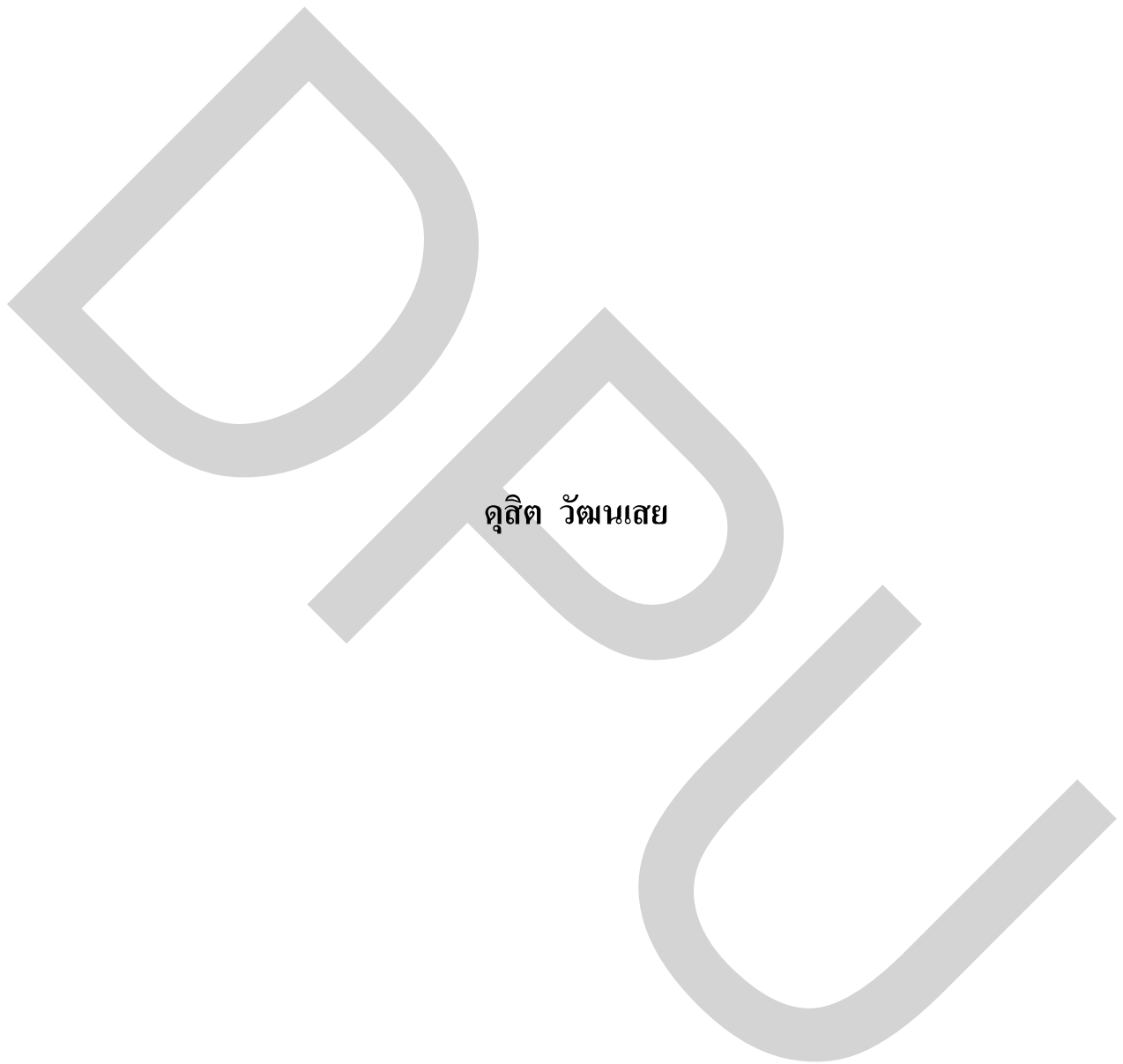


## ระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิศวกรรมคอมพิวเตอร์และโทรคมนาคมบัณฑิตวิทยาลัยมหาวิทยาลัยราชภัฏวชิรวิทยาคาร

พ.ศ. 2555

**IVR based Alarm and Control Systems for Electrical Equipment using  
VoIP System**



**Dusit Wattanasoei**

**A Thesis Submitted in Partial Fulfillment of the Requirements**

**for the Degree of Master of Engineering**

**Department of Computer and Telecommunication Engineering**

**Graduate School, Dhurakij Pundit University**

**2012**

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สามารถ สำเร็จ ลุล่วง ได้อย่างสมบูรณ์ โดยได้รับความอนุเคราะห์ อย่างดีเยี่ยมจาก อาจารย์ ดร.ชัยพร เหมะภาคะพันธ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์และอาจารย์ ดร.ชนัญ จารุวิทย์โกวิท อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ซึ่งได้ให้ความกรุณาแนะนำ ให้คำปรึกษาตรวจสอบ แก้ไขข้อบกพร่องต่าง ๆ เพื่อให้วิทยานิพนธ์ฉบับนี้ถูกต้อง มีความสมบูรณ์ยิ่งขึ้น ข้าพเจ้ารู้สึกซาบซึ้งใน ความอนุเคราะห์ที่ได้รับ จึงขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณ รศ.ปทุมวิวี งามจริกุลและ ดร.ประศาสน์ จันทราทิพย์ กรรมการสอบ วิทยานิพนธ์ ที่สละเวลามาเป็นกรรมการสอบวิทยานิพนธ์และให้คำแนะนำที่เป็นประโยชน์ต่องาน วิทยานิพนธ์ฉบับนี้ ขอขอบคุณเจ้าหน้าที่ทุกท่านที่ช่วยดำเนินการเรื่องต่างๆ ให้เป็นอย่างดี

ขอขอบพระคุณเป็นอย่างสูงต่อ นายสุเทพ รุ่งเรือง ผู้จัดการส่วน 8 บริษัท กสท โทรคมนาคม จำกัด (มหาชน) ที่ให้คำแนะนำแก้ปัญหาในวิทยานิพนธ์ฉบับนี้จนสามารถดำเนินการให้เสร็จ เรียบร้อย

ขอขอบพระคุณเจ้าของเอกสาร บทความ ตำรา หนังสือทุกท่านที่ข้าพเจ้าใช้ในการสืบค้น ข้อมูลที่ไม่ได้กล่าวนามไว้ ณ ที่นี้

ท้ายสุดนี้ คุณค่าและประโยชน์จากการค้นคว้าอันพึงมีของวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอ มอบทดแทนบุญคุณต่อบิดา มารดาและครูอาจารย์ทุกท่านที่ได้อบรมสั่งสอนศิษย์มาตลอด ด้วย จิต- วิญญาณของความเป็นครู ตลอดจนผู้มีพระคุณทุกท่านมา ณ โอกาสนี้ หากมีข้อบกพร่องประการใด ข้าพเจ้าขอน้อมรับไว้แต่เพียงผู้เดียว

ดุสิต วัฒนเสย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	ฉ
บทคัดย่อภาษาอังกฤษ .....	ง
กิตติกรรมประกาศ.....	จ
สารบัญตาราง .....	ช
สารบัญรูป .....	ฉ
บทที่	
1. บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 วัตถุประสงค์ของงานวิจัย .....	2
1.3 ขอบเขตของงานวิจัย .....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ .....	3
2. ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง .....	4
2.1 เทคโนโลยี Voice over Internet Protocol (VoIP) .....	4
2.2 ผู้สาขาโทรศัพท์ .....	11
2.3 Asterisk Gateway Interface (AGI) .....	16
2.4 ภาษาไพธอน (Python language) .....	18
2.5 Interactive Voice Response (IVR) .....	21
2.6 ไมโครคอนโทรลเลอร์ (Microcontroller) .....	22
2.7 ส่วนควบคุมอุปกรณ์ไฟฟ้า .....	26
2.8 งานวิจัยหรือผลิตภัณฑ์ที่เกี่ยวข้อง .....	27
3. ระเบียบวิธีวิจัย .....	34
3.1 แนวทางการวิจัยและพัฒนา .....	34
3.2 เครื่องมือที่ใช้ในงานวิจัย .....	35
3.3 แผนการดำเนินงาน .....	36
3.4 ขั้นตอนและวิธีดำเนินงาน .....	37
3.5 การออกแบบทางด้านฮาร์ดแวร์.....	55
3.6 การออกแบบ IVR.....	57

## สารบัญ (ต่อ)

	หน้า
4. การทดสอบระบบ .....	60
4.1 ทดสอบระบบที่สามารถควบคุมชุดควบคุมหรือสถานีได้หลายจุด .....	62
4.2 ทดสอบระบบสามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า .....	63
4.3 ทดสอบระบบสามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า .....	64
4.4 ทดสอบระบบสามารถแจ้งเหตุผิดปกติมายังผู้ใช้ผ่านทางระบบโทรศัพท์ พื้นฐาน PSTN หรือระบบโทรศัพท์เคลื่อนที่.....	65
5. บทสรุปและข้อเสนอแนะ .....	66
5.1 สรุปผลการวิจัย .....	66
5.2 ข้อจำกัดของระบบ .....	67
5.3 ข้อเสนอแนะ.....	67
บรรณานุกรม .....	68
ประวัติผู้เขียน .....	71

## สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงความถี่ของระบบ DTMF.....	22
2.2 แสดงการเปรียบเทียบคุณลักษณะของงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์นี้.....	33
3.1 แสดงขั้นตอนการดำเนินงาน .....	37
4.1 แสดงผลการทดสอบระบบกรณีสามารถควบคุมชุดควบคุมหรือสถานีได้หลายจุด .....	62
4.2 แสดงผลการทดสอบระบบกรณีตรวจสอบสถานการณ์ทำงานของอุปกรณ์ไฟฟ้า.....	63
4.3 แสดงผลการทดสอบระบบกรณีสามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า.....	64
4.4 แสดงผลการทดสอบระบบกรณีแจ้งเหตุผิดปกติไปยังผู้ใช้ผ่านทางระบบโทรศัพท์.....	65

## สารบัญรูป

รูปที่	หน้า
2.1 แสดงกระบวนการ sampling จากข้อมูลเสียง .....	5
2.2 แสดงกระบวนการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล.....	6
2.3 แสดงการติดต่อ VoIP แบบ PC-PC ผ่านโปรแกรม MSN .....	7
2.4 แสดงการติดต่อ Phone PC- to - Phone.....	8
2.5 แสดงการติดต่อ Phone-to- Computer.....	8
2.6 แสดงการติดต่อ VoIP แบบ Phone – to – Phone .....	9
2.7 แสดง IP telephony protocol stack.....	10
2.8 แสดงการส่งสัญญาณเพื่อเชื่อมต่อระหว่างสถานีเชื่อมโยง A ไปยังสถานีเชื่อมโยง B ของโพรโทคอล SIP.....	11
2.9 แสดงระบบตู้สาขา PBX เชื่อมต่อกับชุมสายโทรศัพท์ท้องถิ่น .....	12
2.10 แสดงระบบตู้สาขา IP-PBX ที่รองรับโทรศัพท์ PSTN.....	13
2.11 แสดงการเชื่อมต่อโทรศัพท์พื้นฐาน PSTN ผ่านทาง Asterisk Card .....	14
2.12 แสดงโครงสร้างสถาปัตยกรรม Asterisk.....	15
2.13 แสดงบล็อกไดอะแกรมการติดต่อสื่อสาร Asterisk กับอุปกรณ์ ผ่านทาง AGI Module.....	18
2.14 แสดงไดอะแกรมการคอมไพล์ของภาษาซี.....	21
2.15 แสดงไดอะแกรมของอินเตอร์พรีเตอร์.....	21
2.16 แสดงโครงสร้างทั่วไปของไมโครคอนโทรลเลอร์.....	24
2.17 แสดง Microcontroller Unit (MCU) ที่สามารถเชื่อมต่อโครงข่ายไอพีผ่านทางพอร์ต RJ45 .....	25
2.18 แสดงการเปรียบเทียบระหว่าง Microchip Stack กับ TCP/IP Model.....	26
2.19 แสดงวงจรพื้นฐานของโซลิดสเตทรีเลย์.....	27
2.20(a) แสดงความสามารถของเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านระบบโทรศัพท์.....	28
2.20(b) แสดงเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านระบบโทรศัพท์.....	28

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.21 แสดงการเชื่อมต่อของอุปกรณ์ X10.....	29
2.22 แสดงการใช้งาน X10 ภายในที่พักอาศัย.....	30
2.23 แสดงการทำงานของระบบการจัดการเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต.....	31
2.24 แสดงการทำงานของระบบการจัดการเครื่องใช้ไฟฟ้า ด้วยหน้าเว็บเพจ.....	32
3.1 แสดงโครงข่ายของระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วยIVRผ่านระบบ VoIP.....	38
3.2 แสดงขั้นตอนการทำงานของระบบ.....	40
3.3 แสดงลำดับขั้นตอนการทำงาน Sequence Diagram ระบบสามารถควบคุมชุดควบคุมอุปกรณ์ไฟฟ้าหรือเพิ่มสถานีควบคุมระบบได้.....	41
3.4 แสดงการออกแบบซอฟต์แวร์ Dial Plan ของระบบที่สามารถควบคุมชุดควบคุมได้หลายชุด.....	43
3.5 แสดงลำดับขั้นตอนการทำงาน Sequence Diagramกรณีตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า.....	44
3.6 แสดงการออกแบบทางด้านซอฟต์แวร์กรณีตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า.....	45
3.7 แสดงลำดับขั้นตอนการทำงาน Sequence Diagram กรณีสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า.....	49
3.8 แสดงลำดับขั้นตอนการทำงาน Sequence Diagram กรณีแจ้งเหตุผิดปกติไปยังผู้ใช้ผ่านทางระบบโทรศัพท์.....	51
3.9 แสดง script file สำหรับตรวจสอบสถานะจาก xml file.....	52
3.10 แสดง script file auto call สำหรับเรียกใช้ check-status.py.....	52
3.11 แสดง Dial Plan สำหรับโทรออกจากระบบไปยังผู้ใช้.....	53
3.12 แสดง crontab script file.....	54
3.13 แสดงขั้นตอนการเขียนซอฟต์แวร์สำหรับระบบแจ้งเหตุผิดปกติไปยังผู้ใช้ผ่านทางระบบโทรศัพท์.....	54
3.14 แสดงการออกแบบส่วนที่ทำหน้าที่ติดต่อสื่อสารกับระบบโทรศัพท์ภายนอก.....	55



สารบัญรูป (ต่อ)

รูปที่	หน้า
3.15 แสดงการเชื่อมต่อ Asterisk Server บอร์ดไมโครคอนโทรลเลอร์ชุด ทรานซิสเตอร์ขับกระแสไฟฟ้าและบอร์ดควบคุมอุปกรณ์ไฟฟ้า.....	56
3.16 แสดงบล็อกไดอะแกรม Transistor Drive Module.....	56
3.17 แสดงซอฟต์แวร์สังเคราะห์เสียงพูดภาษาไทย “วาจา” (VAJA) ของ NECTEC .....	57
3.18 แสดงขั้นตอนการแปลงข้อความเสียงพูดด้วยซอฟต์แวร์สังเคราะห์เสียงพูด ภาษาไทย “วาจา” (VAJA) .....	58
4.1 แสดงอุปกรณ์และโครงข่ายที่ใช้ในการทดสอบ.....	59

หัวข้อวิทยานิพนธ์	ระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP
ชื่อผู้เขียน	คุณิต วัฒนเสย
อาจารย์ที่ปรึกษา	อาจารย์ ดร.ชัยพร เขมะภาคะพันธ์
อาจารย์ที่ปรึกษาร่วม	อาจารย์ ดร.ธนัญ จารุวิทย์โกวิท
สาขาวิชา	วิศวกรรมคอมพิวเตอร์และโทรคมนาคม
ปีการศึกษา	2554

### บทคัดย่อ

งานวิจัยนี้นำเสนอการใช้ประโยชน์ของ VoIP สำหรับตรวจสอบและควบคุมการทำงานของอุปกรณ์ไฟฟ้าซึ่งผู้ใช้งานสามารถเข้าถึงระบบได้จากระยะไกลทำให้ประหยัดเวลาและค่าใช้จ่ายด้วยการประยุกต์ใช้ Asterisk เป็น IP-PBX และระบบ IVR (Interactive Voice Response) ร่วมกับอุปกรณ์ฮาร์ดแวร์อื่นๆ เช่น ไมโครคอนโทรลเลอร์ โซลิด-สเตท รีเลย์ เป็นต้น เพื่อเชื่อมต่อกับโครงข่ายโทรศัพท์สาธารณะและโครงข่ายอินเทอร์เน็ตโดยระบบที่พัฒนา นี้สามารถตรวจสอบและควบคุมการทำงานของอุปกรณ์ไฟฟ้าได้นอกจากนี้ผู้ใช้งานสามารถตั้งเปิด-ปิดอุปกรณ์ไฟฟ้าได้จากระยะไกลและเชื่อมต่อบริษัทเพื่อใช้งานระยะไกลได้หลายแห่งในเวลาเดียวกันรวมทั้งสามารถแจ้งเหตุผิดปกติที่เกิดขึ้นกับอุปกรณ์ไฟฟ้าผ่านทางโครงข่ายโทรศัพท์ ด้วยระบบ IVR โดยอัตโนมัติ

จากการทดสอบการทำงานพบว่า ผู้ใช้งานสามารถโทรศัพท์ไปยังระบบที่พัฒนาเพื่อทำการตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าต่างๆได้ พร้อมทั้งสั่งเปิด-ปิดอุปกรณ์ดังกล่าวได้อย่างถูกต้องตามคำสั่งของผู้ใช้งานผ่านระบบ IVR ผู้ใช้งานสามารถเลือกชุดควบคุมอุปกรณ์ไฟฟ้าได้อย่างถูกต้องตามที่ต้องการนอกจากนั้น การทดสอบในสถานะการณ์เหตุผิดปกติที่ได้จำลองขึ้นนี้ระบบที่พัฒนาก็สามารถแจ้งเหตุผิดปกติดังกล่าวให้ผู้ดูแลระบบทราบได้อย่างถูกต้อง

Thesis Title	IVR based Alarm and Control Systems for Electrical Equipment using VoIP System
Author	Dusit Wattanasoei
Thesis Advisor	Chiyaporn Khemapatapan, Ph.D
Co-Thesis Advisor	Tanun Jaruvitayakovit, Ph.D
Department	Computer and Telecommunication Engineering
Academic Year	2011

### **ABSTRACT**

This thesis proposes an applying VoIP for monitoring and controlling of electrical equipments. Users can remotely access to the proposed system. So, users are not necessary to go to working places resulting in saving time and travel expense. This research applied Asterisk as the IP-PBX and IVR system in conjunction with other hardware such as microcontroller, solid-state relay etc. The developed system connected to the public telephone network and internet to control the operation of electrical equipments. Moreover, users can remotely monitor working status and switch on/off electrical equipments including automatically notification of fault of electrical equipments via IVR system for the simulated fault. The system can control and monitor the electrical equipments in many places in the same time.

Testing results indicated that users can remotely access to the system via telephone. The users can check the working status of the electrical equipments that are connected to the solid-state relay. Users can correctly control and switch on/off of these electrical equipments via an IVR system. Moreover, system can correctly monitor and inform administrator for faults simulated.

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

เทคโนโลยี โครงข่าย โดยเฉพาะอินเทอร์เน็ตเป็นเทคโนโลยีที่มีการพัฒนาอย่างต่อเนื่อง โดยโครงข่ายอินเทอร์เน็ตได้แพร่หลายและมีผู้ใช้งานมาก อินเทอร์เน็ตจึงมีบทบาทในชีวิตประจำวันเป็นอย่างมาก ก็เพราะว่ามีการขยายตัวของระบบโครงข่ายไอพีมากขึ้น ครอบคลุมพื้นที่บริการที่กว้างขวาง ผู้บริโภคสามารถใช้บริการอินเทอร์เน็ตได้ ทุกที่ทุกเวลา ทั้งจากระบบโครงข่ายสื่อสารข้อมูลหรือระบบ LAN (Local Area Network) โครงข่ายไร้สายและโครงข่ายโทรศัพท์เคลื่อนที่ (Mobile Network) เนื่องจากบริการอินเทอร์เน็ตเป็นที่นิยมของผู้บริโภคทั้งในกลุ่มคนทำงาน กลุ่มนักศึกษา องค์กรต่างๆ หรือกลุ่มบ้านที่อยู่อาศัย จึงทำให้มีการนำเสนอหรือพัฒนาบริการใหม่ๆ เกิดขึ้นบนอินเทอร์เน็ต ทั้งในแง่ของการเข้าใช้อินเทอร์เน็ตและบริการบนโครงข่ายอินเทอร์เน็ต อาทิ เช่น VoIP ที่กำลังเป็นที่นิยม เนื่องจากใช้งานง่ายและต้นทุนค่าบริการต่ำ หรือไม่ต้องจ่ายเพิ่ม รวมทั้งมีการผลิตอุปกรณ์ไอทีที่รองรับการสื่อสารผ่านโครงข่ายไอพีมากขึ้น ระบบโทรศัพท์ไอพีหรือ VoIP ซึ่งเป็นบริการอย่างหนึ่งที่ได้รับการพัฒนาจนเป็นที่นิยมของผู้บริโภค ก็เนื่องจากโครงข่ายด้านเสียง (Circuit Switching) เกิดชะงักหรือมีการพัฒนาค่อนข้างน้อย ทำให้ผู้ให้บริการนำบริการด้านเสียง มาพัฒนาให้สามารถใช้งานได้บนโครงข่ายไอพี จากที่กล่าวมาข้างต้น การใช้บริการอินเทอร์เน็ตสามารถใช้บริการได้ทุกที่ทุกเวลาที่มีโครงข่ายอินเทอร์เน็ต ทำให้การดูแลระบบ (Monitoring System) ที่ให้บริการด้านต่างๆ ด้วยวิธีการเข้าถึงระบบจากระยะไกล (Remote Control) สามารถทำได้โดยง่าย

กรณีศึกษาหนึ่งคือกรณีของ Base Transceiver Station (BTS) หรือสถานีฐานของระบบโทรศัพท์เคลื่อนที่ ทำหน้าที่ติดต่อสื่อสารกับเครื่องโทรศัพท์ โดยทั่วไปแล้วสถานีฐานมีลักษณะเป็นตู้คอนเทนเนอร์หรือเป็นอาคารขนาดเล็ก ซึ่งจะต้องมีการรักษาอุณหภูมิ โดยใช้เครื่องปรับอากาศให้กับอุปกรณ์ประเภทอิเล็กทรอนิกส์ให้คงที่ตลอดเวลา ทำให้อุปกรณ์มีอายุการใช้งานที่ยาวนาน BTS นั้นตั้งอยู่ห่างไกลจาก Base Station Control (BSC) และไม่มีเจ้าหน้าที่ประจำอยู่ การบำรุงรักษาหรือตรวจซ่อมต้องใช้เวลามากในการเดินทาง หากไม่ได้รับแก้ไขในเวลาอันรวดเร็วอาจทำให้อุปกรณ์เกิดความเสียหายได้และไม่สามารถให้บริการลูกค้าได้ทำให้สูญเสียโอกาสทางธุรกิจ

ผู้พัฒนาจึงมีแนวความคิดในการ พัฒนาระบบแจ้งเตือน และควบคุม อุปกรณ์ไฟฟ้าด้วย ระบบ Interactive Voice Response (IVR) ผ่านระบบ VoIP เพื่อจัดการกับปัญหาดังกล่าว

ระบบที่พัฒนาขึ้นนี้ยังสามารถใช้ในการรีเซ็ตอุปกรณ์โมเด็ม (MODEM) ได้ในกรณีที่ผู้ใช้บริการอินเทอร์เน็ตนำไปติดตั้งให้กับผู้ใช้บริการตามสถานที่ต่างๆ เนื่องจากปัญหาสภาพอากาศ เช่น ฤดูฝน เกิดฟ้าร้องฟ้าผ่า อันเป็นผลให้โมเด็มทำงานไม่เป็นปกติหรือไม่ทำงาน มีความจำเป็นต้องทำการรีเซ็ตโมเด็มให้เริ่มทำงานใหม่ ซึ่งก็สามารถทำการรีเซ็ตผ่านระบบนี้ได้ หรืออาจนำไปประยุกต์ใช้ในระบบบ้านอัจฉริยะหรือ Smart Home ได้ ซึ่งระบบนี้ก็สามารถควบคุมอุปกรณ์ไฟฟ้าภายในบ้านได้ ระบบนี้มีต้นทุนต่ำเพราะใช้ซอฟต์แวร์ระบบเปิด (Open Source) ลงทุนเฉพาะฮาร์ดแวร์เท่านั้น เมื่อเทียบกับมาตรฐาน X10 แล้วระบบที่พัฒนาขึ้นจะมีต้นทุนที่ต่ำกว่ามาก

## 1.2 วัตถุประสงค์ของการวิจัย

การพัฒนาระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP มีวัตถุประสงค์ ดังต่อไปนี้

1. เพื่อศึกษา ออกแบบและพัฒนาเครื่องต้นแบบ ระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วยระบบ IVR ผ่านระบบ VoIP โดยการประยุกต์ใช้ซอฟต์แวร์ Asterisk (IP-PBX) บอร์ดไมโครคอนโทรลเลอร์ (Microcontroller Board) และบอร์ดควบคุมอุปกรณ์ไฟฟ้า (Solid State Relay)
2. เพื่อจำลองสถานการณ์ โดยการทดสอบระบบที่พัฒนาและพิสูจน์ว่าระบบสามารถใช้แจ้งสิ่งผิดปกติ ตรวจสอบสถานะการทำงานและสั่งเปิด-ปิดอุปกรณ์ไฟฟ้าได้จริง

## 1.3 ขอบเขตของงานวิจัย

ขอบเขตของการศึกษางานวิจัย นี้ โดยในการ ออกแบบ ระบบแจ้งเตือน และควบคุม อุปกรณ์ไฟฟ้าด้วย IVR นั้นจะต้องให้ความสำคัญกับ การประยุกต์ใช้ ซอฟต์แวร์ Asterisk IP-PBX บอร์ดไมโครคอนโทรลเลอร์และบอร์ดควบคุมอุปกรณ์ไฟฟ้า ด้วยเทคโนโลยี VoIP

1. ออกแบบและพัฒนา ระบบแจ้งเตือน และควบคุม อุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP โดยมีคุณสมบัติดังนี้
  - 1.1) ผู้ใช้งานสามารถโทรศัพท์ผ่านระบบ VoIP หรือโทรศัพท์เคลื่อนที่โทรเข้ามาในระบบระบบจะตอบรับด้วย IVR เพื่อสอบถามว่า ผู้ใช้งานต้องการตรวจสอบการทำงานของอุปกรณ์ไฟฟ้าได้
  - 1.2) ผู้ใช้สามารถเลือกสถานีควบคุมได้

1.3) ผู้ใช้สามารถเลือกรหัสหมายเลขอุปกรณ์ไฟฟ้าที่ต้องการ ตรวจสอบ โดยการกดเป็นโทรศัพท์ผ่านระบบ DTMF (Dial Tone Multi Frequency)

1.4) ระบบจะตั้งงานผ่านบอร์ดไมโครคอนโทรลเลอร์ เพื่อที่จะตรวจสอบการทำงานของอุปกรณ์ไฟฟ้าและจะแจ้งให้ผู้ใช้งานทราบถึงสถานะการทำงานของอุปกรณ์ไฟฟ้าผ่านระบบ IVR

1.5) ผู้ใช้งานสามารถสั่งเปิดหรือปิดอุปกรณ์ไฟฟ้างดผ่านระบบ DTMF โดยไมโครคอนโทรลเลอร์จะสั่งการทำงานผ่านบอร์ดควบคุมอุปกรณ์ไฟฟ้า (Solid State Relay)

2. ระบบที่พัฒนาสามารถตรวจสอบและสั่งเปิด-ปิดอุปกรณ์ไฟฟ้าได้ไม่น้อยกว่า 3 ระบบ

3. วิทยานิพนธ์ฉบับนี้จะจำลองเหตุผิดปกติที่เกิดขึ้นกับอุปกรณ์ไฟฟ้าที่ควบคุม เมื่อระบบรับทราบถึงเหตุผิดปกติดังกล่าว ระบบจะโทรศัพท์เพื่อแจ้งให้ผู้ใช้งานทราบถึงเหตุผิดปกติ ด้วยระบบ IVR

4. ซอฟต์แวร์ที่ใช้พัฒนาโปรแกรมสคริปคือ โปรแกรมภาษาไพธอน

#### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เครื่องต้นแบบระบบแจ้งเตือนและความคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP โดยการประยุกต์ใช้ซอฟต์แวร์ Asterisk (IP-PBX) บอร์ดไมโครคอนโทรลเลอร์ และบอร์ดควบคุมอุปกรณ์ไฟฟ้า

2. ผู้ใช้งานสามารถทราบถึงเหตุผิดปกติที่เกิดขึ้นกับอุปกรณ์ไฟฟ้าได้อย่างทันทีทันใด

3. ผู้ใช้สามารถตรวจสอบสถานะทำงานของอุปกรณ์ไฟฟ้าได้

4. ผู้ใช้สามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้าได้

5. เป็นการเพิ่มประสิทธิภาพในการใช้งานระบบเครือข่ายอินเทอร์เน็ตที่มีอยู่ให้เกิดประโยชน์

สูงสุด

6. สามารถนำไปประยุกต์ใช้งานกับหน่วยงาน องค์กรหรือบ้านพักอาศัยได้

## บทที่ 2

### ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ในวิทยานิพนธ์ฉบับนี้ มีทฤษฎีที่เกี่ยวข้องในด้านต่างๆ เพื่อช่วยในการพัฒนาระบบ คือ เทคโนโลยี VoIP, Asterisk IP-PBX, AGI - Asterisk Gateway Interface, IVR, ภาษาไพธอน, บอร์ดไมโครคอนโทรลเลอร์, บอร์ดควบคุมอุปกรณ์ไฟฟ้า รวมถึงงานวิจัยและผลงานที่เกี่ยวข้อง

