพยาบาลติดตั้งกับชุดสวิทช์สำหรับกดเรียกพยาบาลที่โดยปกติจะติดตั้งอยู่ประจำแต่ละห้องพักรักษาตัว

องค์ประกอบที่ใช้ในการทดสอบ ได้แก่ เครื่องเรียกพยาบาล 1 เครื่อง ชุดสวิทช์สำหรับติดตั้งในห้องพักรักษาตัว 3 ชุด แยกย่อยเป็นสวิทช์สำหรับ General call และ Emergency call ปุ่มเพื่อกดรีเซต พร้อมไฟแสดงสถานะการเรียก โดยที่เครื่องเรียกพยาบาลมีการต่อเชื่อมเข้ากับบอร์ด Raspberry Pi สำหรับ Master Node ผ่านพอร์ต RS232 เพื่อเก็บข้อมูลที่สร้างจากเครื่องเรียกพยาบาลมาเก็บไว้ในฐานข้อมูลของเซิร์ฟเวอร์และนำไปติดต่อกับส่วนประมวลผลเพื่อแสดงผลออกทางหน้าจอ การทดสอบจะใช้โปรแกรม Putty เพื่อเชื่อมต่อแบบ SSH ไปยังบอร์ด Raspberry Pi และบอร์ด

Raspberry Pi อีกหนึ่งตัวสำหรับ Display Node ซึ่งมีการต่อเข้ากับหน้าจอผ่านพอร์ต HDMI พร้อมเมาส์และคีย์บอร์ดเพื่อใช้ในการเขียนโปรแกรมแสดงผล GUI

4.1 Master Node

Master node เป็นส่วนการทำงานที่เรียกว่า Backend เพื่อนำข้อมูลที่ได้มาจากเครื่องเรียกพยาบาล รุ่น BCRC ระบบบัส RS232 ส่งค่าต่อไปแสดงผลบน user interface ผ่าน crossbar และเก็บข้อมูลลง database ผลการทดลองในส่วนนี้จะสามารถตรวจสอบได้จาก log เพื่อแสดงผลว่าการทำงานนั้น ๆ ถูกต้องหรือไม่

```
wattana@wattana-VirtualBox:~/Desktop/bcrcnursecallwamp/communication$ python bcr
c232wamp.py
Component created..
Transport connected..
Session joined..
Start bcrc232 thread...
2015-06-04 19:45:42,575 - BCRC-Serial232 - INFO - Starting BCRC communication se
rial protocol!
2015-06-04 19:45:42,578 - BCRC-Serial232 - DEBUG - Trying to connect serial port
...
2015-06-04 19:45:42,844 - BCRC-Serial232 - INFO - Serial connected! portname /de
v/ttyUSB0
```

ภาพที่ 4-1 ภาพตัวอย่างการทำงาน เมื่อเริ่มให้โปรแกรมเริ่มทำงาน

```
2015-06-04 19:45:44,557 - BCRC-Serial232 - DEBUG - Received >> STX
2015-06-04 19:45:44,570 - BCRC-Serial232 - INFO - Send << ACK
2015-06-04 19:45:44,571 - BCRC-Serial232 - DEBUG - Turn On BCRC
('System: ', None, None)
```

ภาพที่ 4-2 ภาพตัวอย่างเมื่อเปิดเครื่อง BCRC

```
2015-06-04 19:46:24,468 - BCRC-Serial232 - DEBUG - Received >> STX
2015-06-04 19:46:24,482 - BCRC-Serial232 - INFO - Send << ACK
2015-06-04 19:46:24,483 - BCRC-Serial232 - DEBUG - Received General Call on Channel(005)
('System: ', '005', '005')
```

ภาพที่ 4-3 ภาพตัวอย่างเมื่อมีการกดเรียกพยาบาลจากข้างเตียงผู้ป่วย (General Call) จากห้อง 005

```

2015-06-04 19:47:03,624 - BCRC-Serial232 - DEBUG - Received >> STX
2015-06-04 19:47:03,637 - BCRC-Serial232 - INFO - Send << ACK
2015-06-04 19:47:03,637 - BCRC-Serial232 - DEBUG - Received Call Reset on Channel(005)
('System: ', ' ', '005')

```

ภาพที่ 4-4 ภาพตัวอย่างเมื่อมีการกดรีเซ็ตการเรียกพยาบาลจากข้างเตียงผู้ป่วยห้อง 005

```

2015-06-04 19:47:35,033 - BCRC-Serial232 - DEBUG - Received >> STX
2015-06-04 19:47:35,047 - BCRC-Serial232 - INFO - Send << ACK
2015-06-04 19:47:35,048 - BCRC-Serial232 - DEBUG - Received Emergency Call on Channel(005)
('System: ', ' ', '005')
2015-06-04 19:47:39,055 - BCRC-Serial232 - DEBUG - Received >> STX
2015-06-04 19:47:39,069 - BCRC-Serial232 - INFO - Send << ACK
2015-06-04 19:47:39,069 - BCRC-Serial232 - DEBUG - Received Call Reset on Channel(005)
('System: ', ' ', '005')

