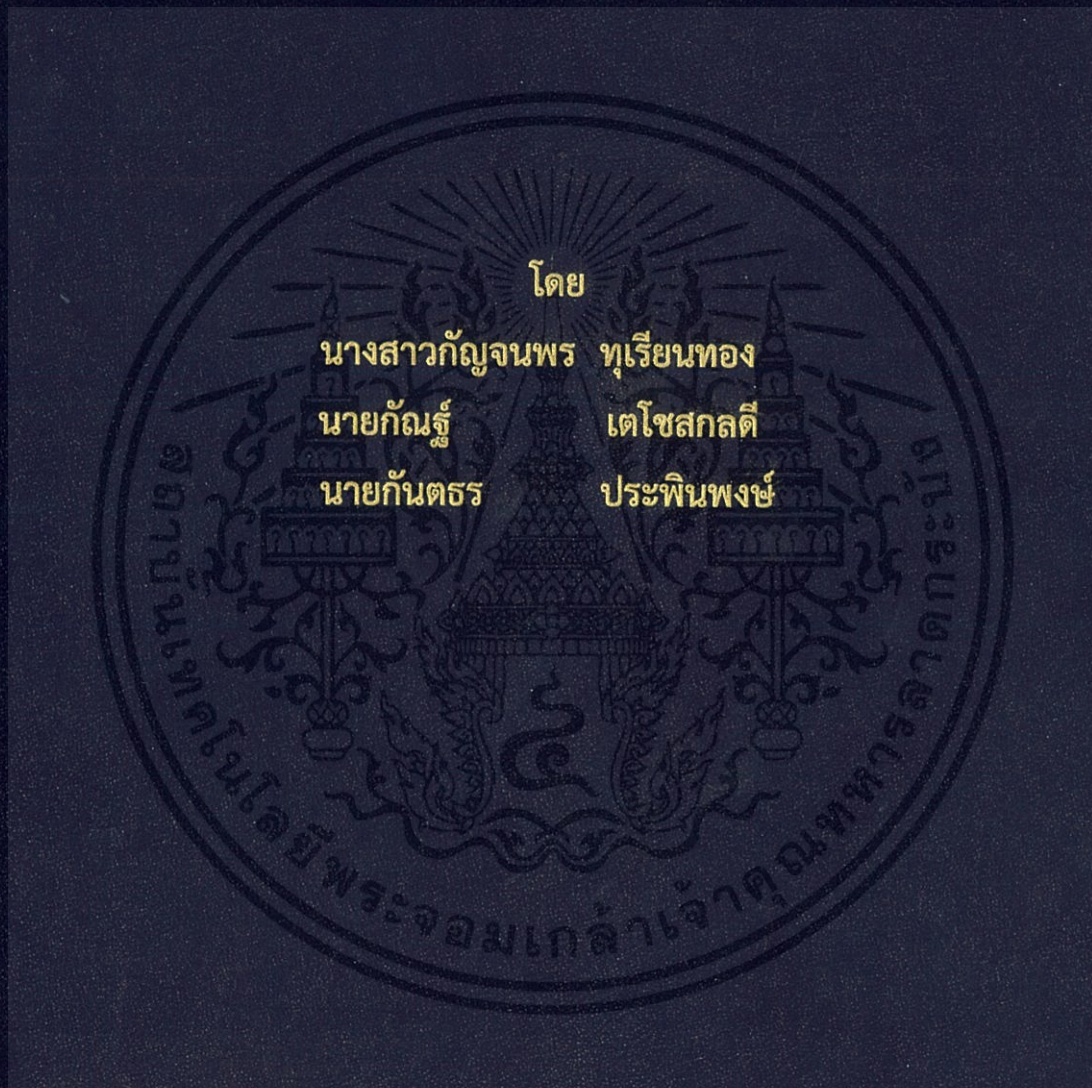


การพัฒนาต้นแบบเครือข่ายสื่อสารสำหรับการระบุตำแหน่งในอาคาร  
IMPLEMENTATION OF COMMUNICATION SYSTEM FOR INDOOR  
POSITIONING



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมโทรคมนาคม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2561

การพัฒนาต้นแบบเครือข่ายสื่อสารสำหรับการระบุตำแหน่งในอาคาร  
IMPLEMENTATION OF COMMUNICATION SYSTEM FOR INDOOR  
POSITIONING



โดย  
นางสาวกัญจนพร ทูเรียนทอง 58010062  
นายกัณฐ์ เตโชสกลดี 58010067  
นายกันตธร ประพินพงษ์ 58010070

อาจารย์ที่ปรึกษา  
รศ.ดร. ยุทธพงษ์ รังสรรค์เสรี

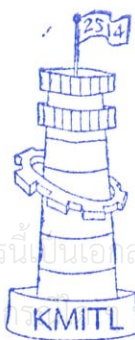
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

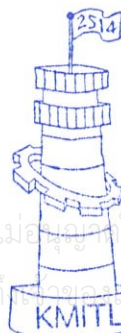
ปีการศึกษา 2561



ผ่านการตรวจรูปเล่มแล้ว

(.....)  
อาจารย์ที่ปรึกษา

23/5/62



ผ่านการตรวจชิ้นงานแล้ว

(.....)  
กรรมการผู้ตรวจชิ้นงาน

23/5/62

ปริญญาโทปีการศึกษา 2561

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาต้นแบบเครือข่ายสื่อสารสำหรับการระบุตำแหน่งในอาคาร

IMPLEMENTATION OF COMMUNICATION SYSTEM FOR INDOOR POSITIONING

ผู้จัดทำ

- |                  |            |          |
|------------------|------------|----------|
| 1. นางสาวกัญจนพร | ทุเรียนทอง | 58010062 |
| 2. นายกัณฐ์      | เดโชสกลดี  | 58010067 |
| 3. นายกันตธร     | ประพินพงษ์ | 58010070 |

(รศ.ดร. ยุทธพงษ์ รังสรรค์เสรี)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

การดำเนินปฏิญาณพันธนี้จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี ถ้าหากขาดการสนับสนุนและกำลังใจจากหลาย ๆ ฝ่าย เช่น

รศ.ดร.ยุทพงษ์ รังสรรค์เสรี และอาจารย์สมรักษ์ เพชรชาติรี อาจารย์ที่ปรึกษาปฏิญาณพันธ สำหรับคำปรึกษา คำแนะนำและแนวทางการแก้ไขเมื่อเกิดปัญหาต่าง ๆ ในการจัดทำปฏิญาณพันธในครั้งนี้ รวมไปถึงการสนับสนุนสถานที่ เครื่องมือและอุปกรณ์ต่าง ๆ ที่จำเป็นต้องใช้ในระหว่างการจัดทำปฏิญาณพันธ

คณาจารย์ทุกท่านของภาควิชาโทรคมนาคมที่ให้คำปรึกษาและให้การสนับสนุนอุปกรณ์ต่าง ๆ รวมถึงคำปรึกษาเพื่อใช้ในการแก้ปัญหาที่เกิดขึ้นระหว่างการทำปฏิญาณพันธ

เพื่อน ๆ นักศึกษาชั้นปีที่ 4 ทุกคนในภาควิชาที่ได้คอยให้คำแนะนำ วิธีแก้ปัญหาและสนับสนุนอุปกรณ์ต่าง ๆ คำปรึกษาและให้ความรู้เพิ่มเติมเกี่ยวกับการจัดทำปฏิญาณพันธ

ผู้จัดทำขอขอบพระคุณทุกท่านเป็นอย่างสูง ณ ที่นี้ ที่ได้ช่วยให้การจัดทำปฏิญาณพันธในครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี

นางสาวกัญจนพร ทูเรียนทอง

นายกันฐ์ เตโชสกลดี

นายกันตธร ประพินพงษ์

ผู้จัดทำ

การพัฒนาต้นแบบเครือข่ายสื่อสารสำหรับการระบุตำแหน่งในอาคาร  
IMPLEMENTATION OF COMMUNICATION SYSTEM FOR  
INDOOR POSITIONING

โดย นางสาวกัญจนพร ทูเรียนทอง 58010062  
นายกันฐ์ เตโชสกลดี 58010067  
นายกันตธร ประพินพงษ์ 58010070

อาจารย์ที่ปรึกษา รศ.ดร. ยุทธพงษ์ ริงสรค์เสรี

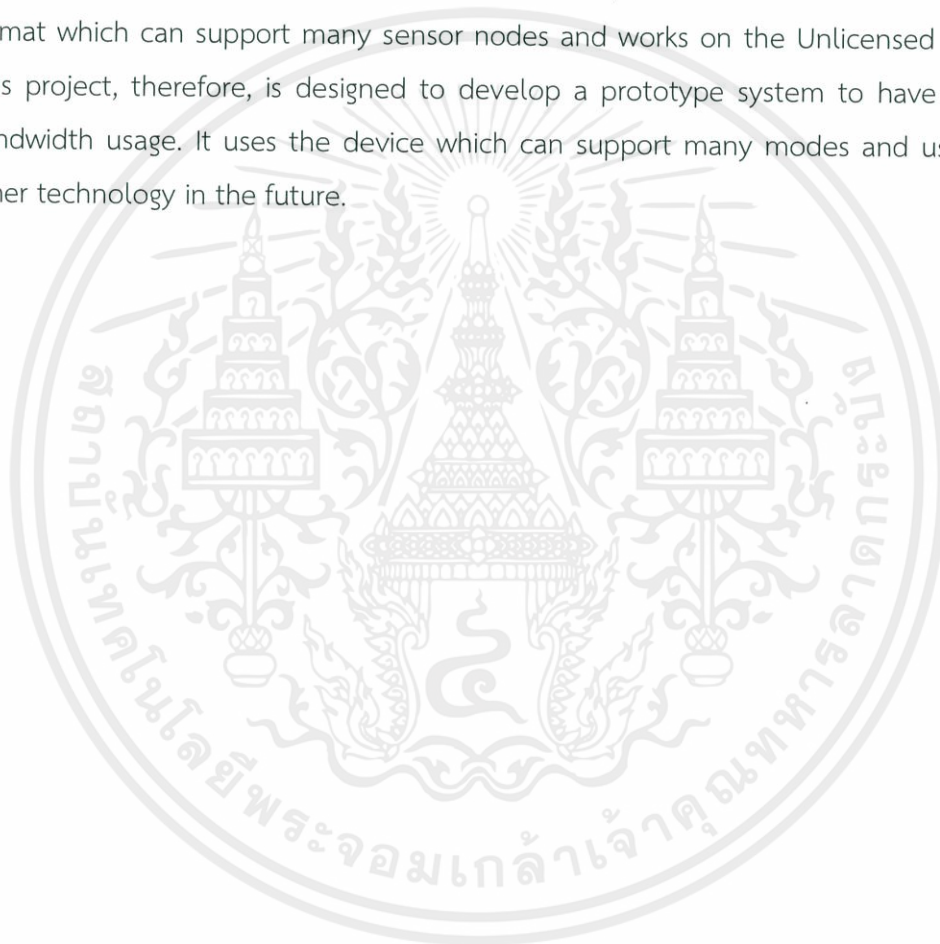
**บทคัดย่อ**

โครงการนี้นำเสนอการออกแบบและพัฒนาต้นแบบเครือข่ายการสื่อสารไร้สายเพื่อรองรับการใช้งานสำหรับเทคโนโลยีต่าง ๆ อย่างเหมาะสมตามความต้องการของผู้ใช้งาน เช่น การระบุตำแหน่งภายในอาคาร (Indoor positioning) ระบบฟาร์มอัจฉริยะ (Smart farming) เป็นต้น โดยระบบมีการทำงานบนช่องสัญญาณความถี่ร่วมสาธารณะ (Unlicensed band) โครงการนี้จึงจัดทำขึ้นเพื่อพัฒนาต้นแบบเครือข่ายสื่อสารซึ่งประกอบด้วยอุปกรณ์รับส่งสัญญาณไร้สาย ทำหน้าที่ Sensor node และ Gateway โดยในการรับส่งข้อมูลจะใช้รูปแบบการมอดูเลชันแบบ FSK รวมไปถึงทำการพัฒนาให้สามารถใช้งานในรูปแบบ Multi access กับอุปกรณ์ลูกข่ายหรือที่เรียกว่า Sensor node พร้อม ๆ กันจำนวนมากได้ และทำงานบนช่องสัญญาณความถี่ร่วมสาธารณะ ให้มีการใช้แบนด์วิดธ์ได้อย่างมีประสิทธิภาพมากขึ้น โครงการนี้ได้เลือกใช้อุปกรณ์ที่สามารถรองรับการมอดูเลชันได้หลายรูปแบบ และยังสามารถนำไปประยุกต์ใช้ในเทคโนโลยีอื่น ๆ ในอนาคตได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ABSTRACT

This project presents the design and development of wireless communication network prototypes to support technology such as indoor positioning, smart farming, etc. It consists of wireless transceivers which acting as Sensor node and Gateway. For the data transmission, the Frequency Shift Keying modulation model will be used as well as developed for the use in Multi access format which can support many sensor nodes and works on the Unlicensed band. This project, therefore, is designed to develop a prototype system to have more bandwidth usage. It uses the device which can support many modes and used in other technology in the future.



## สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	IX
<b>บทที่ 1</b>	
<b>บทนำ</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
<b>บทที่ 2</b>	
<b>ทฤษฎีและหลักการที่เกี่ยวข้อง</b>	<b>3</b>
2.1 การระบุตำแหน่งวัตถุภายในอาคาร	3
2.2 ไมโครคอนโทรลเลอร์ STM32L476	7
2.3 TRANSCEIVER	10
2.4 การใช้งานคลื่นความถี่ UNLICENSED BAND 920-925 MHZ	12
2.5 การมอดูเลชันแบบ FSK (FREQUENCY SHIFT KEYING)	14
<b>บทที่ 3</b>	
<b>การออกแบบและการจัดทำปริญญานิพนธ์</b>	<b>16</b>
3.1 การออกแบบ	16
3.2 เครื่องมือที่ใช้ในการทดลอง	31
3.3 การจัดเก็บผลการทดลอง	32

## สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง	33
4.1 ผลการทดสอบการทำงานของ SENSOR NODE	33
4.2 ผลการทดสอบการทำงานของ GATEWAY	37
4.3 ผลการทดสอบการสื่อสารกันระหว่าง SENSOR NODE และ GATEWAY	38
บทที่ 5 สรุปผลและข้อเสนอแนะ	51
5.1 สรุปผล	51
5.2 ข้อเสนอแนะ	52
บรรณานุกรม	53
ภาคผนวก ก ภาพขึ้นงาน	54
ภาคผนวก ข ภาษาที่ใช้ในโปรแกรมคำสั่ง	57
ภาคผนวก ค เอกสารอ้างอิง	69

## สารบัญรูป

รูปที่	หน้า
2.1	4
2.2	4
2.3	5
2.4	6
2.5	7
2.6	10
2.7	11
2.8	13
2.9	13
2.10	14
2.11	14
3.1	16
3.2	17
3.3	17
3.4	18
3.5	21
3.6	22
3.7	22
3.8	23
3.9	24
3.10	25
3.11	26

## สารบัญรูป (ต่อ)

รูปที่	หน้า	
3.12	โปรโตคอลการตัดสินใจเลือกใช้ช่องสัญญาณ	28
3.13	การทำงานในกรณีที่ CHANNEL REQUEST PACKET สูญหาย	29
3.14	การทำงานในกรณีที่ CHANNEL REPLY PACKET สูญหาย	29
3.15	PACKET FIELD	30
3.16	REQUEST PACKET	30
3.17	REPLY PACKET	31
3.18	DATA PACKET	31
4.1	SIMULINK การทำงานของระบบ	33
4.2	การเปรียบเทียบค่าระหว่าง BER และ SNR	34
4.3	การทำงานของฝั่งส่งและฝั่งรับที่ขนาดข้อมูล 255 ไบต์	35
4.4	การทำงานของฝั่งส่งและฝั่งรับที่ขนาดข้อมูล 2047 ไบต์	35
4.5	ทดสอบการทำงานของฝั่งรับและฝั่งส่ง	36
4.6	ผลลัพธ์จาก SDRSHARP ของ SENSOR NODE	37
4.7	ผลลัพธ์จาก SDRSHARP ของ GATEWAY	37
4.8	การส่งข้อมูลจาก SENSOR NODE ไป GATEWAY	38
4.9	สเปกตรัมของ SENSOR NODE ที่อยู่คนละช่องสัญญาณ	39
4.10	สเปกตรัมข้อมูลที่ใช้ในการร้องขอและการตอบกลับของช่องสัญญาณ	40
4.11	สเปกตรัมการส่งข้อมูลของ SENSOR NODE ไปยังช่องสัญญาณที่ 1	40
4.12	สเปกตรัมการตอบกลับของ GATEWAY ไปให้ SENSOR NODE ใช้ช่องสัญญาณที่ 2	41
4.13	สเปกตรัมการส่งข้อมูลของ SENSOR NODE ไปยังช่องสัญญาณที่ 2	42

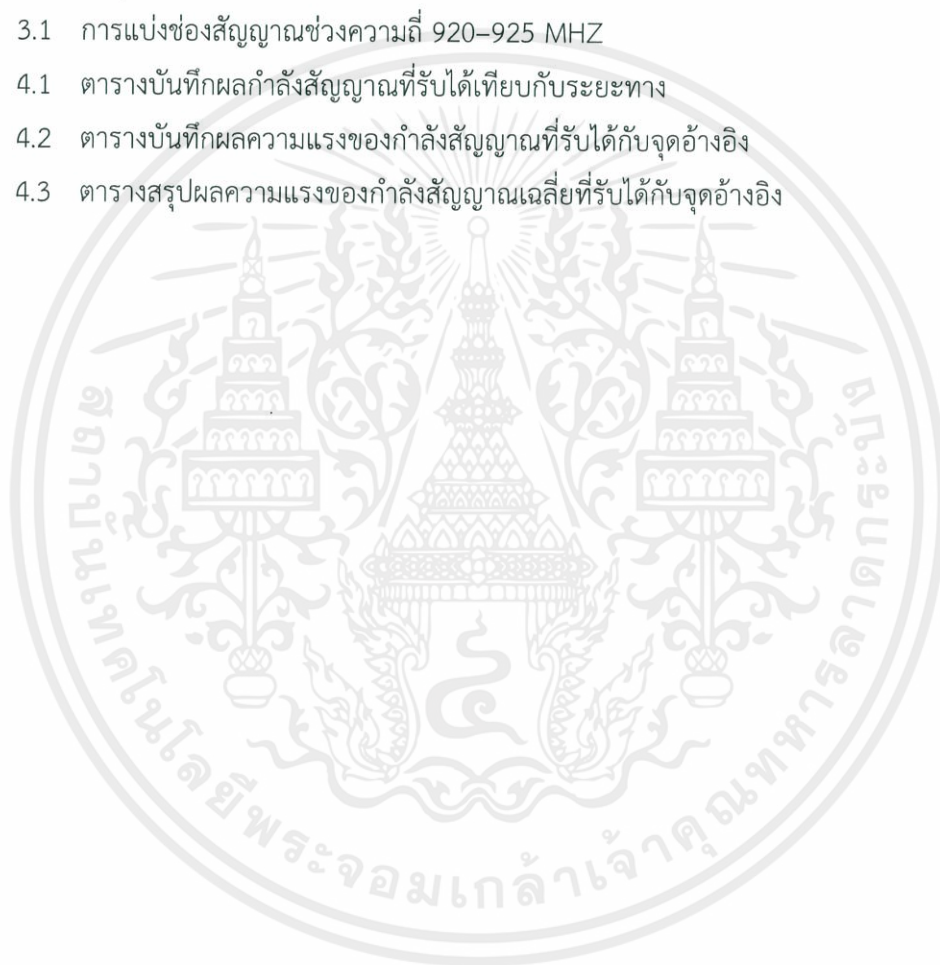
## สารบัญรูป (ต่อ)

รูปที่		หน้า
4.14	สเปกตรัมการตอบกลับของ GATEWAY ไปให้ SENSOR NODE ใช้ช่องสัญญาณที่ 3	43
4.15	สเปกตรัมการส่งข้อมูลของ SENSOR NODE ไปยังช่องสัญญาณที่ 3	43
4.16	กำลังสัญญาณที่รับได้จาก SENSOR NODE เทียบกับระยะทาง	46
4.17	แผนที่แสดงตำแหน่งติดตั้ง SENSOR NODE และจุดอ้างอิง	47



## สารบัญตาราง

ตารางที่		หน้า
2.1	ข้อมูลจำเพาะของ S76S MODULE	12
3.1	การแบ่งช่องสัญญาณช่วงความถี่ 920–925 MHz	27
4.1	ตารางบันทึกผลกำลังสัญญาณที่รับได้เทียบกับระยะทาง	45
4.2	ตารางบันทึกผลความแรงของกำลังสัญญาณที่รับได้กับจุดอ้างอิง	48
4.3	ตารางสรุปผลความแรงของกำลังสัญญาณเฉลี่ยที่รับได้กับจุดอ้างอิง	50



## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

ฝ่ายวิจัยของบริษัท กสท โทรคมนาคม จำกัด (มหาชน) ได้ริเริ่มโครงการพัฒนาเครือข่ายสื่อสารไร้สายเพื่อรองรับการใช้งาน IoT ที่ทำงานบนช่องสัญญาณความถี่ร่วมสาธารณะ (Unlicensed band) สำหรับรองรับการใช้งานขององค์กรหรือธุรกิจขนาดใหญ่ เพื่อให้ผู้ใช้งานระบบสื่อสารนี้ไปใช้ได้โดยไม่ต้องเสียค่าบริการ ซึ่งจะสามารถนำไปประยุกต์ใช้งานได้หลากหลาย เช่น

การระบุตำแหน่งวัตถุภายในอาคาร (Indoor positioning) สำหรับใช้ในอุตสาหกรรมขนาดใหญ่ เช่น ใช้ติดตามตำแหน่งของรถยกสินค้า (Forklift) ที่มีจำนวนมากในโรงงาน เพื่อหลีกเลี่ยงการชนกันระหว่างรถยกกับรถยก หรือระหว่างรถยกกับพนักงาน และเพื่อการบริหารจัดการที่มีประสิทธิภาพ เป็นต้น

ระบบฟาร์มอัจฉริยะ (Smart farming) ซึ่งต้องการเครือข่ายเซ็นเซอร์ไร้สาย (Wireless sensor network) ติดตั้งตามจุดต่าง ๆ ในพื้นที่เกษตรขนาดใหญ่ เพื่อวัดข้อมูลที่หลากหลาย เช่น อุณหภูมิ ความชื้น ในดินและในอากาศ แสง ลม เป็นต้น

การจัดการโรงจอดรถอัจฉริยะ (Smart parking) เป็นระบบ IoT สำหรับใช้เซ็นเซอร์ตรวจสอบว่าที่จอดรถในช่องใดยังว่างอยู่บ้าง สามารถนำทางรถไปยังช่องว่างดังกล่าว และเจ้าของที่จอดรถก็สามารถเก็บเงินค่าจอดรถได้อย่างแม่นยำยิ่งขึ้น

แนวทางการพัฒนาระบบสื่อสารไร้สายดังกล่าวนี้ จะใช้บอร์ดสำเร็จรูปของบริษัท SEMTECH ที่มีความสามารถรับส่งข้อมูลได้ระยะไกล รองรับการผลิตได้หลายแบบ และสามารถเลือกปรับขนาดของเฟรมข้อมูลได้ตามความเหมาะสมของการใช้งานแต่ละประเภท นำมาประกอบกันเป็นระบบเครือข่ายไร้สายที่ประกอบด้วยอุปกรณ์หลัก 2 ส่วน ได้แก่ Gateway ทำหน้าที่เป็นสถานีหลักรองรับการสื่อสารข้อมูลในลักษณะ Multi access กับอุปกรณ์ลูกข่ายหรือที่เรียกว่า Sensor node พร้อม ๆ กันจำนวนมาก โดยจะต้องทำงานได้ในช่องสัญญาณความถี่ร่วมสาธารณะ (Unlicensed band)

