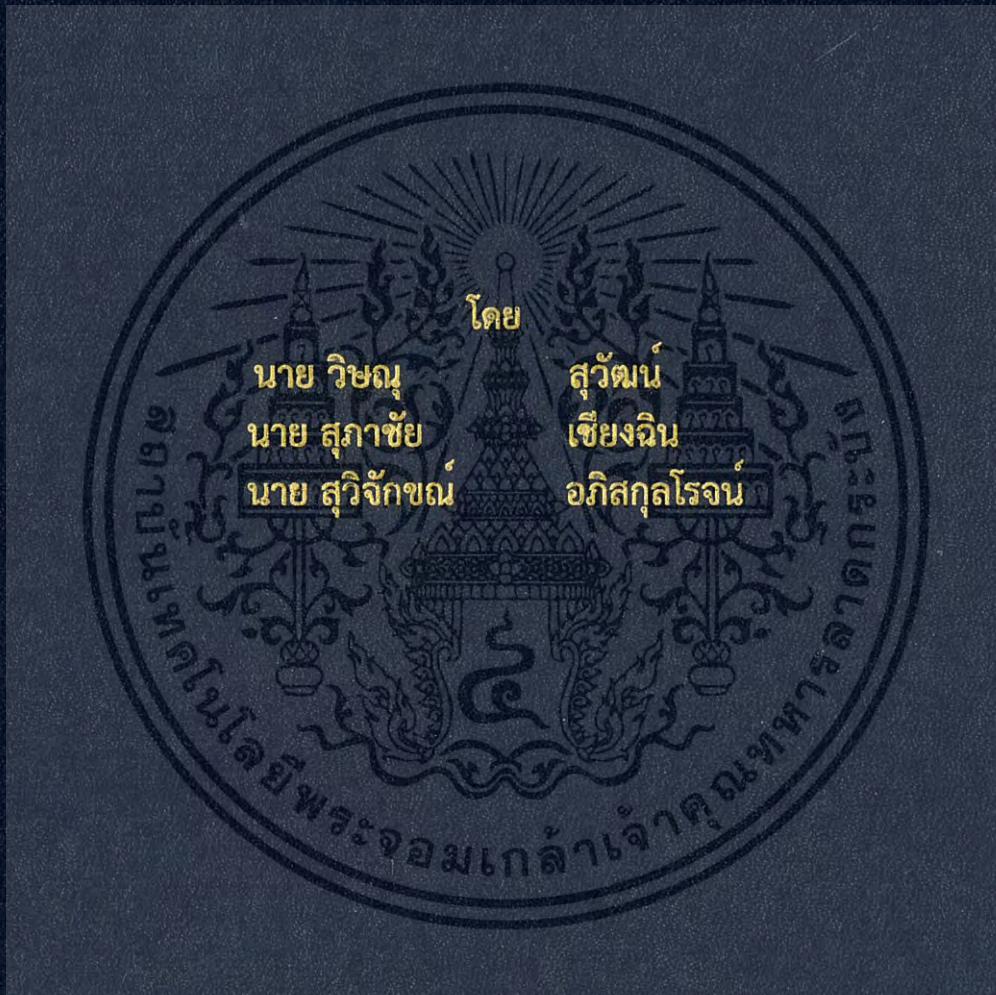


การสร้างแบบจำลองดาวเทียมขนาดเล็กสำหรับการแข่งขัน: ภาครับ
CanSat Design for Competition: Ground Station



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560


การสร้างแบบจำลองดาวเทียมขนาดเล็กสำหรับการแข่งขัน: ภาครับ
CanSat Design for Competition: Ground Station

โดย
นาย วิษณุ สุวัฒน์ 57011195
นาย สุภาชัย เชียงฉิน 57011412
นาย สุวิจักขณ์ อภิสกุลโรจน์ 57011429

อาจารย์ที่ปรึกษา
ดร.พีระเมศร์ โชติกวีกิจญาดา


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ผ่านการตรวจรูปเล่มแล้ว

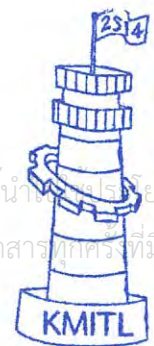
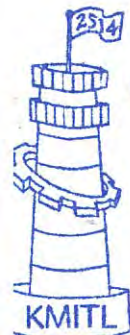

.....
อาจารย์ที่ปรึกษา
ไม่ว่ากรณีใดๆ ก็ตามอีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว


.....
กรรมการผู้ตรวจชิ้นงาน
21/5/61

วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การสร้างแบบจำลองดาวเทียมขนาดเล็กสำหรับการแข่งขัน: ภาครับ

CANSAT DESIGN FOR COMPETITION: GROUND STATION

ผู้จัดทำ

- | | | |
|-------------------|--------------|----------|
| 1. นาย วิษณุ | สุวัฒน์ | 57011195 |
| 2. นาย สุภาชัย | เชียงฉิน | 57011412 |
| 3. นาย สุวิจักขณ์ | อภิสกุลโรจน์ | 57011429 |

อาจารย์ที่ปรึกษา

(ดร.พีระเมศร์ โชติกวีกิจญาดา)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความอนุเคราะห์อย่างยิ่งจากท่านอาจารย์ที่ปรึกษา คือ ดร. พิระเมศร์ โชติทวีกิจญาติ ที่ให้คำแนะนำ คำสั่งสอน ให้ความรู้ ความเข้าใจ คอยดูแลเอาใจใส่ อีกทั้งยังคอยเป็นแรงผลักดันที่สำคัญตลอดระยะเวลาในการทำโครงการนี้ ขอขอบพระคุณท่านในความห่วงใยและความหวังดีที่ให้แก่มารดาเป็นอย่างยิ่ง

ขอขอบคุณ รศ.ดร. ชูวงศ์ พงศ์เจริญพาณิชย์ พี่ๆทุกคน และขอขอบคุณอาจารย์ ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้อบรมสั่งสอนและประสิทธิ์ประสาทวิชาความรู้ให้แก่ผู้จัดทำ

สุดท้ายนี้ขอกราบขอบพระคุณบิดามารดา ที่ให้ความสนับสนุนทางด้านทุนทรัพย์และที่สำคัญคือ คอยเป็นแรงผลักดัน เป็นกำลังใจในการเรียนและการทำโครงการนี้จนประสบความสำเร็จ

วิชณ สุวัฒน์
สุภาชัย เชียงฉิน
สุวิจักขณ์ อภิสกุลโรจน์
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างแบบจำลองดาวเทียมขนาดเล็กสำหรับใช้ในการ
แข่งขัน: ภาครีบ

CANSAT DESIGN FOR COMPETITION: GROUND
STATION

โดย	นาย วิษณุ	สุวัฒน์	57011195
	นาย สุภาชัย	เชียงฉิน	57011412
	นาย สุวิจักขณ์	อภิสกุลโรจน์	57011429

อาจารย์ที่ปรึกษา ดร. พิระเมศร์ โชติทวีกิจญาตา

บทคัดย่อ

ปฏิญานีพนธ์นี้ถูกจัดทำโดยมีจุดประสงค์เพื่อที่จะใช้ในการเข้าร่วมการแข่งขัน CANSAT ในปี 2018 ซึ่งจะกล่าวถึงวิธีการออกแบบและสร้างแบบจำลองของดาวเทียมขนาดเล็ก ในส่วนของสถานีภาคพื้นดินโดยมีขอบเขตในการรับผิดชอบเป็น 2 ส่วนหลักๆ ได้แก่ โปรแกรมที่ใช้ในการแสดงผลข้อมูล และ ระบบของสายอากาศที่ใช้ในการติดตามแบบจำลองดาวเทียมขนาดเล็ก โดยที่โปรแกรมที่ถูกออกแบบมานั้นสามารถที่จะแสดงผลข้อมูลที่เป็นตัวเลขที่ซับซ้อนให้อยู่ในรูปแบบของกราฟ, ตำแหน่งที่แสดงอยู่บน GOOGLE MAP , ภาพแสดงการเคลื่อนที่ของวัตถุ และยังสามารถบันทึกข้อมูลเก็บไว้ให้อยู่ในรูปแบบของ EXCEL ในส่วนของระบบของสายอากาศที่ใช้ในการติดตามแบบจำลองดาวเทียมขนาดเล็กนั้นยังสามารถแบ่งออกได้อีกเป็น 2 ส่วนคือ 1.สายอากาศซึ่งสายอากาศที่เราทำการออกแบบและสร้างนั้นอยู่ในย่านความถี่กลางที่ 2.45 GHz เป็นสายอากาศชนิด แผ่นพิมพ์โครงสร้างแบบ 2 ชั้น ที่มีคุณลักษณะที่ดีในด้านของน้ำหนักที่เบา รวมทั้งขนาดที่พกพาได้ง่ายและยังมีอัตราขยายที่เหมาะสมในการใช้งานอีกด้วย และส่วนที่ 2 คือระบบที่ใช้ในการติดตามดาวเทียมขนาดเล็กเป็นระบบที่ช่วยในการปรับทิศทางของสายอากาศให้ชี้ไปยังเป้าหมายโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

The purpose of this thesis is to develop a CanSat to attend the 2018 CanSat competition. Specifically, this thesis is about the design and implementation of a PC-based ground station for CANSAT. The scope of work of the ground station can be divided into two main parts: the graphical user interface and the system of antenna used for tracking the CanSat. The software for the ground station operation was designed to process complex numerical data and display them in the form of graph, as positions on google map, and as 3D motion as well as recording them in Excel format. The tracking antenna system can be divided into 2 parts: an antenna with suitable gain made of a stack of microstrip operating at 2.45 GHz which is compact and portable and a tracking platform that tracks the CanSat and move the antenna into the corresponding direction.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	X
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของปริิณยานิพนธ์	2
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	4
2.1 แบบจำลองดาวเทียมขนาดเล็กสำหรับการแข่งขัน (CANSAT)	4
2.1.1 ระบบการสื่อสารผ่านดาวเทียม (Satellite Communication System)	4
2.1.2 องค์ประกอบระบบสื่อสารดาวเทียม (Satellite System)	5
2.2 รายละเอียดการแข่งขัน Cansat Competition 2018	7
2.3 ทฤษฎีพื้นฐานและการประยุกต์ใช้งานสายอากาศ	8
2.3.1 หน้าที่และความหมายของสายอากาศ	8
2.3.2 ความถี่และความยาวคลื่น	8
2.3.3 คุณลักษณะและพารามิเตอร์ต่าง ๆ ของสายอากาศ	9
2.3.3.1 อัตราส่วนแรงดันคลื่นนิ่ง	9
2.3.3.2 การสูญเสียย้อนกลับ (Return Loss)	10
2.3.3.3 ประสิทธิภาพของสายอากาศ (Antenna Efficiency)	11
2.3.3.4 สภาพเจาะจงทิศทาง (Directivity)	11
2.3.3.5 อัตราขยายของสายอากาศ (Gain)	12
2.3.3.6 อิมพีแดนซ์ขาเข้า (Input Impedance)	13
2.3.3.7 แบนด์วิดท์ (Bandwidth)	14
2.4 หลักการการทำงานของสเต็ปเปอร์มอเตอร์	14
2.4.1 ข้อมูลและหลักการทํางานของสเต็ปเปอร์มอเตอร์	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.5 หลักการทำงานของจอยสติ๊กแบบแอนะล็อก	18
2.5.1 ทฤษฎีและหลักการทำงานของจอยสติ๊กแบบแอนะล็อก	18
2.6 ทฤษฎีและหลักการของ Zigbee	20
2.6.1 คุณสมบัติของ Zigbee	20
2.6.2 Zigbee Protocol Stack	20
2.6.3 Zigbee Protocol Stack: Physical Layer	21
2.6.4 Zigbee Protocol Stack: Mac Layer	21
2.6.4.1 Device Class	21
2.6.4.2 Topology	22
2.6.5 การส่งข้อมูลแบบ Serial ด้วย Zigbee Module	23
2.6.6 Zigbee Security	25
บทที่ 3 การออกแบบและการจัดทำปริญญานิพนธ์	26
3.1 การออกแบบ	27
3.1.1 การออกแบบและสร้างสายอากาศ	27
3.1.1.1 ข้อมูลเบื้องต้น (Introduction)	27
3.1.1.2 ผลการจำลองการทำงานสายอากาศ	28
3.1.2 การออกแบบระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก	31
3.1.2.1 การออกแบบโครงสร้างด้วยโปรแกรม Solidwork	31
3.1.2.2 การคำนวณและวิธีการเลือกอุปกรณ์	33
3.1.2.3 การคำนวณมุมในการหมุนของสเต็ปเปอร์มอเตอร์	36
3.1.2.4 การเชื่อมต่อระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก	39
3.1.3 การออกแบบโปรแกรมที่ใช้ในการแสดงผลข้อมูลด้วยโปรแกรม LabVIEW	43
3.1.3.1 รูปแบบข้อมูล	43
3.1.3.2 โพล์ชาร์ตการแยกข้อมูล	44
3.1.3.3 โปรแกรมออกแบบหน้าต่างแสดงผล	45
3.2 เครื่องมือที่ใช้ในการทดลอง	56
3.2.1 สายอากาศ Stack Microstrip	56

สารบัญ (ต่อ)

	หน้า
3.2.2 เครื่องสเปคตรัมอนาลิเซอร์(Spectrum Analyzer)	55
3.2.3 Stepper Motor NEMA 17	56
3.2.4 Stepper Motor NEMA 23	57
3.2.5 TB6600 Stepper Motor Driver	58
3.2.6 3.2” HVGA LCD TFT Non-Touch Screen, Arduino Mega-2560 Shield	59
3.2.7 Arduino Mega 2560	60
3.2.8 Xbee Pro 60mW U.FL Connection - Series 1	61
3.2.9 Xbee USB Dongle	62
3.2.10 ออสซิลโลสโคป	62
3.3 การจัดเก็บผลการทดลอง	63
3.3.1 การทดสอบการทำงานของสายอากาศ	63
3.3.2 การทดสอบการทำงานของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก	63
3.3.3 การทดสอบโปรแกรม	63
บทที่ 4 ผลการทดลอง	64
4.1 ผลการทดสอบการทำงานของสายอากาศ	64
4.1.1 เปรียบเทียบผลการจำลองด้วยโปรแกรมกับการสร้างจากแบบจำลอง	66
4.2 ผลการทดสอบการทำงานของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก	67
4.3 ผลการทดสอบการทำงานของโปรแกรม	68
บทที่ 5 สรุปผลและข้อเสนอแนะ	72
5.1 สรุปผล	72
5.2 ข้อเสนอแนะ	72
บรรณานุกรม	74
ภาคผนวก โค้ดของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก	77

สารบัญรูป

รูปที่		หน้า
1.1	แผนภาพการทำงานโดยรวมของแบบจำลองดาวเทียมขนาดเล็ก (CANSAT)	2
1.2	บล็อกไดอะแกรมการทำงานของภาครับ	3
2.1	ความสัมพันธ์ระหว่างความถี่และความยาวคลื่น	9
2.2	สแต็ปเปอร์มอเตอร์แบบ Unipolar	15
2.3	สแต็ปเปอร์มอเตอร์แบบ Bipolar	15
2.4	การควบคุมการหมุนของสแต็ปเปอร์มอเตอร์	18
2.5	หลักการภายในจอยสติ๊กแบบแอนะล็อก	19
2.6	การกำหนดค่าความต้านทานจากตำแหน่งของสติ๊ก	19
2.7	องค์ประกอบของ Zigbee Protocol Stack	20
2.8	รายละเอียดของ Physical Layer ตามมาตรฐาน IEEE 802.15.4	21
2.9	Topology ในการใช้งาน Zigbee Network	22
2.10	Uart Data Packet ของข้อมูล 0X1F (Decimal Number “31”)	23
2.11	รูปแบบของ AT Command	24
2.12	รูปแบบของ API Frame.	24
3.1	บล็อกไดอะแกรมภาพรวมการทำงาน	26
3.2	โครงสร้างสายอากาศ Stack Microstrip (ด้านหน้า-ด้านข้าง-ด้านหลัง) ผล	27
3.3	ของการปรับระยะห่างระหว่างแผ่นพิมพ์สองชั้น	28
3.4	ผลการปรับขนาดความกว้างพื้นที่ระนาบกราวด์ L_g บนแผ่น Ground Plane	29
3.5	ผลของการปรับระยะห่างระหว่างแผ่นพิมพ์สองชั้น	29
3.6	ผลการปรับค่า VSWR (Voltage Standing Wave Ratio)	30
3.7	ผลการจำลองแบบรูปการแพร่กระจายคลื่นในระนาบ YZ (YZ-PLANE)	31
3.8	บล็อกไดอะแกรมระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก	31
3.9	โครงสร้างของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก	32
3.10	อคูมิเนี่ยมโพรไฟล์	33