```

ภาพที่ 4-5 ภาพตัวอย่างเมื่อมีการกดการเรียกพยาบาลจากห้องน้ำ (Emergency Call) และมีการกดรีเซ็ตจากห้อง 005

```

2015-06-04 19:46:24,468 - BCRC-Serial232 - DEBUG - Received >> STX
2015-06-04 19:46:24,482 - BCRC-Serial232 - INFO - Send << ACK
2015-06-04 19:46:24,483 - BCRC-Serial232 - DEBUG - Received General Call on Channel(005)
('System: ', ' ', '005')
2015-06-04 19:47:03,624 - BCRC-Serial232 - DEBUG - Received >> STX
2015-06-04 19:47:03,637 - BCRC-Serial232 - INFO - Send << ACK
2015-06-04 19:47:03,637 - BCRC-Serial232 - DEBUG - Received Call Reset on Channel(005)
('System: ', ' ', '005')
2015-06-04 19:47:35,033 - BCRC-Serial232 - DEBUG - Received >> STX
2015-06-04 19:47:35,047 - BCRC-Serial232 - INFO - Send << ACK
2015-06-04 19:47:35,048 - BCRC-Serial232 - DEBUG - Received Emergency Call on Channel(005)
('System: ', ' ', '005')
2015-06-04 19:47:39,055 - BCRC-Serial232 - DEBUG - Received >> STX
2015-06-04 19:47:39,069 - BCRC-Serial232 - INFO - Send << ACK
2015-06-04 19:47:39,069 - BCRC-Serial232 - DEBUG - Received Call Reset on Channel(005)
('System: ', ' ', '005')

```

ภาพที่ 4-6 ภาพตัวอย่างเมื่อมีการกดการเรียกพยาบาลจากเตียงผู้ป่วย (General Call) และมีการกดหูเพื่อคุยกับผู้ป่วยจากห้อง 005

```

2015-06-04 19:47:05+0700 [WampWebSocketClientProtocol,client] adbapl connecting: MySQLdb {'init_command': 'SELECT 1', 'db': 'brcrc', 'connect_timeout': 5, 'passwd': 'u'password', 'host': 'u'127.0.0.1', 'user': 'u'root', 'port': 3306}
2015-06-04 19:47:05+0700 [WampWebSocketClientProtocol,client] Saved call transaction: serial: brcrc-serial-demo1 channel: 005 detail: {'call_end_time': '2015-06-04 19:47:03',
'call_start_time': '2015-06-04 19:46:24',
'call_state': False,
'status': ' ',
'talk_end_time': None,
'talk_start_time': None}

```

ภาพที่ 4-7 ภาพตัวอย่างผลการส่งข้อมูลไปลง database

4.2 Display Node

4.2.1 การพัฒนาการแสดงผล มีข้อกำหนดในการแสดงผลดังนี้

4.2.1.1 แบ่งการแสดงผลบนหน้าจอเป็นสองส่วนคือ ส่วนแสดงแผนผังของหอพักผู้ป่วย (Ward) ในชั้นที่มีการเฝ้าดู และ ส่วนแสดงข้อมูลห้องพักรักษาผู้ป่วยที่มีการโทรเข้ามา

4.2.1.2 การจัดการลำดับการโทรเลือกจาก Emergency call ก่อนจากนั้นจึงตามด้วย General call หากมีระดับความฉุกเฉินเท่ากันให้เรียงตามลำดับที่มีการโทรเข้ามาก่อน

4.2.1.3 เมื่อมีการโทรเข้ามาแล้วจะมีการแจ้งเตือน โดยต้องมีสัญลักษณ์เตือน ณ ตำแหน่งห้องพักรักษาผู้ป่วยนั้น พร้อมแสดงระยะเวลาที่โทรเข้ามา เพื่อให้ทราบว่ามีสายเรียกเข้าจากห้องนี้นานเท่าใดแล้ว สัญลักษณ์ที่ปรากฏนั้นแบ่งสีตามระดับความฉุกเฉิน

- สีเขียว สำหรับ General call
- สีแดง สำหรับ Emergency call

4.2.1.4 พร้อมกันนั้นเมื่อมีการโทรเข้าจะต้องแสดงหน้าต่างเตือนในด้านมุมบนขวาของจอ ซึ่งกำหนดให้แสดงได้สูงสุด 2 หน้าต่าง หากมีสายเรียกเข้าเกินกว่านั้นจะแสดงอยู่ในส่วนของ Waiting list

4.2.1.5 Waiting list ใช้หลักการเรียงตามลำดับก่อนหลังในการโทรเข้ามาเช่นเดียวกับการแจ้งเตือนในข้อ 2. หากมีจำนวนเกินกว่าที่จะแสดงได้ สามารถเลื่อนดูลำดับได้ (Scroll view)

4.2.1.6 การแสดงข้อความบนหน้าต่างแจ้งเตือน เบื้องต้นกำหนดให้แสดงได้สูงสุด 8 ตัวอักษร (แต่สามารถปรับเปลี่ยนเงื่อนไขดังกล่าวได้)

4.2.1.7 มีส่วนแสดงสถานะ การเชื่อมต่อกับเครื่อง BCRC

4.2.1.8 แสดงเวลาและวันที่ในปัจจุบัน

4.2.2 การเขียนโปรแกรมด้วย Kivy [12]

การใช้ Kivy สร้างแอปพลิเคชันสำหรับแสดงผลจะใช้ภาษา Python เป็นหลักในการพัฒนา นอกจากนี้ Kivy ยังมีส่วนของภาษา Kv หรือภาษา Kivy เพื่อใช้สำหรับออกแบบซึ่งแยกจากส่วนการเขียนโปรแกรมแอปพลิเคชัน (Application) ระบบเงื่อนไขต่าง ๆ เหมาะสำหรับการเขียนระบบที่มีความซับซ้อนมากยิ่งขึ้น

