



## รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต  
สำหรับการสื่อสารการเดินอากาศ  
VoIP Monitoring System for Aeronautical Communication

นางสาวณัฏฐา วุฒิประสิทธิ์ผล

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการ ระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสาร  
การเดินทางอากาศ

ชื่อ-สกุล นักศึกษา นางสาวณัฐฐา วุฒิประสิทธิผล

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมโทรคมนาคม

ชื่อ-สกุล อาจารย์นิเทศ ผศ.ดร. สถาพร พรหมวงศ์

ชื่อ-สกุล ผู้นิเทศงาน นายสุวัฒน์ ศรีทองคำ และ นายปัฐยาวัตร คงประดิษฐ์

สถานประกอบการ บริษัท วิทย์การบินแห่งประเทศไทย จำกัด

## บทคัดย่อ

กิจการการเดินทางอากาศ การสื่อสารระหว่างเจ้าหน้าที่ควบคุมจราจรทางอากาศ และนักบิน เป็นสิ่งสำคัญที่จะต้องดำเนินการด้วยความปลอดภัย และสามารถตรวจสอบได้ ซึ่งในปัจจุบันระบบการสื่อสารการเดินทางอากาศและการสังเกตการณ์การสื่อสารการเดินทางอากาศของบริษัท วิทย์การบินแห่งประเทศไทย จำกัด ใช้ระบบการสื่อสารแบบแอนะล็อก ภายหลังได้เห็นถึงความสำคัญของการนำการสื่อสารแบบ VoIP มาใช้เพื่อตอบสนองความต้องการที่เพิ่มขึ้นจากระบบแอนะล็อก อย่างไรก็ตามยังขาดอุปกรณ์สำหรับสังเกตการณ์การสื่อสารแบบ VoIP

โครงการนี้จึงมีวัตถุประสงค์ในการสร้างและออกแบบโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินทางอากาศ โดยใช้ภาษาไพทอน โปรแกรมนี้จะช่วยในการสังเกตการณ์การสื่อสารในด้านของสถานะการเชื่อมต่อ การรับ-ส่งข้อมูลเสียงของวิทย์ที่มี IP Address ที่ต้องการ รวมทั้งสังเกตเสียงที่ผ่านระบบ VoIP จากการบันทึกเสียงผ่านโปรแกรมที่จัดทำขึ้นสามารถใช้งานผ่าน Graphical User Interface (GUI) ซึ่งสะดวกต่อการใช้งาน และไม่จำเป็นต้องใช้อุปกรณ์จำนวนมาก อีกทั้งยังลดค่าใช้จ่ายของอุปกรณ์ต่าง ๆ ลงไปได้ โดยทั้งนี้โปรแกรมหวังว่ายังคงต้องมีการพัฒนาไปเรื่อย ๆ เพื่อให้การใช้งานมีประสิทธิภาพ และเสถียรภาพยิ่งขึ้นไป

คำสำคัญ : Voice over IP, Python, GUI, ระบบสังเกตการณ์, การสื่อสารการเดินทางอากาศ

**Cooperative Title:** VoIP Monitoring System for Aeronautical Communication

**Student intern name:** Miss. Nattha vudhiprasittipol

**Faculty:** Engineering **Department:** Telecommunication Engineering

**ชื่อ-สกุล อาจารย์นิเทศ** Asst. Prof. Dr. Sathaporn Promwong

**ชื่อ-สกุล ผู้นิเทศงาน** Mr. Suwat Srithongkom and Mr. Piyawat Kongpradit

**สถานประกอบการ** Aeronautical Radio of Thailand Ltd.

## ABSTRACT

In the aeronautical communication, it is important that a communication between air traffic controller (ATC) and pilot must be safe and its evidence must be available for any necessary investigation. AEROTHAI has been using only analog communication system. Nowadays Voice over IP (VoIP) communication has an important role in meeting increasing demand of channels besides the analog system. However, there is still a lack of equipment for monitoring VoIP communications.

This project aims to create and develop the VoIP monitoring system for aeronautical communication program by using Python. This program can be applied for monitoring radio connection status, record and play audio through the created program which can be used via Graphical User Interface (GUI). It easy to use and do not need many devices and reduces the cost of devices. However, the created program still needs to be developed continuously to make more effective and stability.

**Keywords :** Voice over IP, Python, GUI, Monitoring, Aeronautical communication

## กิตติกรรมประกาศ

โครงการ “ระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ” ต้องขอขอบคุณเป็นอย่างสูงที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง คณะวิศวกรรมศาสตร์ที่ได้กรุณาจัดโครงการสหกิจศึกษา ซึ่งโครงการนี้ไม่อาจสำเร็จลุล่วงได้ หากไม่ได้รับการสนับสนุนจากผู้สนับสนุนหลายฝ่าย ดังนี้

ขอขอบคุณ ผศ.ดร. สถาพร พรหมวงศ์ ที่ได้ให้เกียรติเป็นอาจารย์ที่ปรึกษา คอยให้คำปรึกษา สำหรับปัญหาต่าง ๆ และชี้แนะแนวทางตลอดการดำเนินโครงการนี้

ขอขอบคุณ บริษัท วิद्यุการบินแห่งประเทศไทย จำกัด ให้นักศึกษาได้มีโอกาสดำเนินการฝึกงานที่บริษัทฯ ขอขอบคุณ คุณสุวัฒน์ ศรีทองคำ และคุณปัญญาวัตร คงประดิษฐ์ ที่ได้ให้เกียรติเป็นพนักงานที่ปรึกษา ที่คอยดูแล มอบความรู้ ให้คำปรึกษา รวมทั้งพี่ ๆ พนักงานภายในบริษัทฯ ที่อำนวยความสะดวกในเรื่องการขอใช้อุปกรณ์ และสถานที่ที่ใช้ในการดำเนินงาน ตลอดโครงการสหกิจศึกษา

ขอขอบคุณ พี่ ๆ และเพื่อน ๆ ที่คอยให้ความช่วยเหลือ และคำปรึกษาในเรื่องที่เป็นประโยชน์ ต่อผู้จัดทำ

สุดท้ายนี้ขอขอบพระคุณ บิดา มารดา และญาติพี่น้อง ที่คอยให้การสนับสนุน และเป็นกำลังใจในการทำโครงการสหกิจศึกษา

นางสาวณัฐฐา วุฒิประสิทธิ์ผล  
ผู้จัดทำ

## สารบัญ

		หน้า
	บทคัดย่อ	I
	ABSTRACT	II
	กิตติกรรมประกาศ	III
	สารบัญ	IV
	สารบัญรูป	VI
	สารบัญตาราง	IX
บทที่ 1	บทนำ	1
	1.1 ความเป็นมาและความสำคัญ	1
	1.2 วัตถุประสงค์ของโครงการ	1
	1.3 ขอบเขตของโครงการ	2
	1.4 วิธีการดำเนินงานโครงการ	3
	1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
	1.6 ขั้นตอนของการศึกษา	4
บทที่ 2	ทฤษฎีและหลักการที่เกี่ยวข้อง	6
	2.1 ภาษาไพทอน	6
	2.2 เสียง	7
	2.3 หลักการส่งข่าวสารข้อมูลทางคลื่นวิทยุแบบการมอดูเลตทางขนาด	7
	2.4 การมอดูเลตรหัสพัลส์	10
	2.5 Voice over IP (VoIP)	11
	2.6 Air Traffic Management (ATM)	15
	2.7 ED 137 หรือ Interoperability Standard for VoIP ATM Components Part1 Radio	18
	2.8 โปรแกรม Sniffer	30
	2.9 โปรแกรม Wireshark	32
	2.10 Port mirroring	33
	2.11 มาตรฐานเสียง G.711	36
	2.12 ฟิลเตอร์ C-message weighting	37
	2.13 สรุป	39
บทที่ 3	การออกแบบระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่าย อินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ	40
	3.1 การออกแบบ	40
	3.2 เครื่องมือที่ใช้ในการทดลอง	44
	3.3 การจัดเก็บผลการทดลอง	49
	3.4 สรุป	69

## สารบัญ (ต่อ)

	หน้า
<b>บทที่ 4</b>	<b>ผลการทดลอง</b>
4.1 ผลการทดลองการสังเกตแพ็กเก็ตที่อ่านได้จากโปรแกรม Wireshark	70
4.2 ผลการทดลองเขียนโปรแกรมในการดักจับแพ็กเก็ต VoIP ด้วยภาษาไพทอน	73
4.3 ผลการทดลองเขียนโปรแกรมการดักจับแพ็กเก็ต VoIP ที่ได้จากวิทย์ IP Address ที่ต้องการ	74
4.4 ผลการทดลองตรวจสอบสถานะการเชื่อมต่อ และการรับ/ส่งข้อมูลของวิทย์	75
4.5 ผลการทดลองการบันทึกเสียงที่ได้จากแพ็กเก็ต VoIP ของวิทย์	78
4.6 ผลการทดลองเล่นไฟล์เสียง	79
4.7 ผลการทดลองกรองสัญญาณสัญญาณรบกวนออกจากไฟล์เสียง	79
4.8 ผลการทดลองเปรียบเทียบคุณภาพเสียงที่ได้จากการบันทึกของโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ กับเสียงที่ได้จากระบบแอนะล็อกเดิม	80
4.9 ผลการทดลองใช้งานโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ ร่วมกับ GUI	82
4.10 สรุป	84
<b>บทที่ 5</b>	<b>สรุปผลและข้อเสนอแนะ</b>
5.1 สรุปผล	85
5.2 ประโยชน์ที่ได้จากโครงงาน	86
5.3 ปัญหาและอุปสรรค	86
5.4 แนวทางการพัฒนาต่อ	86
<b>บรรณานุกรม</b>	<b>87</b>
<b>ภาคผนวก ก</b>	<b>คำสั่งการทำงานของระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ</b>
	<b>89</b>

## สารบัญรูป

รูปที่		หน้า
รูปที่ 1.1	บล็อกไดอะแกรมระบบการสื่อสารการเดินอากาศแบบแอนะล็อก	2
รูปที่ 1.2	บล็อกไดอะแกรมระบบการสื่อสารการเดินอากาศแบบ VoIP	2
รูปที่ 1.3	บล็อกไดอะแกรมของระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ	4
รูปที่ 2.1	ตราสัญลักษณ์ของโปรแกรมภาษาไพทอน	7
รูปที่ 2.2	หลักการส่งข่าวสารข้อมูลทางคลื่นวิทยุ	7
รูปที่ 2.3	การมอดูเลตแอมพลิจูด	8
รูปที่ 2.4	การวัดระดับความแรงของคลื่น (AM)	9
รูปที่ 2.5	การมอดูเลตรหัสพัลส์	11
รูปที่ 2.6	การแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล	13
รูปที่ 2.7	การนำสัญญาณสะท้อนกลับออก	13
รูปที่ 2.8	การจัดของมูลในรูปของเฟรม	13
รูปที่ 2.9	การแปลงเฟรมให้อยู่ในรูปแพ็กเก็ต	14
รูปที่ 2.10	การใส่ค่า IP Address	14
รูปที่ 2.11	การแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก	14
รูปที่ 2.12	การเชื่อมต่อ SIP ระหว่าง VCS และวิทยุเครื่องรับ/ส่ง (1 เครื่อง)	20
รูปที่ 2.13	RTP header	22
รูปที่ 2.14	RTP header extension	22
รูปที่ 2.15	Topology ของระบบวิทยุพื้นฐาน	23
รูปที่ 2.16	ฟิลด์ข้อมูลของ RTPTx	24
รูปที่ 2.17	การแบ่งส่วน Extension for additional features	26
รูปที่ 2.18	ฟิลด์ข้อมูลของ RTPRx	26
รูปที่ 2.19	โครงสร้างของ Keep alive message	29
รูปที่ 2.20	ตัวอย่างการใช้โปรแกรม Sniffer ในโครงข่าย	30
รูปที่ 2.21	ตราสัญลักษณ์โปรแกรม Wireshark	32
รูปที่ 2.22	การส่งแพ็กเก็ตของสวิตช์ปกติ กับสวิตช์ที่มีการทำ Port mirroring	33
รูปที่ 2.23	การวิ่งของแพ็กเก็ตของสวิตช์ปกติ กับสวิตช์ที่มี Port mirroring รูปที่ 1	34
รูปที่ 2.24	การวิ่งของแพ็กเก็ตของสวิตช์ปกติ กับสวิตช์ที่มี Port mirroring รูปที่ 2	35
รูปที่ 2.25	การวิ่งของแพ็กเก็ตของสวิตช์ปกติ กับสวิตช์ที่มี Port mirroring รูปที่ 3	35
รูปที่ 2.26	รูปร่างของฟิลด์เตอร์ CCITT และฟิลด์เตอร์ C-message weighting	38

## สารบัญรูป (ต่อ)

รูปที่		หน้า
รูปที่ 3.1	ผังงานการทำงานของโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่าน โครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ	42
รูปที่ 3.2	หน้าต่าง Record-Play tab และ Status tab	44
รูปที่ 3.3	วิทยุ Park Air Sapphire T6 (ด้านหน้า)	45
รูปที่ 3.4	วิทยุ Park Air Sapphire T6 (ด้านหลัง)	45
รูปที่ 3.5	สวิตช์ HPE OfficeConnect 1920s Series switch JL381	46
รูปที่ 3.6	โปรแกรม Park Air S4 IP Controller	47
รูปที่ 3.7	CMA 180 Radio Test Set	48
รูปที่ 3.8	ตัวลดทอนสัญญาณ	48
รูปที่ 3.9	วิทยุเครื่องรับความถี่ 120.5 เมกกะเฮิรตซ์ที่ช่อง IP3 ต่อไปยังสวิตช์	49
รูปที่ 3.10	หน้าจอเข้าระบบ Park Air S4 IP Controller	50
รูปที่ 3.11	หน้าจอการกรหัสเข้าระบบ Park air S4 IP Controller	50
รูปที่ 3.12	หน้าจอสำหรับ Setting	51
รูปที่ 3.13	หน้าจอ Controller Setup	51
รูปที่ 3.14	หน้าจอ Radios Setup	52
รูปที่ 3.15	หน้าจอการตั้งค่า Radios Setup ของวิทยุเครื่องรับ	52
รูปที่ 3.16	หน้าจอการตั้งค่า Radios Setup ของวิทยุเครื่องส่ง	53
รูปที่ 3.17	หน้าจอ Channels Setup	53
รูปที่ 3.18	หน้าจอการตั้งค่า Channels Setup ของ Channel Test	54
รูปที่ 3.19	หน้าจอ Roles Setup	54
รูปที่ 3.20	หน้าจอการตั้งค่า Initial Role	55
รูปที่ 3.21	ตำแหน่งของ Channel ที่เลือก	55
รูปที่ 3.22	การเลือก Channel ที่ใช้กับตำแหน่งของ Channel ที่เลือก	55
รูปที่ 3.23	ผลของหน้าจอการตั้งค่า Roles Setup	56
รูปที่ 3.24	หน้าจอตอนที่ Session การเชื่อมต่อกับวิทยุยังไม่สมบูรณ์	56
รูปที่ 3.25	หน้าจอตอนที่ Session การเชื่อมต่อกับวิทยุสมบูรณ์แล้ว	57
รูปที่ 3.26	หน้าจอเข้าระบบ Config ของสวิตช์	58
รูปที่ 3.27	การตั้งค่า Port ต้นทางของสวิตช์	58
รูปที่ 3.28	การตั้งค่า Port ปลายทางของสวิตช์	59
รูปที่ 3.29	หน้าจอ Config port mirroring ของสวิตช์	59
รูปที่ 3.30	หน้าจอแสดงผล เมื่อทำการ Config port mirroring แล้ว	60
รูปที่ 3.31	บล็อกไดอะแกรมการเชื่อมต่ออุปกรณ์การทดลองที่ 3.3.3	61
รูปที่ 3.32	การเชื่อมต่อสวิตช์เข้ากับอุปกรณ์ต่าง ๆ	61
รูปที่ 3.33	บล็อกไดอะแกรมการเชื่อมต่ออุปกรณ์การทดลองที่ 3.3.6	62
รูปที่ 3.34	บล็อกไดอะแกรมสำหรับการทดลองที่ 3.3.11.1	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่		หน้า
รูปที่ 3.35	หน้าจอเลือกฟังก์ชันของเครื่อง CMA 180 Radio Test Set	65
รูปที่ 3.36	หน้าจอ RX Test	66
รูปที่ 3.37	การตั้งค่าฟิลเตอร์	66
รูปที่ 3.38	หน้าจอ tab ย่อย AF Spectrum	67
รูปที่ 3.39	บล็อกไดอะแกรมสำหรับการทดลอง 3.3.11.2	67
รูปที่ 3.40	หน้าจอ Record-Play tab	68
รูปที่ 3.41	หน้าจอ Status tab	69
รูปที่ 4.1	การเชื่อมต่ออุปกรณ์ระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ	70
รูปที่ 4.2	โพรโทคอล SIP ทำการสร้าง Session สำหรับเริ่มสร้างเส้นทางการติดต่อ	71
รูปที่ 4.3	โพรโทคอล RTP ที่ใช้ใน Session สำหรับการรับ-ส่งแพ็กเก็ตข้อมูล VoIP	72
รูปที่ 4.4	โพรโทคอล SIP ทำการสร้าง Session สำหรับยกเลิกเส้นทางการติดต่อ	72
รูปที่ 4.5	ไดอะแกรมแพ็กเก็ต IP	73
รูปที่ 4.6	แพ็กเก็ตข้อมูลที่ดักจับได้	74
รูปที่ 4.7	แพ็กเก็ตข้อมูลที่ดักจับได้จาก IP Address ของวิทยุ	75
รูปที่ 4.8	บิตข้อมูลของแพ็กเก็ต RTP	76
รูปที่ 4.9	บิตข้อมูลของ Header extension ภายในแพ็กเก็ต RTP	76
รูปที่ 4.10	โครงสร้างของแพ็กเก็ต RTP	77
รูปที่ 4.11	ผลการตรวจสอบสถานะของตัวอย่างที่ 1	77
รูปที่ 4.12	ผลการตรวจสอบสถานะของตัวอย่างที่ 2	78
รูปที่ 4.13	ไฟล์เสียง sys.wav ที่ถูกสร้างขึ้น	79
รูปที่ 4.14	รูปร่างของฟิลเตอร์ C-message weighting ที่ใช้	79
รูปที่ 4.15	สเปกตรัมของสัญญาณคลื่นรูปไซน์ที่ 3,000 เฮิรตซ์	80
รูปที่ 4.16	Record-Play tab ของ GUI เมื่อมีการกำหนด IP และชื่อไฟล์เสียงนามสกุล .wav	83
รูปที่ 4.17	ไฟล์เสียง record_test.wav ถูกสร้างขึ้น	84
รูปที่ 4.18	การแสดงผลของ Status tab สำหรับวิทยุเครื่องรับ	84

## สารบัญตาราง

ตารางที่		หน้า
ตารางที่ 1.1	แผนการดำเนินงาน	3
ตารางที่ 2.1	คุณสมบัติของ CODEC	21
ตารางที่ 2.2	Payload types (PT) สำหรับ CODEC	21
ตารางที่ 2.3	รายการของ PTT Value	25
ตารางที่ 2.4	บิตข้อมูล Squelch	27
ตารางที่ 2.5	การแทนค่าตัวเลขด้วยรูปแบบบิต/ไบต์	37
ตารางที่ 2.6	C-message weighting coefficients and accuracy limits	39
ตารางที่ 3.1	คุณสมบัติของวิทยุ Park Air Sapphire T6 Series3	44
ตารางที่ 3.2	คุณสมบัติของสวิตช์ HPE OfficeConnect 1920S Series Switch JL381	46
ตารางที่ 3.3	คุณสมบัติของโปรแกรม Park Air S4 IP Controller	46
ตารางที่ 4.1	ค่าความถี่ของเสียงระบบแอนะลอก และดิจิทัล	81
ตารางที่ 4.2	ค่า SINAD ของเสียงระบบแบบแอนะลอก และดิจิทัล	82

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญ

ในกิจการการบินอากาศ การสื่อสารระหว่างเจ้าหน้าที่ควบคุมจราจรทางอากาศ และนักบิน เป็นสิ่งสำคัญ ที่จะต้องดำเนินการด้วยความปลอดภัย และสามารถตรวจสอบได้ ซึ่งในปัจจุบันระบบการสื่อสารการบินอากาศและการสังเกตการณ์การสื่อสารการบินอากาศของบริษัท วิทยุการบินแห่งประเทศไทย จำกัด ใช้ระบบการสื่อสารแบบแอนะล็อก ซึ่งมีเครื่องมือสำหรับสังเกตการณ์การสื่อสารอยู่ทั้งหมด 2 ตำแหน่ง แสดงดังรูปที่ 1.1 เมื่อบริษัทฯ มีการใช้งานช่องสัญญาณ (Channel) เป็นจำนวนมากขึ้น ทำให้ประสบปัญหาข้อจำกัดของจำนวนอุปกรณ์ที่ใช้ เมื่อต้องการขยายจำนวนช่องสัญญาณ

จากข้อจำกัดด้านการสื่อสาร และการเพิ่มจำนวนอุปกรณ์ในข้างต้น ทำให้ทางบริษัทฯ ได้เห็นถึงความสำคัญที่จะนำการสื่อสารผ่าน Voice over IP (VoIP) มาใช้งาน เพื่อแก้ไขข้อจำกัดดังกล่าว แต่ระบบในปัจจุบันนั้นยังไม่เปิดใช้งานอย่างเต็มระบบ เป็นเพียงการสำรองข้อมูล (Backup) ในบริเวณกรอบเส้นประในรูปที่ 1.2 เท่านั้น ซึ่งเมื่อเปิดใช้งานเต็มระบบ หรือบริเวณกรอบจุดไข่ปลา ในรูปที่ 1.2 จะเห็นได้ว่าระบบนั้นมีเครื่องมือสำหรับสังเกตการณ์ในตำแหน่งที่ 2 ยังคงขาดเครื่องมือที่ใช้สำหรับการสังเกตการณ์ระบบการสื่อสารการบินอากาศผ่านระบบ VoIP ในตำแหน่งที่ 1 เมื่อเทียบกับการสื่อสารระบบแอนะล็อก

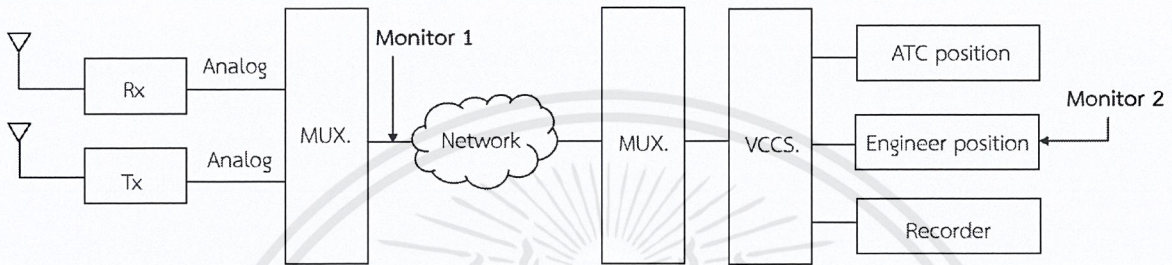
โครงการนี้จึงมีวัตถุประสงค์เพื่อออกแบบ และพัฒนาระบบสังเกตการณ์การสื่อสารการบินอากาศ สำหรับใช้ในการตรวจสอบสถานะการเชื่อมต่อ และการรับ-ส่งข้อมูลเสียงของวิทยุ รวมทั้งการบันทึก และเล่นเสียงที่ผ่านการสื่อสารการบินอากาศผ่านระบบ VoIP ดังแสดงในรูปที่ 1.3

### 1.2 วัตถุประสงค์ของโครงการ

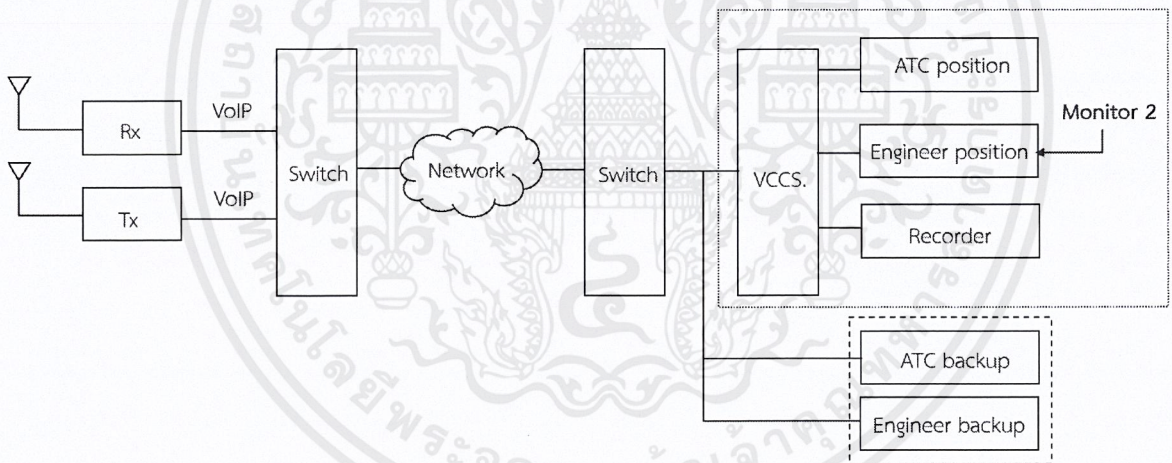
- 1) เพื่อศึกษาการสื่อสารการบินอากาศทางเสียงผ่านระบบ VoIP
- 2) เพื่อออกแบบโปรแกรมต้นแบบสำหรับสังเกตการณ์การสื่อสารการบินอากาศผ่านระบบ VoIP
- 3) เพื่อศึกษาการเขียนโปรแกรมระบบสังเกตการณ์การสื่อสารการบินอากาศผ่านระบบ VoIP โดยใช้ภาษาไพทอน (Python)
- 4) เพื่อศึกษาการเขียน Graphical User Interface (GUI) หรือส่วนต่อประสานกราฟิกกับผู้ใช้

### 1.3 ขอบเขตของโครงการ

ทำการออกแบบ และจำลองโปรแกรมต้นแบบสำหรับระบบสังเกตการณ์การสื่อสารการเดินอากาศทางเสียงผ่านระบบ VoIP โดยสามารถตรวจสอบสถานะ การรับ-ส่งข้อมูลเสียงของวิทยุเครื่องรับ-ส่ง บันทึกลงและเล่นเสียงผ่านโปรแกรมต้นแบบสำหรับสังเกตการณ์การสื่อสารการเดินอากาศ ซึ่งสามารถแสดงผลผ่านทาง GUI โดยใช้ภาษาไพทอนเพื่ออำนวยความสะดวกการใช้งาน



รูปที่ 1.1 บล็อกไดอะแกรมระบบการสื่อสารการเดินอากาศแบบแอนะล็อก



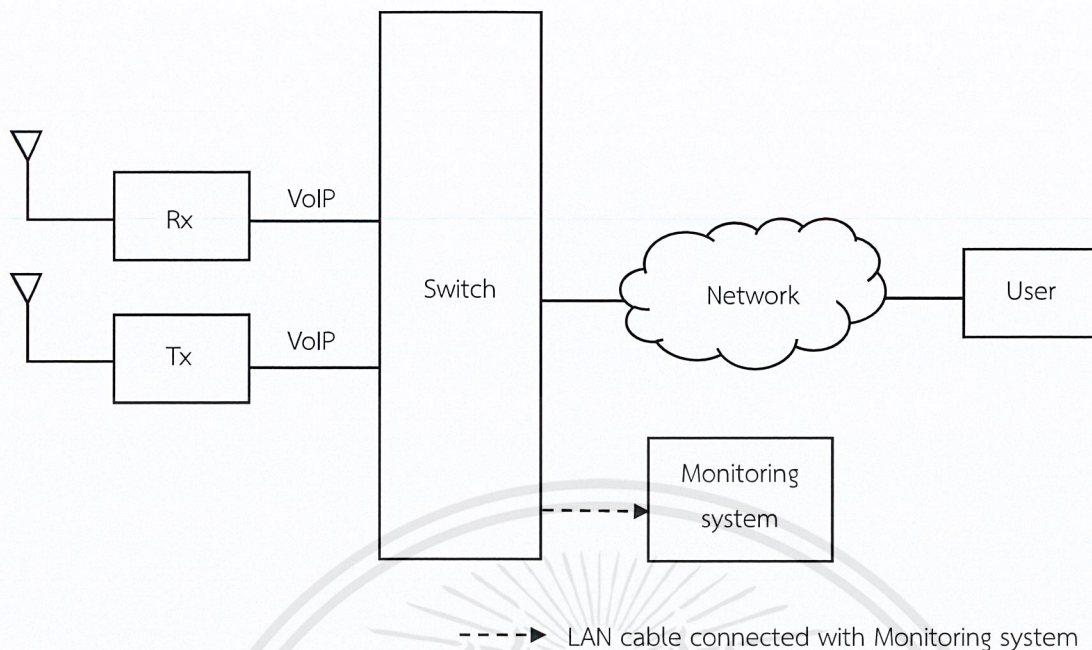
รูปที่ 1.2 บล็อกไดอะแกรมระบบการสื่อสารการเดินอากาศแบบ VoIP

## 1.4 วิธีการดำเนินงานโครงการ

ตารางที่ 1.1 แผนการดำเนินงาน

ลำดับ	รายละเอียด	ภาคการศึกษาที่ 1 ปีการศึกษา 2562			
		เดือนที่ 1	เดือนที่ 2	เดือนที่ 3	เดือนที่ 4
1	ศึกษารูปแบบการรับ-ส่งข้อมูล เสียงสำหรับการสื่อสารการ เดินอากาศผ่านระบบ VoIP	↔			
2	สังเกตรูปแบบการรับ-ส่งข้อมูลใน เครือข่ายผ่านโปรแกรม Wireshark	↔			
3	เปรียบเทียบรูปแบบการรับ-ส่ง ข้อมูลจากโปรแกรม Wireshark กับการศึกษาในลำดับที่ 1	↔			
4	ศึกษาภาษาไพทอน		↔		
5	เขียนโปรแกรมต้นแบบสำหรับ การตรวจสอบการสื่อสาร		↔	↔	
6	ทดสอบโปรแกรมต้นแบบสำหรับ การตรวจสอบการสื่อสาร			↔	
7	เขียนโปรแกรมส่วน GUI ให้ใช้ งานร่วมกับโปรแกรมต้นแบบ สำหรับการตรวจสอบการสื่อสาร				↔
8	ทดสอบ และแก้ไขโปรแกรม ต้นแบบสำหรับการตรวจสอบการ สื่อสาร				↔

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.3 บล็อกไดอะแกรมของระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้รู้ถึงหลักการการสื่อสารการเดินอากาศผ่านระบบ VoIP
- 2) สามารถสร้างระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ เพื่อนำไปใช้ในการตรวจสอบสถานการณ์เชื่อมต่อของวิทยุเครื่องรับ-ส่ง บันทึกลง และเล่นเสียงได้
- 3) สามารถนำความรู้ที่ได้เรียนในภาคทฤษฎีมาประยุกต์ใช้ในเชิงปฏิบัติได้
- 4) ได้ฝึกฝนทักษะความสามารถในการเขียนโปรแกรม และการทำ GUI

### 1.6 ขั้นตอนของการศึกษา

โครงงานเล่มนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกัน คือ

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินงานโครงงาน และประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

บทที่ 2 กล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้องในการเขียนโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ ประกอบด้วยภาษาไพทอนที่ใช้ในการเขียนโปรแกรม เสียง การมอดูเลตทางขนาด การมอดูเลตรหัสพัลส์ VoIP ED-137 โปรแกรม Sniffer หลักการ Port mirroring และมาตรฐานเสียง G.711

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 4 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 กล่าวถึงการออกแบบระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ ประกอบด้วย การออกแบบการทำงานของระบบ เครื่องมือที่ใช้ในการทดลอง และการจัดเก็บผลการทดลองของโครงการนี้

บทที่ 4 กล่าวถึงผลการทดลอง โดยจะแสดงผลการทดลองต่าง ๆ เช่น ผลของการเชื่อมต่อ และรับข้อมูลจากวิทยุแบบ VoIP ผลการทดลองบันทึกผลการทดลองจากการใช้งานโปรแกรมสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ ร่วมกับ GUI และผลการทดลองเปรียบเทียบคุณภาพเสียงที่ได้จากการบันทึกของโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ กับเสียงที่ได้จากระบบแอนะล็อกเดิม

บทที่ 5 กล่าวถึงบทสรุปผลการทดลองและข้อเสนอแนะ



## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

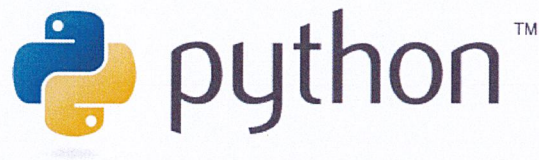
โครงการนี้ได้จัดทำขึ้นเพื่อการออกแบบ และจำลองโปรแกรมต้นแบบสำหรับระบบสังเกตการณ์ การสื่อสารการเดินอากาศทางเสียงผ่านระบบ VoIP โดยสามารถตรวจสอบสถานะ การรับ-ส่งข้อมูลเสียง ของวิทยุเครื่องรับ-ส่ง บันทึกลง และเล่นเสียงผ่านโปรแกรมต้นแบบสำหรับสังเกตการณ์การสื่อสารการเดินอากาศ ซึ่งสามารถแสดงผลผ่านทาง GUI โดยใช้ภาษาไพทอน เพื่อง่ายต่อการใช้งาน จึงมีทฤษฎี และ หลักการที่เกี่ยวข้องดังต่อไปนี้

#### 2.1 ภาษาไพทอน

ภาษาโปรแกรมไพทอน (Python) คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้ เป็นภาษาบทคำสั่ง (Scripting language) ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้าง และไวยากรณ์ของ ภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เขียน ให้เป็นภาษาเครื่องไพทอน มีการทำงานแบบ Interpreter คือ เป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงาน ตามที่เราต้องการ โดยมีตราสัญลักษณ์ของโปรแกรมภาษาไพทอน แสดงได้ดังรูปที่ 2.1

ภาษาโปรแกรมไพทอนสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้ จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลาย ในหลายองค์กรใหญ่ระดับโลก เช่น Google YouTube Instagram Dropbox และ NASA เป็นต้น นอกจากนี้ยังถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถทำงานได้ทั้งบนระบบ Unix Linux Windows NT Windows 2000 Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่ง ภาษาไพทอนเป็น OpenSource เหมือนอย่างภาษา PHP ทำให้ทุกคนสามารถที่จะนำไพทอนมาพัฒนาโปรแกรม ของตัวเองได้ โดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ ภาษาไพทอนมีความสามารถสูงขึ้น และใช้งานได้ครอบคลุมกับทุกลักษณะงาน

คุณลักษณะเด่นของภาษาไพทอนที่ทำให้ผู้จัดทำโครงการเลือกใช้ภาษาไพทอน คือ ภาษาไพทอน มีมอดูล (Module) และฟังก์ชัน (Function) ที่หลากหลาย เช่น มอดูล Sockets ที่ใช้เชื่อมต่อกับ อินเทอร์เน็ต มอดูล Scipy.signal ที่ใช้ในการสร้างตัวกรองสัญญาณแบบดิจิทัล (Digital filter) และมอดูล PyQt5 ที่สามารถใช้ในการสร้าง GUI ฯลฯ นอกจากนี้ PyQt5 นั้นยังมีโปรแกรม Qt Designer ที่ใช้ช่วยให้ การออกแบบหน้าต่าง GUI ง่าย และประหยัดเวลามากขึ้นอีกด้วย



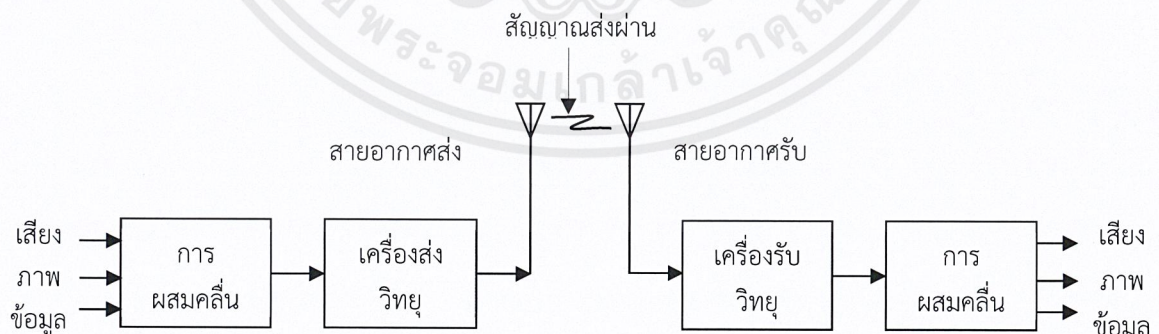
## รูปที่ 2.1 ตราสัญลักษณ์ของโปรแกรมภาษาไพทอน [1]

### 2.2 เสียง

เสียงเป็นคลื่นกลตามยาวชนิดหนึ่ง ซึ่งอาศัยตัวกลางในการเคลื่อนที่ ดังนั้นคุณสมบัติของเสียงจึงเหมือนคลื่นครบทั้ง 4 ประการ ได้แก่ การสะท้อน การหักเห การแทรกสอด และการเลี้ยวเบน คลื่นเสียงที่มนุษย์สามารถได้ยิน (Audible waves) จะมีความถี่โดยประมาณ 20 เฮิรตซ์ (Hertz หรือ Hz) ถึง 20 กิโลเฮิรตซ์ (Kilohertz หรือ kHz) [2]