#### 2.1 เทคโนโลยี Voice over Internet Protocol (VoIP) (ปรเมศวร์ กุมารบุญ, 2553)

นับตั้งแต่ ระบบ โทรศัพท์ ในยุคแรกๆ กำเนิดขึ้นมา ทำให้มนุษย์ชาติได้พูดคุยกัน จากระยะไกล ได้ด้วยการเปลี่ยนเสียงมนุษย์เป็นคลื่นไฟฟ้าขนาดเล็ก ที่เรียกกันว่า สัญญาณแอนะล็อก (Analog) ส่งผ่านตัวกลางที่เป็นสายส่งสัญญาณ และมีวิวัฒนาการ มาแล้วมากมายหลายระบบ เช่นจากระบบ Cross Bar มาเป็น SPC แบบดิจิทัล เป็นต้น โครงข่ายโทรศัพท์พื้นฐานหรือที่เรียกกันในวงการโทรคมนาคมทั่วโลกว่า Fixed Line Telephone ที่ทำงานแบบ Circuit Switching ซึ่งแต่เดิมถือว่าเป็นโครงข่ายโทรคมนาคมหลักชนิดหนึ่งในโลก

แต่ปัจจุบันหลังจากมีการแพร่กระจายเทคโนโลยีอินเทอร์เน็ตอย่างรวดเร็ว จึงเกิดการพัฒนาระบบสื่อสารในแบบ Packet Switching ทำให้เกิดการพัฒนาโปรแกรมประยุกต์ (Software and Applications) อย่างมากมายและทั้งนี้ได้รวมถึง โปรแกรมประเภทหนึ่งที่ถูกพัฒนาจน มีผลต่อการเปลี่ยนแปลงในวงการโทรคมนาคมในปัจจุบันและในอนาคต ด้วย ซึ่งนั่นก็คือการใส่ Feature การสนทนาเสียงผ่านโปรแกรม Chat แล้วก็ส่งไปบนเครือข่ายอินเทอร์เน็ตหรืออาจจะมองเห็นภาพง่ายขึ้นก็คือ ในหมู่วัยรุ่นที่เล่น เอ็มเอสเอ็น (MSN) ที่ใช้ลำโพงและไมโครโฟน สนทนาเสียงผ่านอินเทอร์เน็ต

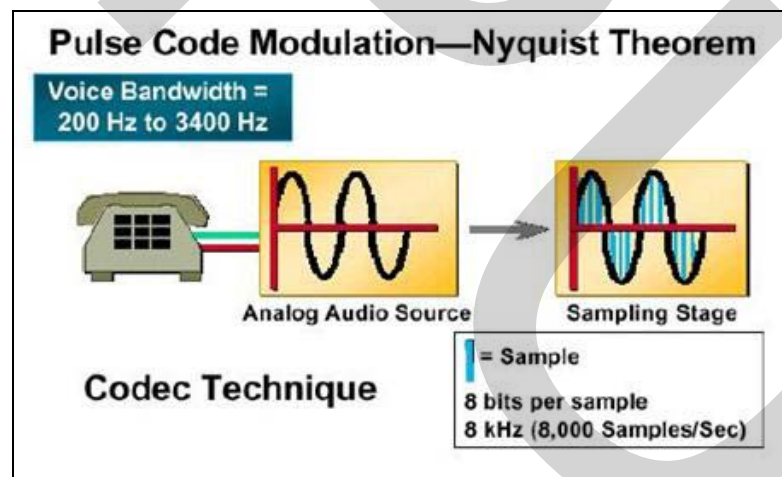
การสนทนาเสียงผ่านอินเทอร์เน็ต นั้นมีด้วยกัน หลากหลายวิธีและการสนทนาเสียงได้รับความนิยมแพร่หลายอย่างรวดเร็ว จนเกิดมีการพัฒนาโปรแกรมการสนทนาเสียงโดย มากมายส่วนมากให้ดาวน์โหลด มาใช้แบบไม่คิด ค่าใช้จ่าย เพื่อการสนทนาผ่านอินเทอร์เน็ต เพียงแต่คอมพิวเตอร์ที่สนทนานั้นต้องมีการติดตั้งโปรแกรมทั้งสองฝ่าย จากความนิยมในการสนทนาเสียงผ่านอินเทอร์เน็ตเป็นอย่างมาก ถึงแม้จะไม่มีค่าใช้จ่ายแต่ก็ได้รับคุณภาพเสียงจากการสนทนาที่ไม่

น่าพอใจนัก จึงเกิดความพยายามมากมายในการพัฒนาระบบการสนทนาผ่านอินเทอร์เน็ต ให้มีคุณภาพเสียงที่ดีเพียงพอที่จะยอมรับได้ จึงเกิดการพัฒนาเทคโนโลยี VoIP ขึ้นมา

VoIP ย่อมาจาก Voice over Internet Protocol หรือเป็นชื่อเรียกเทคโนโลยีนี้ เช่น IP Telephony, Internet telephony, หรือ Digital Phone ทั้งหมดล้วนเป็นการติดต่อสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ตหรือโครงข่ายอื่นๆ ที่ใช้อินเทอร์เน็ต โพรโทคอลหรือ IP (Internet Protocol) โดยสัญญาณเสียงจะถูกแปลงให้เป็นสัญญาณดิจิทัล จากนั้นก็ถูกตัดแบ่งเป็นกลุ่มข้อมูล (Packet) วิ่งไปบนโครงข่ายแบบ Packet Switching ที่ใช้สำหรับการสื่อสารข้อมูลทั่วไปแทนการใช้วงจรเฉพาะตามวิธีการสื่อสารในระบบโทรศัพท์แบบดั้งเดิม (Circuit Switching) โดยมีขั้นตอนการทำงานเป็นดังนี้

### 2.1.1 VoIP และขั้นตอนการทำงาน

2.1.1.1 กระบวนการสุ่มข้อมูล (Sampling) (Waveform Coding Techniques, 2553) เสียงมีลักษณะเป็นสัญญาณแอนะล็อก แต่ข้อมูลที่ส่งออกไปต้องแปลงเป็นสัญญาณดิจิทัล จึงต้องผ่านกระบวนการสุ่มข้อมูลปกติจะสุ่มข้อมูลด้วยความถี่ 8000 sample/sec (ความถี่เสียงอยู่ระหว่าง 200 Hz - 3400 Hz) โดย 1 sample แทนด้วย 8 bits

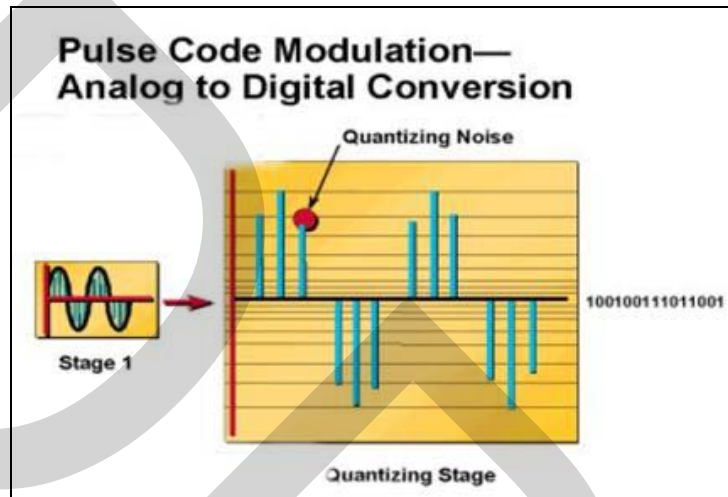


รูปที่ 2.1 แสดงกระบวนการ sampling จากข้อมูลเสียง

ที่มา: [http://www.cisco.com/en/US/tech/tk1077/technologies\\_tech\\_note09186a00801149b3.shtml](http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00801149b3.shtml)



2.1.1.2 กระบวนการ Quantization โดยข้อมูลที่ส่งมาซึ่งเป็นแบบ Continuous state จะแทนด้วยรหัสแบบ discrete state ขนาด 8 bits ดังนั้นจึงต้องปรับข้อมูลให้อยู่ในช่วงที่สามารถแทนค่าในช่วง 8 bit ( $2^8 = 256$  ค่า) ด้วยวิธี Quantization



รูปที่ 2.2 แสดงกระบวนการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

ที่มา: [http://www.cisco.com/en/US/tech/tk1077/technologies\\_tech\\_note09186a00801149b3.shtml](http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00801149b3.shtml)

2.1.1.3 กระบวนการ encapsulation กระบวนการนี้เป็นการนำเอาข้อมูลที่ได้หลังจากผ่านกระบวนการ Quantization มาแบ่งเป็นแพคเกจ เพื่อส่งแต่ละแพคเกจไปยังจุดหมายปลายทาง เมื่อแพคเกจถึงปลายทางก็จะประกอบกันเพื่อให้เป็นข้อมูลเสียงถึงผู้รับต่อไป

2.1.2 รูปแบบการติดต่อของ VoIP (คณิน นิติวงศ์, นาคกร ชูระเจน, 2554)

2.1.2.1 รูปแบบนี้เป็นการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลกับเครื่องคอมพิวเตอร์ส่วนบุคคลหรือ PC หรือ Personal Computer (PC-to-PC) โดยวิธีการนี้จำเป็นต้องอาศัยเครื่องคอมพิวเตอร์ทั้งต้นทางและปลายทาง พร้อมทั้งติดตั้งโปรแกรมเดียวกันหรือติดตั้งโปรแกรมที่สามารถใช้งานร่วมกันได้ ซึ่งรูปแบบนี้เป็นวิธีการสื่อสารที่ไม่ต้องเสียค่าบริการโทรศัพท์แต่อย่างใดเลยและต้องนัดแนะเวลาในการใช้อินเทอร์เน็ตในเวลาเดียวกันเนื่องจากไม่สามารถส่งสัญญาณเรียกไปยังคอมพิวเตอร์ที่ปิดอยู่ได้ เช่น การเล่น MSN ดังแสดงตามรูปที่ 2.3

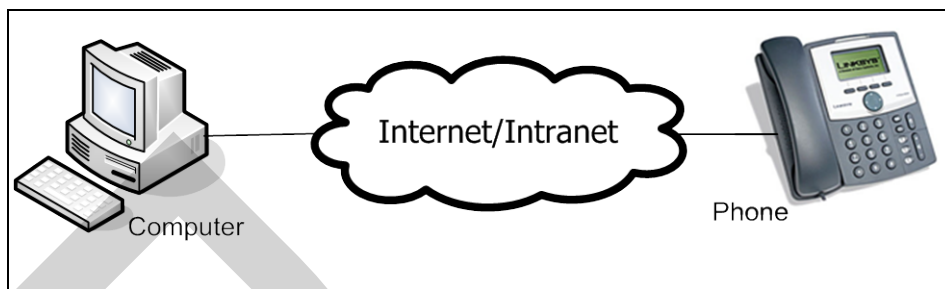


รูปที่ 2.3 แสดงการติดต่อ VoIP แบบ PC-PC ผ่านโปรแกรม MSN

ที่มา: [http://www.student.chula.ac.th/~49702413/for\\_other.htm](http://www.student.chula.ac.th/~49702413/for_other.htm)

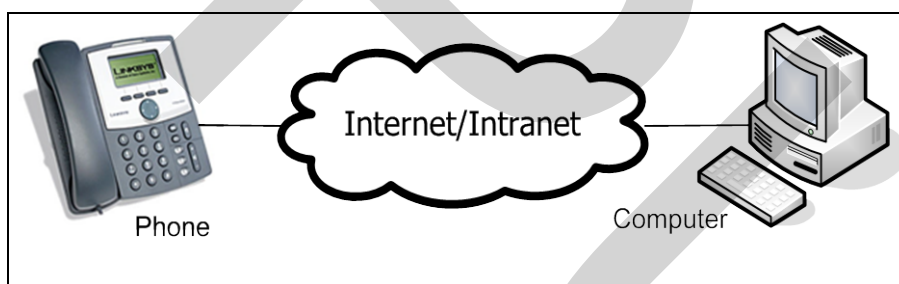
2.1.2.2 รูปแบบการติดต่อ ระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคล กับเครื่องโทรศัพท์ (PC-to-Phone) ในการติดต่อประเภทนี้เครื่องคอมพิวเตอร์ส่วนบุคคลต้องติดตั้งโปรแกรมที่เรียกว่า softphone ซึ่งเป็นโปรแกรมที่จำลองโทรศัพท์บนคอมพิวเตอร์ เพื่อให้สามารถส่งข้อมูลเสียงไปยังปลายทางที่ไม่ใช่คอมพิวเตอร์ได้ ดังแสดงตามรูปที่ 2.4 สำหรับผู้ใช้งานโดยผ่าน PC มีอุปกรณ์ที่จำเป็นดังต่อไปนี้

- 1) ระบบปฏิบัติการบนคอมพิวเตอร์ เช่น windows, Linux เป็นต้น
- 2) การ์ดเสียง (sound card)
- 3) หูฟัง (headset)
- 4) ไมโครโฟน (microphone)
- 5) การเชื่อมต่ออินเทอร์เน็ตความเร็วสูง เช่น DSL หรือ บรอดแบนด์อินเทอร์เน็ต
- 6) Softphone เช่น Skype, X-Lite เป็นต้น



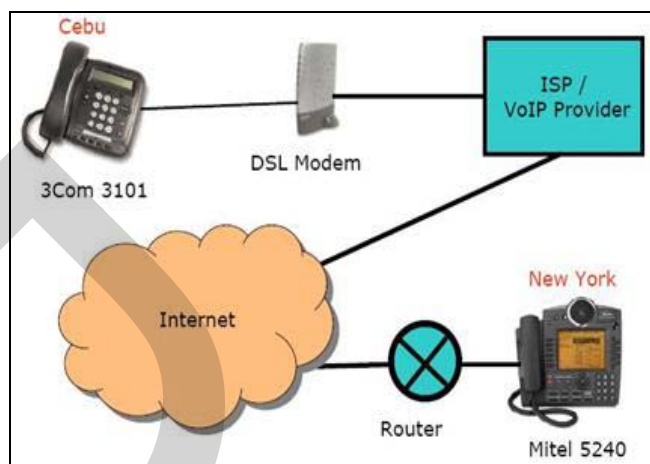
รูปที่ 2.4 แสดงการติดต่อ Phone PC – to – Phone

2.1.2.3 เป็นรูปแบบการติดต่อระหว่างเครื่องโทรศัพท์กับเครื่องคอมพิวเตอร์ส่วนบุคคล (Phone-to-PC) ดังแสดงตามรูปที่ 2.5 วิธีการนี้ใช้หลักการเช่นเดียวกับ PC-to-Phone แต่เส้นทางจะเป็นเครื่องโทรศัพท์ธรรมดา ขณะที่ปลายทางนั้นเป็นเครื่องคอมพิวเตอร์และโปรแกรมโทรศัพท์



รูปที่ 2.5 แสดงการติดต่อ Phone – to – Computer

2.1.2.4 เป็นรูปแบบการติดต่อระหว่าง เครื่องโทรศัพท์ กับเครื่องโทรศัพท์ (Phone- to Phone) ดังแสดงตามรูปที่ 2.6 สำหรับโทรศัพท์ที่ใช้งานในระบบ VoIP แบ่งออกเป็น 2 ประเภทคือ โทรศัพท์ปกติที่ต้องเชื่อมต่อกับ ATA (Analog Telephone Adapter) และโทรศัพท์ที่สามารถใช้ติดต่อผ่านโครงข่ายไอพีได้ (IP Phone)



รูปที่ 2.6 แสดงการติดต่อ VoIP แบบ Phone – to – Phone

ที่มา: [http://www.student.chula.ac.th/~49702413/for\\_other.htm](http://www.student.chula.ac.th/~49702413/for_other.htm)

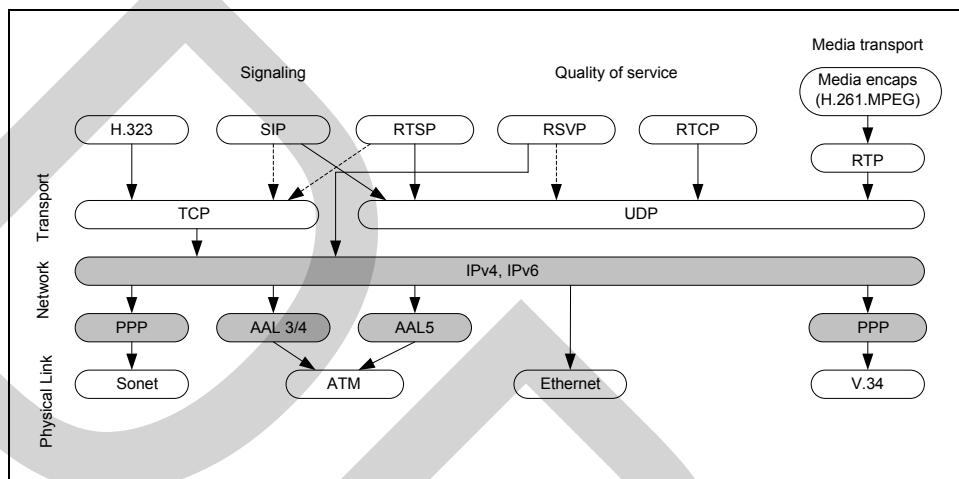
### 2.1.3 องค์ประกอบของ VoIP

2.1.3.1 Software Client หรือ IP Telephony อาจจะเป็นเครื่องคอมพิวเตอร์ ที่ได้รับการติดตั้งโปรแกรมสื่อสารไอพีหรืออุปกรณ์ที่ได้รับการออกแบบขึ้นมา สำหรับการใช้งานโทรศัพท์ผ่านระบบอินเทอร์เน็ตโดยเฉพาะ

2.1.3.2 VoIP Gateway ทำหน้าที่เป็นเครื่องเซิร์ฟเวอร์ที่ถูกใช้งานสำหรับให้บริการโทรศัพท์ผ่าน ระบบอินเทอร์เน็ตเพื่อเป็นตัวกลางในการเชื่อมต่อเข้ากับเครื่องโทรศัพท์ ผู้ชุมสายโทรศัพท์สาธารณะ PSTN (Public Switched Telephone Network) กับระบบ โครงข่าย อินเทอร์เน็ต อย่างโครงข่ายไอพี ซึ่งการจะใช้งานระบบโทรศัพท์ไอพีต้องอาศัยอุปกรณ์นี้เป็นตัวกลางก่อน VoIP Gateway มีลักษณะเป็นอุปกรณ์ในรูปแบบเราเตอร์ (Router) ที่มีคุณสมบัติเช่นเดียวกับเราเตอร์ ที่ใช้งานกันอยู่ทั่วไป แต่มีคุณสมบัติที่ถูกเพิ่มเติมให้รองรับ โพรโทคอลการสื่อสารของ VoIP

2.1.3.3 SIP Server/Gatekeeper เป็นเครื่องเซิร์ฟเวอร์ที่เชื่อมต่อเข้ากับระบบอินเทอร์เน็ต เป็นตัวกลางที่ใช้บริหารจัดการและควบคุมการให้บริการของ VoIP Gateway กับเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรมสำหรับใช้งานสื่อสาร VoIP หรือเครื่องโทรศัพท์ แบบไอพี ซึ่ง Bandwidth ที่ต้องการในการให้บริการ VoIP โดยทั่วไปขึ้นอยู่กับชนิดของการเข้ารหัสและบีบอัดระบบเสียง (Voice Codec) ซึ่ง VoIP Packet มีขนาดเล็กมากแต่ Bandwidth ส่วนใหญ่จะถูกใช้ไปกับ header ของ IP และ UDP ซึ่งมีขนาดใหญ่กว่ามาก

2.1.4 มาตรฐาน โพรโทคอลที่สำคัญในระบบ VoIP (รังสิมา เกียรติยุทธชาติ , สมุทรชัย ไชยวงศ์, 2554) มาตรฐาน โพรโทคอลที่สำคัญสำหรับ VoIP ก็คือ มาตรฐาน โพรโทคอล H.323 และ มาตรฐานโพรโทคอล SIP ซึ่งสามารถแสดงสถาปัตยกรรมของโทรศัพท์ไอพีได้ตามรูปที่ 2.7



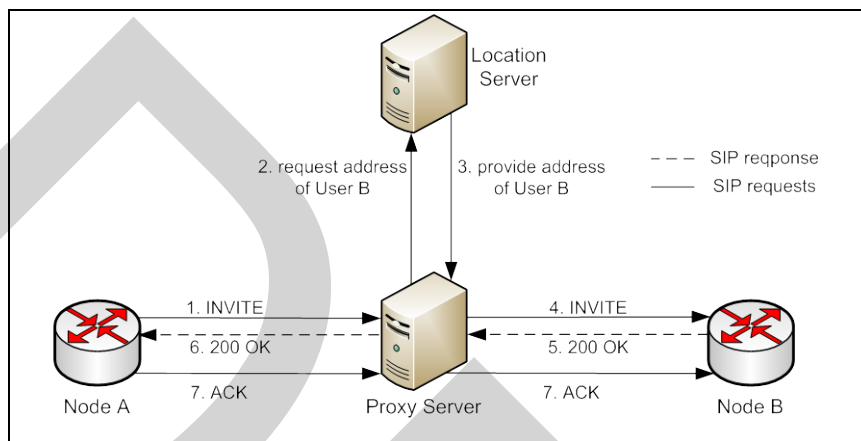
รูปที่ 2.7 แสดง IP telephony protocol stack

2.1.4.1 มาตรฐานโพรโทคอลH.323เป็นโพรโทคอลที่ International Telecommunication Union Telecommunication Standardization Sector (ITU-T) กำหนดสำหรับการส่งผ่านเสียง วิดีโอ และข้อมูลผ่านทางโครงข่าย ซึ่งครอบคลุมถึงระบบอินเทอร์เน็ตด้วย มาตรฐานของ H.323 เกี่ยวข้องกับการให้สัญญาณและการควบคุมเลขที่อยู่ (address call signaling and control) การขนส่งและการควบคุมสื่อประสม การควบคุมแบนด์วิดท์สำหรับการประชุมแบบ point-to-point และ multipoint ส่วนใหญ่โพรโทคอล H.323 จะมุ่งไปที่การประยุกต์ใช้งานของสื่อประสมโดยเฉพาะวิดีโอและการร่วมใช้ข้อมูล ถือว่าเป็นการประยุกต์ใช้งานหลักของโพรโทคอลนี้

2.1.4.2 มาตรฐานโพรโทคอล SIP โดยโพรโทคอล Session Initiation Protocol (SIP) นั้น นิยามโดย Internet Engineering Task Force (IETF) ใช้สำหรับควบคุมการเริ่มต้น ของการเชื่อมต่อ การเปลี่ยนแปลงระบบและการยกเลิกการเชื่อมต่อ ของการสื่อสารแบบสื่อประสม อาทิเช่น เสียง วิดีโอ การส่งข้อความแบบ instant messaging รูปแบบข้อมูลของ SIP จะอยู่ในรูปตัวอักษร (Text-Based) ซึ่งคล้ายกับโพรโทคอล HTTP หรือ Simple Mail Transfer Protocol (SMTP)

อุปกรณ์สำคัญที่ใช้กับโพรโทคอล SIP คือ Proxy ซึ่งหน้าที่หลักของ Proxy ก็คือการรับ และการส่งสัญญาณ Request หรือ Response ของสถานีเชื่อมโยง ต่างๆ รวมทั้งจัดหาเส้นทางของ signaling ที่เดินทางภายในโครงข่าย นอกจากนี้ Proxy อาจใช้สำหรับควบคุมนโยบายของระบบ

เช่น การ authentication ผู้ใช้ ก่อนที่จะอนุญาตให้เข้ามาใช้งานได้ ในการเชื่อมต่อระบบของโพรโทคอล SIP จะมีรูปแบบการส่งสัญญาณของอุปกรณ์ต่างๆ ดังแสดงตามรูปที่ 2.8



รูปที่ 2.8 แสดงการส่งสัญญาณเพื่อเชื่อมต่อระหว่างสถานีเชื่อมต่อ A ไปยังสถานีเชื่อมต่อ B ของโพรโทคอล SIP

ที่มา: [http://www.student.chula.ac.th/~49702413/for\\_other.htm](http://www.student.chula.ac.th/~49702413/for_other.htm)

จากรูปที่ 8 แสดงกระบวนการเชื่อมต่อระหว่างสถานีเชื่อมต่อ A ไปยังสถานีเชื่อมต่อ B มีขั้นตอนการทำงานเป็น ดังนี้

ขั้นตอนที่ 1 อุปกรณ์ Proxy Server รับสัญญาณ INVITE request จากสถานีเชื่อมต่อ A

ขั้นตอนที่ 2 อุปกรณ์ Proxy Server ติดต่อกับ location Server เพื่อขอที่อยู่ (address) ของสถานีเชื่อมต่อ B

ขั้นตอนที่ 3 อุปกรณ์ location Server ค้นหาที่อยู่ของ สถานีเชื่อมต่อ B และส่งข้อมูลที่อยู่กลับมาที่ Proxy Server

ขั้นตอนที่ 4 อุปกรณ์ Proxy Server ส่ง INVITE ไปตามที่อยู่ของ สถานีเชื่อมต่อ B ที่ได้รับ โดย Proxy Server จะบันทึกเส้นทางไปยัง สถานีเชื่อมต่อ B โดยไม่จำเป็นต้องขอที่อยู่จาก location Server ใหม่ในกรณีที่ต้องการส่งสัญญาณไปที่สถานีเชื่อมต่อ B ในครั้งต่อไป

ขั้นตอนที่ 5 สถานีเชื่อมต่อ B ส่งสัญญาณตอบรับ 200 OK กลับมาที่ Proxy Server

ขั้นตอนที่ 6 อุปกรณ์ Proxy Server ส่ง 200 OK ต่อไปยังสถานีเชื่อมต่อ A

ขั้นตอนที่ 7 สถานีเชื่อมต่อ A ยืนยันการรับสัญญาณ 200 OK โดยการส่ง ACK ไปยัง Proxy Server (Proxy Server จำเส้นทางไปยังสถานีเชื่อมต่อ B จากขั้นตอนที่ 4)

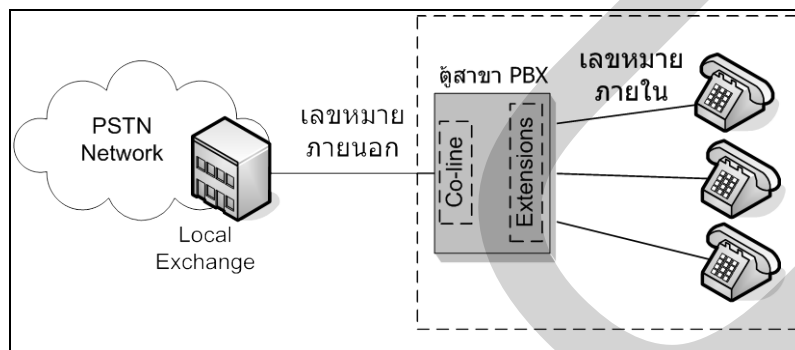
ขั้นตอนที่ 8 อุปกรณ์ Proxy Server ส่ง ACK ต่ไปยังสถานีเชื่อมโยง B

## 2.2 ผู้สาขาโทรศัพท์

ในปัจจุบันผู้สาขาโทรศัพท์มีด้วยกัน 2 ชนิด คือ ผู้สาขาโทรศัพท์ PBX และผู้สาขาโทรศัพท์ IP-PBX

### 2.2.1 ผู้สาขาโทรศัพท์ PBX

ระบบผู้สาขาโทรศัพท์หรือที่เรียกว่า PBX (Private Branch eXchange) จะทำหน้าที่เป็นระบบชุมสายโทรศัพท์ย่อย ที่ทำให้ภายในบริษัทหรือหน่วยงานนั้นๆ มีเบอร์โทรศัพท์เป็นของตนเอง ใช้ติดต่อสื่อสารกันภายในหน่วยงาน โดยเชื่อมต่อโทรศัพท์ผ่านทางจุดเชื่อมต่อ Extensions และสามารถติดต่อไปยังระบบโทรศัพท์ภายนอกหรือชุมสายโทรศัพท์ภายนอกที่เรียกว่า PSTN (Public Switched Telephone Network) ผ่านทางจุดเชื่อมต่อ Co-line ได้ PBX แบบเดิมทำงานด้วยระบบแอนะล็อก จึงมีข้อจำกัดที่เป็นศูนย์กลางใหญ่ใช้พื้นที่มากในการติดตั้ง เป็นระบบปิดไม่มีความยืดหยุ่นด้านฟังก์ชันการโทร (Call Feature) ไม่สามารถรองรับการทำงานร่วมกับอุปกรณ์รุ่นใหม่ ที่ทำงานบนเครือข่ายไอพีได้ อีกทั้งจะขยายหรือเพิ่มเบอร์โทรศัพท์ (Capacity) ก็ทำได้ลำบากหรือต้องใช้ต้นทุนที่สูง ระบบผู้สาขาโทรศัพท์ที่เชื่อมต่อกับชุมสายโทรศัพท์ท้องถิ่น (Local Exchange) แสดงได้ตามรูปที่ 2.9

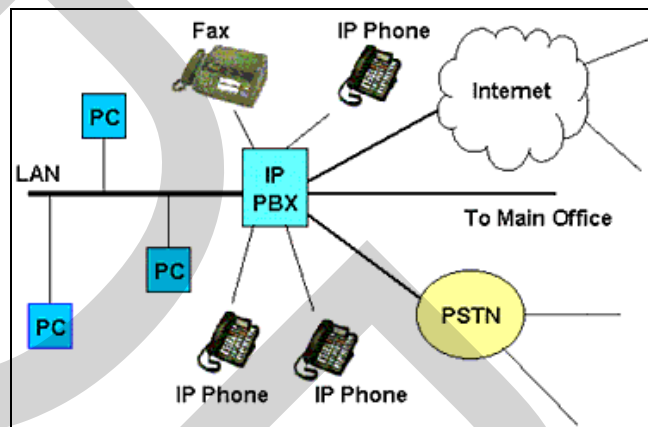


รูปที่ 2.9 แสดงระบบผู้สาขา PBX เชื่อมต่อกับชุมสายโทรศัพท์ท้องถิ่น

### 2.2.2 ผู้สาขาโทรศัพท์ IP-PBX (SPsilicon Press, 2553)

IP-PBX (Internet Protocol-PBX) เป็นระบบผู้สาขาโทรศัพท์ที่ทำงานบนโครงข่ายไอพี ซึ่งได้รับการพัฒนาเพื่อแก้ปัญหาที่เป็นข้อจำกัดของ PBX แบบเดิมและ IP-PBX มีคุณสมบัติที่ดีคือสามารถเพิ่มความยืดหยุ่นในกา รติดต่อสื่อสารให้กับหน่วยงาน ซึ่งเป็นการรวมเทคโนโลยี ยิงของ