การสร้างแอปพลิเคชันอย่างง่ายจะต้องประกอบไปด้วย

- การสร้างคลาส (Class) แอปพลิเคชัน
- ฟังก์ชัน build() เพื่อใช้คืนค่า (Return) Widget instance ซึ่งตรงนี้จะป็นรากของการเก็บฝังต้นไม้ Widget

- การสร้าง instance ของคลาสแอปพลิเคชัน เพื่อให้มีการเรียกใช้งาน

การเขียนแอปพลิเคชันด้วย Kivy จะเป็นการสร้างวัตถุซ้อนขึ้นมาในผังหรือที่เรียกว่า Layout ซึ่งจะมีหลากหลายลักษณะ จากนั้นจะทำการสร้างวัตถุต่าง ๆ ไม่ว่าจะเป็นข้อความ ปุ่ม รูปทรงเลขคณิต หรือผังที่สร้างซ้อนขึ้นมาอีกชั้นหนึ่งก็ตาม

```
import kivy
kivy.require('1.0.6') # replace with your current kivy version !

from kivy.app import App
from kivy.uix.label import Label

class MyApp(App):

    def build(self):
        return Label(text='Hello world')

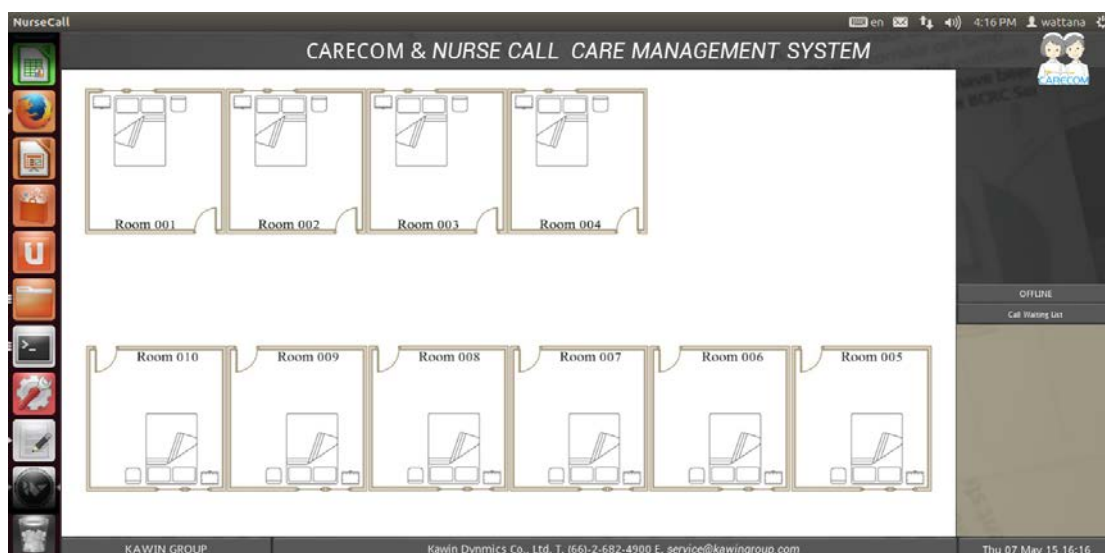
if __name__ == '__main__':
    MyApp().run()
```

ภาพที่ 4-8 แสดงตัวอย่างโค้ดอย่างง่ายที่ใช้ในการเขียน Kivy

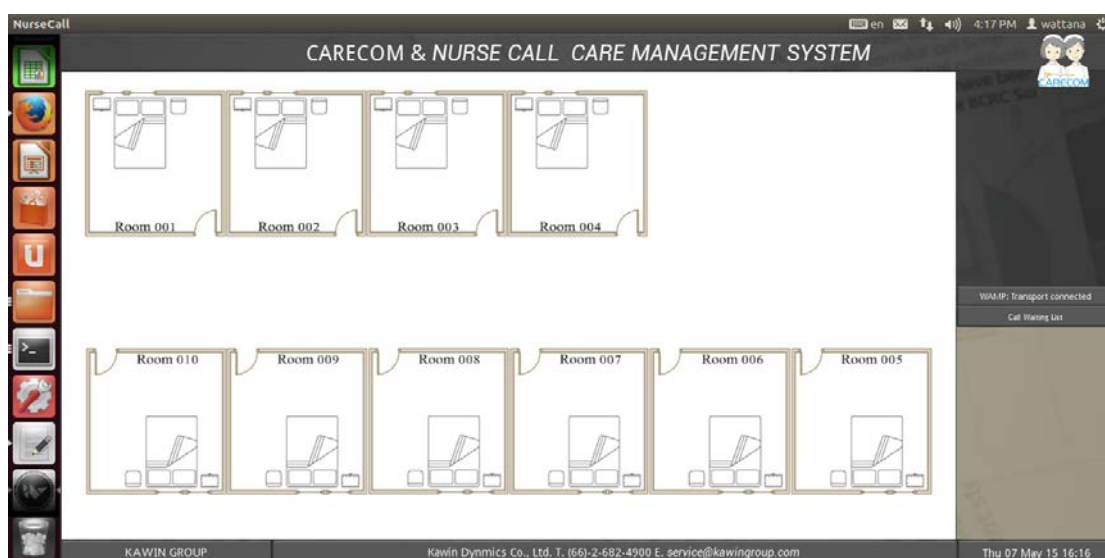
การรัน Kivy เพื่อทดสอบโค้ด ให้แสดงผลทางหน้าจอ คำสั่งที่ใช้รันจะแตกต่างกันไปตามระบบปฏิบัติการ ดังนี้