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.11 Stepper Motor NEMA 17	35
3.12 ประสิทธิภาพของ Stepper Motor NEMA 17 เมื่อใช้งานต่อเนื่องเทียบกันแต่ละแรงดันไฟฟ้า	35
3.13 Stepper Motor NEMA 23	36
3.14 ประสิทธิภาพของ Stepper Motor NEMA 23 เมื่อใช้งานต่อเนื่องเทียบกันแต่ละ แรงดันไฟฟ้า	36
3.15 การคำนวณค่าของมุมจากโปรแกรม Simulator	37
3.16 หลักการในการหามุมในแนวแกนตั้ง	38
3.17 อุปกรณ์ HVGA LCD TFT Non-Touch Screen, Arduino Mega-2560 Shield	39
3.18 TB6600 Stepper Motor Driver	40
3.19 จอยสติ๊กแบบแอนะล็อก	40
3.20 การเชื่อมต่อสายของจอยสติ๊กแบบแอนะล็อกกับอุปกรณ์อื่น	41
3.21 โพล์ชาร์ตการทำงานของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก	42
3.22 ความสามารถของโปรแกรม LabVIEW	43
3.23 รูปแบบของข้อมูลที่ทำการรับจากแบบจำลองดาวเทียมขนาดเล็ก	43
3.24 โพล์ชาร์ตการแยกข้อมูล	44
3.25 ส่วนประกอบต่างๆของหน้าจอแสดงผล	45
3.26 โปรแกรมที่ทำการออกแบบ	47
3.27 โปรแกรมที่ทำงานในส่วนของการรับข้อมูล	47
3.28 โปรแกรมที่ทำงานในส่วนของการตาราง	48
3.29 โปรแกรมที่ทำงานในส่วนของการแยกข้อมูลและแสดงผลกราฟ	48
3.30 โปรแกรมที่ทำงานในส่วนของการแสดงผล GPS	49
3.31 Visa Resource Name	49
3.32 Visa Configure Serial Port	50
3.33 Visa Flush I/O Buffer	50
3.34 Table	50
3.35 Array Size	50
3.36 Index Array	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.37 Wait (ms)	51
3.38 Read Buffer	51
3.39 Case Structure	51
3.40 Fract/Exp String To Number	52
3.41 Match Pattern	52
3.42 Build Array	52
3.43 Get Date/Time In Seconds	53
3.44 Get Date/Time String	53
3.45 Build Waveform	53
3.46 Insert Into Array	54
3.47 Waveform Chart	54
3.48 Number To Decimal String	54
3.49 Invoke Node	54
3.50 Close Reference	55
3.51 Visa Close	55
3.52 While Loop	55
3.53 สายอากาศ Stack Microstrip	56
3.54 เครื่องสเปคตรัมอนาล็อกเซอร์	56
3.55 ประสิทธิภาพของ Stepper Motor NEMA 17 เมื่อใช้งานต่อเนื่องเทียบ กันแต่ละแรงดันไฟฟ้า	57
3.56 ประสิทธิภาพของ Stepper Motor NEMA 23 เมื่อใช้งานต่อเนื่องเทียบ กันแต่ละแรงดันไฟฟ้า	58
3.57 TB6600 Stepper Motor Driver	58
3.58 อุปกรณ์ HVGA LCD TFT Non-Touch Screen, Arduino Mega-2560 Shield	59
3.59 บอร์ด Arduino Mega 2560	60
3.60 Xbee Pro 60mW U.FL Connection - Series 1 (802.15.4)	61
3.61 Xbee USB Dongle	62
3.62 ออสซิลโลสโคป	6

สารบัญตาราง

ตารางที่	หน้า
2.1	7
2.2	9
3.1	30
3.2	32
3.3	57
3.4	57
3.5	59
3.6	59
3.7	61
3.8	62
4.1	67
4.2	71

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันนั้นเทคโนโลยีทางด้านดาวเทียมเข้ามามีบทบาทกับชีวิตประจำวันของเราเป็นอย่างมากไม่ว่าจะเป็น การสื่อสาร, สื่อโทรทัศน์, การทหาร, กรมอุตุนิยมวิทยาและการสำรวจทรัพยากรธรรมชาติ ทั้งหมดล้วนแล้วเป็นสิ่งจำเป็น แต่เนื่องจากขนาดที่ใหญ่ทำให้ระบบมีความซับซ้อนรวมถึงต้นทุนที่สูงทำให้เป็นเรื่องยากที่จะนำอุปกรณ์ดาวเทียมมาศึกษากันว่า จึงได้มีการสร้างแบบจำลองของดาวเทียมขึ้นมาเพื่อเป็นการลดต้นทุนในการสร้างดาวเทียมขนาดใหญ่ขึ้นและง่ายต่อการนำไปศึกษาหาความรู้อีกด้วย ซึ่งเราจะใช้ชื่อในการเรียกอุปกรณ์นี้ว่า แบบจำลองดาวเทียมขนาดเล็ก (CanSat)

แบบจำลองดาวเทียมขนาดเล็ก (CanSat) หรือถูกเรียกอีกในชื่อหนึ่งว่า ดาวเทียมกระป๋อง เป็นเทคโนโลยีอย่างหนึ่งที่ถูกสร้างขึ้นเพื่อใช้ในการจำลองการทำงานของดาวเทียมขนาดใหญ่ ทำให้มีขนาดเล็กและราคาถูกกว่าดาวเทียมขนาดใหญ่เป็นอย่างมากแต่เนื่องจากขนาดที่เล็กทำให้ความสูงที่ทำการปล่อยออกไปจะมีความสูงไม่เกิน 4 กิโลเมตรจากระดับพื้นดินซึ่งไม่ได้ถูกนำไปปล่อยเข้าสู่วงโคจรในอวกาศจริงๆ ภายในของดาวเทียมกระป๋องจะถูกติดตั้งอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ทำให้ดาวเทียมกระป๋องนั้นสามารถนำมาใช้ในการตรวจวัดค่าของสภาพอากาศไม่ว่าจะเป็น อุณหภูมิ, ความดันและความเร็วของลม และยังมีระบบที่ทำให้เราทราบตำแหน่งของดาวเทียมกระป๋องได้อีกด้วย

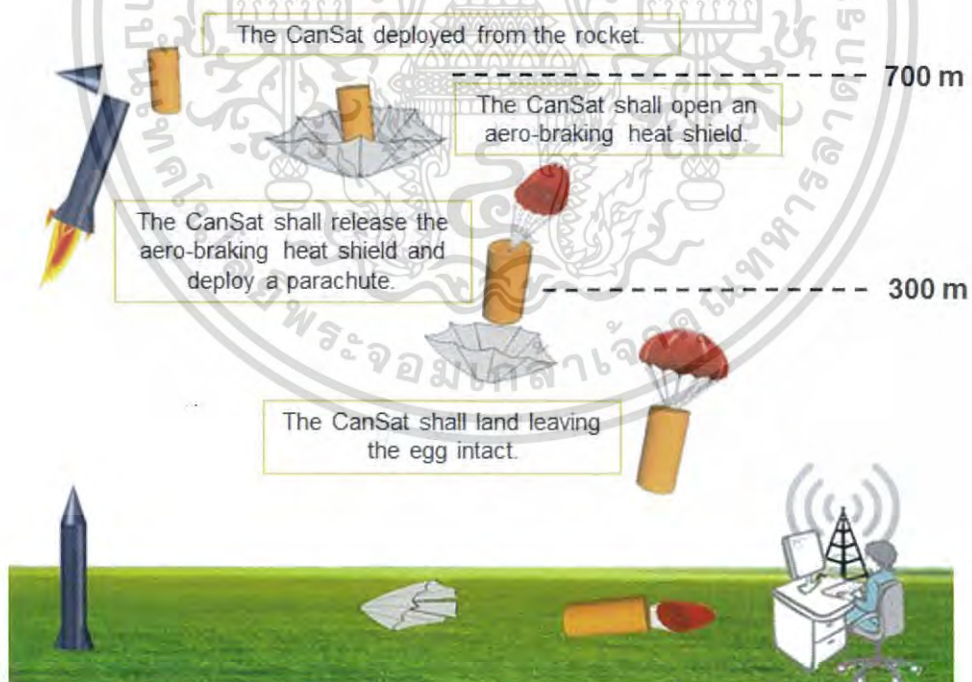
วัตถุประสงค์หลักของปริญญานิพนธ์นี้ถูกจัดทำขึ้นเพื่อใช้ในการแข่งขันในปี 2018 และจะมีการจัดขึ้นทุกๆปี ซึ่งการแข่งขันถูกจัดทำขึ้นสำหรับให้ผู้ที่ความสนใจทางด้านของดาวเทียมได้ทำการออกแบบและสร้างแบบจำลองดาวเทียมขึ้นมาแข่งขันกัน โดยจะมีกฎกติกาและภารกิจ แตกต่างกันไปในแต่ละปีทำให้มีความท้าทายมากยิ่งขึ้น สำหรับเป้าหมายของการแข่งขันนั้นเป็นการเริ่มต้นจากการสร้างแบบจำลองดาวเทียมขึ้นมาเพื่อก้าวสำคัญที่สามารถนำความรู้ที่มีไปต่อยอดได้ในอนาคตซึ่งเป็นพื้นฐานในการสร้างเทคโนโลยีที่สูงขึ้นเพื่อปล่อยสู่อวกาศได้จริง

1.2 วัตถุประสงค์

- 1) ออกแบบและสร้างสถานีภาครับของแบบจำลองดาวเทียมขนาดเล็กตามภารกิจการแข่งขันประจำปี 2018
- 2) ออกแบบและสร้างสายอากาศที่สามารถใช้ในการรับข้อมูลจากแบบจำลองดาวเทียมขนาดเล็ก
- 3) ออกแบบระบบของสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก

1.3 ขอบเขตของปริญญาณิพนธ์

- 1) โปรแกรมของภาครับสามารถรับข้อมูลจากแบบจำลองดาวเทียมขนาดเล็กตรงตามกฎการแข่งขันที่ได้กำหนดไว้
- 2) สายอากาศรูปแบบ Stack microstrip สามารถใช้งานได้จริงในย่านความถี่ 2.4 GHz โดยมีอัตราขยายประมาณ 6 dB ขึ้นไป
- 3) โครงสร้างของระบบของสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็กสามารถรับน้ำหนักของสายอากาศได้

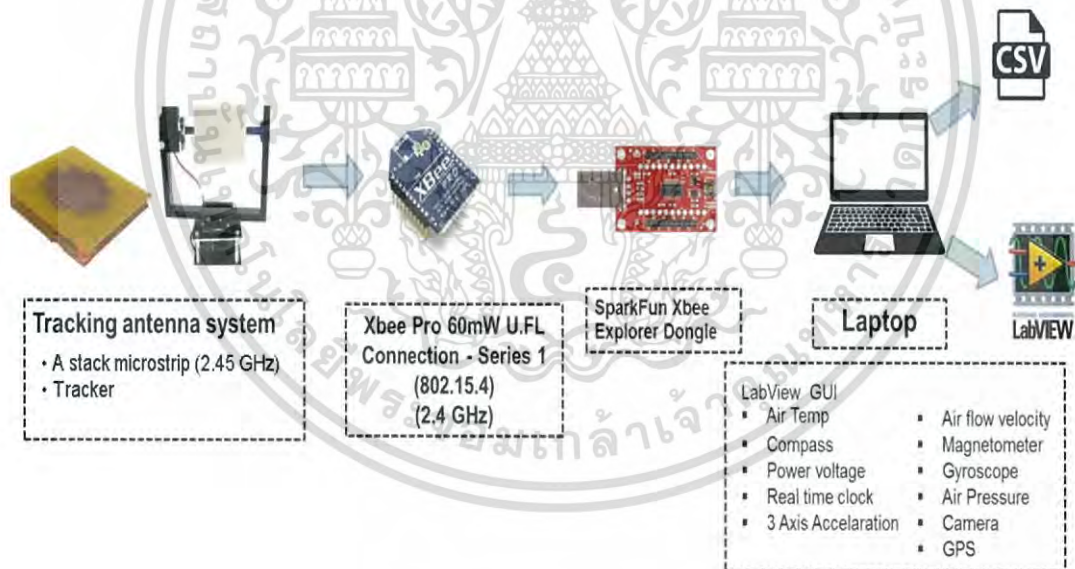


รูปที่ 1.1 แผนภาพการทำงานโดยรวมของแบบจำลองดาวเทียมขนาดเล็ก (CanSat)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 1.1 เป็นแผนภาพการทำงานโดยรวมของขั้นตอนการปล่อยแบบจำลองดาวเทียมขนาดเล็กตามกติกาการแข่งขันในปี 2018 โดยเริ่มจากการปล่อยจรวดออกสู่ชั้นบรรยากาศจากนั้นทำการปล่อยแบบจำลองของดาวเทียมขนาดเล็กออกจากจรวด ในขณะที่ปล่อยนั้นแบบจำลองของดาวเทียมขนาดเล็กจะเริ่มทำการส่งข้อมูลสภาพอากาศ ณ ขณะนั้นมายังสถานีภาคพื้นดินหรือภาครับ โดยในส่วนของภาครับก็จะนำข้อมูลที่ถูกส่งมานั้นมาประมวลผลผ่านโปรแกรมที่ได้ทำการออกแบบไว้และแสดงผลออกมาผ่านจอมอนิเตอร์ ซึ่งข้อมูลที่ส่งมาจากแบบจำลองของดาวเทียมขนาดเล็กจะถูกส่งมาอย่างต่อเนื่องจนกระทั่งแบบจำลองดาวเทียมขนาดเล็กตกลงสู่พื้นดิน

โดยปรัชญาในพินิจนี้จัดทำในส่วนของภาครับซึ่งจะมีขั้นตอนการทำงานเป็นดังนี้ รับข้อมูลมาจากแบบจำลองดาวเทียมขนาดเล็กผ่านสายอากาศ Stack microstrip โดยใช้โมดูล Xbee pro series 1 ซึ่งเป็นอุปกรณ์ที่ใช้ในการสื่อสารระหว่างแบบจำลองขนาดเล็กกับภาครับ จากนั้นจะทำการส่งค่าที่ได้ออกมาประมวลผลผ่านโปรแกรม LabVIEW และนำมาแสดงผลผ่านจอมอนิเตอร์ ซึ่งรูปแบบที่แสดงออกมานั้นจะขึ้นอยู่กับกฎระเบียบข้อบังคับของการแข่งขัน โดยจะมีบล็อกไดอะแกรมดังรูปที่ 1.2



รูปที่ 1.2 บล็อกไดอะแกรมการทำงานของภาครับ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

CanSat [1] คือดาวเทียมขนาดเล็กเท่าขนาดกระป๋องเครื่องดื่ม ที่จำลองการทำงานเหมือนดาวเทียมจริง โดยติดตั้งอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ไว้ภายในกระป๋องเครื่องดื่มหรือวัสดุรูปทรงขนาดเท่ากระป๋องเครื่องดื่ม

การปล่อย CanSat จะใช้จรวดยิงขึ้นไปบนท้องฟ้า หรือใช้ Drone หรือ บอลลูนอัดแก๊สปล่อยลงมาจากที่สูง ในระดับหลายร้อยเมตร ตัว CanSat เมื่อถูกปล่อยออกมาจากอุปกรณ์ขนส่ง ก็จะกางร่มชูชีพโดยอัตโนมัติ และระบบต่างๆ ภายในอุปกรณ์ CanSat ก็จะเริ่มทำงาน ซึ่งได้แก่ บันทึกภาพถ่ายหรือวิดีโอ ระบุตำแหน่งพิกัดของตัว CanSat วัดอุณหภูมิและความดันอากาศ สามารถรับส่งข้อมูลได้ทันทีผ่านอุปกรณ์ไร้สายส่งสู่อุปกรณ์รับสัญญาณภาคพื้นดิน

สถานีภาคพื้นดินหรือสถานีภาครับคือส่วนที่ทำหน้าที่ในการรับสัญญาณจากอุปกรณ์ CanSat แล้วจากนั้นจะทำการประมวลผลข้อมูลที่รับจากตัวอุปกรณ์ CanSat เนื่องจากข้อมูลของทาง CanSat ที่ถูกส่งมานั้นมีความซับซ้อนยากที่จะอ่านข้อมูลเพื่อให้สะดวกในการอ่านข้อมูล จึงได้นำโปรแกรมเข้ามาช่วยในสถานีภาคพื้นดิน โดยอุปกรณ์ในส่วนภาคพื้นดินจะประกอบไปด้วย โมดูลที่ใช้ในการรับข้อมูล, สายอากาศ, ระบบสายอากาศสำหรับติดตาม CanSat ,และ โน้ตบุคที่ใช้ในการแสดงโปรแกรม

การสื่อสารผ่านดาวเทียมนับว่าเป็นวิธีการส่งข้อมูลข่าวสารในปัจจุบันที่นิยมกันมาก ซึ่งอำนวยความสะดวกในการใช้งานได้อย่างกว้างขวาง และรวดเร็วเป็นการสื่อสารที่วิวัฒนาการมาจากการสื่อสารแบบไมโครเวฟ ในช่วงแรกๆ ดาวเทียมได้ถูกประดิษฐ์ขึ้นมาเพื่อใช้ในการด้านการทหารและพัฒนามาใช้ทางด้านการพยากรณ์อากาศ การค้นหาทรัพยากรธรณี และการสื่อสารคือ ดาวเทียมสื่อสาร ที่ใช้ในวงการระบบโทรทัศน์ ดาวเทียมสื่อสารที่ถูกส่งไปครั้งแรกในปี พ.ศ. 2508 โดยองค์การโทรคมนาคมระหว่างประเทศ (International Telecommunications Satellite Organization) หรือเรียกย่อว่า INTELSAT

2.1 แบบจำลองดาวเทียมขนาดเล็กสำหรับการแข่งขัน (CANSAT)

2.1.1 ระบบการสื่อสารผ่านดาวเทียม (Satellite Communication System)[2]

ดาวเทียมสื่อสารมีต้นกำเนิดมาจากความคิดของนักเขียนนิยายวิทยาศาสตร์ชาวอังกฤษ ชื่อ อาร์เธอร์ ซี. คลาร์ก (Arthur C. Clarke) ซึ่งเขียนบทความเรื่อง “Extra Terrestrial Relays ” ลงในนิตยสาร Wireless World เมื่อ พฤษภาคม ค.ศ.1945 โดย อาร์เธอร์ ซี.คลาร์ก ได้เสนอแนวความคิดในการติดต่อสื่อสารรอบโลก โดยใช้สถานีถ่ายทอดสัญญาณในอวกาศที่ความสูงระยะประมาณ 42,000 กิโลเมตร จากจุดศูนย์กลางโลก โดยมีความเร็วเท่ากับความเร็วที่โลกหมุนรอบตัวเอง 1 รอบ และใช้