ปฏิญานี้ เป็นงานส่วนหนึ่งของการพัฒนาระบบสื่อสารไร้สายนี้ โดยรับผิดชอบในส่วนการออกแบบและพัฒนาเทคนิคที่จำเป็นต้องใช้ในการจัดการสื่อสารระหว่าง Gateway กับ Sensor node ได้แก่ วิธีการมอดูเลชัน (Modulation) การกำหนดขนาดของ Packet ข้อมูล (Packet size) การจัดสรรช่องสัญญาณ (Channel allocation) โพรโตคอลสำหรับการตัดสินใจเลือกใช้ช่องสัญญาณ (Channel assignment protocol) เป็นต้น นอกจากนี้ ยังได้เลือกยกตัวอย่างการประยุกต์ใช้ในระบบการระบุตำแหน่งวัตถุภายในอาคาร มาอธิบายประกอบด้วย

## 1.2 วัตถุประสงค์

เพื่อทำการออกแบบและพัฒนาต้นแบบเครือข่ายสื่อสารไร้สายให้รองรับการใช้งาน IoT ที่ทำงานบนช่องสัญญาณความถี่ร่วมสาธารณะ (Unlicensed band) และเพื่อให้สามารถนำไปประยุกต์ใช้ได้ในอนาคต เช่น การระบุตำแหน่งวัตถุภายในอาคาร (Indoor positioning) เป็นต้น ซึ่งการพัฒนาระบบจะอยู่บนพื้นฐานของการมอดูเลชัน ร่วมกับวิธีการเข้าถึงแบบ Multi access ของอุปกรณ์ในเครือข่ายเพื่อให้สามารถสื่อสารพร้อมกันได้เมื่อมีจำนวนมากขึ้น

## 1.3 ขอบเขตของปฏิญานี้

- 1) ทำการออกแบบรูปแบบของ Packet ข้อมูลที่ใช้ในการสื่อสาร โดยเมื่อมีการส่งข้อมูลระหว่างอุปกรณ์ในเครือข่าย จะสามารถทราบได้ว่า Packet นั้นถูกส่งมาเพื่อจุดประสงค์ใด และกำหนดหน้าที่ให้แก่อุปกรณ์ในระบบเครือข่าย
- 2) ทำการออกแบบรูปแบบของ Packet ข้อมูลให้สามารถปรับขนาดได้ตามที่ผู้ใช้งานต้องการ เพื่อให้เหมาะสมต่อการนำไปประยุกต์ใช้ในงานแต่ละประเภทได้
- 3) ทำการออกแบบการสื่อสารด้วยวิธีการเข้าถึงแบบ Multi access เพื่อให้อุปกรณ์ในเครือข่ายสามารถทำการสื่อสารกันได้ในขณะที่มีจำนวนของตัวอุปกรณ์เพิ่มมากขึ้น

## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

#### 2.1 การระบุตำแหน่งวัตถุภายในอาคาร

การระบุตำแหน่งวัตถุภายในอาคาร หรือ Indoor Positioning เป็นการระบุตำแหน่งของวัตถุซึ่งพุดง่าย ๆ ก็เหมือนเป็น GPS สำหรับใช้งานภายในอาคาร โดยการระบุตำแหน่งวัตถุนี้จะใช้ในการค้นหาวัตถุภายในอาคารโดยใช้งานผ่านอุปกรณ์ ตัวอย่างการใช้งานเช่น ในโรงพยาบาล สนามบินหรือสถานที่ภายในอาคารอื่น ๆ

เทคโนโลยีการระบุตำแหน่งวัตถุภายในอาคารจะมีการติดตั้ง Sensor node ไว้ภายในอุปกรณ์พกพาหรือสมาร์ทโฟน เพื่อใช้สำหรับการหาตำแหน่งของอุปกรณ์ภายในอาคาร โดยใช้อัลกอริทึมทางคณิตศาสตร์ที่มีความซับซ้อนโดยการนำเอาข้อมูลที่ได้รับจาก Sensor node เหล่านี้รวมเข้าด้วยกัน และจะทำให้สามารถคำนวณตำแหน่งที่แม่นยำได้โดยจะอาศัยความหนาแน่นหรือจะไม่มีควมหนาแน่นก็ได้ โดยระบบการระบุตำแหน่งวัตถุภายในอาคารในปัจจุบันมีการนำมาใช้กันอย่างแพร่หลายยกตัวอย่างเทคโนโลยีการระบุตำแหน่งวัตถุภายในอาคารที่มีการใช้ในปัจจุบันได้ดังนี้

##### 2.1.1 การระบุตำแหน่งภายในอาคารโดยใช้ Wi-Fi

การระบุตำแหน่งภายในอาคารโดยใช้ Wi-Fi หรือ Wi-Fi Positioning นั้นสามารถนำมาใช้ได้อย่างหลากหลาย ด้วยเหตุผลที่ว่า Wi-Fi hotspot มีการใช้งานกันอย่างแพร่หลายจึงเหมาะกับการนำมาใช้ในลักษณะนี้ นอกจากนี้การระบุตำแหน่งจะสามารถทำงานได้เมื่อผู้ใช้เปิดใช้งาน Wi-Fi บนอุปกรณ์พกพา แต่อย่างไรก็ตามการระบุตำแหน่งภายในอาคารโดยใช้ Wi-Fi ก็มีข้อเสียเช่นกัน เพราะการระบุตำแหน่งภายในอาคารโดยใช้ Wi-Fi มีความแม่นยำอยู่ที่ 5-15 เมตร ซึ่งน้อยกว่าการใช้บลูทูธพลังงานต่ำ

ในแต่ละอุปกรณ์ที่มีความสามารถในการเชื่อมต่ออินเทอร์เน็ต เช่น Wi-Fi access point หรือ Hotspot Router จะมีการส่งข้อมูลชนิดพิเศษ ซึ่งจะมีการใช้วิธีการระบุตำแหน่งคือ RSSI (Received Signal Strength Indication) และ MAC address (Media Access Control) ไว้ใช้ในการคำนวณหาตำแหน่งอุปกรณ์ของผู้ใช้ปลายทาง (Client-Based Positioning) และจำเป็นต้องมีฐานข้อมูล (Database) รวมถึงข้อมูลที่เกี่ยวข้องกับสถานที่ที่จะสามารถนำมาเปรียบเทียบข้อมูลนี้ได้ด้วย ซึ่งวิธีการนี้เรียกว่า Fingerprinting



รูปที่ 2.1 การระบุตำแหน่งภายในอาคารโดยใช้ Wi-Fi [8]

ข้อจำกัดการใช้งานในลักษณะนี้ขึ้นอยู่กับว่าอุปกรณ์ถูกเชื่อมต่อเข้ากับเครือข่าย Wi-Fi หรือไม่ รวมถึงข้อจำกัดอื่น ๆ เช่น หากอุปกรณ์ของผู้ใช้ไม่ได้เชื่อมต่อ Wi-Fi เวลาในการแสวงหา Wi-Fi access point อาจจะล่าช้าถึง 5 นาที

	not connected to Wi-Fi network	connected to Wi-Fi network
Latency times	<p>high (probe requests sent at irregular intervals)</p>	<p>lower (when generating Wi-Fi traffic)</p>
Hardware address of the Wi-Fi client (smart device)	<p>MAC randomization randomly generated MAC addresses (ambiguous correlation)</p>	<p>no MAC randomization unambiguous correlation</p>
Accuracy	<p>&lt; 15 m</p>	<p>&lt; 15 m</p>

รูปที่ 2.2 ข้อจำกัดของการระบุตำแหน่งภายในอาคารโดยใช้ Wi-Fi [8]

จากรูปที่ 2.2 จึงแสดงความแตกต่างระหว่างการเชื่อมต่อ Wi-Fi กับการไม่เชื่อมต่อ Wi-Fi ที่มีความหน่วงเวลา (Latency times), ความแม่นยำ (Accuracy) และ Hardware address ของผู้ใช้ Wi-Fi

ในด้านของการใช้ Wi-Fi มีข้อดีและข้อเสียหลัก ๆ ดังนี้

ข้อดี : โดยทั่วไปตัวอุปกรณ์นั้นมีการติดตั้งอยู่ในแต่ละสถานที่อยู่แล้ว จึงไม่มีความจำเป็นที่จะต้องวางโครงข่าย Network ใหม่

ข้อเสีย : มีความแม่นยำต่ำ เพราะการใช้เทคนิค Fingerprint อาจมีความคลาดเคลื่อนได้ในระยะ 5-15 เมตร ซึ่งหากต้องการความแม่นยำที่เพิ่มขึ้น ก็จะมีเรื่องของค่าใช้จ่ายที่เพิ่มขึ้นในการเพิ่มจำนวนของตัว Access point

### 2.1.2 การระบุตำแหน่งภายในอาคารโดยใช้บลูทูธพลังงานต่ำ

การระบุตำแหน่งภายในอาคารโดยใช้บลูทูธพลังงานต่ำ หรือ Bluetooth Low Energy Positioning ซึ่งมีประเภทการใช้งานที่พบบ่อยสุดคือ iBeacon และ Eddystone โดยมีการใช้วิธี RSSI (Received Signal Strength Indication) ในการระบุหาตำแหน่งของวัตถุ ซึ่งข้อได้เปรียบของการใช้บลูทูธพลังงานต่ำคือมีความแม่นยำสูง ซึ่งมีความแม่นยำอยู่ที่ 1 เมตร อีกทั้ง บลูทูธพลังงานต่ำยังสามารถประหยัดการใช้งานแบตเตอรี่ได้ถึง 2-8 ปี โดยไม่จำเป็นต้องใช้แหล่งจ่ายไฟจากภายนอก



รูปที่ 2.3 ตำแหน่งการวางของ Beacon [9]

รูปที่ 2.3 เป็นตัวอย่างการวางตำแหน่ง Beacon ภายในอาคาร เพื่อให้มีพื้นที่การใช้งานที่ครอบคลุมทั้งบริเวณ



รูปที่ 2.4 รัศมีพื้นที่ของสัญญาณที่ครอบคลุมของ Beacon [9]

รูปที่ 2.4 แสดงถึงรัศมีของสัญญาณที่ถูกปล่อยออกมาจาก Beacon เพื่อแสดงให้เห็นว่าการวางตำแหน่ง Beacon ในแต่ละจุด สามารถครอบคลุมการใช้งานได้ทั่วทั้งบริเวณ

ในด้านของการใช้ลูทูลพลังงานต่ำมีข้อดีและข้อเสียหลัก ๆ ดังนี้

ข้อดี : มีราคาถูก และตัวอุปกรณ์มีการใช้พลังงานต่ำ มีความแม่นยำกว่าเทคนิค Wi-Fi โดยมีความคลาดเคลื่อนอยู่ที่ประมาณ 1 เมตร

ข้อเสีย : เทคนิคนี้จำเป็นต้องมีอุปกรณ์เสริมเพิ่ม เช่น Beacon ไว้ทำหน้าที่เสมือนเป็นระบบเครือข่าย ซึ่งระหว่างตัว Beacon เองนั้นมีระยะในการทำงานเพียง 20-30 เมตร

### 2.1.3 การระบุตำแหน่งภายในอาคารโดยใช้ Ultra-wideband (UWB)

การระบุตำแหน่งภายในอาคารโดยใช้ Ultra-wideband (UWB) หรือ Ultra-wideband Positioning เป็นเทคโนโลยีแบบ Short-range radio ใช้ในงานประเภทที่ต้องการความแม่นยำเป็นอย่างมาก อาศัยเทคนิคการวัดค่า ToF (Time of Flight) แทนวิธีการ RSSI (Received Signal Strength Indication) ที่ Wi-Fi และ Bluetooth Low Energy ใช้กัน ซึ่งวิธีการ ToF คือ การวัดเวลาการทำงานของแสงระหว่างวัตถุกับตัวรับสัญญาณหลาย ๆ ตัว และเพื่อความแม่นยำและความถูกต้องจำเป็นจะต้องมีตัวรับอย่างน้อย 3 ตัว นอกจากนี้จำเป็นจะต้องมีระยะ Line-of-Sight ระหว่างตัวรับและตัวส่งด้วย

ในด้านของการใช้ Ultra-Wideband มีข้อดีและข้อเสียหลัก ๆ ดังนี้

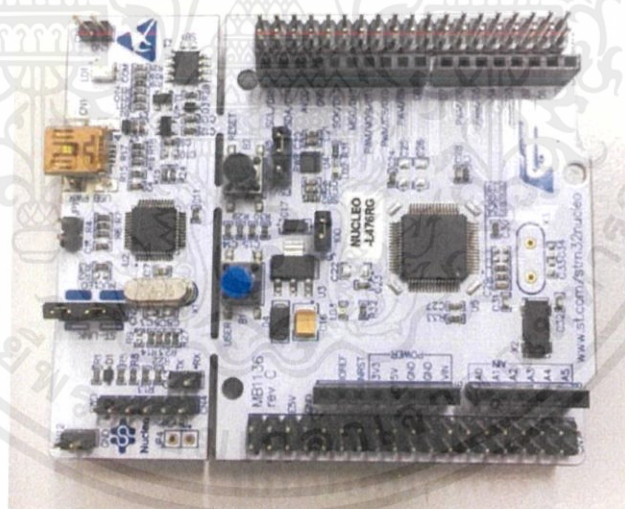
ข้อดี : มีความแม่นยำในการระบุตำแหน่งสูงมาก ความคลาดเคลื่อนอยู่ที่ 0.3 เมตร รวมถึงระยะ

ในการทำงานของอุปกรณ์นั้นสามารถทำงานได้ไกลมากในระยะไกลถึง 150 เมตร

ข้อเสีย : ตัวอุปกรณ์นั้นกินพลังงานสูงมาก รวมทั้งยังมีราคาสูงอีกด้วย

## 2.2 ไมโครคอนโทรลเลอร์ STM32L476

ไมโครคอนโทรลเลอร์ STM32L476 ในรูปที่ 2.5 เป็นไมโครคอนโทรลเลอร์ที่ใช้พลังงานต่ำ (Ultra-low-power) ทำงานอยู่บน High-performance Arm® Cortex®-M4 32-bit RISC core โดยตัวอุปกรณ์มีการใช้งานที่ความถี่ 80 MHz คุณสมบัติของ Cortex-M4 core ประกอบด้วย Floating Point Unit (FPU) ที่ช่วยในการประมวลผลข้อมูล นอกจากนี้ยังมีชุดคำสั่งทางด้าน DSP และ Memory Protection Unit (MPU) ซึ่งจะช่วยให้เรื่องความปลอดภัยของแอปพลิเคชันได้อีกด้วย



รูปที่ 2.5 บอร์ดไมโครคอนโทรลเลอร์ STM32L476 [1]

### ข้อมูลจำเพาะ

- Ultra-low-power with FlexPowerControl
- 1.71 V to 3.6 V power supply

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- -40 °C to 85/105/125 °C temperature range
- 300 nA in VBAT mode: supply for RTC and 32x32-bit backup registers
- 30 nA Shutdown mode (5 wakeup pins)
- 120 nA Standby mode (5 wakeup pins)
- 420 nA Standby mode with RTC
- 1.1  $\mu$ A Stop 2 mode, 1.4  $\mu$ A with RTC
- 100  $\mu$ A/MHz run mode (LDO Mode)
- 39  $\mu$ A/MHz run mode (@3.3 V SMPS Mode)
- Batch acquisition mode (BAM)
- 4  $\mu$ s wakeup from Stop mode
- Brown out reset (BOR)
- Interconnect matrix
- Core: ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait-state execution from Flash memory, frequency up to 80 MHz, MPU, 100DMIPS and DSP instructions
- Performance benchmark
  - 1.25 DMIPS/MHz (Drystone 2.1)
  - 273.55 CoreMark® (3.42 CoreMark/MHz @ 80 MHz)
- Energy benchmark
  - 220 ULPBENCH® score
- Clock Sources
  - 4 to 48 MHz crystal oscillator
  - 32 kHz crystal oscillator for RTC (LSE)
  - Internal 16 MHz factory-trimmed RC ( $\pm$ 1%)
  - Internal low-power 32 kHz RC ( $\pm$ 5%)
  - Internal multispeed 10.0 kHz to 48 MHz oscillator, auto-trimmed by LSE (better than  $\pm$ 0.25 % accuracy)
  - 3 PLLs for system clock, USB, audio, ADC

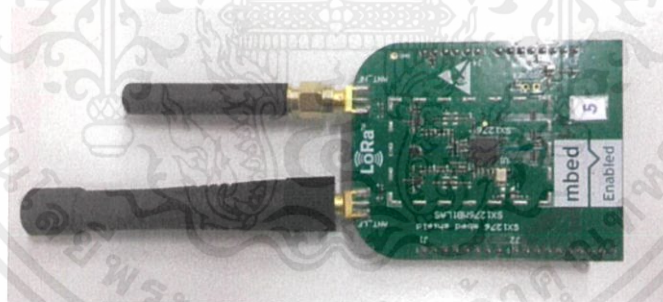
- 114 fast I/Os, most 5 V-tolerant, up to 14 I/Os with independent supply down to 1.08 V
- RTC with HW calendar, alarms and calibration
- LCD 8x 40 or 4x 44 with step-up converter
- 24 capacitive sensing channels: support touchkey, linear and rotary touch sensors
- 16x timers: 2x 16-bit advanced motor-control, 2x 32-bit and 5x 16-bit general purpose, 2x 16-bit basic, 2x low-power 16-bit timers (available in Stop mode), 2x watchdogs, SysTick timer
- Memories
  - 1 MB Flash, 2 banks read-while-write, proprietary code readout protection
  - 128 KB of SRAM including 32 KB with hardware parity check
  - External memory interface for static memories supporting SRAM, PSRAM, NOR and NAND memories
  - Quad SPI memory interface
- 4x digital filters for sigma delta modulator
- Rich analog peripherals (independent supply)
  - 3x 12-bit ADC 5 Msps, up to 16-bit with hardware oversampling, 200  $\mu$ A/Msps
  - 2x 12-bit DAC, low-power sample and hold
  - 2x operational amplifiers with built-in PGA
  - 2x ultra-low-power comparators
- 20x communication interfaces
  - USB OTG 2.0 full-speed, LPM and BCD
  - 2x SAs (serial audio interface)
  - 3x I2C FM+(1 Mbit/s), SMBus/PMBus
  - 5x USARTs (ISO 7816, LIN, IrDA, modem)
  - 1x LPUART (Stop 2 wake-up)

- 3x SPIs (4x SPIs with the Quad SPI)
- CAN (2.0B Active) and SDMMC interface
- SWPMI single wire protocol master I/F
- IRTIM (Infrared interface)
- 14-channel DMA controller
- True random number generator
- CRC calculation unit, 96-bit unique ID
- Development support: serial wire debug (SWD), JTAG, Embedded Trace Macrocell™

## 2.3 Transceiver

### 2.3.1 Sx1276 Transceiver

อุปกรณ์โมดูล Sx1276 Transceiver ในรูปที่ 2.6 เป็นอุปกรณ์ที่รองรับมาตรฐาน LoRa ซึ่งมีฟังก์ชันการสื่อสารแบบ Long range spread spectrum รวมถึงมีความสามารถป้องกันสัญญาณรบกวนและมีการใช้พลังงานจากกระแสไฟฟ้าต่ำอีกด้วย



รูปที่ 2.6 Sx1276 LoRa Module [2]

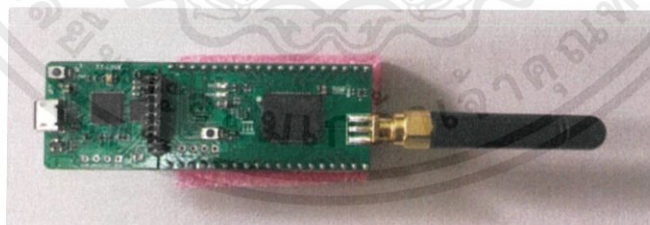
#### ข้อมูลจำเพาะ

- LoRa Modem
- 168 dB maximum link budget
- +20 dBm - 100 mW constant RF output vs. V supply
- +14 dBm high efficiency PA

- Programmable bit rate up to 300 kbps
- High sensitivity: down to -148 dBm
- Bullet-proof front end: IIP3 = -11 dBm
- Excellent blocking immunity
- Low RX current of 9.9 mA, 200 nA register retention
- Fully integrated synthesizer with a resolution of 61 Hz
- FSK, GFSK, MSK, GMSK, LoRa and OOK modulation
- Built-in bit synchronizer for clock recovery
- Preamble detection
- 127 dB Dynamic Range RSSI
- Automatic RF Sense and CAD with ultra-fast AFC
- Packet engine up to 256 bytes with CRC
- Built-in temperature sensor and low battery indicator

### 2.3.2 S76S Transceiver

อุปกรณ์โมดูล S76S Transceiver ในรูปที่ 2.7 เป็นอุปกรณ์ MCU ชนิดหนึ่งที่รองรับมาตรฐาน LoRa ซึ่งมีความสามารถในการสื่อสารแบบ Ultra-long range spread spectrum และยังสามารถป้องกันการรบกวนได้ดี



รูปที่ 2.7 S76S Module [6]

#### คุณสมบัติ

- Small footprint : 13 mm x 1 mm x 1mm

- LoRa™ Modem
- +20 dBm constant RF output vs. V supply
- Programmable bit rate up to 37500 bps
- High sensitivity: down to -137 dBm
- Excellent blocking immunity
- Preamble detection
- Automatic RF Sense and CAD with ultra-fast AFC
- Payload up to 128 bytes with CRC
- Embedded memories (up to 192 Kbytes of Flash memory and 20 Kbytes of RAM)

ตารางที่ 2.1 ข้อมูลจำเพาะของ S76S Module

Model Name	S76S
Product Description	LoRa Wireless Communication Module
Host Interface	UART
Operation Condition	
Temperature	<ul style="list-style-type: none"> <li>■ Storage : -50°C ~ +105°C</li> <li>■ Operating : -20°C ~ +70°C</li> </ul>
Humidity	<ul style="list-style-type: none"> <li>■ Operating : 10 ~ 95% (Non-Condensing)</li> <li>■ Storage : 5 ~ 95% (Non-Condensing)</li> </ul>
Dimension	13mm x 11mm x 1.1mm
Package	LGA type

## 2.4 การใช้งานคลื่นความถี่ Unlicensed band 920-925 MHz

ในยุคปัจจุบัน Smart Device จำนวนมากได้เข้ามามีบทบาทในการช่วยอำนวยความสะดวกในชีวิตให้กับเรามากขึ้น เพื่อที่จะทำให้ Smart Device แต่ละตัวทำงานร่วมกันได้อย่างมีประสิทธิภาพ จึงได้มีการนำอุปกรณ์เหล่านั้นไปเชื่อมต่อกับโครงข่าย Internet ซึ่งในการให้บริการ