- Linux python main.py
- Windows python main.py หรือ C:\appdir>kivy.bat main.py
- Mac OS X kivy main.py

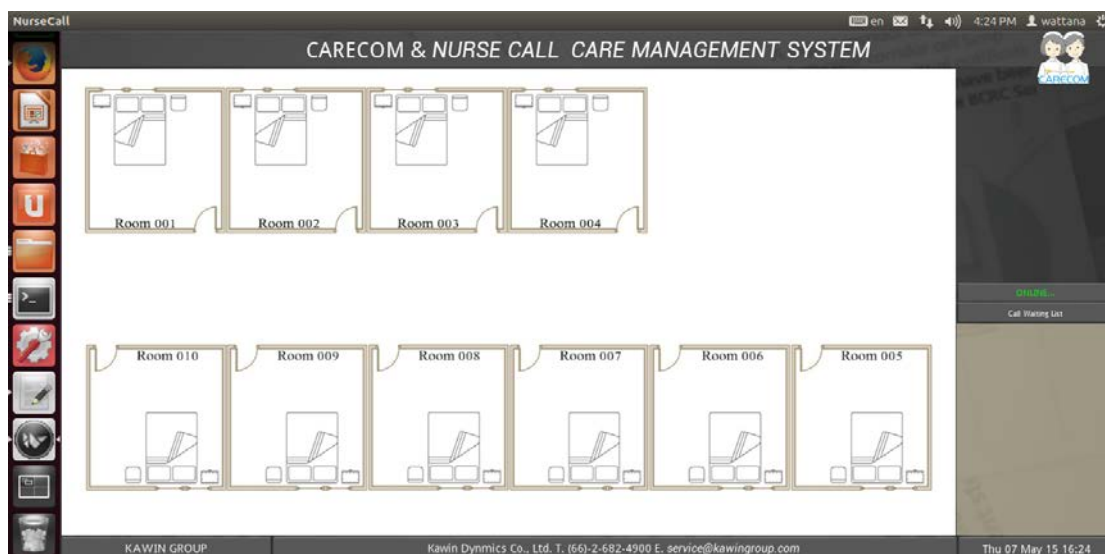
4.3 ภาพแสดงผลการดำเนินงาน ที่ได้จากการพัฒนาระบบ



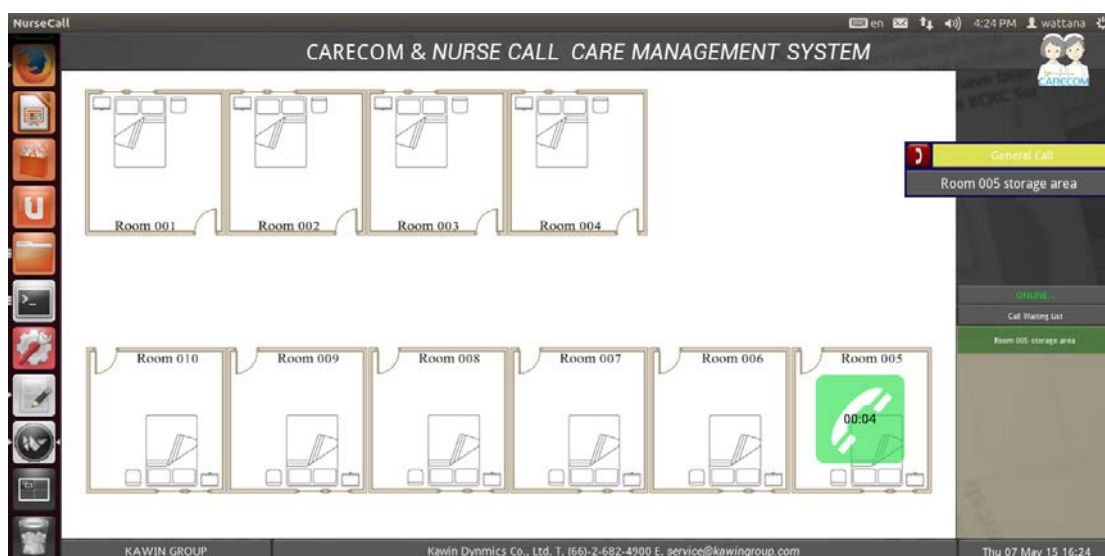
ภาพที่ 4-9 ภาพขณะเปิดหน้าจอครั้งแรก โดยยังไม่ได้ทำการต่อเชื่อมรับข้อมูลจาก Master node