เพียง 3 สถานีก็จะครอบคลุมพื้นที่รอบโลกทั้งหมด ซึ่งตรงกับหลักการของดาวเทียมวงโคจรค้างฟ้า (GEO Stationary Orbit)

2.1.2 องค์ประกอบระบบสื่อสารดาวเทียม (Satellite System)

ในระบบการสื่อสารดาวเทียมจะมององค์ประกอบหลัก 2 ส่วน คือ ดาวเทียมอยู่ในอวกาศ ระบบควบคุมและสั่งการสถานีดาวเทียมภาคพื้นดิน โดยมีการทำงานง่ายๆ ดังนี้ สถานีภาคพื้นดินจะส่งสัญญาณขาขึ้น (Uplink) กำลังส่งสูงผ่านจานสายอากาศไปยังจานสายอากาศไปยังจานสายอากาศและเครื่องบนดาวเทียม ทำการขยายสัญญาณ, แปลงความถี่ แล้วขยายให้กำลังสูงส่งผ่านจานสายอากาศเป็นสัญญาณขาลง (Downlink) มาถึงจานสายอากาศรับสถานีภาคพื้นดิน สถานีรับจะทำการขยายสัญญาณแล้วดำเนินการวิธีนำข้อมูลต่างๆ ไปใช้งาน

1) สถานีภาคพื้นดิน (Earth Station) ประกอบด้วยส่วนหลักๆ คือระบบจานสายอากาศ, ระบบการส่ง, ระบบการรับ และอุปกรณ์ช่องสัญญาณ โดยมีภาคย่อยที่สำคัญ คือ

1. จานสายอากาศ (Antenna) ทำหน้าที่แพร่กระจายคลื่นสัญญาณขาขึ้นไปยังดาวเทียม และทำหน้าที่รับคลื่นสัญญาณขาลงมาเข้าเครื่องรับจานสายอากาศที่ดีต้องมีคุณสมบัติ อัตราขยายกำลังสูง, ลำคลื่น (Beamwidth) แคบ, ลำคลื่นข้าง (Sidelobe) ต่ำ, ค่า Noise Temperature ต่ำ และมีความเที่ยงตรงสูง สามารถปรับทิศทางไปยังตำแหน่งดาวเทียมได้ตามต้องการ ปกตินิยมใช้สายอากาศแบบพาราโบลอยด์เป็นตัวสะท้อนสัญญาณ (Reflector) เพื่อให้รวมลำคลื่นได้แคบ ขนาดของจานสายอากาศโดยทั่วไปขึ้นกับความถี่ใช้งาน ความถี่ยิ่งสูงขนาดจานสายอากาศยิ่งเล็ก เช่น จานสายอากาศย่านความถี่ Ku-band จะเล็กกว่าย่านความถี่ C-band นอกจากนี้ยังขึ้นกับอัตราขยายกำลัง (Gain) ของสายอากาศ ถ้าต้องการอัตราขยายกำลังขยายสูง จานสายอากาศจะมีขนาดใหญ่ขึ้นเพื่อให้สามารถติดตามตำแหน่งดาวเทียมได้แม่นยำ จะต้องมีการควบคุมการหันของจานสายอากาศทั้งทางมุมทิศ (Azimuth) และทางมุมสูง (Elevation) อย่างดี ค่า G/T หรือ Gain ต่อ Thermal noise จะเป็นตัวกำหนดคุณภาพของจานสายอากาศ

2. ภาคขยายกำลังสูง (High Power Amplifier:HPA) ทำหน้าที่ขยายกำลังให้สูงก่อนส่งกำลังออกอากาศ อาจใช้หลอด Klystron, TWT (Travelling Wave Tube) หรือ Solid State เป็นภาคขยายกำลังก็ได้โดยหลอด Klystron จะให้กำลังขยายค่อนข้างสูงแต่ค่อนข้างยุ่งยากในการใช้งานแบบ Solid State ที่เรียกว่า SSPA (Solid State Power Amplifier) ให้กำลังขยายไม่สูงหนักแต่สะดวกในการใช้งาน ส่วนภาคขยายปานกลางและมีใช้งานมาพอสมควร

3. ภาคขยายสัญญาณรบกวนต่ำ (Low Noise Amplifier: LNA) ทำหน้าที่ขยายสัญญาณกำลังต่ำมากๆ ที่เครื่องรับรับได้เพื่อให้มีกำลังพอที่จะนำมาใช้งาน โดยให้มีสัญญาณรบกวนต่ำที่สุดซึ่งจะดูคุณสมบัติได้จากค่า Noise Temperature

4. ภาคแปลงความถี่ขาขึ้น (Up Converter) และภาคแปลงความถี่ขาลง (Down Converter) ภาคแปลงความถี่ขาขึ้น ทำหน้าที่แปลงความถี่ IF ให้เป็นความถี่ RF ก่อนส่งอากาศ และภาคแปลงความถี่ IF เพื่อให้สะดวกในการขยายสัญญาณ

2) ดาวเทียม [3] เป็นเครื่องมือทางอิเล็กทรอนิกส์ที่ซับซ้อน มีส่วนประกอบหลายๆ อย่างประกอบเข้าด้วยกันและสามารถทำงานได้โดยอัตโนมัติ สามารถโคจรรอบโลกด้วยความเร็วที่สูงพอที่จะหนีจากแรงดึงดูดของโลกได้ การสร้างดาวเทียมนั้นมีความพยายามออกแบบให้ชิ้นส่วนต่างๆ ทำงานได้อย่างมีประสิทธิภาพมากที่สุด ดาวเทียมประกอบด้วยส่วนประกอบเป็นจำนวนมาก แต่ละส่วนจะมีระบบควบคุมการทำงานแยกย่อยกันไป และมีอุปกรณ์เพื่อควบคุมให้ระบบต่างๆ ทำงานร่วมกัน โดยองค์ประกอบส่วนใหญ่ของดาวเทียมประกอบด้วยส่วนต่างๆ ดังนี้

1. ระบบควบคุมตำแหน่งและวงโคจรดาวเทียมปกติจะประกอบด้วยอุปกรณ์ ดังนี้ ได้แก่มอเตอร์จรวดที่คอยทำหน้าที่ปรับเปลี่ยนวงโคจรดาวเทียมให้อยู่ในวงโคจรถูกต้องเมื่อเกิดการคลาดเคลื่อน(Beam) มายังตำแหน่งบนพื้นโลกอย่างถูกต้องระบบการควบคุมตำแหน่งอาจใช้ตัวดาวเทียมหมุน ที่เรียกว่า spinners หรือ ใช้ Momentum wheels ช่วยวิธีหลังนี้นิยมใช้ในปัจจุบัน เพราะทำให้ลดขนาดแผงโซลาร์เซลล์ลงได้ถึง 1/3 เท่า ส่วนระบบควบคุมวงโคจรนั้นเราใช้ Gas Jet ควบคุมวงโคจรให้อยู่ในระนาบเส้นศูนย์สูตร

2. ระบบตรวจจับและสั่งการดาวเทียม (Telemetry, Tracking and Command:TT&C) ระบบนี้มีทั้งส่วนที่อยู่บนดาวเทียมและบนพื้นดินทำงานสัมพันธ์กัน โดย Telemetry จะส่งข้อมูลได้จากการตรวจจับ (Sensor) สัญญาณควบคุมต่างๆ บนดาวเทียม แล้วส่งกลับมายังสถานีภาคพื้นดิน ระบบ Tracking บนภาคพื้นดินจะติดตามดาวเทียมและรับสัญญาณจากระบบ Telemetry ส่งให้ระบบ Command นำเอาสัญญาณไปประมวลในระบบคอมพิวเตอร์ เป็นสัญญาณสั่งการส่งไปยังดาวเทียม เพื่อปรับแก้ตำแหน่งวงโคจรและระบบควบคุมต่างๆ ในตัวดาวเทียมให้ถูกต้อง

3. ระบบจ่ายกำลังไฟฟ้า ดาวเทียมทุกแบบได้รับพลังงานมาจากแผงรับพลังงานแสงอาทิตย์ (Solar Cells) เพื่อนำไปใช้ในระบบสื่อสารของดาวเทียมโดยเฉพาะภาคส่งพลังงานที่เหลือจะนำไปใช้ในส่วนอื่นๆ ซึ่งเรียกว่า Housekeeping เพื่อสนับสนุนดาวเทียมให้ทำงานอย่างมีประสิทธิภาพ

4. ระบบสื่อสารดาวเทียม เป็นส่วนประกอบหลักของดาวเทียมสื่อสารระบบอื่นเป็นเพียงส่วนสนับสนุนระบบนี้จะประกอบด้วยจานสายอากาศที่คอยรับส่งสัญญาณแบนด์กว้าง , ภาครับ-ส่ง และขยายกำลังของสัญญาณ ที่เรียกว่า Transponder ซึ่งเป็นหน่วยรับ-ส่งสัญญาณแต่ละช่องในตัวดาวเทียม

5. ระบบสายอากาศ ระบบนี้อาจถือว่าเป็นส่วนหนึ่งของระบบสื่อสารดาวเทียมโดยแยกออกมาจาก ทรานสปอนเดอร์ (Transponder) ปัจจุบันดาวเทียมมีระบบจานสายอากาศที่ซับซ้อนเพื่อให้สามารถแยกลำคลื่น (beam) ส่งมาครอบคลุมพื้นโลกในรูปแบบต่างๆ ได้ตามต้องการ

2.2 รายละเอียดการแข่งขัน Cansat Competition 2018

ปริญญาณิพนธ์เรื่องนี้จัดทำขึ้นเพื่อใช้ในการแข่งขัน CanSat Competition 2018 [4] ที่จะจัดขึ้นทุกปี โดยการแข่งขันนี้ถูกจัดขึ้นโดยหน่วยงาน American Astronautical Society (AAS) และ American Institute of Aeronautics and Astronautics (AIAA) ซึ่งการแข่งขันนี้คือ การแข่งขันการออกแบบและสร้างอุปกรณ์ที่มีการทำงานเลียนแบบดาวเทียม (CanSat) โดยต้อง ออกแบบและสร้างให้เหมาะสมตามกฎกติกาการแข่งขัน และในแต่ละปีก็จะมีการเพิ่มภารกิจต่างๆ เพื่อให้มีความท้าทายแก่ผู้แข่งขันโดยจะมีการออกข้อบังคับที่ใช้สำหรับการแข่งขันดังนี้

ภารกิจประจำปี 2018 [5] คือจำลอง Space probe (CanSat) เพื่อเข้าสู่ชั้นบรรยากาศ โดย Probe จะต้องบรรจุใส่กล่องขนาดใหญ่ โดยจะต้องรักษาไม่ให้ไข่วัสดุแตกระหว่างการร่อนลงสู่พื้นดิน ซึ่งลำดับการดำเนินงานต้องเป็นดังนี้ [6]

- 1) Probe (CanSat) จะถูกปล่อยที่ระดับความสูง 670 เมตร ถึง 725 เมตร และจากนั้นก็จะถูกปล่อยออกจากจรวด (Rocket) โดยจะต้องไม่ถูกควบคุมทิศทางในการปล่อย
- 2) ในขณะที่ปล่อย Probe จะเปิด aero-braking heat shield ทันที และร่อนลงด้วยอัตราความเร็ว 50 เมตรต่อวินาที ซึ่งจะต้องรักษาความเสถียรภาพให้มั่นคง
- 3) ที่ระดับความสูง 300 เมตร Probe (CanSat) จะปล่อย aero-braking heat shield ทิ้งไป พร้อมกับปล่อยร่มชูชีพและลดความเร็วในการร่อนลงให้เหลือ 5 เมตรต่อวินาที
- 4) ขณะที่ร่อนลงสู่พื้นดิน Probe (CanSat) จะต้องปกป้องไข่ไก่เพื่อไม่ให้ไข่แตกซึ่งรายละเอียดข้อบังคับการแข่งขันพื้นฐานในส่วนของภาครับ จะแสดงในตารางที่ 2.1

ตารางที่ 2.1 ข้อบังคับพื้นฐานในส่วนของภาครับ

ข้อที่	ข้อกำหนด
1	XBEE ที่ใช้สำหรับการส่ง telemetry ต้องมีความถี่ที่ 2.4 GHz ทั้ง Series 1 และ Series2 หรือที่ความถี่ 900 MHz
2	XBEE ต้องมี NETID/PANID ซึ่งตั้งตามหมายเลขของแต่ละทีม
3	XBEE ห้ามใช้โหมดบรอดแคสต์
4	แต่ละทีมต้องพัฒนาสถานีภาคพื้นดินของตัวเอง
5	Telemetry ทั้งหมดจะต้องแสดงแบบเรียลไทม์ขณะที่ร่อนลงสู่พื้นดิน
6	Telemetry ทั้งหมดจะต้องแสดงผลในหน่วยทางวิศวกรรม เช่น เมตร, เมตรต่อวินาที และ องศาเซลเซียส เป็นต้น
7	ไม่อนุญาตให้ใช้เลเซอร์
8	สถานีภาคพื้นดินจะต้องมีคอมพิวเตอร์แล็ปท็อปที่มีการใช้งานแบตเตอรี่อย่างน้อย 2 ชั่วโมง XBEE และสายอากาศ

ข้อที่	ข้อกำหนด
9	สถานีภาคพื้นดินต้องเป็นแบบพกพาเพื่อให้ทีมสามารถวางตำแหน่งไว้ที่สถานีปฏิบัติงานภาคพื้นดิน โดยจะไม่มีไฟ AC อยู่ที่บริเวณที่ทำการปฏิบัติงานของสถานีภาคพื้นดิน

2.3 ทฤษฎีพื้นฐานและการประยุกต์ใช้งานสายอากาศ [7]

2.3.1 หน้าที่และความหมายของสายอากาศ

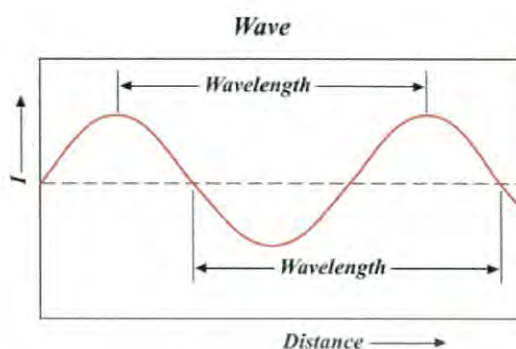
สายอากาศ อุปกรณ์สำหรับรับและส่งคลื่นความถี่วิทยุ ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าเป็นคลื่นแม่เหล็กไฟฟ้า และในทางกลับกัน ก็เปลี่ยนคลื่นแม่เหล็กไฟฟ้าเป็นพลังงานไฟฟ้าเช่นกัน สายอากาศมีหลายขนาดและรูปแบบขึ้นอยู่กับการใช้งาน เช่น สายอากาศสำหรับเครื่องรับโทรทัศน์ในบ้านส่วนใหญ่เป็นสายอากาศชนิด ยากิ-อูตะ มักติดตั้งไว้บนหลังคาทำด้วยอะลูมิเนียมเพราะน้ำหนักเบาและทนต่อสภาพอากาศได้ดีกว่าโลหะทั่วไป สายอากาศของไม้คล้องเป็นเพียงสายไฟสั้น ๆ หรือสายอากาศของโทรศัพท์มือถือเป็นเพียงจุดเชื่อมต่อเล็ก ๆ เท่านั้น คำว่าสายอากาศเป็นศัพท์เฉพาะด้านไฟฟ้า และ อิเล็กทรอนิกส์ บัญญัติขึ้นจากคำศัพท์ในภาษาอังกฤษ "Antenna" หรือ "Aerial" ในวงจรอิเล็กทรอนิกส์อาจเขียนอักษรย่อ Ant. อย่างไรก็ตามบุคคลทั่วไปนิยมเรียกว่า เสาอากาศ อาจจะเป็นเพราะเดิมใช้เสาสูงๆสำหรับติดตั้งสายอากาศนั่นเอง

สายอากาศแบ่งตามรูปแบบการรับ-ส่งคลื่นได้ดังนี้

- (1) สายอากาศแบบรอบตัว สามารถรับ-ส่งคลื่นได้ดีในทุกทิศทางเฉลี่ยกันไปโดยรอบ
- (2) สายอากาศแบบกึ่งรอบตัว สามารถรับ-ส่งคลื่นได้ดีเกือบรอบตัวแต่มีอัตราขยายสูงกว่า แบบรอบตัว
- (3) สายอากาศแบบทิศทางเดียว สามารถรับ-ส่งคลื่นได้ดีในทิศทางที่กำหนดและจะมี อัตราขยาย (Gain) สูงกว่าประเภทอื่น

2.3.2 ความถี่และความยาวคลื่น

โดยทั่วไปนิยมแบ่งคลื่นวิทยุออกเป็นย่านความถี่ต่าง ๆ โดยมีหน่วยเป็นเฮิรตซ์ (Hertz) ใน ประวัติศาสตร์การวิทยุแบ่งคลื่นวิทยุตามความยาวคลื่น (Wavelength) ความสัมพันธ์ระหว่างความถี่และความยาวคลื่นดังแสดงในรูปที่ 2.1 [8] และมีย่านความถี่และความยาวคลื่นแสดงดังตารางที่ 2.2



รูปที่ 2.1 ความสัมพันธ์ระหว่างความถี่และความยาวคลื่น

ตารางที่ 2.2 ย่านความถี่และความยาวคลื่น

ย่านความถี่	ความถี่	ความยาวคลื่น
Very Low Frequency (VLF)	ต่ำกว่า 30 kHz	ยาวกว่า 10 km
Low Frequency(LF)	30-300 kHz	10-1 km
Medium Frequency(MF)	300-3000 kHz	1000-100 m
High Frequency (HF)	3-30 MHz	100-10 m
Very High Frequency (VHF)	30-300 MHz	10-1 m
Ultra High Frequency (UHF)	300-3000 MHz	100-10 cm
Super High Frequency (SHF)	3-30 GHz	10-1 cm
Extremely High Frequency (EHF)	30-300 GHz	mm