ระบบโทรศัพท์พื้นฐานเข้ากับระบบ VoIP ดังแสดงได้ตามรูปที่ 2.10 ทำให้ระบบโทรศัพท์ที่มีความสามารถมากขึ้นและรองรับการขยายตัวของระบบในอนาคตได้ IP-PBX ทำงานร่วมกับโทรศัพท์ไอพีที่รองรับหลายมาตรฐาน เช่น มาตรฐาน SIP และมาตรฐาน H.323 (รังสิมา เกียรติยุทธชาติ, สมทิธิชัย ไชยวงศ์, 2554) ซึ่งงานวิจัยนี้ได้ใช้มาตรฐาน SIP ในการพัฒนาระบบ



รูปที่ 2.10 แสดงระบบตู้สาขา IP-PBX ที่รองรับโทรศัพท์ PSTN

ที่มา: <http://www.silicon-press.com>

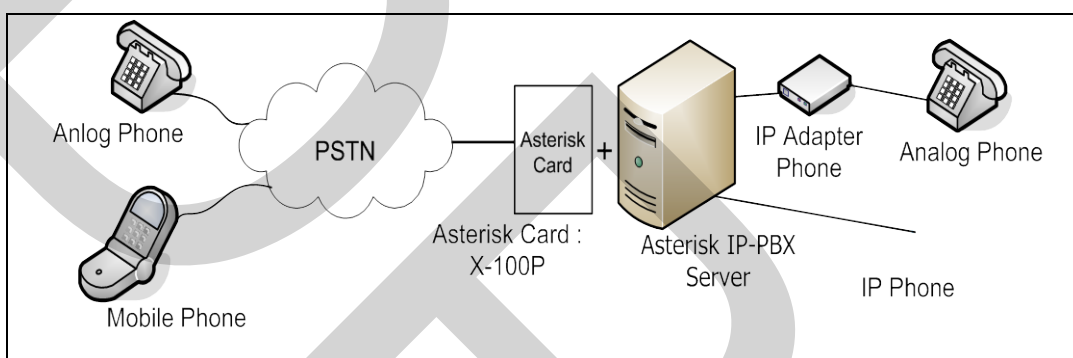
### 2.2.3 Asterisk IP-PBX (ประจักษ์ ปุณยวิจน์พรกุล, 2554), (กิตติพงษ์ สุวรรณราช, 2551)

Asterisk เป็นซอฟต์แวร์แบบเปิด (Open-source Software) ที่ไม่ต้องเสียค่าใช้จ่ายในการใช้งาน สามารถทำงานได้บนหลายๆ ระบบปฏิบัติการ เช่น Linux, Mac OS X, OpenBSD, FreeBSD เป็นต้น หน้าที่หลักก็คือ ทำหน้าที่เป็น IP-PBX (Internet Protocol-Private Branch eXchange) หรือที่เรียกว่าตู้ชุมสายโทรศัพท์ระบบ IP โดยเป็นการนำเอาเทคโนโลยีของระบบโทรศัพท์พื้นฐานที่ใช้กันทั่วไปรวมเข้ากับเทคโนโลยี Voice over IP (VoIP) ทำให้ได้คุณสมบัติของระบบโทรศัพท์ที่มีความสามารถมากขึ้น สามารถสื่อสารกันได้ทั้งระบบโทรศัพท์พื้นฐานที่มีอยู่เดิมกับระบบโทรศัพท์ไอพีได้พร้อมๆ กัน อีกทั้งยังสามารถเพิ่มเติมประสิทธิภาพและ ความสามารถในการทำงานได้โดยง่าย ซึ่งได้มีการจัดเตรียมฟังก์ชันการใช้งานของตู้สาขาโทรศัพท์ PBX ที่มีคุณภาพสูงไว้ในตัว Asterisk สามารถรองรับกับระบบ VoIP ได้หลายโพรโทคอล อาทิเช่น SIP, H.323, IAX เป็นต้น ซึ่งโพรโทคอลดังกล่าว รองรับกับอุปกรณ์โทรศัพท์ที่เป็นมาตรฐานและใช้ฮาร์ดแวร์ที่ราคาไม่แพง

Asterisk จะแตกต่างจาก PBX แบบเดิมตรงที่ Asterisk ทำงานบน โครงข่ายไอพี เป็น Server ที่คอยบริหารจัดการกับการใช้งานโทรศัพท์ ซึ่งหลักๆ ก็คือพนักงานภายในออฟฟิศ IP Phone



หรือคอมพิวเตอร์ที่ติดตั้งโปรแกรม Softphone (เช่น X-Lite) ที่ทำหน้าที่เหมือนโทรศัพท์ทั่วๆ ไป แต่ นำไปใช้งานอยู่บนคอมพิวเตอร์ และทั้งนี้ IP Phone และ Softphone ทำหน้าที่เป็น Client ซึ่งเมื่อ ลงทะเบียน (Register) กับ Asterisk แล้วจะทำให้โทร เรียกไปหาหมายเลขอื่นๆ ที่ลงทะเบียนกับ Asterisk ไปได้ ส่วนกรณีที่โทรออกไปยังหมายเลขอื่นภายนอกบริษัท นั้น Server ที่ติดตั้ง Asterisk จะต้องมีการ์ด (เช่น การ์ด FXO : X-100P) ที่ใช้ต่อกับระบบโทรศัพท์พื้นฐาน PSTN (Public Switch Telephone Network) ซึ่งแสดงได้ตามรูปที่ 2.11



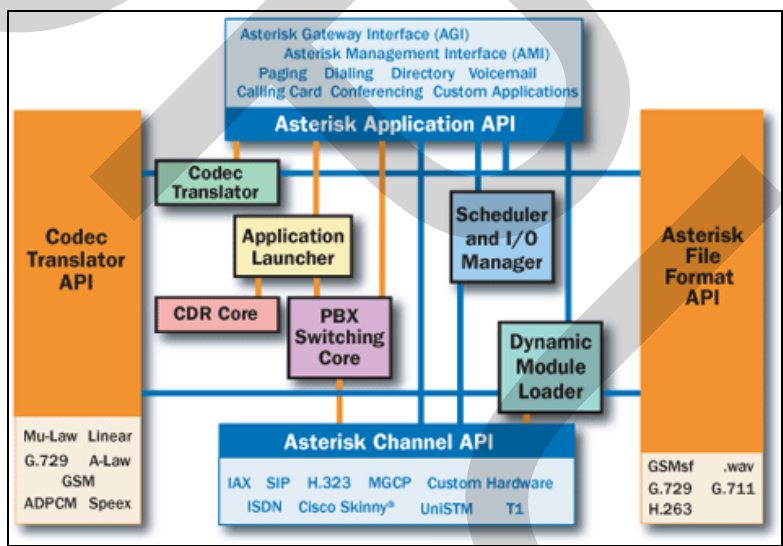
รูปที่ 2.11 แสดงการเชื่อมต่อโทรศัพท์พื้นฐาน PSTN ผ่านทาง Asterisk Card

2.2.3.1 หน้าที่ของ Asterisk ในระบบ IP Telephony คือ Asterisk ทำหน้าที่เป็น Back-to-back user agent นั่นก็คือ Asterisk จะมีความสามารถเป็น UAS (User Agent Server) ที่อนุญาตให้ SIP Client เข้ามา Register และคอยส่งต่อสัญญาณ SIP ไปให้ SIP Proxy Server และ Asterisk จะทำหน้าที่เป็น UAC (User Agent Client) ที่คอยรับสัญญาณจาก SIP Proxy Server และส่งกลับมาให้ SIP Client อีกด้วย จากความสามารถในการเป็นตัวกลางระหว่าง SIP Proxy Server และ SIP Client ทำให้ Asterisk สามารถรับรู้ถึงสถานะของ SIP ที่เกิดขึ้นระหว่างการสนทนาและสามารถให้บริการ ความสามารถต่างๆแก่ระบบ IP Telephony ได้ดังตัวอย่างต่อไปนี้

- 1) Call Detail Records หมายถึง ความสามารถในการเก็บประวัติการใช้งาน Asterisk ของผู้ใช้แต่ละคนในรูปแบบของ Database
- 2) Call Forward หมายถึง ความสามารถในการโอนสายเมื่อมีเหตุการณ์ต่างๆเกิดขึ้นบน SIP Client เช่น เหตุการณ์ Busy (สายไม่ว่าง) หรือ No Answer (ไม่มีผู้รับสาย)
- 3) Conference Bridging หมายถึง บริการที่สามารถทำให้ SIP Client มากกว่า 2 เครื่อง ขึ้นไปสามารถติดต่อกันได้

- 4) Extension number หมายถึง ความสามารถในการนำตัวเลขมาใช้แทน SIP URL
- 5) Interactive Voice Response (IVR) หมายถึงเป็นการนำไฟล์เสียงมาเรียกใช้ เช่น เล่นไฟล์เสียงเพื่อบอกให้ผู้ใช้รู้ว่าระบบได้มีการตอบรับแล้วหรือบอกผู้ใช้ว่าจะทำอะไรเป็นลำดับต่อไป
- 6) Music On Hold หมายถึง ความสามารถในการเล่นเสียงเพลงรอสาย
- 7) Voicemail หมายถึง การเก็บเสียงหรือข้อความของ SIP Client ที่ติดต่อมาในกรณีที่ SIP Client ฟังผู้รับไม่สามารถรับสายดังกล่าวได้

2.2.3.2 โครงสร้างสถาปัตยกรรมของ Asterisk (กิตติพงษ์ สุวรรณราช, 2551) Asterisk ถูกออกแบบให้มีความยืดหยุ่นสูง ประกอบด้วย APIs (Application Programming Interface) ที่เป็นโครงสร้างอยู่ภายนอก ซึ่งทำให้มีประโยชน์มากหากมีผู้พัฒนา API นำไปต่อยอดเพื่อทำงานร่วมกับระบบโทรศัพท์ Asterisk ก็สามารทำได้ทันทีและจะมีระบบ Central PBX อยู่เป็นโครงสร้างภายใน ดังแสดงได้ตามรูปที่ 2.12



รูปที่ 2.12 แสดงโครงสร้างสถาปัตยกรรมของ Asterisk

ที่มา: <http://www.digium.com/images/graphics/asteriskarch.gif>

จากรูปที่ 2.12 ส่วนที่เป็นโครงสร้างภายนอกมีอยู่ด้วยกัน 4 APIs คือ

- 1) Asterisk Channel API เป็น API ที่ทำหน้าที่ในการจัดการกับประเภทของการเชื่อมต่อที่เข้ามา เช่น IAX, ASDN, H.322, MGCP Custom Hardware, Cisco Skinny รวมทั้งเชื่อมต่อกับสายส่งสัญญาณชนิด T1 ด้วย

2) Asterisk Application API เป็นAPI ที่ทำหน้าที่ยอมให้งานหลายๆ อย่างสามารถทำหน้าที่ได้หลายๆหน้าที่ เช่น Asterisk Gateway Interface (AGI), Asterisk Management Interface (AMI), Paging, Dialling, Directory, Voicemail, Calling Card, Conference, Custom Application เป็นต้น

3) Asterisk File Format API เป็นAPI ที่ทำหน้าที่ในการอ่านและบันทึกไฟล์ในหลายรูปแบบ เช่น ไฟล์เสียงนามสกุล gsm, นามสกุล Wave, H.263, G.729, G.711 เป็นต้น

4) Asterisk Translator API เป็นAPI ที่ทำหน้าที่โหลดตัวเข้ารหัส – ถอดรหัสของไฟล์เสียงรูปแบบต่างๆ ไม่ว่าจะเป็น GSM, G.729, Mu-Law, A-law, ADPCM เป็นต้น

ส่วนที่เป็น Central PBX ที่เป็น โครงสร้างภายในของ Asterisk มีอยู่ด้วยกัน 6 โมดูล คือ

1) CODEC Translator Module โดยจะมีการใช้ codec โมดูลเพื่อเข้ารหัสและถอดรหัสไฟล์เสียงที่ถูกบีบอัดเพื่อใช้ในการส่งสัญญาณเสียงโดยมีการเลือกมาตรฐานcodecให้มีความเหมาะสมโดยจะคำนึงถึงคุณภาพเสียงและการใช้งานช่องสัญญาณในการส่งผ่านข้อมูล (Bandwidth Usage) ด้วย

2) Application Launcher Module เป็นส่วนที่ประกาศว่า บริการใดจะเริ่มทำงาน เช่น Voice Mail, File Playback, File Background เป็นต้น

3) Scheduler and I/O Management Module ทำหน้าที่ในการจัดสรรและดูแลตารางงานต่างๆ ที่อยู่ในระบบล่าง ซึ่งเพื่อให้งานเหล่านั้นสามารถที่จะทำงานได้อย่างมีประสิทธิภาพภายใต้เงื่อนไขที่กำหนด

4) CDR Core (Call Detail Record) ทำหน้าที่บันทึกข้อมูลการใช้งานของโทรศัพท์ทั้งหมดในระบบ

5) PBX Switching Core เป็นส่วนที่มีความสำคัญ เป็นอย่างมากสำหรับ Asterisk โดย PBX Switching Core จะทำหน้าที่ในการเชื่อมต่อสื่อสารระหว่างผู้ใช้หลายๆ คนและการทำงานอัตโนมัติ รวมถึงการจัดการและดูแลเกี่ยวกับซอฟต์แวร์และฮาร์ดแวร์ของระบบด้วย

6) Dynamic Module Loader ทำหน้าที่ในการจัดการ โมดูลต่างๆที่มีความจำเป็นต่อการทำงานของระบบ

### 2.3 Asterisk Gateway Interface (AGI) (Nir Simionovich, 2009)

AGI เป็นส่วนหนึ่งของโมดูล Asterisk Application API ตามโครงสร้างสถาปัตยกรรมของ Asterisk ที่แสดงตามรูปที่ 2.12 โดย Asterisk ได้จัดเตรียมช่องทางสำหรับการติดต่อของผู้ใช้กับอุปกรณ์หรือซอฟต์แวร์ต่างๆ ซึ่งทำให้ง่ายต่อการพัฒนาออกแบบระบบหรือการประยุกต์ใช้งาน

นั่นก็หมายความว่าผู้พัฒนาสามารถเขียนโปรแกรมด้วยภาษาอะไรก็ได้ซึ่ง AGI ถูกออกแบบให้สามารถรองรับได้หลายภาษา เช่น PERL, PHP, Python, Ruby, Java, C/C++, .Net Language เป็นต้น ไม่เว้นแม้แต่ shell ต่างๆ ก็สามารถติดต่อกับ AGI ได้อย่างง่ายดาย โดย AGI จะติดต่อกับ Dial Plan และถูกเรียกใช้จากไฟล์ extensions.conf AGI มี 4 ชนิดด้วยกันคือ Standard AGI, Dead AGI, Fast AGI, และ Enhanced AGI

2.3.1 Standard AGI เป็นมาตรฐาน AGI ง่ายๆและใช้กันอย่างกว้างขวาง เป็น Script รันอยู่บน PBX ท้องถิ่น (Local PBX) และติดต่อสื่อสารกับ Asterisk ผ่านทาง Socket ที่มีชื่อว่า STDIN และ STDOUT

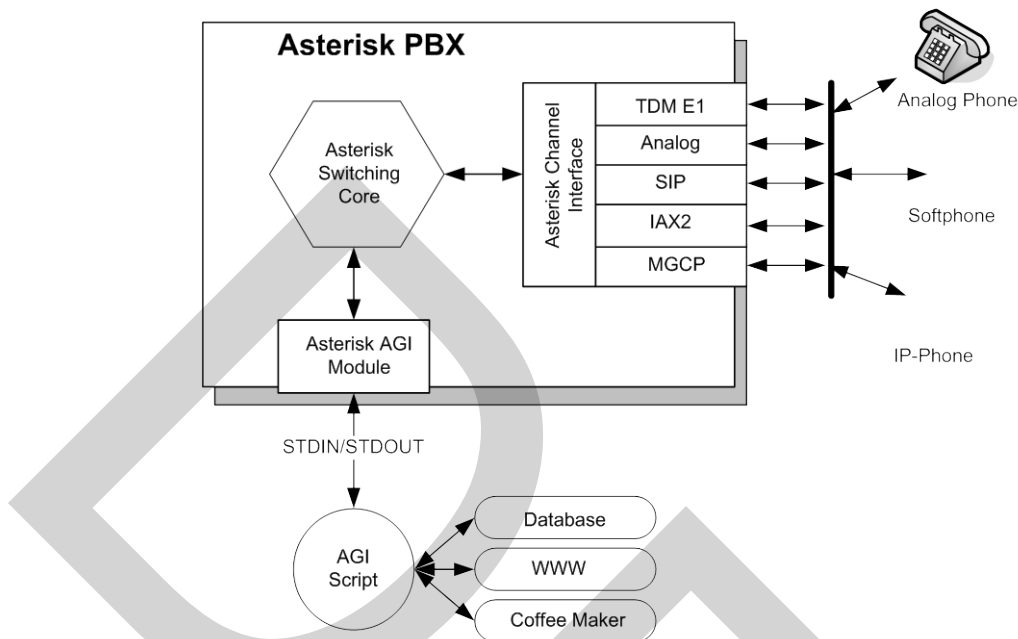
2.3.2 Dead AGI เป็นส่วนที่กระทำต่อเนื่อง โดยทำงานหลังจากมีการเรียก (call) และได้มีการวางสายไปแล้ว

2.3.3 Fast AGI เป็น AGI ที่ทำงานบน TCP sockets Protocol จึงทำให้ผู้ใช้งานสามารถกำหนดการติดต่อกับ AGI ผ่านทาง TCP ได้

2.3.4 EAGI (Enhance AGI) จะติดต่อเกี่ยวกับ channel ต่างๆในกรณีที่มีการเรียกเข้ามาผู้พัฒนาระบบสามารถกำหนดทิศทาง การเข้าถึง audio channel ได้โดยตรงผ่านทาง STDIN และ STDOUT

AGI Script ทำหน้าที่คล้ายกับเป็นตัวกลางหรือคนกลาง (Middle Man) ในการติดต่อสื่อสาร โดยผู้พัฒนาสามารถเขียน AGI Script ได้ด้วยหลายภาษาตามวัตถุประสงค์ในการออกแบบใช้งานตามที่ได้กล่าวเอาไว้ในตอนต้น ซึ่งบทวิจยนี้ผู้พัฒนาใช้ภาษาไพธอนในการพัฒนา AGI Script จากรูปที่ 2.13 แสดงให้เห็นว่าเมื่อผู้ใช้ต้องการติดต่อ Web page (WWW) หรือ Data Base ผ่านทาง E1 Transmission, SIP หรือ Analog ฯลฯ ก็ตาม จะต้องติดต่อผ่านทาง Asterisk Switching Core, Asterisk AGI Module และ AGI Script ด้วย Asterisk ได้กำหนดช่องทางการติดต่อสื่อสารระหว่าง Asterisk AGI Module และ AGI Script ผ่านทาง socket ที่เป็นมาตรฐาน คือ STDIN และ STDOUT โดยมีหน้าที่การทำงานดังนี้

- 1) STDIN (Standard INPUT) เป็นช่องทางที่ AGI Script อ่านข้อมูลที่ได้รับจาก Asterisk
- 2) STDOUT (Standard OUTPUT) เป็นช่องทางที่ AGI Script เขียนข้อมูลแล้วส่งไปยัง Asterisk



รูปที่ 2.13 แสดงบล็อกโคไดอะแกรมการติดต่อสื่อสารของ Asterisk กับอุปกรณ์ผ่านทาง AGI Module

เริ่มต้นใช้งาน AGI Script ที่เขียนด้วยภาษาต่าง ๆ นั้น โดยตาม default แล้วจะเก็บเอาไว้ที่ไดเรกทอรี `/var/lib/asterisk/agi-bin` ภายใน Asterisk เมื่อมีการเรียกใช้ก็สามารถเรียกใช้ด้วยการกำหนดจาก Dial Plan ผ่านทาง `extensions.conf` โดยมีรูปแบบคือ

**exten => extension-number,priority,application,arguments**

จากรูปแบบการเขียน Dial Plan ประกอบไปด้วย 4 ส่วนคือ

- 1) extension-number หมายถึง เลขหมายภายในที่ถูกกำหนดขึ้นมาได้ในหลายรูปแบบ
- 2) priority หมายถึงตัวเลขที่ระบุลำดับการทำงาน โดยส่วนมากจะเริ่มที่ 1 และเพิ่มขึ้นทีละ 1

3) application เป็นการกำหนดให้โทรศัพท์ Asterisk ได้ทำงานตามโปรแกรมที่ตั้งให้ทำงานเช่น Dial, Playback, AGI, Hangup ฯลฯ

4) arguments หมายถึง ชื่อของไฟล์สคริปต์ที่เขียนขึ้นมาด้วยภาษาต่างๆ เช่น test.py เป็นสคริปต์ที่เขียนขึ้นด้วยภาษาไพธอน ข้อสังเกตคือ มีนามสกุล .py เป็นต้น

ตัวอย่างการเขียน Dial Plan ที่ระบุเรียกใช้ AGI application คือ

**exten => 1,2,agi,test.py**

- “1” หมายถึง extension number
- “2” หมายถึง Priority
- “agi” หมายถึง เป็นการระบุเรียกใช้ Application “AGI”
- “test.py” หมายถึง Script ที่เขียนขึ้นด้วยภาษาไพธอน

## 2.4 ภาษาไพธอน (Python language) (bbee, 2010)

ไพธอน เป็นชื่อของภาษาที่ใช้ในการเขียนโปรแกรมภาษา อย่างหนึ่งที่มีความสามารถสูงไม่แพ้ภาษาอื่นๆ ซึ่งภาษาไพธอนเป็นซอฟต์แวร์แบบเปิดที่ไม่คิดค่าใช้จ่าย ทำให้ผู้พัฒนาสามารถที่จะนำภาษาไพธอน มาพัฒนาโปรแกรมของตัวเองได้แบบฟรีๆ และความเป็น โปรแกรมประเภทไม่คิดค่าใช้จ่ายนี้ ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ ภาษาไพธอน มีความสามารถสูงขึ้นและใช้งานได้ครอบคลุมกับทุกลักษณะงาน

### 2.4.1 คุณลักษณะเด่นของภาษาไพธอน

1) ภาษาไพธอนนั้นง่ายต่อการเรียนรู้ โดยมีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่งคล้ายกับภาษาซีมาก เพราะถูกสร้างขึ้นมาใช้ภาษาซีทำให้ผู้ที่คุ้นเคยภาษาซีอยู่แล้วใช้งานภาษาไพธอนได้ไม่ยาก นอกจากนี้ตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความ และ Text File ได้เป็นอย่างดี

2) ภาษาไพธอนเป็นซอฟต์แวร์แบบเปิด จึงไม่ต้องเสียค่าใช้จ่ายใดๆ ทั้งสิ้นในการพัฒนา เพราะตัวแปลภาษาไพธอนอยู่ภายใต้ลิขสิทธิ์ Python Software Foundation License (PSFL) ซึ่งเป็นของ Python Software Foundation (PSF) ซึ่งมีลักษณะคล้ายกับลิขสิทธิ์แม่แบบอย่าง General Public License (GPL) ของ Free Software Foundation (FSF)

3) ภาษาไพธอนนั้นใช้ได้กับหลายๆ แพลตฟอร์ม (Platform) ในช่วงแรกถูกออกแบบให้ทำงานกับระบบ Unix แต่ในปัจจุบัน นี้ได้มีการพัฒนาตัวแปลภาษาไพธอน ให้สามารถใช้ได้ดีกับระบบปฏิบัติการอื่นๆ ได้ อาทิเช่น Linux Platform, Windows Platform, OS/2, Amiga, Mac OS X และรวมไปถึงระบบปฏิบัติการ .NET Framework, Java virtual machine เป็นต้น

4) ภาษาไพธอนถูกสร้างขึ้น โดยได้รวบรวมเอาส่วนดีของภาษาต่างๆ เข้ามาไว้ด้วยกัน อาทิเช่น ภาษา ABC, Modula-3, Icon, ANSI C, Perl, Lisp, Smalltalk และ Tcl

5) ภาษาไพธอนสามารถรวมการพัฒนาของระบบเข้ากับ COM, .NET และ CORBA objects

6) สำหรับ Java libraries แล้วสามารถใช้ Python เพื่อทำการพัฒนาซอฟต์แวร์จากภาษาไพธอนสำหรับ Java Virtual Machine

7) สำหรับ .NET Platform แล้วสามารถใช้ Iron Python ซึ่งเป็นการพัฒนาของ Microsoft เพื่อจะทำให้ไพธอนนั้นสามารถทำงานได้บน .Net Framework ซึ่งใช้ชื่อว่า Python for .NET

8) ภาษาไพธอนนั้นสนับสนุน Internet Communications Engine (ICE) และการรวมกันของเทคโนโลยีอื่น ๆ อีกมากมายในอนาคต

9) บางครั้งนักพัฒนา อาจจะพบว่าภาษาไพธอน นั้นไม่สามารถทำงานบางอย่างได้ แต่นักพัฒนาต้องการให้มันทำงานได้ ก็สามารถพัฒนาเพิ่มได้ในรูปแบบของ extension modules ซึ่งอยู่ในรูปแบบของโค้ด C หรือ C++

10) ภาษาไพธอนสามารถพัฒนาเป็นภาษาประเภท Server side Script คือการทำงานของภาษาไพธอนจะทำงานด้านฝั่ง Server แล้ว ส่งผลลัพธ์กลับมาฝั่ง Client ทำให้มีความปลอดภัยสูง และยังใช้ภาษาไพธอนนำมาพัฒนาเว็บเซอร์วิสได้อีกด้วย

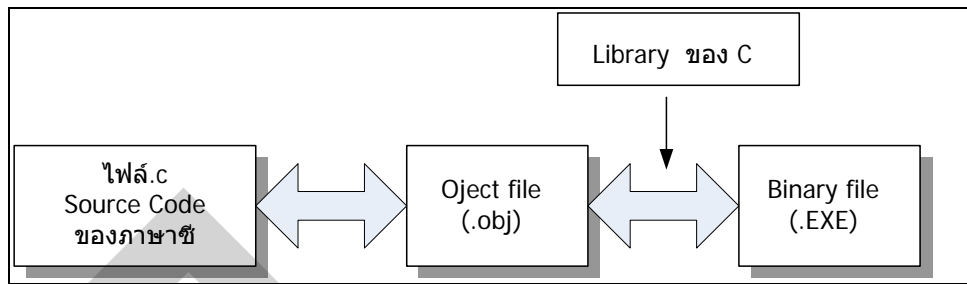
11) ภาษาไพธอนใช้พัฒนาระบบบริหารการสร้างเว็บไซต์สำเร็จรูปที่เรียกว่า Content Management Systems (CMS) ซึ่ง CMS ที่มีชื่อเสียงมาก และเบื้องหลังทำงานด้วยไพธอน คือ Plone <http://www.plone.org/>

จะเห็นได้ว่าคุณลักษณะเด่นของภาษา Python นั้นมีอยู่มากและใช้งานได้อย่างรวดเร็ว ซึ่งหาได้ยากจากภาษาเขียนโปรแกรมภาษาอื่นๆ โดยรองรับทั้ง Object-oriented programming, Imperative, Functional programming และ Logic programming ซึ่งไพธอนสามารถนำไปพัฒนาซอฟต์แวร์ประยุกต์ได้มากมาย

#### 2.4.2 หลักการทำงานของภาษาไพธอน

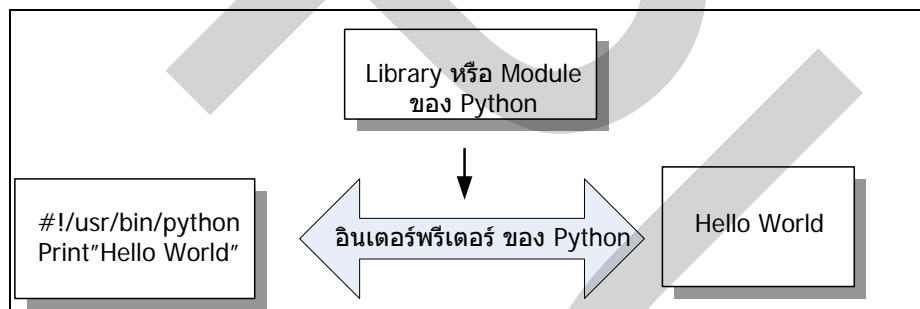
ตัวแปลภาษาคือ เมื่อเราได้เขียนโค้ดขึ้นมาตามโครงสร้างของโปรแกรมภาษาใดก็ตาม และการที่จะให้โค้ดคำสั่งเหล่านั้นทำงานได้ ก็จะต้องมีตัวแปลภาษามาจัดการแปลโค้ดคำสั่งเพื่อให้ทำงานตามที่เราต้องการ ตัวแปลภาษานั้นแบ่งออกเป็น 2 ประเภท คือ

1) คอมไพเลอร์ (Compiler) เป็นตัวแปลภาษาสำหรับภาษา C, C++, Pascal เป็นต้น การทำงานก็คือจะทำหน้าที่ตรวจสอบความผิดพลาดของโค้ดคำสั่งตั้ง แต่ต้นจนจบก่อน หรือที่เรียกว่าการคอมไพล์นั่นเอง ถ้าไม่มีข้อผิดพลาดใดๆ จากนั้นก็จะแปลโค้ดคำสั่งให้เป็นไฟล์นามสกุล .obj (object file) และก็จะทำการแปลไฟล์ .obj ให้เป็นไบนารีไฟล์ .EXE เพื่อทำงานต่อไป ดังตัวอย่างการทำงานของคอมไพเลอร์ภาษา C ดังรูปที่ 2.14



รูปที่ 2.14 แสดงไคอะแกรมการคอมไพล์ของภาษาซี

2) อินเตอร์พรีเตอร์ (Interpreter) เป็นการอ่านโค้ดทีละบรรทัดต่อบรรทัด คือ อ่านโค้ดคำสั่งมาหนึ่งบรรทัดแล้วก็ทำงานแสดงผลออกมาเลยตาม บล็อกไคอะแกรมตามรูปที่ 2.15 ซึ่งแสดงในกรณีที่มีการเรียกใช้ฟังก์ชันจากไลบรารีหรือโมดูลของภาษาไพธอน อินเตอร์พรีเตอร์ของภาษาไพธอนก็จะไปทำการเรียกใช้ฟังก์ชันจากไลบรารีเหล่านั้นให้ทำงานแล้วจึงแสดงผลออกมา เช่น Hello World