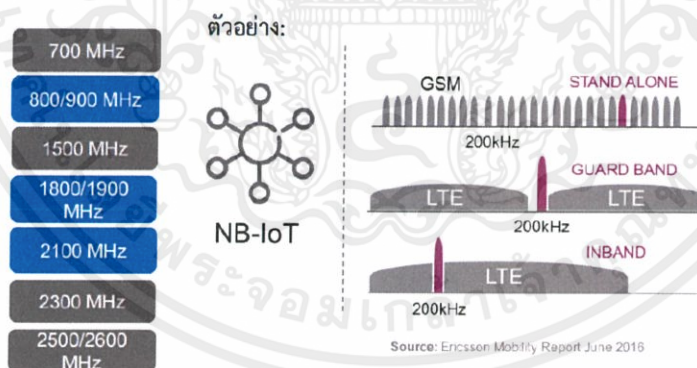
Internet แบบไร้สายนั้นมีการให้บริการอยู่ 2 แบบ ดังแสดงในรูปที่ 2.8 คือ การให้บริการย่านความถี่ที่ต้องมีใบอนุญาตและไม่ต้องมีใบอนุญาต



รูปที่ 2.8 การให้บริการย่านความถี่ที่ต้องมีใบอนุญาตและไม่ต้องมีใบอนุญาต [3]

1) ตัวอย่างการให้บริการบนคลื่นความถี่ที่ต้องมีใบอนุญาต (Licensed Spectrum)

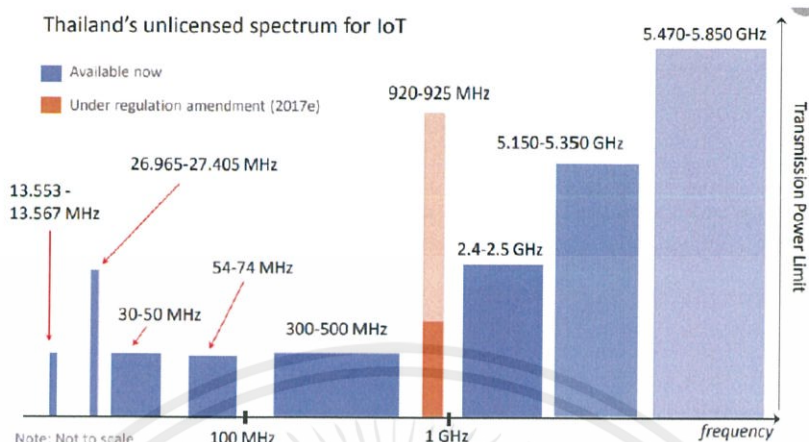
เช่น NB-IoT ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 ย่านความถี่ที่ต้องมีใบอนุญาต NB-IoT [3]

2) ตัวอย่างการให้บริการบนคลื่นความถี่ที่ได้รับการยกเว้นใบอนุญาต (Unlicensed band)

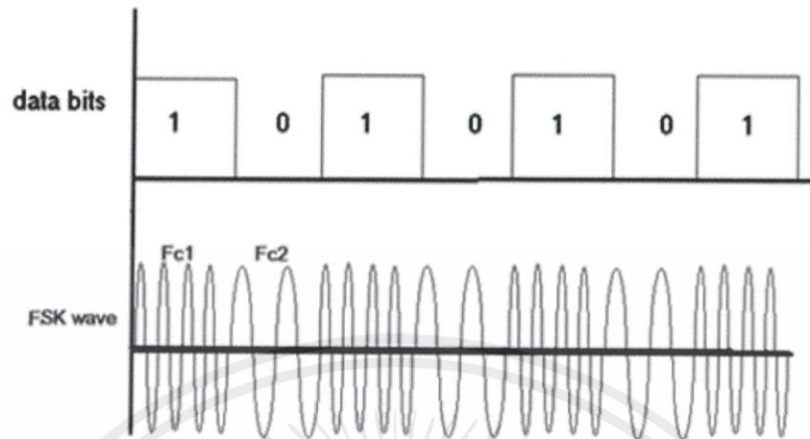
ในไทยนั้นมีการกำหนดสเปกตรัมไว้ในช่วง 920-925 MHz ดังแสดงในรูปที่ 2.10



รูปที่ 2.10 สเปกตรัมของย่านความถี่ที่ได้รับการยกเว้นใบอนุญาต [3]

## 2.5 การมอดูเลชันแบบ FSK (Frequency Shift Keying)

การมอดูเลชันแบบ FSK หรือ Frequency Shift Keying เป็นวิธีการมอดูเลชันที่ขนาดของคลื่นพาหะจะไม่มีการเปลี่ยนแปลง แต่ที่เปลี่ยนแปลงก็คือความถี่ของคลื่นพาหะ เมื่อบิตมีค่าเป็น 1 ความถี่จะสูงกว่าปกติ และเมื่อบิตมีค่าเป็น 0 ความถี่ของคลื่นจะต่ำกว่าปกติ เทคนิคนี้เป็นการมอดูเลตสัญญาณที่มีความถี่ 2 ค่า มีลักษณะสัญญาณดังรูปที่ 2.11 เมื่อสัญญาณมีสถานะ “High” สัญญาณ FSK มีความถี่เป็น  $F_{c1}$  และเมื่อสัญญาณมีสถานะ “Low” สัญญาณ FSK จะมีความถี่เป็น  $F_{c2}$  ซึ่งถ้าเมื่อค่าความแตกต่างระหว่าง  $F_{c1}$  และ  $F_{c2}$  มีค่ามากพอ จะทำให้การส่งและรับสัญญาณทำได้ดี เทคนิคการมอดูเลชันแบบ FSK เป็นเทคนิคที่นิยมใช้กันอย่างแพร่หลาย ซึ่งจะเห็นได้ชัดในด้านการสื่อสาร เช่น ในระบบการสื่อสารไร้สายและระบบสื่อสารแบบใช้สาย



รูปที่ 2.11 การมอดูเลชันแบบ FSK [4]

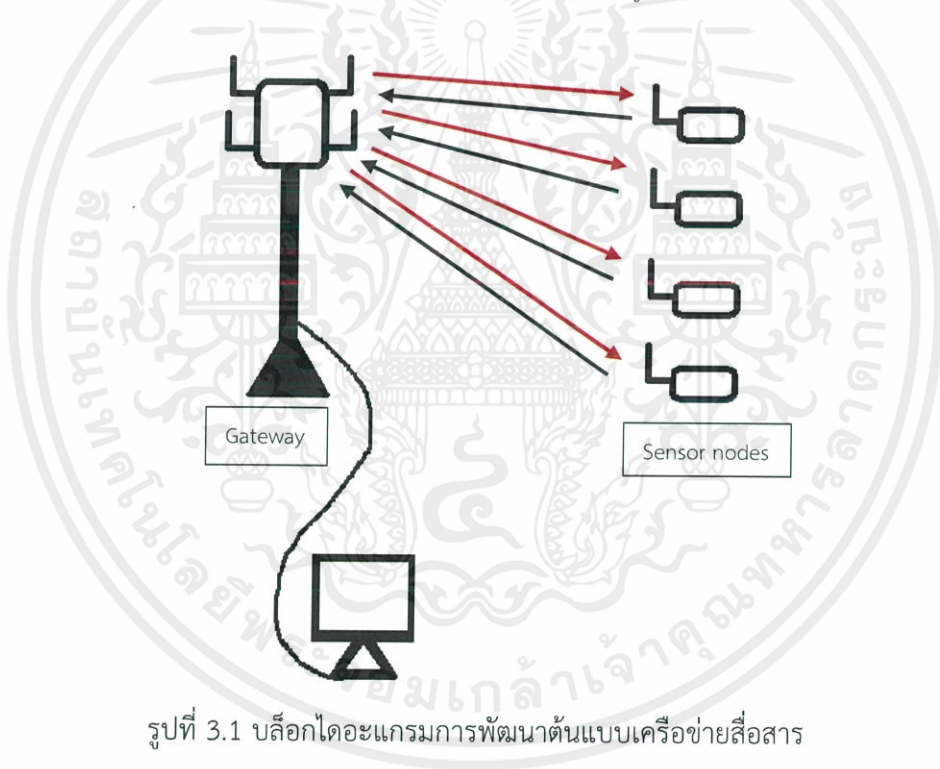
จากการมอดูเลชันแบบ FSK ที่กล่าวมาข้างต้นทำให้พบข้อดีหลายประการ จึงตัดสินใจนำมาใช้งานในโครงงานนี้ เช่น การมอดูเลชันแบบ FSK สามารถส่งข้อมูลได้เร็ว เนื่องจากความเร็วในการส่งไม่ขึ้นอยู่กักระยะทาง และสามารถจัดสรรช่องสัญญาณหรือแบ่งช่องสัญญาณได้ เนื่องจากมีการใช้งานแบนด์วิดธ์น้อย

### บทที่ 3

## การออกแบบและการจัดทำปฏิญญานิพนธ์

### 3.1 การออกแบบ

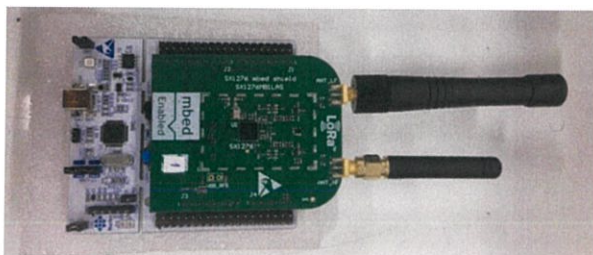
ปฏิญญานิพนธ์นี้ได้ทำการศึกษาและพัฒนาต้นแบบเครือข่ายสื่อสารเพื่อการประยุกต์ใช้ภายในอาคารหรือนำไปประยุกต์ใช้กับเทคโนโลยีอื่น ๆ ให้เกิดประโยชน์มากยิ่งขึ้น ด้วยความที่ระบบที่ถูกพัฒนาขึ้นมานี้มีข้อได้เปรียบหลายอย่างคือ ติดตั้งง่าย ราคาต่ำ ส่งได้ในระยะทางไกล มีความแม่นยำ และใช้งานบนย่านความถี่ Unlicensed band ทำให้ระบบนี้สามารถนำไปต่อยอดได้อีกตามความต้องการของผู้ใช้ ระบบที่ออกแบบไว้แสดงเป็นแผนภาพในรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมการพัฒนาต้นแบบเครือข่ายสื่อสาร

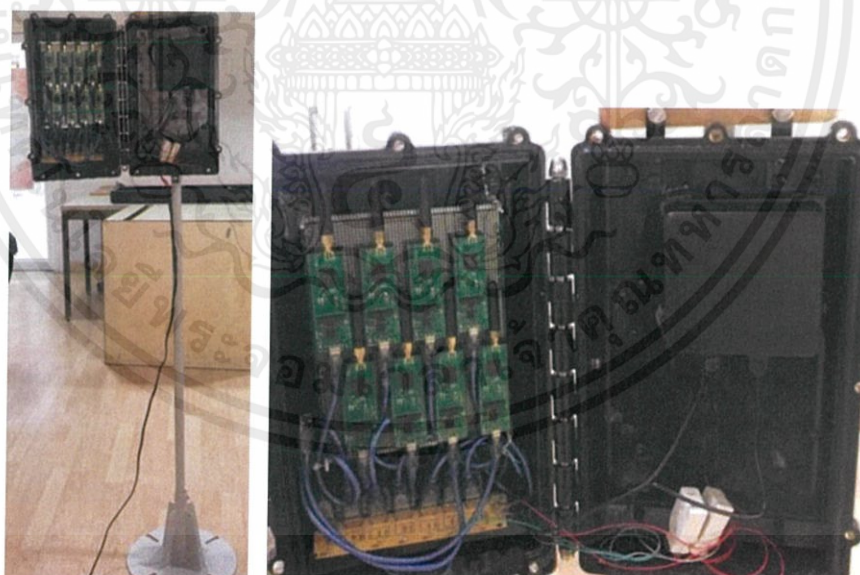
ระบบนี้ประกอบด้วยอุปกรณ์ 2 ส่วนหลัก ๆ ได้แก่

- 1) Sensor node ทำหน้าที่ในการส่งข้อมูลที่ต้องการไปให้ Gateway ในขั้นตอนการพัฒนา ระบบ ได้เลือกใช้บอร์ดไมโครคอนโทรลเลอร์ STM32L476 ทำหน้าที่ควบคุมการทำงาน โมดูลรับส่งข้อมูล Sx1276 สำหรับทำหน้าที่ Sensor node แต่ละตัว ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 บอร์ดไมโครคอนโทรลเลอร์ STM32L476 กับโมดูล Sx1276

2) Gateway ทำหน้าที่เป็นสถานีฐานรับข้อมูลจาก Sensor node ได้พร้อมกันหลายตัว เพื่อนำข้อมูลไปประมวลผล หรือส่งเข้าไปเครือข่ายอินเทอร์เน็ตต่อไป คณะผู้จัดทำได้เลือกใช้โมดูลรับส่งข้อมูล S76S ซึ่งมีไมโครคอนโทรลเลอร์ฝังมาบนบอร์ดแล้ว จำนวน N ตัว แต่ละตัวทำงานอิสระรับหน้าที่สื่อสารกับ Sensor node เพียงตัวเดียวในขณะเวลาหนึ่ง ๆ และทุกโมดูลนี้จะเชื่อมต่อเข้ากับเครื่องคอมพิวเตอร์ระบบ Windows (เลือกใช้ Intel NUC Mini PC) เพื่อความสะดวกในการบริหารระบบต่อไป รูปที่ 3.3 เป็นภาพถ่าย Gateway หนึ่งสถานีที่ได้สร้างขึ้นโดยบริษัท กสท โทรคมนาคม จำกัด (มหาชน)



(ก)

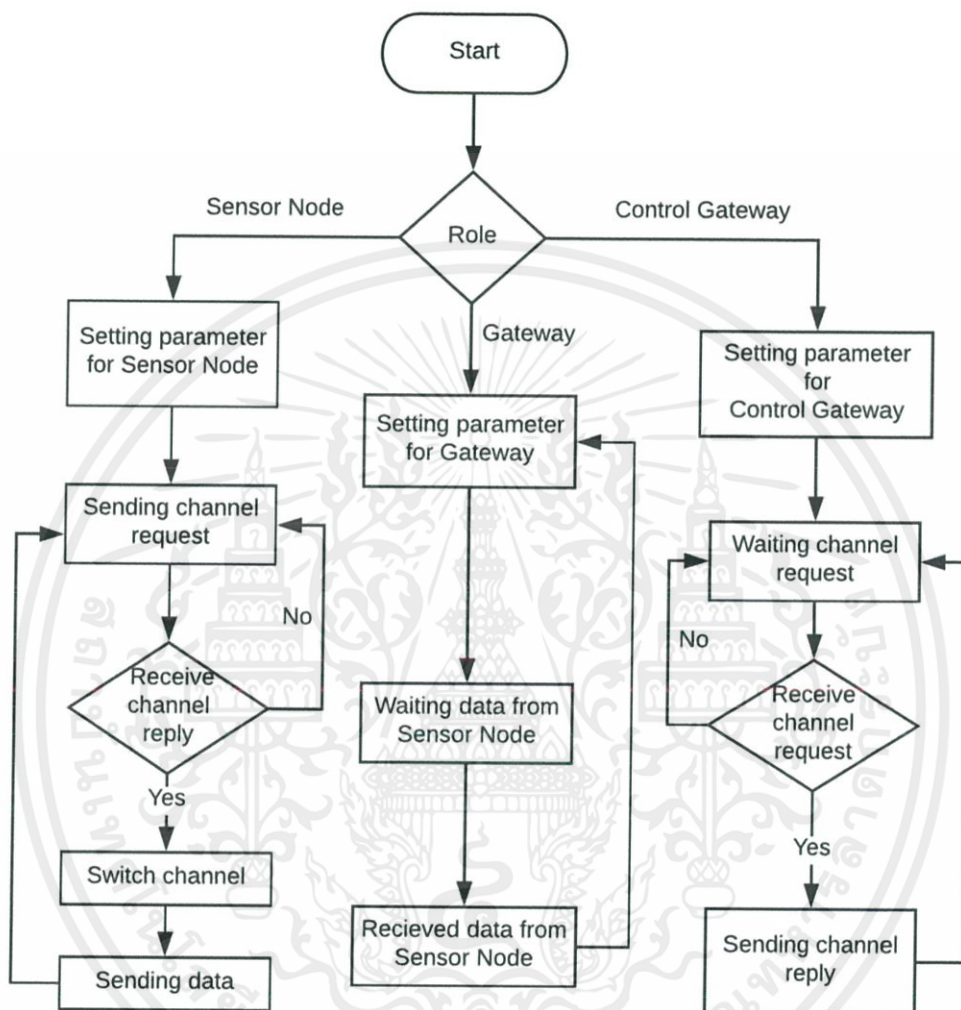
(ข)

รูปที่ 3.3 (ก) เสาที่ทำการประกอบ Gateway เข้าด้วยกัน

(ข) ภายในเสา Gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.1.1 การออกแบบการทำงาน



รูปที่ 3.4 แผนผังการทำงานของการพัฒนาต้นแบบเครือข่ายสื่อสาร

จากรูปที่ 3.4 มีขั้นตอนการทำงานโดยเริ่มต้นจากการกำหนดค่าพารามิเตอร์ของการสื่อสาร ซึ่งได้แก่ กำลังส่งสัญญาณ ขนาดข้อมูล ค่าความถี่ที่ใช้และเลือกรูปแบบการมอดูเลชัน โดยกำหนดพารามิเตอร์แบนด์วิดท์และความเร็วในการส่งข้อมูลเป็นพารามิเตอร์ในการมอดูเลชัน และต่อมาจึงเป็นการกำหนดหน้าที่ของโมดูลให้เป็นโมดูลที่ใช้สำหรับการส่งหรือรับข้อมูล โดยโมดูลที่ทำหน้าที่ส่งข้อมูลจะมีขั้นตอนการทำงานคือ เริ่มต้นจากการสุ่มค่าข้อมูลแล้วแสดงผ่านเทอร์มินอล

จากนั้นจึงส่งข้อมูลผ่านสัญญาณไร้สายไปยังโมดูลที่ทำหน้าที่รับข้อมูล แล้วกลับไปทำซ้ำใหม่ต่อไป ส่วนทางฝั่งโมดูลที่ทำหน้าที่รับข้อมูลจะมีขั้นตอนการทำงานคือ เริ่มต้นจากการอยู่ในสถานะรอรับข้อมูล โดยกำหนดค่าเวลา Timeout จากนั้นจึงรอจนกว่าจะมีข้อมูลส่งเข้ามา หากมีข้อมูลส่งเข้ามา จะเกิดการ Interrupt ขึ้น และเมื่อโมดูลได้รับข้อมูลจนเสร็จสิ้น จึงแสดงค่าผ่านทางเทอร์มินอลแล้วกลับไปทำซ้ำใหม่ต่อไป

### 3.1.2 การออกแบบการทำงานของ Sensor node

การออกแบบการทำงานของโมดูล Sx1276 ที่ทำหน้าที่เป็น Sensor node ถูกควบคุมการทำงานโดยการโปรแกรมคำสั่งผ่าน STM32L476 ซึ่งทำหน้าที่ในการส่งข้อมูล ซึ่งสามารถแบ่งส่วนของการออกแบบเป็น 1) ส่วนการแก้ไขการมอดูเลชันที่กำหนดการมอดูเลชันเป็นรูปแบบ FSK 2) ส่วนการแก้ไขขนาดข้อมูลที่จะทำให้สามารถส่งข้อมูลได้มากขึ้น และ 3) ส่วนการทำ Multi access เพื่อที่จะทำให้ Sensor node ทำงานพร้อมกันได้หลายตัว

#### 3.1.2.1 ส่วนการแก้ไขการมอดูเลชัน

ส่วนนี้เป็นการแก้ไขส่วนการมอดูเลชัน ซึ่งผู้จัดทำได้ทำการออกแบบให้สามารถมอดูเลชันแบบ FSK โดยจากเอกสารอ้างอิงของโมดูล Sx1276 ระบุว่าสามารถทำการมอดูเลชันได้หลากหลายรูปแบบ ซึ่งเหตุผลที่เลือกใช้การมอดูเลชันแบบ FSK เนื่องจากโลบาร์ของโมดูลรองรับการมอดูเลชันแค่ LoRa และ FSK เท่านั้น และด้วยความสามารถของ LoRa ที่มีการใช้พื้นฐานของเทคโนโลยี Chirp spread- spectrum (CSS) ทำให้ใช้แบนด์วิดธ์ในปริมาณมากในการส่งข้อมูลเพียง 1 ครั้ง และมีความยุ่งยากในการกำหนดพารามิเตอร์ตัวอื่น ๆ เช่น Spreading Factor (SF), กำลังส่งและแบนด์วิดธ์ ดังนั้นจึงไม่สามารถทำการแบ่งช่องสัญญาณเพื่อจะใช้ในการส่งได้ จึงเลือกใช้การมอดูเลชันแบบ FSK เนื่องจากสามารถกำหนดความกว้างของช่องสัญญาณได้ และใช้งานแบนด์วิดธ์ได้อย่างคุ้มค่า รวมถึงมีความเร็วในการส่งข้อมูลที่ดีกว่าการใช้ LoRa

#### 3.1.2.2 ส่วนการแก้ไขขนาดข้อมูล

ส่วนนี้จะทำการแก้ไขขนาดของข้อมูล Packet size ของโมดูล Sx1276 ที่ถูกควบคุมผ่านไมโครคอนโทรลเลอร์ STM32L476 จำนวน 2 บอร์ด ซึ่งทำหน้าที่เป็นฝั่งส่งและฝั่งรับกันให้สามารถทำการส่งข้อมูลได้อย่างมีประสิทธิภาพ โดยการเพิ่มปริมาณข้อมูลให้ส่งข้อมูลได้มากขึ้น และสามารถปรับขนาด Packet ได้ตามที่ต้องการ แต่เนื่องจากโหมด Variable length format นั้นสามารถส่งข้อมูลได้เท่ากับขนาดของ Buffer FIFO ของโมดูล (255 ไบต์) ดังนั้นถ้าต้องการส่งข้อมูลในปริมาณที่มากขึ้น ผู้จัดทำจึงได้ทำการพัฒนาให้ตัวโมดูลทำการส่งข้อมูลในโหมด Fix length

format แล้วทำการแก้ไขค่ารีจิสเตอร์ Fix length format ที่ควบคุมขนาดของ Payload ข้อมูลตามเอกสารอ้างอิงของโมดูล Sx1276 ได้มีการอธิบายเอาไว้ว่าสามารถที่จะทำการตั้งค่าขนาดของข้อมูลได้จากการเขียนโปรแกรมคำสั่ง โดยผู้จัดทำได้มีการแก้ไขขนาดข้อมูลดังนี้