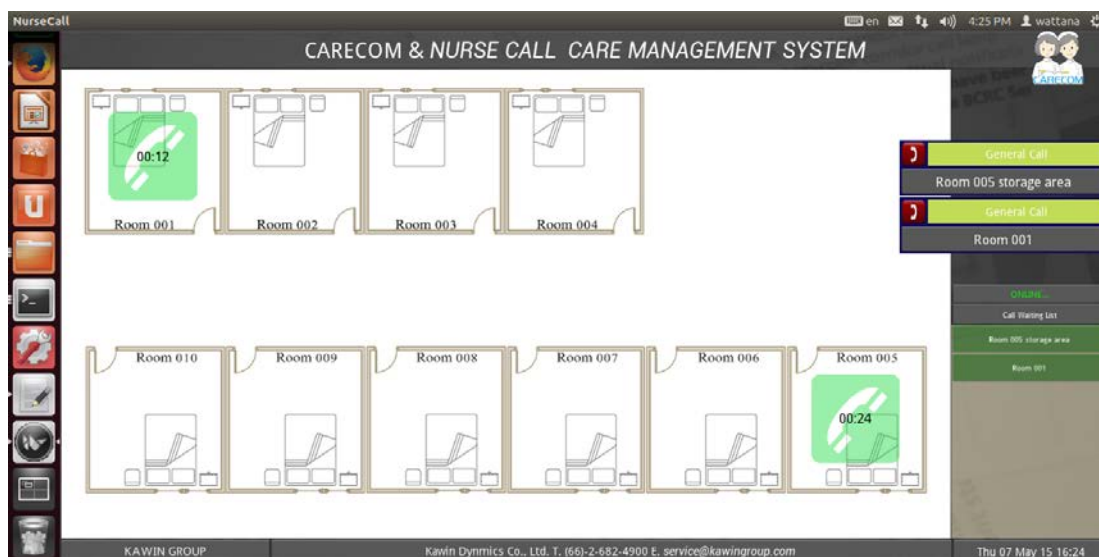
ภาพที่ 4-10 ภาพตัวอย่างหน้าจอขณะรอการเชื่อมต่อกับ Master node



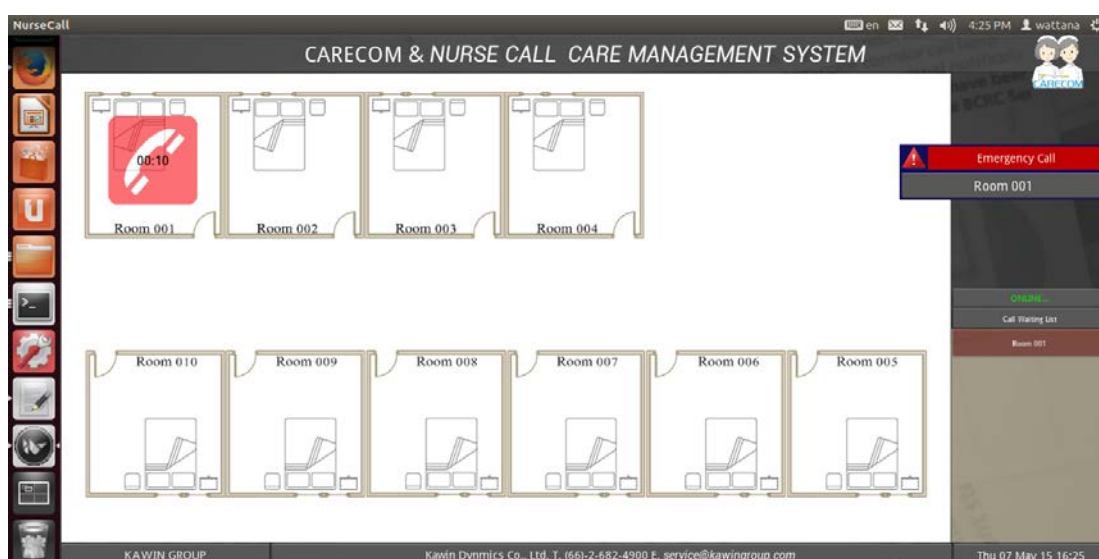
ภาพที่ 4-11 ภาพหลังจากเชื่อมต่อกับ Master node สำเร็จ



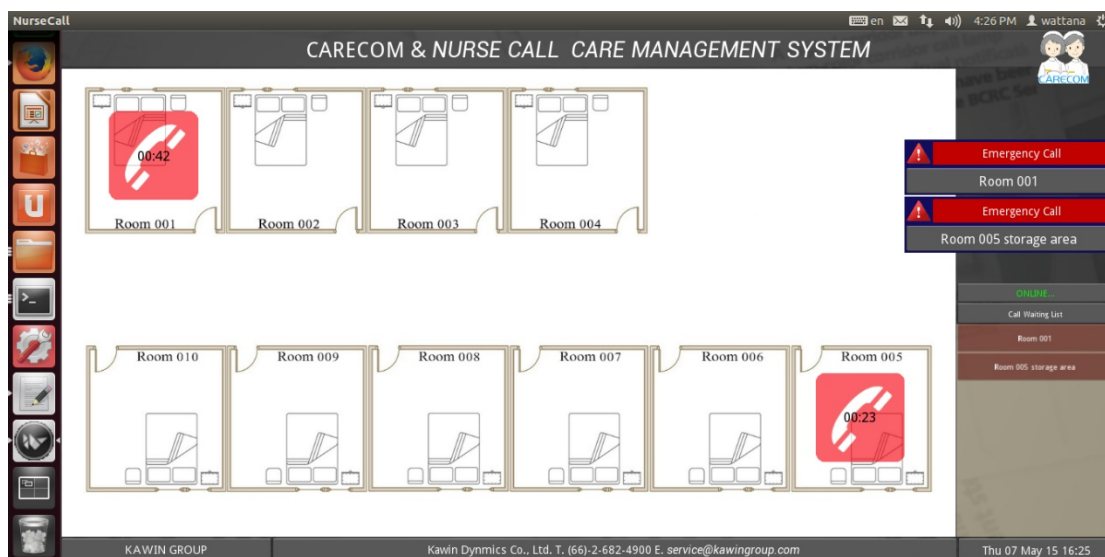
ภาพที่ 4-12 ภาพตัวอย่างที่มีการเรียก General Call จากห้อง 005