รูปที่ 2.15 แสดงไคอะแกรมของอินเตอร์พรีเตอร์

## 2.5 Interactive Voice Response (IVR)

ระบบ IVR หรือระบบตอบรับอัตโนมัติทางโทรศัพท์ เป็นระบบที่เชื่อมต่อระหว่างมนุษย์กับคอมพิวเตอร์ โดยอาศัยระบบโทรศัพท์เป็นสื่อกลางในการสื่อสารกัน โดยจะได้ยินเสียงพูด ส่วสดีค๊ะที่นี่..... เป็นคำทักทายของระบบตอบรับอัตโนมัติทางโทรศัพท์สำหรับบุคคลที่โทรศัพท์มายัง บริษัท ห้างร้าน หน่วยงาน หรือองค์กรต่างๆ ซึ่งเป็นคำทักทายด่านแรกที่ใช้ในการต้อนรับ แล้วตามด้วยคำแนะนำในส่วนงานต่างๆว่ามีส่วนไหนบ้างที่ท่านต้องการจะติดต่อและเป็น การแนะนำในส่วนงานต่างๆ ของบริษัท ห้างร้าน หน่วยงาน องค์กร แบบ สั้นๆ เพื่อลดระยะเวลาที่จะให้บุคคลที่โทรศัพท์มาติดต่อ ส่วนในขั้นตอนของการกดหมายเลขภายในนั้น โดยส่วนใหญ่แล้ว



จะให้กำหนดหมายเลขในการติดต่อไปยังส่วนของหน่วยงานภายใน ให้กดเพียงแค่ 1 - 2 ตัว เพื่อความสะดวกของบุคคลที่เข้ามาใช้บริการของระบบตอบรับอัตโนมัติ หลังจากนั้นให้ส่วนของหน่วยงานภายในนั้นๆ ติดต่อไปยังปลายทางอีกครั้งหนึ่ง แต่ถ้าบุคคลที่เข้ามาใช้บริการระบบตอบรับอัตโนมัติทราบหมายเลขภายในปลายทางที่ต้องการติดต่อก็สามารถกดหมายเลขได้โดยตรง เพื่อที่จะสามารถเข้าถึงส่วนงานหรือบุคคลที่ท่านต้องการได้เลย ตัวอย่างขั้นตอนระบบ ตอบรับอัตโนมัติ เมื่อมีผู้โทรศัพท์ติดต่อเข้ามายังหน่วยงานนั้นๆเป็น ดังนี้คือ สวัสดีค่ะที่นี่....., กรุณากดหมายเลขภายในที่ท่านต้องการติดต่อ, กด “111” เพื่อติดต่อ....., กด “222” เพื่อติดต่อ....., กด “222” เพื่อติดต่อ....., ถ้าไม่ทราบให้กด “0” เพื่อติดต่อโอปอเรเตอร์ เป็นต้น

ในระบบ IVR โดยทั่วไปที่โต้ตอบกับผู้ใช้งานได้ ก็จะต้องให้ผู้ใช้งานกดปุ่มบนแป้นโทรศัพท์ (Dial Pad) เพื่อเลือกตัวเลขที่ได้ยิน เพื่อเข้าสู่เมนูที่ต้องการหรือไม่ก็ต้องป้อนข้อมูลของตัวเองเข้าไป ซึ่งการกดปุ่มบนแป้นโทรศัพท์นั้น เป็นการเรียก ใช้ระบบ DTMF (Dial Tone Multi Frequency) ที่แต่ละปุ่มที่ถูกกดนั้นจะเป็นการนำความถี่ 2 ความถี่มอดูเลต (Modulate) เข้าด้วยกันแล้วส่งออกไป ตัวอย่างเช่น ปุ่มหมายเลข “1” จะเป็นการมอดูเลตความถี่ 697 Hz และ 1209 Hz ดังแสดงตามตารางที่ 1 เป็นต้น โดยตามมาตรฐานแล้วจะ กำหนด DTMF เป็นแบบ 4x4 ความถี่เสียงด้วยกัน

ตารางที่ 2.1 แสดงความถี่ของระบบ DTMF

	1209 Hz	1336 Hz	1477 Hz	1633 Hz <sup>a</sup>
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

ที่มา: Asterisk the Future of Telephony, 2<sup>nd</sup> Edition

## 2.6 ไมโครคอนโทรลเลอร์ (Microcontroller) (วิกิพีเดีย สารานุกรมเสรี, 2554)

ไมโครคอนโทรลเลอร์ (Microcontroller) มาจากคำ 2 คำ คำหนึ่งคือ ไมโคร (Micro) หมายถึงขนาดเล็กและคำว่าคอนโทรลเลอร์ (controller) หมายถึงตัวควบคุมหรืออุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์ จึงหมายถึงอุปกรณ์ควบคุมขนาดเล็ก แต่ในตัวอุปกรณ์ควบคุมขนาดเล็กนี้ได้บรรจุความสามารถ ที่คล้ายคลึง กันกับระบบคอมพิวเตอร์ที่คนโดยส่วนใหญ่คุ้นเคย กัน

กล่าวคือภายในตัวไมโครคอนโทรลเลอร์ นั้นได้รวมเอาซีพียู (CPU) หน่วยความจำและพอร์ต ซึ่งเป็นส่วนประกอบหลักที่สำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกัน

ไมโครคอนโทรลเลอร์เป็นอุปกรณ์อิเล็กทรอนิกส์ ที่สามารถทำงานตามเงื่อนไขต่างๆ ตามที่เราเขียนหรือตั้งโปรแกรมไว้ โดยที่ตัวไมโครคอนโทรลเลอร์เองสามารถเชื่อมต่อกับอุปกรณ์ภายนอกได้ทันทีและทำให้เราสามารถนำไมโครคอนโทรลเลอร์ไปประยุกต์ใช้ในงานการควบคุมต่างๆ มากมาย เช่น การควบคุมมอเตอร์ การควบคุมหลอดไฟหรือการควบคุมการทำงานของหุ่นยนต์ เป็นต้น ในปัจจุบันนี้ไมโครคอนโทรลเลอร์นั้น มีผู้ผลิตหลายบริษัทและมีหลายเบอร์หลายตระกูล อาทิ เช่น 80C51 ของ บริษัท Intel, PIC ของบริษัท Microchip, PSoC ของบริษัท Cypress Microsystems เป็นต้น ทำให้ง่ายต่อการเลือกใช้ให้เหมาะสมกับงาน (Application) นอกจากนั้นแล้วยังสามารถเขียนโปรแกรมได้หลายภาษา เช่น ภาษาแอสเซมบลี, ภาษา Basic, ภาษาซี เป็นต้น ทำให้เรียนรู้ได้เร็ว มีเครื่องมือสนับสนุนในการทำงานมากมาย

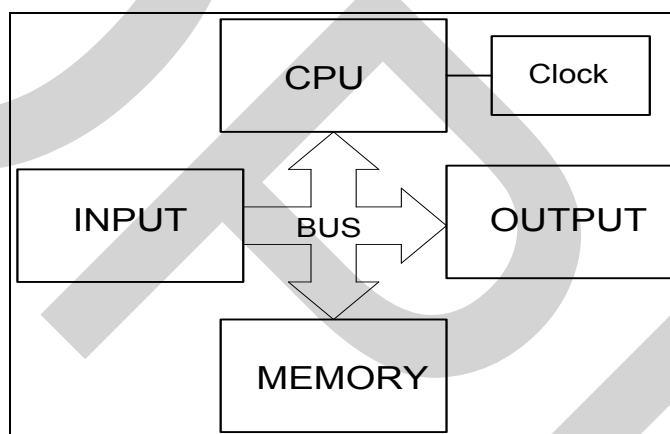
#### 2.6.1 โครงสร้างทั่วไปของไมโครคอนโทรลเลอร์ (คารม กร้าเพียร, 2554)

โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์ ที่แสดงตามรูปที่ 2.16 สามารถแบ่งออกได้เป็น 5 ส่วนใหญ่ๆ ด้วยกัน คือ

- 1) หน่วยประมวลผลกลางหรือ CPU (Central Processing Unit)
- 2) หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะคือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำ (RAM) ในเครื่องคอมพิวเตอร์ทั่วไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็น EEPROM (Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง
- 3) ส่วนติดต่อกับอุปกรณ์ภายนอกหรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณ หรือพอร์ตเอาต์พุต (Output Port) ซึ่งส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะใช้การกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปยังพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

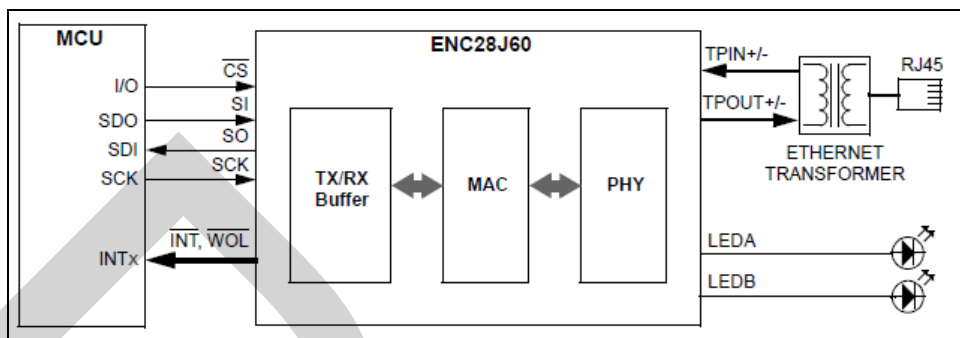
4) ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5) วงจรกำเนิดสัญญาณนาฬิกาหรือ Clock นับว่าเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่งเนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับกรกำหนดจังหวะของสัญญาณนาฬิกา หากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วย



รูปที่ 2.16 แสดงโครงสร้างทั่วไปของไมโครคอนโทรลเลอร์

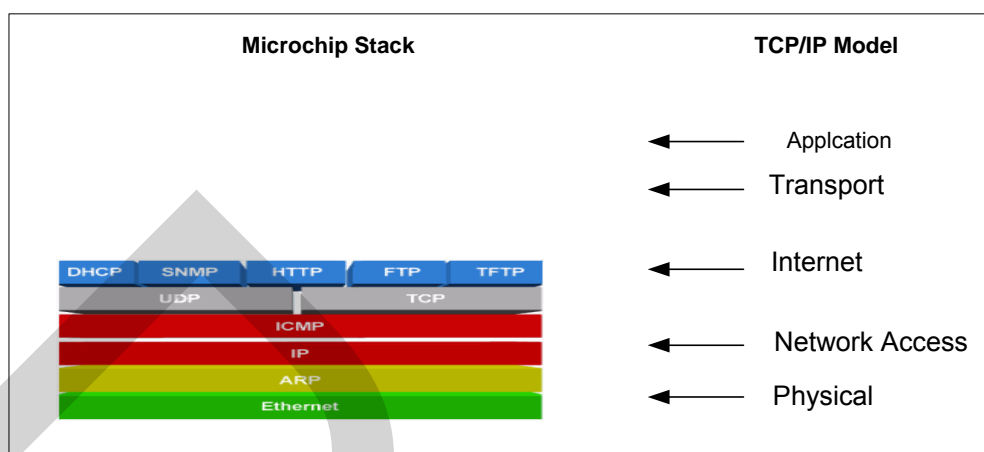
ในยุคแรกๆ นั้นไมโครคอนโทรลเลอร์รองรับการเชื่อมต่อกับอุปกรณ์ภายนอก เช่น คอมพิวเตอร์ ผ่านทางพอร์ต RS232 แต่ในปัจจุบันไมโครคอนโทรลเลอร์ได้รับการพัฒนาให้สามารถใช้งานกับเครือข่ายไอพีได้ โดยเชื่อมต่อผ่านทางพอร์ต RJ45 (Ethernet Port) ซึ่งแสดงได้ตามรูปที่ 2.17



รูปที่ 2.17 แสดง Microcontroller Unit (MCU) ที่สามารถเชื่อมต่อโครงข่ายไอพีผ่านทางพอร์ต RJ45

จากบล็อกไดอะแกรมที่แสดงตามรูปที่ 2.17 เป็นลักษณะการทำงานของบอร์ดไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อกับระบบโครงข่ายไอพีได้ โดยใช้ไอซี ENC28J60 ทำหน้าที่เป็นตัวกลางในการเชื่อมต่อกับระบบแลนที่ระดับฟิสิกัลเลเยอร์ (Physical Layer) ด้วยความเร็วที่ใช้ในการส่งข้อมูลสูงสุด 10 Mbps ผ่านทาง Ethernet Port (RJ45) จึงทำให้บอร์ดไมโครคอนโทรลเลอร์ถูกนำมาใช้เป็น Web Server ได้ ด้วยข้อมูลเว็บไซต์ที่สามารถเลือกเก็บไว้ได้ทั้งภายในตัวไมโครคอนโทรลเลอร์เองและในหน่วยความจำภายนอกและใช้มาตรฐาน Microchip Stack เวอร์ชัน 4.55 ตามรูปที่ 2.18 นอกจากจะเก็บข้อมูลและแสดงผลทางเว็บไซต์ได้แล้วยังสามารถควบคุมการทำงานของไมโครคอนโทรลเลอร์ผ่านทาง Web Browser (Explorer, FireFoX) ได้อีกด้วย เช่น ควบคุมขาอินพุท/เอาต์พุท หรือส่งค่าออกทางพอร์ตอนุกรมเพื่อแสดงผลทางจอ LCD สำหรับการนำไปใช้งานนั้นผู้ใช้สามารถนำไปประยุกต์ใช้ได้หลากหลายรูปแบบด้วยกัน อาทิ เช่น ควบคุมระยะไกลผ่านอินเทอร์เน็ต หรือสร้างเป็นระบบมอนิเตอร์ต่างๆ เป็นต้น

จากรูปที่ 2.18 แสดงการทำงานของ Microchip TCP/IP Stack ในส่วนฟิสิกัลเลเยอร์จะเป็นหน้าที่ของไอซี ENC28J60 ที่จะจัดการเกี่ยวกับการรับส่งข้อมูลและการจัดลำดับข้อมูล ส่วนที่เป็นเลเยอร์ที่สูงขึ้นจะทำการควบคุมโดย Microchip Stack ซึ่งรองรับโพรโทคอลได้หลายโพรโทคอล ขึ้นอยู่กับว่าผู้พัฒนาจะใช้โพรโทคอลใด ซึ่งสามารถเลือกใช้ได้ตามความเหมาะสมของงาน ทำให้ผู้พัฒนาสามารถควบคุมขนาดของซอสโค้ดได้



รูปที่ 2.18 แสดงการเปรียบเทียบระหว่าง Microchip Stack กับ TCP/IP Model

ที่มา: <http://www.microchip.com>

การติดต่อกับ Web service (บอร์ดไมโครคอนโทรลเลอร์) สามารถทำได้ด้วยใช้โพรโทคอลหลักๆ ก็คือ HTTP, ICMP, IP, TCP, FTP และ DHCP เป็นต้น ทั้งนี้ขึ้นอยู่กับความจำเป็นของผู้ใช้ เช่น ในบางงานผู้ใช้อาจจะปิด DHCP ถ้าจะใช้ IP address แบบ static

## 2.7 ส่วนควบคุมอุปกรณ์ไฟฟ้า

ส่วนควบคุมอุปกรณ์ไฟฟ้ามีหน้าที่รับสัญญาณที่ส่งมาจากบอร์ดไมโครคอนโทรลเลอร์เพื่อนำไปควบคุมการทำงานของอุปกรณ์ไฟฟ้าอีกทีหนึ่ง ส่วนที่ทำหน้าที่ควบคุมอุปกรณ์ไฟฟ้าที่เป็นที่นิยมมีด้วยกัน 2 ชนิด คือ รีเลย์ และ โซลีนอยด์-รีเลย์

### 2.7.1 รีเลย์ (Relay)

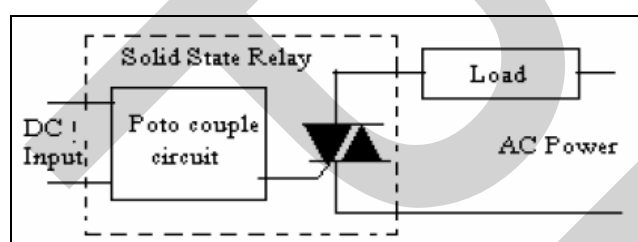
รีเลย์เป็นอุปกรณ์เชิงกลชนิดหนึ่งที่ทำหน้าที่เป็น สวิตช์ไฟฟ้า คือ ตัด-ต่อวงจรไฟฟ้า แตกต่างกันตรงที่รีเลย์ทำงานด้วยขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (Solenoid) ที่ใช้หลักการหน้าสัมผัสของคอนแทค (contact) และการที่จะให้รีเลย์ทำงานก็ จะต้องจ่ายไฟให้ตาม ปริมาณแรงดัน ที่กำหนดเพราะเมื่อจ่ายไฟให้กับตัวรีเลย์ และรีเลย์จะเป็นตัวกำหนดให้หน้าสัมผัสติดกัน (Normal close : NC) เป็น วงจรปิด หรือหน้าสัมผัสห่างกัน (Normal Open: NO) เป็น วงจรเปิด ก็จะจ่ายไฟให้กับตัวรีเลย์หรือไม่ก็ตาม ผู้ใช้สามารถเลือกใช้งานได้ทั้งกรณี NC และ NO ขึ้นอยู่กับ การออกแบบ รีเลย์นั้น ได้ถูกพัฒนามาตามลำดับในแง่ของความเร็วในการทำงาน ขนาดของตัว อุปกรณ์ ความหลากหลายของการเชื่อมต่อสัญญาณ แต่อย่างไรก็ตามเนื่องจากอุปกรณ์รีเลย์เองนั้น

ประกอบไปด้วยชิ้นส่วน ที่ต้องมีการเคลื่อนไหวซึ่งทำให้เป็นอุปสรรคต่อความเร็วในการควบคุม อุปกรณ์อื่นๆ และต้องการพื้นที่ในการติดตั้งมาก

### 2.7.2 โซลิด-สเตท รีเลย์ (Solid State Relay)

โซลิด-สเตท รีเลย์ มีหน้าที่เหมือนกับการทำงานของรีเลย์ แต่ที่มีข้อแตกต่างกันก็คือ ตัว โซลิด-สเตท รีเลย์ นั้นจะเป็นการตัดต่อโดยไม่มีการสปาร์กทางไฟฟ้า เพราะที่ไม่มีหน้าสัมผัสของ คอนแทค ซึ่งข้อดีคือทำให้ไม่มีการติดไฟ รองรับการตัดต่อวงจรที่ใช้ความเร็วได้ดีกว่ารีเลย์และมี อายุการทำงานที่ยาวนานกว่า

การทำงานของ โซลิด-สเตท รีเลย์ โดยเป็นการเชื่อมต่อ (Interface) ระหว่างภาคควบคุม (Control) ซึ่งเป็นส่วนของวงจรอิเล็กทรอนิกส์กับวงจรภาคไฟฟ้ากำลัง (Power) โดยที่ภาคทั้งสอง จะมีระบบกราวด์ (Ground) ที่แยกออกจากกันทำให้สามารถป้องกันการเกิดลัดวงจร (Short circuit) และการรบกวนซึ่งกันและกันได้ ดังแสดงตามรูปที่ 2.19



รูปที่ 2.19 แสดงวงจรพื้นฐานของ โซลิด-สเตท รีเลย์

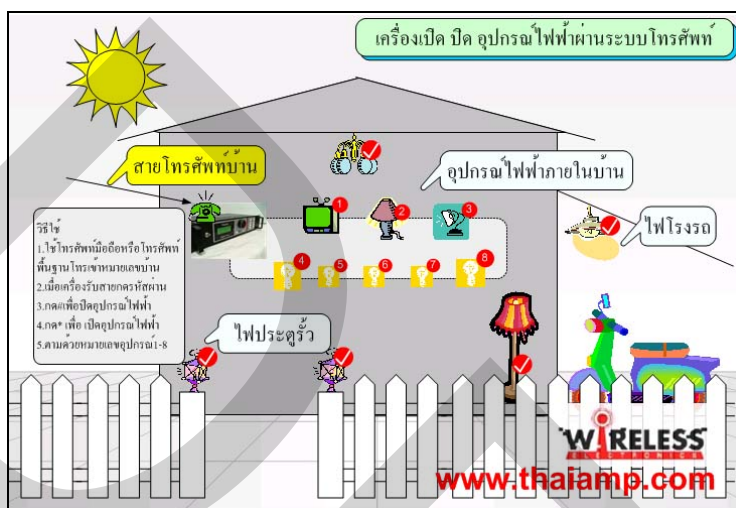
จากที่กล่าวมาในข้างต้น โซลิด-สเตท รีเลย์ มีข้อดีตรงที่ไม่มีส่วนที่เคลื่อนไหวหรือส่วน ที่เป็นหน้าสัมผัสของคอนแทค อย่างเช่นของตัวรีเลย์ ดังนั้นจึงไม่เสี่ยงต่อการเกิดประกายไฟ ซึ่ง อาจจะเป็นการสร้างความถี่รบกวนต่อระบบได้ และตอบสนองต่อสัญญาณทางไฟฟ้าได้เร็ว ดังนั้น งานวิจัยนี้จึงใช้ โซลิด-สเตท รีเลย์ในการพัฒนาระบบ

## 2.8 งานวิจัยและผลิตภัณฑ์ที่เกี่ยวข้อง

### 2.8.1 เครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านระบบโทรศัพท์ (ทีมงาน thaiamp, 2554)

การทำงานของเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านระบบโทรศัพท์ เมื่ออยู่นอกบ้านหรือนอกสำนักงาน แต่ต้องการเปิด-ปิดอุปกรณ์ไฟฟ้าภายในบ้านหรือสำนักงาน สามารถทำได้ โดยการติดตั้งเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านระบบโทรศัพท์เอาไว้กับบ้านหรือสำนักงาน วิธีการติดตั้งก็

คือ นำเครื่องใช้ไฟฟ้าเสียบเข้าที่หลังเครื่องนี้ ก็สามารถควบคุมการทำงานของเครื่องใช้ไฟฟ้าต่างๆ ได้จากโทรศัพท์มือถือหรือโทรศัพท์พื้นฐาน



รูปที่ 2.20 (a) แสดงความสามารถของเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านระบบ โทรศัพท์



รูปที่ 2.20 (b) แสดงเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านระบบ โทรศัพท์

ที่มา: <http://www.thaiamp.com>

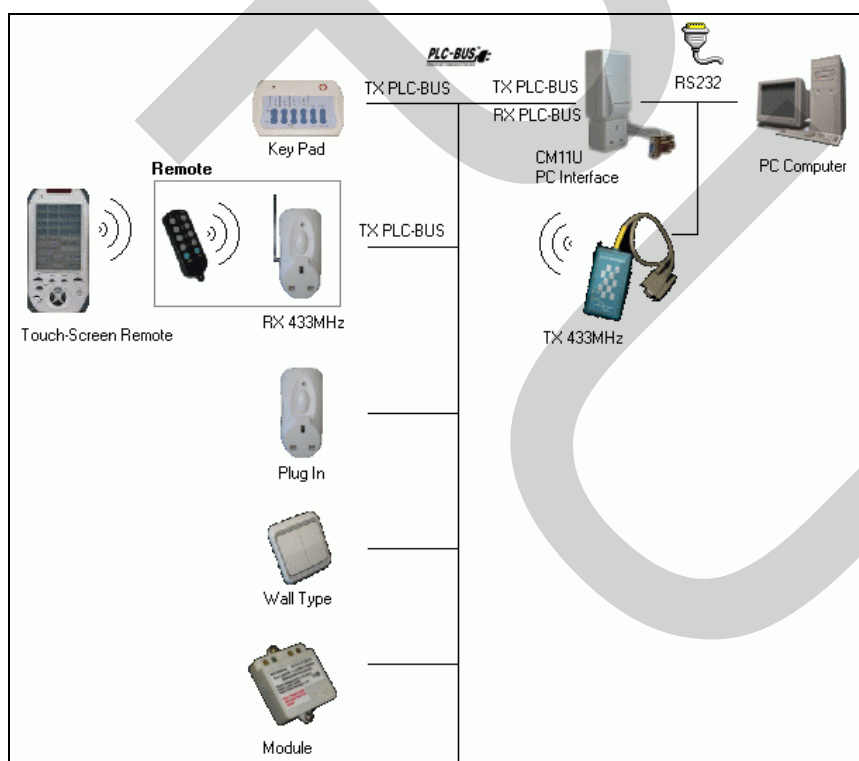
คุณสมบัติของเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านระบบ โทรศัพท์

- 1) สามารถสั่งให้เปิด-ปิดอุปกรณ์ไฟฟ้าผ่านทางระบบโทรศัพท์ พื้นฐาน PSTN และ โทรศัพท์เคลื่อนที่
- 2) สามารถต่อใช้งานผ่านตู้สาขาได้

- 3) มีสัญญาณยืนยันการทำงานหน้าเครื่อง
- 4) สามารถควบคุมการทำงานของอุปกรณ์ไฟฟ้าที่ใช้กำลังไฟฟ้าที่สูงได้
- 5) จัดจำสถานการณ์ทำงานได้แม่ไฟดับ

2.8.2 ระบบควบคุมอุปกรณ์ไฟฟ้าตามมาตรฐาน X10 (วิกิพีเดีย สารานุกรมเสรี, 2554, Thaimicrotron, 2553, เอ-เทค โซลูชั่น, 2554) เทคโนโลยี X10 เป็นมาตรฐานทางอุตสาหกรรมแบบเปิด เป็นมาตรฐานสำหรับการติดต่อสื่อสารในหมู่ของอุปกรณ์ไฟฟ้าตามที่พักอาศัยหรือเรียกกันทั่วไปว่า Home Automation การทำงานของ X10 นั้น ทำงานโดยการส่งข้อมูลไปบน AC Line (Power Line Carrier (P.L.C) Transmission) หรือสายไฟ AC ที่เป็นที่รู้จักกันทั่วไปนั่นเอง

ปัจจุบัน X10 เป็นมาตรฐานที่แพร่หลายไปทั่วโลกซึ่งมี ลักษณะการใช้งานเป็นแบบ Plug In คือเป็นโมดูลมาตรฐานที่เมื่อซื้อไปแล้วสามารถติดตั้งใช้งานได้ทันทีโดยไม่ต้องมีการเดินสายไฟฟ้าใหม่ ซึ่งทุกบริษัทผู้ผลิตอุปกรณ์ X10 จะใช้มาตรฐานเดียวกัน จึงสามารถใช้ร่วมกันได้ เพียงแต่ตั้งแอดเดสของอุปกรณ์ให้ตรงกันเท่านั้น ดังแสดงตามรูปที่ 2.21 และ 2.22



รูปที่ 2.21 แสดงการเชื่อมต่อของอุปกรณ์ X10

ที่มา: <http://www.thaimicrotron.com/X10/X10-MainPage.htm>





รูปที่ 2.22 แสดงการใช้งานอุปกรณ์ X10 ภายในที่พักอาศัย

ที่มา: [http://www.smartihome.com/atech/product/x10/home\\_x10.htm](http://www.smartihome.com/atech/product/x10/home_x10.htm)

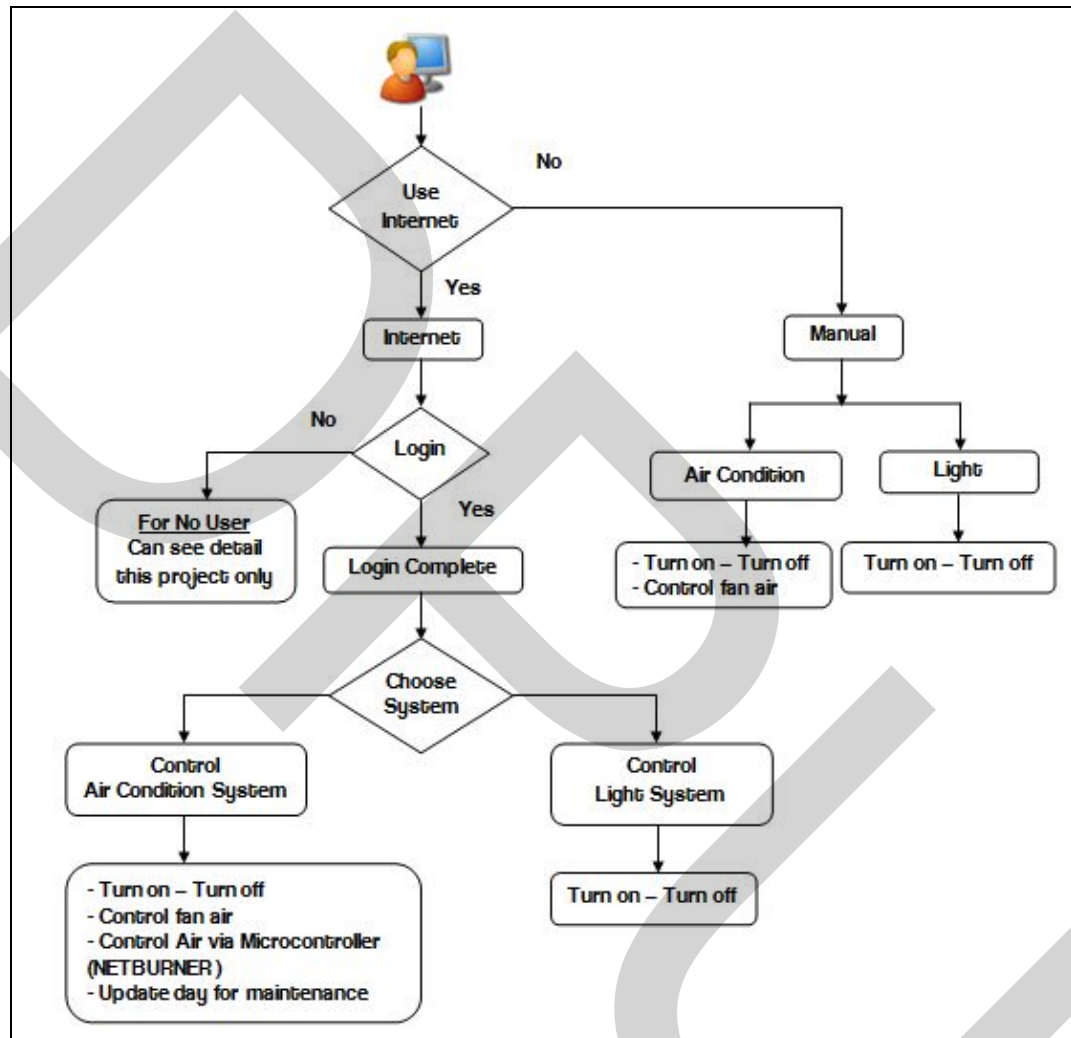
คุณสมบัติของบ้านอัจฉริยะ X10 มีดังต่อไปนี้