1) ส่วนการแก้ไขขนาดข้อมูล 255 ไบต์

คือส่วนที่ใช้ในการกำหนดขนาดของข้อมูล โดยการกำหนดขนาดของข้อมูล (BUFFER\_SIZE) เป็น 255 ก็จะสามารถทำการส่งที่ขนาดข้อมูลขนาด 255 ไบต์ได้

```
#if defined(TXSite)
    random1=randr(1,0xffff);
    random2=randr(1,0xffff);
    Buffer[0] = 0xFF&random1;
    Buffer[1] = (0xFF00&random1)>>8;
    Buffer[2] = 0xFF&random2;
    Buffer[3] = (0xFF00&random2)>>8;
    Buffer[BufferSize-1] = 9;
    for( i = 4; i < BufferSize-1; i++ )
    {
        Buffer[i] = 1 ;
    }
    PRINTF("Payload<%d> : ",BufferSize);
    for( i = 0; i < 64; i++ )
    {
        PRINTF("%d",Buffer[i]);
    }
    PRINTF("\n\r");
    Radio.Send( Buffer, BufferSize );
    DelayMs(2000);
    State = LOWPOWER ;
    DISABLE_IRQ( );
    ENABLE_IRQ( );
#endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

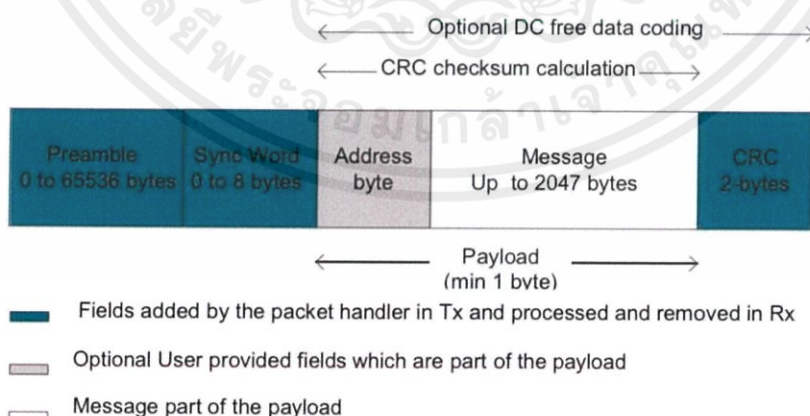
หลังจากนั้นจึงทำการเขียนโปรแกรมคำสั่ง สำหรับสุมตัวเลขเพื่อใช้ในการทดลองการส่งข้อมูลอีกครั้ง เพื่อทดสอบการส่งข้อมูลและให้แสดงผ่านหน้าเทอร์มินอล

## 2) ส่วนการแก้ไขขนาดข้อมูล 2047 ไบต์

จากเอกสารอ้างอิงของโมดูล Sx1276 พบว่าสามารถตั้งค่าการส่งให้มีขนาดข้อมูลสูงสุดที่ค่า 2047 ไบต์ได้ แต่ต้องทำการเขียนโปรแกรมแก้ไขที่รีจิสเตอร์รวมถึงต้องเขียนโปรแกรมเพิ่มเติมในตัวไลบรารีของโมดูล Sx1276 อีกด้วย โดยมีขั้นตอนการเขียนโปรแกรมแก้ไขดังนี้

2.1) ทำการแก้ไขชนิดของตัวแปร payloadLen เนื่องจากว่าในรูปแบบของขนาดข้อมูล 255 ไบต์ มีการกำหนดชนิดของตัวแปร payloadLen เพียง 8 บิต จึงทำให้สามารถกำหนดขนาดของข้อมูลการส่งได้เพียง 255 ไบต์ เท่านั้น ดังนั้นจึงทำการแก้ไขชนิดของตัวแปรให้เป็น 16 บิต เพื่อรองรับการเพิ่มขนาดของข้อมูลให้เป็น 2047 ไบต์

2.2) กำหนดค่าของรีจิสเตอร์สำหรับใช้งานในขนาดของข้อมูลการส่ง 2047 ไบต์ โดยได้ทำการแก้ไขที่รีจิสเตอร์ที่มีชื่อว่า REG\_PACKETCONFIG2 และ REG\_PAYLOADLENGTH จากนั้นทำการแก้ไขคำสั่งให้ดึงค่าจากรีจิสเตอร์ เพื่อที่จะอ่านค่าจากรีจิสเตอร์ และนำไปปรับค่าพารามิเตอร์ที่เกี่ยวข้องกับการรับส่งข้อมูล REG\_PAYLOADLENGTH จากนั้นทำการแก้ไขคำสั่งให้ดึงค่าจากรีจิสเตอร์เพื่อที่จะอ่านค่าจากรีจิสเตอร์ และนำไปปรับค่าพารามิเตอร์ที่เกี่ยวข้องกับการรับส่งข้อมูล แสดงถึงรูปแบบ Packet ของข้อมูล ดังรูปที่ 3.5 และค่ารีจิสเตอร์ที่ใช้ในการควบคุม Packet format ดังรูปที่ 3.6 ค่ารีจิสเตอร์ที่ใช้ในการควบคุมขนาดของ Payload ดังรูปที่ 3.7



รูปที่ 3.5 Fix Length Packet Format

Name (Address)	Bits	Variable Name	Mode	Default value	FSK/OOK Description
RegPacketConfig1 (0x30)	7	PacketFormat	rw	0x01	Defines the packet format used: 0 → Fixed length 1 → Variable length
	6-5	DcFree	rw	0x00	Defines DC-free encoding/decoding performed: 00 → None (Off) 01 → Manchester 10 → Whitening 11 → reserved
	4	CrcOn	rw	0x01	Enables CRC calculation/check (Tx/Rx): 0 → Off 1 → On
	3	CrcAutoClearOff	rw	0x00	Defines the behavior of the packet handler when CRC check fails: 0 → Clear FIFO and restart new packet reception. No <i>PayloadReady</i> interrupt issued. 1 → Do not clear FIFO. <i>PayloadReady</i> interrupt issued.
	2-1	AddressFiltering	rw	0x00	Defines address based filtering in Rx: 00 → None (Off) 01 → Address field must match <i>NodeAddress</i> 10 → Address field must match <i>NodeAddress</i> or <i>BroadcastAddress</i> 11 → reserved
	0	CrcWhiteningType	rw	0x00	Selects the CRC and whitening algorithms: 0 → CCITT CRC implementation with standard whitening 1 → IBM CRC implementation with alternate whitening

รูปที่ 3.6 รีจิสเตอร์ที่ใช้ในการควบคุม Packet format

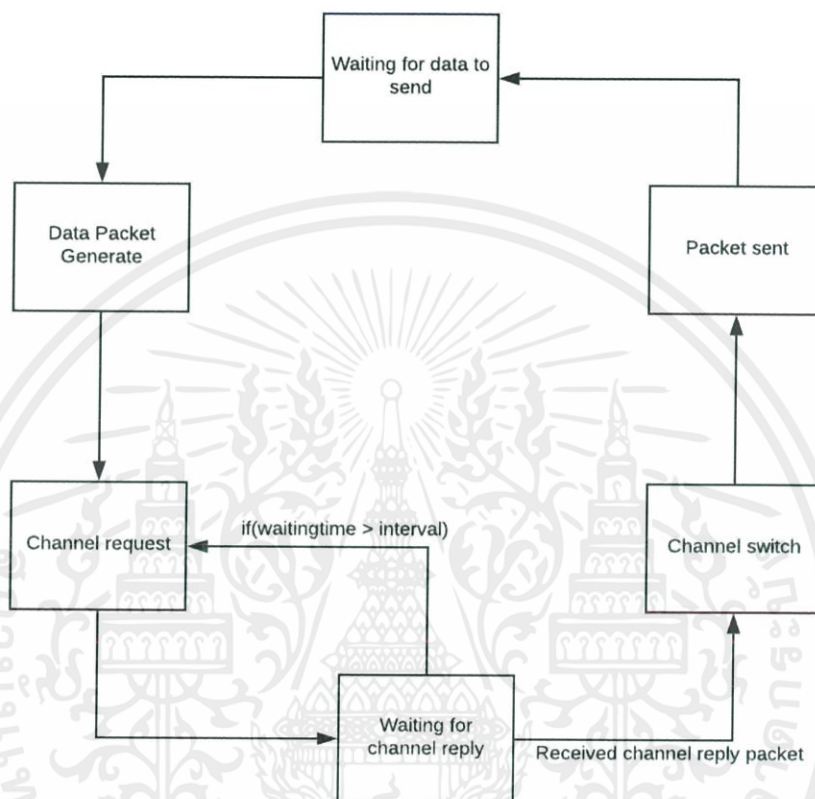
RegPacketConfig2 (0x31)	7	unused	r	-	unused
	6	DataMode	rw	0x01	Data processing mode: 0 → Continuous mode 1 → Packet mode
	5	IoHomeOn	rw	0x00	Enables the io-homecontrol <sup>®</sup> compatibility mode 0 → Disabled 1 → Enabled
	4	IoHomePowerFrame	rw	0x00	reserved - Linked to io-homecontrol <sup>®</sup> compatibility mode
	3	BeaconOn	rw	0x00	Enables the Beacon mode in Fixed packet format
RegPayloadLength (0x32)	2-0	PayloadLength(10:8)	rw	0x00	Packet Length Most significant bits
	7-0	PayloadLength(7:0)	rw	0x40	If PacketFormat = 0 (fixed), payload length. If PacketFormat = 1 (variable), max length in Rx, not used in Tx.
RegNodeAdrs (0x33)	7-0	NodeAddress	rw	0x00	Node address used in address filtering.
RegBroadcastAdrs (0x34)	7-0	BroadcastAddress	rw	0x00	Broadcast address used in address filtering.

รูปที่ 3.7 รีจิสเตอร์ที่ใช้ในการควบคุมขนาดของ Payload

### 3.1.2.3 ส่วนการทำ Multi access ของ Sensor node

ส่วนนี้เป็นการทำงานของโมดูล Sx1276 ที่ทำหน้าที่เป็น Sensor node ซึ่งส่วนนี้จะเป็นการออกแบบและอธิบายการทำงานของ Sensor node ตั้งแต่การเริ่มต้นการทำงาน

การร้องขอช่องสัญญาณจนไปถึงการส่งข้อมูลเมื่อช่องสัญญาณพร้อมใช้งาน รูปที่ 3.8 แสดงให้เห็นถึง State diagram การทำงานของ Sensor node



รูปที่ 3.8 State diagram การทำงานของ Sensor node

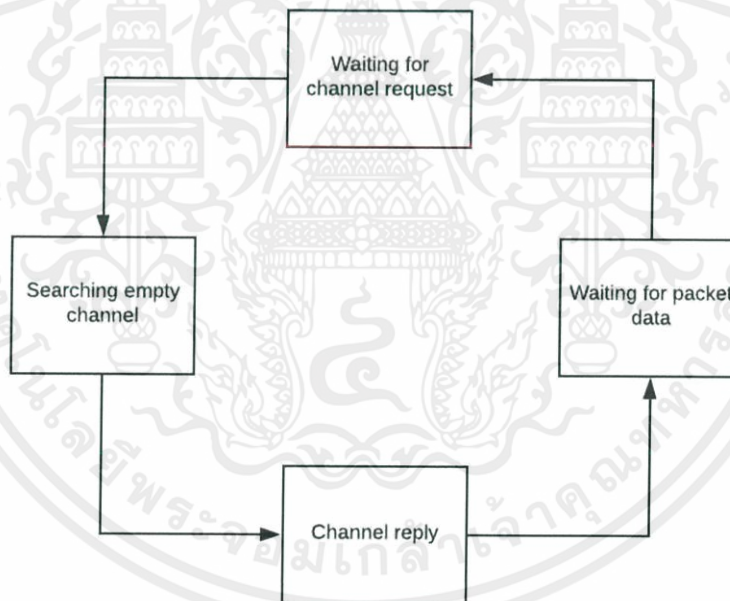
เริ่มต้นที่ Sensor node จะทำการรออยู่ในสถานะปกติ และเมื่อถึงเวลาที่ต้องการจะส่งข้อมูลไปให้ Gateway ตัว Sensor node จะเข้าสู่สถานะสร้าง Packet ข้อมูล จากนั้นจะเข้าสู่สถานะร้องขอช่องสัญญาณที่จะใช้ในการส่ง หลังจากร้องขอช่องสัญญาณเสร็จเรียบร้อย จะเข้าสู่สถานะรอการตอบกลับช่องสัญญาณที่จะใช้ในการส่ง เมื่อได้รับการตอบกลับช่องสัญญาณที่ใช้ในการส่งแล้วจะทำการส่งข้อมูลที่สร้างไว้ในช่องสัญญาณที่ได้รับมอบหมาย จากนั้นจะกลับไปรอส่งข้อมูลชุดถัดไป และทำการวนซ้ำในแต่ละสถานะต่อไป

### 3.1.3 การออกแบบการทำงานของ Gateway

การออกแบบการทำงานของโมดูล S76S ที่ทำหน้าที่เป็น Gateway เพื่อไว้สำหรับเป็นตัวรับข้อมูลที่ถูส่งมาจาก Sensor node และนำไปประมวลผลต่อไป ซึ่งส่วนของการออกแบบ คือ ส่วนการทำ Multi access ของ Gateway

#### 3.1.3.1 ส่วนการทำ Multi access ของ Gateway

ส่วนนี้เป็นการทำงานของโมดูล S76S ที่ทำหน้าที่เป็น Gateway ซึ่งส่วนนี้จะเป็นการออกแบบและอธิบายการทำงานของ Gateway ตั้งแต่การเริ่มต้นการทำงาน การควบคุมการให้ใช้ช่องสัญญาณ และทำการตอบกลับช่องสัญญาณเมื่อช่องสัญญาณพร้อมใช้งาน ซึ่งแสดง State diagram การทำงานของ Gateway ตามรูปที่ 3.9 แสดงให้เห็นถึง State diagram การทำงานของ Gateway



รูปที่ 3.9 State diagram การทำงานของ Gateway

เริ่มต้นที่ Gateway จะรอการร้องขอช่องสัญญาณจาก Sensor node หลังจากได้รับการร้องขอช่องสัญญาณจาก Sensor node แล้ว Gateway จะเข้าสู่สถานะค้นหาช่องสัญญาณโดยจะค้นหาช่องสัญญาณที่ไม่มีการใช้งานเพื่อนำไปเสนอให้กับ Sensor node หลังจากนั้นจะเข้าสู่

สถานะตอบกลับช่องสัญญาณที่ว่างหรือไม่มีการใช้งานไปยัง Sensor node แล้วจึงรอรับ Packet ข้อมูลในช่องสัญญาณนั้น ๆ และทำการวนซ้ำในแต่ละสถานะต่อไป

### 3.1.4 การออกแบบการทำงานร่วมกันของ Sensor node และ Gateway

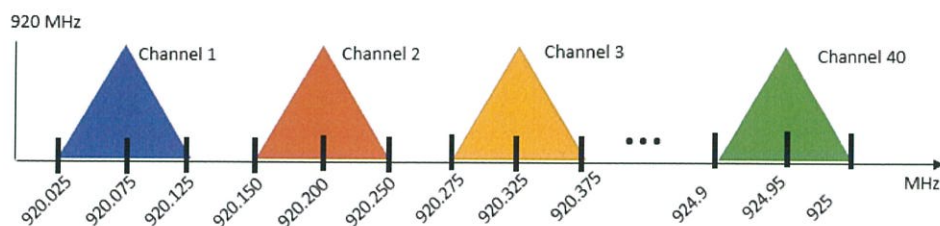
ส่วนนี้จะทำการออกแบบผ่านการเขียนโปรแกรมคำสั่งควบคุมการทำงานของโมดูล Sx1276 ที่ทำหน้าที่เป็น Sensor node และโมดูล S76S ที่ทำหน้าที่เป็น Gateway ให้สามารถส่งข้อมูลจาก Sensor node ไปยัง Gateway ได้ ดังแสดงตามรูปที่ 3.10 บล็อกไดอะแกรมการส่งข้อมูลจาก Sensor node ไปยัง Gateway



รูปที่ 3.10 บล็อกไดอะแกรมการส่งข้อมูลจาก Sensor node ไปยัง Gateway

#### 3.1.4.1 การแบ่งช่องสัญญาณ

จากการสื่อสารกันระหว่าง Sensor node และ Gateway นั้น ในกรณีที่มีจำนวนของ Sensor node มากกว่า 1 ตัวขึ้นไป อาจส่งผลให้เกิดการชนกันหรือรบกวนกันของข้อมูลได้ จึงต้องมีการออกแบบช่องสัญญาณขึ้นระหว่าง Sensor node และ Gateway โดยทำการกำหนดช่วงความถี่ที่ใช้ในการสื่อสารไว้ที่ 920–925 MHz ซึ่งเป็นย่านความถี่ Unlicensed band และมีการกำหนดขนาดแบนด์วิดท์ในแต่ละช่องสัญญาณไว้ที่ช่องละ 100 kHz รวมไปถึงกำหนดการ์ดแบนด์ (Guard band) เพื่อป้องกันการชนกันของสัญญาณเป็น 25 kHz ตามรูปที่ 3.11 ภาพจำลองการแบ่งช่องสัญญาณช่วงความถี่ 920–925 MHz



รูปที่ 3.11 จำลองการแบ่งช่องสัญญาณช่วงความถี่ 920–925 MHz

จากการคำนวณการแบ่งช่องสัญญาณในช่วงความถี่ที่ใช้ในการสื่อสารระหว่างความถี่ที่ 920-925 MHz จะสามารถแบ่งช่องสัญญาณได้ประมาณ 40 ช่องสัญญาณ โดยกำหนด 1 ช่อง เป็นช่องที่ใช้ในการควบคุม ดังแสดงการแบ่งช่องสัญญาณได้ตามตารางที่ 3.1

ตารางที่ 3.1 การแบ่งช่องสัญญาณช่วงความถี่ 920–925 MHz

Channel	Frequency (MHz)		
	Lower	Center	Upper
Control	920.025	920.075	920.125
1	920.150	920.200	920.250
2	920.275	920.325	920.375
3	920.400	920.450	920.500
4	920.525	920.575	920.625
5	920.650	920.700	920.750
6	920.775	920.825	920.875
7	920.900	920.950	921.000
8	921.025	921.075	921.125
9	921.150	921.200	921.250
10	921.275	921.325	921.375
11	921.400	921.450	921.500
12	921.525	921.575	921.625
13	921.650	921.700	921.750
14	921.775	921.825	921.875

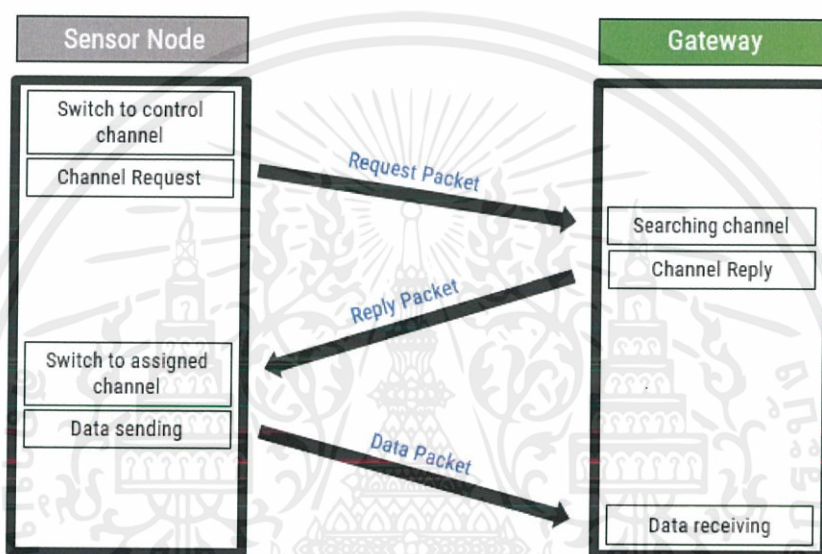
ตารางที่ 3.1 การแบ่งช่องสัญญาณช่วงความถี่ 920–925 MHz (ต่อ)

Channel	Frequency (MHz)		
	Lower	Center	Upper
15	921.900	921.950	922.000
16	922.025	922.075	922.125
17	922.150	922.200	922.250
18	922.275	922.325	922.375
19	922.400	922.450	922.500
20	922.525	922.575	922.625
21	922.650	922.700	922.750
22	922.775	922.825	922.875
23	922.900	922.950	923.000
24	923.025	923.075	923.125
25	923.150	923.200	923.250
26	923.275	923.325	923.375
27	923.400	923.450	923.500
28	923.525	923.575	923.625
29	923.650	923.700	923.750
30	923.775	923.825	923.875
31	923.900	923.950	924.000
32	924.025	924.075	924.125
33	924.150	924.200	924.250
34	924.275	924.325	924.375
35	924.400	924.450	924.500
36	924.525	924.575	924.625
37	924.650	924.700	924.750
38	924.775	924.825	924.875
39	924.900	924.950	925.000

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

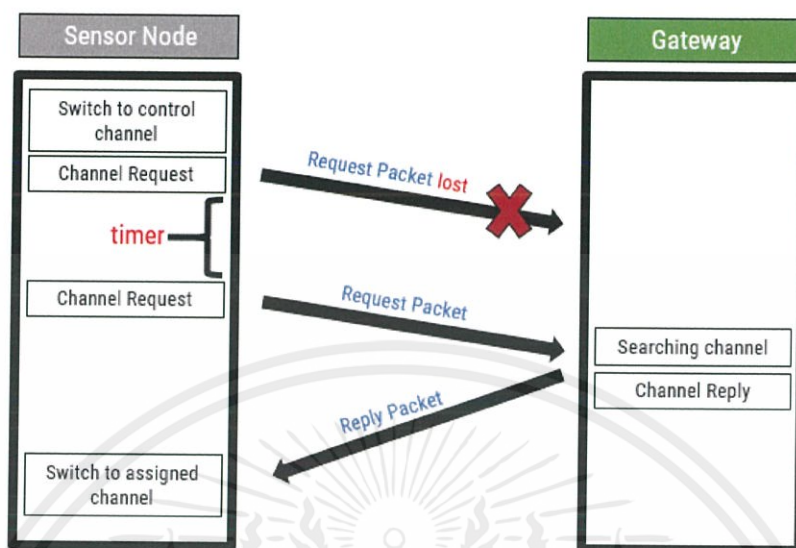
### 3.1.4.2 การตัดสินใจเลือกใช้ช่องสัญญาณ

จากหลักการแบ่งช่องสัญญาณข้างต้น เป็นการแบ่งช่องสัญญาณเพื่อให้ Sensor node สามารถส่งข้อมูลได้โดยไม่เกิดการชนกันหรือการรบกวนกันของสัญญาณข้อมูล แต่ถึงอย่างนั้นตัว Sensor node เองก็ไม่อาจทราบได้ว่าตัวเองต้องใช้ช่องสัญญาณช่องใด จึงได้มีการออกแบบโปรโตคอลที่ใช้สำหรับตัดสินใจใช้ช่องสัญญาณขึ้น



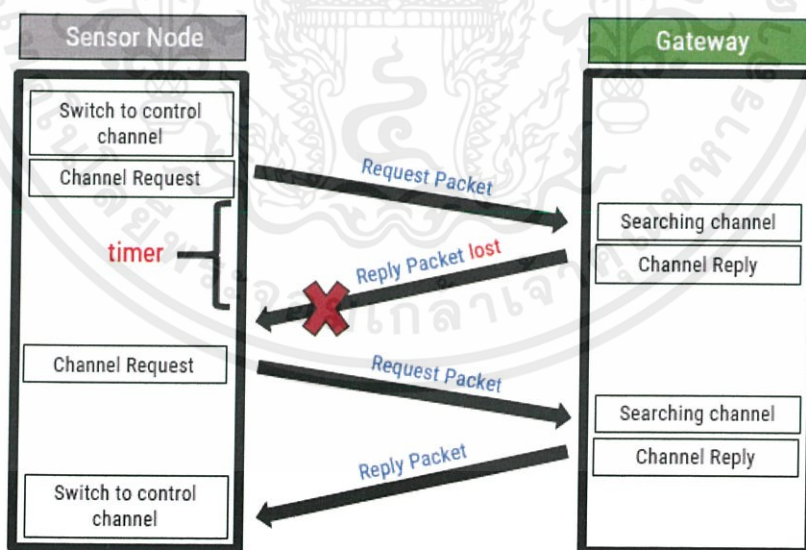
รูปที่ 3.12 โปรโตคอลการตัดสินใจเลือกใช้ช่องสัญญาณ

ในรูปที่ 3.12 เมื่อ Sensor node ต้องการส่งข้อมูลจะทำการส่ง Request Packet ผ่านช่องสัญญาณหลักที่ใช้ในการทำ Channel Control (Channel 0) ไปยัง Gateway Channel Control เมื่อ Gateway Channel Control ได้รับ Request Packet จะทำการมองหาช่องสัญญาณที่ว่างอยู่ จากนั้นจะทำการส่ง Reply Packet ไปยัง Sensor node และเมื่อ Sensor node ได้รับ Reply Packet แล้วจะย้ายไปยังช่องสัญญาณที่ Gateway Channel Control บอกมา แล้วจึงทำการส่งข้อมูลไป จากนั้นจะย้ายกลับมายังช่องหลักที่ใช้ในการทำ Channel Control (Channel 0) เช่นเดิม



รูปที่ 3.13 การทำงานในกรณีที่ Channel Request Packet สูญหาย

รูปที่ 3.13 ในกรณีที่ Request Packet สูญหาย ซึ่งอธิบายได้ว่าเมื่อ Sensor node ส่ง Request Packet ไปแล้วและไม่ได้รับการตอบ Reply Packet กลับมา เมื่อผ่านไประยะเวลาหนึ่ง Sensor node จะทำการส่ง Request Packet อีกครั้ง (ระยะเวลา Timer)



รูปที่ 3.14 การทำงานในกรณีที่ Channel Reply Packet สูญหาย

รูปที่ 3.14 ในกรณีที่ Reply Packet สูญหาย ใช้รูปแบบการตรวจจับแบบเดียวกันกับกรณีที่ Request Packet สูญหาย เมื่อ Sensor node ทำการส่ง Request Packet แล้วไม่ได้รับการตอบ Reply Packet กลับมา เมื่อผ่านไประยะเวลาหนึ่ง Sensor node จะทำการส่ง Request Packet อีกครั้งหนึ่ง

#### 3.1.4.3 รูปแบบข้อมูลของการเลือกใช้ช่องสัญญาณ

จากหลักการเลือกช่องสัญญาณที่ถูกกล่าวมาข้างต้น ในการเลือกใช้ช่องสัญญาณจำเป็นจะต้องใช้ Request Packet และ Reply Packet ซึ่ง Packet ทั้งสองนี้จะต้องมีเอกลักษณ์เฉพาะตัวที่จะกำหนดประเภทของ Packet นั้น ๆ เพื่อที่จะให้ Sensor node และ Gateway สามารถเข้าใจได้ว่าควรจะทำสิ่งใดหลังจากได้รับ Packet นั้น ผู้จัดทำจึงได้ทำการออกแบบ Packet ที่ใช้ภายในระบบ ดังแสดงในรูปที่ 3.15

Preamble	Dest.	Src.	Type	Rssi
Payload 2039 byte			Preamble	

รูปที่ 3.15 Packet Field

โดย

Preamble คือ ส่วนเริ่มต้นและปิดท้ายของ Header