### 2.3 หลักการส่งข่าวสารข้อมูลทางคลื่นวิทยุแบบการมอดูเลตทางขนาด

การสื่อสารโทรคมนาคม นอกเหนือจากการสื่อสารทางสายแล้ว การสื่อสารทางคลื่นวิทยุนับว่าเป็นการสื่อสารที่มีความสำคัญ ถูกนำไปใช้งานอย่างกว้างขวางแพร่หลาย เพราะสามารถส่งข่าวสารข้อมูลไปได้เป็นระยะทางไกล ๆ ช่วยแก้ปัญหาการสื่อสารทางสายที่ในบางพื้นที่ทำไม่ได้ หรือไม่คุ้มค่าในแง่การลงทุน และการส่งข้อมูลไปได้ครั้งละมาก ๆ ทำให้เกิดการประหยัด มีความรวดเร็ว และไม่เกิดความผิดพลาด หลักการส่งข่าวสารข้อมูลทางคลื่นวิทยุ แสดงดังรูปที่ 2.2



รูปที่ 2.2 หลักการส่งข่าวสารข้อมูลทางคลื่นวิทยุ [3]

จากรูปที่ 2.2 แสดงหลักการส่งข่าวสารข้อมูลทางคลื่นวิทยุ โดยการนำข่าวสารข้อมูลต่าง ๆ และสัญญาณไฟฟ้าต่าง ๆ ไปเข้าขบวนการผสมคลื่นกับสัญญาณคลื่นวิทยุ ส่งต่อไปให้เครื่องส่งวิทยุออกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายอากาศแพร่กระจายคลื่นออกไปในอากาศในรูปคลื่นแม่เหล็กไฟฟ้า ทางด้านรับคลื่นแม่เหล็กไฟฟ้า ส่งผ่านเข้าสายอากาศ ส่งต่อไปให้เครื่องรับวิทยุรับสัญญาณคลื่นเหล่านี้เข้ามา ผ่านไปเข้าขบวนการแยกคลื่น ส่งออกไปทางปลายทางเพื่อใช้งาน

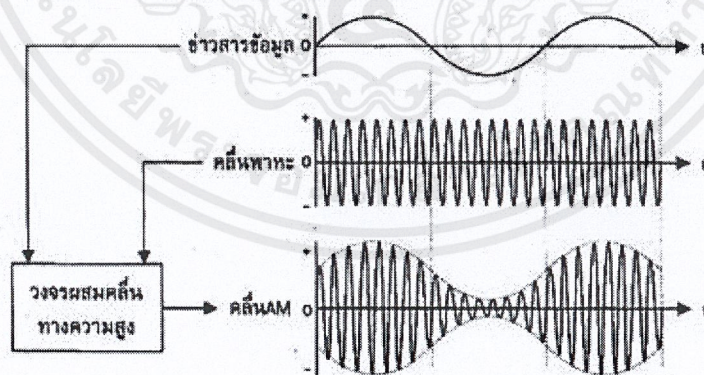
คลื่นวิทยุที่ใช้เป็นตัวพาข่าวสารข้อมูลต่าง ๆ ไปได้ นั่น เพราะคลื่นวิทยุสามารถเดินทางไปได้ไกล มีความเร็วในการเดินทางเท่ากับคลื่นแสง เกิดการสูญเสียขณะเดินทางต่ำ คลื่นวิทยุนี้มีชื่อเรียกว่า คลื่นพาห้ (Carrier) การนำข่าวสารข้อมูลไปผสมกับคลื่นพาห้ นั้น คือการนำข่าวสารฝากไปกับคลื่นพาห้ นั้นเอง เพื่อให้ข่าวสารเดินทางได้ไกล ที่นิยมใช้งานมี 3 แบบคือ

- 1) การผสมคลื่นทางแอมพลิจูด หรือ การมอดูเลตแอมพลิจูด (Amplitude Modulation หรือ AM)
- 2) การผสมคลื่นทางความถี่ หรือ การมอดูเลตความถี่ (Frequency Modulation หรือ FM)
- 3) การผสมคลื่นทางเฟส หรือ การมอดูเลตเฟส (Phase Modulation หรือ PM)

การการผสมสัญญาณข่าวสารข้อมูลเข้ากับคลื่นพาห้ ในแต่ละแบบมีความแตกต่างกันในลักษณะ การผสมสัญญาณเข้าด้วยกัน ได้รูปแบบคลื่นไม่เหมือนกัน และใช้งานแตกต่างกัน ซึ่งในโครงการนี้ได้ ทำการศึกษาเกี่ยวกับการมอดูเลตแอมพลิจูดเท่านั้น

### 2.3.1 การมอดูเลตแอมพลิจูด

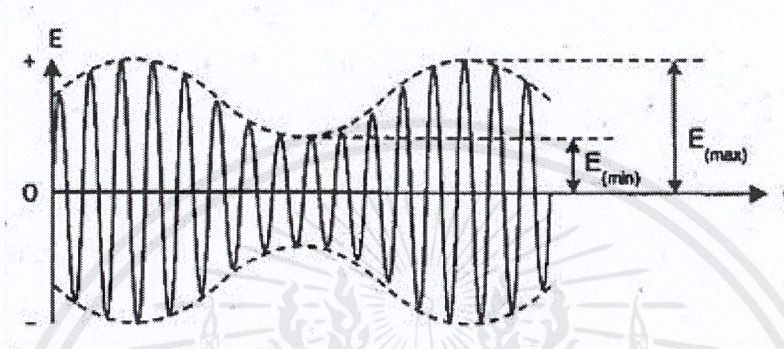
การผสมคลื่นทางแอมพลิจูด คือการนำสัญญาณข่าวสารข้อมูลไปผสมกับคลื่นพาห้ ควบคุมโดยสัญญาณข้อมูลข่าวสารเพิ่มขึ้น หรือลดลงโดยความถี่ของคลื่นพาห้ยังคงเดิม การผสมคลื่นทางความสูงแสดง ดังรูปที่ 2.3



รูปที่ 2.3 การมอดูเลตแอมพลิจูด [3]

จากรูปที่ 2.3 แสดงการผสมคลื่นทางความสูง มีภาคผสมคลื่นทางความสูงทำหน้าที่ผสมสัญญาณข่าวสารข้อมูลควบคุมคลื่นพาห้ เกิดการเปลี่ยนแปลงเพิ่มสูงหรือต่ำ ตามช่วงบวก หรือลบ ช่วงบวกทำให้คลื่นพาห้สูงขึ้น ช่วงลบทำให้คลื่นพาห้ต่ำลง และถ้าไม่มีข้อมูลข่าวสารมาผสม คลื่นพาห้จะสูงกว่าปกติ

การผสมคลื่นทางความสูงนั้น สิ่งสำคัญของการผสมคลื่นแบบนี้อยู่ที่เปอร์เซ็นต์ของการผสมคลื่น โดยคิดจากระดับความแรงของข่าวสารไปเปรียบเทียบกับความแรงของคลื่นพาห์ด้วยดัชนี และเปอร์เซ็นต์ การผสมคลื่นออกมา ตำแหน่งการวัดระดับความแรงของคลื่นทางความสูง แสดงดังรูปที่ 2.4 นำค่าที่วัดได้ จากรูปไปคำนวณค่าดัชนีการผสมคลื่น (Modulation index หรือ Modulation depth) ได้ด้วยสมการที่ 2.1



รูปที่ 2.4 การวัดระดับความแรงของคลื่น (AM) [3]

นำค่าความแรงของคลื่นทางความสูงวัดที่ระดับต่ำสุด และวัดที่ระดับสูงสุด มาเข้าสมการเขียนเป็น สูตรคำนวณได้ดังนี้

$$M = \frac{E(\max) - E(\min)}{E(\max) + E(\min)} \quad (2.1)$$

- เมื่อ M คือ ดัชนีการผสมคลื่นแบบ AM ไม่มีหน่วย
- E(max) คือ ระดับแรงต้นวัดจากแกนศูนย์ยอดสูงสุดตำแหน่งแรงสุด
- E(min) คือ ระดับแรงกันวัดจากแกนศูนย์ถึงยอดสูงสุดตำแหน่งเบาสุด

การผสมคลื่นแบบ AM สามารถผสมคลื่นได้สูงสุดไม่เกิน 100% ในช่วงการผสมคลื่นไม่เกิน 100% นี้สัญญาณข้อมูลข่าวสารมีความสามารถเกาะติดพาหะได้ ทำให้คลื่นทั้งหมด ทำให้คลื่นสัญญาณข่าวสาร ข้อมูลที่รับได้ทางปลายทางสมบูรณ์ทุกประการ

คลื่นวิทยุกับสัญญาณไฟฟ้ามีความแตกต่างกัน คือ ค่าความถี่ของสัญญาณไฟฟ้ากระแสสลับ (Alternating current หรือ AC) มีความถี่ 50 เฮิรตซ์ หรือ 60 เฮิรตซ์ ส่วนคลื่นวิทยุจะมีค่าความถี่เป็น เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เหมือนญาติเห็นไปเซประยะขึ้นดำเนินการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิโลเฮิร์ตซ์ (kHz) เมกะเฮิร์ตซ์ (MHz) หรือจิกะเฮิร์ตซ์ (GHz) ซึ่งสูงมาก แต่ความเหมือนระหว่างสัญญาณไฟฟ้ากับคลื่นวิทยุ คือ ทั้งสองต่างเป็นไพล์แอสลับ พลังงานคลื่นวิทยุที่ส่งผ่านสายอากาศจากเครื่องส่งไปยังเครื่องรับก็เป็นสัญญาณไพล์แอสลับเช่นกัน แต่ความถี่จะสูงมาก เช่น คลื่นวิทยุมีความถี่ 27 เมกะเฮิร์ตซ์ มีพลังงานสูงเพียงพอที่จะส่งคลื่นให้เดินทางผ่านอากาศไปยังเครื่องรับที่อยู่ไกล ๆ หลายร้อยกิโลเมตรได้

ความถี่เสียง (Audio Frequency หรือ AF) และความถี่วิทยุ มนุษย์สามารถรับฟังคลื่นความถี่เสียงได้ในช่วง 2 เฮิร์ตซ์ ถึง 20,000 เฮิร์ตซ์ ความถี่ในช่วงดังกล่าวจึงเรียกว่าเป็นความถี่เสียง เสียงทุกแบบไม่ว่าจะเป็นเสียงดนตรี หรือเสียงพูด เสียงพูดที่มนุษย์ได้ยินจึงถูกจัดเป็นคลื่นความถี่เสียงทั้งหมด ส่วนสัญญาณไฟฟ้าที่มีความถี่สูงเกินกว่า 20,000 เฮิร์ตซ์ขึ้นไป หูของมนุษย์ไม่สามารถรับฟังได้เรียกว่าคลื่นความถี่วิทยุ (Radio Frequency หรือ RF) คุณสมบัติของคลื่นความถี่วิทยุจะแตกต่างจากคลื่นความถี่เสียงที่สามารถรับฟังเสียงผ่านลำโพงได้ แต่ไม่สามารถส่งสัญญาณผ่านเครื่องส่งวิทยุไปในพื้นที่เขตบริการได้ แต่คลื่นความถี่วิทยุสามารถทำได้ ดังนั้นการส่งสัญญาณวิทยุกระจายเสียงที่มนุษย์สามารถรับฟังได้ก็เกิดจากการส่งคลื่นความถี่วิทยุนั่นเอง

## 2.4 การมอดูเลตรหัสพัลส์

การมอดูเลตรหัสพัลส์ หรือ Pulse Code Modulation (PCM) เป็นขั้นตอนในการเปลี่ยนสัญญาณแอนะล็อกไปเป็นสัญญาณดิจิทัล แสดงดังรูปที่ 2.5 ซึ่งมีกระบวนการทำงาน 3 ขั้นตอน ดังนี้

### 1) การสุ่มสัญญาณ (Sampling)

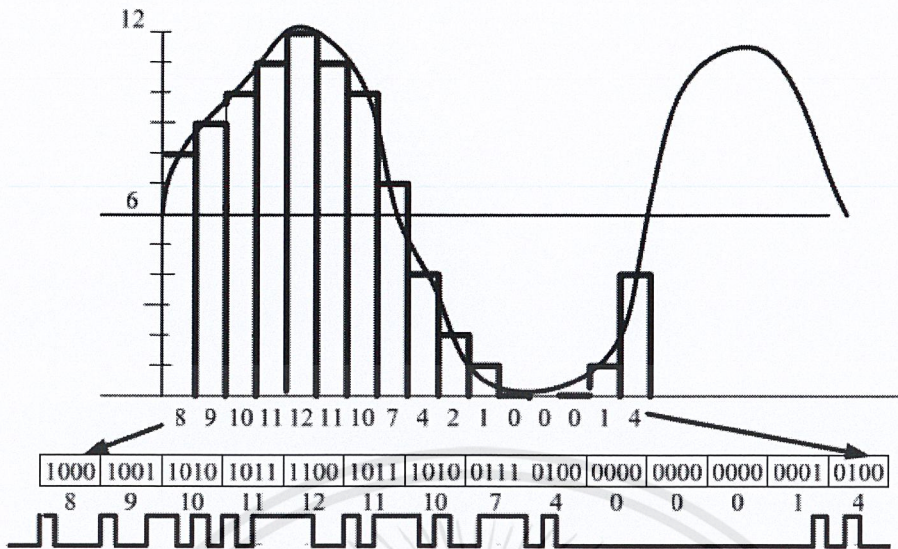
เป็นการเทียบข้อมูลของสัญญาณแอนะล็อกกับสัญญาณดิจิทัลว่า ในช่วงเวลานั้นสัญญาณแอนะล็อกมีค่าแรงดันเท่าไร ซึ่งเมื่อเทียบแล้วได้ข้อมูลสัญญาณดิจิทัลเป็นเท่าไร

### 2) การควอนไทซ์ (Quantization)

เป็นการนำผลของการสุ่มค่ามาเทียบค่า ของจุดในการสุ่มระดับสัญญาณ 1 ลูกคลื่น เพื่อให้ทราบค่าของระดับสัญญาณที่ทำการสุ่มว่ามีค่าเท่าไร

### 3) การเข้ารหัสข้อมูล (Encoder)

เป็นการให้ข้อมูลตัวเลขแบบดิจิทัลตามระดับของสัญญาณที่เกิดจากการเทียบค่าในการทำควอนไทซ์



รูปที่ 2.5 การมอดูเลตรหัสพัลส์ [4]

## 2.5 Voice over IP (VoIP)

VoIP คือเทคโนโลยีการนำสัญญาณเสียงมาผสมกับสัญญาณข้อมูล เพื่อส่งไปบนระบบเครือข่าย Internet Protocol (IP) โดยปกติจะใช้ IP ในการส่งสัญญาณข้อมูลเท่านั้น เทคโนโลยี VoIP นี้ได้พัฒนาจนสามารถสื่อสารสัญญาณเสียงผ่าน IP ได้ ทำให้ประหยัดค่าใช้จ่ายในส่วนของการเช่าโทรศัพท์ได้มากขึ้น ซึ่งการติดต่อสื่อสารแบบเดิมเป็นแบบแอนะล็อก ทำให้สิ้นเปลืองเวลา และอุปกรณ์ เช่น การเชื่อมต่อโทรศัพท์ระหว่างปลายทาง และต้นทาง เมื่อระบบเชื่อมต่อโทรศัพท์ได้แล้วหมายความว่า ช่องทางการติดต่อของคู่สายนี้จะถูกจองทั้งสาย ไม่สามารถรับการเชื่อมต่อจากคู่สายอื่นได้จนกว่าจะจบการเชื่อมต่อของคู่สนทนาสายโทรศัพท์เส้นนี้จึงจะว่าง หากเป็นระบบดิจิทัล ซึ่งไม่ใช่เป็นการเพิ่มสายให้มีช่องทางมากขึ้น เพียงแต่เสมือนการแบ่งเลน ให้มีหลายช่องจราจร มีหลายระดับความเร็วแบ่งกันใช้

เมื่อเอาโทรศัพท์ที่สามารถใช้ระบบ IP Telephony มาเชื่อมต่อก็เหมือนกับว่าโทรศัพท์ 2 เครื่องต่อผ่านสายโทรศัพท์ 1 เส้น แต่การส่งสัญญาณไปกลับจะถูกแบ่งเป็นแพ็กเกต (Packet) แล้วทยอยส่งเป็นช่วง ช่วงว่างก็เป็นโอกาสให้ผู้อื่นส่งบ้าง เรียกว่าไปด้วยกัน แบ่งเลนกัน แบ่งเวลากัน ดังนั้นในช่วงเวลาเท่า ๆ กัน ระบบ IP Telephony จึงสามารถคุยกันได้

VoIP สามารถประยุกต์ใช้กับข้อกำหนดของการสื่อสารได้เกือบทั้งหมด ตั้งแต่การติดต่อภายในองค์กร ระหว่างองค์กร ระหว่างประเทศ อุปกรณ์ที่ใช้ต้องมีความยืดหยุ่นสูง สามารถสร้างความพึงพอใจให้กับทุกฝ่ายที่ใช้งาน และใช้งานได้ในสภาพแวดล้อมที่หลากหลาย ที่สำคัญต้องสามารถใช้ร่วมกับระบบโทรศัพท์แบบเดิมได้ด้วย ยังคงต้องใช้ควบคู่กันไป เพราะปัจจุบันส่วนใหญ่องค์กรต่าง ๆ ยังคงใช้ระบบโทรศัพท์แบบเดิมอยู่ ทำให้สามารถติดต่อสื่อสารกันได้ทั้งระบบ VoIP และระบบโทรศัพท์แบบเดิม

เมื่อมองว่าข้อมูลเสียงเหมือนกับข้อมูลทั่วไป เช่น E-mail, Web, FTP ซึ่งความเหมือนนี้หมายถึงมีผู้ส่ง และผู้รับเช่นเดียวกัน เช่น E-mail มีผู้ส่ง และมีผู้รับ ส่วนข้อมูลเสียงก็มีฝั่งต้นทาง และปลายทาง โดยทั้งเสียง และ E-mail เดินทางผ่าน Internet Protocol Network (IP Network) [5]

สำหรับโครงงานนี้ที่ใช้วิทยุสื่อสารแบบ IP นั้น เป็นบริการวิทยุสื่อสารเฉพาะกลุ่มที่สามารถสื่อสารได้ครอบคลุมได้ทุกพื้นที่ โดยการเชื่อมต่อผ่าน IP สามารถเชื่อมต่อกับระบบผ่านสัญญาณ Wi-Fi หรือโครงข่ายโทรศัพท์ และอ้างอิงสัญญาณข้อมูลผ่านโครงข่ายโทรศัพท์ ทำให้มีความจำเป็นต้องใช้โครงข่ายร่วมกับโครงข่ายโทรศัพท์

### 2.5.1 การทำงานของ IP Network

Internet Protocol หรือ IP จะเป็นโพรโทคอล (Protocol) ที่ใช้ในการสื่อสารข้อมูลในระบบอินเทอร์เน็ต (Internet) ซึ่งในส่วนของการทำงานคร่าว ๆ IP นี้สามารถสรุปอย่างย่อได้ดังต่อไปนี้

สัญญาณข้อมูลจะถูกแบ่งออกเป็น ส่วน ๆ แต่ละส่วนของข้อมูลจะถูกส่งออกไปในเส้นทางที่อาจแตกต่างกันบนระบบอินเทอร์เน็ต ข้อมูลย่อยแต่ละส่วนนั้นจะไปถึงยังปลายทางในเวลา และลำดับไม่พร้อมเพรียงกัน หลังจากนั้นจะมีโพรโทคอลอีกหนึ่งตัวเข้ามาเกี่ยวข้อง คือ Transmission Control Protocol (TCP) ซึ่ง TCP นี้จะเข้ามาช่วยเกี่ยวกับการเรียงลำดับข้อมูลที่มาถึงยังปลายทางนี้ให้อยู่ในลำดับ และรูปแบบที่ถูกต้องเหมือนข้อมูลต้นแบบก่อนที่จะถูกส่งออกมา

IP นี้จะเป็นโพรโทคอลการสื่อสารแบบที่เรียกว่า Connectionless Protocol ซึ่งเป็นการสื่อสารที่จุดต้นทาง และปลายทางของการสื่อสารไม่จำเป็นต้องสร้างการเชื่อมต่อขึ้นมา ณ เวลาที่ต้องการทำการสื่อสาร

### 2.5.2 มาตรฐานของเทคโนโลยี VoIP

โดยทั่วไปจะมีอยู่ 2 มาตรฐาน คือ

#### 1) มาตรฐาน H.232

โดยทั่วไปแล้วระบบนี้ไม่ได้ถูกออกแบบมาให้ใช้งานกับระบบเครือข่ายที่ใช้ IP นอกจากนั้นมาตรฐานนี้มีการทำงานที่ค่อนข้างช้า โดยปกติจะใช้ก็ต่อเมื่อในระบบเดิมของผู้ใช้มีการใช้มาตรฐาน H.323 อยู่แล้วเท่านั้น

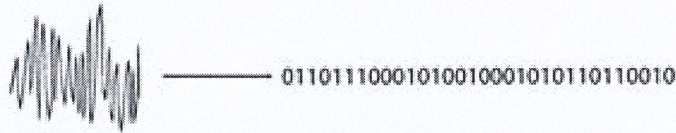
#### 2) มาตรฐาน Session Initiation Protocol (SIP)

เป็นมาตรฐานใหม่ในการใช้งานของเทคโนโลยี VoIP ซึ่งมาตรฐาน SIP ได้ถูกออกแบบมาเพื่อใช้กับระบบ IP โดยเฉพาะ ซึ่งโดยปกติแล้วจะแนะนำให้ผู้ใช้ใหม่ที่จะมีการใช้ VoIP ให้มีการใช้งานอยู่บนมาตรฐาน SIP และเป็นมาตรฐานที่มีความเชื่อถือได้ค่อนข้างสูง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.3 กระบวนการทำงานของเทคโนโลยี VoIP

#### 1) การแปลงสัญญาณเป็น PCM (Pulse Code Modulation)

เป็นการแปลงสัญญาณแอนะล็อกให้ไปอยู่ในรูปแบบสัญญาณดิจิทัล หรือเรียกว่า PCM แสดงดังรูปที่ 2.6



รูปที่ 2.6 การแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล [6]

#### 2) การนำสัญญาณสะท้อนกลับออก

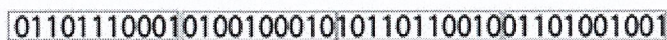
เป็นการแยกสัญญาณออกเป็นส่วน ๆ เพื่อทำการตัดสัญญาณสะท้อนกลับ (echo) ออก ซึ่งกระบวนการนี้จะถูกจัดการโดย DSP (digital signal processor) แสดงดังรูปที่ 2.7



รูปที่ 2.7 การนำสัญญาณสะท้อนกลับออก [6]

#### 3) การทำเฟรมมิ่ง (Framing)


สัญญาณส่วนที่เหลือจะถูกแบ่งและจัดรูปแบบขึ้นมาใหม่ในรูปของ เฟรม (frame) ซึ่งกระบวนการนี้จะถูกจัดการโดยรูปแบบการบีบอัดที่เรียกว่า CODEC (Coder/Decode) หลังจากนั้นเฟรมของสัญญาณเสียงจะถูกสร้างขึ้น แสดงดังรูปที่ 2.8



รูปที่ 2.8 การจัดข้อมูลในรูปของเฟรม [6]

#### 4) Packetization

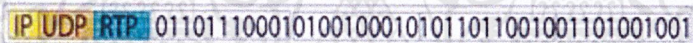
ในกระบวนการนี้จะเป็นการแปลง เฟรมของสัญญาณให้มาอยู่ในรูปของ แพ็กเก็ตซึ่งจะมีการเพิ่ม header เข้าไปในแพ็กเก็ต โดยในส่วนของส่วนหัวนั้น ก็จะประกอบไปด้วยข้อมูลที่เรียกว่า Sequence Number และ Time Stamp หลังจากนั้นแพ็กเก็ตนี้จะถูกส่งต่อไปที่ Host Processor แสดงดังรูปที่ 2.9



รูปที่ 2.9 การแปลงเฟรมให้อยู่ในรูปแพ็กเก็ต [6]

#### 5) Address and Delivery

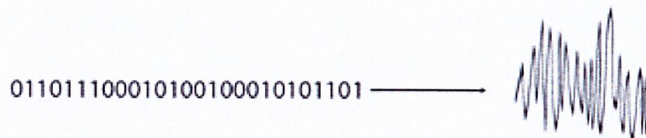
หลังจากที่ได้แปลงสัญญาณให้อยู่ในรูปของแพ็กเก็ตแล้ว ข้อมูลนั้นจะถูกนำมาวิเคราะห์ และใส่ค่า IP Address ปลายทาง แสดงดังรูปที่ 2.10



รูปที่ 2.10 การใส่ค่า IP Address [6]

#### 6) การแปลงเป็นสัญญาณแอนะล็อก (Conversion to Analog)

หลังจากที่ได้ทำการใส่ค่าของ IP Address ปลายทางไปใน header ของแพ็กเก็ตแล้วนั้น เมื่อแพ็กเก็ตไปถึงปลายทาง ข้อมูล header เหล่านี้จะถูกแยกออกเพื่อให้เหลือแค่เฟรมเสียง (Voice frame) หลังจากนั้นก็จะทำการแปลงสัญญาณดิจิทัลแบบ PCM ให้กลับมาเป็นสัญญาณรูปแบบแอนะล็อก ที่เป็นสัญญาณเสียงที่เราได้ยินกันอีกครั้งหนึ่ง แสดงดังรูปที่ 2.11



รูปที่ 2.11 การแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก [6]

## 7) การแก้ความผิดพลาด (Error Correction)

กระบวนการนี้จะเป็นกระบวนการที่ใช้ในการตรวจสอบและแก้ไขข้อผิดพลาด ซึ่งอาจเกิดขึ้นระหว่างการส่งสัญญาณ และนำมาซึ่งความผิดพลาด หรือความเสียหายของสัญญาณจนทำให้เราไม่สามารถทำการสื่อสารอย่างถูกต้องได้

## 2.6 Air Traffic Management (ATM)

Air Traffic Management (ATM) คือการบริหารจัดการจราจรทางอากาศ ซึ่งแบ่งการบริหารออกเป็น 4 ประเภท [7] ได้แก่

- 1) การให้บริการจราจรทางอากาศ (ATS หรือ Air Traffic Services)
- 2) การควบคุมจราจรทางอากาศ (ATC หรือ Air Traffic Control)
- 3) การจัดการห้วงอากาศ (ASM หรือ Air Space Management)
- 4) การจัดการความคล่องตัวจราจรทางอากาศ (ATFM หรือ Air Traffic Flow Management)

### 2.6.1 การให้บริการจราจรทางอากาศ (ATS)

รัฐบาลได้มอบหมายให้บริษัท วิทยุการบินแห่งประเทศไทย จำกัด (บวท.) เป็นผู้ดำเนินงานเมื่อ 1 กันยายน พ.ศ.2492 จากการจัดตั้ง และแบ่งอาณาเขตแกลงข่าวการบิน (Flight Information Region) ของ แต่ละประเทศ ในการประชุม ICAO 1st Southeast Asia Regional Air Navigation Meeting, New Delhi, 1948 ซึ่งประกอบไปด้วยงานบริหารด้านต่าง ๆ ดังนี้

- En-Route งานด้านเส้นทางบิน
- Terminal ด้านบริการสนามบินที่จัดการจราจรทางอากาศโดยรอบ ๆ
- Flight Information Service งานแกลงข่าวการบิน
- Alerting Service งานเตือนภัยการบิน
- Search and Rescue งานด้านค้นหา และช่วยเหลือ

### 2.6.2 การควบคุมจราจรทางอากาศ (ATC)

#### 2.6.2.1 จุดประสงค์

- เพื่อป้องกันการเกิดอุบัติเหตุระหว่างอากาศยานที่บินอยู่ในอากาศ
- เพื่อป้องกันการเกิดอุบัติเหตุระหว่างอากาศยานกับสิ่งกีดขวางภาคพื้น
- เพื่อให้การจราจรทางอากาศเป็นไปด้วยความสะดวก รวดเร็ว และปลอดภัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่นับผูกพันไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.2.2 ขั้นตอน และการทำงาน

- การนำเครื่องบินออกจากสนามบิน
- การนำเครื่องบินเข้ามายังสนามบิน

### 2.6.2.3 การแบ่งเขตพื้นที่

ซึ่งแบ่งเป็น 3 พื้นที่ตามระยะทาง ได้แก่

#### 1) Aerodrome Control ซึ่งแบ่งหน้าที่เป็น 2 ตำแหน่ง ได้แก่

- Ground Control

ทำหน้าที่ควบคุมจราจรทางอากาศ และลานจอดให้ขับเคลื่อนด้วยความปลอดภัย เป็นระเบียบ และรวดเร็ว ให้คำอนุญาตการขับเคลื่อนบนทางขับ/ลานจอด และแจ้งข่าวสารข้อมูลแก่นักบิน และหน่วยสนับสนุนภาคพื้นดินในการอำนวยความสะดวกให้แก่อากาศยานที่มาใช้บริการ

- Local Control/Tower Control

ทำหน้าที่รับผิดชอบการควบคุมจราจรทางอากาศบนทางวิ่งและอากาศยาน โดยรอบท่าอากาศยาน กำหนดทางวิ่งขึ้น-ลง ให้เหมาะสมกับสภาพอากาศ (สวนทางกับทิศทางลม) ติดต่อประสานงานกับ Approach Control ในการจัดลำดับให้แก่อากาศยานเข้า-ออก และแจ้งเตือนนักบินให้ปฏิบัติตามข้อกำหนดการบิน

#### 2) Approach Control

การควบคุมจราจรทางอากาศเขตประชิดสนามบินมีหน้าที่ และความรับผิดชอบในการควบคุมจราจร และจัดการจราจรทางอากาศ เพื่อนำอากาศยานเข้า-ออก (Arrival-Departure) และบินผ่าน (Overflight) ในพื้นที่ความรับผิดชอบบริเวณโดยรอบสนามบิน ระยะ 50 ไมล์ทะเล ระยะสูง 11,000 ฟุต (1 ไมล์ทะเลเท่ากับ 1.852 กิโลเมตร)

#### 3) Area Control Ground

จัดการจราจรทางอากาศตามเส้นทางบินทั่วอาณา-เขตของประเทศไทย (Bangkok FIR) โดยมีศูนย์ควบคุมจราจรทางอากาศ (Bangkok Area Control Centre) ตั้งอยู่ที่ตึกปฏิบัติการชั้น 2 สำนักงานใหญ่ (ทุ่งมหาเมฆ)

## 2.6.3 การจัดการห้วงอากาศ

มีหน้าที่บริหารจัดการห้วงอากาศให้มีความยืดหยุ่น มีการผ่อนปรนในการใช้งาน ห้วงอากาศที่มีการใช้งานพิเศษ (Special Use Airspace) ให้เป็นประโยชน์สูงสุดต่อเส้นทางบิน เช่น การใช้ความเร็ว การบินตรงสู่จุดบิน การรักษาความสูง การกำหนดห้วงอากาศ ซึ่งต้องคำนึงถึง ขนาด รูปร่าง ห้วงเวลา และกฎ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานห้วงอากาศ เพื่อให้มีผลกระทบน้อยที่สุด ประสานความร่วมมือกับทุกฝ่ายที่เกี่ยวข้องเพื่อให้เกิดผลกระทบน้อยที่สุด (หน่วยงานควบคุมการจราจรทางอากาศ สายการบิน อากาศยานทหารและการบินทั่วไป) การขอใช้ห้วงอากาศในกรณีเฉพาะ จำเป็นต้องมีการแจ้งขอล่วงหน้า (เช่น airshow SAREX) และความซับซ้อนของการปฏิบัติการบิน อาจมีผลทำให้ความยืดหยุ่นของห้วงอากาศลดลง

#### 2.6.4 การบริหารสภาพคล่องการจราจรทางอากาศ

การบริหารสภาพคล่องการจราจรทางอากาศ คือ กระบวนการบริหารจัดการ ปริมาณการจราจรทางอากาศ (Traffic Demand) ให้เหมาะสมกับขีดความสามารถในการรองรับปริมาณการจราจรทางอากาศ (Capacity) ในพื้นที่รับผิดชอบของผู้ให้บริการจราจรทางอากาศ (ANSP) ไม่ว่าจะเป็นพื้นที่ภายในน่านฟ้า (Airspace Sector) หรือบริเวณท่าอากาศยาน ซึ่งขีดความสามารถดังกล่าวก็ขึ้นกับปัจจัยหลายอย่าง เช่น ลักษณะกายภาพของน่านฟ้า/ท่าอากาศยาน ความสามารถของเจ้าหน้าที่ควบคุมจราจรทางอากาศ และระบบสนับสนุน เป็นต้น

##### 2.6.4.1 จุดประสงค์

เป้าหมายหลักของกระบวนการ ATFM คือ การบริหารจัดการให้สภาพการจราจรทางอากาศภายในพื้นที่รับผิดชอบ มีความคล่องตัวและมีปริมาณที่เหมาะสม (Demand-Capacity Balancing) ไม่มากหรือน้อยเกินไป มีความปลอดภัย และมีประสิทธิภาพ โดยคำนึงถึงผลประโยชน์และเน้นความร่วมมือจากผู้มีส่วนได้ส่วนเสียทุกส่วน ไม่ว่าจะเป็นผู้ให้บริการจราจรทางอากาศ ผู้ให้บริการท่าอากาศยาน และผู้ใช้บริการหรือสายการบิน

##### 2.6.4.2 ปัจจัยที่ส่งผลต่อความคล่องตัวการจราจรทางอากาศ

- จำนวนทางขับ (taxiway)
- จำนวนทางวิ่ง (runway)
- ปริมาณอากาศยาน
- สภาพอากาศ
- สาธารณูปโภคที่รองรับ
- ความชำนาญของ ATC
- เครื่องช่วยเดินอากาศ
- สมรรถนะอากาศยาน

## 2.7 ED 137 หรือ Interoperability Standard for VoIP ATM Components Part 1 Radio

ED 137 คือ กฎสำหรับการใช้งาน VoIP เพื่อสนับสนุนการสื่อสารของ ATM

### 2.7.1 โพรโทคอลพื้นฐานที่จำเป็น (Basic Protocol Requirement)

ซึ่งโพรโทคอลพื้นฐานที่จำเป็นในมาตรฐานนี้มีอยู่ 2 โพรโทคอล ได้แก่

#### 1) Session Initiation Protocol (SIP)

โพรโทคอล SIP คือ โพรโทคอลที่ใช้ในการส่งข้อความควบคุมที่สร้าง และยกเลิก Session การสื่อสารระหว่างหลาย Endpoints บน IP Network

ในการใช้งานจริงนั้นโพรโทคอล SIP ใช้ในการเชื่อมต่อลิงค์ระหว่าง Voice Communication System (VCS) กับอุปกรณ์วิทยุ และประกาศค่าพารามิเตอร์ (Parameters) ที่ใช้ระหว่าง Session นั้น

ซึ่งหลัก ๆ แล้ว โปรแกรม Park Air S4 IP Controller จะทำหน้าที่ในการสร้าง Session โดยใช้โพรโทคอล SIP

#### 2) Real time Transport Protocol (RTP)

โพรโทคอล RTP คือ โพรโทคอลที่ถูกใช้ในการส่งแพ็กเก็ตเสียงแบบ Real time

ในการใช้งานจริงนั้นโพรโทคอล RTP ใช้ในการอนุญาตการส่งแพ็กเก็ตเสียง และสัญญาณวิทยุ โดยใช้ Header extension (ส่วนต่อขยายส่วนหัว) ที่อยู่ในโพรโทคอล RTP

### 2.7.2 Real time Transport Protocol Header Extension

Real time transport Protocol Header Extension จะประกอบไปด้วยข้อมูลของการให้สัญญาณ (Signalling) ที่ถูกส่งมาด้วยแบบ Real time ได้แก่

- PTT (Push To Talk)

การกระทำทางกายภาพที่ถูกกระทำโดยผู้ใช้ ในการดำเนินการส่ง Key (ซึ่งคำว่า Key ในที่นี้หมายถึงรูปแบบ หรือประเภทของเครื่องมือ รวมไปถึงปุ่ม และคันโยก ฯลฯ)

- Aircraft call (A/C call)

Aircraft Call หรือเรียกว่า nominated Squelch (SQL) เป็นคำที่ถูกใช้ที่ Controller Working Position (CWP) บริเวณพื้นที่การทำงานของเจ้าหน้าที่ควบคุมจราจรทางอากาศ หมายถึง การรับสัญญาณที่ส่งมาบนความถี่เฉพาะ

- Quality Index
- Time stamping information
- Keep alive message

เมื่อไม่มีการแลกเปลี่ยนข้อมูลเสียง โพรโทคอล RTP จึงไม่มีความจำเป็นต้องส่งแพ็กเก็ตส่วนเสียง แต่จะส่งแพ็กเก็ต RTP ขนาดเล็กที่ประกอบด้วย Header extension ซึ่งใช้สำหรับการแลกเปลี่ยนการให้สัญญาณที่ได้กล่าวไปข้างต้นเท่านั้น

### 2.7.3 Communication initiation between VCS and Radios

การเริ่มต้นการสื่อสารระหว่าง VCS และวิทยุ แบ่งได้เป็น 2 เฟส (phase) ได้แก่

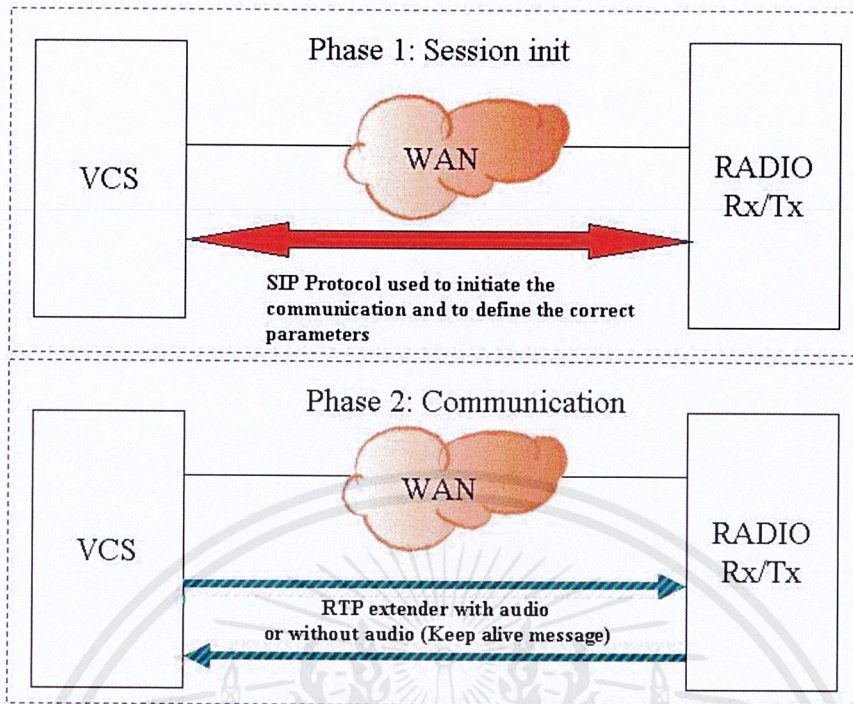
#### 1) เฟส 1 : การเริ่มต้น session

- SIP session จะถูกสร้างจากฝั่ง VCS
- โพรโทคอล SIP จะถูกใช้เพื่อเริ่มการเชื่อมต่อระหว่าง VCS และวิทยุ และเพื่อกำหนดหรือแลกเปลี่ยนพารามิเตอร์ในระหว่าง Session ผ่าน Session Description Protocol (SDP)