2.3.3 คุณลักษณะและพารามิเตอร์ต่าง ๆ ของสายอากาศ

สายอากาศชนิดต่าง ๆ ที่มีการใช้งานอยู่ทั่วไปมีคุณลักษณะและพารามิเตอร์ต่าง ๆ ที่จำเป็นต้องพิจารณาประกอบการประเมินประสิทธิภาพของสายอากาศเพื่อช่วยในการตัดสินใจและประยุกต์ใช้ให้เหมาะสมกับงานต่าง ๆ มากมาย โดยมีส่วนสำคัญ ดังนี้

2.3.3.1 อัตราส่วนแรงดันคลื่นนิ่ง (Voltage standing Wave Ratio)

อัตราส่วนระหว่างแรงดันค่าสูงสุดและค่าต่ำสุดในคลื่นนิ่งเรียกว่า อัตราส่วนแรงดันคลื่นนิ่ง (Voltage Standing Wave Ratio: VSWR) ดังสมการที่ (2.1)

$$VSWR = \frac{1+|\Gamma|}{1-|\Gamma|} \quad (2.1)$$

Γ คือ สัมประสิทธิ์การสะท้อนกลับของแรงดัน (Voltage Reflection Coefficients) สัมประสิทธิ์การสะท้อนกลับของแรงดัน ยังสามารถหาได้จากอัตราส่วนผลต่างและผลรวมระหว่างโวลต์กับอิมพีแดนซ์คุณลักษณะของสายนำสัญญาณ ดังสมการที่ (2.2)

$$\Gamma = \frac{V_r}{V_i} = \frac{Z_L - Z_0}{Z_L + Z_0} \quad (2.2)$$

Γ คือ สัมประสิทธิ์การสะท้อนกลับของแรงดัน (Voltage Reflection Coefficients)

V_r คือ แรงดันสะท้อนกลับ

V_i คือ แรงดันตกกระทบ

Z_L คือ โหลดอิมพีแดนซ์

Z_0 คือ อิมพีแดนซ์คุณลักษณะของสายนำสัญญาณ

ในกรณีที่ต่อไว้ด้วยแมตซ์ซึ่งโหลดนั้นค่า VSWR เป็น 1 ซึ่งเป็นค่าที่ดีที่สุด

2.3.3.2 การสูญเสียย้อนกลับ (Return Loss)

การสูญเสียย้อนกลับของสายอากาศแสดงค่ากำลังที่สูญเสียที่โหลด เมื่ออิมพีแดนซ์ของสายส่งและสายอากาศไม่แมตซ์กัน การสูญเสียย้อนกลับมีความสัมพันธ์กับ VSWR ซึ่งเป็นการแสดงการแมตซ์อิมพีแดนซ์ระหว่างสายส่งกับสายอากาศตามสมการ โดยการสูญเสียย้อนกลับสามารถหาได้จากสมการที่ (2.3)

$$S_{11} = -20 \log_{10} |\Gamma| \text{ (dB)} \quad (2.3)$$

สำหรับการแมตซ์อิมพีแดนซ์ที่สมบูรณ์ระหว่างสายส่งและสายอากาศ เมื่อ $\Gamma = 0$ ค่าความสูญเสียย้อนกลับเป็นอนันต์ แสดงว่าไม่มีกำลังงานสะท้อนกลับ ในทำนองเดียวกันเมื่อ $\Gamma = 1$ ค่าความสูญเสียย้อนกลับจะเป็น 0 dB ซึ่งแสดงว่ากำลังงานสะท้อนกลับหมด

2.3.3.3 ประสิทธิภาพของสายอากาศ (Antenna Efficiency)

ประสิทธิภาพของสายอากาศเป็นพารามิเตอร์ที่รวมประสิทธิภาพการสูญเสียที่สายอากาศและในโครงสร้างของสายอากาศการสูญเสียต่าง ๆ หาได้จากการสะท้อนกลับเนื่องจากการไม่แมตช์กันระหว่างสายส่งกับสายอากาศการสูญเสียจากตัวนำและฉนวนโดยประสิทธิภาพรวมของสายอากาศสามารถเขียนเป็นสมการที่ (2.4)

$$e_t = e_r e_c e_d \quad (2.4)$$

e_t คือ ประสิทธิภาพทั้งหมดของสายอากาศ

e_r คือ $(1 - |\Gamma|^2)$ ประสิทธิภาพการสะท้อนกลับเนื่องจากการไม่แมตช์

e_c คือ ประสิทธิภาพของตัวนำ

e_d คือ ประสิทธิภาพของฉนวน (dielectric) และโดยทั่วไป e_c จะรวมเป็นตัวเดียวกันตามสมการที่ (2.5)

$$e_{cd} = e_c e_d = \frac{R_R}{R_R + R_L} \quad (2.5)$$

R_R คือ ความต้านทานจากการแผ่พลังงานคลื่นออกไป

R_L คือ ความต้านทานที่โหด

2.3.3.4 สภาพเจาะจงทิศทาง (Directivity)

ไดเรกตีวีตีเป็นการบอกความสามารถเชิงทิศทางของสายอากาศ เป็นอัตราส่วนระหว่าง ความเข้มของการแผ่พลังงานในทิศทางที่สนใจกับความเข้มของการแผ่พลังงานโดยเฉลี่ย เมื่อมีการแผ่ พลังงานออกไป รอบทิศทางอย่างเท่าเทียมกันโดยไม่คิดกำลังงานส่วนที่สูญเสียไปดังสมการที่ (2.6) และ สมการที่ (2.7)

$$D = \frac{U}{U_i} = \frac{4\pi U}{P_{rad}} \quad (2.6)$$

D คือ สภาพเจาะจงทิศทางของสายอากาศ

U คือ ความเข้มของการแผ่กำลังงาน

U_i คือ ความเข้มของการแผ่กำลังงานเฉลี่ย

P_{rad} คือ กำลังงานที่สายอากาศแผ่ออกไป

โดยทั่วไปถ้าไม่กำหนดทิศทางใช้สภาพเจาะจงทิศทางในทิศที่สายอากาศแผ่พลังงานได้ดีที่สุด

$$D_0 = \frac{U_{\max}}{U_i} = \frac{4\pi U_{\max}}{P_{\text{rad}}} \quad (2.7)$$

2.3.3.5 อัตราขยายของสายอากาศ (Gain)

อัตราขยายของสายอากาศเป็นความสัมพันธ์ที่ได้จากไดเรกทิวิตี โดยรวมประสิทธิภาพ ของสายอากาศเข้ามาด้วย ในขณะที่ไดเรกทิวิตีแสดงคุณสมบัติในการชี้ทิศทางของสายอากาศเท่านั้น การคิดอัตราขยายของสายอากาศ วัดเทียบกับสายอากาศอ้างอิง โดยอัตราขยายของสายอากาศส่ง คือกำลัง สองของอัตราส่วนระหว่างความเข้มสนามตามทิศที่มีการแพร่กระจายคลื่นมากที่สุดเมื่อเทียบกับความเข้ม สนามที่จุดเดียวกันของสายอากาศอ้างอิง หรือแสดงในรูปของอัตราส่วนของค่าพลังงานที่ต้องใช้ในการส่ง ของสายอากาศทั้งสอง เพื่อให้เกิดความเข้มสนามขนาดเท่ากัน (ณ จุดเดียวกัน) ในทิศทางที่มีการ แพร่กระจายคลื่นมากที่สุด หรืออัตราขยายของสายอากาศรับ คือ อัตราส่วนระหว่างค่าความเข้มสนามแม่ พลังงานของสายอากาศทดสอบกับสายอากาศอ้างอิง ณ จุดตั้งสายอากาศที่เดียวกัน

การใช้สายอากาศอ้างอิงมักเป็นแบบไดโพลขนาด $\lambda/2$ หรือแบบไอโซโทรปิก (Isotropic) ซึ่งมีลักษณะพิเศษ คือ กระจายคลื่นได้รอบตัวทุกทิศในปริมาณที่เท่ากัน

อัตราขยายกำลัง (Power Gain) ของสายอากาศ ในทิศทางที่กำหนดให้นั้นมีค่าเท่ากับ A_n คูณอัตราส่วนของความเข้มของการแพร่กระจายคลื่นในทิศทางนั้น ต่อ (หาร) กำลังงานสุทธิที่ สายอากาศรับจากขั้วต่อของเครื่องส่งเมื่อไม่กำหนดทิศทางไว้โดยทั่วไปคิดอัตราขยายกำลังในทิศทางที่มีการแพร่กระจายคลื่นแรงที่สุดตามสมการที่ (2.8)

$$\text{Gain} = \frac{4\pi U(\theta, f)}{P_{\text{in}}} \quad (2.8)$$

ตราขยายสัมพันธ์ เป็นอัตราส่วนของอัตราขยายกำลังในทิศทางที่กำหนดให้ ต่ออัตราขยายกำลังของสายอากาศที่ใช้เปรียบเทียบในทิศทางนั้นโดยกำลังงานที่เอนเข้าสายอากาศทั้งสอง นั้นต้องเท่ากัน สายอากาศที่ใช้เปรียบเทียบเป็นสายอากาศไดโพล สายอากาศปากแตร หรือสายอากาศอื่น ๆ ซึ่งคำนวณอัตราขยายได้ง่ายหรือรู้ค่าอยู่แล้ว แต่อย่างไรก็ตามโดยส่วนใหญ่สายอากาศที่ใช้เปรียบเทียบ เป็นไอโซโทรปิกพอยท์ซอร์สที่ไม่มีการสูญเสีย ดังนั้นจึงได้เป็นสมการที่ (2.9)

$$G_g = \frac{4\pi U(\theta, f)}{P_{\text{in}}} \quad (2.9)$$

เมื่อ P_{in} คือ กำลังงานที่ป้อนให้กับไอโซโทรปิกพอยท์ซอร์สที่ไม่มีการสูญเสีย กำลังงานที่แพร่กระจายทั้งหมด (p_{rad}) สัมพันธ์กับกำลังงานที่เอนให้สายอากาศ (p_{in}) ดังสมการที่ (2.10)

$$p_{rad} = e_t P_{in} \quad (2.10)$$

เมื่อ e_t คือประสิทธิภาพรวมของสายอากาศ (ไม่มีหน่วย) ทำให้สมการที่ (2.9) และ (2.10) มีความสัมพันธ์กันตามสมการที่ (2.11)

$$G_g(\theta, f) = \frac{[4\pi U(\theta, f)]}{P_{rad}} \quad (2.11)$$

และมีความสัมพันธ์กับอัตราขยายไดเรกทิฟ ตามสมการที่ (2.12)

$$G_g(\theta, f) = e_t D_g \quad (2.12)$$

ในทำนองเดียวกัน ค่าสูงสุดของอัตราขยายจะสัมพันธ์กับไดเรกทิวิตี แสดง ดัง สมการที่ (2.13)

$$\begin{aligned} G_0 &= G_g(\theta, f)|_{\max} \\ &= e_t D_g(\theta, f)|_{\max} \\ &= e_t D_0 \end{aligned} \quad (2.13)$$

ในทางปฏิบัติเมื่อกล่าวถึงอัตราขยายถึง อัตราขยายกำลังที่มีค่าสูงสุด แสดง ดังสมการที่ (2.14)

$$G_0(\text{dB}) = 10 \log_{10} [e_t D_0] \quad (2.14)$$

2.3.3.6 อิมพีแดนซ์ขาเข้า (Input Impedance)

พิจารณาสายอากาศเสมือนเป็นชิ้นส่วนหนึ่งในวงจรไฟฟ้า เมื่อต่อ แหล่งกำเนิด สัญญาณเพื่อเอนาพลังงานให้กับสายอากาศ พลังงานจะไหลเข้าสู่สายอากาศที่ละน้อย เนื่องจากการต้าน การไหลของพลังงานที่เรียกว่าอิมพีแดนซ์หรือความต้านทานเชิงซ้อนเกิดขึ้น อิมพีแดนซ์ดังกล่าวจะปรากฏที่ ขั้วของสายอากาศ เรียกว่าอิมพีแดนซ์ขาเข้า (Z_{in}) ดังสมการที่ (2.15)

$$Z_{in} = R_{in} + jX_{in} \quad (2.15)$$

X_{in} คือความต้านทานเชิงจินตภาพที่ทำให้เกิดการสะสมของพลังงานในบริเวณสนาม ใกล้สายอากาศโดยไม่แผ่กระจายออกไป และ R_{in} ประกอบด้วยสองส่วนคือ R_r หมายถึงความต้านทานพลังงานคลื่นที่แผ่ออกไปโดยสายอากาศ และ R_L หมายถึงความต้านทานที่โหลด ซึ่งรวมถึงความต้านทาน จากการสูญเสียที่เกิดขึ้นจากความร้อน สารไดอิเล็กตริก และตัวนำ

2.3.3.7 แบนด์วิดท์ (Bandwidth)

แบนด์วิดท์ของสายอากาศเป็นช่วงของความถี่ที่สามารถนำไปใช้งานได้ดี ซึ่งช่วงความถี่ ถูกกำหนดโดย VSWR ประมาณ 2 หรือพิจารณาจากการสูญเสียย้อนกลับ (S_{11}) ที่ระดับ -10 dB ดังสมการที่ (2.16)

$$BW_{\text{narrowband}} (\%) = \frac{f_u - f_l}{f_c} \times 100 \quad (2.16)$$

BW คือ แบนด์วิดท์ของสายอากาศ

f_u คือ ขอบความถี่สูงของย่านความถี่

f_l คือ ขอบความถี่ต่ำของย่านความถี่

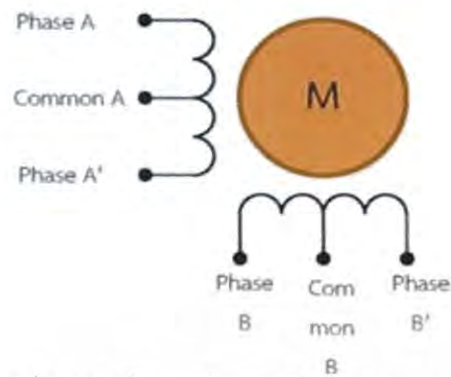
f_c คือ ความถี่กลางของย่านความถี่

2.4 หลักการทำงานของสเต็ปเปอร์มอเตอร์

2.4.1 ข้อมูลของสเต็ปเปอร์มอเตอร์และหลักการทำงานของสเต็ปเปอร์มอเตอร์ [9]

สเต็ปเปอร์มอเตอร์ คือ มอเตอร์ที่มีการหมุนเป็นขั้นๆ โดยแกนของมอเตอร์จะหมุนด้วยมุมค่าคงที่เมื่อมีสัญญาณ Pulse มากกระตุ้น เช่น 0.9, 1.8, 5, 7.5, 15 หรือ 45 องศาต่อสเต็ป เนื่องจากการควบคุมสเต็ปเปอร์มอเตอร์สามารถใช้สัญญาณดิจิทัลควบคุมได้โดยตรง และสามารถบังคับทิศทางและความเร็วของแกนหมุน สเต็ปเปอร์มอเตอร์ โดยมีข้อดี ดังนี้ หมุนได้ครบ 360 องศา ต่อเนื่อง, การควบคุมตำแหน่งไม่ต้องอาศัยตัวตรวจจับการหมุน, ไม่ต้องใช้แปรงถ่านดังนั้นจึงทำให้ไม่มีส่วนที่สึกหรอและปัญหาจากประกายไฟ และการควบคุมโดยทางวงจรถิจิตอลหรือไม่โครคอนโทรลเลอร์ทำได้ง่ายและสะดวก

โดยทั่วไป สเต็ปเปอร์มอเตอร์ (Two-phase Stepper Motor) สามารถแบ่งได้ 2 ประเภท คือ Unipolar และ Bipolar ดังรูปที่ 2.2 และรูป 2.3 ตามลำดับ ผู้ใช้สามารถสังเกตข้อแตกต่างได้จากจำนวนสายไฟ กล่าวคือ แบบ Unipolar จะมีสายไฟต่อ 5 (Common A และ Common B เชื่อมต่อกัน) หรือ 6 เส้น และแบบ Bipolar จะมีสายไฟต่อ 4 เส้น เท่านั้น



รูปที่ 2.2 สเต็ปเปอร์มอเตอร์แบบ Unipolar



รูปที่ 2.3 สเต็ปเปอร์มอเตอร์แบบ bipolar

การควบคุมการหมุนของสเต็ปเปอร์มอเตอร์แบบ Unipolar สามารถทำได้โดย ควบคุมกระแสไฟที่จ่ายให้กับขดลวดในแต่ละเฟส อย่างเป็นลำดับที่แน่นอน โดยถ้าหากเราต้องการให้กระแสไหลในเฟสใดๆ ก็จะทำให้สถานะของเฟสนั้นๆ เป็นสถานะลอจิก “1” การควบคุมการหมุนของสเต็ปเปอร์มอเตอร์สามารถทำได้ 3 วิธีดังนี้

1. การควบคุมการหมุนแบบเฟสเดียว (Single Phase) หรือ Wave Drive เป็นการป้อนกระแสไฟให้กับขดลวดของสเต็ปเปอร์มอเตอร์ทีละขด โดยจะป้อนกระแสเรียงตามลำดับกันไปตามตาราง 2.3 (ตัวเลข 1 หมายถึง มีการจ่ายแรงดันให้กับ ขดลวด 0 คือไม่จ่ายแรงดันให้กับขดลวด) ดังนั้นกระแสที่ไหลในขดลวดจะทำการไหลในทิศทางเดียวกันทุกขด ลักษณะเช่นนี้จึงทำให้แรงขับของสเต็ปเปอร์มอเตอร์มีน้อย จึงไม่เป็นที่นิยมใช้