Src. และ Dest. คือ ฟิ��ลด์ที่ใช้ระบุตัวตนของ Sensor node หรือ Gateway

Type คือ ฟิล์ดที่ใช้ในการระบุว่า Packet นี้เป็น Request Packet, Reply Packet หรือเป็น Data Packet ที่ใช้ในการส่งข้อมูลทั่วไป

Rssi คือ Field strength ใช้สำหรับประกอบการตัดสินใจเลือกช่องสัญญาณของ Gateway

Preamble	0xFF	Src.	0xFF	Rssi
Unused			Preamble	

รูปที่ 3.16 Request Packet

รูปที่ 3.16 แสดง Request Packet ที่มีการแสดงฟิลด์ของ Dest. ที่ถูกระบุไว้เป็น 0xFF ซึ่งเสมือนเป็น Broadcast Address ให้กับ Gateway และฟิลด์ Type ก็จะเป็น 0xFF เช่นกัน ส่วนในฟิลด์ของ Unused จะถูกระบุไว้เป็น 0x41

Preamble	Dest.	0xFF	0xFE	Rssi
CH Number	Unused		Preamble	

รูปที่ 3.17 Reply Packet

รูปที่ 3.17 แสดง Reply Packet มีการแสดงฟิลด์ของ Src. ที่ถูกระบุไว้เป็น 0xFF และแสดงฟิลด์ Type เป็น 0xFE ส่วนในฟิลด์ของ Unused จะถูกระบุไว้เป็น 0x41 และจะมีการเพิ่มฟิลด์ Channel number ที่ Gateway เลือกให้ Sensor node เพิ่มเข้ามาอีกด้วย

Preamble	Dest.	Src.	0x6A	Rssi
Payload 2039 byte			Preamble	

รูปที่ 3.18 Data Packet

รูปที่ 3.18 เป็นฟิลด์ของ Data Packet ที่ใช้ในการส่งข้อมูลทั่วไป หลังจากที่ได้มีการเลือกช่องสัญญาณโดย Gateway Channel Control แล้วจึงทำการส่งข้อมูลที่ต้องการส่งได้ ส่วนฟิลด์ Type จะระบุไว้เป็น 0x6A และ Payload ระบุขนาดของฟิลด์เป็น 2039 byte เพื่อรองรับจำนวนข้อมูลที่ต้องการส่ง

### 3.2 เครื่องมือที่ใช้ในการทดลอง

- |                                   |         |
|-----------------------------------|---------|
| 3.2.1 ไมโครคอนโทรลเลอร์ STM32L476 | 3 บอร์ด |
| 3.2.2 โมดูล Sx1276 Transceiver    | 3 บอร์ด |
| 3.2.3 โมดูล S76S Transceiver      | 5 บอร์ด |
| 3.2.4 RTL-SDR                     | 1 ตัว   |

### 3.3 การจัดเก็บผลการทดลอง

#### 3.3.1 ผลการทดสอบการทำงานของ Sensor node

- 1) ผลการทดสอบประสิทธิภาพในการส่งข้อมูลของการมอดูเลชันแบบ FSK
- 2) ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์ควบคุม Sensor node โดยการเขียนโปรแกรมแก้ไขขนาดของข้อมูล
- 3) ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์ควบคุม Sensor node โดยการเขียนโปรแกรมคำสั่งฝั่งส่งและฝั่งรับให้สามารถสื่อสารกันได้
- 4) ผลการทดสอบการวัดสเปกตรัมที่ได้จากการทำงานของ Sensor node

#### 3.3.2 ผลการทดสอบการทำงานของ Gateway

- 1) ผลการทดสอบการวัดสเปกตรัมที่ได้จากการทำงานของ Gateway

#### 3.3.3 ผลการทดสอบการสื่อสารกันระหว่าง Sensor node และ Gateway

- 1) ผลการทดสอบการส่งข้อมูลจาก Sensor node ไปยัง Gateway
- 2) ผลการทดสอบการส่งข้อมูลระหว่าง Sensor node และ Gateway โดยทำการเลือกช่องสัญญาณที่แตกต่างกัน เพื่อทดสอบการแบ่งช่องสัญญาณ
- 3) ผลการทดสอบการสื่อสารกันระหว่าง Sensor node และ Gateway ว่าสามารถทำงานได้จริง

## บทที่ 4

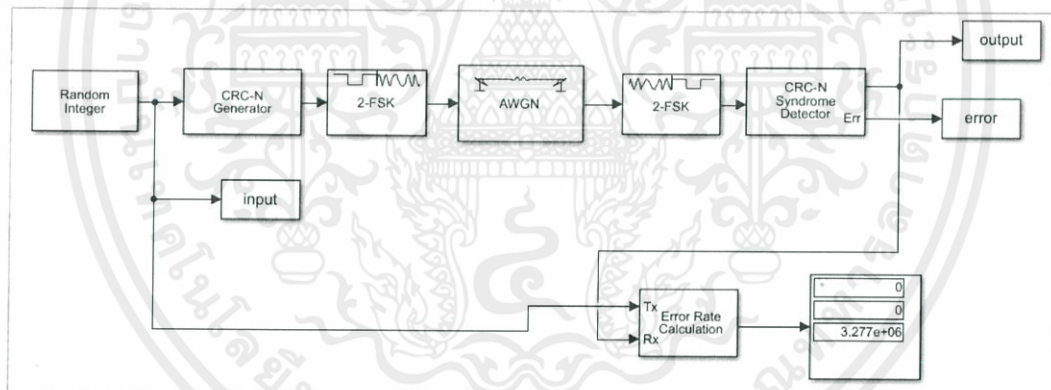
### ผลการทดลอง

สำหรับการทดลองการพัฒนาต้นแบบเครือข่ายสื่อสารเพื่อการประยุกต์ใช้ในการระบุตำแหน่งภายในอาคารนั้น ได้ทำการจัดเก็บผลการทำงานของระบบ โดยแบ่งการทดลองและจัดเก็บผลการทดลองเป็นส่วนต่าง ๆ ดังต่อไปนี้

#### 4.1 ผลการทดสอบการทำงานของ Sensor node

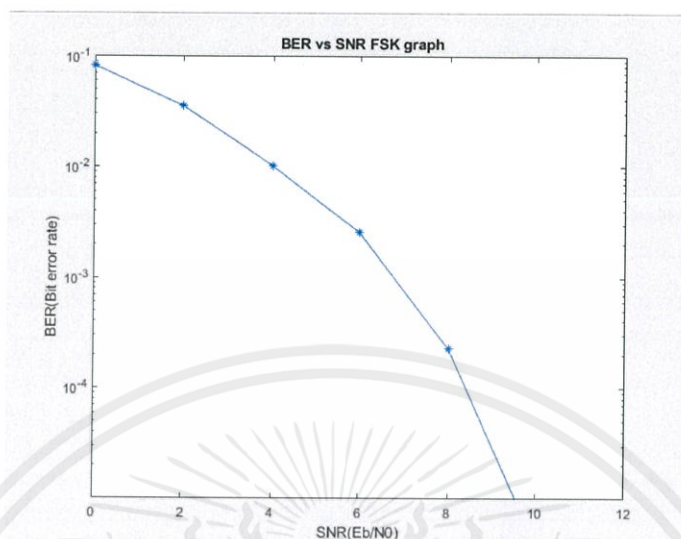
##### 4.1.1 ผลการทดสอบประสิทธิภาพในการส่งข้อมูลของการมอดูเลชันแบบ FSK

ทำการสร้าง Simulink ในโปรแกรม Matlab เพื่อจำลองระบบการทำงานของการพัฒนาต้นแบบเครือข่ายสื่อสารว่ามีประสิทธิภาพการทำงานเป็นอย่างไร โดยได้ทำการบันทึกค่า BER (Bit Error Rate) เทียบกับ SNR (Signal to Noise Ratio) ซึ่งได้แสดง Simulink การทำงานของระบบดังรูปที่ 4.1 และแสดงกราฟการเปรียบเทียบค่าระหว่าง BER และ SNR ดังรูปที่ 4.2



รูปที่ 4.1 Simulink การทำงานของระบบ

SNR (Signal to Noise Ratio) คือ การวัดประสิทธิภาพของการส่งสัญญาณ คำนวณได้จากอัตราส่วนกำลังของสัญญาณที่ส่งออกไปต่อสัญญาณรบกวน มีหน่วยเป็น เดซิเบล (dB)



รูปที่ 4.2 การเปรียบเทียบค่าระหว่าง BER และ SNR

จากกราฟสามารถอธิบายได้ว่า เมื่อค่า BER มากขึ้นเรื่อย ๆ จะมีความผิดพลาดในการส่งข้อมูลน้อยลง กล่าวคือค่า SNR มีผลต่อการรับส่งสัญญาณ เมื่อมีการรับส่งสัญญาณในบริเวณที่มีกำลังของสัญญาณรบกวนต่ำหรือช่องสัญญาณมีค่า SNR ที่ดี จะส่งผลให้เกิดความผิดพลาดน้อยลง และเป็นประโยชน์ต่อการส่งสัญญาณ ทำให้การรับส่งสัญญาณมีประสิทธิภาพมากขึ้น

#### 4.1.2 การเขียนโปรแกรมแก้ไขขนาดของข้อมูล

ในส่วนของการทำงานส่วนอุปกรณ์ไมโครคอนโทรลเลอร์ STM32L476 ซึ่งเป็นการเขียนโปรแกรมภาษาซี เพื่อทำการควบคุมการทำงานของโมดูล Sx1276 จำนวน 2 บอร์ด โดยการเขียนโปรแกรมแก้ไขขนาดของข้อมูลผ่านโปรแกรม Keil version 5 ที่ใช้สำหรับควบคุมส่วนฮาร์ดแวร์ เป็นการทดสอบการทำงานของฝั่งส่งและฝั่งรับ แสดงผลผ่านหน้าเทอร์มินอล ดังแสดงตามรูปที่ 4.3 แสดงการเขียนโปรแกรมควบคุมการทำงานของฝั่งส่งและฝั่งรับที่ขนาดข้อมูล 255 ไบต์



### 4.1.3 การเขียนโปรแกรมคำสั่งควบคุม Sensor node ฝั่งส่งและฝั่งรับให้สื่อสารกันได้

ดำเนินการศึกษาและทำความเข้าใจการเขียนโปรแกรมภาษาซี ในโปรแกรม Keil version 5 ที่ใช้ในการควบคุมส่วนฮาร์ดแวร์ ซึ่งได้แก่ ไมโครคอนโทรลเลอร์ STM32L476 ที่ทำหน้าที่ควบคุมโมดูล Sx1276 โดยเป็นการทดสอบการทำงานของฝั่งส่งและฝั่งรับ แสดงผลลัพธ์ได้ดังแสดงตามรูปที่ 4.5

Sensor node 1	Sensor node 2
txDone	txDone
OnRxDone	OnRxLineOut
RssiValue=-83 dBm, SnrValue=-19	rxLineOut
rxDone	OnRxDone
...PING	RssiValue=-80 dBm, SnrValue=-16
OnTxDone	rxDone
txDone	...PONG
OnRxDone	OnTxDone
RssiValue=-85 dBm, SnrValue=-19	txDone
rxDone	OnRxDone
...PING	RssiValue=-80 dBm, SnrValue=-17
OnTxDone	rxDone

รูปที่ 4.5 ทดสอบการทำงานของฝั่งรับและฝั่งส่ง

ตามรูปแสดงผลพัลส์ระหว่างการส่งและรับกันระหว่างโมดูล Sx1276 จำนวน 2 บอร์ด โดยกำหนดทางฝั่ง Sensor node 1 ให้ ส่งข้อความ “Ping” และให้ทางฝั่ง Sensor node 2 รับข้อความ “Pong” และมีการแสดงค่า Rssi value รวมไปถึงค่า SNR ด้วย

### 4.1.4 ผลการทดสอบการวัดสเปกตรัมที่ได้จากการทำงานของ Sensor node

ทำการทดสอบรับค่าสัญญาณที่ถูกส่งออกมาจากโมดูล Sx1276 ด้วย RTL-SDR โดยการรับสัญญาณนั้นและใช้โปรแกรม SDRsharp



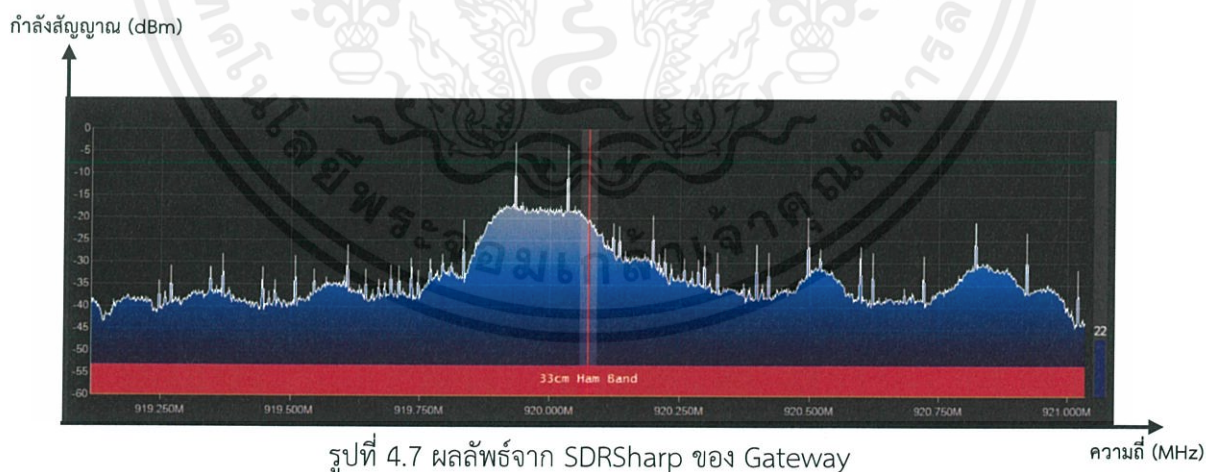
รูปที่ 4.6 ผลลัพธ์จาก SDRSharp ของ Sensor node

จากรูปที่ 4.6 แสดงการรับสัญญาณจากการใช้โปรแกรม SDRsharp จากรูปเป็นการแสดงสเปกตรัมของสัญญาณที่ส่งออกมาเป็นช่วง ๆ ซึ่งแสดงให้เห็นถึงค่าแบนด์วิดธ์ขนาด 100 kHz กำลังของสัญญาณขนาด -35 dBm และความถี่กลางอยู่ที่ 920.900 MHz ตามที่ได้ตั้งโปรแกรมไว้

## 4.2 ผลการทดสอบการทำงานของ Gateway

### 4.2.1 ผลการทดสอบการวัดสเปกตรัมที่ได้จากการทำงานของ Gateway

ทำการทดสอบรับค่าสัญญาณที่ถูกส่งออกมาจากโมดูล S76S ด้วย RTL-SDR โดยการรับสัญญาณนั้นและใช้โปรแกรม SDRsharp



รูปที่ 4.7 ผลลัพธ์จาก SDRSharp ของ Gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.7 แสดงการรับสัญญาณจากการใช้โปรแกรม SDRsharp จากรูปแสดงสเปกตรัมของสัญญาณที่ส่งออกมาเป็นช่วง ๆ ซึ่งแสดงให้เห็นถึงค่าแบนด์วิดธ์ขนาด 100 kHz กำลังของสัญญาณขนาด -25 dBm และความถี่กลางอยู่ที่ 920.900 MHz

### 4.3 ผลการทดสอบการสื่อสารกันระหว่าง Sensor node และ Gateway

ทดลองการทำงานร่วมกันของโมดูล Sx1276 และโมดูล S76S ที่ทำหน้าที่เป็น Sensor node และ Gateway ตามลำดับ

#### 4.3.1 ผลการทดสอบการส่งข้อมูลจาก Sensor node ไปยัง Gateway

การทดสอบการส่งข้อมูลจากโมดูล Sx1276 ที่ทำหน้าที่เป็น Sensor node ไปยังโมดูล S76S ที่ทำหน้าที่เป็น Gateway นั้นเป็นการสื่อสารกันระหว่างอุปกรณ์ 2 ตัว โดยที่ Sensor node จะทำการส่ง Packet ที่มีการสุ่มไปยัง Gateway และทางฝั่ง Gateway ก็สามารถรับ Packet ที่ถูกส่งมานี้ได้อย่างถูกต้อง ดังแสดงตามรูปที่ 4.8 การส่งข้อมูลจาก Sensor node ไป Gateway

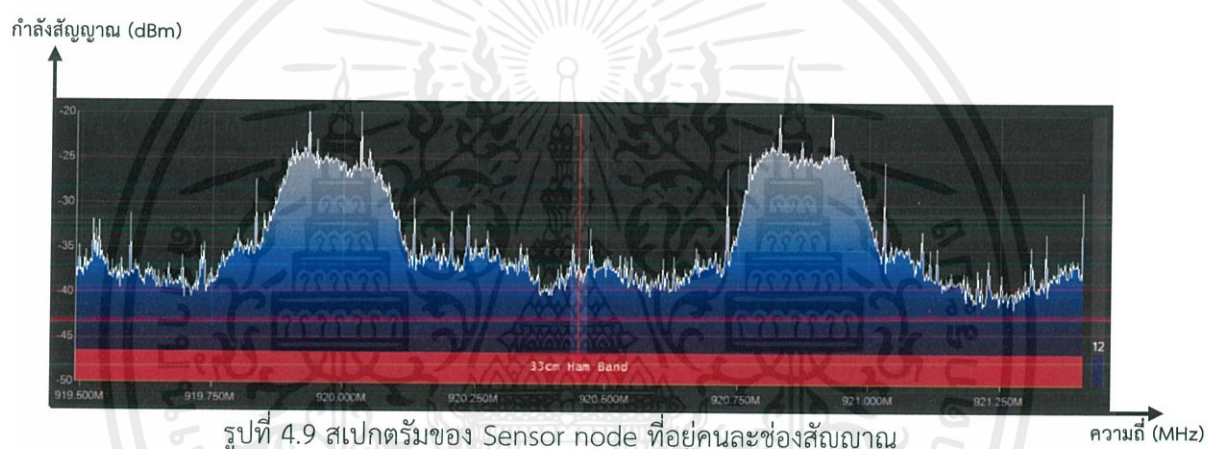
Sensor node	Gateway
OnTxDone	RxDone<207> : 701000111111...252253254255
txDone	RxDone<247> : 1201040111111...252253254255
Payload<2047> : 20290111111...252253254255	RxDone<207> : 1801360111111...252253254255
OnTxDone	RxDone<2047> : 4402460111111...252253254255
txDone	RxDone<2047> : 2050900111111...252253254255
Payload<2047> : 1150115011111...252253254255	RxDone<2047> : 2260142011111...252253254255
OnTxDone	RxDone<2047> : 1110168011111...252253254255
txDone	RxDone<2047> : 2170180111111...252253254255
Payload<2047> : 3201290111111...252253254255	RxDone<2047> : 2150111011111...252253254255
OnTxDone	RxDone<247> : 5012401111111...252253254255
txDone	RxDone<2047> : 20290111111...1111
Payload<2047> : 1990180011111...252253254255	RxDone<207> : 1101150111111...252253254255
OnTxDone	RxDone<2047> : 3201290111111...1111
txDone	

รูปที่ 4.8 การส่งข้อมูลจาก Sensor node ไป Gateway

การส่งข้อมูลโดยการสุ่มตัวเลข 1 ชุด ต่อการส่งข้อมูล 1 ครั้ง และส่งไปที่ปลายทาง Gateway ซึ่งที่ Gateway แสดงผลลัพธ์ว่าสามารถรับชุดตัวเลขนั้น ๆ มาได้ แม้ในบางครั้งจะมีการตกลงของข้อมูล ที่เกิดจากสัญญาณรบกวนในขณะนั้นด้วย

#### 4.3.2 ผลการทดสอบการส่งข้อมูลระหว่าง Sensor node และ Gateway โดยทำการเลือกช่องสัญญาณที่แตกต่างกัน

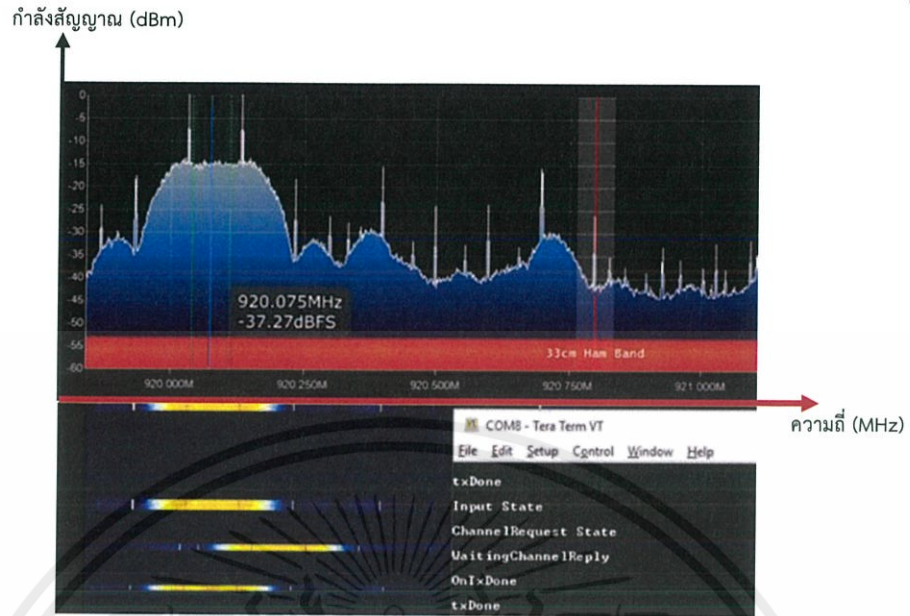
หลังจากที่ทำการทดสอบการส่งข้อมูลจาก Sensor node ไปยัง Gateway แล้วนั้น โดยมีการใช้ช่องสัญญาณความถี่เพียงช่องสัญญาณเดียวไปแล้ว จากนั้นก็ได้ทำการทดสอบการส่งข้อมูลโดยการเลือกใช้ช่องสัญญาณที่แตกต่างกันในเวลาเดียวกันด้วย ดังรูปที่ 4.9 แสดงการส่งข้อมูลในช่องสัญญาณที่แตกต่างกันภายในเวลาเดียวกัน เพื่อแสดงให้เห็นว่าไม่เกิดการรบกวนกันของช่องสัญญาณทั้งสอง ตามที่ได้กำหนดการแบ่งช่องสัญญาณให้มีการ์ดแบนด์ (Guard band) ขนาด 25 kHz



รูปที่ 4.9 สเปกตรัมของ Sensor node ที่อยู่คนละช่องสัญญาณ

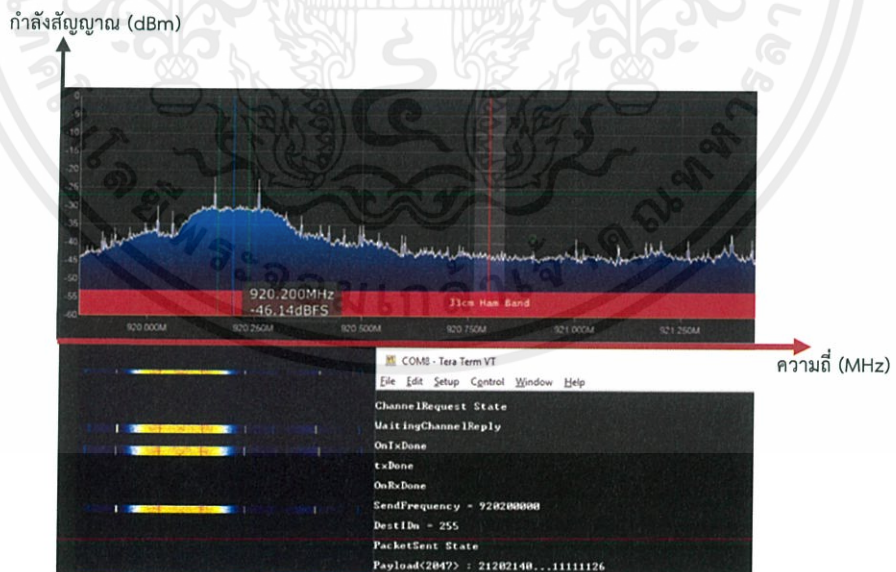
#### 4.3.3 ผลการทดสอบการสื่อสารกันระหว่าง Sensor node และ Gateway ว่าสามารถทำงานได้จริง

เป็นการทดสอบและเก็บผลการทดสอบที่ได้จากการสื่อสารกันระหว่าง Sensor node และ Gateway ในรูปแบบของการสลับช่องสัญญาณร่วมกับการใช้ Multi access ซึ่งได้ทำการทดสอบและเก็บผลการทดลองจำนวน 3 ช่องสัญญาณ ดังนี้



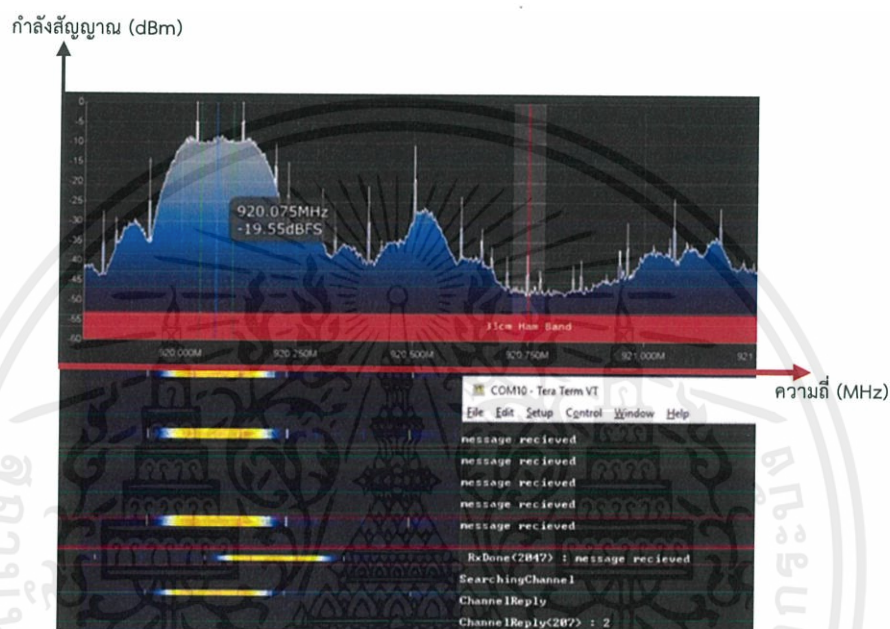
รูปที่ 4.10 สเปกตรัมข้อมูลที่ใช้ในการร้องขอและการตอบกลับของช่องสัญญาณ

จากรูปที่ 4.10 สเปกตรัมของข้อมูลที่ใช้ในการร้องขอและการตอบกลับของช่องสัญญาณ แสดงให้เห็นว่ามีการร้องขอช่องสัญญาณผ่านหน้าเทอร์มินอล และยังแสดงการส่งข้อมูลผ่านสเปกตรัม โดยข้อมูลการร้องขอช่องสัญญาณของ Sensor node และข้อมูลจากการตอบกลับของข้อมูลจะอยู่ในช่องสัญญาณเดียวกัน



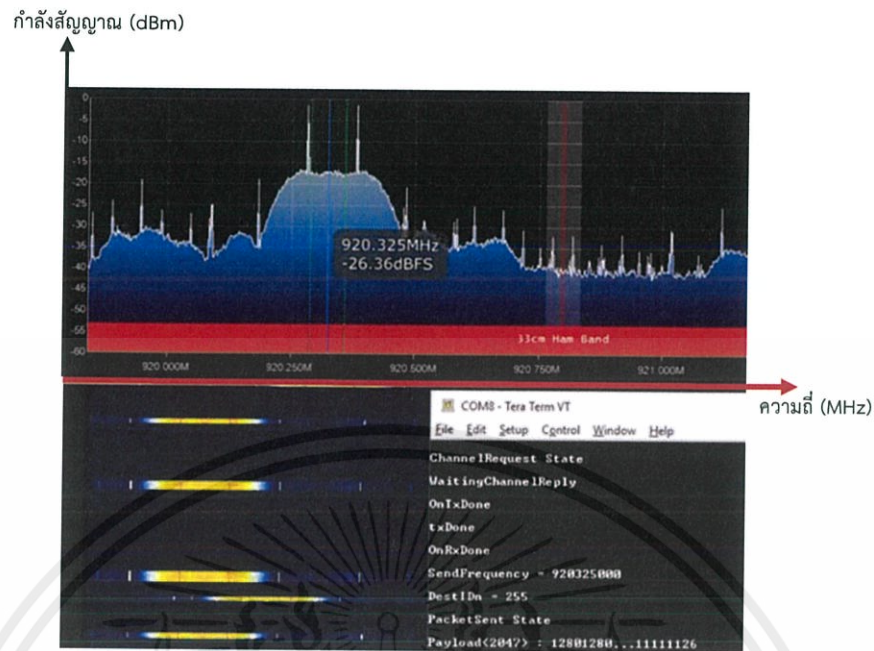
รูปที่ 4.11 สเปกตรัมการส่งข้อมูลของ Sensor node ไปยังช่องสัญญาณที่ 1

จากรูปที่ 4.11 สเปกตรัมการส่งข้อมูลของ Sensor node ไปยังช่องสัญญาณที่ 1 อธิบายได้ว่าเมื่อ Sensor node ได้รับข้อมูลการตอบกลับของช่องสัญญาณจาก Gateway ให้สามารถใช้ช่องสัญญาณที่ 1 ได้ หลังจากนั้น Sensor node จึงทำการสลับช่องสัญญาณไปยังช่องสัญญาณที่ 1 ซึ่งมีความถี่ 920.200 MHz และทำการส่งข้อมูลไปยังปลายทาง



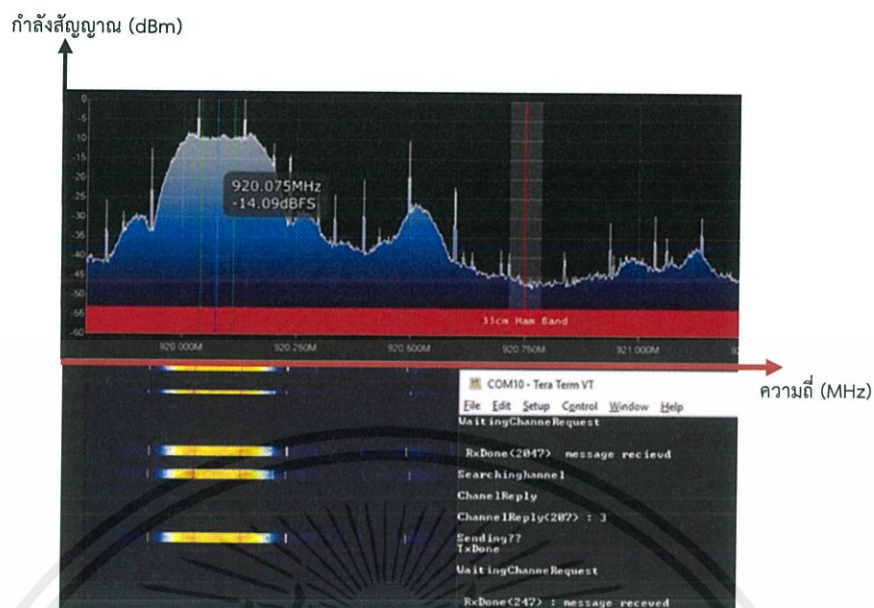
รูปที่ 4.12 สเปกตรัมการตอบกลับของ Gateway ไปให้ Sensor node ใช้ช่องสัญญาณที่ 2

จากรูปที่ 4.12 สเปกตรัมการตอบกลับของ Gateway ไปให้ Sensor node ใช้ช่องสัญญาณที่ 2 อธิบายได้ว่าเมื่อ Sensor node ส่งการร้องขอการใช้ช่องสัญญาณไปให้ Gateway แล้ว จากนั้น Gateway จึงทำการตอบกลับและอนุญาตให้ใช้ช่องสัญญาณที่ 2 ได้ที่ความถี่กลาง 920.075 MHz



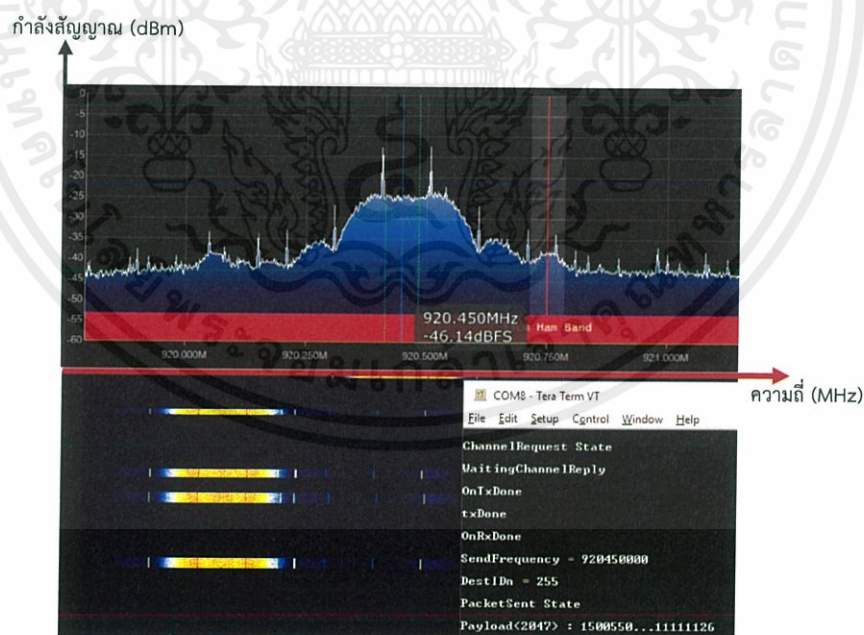
รูปที่ 4.13 สเปกตรัมการส่งข้อมูลของ Sensor node ไปยังช่องสัญญาณที่ 2

จากรูปที่ 4.13 สเปกตรัมการส่งข้อมูลของ Sensor node ไปยังช่องสัญญาณที่ 2 อธิบายได้ว่าเมื่อ Sensor node ได้รับข้อมูลการตอบกลับของช่องสัญญาณจาก Gateway ให้สามารถใช้ช่องสัญญาณที่ 2 ได้ หลังจากนั้น Sensor node จึงทำการสลับช่องสัญญาณไปยังช่องสัญญาณที่ 2 ซึ่งมีความถี่ 920.325 MHz และทำการส่งข้อมูลไปยังปลายทาง ซึ่งเห็นได้จากหน้าเทอร์มินอล



รูปที่ 4.14 สเปกตรัมการตอบกลับของ Gateway ไปให้ Sensor node ใช้ช่องสัญญาณที่ 3

จากรูปที่ 4.14 สเปกตรัมการตอบกลับของ Gateway ไปให้ Sensor node ใช้ช่องสัญญาณที่ 3 อธิบายได้ว่าเมื่อ Sensor node ส่งการร้องขอการใช้ช่องสัญญาณไปให้ Gateway แล้ว จากนั้น Gateway จึงทำการตอบกลับและอนุญาตให้ใช้ช่องสัญญาณที่ 3 ได้ที่ความถี่กลางที่ 920.075 MHz ซึ่งเห็นได้จากหน้าเทอร์มินอล



รูปที่ 4.15 สเปกตรัมการส่งข้อมูลของ Sensor node ไปยังช่องสัญญาณที่ 3