#### 2) เฟส 2 : การสื่อสาร

- RTP จะถูกใช้ในการขนส่งเสียง ในขณะที่ RTP header extension จะถูกใช้ในการขนส่งสัญญาณ PTT แบบ real time จาก VCS ไปยังวิทยุ และ A/C Call และส่งสัญญาณคุณภาพ BSS (Best Signal Selection) จากวิทยุสู่ VCS
- นอกจากนี้โพรโทคอล RTP จะถูกใช้เพื่อควบคุมการเชื่อมโยงระหว่าง VCS และวิทยุโดยใช้กลไก “Keep alive”

ซึ่งการเชื่อมต่อของทั้ง 2 เฟสนั้น สามารถแสดงได้ดังรูปที่ 2.12



รูปที่ 2.12 การเชื่อมต่อ SIP ระหว่าง VCS และวิทยุเครื่องรับ/ส่ง (1 เครื่อง) [8]

#### 2.7.4 Voice Coding

เสียงจะถูกเข้ารหัสตามมาตรฐานของ ITU-T แบบ G.711 A-law หรือ  $\mu$ -law G.711 เป็น อัลกอริทึมการรวบรวมขั้นพื้นฐานที่ใช้สำหรับแอปพลิเคชัน (application) การขนส่งเสียงในโลกการสื่อสาร

G.711 ถือเป็นเกณฑ์มาตรฐานสำหรับคุณภาพเสียงที่เข้ารหัสด้วยระบบดิจิทัล ซึ่งมีการวัดการเข้ารหัสด้วยเสียง และขั้นตอน (algorithm) การบีบอัดเสียงอื่น ๆ

ซึ่ง G.711 A-law เป็นมาตรฐานในการเข้ารหัสเสียง สำหรับโครงการนี้

#### 2.7.5 Audio CODECS

CODEC หรือตัวแปลงสัญญาณเสียงที่แสดงในตารางที่ 2.1 ด้านล่างถูกกำหนดโดย RFC 3551 payload type numbers ถูกกำหนดดังแสดงในตารางที่ 2.2 ด้านล่าง

โดยในโครงการนี้จะเลือกใช้ และแสดงรายละเอียดของ PCMA (Pluse Code Modulation A-law) หรือ G.711 A-law และ PCMU (Pluse Code Modulation  $\mu$ -law) หรือ G.711  $\mu$ -law

ตาราง 2.1 คุณสมบัติของ CODEC [8]

Audio CODEC name	Sample/frame	Bits/sample	Sampling rate	Default ms/packet
PCMA	sample	8	8,000	20
PCMU	Sample	8	8,000	20

ตาราง 2.2 Payload types (PT) สำหรับ CODEC [8]

Payload type	Encoding name	Clockrate (Hz)	Channels
0	PCMU	8,000	1
8	PCMA	8,000	1

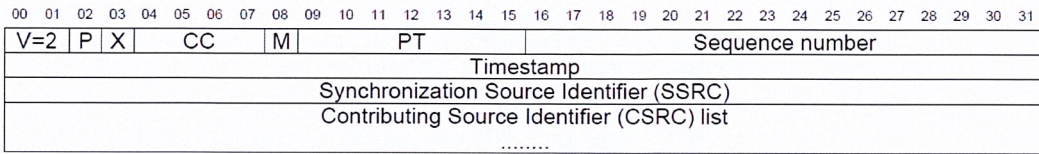
### 2.7.6 RTP Header Specifications

รายละเอียดของ RTP Header มีดังต่อไปนี้

- Version (V) : 필드 (field) นี้ระบุเวอร์ชันของ RTP ซึ่งมีค่าเท่ากับ 2
- Padding-Bit (p) : ค่าจะเท่ากับ 0 ถ้าไม่มีการเพิ่ม octets ที่ท้ายของ telegram
- Extension (X) : บิตนี้จะถูกตั้งค่าเป็น 1 ซึ่งคือ fixed header จะถูกตาม ด้วย Header extension 1 ชุด
- CSRC count (CC) : ประกอบด้วยหมายเลขของตัวระบุ CSRC ตาม Fixed header
- Marker (M) : การตีความ (interpretation) ของ Marker ถูกกำหนดโดยโปรไฟล์ (profile)
- Payload type (PT) : 필드ระบุรูปแบบ Payload type ของ RTP และกำหนดการตีความโดยแอปพลิเคชัน
- Sequence number : จะเพิ่มขึ้นทีละหนึ่ง สำหรับแต่ละแพ็กเก็ตข้อมูล RTP (RTP data packet) ที่ส่ง และอาจถูกใช้โดยผู้รับ เพื่อตรวจหาแพ็กเก็ตที่สูญหาย (Packet loss) และเรียกคืนลำดับแพ็กเก็ต (Packet sequence)
- Time stamp : สะท้อนให้เห็นถึงการสุ่มตัวอย่างทันทีของ octet แรกใน แพ็กเก็ตข้อมูล RTP
- SSRC : 필드ระบุแหล่งที่มาของการซิงโครไนซ์ (synchronization) และเลือกสุ่ม
- CSRC : 필드ระบุแหล่งที่มาสำหรับ payload ที่บรรจุในแพ็กเก็ต

12 octets แรกนั้นมีอยู่ในทุก ๆ แพ็กเก็ต RTP ในขณะที่รายการของตัวระบุ CSRS นั้นจะปรากฏเฉพาะเมื่อมีการแทรกโดย mixer และฟิลด์ CC ที่ถูกตั้งค่าไว้สอดคล้องกันเท่านั้น โดยโครงสร้างของแพ็กเก็ต RTP header แสดงดังรูปที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและ 21 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 RTP header [8]

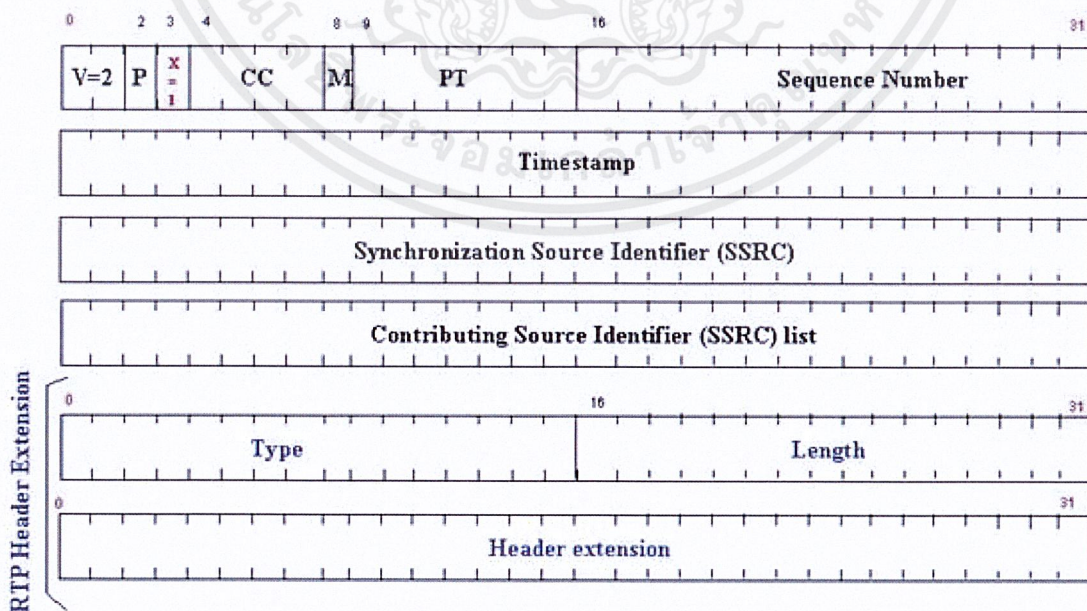
### 2.7.7 คำอธิบาย RTP Header Extension

RTP header extension ใช้สำหรับการส่งข้อมูลเพิ่มเติมที่จำเป็นสำหรับการสื่อสารวิทยุ (ตัวอย่างเช่น สัญญาณ PTT หรือ SQL ฯลฯ)

เมื่อบิต X ถูกตั้งค่าเป็น 1 ตัวแปร Length-header extension จะถูกเพิ่มต่อท้ายไปยัง RTP header

Header extension มีความยาว 16 บิต ที่ระบุจำนวน 32-bit words ไม่รวมส่วน 4 octet ของ Header extension

เพื่อให้การใช้งานระหว่างปฏิบัติการหลายครั้ง กับการทดสอบแต่ละครั้งอย่างอิสระพร้อมกับส่วนขยายส่วนหัวที่แตกต่างกัน 16 บิตแรกของ Header extension จะถูกเปิดทิ้งไว้ เพื่อแยกตัวระบุ หรือ พารามิเตอร์ รูปแบบของบิตทั้ง 16 บิตเหล่านี้จะถูกกำหนดโดยข้อกำหนดคุณสมบัติของการนำไปใช้งาน แสดงดังรูปที่ 2.14

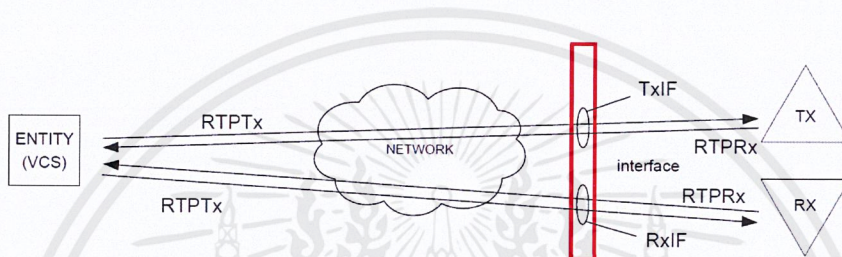


รูปที่ 2.14 RTP header extension [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟิลด์ Type ควรระบุรูปแบบ Header extension เฉพาะ สำหรับแอปพลิเคชันฟิลด์ Type จะเป็นค่า 0x67h รายการประเภท Header extension พร้อมกับรูปแบบของ Extension นั้นจะต้องได้รับการคงไว้ เพื่อสร้างแอปพลิเคชันที่แตกต่างกัน โดยมีรูปแบบ Header extension ที่ต่างกัน

วิธีนี้ขึ้นอยู่กับการใช้ RTP extension เพื่อรวมการส่งสัญญาณแบบ Real time ทั้งหมดเช่น PTT SQL และการรับดัชนีคุณภาพสัญญาณ ข้อมูลการให้สัญญาณวิทยุที่ส่งใน RTP header ของสตรีมเสียง (Audio stream) จะถูกนำไปใช้กับอินเทอร์เฟซ (Interface) ระหว่างวิทยุ (ตัวรับสัญญาณเครื่องส่งสัญญาณหรือตัวรับส่งสัญญาณ) และเอนทิตีกราวด์ (Ground entity) เช่น VCS ดังแสดงในรูปที่ 2.15



รูปที่ 2.15 Topology ของระบบวิทยุพื้นฐาน [8]

จากรูปที่ 2.15 ปรากฏอินเทอร์เฟซ 2 ประเภทที่ต่างกัน คือ

- TxIF : อินเทอร์เฟซระหว่างเอนทิตีกราวด์ และวิทยุเครื่องส่ง
- RxIF : อินเทอร์เฟซระหว่างเอนทิตีกราวด์ และวิทยุเครื่องรับ

ดังนั้นจาก RTP header extension บอกได้ว่า

- RTPTx RTP extension สำหรับการให้สัญญาณแบบ Real time จากเอนทิตีกราวด์ (เช่น VCS) ไปยังวิทยุเครื่องส่ง วิทยุรับ-ส่ง หรือวิทยุเครื่องรับ มี 2 กรณีที่เกิดขึ้นได้ คือ
  - RTPTx ส่งแพ็กเก็ตที่มีแพ็กเก็ตเสียง
  - RTPTx ส่งแพ็กเก็ตที่ไม่มีแพ็กเก็ตเสียงระหว่างส่งข้อความ Keep alive (ในหัวข้อ 2.7.8.1)
- RTPRx RTP Extension สำหรับการให้สัญญาณแบบ Real time จากวิทยุเครื่องรับ วิทยุรับ-ส่ง หรือ วิทยุเครื่องส่ง ไปยังเอนทิตีกราวด์ มี 2 กรณีที่เกิดขึ้นได้ คือ
  - RTPRx ส่งแพ็กเก็ตที่มีแพ็กเก็ตเสียงประกอบด้วย
  - RTPRx ส่งแพ็กเก็ตที่ไม่มีแพ็กเก็ตเสียงระหว่างส่งข้อความ Keep alive (ในหัวข้อ 2.7.8.1)



- Priority PTT : จากผู้ใช้ที่มีสิทธิ์ในการลบการส่งสัญญาณที่ดำเนินอยู่
- Emergency PTT : PTT ที่มีลำดับความสำคัญสูงสุดจะไม่ถูกขัดจังหวะโดย PTT อื่น
- SQU (1 bit : b3) : ไม่ใช่ใน RTPTx จึงถูกตั้งค่าเป็น 0
- PTT-ID (4 bits : b4 to b7) : PTT-ID ของอุปกรณ์ส่งสัญญาณซึ่ง ID นี้ ถูกกำหนดโดยวิทยุส่งสัญญาณ และจะถูกกำหนดระหว่าง SIP session หากไม่มีการกำหนด PTT-ID ให้ใช้ค่า PTT-ID = 0
- SCT (1 bit : b8) - Simultaneous transmission : ไม่ใช่ใน RTPTx ถูกตั้งค่าเป็น 0
- X (1 bit : b9) : บิต marker ซึ่งจะถูกรับค่าเป็น 1 หากมีการขยายข้อมูลเพิ่มเติมเมื่อคุณลักษณะเพิ่มเติมถูกใช้

ตาราง 2.3 รายการของ PTT Value [8]

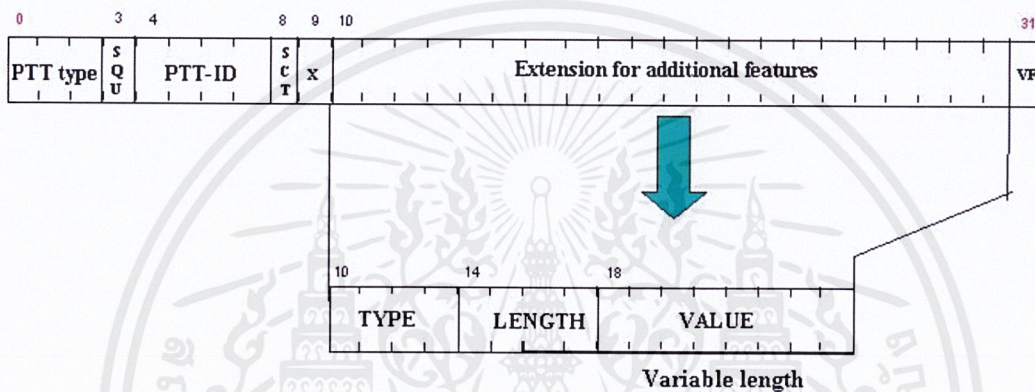
PTT Value	Description
0x00	PTT OFF
0x01	Normal PTT ON
0x02	Coupling PTT ON
0x03	Priority PTT ON
0x04	Emergency PTT ON
0x05	Reserved
0x06	Reserved
0x07	Reserved

- VF (1 bit : b31) : บิต Visibility Flag marker ซึ่งจะถูกรับค่าเป็น 1 เมื่อ end point ได้รับแพ็กเก็ต RTP จากต้นทาง (originator)
- Extension for additional features : ข้อมูลในบล็อกคุณลักษณะเพิ่มเติมจะถูกเข้ารหัสในรูปแบบ Type-Length-Value (TLV) สำหรับ backward compatibility จะต้องมีการกำหนดค่าที่เหมาะสมภายในส่วนบังคับ แม้ว่าจะมีข้อมูลซ้ำซ้อนในเนื้อหาคุณลักษณะเพิ่มเติมก็ตาม เมื่อไม่ได้ใช้ฟิลด์ Extension จะถูกตั้งค่าเป็น 0
  - Type (4 bit) : ฟิลด์ Type ใช้เพื่อระบุเนื้อหาคุณลักษณะภายในการตั้งค่า session ที่ได้รับการสนับสนุนจากแต่ละ entity (เช่น หมายเลข

ประเภทเพย์โหลดของ RTP เพื่อแยกความแตกต่างระหว่างตัวแปลงสัญญาณ)

- Length (4 bit) : ฟিলด์ Length กำหนดความยาวของฟিলด์ Value ในหน่วยบิต ดังนั้นความยาวสูงสุดเท่ากับ 13 บิต
- Value (Variable length) : บล็อกนี้จะนำส่งข้อมูลเกี่ยวกับคุณลักษณะเพิ่มเติม

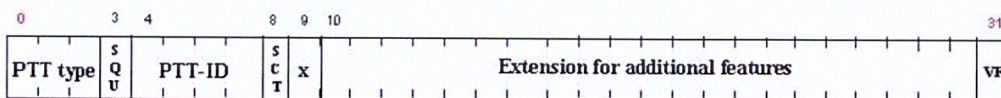
รูปที่ 2.17 แสดงตัวอย่างของวิธีการเข้ารหัสข้อมูลภายในบล็อกคุณสมบัติเพิ่มเติม



รูปที่ 2.17 การแบ่งส่วน Extension for additional features [8]

### 2.7.7.2 ฟิลด์ข้อมูล RTPRx

ฟิลด์ข้อมูล RTPRx ที่บรรจุภายใน Header extension ของสตรีม RTP จากตัวรับ โดยจะไม่แสดงประเภท Extension และฟิลด์ความยาว ดังรูปที่ 2.18



รูปที่ 2.18 ฟิลด์ข้อมูลของ RTPRx [8]

เพื่อให้แน่ใจว่า RTP header สิ้นสุดที่ขอบเขต 32 บิต ส่วน Extension น้อยที่สุดที่เป็นไปได้ คือ 4 ไบต์ โดยที่บิต 0 ถึง 15 จะแบ่งออกเป็นตามฟิลด์ของ โพรโทคอล ดังต่อไปนี้

- PTT-type (3 bits : b0 to b2) : Operator Position (Client) แจ้งผู้ฟัง (Listeners) หรือ Client ให้ทราบเกี่ยวกับเงื่อนไข trigger (เช่น PTT type) ของการส่ง พิลด์นี้อาจเป็นตัวเลือกใน RTPRx directionเนื่องจากข้อมูลไม่ได้ Available ด้านวิทยุเครื่องรับเสมอ

ในกรณีที่มี Aircraft Call (A/C) ค่า PTT type จะถูกตั้งค่าเป็น PTT OFF หากไม่ได้ใช้พิลด์นี้จะตั้งค่าเป็น 0

รายการ PTT type ต่าง ๆ และรายละเอียดของ RTPRx เหมือนกับของ RTPTx ในหัวข้อ 2.7.7.1

- SQU (1 bit : b3) : การระบุ Squelch จากวิทยุ : พิลด์นี้จะใช้เพื่อระบุการ A/C call (เช่น การกระทำ Squelch) ไปยังระบบภายนอก ค่าที่เป็นไปได้ถูกกำหนดไว้ในตารางที่ 2.4

ตาราง 2.4 บิตข้อมูล Squelch [8]

SQU	Description
0x00	SQ OFF
0x01	SQ ON

- PTT-ID (4 bits : b4 to b7) : PTT-ID ของอุปกรณ์ส่งสัญญาณ ID นี้ถูกกำหนดโดยวิทยุส่งสัญญาณและจะถูกกำหนดในระหว่าง SIP session หากไม่มีการกำหนด PTT-ID ให้ใช้ค่า PTT-ID = 0
- SCT (1 bit : b8) : Simultaneous transmission บิตจะถูกตั้งค่าเป็น SCT = 1 หากตรวจพบการส่งสัญญาณพร้อมกัน ค่าจะถูกตั้งค่าในแต่ละแพ็กเก็ต RTP トラフィックที่มีการส่งพร้อมกัน
- X (1 bit : b9) : บิต marker ซึ่งจะถูกตั้งค่าเป็น 1 หากใช้พิลด์ข้อมูลเสริม
- VF (1 bit : b31) : บิต Visibility Flag marker ซึ่งจะถูกตั้งค่าเป็น 1 เมื่อ end point ได้รับแพ็กเก็ต RTP จากต้นทาง
- Extension for additional features : เช่นเดียวกับในพิลด์ข้อมูล RTPTx

## 2.7.8 Real Time Session Supervision

หากไม่มีกลไกการตรวจสอบปลายทาง Session ที่ไม่ได้รับแพ็กเก็ตใด ๆ จะไม่สามารถแยกแยะความแตกต่างระหว่างกรณีเมื่อไม่มีแพ็กเก็ตที่ออกโดยผู้ส่ง และกรณีเมื่อแพ็กเก็ตที่ออกโดยผู้ส่งสูญหายไประหว่างการเดินทาง

The Real-time Session Supervision เรียกสั้น ๆ ว่า R2S มาพร้อมกับการสร้างแต่ละ SIP session เพื่อให้ปลายทางตระหนักถึงความสามารถในการเชื่อมต่อในระดับเดียวกัน (peer)

กลไกที่ใช้ในการตรวจจับข้อบกพร่อง คือ ใช้การแลกเปลี่ยนข้อความการให้สัญญาณเป็นระยะขึ้นอยู่กับ "การมองเห็น" ของข้อความ Hello ที่ได้รับ

คุณลักษณะที่โดดเด่นของข้อความการให้สัญญาณ R2S คือสิ่งเหล่านี้ใช้ของ (envelope) เดียวกันกับสตรีมเสียงที่เชื่อมโยงกับ Session นั้นจนถึงเลเยอร์ 3 (Layer 3) ที่ครอบคลุม ดังนั้นจึงไม่สามารถแยกออกได้จากอีกมุมมองเครือข่ายอื่น

### 2.7.8.1 R2S Keep Alive Signalling Operation

[RTP] Keep Alive messages

ข้อความการให้สัญญาณ Keep alive เป็นแพ็กเก็ต RTP และมีรายละเอียดดังต่อไปนี้

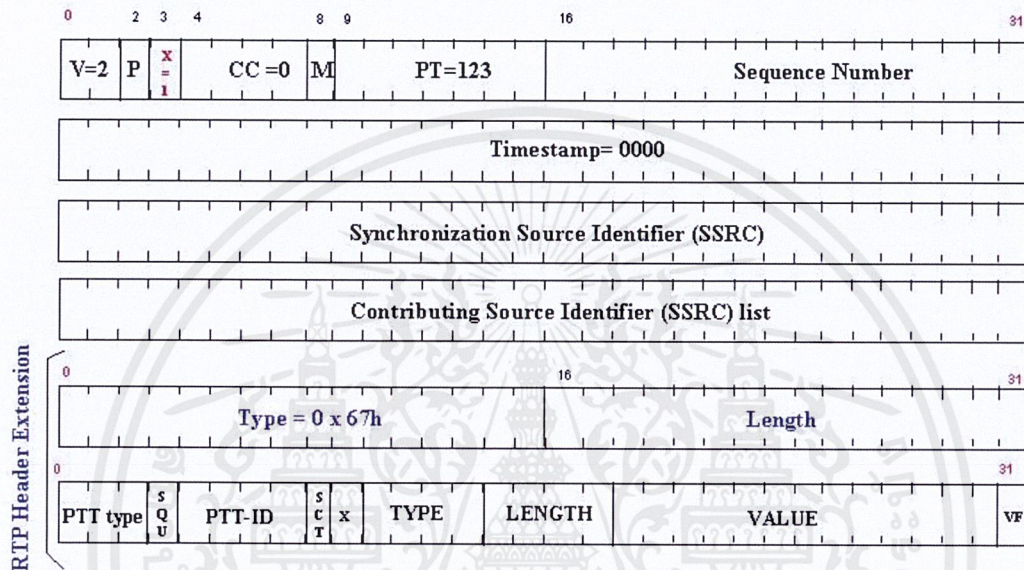
- X = 1
- PT = 123 (ไดนามิก) ไม่มีการรายงาน RTCP
- CC = 0
- Sequence Number
- Timestamp = 0x00000000 ( null เสมอ)
- SSRC identifier : จะแตกต่างจากแพ็กเก็ตเสียง ค่าที่ต้องการแต่ไม่บังคับ = 0x55555555 หรือ 0xAAAAAAAA
  - 0x55555555 สำหรับข้อความที่สร้างโดย Endpoint ที่มี IP Address ต่ำกว่า
  - 0xAAAAAAAA สำหรับข้อความที่สร้างขึ้นโดย Endpoint ที่มี IP Address สูงกว่า
  - IP Address และ Transport Address นั้นเหมือนกันในแพ็กเก็ตเสียง RTP สร้างความมั่นใจสำหรับการให้สัญญาณข้อความ และสำหรับแพ็กเก็ตเสียงในระดับคุณสมบัติเดียวกันเมื่อทำการส่งผ่าน
- Type = 0x67h ฟิ ลด์ type ควรระบุรูปแบบ Header extension เฉพาะแอปพลิเคชัน

โครงสร้างของ Keep Alive message แสดงดังรูปที่ 2.19

เอกสารนี้เป็นเอกสารทงสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยนาให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาแ่ 28 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์ที่ถูกเจรจาระหว่าง SDP :

- R2S-KeepalivePeriod : ค่าในหน่วย มิลลิวินาที (millisecond หรือ ms) ของช่วงเวลาระหว่างข้อความ RTP ที่ไม่มีเสียง
- R2S-KeepaliveMultiplier : ค่าที่คูณด้วยR2S-KeepalivePeriod เพื่อระบุเวลาที่ R2S-Local



รูปที่ 2.19 โครงสร้างของ Keep alive message [8]

ค่าเริ่มต้นสำหรับการจับเวลา R2S-LocalHoldtime ใน session เริ่มต้นขึ้นเท่ากับมูลค่าของ R2S-KeepalivePeriod คูณกับค่าของ R2S-KeepaliveMultiplier

- กฎในการส่งข้อความการให้สัญญาณ :

ข้อความการส่งสัญญาณจะถูกส่งด้วยจังหวะที่ระบุโดยค่า R2S-KeepalivePeriod เมื่อเสียงแสดงข้อความการส่งสัญญาณจะถูกส่งเป็นภายใน RTP header extension ในขณะที่เสียงจะถูกส่งเป็นเพย์โหลด

- กฎสำหรับการรีเซ็ตการตั้งค่าสถานะการมองเห็น SIP session ที่ Tearing down และการตั้งค่าอีกครั้ง :

เมื่อได้รับข้อความการให้สัญญาณจากปลายทางที่สอดคล้องกัน โดยมีค่าการตั้งค่าสถานะการมองเห็น (VF) เป็น 1 ตัวจับเวลา Local R2S-LocalHoldtime จะถูกกำหนดค่าเริ่มต้น ซึ่ง R2S-LocalHoldtime ถูกกำหนดค่าเริ่มต้นด้วยค่าของ R2S-KeepaliveMultiplier

ตัวจับเวลา R2S-LocalHoldtime จะถูกลดขนาด โดยทุก ๆ หน่วยของข้อความ R2S-KeepalivePeriod ที่ส่งโดยระบบ トラバドที่ค่าสถานการณ์มองเห็นที่ได้รับจาก Endpoint อื่นถูกรีเซ็ต (Reset) หรือトラバドที่ไม่มีข้อความกลับมาจาก Endpoint

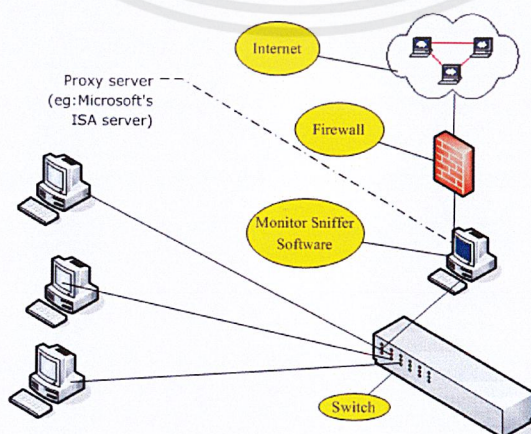
เมื่อ R2S-LocalHoldtime ลดมาถึง 0 SIP session จะถูกนำออกใช้ โดยการส่งข้อความ SIP BYE ซึ่ง SIP session จะยังคงอยู่ในสถานะ Unacknowledged จนกว่าตัวจับเวลา SIP stack จะหมดอายุ หรือจนกว่าข้อความ Acknowledge จะมาถึง

SIP initiator จะพยายามสร้าง Session ใหม่ทันที หลังจากข้อความ BYE แรกของ Session ก่อนหน้าถูกส่ง

## 2.8 โปรแกรม Sniffer

โปรแกรม Sniffer คือโปรแกรมที่เอาไว้ดักจับข้อมูลบนระบบโครงข่าย (Network) เนื่องจากคอมพิวเตอร์เน็ตเวิร์คเป็นระบบการสื่อสารที่ใช้ร่วมกัน เพื่อประหยัดค่าใช้จ่าย การแบ่งกันใช้ (Sharing) หมายถึงคอมพิวเตอร์สามารถรับข้อมูลที่คอมพิวเตอร์เครื่องอื่นตั้งใจจะส่งไปให้อีกเครื่องหนึ่ง การดักจับข้อมูลที่ผ่านไปมาระหว่างโครงข่ายเรียกว่า Sniffing ซึ่งคล้ายกับการดักฟังโทรศัพท์ แตกต่างกันที่การดักฟังโทรศัพท์จะทำได้ทีละเครื่อง แต่ Sniffer สามารถดักข้อมูลได้ที่เดียวทั้งโครงข่าย ตัวอย่างของการใช้ Sniffer ในโครงข่าย ซึ่งเป็นการดักจับข้อมูลผ่าน Port ของสวิตช์ (Switch) แสดงดังรูปที่ 2.20

แรกเริ่มโปรแกรม Sniffer เป็นชื่อโปรแกรมของบริษัท Network Associates Inc. สหรัฐฯ เพื่อใช้ในผลิตภัณฑ์ของตนเองในเครือ Sniffer Network Analyzer ซึ่งเป็นโปรแกรมวิเคราะห์โครงข่าย โดยอาศัยการดักอ่านข้อมูล แรกเริ่มโปรแกรม sniffer ถูกสร้างมาเพื่อการป้องกัน ซึ่งคือเอามาตรวจสิ่งที่วิ่งอยู่บนโครงข่าย แต่เมื่อมีผู้ประสงค์ไม่ดีนำโปรแกรมที่ใช้ป้องกัน ไปทำลาย จึงก่อให้เกิดอันตราย หากนำมาใช้ในการตรวจระบบโครงข่าย ก็เป็นประโยชน์ แต่หากนำเอาไปใช้ในการดักอ่านของข้อมูลเพื่อจุดประสงค์ที่ไม่ดี เช่น เอาไปวางไว้บนทางของระบบ E-mail ก็ทำให้ข้อมูลทั้งหมดของ E-mail โดนอ่าน ในขณะที่ควรเป็นความลับ



เอกสารนี้เป็นเอกสารที่ **รูปที่ 2.20** ตัวอย่างการใช้โปรแกรม Sniffer ในโครงข่าย [9] ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## 2.9 โปรแกรม Wireshark

Wireshark (โปรแกรมวิเคราะห์ข้อมูลโครงข่าย ฟรี) : โปรแกรมนี้มีชื่อว่า โปรแกรม Wireshark เป็นอีกหนึ่งโปรแกรมที่เอาไว้ใช้ในการวิเคราะห์ข้อมูลบนโครงข่าย (Network Analyzer) ชื่อระดับโลก ซึ่งเป็นที่นิยมมากทั่วโลก และเป็นโปรเจกแบบ OpenSource ซึ่งพัฒนาโดยโปรแกรมเมอร์อาสาสมัครกว่า 100 คนจากทั่วโลก ที่ถูกเริ่มต้นพัฒนาขึ้นมาตั้งแต่ปี ค.ศ.1998 (พ.ศ. 2541) ซึ่งโปรแกรมนี้สามารถวิเคราะห์ข้อมูลบนโครงข่ายได้หลากหลายรูปแบบ สนับสนุนโพรโทคอล หลากหลายมาตรฐาน และมีการพัฒนาอยู่ตลอดเวลา นอกจากนี้ยังสามารถดูปริมาณ และประเภทของข้อมูลที่เข้าออกได้แบบ Real time และดูย้อนหลังได้แบบออฟไลน์ (Offline) ซึ่งสามารถสามารถทำงานได้บนหลากหลายแพลตฟอร์ม ได้แก่ ระบบปฏิบัติการ Windows Linux MacOSX Solaris FreeBSD NetBSD และอื่น ๆ อีกมากมาย

นอกจากนี้แล้ว โปรแกรมวิเคราะห์ข้อมูลโครงข่าย WireShark ก็ยังมีข้อมูลเชิงลึก ด้านเครือข่าย เน็ตเวิร์คต่าง ๆ มากมาย ดังนั้นจึงเหมาะสำหรับผู้ดูแลระบบ (System Administrator) ที่มีความรู้ด้านนี้โดยตรง นอกจากนี้ยังสามารถที่จะแสดงข้อมูลได้หลากหลายประเภท ไม่ว่าจะเป็นในส่วนที่มาของข้อมูล (Source) ข้อมูลปลายทาง (Destination) ประเภทโพรโทคอล มาตรฐานการรับส่งของข้อมูล (Protocol) และ รายละเอียดเชิงลึกต่าง ๆ ซึ่งข้อมูลทั้งหมดนี้ สามารถกำหนดเงื่อนไข ฟิลเตอร์ข้อมูลได้เอง พร้อมออกรายงานได้หลากหลาย

คุณลักษณะเด่นของโปรแกรม Wireshark ที่ทำให้ผู้จัดทำเลือกใช้ คือ เป็นโปรแกรมที่ทำหน้าที่ดักจับข้อมูล หรือแพ็กเก็ตที่วิ่งบนโครงข่ายได้ และสามารถนำแพ็กเก็ตนั้นมาตรวจสอบ และวิเคราะห์รายละเอียดในส่วนต่าง ๆ ของแพ็กเก็ตได้ ตราสัญลักษณ์ของโปรแกรม Wireshark แสดงได้ดังรูปที่ 2.21

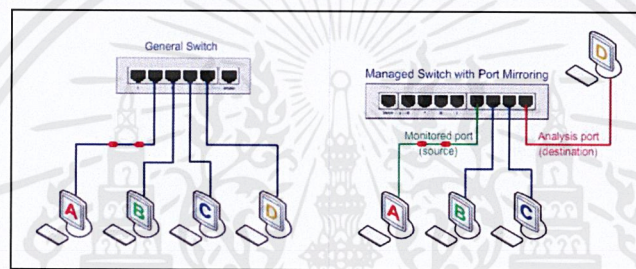


รูปที่ 2.21 ตราสัญลักษณ์โปรแกรม Wireshark [10]

## 2.10 Port mirroring

Port mirroring หรือที่รู้จักในชื่อ SPAN (Switched Port Analyzer) เป็นวิธีการตรวจสอบการใช้โครงข่าย เมื่อเปิดใช้งาน Port mirroring สวิตช์จะส่งสำเนาของแพ็กเก็ตโครงข่ายทั้งหมดที่เห็นบน Port หนึ่ง (หรือ VLAN ทั้งหมด) ไปยัง Port อื่น ซึ่งสามารถวิเคราะห์แพ็กเก็ตได้

ฟังก์ชัน Port mirroring รองรับโดยสวิตช์ระดับองค์กรเกือบทั้งหมด (สวิตช์จัดการ) ฟังก์ชัน Port mirroring อธิบายได้ดีที่สุด เมื่อเปรียบเทียบกับสวิตช์ปกติ ในรูปที่ 2.22 (ซ้าย) กับสวิตช์ที่มีการทำ Port mirroring ในรูปที่ 2.22 (ขวา) โดยนับเรียง Port จากด้านซ้ายสุดเป็น Port ที่ 1 ไปด้านขวาสุดเป็น Port ที่ 6 ตามลำดับ ดังรูปที่ 2.22



รูปที่ 2.22 การส่งแพ็กเก็ตของสวิตช์ปกติ กับสวิตช์ที่มีการทำ Port mirroring [11]

จากรูปที่ 2.22 ในกรณีของสวิตช์ปกติ นั้น สวิตช์จะเก็บข้อมูลของพอร์ตที่เชื่อมต่ออยู่กับคอมพิวเตอร์นั้น ๆ เพราะฉะนั้นสวิตช์จะรู้เพียงแค่ว่า

Port ที่ 1 (ด้านซ้ายสุด) ไม่มีการเชื่อมต่อ

Port ที่ 2 เชื่อมต่อกับคอมพิวเตอร์เครื่อง A

Port ที่ 3 เชื่อมต่อกับคอมพิวเตอร์เครื่อง B

Port ที่ 4 เชื่อมต่อกับคอมพิวเตอร์เครื่อง C

Port ที่ 5 เชื่อมต่อกับคอมพิวเตอร์เครื่อง D

Port ที่ 6 (ด้านขวาสุด) ไม่มีการเชื่อมต่อ

เมื่อสวิตช์รับแพ็กเก็ตจากคอมพิวเตอร์ A ไปยัง B เราเตอร์จะส่งแพ็กเก็ตนี้ไปยัง Port ที่ 3 (เนื่องจากคอมพิวเตอร์ B เชื่อมต่อกับ Port ที่ 3) แสดงดังรูปที่ 2.23 (ซ้าย)