ตาราง 2.3 ลำดับการป้อนกระแสแบบเฟสเดียว

Step	Phase A	Phase B	Phase A'	Phase B'
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	Back to step 1			

2.การควบคุมการหมุนแบบสองเฟส (Two Phase) หรือ Full Step Drive เป็นการป้อนกระแสไฟให้กับขดลวดของสเต็ปเปอร์มอเตอร์ทั้ง 2 ขด พร้อมๆ กันไป และป้อนกระแสเรียงตามลำดับกันไป ตามตาราง 2.4 ดังนั้นจึงมีกระแสไหลในขดลวดของมอเตอร์มากขึ้น และจะทำให้มอเตอร์มีแรงบิดมากขึ้น

ตาราง 2.4 ลำดับการป้อนกระแสแบบสองเฟส

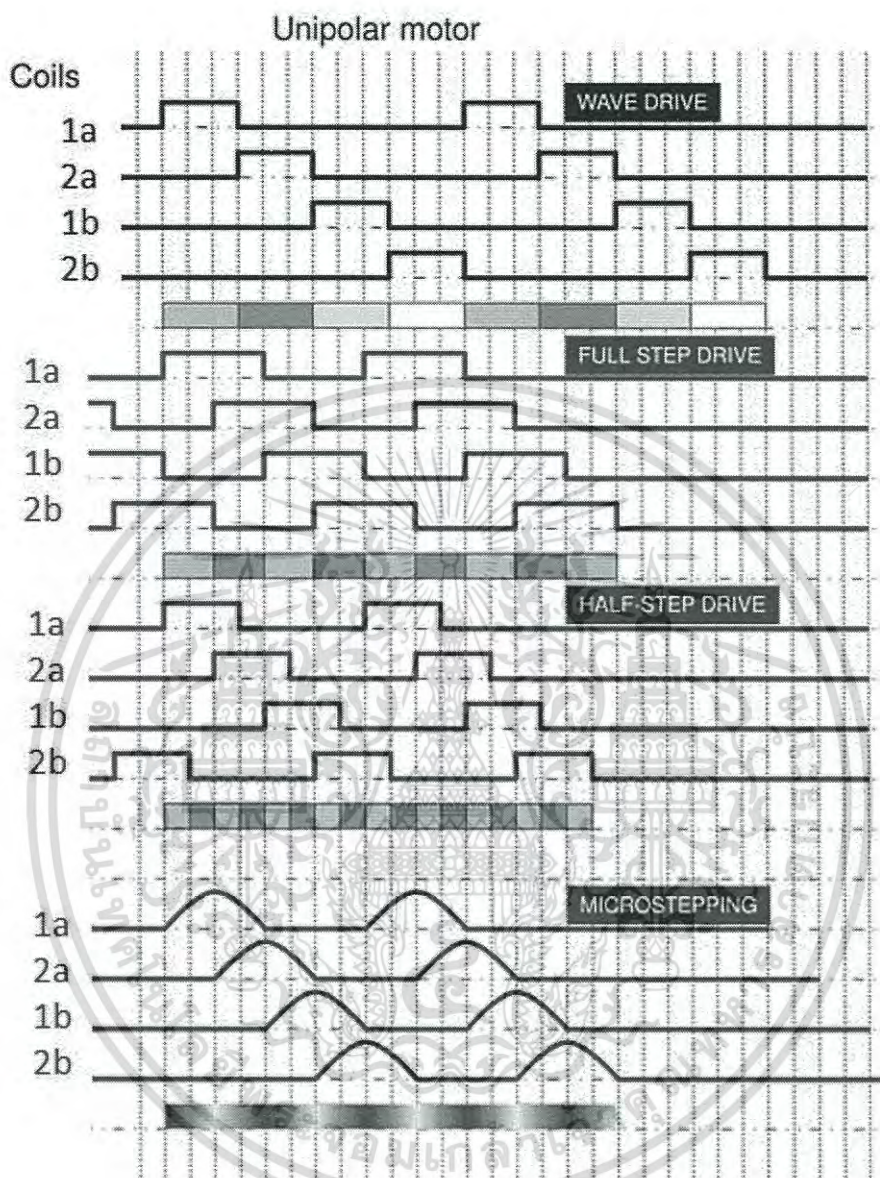
Step	Phase A	Phase B	Phase A'	Phase B'
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1
5	Back to step 1			

3.การควบคุมการหมุนแบบครึ่งเฟส (Half Step) เป็นการป้อนกระแสแบบ Two Phase และ Single Phase สลับกันไป ตามตาราง 2.5 ซึ่งเป็นการเพิ่มความละเอียดของตำแหน่งในการหมุน

ตาราง 2.5 ลำดับการป้อนกระแสแบบครึ่งเฟส

Step	Phase A	Phase B	Phase A'	Phase B'
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	Back to Step 1			

ซึ่งมอเตอร์ประกอบด้วยส่วนสำคัญ 2 ส่วนคือ 1. rotor ซึ่งเป็นส่วนที่หมุนหรือเคลื่อนที่ได้ 2. คือ stator ซึ่งเป็นส่วนที่อยู่กับที่ไม่สามารถเคลื่อนที่ได้ สำหรับสเต็ปเปอร์มอเตอร์ก็เช่นเดียวกันที่ประกอบไปด้วยส่วนสำคัญ 2 ส่วนนี้ โดยที่ส่วนที่เป็น stator ของสเต็ปเปอร์มอเตอร์นี้จะเป็นส่วนซึ่งมีขดลวดพันล้อมอยู่บนแกนเหล็ก เมื่อเราจ่ายไฟเข้าที่ขดลวดที่พันอยู่บน stator ที่ขด 1 ก่อน ก็จะทำให้เกิดอำนาจแม่เหล็กขึ้นที่ตำแหน่ง 1 นี้ และผลักให้ rotor เกิดการเคลื่อนที่ได้ การเคลื่อนที่นี้เรียกว่ามันเคลื่อนที่ไป 1 step ซึ่งการที่มันจะเคลื่อนที่ไปเป็นมุมเท่าไรก็ขึ้นอยู่กับ spec ของสเต็ปเปอร์มอเตอร์ตัวนั้นว่ามันสามารถที่จะหมุนได้ step ละกี่องศา และต่อไปถ้าเราหยุดจ่ายไฟเข้าที่ขดลวด 1 และไปจ่ายไฟเข้าที่ขดลวด 2 แทนก็จะทำให้เกิดสนามแม่เหล็กขึ้นที่บริเวณ 2 บน stator และก็จะทำให้สนามแม่เหล็กนี้ผลัก rotor ให้เคลื่อนที่ต่อไปได้ และถ้าเราหยุดจ่ายกระแสเข้าที่ 2 และไปจ่ายเข้าที่ขด 3 แทนก็จะทำให้สนามแม่เหล็กมาเกิดที่บริเวณ 3 และผลักให้ rotor เคลื่อนที่ต่อไปได้ โดยที่จะแสดงดังรูปที่ 2.4[10] ซึ่งเป็นภาพแสดงการควบคุมการหมุนของสเต็ปเปอร์มอเตอร์ และความเร็วการหมุนของแกนสเต็ปเปอร์มอเตอร์จะขึ้นอยู่กับระยะเวลา (Time Delay) ในแต่ละลำดับ ถ้า Time Delay มีค่าน้อยแกนจะหมุนเร็วขึ้น ถ้า Time Delay มีค่ามากแกนจะหมุนช้าและหมุนไม่สม่ำเสมอ

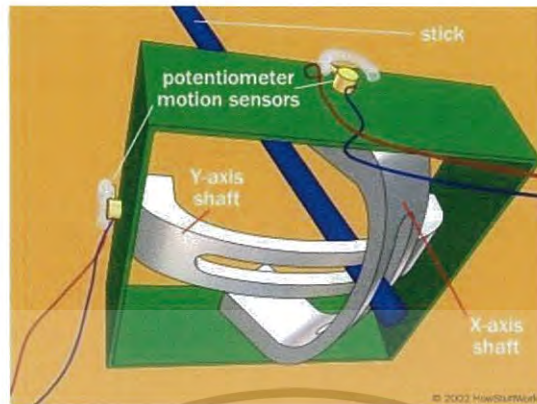


รูปที่ 2.4 การควบคุมการหมุนของสเต็ปเปอร์มอเตอร์ [10]

2.5 หลักการทำงานของจอยสติ๊กแบบแอนะล็อก

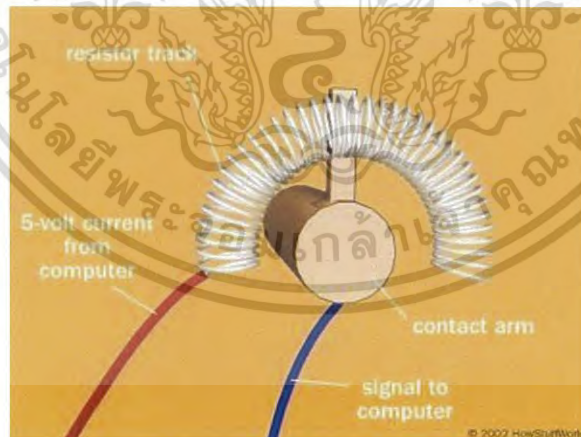
2.5.1 ทฤษฎีและหลักการทำงานของจอยสติ๊กแบบแอนะล็อก [11]

ในการที่จะสื่อสารรูปแบบการเคลื่อนไหวอย่างสมบูรณ์ จอยสติ๊กจะต้องประมวลตำแหน่งของ สติ๊กหรือแกนแห่งในสองทาง คือ X-axis (จากซ้ายไปขวา) และ Y-axis (จากบนลงล่าง) ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 หลักการภายในจอยสติ๊กแบบแอนะล็อก [11]

จอยสติ๊กที่ได้รับการออกแบบอย่างมีมาตรฐาน มือจับจะทำหน้าที่ขยับท่อเล็กๆ ที่วางอยู่บนด้ามเพลลาที่ถูกเจาะและสามารถหมุนรอบได้ การเอียงสติ๊กไปข้างหน้าและข้างหลังจะไปหมุนทิศทางของ Y-axis จากด้านหนึ่งไปยังอีกด้านหนึ่ง ถ้าเอียงสติ๊กจากซ้ายไปขวา จะทำให้เพลลาของ X-axis shaft หมุน แต่ถ้าขยับสติ๊กตามแนวขวาง มันจะไปหมุนเพลลาทั้งสองข้าง ในการกำหนดตำแหน่งของสติ๊ก ระบบควบคุมจอยสติ๊กจะตรวจจับตำแหน่งเพลลาทั้งสองด้านได้อย่างง่ายดาย the conventional analog design จอยสติ๊ก จะทำเช่นนี้กับเครื่องวัดระดับไฟฟ้าสองตัว หรือกับเครื่องต้านทานกระแสไฟฟ้า ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 การกำหนดค่าความต้านทานจากตำแหน่งของสติ๊ก [11]

2.6 ทฤษฎีและหลักการของ Zigbee

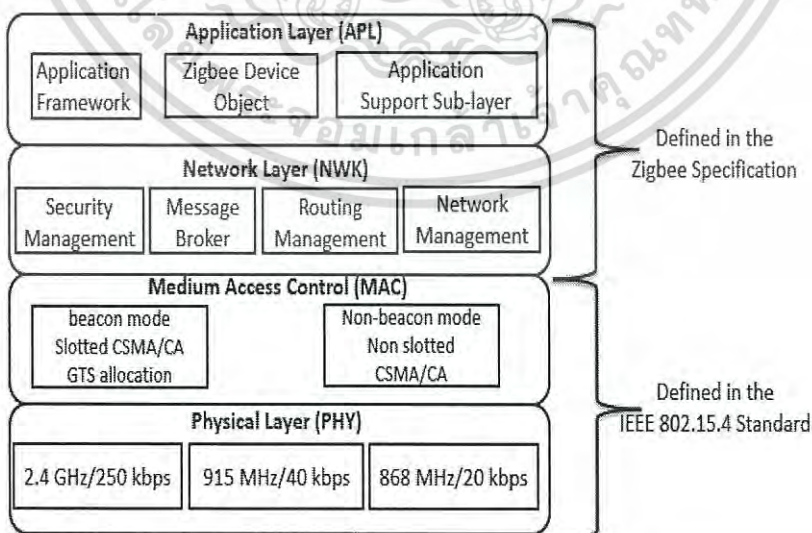
2.6.1 คุณสมบัติของ Zigbee [12]

เป็นมาตรฐานของอุปกรณ์ไร้สาย ถูกกำหนดโดยกลุ่ม Zigbee Alliance ซึ่งเริ่ม ก่อตั้งขึ้นเมื่อปี 2002 เพื่อใช้ควบคุมอุปกรณ์ต่างๆ แบบไร้สายซึ่งมีความเร็วสูงไม่มากนัก เช่น สวิตช์เปิดปิดแสงสว่าง ระบบควบคุมอุณหภูมิห้อง รวมไปถึงใช้ในการรับค่าจาก Sensor ต่างๆ เป็นต้น โดย Zigbee ถูกออกแบบโดยมุ่งให้มีคุณลักษณะดังนี้

1. เป็นเครือข่ายไร้สายในระยะใกล้
2. ราคาประหยัด อุปกรณ์ที่ใช้มีราคาถูก เสียค่าติดตั้งและดูแลน้อย เนื่องจากแบตเตอรี่ที่ใช้เป็นแบบ primary cell ซึ่งมีราคาถูกและอายุการใช้งานนาน ไม่ใช่แบตเตอรี่ที่สามารถนำมาชาร์ตเพื่อใช้งานใหม่ได้
3. ติดตั้งง่าย สามารถประยุกต์ใช้งานได้หลากหลาย
4. สามารถรับส่งข้อมูลได้โดยเชื่อมั่นในความถูกต้องได้
5. ประหยัดพลังงานโดยพลังงานในการทำงานต่ำ โดยมาตรฐานซิกบีตาม IEEE 802.15.4 กำหนดให้มีโหมดของการใช้พลังงาน 2 สถานะ คือ สถานะที่มีการรับและส่งของข้อมูล ซึ่งเรียกว่า Active และอีกสถานะคือ สถานะที่ไม่มีการทำงานชั่วขณะหนึ่ง หรือเรียกว่า Sleep

2.6.2 Zigbee Protocol Stack

Zigbee Protocol Stack นั้นถูกออกแบบขึ้นให้ทำงานตาม Zigbee Specification ซึ่งถูกกำหนดจาก Zigbee Alliance โดย Zigbee Specification ฉบับแรกถูกกำหนดขึ้นเมื่อปี 2004 และได้รับการปรับปรุงเรื่อยมาสำหรับรูปแบบของ Zigbee Protocol Stack นั้นมีรูปแบบดังแสดงในรูปที่ 2.7

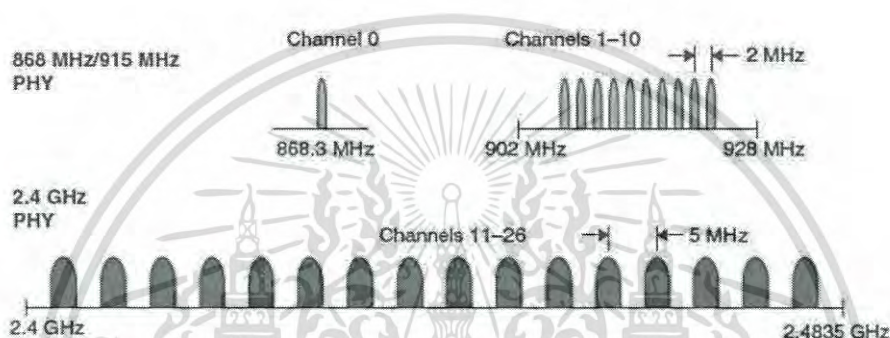


รูปที่ 2.7 องค์ประกอบของ Zigbee Protocol Stack [13]

จากรูปที่ 2.7 แสดงองค์ประกอบของ Zigbee Protocol Stack โดยประกอบด้วย Application Layer และ Zigbee Network Layer ซึ่งสร้างขึ้นตาม Zigbee Specification และ MAC Layer กับ Physical Layer ซึ่งเป็นมาตรฐานเดียวกันกับ IEEE 802.15.4

2.6.3 Zigbee Protocol Stack: Physical Layer

ในระดับ Physical Layer ของ Zigbee นั้นเป็นมาตรฐานเดียวกันกับ IEEE 802.15.4 ซึ่งมีรายละเอียดดังรูปที่ 2.9



รูปที่ 2.8 รายละเอียดของ Physical Layer ตามมาตรฐาน IEEE 802.15.4 [14]

จากรูป 2.8 [14] แสดงคลื่นความถี่ที่ Zigbee ใช้นั้นเป็นคลื่นความถี่ในช่วง Industrial Sciences Medicine Band (ISM) 3 ช่วงคลื่น คือ 868 MHz, 915 MHz, 2.4 GHz ด้วยวิธีการส่งแบบ Direct Sequence Spread Spectrum (DSSS) ทำให้มีคุณสมบัติทนทานต่อ Noise เหมาะสำหรับสถานที่ที่มีความถี่รบกวนสูง โดยแต่ละช่วงจะได้อัตราเร็วในการรับส่งข้อมูล ดังนี้

1. ความถี่ 868 MHz Modulation แบบ Binary Phase-shift Keying อัตรารับส่งข้อมูล 20 kb/s
2. ความถี่ 915 MHz Modulation แบบ Binary Phase-shift Keying อัตรารับส่งข้อมูล 40 kb/s
3. ความถี่ 2.4 GHz Modulation แบบ Quadrature Phase-shift Keying อัตรารับส่งข้อมูล 250 kb/s