- 1) สามารถโทรสั่งงานด้วยระบบโทรศัพท์เคลื่อนที่และโทรศัพท์พื้นฐาน
- 2) รองรับระบบ Telephone/Voice Responder ด้วยอุปกรณ์ FH-T5010E X10
- 3) สั่งงานให้กับ FH-R3010E X10 Air - Conditioner Controller สำหรับเปิด-ปิด

เครื่องปรับอากาศ

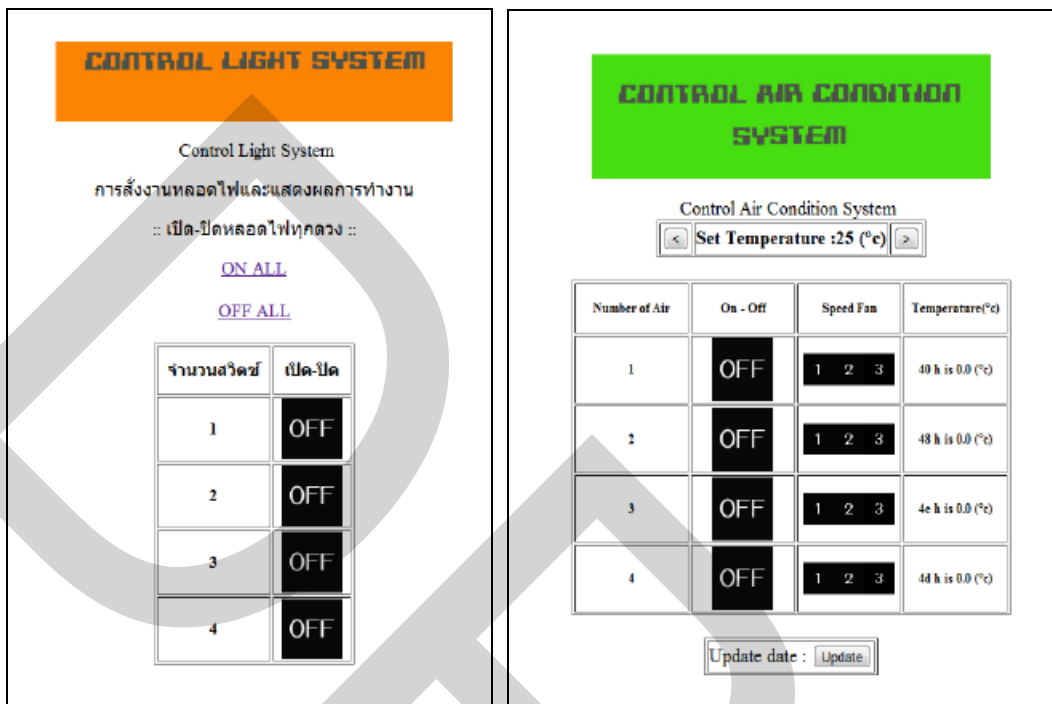
- 4) ควบคุมอุปกรณ์ไฟฟ้าด้วยรีโมทคอนโทรลไร้สายระยะไกล

2.8.3 ระบบการจัดการเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต เป็นการนำโครงข่ายอินเทอร์เน็ต (คณะวิศวกรรมศาสตร์ภาควิชาอิเล็กทรอนิกส์และโทรคมนาคม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี พระจอมเกล้าธนบุรี, 2554) มาจัดการการทำงานของเครื่องใช้ไฟฟ้า โดยทำงานโดยควบคุมผ่านไมโครคอนโทรลเลอร์ และมุ่งเน้นไปที่การทำงาน 3 ส่วนหลักคือ การควบคุมเครื่องใช้ไฟฟ้า การบำรุงรักษาอุปกรณ์เครื่องใช้ไฟฟ้าและรวมถึง การตรวจสอบสถานะของการทำงานของอุปกรณ์เครื่องใช้ไฟฟ้าต่างๆ ได้ เช่น เครื่องปรับอากาศ และระบบแสงสว่าง เป็นต้น โดยที่สามารถควบคุมได้ในทุกที่ที่สามารถเข้าถึงอินเทอร์เน็ต โดยจะมีไมโครคอนโทรลเลอร์ คือ Net burner MOD5270 ซึ่งทำหน้าที่ควบคุมระบบทั้งหมด รวมถึงเป็น Web Server ที่เป็นศูนย์กลางในการเชื่อมต่อระหว่างผู้ใช้กับเครื่องใช้ไฟฟ้า ที่ต้องการควบคุม โดยการทำงานอย่างหลักๆ สามารถแสดงได้ตามรูปที่ 2.23



รูปที่ 2.23 แสดงการทำงานของระบบการจัดการเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต

ที่มา: <http://www.st.kmutt.ac.th/~s0212445/index.htm>



(a)

(b)

รูปที่ 2.24 (a) และ (b) แสดงการทำงานของระบบการจัดการเครื่องใช้ไฟฟ้าด้วยหน้าผ่านเว็บเพจ

ที่มา: <http://www.st.kmutt.ac.th/~s0212445/index.htm>

คุณสมบัติของระบบการจัดการเครื่องใช้ไฟฟ้าผ่านอินเทอร์เน็ต

- 1) ควบคุมการสั่งงานเครื่องใช้ไฟฟ้า เช่น เครื่องปรับอากาศ, หลอดไฟ ผ่านเว็บเพจ
- 2) แสดงสถานะการใช้งานผ่านเว็บเพจ เช่น เปิด/ปิด เครื่องใช้ไฟฟ้า
- 3) ใช้ไมโครคอนโทรลเลอร์ควบคุมการ เปิด-ปิด เครื่องปรับอากาศ
- 4) สามารถเก็บบันทึกประวัติการใช้งานและนับอายุการใช้งานของเครื่องใช้ไฟฟ้า
- 5) IP Address ของตัว Server ที่ใช้ต้องเป็นแบบ Static เท่านั้น ถ้าเป็นแบบ Dynamic จะต้องมีขั้นตอนในการขอใช้บริการ Dynamic DNS เพิ่มเติม
- 6) สามารถสั่งงานผ่านทางระบบโครงข่ายอินเทอร์เน็ต เช่น ADSL เป็นต้น

ตารางที่ 2.2 แสดงการเปรียบเทียบคุณลักษณะของงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์นี้

ลำดับที่	ความสามารถของระบบ	ระบบที่พัฒนาด้วย VoIP	ควบคุมผ่านทางระบบโทรศัพท์	มาตรฐาน X10	ควบคุมผ่านเว็บเพจ
1	โทรสั่งงานเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านทางระบบโทรศัพท์พื้นฐาน PSTN	√	√	√	
2	โทรสั่งงานเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านทางระบบโทรศัพท์ VoIP	√			
3	แสดงสถานะการทำงานของอุปกรณ์ไฟฟ้าด้วยระบบ IVR	√		√	
4	แสดงสถานะการทำงานของอุปกรณ์ไฟฟ้าด้วยหน้าเว็บเพจ				√
5	แสดงสถานะการทำงานของอุปกรณ์ไฟฟ้าด้วยหลอด LED		√		
6	ใช้เครือข่ายอินเทอร์เน็ต	√			√
7	เก็บบันทึกประวัติการใช้งานและนับอายุการใช้งานของอุปกรณ์ไฟฟ้า				√
8	รองรับอุปกรณ์ไฟฟ้าที่ใช้กำลังไฟฟ้าสูงๆได้	√		√	√

## บทที่ 3

### ระเบียบวิธีวิจัย

#### 3.1 แนวทางการวิจัยและพัฒนา

งานวิจัยนี้มีวัตถุประสงค์ เพื่อศึกษา ออกแบบ และพัฒนา ระบบแจ้งเตือน และควบคุม อุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP โดยการประยุกต์ใช้ ซอฟต์แวร์ Asterisk (IP-PBX) บอร์ด ไมโครคอนโทรลเลอร์และบอร์ดควบคุมอุปกรณ์ไฟฟ้า เพื่อให้ระบบสามารถแจ้งเตือนสิ่งผิดปกติ และผู้ใช้งานระบบสามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าได้ อีกทั้งยังสามารถสั่ง เปิดหรือสั่งปิดได้ในคราวเดียวกัน โดยมีแนวทางในการวิจัยและพัฒนาดังนี้

##### 3.1.1 ศึกษาและรวบรวมข้อมูล

- 1) ศึกษาการใช้งานของโปรแกรม Asterisk
- 2) ศึกษาการใช้งานบอร์ดไมโครคอนโทรลเลอร์
- 3) ศึกษาการใช้งานบอร์ดควบคุมอุปกรณ์ไฟฟ้า
- 4) ศึกษาเพิ่มเติมการใช้งานภาษาไพธอนสำหรับเขียน Python AGI Script

##### 3.1.2 การออกแบบระบบงาน

ในส่วนนี้ออกแบบระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP จุดเชื่อมต่อของอุปกรณ์ต่างๆ โดยศึกษาเครื่องมือและอุปกรณ์ที่ใช้พัฒนาอย่างละเอียด

##### 3.1.3 พัฒนาระบบงาน

ทำการพัฒนาระบบให้สามารถทำงานได้ตามวัตถุประสงค์ที่ตั้งไว้และ มีการทดสอบ ย่อยเพื่อหาข้อผิดพลาดต่างๆ ภายในระบบ แล้วทำการแก้ไข

##### 3.1.4 ทดสอบการใช้งาน

การทดสอบนี้เป็นการทดสอบเพื่อดูประสิทธิภาพของทั้งระบบ โดยตรวจสอบคุณภาพ ของการแสดงผลสถานะการทำงานของอุปกรณ์ไฟฟ้า ตรวจสอบความสามารถในการสั่งเปิดหรือสั่ง ปิดอุปกรณ์ไฟฟ้าและตรวจสอบความสามารถของระบบในการแจ้งเหตุผิดปกติ

##### 3.1.5 สรุปผลการพัฒนา

นำข้อมูลที่ได้จากการจำลองสถานการณ์มาสรุปผล เพื่อใช้ในการวิเคราะห์การทำงาน และประเมินประสิทธิภาพของระบบ

### 3.2 เครื่องมือที่ใช้ในงานวิจัย

3.2.1 ฮาร์ดแวร์ ที่นำมาใช้สำหรับพัฒนา ระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP ประกอบด้วย

1) เครื่องคอมพิวเตอร์

CPU : Intel Pentium 1.6 GHz

RAM : 768 MB

Hard Disk : 40 GB

2) บอร์ดไมโครคอนโทรลเลอร์ จะทำหน้าที่รับคำสั่งจาก Asterisk IP-PBX และนำคำสั่งนั้นๆ ไปควบคุมอุปกรณ์ไฟฟ้าผ่านทาง บอร์ดควบคุมอุปกรณ์ไฟฟ้า

3) บอร์ดควบคุมอุปกรณ์ไฟฟ้า ทำหน้าที่ควบคุมอุปกรณ์ไฟฟ้า

4) ADSL Modem Wireless Router ทำหน้าที่เป็นศูนย์กลางเชื่อมต่ออุปกรณ์ต่างๆ ให้ติดต่อสื่อสารกันเป็นระบบ Local Area Network และเป็น Gateway เชื่อมต่อกับผู้ให้บริการ ADSL

5) เครื่องโทรศัพท์เคลื่อนที่ เป็นส่วนที่ทำหน้าที่โทรศัพท์เข้ามาในระบบและกดสั่งงานควบคุมอุปกรณ์ไฟฟ้า

6) การ์ดโทรศัพท์ FXO รุ่น X-100P ใช้ทำหน้าที่เชื่อมต่อกับโครงข่ายโทรศัพท์ภายนอก เช่น โครงข่ายโทรศัพท์พื้นฐาน PSTN และโครงข่ายโทรศัพท์เคลื่อนที่ เป็นต้น

7) Fixed Wireless Terminal ทำหน้าที่เป็น โมดูลโครงข่ายโทรศัพท์สำหรับติดต่อสื่อสารกับโครงข่ายโทรศัพท์เคลื่อนที่

3.2.2 ซอฟต์แวร์ ที่ใช้สำหรับพัฒนาระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP ประกอบด้วย

1) โปรแกรมภาษาไพธอน ผู้พัฒนาได้นำโปรแกรมภาษาไพธอนเขียนเป็น Script file โดย Script file นี้มีหน้าที่ติดต่อสื่อสารกับ XML File จาก Microcontroller Board

2) Asterisk เวอร์ชัน 1.4 เป็น IP-PBX ที่ทำหน้าที่ควบคุมบริหารจัดการเกี่ยวกับการโทรออกหรือรับสายที่โทรเข้ามาในระบบ ที่เรียกว่า 'Dial Plan'

3) ระบบปฏิบัติการ Linux Ubuntu version 8.04 เป็นซอฟต์แวร์ระบบปฏิบัติการ หรือ Operating System (OS)

4) Softphone X-lite เป็นซอฟต์แวร์ที่มีคุณสมบัติทำหน้าที่เป็นเครื่องโทรศัพท์ ที่ใช้งานในระบบ VoIP ซึ่งสะดวกในการนำมาใช้ทดสอบระบบ ซึ่งการใช้งานนั้นจะต้องติดตั้งลงบนเครื่องคอมพิวเตอร์

### 3.3 แผนการดำเนินงาน

#### 3.3.1 รวบรวมข้อมูลและปัญหาของระบบ

รวบรวมข้อมูลที่เกี่ยวข้องกับระบบแจ้งเตือน และควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP พร้อมทั้งศึกษาถึงปัญหา ขอบเขต ข้อจำกัดของระบบและวิธีการแก้ปัญหาต่างๆ ซึ่งจะทำให้การออกแบบระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP ให้มีความเหมาะสมในการใช้งานมากขึ้น

#### 3.3.2 ศึกษาการใช้งาน โปรแกรม Asterisk

ศึกษาทฤษฎีและหลักการเขียน Dial Plan เพื่อให้ Asterisk ทำงานตามความต้องการของระบบการติดตั้งโปรแกรมบนระบบปฏิบัติการ และการตั้งค่าการใช้งาน

#### 3.3.3 ศึกษาการใช้งานบอร์ดไมโครคอนโทรลเลอร์ควบคุมอุปกรณ์ไฟฟ้า

ศึกษาการใช้งานบอร์ดไมโครคอนโทรลเลอร์มาเพื่อทำหน้าที่ติดต่อสื่อสารกับ Asterisk Server และบอร์ดควบคุมอุปกรณ์ไฟฟ้าเพื่อให้ทำงานอย่างสอดคล้องกัน

#### 3.3.4 ศึกษาเพิ่มเติมการใช้งานของโปรแกรมภาษาไพธอน

ศึกษาทฤษฎีและหลักการทำงานของคำสั่ง (Command line) การติดตั้งโปรแกรมภาษาไพธอนและศึกษาเพิ่มเติมการใช้งานภาษาไพธอน โดยศึกษาการติดต่อระหว่างภาษาไพธอนกับ Web Server

#### 3.3.5 ออกแบบระบบงานและรวบรวมอุปกรณ์ที่จำเป็นต้องใช้ในระบบ

ออกแบบระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP ในส่วนของจุดเชื่อมต่อของอุปกรณ์ต่างๆ โดยศึกษาเครื่องมือและอุปกรณ์ที่ใช้พัฒนาอย่างละเอียด พร้อมทั้งรวบรวมอุปกรณ์ในการพัฒนาระบบให้พร้อมที่สุด

#### 3.3.6 พัฒนาและทดสอบระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP

หลังจากเตรียมความพร้อมมาทั้งหมดแล้ว ก็ทำการพัฒนาระบบให้สามารถทำงานได้ตามวัตถุประสงค์ที่ตั้งไว้ โดยเมื่อมีการพัฒนาระบบไประยะหนึ่ง แล้ว หลังจากนั้น จะเริ่มทำการทดสอบย่อยเพื่อหาข้อผิดพลาดต่างๆ ภายในระบบ พร้อมทั้งทำการแก้ไข

เมื่อแก้ไขการทำงานต่างๆ ของระบบเป็นที่เรียบร้อยแล้ว จะนำไปสู่การทดสอบเพื่อวัดประสิทธิภาพของทั้งระบบ คุณภาพการแจ้งเตือน คุณภาพของการแสดงสถานะการทำงาน คุณภาพของการสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า รวมทั้งอัตราความผิดพลาดที่เกิดขึ้น ด้วยการจำลองสถานการณ์การควบคุมอุปกรณ์ไฟฟ้า

### 3.3.7 สรุปผลการพัฒนาและประโยชน์ที่จะได้รับ

นำข้อมูลที่ได้ในการจำลองสถานการณ์มาสรุปผล เพื่อใช้ในการวิเคราะห์การทำงาน และประเมินประสิทธิภาพของระบบ

### 3.3.8 ตารางการทำงาน

ตารางการทำงานวิจัยในวิทยานิพนธ์ฉบับนี้แสดงได้ในตารางที่ 3.1

ตารางที่ 3.1 แสดงขั้นตอนการดำเนินงานในระยะเวลา 14 เดือน

งาน	เดือน															
		ก.ค.-53	ก.ย.-53	ต.ค.-53	พ.ย.-53	ธ.ค.-53	ม.ค.-54	ก.พ.-54	มี.ค.-54	เม.ย.-54	พ.ค.-54	มิ.ย.-54	ก.ค.-54	ส.ค.-54	ก.ย.-54	
รวบรวมข้อมูลและปัญหาของระบบตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า																
ศึกษาการใช้งานโปรแกรม Asterisk																
ศึกษาการใช้งานบอร์ดไมโครคอนโทรลเลอร์และบอร์ดควบคุมอุปกรณ์ไฟฟ้า																
ศึกษาเพิ่มเติมการใช้โปรแกรมภาษาไพธอน																
ออกแบบระบบงานและรวบรวมอุปกรณ์ที่จำเป็นต้องใช้ในระบบ																
พัฒนาและทดสอบการใช้งานระบบ																
สรุปผลและพิจารณาประโยชน์ที่ได้รับจากระบบ																

## 3.4 ขั้นตอนและวิธีดำเนินงาน

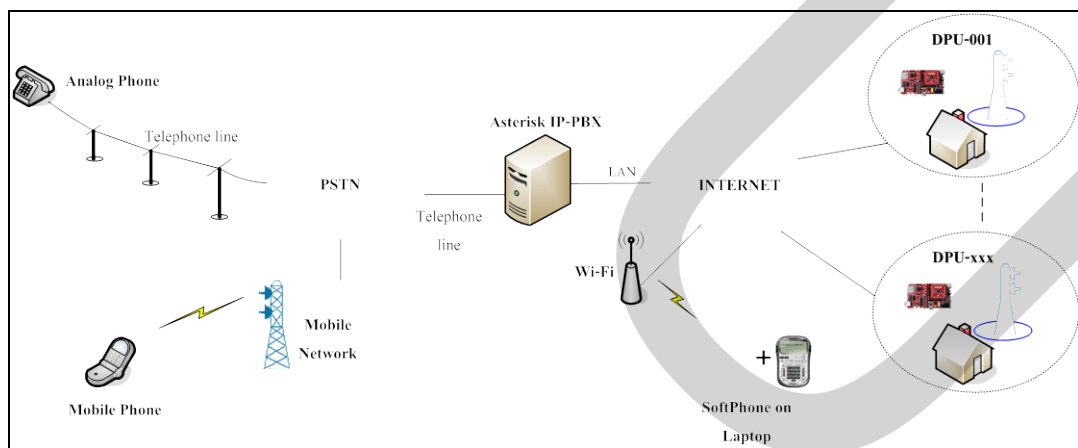
### 3.4.1 แนวคิดการทำงานของระบบ

ระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP โดยเป็นระบบที่พัฒนาขึ้นมาเพื่อนำไปใช้ในหน่วยงานหรือองค์กรต่างๆหรือตามอาคารบ้านพักอาศัยที่ต้องมีระบบดังกล่าว ซึ่งระบบสามารถแจ้งเตือนสิ่งผิดปกติ ตรวจสอบสถานะการทำงานของ อุปกรณ์ไฟฟ้า พร้อมทั้งสามารถสั่งเปิดหรือสั่งปิดได้ในคราวเดียวกัน มีขั้นตอนการทำงาน ดังนี้



- 1) ใช้บอร์ดควบคุมอุปกรณ์ไฟฟ้าหรือโซลิด-สเตต รีเลย์ (Solid state relay) ทำหน้าที่ส่งสัญญาณไฟฟ้า สำหรับสั่งให้อุปกรณ์ไฟฟ้าเปิดหรือปิดได้
- 2) ใช้บอร์ดไมโครคอนโทรลเลอร์ติดต่อกับ Asterisk ได้
- 3) ผู้ใช้สามารถโทรศัพท์เข้ามาในระบบ ด้วยโทรศัพท์ไอพีได้
- 4) ผู้ใช้สามารถโทรศัพท์เข้ามาในระบบ โดยผ่านทาง โครงข่าย โทรศัพท์พื้นฐาน PSTN หรือโครงข่ายโทรศัพท์เคลื่อนที่ก็ได้
- 5) ผู้ใช้สามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าได้
- 6) ผู้ใช้สามารถสั่งเปิดหรือสั่งปิดอุปกรณ์ไฟฟ้าได้
- 7) ระบบสามารถแจ้งเตือนสิ่งผิดปกติ ผ่านทางโครงข่ายโทรศัพท์ ได้
- 8) ระบบรองรับการขยายสถานีควบคุมได้ (Distribute)

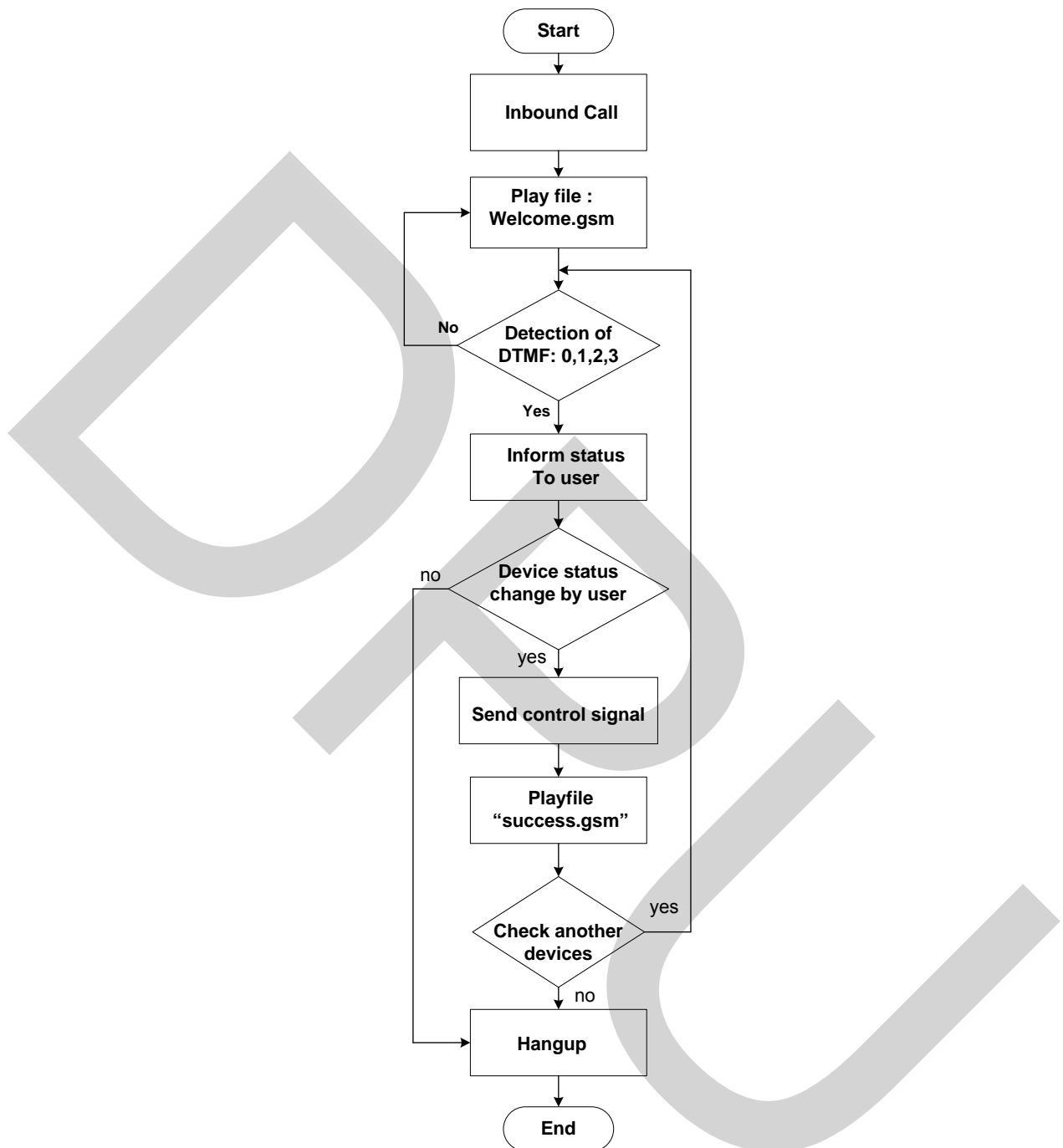
3.4.2 ระบบที่ได้ออกแบบและพัฒนาในวิทยานิพนธ์ฉบับนี้ แสดงได้ตามรูปที่ 3.1 โดยเป็นการแสดง Network Diagram ของระบบจริง โดยใช้ Asterisk IP-PBX Server จำนวน 1 เครื่องออกแบบให้มีความสามารถรองรับการขยายหรือเพิ่มชุดควบคุมได้หลายสถานี เช่น DPU-001 จนถึง DPU-xxx เป็นต้น



รูปที่ 3.1 แสดงโครงข่ายของระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP

การทำงานของระบบมีแนวทางการทำงาน ตามแผนภาพการทำงานแสดงได้ตาม รูปที่-3.2 ซึ่งมีลำดับขั้นตอนเป็น ดังนี้

- 1) เมื่อผู้ใช้โทรศัพท์เข้ามาในระบบ (Inbound Call)
- 2) จากนั้นระบบจะเล่นไฟล์เสียง Welcome.gsm
- 3) ระบบจะคอยตรวจสอบสัญญาณ ที่ผู้ใช้งานกดส่งจากแป้นโทรศัพท์ระบบDTMF ถ้าไม่ใช่ สัญญาณของหมายเลข 1, 2 หรือ 3 หรือผู้ใช้ไม่ได้กดแป้นโทรศัพท์ถ้าไม่กระทำภายในเวลาที่กำหนด ระบบจะกลับไปเล่น ไฟล์เสียง IVR อีกครั้ง
- 4) กรณีเมื่อผู้ใช้งาน กดส่งสัญญาณจากแป้นโทรศัพท์ หมายเลข 1 2 หรือ 3 ระบบจะตอบรับ ด้วยการเรียกใช้ไฟล์เสียงบอกสถานะการทำงานของอุปกรณ์ไฟฟ้า ตรงตามหมายเลขของอุปกรณ์ไฟฟ้าที่ผู้ใช้เลือกกด เช่น กด '1' เป็นการตรวจสอบสถานะการทำงานของเครื่องปรับอากาศเครื่องที่หนึ่ง โดยไฟล์เสียง IVR จะบอกว่าขณะนี้ เครื่องปรับอากาศเครื่องที่หนึ่งไม่ทำงานต้องการเปิดให้กด '1' เป็นต้น
- 5) ถ้าหากผู้ใช้ไม่ต้องการเปลี่ยนแปลงสถานะการทำงานของอุปกรณ์ไฟฟ้า แต่ต้องการออกจากระบบ ให้กดเครื่องหมายดอกจัน (\*\*\*) หรือทำการวางสาย
- 6) ถ้าผู้ใช้ต้องการเปลี่ยนสถานะการทำงานของอุปกรณ์ไฟฟ้า กรณีที่อุปกรณ์ไฟฟ้าไม่ทำงาน หากต้องการเปิดให้กด '1' และหากอุปกรณ์ไฟฟ้ากำลังทำงาน ต้องการปิดให้กด '0' ระบบก็จะเปลี่ยนสถานะการทำงานของอุปกรณ์ไฟฟ้านั้น ตามคำสั่ง ต่อจากนั้นระบบจะเล่นไฟล์เสียง IVR ไฟล์ success.gsm เป็นการบอกว่าอุปกรณ์ไฟฟ้านั้นๆ ได้เปลี่ยนสถานะการทำงานเรียบร้อยแล้ว และหากผู้ใช้ต้องการกลับไปตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าอื่น ๆ อีกระบบจะเล่นไฟล์เสียงวนลูป 3 รอบ เมื่อสิ้นสุดจะวางสาย เป็นอันดับต่อไป
- 7) หากผู้ใช้งานไม่ต้องการตรวจสอบอุปกรณ์ไฟฟ้าอื่นๆ อีกก็ให้ทำการกดเครื่องหมายดอกจัน '\*' เพื่อออกจากระบบหรือทำการวางสาย (Hangup)