คอมพิวเตอร์ตัวอื่น (คอมพิวเตอร์ C และ D) จะไม่เห็นการรับ-ส่งข้อมูลในโครงข่ายนั้น

สรุป คือ สวิตช์ปกติ การรับ-ส่งข้อมูลจะสามารถเห็นได้เฉพาะคอมพิวเตอร์ที่มีการรับ-ส่งข้อมูลกัน คอมพิวเตอร์เครื่องอื่นจะไม่เห็นการรับ-ส่งข้อมูล ที่ไม่ได้กำหนดไว้สำหรับคอมพิวเตอร์นั้น

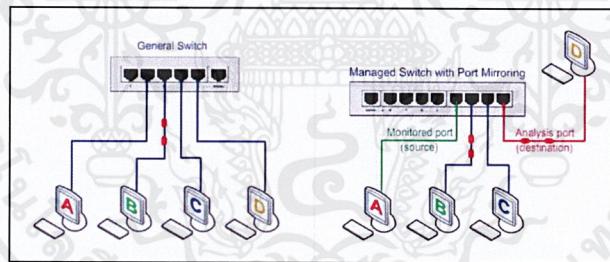
ในกรณีของสวิตช์ที่มี Port ทำหน้าที่เป็น Port mirroring ซึ่งผู้ดูแลระบบได้มีการกำหนดค่าสวิตช์ให้ส่งสำเนาทุกแพ็กเก็ตที่ถูกส่งระหว่างคอมพิวเตอร์ A และ B ไปยังคอมพิวเตอร์ D แสดงดังรูปที่ 2.23 (ขวา)

คอมพิวเตอร์ D เป็นผู้ที่ฟังการรับ-ส่งข้อมูลของโครงข่ายนั้น และสามารถใช้สำหรับบันทึกการโทรหากเรามีโทรศัพท์ IP แทนคอมพิวเตอร์ A และ B

สรุป คือ Port mirroring อนุญาตให้คอมพิวเตอร์เครื่องใดเครื่องหนึ่งเห็นการรับ-ส่งข้อมูลเครือข่าย ซึ่งปกติจะซ่อนอยู่

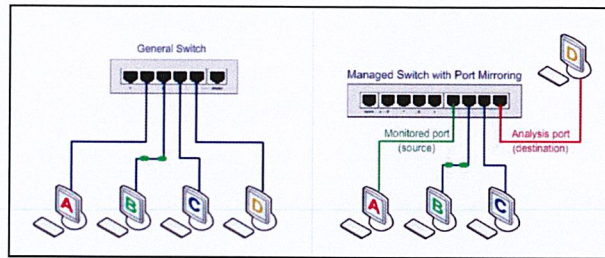
ตัวอย่างการเปรียบเทียบการส่งแพ็กเก็ตของสวิตช์ปกติ กับสวิตช์ที่ทำเป็น Port mirroring แสดงได้ดังรูปที่ 2.23-2.25

จากรูปที่ 2.22-2.25 จุดสีแดง หมายถึงแพ็กเก็ตถูกส่งจากคอมพิวเตอร์เครื่อง A โดยมีจุดประสงค์คือ ส่งไปให้คอมพิวเตอร์เครื่อง B และจุดสีเขียว หมายถึงแพ็กเก็ตถูกส่งจากคอมพิวเตอร์เครื่อง B โดยมีจุดประสงค์ คือ ส่งกลับไปให้คอมพิวเตอร์เครื่อง A



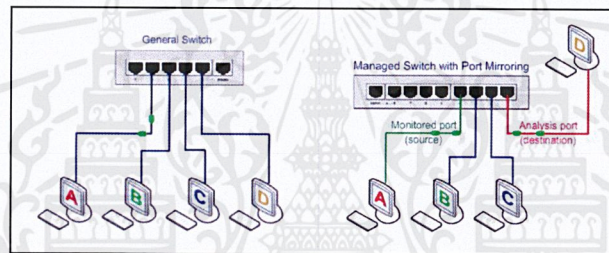
รูปที่ 2.23 การวิ่งของแพ็กเก็ตของสวิตช์ปกติ กับสวิตช์ที่มี Port mirroring รูปที่ 1 [11]

จากรูปที่ 2.23 เป็นรูปของแพ็กเก็ตข้อมูลถูกส่งไปยังคอมพิวเตอร์ B เครื่องเดียว (ซ้าย) และแพ็กเก็ตข้อมูลถูกส่งไปทั้งเครื่อง B และ D ที่เป็น Port mirroring (ขวา)



รูปที่ 2.24 การวิ่งของแพ็กเก็ตของสวิตช์ปกติ กับสวิตช์ที่มี Port mirroring รูปที่ 2 [11]

จากรูปที่ 2.24 คอมพิวเตอร์เครื่อง B พยายามส่งแพ็กเก็ตข้อมูลตอบไปยังคอมพิวเตอร์ A (ทั้งด้านซ้าย และขวา)



รูปที่ 2.25 การวิ่งของแพ็กเก็ตของสวิตช์ปกติ กับสวิตช์ที่มี Port mirroring รูปที่ 3 [11]

จากรูปที่ 2.25 เป็นรูปของแพ็กเก็ตข้อมูลถูกส่งตอบไปยังคอมพิวเตอร์ A เครื่องเดียว (ซ้าย) และแพ็กเก็ตข้อมูลถูกส่งตอบไปทั้งเครื่อง A และ D ที่เป็น Port mirroring (ขวา)

ในโครงการนี้ผู้จัดทำได้ทำการใช้คุณสมบัติของ Port mirroring เพื่อต่อสาย LAN เพียง 1 เส้นจาก Port mirroring ของสวิตช์ที่มีการเชื่อมต่อกับวิทยุเครื่องรับ-ส่งจำนวนมาก เนื่องจากต้องการสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินทางอากาศ โดยนำสำเนาแพ็กเก็ตเสียงที่ได้รับจาก Port mirroring มาวิเคราะห์ต่อไป

## 2.11 มาตรฐานเสียง G.711

G.711 เป็นโคเดก (CODEC) ที่ใช้การเข้ารหัส และถอดรหัสสัญญาณเสียงที่มีขนาด 64 kbps โดยจะไม่มีการบีบอัดสัญญาณเสียง และมีการใช้งานซีพียูในการเข้ารหัสและถอดรหัสน้อยมากจึงทำให้คุณภาพที่ได้มานั้นคุณภาพดีแต่จะใช้งานแบนด์วิดท์ที่มากกว่าโคเดกชนิดอื่น ๆ โดยปกติแล้วมาตรฐาน G.711 นั้นจะแบ่งออกเป็นอีก 2 มาตรฐานย่อยคือ A-law หรือ  $\mu$ -law

โดยที่ G.711 A-law นั้นจะใช้ในยุโรป (Europe) ส่วน G.711  $\mu$ -law นั้นจะใช้ในสหรัฐอเมริกา ซึ่งทั้งสองมาตรฐานก็ต้องการแบนด์วิดท์ (Bandwidth) ที่ 64 kbps โดยทั่วไปแล้วอุปกรณ์ที่ใช้งานในระบบ VoIP นั้นจะรองรับทั้งสองมาตรฐานนี้เป็นหลัก ไม่ว่าจะใช้อุปกรณ์ที่เป็นโทรศัพท์แบบ IP ทั้งฮาร์ดแวร์ และซอฟต์แวร์ รวมถึงอุปกรณ์แปลงสัญญาณเสียงอย่าง ATA ก็รองรับด้วยเช่นกัน หากมีการนำโคเดกนี้ไปใช้งานกับการสื่อสารผ่านทาง Dial up ที่มีช่องสัญญาณเพียง 56 kbps อาจจะทำให้คุณภาพเสียงออกมาไม่ดีนัก เสียงจะขาด ๆ หาย ๆ ได้ เนื่องจากช่องสัญญาณที่ใช้ในการสื่อสารมีขนาดเล็กกว่าความต้องการของมาตรฐานนี้ [12]

### 2.11.1 การเข้ารหัส/ถอดรหัสมาตรฐานเสียง G.711

การมอดูเลตรหัสพัลส์เป็นวิธีการเข้ารหัสรูปแบบ Time-Domain Waveform และกำหนดไว้ใน CCITT G.711 และ AT&T 43801 โดยทั่วไปสัญญาณแอนะล็อกจะถูกสุ่มตัวอย่างในอัตรา 8000 ครั้งต่อวินาที ในแต่ละตัวอย่างแอมพลิจูดของสัญญาณจะถูกกำหนด (Quantized) เป็นค่าดิจิทัล

สำหรับระบบเชิงเส้นที่แท้จริงค่าของ “ระดับ” (Step) แต่ละอันนั้นเหมือนกัน ซึ่งต้องมีตั้งแต่ 11 ถึง 14 บิตในการส่ง อย่างไรก็ตาม PCM ตามที่กำหนด และเป็นมาตรฐานใช้เครื่องซิงโครไนซ์ในการถ่วงน้ำหนักของแต่ละขั้นตอน

มีอัลกอริทึมการมอดูเลตรหัสพัลส์ 2 แบบที่กำหนดไว้ใน CCITT G.711 ในทั้ง A-law และ  $\mu$ -law ค่าที่ใช้เพื่อเป็นตัวแทนของแอมพลิจูด คือ ตัวเลขระหว่าง 0 ถึง  $\pm 127$  ดังนั้นจำเป็นต้องใช้ 8 บิตเพื่อแสดงตัวอย่างแต่ละตัวอย่าง (2 ยกกำลัง 8 = 256)

จะเห็นได้ว่าการมอดูเลตรหัสพัลส์ทำงานในอัตรา (8 บิต/ตัวอย่าง)  $\times$  (8000 ตัวอย่าง/วินาที) = 64000 (64k) บิต/วินาที

ตารางที่ 2.5 ต่อไปนี้ระบุรูปแบบบิต/ไบต์ที่ส่ง สำหรับค่าตัวเลขโดยเฉพาะ

ตารางที่ 2.5 การแทนค่าตัวเลขด้วยรูปแบบบิต/ไบต์ [13]

Numerical Value	Bit Number	
	$\mu$ -law	A-law
	12345678	12345678
+127	10000000	11111111
+96	10011111	11100000
+64	10111111	11000000
+32	11011111	10100000
0	11111111	10000000
0	01111111	00000000
-32	01011111	00100000
-64	00111111	01000000
-96	00011111	01100000
-126	00000001	01111110
-127	00000000	01111111

บิตซ้ายสุด (บิต 1) จะถูกส่งก่อน และเป็นบิตที่สำคัญที่สุด (MSB) บิตนี้เรียกว่า บิตเครื่องหมาย และเป็น 1 สำหรับค่าบวก และเป็น 0 สำหรับค่าลบ (ทั้ง A-law และ  $\mu$ -law) โดยที่บิต 2 ถึง 8 จะกลับหัวระหว่าง A-law และ  $\mu$ -law

## 2.12 ฟิลเตอร์ C-message weighting

ฟิลเตอร์ CCITT และฟิลเตอร์ C-message weighting เป็นฟิลเตอร์แถบความถี่ผ่าน (Bandpass filter) ที่ใช้ในการวัดเสียงรบกวนความถี่เสียงในวงจรโทรศัพท์ (Telephone circuits)

ฟิลเตอร์ CCITT (ITU-T) ใช้สำหรับวงจรโทรศัพท์ระหว่างประเทศ โดยทั่วไปฟิลเตอร์ C-message weighting จะใช้สำหรับวงจรโทรศัพท์ในอเมริกาเหนือ

ซึ่งมีรูปร่างฟิลเตอร์ ดังรูปที่ 2.26

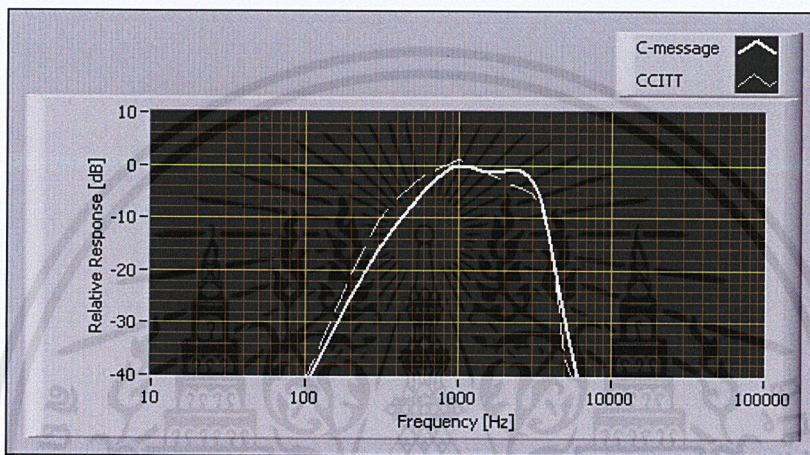
จากเอกสาร ITU-T O.41 [15] มีการแสดง C-message weighting coefficients and accuracy limits ดังรูปที่ 2.26 ที่ความถี่ 1000 เฮิรตซ์ คือความถี่ที่ใช้เป็นค่าอ้างอิงที่มี Relative weight (dB) และ Tolerance ( $\pm$ dB) ประมาณ 0.0 ค่าที่โครงการนี้ให้ความสนใจ คือ ค่า Relative weight

ประกอบกับเอกสาร ขั้นตอนการปฏิบัติงาน (Work Instruction) การซ่อมบำรุงเชิงป้องกันระบบเครื่องรับวิทยุ WI-CE.SE-01 [16] ของทางบริษัทฯ ได้มีการกำหนดค่า Line Audio Output : SINAD > 10 dB ดังนั้นค่า SINAD ที่ได้วัดได้ที่ความถี่ต่าง ๆ นั้น ควรมีค่ามากกว่า 10 (dB)  $\pm$  ค่า Relative weight (dB) ดังตารางที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา 37 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 1 เช่น ถ้าที่ความถี่ 1000 เฮิรตซ์ สามารถวัดค่า SINAD ได้ 10 dB แล้วที่ความถี่ 800 เฮิรตซ์ ตามรูปที่ 2.26 มีค่า Relative weight เท่ากับ -1.3 dB ดังนั้นที่ความถี่ 800 เฮิรตซ์ ควรวัดค่า SINAD ได้ > 8.7-10 dB

ตัวอย่างที่ 2 เช่น ถ้า ที่ความถี่ 1000 เฮิรตซ์ สามารถวัดค่า SINAD ได้ 14 dB แล้วที่ความถี่ 800 เฮิรตซ์ ตามรูปที่ 2.26 มีค่า Relative weight เท่ากับ -1.3 dB ดังนั้นที่ความถี่ 800 เฮิรตซ์ ควรวัดค่า SINAD ได้ > 12.7-14 dB



รูปที่ 2.26 รูปร่างของฟิลเตอร์ CCITT และฟิลเตอร์ C-message weighting [14]

SINAD (Signal to Noise and Distortion ratio)

คำว่า SINAD เป็นคำขยายความของรูปแบบ และวิธีการตรวจวัดภาครับแบบ SINAD ดังนั้นถ้าการตรวจวัดภาครับของเครื่องรับส่งโดยทั่วไปแล้ว จะคำนึงถึงสัญญาณต่อเสียงรบกวนเป็นหลัก แต่ SINAD จะรวมถึง Noise and distortion ด้วย จึงแปลความหมายได้ว่า SINAD คือ Signal to Noise and Distortion ซึ่งมีสูตรการคำนวณดังสมการที่ 2.2

$$\text{SINAD}_{(\text{dB})} = 20 \log \left[ \frac{\text{Signal}_{(V)} + \text{Noise}_{(V)} + \text{Distortion}_{(V)}}{\text{Noise}_{(V)} + \text{Distortion}_{(V)}} \right] \quad (2.2)$$

โดยทั่วไปแล้ว SINAD จะแสดงเป็น dB หรือเดซิเบล และแสดงควบคู่กับความไวของตัวรับ RF เพื่อให้การประเมินเชิงปริมาณของความไวของตัวรับ ซึ่งแตกต่างจาก SNR การอ่าน SINAD จะต้องไม่น้อยกว่า 1 (เช่น มันจะเป็นค่าบวกเสมอ เมื่อยกมาเป็น dB) [17]

ตารางที่ 2.6 C-message weighting coefficients and accuracy limits [15]

Frequency (Hz)	Relative weight (dB)	Tolerance ( $\pm$ dB)
60	-55.7	2
100	-42.5	2
200	-25.1	2
300	-16.3	2
400	-11.2	1
500	-7.7	1
600	-5.0	1
700	-2.8	1
800	-1.3	1
900	-0.3	1
1,000	0.0	0.0 (Reference)
1,200	-0.4	1
1,300	-0.7	1
1,500	-1.2	1
1,800	-1.3	1
2,000	-1.1	1
2,500	-1.1	1
2,800	-2.0	1
3,000	-3.0	1
3,300	-5.1	2
3,500	-7.1	2
4,000	-14.6	3
4,500	-22.3	3
5,000	-28.7	3

### 2.13 สรุป

ในบทนี้ได้มีการศึกษาถึงภาษาไพทอนที่ใช้ในการเขียนโปรแกรม หลักการมอดูเลตแอมพลิจูดที่ใช้ในการสื่อสารแบบแอนะล็อก หลักการมอดูเลตรหัสพัลส์ที่ใช้ในการสื่อสารแบบดิจิทัล มาตรฐาน G.711 หรือ การมอดูเลตรหัสพัลส์แบบ A-law ซึ่งถูกใช้ในการสื่อสารแบบ VoIP รวมไปถึงหลักการของ VoIP และกฎสำหรับการใช้งาน VoIP เพื่อสนับสนุนการสื่อสารของการบริหารจัดการจราจรทางอากาศ นอกจากนี้ ได้ศึกษาโปรแกรม Wireshark ซึ่งเป็นโปรแกรม Sniffer ใช้ในการดักจับข้อมูลบนระบบโครงข่าย และหลักการของ Port mirroring ที่สามารถช่วยในการตรวจสอบ หรือวิเคราะห์โครงข่ายได้ และท้ายที่สุดได้ศึกษาฟิลเตอร์ C-message weighting ซึ่งเป็นฟิลเตอร์ที่นิยมใช้กับเสียง เพื่อกรองสัญญาณรบกวนออกไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา 39 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

# การออกแบบระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ

### 3.1 การออกแบบ

#### 3.1.1 การออกแบบการทำงานระบบ

ผู้จัดทำได้ออกแบบการทำงานระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ โดยทำการเชื่อมต่อสาย LAN 1 เส้นเข้าระหว่าง Port mirroring ของสวิตช์ที่มีการเชื่อมต่อกับวิทยุเครื่องรับ และวิทยุเครื่องส่งอย่างละ 1 เครื่อง และโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศที่ได้จัดทำขึ้น เพื่อตรวจสอบสถานะการเชื่อมต่อของวิทยุ และการรับ-ส่งข้อมูลเสียง จากสำเนาแพ็กเก็ตของวิทยุที่มี IP Address ของวิทยุเครื่องรับ หรือวิทยุเครื่องส่งที่ได้รับจาก Port mirroring นอกจากนี้ผู้จัดทำได้ออกแบบให้สามารถบันทึก และเล่นไฟล์เสียงได้อีกด้วย โดยบล็อกไดอะแกรมของระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศแสดงได้ดังรูปที่ 1.1

#### 3.1.2 การออกแบบการทำงานของโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ

ผู้จัดทำได้ทำการออกแบบให้โปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ โดยออกแบบให้โปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศสามารถเชื่อมต่อกับ IP Address ของวิทยุเพื่อรับแพ็กเก็ตข้อมูลเสียงจากวิทยุเครื่องรับ/ส่ง ได้ครั้งละ 1 เครื่อง หรือ 1 IP Address โดยแรกเริ่มให้สามารถกำหนด IP Address ของวิทยุที่ต้องการสังเกตการณ์ รวมทั้งกำหนดชื่อไฟล์เสียงที่ต้องการบันทึกเป็นนามสกุล .wav จากนั้นตรวจสอบเงื่อนไขว่ามีการเริ่มการทำงานโปรแกรม จากการสร้างการเชื่อมต่อ (Connect) หรือไม่ ถ้าเชื่อมต่อแล้ว จะเริ่มสร้างไฟล์เสียงนามสกุล .wav ที่ได้ทำการกำหนดชื่อไปข้างต้นรอเพื่อรอรับการบันทึกเสียง และทำการรับแพ็กเก็ต VoIP จากวิทยุที่มี IP Address ตามที่ได้กำหนดไว้ ซึ่งจากแพ็กเก็ต VoIP ที่ได้รับนั้น จะถูกนำมาตรวจสอบสถานะการรับ/ส่งข้อมูลเสียงของวิทยุได้ ซึ่งหลักจากโปรแกรมเริ่มทำงานสามารถแบ่งการทำงานภายในได้เป็น 2 ส่วนหลัก ๆ ได้แก่

### 3.1.2.1 ส่วนของการบันทึก และเล่นไฟล์เสียง

ในกรณีที่ต้องการบันทึกไฟล์เสียง โปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ จะทำการตรวจสอบเงื่อนไขว่ามีการเริ่มบันทึก (Record) หรือไม่ หากมีการเริ่มบันทึก จะทำการตรวจสอบว่าภายในแพ็คเกจ VoIP ที่ส่งมานั้นมีส่วนของข้อมูลเสียงหรือไม่ ถ้ามีส่วนของข้อมูลเสียง จะนำข้อมูลเสียงนั้นไปถอดรหัส และเขียนบันทึกลงในไฟล์เสียงนามสกุล .wav ที่ได้กำหนดไปในข้างต้น

ในกรณีที่ต้องการหยุดการบันทึกไฟล์เสียง โปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ จะทำการตรวจสอบเงื่อนไขว่ามีการหยุดการบันทึก (Stop record) หรือไม่ หากมีการหยุดการบันทึก โปรแกรมจะทำการปิดการเขียนไฟล์เสียงนามสกุล .wav นั้น

ในกรณีที่ต้องการเล่นเสียงจากไฟล์เสียงนามสกุล .wav ให้ทำการกำหนดชื่อไฟล์เสียงนามสกุล .wav ที่ต้องการ โปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ จะทำการตรวจสอบเงื่อนไขว่ามีความต้องการจะเล่นไฟล์เสียงหรือไม่ หากมีความต้องการเล่นไฟล์เสียง โปรแกรมจะนำไฟล์เสียงนามสกุล .wav ที่ต้องการ ไปเข้าฟิลเตอร์ เพื่อลดสัญญาณรบกวน และทำการเล่นเสียงออกมา

### 3.1.2.2 ส่วนของการตรวจสอบสถานะ

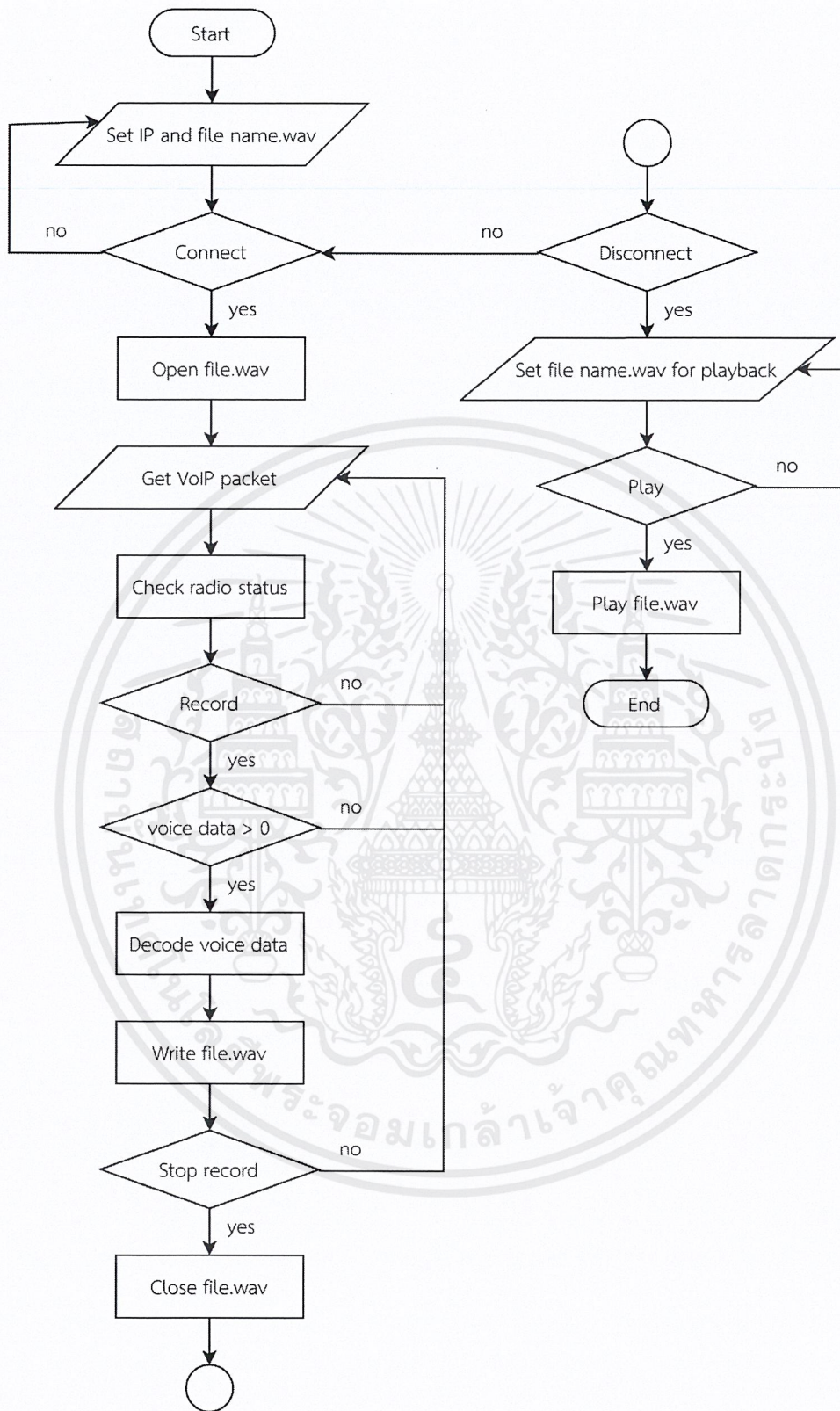
สำหรับการตรวจสอบสถานะการเชื่อมต่อ และการรับ/ส่งข้อมูล เมื่อมีการเชื่อมต่อโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศกับ IP Address ของวิทยุเครื่องรับ หรือวิทยุเครื่องส่งแล้วนั้น

ในกรณีการตรวจสอบสถานะการเชื่อมต่อ ทั้งวิทยุเครื่องรับ และวิทยุเครื่องส่ง จะทำการตรวจสอบจาก Keep alive message

ในกรณีการตรวจสอบการส่งข้อมูลเสียงของวิทยุเครื่องส่ง จะทำการตรวจสอบจากสถานะบิต Push To Talk ซึ่งเป็นสถานะที่ให้สัญญาณก่อนการเริ่มส่งข้อมูลเสียง

ในกรณีการตรวจสอบการรับข้อมูลเสียงของวิทยุเครื่องรับ จะทำการตรวจสอบจากสถานะบิต Squelch ซึ่งเป็นสถานะที่ให้สัญญาณเมื่อวิทยุเครื่องรับได้รับข้อมูลเสียงเข้ามา

ซึ่งโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศนี้ มีผังงาน (Flowchart) การทำงานดังแสดงในรูปที่ 3.1 จะแสดงผลผ่าน GUI ซึ่งจะกล่าวต่อไปในหัวข้อ 3.3



รูปที่ 3.1 ผังงานการทำงานของโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ

### 3.1.3 การแบบออกแบบ GUI

ในการออกแบบ GUI นั้น ผู้จัดทำได้ทำการออกแบบเป็น 2 tab ได้แก่ Record-Play tab และ Status tab ดังแสดงในรูปที่ 3.2

สำหรับ Record-Play tab มีหน้าที่หลักในการเริ่มโปรแกรมจากการเชื่อมต่อ IP Address ของวิทยุที่ต้องการตรวจสอบสถานะรับ/ส่ง รวมทั้งการบันทึก และเล่นไฟล์เสียงนามสกุล .wav โดยจะสามารถทำการบันทึกไฟล์เสียงนามสกุล .wav และเล่นเสียงผ่านไฟล์นามสกุล .wav ซึ่งสามารถกำหนดชื่อไฟล์เสียงที่ต้องการบันทึก และเล่นไฟล์เสียงได้ตามต้องการ

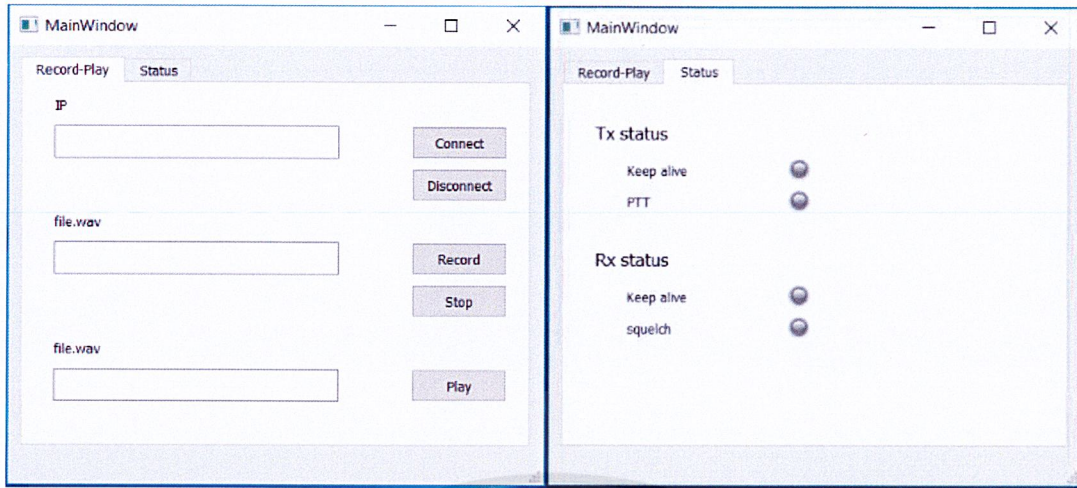
สำหรับ Status tab มีหน้าที่หลักในการแสดงสถานะการเชื่อมต่อของ IP Address วิทยุที่ต้องการตรวจสอบ และสังเกตการณ์การเชื่อมต่อ

เริ่มจาก Record-Play tab ถ้ากดปุ่ม Connect เพื่อทำการเชื่อมต่อ ซึ่งได้กำหนด IP ของเครื่องวิทยุรับ/ส่งที่ต้องการสังเกตการณ์ จากนั้นสามารถกดปุ่ม Record เพื่อทำการบันทึกเสียงจากแพ็กเก็ต VoIP ที่มาจากวิทยุเครื่องรับ/ส่ง สามารถกดปุ่ม Stop เมื่อต้องการหยุดบันทึกไฟล์เสียง และสามารถกดปุ่ม Play เพื่อทำการเล่นไฟล์เสียงที่ได้บันทึกไปก่อนหน้านี้ ซึ่งก่อนการเล่นไฟล์เสียงนั้น ต้องทำการกดปุ่ม Disconnect เพื่อยกเลิกการเชื่อมต่อก่อน

ในขณะเดียวกัน เมื่อมีการกด Connect ใน Record-Play tab เพื่อเชื่อมต่อกับวิทยุที่มี IP Address ตามที่ต้องการ Status tab จะแสดงสถานะของวิทยุ โดยแสดงออกผ่านทางสัญลักษณ์ LED สีเขียว ซึ่งหมายถึงสถานะนั้นมีการทำงานอยู่ และสีเทา ซึ่งหมายถึงสถานะนั้นไม่ได้ทำงานอยู่ ในกรณีที่มีการกดปุ่ม Disconnect ใน Record-Play tab การแสดงสถานะจะหยุดลงเช่นกัน

สรุป คือ โปรแกรมที่เขียนขึ้นนั้นในหน้าต่าง GUI สามารถกดปุ่ม Connect และปุ่ม Disconnect เพื่อทำการเชื่อมต่อ และยกเลิกการเชื่อมต่อกับ IP Address ของวิทยุ สามารถกดปุ่ม Record และปุ่ม Stop เพื่อทำการบันทึก และหยุดบันทึกได้มากกว่า 1 ครั้ง เนื่องจากโปรแกรมมีการทำงานแบบวน loop แต่ในกรณีของการเล่นเสียงนั้น ควรที่จะกดปุ่ม Play เพื่อเล่นเสียง หลังมีการยกเลิกการเชื่อมต่อแล้ว

โปรแกรมนี้ตรวจสอบสถานะของวิทยุเครื่องส่ง หรือเครื่องรับได้ครั้งละ 1 เครื่อง ซึ่งเป็นไปตาม IP Address ของวิทยุเครื่องรับ หรือเครื่องส่งที่ได้เลือกเชื่อมต่อไว้ในตอนแรกนั่นเอง



รูปที่ 3.2 หน้าต่าง Record-Play tab และ Status tab

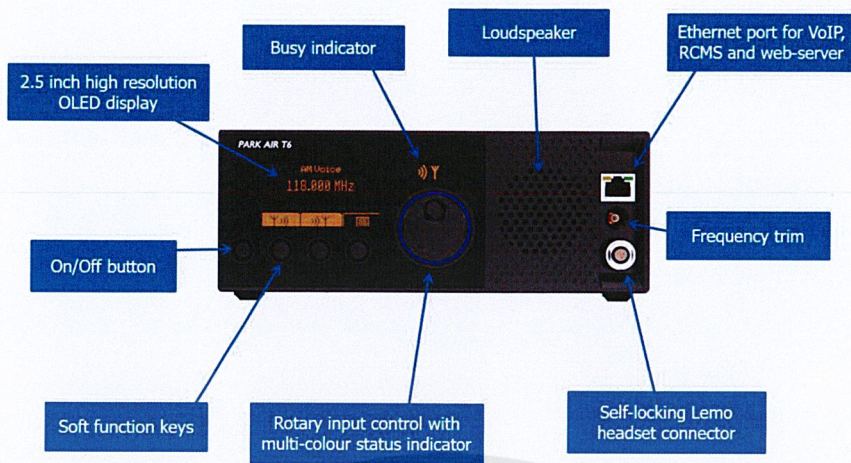
## 3.2 เครื่องมือที่ใช้ในการทดลอง

### 3.2.1 วิทยุ Park Air Sapphire T6 Series3

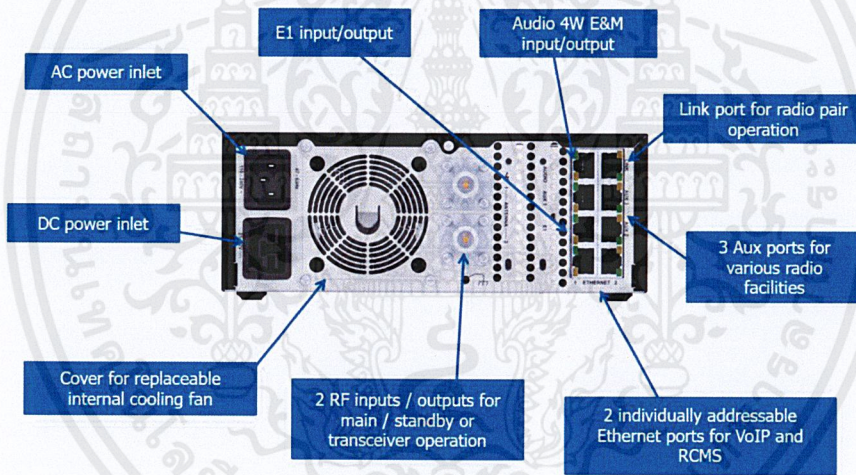
Park Air T6 Radios คือ วิวัฒนาการล่าสุดของแพลตฟอร์มวิทยุภาคพื้นดินสู่อากาศ (Air - Ground Communication) ชั้นนำของโลก ซึ่งเต็มไปด้วยเทคโนโลยีล่าสุด มาตรฐานสำหรับความปลอดภัย และประสิทธิภาพสูง ที่ให้ประสิทธิภาพโดดเด่นสำหรับการสื่อสารและครอบคลุมย่านความถี่ VHF และ UHF ซึ่งตัวเครื่องวิทยุด้านหน้า และด้านหลังแสดงดังรูปที่ 3.3 และ 3.4 ตามลำดับ

ตารางที่ 3.1 คุณสมบัติของวิทยุ Park Air Sapphire T6 Series3 [18]

ช่วงความถี่	112-156 MHz หรือ 225-400 MHz
กำลังขาออก	5-50 วัตต์
รูปแบบของคลื่น	AM Voice; ACARS; VDL-M2; AM Wideband
สัญญาณรบกวนบรรดแบนด์ของเครื่อง	< -160 dBc/Hz at 1% f0 offset
ความไวของตัวรับ	< -107 dBm
น้ำหนักตัวรับ-ส่งสัญญาณ	< 6 กิโลกรัม
ขนาด	79 (2U) x 210 มิลลิเมตร



รูปที่ 3.3 วิทย์ Park Air Sapphire T6 (ด้านหน้า) [18]



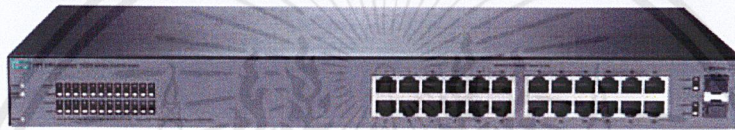
รูปที่ 3.4 วิทย์ Park Air Sapphire T6 (ด้านหลัง) [18]

### 3.2.2 สวิตช์ HPE OfficeConnect 1920S Series Switch JL381

HPE OfficeConnect 1920S Switch Series นี้ประกอบไปด้วย Gigabit Switch หลายรุ่นที่สามารถบริหารจัดการได้อย่างชาญฉลาดแบบพร้อมใช้งาน สำหรับใช้งานภายในองค์กรธุรกิจขนาดเล็กที่ต้องการระบบเครือข่ายซึ่งดูแลรักษาได้ง่าย แสดงดังรูปที่ 3.5

ตารางที่ 3.2 คุณสมบัติของสวิตช์ HPE OfficeConnect 1920S Series Switch JL381 [19]

แรงดันไฟฟ้าขาเข้า	15.7 วัตต์ (สูงสุด), 11.6 วัตต์ (ไม่ได้ใช้งาน)
พอร์ตขาเข้า-ออกภายนอก	RJ-45 10/100/1,000 Mbps 24 พอร์ต และ SFP 100/1,000 Mbps 2 พอร์ต
ความสามารถในการกำหนดเส้นทาง	52 Gbps
ปริมาณงาน	สูงสุดถึง 38.6 Mbps
ลักษณะการจัดการ	Web browser SNMP IEEE 802.3 Ethernet mib
น้ำหนัก	2.72 กิโลกรัม
ขนาด	4.39 ซม. x 44.25 ซม. x 24.61 ซม.