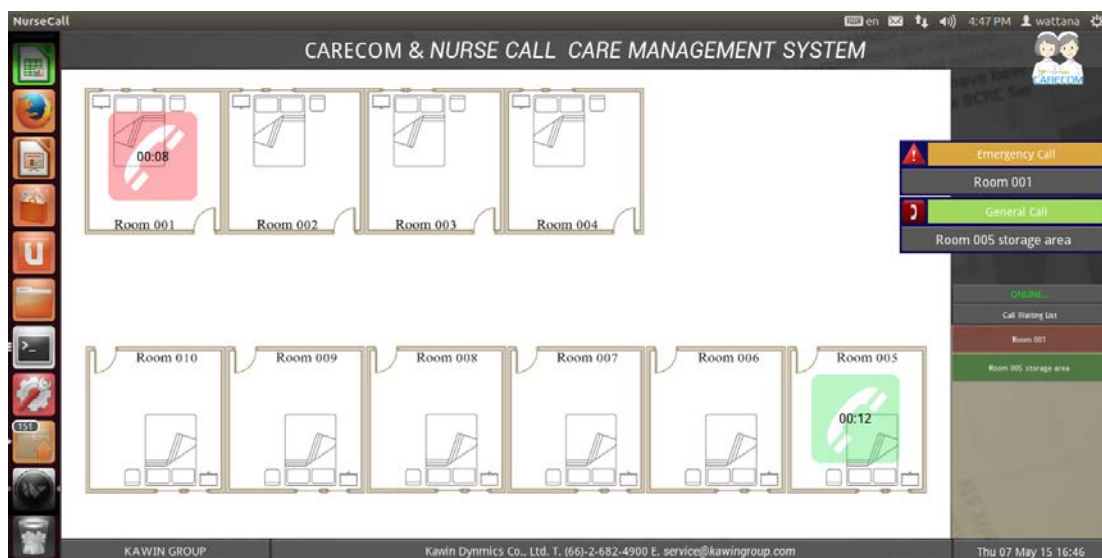
ภาพที่ 4-13 ภาพตัวอย่างที่มีการเรียก General Call จากห้อง 005 และ
เกิด General Call จากห้อง 001 แทรกเข้ามาภายหลัง



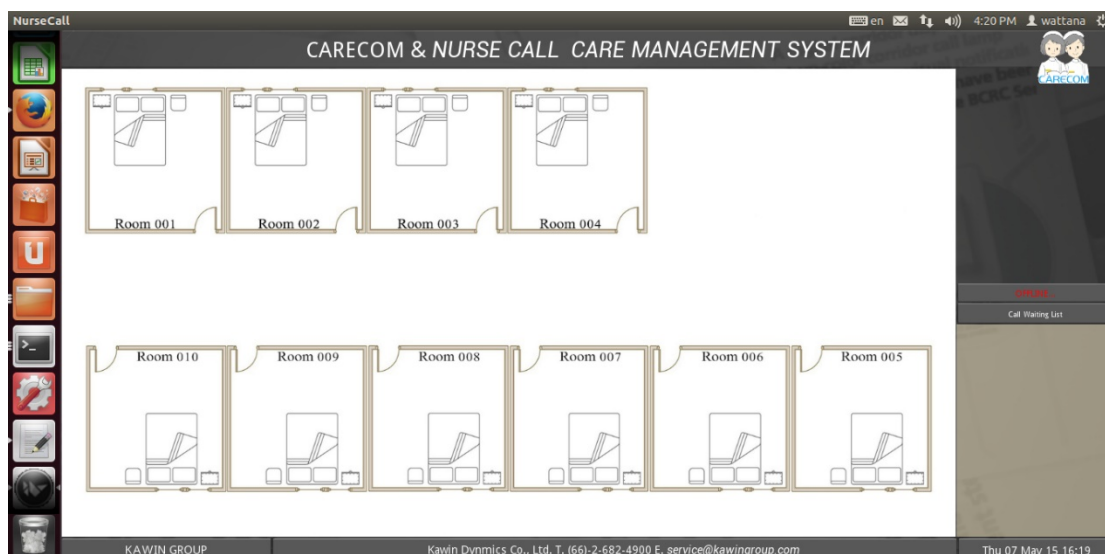
ภาพที่ 4-14 ภาพตัวอย่างที่มีการเรียก Emergency Call จากห้อง 001



ภาพที่ 4-15 ภาพตัวอย่างที่มีการเรียก Emergency Call จากห้อง 001 และ
มีการเรียก Emergency Call จากห้อง 005 แทรกเข้ามาภายหลัง



ภาพที่ 4-16 ภาพตัวอย่างที่มีการเรียก General Call จากห้อง 005 และ
มีการเรียก Emergency Call จากห้อง 001 แทรกเข้ามาภายหลัง



ภาพที่ 4-17 ภาพหลังจากหลุดจากการเชื่อมต่อกับ Master node

การทดลองนั้นมีข้อจำกัดในการเคลื่อนย้ายอุปกรณ์เนื่องจากกล่องใส่ป้องกันเครื่องเรียกพยาบาลมีน้ำหนักมากทำให้ไม่สะดวกในการเคลื่อนย้ายไปติดตั้งในบริเวณอื่นได้จึงส่งผลให้มีการทดสอบในสภาพแวดล้อมเพียงที่เดียวคือห้องปฏิบัติการเครือข่าย