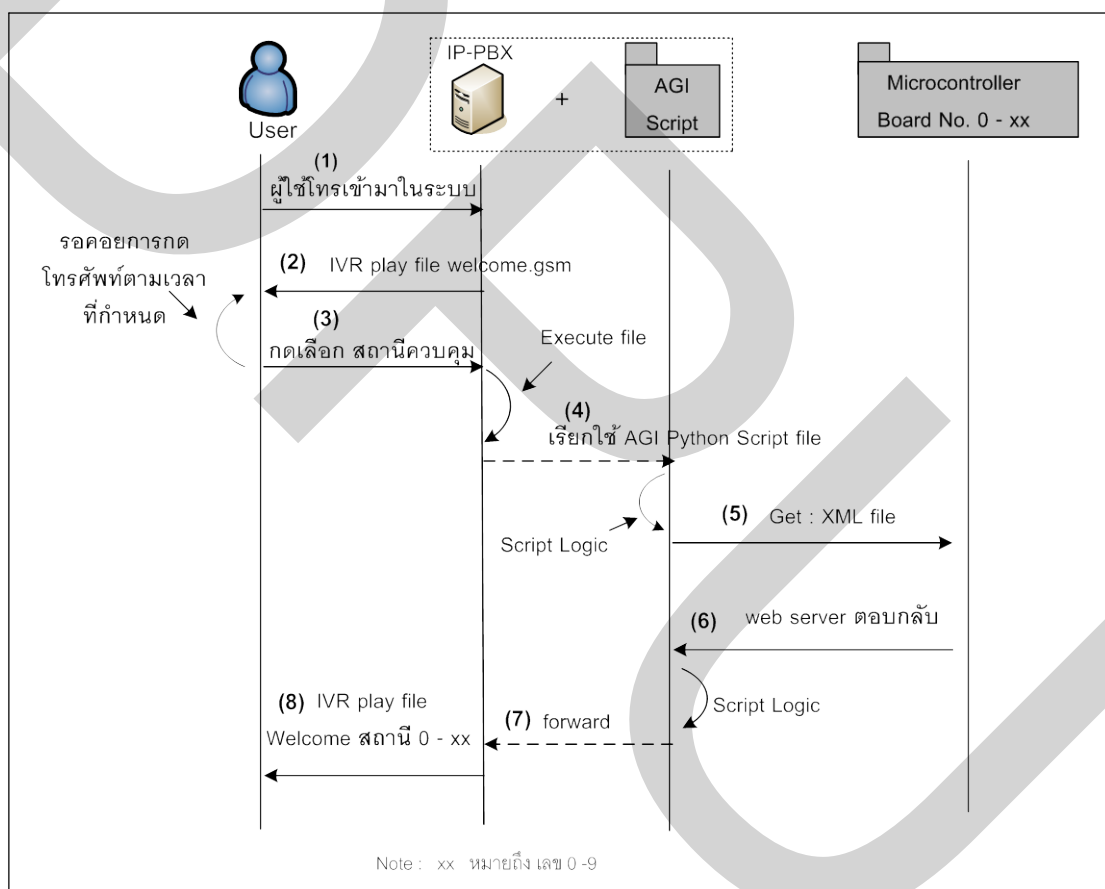


รูปที่ 3.2 แสดงขั้นตอนการทำงานของระบบ

เพื่อให้เข้าใจในรายละเอียดการออกแบบระบบมากขึ้น ดังนั้นจึงแยกส่วนของการออกแบบเป็นส่วนๆ พร้อมแนวทางการออกแบบทางด้านซอฟต์แวร์ควบคู่กันไป เพื่อให้เป็นไปตามแนวคิดการทำงานของระบบ ซึ่งสามารถแบ่งการออกแบบเป็นหัวข้อ ดังนี้

- 1) ออกแบบระบบ ให้สามารถควบคุมชุดควบคุมหรือสถานีได้หลายแห่ง
- 2) ออกแบบระบบ ให้สามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าได้
- 3) ออกแบบระบบ ให้สามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้าได้
- 4) ออกแบบระบบ ให้สามารถแจ้งเหตุผิดปกติไปยังผู้ใช้งานโดยผ่านทางโครงข่ายโทรศัพท์ PSTN หรือโครงข่ายโทรศัพท์เคลื่อนที่

1) การออกแบบระบบให้ สามารถควบคุมชุดควบคุมหรือสถานีได้หลายแห่ง มีหลักการทำงานตาม Sequence Diagram ดังรูปที่ 3.3 ดังนี้



รูปที่ 3.3 แสดงลำดับขั้นตอนการทำงาน Sequence Diagram ระบบสามารถควบคุมชุดควบคุม อุปกรณ์ไฟฟ้าหรือเพิ่มสถานีควบคุมระบบได้

การทำงานกรณีที่ระบบเชื่อมต่อกับชุดควบคุมหลายชุด นั้นผู้ใช้สามารถกดเลือกสถานีที่ต้องการตรวจสอบได้ โดยมีลำดับขั้นตอนการทำงานเป็นไปตาม Sequence Diagram รูปที่ 3.3 ซึ่ง

ในวิทยานิพนธ์ฉบับนี้ได้ยกตัวอย่างของระบบที่เชื่อมต่อกับชุดควบคุมอุปกรณ์ไฟฟ้า จำนวน 2 ชุด หรือ 2 สถานี ซึ่งมีลำดับขั้นตอนการทำงานเป็นดังนี้

ลำดับที่ 1 เมื่อผู้ใช้โทรศัพท์เข้ามาในระบบ

ลำดับที่ 2 ระบบจะทำการรับสายพร้อมกับตอบรับการโทร ด้วยการเล่นไฟล์เสียง IVR ยินดีต้อนรับ 'welcome.gsm' ซึ่งไฟล์เสียงนี้ได้ถูกกำหนดให้ทำการการวนลูปใน กรณีที่ผู้ใช้ไม่กดส่งสัญญาณที่เป็นตัวเลขใดๆ จากแป้นโทรศัพท์ภายในเวลาที่กำหนด (default 10 วินาที)

ลำดับที่ 3 ผู้ใช้สามารถกดเลือกสถานีที่ต้องการจะตรวจสอบตามคำบอกของ IVR โดยในการพัฒนาครั้งนี้ ระบบถูกออกแบบให้เลือกได้ 2 สถานีคือ เลือกสถานี DPU-001 ให้กด '1' และถ้าเลือกสถานี DPU-002 ให้กด '2'

ลำดับที่ 4 และ 5 เมื่อผู้ใช้กดเลือกสถานีใดสถานีหนึ่งแล้ว ระบบจะประมวลผลการร้องขอด้วยการตรวจสอบ XML file จาก Microcontroller Board ที่ทำหน้าที่เป็น Web Server

ลำดับที่ 6 7 และ 8 เมื่อระบบรับรู้ว่ามีสถานีใดสถานีหนึ่งถูกเลือกแล้ว ระบบก็จะเรียกใช้ไฟล์เสียง welcome.gsm หรือ ไฟล์เสียง "ยินดีต้อนรับเข้าสู่สถานี DPU-xxx" เป็นต้น

การออกแบบทางด้านซอฟต์แวร์ เป็นการ กำหนดให้ระบบทำงาน เป็นไปตาม Sequence Diagram รูปที่ 3.3 โดยซอฟต์แวร์นี้ จะทำหน้าที่เป็นแผนการ โทรศัพท์ที่เรียกว่า Dial Plan ที่เป็น Function การทำงานของ Asterisk และ Dial Plan นี้ จะถูกวางเอาไว้ใน ไดรเร็กทอรี (Directory): /etc/asterisk/extensions.conf ดังแสดงได้ตามรูปที่ 3.4

จากแผนการ โทรศัพท์หรือ Dial Plan ตามรูปที่ 3.4 มีลำดับขั้นตอนการทำงานเป็นดังนี้ โดยในการออกแบบได้กำหนดให้หมายเลขโทรศัพท์ของระบบเป็นหมายเลข SIP เบอร์ 999 และเมื่อผู้ใช้โทรศัพท์เรียกหมายเลข 999 ด้วยโทรศัพท์ระบบ VoIP หรือ Softphone ระบบจะทำการตอบรับด้วย Application 'Answer' และเล่นไฟล์เสียง welcome01.gsm โดยผ่านทาง Application 'Background' จากนั้นจะหน่วงเวลาเป็นเวลา 10 วินาที ด้วย Application WaitExten (10) ในช่วงเวลานี้ถ้าผู้ใช้ไม่กดส่งสัญญาณใดๆ จากแป้นโทรศัพท์ ระบบก็จะวางสายโดยเรียกใช้ Application Hangup ()

```

[internal_calls]
    exten => 999,1,Answer()
    exten => 999,n,Background(topup/welcome01)
    exten => 999,n,WaitExten(10)
    exten => 999,n,Hangup()
    exten => s,1,Answer()
    exten => s,n,Background(topup/welcome01)
    exten => s,n,WaitExten(5)
    exten => s,n,Hangup()
    exten => 1,1,Goto(select-site,777,1)
    exten => 2,1,Goto(select-site,888,1)

[select-site]
    exten => 777,1,AGI(/var/lib/asterisk/agi-bin/menu01.py)
    exten => 888,1,AGI(/var/lib/asterisk/agi-bin/menu02.py)

```

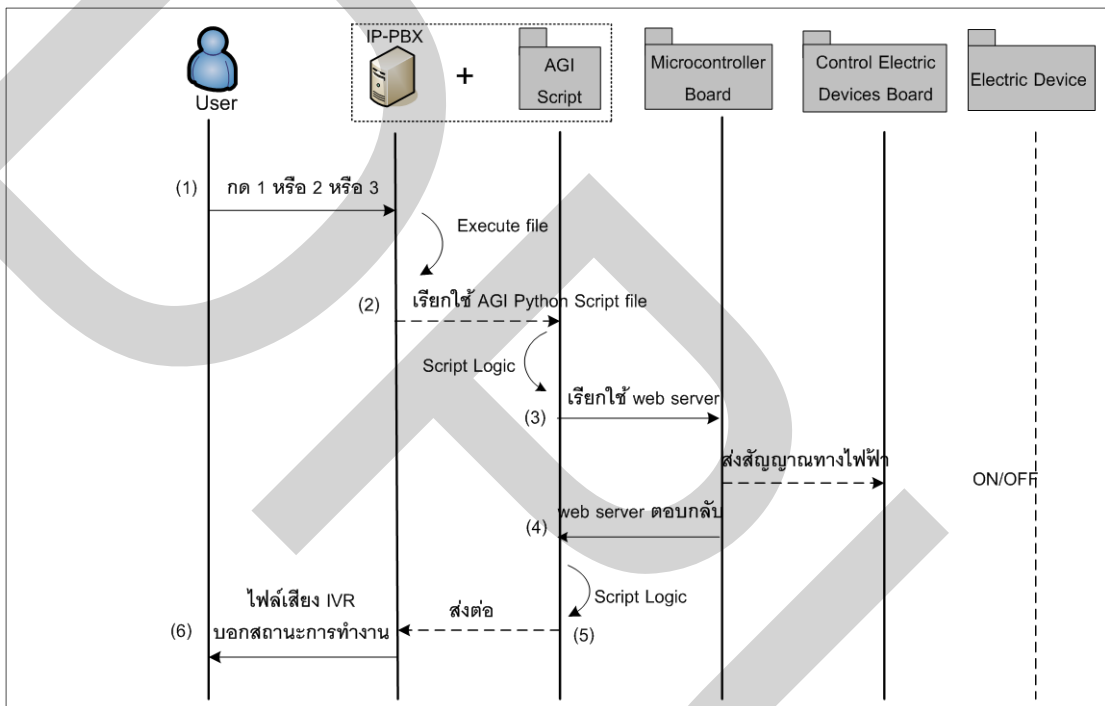
รูปที่ 3.4 แสดงการออกแบบซอฟต์แวร์ ในส่วนของ Dial Plan ของระบบ ที่สามารถควบคุมชุดควบคุมได้หลายชุด

ทั้งนี้ระบบ ได้ถูกออกแบบให้ผู้ใช้ สามารถกดเลือก '1','2',.... xx หมายความว่า ผู้ใช้สามารถกดเลือกชุดควบคุมหรือสถานีได้ตามที่ได้ออกแบบไว้ จนถึง xx สถานี โดย 'xx' หมายถึงจำนวนของชุดควบคุมหรือจำนวนสถานี กำหนดเป็นตัวเลข 0-9 ตัวอย่างเช่น เมื่อผู้ใช้กด '1' ก็จะเป็นการเรียกใช้ Application Goto เป็นผลให้มีเรียกใช้ Extensions [select-site] -> เรียกใช้ หมายเลข 777 โดยภายใน Extension นั้นได้ถูกกำหนดให้เรียกใช้ AGI Module ของ Asterisk software และเรียกใช้ AGI Script ไฟล์ Menu01.py ที่ผู้พัฒนาเขียนขึ้นมาด้วยภาษาไพธอน และเช่นเดียวกันหากผู้ใช้กดเลือกสถานีอื่นๆ (2 -xx) ขั้นตอนการทำงานก็จะเป็นไปในรูปแบบเดียวกัน

ในทำนองเดียวกันผู้ใช้สามารถโทรศัพท์เข้ามาในระบบด้วยโครงข่ายโทรศัพท์พื้นฐาน PSTN และโครงข่ายโทรศัพท์เคลื่อนที่ ได้ด้วยเช่นเดียวกัน ผ่านทางช่องสัญญาณ โทรศัพท์และเขียน

สำหรับรับโทรศัพท์ได้เป็น exten => s,1,Answer() ต่อจากนั้นขั้นตอนการทำงานต่างๆ ก็จะเหมือนกันกับที่ได้กล่าวมาแล้วข้างต้น

2) ออกแบบระบบ ให้สามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าได้ ซึ่งระบบที่ออกแบบ ให้สามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า มีหลักการทำงานตาม Sequence Diagram ดังรูปที่ 3.5



รูปที่ 3.5 แสดงลำดับขั้นตอนการทำงาน Sequence Diagram กรณีตรวจสอบสถานะการทำงานของ อุปกรณ์ไฟฟ้า

จากรูปที่ 3.5 แสดงลำดับขั้นตอนการทำงาน ในกรณีตรวจสอบสถานะการทำงานของ อุปกรณ์ไฟฟ้า ซึ่งการทำงานในส่วนนี้จะอธิบายขั้นตอนการทำงานต่อจากขั้นตอนการเลือกสถานี ตามรูปที่ 3.4 โดยเมื่อผู้ใช้กดเลือกสถานีใดสถานีหนึ่งแล้วนั้น IVR เล่นไฟล์เสียงยินดีต้อนรับพร้อม กับบอกให้เลือกอุปกรณ์ที่ต้องการตรวจสอบ ระบบที่พัฒนาขึ้นนี้ออกแบบให้มีความสามารถเลือก ตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าได้จำนวน 3 อุปกรณ์ คือ กด ‘1’ เป็นการตรวจสอบ สถานะการทำงานของเครื่องปรับอากาศเครื่องที่หนึ่ง กด ‘2’ เป็นการตรวจสอบสถานะการทำงานของ เครื่องปรับอากาศเครื่องที่สอง และหากกด ‘3’ เป็นการตรวจสอบสถานะการทำงานของไฟแสงสว่าง ซึ่งมีลำดับขั้นตอนการทำงานเป็นดังนี้

ลำดับที่ 1 เมื่อผู้ใช้ได้กดเลือกอุปกรณ์ที่ต้องการตรวจสอบด้วยการเลือกกด ‘1’ ‘2’ หรือ ‘3’ (จากการกดเลือกจากแป้นโทรศัพท์)

ลำดับที่ 2 ระบบจะทำการประมวลผล หรือ Execute File ด้วยการเรียกใช้ AGI Python Script File

ลำดับที่ 3 ประมวลผลทางลอจิก (Script Logic) พร้อมกับตรวจสอบ XML File จากบอร์ดไมโครคอนโทรลเลอร์

ลำดับที่ 4 5 และ 6 เมื่อเครื่อง Asterisk Server ซึ่งเมื่อได้รับค่าที่ส่งกลับมาจาก บอร์ดไมโครคอนโทรลเลอร์ แล้วพร้อมกับประมวลผลว่า เป็นค่าของอุปกรณ์ตัวที่ผู้ใช้ต้องการให้แสดงสถานะแล้ว จากนั้นระบบจะเรียกใช้ไฟล์เสียง IVR เพื่อบอกว่าอุปกรณ์ตัวที่ถูกเลือกกำลังทำงานอยู่ในสถานะใด (เปิดหรือปิดอยู่) ยกตัวอย่างเช่น ผู้ใช้กด ‘1’ ซึ่งเป็นการตรวจสอบสถานะการทำงานของเครื่องปรับอากาศเครื่องที่หนึ่ง สมมติว่าเครื่องปรับอากาศเครื่องที่หนึ่งกำลังทำงานหรือเปิดอยู่ IVR ก็จะบอกว่า “ขณะนี้เครื่องปรับอากาศเครื่องที่หนึ่งกำลังทำงาน ” เป็นต้น หากผู้ใช้กดเลือก 2 หรือ 3 ขั้นตอนการทำงานก็จะเหมือนกับที่ได้อธิบายข้างต้น

การออกแบบทางด้านซอฟต์แวร์ กรณีตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า เพื่อให้เป็นไปตาม Sequence Diagram รูปที่ 3.5 โดยซอฟต์แวร์นี้เขียนเป็น Script file ด้วยโปรแกรมภาษาไพธอน และได้ฝังตัวเอาไว้ใน Asterisk Program ตาม Directory: /var/lib/asterisk/agi-bin /Menu01.py และสามารถแสดงโปรแกรมสคริป Menu01.py ดังแสดงตามรูปที่ 3.6 ดังนี้

```

.
.
8  """***** AirCondition-01 *****"""
9  if i == '1':
10     txt = urllib.urlopen("http://192.168.1.201/status.xml").read()
11     stxt = []
12     stxt = txt.splitlines()
13     LED = ""

```

รูปที่ 3.6 แสดงการออกแบบทางด้านซอฟต์แวร์ กรณีตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า



```

14 LED1 = ""
15 for x in stxt:
16     if "<led1>" in x:
17         LED1 = str(x[6:7])
18         if LED1 == '1':
19             j = agi.get_data('./topup/air01-on01',5000,1)
20             if j == '0':
21                 txt = urllib.urlopen("http://192.168.1.201/leds.cgi?led=1").read()
22                 txt = urllib.urlopen("http://192.168.1.201/status.xml").read()
23                 i = agi.get_data('./topup/air01-offsuccess01',2000,1)
24             if LED1 == '0':
25                 j = agi.get_data('./topup/air01-off01',5000,1)
26             if j == '1':
27                 txt = urllib.urlopen("http://192.168.1.201/leds.cgi?led=1").read()
28                 txt = urllib.urlopen("http://192.168.1.201/status.xml").read()
29                 i = agi.get_data('./topup/air01-onsuccess01',2000,1)
30             """***** AirCondition-02 *****"""
31             if i == '2':
32                 txt = urllib.urlopen("http://192.168.1.201/status.xml").read()
33                 stxt = []
34                 stxt = txt.splitlines()
35                 LED = ""
36                 LED2 = ""
37                 for x in stxt:
38                     if "<led2>" in x:

```

รูปที่ 3.6 แสดงการออกแบบทางด้านซอฟต์แวร์ กรณีตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า (ต่อ)

```

38     LED2 = str(x[6:7])
39     if LED2 == '1':
40         j = agi.get_data('./topup/air02-on01',5000,1)
41         if j == '0':
42             txt = urllib.urlopen("http://192.168.1.201/leds.cgi?led=2").read()
43             txt = urllib.urlopen("http://192.168.1.201/status.xml").read()
44             i = agi.get_data('./topup/air02-offsuccess01',2000,1)
45             if LED2 == '0':
46                 j = agi.get_data('./topup/air02-off01',5000,1)
47                 if j == '1':
48                     txt = urllib.urlopen("http://192.168.1.201/leds.cgi?led=2").read()
49                     txt = urllib.urlopen("http://192.168.1.201/status.xml").read()
50                     i = agi.get_data('./topup/air02-onsuccess01',2000,1)
51
52     """***** Light *****"""
53     if i == '3':
54         txt = urllib.urlopen("http://192.168.1.201/status.xml").read()
55         stxt = []
56         stxt = txt.splitlines()
57         LED = ""
58         LED3 = ""
59         for x in stxt:
60             if "<led3>" in x:

```

รูปที่ 3.6 แสดงการออกแบบทางด้านซอฟต์แวร์ กรณีตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า (ต่อ)

```

61     LED3 = str(x[6:7])
62     if LED3 == '1':
63         j = agi.get_data('./topup/light-on01',5000,1)
64         if j == '0':
65             txt = urllib.urlopen("http://192.168.1.201/leds.cgi?led=3").read()
66             txt = urllib.urlopen("http://192.168.1.201/status.xml").read()
67             i = agi.get_data('./topup/light-offsuccess01',2000,1)
68     if LED3 == '0':
69         j = agi.get_data('./topup/light-off01',5000,1)
70         if j == '1'
71             txt= urllib.urlopen("http://192.168.1.201/leds.cgi?led=3").read()
72             txt = urllib.urlopen("http://192.168.1.201/status.xml").read()
73             i = agi.get_data('./topup/light-onsuccess01',2000,1)
74 except:
75     #agi.hangup()
76     #os.abort()
77     pass
78 agi.hangup()
79 os.abort()
80 """"***** END *****"""

```

รูปที่ 3.6 แสดงการออกแบบทางด้านซอฟต์แวร์ กรณีตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า (ต่อ)

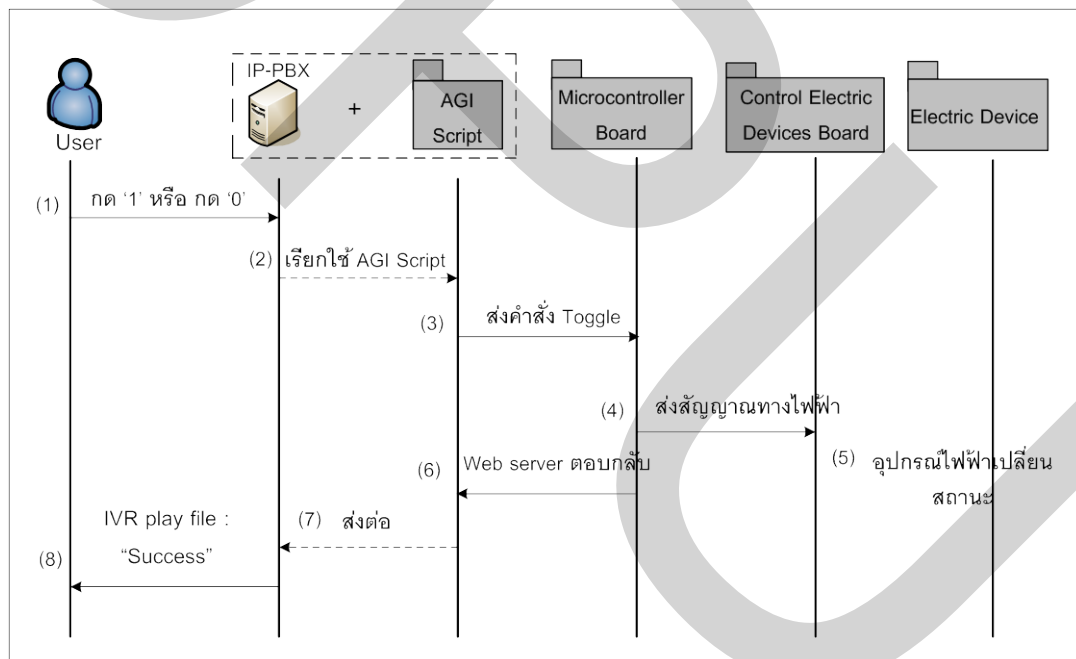
จาก AGI Script Program รูปที่ 3.6 สามารถอธิบายหลักการทำงานได้ดังนี้ หลังจากที่ผู้ใช้กด '1' และเข้าสู่สถานี DPU001 จากนั้น IVR ก็จะเล่นไฟล์เสียงบอกว่า “ขณะนี้ได้เข้าสู่สถานี DPU001 ต้องการตรวจสอบสถานะการทำงานของเครื่องปรับอากาศเครื่องที่หนึ่งให้กด '1' ต้องการ

ตรวจสอบสถานะการทำงานของเครื่องปรับอากาศเครื่องที่สองให้กด '2' และต้องการตรวจสอบสถานะการทำงานของไฟแสงสว่างให้กด 3"

สมมติว่าผู้ใช้เลือกกด '1' (โปรแกรมบรรทัดที่ 9) เพื่อเลือกตรวจสอบสถานะการทำงานของเครื่องปรับอากาศเครื่องที่หนึ่ง (สมมติว่าเครื่องปรับอากาศเครื่องที่หนึ่งขณะนี้ไม่ทำงาน) ระบบจะเรียกใช้ไฟล์เสียง air01-off01.gsm (บรรทัดที่ 25) ซึ่งไฟล์เสียงนี้จะบอกว่าขณะนี้เครื่องปรับอากาศเครื่องที่หนึ่งไม่ทำงาน เป็นต้น

ถ้าผู้ใช้กดเลือกตรวจสอบสถานะการทำงานของเครื่องปรับอากาศเครื่องที่สองให้กด '2' (บรรทัดที่ 30) และหากผู้ใช้กดเลือกตรวจสอบสถานะการทำงานของไฟแสงสว่าง ก็ให้กด '3' (บรรทัดที่ 53) โปรแกรมสคริปก็ทำงานไปในทำนองเดียวกัน

3) ออกแบบระบบให้สามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้ามีหลักการทำงานตาม Sequence Diagram ดังรูปที่ 3.7



รูปที่ 3.7 แสดงลำดับขั้นตอนการทำงาน Sequence Diagram กรณีสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า

จากรูปที่ 3.7 แสดงลำดับขั้นตอนการทำงาน กรณีสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า คือเมื่อผู้ใช้กดเลือกสถานีใดสถานีหนึ่งแล้วนั้น IVR เล่นไฟล์เสียงยินดีต้อนรับและบอกว่าขณะนี้อุปกรณ์ที่ถูกเลือกนั้นกำลังอยู่ในสถานะใดเปิดหรือปิดอยู่ พร้อมกับบอกว่าหากต้องการเปลี่ยนสถานะของอุปกรณ์ โดยต้องการเปิดให้กด '1' หรือหากต้องการปิดให้กด '0' สมมติว่าขณะนี้เครื่องปรับอากาศ

เครื่องที่หนึ่งที่อยู่ภายในสถานี DPU001 ไม่ทำงาน ซึ่งขั้นตอนการทำงานเพื่อสั่งให้เครื่องปรับอากาศเครื่องที่หนึ่งทำงาน เป็นดังนี้

ลำดับที่ 1 เมื่อผู้ใช้ต้องการเปิดเครื่องปรับอากาศเครื่องที่หนึ่ง ด้วยการกด ‘1’

ลำดับที่ 2 และ 3 ระบบเรียกใช้ AGI Script และส่งคำสั่งให้เปิดเครื่องปรับอากาศเครื่องที่หนึ่งไปที่ Microcontroller Board

ลำดับที่ 4 และ 5 เมื่อ Microcontroller Board รับรู้ถึงคำสั่งนั้น ก็ส่งสัญญาณทางไฟฟ้าไปสั่งให้ชุดควบคุมอุปกรณ์ไฟฟ้า ให้ทำการเปิดเครื่องปรับอากาศเครื่องที่หนึ่งและเครื่องปรับอากาศเครื่องที่หนึ่งก็เริ่มทำงาน

ลำดับที่ 6, 7 และ 8 เป็นลำดับที่ต่อเนื่องกันซึ่งจะ เรียกใช้ไฟล์เสียง ‘success.gsm’ เป็น IVR บอกว่าเครื่องปรับอากาศเครื่องที่หนึ่งได้ถูกเปิดเรียบร้อยแล้ว

การออกแบบทางด้านซอฟต์แวร์ ในกรณีสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า แสดงได้ตามรูปที่ 3.5 โดยบรรทัดที่ 26 – 29 คือ เมื่อผู้ใช้กด ‘1’ เป็นการส่งคำสั่งไปเปิดเครื่องปรับอากาศเครื่องที่หนึ่งพร้อมกับเรียกใช้ไฟล์เสียงไฟล์ air01-onsuccess01.gsm ซึ่งบอกว่าเครื่องปรับอากาศเครื่องที่หนึ่งได้ถูกเปิดเรียบร้อยแล้ว

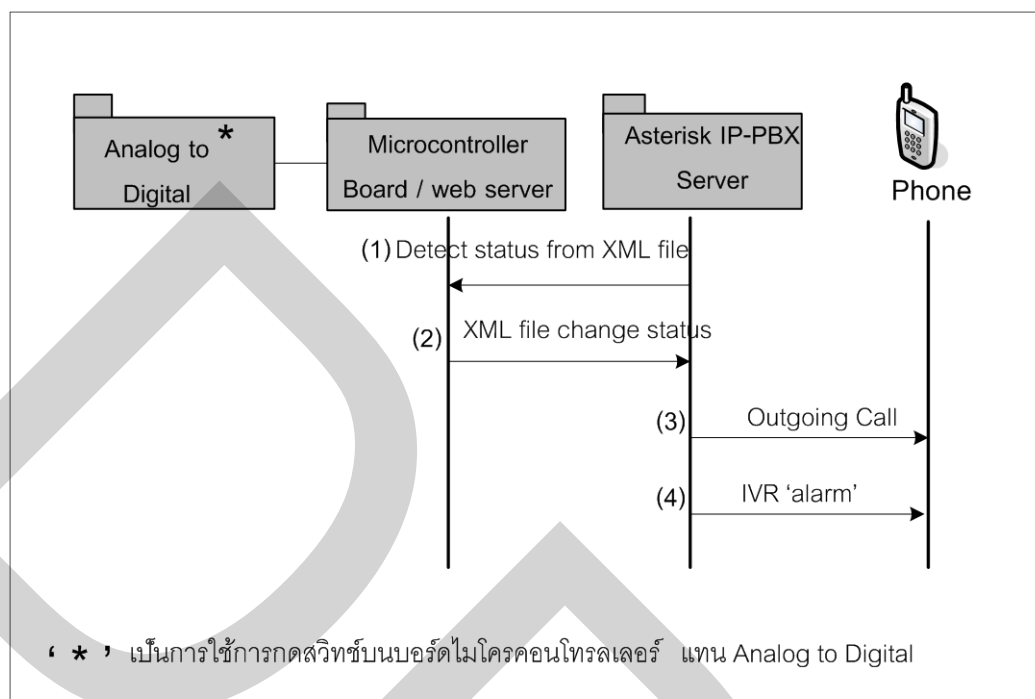
4) ออกแบบระบบให้สามารถแจ้งเหตุผิดปกติมายังผู้ใช้ โดยผ่านทางโครงข่ายโทรศัพท์พื้นฐาน PSTN หรือโครงข่ายโทรศัพท์เคลื่อนที่ ระบบที่ออกแบบให้สามารถแจ้งเหตุผิดปกติมายังผู้ใช้ ผ่านทางโครงข่ายโทรศัพท์พื้นฐาน PSTN หรือโครงข่ายโทรศัพท์เคลื่อนที่ได้ ซึ่งมีหลักการทำงานเป็นไปตาม Sequence Diagram ดังรูปที่ 3.8

การออกแบบระบบให้สามารถแจ้งเหตุผิดปกติมายังผู้ใช้ผ่านทางโครงข่ายโทรศัพท์โดยมีขั้นตอนการทำงานตาม Sequence Diagram รูปที่ 3.8 ดังนี้

ลำดับที่ 1 Asterisk IP-PBX Server จะทำหน้าที่คอยตรวจสอบสถานะของค่า ‘up’ (เป็นค่า default) และ ‘dn’ จาก XML file ของ Microcontroller Board โดยถูกออกแบบให้คอยตรวจสอบสถานะตามเวลาที่กำหนด เช่น ตั้งเวลาให้ทำการตรวจสอบทุกๆวัน เป็นต้น