รูปที่ 3.5 สวิตช์ HPE OfficeConnect 1920S Series Switch JL381 [19]

### 3.2.3 โปรแกรม Park Air S4 IP Controller

คือโปรแกรมที่ใช้ในการทดสอบการรับ-ส่งวิทยุ ด้วยส่วนต่อประสานผู้ใช้ที่ใช้งานง่าย และเข้าถึงโดยตรงไปยังช่องสัญญาณเสียงวิทยุตัวควบคุม IP ซึ่ง IP ทำให้สามารถควบคุมระบบการสื่อสารได้อย่างสมบูรณ์ สอดคล้องกับมาตรฐาน VoIP ล่าสุดสำหรับการควบคุมจราจรทางอากาศ (Air Traffic Control : ATC) เหมาะสำหรับการใช้งานในสนามบิน การจัดส่ง หรือการใช้งานอื่น ๆ แสดงดังรูปที่ 3.6

ตารางที่ 3.3 คุณสมบัติของโปรแกรม Park Air S4 IP Controller [20]

หน่วยควบคุม IP ระยะไกล	All in one
มาตรฐาน	ED-137 A B และ C
สถานีวิทยุ	8 สถานีวิทยุ main standby
ความถี่ต่อช่องสัญญาณ	100 ความถี่
เครื่องบันทึก IP	4 sessions
ขนาด	204 มม. x 294 มม. x 48 มม.



รูปที่ 3.6 โปรแกรม Park Air S4 IP Controller

### 3.2.4 CMA 180 Radio Test Set

ในการทดลอง จะเลือกใช้เครื่องมือ Rx Test ซึ่งเป็นส่วนที่เกี่ยวข้องกับวิทยุเครื่องรับเท่านั้น เนื่องจากไม่ต้องการให้ส่งผลกระทบต่อส่วนของวิทยุเครื่องส่ง ซึ่งอาจเกิดข้อผิดพลาดได้ เมื่อระมัดระวังไม่มากพอ ซึ่งตัวเครื่อง CMA 180 Radio Test Set แสดงดังรูปที่ 3.7

เครื่องมือ Rx Test ในส่วนตัวกำเนิดสัญญาณที่นำมาใช้นั้น สามารถกำเนิดสัญญาณคลื่นรูปไซน์ที่มีความถี่ต่าง ๆ ได้ ในการทดลองเลือกใช้ความถี่ 400-3,000 เฮิรตซ์ และสามารถนำสัญญาณมาวิเคราะห์ค่าต่าง ๆ ได้

- Signal generator สามารถกำเนิดสัญญาณดังต่อไปนี้แบบคู่ขนานได้
  - Audio signals (สัญญาณเสียง) – สัญญาณเสียงแอนะล็อก 2 สัญญาณ และสัญญาณเสียงดิจิทัล 2 สัญญาณ
  - RF signal (สัญญาณความถี่วิทยุ) – สัญญาณที่ถูกมอดูเลตระหว่างสัญญาณเสียงดิจิทัล 1 หรือ 2 สัญญาณ รวมกับ Additional tone
  - RF interferer
- Analyzer



รูปที่ 3.7 CMA 180 Radio Test Set [21]

### 3.4.3 ตัวลดทอนสัญญาณ (Attenuator)

ตัวลดทอนสัญญาณ ในที่นี้มีหน้าที่เป็น Dummy Load หรือก็คือ อุปกรณ์ที่ต่อแทนสายอากาศ ในขณะที่ปรับแต่ง และทดสอบวิทยุ ซึ่งมีค่าความต้านทาน 50 โอห์ม รับกำลังงานได้สูงสุด 100 วัตต์ [22] แสดงดังรูปที่ 3.8

ตัวลดทอนสัญญาณนี้จะต่อแทนสายอากาศของวิทยุเครื่องส่งเพื่อไม่ให้เกิดการส่งสัญญาณออกไปรบกวนการทำงานจริงภายในบริษัทฯ



รูปที่ 3.8 ตัวลดทอนสัญญาณ

### 3.3 การจัดเก็บผลการทดลอง

ในการทดลองจะทำการทดลองเกี่ยวกับวิทยุเครื่องรับเป็นหลัก เนื่องจากไม่ต้องการให้การทดลองส่งผลกระทบต่อการทำงานจริงภายในบริษัทฯ

ในหัวข้อ 3.3 จะแบ่งเป็น 2 ส่วนหลัก ๆ ได้แก่ ส่วนของการตั้งค่าอุปกรณ์ ซึ่งในทุกการทดลองจะต้องทำการตั้งค่าวิทยุ Park Air Sapphire T6 Series 3 โปรแกรม Park Air S4 IP Controller ที่ใช้ในการสร้าง Session ในการเชื่อมต่อวิทยุไว้เสมอ และการตั้งค่า Port mirroring และการทดลองต่าง ๆ

#### 3.3.1 การตั้งค่าวิทยุ Park Air Sapphire T6 Series 3

ในการทดลองจะทำการตั้ง IP Address 192.168.1.10 สำหรับวิทยุเครื่องรับ และ 192.168.1.20 สำหรับวิทยุเครื่องส่ง ส่วนความถี่ที่ใช้รับ/ส่งสัญญาณมานั้นจะขึ้นกับการใช้งานการทดลองนั้น ๆ ทางด้านหลังของวิทยุบริเวณของ RF input จะทำการเชื่อมต่อกับเสาอากาศ หรือ CMA 180 Radio Test Set ขึ้นกับการทดลอง

ส่วนการตั้งค่าให้วิทยุรับ/ส่งแพ็คเกจแบบ VoIP นั้น สามารถทำได้โดยการเสียบสาย LAN บริเวณด้านหน้าของวิทยุช่อง IP3 ดังตัวอย่างในรูปที่ 3.9 เข้ากับสวิตช์ที่เชื่อมต่อกับโปรแกรม Park Air S4 IP Controller อยู่ เพื่อให้วิทยุสามารถติดต่อกับโปรแกรม Park Air S4 IP Controller ในการสร้าง Session ได้



รูปที่ 3.9 วิทยุเครื่องรับความถี่ 120.5 เมกกะเฮิร์ตซ์ที่ช่อง IP3 ต่อไปยังสวิตช์

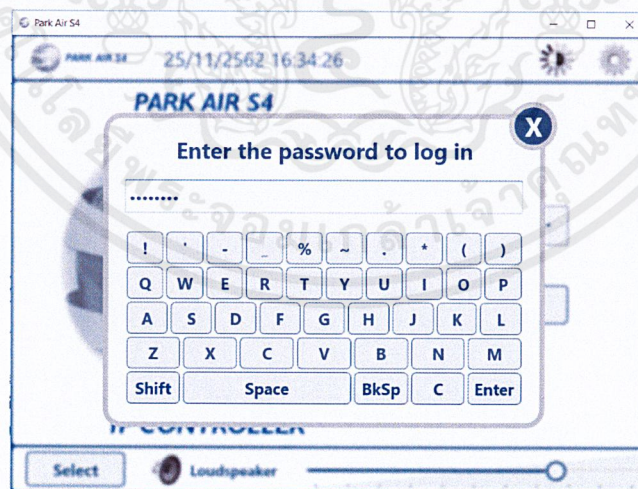
### 3.3.2 การตั้งค่า Park Air S4 IP Controller

1) เปิดโปรแกรม Park Air S4 IP Controller หน้าจอจะแสดงดังรูปที่ 3.10




รูปที่ 3.10 หน้าจอเข้าระบบ Park Air S4 IP Controller

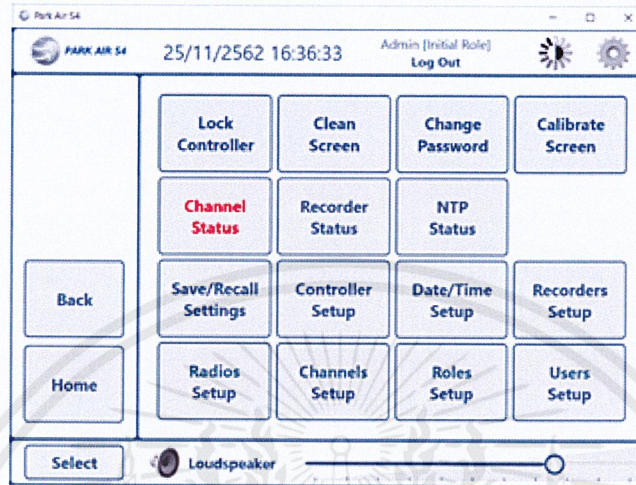
2) Enter > ใส่รหัส “Password” > Enter ดังรูปที่ 3.11



รูปที่ 3.11 หน้าจอกรอกรหัสเข้าระบบ Park Air S4 IP Controller

### 3) เมื่อเข้าระบบ Park Air S4 IP Controller

กดปุ่ม  (Setting) หน้าจอแสดงดังรูปที่ 3.12

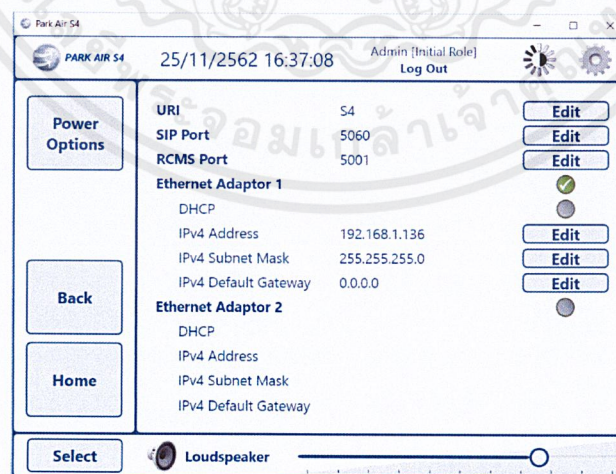


รูปที่ 3.12 หน้าจอสำหรับ Setting

### 4) เริ่มการสร้าง Channel

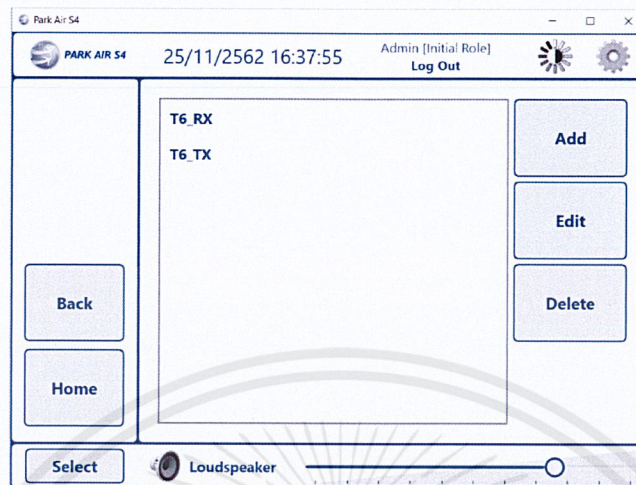
Setting > Controller Setup

ตั้งค่า URI IP Address และ Port ต่าง ๆ ให้ตรงตามรูปที่ 3.13



รูปที่ 3.13 หน้าจอ Controller Setup

5) Setting > Radios Setup > Add > ตั้งชื่อวิทยุ หน้าจอแสดงดังรูปที่ 3.14



รูปที่ 3.14 หน้าจอ Radios Setup

6) Setting > Radios Setup > เลือกชื่อวิทยุ > Edit

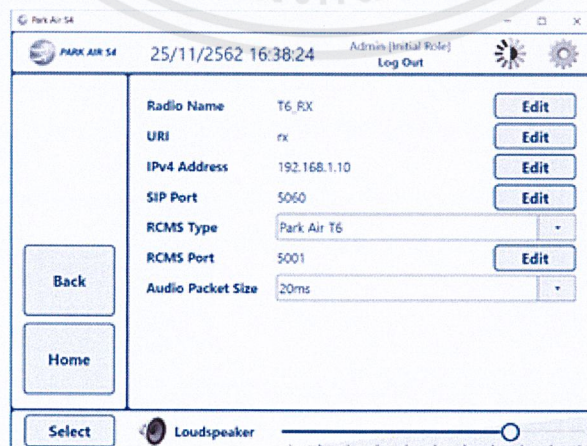
ตั้งค่า URI ให้กับวิทยุเครื่องรับ/ส่ง

IPv4 Address ให้ตรงตามวิทยุเครื่องรับ/ส่งที่ต้องการ

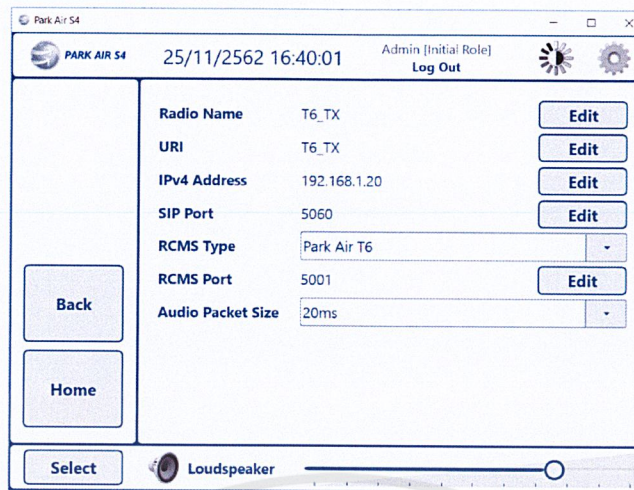
Port SIP Port กำหนดให้เป็น 5060

RCMS Port กำหนดให้เป็น 5001

โดยการตั้งค่าสำหรับวิทยุเครื่องรับ และเครื่องส่งแสดงดังรูปที่ 3.15 และ 3.16 ตามลำดับ



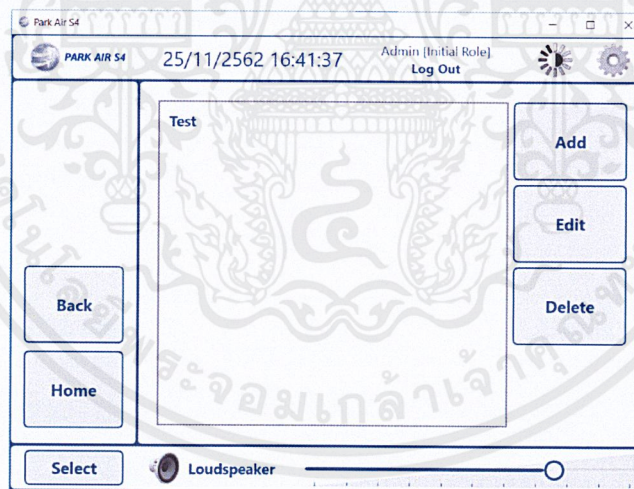
เอกสารนี้เป็นเอกสารที่รูปที่ 3.15 หน้าจอการตั้งค่า Radios Setup ของวิทยุเครื่องรับ  
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การข่งขันเพื่อสิทธิประโยชน์นี้ มิใช่ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 หน้าจอการตั้งค่า Radios Setup ของวิทยุเครื่องส่ง

7) Setting > Channels Setup > Add

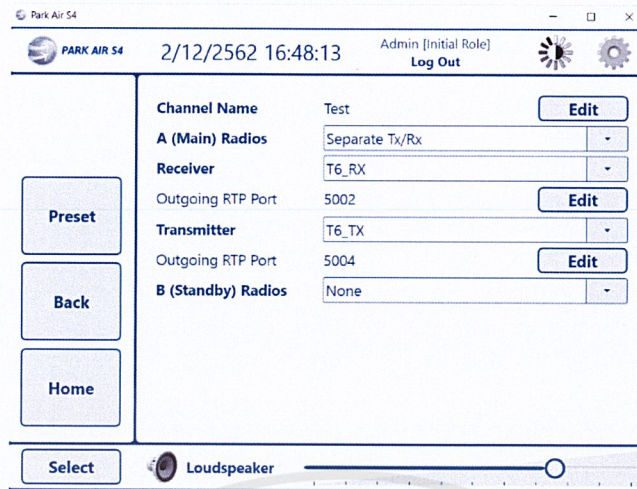
ตั้งชื่อ Channel เป็น Test แสดงดังรูปที่ 3.17



รูปที่ 3.17 หน้าจอ Channels Setup

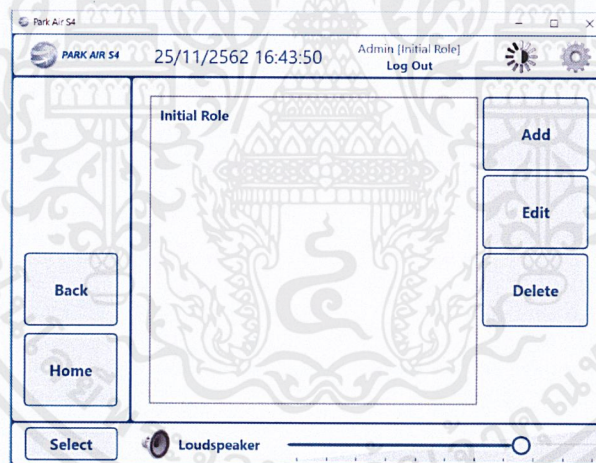
8) Setting > Channels Setup > เลือก Channel > Edit

ตั้งค่า A (Main) Radios เป็น Separate Tx/Rx จากนั้นเลือกชื่อ Receiver ให้ตรงกับวิทยุเครื่องรับที่ต้องการ ทำการตั้ง Outgoing Port เป็น 5002 หรือ 500X และเลือกชื่อ transmitter ให้ตรงกับวิทยุเครื่องส่งที่ต้องการ ทำการตั้ง Outgoing Port เป็น 5004 หรือ 500X (โดยที่ X เป็นเลขคู่ ตามมาตรฐานของ RTP Port และต้องไม่ซ้ำกับ Port ที่เลือกใช้ไปแล้ว) แสดงดังรูปที่ 3.18



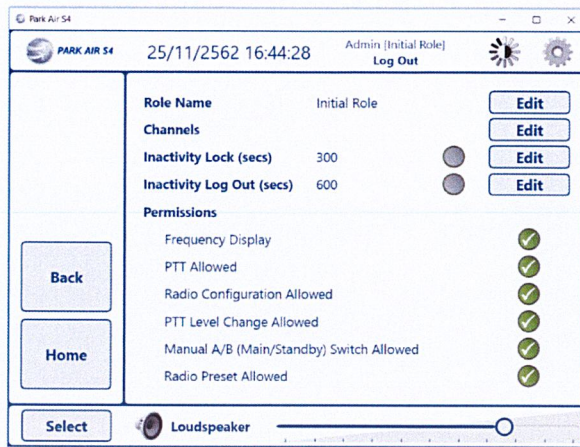
รูปที่ 3.18 หน้าจอการตั้งค่า Channels Setup ของ Channel Test

9) Setting > Roles Setup > เลือก Initial Role > Edit หน้าจอแสดงดังรูปที่ 3.19

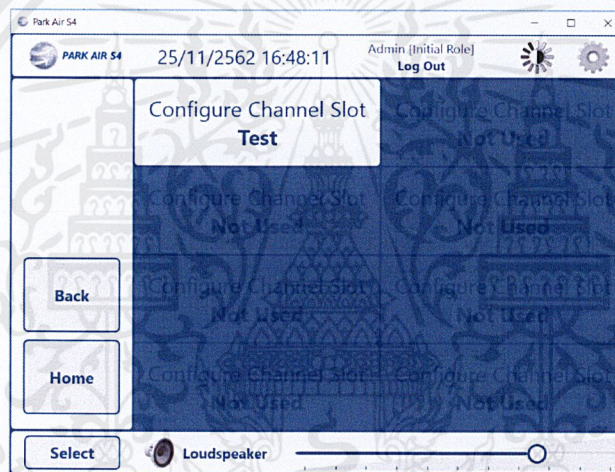


รูปที่ 3.19 หน้าจอ Roles Setup

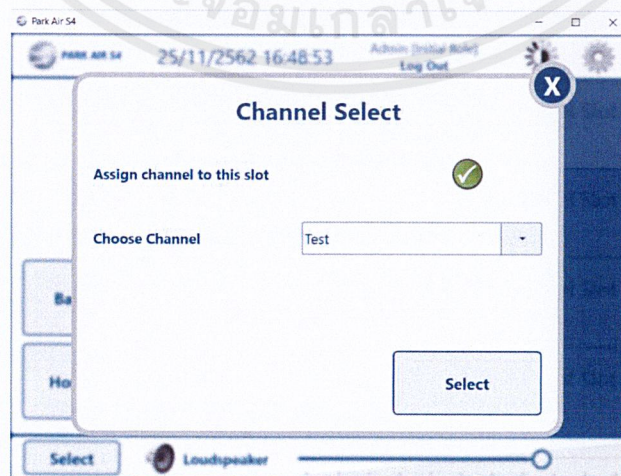
10) Setting > Roles Setup > เลือก Initial Role > Edit > Channels Edit > เลือกตำแหน่งของ Channel ที่ต้องการแสดงในโปรแกรม หน้าจอแสดงดังรูปที่ 3.20 ซึ่งในการทดลองเลือกเป็นช่องด้านซ้ายบนสุด ดังรูปที่ 3.21 > เลือก Assign channel to slot และเลือก Channel ที่ต้องการ ซึ่งในการทดลองกำหนดให้ชื่อ Test แสดงดังรูปที่ 3.22 ผลที่ได้นั้นแสดงดังรูปที่ 3.23



รูปที่ 3.20 หน้าจอการตั้งค่า Initial Role

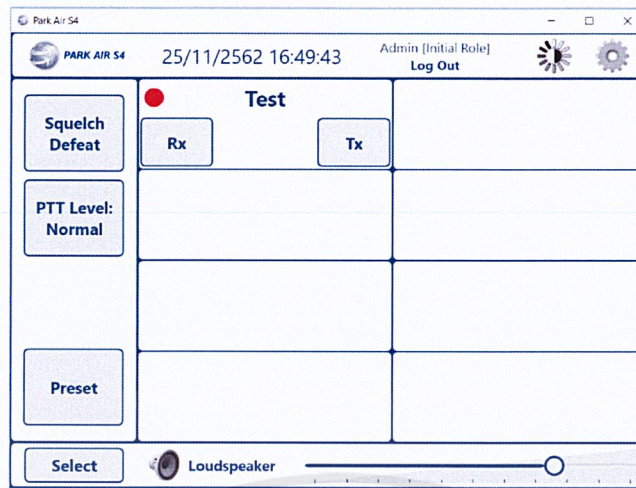


รูปที่ 3.21 ตำแหน่งของ Channel ที่เลือก



รูปที่ 3.22 การเลือก Channel ที่ใช้กับตำแหน่งของ Channel ที่เลือก

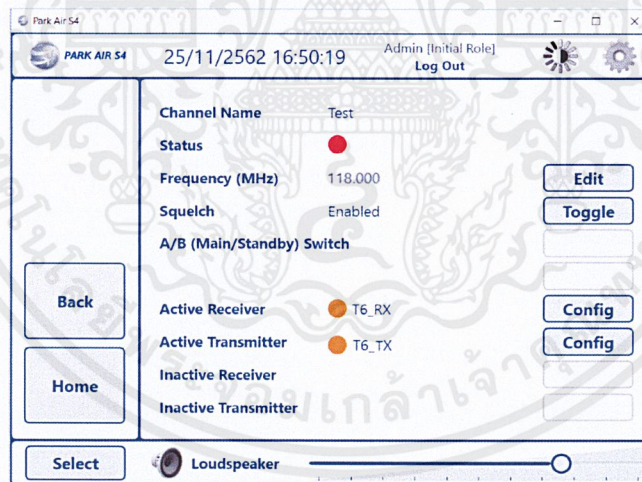
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิฉะนั้นผู้ใดเห็นใบใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 ผลของหน้าจอการตั้งค่า Roles Setup

11) Setting > Channel Status

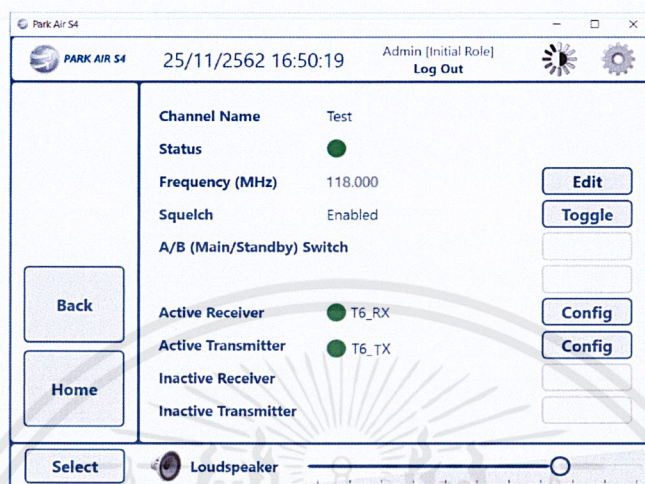
หน้าจอต้อนรับ Session การเชื่อมต่อกับวิทยุยังไม่สมบูรณ์ แสดงดังรูปที่ 3.24



รูปที่ 3.24 หน้าจอต้อนรับ Session การเชื่อมต่อกับวิทยุยังไม่สมบูรณ์

## 12) Setting > Channel Status

หน้าจอตอนที่ Session การเชื่อมต่อกับวิทยุสมบูรณ์แล้ว แสดงดังรูปที่ 3.25



รูปที่ 3.25 หน้าจอตอนที่ Session การเชื่อมต่อกับวิทยุสมบูรณ์แล้ว

### 3.3.3 การตั้งค่า Port mirroring ของสวิตช์

ในการทดลองจะทำการเชื่อมต่อสวิตช์กับอุปกรณ์ 4 อุปกรณ์ ได้แก่ วิทยุเครื่องรับ วิทยุเครื่องส่ง โปรแกรม Park Air S4 IP Controller และคอมพิวเตอร์ที่ใช้โปรแกรม Wireshark หรือโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศที่ได้จัดทำขึ้น

โดยสวิตช์ Port ที่ 1 เชื่อมต่อกับ วิทยุเครื่องรับ

Port ที่ 3 เชื่อมต่อกับ วิทยุเครื่องส่ง

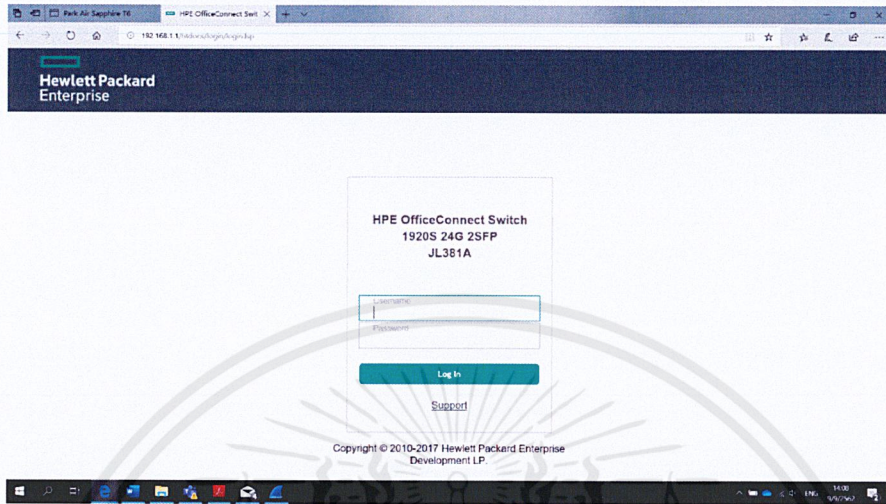
Port ที่ 9 เชื่อมต่อกับ โปรแกรม Park Air S4 IP Controller

Port ที่ 23 เชื่อมต่อกับ คอมพิวเตอร์

สำหรับ Port ที่ต้องการทำเป็น Port mirroring คือ Port ที่ 23 เนื่องจากต้องการสำเนาแพ็กเก็ตจาก Port อื่น ๆ มายัง Port ที่เชื่อมต่อกับคอมพิวเตอร์ เพื่อนำไปวิเคราะห์ในโปรแกรม Wireshark หรือโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศที่ได้จัดทำขึ้น ซึ่งการตั้งค่า Port mirroring สามารถทำได้ต่อไปดังต่อไปนี้

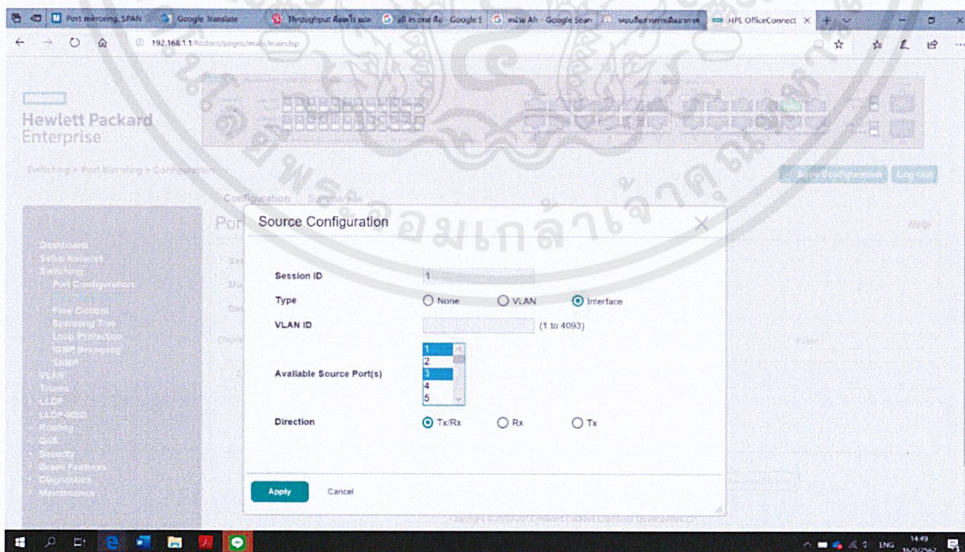
3.26

1) เข้าระบบ Config ของอุปกรณ์ HPE OfficeConnect 1920S Series Switch JL381 ดังรูปที่



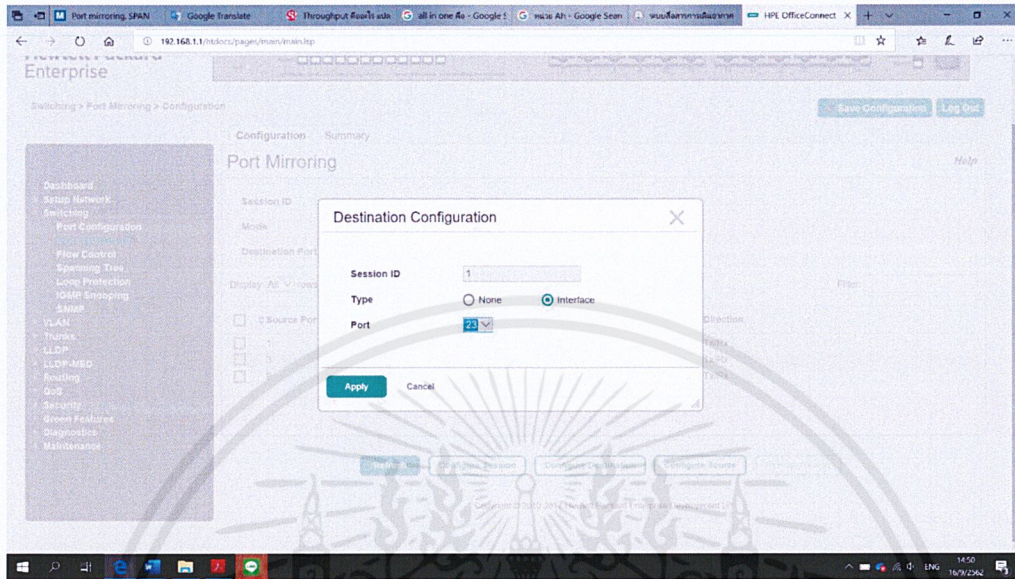
รูปที่ 3.26 หน้าจอเข้าระบบ Config ของสวิตช์

2) เลือก Port Mirroring > Configure Source และเลือก Port หมายเลข 1, 3 และ 9 > Apply  
ดังรูปที่ 3.27



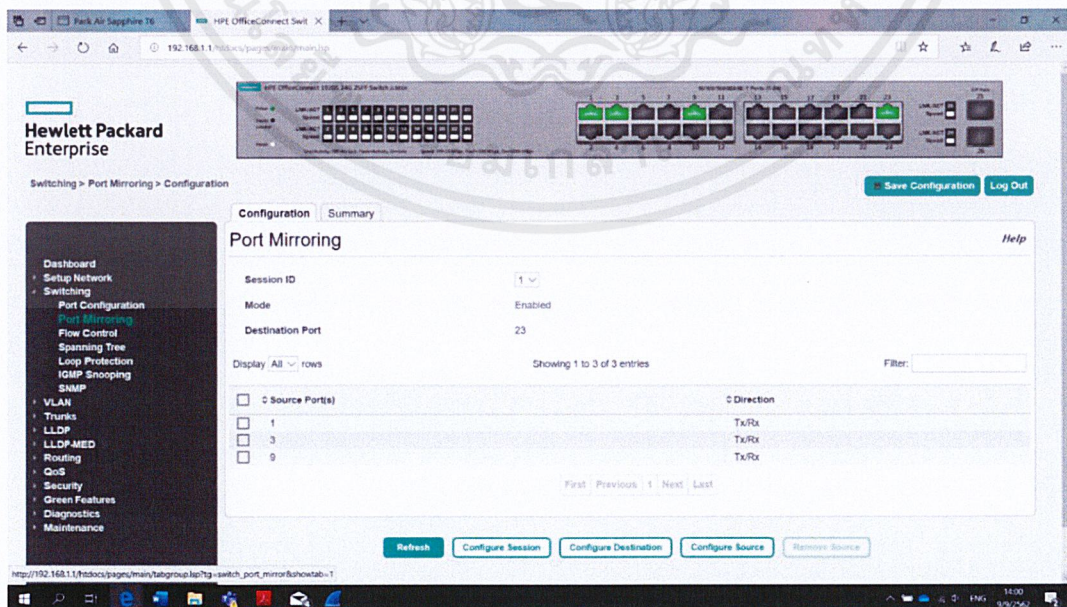
รูปที่ 3.27 การตั้งค่า Port ต้นทางของสวิตช์

3) เลือก Port Mirroring > Configure Destination และเลือก Portหมายเลข 23 > Apply ดังรูปที่ 3.28



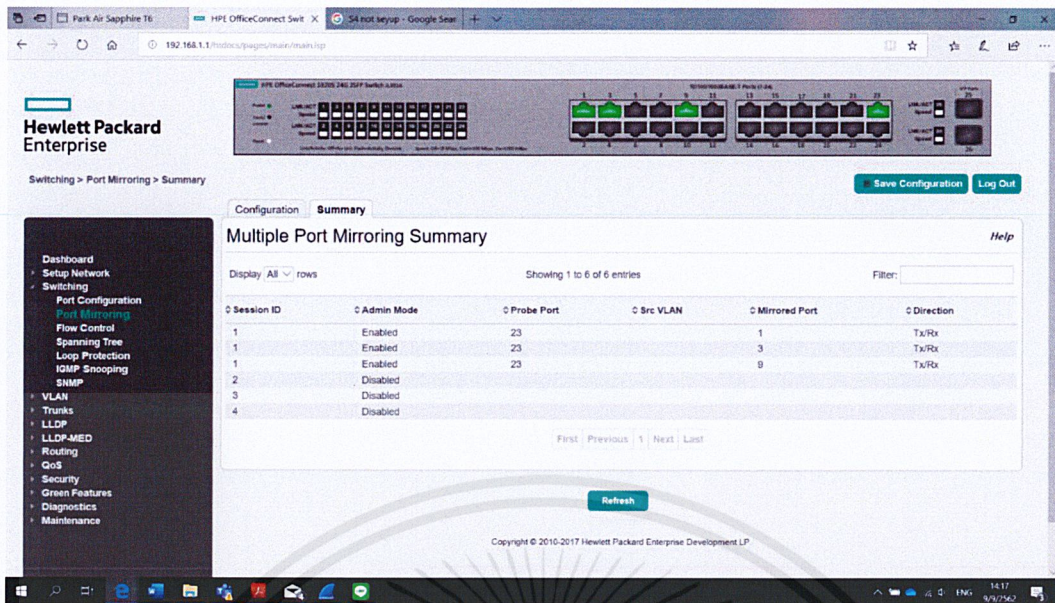
รูปที่ 3.28 การตั้งค่า Port ปลายทางของสวิตช์

เมื่อตั้งค่าเสร็จเรียบร้อยแล้ว สามารถดูการตั้งค่าพอร์ตได้ดังรูปที่ 3.29 และผลของการตั้งค่า Port mirroring ทั้งหมดได้ดังรูปที่ 3.30



รูปที่ 3.29 หน้าจอ Config port mirroring ของสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.30 หน้าจอแสดงผล เมื่อทำการ Config port mirroring แล้ว

### 3.3.4 การทดลองการสังเกตแพ็กเก็ต VoIP ที่อ่านได้จากโปรแกรม Wireshark

การทดลองนี้ ผู้จัดทำได้ทำการเชื่อมต่ออุปกรณ์ทั้งหมด 6 อุปกรณ์เข้าด้วยกัน ได้แก่คอมพิวเตอร์ สำหรับเปิดโปรแกรม Wireshark วิทยุเครื่องรับ วิทยุเครื่องส่ง สวิตช์ โปรแกรม Park Air S4 IP Controller และตัวลดทอนสัญญาณ