2.6.4 Zigbee Protocol Stack: MAC Layer

2.6.4.1 Device Class

อุปกรณ์ Zigbee แบ่ง Device Class ไว้เป็น 2 ประเภท ได้แก่

- 1) Full Function Device (FFD) ซึ่งมีคุณสมบัติคือ

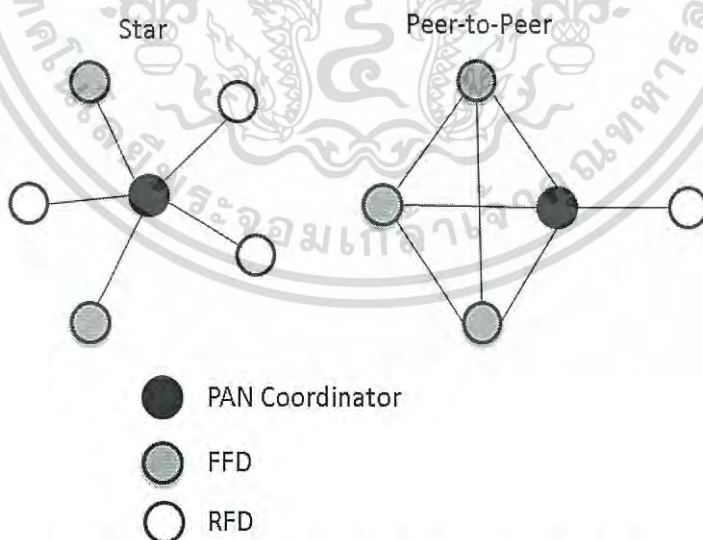
1. สามารถทำการ Routing เพื่อหาเส้นทางสำหรับส่งข้อมูลได้ ใช้งานได้ในทุกรูปแบบ Topology
2. ติดต่อกับอุปกรณ์ Zigbee โหนดอื่นได้ทั้งประเภท FFD และ RFD
3. สามารถเป็น PAN Coordinator ซึ่งเป็นศูนย์กลางของเครือข่ายไร้สาย Zigbee โดยจะทำหน้าที่จัดตั้ง Zigbee Network ขึ้นมาพร้อมทั้งเก็บสารสนเทศของ Zigbee Network นั้น เช่น Key ที่ใช้ในการเข้ารหัสข้อมูลภายใน Zigbee Network นี้เป็นต้น โดยในหนึ่ง Zigbee Network จะมี Coordinator นี้เพียงหนึ่งโหนดเท่านั้น

2) Reduced Function Device (RFD)

1. ไม่สามารถหาเส้นทางให้ Zigbee Packet ได้จึงใช้งานได้แบบเฉพาะ Peer-to-Peer Topology เท่านั้น
2. เป็น PAN Coordinator ไม่ได้
3. เป็นได้เฉพาะ End Node ใน Zigbee Network

2.6.4.2 Topology

ในการทำงานแบบเครือข่ายของ Zigbee รูปแบบของ Topology ที่ใช้เป็นมาตรฐานในการทำงานของ Zigbee Network ตามมาตรฐาน 802.15.4 คือ Star Topology และ Peer-to-Peer Topology แสดงได้ดังรูปที่ 2.9

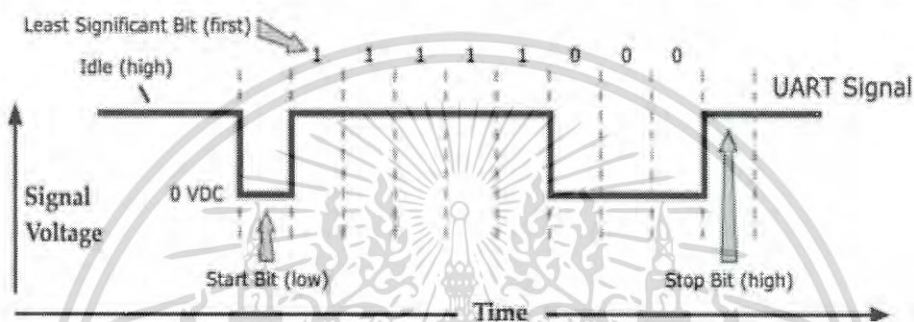


รูปที่ 2.9 ภาพ Topology ใ้การใช้งาน Zigbee Network [15]

จากการที่ Zigbee รองรับ Topology ที่หลากหลายนี้เองทำให้เหมาะสมและยืดหยุ่นในการนำไปใช้งานเป็นเครือข่ายระยะใกล้ในการส่งข้อมูล

2.6.5 การส่งข้อมูลแบบ Serial ด้วย Zigbee Module

Zigbee Module ที่เลือกใช้สามารถทำหน้าที่เป็น DCE (Data Communication Equipment) คือ เป็นอุปกรณ์ที่ทำหน้าที่สื่อสาร ซึ่งอาจจะทำหน้าที่แปลงสัญญาณ หรือเป็นตัวส่งต่อสัญญาณ เพื่อให้ DTE (Data Terminating Equipment) ที่เป็นอุปกรณ์ที่ทำหน้าที่เป็นแหล่งกำเนิดสัญญาณ หรือ รับ/ส่งสัญญาณ (อุปกรณ์ ต้นทาง หรือ ปลายทาง) ทั้งสองฝั่งสามารถติดต่อกันได้ เสมือนกับว่า DTE ทั้งสองฝั่งติดต่อกันด้วยสาย Serial โดยตรงนั่นเอง ดังรูปที่ 2.10

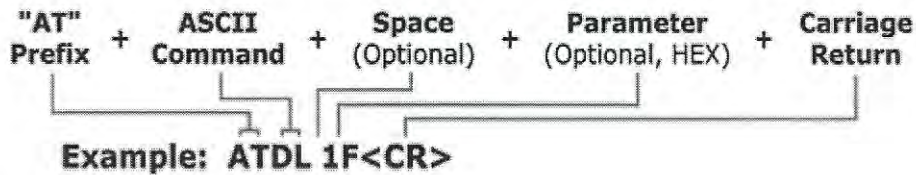


รูปที่ 2.10 UART Data Packet ของข้อมูล 0x1F (Decimal Number “31”) [16]

จากรูปที่ 2.10 แสดง UART Data Packet ที่ถูกส่งเข้ามายัง UART Module เริ่มต้นที่ สัญญาณ Idle มีค่าเป็น High จากนั้นจะเป็น Start Bit จำนวน 1 Bit ตามด้วยข้อมูล จำนวน 8 Bit ปิดท้ายด้วย Stop Bit จำนวน 1 Bit ซึ่งเป็นรูปแบบที่ได้กำหนดไว้ในกาส่งข้อมูลคือ 8-N-1 คือ 8 Bit Data – Non Parity Bit – 1 Stop Bit ข้อมูลที่แสดงในตัวอย่าง เป็นข้อมูลที่ถูกส่งเข้ามาใน UART Module โดยมีค่าในระบบเลขฐานสิบคือ 31 เมื่อเขียนในรูปเลขฐานสอง คือ 0001 1111 ซึ่งตรงตามรูปแบบที่แสดงในตัวอย่างที่ส่งแบบ Least Significant Bit

Zigbee Module ที่ใช้ในงานวิจัยนี้ มีโหมดการทำงานสองโหมด คือ Transparent Mode และ API Mode ซึ่งสองโหมดนั้นจะมีรูปแบบการส่งข้อมูลและคำสั่งที่ใช้ติดต่อกับ Zigbee Module ต่างกันออกไป

ใน Transparent Mode นั้น Zigbee Module จะทำหน้าที่เป็น Serial Line Replacement นั่นคือ มีรูปแบบการใช้งานเหมือนกับการเชื่อมต่อด้วยสาย Serial ธรรมดา โดยใน Transparent Mode นี้เราสามารถสั่งงาน Zigbee Module ด้วย AT Command ผ่านทางโปรแกรม Terminal ต่างๆ ได้ ซึ่ง AT Command มีรูปแบบดังนี้



รูปที่ 2.11 รูปแบบของ AT Command [17]

จากรูปที่ 2.11 เป็นตัวอย่างการใช้งาน AT Command ซึ่งเป็นการเซตค่าของ Address ของ Zigbee Module ปลายทางที่ต้องการติดต่อด้วย ในส่วน Low Address ที่เป็น 1F ตัวอย่างของ AT Command อื่นที่สามารถเลือกใช้ได้ เช่น

(1) ATRR X โดย X เป็นมีค่าได้ตั้งแต่ 0 ถึง 6 ซึ่งคำสั่งนี้จะเป็นการกำหนดจำนวน ครั้งของ Xbee Retry ที่ต้องการให้ทำการลองส่งใหม่เมื่อเกิดการส่งผิดพลาดขึ้นที่ชั้น MAC Layer โดย Xbee Retry 1 ครั้งนั้น จะสั่งให้ทำการส่งข้อมูลใหม่ในชั้น MAC Layer 3 ครั้ง

(2) ATND เป็นการสั่งให้ทำการค้นหา Zigbee Module ซึ่งอยู่ภายในระยะที่สามารถรับส่งข้อมูลกันได้ โดยจะแสดงรายละเอียดเช่น Address แบบ 16 Bits เลข Serial Number ของ Zigbee Module นั้นระดับของสัญญาณที่ติดต่อกับ Node นั้นได้ เป็นต้น

(3) ATAP X โดย X มีค่าได้ตั้งแต่ 0 ถึง 2 คำสั่งนี้จะเป็นการสั่งให้ Zigbee ทำการ เปิด/ปิดการใช้ API Mode

สำหรับ API Mode (Application Programming Interface Mode) ทั้งข้อมูลและคำสั่ง จะอยู่ในรูปแบบ Frame-Based ทั้งหมด โดย API Frame จะมีรูปแบบดังรูปที่ 2.12



รูปที่ 2.12 รูปแบบของ API Frame. [18]

Byte 1 เป็น Start Delimiter บอกการเริ่มต้นของ Frame มีค่าเป็น 0x7E เสมอ Byte 2 ถึง 3 เป็นจำนวน Byte ของส่วน Frame Data Byte 4 ถึง n เป็นส่วนคำสั่งและพารามิเตอร์ หรือเป็นข้อมูล Byte n+1 ใช้เป็น checksum

ตัวอย่าง ของ API Frame คำสั่ง ATND สำหรับค้นหาโหมดซึ่งอยู่ภายในรีซีมี เมื่ออยู่ในรูปแบบ API Command จะมีรูปแบบ 0x7E 0x00 0x04 0x08 0x00 0x4E 0x44 0x64 โดยมีความหมายคือ

0x7E	เป็น Start Delimiter
0x04	เป็น Length
0x08	เป็นการระบุว่าเป็น AT Command
0x01	เป็นการระบุว่าเป็นคำสั่งชนิดที่จะมีผลการทำงานตอบกลับมา
0x4E 0x44	เป็นคำสั่ง ND ในรูปแบบ Hexadecimal
0x64	เป็น Checksum

2.6.6 Zigbee Security

การรักษาความปลอดภัยของ Zigbee นั้นจะใช้การเข้ารหัสแบบ 128-bit AES เป็นหลักโดยจะมีศูนย์กลางในการรักษาความปลอดภัยของ Zigbee Network อยู่ที่ Coordinator Node โดยเป็น Trust Center ซึ่งทำหน้าที่จัดการด้านความปลอดภัยในด้านต่างๆ ดังนี้

Trust Manager คือ ตรวจสอบว่าอุปกรณ์ที่ร้องขอเข้าใน Zigbee Network นั้นมีสิทธิในการเข้าใช้เครือข่ายหรือไม่

Network Manager คือ แจกจ่าย Network Key ให้สมาชิกใน Zigbee Network เพื่อให้ทุก Node มี Key ร่วมกัน ซึ่งทำให้การเข้ารหัสเป็นแบบ Group Encryption คือ ทุก Node Configuration Manager คือการสร้างความปลอดภัยในการส่งข้อมูลในรูปแบบระหว่าง Node ต่อ Node ใน Network โดยการใช้ Link Key

สำหรับ Key ที่ใช้ในระบบ Zigbee นั้น มี 3 ประเภทได้แก่ Master Key Network Key และ Link Key ซึ่งมีวิธีใช้แตกต่างกันออกไป ดังนี้

(1) Master Keys เป็น Key ร่วมเพื่อใช้สร้าง Link Keys ที่จะใช้เข้ารหัสระหว่าง Zigbee 2 Nodes

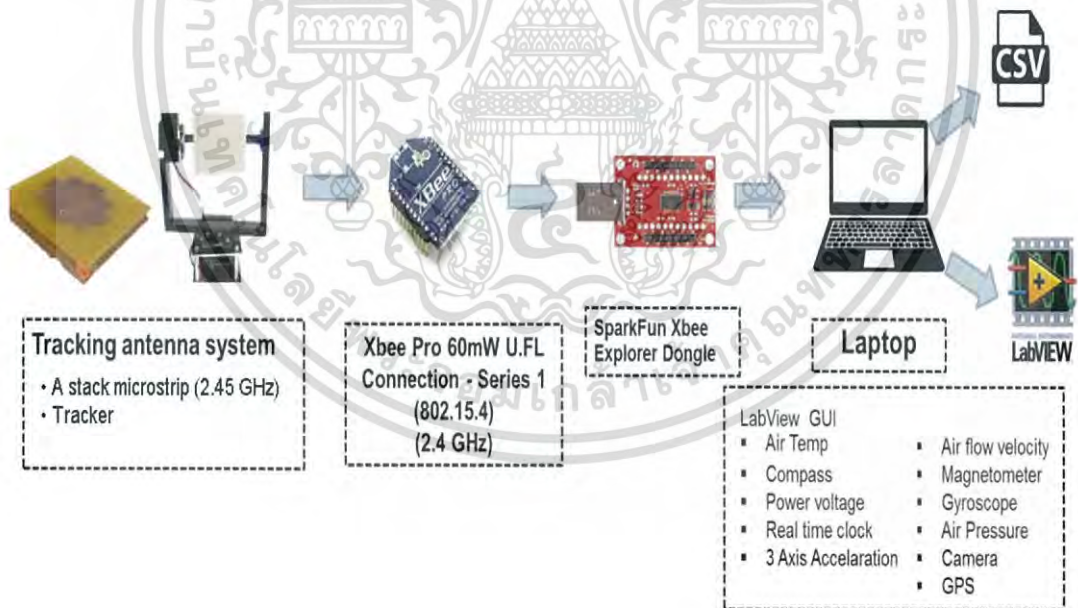
(2) Network Keys เป็น Key ที่ใช้เข้ารหัสข้อมูลในระดับ Network Layer ของ Zigbee Network โดย Zigbee ทุก Node จะใช้ Key เดียวกัน จุดประสงค์ของ Network Keys เพื่อใช้ ปกป้องความลับของ Zigbee Network จากภายนอก

(3) Link Keys เป็น key ที่ใช้ในการเข้ารหัสเพื่อส่งข้อมูล โดยจะใช้ในการรักษาความลับของ การส่งข้อมูลระหว่าง Node 2 Node ซึ่งอยู่ภายในเครือข่าย Zigbee Network เดียวกัน

บทที่ 3

การออกแบบและการจัดทำปฏิญานิพนธ์

ปฏิญานิพนธ์นี้นำเสนอในส่วนของภาคพื้นดินหรือภาครับ โดยมีหน้าที่รับข้อมูลจากภาคส่งโดยใช้สายอากาศ Stack microstrip ที่ทำขึ้นเองในการรับข้อมูล ซึ่งสายอากาศที่ทำขึ้นเองนั้นจะถูกติดตั้งอยู่บนระบบติดตามดาวเทียมขนาดขนาดเล็กเพื่อช่วยในเรื่องของความสะดวกสบายและกำลังในการรับข้อมูลสูงที่สุด โดยสายอากาศจะถูกใช้กับโมดูล xbee ที่เป็นอุปกรณ์ที่ทำหน้าที่สื่อสารระหว่างอุปกรณ์ภาคส่งและสถานีภาคพื้นดิน จากนั้นข้อมูลจะถูกส่งต่อไปให้คอมพิวเตอร์เพื่อนำค่าที่ได้ไปประมวลผลผ่านโปรแกรม LabVIEW ที่ทำการออกแบบไว้ ค่าที่ถูกประมวลผลนั้นจะถูกไปแสดงในแต่ละส่วนต่าง ๆ ภายในโปรแกรมเพื่อความสะดวกในการวิเคราะห์ข้อมูล ซึ่งข้อมูลเหล่านั้นจะถูกแสดงรูปแบบของกราฟ รวมไปถึงมีการระบุตำแหน่งของ CanSat โดยใช้ข้อมูลจาก GPS ข้อมูลทั้งหมดที่กล่าวมานั้นจะแสดงผลอยู่บนหน้าจอคอมพิวเตอร์เพียงอย่างเดียว โดยมีบล็อกไดอะแกรมภาพรวมการทำงานแสดงดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมภาพรวมการทำงาน

3.1 การออกแบบ

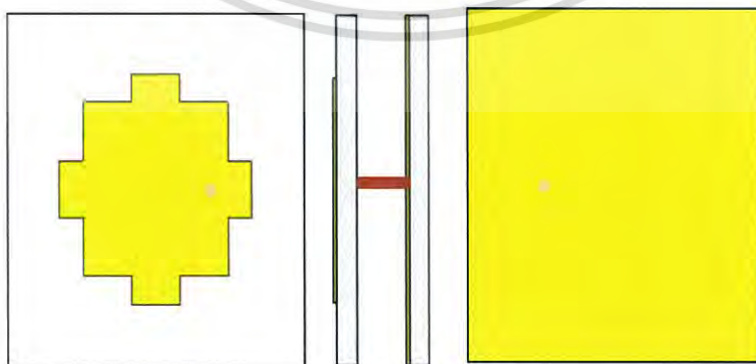
การออกแบบเป็นการแสดงขั้นตอนกระบวนการคิดในวางแผนหรือจำลองรูปแบบโครงสร้าง หรือการทำงาน ก่อนที่จะลงมือทำจริงซึ่งในส่วนของปริญญาบัตรนี้การออกแบบจะถูกแบ่งออกเป็นทั้งหมด 3 ส่วนซึ่งได้แก่