ลำดับที่ 2 โดยการทำงานเมื่อเกิดเหตุผิดปกติเกิดขึ้น Analog to Digital Module จะส่งค่ามายัง Microcontroller Board ซึ่งทำให้เกิดการเปลี่ยนแปลงสถานะจาก ‘up’ เป็น ‘dn’ เกิดขึ้น พร้อมกับแจ้งไปยัง Asterisk IP-PBX Server

ลำดับที่ 3 ในระยะเวลาอันสั้น Asterisk Server ก็จะทำการโทรออก (Outgoing call) ไปแจ้งให้ผู้ใช้ผ่านทางโทรศัพท์เคลื่อนที่ให้ได้รับรู้ถึงปัญหาที่เกิดขึ้น



รูปที่ 3.8 แสดงลำดับขั้นตอนการทำงาน Sequence Diagram กรณีแจ้งเหตุผิดปกติไปยังผู้ใช้ผ่านทางระบบ โทรศัพท์ PSTN หรือระบบโทรศัพท์เคลื่อนที่

ลำดับที่ 4 ระบบบอกเป็นไฟล์เสียง IVR เพื่อแจ้งเหตุผิดปกตินั้นๆ โดยในระบบนี้ผู้พัฒนา กำหนดให้แจ้งเหตุผิดปกติเป็น 2 แบบคือ IVR บอกถึงอุปกรณ์ไฟฟ้าชำรุดไม่ทำงานและบอกว่า มีเหตุอุณหภูมิสูงผิดปกติเกิดขึ้น

หมายเหตุ ผู้พัฒนาได้จำลองเหตุการณ์การเกิดเหตุผิดปกติด้วยการกดสวิตช์บน Microcontroller Board แทน Analog to Digital

การออกแบบทางด้านซอฟต์แวร์ให้ ระบบสามารถแจ้งเหตุผิดปกติมายังผู้ใช้ ผ่านทางระบบโทรศัพท์ มีขั้นตอน ดังนี้

1) อันดับแรกเขียน script file : check-status.py เพื่อทำหน้าที่ตรวจจับการเปลี่ยนแปลงค่าของ xml file (status.xml) จากบอร์ดไมโครคอนโทรลเลอร์และ วางไว้ที่ Directory : /var /lib /asterisk/agi-bin/ โดยการออกแบบ script file นั้นสามารถเขียนด้วยภาษาไพธอนได้ ตามรูปที่ 3.9 ดังต่อไปนี้

```

1  #!/usr/bin/python
2  import sys,os,urllib,cgi
3  from asterisk.agi import AGI
4  agi = AGI()
5  data = urllib.urlopen('http://192.168.1.200/status.xml').read()
6  if <btn0>dn</btn0>in data:
7      agi.appexec('Dial zap/0812320053')
8      agi.hangup()

```

รูปที่ 3.9 แสดง script file สำหรับตรวจสอบสถานะจาก xml file

จากรูปที่ 3.9 เริ่มแรกเป็นการประกาศว่าใช้ภาษาไพธอน ต่อไปบรรทัดที่ 2 เป็นการเรียกใช้ library sys, OS, urllib และ cgi บรรทัดที่ 3 กำหนดเรียกใช้ function AGI จาก Asterisk agi เป็นการกำหนดค่าตัวแปรโดยให้ agi มีค่าเป็น AGI () บรรทัดที่ 5 กำหนดให้ data เป็นค่าที่ได้จากการอ่าน xml file บรรทัดที่ 6 กำหนด condition if โดยเมื่อปุ่ม btn ถูกกดก็จะมี การเปลี่ยนสถานะจาก up ไปเป็น dn บรรทัดที่ 7 กำหนดเรียกใช้ application agi.appexec เพื่อโทรศัพท์ไปยังเบอร์ 08-1232-0053 และบรรทัดที่ 8 กำหนดให้วางสาย

2) เขียน Script Asterisk Auto Call file: auto-call.call สำหรับเรียกใช้ check-status.py และเอาวางไว้ที่ Directory: /var/spool/asterisk/tmp/auto-call.call โดยการออกแบบ script file มีรายละเอียดตามรูปที่ 3.10 ดังต่อไปนี้

```

1  Channel: Local/555@default
2  MaxRetries: 0
3  RetryTime: 15
4  WaitTime: 15
5  Applications: AGI
6  Data: /var/lib/asterisk/agi-bin/check-status.py

```

รูปที่ 3.10 แสดง script file auto call สำหรับเรียกใช้ check-status.py

จากรูปที่ 3.10 แสดง script file สำหรับการโทรออกแบบอัตโนมัติหรือที่ auto call โดยบรรทัดแรก กำหนดให้โทรออกจาก SIP เบอร์ 555 ที่อยู่ใน extension default บรรทัดที่ 3 ซึ่งถูกกำหนดให้โทรทุก 15 วินาที บรรทัดที่ 4 กำหนดให้เวลารอคอย 15 วินาที บรรทัดที่ 5 กำหนดให้ใช้ application AGI บรรทัดที่ 6 กำหนดให้เรียกใช้สคริปต์ check-status.py

3) เขียน Dial Plan ที่ Directory : /etc/asterisk/extensions/ สำหรับกำหนดให้โทรออกไปยังระบบโทรศัพท์พื้นฐาน PSTN หรือระบบโทรศัพท์เคลื่อนที่ พร้อมกับเรียกใช้ไฟล์เสียงสำหรับแจ้งเหตุที่เกิดขึ้นต่อผู้ใช้ ด้วยระบบ IVR โดยการออกแบบ Dial Plan นี้กำหนดให้โทรออกจากระบบด้วย SIP เบอร์ 555 ซึ่งการออกแบบ Dial Plan มีรายละเอียดตามรูปที่ 3.11 ดังนี้

```
[default]
    exten => 555,1,Answer()
    exten => 555,n,WaitExten(35)
    exten => 555,n,Background(followme/dpu001-tem-alarm)
    exten => 555,n,WaitExten(5)
    exten => 555,n,Hangup()
```

รูปที่ 3.11 แสดง Dial Plan สำหรับโทรออกจากระบบไปยังผู้ใช้

จากรูปที่ 3.11 เป็นการกำหนด Dial Plan สำหรับให้ระบบโทรออกไปยังผู้ใช้ Dial Plan นี้กำหนดให้อยู่ใน Extension default ลำดับแรกนั้น เมื่อระบบโทรออกจากระบบด้วย SIP เบอร์ 555 จากนั้นเมื่อผู้รับสายแล้วระบบจะหน่วงเวลา 35 วินาที (WaitExten(35)) ต่อจากนั้นผู้ใช้จะได้ยินเสียง IVR ที่เป็นการแจ้งเหตุผิดปกติ เช่น dpu001-tem-alarm ซึ่งเป็นการแจ้งว่า ‘ที่สถานี dpu001 มี อุณหภูมิสูงผิดปกติโปรดตรวจสอบโดยด่วน’ จากนั้นระบบจะหน่วงเวลา 5 วินาที (WaitExten(5)) และวางสายเป็นอันดับต่อไป

4) เขียน Crontab สำหรับตั้งเวลาการโทร ศัพท์เข้าไปในระบบ เพื่อวนลูปตรวจเช็คค่าจาก XML File โดยใช้คำสั่ง crontab -e บน Linux OS ซึ่งเป็นการกำหนดให้ทำการนำ ไฟล์ auto-call.call ที่ถูกวางอยู่ใน Directory: /var/spool/asterisk/tmp ไปวางเอาไว้ใน Directory : /var/spool/asterisk/outgoing ตามช่วงเวลาที่กำหนด เช่น กำหนดให้ทำการตรวจสอบ status.xml ช่วงเวลา 10-11 โมงเช้าทุกวัน เป็นต้น แต่สำหรับในการทดสอบระบบในส่วนนี้ จะกำหนดให้มีการตรวจสอบทุกๆ 1 นาที ซึ่งการเขียน script file เป็นดังต่อไปนี้

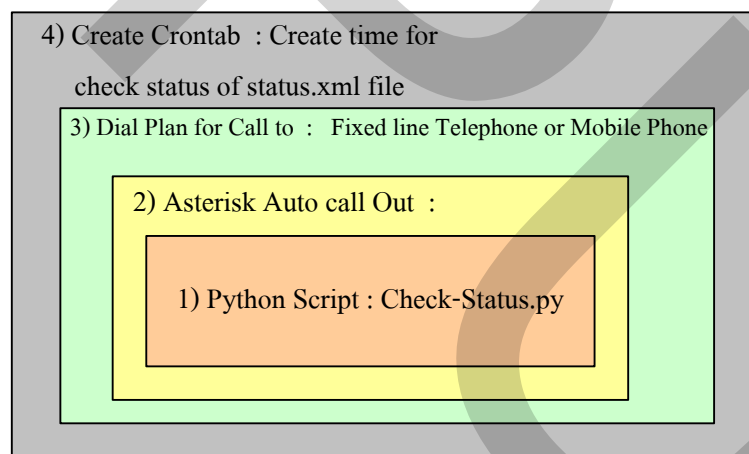


```
# m h dom mom dow  command
***** cp /var/spool/asterisk/tmp/script01.call /var/spool/asterisk/outgoing
```

รูปที่ 3.12 แสดง crontab script file

จากรูปที่ 3.12 สามารถอธิบายการทำงานได้ดังนี้ โดยในบรรทัดแรกเป็นรูปแบบทั่วไปของ crontab และบรรทัดที่ 2 ‘\*\*\*\*\*’ เป็นการกำหนดช่วงเวลา สำหรับกำหนดให้ระบบทำการ copy ไฟล์จาก directory หนึ่งเอาไปวางในอีก directory หนึ่ง เช่น กำหนดให้ทุกนาที ให้ทำการ copy ไฟล์ script01.call จากdirectory: /var/spool/asterisk/tmp เอาไปวางไว้ใน directory: /var/spool/asterisk/outgoing ซึ่งเป็นการกำหนดให้โทรออกแบบอัตโนมัติ นั่นเอง

การออกแบบทางด้านซอฟต์แวร์ให้ระบบสามารถแจ้งเหตุผิดปกติมายังผู้ใช้ผ่านทางโครงข่ายโทรศัพท์ สามารถสรุปได้ตามรูปที่ 3.13

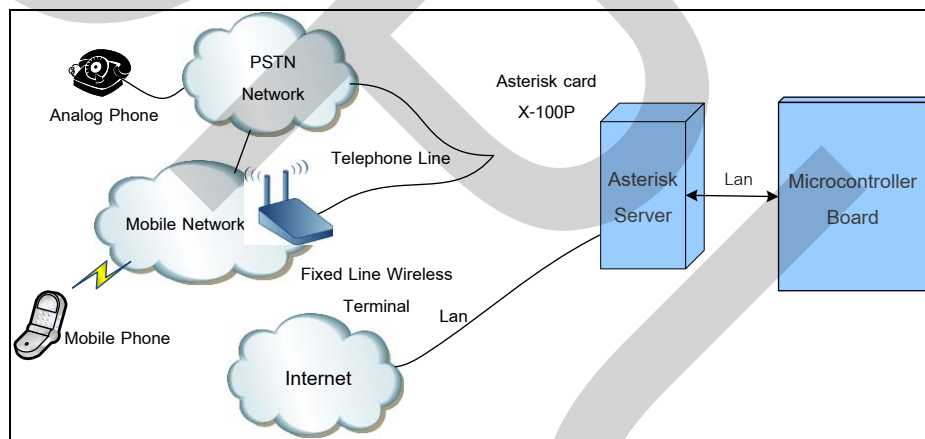


รูปที่ 3.13 แสดงขั้นตอนการเขียนซอฟต์แวร์สำหรับระบบแจ้งเหตุผิดปกติมายังผู้ใช้ผ่านทางโครงข่ายโทรศัพท์พื้นฐาน PSTN หรือโครงข่ายโทรศัพท์เคลื่อนที่

### 3.5 การออกแบบทางด้านฮาร์ดแวร์

การออกแบบทางด้านฮาร์ดแวร์ โดยเป็นการ นำเอา Asterisk IP-PBX Server บอร์ดไมโครคอนโทรลเลอร์ บอร์ดทรานซิสเตอร์ขับเคลื่อนและบอร์ดควบคุมอุปกรณ์ไฟฟ้า เชื่อมต่อเข้าด้วยกันเป็นระบบ ซึ่งหลักๆแล้วการออกแบบทางด้านฮาร์ดแวร์จะมีด้วยกัน 2 ส่วนคือ

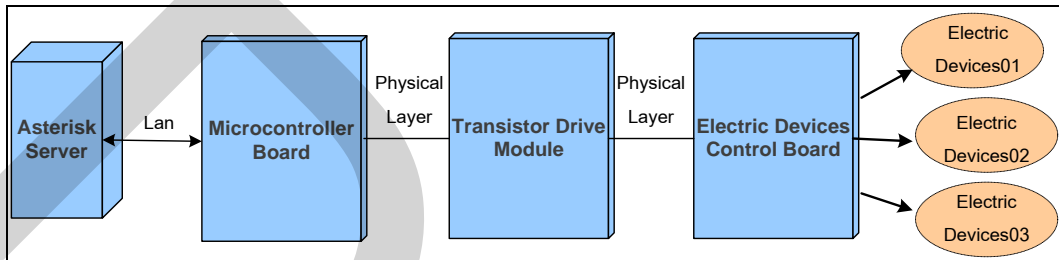
3.5.1 ส่วนติดต่อสื่อสารกับโครงข่ายโทรศัพท์ภายนอก ซึ่งส่วนของการติดต่อสื่อสารกับโครงข่ายโทรศัพท์ภายนอกสามารถแสดงได้ตามรูปที่ 3.14 ซึ่งมีจุดประสงค์ ออกแบบให้ระบบ สามารถติดต่อสื่อสาร กับโครงข่าย โทรศัพท์พื้นฐาน PSTN (TOT และ TT&T) และโครงข่าย โทรศัพท์เคลื่อนที่ได้นั้น จะต้องติดตั้งการ์ดเพิ่มเติม เพื่อทำหน้าที่รับโทรศัพท์จากภายนอก โดยทั่วไปแล้ว การ์ดชนิดนี้จะเรียกว่า การ์ด FXO (FXO card) ที่นิยมนำมาใช้งานจะเป็นการ์ด FXO รุ่น X-100P ซึ่งมีคุณสมบัติเป็นการ์ดแบบ 1 FXO กล่าวคือการ์ดนี้สามารถรองรับการ เชื่อมต่อกับ โครงข่าย โทรศัพท์ได้จำนวน 1 พอร์ต หรือ 1 คู่สายโทรศัพท์เท่านั้น ดังนั้น การเชื่อมต่อกับโครงข่ายโทรศัพท์จึงต้องเลือกอย่าง ใดอย่างหนึ่ง ขึ้นอยู่กับข้อจำกัดของสถานที่ที่จะติดตั้งระบบ ตัวอย่างเช่น เมื่อนำระบบไปใช้งานในสถานที่ ที่ไม่มีโครงข่ายโทรศัพท์พื้นฐาน PSTN รองรับ ผู้ใช้ก็สามารถนำ Fixed Line Wireless Terminal ที่รองรับโครงข่ายโทรศัพท์เคลื่อนที่มาใช้แทนทดแทนได้เช่นกัน



รูปที่ 3.14 แสดงการออกแบบ ส่วนที่ทำหน้าที่ติดต่อสื่อสารกับระบบ โทรศัพท์ภายนอก

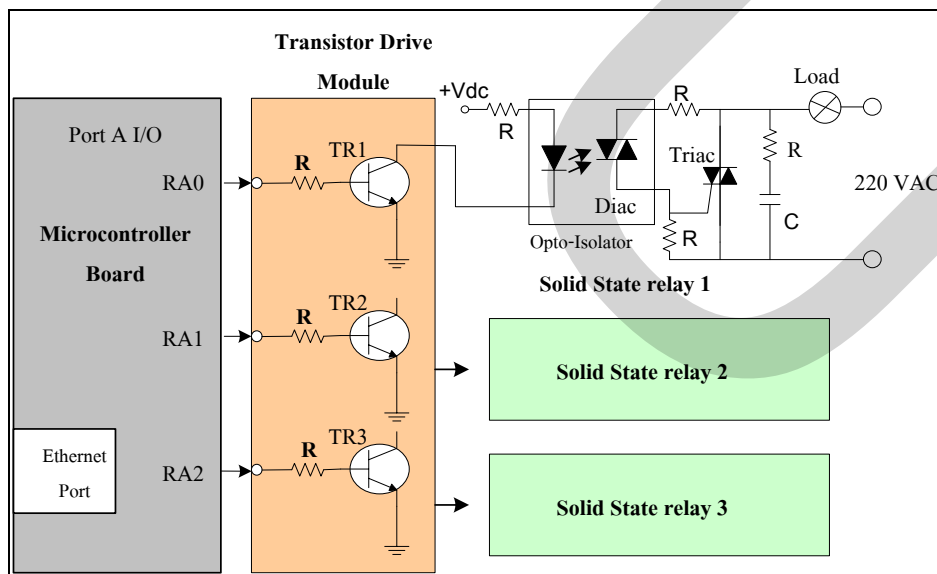
3.5.2 ส่วนรับสั่งและควบคุมอุปกรณ์ไฟฟ้า โดยที่การออกแบบทางด้านฮาร์ดแวร์นั้นส่วนรับคำสั่งควบคุมอุปกรณ์ไฟฟ้าแสดงได้ตามรูปที่ 3.15 โดยเป็นการนำเอา asterirk server บอร์ดไมโครคอนโทรลเลอร์เชื่อมต่อกับบอร์ดควบคุมอุปกรณ์ไฟฟ้าและจากที่ได้ศึกษาและเก็บข้อมูลคุณสมบัติของบอร์ดไมโครคอนโทรลเลอร์และบอร์ดควบคุมอุปกรณ์ไฟฟ้าพบว่าบอร์ดควบคุมอุปกรณ์ไฟฟ้านั้นทางด้านอินพุตถ้าต้องการสั่งให้ทำงานจะต้องใช้กระแสไฟฟ้าที่สูงแต่ถ้าจะนำเอาบอร์ดไมโครคอนโทรลเลอร์ จ่ายกระแสไฟฟ้าให้กับบอร์ดควบคุมอุปกรณ์ไฟฟ้าโดยตรง ซึ่ง ไม่สามารถทำได้ก็เพราะว่าบอร์ดไมโครคอนโทรลเลอร์นั้นสามารถจ่ายกระแสไฟฟ้าได้ในปริมาณที่ค่าต่ำๆทุกและ

ดังนั้นจำเป็นต้องหาตัวช่วยทำหน้าที่ขับกระแสไฟฟ้าให้สูงๆ แทนนั้นก็คือโดยการนำโมดูลทรานซิสเตอร์ขับกระแสไฟฟ้ามาช่วยเพื่อให้จ่ายกระแสไฟฟ้าได้เพียงพอที่จะนำไปสั่งบอร์ดควบคุมอุปกรณ์ไฟฟ้า อีกทีหนึ่ง



รูปที่ 3.15 แสดงการเชื่อมต่อ Asterisk Server บอร์ดไมโครคอนโทรลเลอร์ ชุดทรานซิสเตอร์ขับกระแสไฟฟ้าและบอร์ดควบคุมอุปกรณ์ไฟฟ้า

โมดูลทรานซิสเตอร์ขับกระแสไฟฟ้า (Transistor Drive Module) แสดงได้ตามรูปที่ 3.16 โดยใช้ทรานซิสเตอร์เบอร์ B548 ที่มีคุณสมบัติเป็น Amplifier transistor ชนิด NPN สามารถจ่ายกระแสไฟฟ้าตรงได้ 100 มิลลิแอมป์ (mA) ซึ่งเพียงพอในการนำมาใช้ขับกระแสไฟฟ้าให้กับชุดควบคุมอุปกรณ์ไฟฟ้าแทนบอร์ดไมโครคอนโทรลเลอร์

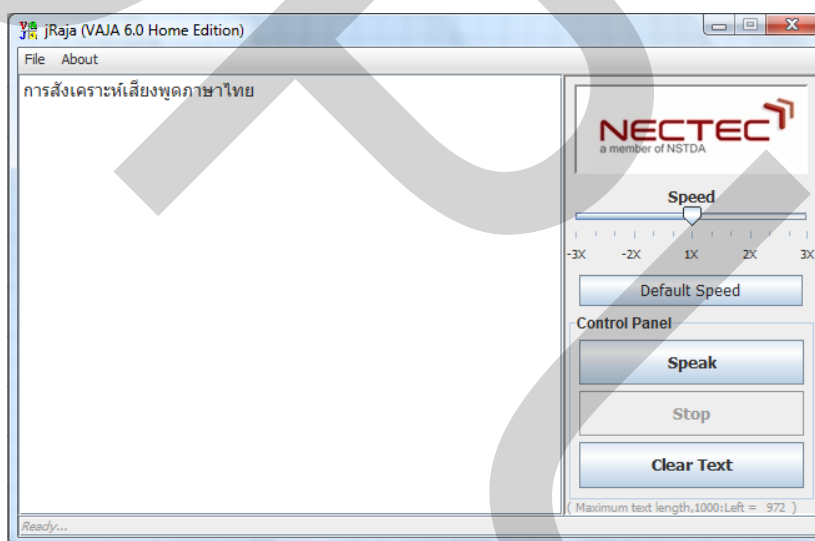


รูปที่ 3.16 แสดงบล็อกไดอะแกรม Transistor Drive Module

### 3.6 การออกแบบระบบ IVR

การออกแบบระบบ IVR หรือระบบตอบรับด้วยเสียงพูด โดยทั่วไปมีวิธีการสังเคราะห์เสียงพูดด้วยกันหลายวิธี เช่น วิธีการพูดใส่ไมโครโฟนแล้วทำการบันทึกเสียง วิธีนี้คุณภาพเสียงไม่ชัดเจนขึ้นอยู่กับคุณภาพของอุปกรณ์หรือเครื่องมือในการบันทึกเสียง อีกอย่างหนึ่งคือใช้เวลาในการแก้ไขมาก หากขณะบันทึกเสียงแล้วพูดผิด วิธีการสังเคราะห์เสียงพูดอีกวิธีหนึ่งซึ่งเป็นที่นิยมใช้ในปัจจุบันก็คือวิธีการสังเคราะห์เสียงพูดด้วยซอฟต์แวร์ มีลักษณะในการแปลงข้อความเป็นเสียงพูดซึ่งมีความสะดวกสบายไม่ยุ่งยาก คุณภาพเสียงก็ชัดเจน

การออกแบบระบบ IVR สำหรับงานวิจัยนี้ ใช้วิธีการสังเคราะห์เสียงพูดด้วยซอฟต์แวร์ หรือ Text to Speed ที่รู้จักกันทั่วไปก็คือ ซอฟต์แวร์สังเคราะห์เสียงพูดภาษาไทย “วาจา” (VAJA) ซึ่งเป็นงานวิจัยของ NECTEC หรือ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ รุ่นที่นำมาใช้งานเป็นรุ่น VAJA 6.0 Home Edition ดังแสดงตามรูปที่ 3.17



รูปที่ 3.17 แสดงซอฟต์แวร์สังเคราะห์เสียงพูดภาษาไทย “วาจา” (VAJA) ของ NECTEC

ไฟล์ที่ได้จากการแปลงข้อความเป็นเสียงพูดจะมีนามสกุล .wav ซึ่งโดยปกติแล้วการบันทึกเสียงต่าง ๆ ผ่านคอมพิวเตอร์จะบันทึกเป็น .wav และคุณภาพของไฟล์จะเป็นตามต้นฉบับทุกประการ คุณภาพเสียงดี แต่ว่ามีขนาดใหญ่ ดังนั้น Asterisk จึงใช้ไฟล์ .gsm แทน .wav ซึ่งขนาดของไฟล์จะเล็กลงมาก คุณภาพก็อยู่ในเกณฑ์ที่ยอมรับได้

เหตุผลที่ต้องทำการแปลงไฟล์ นามสกุล .wav เป็นไฟล์ นามสกุล .gsm ก็เพราะว่าการสื่อสารในโครงข่าย โทรศัพท์ นั้นต้องใช้ช่องสัญญาณที่มีขนาดเล็ก ซึ่งต้องการเปลี่ยนจากไฟล์

นามสกุล .wav ไปเป็นไฟล์นามสกุล .gsm นั้นสามารถทำได้ด้วยโปรแกรม sox ซึ่งในบางครั้งพบว่า Linux OS ยังไม่ได้ติดตั้งโปรแกรม sox ถ้าจะใช้โปรแกรม sox ในการแปลงไฟล์ นามสกุล จำเป็นจะต้องติดตั้งใน linux OS ก่อนจึงจะสามารถใช้คำสั่ง sox ในการเปลี่ยน ไฟล์นามสกุลได้และการติดตั้งโปรแกรม sox บน Linux Ubuntu ใช้คำสั่ง ดังนี้

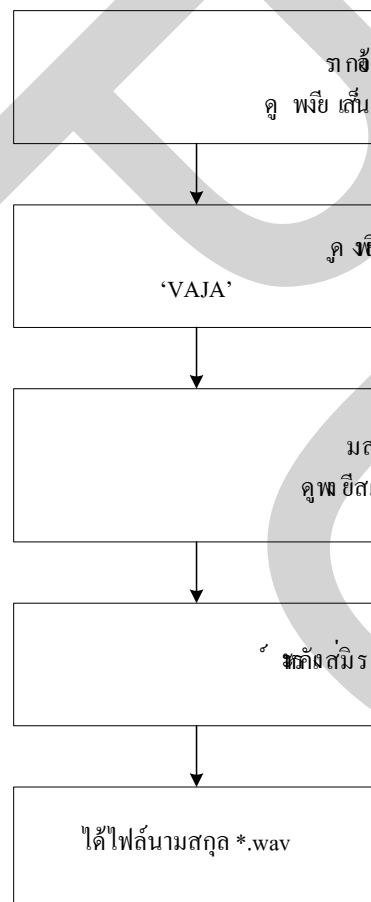
```
$sudo apt-get install sox
```

การเปลี่ยนไฟล์นามสกุล .wav ไปเป็นไฟล์นามสกุล .gsm มีรูปแบบคือ

```
sox <filename>.wav -r 8000 <filename>.gsm
```

ไฟล์เสียงที่ได้จะจัดเก็บเอาไว้ที่ Directory : /etc/asterisk/sounds/filename.gsm

วิธีการแปลงข้อความเป็นเสียงพูดด้วยซอฟต์แวร์สังเคราะห์เสียงพูดภาษาไทย ‘วาจา’ มีขั้นตอนการทำงานตามรูปที่ 3.18

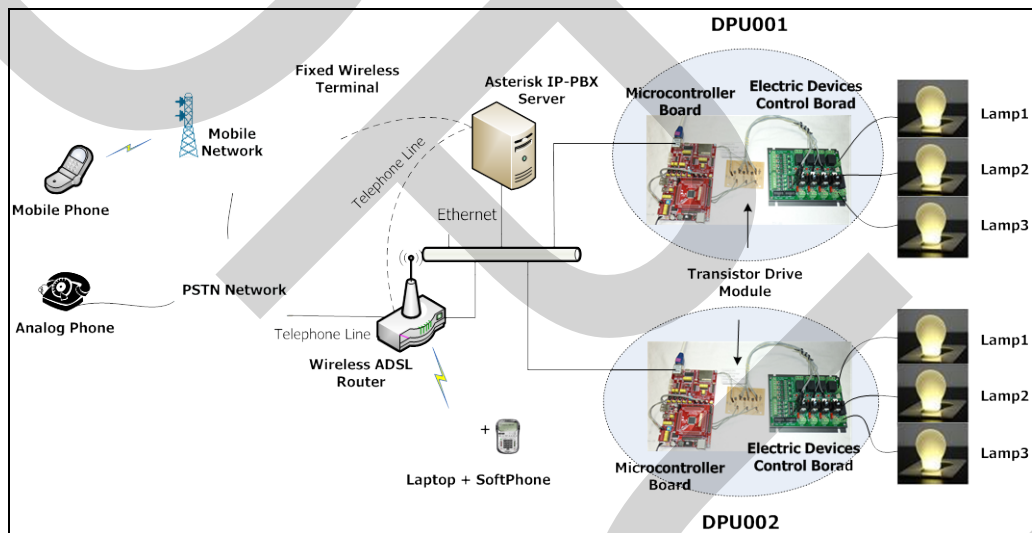


รูปที่ 3.18 แสดงขั้นตอนการแปลงข้อความเป็นเสียงพูด ด้วยซอฟต์แวร์สังเคราะห์เสียงพูดภาษาไทย “วาจา” (VAJA)

## บทที่ 4

### การทดสอบระบบ

เนื้อหาในบทนี้จะกล่าวถึงการทดสอบระบบ ที่ได้พัฒนาขึ้นพร้อมกับทำการประเมินผลการใช้งานและปรับปรุงข้อผิดพลาดที่เกิดขึ้นจากการทดสอบ โดยทำการจำลองสถานการณ์ขึ้นม ๑ และทดสอบการทำงาน ซึ่งโครงข่ายของระบบที่ใช้ในการทดสอบแสดงได้ตามรูปที่ 4.1



รูปที่ 4.1 แสดงอุปกรณ์และโครงข่ายที่ใช้ในการทดสอบ

จากรูปที่ 4.1 แสดงโครงข่ายที่ใช้ในการทดสอบซึ่งประกอบด้วยอุปกรณ์ต่างๆ ดังนี้

- 1) Fixed Wireless Terminal : รุ่น HR8100 (23) TG ทำหน้าที่เป็นGateway รับสัญญาณโทรศัพท์เคลื่อนที่ GSM ย่านความถี่ 1800 MHz
- 2) Wireless ADSL Router : DLINK DSL-2640T ทำหน้าที่เป็นGateway เชื่อมต่อกับระบบโทรศัพท์ พื้นฐาน PSTN และอินเทอร์เน็ต
- 3) Asterisk IP-PBX Server : เครื่องคอมพิวเตอร์ส่วนบุคคล Intel Pentium 1.6 GHz พร้อมติดตั้ง Linux Ubuntu 8.04 Asterisk IP-PBX Software และโปรแกรมสคลิปทั้งหมดนี้ทำหน้าที่บริหารจัดการแผนการโทรศัพท์

4) บอร์ดไมโครคอนโทรลเลอร์ : รุ่น ET\_PIC 16/32 Start Kit ทำหน้าที่ควบคุมการทำงานของบอร์ดควบคุมอุปกรณ์ไฟฟ้า