โดยสวิตช์ Port ที่ 1 เชื่อมต่อกับ วิทยุเครื่องรับ

Port ที่ 3 เชื่อมต่อกับ วิทยุเครื่องส่ง

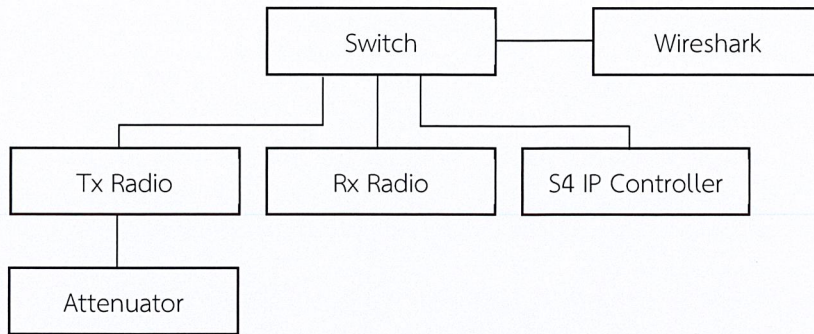
Port ที่ 9 เชื่อมต่อกับ โปรแกรม Park Air S4 IP Controller

Port ที่ 23 เชื่อมต่อกับ คอมพิวเตอร์

สำหรับ IP Address วิทยุเครื่องรับ กำหนดให้เป็น 192.168.1.10

สำหรับ IP Address วิทยุเครื่องส่ง กำหนดให้เป็น 192.168.1.20

นอกจากนี้ยังมีการต่อตัวลดทอนสัญญาณเข้ากับวิทยุเครื่องส่ง เพื่อป้องกันสัญญาณที่อาจเกิดจากการผิดพลาดทางการทดลองมีกำลังส่งเยอะจนไปกระทบต่อการทำงานภายในบริษัทซึ่งแสดงบล็อกไดอะแกรมได้ดังรูปที่ 3.31 และการเชื่อมต่ออุปกรณ์จริงดังรูปที่ 3.32



รูปที่ 3.31 บล็อกไดอะแกรมการเชื่อมต่ออุปกรณ์การทดลองที่ 3.3.3



รูปที่ 3.32 การเชื่อมต่อสวิตช์เข้ากับอุปกรณ์ต่าง ๆ

โดยในการทดลองนี้จะทำการดักจับแพ็กเก็ตข้อมูล VoIP ผ่านโปรแกรม Wireshark แล้วสังเกต Session การเชื่อมต่อของวิทย์ที่เกิดขึ้น และโพรโทคอลที่ถูกใช้ในแต่ละ Session แล้วนำมาเปรียบเทียบกับข้อมูลที่ได้ศึกษาในหัวข้อ 2.6 ข้างต้น

### 3.3.5 การทดลองเขียนโปรแกรมในการดักจับแพ็กเก็ต VoIP ด้วยภาษาไพทอน

การทดลองนี้เป็นการทดลองดักจับแพ็กเก็ตข้อมูล VoIP โดยแพ็กเก็ตข้อมูลที่ดักจับได้นั้น จะเป็น IP Address ของ Public Network หรือแพ็กเก็ตข้อมูลที่วิ่งผ่านคอมพิวเตอร์ที่ผู้จัดทำใช้ เช่น แพ็กเก็ตที่มาจาก การเชื่อมต่อ Wi-Fi ซึ่งการดักจับแพ็กเก็ต VoIP นั้น เป็นกระบวนการดักจับข้อมูลคล้ายกับโปรแกรม Wireshark ที่ได้ทำการศึกษาไปในเบื้องต้น

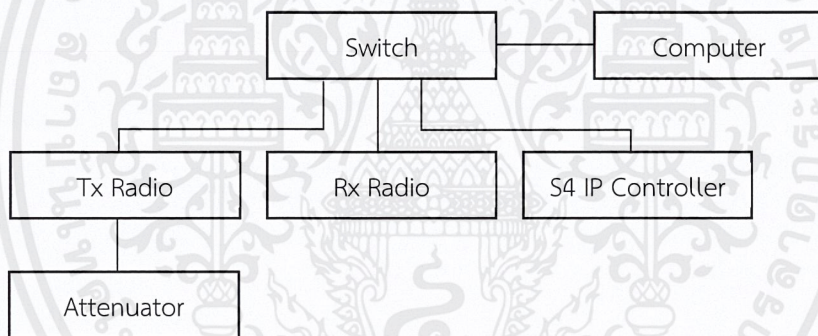
โดยจะแสดงผลผ่าน Terminal ของโปรแกรม PyCharm ซึ่งเป็นโปรแกรมที่ใช้หัดเขียนภาษาไพทอนเป็นโปรแกรมที่ใช้งานง่าย และรองรับการทำงานบนระบบปฏิบัติการ Windows Linux และ MacOS อีกด้วย  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.6 การทดลองเขียนโปรแกรมการดักจับแพ็กเก็ต VoIP ที่ได้จากเชื่อมต่อกับวิทยุ IP Address ที่ต้องการ

การทดลองนี้จะทำการเชื่อมต่ออุปกรณ์เช่นเดียวกับการทดลองที่ 3.3.4 แต่จะใช้โปรแกรมการดักจับแพ็กเก็ต VoIP ที่ผู้จัดทำเขียนขึ้นแทนการใช้โปรแกรม Wireshark แสดงได้ดังบล็อกไดอะแกรมในรูปที่ 3.33 โดยทำการทดลองดักจับแพ็กเก็ตข้อมูล VoIP ที่ได้จาก Port mirroring

ในการทดลองนี้จะทำการดักจับแพ็กเก็ต VoIP ที่ได้จาก Port mirroring และนำแพ็กเก็ตที่ได้มาทำการแยกส่วนประกอบแพ็กเก็ตให้อยู่ในรูปแบบข้อมูลที่อ่านง่าย ข้อมูลบางส่วนจะแสดงเป็นรูปของไบนารีและเรียงเรียงการแสดงผล เพื่อให้ง่ายต่อการเข้าใจ

โดยจะแสดงผลผ่าน Terminal ของโปรแกรม PyCharm จากนั้นทำการตรวจสอบว่ารายละเอียดข้อมูลของแพ็กเก็ตที่ได้รับจากการเขียนโปรแกรมในการดักจับแพ็กเก็ต VoIP ด้วยภาษาไพทอน กับรายละเอียดข้อมูลของแพ็กเก็ตที่ได้รับจากการดักจับแพ็กเก็ต VoIP ผ่านโปรแกรม Wireshark มีความสอดคล้อง ถูกต้องหรือไม่



รูปที่ 3.33 บล็อกไดอะแกรมการเชื่อมต่ออุปกรณ์การทดลองที่ 3.3.6

### 3.3.7 การทดลองตรวจสอบสถานะการเชื่อมต่อ และการรับ/ส่งข้อมูลของวิทยุ

การทดลองนี้จะทำการตรวจสอบสถานะการเชื่อมต่อของวิทยุเครื่องรับ จากแพ็กเก็ต VoIP ที่ได้จากการทดลองที่ 3.3.6 โดยวิทยุเครื่องรับนั้น สามารถตรวจสอบสถานะได้ 2 สถานะ ได้แก่ สถานะ Keep Alive ที่บอกว่าวิทยุเครื่องรับนั้นทำงานอยู่ และสถานะ Squelch ที่บอกว่าวิทยุเครื่องรับนั้นได้รับข้อมูลเสียงเข้ามา ทั้ง 2 สถานะนี้เป็นสถานะหลัก ๆ ที่ต้องการ เพื่อตรวจสอบการใช้งานเบื้องต้นของวิทยุเครื่องรับ

โดยจะแสดงผลผ่าน Terminal ของโปรแกรม PyCharm เช่นเดียวกับการทดลอง 3.3.6 แต่มีการเพิ่มเติมในส่วนของการแสดงสถานะให้เห็นในรูปของสถานะ “ON” หรือ “OFF”

### 3.3.8 การทดลองการบันทึกเสียงที่ได้จากแพ็คเกจ VoIP ของวิทยุ

การทดลองนี้จะทำการทดลองบันทึกเสียงที่ได้จากแพ็คเกจ VoIP ที่ได้จากการทดลองที่ 3.3.6 ซึ่งในการบันทึกนั้นจะบันทึกเป็นไฟล์นามสกุล .wav และทำการบันทึกไฟล์ด้วยเสียงที่มีการถอดรหัสเสียง G.711 แบบ A-law เนื่องจากแพ็คเกจจากวิทยุ นั้นเป็นแพ็คเกจเสียงที่เข้ารหัส G.711 แบบ A-law โดยทำการนำบิตเครื่องหมายออก และนำบิตที่เหลือมาแปลงเป็นค่าแอมพลิจูดของเสียง ซึ่งมีค่าตั้งแต่ -127 ถึง +127 ถ้าบิตเครื่องหมายเป็นเลข 0 ค่าที่แปลงมาจะเป็นค่าบวก แต่ถ้าบิตเครื่องหมายเป็นเลข 1 ค่าแอมพลิจูดที่แปลงมาจะเป็นค่าลบ ในทดลองนี้จะตั้งค่าความถี่ของวิทยุเครื่องรับเป็นคลื่นความถี่ที่ต้องการ โดยใช้ความถี่ 120.50 เมกกะเฮิรตซ์ ซึ่งเสียงที่วิทยุได้รับเข้ามานั้นจะเป็นเสียงของเจ้าหน้าที่ควบคุมจราจรทางอากาศ (Air Traffic Controller หรือ ATC)

โดยจะทำการตรวจสอบดูว่ามีไฟล์เสียงที่มีขนาดมากกว่า 0 KB (หมายถึง ไฟล์นั้นมีการเขียนข้อมูลลงไป) ถูกสร้างขึ้นหรือไม่ ถ้ามี แสดงว่ามีการบันทึกข้อมูลเสียงลงไปแล้ว

### 3.3.9 การทดลองเล่นไฟล์เสียง

การทดสอบการเล่นไฟล์เสียงนี้ จะทำการทดลองฟังไฟล์เสียงที่ได้ถูกสร้างขึ้นในการทดลองที่ 3.3.8 ทำการฟังเสียงที่เล่นออกมาว่าสามารถฟังรู้เรื่อง มีความใกล้เคียงกับเสียงที่ได้ยินผ่านวิทยุโดยตรงหรือไม่ หรือมีสัญญาณรบกวนหรือไม่

### 3.3.10 การทดลองกรองสัญญาณรบกวนออกจากไฟล์เสียง

การทดลองนี้ ในกรณีที่ไฟล์เสียงที่ทดลองเล่นออกมาในการทดลองที่ 3.3.9 นั้นมีสัญญาณรบกวนหรือเสียงซ่า ผู้จัดทำจึงทำการทดลองนำไฟล์เสียงนั้น ไปผ่านฟิลเตอร์ C-message weighting ซึ่งเป็นฟิลเตอร์ที่เหมาะสมกับเสียงพูดตามมาตรฐานของบริษัทฯ เพื่อลดสัญญาณรบกวนที่เกิดขึ้น จากนั้นทดลองฟังไฟล์เสียงที่ผ่านฟิลเตอร์ C-message weighting ว่าสามารถฟังรู้เรื่อง มีความใกล้เคียงกับเสียงที่ได้ยินผ่านวิทยุโดยตรงหรือไม่ สัญญาณรบกวนลดลงหรือไม่ เมื่อเปรียบเทียบกับทำการทดลองที่ 3.3.9

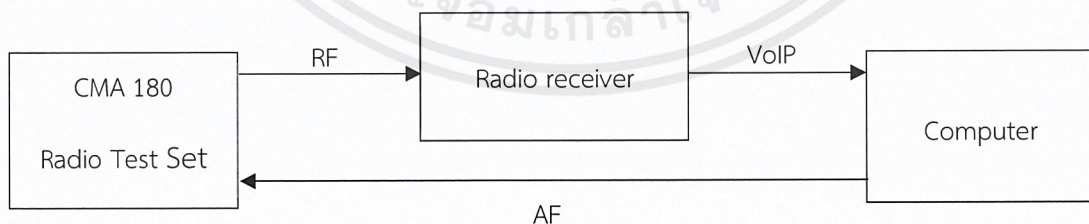
3.3.11 การทดลองเปรียบเทียบคุณภาพเสียงที่ได้จากการบันทึกของโปรแกรมระบบ  
สังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศที่ได้จัดทำ  
ขึ้น กับเสียงที่ได้จากระบบแอนะล็อกเดิม

3.3.11.1 การทดลองเปรียบเทียบเสียงจากโปรแกรมระบบสังเกตการณ์การสื่อสารทาง  
เสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ กับสัญญาณเสียงคลื่นรูปไซน์ ณ ความถี่  
ต่าง ๆ

การทดลองนี้จะทำการเปรียบเทียบเสียงดิจิทัลที่ได้จากโปรแกรมระบบสังเกตการณ์การ  
สื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ กับสัญญาณเสียงคลื่นรูปไซน์ที่  
ความถี่ต่าง ๆ จากเครื่อง CMA 180 Radio Test Set

การทดลองนี้ผู้จัดทำได้ทำการป้อนสัญญาณเสียงคลื่นรูปไซน์ที่ความถี่ต่าง ๆ จาก CMA  
180 Radio Test Set ซึ่งเป็นคลื่นความถี่วิทยุ (Radio Frequency : RF) โดยเชื่อมต่อสาย RG214 ซึ่งเป็น  
สายนำสัญญาณแบบสูญเสียต่ำ (Low loss) ไปยังวิทยุเครื่องรับ Park Air Sapphire T6 Series3 ที่มีกร  
รับแพ็คเกจข้อมูลเสียงแบบ VoIP ผ่านสาย LAN ไปยังคอมพิวเตอร์ที่เปิดใช้งานโปรแกรม Park Air S4 IP  
Controller สำหรับสร้าง session การเชื่อมต่อกับวิทยุ รวมถึงการรับ-ส่งแพ็คเกจข้อมูลเสียง และเปิดใช้  
งานโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการ  
เดินอากาศที่ผู้จัดทำเขียนขึ้น เพื่อใช้ในการบันทึกสัญญาณเสียงคลื่นรูปไซน์ที่กำเนิดจากเครื่อง CMA 180  
Radio Test Set และเล่นไฟล์เสียงที่บันทึก

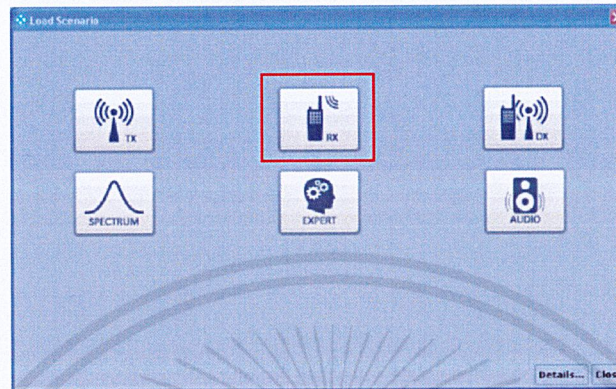
จากนั้นทำการป้อนเสียงที่เล่น ซึ่งเป็นคลื่นเสียง (Audio Frequency : AF) ของไฟล์เสียง  
ที่ได้จากการบันทึก ผ่านช่อง AUX ของคอมพิวเตอร์ไปยัง CMA 180 Radio Test Set ดังบล็อกไดอะแกรม  
ในรูปที่ 3.34 เพื่อทำการวิเคราะห์สัญญาณที่ป้อนกลับเข้ามา กับที่ป้อนออกไปในตอนแรก ซึ่งค่าที่ได้นั้นจะ  
นำไปเปรียบเทียบความถี่ (frequency) และ SINAD



รูปที่ 3.34 บล็อกไดอะแกรมสำหรับการทดลองที่ 3.3.11.1

ดังนั้นการทดลองนี้จึงต้องมีการตั้งค่า เครื่อง CMA 180 Radio Test Set ก่อนดังนี้

1) Scenario > Rx Test แสดงหน้าจอดังรูปที่ 3.35



รูปที่ 3.35 หน้าจอเลือกฟังก์ชันของเครื่อง CMA 180 Radio Test Set

2) เมื่อเข้า Rx Test จะมี tab ต่างๆ ที่ใช้ในการทดลอง ผู้จัดทำจะใช้ tab อยู่ 2 tab หลักๆ ได้แก่ Generator tab และ Analyzer tab แสดงดังรูปที่ 3.36

- Generator tab มีหน้าที่ในการให้กำเนิดเสียงคลื่นรูปไซน์ตามความถี่ที่ต้องการ
- Analyzer tab มีหน้าที่ในการวิเคราะห์ค่าที่ต้องการเปรียบเทียบระหว่างสัญญาณที่เครื่องป้อนออกไป กับสัญญาณที่เครื่องรับเข้ามา สำหรับ tab ย่อยของ Analyzer tab จะใช้ tab ย่อย AF Results และ AF Spectrum เป็นหลัก

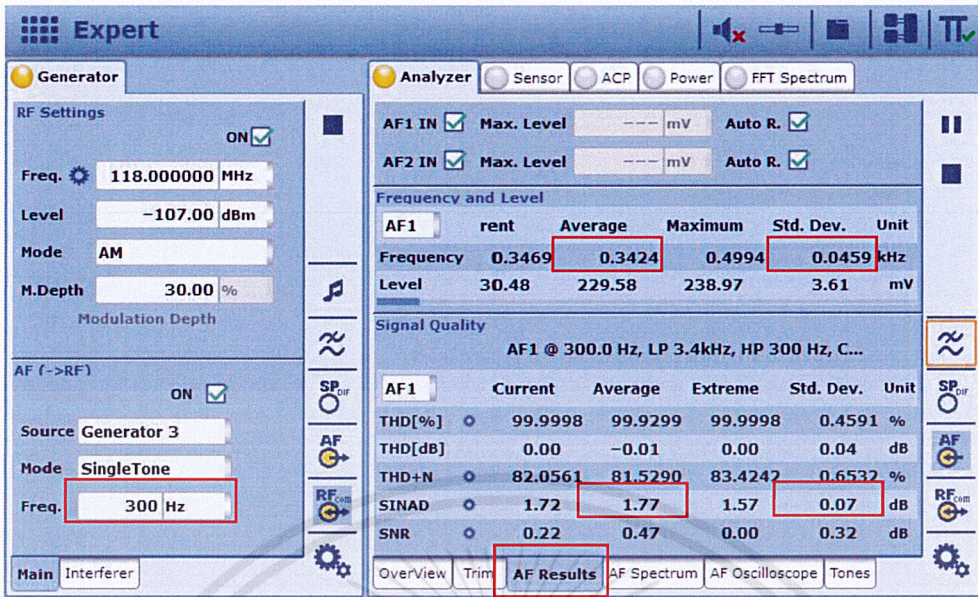
สำหรับ tab ย่อย AF Results มีหน้าที่ในการแสดงค่า Frequency and Level และ Signal Quality ที่ต้องมีการกำหนดฟังก์ชันของ filter โดยจะใช้เฉพาะ AF 1 IN

นอกจากนี้ต้องกำหนดการใช้ CCITT ของ Signal Quality ซึ่งเป็นฟิลเตอร์ที่บริษัทใช้ มีลักษณะคล้ายกับฟิลเตอร์ C-message weighting โดยมีการตั้งค่าฟิลเตอร์ Lowpass ที่ 3.4 กิโลเฮิร์ตซ์ และฟิลเตอร์ Highpass ที่ 300 เฮิร์ตซ์ ส่วนความถี่กลาง (Center frequency) ที่ใช้นั้นก็ขึ้นกับความถี่ที่ต้องการทดลอง แสดงดังรูปที่ 3.37

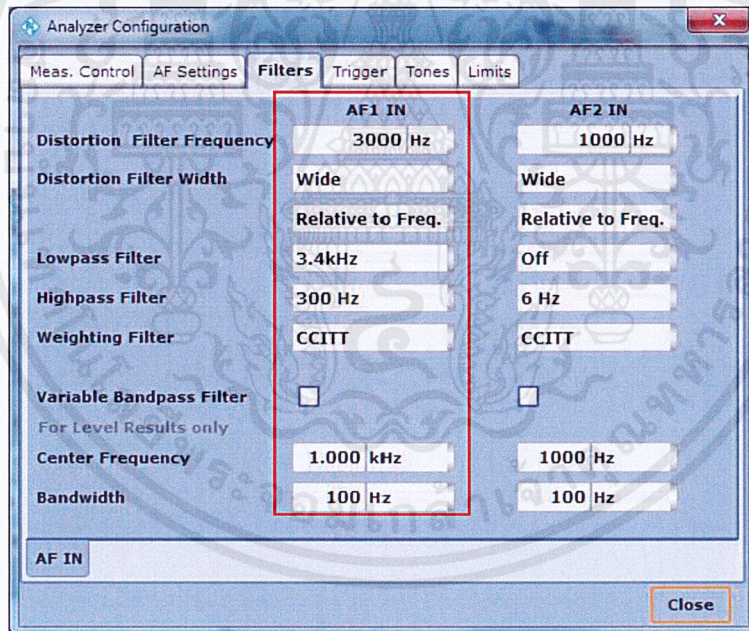
ในการทดลองนี้ค่าใน Frequency and Level ที่จะนำมาเปรียบเทียบเป็นหลัก คือค่า Average Frequency และ Std. Dev. ส่วนค่าใน Signal Quality ที่จะนำมาเปรียบเทียบเป็นหลัก คือค่า Average SINAD และ Std. Dev.

สำหรับ tab ย่อย AF Spectrum มีหน้าที่ในการแสดงกราฟสเปกตรัม (Spectrum) แสดงตัวอย่างดังรูปที่ 3.38

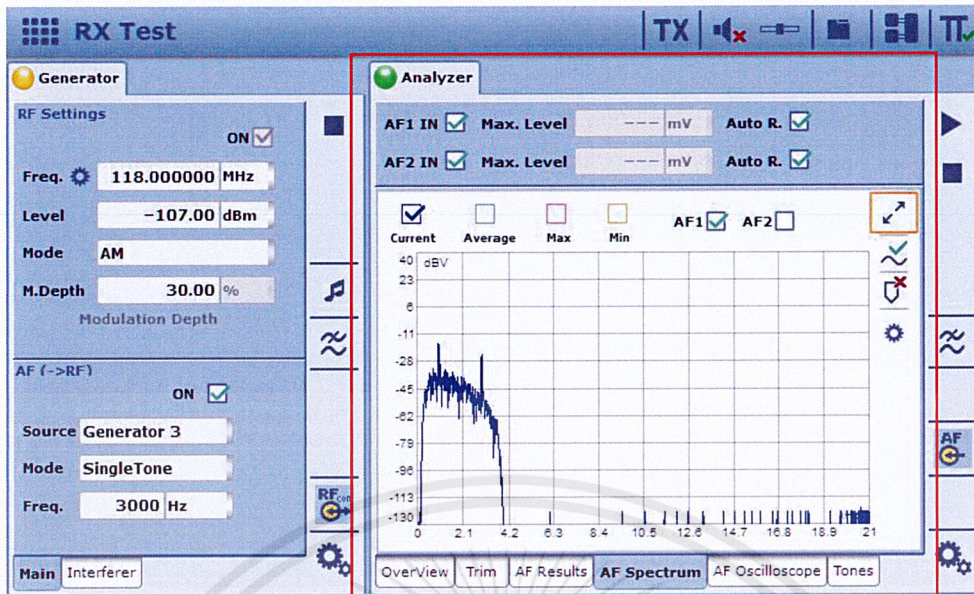
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.36 หน้าจอ Rx Test



รูปที่ 3.37 การตั้งค่าฟิลเตอร์



รูปที่ 3.38 หน้าจอ tab ย่อย AF Spectrum

3.3.11.2 การทดลองเปรียบเทียบเสียงจากระบบแอนะล็อก กับสัญญาณเสียงคลื่นรูปไซน์ ความถี่ต่าง ๆ

ผู้จัดทำได้ทำการป้อนสัญญาณเสียงคลื่นรูปไซน์ที่มีความถี่ต่าง ๆ จาก CMA 180 Radio Test Set ซึ่งเป็นคลื่นความถี่วิทยุ (Radio Frequency : RF) โดยเชื่อมต่อสาย RG214 ไปยังวิทยุเครื่องรับ Park Air Sapphire T6 Series3 และนำเสียงที่วิทยุทำการดีมอดูเลต (Demodulate) ซึ่งเป็นเสียงแอนะล็อก จาก Port MARC Audio ของวิทยุ ป้อนกลับไปยัง CMA 180 Radio Test Set ดังบล็อกไดอะแกรมในรูปที่ 3.39 เพื่อทำการวิเคราะห์สัญญาณที่ป้อนกลับเข้ามา กับที่ป้อนออกไปในตอนแรก ซึ่งค่าที่ได้นั้นจะนำไปเปรียบเทียบค่าความถี่ และ SINAD



รูปที่ 3.39 บล็อกไดอะแกรมสำหรับการทดลอง 3.3.11.2

ซึ่งตามเอกสาร ITU-T O.41 และขั้นตอนการปฏิบัติงานการซ่อมบำรุงเชิงป้องกันระบบเครื่องรับวิทยุ WI-CE.SE-01 ของทางบริษัทฯ ค่า SINAD ที่ความถี่ 1,000 Hz ควรวัดได้มากกว่า 10 dB และที่ความถี่อื่น ๆ SINAD จะลดลงตาม Relative weight ตามตารางที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.12 การทดลองใช้งานโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ ร่วมกับ GUI

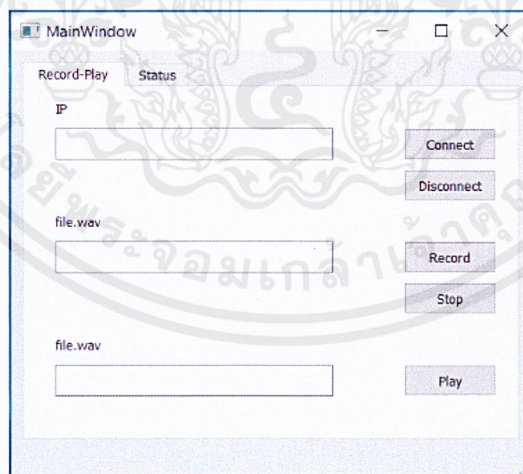
การทดลองนี้เป็นการทดสอบการใช้งานของโปรแกรมระบบรวมทั้งหมดเพื่อตรวจสอบการใช้งานว่าสามารถใช้งานได้จริงหรือไม่

โดยเริ่มจากเปิดหน้าต่าง GUI ดังรูปที่ 3.40 จากนั้น

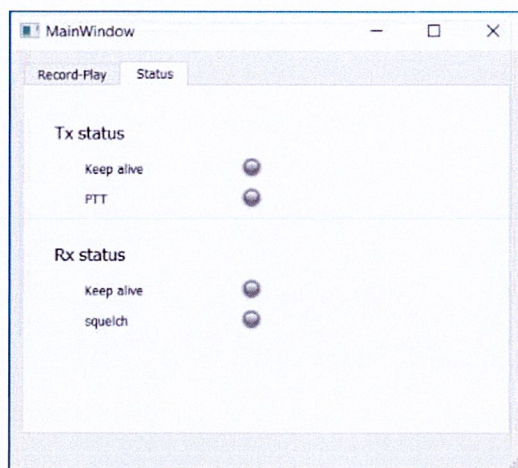
- 1) กรอกหมายเลข IP ของวิทยุที่ต้องการเชื่อมต่อ
- 2) กรอกชื่อไฟล์เสียงที่ต้องการบันทึก โดยใส่นามสกุล .wav
- 3) กดปุ่ม Connect เพื่อทำการเริ่มการรับแพ็กเก็ต VoIP ที่มาจาก IP Address ที่ได้ใส่ไว้ในข้อที่ 1 และเริ่มการสร้างไฟล์นามสกุล .wav

เมื่อกดปุ่ม Connect แล้วนั้น สามารถทำการตรวจสอบสถานะของวิทยุ ที่มี IP Address ตามที่ได้ใส่ไว้ในข้อที่ 1 ใน Status tab ดังแสดงในรูปที่ 3.41 และสามารถบันทึกเสียง และเล่นเสียงได้ต่อไป ใน Record-Play tab ดังแสดงในรูปที่ 3.40

- 4) กดปุ่ม Record เพื่อเริ่มการบันทึกเสียงที่มาจากหมายเลข IP ของวิทยุที่ต้องการ
- 5) กดปุ่ม Stop เพื่อหยุดการบันทึกเสียง
- 6) กดปุ่ม Disconnect เพื่อยกเลิกการเชื่อมต่อกับ IP Address ของวิทยุ
- 7) กรอกชื่อไฟล์เสียงที่ต้องการเล่นเสียง โดยใส่นามสกุล .wav
- 8) กดปุ่ม Play เพื่อเริ่มเล่นไฟล์เสียงที่ต้องการในข้อที่ 7



รูปที่ 3.40 หน้าจอ Record-Play tab



รูปที่ 3.41 หน้าจอ Status tab

### 3.4 สรุป

ในบทนี้ได้กล่าวถึงการออกแบบระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ มีดังนี้

การออกแบบ ซึ่งประกอบไปด้วยภาพรวมการทดลองระบบ การออกแบบการทำงานของโปรแกรม ระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ ที่ใช้งานร่วมกับหน้าต่าง GUI และการออกแบบ GUI ซึ่งได้ออกแบบ 2 tab ได้แก่ Record-Play tab และ Status tab เพื่อใช้ในส่วนของการบันทึกและเล่นไฟล์เสียง และส่วนของการตรวจสอบสถานะวิทยุตามลำดับ

เครื่องมือที่ใช้ในการทดลอง ซึ่งจะกล่าวถึงรายละเอียด และหน้าที่ของเครื่องมืออื่น ๆ ได้แก่ วิทยุ Park Air Sapphire T6 Series3 สวิตช์ HPE OfficeConnect 1920S Series Switch JL381 โปรแกรม Park Air S4 IP Controller ตัวลวดทองสัญญาณ และ CMA 180 Radio Test Set

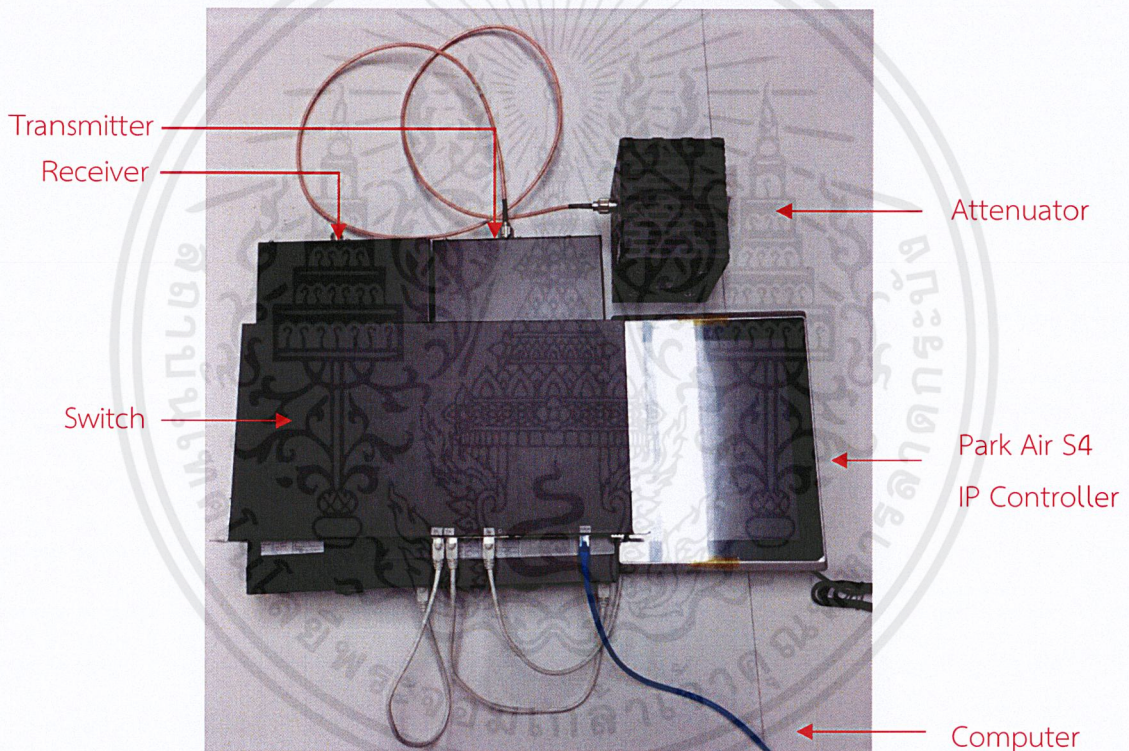
การจัดเก็บผลทดลอง ซึ่งจะประกอบไปด้วยวิธีการทดลอง ค่าที่ใช้ในการทดลอง รวมทั้งการตั้งค่าโปรแกรมต่าง ๆ โดยการทดลองในหัวข้อแรก ๆ นั้น จะเป็นการทดลองเพื่อนำมาใช้ในการทดลองเขียนโปรแกรมการดักจับแพ็กเก็ต VoIP ที่ได้จากเชื่อมต่อกับวิทยุ IP Address ที่ต้องการ เพื่อนำไปใช้ในการทดลองการบันทึก เล่นเสียง และตรวจสอบสถานะที่ได้จากแพ็กเก็ต VoIP ของวิทยุ และท้ายที่สุดจะเป็นการทดลองเปรียบเทียบคุณภาพเสียงที่ได้จากการบันทึกของโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศที่ได้จัดทำขึ้น กับเสียงที่ได้จากระบบแอนะล็อกเดิม เพื่อเป็นการตรวจสอบว่าไฟล์เสียงจากโปรแกรมที่จัดทำขึ้นมานั้นอยู่ในมาตรฐานที่สามารถนำมาใช้ได้หรือไม่

## บทที่ 4

### ผลการทดลอง

#### 4.1 ผลการทดลองการสังเกตแพ็กเก็ตที่อ่านได้จากโปรแกรม Wireshark

ผู้จัดทำได้ทำการเชื่อมต่ออุปกรณ์ทั้งหมด 6 อุปกรณ์เข้าด้วยกันได้แก่ คอมพิวเตอร์สำหรับเปิดโปรแกรม Wireshark วิทยุเครื่องรับ วิทยุเครื่องส่ง สวิตช์ และโปรแกรม Park Air S4 IP Controller รวมทั้งต่อตัวลดทอนสัญญาณเข้ากับวิทยุเครื่องส่ง ซึ่งแสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 การเชื่อมต่ออุปกรณ์ของระบบสังเกตการณ์สื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ

ผลการสังเกตแพ็กเก็ตที่อ่านได้จากโปรแกรม Wireshark สังเกตเห็นได้ว่ามีโปรโตคอลที่ให้ความสนใจอยู่ 2 โปรโตคอล ได้แก่ SIP และ RTP ซึ่งเป็นการ Session ระหว่างวิทยุ กับโปรแกรม Park Air S4 IP Controller ที่ช่วยในการสร้าง Session ต่าง ๆ

โดย IP Address ของวิทยุเครื่องรับ คือ 192.168.1.10

IP Address ของวิทยุเครื่องส่ง คือ 192.168.1.20

IP Address ของโปรแกรม Park Air S4 IP Controller คือ 192.168.1.136

จากการทดลองพบว่า Session ในการเชื่อมต่อ และการรับ-ส่งแพ็กเก็ตข้อมูลมีกระบวนการคร่าว ๆ ดังนี้

1) โพรโทคอล SIP ทำการสร้าง Session สำหรับเริ่มสร้างเส้นทางการติดต่อ โดยมีลำดับการส่งแพ็กเก็ต ข้อมูลคร่าว ๆ ดังนี้

- INVITE

ซึ่งเป็นแพ็กเก็ตที่มีต้นทาง (Source) จากโปรแกรม Park Air S4 IP Controller ที่มี IP Address เท่ากับ 192.168.1.136 ไปยังปลายทาง (Destination) ที่เป็นวิทยุเครื่องส่งที่มี IP Address เท่ากับ 192.168.1.20 เพื่อเป็นการร้องขอการเชื่อมต่อ

- 200 OK

ซึ่งเป็นแพ็กเก็ตที่มีต้นทางจากวิทยุเครื่องส่งที่มี IP Address เท่ากับ 192.168.1.20 ไปยังปลายทางที่เป็นโปรแกรม Park Air S4 IP Controller ที่มี IP Address เท่ากับ 192.168.1.136 เพื่อตอบตกลงการเชื่อมต่อ

- ACK

ซึ่งเป็นแพ็กเก็ตที่มีต้นทาง (Source) จากโปรแกรม Park Air S4 IP Controller ที่มี IP Address เท่ากับ 192.168.1.136 ไปยังปลายทาง (Destination) ที่เป็นวิทยุเครื่องส่งที่มี IP Address เท่ากับ 192.168.1.20 เพื่อเป็นการร้องขอการเชื่อมต่อ

แสดงกระบวนการดังกล่าวได้ผลดังรูปที่ 4.2

Source	Destination	Protocol	Length	Info
192.168.1.136	192.168.1.20	SIP/SDP	898	Request: INVITE sip:T6_TX@192.168.1.20:5060   Tx Radio-TxRx
192.168.1.136	192.168.1.20	SIP/SDP	898	Request: INVITE sip:T6_TX@192.168.1.20:5060   Tx Radio-TxRx
192.168.1.20	192.168.1.136	SIP/SDP	827	Status: 200 OK Tx Radio-TxRx
192.168.1.20	192.168.1.136	SIP/SDP	827	Status: 200 OK Tx Radio-TxRx
192.168.1.136	192.168.1.20	SIP	439	Request: ACK sip:T6_TX@192.168.1.20:5060
192.168.1.136	192.168.1.20	SIP	439	Request: ACK sip:T6_TX@192.168.1.20:5060

รูปที่ 4.2 โพรโทคอล SIP ทำการสร้าง Session สำหรับเริ่มสร้างเส้นทางการติดต่อ

2) โพรโทคอล RTP ที่ใช้ใน Session สำหรับการรับ-ส่งแพ็กเก็ตข้อมูล VoIP หลักจากมีการทำ Session ในการเชื่อมต่อเรียบร้อยแล้ว ซึ่งแสดงผลได้ดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและที่อยู่อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