ผลการทดลองในระยะแรกเป็นผลจากการจำลองการส่งข้อมูลของโปรแกรมที่เขียนขึ้นเลียนแบบการทำงานของเครื่องเรียกพยาบาล เพื่อใช้ในการพัฒนา Display Node

ขณะที่ Master Node มีการปรับเปลี่ยนการเก็บข้อมูลมาเก็บไว้ในฐานข้อมูลของเซิร์ฟเวอร์ ซึ่งแต่เดิมใช้ WAMP ในการจัดเก็บข้อมูลเพราะเป็นตัวกลางในการติดต่อระหว่าง Master node และ Display Node

บทที่ 5

สรุป

การสร้างระบบแสดงผลการแจ้งเตือนการเรียกเข้าเครื่อง BCRC บนจอแสดงผล โดยใช้เครื่อง BCRC ทำหน้าที่รับสัญญาณจากการเรียกของคนไข้และส่งข้อมูลไปยังบอร์ด Raspberry Pi ผ่าน Protocol RS232 และใช้บอร์ด Raspberry Pi เป็นส่วนรับข้อมูลไปเก็บไว้ในฐานข้อมูลของเว็บเซิร์ฟเวอร์และแสดงผลผ่าน Application User interface บนจอแสดงผล (ใน Raspberry Pi อีกตัวหนึ่ง) โดยผ่านเครือข่ายทั้งแบบมีสายและแบบไร้สาย

การนำบอร์ด Raspberry Pi มาพัฒนาเพื่อเป็นการเพิ่มความรู้ความเข้าใจและให้เห็นถึงความสามารถและประสิทธิภาพในการทำงานของบอร์ด Raspberry Pi โดยได้ติดตั้งซอฟต์แวร์ดังรายการต่อไปนี้

1. pySerial ทำหน้าที่เป็นตัวเชื่อมโยงไฟรอนให้สามารถส่งข้อมูลออกทางพอร์ตอนุกรมของคอมพิวเตอร์ได้

2. Crossbar.io และ Autobahn.ws ซึ่งเป็นการติดต่อสื่อสารรูปแบบใหม่ โดย Crossbar.io เปรียบเสมือน router ในการส่งข้อมูล และ Autobahn|Python เป็นไลบรารีสำหรับพัฒนา Application ด้วยภาษาไพธอน

ผลการดำเนินงานพบว่าบอร์ด Raspberry Pi สามารถแสดงผลการแจ้งเตือนการเรียกเข้าเครื่อง BCRC บนจอแสดงผลได้ ซึ่งสามารถติดตั้งระบบนี้ได้หลายจุด ในส่วนการพยาบาลมี Raspberry Pi หนึ่งตัวที่ทำหน้าที่เป็น Master Node ควบคุมการติดต่อสื่อสารข้อมูลกับเครื่อง BCRC แล้วนำข้อมูลที่ได้ไปกำหนดค่าในฐานข้อมูลและในขณะเดียวกันก็ส่งข้อมูลนั้นไปแสดงผลทางหน้าจอซึ่งเชื่อมต่อกับ Raspberry Pi อีกตัวหนึ่งผ่าน Crossbar.io ได้โดยที่หน้าจอแสดงผลนั้นสามารถแสดงผลการแจ้งเตือนได้เหมือนกับเครื่อง BCRC ที่แสดงผลทั้งแจ้งเตือนการเรียกเข้าแบบปกติและแบบฉุกเฉิน ทำให้พยาบาลสามารถดูผลของการแจ้งเตือนและเข้าถึงผู้ป่วยได้สะดวกและคล่องตัวมากขึ้น ไม่จำเป็นต้องดูการแสดงผลจากเครื่อง BCRC เพียงอย่างเดียว

การแสดงผลนี้ยังคงเป็นการสื่อสารแบบเพียงทางเดียว เพราะสามารถรับรู้ข้อมูลที่แสดงผ่านทางหน้าจอ ในการพัฒนาต่อขอระบบต่อไป อาจทำให้ระบบสามารถติดต่อสื่อสารแบบสองทางได้ เช่น วิธีการที่ทำให้พยาบาลสามารถกดตอบรับการเข้าไปให้บริการผู้ป่วยผ่านหน้าจอ ทำให้สามารถทราบได้ว่าใครเป็นคนเข้าไปให้บริการดูแลคนไข้ห้องไหน พร้อมทั้งไปปิดสัญญาณแจ้งเตือนที่ระบบ และยังสามารถต่อขอในส่วนการสื่อสารโดยนำมารวมเข้ากับอุปกรณ์อัจฉริยะ เช่น โทรศัพท์มือถือ หรือ นาฬิกาอัจฉริยะ เพื่อให้พยาบาลสามารถตอบสนองและมีการเข้าถึงผู้ป่วยได้สะดวกรวดเร็วมากยิ่งขึ้น และยังเป็นการช่วยให้พยาบาลสามารถดูแลผู้ป่วยที่พักในสถานพยาบาลได้อย่างทั่วถึง