จากรูปที่ 4.15 สเปกตรัมการส่งข้อมูลของ Sensor node ไปยังช่องสัญญาณที่ 3 อธิบายได้ว่าเมื่อ Sensor node ได้รับข้อมูลการตอบกลับของช่องสัญญาณจาก Gateway ให้สามารถใช้ช่องสัญญาณที่ 3 ได้ หลังจากนั้น Sensor node จึงทำการสลับช่องสัญญาณไปยังช่องสัญญาณที่ 3 ซึ่งมีความถี่ 920.450 MHz และทำการส่งข้อมูลไปยังปลายทาง

หลังจากที่ผู้จัดทำได้ทดสอบและเก็บข้อมูลการสื่อสารกันระหว่าง Sensor node และ Gateway ผ่านการแสดงสเปกตรัมดังที่กล่าวมาแล้วข้างต้น จากนั้นผู้จัดทำจึงได้ทำการทดสอบและเก็บข้อมูลอีกครั้งจากการวัดกำลังสัญญาณที่ได้รับได้จาก Sensor node เปรียบเทียบกับระยะทาง โดยใช้ระยะทางทั้งหมด 200 และคำนวณทางทฤษฎีกำลังของสัญญาณที่รับได้เทียบกับระยะทาง ซึ่งมีการอ้างอิงการหาค่าความสูญเสียกำลังของสัญญาณที่รับได้จากสูตรการคำนวณ ดังนี้

$$FSPL = 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}\left(\frac{4\pi}{c}\right) - G_{Tx} - G_{Rx}$$

โดย  $d$  คือ ระยะทางระหว่างสายอากาศส่งและรับมีหน่วยเป็นเมตร ( $m$ )  
 $f$  คือ ความถี่มีหน่วยเป็นเมกะเฮิรตซ์ ( $MHz$ )  
 $G_{Tx}$  คือ กำลังขยายของสายอากาศฝั่งส่ง  
 $G_{Rx}$  คือ กำลังขยายของสายอากาศฝั่งรับ  
 $c$  คือ ความเร็วแสง

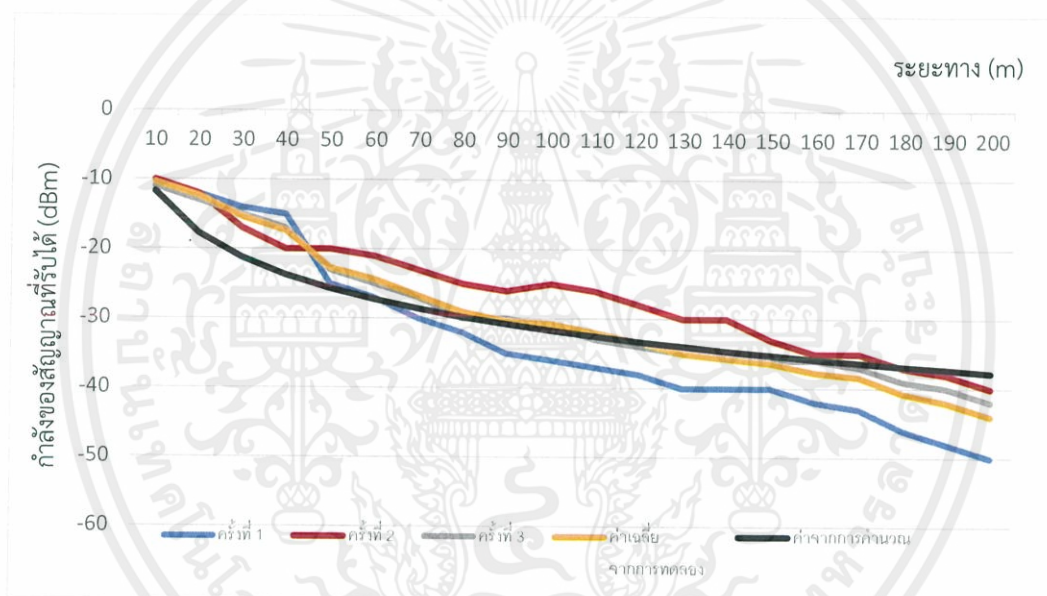
เมื่อได้ค่าความสูญเสียกำลังของสัญญาณที่รับได้แล้ว จึงนำไปหาค่ากำลังของสัญญาณที่รับได้จากการนำไปลบกับกำลังของสัญญาณที่ใช้ส่งที่มีค่าเป็น 20 dBm ดังแสดงตารางบันทึกผลกำลังสัญญาณที่รับได้เทียบกับระยะทางได้ตามตารางที่ 4.1

ตารางที่ 4.1 ตารางบันทึกผลกำลังสัญญาณที่รับได้เทียบกับระยะทาง

ระยะทาง (m)	กำลังสัญญาณที่รับได้ (dBm)			ค่าเฉลี่ย จากการทดลอง (dBm)	ค่าจากการคำนวณ (dBm)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
10	-10	-10	-11	-10.33	-11.72
20	-12	-12	-13	-12.33	-17.74
30	-14	-17	-15	-15.33	-21.26
40	-15	-20	-17	-17.33	-23.76
50	-25	-20	-23	-22.67	-25.69
60	-27	-21	-25	-24.33	-27.28
70	-30	-23	-27	-26.67	-28.62
80	-32	-25	-30	-29.00	-29.78
90	-35	-26	-30	-30.33	-30.80
100	-36	-25	-31	-30.67	-31.71
110	-37	-26	-33	-32.00	-32.54
120	-38	-28	-34	-33.33	-33.30
130	-40	-30	-35	-35.00	-33.99
140	-40	-30	-35	-35.67	-34.63
150	-40	-33	-36	-36.33	-35.24
160	-42	-35	-36	-37.67	-35.80
170	-43	-35	-37	-38.33	-36.32
180	-46	-37	-39	-40.67	-36.82
190	-48	-38	-40	-42.00	-37.29
200	-50	-40	-42	-44.00	-37.74

จากตารางการเก็บผลข้างต้น สามารถสรุปได้ว่ายิ่งระยะทางเพิ่มขึ้น ส่งผลให้กำลังที่รับได้จาก Sensor node มีค่าลดลง และจากการทดสอบและเก็บผลนี้อาจมีปัญหาที่มาจากสัญญาณรบกวน ทำให้กำลังสัญญาณที่รับได้ อาจมีความผิดพลาดหรือความไม่คงที่ของสัญญาณเมื่อเปรียบเทียบกับค่าที่เกิดจากการคำนวณทางทฤษฎี


จากนั้นผู้จัดทำจึงได้ทำการพล็อตกราฟของกำลังสัญญาณที่รับได้จาก Sensor node เทียบกับระยะทาง โดยเพิ่มระยะทางครั้งละ 10 เมตร ได้กราฟการทดสอบดังรูปที่ 4.16 สามารถอธิบายได้ว่า เมื่อระยะเพิ่มขึ้นค่ากำลังของสัญญาณที่รับได้จาก Sensor node จะมีค่าลดลง จากการเก็บผลการทดสอบทั้ง 3 ครั้ง และทำค่าเฉลี่ยของกำลังสัญญาณที่รับได้



รูปที่ 4.16 กำลังสัญญาณที่รับได้จาก Sensor node เทียบกับระยะทาง

จากนั้นทำการทดสอบการระบุตำแหน่งภายในอาคารโดยใช้พื้นที่โล่งภายในอาคาร ซึ่งมีการใช้ Sensor node สำหรับกระจายสัญญาณ จำนวน 4 ตัว และ Gateway สำหรับรับสัญญาณ จำนวน 1 ตัว โดยพื้นที่ทำการทดสอบเป็นรูปสี่เหลี่ยมผืนผ้าขนาด 8x7 เมตร และถูกแบ่งออกเป็นจุดพื้นที่อ้างอิงจำนวน 52 จุด ด้วยขนาด 1x1 เมตร ดังรูปที่ 4.17 แผนที่แสดงตำแหน่งติดตั้งอุปกรณ์และจุดอ้างอิง

	7	15	23	31	39	
1	8	16	24	32	40	47
2	9	17	25	33	41	48
3	10	18	26	34	44	49
4	11	19	27	35	43	50
5	12	20	28	36	44	51
6	13	21	29	37	15	52
	14	22	30	38	46	

 Sensor node

รูปที่ 4.17 แผนที่แสดงตำแหน่งติดตั้ง Sensor node และจุดอ้างอิง

เมื่อได้ทำการจำลองแผนที่แสดงตำแหน่งติดตั้ง Sensor node และจุดอ้างอิงแล้ว ผู้จัดทำจึงได้ทำการวัดความแรงของกำลังสัญญาณที่รับได้กับจุดอ้างอิง และบันทึกผลที่ได้ โดยผลความแรงของกำลังสัญญาณที่รับได้กับจุดอ้างอิง แสดงตามตารางที่ 4.3

ตารางที่ 4.2 ตารางบันทึกผลความแรงของกำลังสัญญาณที่รับได้กับจุดอ้างอิง

จุดอ้างอิง	ตำแหน่ง Sensor node			
	1	2	3	4
1	-16	-19	-24	-25
2	-18	-19	-23	-24
3	-20	-21	-21	-23
4	-21	-22	-20	-20
5	-22	-23	-19	-19
6	-23	-24	-17	-18
7	-15	-18	-25	-27
8	-17	-19	-24	-26
9	-18	-21	-23	-25
10	-19	-21	-21	-23
11	-20	-23	-18	-22
12	-21	-25	-18	-21
13	-24	-26	-17	-20
14	-26	-26	-16	-19
15	-15	-19	-24	-26
16	-15	-19	-23	-25
17	-15	-20	-21	-24
18	-16	-21	-19	-23
19	-18	-23	-19	-21
20	-19	-24	-18	-21
21	-20	-25	-17	-20
22	-25	-25	-16	-19
23	-16	-18	-25	-25

ตารางที่ 4.2 ตารางบันทึกผลความแรงของกำลังสัญญาณที่รับได้กับจุดอ้างอิง (ต่อ)

จุดอ้างอิง	ตำแหน่ง Sensor node			
	1	2	3	4
24	-18	-18	-24	-24
25	-19	-19	-22	-24
26	-20	-20	-20	-22
27	-21	-21	-19	-20
28	-23	-23	-18	-18
29	-24	-24	-17	-17
30	-26	-25	-17	-17
31	-16	-16	-27	-23
32	-16	-16	-25	-21
33	-17	-17	-24	-20
34	-18	-19	-22	-20
35	-19	-19	-19	-19
36	-22	-21	-19	-18
37	-24	-21	-18	-16
38	-27	-22	-17	-15
39	-17	-15	-25	-23
40	-18	-15	-23	-21
41	-20	-16	-23	-19
42	-19	-17	-19	-17
43	-23	-18	-20	-16
44	-24	-20	-18	-16
45	-25	-22	-17	-15
46	-26	-22	-17	-15
47	-19	-15	-26	-20
48	-20	-15	-25	-18

ตารางที่ 4.2 ตารางบันทึกผลความแรงของกำลังสัญญาณที่รับได้กับจุดอ้างอิง (ต่อ)

จุดอ้างอิง	ตำแหน่ง Sensor node			
	1	2	3	4
49	-21	-17	-23	-17
50	-24	-17	-21	-16
51	-25	-19	-20	-15
52	-27	-20	-19	-15

จากตารางการบันทึกผลข้างต้น สามารถสรุปได้ว่าเมื่อระยะทางเพิ่มขึ้น ส่งผลให้กำลังที่รับได้จาก Sensor node มีค่าลดลง ซึ่งจะส่งผลในเรื่องของการนำไปใช้ในการระบุตำแหน่งภายในอาคารต่อไป

จากนั้นทำการวัดระยะทางจาก Sensor node ในระยะต่าง ๆ ดังนี้ คือ 1, 2, 3, 4 และ 5 เมตร และทำการเฉลี่ยจากกำลังสัญญาณที่รับได้ในระยะทางต่าง ๆ ข้างต้น ได้การสรุปผลความแรงของกำลังสัญญาณเฉลี่ยที่รับได้กับจุดอ้างอิงเฉลี่ยตามตารางที่ 4.3

ตารางที่ 4.3 ตารางสรุปผลความแรงของกำลังสัญญาณเฉลี่ยที่รับได้กับจุดอ้างอิง

ระยะทาง (m)	Sensor node			
	1	2	3	4
1	-18.00	-18.67	-16.67	-15.00
2	-18.00	-19.14	-17.14	-15.43
3	-18.14	-19.43	-17.64	-16.36
4	-19.13	-19.52	-18.26	-17.39
5	-19.40	-19.69	-19.26	-18.68

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผล

ปริญญานิพนธ์นี้นำเสนอการพัฒนาต้นแบบเครือข่ายสื่อสารเพื่อการประยุกต์ใช้ในงานต่าง ๆ โดยการนำโมดูล Sx1276 ให้ทำหน้าที่ในการเป็น Sensor node และโมดูล S76S ทำหน้าที่เป็น Gateway ให้แก่ระบบ ในการสื่อสารระหว่างโมดูลทั้งสอง ได้ปรับการมอดูเลชันให้อยู่ในรูปแบบของ FSK ซึ่งเป็นการมอดูเลชันที่ทำให้การส่งข้อมูลมีความเร็วสูงที่สุดเท่าที่ตัวโมดูลจะสามารถทำได้ และได้ทำการเพิ่มขนาดของ Packet size ทำให้สามารถส่งข้อมูลได้ในปริมาณมากขึ้นในการส่งในแต่ละครั้ง โดยขนาดของ Packet size สามารถปรับได้ตามความต้องการหรือตามความเหมาะสมของงานที่จะนำไปประยุกต์ใช้ จากนั้นทำการออกแบบช่องสัญญาณขึ้นระหว่าง Sensor node และ Gateway โดยกำหนดช่วงความถี่ที่ใช้ในการสื่อสารไว้ที่ 920–925 MHz เพื่อไม่ให้ข้อมูลชนกันหรือรบกวนกันในกรณีที่มี Sensor node มากกว่า 1 ตัว รวมทั้งยังทำการออกแบบประเภทของ Packet ข้อมูล เพื่อที่จะให้ Sensor node และ Gateway สามารถเข้าใจได้ว่าควรจะทำสิ่งใดหลังจากได้รับ Packet นั้น สุดท้ายยังทำการออกแบบวิธีการเข้าใช้ช่องสัญญาณด้วยรูปแบบ Multi access เพื่อให้สามารถใช้งาน Sensor node หลาย ๆ ตัวพร้อมกันได้ ซึ่งในการทำงานส่วน Multi access สามารถแบ่งการทำงานออกเป็น 2 ส่วน คือส่วนการทำงานของ Sensor node และส่วนการทำงานของ Gateway ซึ่งอธิบายเพิ่มเติมได้ดังนี้

1) ส่วนการทำงานของ Sensor node คือส่วนที่มีการใช้ Sx1276 ซึ่งเริ่มต้นการทำงานด้วยการส่งข้อมูลการร้องขอช่องสัญญาณไปยัง Gateway หลังจากนั้นจะทำการรอรับการตอบกลับของ Gateway เพื่อใช้ในการสลับไปยังช่องสัญญาณที่ถูกกำหนดโดย Gateway และจะทำการส่งข้อมูลที่ต้องการส่งไปยังช่องสัญญาณนั้น ๆ ซึ่ง Sensor node แต่ละตัวจะมีเลขไอดีไม่ซ้ำกัน เพื่อที่จะใช้ในการระบุเอกลักษณ์เฉพาะตัวสำหรับการใช้งานในรูปแบบ Multi access

2) ส่วนการทำงานของ Gateway คือส่วนที่มีการใช้ S76S ซึ่งรอรับการร้องขอจาก Sensor node แต่ละตัว ซึ่งหลังจากที่ได้รับการร้องขอแล้ว จะทำการเลือกช่องสัญญาณที่สามารถใช้งานได้ขณะนั้น และจะทำการตอบกลับไปยัง Sensor node แต่ละตัว เพื่อให้ Sensor node นั้น ๆ ทำการสลับไปใช้ช่องสัญญาณที่ Gateway ได้กำหนดมาให้

จากที่กล่าวมาข้างต้น โครงการนี้จะนำไปประยุกต์ใช้งานร่วมกับเทคโนโลยีการระบุตำแหน่งภายในอาคาร หรือแอปพลิเคชันอื่น ๆ ที่สามารถนำไปประยุกต์ใช้ต่อได้ โดยมีการทำหน้าที่เป็นเครือข่ายในการรับส่งข้อมูลการระบุตำแหน่ง ซึ่งเมื่อเทียบกับเทคโนโลยีอื่น ๆ ระบบที่ได้ทำการพัฒนาขึ้นมานี้มีข้อดี ดังนี้

- 1) มีความหน่วงเวลา (Latency) ในการส่งข้อมูลต่ำ เนื่องจากใช้งานในย่านความถี่ Unlicensed band เมื่อเทียบกับระบบ Wi-Fi ที่ใช้ในย่านความถี่ 2.4 GHz
- 2) มีระยะในการใช้งานมากกว่าเทคโนโลยีอื่น ๆ โดยจากการทดสอบสามารถใช้งานได้ที่ระยะ 200 เมตร หรือมากกว่านั้นซึ่งขึ้นอยู่กับกำลังที่ใช้ในการส่งและกำลังรับกวนในช่วงเวลานั้น ๆ
- 3) มีการใช้พลังงานต่ำและอย่างมีประสิทธิภาพ เมื่อเทียบกับ Wi-Fi และ Ultra-wideband

## 5.2 ข้อเสนอแนะ

การพัฒนาต้นแบบเครือข่ายสื่อสารเพื่อการประยุกต์ใช้ ยังมีจำกัดในหลาย ๆ ด้าน เช่น ในส่วนของตัวโมดูลนั้นยังรองรับการมอดูเลชันได้หลากหลายแบบมาก โดยปริญญาณิพนธ์นี้ได้เลือกใช้การมอดูเลชันแบบ FSK ซึ่งมีผลทำให้ความเร็วในการส่งข้อมูลมีข้อจำกัด และความสามารถในการประมวลผลของโมดูล S76S ที่ทำหน้าที่เป็น Gateway นั้นยังไม่สามารถรองรับความแออัดจากการร้องขอช่องสัญญาณจาก Sensor node ที่มีจำนวนมาก ๆ ได้ ทำให้การตอบกลับการเลือกใช้ช่องสัญญาณมีความล่าช้าหรืออาจจะมีการตกลงของการตอบกลับช่องสัญญาณ ซึ่งอาจแก้ปัญหาได้โดยเพิ่มจำนวน Gateway ให้เพียงพอต่อจำนวนของ Sensor node หรืออาจจะทำการแก้ไขโปรโตคอลให้มีการใช้ช่วงเวลาต่าง ๆ ในการทำงานให้เหมาะสม และจากการทดสอบการใช้งานในระยะทางต่าง ๆ ซึ่งทางผู้จัดทำได้ทำการทดสอบวัดระยะทางเพียง 200 เมตร ทำให้ยังไม่สามารถทราบถึงระยะการใช้งานของโมดูลที่ไกลที่สุดรวมถึงกำลังที่ต่ำที่สุดที่โมดูลจะสามารถทำการสื่อสารกันได้

## บรรณานุกรม

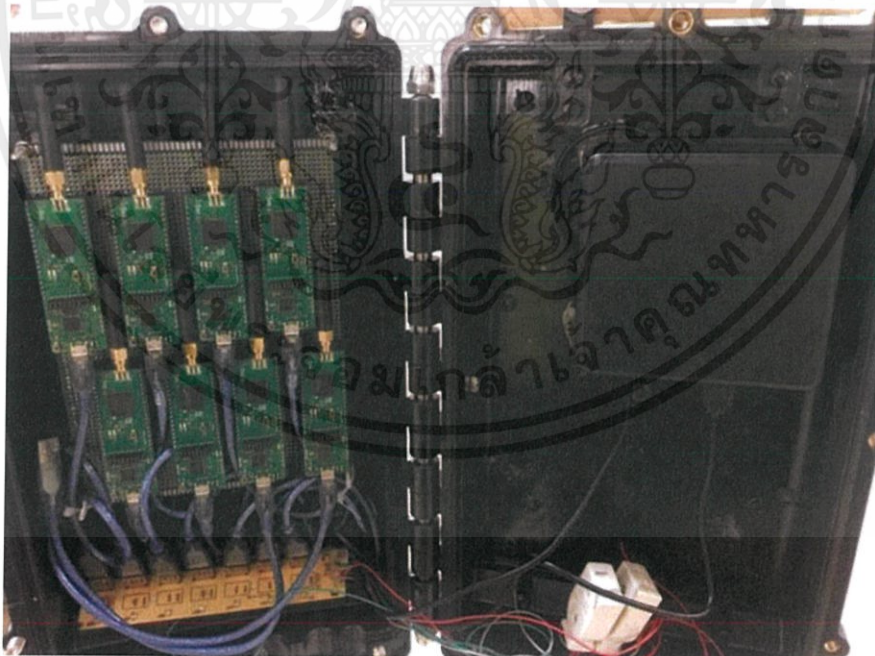
- [1] STMicroelectronics. “STM32L476xx. Ultra-low-power Arm® Cortex®-M4 32-bit MCU+FPU, 100DMIPS, up to 1MB Flash, 128 KB SRAM, USB OTG FS, LCD, ext. SMPS.”  
<https://www.st.com/resource/en/datasheet/stm32l476je.pdf>
- [2] Semtech Corporation. “WIRELESS & SENSING PRODUCTS SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver.”  
<https://www.mouser.com/ds/2/761/sx1276-944191.pdf>
- [3] สำนักบริหารคลื่นความถี่. สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และ กิจการโทรคมนาคมแห่งชาติ. “การใช้คลื่นความถี่ 920-925 เมกะเฮิรตซ์.”  
<https://nbt.go.th/getattachment/News/A8-920-925-MHz-Public-Hearing-18-8-2017.pdf.aspx>
- [4] elprocus. “FSK modulation.”  
<https://www.elprocus.com/fsk-modulation-demodulation-circuit-diagram/>
- [5] kritsada arjchariyaphat. “LoRA, LoRaWAN คืออะไร”  
<https://medium.com/deaware/lora-lorawan-%E0%B8%84%E0%B8%B7>
- [6] AcSiP Technology Corp. “S76S Product Brief.”  
<https://techship.com/downloads/acsip-s76s-product-brief-datasheet-v2/>
- [7] Senion Inc. “What is Indoor Positioning System.”  
<https://senion.com/indoor-positioning-system/#what>
- [8] infsoft GmbH. “Indoor Positioning, Tracking and Indoor Navigation with Wi-Fi.”  
<https://www.infsoft.com/technology/sensors/wifi>
- [9] infsoft GmbH. “Indoor Positioning, Tracking and Indoor Navigation with Beacons.”  
<https://www.infsoft.com/technology/sensors/bluetooth-low-energy-beacons>
- [10] infsoft GmbH. “Indoor Positioning with Ultra-wideband”  
<https://www.infsoft.com/technology/sensors/ultra-wideband>



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

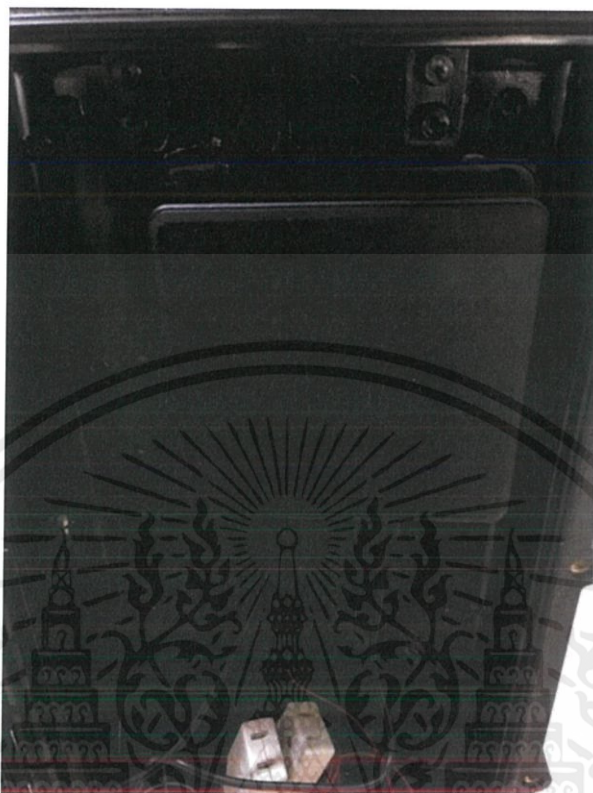


เสาและกลุ่มอุปกรณ์ Gateway ที่ทำการประกอบแล้ว



กลุ่มของอุปกรณ์ Gateway ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



อุปกรณ์ High Power Process

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข  
ภาษาที่ใช้ในโปรแกรมคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาษาที่ใช้ในโปรแกรมคำสั่งส่วน Sensor node