192.168.1.136	192.168.1.10	RTP	62 PT=R2S, SSRC=0x18BE, Seq=1465, Time=2344000
192.168.1.20	192.168.1.136	RTP	62 PT=R2S, SSRC=0x78466DAF, Seq=24, Time=38400
192.168.1.20	192.168.1.136	RTP	62 PT=R2S, SSRC=0x78466DAF, Seq=24, Time=38400
192.168.1.136	192.168.1.20	RTP	62 PT=R2S, SSRC=0x29, Seq=1564, Time=2502400
192.168.1.136	192.168.1.20	RTP	62 PT=R2S, SSRC=0x29, Seq=1564, Time=2502400
192.168.1.10	192.168.1.136	RTP	62 PT=R2S, SSRC=0x78466DAF, Seq=8, Time=12800
192.168.1.10	192.168.1.136	RTP	62 PT=R2S, SSRC=0x78466DAF, Seq=8, Time=12800
192.168.1.136	192.168.1.10	RTP	62 PT=R2S, SSRC=0x18BE, Seq=1466, Time=2345600
192.168.1.136	192.168.1.10	RTP	62 PT=R2S, SSRC=0x18BE, Seq=1466, Time=2345600
192.168.1.20	192.168.1.136	RTP	62 PT=R2S, SSRC=0x78466DAF, Seq=25, Time=40000
192.168.1.20	192.168.1.136	RTP	62 PT=R2S, SSRC=0x78466DAF, Seq=25, Time=40000
192.168.1.136	192.168.1.20	RTP	62 PT=R2S, SSRC=0x29, Seq=1565, Time=2504000
192.168.1.136	192.168.1.20	RTP	62 PT=R2S, SSRC=0x29, Seq=1565, Time=2504000

รูปที่ 4.3 โพรโทคอล RTP ที่ใช้ใน Session สำหรับการรับ-ส่งแพ็กเก็ตข้อมูล VoIP

3) โพรโทคอล SIP ทำการสร้าง Session สำหรับยกเลิกเส้นทางการติดต่อ โดยมีลำดับการส่งแพ็กเก็ตข้อมูลคร่าว ๆ ดังนี้

- BYE

ซึ่งเป็นแพ็กเก็ตที่มีต้นทางจากโปรแกรม Park Air S4 IP Controller ที่มี IP Address เท่ากับ 192.168.1.136 ไปยังปลายทางที่เป็นวิทยุเครื่องรับที่มี IP Address เท่ากับ 192.168.1.10 เพื่อเป็นการร้องขอยกเลิกการเชื่อมต่อ

- 200 OK

ซึ่งเป็นแพ็กเก็ตที่มีต้นทางจากวิทยุเครื่องรับที่มี IP Address เท่ากับ 192.168.1.10 ไปยังปลายทางที่เป็นโปรแกรม Park Air S4 IP Controller ที่มี IP Address เท่ากับ 192.168.1.136 เพื่อตอบตกลงยกเลิกการเชื่อมต่อ

- ACK

ซึ่งเป็นแพ็กเก็ตที่มีต้นทางจากโปรแกรม Park Air S4 IP Controller ที่มี IP Address เท่ากับ 192.168.1.136 ไปยังปลายทางที่เป็นวิทยุเครื่องส่งที่มี IP Address เท่ากับ 192.168.1.20 เพื่อเป็นการรับทราบโดยการตอบ Acknowledge กลับไป

แสดงกระบวนการดังกล่าวได้ผลดังรูปที่ 4.2

192.168.1.136	192.168.1.10	SIP	439 Request: BYE sip:T6_RX@192.168.1.10:5060
192.168.1.136	192.168.1.10	SIP	439 Request: BYE sip:T6_RX@192.168.1.10:5060
192.168.1.10	192.168.1.136	SIP	394 Status: 200 OK
192.168.1.10	192.168.1.136	SIP	394 Status: 200 OK
192.168.1.136	192.168.1.10	SIP	439 Request: ACK sip:T6_RX@192.168.1.10:5060
192.168.1.136	192.168.1.10	SIP	439 Request: ACK sip:T6_RX@192.168.1.10:5060
192.168.1.136	192.168.1.20	SIP	439 Request: BYE sip:T6_TX@192.168.1.20:5060
192.168.1.136	192.168.1.20	SIP	439 Request: BYE sip:T6_TX@192.168.1.20:5060
192.168.1.20	192.168.1.136	SIP	394 Status: 200 OK
192.168.1.20	192.168.1.136	SIP	394 Status: 200 OK
192.168.1.136	192.168.1.20	SIP	439 Request: ACK sip:T6_TX@192.168.1.20:5060
192.168.1.136	192.168.1.20	SIP	439 Request: ACK sip:T6_TX@192.168.1.20:5060

รูปที่ 4.4 โพรโทคอล SIP ทำการสร้าง Session สำหรับยกเลิกเส้นทางการติดต่อ

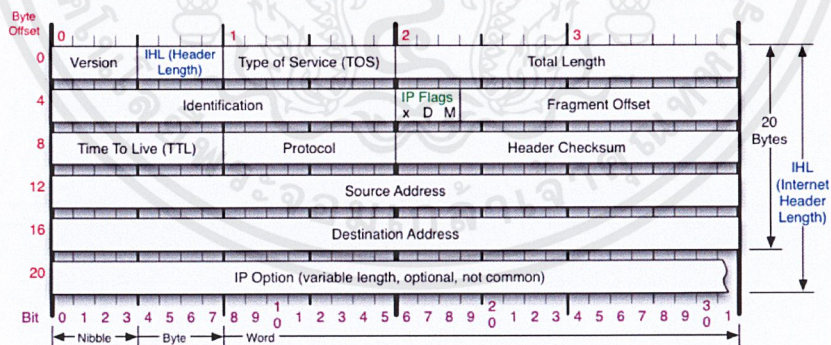
สรุป คือ โพรโทคอล SIP จะใช้ใน Session สำหรับการสร้างเส้นทางการติดต่อ และยกเลิกการติดต่อ ส่วนโพรโทคอล RTP ใช้ใน Session สำหรับการรับ-ส่งแพ็กเก็ตข้อมูล VoIP ซึ่งเป็นไปตามทฤษฎีที่ได้ศึกษาในเอกสาร ED 137

## 4.2 ผลการทดลองเขียนโปรแกรมในการดักจับแพ็กเก็ต VoIP ด้วยภาษาไพทอน

การทดสอบการดักจับแพ็กเก็ตข้อมูล โดยใช้ภาษาไพทอนนี้ ผู้จัดทำได้ทำการแยกส่วนประกอบของแพ็กเก็ตข้อมูลที่ได้รับมา 2 ส่วน ได้แก่ ส่วน Network Layer คือ IPv4 Packet และ Ethernet Layer คือ UDP Segment ซึ่งผลการทดลองแสดงผ่าน Terminal ของโปรแกรม Pycharm ดังรูปที่ 4.6

ส่วน IPv4 Packet จะระบุ

- IP Version คือ เวอร์ชันของ IP
- Header Length คือ ขนาดความยาวส่วนหัว ในหน่วย ไบต์ (bytes)
- Time To Live ในหน่วย มิลลิวินาที (ms)
- Protocol คือ โพรโทคอลที่ใช้แสดงในรูปแบบของตัวเลข เช่น 17 หมายถึง UDP ดังรูปที่ 4.5
- Source คือ IP Address ของต้นทาง
- Target คือ IP Address ของปลายทาง



<b>Version</b> Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.	<b>Protocol</b> IP Protocol ID. Including (but not limited to): 1 ICMP 17 UDP 57 SKIP 2 IGMP 47 GRE 88 EIGRP 6 TCP 50 ESP 89 OSPF 9 IGRP 51 AH 115 L2TP	<b>Fragment Offset</b> Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.	<b>IP Flags</b> x D M x 0x80 reserved (evil bit) D 0x40 Do Not Fragment M 0x20 More Fragments follow RFC 791
<b>Header Length</b> Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.	<b>Total Length</b> Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.	<b>Header Checksum</b> Checksum of entire IP header	<b>RFC 791</b> Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

รูปที่ 4.5 ไดอะแกรมแพ็กเก็ต IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ส่วน UDP Segment จะระบุ

- Source Port คือ พอร์ตต้นทาง
- Destination Port คือ พอร์ตปลายทาง
- Length คือ ขนาดความยาวของ UDP Segment ในหน่วย ไบต์
- Data คือ ข้อมูลของ UDP Segment ซึ่งภายใน Payload ของ UDP Segment ก็คือ ข้อมูล RTP แสดงในรูปของไบต์

```
- IPv4 Packet:  
- Version: 4, Header Length: 20, TTL: 128  
- Protocol: 17, Source: 192.168.44.29, Target: 192.168.47.255  
- UDP Segment:  
- Source Port: 137, Destination Port: 137, Length: 58 bytes  
- Data:  
  \x92\x30\x01\x10\x00\x01\x00\x00\x00\x00\x00\x00\x20\x45\x46\x45\x4f\x45\x49  
  \x45\x45\x46\x41\x46\x43\x45\x4a\x45\x4f\x46\x45\x43\x41\x43\x41\x43  
  \x41\x43\x41\x43\x41\x43\x41\x00\x00\x20\x00\x01
```

รูปที่ 4.6 แพ็กเก็ตข้อมูลที่ดักจับได้

### 4.3 ผลการทดลองเขียนโปรแกรมการดักจับแพ็กเก็ต VoIP ที่ได้จากวิทยุ IP Address ที่ต้องการ

ผู้จัดทำได้ทำการเชื่อมต่ออุปกรณ์ทั้งหมด 6 อุปกรณ์เข้าด้วยกันได้แก่ คอมพิวเตอร์สำหรับเปิดโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายวิทยุ สำหรับการสื่อสารการเดินอากาศ วิทยุเครื่องรับ วิทยุเครื่องส่ง สวิตช์ และโปรแกรม Park Air S4 IP Controller รวมทั้งต่อตัวลดทอนสัญญาณเข้ากับวิทยุเครื่องส่ง ซึ่งมีการเชื่อมต่อเช่นเดียวกับในการทดลองที่ 4.1 ดังรูปที่ 4.1 แตกต่างกันที่โปรแกรมที่ใช้ในคอมพิวเตอร์

ซึ่งผลการทดลองแสดงผ่าน Terminal ของโปรแกรม PyCharm โดยจะมีแยกส่วนประกอบของแพ็กเก็ตเช่นเดียวกับผลการทดลองที่ 4.2 แต่ในที่นี้จะเน้นเป็นสำเนาแพ็กเก็ตที่ได้จากการเชื่อมต่อระหว่างโปรแกรม Park Air S4 IP Controller กับตัววิทยุผ่านสวิตช์ Port mirroring

จากผลการทดลองในรูปที่ 4.7 จะเห็นได้ว่าต้นทางเป็นแพ็กเก็ตที่มาจากต้นทางที่มีเลข IP Address คือ 192.168.1.136 ที่เป็นของโปรแกรม Park Air S4 IP Controller และปลายทางเขียนหมายเลข IP Address คือ 192.168.1.10 ซึ่งเป็นของวิทยุเครื่องรับเรียบร้อย แสดงว่าการทดลองนี้สามารถดักจับแพ็กเก็ต VoIP ที่ได้จากเลข IP Address วิทยุที่ต้องการได้แล้ว

```
- IPv4 Packet:
- Version: 4, Header Length: 20, TTL: 128
- Protocol: 17, Source: 192.168.1.136, Target: 192.168.1.10
- UDP Segment:
- Source Port: 5002, Destination Port: 5002, Length: 28 bytes
- Data:
  \x90\x7b\x45\xab\x01\xb3\xc0\x00\x00\x18\xbe\x01\x67\x00\x01\x00\x00\x00
  \x00
```

รูปที่ 4.7 แพ็กเก็ตข้อมูลที่ดักจับได้จาก IP Address ของวิทยุ

#### 4.4 ผลการทดลองตรวจสอบสถานะการเชื่อมต่อ และการรับ/ส่งข้อมูลของวิทยุ

ในการทดลองนี้จะให้ความสำคัญกับโปรโตคอล RTP เนื่องจากตัวที่บ่งบอกสถานะของวิทยุที่ต้องการตรวจสอบนั้น อยู่ในส่วน Header extension ของแพ็กเก็ต RTP

สำหรับวิทยุเครื่องรับ สถานะของวิทยุที่ต้องการตรวจสอบสอบ มีดังนี้

- 1) สถานะ Keep alive ตรวจสอบได้ 2 กรณี ได้แก่
  - กรณีที่ 1 วิทยุเครื่องรับเปิดใช้งานปกติ ไม่มีการรับข้อมูลเสียง
  - กรณีที่ 2 วิทยุเครื่องรับ มีการรับข้อมูลเสียง
- 2) สถานะ Squelch (SQU) ตรวจสอบได้ ในกรณีที่เครื่องรับได้รับข้อมูลเสียง

สำหรับวิทยุเครื่องส่ง สถานะของวิทยุที่ต้องการตรวจสอบสอบ มีดังนี้

- 1) สถานะ Keep alive ตรวจสอบได้ 2 กรณี ได้แก่
  - กรณีที่ 1 วิทยุเครื่องส่งเปิดใช้งานปกติ ไม่มีการส่งข้อมูลเสียง
  - กรณีที่ 2 วิทยุเครื่องส่ง มีการส่งข้อมูลเสียง
- 2) สถานะ Push To Talk (PTT) ตรวจสอบได้ ในกรณีที่เครื่องส่ง ส่งข้อมูลเสียง

ซึ่งสถานะทั้งหมดที่ได้กล่าวมานั้น สามารถตรวจสอบได้จากบิตข้อมูล ภายใน Header extension ของแพ็กเก็ต RTP ที่มีการให้ข้อมูลของสัญญาณ หรือเรียกว่า Signalling (Real time transport protocol header extension จะประกอบไปด้วยข้อมูลของสัญญาณที่ถูกส่งมาด้วยแบบ Real time)

จากโปรแกรม Wireshark จะสามารถดูตำแหน่งของบิตข้อมูลต่าง ๆ ว่าบิตตำแหน่งไหนบ่งบอกถึงสถานะ หรือข้อมูลอะไรบ้าง จากนั้นนำข้อมูลที่ได้จากโปรแกรม Wireshark รูปที่ 4.8-4.10 มาวิเคราะห์ห์เปรียบเทียบกับโครงสร้างของแพ็กเก็ตในเอกสาร ED137 รูปที่ 4.10 จากการเปรียบเทียบจะเห็นได้ว่าตำแหน่งของบิตต่าง ๆ มีความสัมพันธ์กัน

ซึ่งจากความสัมพันธ์ดังกล่าวนี้ สามารถนำมาใช้ในการตั้งเงื่อนไขในการตรวจสอบสถานะของวิทย์ ได้แก่ Keep alive SQU และ PTT

```

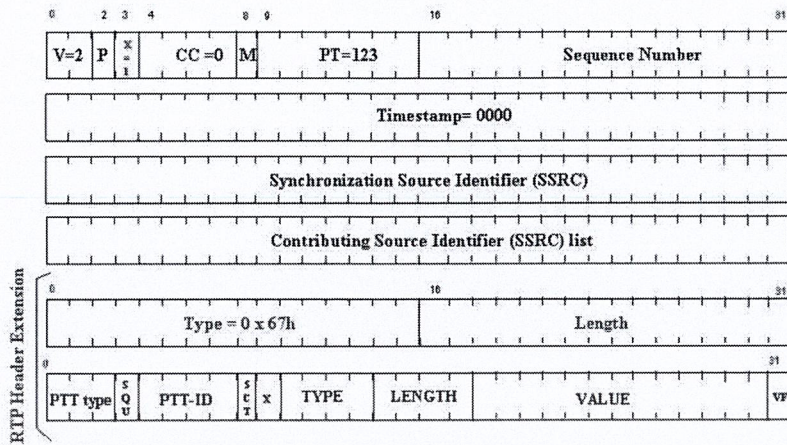
User Datagram Protocol, Src Port: 5002, Dst Port: 5002
Source Port: 5002
Destination Port: 5002
Length: 28
Checksum: 0x22ba [unverified]
[Checksum Status: Unverified]
[Stream index: 2]
[Timestamp]
Real-Time Transport Protocol
[Stream setup by SDP (frame 44)]
10.. .... = Version: RFC 1889 Version (2)
..0. .... = Padding: False
...1 .... = Extension: True
... 0000 = Contributing source identifiers count: 0
0... .... = Marker: False
Payload type: R25 (123)
Sequence number: 1659
[Extended sequence number: 67195]
Timestamp: 2654400
Synchronization Source identifier: 0x000018be (6334)
Defined by profile: ED137A (0x0167)
Extension length: 1
Header extensions
0000 00 15 8b 00 9d c3 7c cb e2 e1 67 68 08 00 45 00 | .....gh..E
0010 00 30 20 9b 00 00 80 11 96 3f c0 a8 01 88 c0 a8 | .....?.....
0020 01 0a 13 8a 13 8a 00 1c 22 ba 90 7b 06 7b 00 28 | .....{({
0030 80 c0 00 00 18 be 01 67 00 01 00 00 00 00 | .....g.....
  
```

รูปที่ 4.8 บิตข้อมูลของแพ็กเก็ต RTP

```

Header extensions
  ED137 extensions
    ED137A extension
      000. .... = PTT Type: PTT OFF (0)
      ...0 .... = SQU: SQ OFF (0)
      .... 0000 00. .... = PTT-id: 0
      .....0. .... = PTT Mute: 0
      .....0 .... = PTT Summation: 0
      ..... 0... = Simultaneous Call Transmissions: 0
      ..... ..00. = Reserved: 0x0 (0)
      ..... ..0. = X: No extended information with additional features is used (0)
    ED137A extension additional feature
      0000 0000 0000 0000 = Not used: 0
0000 00 15 8b 00 9d c3 7c cb e2 e1 67 68 08 00 45 00 | .....gh..E
0010 00 30 20 9b 00 00 80 11 96 3f c0 a8 01 88 c0 a8 | .....?.....
0020 01 0a 13 8a 13 8a 00 1c 22 ba 90 7b 06 7b 00 28 | .....{({
0030 80 c0 00 00 18 be 01 67 00 01 00 00 00 00 | .....g.....
  
```

รูปที่ 4.9 บิตข้อมูลของ Header extension ภายในแพ็กเก็ต RTP



รูปที่ 4.10 โครงสร้างของแพ็กเก็ต RTP

โดยการทดลองนี้ จะทำการเชื่อมต่ออุปกรณ์ เช่นเดียวกับในการทดลองที่ 4.3 และเขียนเงื่อนไขการตรวจสอบสถานะเพิ่มจากโปรแกรมในการทดลอง 4.3 โดยการตรวจสอบสถานะจะแสดงในรูปของสถานะ ON/OFF เช่น

ตัวอย่างที่ 1

Source : 192.168.1.10 แสดงถึง วิทยุเครื่องรับ

สถานะ Keep alive                      KEEP\_ALIVE\_STATUS : ON

สถานะ Squelch                              SQ\_STATUS : OFF

สถานะ Push To Talk (PTT)              PTT\_STATUS : OFF

หมายถึง วิทยุเครื่องรับ ที่มีหมายเลข IP 192.168.1.10 เปิดทำงานอยู่ และยังไม่มีการการรับข้อมูลเสียงเข้ามา แสดงดังรูปที่ 4.11

```

- IPv4 Packet:
  - Version: 4, Header Length: 20, TTL: 128
  - Protocol: 17, Source: 192.168.1.136, Target: 192.168.1.10
- UDP Segment:
  - Source Port: 5002, Destination Port: 5002, Length: 28 bytes
  - Data:
    \x90\x7b\x45\xab\x01\xb3\x6c\xc0\x00\x00\x18\xbe\x01\x67\x00\x01\x00\x00\x00
    \x00
- RADIO STATUS
  - KEEP_ALIVE_STATUS : ON
  - SQ_STATUS : OFF
  - PTT_STATUS : OFF

```

รูปที่ 4.11 ผลการตรวจสอบสถานะของตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้นี้เพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตัวอย่างที่ 2

Source : 192.168.1.10 แสดงถึง วิทยุเครื่องรับ

สถานะ Keep alive                      KEEP\_ALIVE\_STATUS : ON

สถานะ Squelch                         SQ\_STATUS : ON

สถานะ Push To Talk (PTT)        PTT\_STATUS : OFF

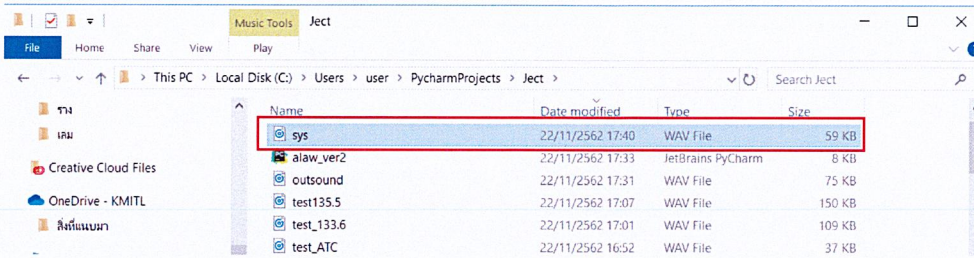
หมายถึง วิทยุเครื่องรับ ที่มีหมายเลข IP 192.168.1.10 เปิดทำงานอยู่ และมีการรับข้อมูลเสียงเข้ามา แสดงดังรูปที่ 4.12

```
= IPv4 Packet:
- Version: 4, Header Length: 20, TTL: 64
- Protocol: 17, Source: 192.168.1.10, Target: 192.168.1.136
- UDP Segment:
- Source Port: 5002, Destination Port: 5002, Length: 188 bytes
- Data:
  \x0d\x08\xc1\xce\x01\x1f\xfe\x8b\x56\xc9\xa8\x27\x01\x67\x00\x01\x18\x01\x11
  \x00\x33\x0e\x91\x6f\xe4\xe2\x01\xfa\x04\xec\x87\x81\x85\x99\x75\x8d\x0e\x04
  \x03\x0e\x9e\xb4\x80\x19\x84\x9f\x1b\x0c\x93\xe1\x18\x6f\x7e\x12\x92\x90
  \xe9\x64\x31\x92\xbd\x32\x87\xeb\xef\x04\x01\xe9\x9e\x8f\x06\x35\x05\x99\x99
  \xf\xef\xe1\xe2\xca\x80\x84\xb5\xfd\x94\xb7\x17\x00\x6e\x95\x4e\xd8\x0f\x98\x07
  \x3d\x1e\x0f\x4a\x81\x83\x88\x12\x05\xb4\xec\x1e\xb5\xf1\x84\xf8\xd8\x07\x02
  \x8c\xb4\xee\x0e\x17\x31\x63\x84\x10\x15\x05\x98\xe1\x8b\x9e\xd0\x3e\xd0\x0d
  \x80\x27\x37\x93\xf1\x95\xd4\x9b\x17\x0e\x41\x93\x64\x84\x9e\x87\x05\x91\x0a
  \xb5\x1e\x01\xe5\x36\x3f\x02\x05\x96\x9c\xf5\x1c\x92\x08\x97\xfe\x8e\x77\x08
  \x16\x0e\xf3\x81\x85\x18\x00\xb4\xb4
- RADIO STATUS
- KEEP_ALIVE_STATUS : ON
- SQ_STATUS : ON
- PTT_STATUS : OFF
```

รูปที่ 4.12 ผลการตรวจสอบสถานะของตัวอย่างที่ 2

## 4.5 ผลการทดลองการบันทึกเสียงที่ได้จากแพ็กเก็ต VoIP ของวิทยุ

การทดลองนี้ได้ทำการตั้งความถี่ของวิทยุเครื่องรับไว้ที่ 120.5 เมกกะเฮิร์ตซ์ แสดงดังรูปที่ 3.10 ผู้จัดทำได้เขียนโปรแกรมเพื่อสร้างไฟล์เสียงชื่อ sys.wav แล้วเก็บเฉพาะข้อมูลเสียงที่ได้รับจากแพ็กเก็ตของวิทยุลงไป หลังจากทำการหยุดการบันทึก เมื่อไปตรวจสอบในโฟลเดอร์ (Folder) ที่รันโปรแกรมไว้ พบว่ามีไฟล์เสียง sys.wav ถูกสร้างขึ้นเรียบร้อยดังรูปที่ 4.13 ซึ่งมีขนาดมากกว่า 0 KB แสดงว่ามีข้อมูลเสียงบันทึกลงไปแล้ว



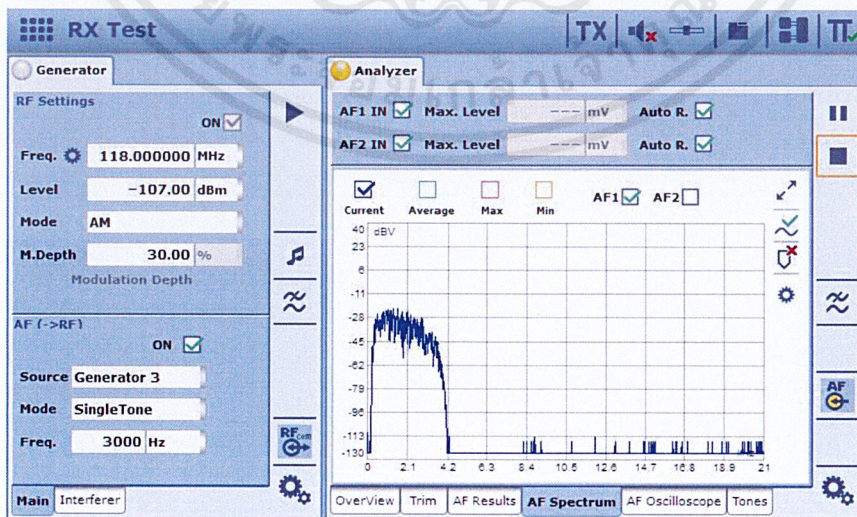
รูปที่ 4.13 ไฟล์เสียง sys.wav ที่ถูกสร้างขึ้น

#### 4.6 ผลการทดลองเล่นไฟล์เสียง

การทดลองนี้จะทำการเล่นไฟล์เสียง sys.wav ที่ได้จากการทดลองที่ 4.5 ซึ่งผลจากการฟังไฟล์เสียงนั้น ปรากฏว่ามีเสียงที่ออกมานั้นมีสัญญาณรบกวนอยู่ หรือมีเสียงซ่า ทำให้เสียงที่ได้ยินนั้นไม่ค่อยชัดเจน แต่ยังไม่พอฟังรู้เรื่องบ้าง

#### 4.7 ผลการทดลองกรองสัญญาณสัญญาณรบกวนออกจากไฟล์เสียง

การทดลองนี้ผู้จัดทำได้ทำการเลือกใช้ฟิลเตอร์ C-message weighting ในการทดลองนำไฟล์เสียง sys.wav ที่ได้จากการทดลอง 4.5 มาผ่านการกรองสัญญาณรบกวนจากฟิลเตอร์ C-message weighting ที่มีรูปร่างหน้าตาดังรูปที่ 4.14 ซึ่งมีความถี่ Cutoff ด้านต่ำอยู่ที่ประมาณ 300 เฮิรตซ์ และ ความถี่ Cutoff ด้านสูงประมาณ 4000 เฮิรตซ์ อยู่ในช่วงของความถี่เสียงพูดที่ต้องการ และเมื่อทดลองเล่นไฟล์เสียงที่ผ่านการกรองสัญญาณรบกวนออกแล้วนั้น ผลปรากฏว่า เสียงรบกวนลดลงจากเดิม ทำให้ได้ยินเสียงที่ต้องการรู้เรื่อง และชัดเจนมากขึ้น



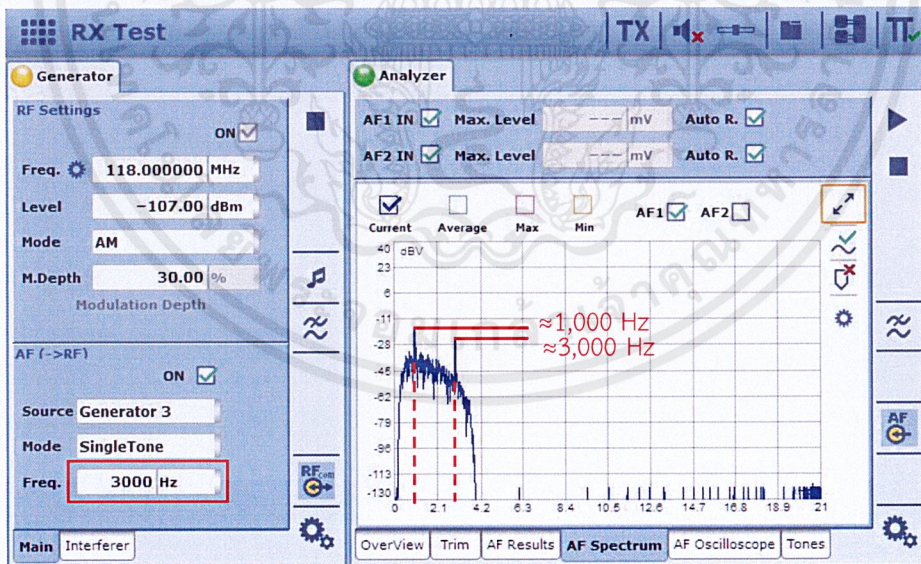
รูปที่ 4.14 รูปร่างของฟิลเตอร์ C-message weighting ที่ใช้

#### 4.8 ผลการทดลองเปรียบเทียบคุณภาพเสียงที่ได้จากการบันทึกของโปรแกรมระบบ สังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการ เดินอากาศ กับเสียงที่ได้จากระบบแอนะล็อกเดิม

4.8.1 ผลการทดลองเปรียบเทียบเสียงจากโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียง  
ผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ กับสัญญาณเสียงคลื่นรูปไซน์ ณ ความถี่  
ต่าง ๆ

ผลการเปรียบเทียบสัญญาณเสียงแบบดิจิทัลที่ผ่านโปรแกรมระบบสังเกตการณ์การสื่อสารทาง  
เสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ กับสัญญาณเสียงคลื่นรูปไซน์ที่กำหนดจาก  
CMA 180 Radio Test Set แสดงค่าเฉลี่ยความถี่ และ SINAD ในตารางที่ 4.1 และ 4.2 ตามลำดับ

ผลการทดลองพบว่า ค่าเฉลี่ยความถี่ของเสียงดิจิทัลที่ผ่านโปรแกรมระบบสังเกตการณ์การสื่อสาร  
ทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศช่วงความถี่ประมาณ 600-1,800  
เฮิร์ตซ์ มีความใกล้เคียงกับสัญญาณเสียงคลื่นรูปไซน์ที่กำหนดจาก CMA 180 Radio Test Set และค่า  
SINAD มีค่าสอดคล้องตามเอกสาร ITU-T O.41 แต่บริเวณความถี่ต่ำ และความถี่เริ่มสูงขึ้นไปนั้น พบว่ามี  
ความถี่ช่วง 1,000 เฮิร์ตซ์ ครอบคลุมอยู่ด้วย แสดงดังรูปที่ 4.15 ทำให้ค่าเฉลี่ยความถี่ และค่า SINAD จาก  
Analyzer ไม่เป็นไปตามสมควร



รูปที่ 4.15 สเปกตรัมของสัญญาณคลื่นรูปไซน์ที่ 3,000 เฮิร์ตซ์

สาเหตุนี้มาจากฟิลเตอร์ที่ออกแบบ จากการทดลองได้ทำการจำลองหน้าตาของฟิลเตอร์ที่ใช้ โดยทำการสร้างสัญญาณรบกวนแบบ White Noise ที่เป็นสัญญาณรบกวนทุกความถี่ แล้วป้อนเข้าไปในฟิลเตอร์ที่ได้ออกแบบไว้ จึงได้หน้าตาฟิลเตอร์ที่ออกแบบแสดงดังรูปที่ 4.14

จากรูปที่ 4.14 จะเห็นได้ว่ารูปร่างของฟิลเตอร์นั้นในช่วงความถี่สูง ประมาณ 2,500-3,000 เฮิรตซ์ นั้นลดลงกว่า จึงทำให้สัญญาณรบกวนในช่วง 1,000 เฮิรตซ์ แรงกว่า ซึ่งส่งผลต่อค่าเฉลี่ยความถี่ที่เครื่องวิเคราะห์ออกมาเป็นค่าเฉลี่ยความถี่ระหว่างความถี่ที่ควรได้กับความถี่รบกวน และส่งผลให้ SINAD ไม่ได้ค่าตามที่ควร

#### 4.8.2 ผลการทดลองเปรียบเทียบเสียงจากระบบแอนะล็อก กับสัญญาณเสียงคลื่นรูปไซน์ ณ ความถี่ต่าง ๆ

ผลการเปรียบเทียบความถี่ และ SINAD ของเสียงแอนะล็อกที่ออกจาก Port MARC Audio กับสัญญาณเสียงคลื่นรูปไซน์ที่กำเนิดจาก CMA 180 Radio Test Set แสดงค่าความถี่ และ SINAD ในตารางที่ 4.1 และ 4.2 ตามลำดับ

ผลการทดลองพบว่า ค่าเฉลี่ยความถี่ของเสียงแอนะล็อกที่ออกจาก Port MARC Audio นั้นมีความใกล้เคียงกับสัญญาณเสียงคลื่นรูปไซน์ที่กำเนิดจาก CMA 180 Radio Test Set และค่า SINAD มีค่าสอดคล้องตามเอกสาร ITU-T ฉบับ O.41 ทั้งช่วงความถี่ตั้งแต่ 400-3,000 เฮิรตซ์

ตารางที่ 4.1 ค่าความถี่ของเสียงระบบแบบแอนะล็อก และดิจิตอล

ความถี่ (เฮิรตซ์)	แอนะล็อก		ดิจิตอล	
	ความถี่ (กิโลเฮิรตซ์)		ความถี่ (กิโลเฮิรตซ์)	
	Avg.	Std Dev.	Avg.	Std Dev.
400	0.4568	0.00485	0.6327	0.0861
600	0.6199	0.0317	0.6250	0.0306
800	0.8062	0.0206	0.8026	0.0130
1,000	1.0032	0.0141	1.0054	0.0169
1,800	1.7994	0.0092	1.8000	0.0166
2,500	2.4991	0.0059	1.7995	0.0128
3,000	2.9937	0.0223	1.6136	0.1263

ตารางที่ 4.2 ค่า SINAD ของเสียงระบบแบบแอนะล็อก และดิจิทัล

ความถี่ (เฮิรตซ์)	แอนะล็อก		ดิจิทัล	
	SINAD (dB)		SINAD (dB)	
	Avg.	Std Dev.	Avg.	Std Dev.
400	6.68	0.09	2.08	0.08
600	10.44	0.19	6.52	0.12
800	13.20	0.22	10.45	0.20
1,000	14.05	0.15	12.91	0.31
1,800	10.78	0.14	7.68	0.32
2,500	8.29	0.28	1.41	0.02
3,000	6.15	0.21	0.59	0.02

จากผลการทดลองที่ 4.8.1 และ 4.8.2 จะเห็นได้ว่าค่าเฉลี่ยความถี่ และ SINAD ของเสียงจากระบบแอนะล็อกเดิม เมื่อเปรียบเทียบกับเสียงดิจิทัลที่ผ่านโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศนั้น ถึงแม้คุณภาพเสียงของระบบแอนะล็อกยังคงดีกว่าระบบดิจิทัล แต่ก็มีค่าใกล้เคียงกัน และ ณ ความถี่ที่สนใจที่ 1,000 เฮิรตซ์นั้น เสียงดิจิทัลที่ผ่านโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศมีค่ามากกว่า 10 dB ซึ่งถือว่าผ่านมาตรฐานตามเอกสารขั้นตอนการปฏิบัติงาน (Work Instruction) การซ่อมบำรุงเชิงป้องกันระบบเครื่องรับวิทยุ WI-CE.SE-01 ของทางบริษัทฯ จึงสามารถสรุปได้ว่า ระบบดิจิทัลนั้นสามารถนำมาใช้งานได้

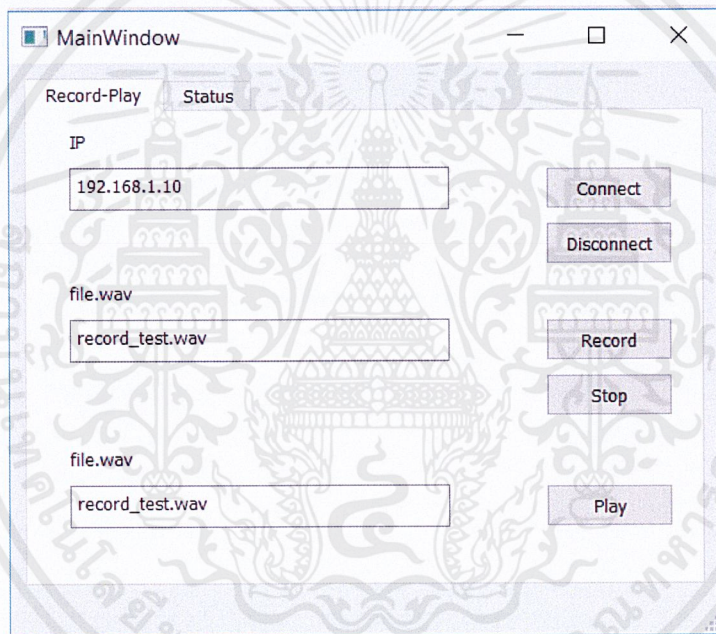
#### 4.9 ผลการทดลองใช้งานโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ ร่วมกับ GUI

ผู้จัดทำได้ทดลองนำโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ ประกอบเข้ากับการแสดงผลผ่านทางหน้าต่าง GUI ซึ่งมี Flow chart การทำงานเป็นไปดังรูปที่ 3.1 โดยในการทดลองนี้จะทำการทดลองรับเสียงจากวิทยุ 1 เครื่อง และทำการบันทึก และเล่นเสียงผ่านชื่อไฟล์ .wav ที่ต้องกรอกชื่อไฟล์ ก่อนเริ่มการทำงานของโปรแกรม โดยมีการทดลองดังต่อไปนี้