5) บอร์ดควบคุมอุปกรณ์ไฟฟ้า : โซลิต-สเตท รีเลย์ รุ่น MCT-02-8 ทำหน้าที่ควบคุมเปิด-ปิดอุปกรณ์ไฟฟ้า

6) เครื่อง Laptop พร้อมติดตั้ง Softphone X-Lite : ทำหน้าที่เป็นเครื่องโทรศัพท์ VoIP จากโครงข่ายของระบบรูปที่ 4.1 มีการทำงานเป็นดังนี้ ผู้ใช้สามารถโทรเข้ามาในระบบผ่านทางโครงข่าย โทรศัพท์ได้ 3 โครงข่ายด้วยกันคือ โครงข่าย โทรศัพท์พื้นฐาน PSTN โครงข่าย โทรศัพท์เคลื่อนที่และ โครงข่าย โทรศัพท์ VoIP ซึ่งมี Asterisk IP-PBX Server ทำหน้าที่บริหารจัดการกระบวนการโทรศัพท์ทั้งหมด และระบบได้ติดตั้ง การ์ด FXO เพิ่มเติมบน Server จึงทำให้ระบบสามารถเชื่อมต่อกับระบบโทรศัพท์ภายนอกได้ การควบคุมอุปกรณ์ไฟฟ้านั้นเป็นการทำงานร่วมกันของบอร์ดไมโครคอนโทรลเลอร์และบอร์ดควบคุมอุปกรณ์ไฟฟ้า ทั้งนี้ผู้ใช้สามารถเลือกกดสั่งงานจากแป้นโทรศัพท์ ด้วยระบบ DTMF เพื่อให้รู้ถึง สถานะการทำงานปัจจุบันของอุปกรณ์ไฟฟ้า พร้อมกับสามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้าได้ในคราวเดียวกัน

ในการทดสอบนี้ใช้บอร์ดควบคุมอุปกรณ์ไฟฟ้าที่รับภาระโหลดได้ประมาณ 3 แอมป์ และใช้หลอดไฟแสงสว่างที่กินกระแสไฟฟ้า 60 วัตต์ จำนวน 3 หลอด มาทำหน้าที่เป็น โหลดแทนเครื่องปรับอากาศ ที่มีข้อจำกัดเมื่อจะใช้เครื่องปรับอากาศเป็นโหลดขณะทดสอบ คือต้องใช้อุปกรณ์ควบคุมอุปกรณ์ไฟฟ้าที่สามารถจ่าย กระแสไฟฟ้าที่สูงๆ ประมาณ 30 แอมป์ ขึ้นไป ซึ่งมีราคาแพง และมีจำหน่ายทั่วไป ซึ่งสามารถนำมาใช้ทดแทน โซลิต-สเตท รีเลย์ที่ใช้ในงานวิจัยนี้ได้ทันที โดยไม่ต้องปรับปรุงสิ่งใดเพิ่มเติม

ระบบนี้สามารถแจ้งเหตุผิดปกติที่เกิดขึ้นกับอุปกรณ์ไฟฟ้าได้ โดยมี ชุดตรวจจับหรือ Analog to Digital อย่างเช่น ชุดตรวจจับอุณหภูมิ ชุดตรวจจับกระแสไฟฟ้ากรณีอุปกรณ์เสีย เป็นต้น เมื่อมีเหตุผิดปกติเกิดขึ้น ระบบจะทำการ โทรไปหาผู้ใช้ และแจ้งเตือนเป็นเสียง IVR บอกเหตุต่างๆ โดยระบบสามารถแจ้งเหตุผ่านทางระบบโทรศัพท์พื้นฐานและระบบโทรศัพท์เคลื่อนที่ได้ การทดสอบการแจ้งเหตุผิดปกตินี้ใช้วิธีการจำลองเหตุการณ์ ด้วยการกดสวิทช์แจ้งเหตุจากบอร์ดไมโครคอนโทรลเลอร์

การทดสอบระบบ สามารถแบ่งเป็นการทดสอบย่อยตามหัวข้อได้ ดังนี้

- 1) ทดสอบระบบที่สามารถควบคุมชุดควบคุมหรือสถานีได้หลายจุด
- 2) ทดสอบระบบที่สามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า
- 3) ทดสอบระบบที่สามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า

4) ทดสอบระบบที่สามารถแจ้งเหตุผิดปกติมายังผู้ใช้ ผ่านทางระบบโทรศัพท์ พื้นฐาน PSTN หรือระบบโทรศัพท์เคลื่อนที่

#### 4.1 ทดสอบระบบที่สามารถควบคุมชุดควบคุมหรือสถานีได้หลายจุด

การทดสอบในส่วน นี้เป็นการทดสอบว่า ระบบที่พัฒนา ขึ้นนี้สามารถควบคุมชุดควบคุมหรือสถานีได้หลายจุด โดยจะขึ้นอยู่กับการออกแบบระบบว่าต้องการควบคุมกี่จุด ในการทดสอบกำหนดให้ระบบควบคุมชุดควบคุมได้ 2 จุด ตามรูปที่ 4.1 โดยกำหนด ให้ชื่อของสถานีเป็น สถานี DPU001 และสถานี DPU002

ขั้นตอนการทดสอบระบบ ใช้ระบบโทรศัพท์สำหรับเป็น Gateway จำนวน 3 โครงข่ายด้วยกัน คือ เชื่อมต่อกับระบบ โทรศัพท์พื้นฐาน PSTN เชื่อมต่อกับระบบโทรศัพท์เคลื่อนที่ ผ่านทาง Fixed Wireless Terminal และเชื่อมต่อกับระบบโทรศัพท์ VoIP ทดสอบด้วยวิธีการ โทรศัพท์เข้ามาในระบบ หลังจากนั้นทำการกดเลือกสถานีสลับกันระหว่างสถานี DPU001 และสถานี DPU002 โดยทดสอบเลือกสถานีละ 20 ครั้งและบันทึกผล โดยผลการทดสอบเป็นไปตามตารางที่ 4.1

ตารางที่ 4.1 แสดงผลการทดสอบระบบ กรณีสามารถควบคุมชุดควบคุมหรือสถานีได้หลายจุด

โครงข่ายโทรศัพท์ที่ใช้ทดสอบ	สถานีที่เลือก				ความถูกต้องคิดเป็นเปอร์เซ็นต์ (%)
	DPU001		DPU002		
	สำเร็จ	ไม่สำเร็จ	สำเร็จ	ไม่สำเร็จ	
ระบบโทรศัพท์พื้นฐาน PSTN	20	0	20	0	100%
ระบบโทรศัพท์เคลื่อนที่	20	0	20	0	100%
ระบบโทรศัพท์ VoIP	20	0	20	0	100%

จากตารางที่ 4.1 แสดงผลการทดสอบระบบที่สามารถควบคุมชุดควบคุมหรือสถานีได้หลายจุด พบว่าผู้ใช้สามารถเลือกสถานีที่ต้องการได้อย่างถูกต้อง 100 % และพบว่าคุณภาพของเสียงกรณีโทรจากระบบโทรศัพท์พื้นฐานและจากระบบโทรศัพท์เคลื่อนที่ นั้นคุณภาพเสียงไม่ชัดเจนเท่าที่ควร แตกต่างจากการโทรจากระบบโทรศัพท์ VoIP ที่มีคุณภาพเสียงที่ดีกว่า



#### 4.2 ทดสอบระบบสามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า

การทดสอบในส่วนนี้เป็นการทดสอบว่า ระบบที่พัฒนาขึ้นนี้ สามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า ซึ่งต่อเนื่อง จากหัวข้อ 4.1 ระบบนี้ถูกออกแบบให้ควบคุมอุปกรณ์ไฟฟ้าได้จำนวน 3 อุปกรณ์ นั่นก็คือทำให้ตรวจสอบสถานะการทำงานได้ 3 อุปกรณ์ โดยกำหนดให้เป็นการตรวจสอบ สถานการณ์ ทำงานของเครื่องปรับอากาศเครื่องที่หนึ่ง ตรวจสอบสถานะการทำงานของเครื่องปรับอากาศเครื่องที่สอง และตรวจสอบสถานะการทำงานของไฟแสงสว่าง

วิธีการทดสอบทำได้โดยทำการเชื่อมต่อระบบเข้ากับโครงข่ายโทรศัพท์ 3 โครงข่าย คือโครงข่าย โทรศัพท์พื้นฐาน PSTN และโครงข่าย โทรศัพท์เคลื่อนที่ โดยผ่านทาง Fixed Wireless Terminal และเชื่อมต่อกับระบบโทรศัพท์ VoIP จากนั้น ทดสอบด้วยวิธีโทรเข้ามาในระบบคือเมื่อเลือกสถานที่ที่ต้องการตรวจสอบได้แล้ว จากนั้นกด เลือกอุปกรณ์ที่ต้องการตรวจสอบ จนครบทุกอุปกรณ์ๆ ละ 20 ครั้งและบันทึกผล โดยผลการทดสอบเป็นไปตามตารางที่ 4.2

ตารางที่ 4.2 แสดงผลการทดสอบระบบ กรณีตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า

โครงข่าย โทรศัพท์ที่ใช้ ทดสอบ	แสดงสถานะของไฟ แสงสว่าง Lamp 1				แสดงสถานะของไฟ สว่าง Lamp 2				แสดงสถานะของไฟ แสงสว่าง Lamp 3				ความ ถูกต้องคิด เป็น เปอร์เซ็นต์ (%)
	เปิด		ปิด		เปิด		ปิด		เปิด		ปิด		
	ถูก	ผิด	ถูก	ผิด	ถูก	ผิด	ถูก	ผิด	ถูก	ผิด	ถูก	ผิด	
ระบบโทรศัพท์ พื้นฐาน PSTN	20	0	20	0	20	0	20	0	20	0	20	0	100%
ระบบโทรศัพท์ เคลื่อนที่	20	0	20	0	20	0	20	0	20	0	20	0	100%
ระบบโทรศัพท์ VoIP	20	0	20	0	20	0	20	0	20	0	20	0	100%

จากตารางที่ 4.2 แสดงผลการทดสอบระบบกรณี ตรวจสอบ สถานะการทำงาน ของอุปกรณ์ไฟฟ้า พบว่าระบบสามารถบอกสถานะการทำงานผ่านทาง IVR ที่เป็นปัจจุบันของทุก ไฟแสงสว่างได้อย่างถูกต้อง 100% และพบว่าคุณภาพของเสียงกรณีโทรจากระบบโทรศัพท์พื้นฐาน

และจากระบบโทรศัพท์เคลื่อนที่ นั้น คุณภาพเสียงไม่ชัดเจนเท่าที่ควร ซึ่งแตกต่างจากการโทรจากระบบโทรศัพท์ VoIP ที่มีคุณภาพเสียงที่ดีกว่า

#### 4.3 ทดสอบระบบสามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า

การทดสอบในส่วนนี้เป็นการทดสอบ ระบบที่พัฒนาขึ้น สามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า โดยวิธีการทดสอบเป็นการทดสอบต่อเนื่องจากหัวข้อ 4.2 ซึ่งหลังจากที่ระบบสามารถแสดงสถานะการทำงานได้แล้ว ต่อจากนั้นก็ทดสอบด้วยการสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า วิธีการทดสอบก็คือ หลังจาก IVR บอกว่าอุปกรณ์นั้นกำลังทำงานอยู่ในสถานะใด ต่อจากนั้นก็ทดสอบด้วยการกด '1' หากต้องการเปิดและกด '0' หากต้องการปิด โดยทำการเปิด-ปิดเป็นจำนวน 20 ครั้งต่ออุปกรณ์และบันทึกผล ผลการทดสอบเป็นไปตามตารางที่ 4.3

ตารางที่ 4.3 แสดงผลการทดสอบระบบ กรณีสามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า

โครงข่าย โทรศัพท์ที่ใช้ ทดสอบ	ไฟแสงสว่าง Lamp 1		ไฟแสงสว่าง Lamp 2		ไฟแสงสว่าง Lamp 3		ความถูกต้อง คิดเป็น เปอร์เซ็นต์						
	เปิด->ปิด		ปิด->เปิด		เปิด->ปิด			ปิด->เปิด					
	ได้	ไม่ได้	ได้	ไม่ได้	ได้	ไม่ได้		ได้	ไม่ได้				
ระบบโทรศัพท์ พื้นฐาน PSTN	20	0	20	0	20	0	20	0	20	0	20	0	100%
ระบบโทรศัพท์ เคลื่อนที่	20	0	20	0	20	0	20	0	20	0	20	0	100%
ระบบโทรศัพท์ VoIP	20	0	20	0	20	0	20	0	20	0	20	0	100%

จากตารางที่ 4.3 แสดงผลการทดสอบระบบกรณี สามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า พบว่าระบบสามารถสั่งเปิด-ปิดไฟแสงสว่าง Lamp1 – Lamp3 ได้ทั้ง 3 หลอดและทำได้ถูกต้อง 100% และทั้งนี้ พบว่าคุณภาพของเสียงกรณีโทรจากระบบโทรศัพท์พื้นฐานและจากระบบโทรศัพท์เคลื่อนที่ นั้นคุณภาพเสียงไม่ชัดเจนเท่าที่ควร ซึ่งแตกต่างจากการโทรจากระบบโทรศัพท์ VoIP ที่มีคุณภาพเสียงที่ดีกว่า

#### 4.4 ทดสอบระบบสามารถแจ้งเหตุผิดปกติมายังผู้ใช้งานทางระบบโทรศัพท์พื้นฐาน PSTN หรือระบบโทรศัพท์เคลื่อนที่

การทดสอบในส่วนนี้เป็นการทดสอบระบบที่พัฒนาขึ้น สามารถแจ้งเหตุผิดปกติมายังผู้ใช้งานทางระบบโทรศัพท์พื้นฐาน PSTN หรือระบบโทรศัพท์เคลื่อนที่ วิธีการทดสอบโดยการจำลองเหตุการณ์แจ้งเหตุผิดปกติด้วยการกดสวิทช์แจ้งเหตุบนบอร์ดไมโครคอนโทรลเลอร์ โดยกดสวิทช์แจ้งเหตุโครงข่ายละ 20 ครั้ง และบันทึกผลโดยผลการทดสอบเป็นไปตามตารางที่ 4.4

ตารางที่ 4.4 แสดงผลการทดสอบระบบ กรณิแจ้งเหตุผิดปกติไปยังผู้ใช้งานทางระบบโทรศัพท์พื้นฐาน PSTN หรือระบบโทรศัพท์เคลื่อนที่

แจ้งเหตุผิดปกติ	รับแจ้งเหตุด้วยระบบโทรศัพท์				ความถูกต้อง คิดเป็น เปอร์เซ็นต์
	ระบบโทรศัพท์พื้นฐาน		ระบบโทรศัพท์เคลื่อนที่		
	สำเร็จ	ไม่สำเร็จ	สำเร็จ	ไม่สำเร็จ	
จำลองการแจ้งเหตุ ผิดปกติ	20	0	20	0	100%

จากตารางที่ 4.4 แสดงผลการทดสอบระบบกรณิแจ้งเหตุผิดปกติไปยังผู้ใช้งานทางระบบโทรศัพท์พื้นฐาน PSTN หรือระบบโทรศัพท์เคลื่อนที่ สามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้าพบว่า ระบบสามารถแจ้งเหตุผิดปกติผ่านทางระบบโทรศัพท์ทั้ง 2 โครงข่ายได้อย่างถูกต้อง 100%

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

ในบทนี้จะเป็นการอภิปรายเพื่อสรุปผลที่ได้จากการทดสอบงานวิจัย รวมทั้งข้อจำกัดของระบบที่พบจากการทดสอบระบบ และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนางานวิจัยนี้ต่อไป เพื่อแก้ข้อบกพร่องของระบบให้มีประสิทธิภาพมากยิ่งขึ้น

#### 5.1 สรุปผลการวิจัย

##### 5.1.1 สรุปผลตามวัตถุประสงค์ของงานวิจัย

1) ในการพัฒนาระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP นั้นตามวัตถุประสงค์ของงานวิจัย คือสามารถศึกษา ออกแบบ และพัฒนาระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP โดยการประยุกต์ใช้ซอฟต์แวร์ Asterisk IP-PBX บอร์ดไมโครคอนโทรลเลอร์ และบอร์ดควบคุมอุปกรณ์ไฟฟ้า ได้อย่างเหมาะสมและมีประสิทธิภาพ

2) ระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP สามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า สั่งปิด -เปิดอุปกรณ์ ไฟฟ้าและ สามารถแจ้ง เตือนเหตุผิดปกติไปยังผู้ใช้งานอย่างมีระบบ

3) การจำลองสถานการณ์สำหรับทดสอบระบบที่พัฒนาขึ้นมาี้ สามารถพิสูจน์ว่าระบบสามารถใช้งานได้จริง

5.1.2 สรุปผลตามขอบเขตของงานวิจัย ซึ่งให้ความสำคัญในเรื่องของการอำนวยความสะดวกการประหยัดงบประมาณ ในการออกแบบระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP จากการทดสอบการทำงานต่างๆ ตามขอบเขตของระบบ สามารถสรุปผลได้ดังนี้

1) ผู้ใช้สามารถตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าได้อย่างสมบูรณ์

2) ระบบสามารถควบคุมชุดควบคุมหรือสถานีได้หลายจุด

3) ผู้ใช้สามารถสั่งเปิด-ปิดอุปกรณ์ไฟฟ้าได้อย่างถูกต้อง

4) กรณีมีเหตุผิดปกติเกิดขึ้น (อุณหภูมิสูงเกินปกติและอุปกรณ์ไฟฟ้าเสีย) ระบบสามารถแจ้งเหตุไปยังผู้ใช้ผ่านทางระบบโทรศัพท์พื้นฐาน PSTN และระบบโทรศัพท์เคลื่อนที่ก็ได้

5.1.3 ภายหลังกการพัฒนาระบบได้มีการทดสอบการใช้งานระบบ โดยทำการจำลองเหตุการณ์และโทรเข้ามาทดสอบด้วยระบบโทรศัพท์ 3 โครงข่ายด้วยกัน คือโครงข่ายโทรศัพท์พื้นฐาน PSTN

โครงข่ายโทรศัพท์เคลื่อนที่และ โครงข่าย โทรศัพท์ VoIP ซึ่งทำการ โทรเข้ามาจากแต่ละโครงข่าย จำนวน 20 ครั้ง กดสั่งงานจากเป็น โทรศัพท์สำหรับตรวจสอบสถานะการทำงานของแต่ละอุปกรณ์ เป็นจำนวน 20 ครั้ง ตั้งเปิด-ปิดอุปกรณ์ไฟฟ้าแต่ละอุปกรณ์จำนวน 20 ครั้ง จำลองเหตุการณ์แจ้งเหตุ ผิดปกติจำนวน 20 ครั้ง พร้อมกับบันทึกผลการทดสอบ

ผลปรากฏว่าระบบสามารถทำงานตามขอบเขตของงานวิจัยได้อย่างถูกต้องโดยไม่มี ข้อผิดพลาดใดๆ (ความถูกต้องคิดเป็นเปอร์เซ็นต์ 100 %)

## 5.2 ข้อจำกัดของระบบ

ข้อจำกัดของระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP มี ดังนี้

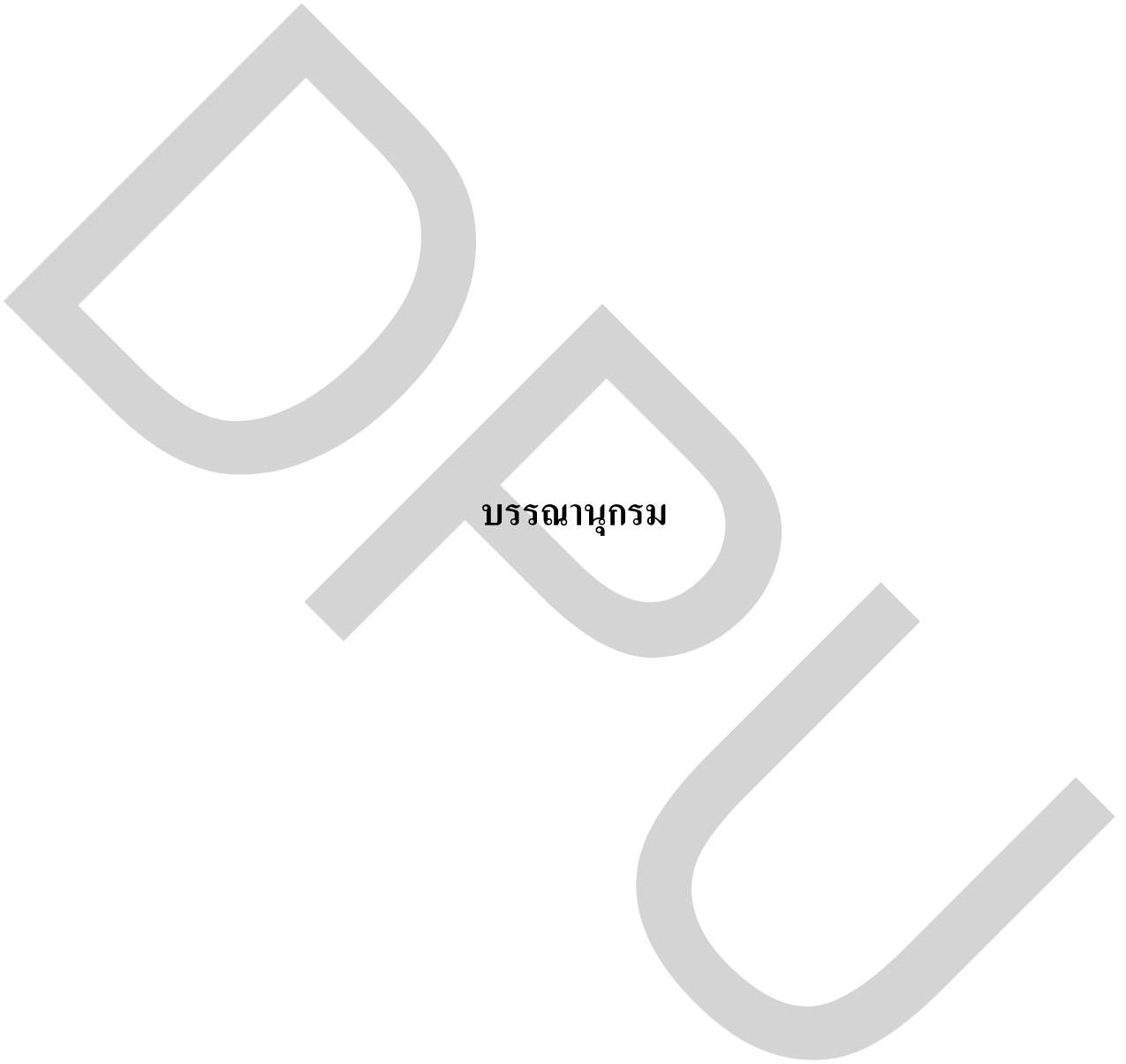
- 5.2.1 การเพิ่มชุดควบคุมหรือเพิ่มสถานีทำได้โดยผู้ดูแลระบบเท่านั้น ทำให้ขาดความสะดวก
- 5.2.2 การแก้ไขในส่วนของ Dial Plan หรือที่เรียกว่าแผนการ โทรศัพท์ นั้นต้องกระทำโดยผู้ดูแลระบบเท่านั้น
- 5.2.3 ระบบที่พัฒนานี้ไม่สามารถตรวจสอบความผิดปกติที่เกิดขึ้นกับอุปกรณ์ไฟฟ้าได้ ซึ่งรองรับเพียงแต่การจำลองเหตุผิดปกติที่บอร์ดไมโครคอนโทรลเลอร์เท่านั้น

## 5.3 ข้อเสนอแนะ

งานวิจัยนี้นำเสนอระบบแจ้งเตือนและควบคุมอุปกรณ์ไฟฟ้าด้วย IVR ผ่านระบบ VoIP ซึ่งเป็นการนำ ซอฟต์แวร์ Asterisk IP-PBX บอร์ดไมโครคอนโทรลเลอร์ บอร์ดควบคุมอุปกรณ์ไฟฟ้า มาเชื่อมต่อติดต่อสื่อสารกันเป็นระบบ ซึ่งจะต้องมีความรู้ด้านไมโครคอนโทรลเลอร์ WebServer เป็นอย่างมาก ข้อสำคัญต้องรู้และเข้าใจระบบโทรศัพท์ VoIP เป็นอย่างดี

ข้อเสนอแนะที่ต้องทำเพิ่มเติมต่อไปก็คือ

- 5.3.1 การทำเป็นเว็บเพจ ทำให้ผู้ใช้สามารถเข้าถึงระบบได้ง่ายขึ้น เป็นผลให้การบริหารจัดการ ก็ไม่ยุ่งยากและที่สำคัญง่ายต่อการเพิ่มจำนวนชุดควบคุมหรือเพิ่มสถานี อีกทั้งทำให้ง่ายต่อการเขียน แผนการ โทรศัพท์หรือ Dial Plan
- 5.3.2 ออกแบบเพิ่มเติมชุดแจ้งเหตุให้สามารถตรวจจับได้ว่าอุปกรณ์ตัวใดอยู่ในภาวะ Stand by หรือเตรียมพร้อมที่จะทำงาน
- 5.3.3 เพิ่มเติมชุดตรวจจับเหตุผิดปกติหรือ Analog to Digital เพื่อให้ระบบทำงานได้อย่างสม -บูรณ์ยิ่งขึ้น



**บรรณานุกรม**

## บรรณานุกรม

ภาษาไทย

หนังสือ

กิตติพงษ์ สุวรรณราช. (2551). การออกแบบและติดตั้งระบบโทรศัพท์ IP-PBX ด้วย Asterisk. กรุงเทพฯ: ออฟเซ็ท เพรส.

### สารสนเทศจากสื่ออิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ ภาควิชาอิเล็กทรอนิกส์และโทรคมนาคม มหาวิทยาลัยเทคโนโลยี

พระจอมเกล้าธนบุรีพระจอมเกล้าธนบุรี. (2554). The Management System of Electric Appliances via Internet. สืบค้นเมื่อ 20 มกราคม 2554, จาก

<http://www.st.kmutt.ac.th/~s0212445/index.htm>

คารม กร้าเพียร. (2554). ไมโครคอนโทรลเลอร์. สืบค้นเมื่อ 15 มกราคม 2554, จาก

<http://www.vcharkarn.com/vblog/35955>

คณิน นิติวงศ์, นาคกร ชูระเจน. (2554). Internet and Voice over IP. สืบค้นเมื่อ 20 มกราคม 2554,

จาก [http://www.student.chula.ac.th/~49702413/for\\_other.htm](http://www.student.chula.ac.th/~49702413/for_other.htm)

ทีมงาน ไทยแอมป์. (2554). เครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่านระบบโทรศัพท์.

สืบค้นเมื่อ 20 มกราคม 2554, จาก

<http://www.thaiamp.com>

ปรเมศวร์ กุमारบุญ. (2553). รู้จัก VOIP เทคโนโลยีที่จะมาแทนระบบโทรศัพท์แบบเดิม.

สืบค้นเมื่อ 15 ธันวาคม 2553, จาก

<http://forum.asteriskthailand.com/index.php?topic=45.0>

ประจักษ์ ปุณยวัฒน์พรกุล. (2554). Asterisk คืออะไร. สืบค้นเมื่อ 25 มกราคม 2554, จาก

<http://www.thaiasteriskclub.com/index.php?topic=5.0>

รังสิมา เกียรติยุทธชาติ, สมิตธิชัย ไชยวงศ์. (2554). เทคโนโลยี VoIP. Standard of VoIP

Technology. สืบค้นเมื่อ 20 มกราคม 2554, จาก

<http://www.vcharkarn.com/varticle/17875#P3>

เวปไซด์ ไมโครตรอน. (2553). การควบคุมอุปกรณ์ไฟฟ้า ผ่านสาย AC-Line ด้วยมาตรฐาน X10. สืบค้นเมื่อ 20 ธันวาคม 2553, จาก

<http://www.thaimicrotron.com/X10/X10-MainPage.htm>

วิกิพีเดีย สารานุกรมเสรี. (2554). ไมโครคอนโทรลเลอร์. สืบค้นเมื่อ 15 มกราคม 2554, จาก

<http://th.wikipedia.org/wiki/>

เอ-เทค โสคูซัน. (2554). บ้านอัจฉริยะ X10 <HomeAutomation>. สืบค้นเมื่อ 20 มกราคม 2554, จาก

[http://www.smartihome.com/atech/product/x10/home\\_x10.htm](http://www.smartihome.com/atech/product/x10/home_x10.htm)

ภาษาต่างประเทศ

## BOOKS

Jim Van Meggelen, Leif Madsen & Jred Smith. (2007). **Asterisk : The future of Telephony** (2<sup>nd</sup> Edition). United States of America: O'REILLY.

Nir Simionovich. (2009). **Asterisk Gateway Interface 1.4 and 1.6 Programming**. PACKT Publishing

## ELECTRONIC SOURCES

Cisco. (2010). Waveform Coding Techniques. Retrieved December 20 2010, from

[http://www.cisco.com/en/US/tech/tk1077/technologies\\_tech\\_note09186a00801149b3.shtml](http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a00801149b3.shtml)

SPsilicon Press. (2010). IP PBX Technology Brief . Retrieved December 25 2010, from

<http://www.silicon-press.com/briefs/brief.ippbx/index.html>

Webside bbee. (2010). Python. Retrieved December 10 2010, from

<http://bbee.exteen.com/20080313/python>

Wikipedia. (2010). X10 (industry standard). Retrieved January 20 2011, from

[http://en.wikipedia.org/wiki/X10\\_\(industry\\_standard\)](http://en.wikipedia.org/wiki/X10_(industry_standard))



## ประวัติผู้เขียน

ชื่อ-นามสกุล

ประวัติการศึกษา

ตำแหน่งและสถานที่ทำงานปัจจุบัน

คูสิต วัฒนเสย

ปริญญาตรีวิศวกรรมศาสตร์

สาขาวิศวกรรมโทรคมนาคม

มหาวิทยาลัยเทคโนโลยีมหานคร

สำเร็จการศึกษาเมื่อวันที่ 18 เมษายน พ.ศ. 2540

วิศวกร 8

บริษัท กสท โทรคมนาคม จำกัด (มหาชน)

ฝ่ายบริหารทรัพยากรร่วมการงาน

ส่วนบริหารทรัพยากรและเทคนิค โทร มูฟ