- 1). การออกแบบและสร้างสายอากาศ
- 2). การออกแบบระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก
- 3). การออกแบบโปรแกรมที่ใช้ในการแสดงผลข้อมูลด้วยโปรแกรม LabVIEW

3.1.1 การออกแบบและสร้างสายอากาศ

3.1.1.1 ข้อมูลเบื้องต้น (Introduction) [19]

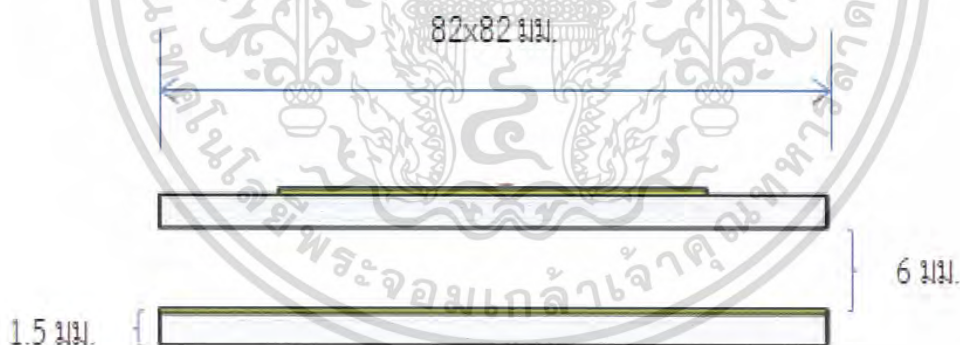
โครงสร้างของสายอากาศ Stack microstrip ที่นำเสนอนี้ สร้างจากแผ่นพิมพ์ชนิด FR4 มีค่าค่าคงที่ของอนุวน (dielectric's relative permittivity) $\epsilon_r = 4.3$ มีค่าความหนาของวัสดุฐานรอง $h = 1.5$ มม. และค่า การสูญเสียแทนเจนท์ (Loss Tangent, $\tan \delta$) = 0.02 มี จำนวน 2 ชั้นด้วยกันประกอบด้วย ชั้นล่างสุดจนถึงชั้นบนสุด ดังนี้คือ ชั้น Ground plane, และชั้น Microstrip line, ซึ่งทั้งสองชั้นมีขนาดเท่ากันหมดคือ 82×82 มม. ดังรูปที่ 3.2 มีระยะห่างระหว่างชั้น Ground plane กับชั้น Microstrip line เท่ากับ 7.5 มม. มีการป้อนสัญญาณทางเข้าที่หัว SMA ที่ชั้น Ground Plane เชื่อมต่อไปชั้น Microstrip line มีค่าอิมพีแดนซ์ทาง อินพุต 50 โอห์ม สำหรับรายละเอียดในแต่ละชั้นมีดังนี้ ชั้น Ground plane เป็นแผ่นวงจรพิมพ์หน้าเดียว ที่มีลายทองแดงเต็มแผ่นส่วนชั้น Microstrip line มีลายทองแดงที่หน้าแผ่นวงจรพิมพ์ ลักษณะคล้ายกับเครื่องหมายบวกประกบกับรูปสี่เหลี่ยมจตุรัสที่อยู่ตรงกลางแผ่น Microstrip line ผลของการตัดมุมที่ตัวแพตช์ จะช่วยให้การแพร่กำลังงานในแนว โพลาริซแบบชี้ทิศทางทำได้ดีขึ้นอีกด้วย โดยจะมีความยาวและระยะห่างที่เท่ากันทุกมุมด้าน



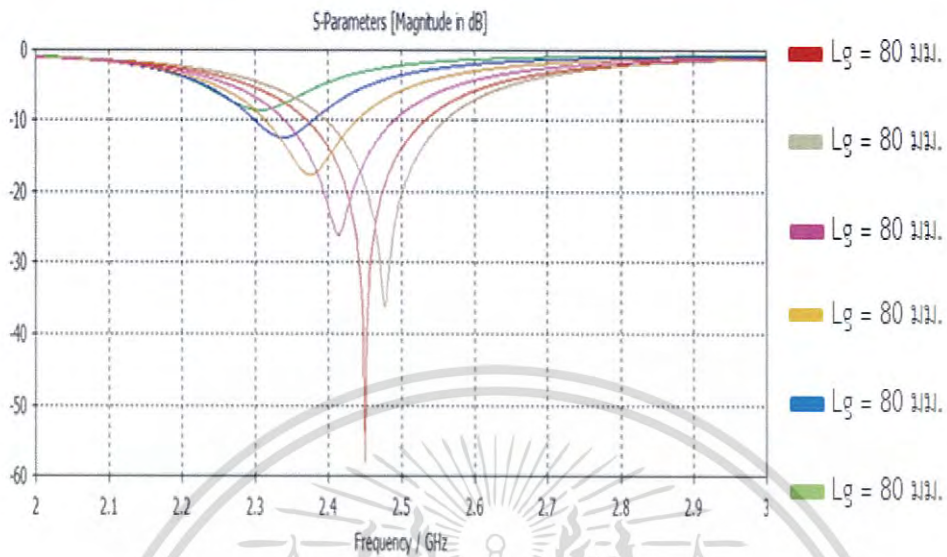
รูปที่ 3.2 โครงสร้างสายอากาศ Stack microstrip (ด้านหน้า-ด้านข้าง-ด้านหลัง)

3.1.2.2 ผลการจำลองการทำงานสายอากาศ [19]

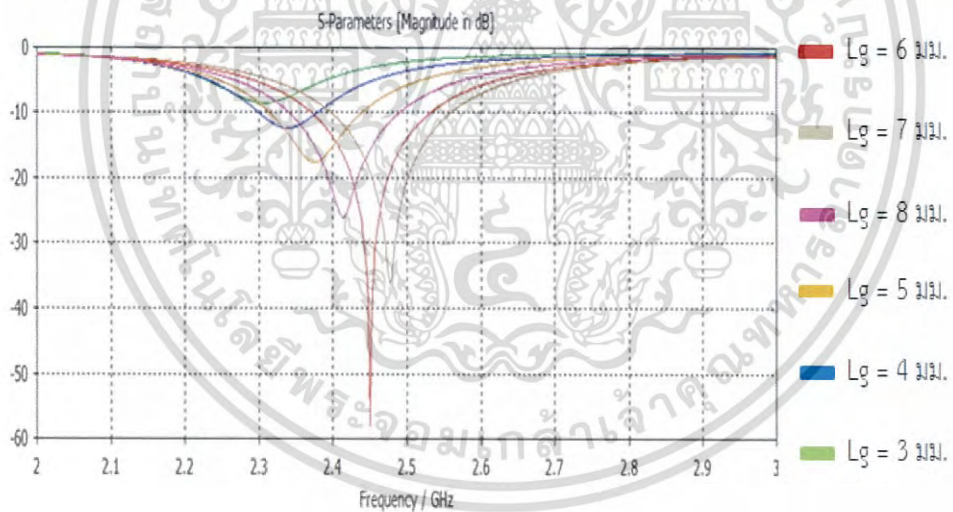
เมื่อนำโครงสร้างของสายอากาศมาทำการออกแบบและจำลองการทำงานด้วยโปรแกรมจำลองการทำงานสายอากาศ พบว่า ผลการปรับขนาดความกว้างพื้นที่ระนาบกราวด์ L_g บนแผ่น Ground plane ตั้งแต่ 50-100 มม. ผลของการปรับ L_g ที่ 50 มม. จะมีผลต่อค่าการสูญเสียย้อนกลับ (Return Loss) หรือ $[S_{11}]$ น้อยกว่ามากกว่า -10 dB ซึ่งในระดับนี้สายอากาศไม่สามารถทำงานได้ การปรับ L_g ที่ 60-100 มม. จะมีผลต่อค่า S-Parameter $[S_{11}]$ ที่ดีขึ้น เมื่อทำการพิจารณาแล้วพบว่าค่าที่ดีที่สุดคือค่า $L_g=80$ มม. ซึ่งสายอากาศที่ทำการออกแบบเสร็จแล้วจะมีขนาดด้านต่างๆตามรูปที่ 3.3 ซึ่งมีค่าในช่วงแบนด์-วิดธ์ 2.37-2.53 GHz ที่ -57.97 dB ทำให้แรงดันของคลื่นที่สะท้อนกลับออกมา มีค่าน้อยกว่าแรงดันที่ป้อนเข้าไปโดยปกติคือต้องน้อยกว่า -10 dB ดังแสดงในรูปที่ 3.4 และผลการปรับระยะห่างระหว่างแผ่นพิมพ์สองชั้น L_d ตั้งแต่ 3-8 มม. ผลของการปรับระยะห่างระหว่างแผ่นพิมพ์สองชั้นมากขึ้น จะมีผลให้ค่าอัตราขยายของสายอากาศมากขึ้น และเมื่อพิจารณาค่าอัตราขยายของสายอากาศพบว่าค่าที่ดีที่สุดคือค่า $L_d=6$ มม. ซึ่งส่งผลให้ค่าอัตราขยายของสายอากาศในช่วงแบนด์วิดธ์ 2.45 GHz ดีที่สุด ดังแสดงในรูปที่ 3.5 และปรับขนาดของลายทองแดงที่อยู่บนแผ่นพิมพ์ชั้นบน หรือ Microstrip line มีลักษณะแบ่งออกเป็นสองลักษณะคล้ายกับเครื่องหมายบวก D_o ประกอบกับรูปสี่เหลี่ยมจตุรัส D_s โดยแต่ละลักษณะทุกมุมด้านจะมีขนาดเท่ากัน



รูปที่ 3.3 ผลของการปรับระยะห่างระหว่างแผ่นพิมพ์สองชั้น

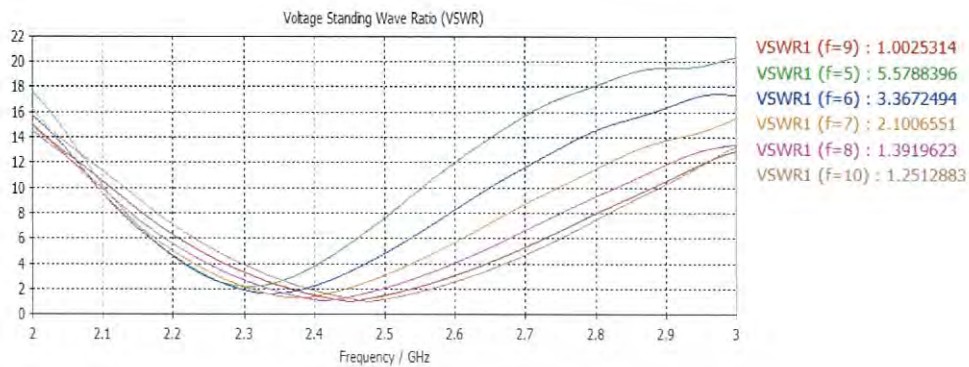


รูปที่ 3.4 ผลการปรับขนาดความกว้างพื้นที่ระนาบกราวด์ L_g บนแผ่น Ground plane



รูปที่ 3.5 ผลของการปรับระยะห่างระหว่างแผ่นพิมพ์สองชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



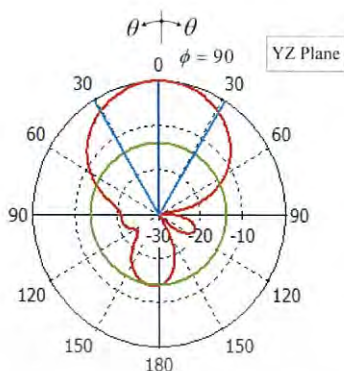
รูปที่ 3.6 ผลการปรับค่า VSWR (Voltage Standing Wave Ratio)

ค่า VSWR (Voltage Standing Wave Ratio) เป็นอัตราส่วนความสูญเสียระหว่างแรงดันไฟฟ้าที่ส่งออกไปและได้มีการสะท้อนกลับมา ซึ่งเกิดจากสาเหตุหลายประการ เช่น เกิดจากสายนำสัญญาณ ขั้วต่อต่างๆ ค่าSWRจะเป็นตัวบอกว่า สายอากาศแมตซ์กับสายนำสัญญาณได้ดีเพียงใด ตามมาตรฐานของญี่ปุ่นนั้นจะให้ไม่เกิน 1:1.7 ส่วนของอเมริกาจะให้ไม่เกิน 1:2 จากรูปที่ 3.6 จากการปรับค่าพารามิเตอร์และพิจารณาค่าต่างๆแล้วพบว่า VSWR ที่ได้จากการจำลองสายอากาศด้วยโปรแกรมมีอัตราส่วนอยู่ที่ 1:1.0025 ซึ่งแมตซ์กับสายอากาศที่ใช้ในย่านความถี่กลาง 2.45 GHz

ตารางที่ 3.1 ค่าพารามิเตอร์ต่างๆ ของสายอากาศที่ได้หลังจากปรับแต่งสายอากาศ

พารามิเตอร์	ขนาด (มม.)
L_g	82
L_d	6
D_p	20
D_s	6.58

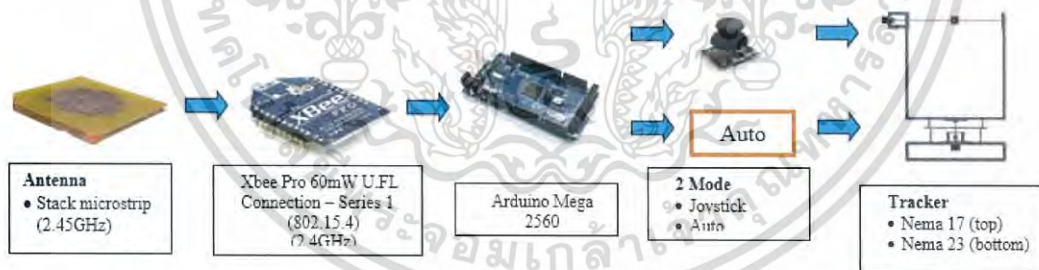
หลังจากที่ได้ทำการปรับแต่งค่าพารามิเตอร์ต่างๆ จนให้ได้คุณสมบัติทางสายอากาศที่มีค่าการสูญเสียย้อนกลับ (Return Loss) น้อยกว่า -10 dB และค่าอัตราส่วนแกน (AR) น้อยกว่า 3 dB ตลอดช่วงแถบความถี่พิจารณา 2.37-2.53 GHz ซึ่งผลสรุปค่าพารามิเตอร์ต่างๆ ที่ได้หลังจากผ่านการปรับแต่งสายอากาศจนได้ค่าที่ดีที่สุดแล้ว ดังแสดงในตารางที่ 3.1 รูปที่ 3.7 แสดงผลการจำลองแบบรูปการแผ่กำลังงานในระนาบ YZ (YZ-Plane) ซึ่งผลการจำลองการทำงานสายอากาศที่นำเสนอ พบว่าสายอากาศมีทิศทางการแผ่กำลังงานในทิศทางเดียวและมีอัตราการขยายของสายอากาศที่ 8.31 dBi



รูปที่ 3.7 ผลการจำลองแบบรูปการแพร่กระจายคลื่นในระนาบ YZ (YZ-Plane)

3.1.2 การออกแบบระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก

ระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็กเป็นระบบที่ทำหน้าที่ช่วยให้สายอากาศทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้นเนื่องจากสายอากาศที่สร้างขึ้นนั้นมีลักษณะการแพร่กระจายคลื่นเป็นแบบ unidirectional ซึ่งระบบติดตามแบบจำลองดาวเทียมขนาดเล็กจะมีหน้าที่เล็งทิศทางของสายอากาศไปยังเป้าหมายได้แม่นยำรวมถึงช่วยในเรื่องการอำนวยความสะดวกสบายให้กับผู้ใช้โดยที่ระบบจะใช้ค่าที่ได้จาก GPS และความสูงเพื่อคำนวณหามุมในแนวแกนตั้งกับมุมในแนวระดับ พร้อมทั้งแสดงผลของข้อมูลผ่านจาก LCD ได้อีกด้วย ดังแสดงในรูปที่ 3.8