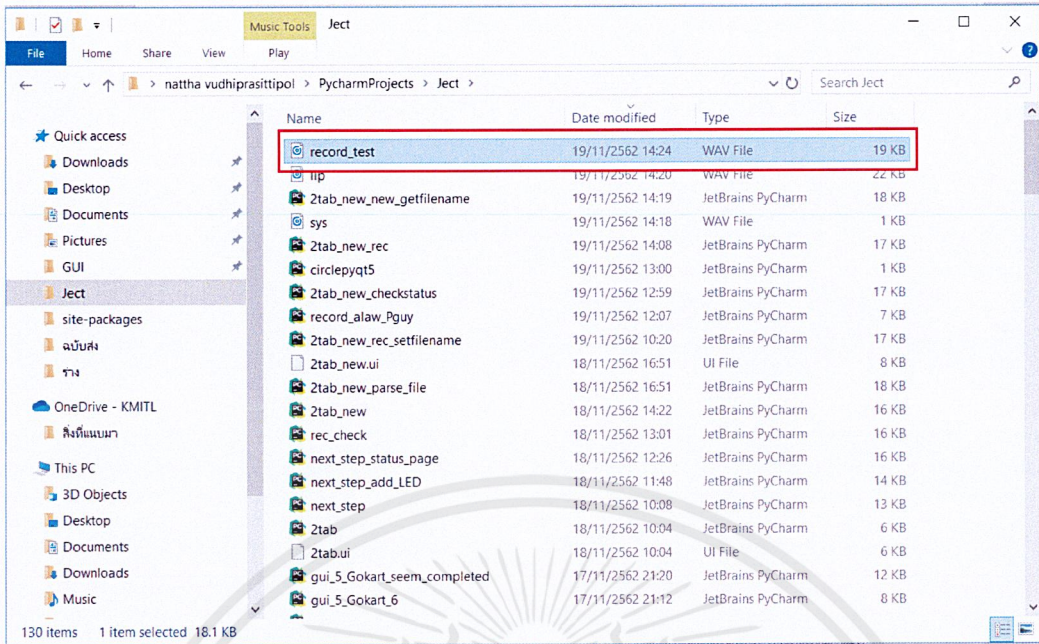
เริ่มจาก Record-Plat tab ผู้จัดทำได้หนด IP คือ 192.168.1.10 ซึ่งเป็นของวิทยุเครื่องรับ และชื่อไฟล์ที่ต้องการบันทึก คือ record\_test.wav รับสัญญาณในย่านความถี่ 118.0 เมกกะเฮิรตซ์ ดังรูปที่ 4.16 เมื่อต้องการเริ่มบันทึกเสียง จึงกดปุ่ม Record เพื่อทำการบันทึกเสียงจาก VoIP แพ็กเก็ต ที่มาจากวิทยุเครื่องรับ IP Address นั้น และกดปุ่ม Stop เมื่อต้องการหยุดบันทึกเสียง ไฟล์ record\_test.wav จะถูก

เขียนเสร็จเรียบร้อย ดังรูปที่ 4.17 จากนั้นให้ทำการกำหนดชื่อไฟล์ที่ต้องการจะเล่นเสียง ในที่นี้ต้องการเล่นเสียงไฟล์ record\_test.wav ทำการกดปุ่ม Play เพื่อทำการฟังไฟล์เสียงที่ได้บันทึกไปก่อนหน้านี้

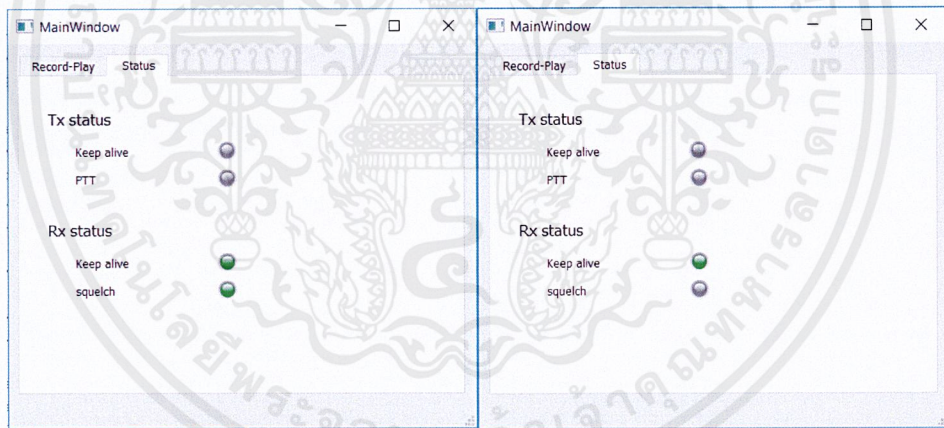
สำหรับใน Status tab เมื่อผู้จัดทำกดปุ่ม Connect ใน Record-Play tab เพื่อเชื่อมต่อกับวิทยุที่มีเลข IP ที่ต้องการ คือ 192.168.1.10 ซึ่งเป็นวิทยุเครื่องรับ ดังนั้นสถานะของวิทยุที่สังเกตนั้นจะบอกได้เพียงสถานะของวิทยุเครื่องรับที่มี IP 192.168.1.10 โดยจะเห็นได้ว่าส่วนของ Rx status แสดง LED ของสถานะ Keep alive เป็นสีเขียว หมายถึงวิทยุเครื่องรับนี้ทำงานอยู่ ส่วนสถานะ Squelch เมื่อมีเสียงส่งมาแล้วเครื่องรับได้รับเสียง LED ของสถานะ Squelch จะเป็นสีเขียว หมายถึงวิทยุเครื่องรับนี้ได้รับข้อมูลเสียง แสดงดังรูปที่ 4.18 (a) แต่ถ้าไม่ได้รับเสียง LED ของสถานะ Squelch จะเป็นสีเทา หมายถึงวิทยุเครื่องรับนี้ไม่ได้รับข้อมูลเสียง ซึ่งแสดงดังรูปที่ 4.18 (b)



รูปที่ 4.16 Record-Play tab ของ GUI เมื่อมีการกำหนด IP และชื่อไฟล์เสียงนามสกุล .wav



รูปที่ 4.17 ไฟล์เสียง record\_test.wav ถูกสร้างขึ้น



(a)

(b)

รูปที่ 4.18 การแสดงผลของ Status tab สำหรับวิทยุเครื่องรับ

## 4.10 สรุป

ในบทนี้ได้แสดงผลการทดลองการสังเกตแพ็กเก็ตในโปรแกรม Wireshark เพื่อนำไปใช้ประกอบกับการทดลองอื่น ๆ และผลการทำงานของโปรแกรมสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศทั้งในหน้าจอ Terminal ของโปรแกรม Pycharm และที่ทำงานร่วมกับหน้าต่าง GUI รวมทั้งมีผลการทดสอบคุณภาพเสียงระหว่างเสียงระบบแอนะล็อกกับดิจิทัล ซึ่งได้อภิปรายผลในข้างต้นอย่างละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผล

โครงการฉบับนี้ ได้รายงานถึงปัญหาด้านข้อจำกัดของจำนวนอุปกรณ์ ในระบบแอนะล็อกเดิม ทำให้บริษัทฯ เห็นความสำคัญของการนำการสื่อสารระบบดิจิทัลผ่านระบบ VoIP แต่ยังคงขาดเครื่องมือในการสังเกตการณ์การสื่อสารการเดินอากาศผ่านระบบ VoIP ซึ่งปัญหานี้ได้รับการแก้ไขด้วยการพัฒนาโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต

ในการทดลองระบบการสื่อสารการทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศที่ได้จัดทำขึ้นนั้น มีการเชื่อมต่ออุปกรณ์ทั้งหมด 6 อุปกรณ์ ได้แก่ คอมพิวเตอร์ที่รองรับการทำงานโปรแกรมระบบการสื่อสารการทางเสียงผ่านโครงข่ายอินเทอร์เน็ต ที่ได้จัดทำ วิทย์เครื่องรับ Park Air Sapphire T6 Series 3 วิทย์เครื่องส่ง Park Air Sapphire T6 Series 3 สวิตช์ HPE OfficeConnect 1920S Series Switch JL381 และโปรแกรม Park Air S4 IP Controller รวมทั้งตัวลดทอนสัญญาณที่ต่อเข้ากับวิทย์เครื่องส่งเพื่อป้องกันการรบกวนการทำงานจริงของทางบริษัทฯ ซึ่งได้นำอุปกรณ์เหล่านี้มาจัดทำ การทดลองต่าง ๆ ได้แก่ การเชื่อมต่อ การตรวจสอบสถานะของวิทย์ การบันทึกเสียง และเล่นเสียง จากนั้นได้นำการทดลองข้างต้นนี้ มาทำงานร่วมกัน และแสดงผลผ่าน GUI นอกจากนี้ยังได้มีการใช้เครื่อง CMA 180 Radio Test Set สำหรับทดลองเกี่ยวกับคุณภาพสัญญาณในการเปรียบเทียบระหว่างคุณภาพเสียงที่ได้จากระบบแอนะล็อก และดิจิทัล เพื่อให้มั่นใจว่าเสียงที่ได้บันทึกมานั้นอยู่ในเกณฑ์มาตรฐานที่สามารถใช้ได้

จากผลการทดลองนั้น พบว่าโปรแกรมระบบการสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศ สามารถสังเกตสถานการณ์เชื่อมต่อ สถานะการรับ/ส่งข้อมูลเสียงของวิทย์ได้ และสามารถสังเกตเสียงจากการบันทึก และเล่นไฟล์เสียงของวิทย์ได้ สำหรับผลการทดลองเรื่องคุณภาพเสียงนั้น พบว่าถึงแม้คุณภาพเสียงของระบบแอนะล็อกจะดีกว่า แต่คุณภาพเสียงที่ได้จากระบบดิจิทัลยังอยู่ในเกณฑ์ที่สามารถใช้งานได้ตามมาตรฐาน ดังนั้นโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศที่จัดทำขึ้นนั้นสามารถใช้งานได้

โดยโปรแกรมระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศที่จัดทำขึ้นนั้น สามารถแก้ไขปัญหาเรื่องของการขาดเครื่องมือที่ใช้ในการสังเกตการณ์การสื่อสารการเดินอากาศผ่านระบบ VoIP และยังสามารถนำไปประยุกต์ใช้กับสวิตช์ที่มีการเชื่อมต่อระหว่างสวิตช์ด้วยกัน ทำให้สามารถเลือกสังเกตการณ์วิทย์ได้หลายเครื่องมากขึ้น นอกจากนี้ยังช่วยในเรื่องของการลดจำนวน และค่าใช้จ่ายด้านอุปกรณ์ที่ใช้ได้อีกด้วย

### 5.3 ประโยชน์ที่ได้จากโครงการ

- 1) ช่วยลดจำนวนอุปกรณ์ที่ใช้ในการสังเกตการณ์ระบบการสื่อสารการเดินอากาศ
- 2) ช่วยลดค่าใช้จ่ายที่เกิดจากเครื่องมือที่ใช้ในการสังเกตการณ์ระบบการสื่อสารการเดินอากาศผ่านระบบ VoIP เนื่องจากเป็นอุปกรณ์ที่มีอยู่แล้ว และสามารถหาใช้ได้ง่าย
- 3) ช่วยให้สามารถสังเกตการณ์ระบบการสื่อสารการเดินอากาศทางเสียงผ่านระบบ VoIP ได้
- 4) ได้ความรู้ด้านการสื่อสารทางเสียงผ่านระบบ VoIP
- 5) ได้ความรู้ และประสบการณ์ด้านการเขียนโปรแกรม และ GUI โดยใช้ภาษาไพทอน

### 5.4 ปัญหาและอุปสรรค

การทำโครงการในครั้งนี้ มีอุปสรรคในการทำงานค่อนข้างหลายด้าน ทั้งในเรื่องของระยะเวลาในการดำเนินการโครงการ เนื่องจากมีเวลาค่อนข้างจำกัด การศึกษามาตรฐานในการสื่อสารผ่านระบบ VoIP ของระบบวิทยุการเดินอากาศที่ต้องศึกษาเพิ่มเติม เนื่องจากเป็นเรื่องที่มีบุคลากรในบริษัทฯ ให้คำปรึกษาได้น้อยราย ส่งผลให้ต้องใช้เวลาในการศึกษาข้อมูลค่อนข้างนาน อีกทั้งในด้านฮาร์ดแวร์ และซอฟต์แวร์ ซึ่งในบางครั้งต้องมีการขอใช้งานอุปกรณ์ หรือความช่วยเหลือด้านการ Config ค่าพารามิเตอร์ต่าง ๆ ซึ่งในขณะนั้นบริษัทฯ ได้มีการส่งบุคลากรออกไปติดตั้งระบบวิทยุนอกพื้นที่ในหลายจังหวัด ซึ่งใช้คนจำนวนมาก ส่งผลให้เกิดความล่าช้าในการทำงาน นอกจากนี้แล้วยังต้องมีการเรียนรู้ภาษาโปรแกรมอื่น ๆ ที่ต้องประยุกต์ใช้เพิ่มมากขึ้นด้วย

ด้วยข้อจำกัดทางด้านเวลา ทำให้ระบบที่สร้างขึ้นนั้นยังมีข้อจำกัดบางอย่างที่ยังทำการศึกษาไม่เสร็จ เช่น การสังเกตการณ์เสียงแบบ Real time และคุณภาพของฟิลเตอร์ที่นำมาใช้เป็นต้น

### 5.5 แนวทางในการพัฒนาต่อ

ในการนำระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต สำหรับการสื่อสารการเดินอากาศที่ได้จัดทำขึ้นไปใช้งานได้จริง ๆ นั้น ระบบควรได้รับการพัฒนาการสังเกตการณ์ทางเสียงแบบ Real time และคุณภาพของเสียง รวมทั้งพัฒนา GUI ให้สามารถสังเกตการณ์การสื่อสารได้ที่หลากหลาย ๆ ช่องสัญญาณ และมีความเสถียรคงที่ เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพ

## บรรณานุกรม

- [1] PhD. Sarayut Nonsiri. *บทความ Python คืออะไร*. แหล่งข้อมูล : <https://www.9experttraining.com/articles/python-คืออะไร>, สืบค้นเมื่อ 27 พฤศจิกายน 2562.
- [2] ครูแอน. *คุณสมบัติของเสียง*. แหล่งข้อมูล : <http://www.prd.go.th/download/12Digital%20audio.pdf>, สืบค้นเมื่อ 7 ธันวาคม 2562.
- [3] *การสื่อสารทางวิทยุ*. แหล่งข้อมูล : [https://www.google.com/url?sa=t&rct=j&q=&esrc=ssource=web&cd=8&cad=rja&uact=8&ved=2ahUKEwj44q2a0ormAhW\\_ILcAHcooCi8QFjAHegQICRAC&url=http%3A%2F%2Fwww.pattayatech.ac.th%2Ffiles%2F150511088445365\\_15051115151352.docx&usq=AOvVaw2L-FZUfFt2Q KPX1A5hjjNk](https://www.google.com/url?sa=t&rct=j&q=&esrc=ssource=web&cd=8&cad=rja&uact=8&ved=2ahUKEwj44q2a0ormAhW_ILcAHcooCi8QFjAHegQICRAC&url=http%3A%2F%2Fwww.pattayatech.ac.th%2Ffiles%2F150511088445365_15051115151352.docx&usq=AOvVaw2L-FZUfFt2Q KPX1A5hjjNk), สืบค้นเมื่อ 7 ธันวาคม 2562.
- [4] *กระบวนการทางดิจิทัล*. แหล่งข้อมูล : <https://www.utt.ac.th/new/elearning/chf/chf-h8l8.pdf>, สืบค้นเมื่อ 7 ธันวาคม 2562.
- [5] เอกลักษณ์ โสภานันท์. *Voice over IP (VoIP) คืออะไร*. แหล่งข้อมูล : <https://www.gotoknow.org/posts/300860>, สืบค้นเมื่อ 11 กันยายน 2562.
- [6] กุลวิทย์ วิไลลักษณ์, สุมนิตา สุภธีระและอุษารัตน์ จุ้ยทองกลาง. *ขั้นตอนการทำงานของ VoIP*. แหล่งข้อมูล : <https://sites.google.com/site/voipvoipgroup/1-hlak-kar-thangan-laea-sthapatykrmm/khan-txn-kar-thangan-khxng-voip>, สืบค้นเมื่อ 11 กันยายน 2562.
- [7] Phavadol Srisarnsakul. *Air traffic management*. แหล่งข้อมูล : <https://www.slideshare.net/hachimae/air-traffic-management-79045476>, สืบค้นเมื่อ 7 ธันวาคม 2562.
- [8] EUROCAE. *Interoperability Standards for VoIP ATM Components – Part 1:Radio*. 2009
- [9] Manus. *การดักจับและวิเคราะห์ข้อมูลในระบบเครือข่ายคอมพิวเตอร์*. แหล่งข้อมูล : <https://www.uni.net.th/backend/uninews/news.php?name=2013022858>, สืบค้นเมื่อ 11 กันยายน 2562.
- [10] *Wireshark*. แหล่งข้อมูล : <https://software.thaiware.com/9460-Wireshark-Download.html>, สืบค้นเมื่อ 11 กันยายน 2562.
- [11] *What is port mirroring?*. แหล่งข้อมูล : <https://www.miarec.com/faq/what-is-portmirroring>, สืบค้นเมื่อ 11 กันยายน 2562.
- [12] *VoIP ทำงานอย่างไร*. แหล่งข้อมูล : <https://www.ippbxthai.com/2013/03/voipทำงานอย่างไร/>, สืบค้นเมื่อ 11 กันยายน 2562.
- [13] *Pulse Code Modulation*. แหล่งข้อมูล : <https://www.jaredsec.com/pcm1/>, สืบค้นเมื่อ 28 ตุลาคม 2562.
- [14] *Telecommunications Weighting Filters*. แหล่งข้อมูล : <http://zone.ni.com/reference/en-XX/help/372416L-01/svtconcepts/svteleweighting/>, สืบค้นเมื่อ 21 พฤศจิกายน 2562.
- [15] International Telecommunication Union. *PSOPHOMETER FOR USE ON TELEPHONE TYPE CIRCUITS*. In Annex A Comparison of ITU-T and North American weightings. pp. 8, 1972.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม (ต่อ)

- [16] นายพิบูลเศรษฐ์ สมบัติธีระ. *ขั้นตอนการปฏิบัติงาน (Work Instruction) การซ่อมบำรุงเชิงป้องกัน ระบบเครื่องรับวิทยุ*. ในการตรวจวัด Line Audio Output : SINAD, หน้า 31.
- [17] *การวัด SINAD sensitivity*. แหล่งข้อมูล : <http://www.153rt.com/index.php?topic=5566.0>, สืบค้นเมื่อ 11 ธันวาคม 2562.
- [18] *Northrop Grumman Park Air Systems*. แหล่งข้อมูล : <https://www.airport-suppliers.com/supplier/northrop-grumman-park-air-systems/>, สืบค้นเมื่อ 16 กันยายน 2562.
- [19] *HPE OfficeConnect 1920S Switch Series Installation and Getting Started Guide*. แหล่งข้อมูล : <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=8&cad=rja&uact=8&ved=2ahUKEwj41dOlpvrlAhXoxzgGHRHFB6wQFjAHegQIAxAC&url=http%3A%2F%2Fdwn.alza.cz%2Fmanual%2F40326&usg=AOvVaw3YF6UO6xRPCaHyyzl0224M>, สืบค้นเมื่อ 16 กันยายน 2562.
- [20] *Park Air S4 IP Controller*. แหล่งข้อมูล : <https://www.parkairsystems.com/components/s4-ip-controller>, สืบค้นเมื่อ 16 กันยายน 2562.
- [21] *R&S@CMA180 Radio Test Set User Manual*. แหล่งข้อมูล : [https://seminars.rohde-schwarz.co.jp/file/man/CMA180\\_UserManual\\_V1-0-30\\_en\\_05.pdf](https://seminars.rohde-schwarz.co.jp/file/man/CMA180_UserManual_V1-0-30_en_05.pdf), สืบค้นเมื่อ 21 พฤศจิกายน 2562.
- [22] สมาคมวิทยุสมัครเล่นจังหวัดเพชรบูรณ์ HS6AB. *ดัมมี่โหลด (Dummy Load) คืออะไร*. แหล่งข้อมูล : <https://www.facebook.com/hs6ab/posts/532461020190941>, สืบค้นเมื่อ 16 กันยายน 2562.



ภาคผนวก

คำสั่งการทำงานของระบบสังเกตการณ์การสื่อสารทางเสียงผ่านโครงข่ายอินเทอร์เน็ต  
สำหรับการสื่อสารการเดินทางอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import os
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtCore import Qt, QThread, pyqtSignal
from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout
from QLed import QLed
import socket
import struct
import textwrap
import wave
from playsound import playsound
from scipy.signal import lfilter
from scipy.signal import bilinear
from numpy import pi, polymul
import soundfile as sf

```

```

sampleRate = 8000.0 # hertz
duration = 1.0 # seconds
frequency = 8000.0 # hertz

```

```

TAB_1 = '\t - '
TAB_2 = '\t\t - '
TAB_3 = '\t\t\t - '
TAB_4 = '\t\t\t\t - '

```

```

DATA_TAB_1 = '\t '
DATA_TAB_2 = '\t\t '
DATA_TAB_3 = '\t\t\t '
DATA_TAB_4 = '\t\t\t\t '

```

```

class MyThread(QThread):

```

```

    global con_discon, rec_stop
    change_Tx_keep_led = pyqtSignal(bool)
    change_Rx_keep_led = pyqtSignal(bool)
    change_PTT_led = pyqtSignal(bool)
    change_Squelch_led = pyqtSignal(bool)
    con_discon = None
    rec_stop = None
    def run(self):

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

global con_discon, rec_stop, record_text, IP_text
print("main start")
HOST = socket.gethostbyname(socket.gethostname())
print(HOST)
conn = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_IP)
conn.bind((HOST, 0))
conn.ioctl(socket.SIO_RCVALL, socket.RCVALL_ON)
obj = wave.open(record_text, 'w')
obj.setnchannels(1) # mono
obj.setsampwidth(1)
obj.setframerate(sampleRate)
while True:
    raw_data, addr = conn.recvfrom(65535)
    (version, header_length, ttl, proto, src, target, data) = ipv4_packet(raw_data)
    print(TAB_1 + 'IPv4 Packet:')
    print(TAB_2 + 'Version: {}, Header Length: {}, TTL: {}'.format(version,
        header_length, ttl))
    print(TAB_2 + 'Protocol: {}, Source: {}, Target: {}'.format(proto, src, target))
    if src == IP_text:
        # UDP
        if proto == 17:
            src_port, dest_port, size, data_udp, voice_data = udp_segment(data)
            print(TAB_1 + 'UDP Segment: ')
            print(TAB_2 + 'Source Port: {}, Destination Port: {}, Length: {}
                bytes'.format(src_port, dest_port, size))
            print(TAB_2 + 'Data: ')
            print(format_multi_line(DATA_TAB_3, data_udp))
            print(TAB_2 + 'Voice Data: ')
            print(TAB_1 + 'RADIO STATUS')
            pt_status = keep_alive_status(data_udp)
            if pt_status == 123:
                print(TAB_2 + 'KEEP_ALIVE_STATUS : ON')
                self.change_Rx_keep_led.emit(True)
                sq_status = squelch_status(data_udp)
                if sq_status == 1:
                    print(TAB_2 + 'SQ_STATUS : ON')
                    self.change_Squelch_led.emit(True)

```

```

else:
    print(TAB_2 + 'SQ_STATUS : OFF')
    self.change_Squelch_led.emit(False)
ptt_status = push_to_talk_status(data_udp)
if ptt_status == 1:
    print(TAB_2 + 'PTT_STATUS : ON')
    self.change_PTT_led.emit(True)
else:
    print(TAB_2 + 'PTT_STATUS : OFF')
    self.change_PTT_led.emit(False)
elif pt_status == 0 or pt_status == 8:
    print(TAB_2 + 'KEEP_ALIVE_STATUS : ON')
    self.change_Tx_keep_led.emit(True)
    sq_status = squelch_status(data_udp)
    if sq_status == 1:
        print(TAB_2 + 'SQ_STATUS : ON')
        self.change_Squelch_led.emit(True)
    else:
        print(TAB_2 + 'SQ_STATUS : OFF')
        self.change_Squelch_led.emit(False)
    ptt_status = push_to_talk_status(data_udp)
    if ptt_status == 1:
        print(TAB_2 + 'PTT_STATUS : ON')
        self.change_PTT_led.emit(True)
    elif ptt_status == 3:
        print(TAB_2 + 'PTT_STATUS : PRIORITY')
    elif ptt_status == 4:
        print(TAB_2 + 'PTT_STATUS : EMERGENCY')
    else:
        print(TAB_2 + 'PTT_STATUS : OFF')
        self.change_PTT_led.emit(False)
else:
    print(TAB_2 + 'KEEP_ALIVE_STATUS : OFF')
    self.change_Tx_keep_led.emit(False)
    self.change_Rx_keep_led.emit(False)
if rec_stop == True:
    print("Start Record")
    if len(voice_data) > 0:

```

```

print(type(voice_data))
# convert data
for i in range(len(voice_data)):
    if voice_data[i] > 127:
        mant = voice_data[i] & 0x7F
    else:
        mant = -voice_data[i]
    data1 = struct.pack('<b', mant)

    # save data to file
    obj.writeframesraw(data1)

if rec_stop == False:
    print("Stop Record")
    obj.close()
    rec_stop = None

# Other
else:
    print(TAB_1 + 'Data: ')
    print(format_multi_line(DATA_TAB_2, data))

if con_discon == True:
    print("IP Disconnect")
    self.change_Tx_keep_led.emit(False)
    self.change_Rx_keep_led.emit(False)
    self.change_PTT_led.emit(False)
    self.change_Squelch_led.emit(False)
    break

```

```
class Ui_MainWindow(object):
```

```
    def setupUi(self, MainWindow):
```

```
        MainWindow.setObjectName("MainWindow")
```

```
        MainWindow.resize(509, 415)
```

```
        self.centralwidget = QtWidgets.QWidget(MainWindow)
```

```
        self.centralwidget.setObjectName("centralwidget")
```

```
        self.gridLayout = QtWidgets.QGridLayout(self.centralwidget)
```

```
        self.gridLayout.setObjectName("gridLayout")
```

```
        self.Rec_Play_Status_Tab = QtWidgets.QTabWidget(self.centralwidget)
```

```
        self.Rec_Play_Status_Tab.setObjectName("Rec_Play_Status_Tab")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

self.Record_Play_tab = QtWidgets.QWidget()
self.Record_Play_tab.setObjectName("Record_Play_tab")
self.Record_textEdit = QtWidgets.QTextEdit(self.Record_Play_tab)
self.Record_textEdit.setGeometry(QtCore.QRect(30, 150, 271, 31))
self.Record_textEdit.setObjectName("Record_textEdit")
self.Play_textEdit = QtWidgets.QTextEdit(self.Record_Play_tab)
self.Play_textEdit.setGeometry(QtCore.QRect(30, 270, 271, 31))
self.Play_textEdit.setObjectName("Play_textEdit")
self.Record_label = QtWidgets.QLabel(self.Record_Play_tab)
self.Record_label.setGeometry(QtCore.QRect(30, 120, 51, 21))
self.Record_label.setObjectName("Record_label")
self.Play_label = QtWidgets.QLabel(self.Record_Play_tab)
self.Play_label.setGeometry(QtCore.QRect(30, 240, 51, 21))
self.Play_label.setObjectName("Play_label")
self.Record_Button = QtWidgets.QPushButton(self.Record_Play_tab)
self.Record_Button.setGeometry(QtCore.QRect(370, 150, 91, 31))
self.Record_Button.setObjectName("Record_Button")
self.Stop_Button = QtWidgets.QPushButton(self.Record_Play_tab)
self.Stop_Button.setGeometry(QtCore.QRect(370, 190, 91, 31))
self.Stop_Button.setObjectName("Stop_Button")
self.Play_Button = QtWidgets.QPushButton(self.Record_Play_tab)
self.Play_Button.setGeometry(QtCore.QRect(370, 270, 91, 31))
self.Play_Button.setObjectName("Play_Button")
self.IP_label = QtWidgets.QLabel(self.Record_Play_tab)
self.IP_label.setGeometry(QtCore.QRect(30, 10, 51, 21))
self.IP_label.setObjectName("IP_label")
self.IP_textEdit = QtWidgets.QTextEdit(self.Record_Play_tab)
self.IP_textEdit.setGeometry(QtCore.QRect(30, 40, 271, 31))
self.IP_textEdit.setObjectName("IP_textEdit")
self.Connect_Button = QtWidgets.QPushButton(self.Record_Play_tab)
self.Connect_Button.setGeometry(QtCore.QRect(370, 40, 91, 31))
self.Connect_Button.setObjectName("Connect_Button")
self.Disconnect_Button = QtWidgets.QPushButton(self.Record_Play_tab)
self.Disconnect_Button.setGeometry(QtCore.QRect(370, 80, 91, 31))
self.Disconnect_Button.setObjectName("Disconnect_Button")
self.Rec_Play_Status_Tab.addTab(self.Record_Play_tab, "")
self.Status_tab = QtWidgets.QWidget()
self.Status_tab.setObjectName("Status_tab")

```

```

self.Tx_status_label = QtWidgets.QLabel(self.Status_tab)
self.Tx_status_label.setGeometry(QRect(30, 30, 71, 31))
font = QtGui.QFont()
font.setPointSize(10)
self.Tx_status_label.setFont(font)
self.Tx_status_label.setObjectName("Tx_status_label")
self.Tx_keep_label = QtWidgets.QLabel(self.Status_tab)
self.Tx_keep_label.setGeometry(QRect(60, 70, 61, 21))
self.Tx_keep_label.setObjectName("Tx_keep_label")

self.Tx_keep_led = QLed(self.Status_tab, onColour=QLed.Green,
                        shape=QLed.Circle)
self.Tx_keep_led.value = False
self.Tx_keep_led.setGeometry(QRect(150, 70, 151, 21))

self.PTT_label = QtWidgets.QLabel(self.Status_tab)
self.PTT_label.setGeometry(QRect(60, 100, 61, 21))
self.PTT_label.setObjectName("PTT_label")

self.PTT_led = QLed(self.Status_tab, onColour=QLed.Green, shape=QLed.Circle)
self.PTT_led.value = False
self.PTT_led.setGeometry(QRect(150, 100, 151, 21))

self.Rx_keep_label = QtWidgets.QLabel(self.Status_tab)
self.Rx_keep_label.setGeometry(QRect(60, 190, 61, 21))
self.Rx_keep_label.setObjectName("Rx_keep_label")

self.Rx_keep_led = QLed(self.Status_tab, onColour=QLed.Green,
                        shape=QLed.Circle)
self.Rx_keep_led.value = False
self.Rx_keep_led.setGeometry(QRect(150, 190, 151, 21))

self.Squelch_label = QtWidgets.QLabel(self.Status_tab)
self.Squelch_label.setGeometry(QRect(60, 220, 61, 21))
self.Squelch_label.setObjectName("Squelch_label")

self.Squelch_led = QLed(self.Status_tab, onColour=QLed.Green,
                        shape=QLed.Circle)

```

```

self.Squelch_led.value = False
self.Squelch_led.setGeometry(QRect(150, 220, 151, 21))

self.Rx_status_label = QtWidgets.QLabel(self.Status_tab)
self.Rx_status_label.setGeometry(QRect(30, 150, 71, 31))
font = QtGui.QFont()
font.setPointSize(10)
self.Rx_status_label.setFont(font)
self.Rx_status_label.setObjectName("Rx_status_label")
self.Rec_Play_Status_Tab.addTab(self.Status_tab, "")
self.gridLayout.addWidget(self.Rec_Play_Status_Tab, 0, 0, 1, 1)
MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(MainWindow)
self.menubar.setGeometry(QRect(0, 0, 509, 26))
self.menubar.setObjectName("menubar")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(MainWindow)
self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)
self.Rec_Play_Status_Tab.setCurrentIndex(0)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
    self.Record_label.setText(_translate("MainWindow", "file.wav"))
    self.Play_label.setText(_translate("MainWindow", "file.wav"))
    self.Record_Button.setText(_translate("MainWindow", "Record"))
    self.Stop_Button.setText(_translate("MainWindow", "Stop"))
    self.Play_Button.setText(_translate("MainWindow", "Play"))
    self.IP_label.setText(_translate("MainWindow", "IP"))
    self.Connect_Button.setText(_translate("MainWindow", "Connect"))
    self.Disconnect_Button.setText(_translate("MainWindow", "Disconnect"))
    self.Rec_Play_Status_Tab.setTabText(self.Rec_Play_Status_Tab.indexOf(self.Record_
        Play_tab), _translate("MainWindow", "Record-Play"))

```

```
self.Tx_status_label.setText(_translate("MainWindow", "Tx status"))
self.Tx_keep_label.setText(_translate("MainWindow", "Keep alive"))
self.PTT_label.setText(_translate("MainWindow", "PTT"))
```

```
self.Rx_keep_label.setText(_translate("MainWindow", "Keep alive"))
self.Squelch_label.setText(_translate("MainWindow", "squelch"))
self.Rx_status_label.setText(_translate("MainWindow", "Rx status"))
self.Rec_Play_Status_Tab.setTabText(self.Rec_Play_Status_Tab.indexOf(self.Status_ta
b), _translate("MainWindow", "Status"))
```

```
self.Connect_Button.clicked.connect(self.startFirstThread)
self.Disconnect_Button.clicked.connect(self.disconnect)
self.Record_Button.clicked.connect(self.record)
self.Stop_Button.clicked.connect(self.stop)
self.Play_Button.clicked.connect(self.play)
```

```
def startFirstThread(self):
```

```
    global con_discon, record_text, IP_text
    IP_text = self.IP_textEdit.toPlainText()
    record_text = self.Record_textEdit.toPlainText()
    con_discon = False
    self.thread = MyThread()
    self.thread.change_Tx_keep_led.connect(self.set_Tx_keep_Led)
    self.thread.change_Rx_keep_led.connect(self.set_Rx_keep_Led)
    self.thread.change_PTT_led.connect(self.set_PTT_Led)
    self.thread.change_Squelch_led.connect(self.set_Squelch_Led)
    self.thread.start()
```

```
def set_Tx_keep_Led(self, bool):
```

```
    self.Tx_keep_led.setValue(bool)
```

```
def set_Rx_keep_Led(self, bool):
```

```
    self.Rx_keep_led.setValue(bool)
```

```
def set_PTT_Led(self, bool):
```

```
    self.PTT_led.setValue(bool)
```

```
def set_Squelch_Led(self, bool):  
    self.Squelch_led.setValue(bool)
```

```
def disconnect(self):  
    print("Disconnect")  
    global con_discon  
    con_discon = True
```

```
def record(self):  
    print("Record")  
    global rec_stop  
    rec_stop = True
```

```
def stop(self):  
    print("Stop")  
    global rec_stop  
    rec_stop = False
```

```
def play(self):  
    global play_text  
    play_text = self.Play_textEdit.toPlainText()  
    print("Play")  
    x, fs = sf.read(play_text)  
    [b, a] = C_weighting(fs)  
    y = lfilter(b, a, x)  
    sf.write(play_text, y, fs)  
    playsound(play_text)  
    print("Finish")
```

```
def C_weighting(fs):  
    f1 = 300 #20.598997  
    f4 = 4000 #12194.217  
    C1000 = 0.0619  
    NUMs = [(2*pi*f4)**2*(10**(C1000/20.0)),0,0]  
    DENs = polymul([1,4*pi*f4,(2*pi*f4)**2.0],[1,4*pi*f1,(2*pi*f1)**2])  
    return bilinear(NUMs, DENs, fs)
```

```

#Unpacking IPv4 Packet
def ipv4_packet(data):
    version_header_length = data[0]
    version = version_header_length >> 4
    header_length = (version_header_length & 15) * 4
    ttl, proto, src, target = struct.unpack('! 8x B B 2x 4s 4s', data[:20])
    return version, header_length, ttl, proto, ipv4(src), ipv4(target), data[header_length:]

```

*#Return properly formatted IPv4 address*

```

def ipv4(addr):
    return '.'.join(map(str, addr))

```

*#Unpacks UDP segment*

```

def udp_segment(data):
    src_port, dest_port, size = struct.unpack('! H H H', data[:6])
    return src_port, dest_port, size, data[8:], data[28:]

```

*def sqelch\_status(data):*

```

sq_bit = data[16]
sq_status = (sq_bit >> 4) & 1
return sq_status

```

*def push\_to\_talk\_status(data):*

```

ptt_bit = data[16]
ptt_status = (ptt_bit >> 5) & 111
return ptt_status

```

*def keep\_alive\_status(data):*

```

pt_bit = data[1]
pt_status = pt_bit
return pt_status

```

*#Formats multi-line data*

```

def format_multi_line(prefix, string, size=80):
    size -= len(prefix)
    if isinstance(string, bytes):
        string = ''.join(r'\x{:02x}'.format(byte) for byte in string)

```

```
if size % 2:  
    size -= 1  
return '\n'.join([prefix + line for line in textwrap.wrap(string, size)])
```

```
if __name__ == "__main__":  
    import sys  
    app = QtWidgets.QApplication(sys.argv)  
    MainWindow = QtWidgets.QMainWindow()  
    ui = Ui_MainWindow()  
    ui.setupUi(MainWindow)  
    MainWindow.show()  
    sys.exit(app.exec_())
```



## ประวัติผู้จัดทำ

ชื่อ-นามสกุล ณีภุชฌา วุฒิประสิทธิ์ผล  
วัน เดือน ปีเกิด 19 ตุลาคม 2540  
ที่อยู่ปัจจุบัน 422 ซอยจรัญสนิทวงศ์ 62 เขตบางพลัด แขวงบางพลัด  
จังหวัดกรุงเทพมหานคร 10700

### ประวัติการศึกษา

พ.ศ. 2558 มัธยมศึกษาตอนปลาย  
โรงเรียนสาธิตมหาวิทยาลัยราชภัฏสวนสุนันทา  
พ.ศ. 2562 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้