เอกสารอ้างอิง

1. ministitch. ยุทธศาสตร์การวิจัยรายประเด็นด้านการเป็นศูนย์กลางทางการแพทย์ (Medical Hub). 2557. [ระบบออนไลน์]. แหล่งที่มา <http://www.kmutt.ac.th/rippc/nrct59/52s28.pdf> (5 มีนาคม 2558).
2. ระบบสารสนเทศโรงพยาบาล. [ระบบออนไลน์]. แหล่งที่มา <http://th.wikipedia.org/wiki/ระบบสารสนเทศโรงพยาบาล>. (5 มีนาคม 2558).
3. นพ.นวันรรณ ชีระอัมพรพันธ์. ระบบสารสนเทศโรงพยาบาล. [ระบบออนไลน์]. แหล่งที่มา <http://www.slideshare.net/nawanan/ss-16951566>. (5 มีนาคม 2558).
4. ระบบเรียกพยาบาล(Nurse Call System). [ระบบออนไลน์]. แหล่งที่มา [http://www.fourtern.com/content-ระบบเรียกพยาบาล\(NurseCallSystem\)-4-643-5816-1.html](http://www.fourtern.com/content-ระบบเรียกพยาบาล(NurseCallSystem)-4-643-5816-1.html). (5 มีนาคม 2558).
5. สมภพ โชคเทอดธรรม จิตติมา สมบูรณ์โกกกันท์ และ วิศรุต สาสนัส. “การพัฒนาระบบการให้บริการแบบกลุ่มเมฆเพื่อเพิ่มประสิทธิภาพการใช้งานบนเครื่องแม่ข่ายที่กระจายอยู่ในเครือข่ายแกนหลัก และระบบแสดงผลการแจ้งเตือนเครื่อง BCRC.” วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2556.
6. อรรถรัตน์ นาวิกาวดาร และ ไชโย ธรรมรัตน์. ความรู้เบื้องต้นเกี่ยวกับระบบเรียกพยาบาล. [ระบบออนไลน์]. แหล่งที่มา <http://www.secu.co.th/index.php?lay=show&ac=article&Id=539682197>. (5 มีนาคม 2558).
7. CARECOM CO., LTD. CARECOM Nurse Call & Care Management System. [ระบบออนไลน์]. แหล่งที่มา http://www.sst.co.th/file_content/pdf/11152111050pdf.pdf. (5 มีนาคม 2558).
8. ETT CO.,LTD. เริ่มต้นการใช้งานบอร์ด Raspberry Pi. [ระบบออนไลน์]. แหล่งที่มา <http://www.ett.co.th/prod2013/et-rasbery-pi/Getting%20Start%20Raspberry%20Pi.pdf>. (5 มีนาคม 2558).
9. ดร.เรวัต ศิริ โกลาภิรมย์. เอกสารประกอบการเรียนวิชา High-Level Design for Digital Systems. สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2555.

เอกสารอ้างอิง (ต่อ)

10. การใช้งานพอร์ตอนุกรม RS232. [ระบบออนไลน์]. แหล่งที่มา <http://www.thaimicrotron.com/CCS-628/Reference/RS232.htm>. (5 มีนาคม 2558).
11. pySerial โมดูลสำหรับใช้ python ติดต่อกับ serial port. [ระบบออนไลน์]. แหล่งที่มา <https://sites.google.com/site/pythonclassroom/module-package-python/pySerial-module>. (5 มีนาคม 2558).
12. Kivy; User's Guide. [Online] Available from: [https:// http://kivy.org](https://http://kivy.org).
13. WAMP; Why WAMP?. [Online] Available from: <http://wamp.ws>.
14. Crossbar.io; Application Scenarios. [Online] Available from: <http://crossbar.io/docs/Application-Scenarios/>
15. Autobahn|Python. Tavendo GmbH. [Online] Available from: <http://autobahn.ws/python/websocket/programming.html>
16. มายเอสคิวเอล. [ระบบออนไลน์]. แหล่งที่มา <https://th.wikipedia.org/wiki/มายเอสคิวเอล>. (5 มีนาคม 2558).
17. paramad. ระบบสารสนเทศในโรงพยาบาล. [ระบบออนไลน์]. แหล่งที่มา <https://bit2alone.wordpress.com/4-2/>. (5 มีนาคม 2558).
18. What the Internet of Things (IoT) Needs to Become a Reality. [Online] Available from: http://www.freescale.com/files/32bit/doc/white_paper/INTOTHNGSWP.pdf.
19. Raspberry Pi. United Kingdom: Raspberry Pi Foundation. [Online] Available from: <https://www.raspberrypi.org>.

ประวัติผู้แต่ง

ปริญญานิพนธ์เรื่อง : การเชื่อมต่ออินเทอร์เน็ตเข้ากับสภาพแวดล้อมรอบตัว
(ระบบแสดงผลการเฝ้าดูและแจ้งเตือนของระบบดูแลและการจัดการ
เรียกพยาบาล)

สาขาวิชา : วิศวกรรมคอมพิวเตอร์

ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะ : วิศวกรรมศาสตร์

ชื่อ : นางสาวธนพร เดชาวิจิตรเลิศ

ประวัติ

เกิดวันที่ 25 มกราคม พ.ศ. 2536 อยู่บ้านเลขที่ 56 ซอยบางแวก 140 แขวงบางไผ่ เขตบางแค กรุงเทพมหานคร สำเร็จการศึกษาในระดับปริญญาตรี สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2557

ชื่อ : นางสาวธัญรดา เกรือรัตน์ชัย

ประวัติ

เกิดวันที่ 5 ธันวาคม พ.ศ. 2535 อยู่บ้านเลขที่ 895/32 ซอยปทุมณวิถี 48 ถนนสุขุมวิท 101 แขวงบางจาก เขตพระโขนง กรุงเทพมหานคร สำเร็จการศึกษาในระดับปริญญาตรี สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2557