```

#include <string.h>
#include "hw.h"
#include "radio.h"
#include "timeServer.h"
#include "delay.h"
#include "low_power.h"
#include "vcom.h"

//define the frequency band
#if defined( USE_BAND_868 )
#define RF_FREQUENCY 868000000 // Hz
#elif defined( USE_BAND_915 )
#define RF_FREQUENCY 920075000 // Hz
#else
#error "Please define a frequency band in the compiler options."
#endif

//defined the radio variable
#define TX_OUTPUT_POWER 20 // dBm
#define( USE_MODEM_FSK )
#define FSK_FDEV 50e3 // Hz
#define FSK_DATARATE 100e3 // bps
#define FSK_BANDWIDTH 100e3 // Hz
#define FSK_AFC_BANDWIDTH 83.333e3 // Hz
#define FSK_PREAMBLE_LENGTH 5 // Same
for Tx and Rx
#define FSK_FIX_LENGTH_PAYLOAD_ON true
#else
#error "Please define a modem in the compiler options."
#endif

//defined the state of radio Rx and Tx
typedef enum
{
    LOWPOWER,
    RX,
    RX_TIMEOUT,
    RX_ERROR,
    TX,
    TX_TIMEOUT,
} States_t;

//defined the state of radio node
typedef enum
{
    InputState,
    ChannelRequest,
    WaitingChannelReply,
    PacketSent
} NodeState_t;

//defined the variable of packet
#define RX_TIMEOUT_VALUE 1000
#define BUFFER_SIZE 2047 // Define the
payload size here
#define LED_PERIOD_MS 200
#define Reply_Interval 3000
#define TXSite

```

```

uint16_t BufferSize = BUFFER_SIZE;
uint8_t RecievedBuffer[BUFFER_SIZE];
uint8_t SendBuffer[BUFFER_SIZE];
uint8_t Buffer[BUFFER_SIZE];
bool MessageRecieved,TimerInterupt;
States_t State = LOWPOWER;
NodeState_t NodeState = InputState;
int8_t RssiValue = 0;
int8_t SnrValue = 0;

static TimerEvent_t timerLed;
static TimerEvent_t timerReply;
/* Private function prototypes -----
---*/
/*!
 * Radio events function pointer
 */
static RadioEvents_t RadioEvents;
/*!
 * \brief Function to be executed on Radio Tx Done event
 */
void OnTxDone( void );
/*!
 * \brief Function to be executed on Radio Rx Done event
 */
void OnRxDone( uint8_t *payload, uint16_t size, int16_t rssi, int8_t snr );
/*!
 * \brief Function executed on Radio Tx Timeout event
 */
void OnTxTimeout( void );
/*!
 * \brief Function executed on Radio Rx Timeout event
 */
void OnRxTimeout( void );
/*!
 * \brief Function executed on Radio Rx Error event
 */
void OnRxError( void );
/*!
 * \brief Function executed on when led timer elapses
 */
static void OnledEvent( void );
static void ReplyEvent( void );
/**
 * Main application entry point.
 */
int main( void )
{
    uint32_t SendFrequency;
    uint16_t i;
    uint8_t SendChannel,SrcIDn,DestIDn;
                SrcIDn = 0x01;

    HAL_Init(); //HAL driver initial
    SystemClock_Config(); //System clock config
    DBG_Init(); //Debug initial
    HW_Init(); //Hardware initial

    /* Led Timers*/
    TimerInit(&timerLed, OnledEvent); //LED Timer

```

```

TimerSetValue( &timerLed, LED_PERIOD_MS);           //LED Timer set value
TimerStart(&timerLed );                             //LED Timer start

// Radio events initialization
RadioEvents.TxDone = OnTxDone;
RadioEvents.RxDone = OnRxDone;
RadioEvents.TxTimeout = OnTxTimeout;
RadioEvents.RxTimeout = OnRxTimeout;
RadioEvents.RxError = OnRxError;
Radio.Init( &RadioEvents );
Radio.SetChannel( RF_FREQUENCY );

//define modulation and config Rx and Tx variable
#if defined( USE_MODEM_FSK )
    Radio.SetTxConfig( MODEM_FSK, TX_OUTPUT_POWER, FSK_FDEV, 0,
                      FSK_DATARATE, 0,
                      FSK_PREAMBLE_LENGTH,
                      FSK_FIX_LENGTH_PAYLOAD_ON,
                      true, 0, 0, 0, 2000 );
    Radio.SetRxConfig( MODEM_FSK, FSK_BANDWIDTH, FSK_DATARATE,
                      0, FSK_AFC_BANDWIDTH,
                      FSK_PREAMBLE_LENGTH,
                      0, FSK_FIX_LENGTH_PAYLOAD_ON, BufferSize,
                      true,
                      0, 0, false, true );
#else
    #error "Please define a frequency band in the compiler options."
#endif
#if defined(RXSite)
    Radio.Rx( RX_TIMEOUT_VALUE );
#endif
    while(1)
    {
        switch (NodeState)
        {
//Input State Creating a random data packet
        case InputState :
            #define TXSite
                DelayMs(1000);
                PRINTF("Input State\n\r");
                random1=randr(1,0xffff);
                random2=randr(1,0xffff);
                Buffer[0] = 0x7E;
                Buffer[1] = 0x7E;
                Buffer[2] = 0xFF; //DestID
                Buffer[3] = 0x01; //SrcID
                Buffer[4] = 0x6A; //Type
                Buffer[5] = 0xFF; //RSSI
                Buffer[6] = 0xFF&random1;
                Buffer[7] = (0xFF00&random1)>>8;
                Buffer[8] = 0xFF&random2;
                Buffer[9] = (0xFF00&random2)>>8;
                Buffer[2045] = 0x7E;
                Buffer[2046] = 0x7E;

                for ( i=10 ; i<BufferSize-2 ; i++ )
                {
                    Buffer[i] = 1;
                }
                NodeState = ChannelRequest;

```

```

        break;

//Channel request state create a channel request packet and send

case ChannelRequest :
    #define TXSite
    PRINTF("ChannelRequest State\n\r");
    SendBuffer[0] = 0x7E;
    SendBuffer[1] = 0x7E;
    SendBuffer[2] = 0xFF;           //DestID
    SendBuffer[3] = 0x01;         //SrcID
    SendBuffer[4] = 0xFF;         //Type
    SendBuffer[5] = 0xFF;         //RSSI
    SendBuffer[2045] = 0x7E;
    SendBuffer[2046] = 0x7E;
    for ( i=5 ; i<BufferSize-2 ; i++ )
    {
        SendBuffer[i] = 0x41;
    }
    Radio.SetChannel( RF_FREQUENCY );
    State = LOWPOWER;
    DISABLE_IRQ( );
    if (State == LOWPOWER)
    {
        #ifndef LOW_POWER_DISABLE
            LowPower_Handler( );
        #endif
    }
    ENABLE_IRQ( );
    Radio.Send( SendBuffer, BufferSize );
    NodeState = WaitingChannelReply;
    break;

/*Waiting channel reply state wait for the channel reply packet and set
interrupt for reply interval and check the packet that received*/

case WaitingChannelReply :
    PRINTF("WaitingChannelReply\n\r");
    #define RXSite
    TimerInterrupt = false;
    TimerInit(&timerReply, ReplyEvent);
    TimerSetValue( &timerReply, Reply_Interval);
    TimerStart(&timerReply );
    ENABLE_IRQ( );
    while (NodeState == WaitingChannelReply)
    {
        switch (State)
        {
        case RX:
            Radio.Rx( RX_TIMEOUT_VALUE );
            State = LOWPOWER;

            break ;
        case TX:
        case TX_TIMEOUT :
        case RX_TIMEOUT:
        case RX_ERROR:
            Radio.Rx( RX_TIMEOUT_VALUE );
            State = LOWPOWER;
            break;
        case LOWPOWER :

```

```

default:
    break;
}
DISABLE_IRQ( );
if (State == LOWPOWER)
{
    #ifndef LOW_POWER_DISABLE
        LowPower_Handler( );
    #endif
}
ENABLE_IRQ( );
if (MessageRecieved == true)
{
if
((RecievedBuffer[0]==0x7E) && (RecievedBuffer[1]==0x7E) && (RecievedBuffer[2]==
0x01) && (RecievedBuffer[4]==0xFE))
{
    SendChannel = RecievedBuffer[6];
    SendFrequency = 920075000 + (125000*SendChannel);
    DestIDn = RecievedBuffer[3];
    PRINTF("SendFrequency = %d\n\r", SendFrequency);
    PRINTF("DestIDn = %d\n\r", RecievedBuffer[3]);
    Radio.SetChannel( SendFrequency );
    MessageRecieved = false;
    NodeState = PacketSent;
}
else
{
    NodeState = WaitingChannelReply;
    MessageRecieved = false;
}
}
if (TimerInterupt == true)
{
    NodeState = ChannelRequest;
}
}
break ;

//Sent the data packet on the channel that was assigned by control gateway
case PacketSent :
    PRINTF("PacketSent State\n\r");
    MessageRecieved = false;
    #define TXSite
    Buffer[2] = DestIDn;
    DelayMs (1000);
    State = LOWPOWER;
    DISABLE_IRQ( );
    if (State == LOWPOWER)
    {
        #ifndef LOW_POWER_DISABLE
            LowPower_Handler( );
        #endif
    }
    ENABLE_IRQ( );
    Radio.Send( Buffer, BufferSize );
    NodeState = InputState;
    PRINTF("Payload<%d> : ", BufferSize);
    for( i = 6; i < 10; i++ )

```

```

        {
            PRINTF("%d",Buffer[i]);
        }
        PRINTF("...");
        for( i = 2040; i < 2046; i++ )
        {
            PRINTF("%d",Buffer[i]);
        }
        PRINTF("\n\r");
        break ;
    }
}

void OnTxDone( void )
{
    Radio.Sleep( );
    #if defined(RXSite)
        State = TX;
    #endif
    PRINTF("OnTxDone\n\r");
}

void OnRxDone( uint8_t *payload, uint16_t size, int16_t rssi, int8_t snr )
{
    uint16_t i;
    Radio.Sleep( );
    BufferSize = size;
    memcpy( RecievedBuffer, payload, BufferSize );
    RssiValue = rssi;
    SnrValue = snr;
    #if defined(RXSite)
        State = RX;
    #endif
    MessageRecieved = true;
    PRINTF("OnRxDone\n\r");
}

void OnTxTimeout( void )
{
    Radio.Sleep( );
    #if defined(RXSite)
        State = TX_TIMEOUT;
    #endif
}

void OnRxTimeout( void )
{
    Radio.Sleep( );
    #if defined(RXSite)
        State = RX_TIMEOUT;
    #endif
}

void OnRxError( void )
{
    Radio.Sleep( );
    #if defined(RXSite)
        State = RX_ERROR;
    #endif
    #if defined(TXSite)
        State = LOWPOWER;
    #endif
    PRINTF("OnRxError\n\r");
}

    TimerStart(&timerLed );
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
static void ReplyEvent( void )
{
    if(NodeState == WaitingChannelReply)
    {
        PRINTF("Timer interrupt\n\r");
        TimerInterrupt = true;
    }
}

```

## ภาษาที่ใช้ในโปรแกรมคำสั่งส่วน Gateway

```

#include <string.h>
#include "board.h"
#include "radio.h"

#define LED_Toggle( x )      BSP_LED_Toggle( x );
#define LED_On( x )         BSP_LED_On( x );
#define LED_Off( x )        BSP_LED_Off( x );
#define BACKUP_PRIMASK()    uint32_t primask_bit= __get_PRIMASK()
#define DISABLE_IRQ()      __disable_irq()
#define ENABLE_IRQ()       __enable_irq()
#define RESTORE_PRIMASK()  __set_PRIMASK(primask_bit)

//define frequency and Tx power
#define RF_FREQUENCY        920075000 // Hz
#define TX_OUTPUT_POWER    20 // dBm

//define radio variable
#if defined( USE_MODEM_FSK )
#define FSK_FDEV            50e3 // Hz
#define FSK_DATARATE       100e3 // bps
#define FSK_BANDWIDTH      100e3 // Hz
#define FSK_AFC_BANDWIDTH  83.333e3 // Hz
#define FSK_PREAMBLE_LENGTH 5 // Same for Tx and Rx
#define FSK_FIX_LENGTH_PAYLOAD_ON true
#else
#error "Please define a modem in the compiler options."
#endif

//defined the state of radio Rx and Tx
typedef enum
{
    LOWPOWER,
    RX,
    RX_TIMEOUT,
    RX_ERROR,
    TX,
    TX_TIMEOUT,
} States_t;

//define the control gateway state
typedef enum
{
    WaitingChannelRequest,
    SearchingChannel,

```

```

    ChannelReply
} GatewayState_t;

#define RX_TIMEOUT_VALUE                2000//2000
#define BUFFER_SIZE                      2047//64 // Define the
payload size here

uint16_t BufferSize = BUFFER_SIZE;
uint8_t Buffer[BUFFER_SIZE];
uint8_t RecievedBuffer[BUFFER_SIZE];
uint8_t sendBuffer[BUFFER_SIZE];
States_t State = LOWPOWER;
GatewayState_t GatewayState = WaitingChannelRequest;
int16_t RssiValue = 0;
int8_t SnrValue = 0;
bool MessageRecieved;
/*!
 * Radio events function pointer
 */
static RadioEvents_t RadioEvents;
/*!
 * \brief Function to be executed on Radio Tx Done event
 */
void OnTxDone( void );
/*!
 * \brief Function to be executed on Radio Rx Done event
 */
void OnRxDone( uint8_t *payload, uint16_t size, int16_t rssi, int8_t snr );
/*!
 * \brief Function executed on Radio Tx Timeout event
 */
void OnTxTimeout( void );
/*!
 * \brief Function executed on Radio Rx Timeout event
 */
void OnRxTimeout( void );
/*!
 * \brief Function executed on Radio Rx Error event
 */
void OnRxError( void );
/**
 * Main application entry point.
 */
int main( void )
{
    uint8_t channelno, DestIDn, SrcIDn;
    uint16_t i;
    int32_t rand1, rand2;
    srand1( 10101 );
    channelno = 1 ;

    // Target board initialisation
    BoardInitMcu( );
    BoardInitPeriph( );
    UartPrint("\nS7678S SDK V1.2 (Multi Channel Gateway Control)\n");

#if POWER_SAVING_DEMO
    //Enter power-saving stop mode and then back to normal mode.
    UartPrint("Enter Power Saving mode for %ds ...\n",
POWER_SAVING_INTERVAL);
#endif

```

```

Demo_Enter_Stop_Mode(POWER_SAVING_INTERVAL, GPIOC);
UartPrint("Leave Power Saving mode ... \n", POWER_SAVING_INTERVAL);
#endif

// Radio initialization
RadioEvents.TxDone = OnTxDone;
RadioEvents.RxDone = OnRxDone;
RadioEvents.TxTimeout = OnTxTimeout;
RadioEvents.RxTimeout = OnRxTimeout;
RadioEvents.RxError = OnRxError;
Radio.Init( &RadioEvents ); //initial the radio event
Radio.SetChannel( RF_FREQUENCY );

//define modulation and config Rx and Tx variable
#if defined( USE_MODEM_FSK )
Radio.SetTxConfig( MODEM_FSK, TX_OUTPUT_POWER, FSK_FDEV, 0,
FSK_DATARATE, 0,
FSK_PREAMBLE_LENGTH, true,
true, 0, 0, 0, 3000 );//3000
Radio.SetRxConfig( MODEM_FSK, FSK_BANDWIDTH, FSK_DATARATE,
0, FSK_AFC_BANDWIDTH,
FSK_PREAMBLE_LENGTH,
0, true, BufferSize, true,
0, 0, false, true );
#else
#error "Please define a frequency band in the compiler options."
#endif

#if defined(RXSite)
Radio.Rx( RX_TIMEOUT_VALUE );
#endif

UartPrint("\n");
SrcIDn = 0xFF;
bool ChannelRequestRecieved;
ChannelRequestRecieved = false;
State = RX;
while(1)
{
switch (GatewayState)
{
//Gateway is waiting for channel request packet
case WaitingChannelRequest :
#define RXSite
UartPrint("WaitingChannelRequest \n\r");
MessageRecieved = false;
//Listen until received the packet
while (ChannelRequestRecieved == false)
{
switch (State)
{
case RX:
Radio.Rx( RX_TIMEOUT_VALUE );
State = LOWPOWER;
break ;
case TX:
case TX_TIMEOUT :
case RX_TIMEOUT:
case RX_ERROR:
Radio.Rx( RX_TIMEOUT_VALUE );

```

```

        State = RX;
        break;
    case LOWPOWER :
    default:
        break;
    }
    DISABLE_IRQ( );
    if (State == LOWPOWER)
    ENABLE_IRQ( );
    //Check the packet Header if it is a channel request packet move to
    thenext stage
    if(MessageRecieved == true)
    { UartPrint("message recieved \n\r");
      if
      ((RecievedBuffer[0]==0x7E)&&(RecievedBuffer[1]==0x7E)&&(Recieved
      Buffer[2]==0xFF)&&(RecievedBuffer[4]==0xFF))
      {
          MessageRecieved = true;
          DestIDn = RecievedBuffer[3];
          ChannelRequestRecieved = true;
          GatewayState = SearchingChannel;
      }
      else
      {
          ChannelRequestRecieved = false;
          GatewayState = WaitingChannelRequest;
      }
    }
    else
    {
        ChannelRequestRecieved = false;
        GatewayState = WaitingChannelRequest;
    }
}
break;
//Searching an empty channel for assign to sensornode
case SearchingChannel :
    UartPrint("SearchingChannel \n\r");
    ChannelRequestRecieved = false;
    MessageRecieved = false;
    if (channelno == 1)
    {
        channelno = 2;
    }
    else if (channelno == 2)
    {
        channelno = 3;
    }
    else if (channelno == 3)
    {
        channelno = 4;
    }
    else if (channelno == 4)
    {
        channelno = 5;
    }
    else if (channelno == 5)
    {
        channelno = 1;
    }
}

```

```

    }
    GatewayState = ChannelReply;
    break;

//Create a channel reply packet and send
case ChannelReply :
    UartPrint("ChannelReply \n\r");
    DelayMs(1000);
    #define TXSite
    State = TX;
    Buffer[0] = 0x7E;
    Buffer[1] = 0x7E;
    Buffer[2] = DestIDn;
    Buffer[3] = 0xFF; //SrcIDn
    Buffer[4] = 0xFE;
    Buffer[5] = 0xFF;
    Buffer[6] = channelno;
    Buffer[2045] = 0x7E;
    Buffer[2046] = 0x7E;
    for ( i=7 ; i<BufferSize-2 ; i++ )
    {
        Buffer[i] = 0x41;
    }
    UartPrint("ChannelReply<%d> : %d \n\r",BufferSize,Buffer[6]);
    DelayMs(1000);
    State = TX;
    Radio.Send( Buffer, BufferSize );
    UartPrint("Sending??\n\r");
    DelayMs(500);
    GatewayState = WaitingChannelRequest;
    break;
    }
}
void OnTxDone( void )
{
    Radio.Sleep( );
    #if defined(RXSite)
        State = TX;
    #endif
    UartPrint("TxDone \n\r",BufferSize);
}
void OnRxDone( uint8_t *payload, uint16_t size, int16_t rssi, int8_t snr )
{
    uint16_t i;
    Radio.Sleep( );
    BufferSize = size;
    memcpy( RecievedBuffer, payload, BufferSize );
    RssiValue = rssi;
    SnrValue = snr;
    UartPrint("\n RxDone(%d) : ",size);
    MessageRecieved = true;
    #if defined(RXSite)
        State = RX;
    #endif
}
void OnTxTimeout( void )
{
    Radio.Sleep( );
    #if defined(RXSite)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        State = TX_TIMEOUT;
    #endif
    UartPrint("TxTimeout\n");
}
void OnRxTimeout( void )
{
    Radio.Sleep( );
    #if defined(RXSite)
        State = RX_TIMEOUT;
    #endif
}
void OnRxError( void )
{
    Radio.Sleep( );
    #if defined(RXSite)
        State = RX_ERROR;
    #endif
    #if defined(TXSite)
        State = LOWPOWER;
    #endif
    UartPrint("RxError\n");
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## STM32L476xx

Ultra-low-power Arm® Cortex®-M4 32-bit MCU+FPU, 100DMIPS, up to 1MB Flash, 128 KB SRAM, USB OTG FS, LCD, ext. SMPS

Datasheet - production data

### Features

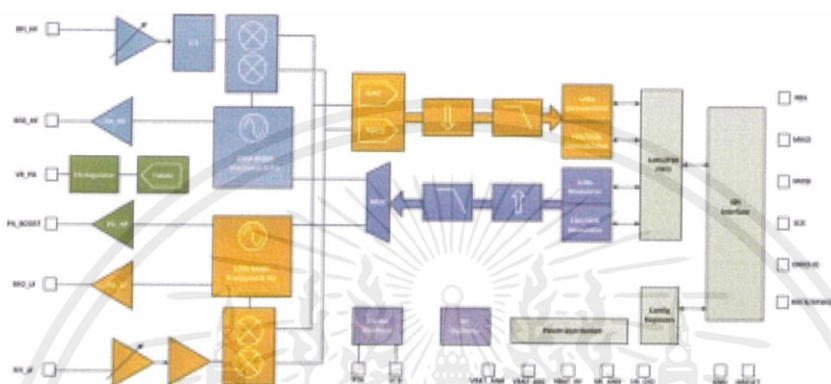
- Ultra-low-power with FlexPowerControl
    - 1.71 V to 3.6 V power supply
    - -40 °C to 85/105/125 °C temperature range
    - 300 nA in V<sub>BAT</sub> mode: supply for RTC and 32x32-bit backup registers
    - 30 nA Shutdown mode (5 wakeup pins)
    - 120 nA Standby mode (5 wakeup pins)
    - 420 nA Standby mode with RTC
    - 1.1 µA Stop 2 mode, 1.4 µA with RTC
    - 100 µA/MHz run mode (LDO Mode)
    - 39 µA/MHz run mode (@3.3 V SMPS Mode)
    - Batch acquisition mode (BAM)
    - 4 µs wakeup from Stop mode
    - Brown out reset (BOR)
    - Interconnect matrix
  - Core: Arm® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait-state execution from Flash memory, frequency up to 80 MHz, MPU, 100DMIPS and DSP instructions
  - Performance benchmark
    - 1.25 DMIPS/MHz (Drystone 2.1)
    - 273.55 CoreMark® (3.42 CoreMark/MHz @ 80 MHz)
  - Energy benchmark
    - 294 ULPMark™ CP score
    - 106 ULPMark™ PP score
  - Clock Sources
    - 4 to 48 MHz crystal oscillator
    - 32 kHz crystal oscillator for RTC (LSE)
    - Internal 16 MHz factory-trimmed RC (±1%)
    - Internal low-power 32 kHz RC (±5%)
    - Internal multispeed 100 kHz to 48 MHz oscillator, auto-trimmed by LSE (better than ±0.25 % accuracy)
    - 3 PLLs for system clock, USB, audio, ADC
- 
- Up to 114 fast I/Os, most 5 V-tolerant, up to 14 I/Os with independent supply down to 1.08 V
  - RTC with HW calendar, alarms and calibration
  - LCD 8×40 or 4×44 with step-up converter
  - Up to 24 capacitive sensing channels: support touchkey, linear and rotary touch sensors
  - 16x timers: 2x 16-bit advanced motor-control, 2x 32-bit and 5x 16-bit general purpose, 2x 16-bit basic, 2x low-power 16-bit timers (available in Stop mode), 2x watchdogs, SysTick timer
  - Memories
    - Up to 1 MB Flash, 2 banks read-while-write, proprietary code readout protection
    - Up to 128 KB of SRAM including 32 KB with hardware parity check
    - External memory interface for static memories supporting SRAM, PSRAM, NOR and NAND memories
    - Quad SPI memory interface
  - 4x digital filters for sigma delta modulator
  - Rich analog peripherals (independent supply)
    - 3x 12-bit ADC 5 Msps, up to 16-bit with hardware oversampling, 200 µA/Msps
    - 2x 12-bit DAC output channels, low-power sample and hold
    - 2x operational amplifiers with built-in PGA
    - 2x ultra-low-power comparators
  - 20x communication interfaces
    - USB OTG 2.0 full-speed, LPM and BCD
    - 2x SAs (serial audio interface)
    - 3x I2C FM+(1 Mbit/s), SMBus/PMBus
    - 5x USARTs (ISO 7816, LIN, IrDA, modem)
    - 1x LPUART (Stop 2 wake-up)

## STM32L476xx

- 3x SPIs (and 1x Quad SPI)
- CAN (2.0B Active) and SDMMC interface
- SWPMI single wire protocol master I/F
- IRTIM (Infrared interface)
- 14-channel DMA controller
- True random number generator
- CRC calculation unit, 96-bit unique ID
- Development support: serial wire debug (SWD), JTAG, Embedded Trace Macrocell™
- All packages are ECOPACK2® compliant

Table 1. Device summary

Reference	Part numbers
STM32L476xx	STM32L476RG, STM32L476JG, STM32L476MG, STM32L476ME, STM32L476VG, STM32L476QG, STM32L476ZG, STM32L476RE, STM32L476JE, STM32L476VE, STM32L476QE, STM32L476ZE, STM32L476RC, STM32L476VC


**SX1276/77/78/79**
**WIRELESS & SENSING PRODUCTS**
**DATASHEET**
**SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver**


### GENERAL DESCRIPTION

The SX1276/77/78/79 transceivers feature the LoRa™ long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption.

Using Semtech's patented LoRa™ modulation technique SX1276/77/78/79 can achieve a sensitivity of over -148dBm using a low cost crystal and bill of materials. The high sensitivity combined with the integrated +20 dBm power amplifier yields industry leading link budget making it optimal for any application requiring range or robustness. LoRa™ provides significant advantages in both blocking and selectivity over conventional modulation techniques, solving the traditional design compromise between range, interference immunity and energy consumption.

These devices also support high performance (G)FSK modes for systems including WMBus, IEEE802.15.4g. The SX1276/77/78/79 deliver exceptional phase noise, selectivity, receiver linearity and IIP3 for significantly lower current consumption than competing devices.

### ORDERING INFORMATION

Part Number	Delivery	MOQ / Multiple
SX1276IMLTRT	T&R	3000 pieces
SX1277IMLTRT	T&R	3000 pieces
SX1278IMLTRT	T&R	3000 pieces
SX1279IMLTRT	T&R	3000 pieces
SX1276WS	Wafer Form	1 Wafer (2000 dies)

- ◆ QFN 28 Package - Operating Range [-40;+85°C]
- ◆ Pb-free, Halogen free, RoHS/WEEE compliant product

### KEY PRODUCT FEATURES

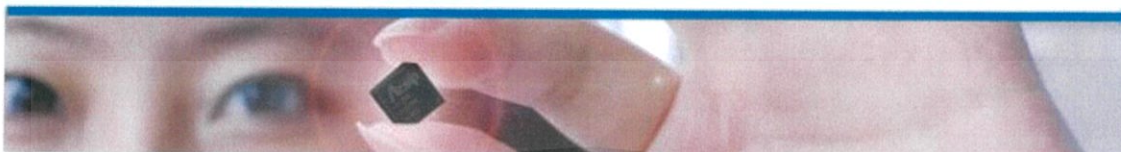
- ◆ LoRa™ Modem
- ◆ 168 dB maximum link budget
- ◆ +20 dBm - 100 mW constant RF output vs. V supply
- ◆ +14 dBm high efficiency PA
- ◆ Programmable bit rate up to 300 kbps
- ◆ High sensitivity: down to -148 dBm
- ◆ Bullet-proof front end: IIP3 = -11 dBm
- ◆ Excellent blocking immunity
- ◆ Low RX current of 9.9 mA, 200 nA register retention
- ◆ Fully integrated synthesizer with a resolution of 61 Hz
- ◆ FSK, GFSK, MSK, GMSK, LoRa™ and OOK modulation
- ◆ Built-in bit synchronizer for clock recovery
- ◆ Preamble detection
- ◆ 127 dB Dynamic Range RSSI
- ◆ Automatic RF Sense and CAD with ultra-fast AFC
- ◆ Packet engine up to 256 bytes with CRC
- ◆ Built-in temperature sensor and low battery indicator

### APPLICATIONS

- ◆ Automated Meter Reading.
- ◆ Home and Building Automation.
- ◆ Wireless Alarm and Security Systems.
- ◆ Industrial Monitoring and Control
- ◆ Long range Irrigation Systems

# S76S

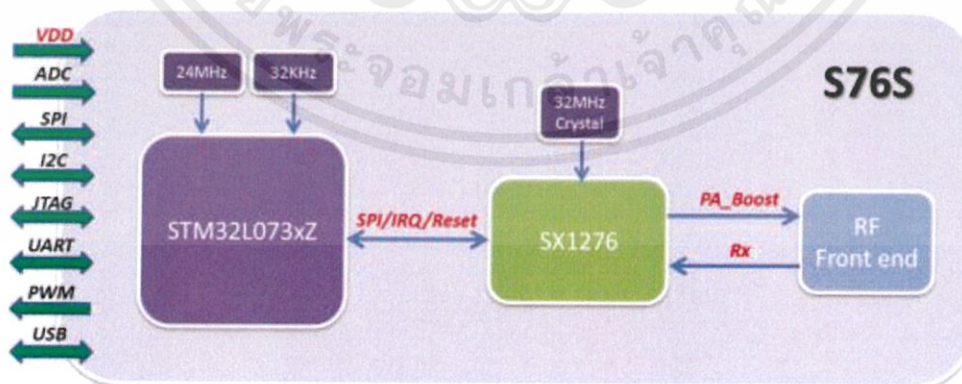
## Product Brief



### FEATURE

- Chipset: STM32L073xZ and SX1276
- High performance ARM® Cortex® -M0+ 32-bit RISC core operating at a 32 MHz frequency
- Embedded memories (up to 192 Kbytes of Flash memory and 20 Kbytes of RAM)
- LoRa Modem
- +20 dBm constant RF output vs. V supply
- Programmable bit rate up to 300 kbps
- High sensitivity: down to -137 dBm
- Excellent blocking immunity
- FSK, GFSK, MSK, GMSK, LoRa and OOK modulation
- Packet engine up to 256 bytes with CRC
- Built-in temperature sensor and low battery indicator
- Epoxy molding finished module in LGA type
- Small footprint : 13 mm x 11 mm x 1.1 mm

### S76S Block Diagram



AcSiP Technology Corp.  
群登科技股份有限公司

3F, -1 No 207 Fusing Rd, Taoyuan City Taoyuan Dist, Taoyuan City 33006, Taiwan(R.O.C)  
T +886 3 286-8368 F +886 3 347-5000

A wireless communication SiP Solution Provider

www.acsip.com.tw

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## S76S Product Brief

Technical Specification	
Model Name	S76S
Product Description	LoRa Wireless Communication Module
Package	LGA type
Host Interface	UART
Dimension	13 mm x 11 mm x 1.1 mm
Operation Conditions	
Operating Voltage	■ 3.3V
Temperature	■ Storage : -50°C ~ +105°C ■ Operating : -40°C ~ +85°C
Humidity	■ Operating : 10 ~ 95% (Non-Condensing) ■ Storage : 5 ~ 95% (Non-Condensing)
Electrical Specifications	
Frequency	EU868 / US915
Band Width	62.5~500KHz
Tx Power	+20dBm max.
Sensitivity	Down to -137dBm



AcSiP Technology Corp.  
群登科技股份有限公司

3F.-1 No 207, Fusing Rd., Taoyuan City, Taoyuan Dist., Taoyuan City 33056, Taiwan (R.O.C.)  
T +886 3 286-8388 F +886 3 347-5000

A wireless communication SiP Solution Provider

www.acsip.com.tw

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้