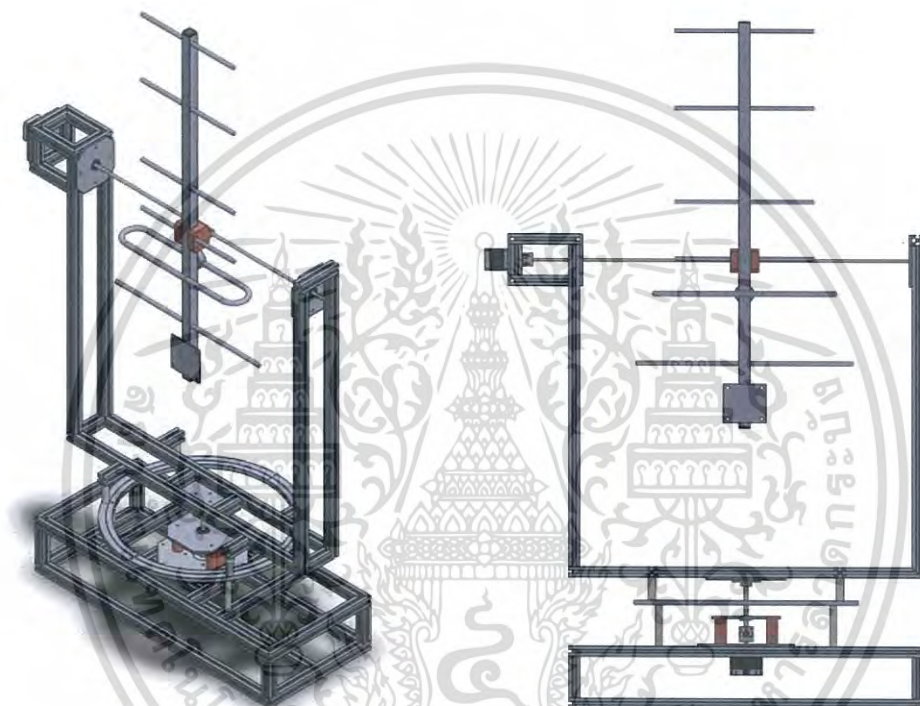


รูปที่ 3.8 บล็อกไดอะแกรมระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก

3.1.2.1 การออกแบบโครงสร้างด้วยโปรแกรม Solidwork

เนื่องจากต้องการให้โครงสร้างของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็กสามารถประยุกต์ใช้ได้กับหลากหลายสายอากาศจึงออกแบบให้ตัวโครงสร้างนั้นมีความกว้างและความยาวรองรับสายอากาศที่มีอยู่ทั้งหมดโดยจะมีสายอากาศที่ใหญ่ที่สุดคือ สายอากาศยาก็-อูตะ ที่มีความถี่กลางที่ 433 MHz (ที่อยู่ในรูปที่ 3.9) และเพื่อให้สามารถหมุนเพื่อติดตามแบบจำลองดาวเทียมขนาดเล็กทั้งแนวแกนตั้งและแนวแกนนอนนั้นการออกแบบจึง

มีมอเตอร์สำหรับหมุนแต่ละแนวแกนอย่างละ 1 ตัวจึงมีมอเตอร์ที่ใช้ทั้งหมด 2 ตัว โดยตัวแรกคือ Stepper Motor NEMA 17 [20] (ตัวที่อยู่ด้านบน) จะนำไปใช้ในการหมุนตามแนวแกนตั้งและตัวที่สองคือ Stepper Motor NEMA 23 [20] (ตัวที่อยู่ด้านล่าง) จะนำไปใช้ในการหมุนตามแนวแกนนอน โดยจะมีโครงสร้างที่ออกแบบได้แสดงดังรูปที่ 3.9 โดยจะมีขนาดและน้ำหนักเมื่อไม่ได้คิดรวมกับสายอากาศแสดงดังตามตารางที่ 3.2



รูปที่ 3.9 โครงสร้างของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก

ตารางที่ 3.2 ขนาดและน้ำหนักของโครงสร้างของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก

ความกว้าง	ความยาว	น้ำหนัก (โดยประมาณ)
77.43 cm	91.70 cm	4.7 kg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งโครงสร้างที่ทำการออกแบบนั้นทางผู้จัดทำได้ตัดสินใจเลือกใช้อลูมิเนียมโปรไฟล์ (Aluminium Profile) [21] ที่มีลักษณะแสดงดังรูปที่ 3.10 เป็นวัสดุในการสร้างโครงสร้างหลักเนื่องจากมีคุณสมบัติที่ดีกว่าโลหะทั่วไปคือ เบาแต่แข็งแรง ,ทนทานต่อการสึกกร่อน , ไม่เป็นสนิม ,มีความสามารถในการรับแรงกดได้สูง ,สามารถถอดประกอบและต่อเติมภายหลังได้สะดวก , ไม่เป็นพิษและไม่ติดไฟ ,สามารถนำไปประยุกต์ใช้งานต่างๆได้มากมาย ,สามารถนำไปประกอบ ตัด เจาะ ตัด บีม ได้ตามต้องการ ,และสามารถเชื่อมต่อได้ด้วยวิธีการที่หลากหลาย



รูปที่ 3.10 อลูมิเนียมโปรไฟล์

3.1.2.2 การคำนวณและวิธีการเลือกอุปกรณ์

ทำการคำนวณหาค่า torque ได้จากสูตร

$$\text{Torque} = F \times R \quad (\text{N.cm}) \quad (3.1)$$

โดยที่ F คือ แรงในหน่วยนิวตัน (N) และ R คือ ระยะจากจุด center (cm)

โดยการใช้สูตรนี้นั้นจะแบ่งเป็น 2 ส่วนโดยจุด center อยู่ที่ center ของแกนเพลลาของ Stepper Motor NEMA 23 (ตัวที่อยู่ด้านล่าง) แบ่งคำนวณที่ละฝั่ง ในด้านฝั่งซ้ายนั้นจะมี

โครงซี่ก้าน	1.2	kg
สายอากาศ	0.65	kg
Stepper Motor NEMA 17 with gear bore	1.5	kg
อื่นๆ	0.5	kg
สุทธิ	3.85	kg

ดังนั้นจะได้ $F = 38.5 \text{ N}$ และในการหาค่า R นั้นต้องมีระยะจาก center ของเพลลา ระยะนี้เกิดจากจุด centroid ของฝั่ช้ายโดยประมาณไว้ที่ 40 cm สรุปคือจะได้ค่า torque ฝั่ช้ายจากการคำนวณเท่ากับ 38.5×40 เท่ากับ 1540 N.cm ในฝั่ชทางด้านขวานั้นก็ใช้หลักการเดียวกันโดยจะมี

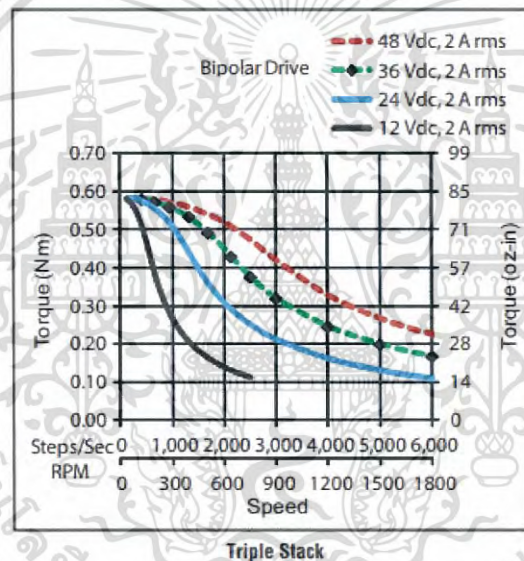
โครงซี่กษา	1	kg
สายอากาศ	0.65	kg
อื่นๆ	0.5	kg
สุทธิ	2.15	kg

ดังนั้นจะได้ $F = 21.5 \text{ N}$ และในการหาค่า R นั้นต้องมีระยะจาก center ของเพลลา ระยะนี้เกิดจากจุด centroid ของฝั่ชขวาโดยประมาณไว้ที่ 20 cm สรุปคือจะได้ค่า torque ฝั่ชขวา จากการคำนวณเท่ากับ 21.5×20 เท่ากับ 430 N.cm เมื่อทำการรวมทั้งสองฝั่ชจะได้ค่า torque สุทธิเท่ากับ 1970 N.cm และเนื่องจากถ้ากระแสเพิ่มขึ้นเนื่องจากโหลด ความสามารถในการรับกระแสของอุปกรณ์ก็ต้องมากขึ้นด้วยจึงต้องเพิ่มขั้ของ motor ไว้ประมาณ 2 เท่าของโหลด แนวแกน ดังนั้นก็จะได้ค่า torque คือ 2 เท่าของโหลดแนวแกนคือจะประมาณ 4000 N.cm แล้วถ้าการออกแบบนั้นเราต้องการความเร็วอยู่ที่ 10 rpm จึงได้ทำการออกแบบการทดเฟืองของมอเตอร์อยู่ที่ $1 : 30$ ซึ่งหมายความว่าความเร็วของมอเตอร์ที่ได้จะอยู่ที่ 300 rpm และเมื่อเทียบจากกราฟในรูปที่ 13 แล้วจะเห็นได้ว่าค่า torque ที่ได้อยู่ที่ 1.75 N.m ซึ่งจากการคำนวณนั้นค่า torque ของเรา อยู่ที่ 4000 N.cm หรือ 40 N.m และทำการทดเฟืองไปอีก $1 : 30$ จะทำให้ได้ค่า torque สุทธิอยู่ที่ 1.33 N.m ซึ่งไม่เกินจากค่าที่ได้จากกราฟจึงหมายความว่ามอเตอร์ที่เราทำการเลือกนั้นสามารถรองรับ torque ที่เราออกแบบไว้ได้

โดยมอเตอร์ตัวแรกที่เราเลือกใช้คือ Stepper Motor NEMA 17 (ตัวที่อยู่ด้านบนของรูปที่ 3.9) จะนำไปใช้ในการหมุนตามแนวแกนตั้งโดยจะมีรูปร่างดังรูปที่ 3.11 โดยจะมี ประสิทธิภาพของ Stepper Motor NEMA 17 เมื่อใช้งานต่อเนื่องเทียบกันแต่ละแรงดันไฟฟ้าแสดงดังในรูปที่ 3.12



รูปที่ 3.11 Stepper Motor NEMA 17 [20]

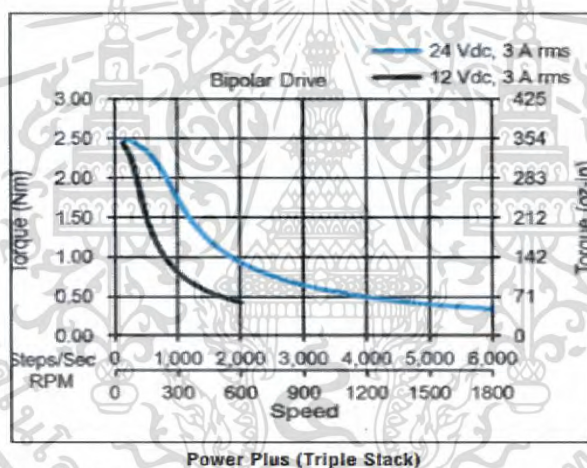


รูปที่ 3.12 ประสิทธิภาพของ Stepper Motor NEMA 17 เมื่อใช้งานต่อเนื่องเทียบกับแต่ละแรงดันไฟฟ้า [20]

ส่วนมอเตอร์ตัวที่สองนั้นคือ Stepper Motor NEMA 23 (ตัวที่อยู่ด้านล่างของรูปที่ 3.9) จะนำไปใช้ในการหมุนตามแนวแกนนอนจะมีรูปร่างดังรูปที่ 3.13 โดยจะมี ประสิทธิภาพของ Stepper Motor NEMA 23 เมื่อใช้งานต่อเนื่องเทียบกับแต่ละแรงดันไฟฟ้าแสดงดังในรูปที่ 3.14



รูปที่ 3.13 Stepper Motor NEMA 23 [20]



รูปที่ 3.14 ประสิทธิภาพของ Stepper Motor NEMA 23 เมื่อใช้งานต่อเนื่องเทียบกันแต่ละแรงดันไฟฟ้า [20]

3.1.2.3 การคำนวณมุมในการหมุนของสเต็ปเปอร์มอเตอร์

1) วิธีการในการคำนวณมุมในแนวแกนนอน

ในการควบคุมมอเตอร์ที่ใช้ในการติดตามแบบจำลองดาวเทียมขนาดเล็กลักษณะนี้ ใช้วิธีการหามุมที่จะมาสั่งให้มอเตอร์หมุนตามแนวแกนนอนจากค่า GPS Latitude และ GPS Longitude ซึ่งเป็นข้อมูลที่จะได้มาจากฝั่งส่ง โดยใช้หลักการในการหาค่ามุมองศาที่จะควบคุมให้มอเตอร์หมุนตามสมการที่ 3.2 ถึง 3.4 [22]

$$\beta = \arctan2(X, Y) \quad (3.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X = \cos(B) \times \sin(C - D) \quad (3.3)$$

$$Y = (\cos(A) \times \sin(B)) - (\sin(A) \times \cos(B) \times \cos(D - C)) \quad (3.4)$$

- โดย β คือมุมในแนวระนาบระหว่างตำแหน่งอ้างอิงกับตำแหน่งที่ต้องการจะชี้ไป
 A คือ Latitude ของตำแหน่งอ้างอิง
 B คือ Latitude ของตำแหน่งที่ต้องการจะชี้ไป
 C คือ Longitude ของตำแหน่งอ้างอิง
 D คือ Longitude ของตำแหน่งที่ต้องการจะชี้ไป

ตัวอย่างการคำนวณ ให้ตำแหน่งอ้างอิงคือ latitude = 8.847099 , longitude = -9.730721 ตำแหน่งที่ต้องการชี้ latitude = 8.847139 longitude = -9.730673 จะได้

$$X = \cos(8.847139) \times \sin(-9.730673 - (-9.730721))$$

$$X = 8.277905 \times 10^{-7}$$

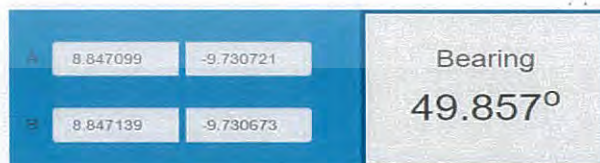
$$Y = (\cos(8.847099) \times \sin(8.847139)) - (\sin(8.847099) \times \cos(8.847139) \times \cos(-9.730673 - (-9.730721)))$$

$$Y = 6.981317 \times 10^{-7}$$

$$\beta = \arctan2(8.27 \times 10^{-7}, 6.98 \times 10^{-7})$$

$$\beta = 49.8568$$

จากนั้นทำการใช้โปรแกรม simulator ในการคำนวณหาค่าของมุมในแนวแกนนอน ดังรูปที่ 3.15 ผลที่ได้คือค่าของมุมที่ได้นั้นมีค่าเท่ากัน



รูปที่ 3.15 การคำนวณค่าของมุมจากโปรแกรม simulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) วิธีการในการคำนวณมุมในแนวแกนตั้ง

การควบคุมมอเตอร์ให้หมุนในแนวแกนตั้งนั้นจะใช้ค่าความสูงและระยะห่างระหว่างสองตำแหน่งในการคำนวณมุมที่จะคอนโทรลให้มอเตอร์หมุนดังแสดงในรูปที่ 3.16 ซึ่งค่าความสูงจะเป็นข้อมูลที่ได้มาจากฝั่งส่งและค่าระยะห่างระหว่างสองตำแหน่งมาจากการคำนวณจากสมการที่ 3.5 ถึง 3.7 [23]

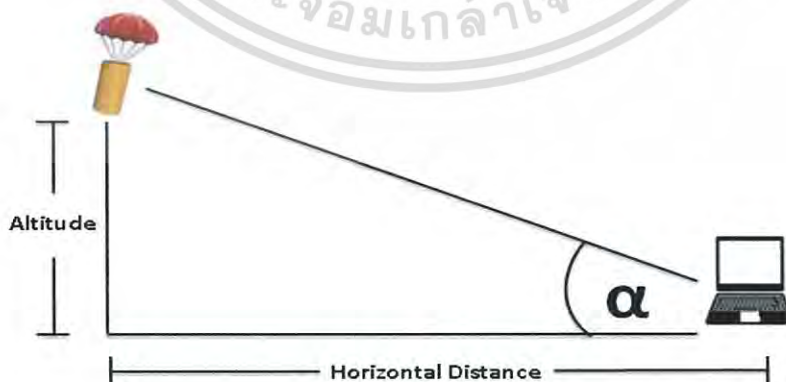
$$a = \left(\sin\left(\frac{dlat}{2}\right)\right)^2 + \cos(lat1) \times \cos(lat2) \times \left(\sin\left(\frac{dlon}{2}\right)\right)^2 \quad (3.5)$$

$$c = 2 \times \arctan2\left(\sqrt{a}, \sqrt{1-a}\right)^2 \quad (3.6)$$

$$d = R \times c \quad (3.7)$$

โดย lon 1 = Latitude ตำแหน่งที่ 1
lon 2 = Latitude ตำแหน่งที่ 2
lat 1 = Longitude ตำแหน่งที่ 1
lat 2 = Longitude ตำแหน่งที่ 2
dlon = lon2 - lon1
dlat = lat2 - lat1
R = รัศมีของโลก (6373 km)
d = ระยะห่างระหว่างสองตำแหน่ง (km)

หลังจากได้ค่าความสูงและระยะห่างระหว่างสองตำแหน่งมาแล้วก็จะทำการหาค่ามุมที่จะสั่งให้หมุนตามแนวแกนตั้งได้จากรูปที่ 3.15



รูปที่ 3.16 หลักการในการหามุมในแนวแกนตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจากรูปที่ 3.16 จะสามารถหาค่ามุมในแนวแกนตั้งได้โดยใช้สมการที่ 3.8

$$\alpha = \tan^{-1}\left(\frac{A}{D}\right) \quad (3.8)$$

โดย A (Altitude) คือ ความสูง (m)

D (Horizontal Distance) คือ ระยะห่างระหว่างสองตำแหน่ง (m)

α คือ มุมในแนวแกนตั้ง (องศา)

3.1.2.4 การเชื่อมต่อระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก

ในส่วนของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก จะมีส่วนแสดงผลของจอ TFT LCD เพิ่มเข้ามาเพื่อแสดงค่า GPS Latitude GPS Longitude และ GPS Altitude เพื่อใช้ในการระบุพิกัดตำแหน่งและเพื่อนำเอาค่ามาใช้ในการติดตามแบบจำลองดาวเทียมขนาดเล็ก โดยหน้าจอแสดงผลที่เลือกใช้ใช้นั้นคือ 3.2" HVGA LCD TFT Non-Touch Screen, Arduino MEGA-2560 Shield [24] ซึ่งเป็นหน้าจอ LCD ที่ความละเอียดอยู่ที่ 490 x 320 และมีทั้งหมด 262K สี โดยอุปกรณ์มีลักษณะเป็นรูปที่ 3.17



รูปที่ 3.17 อุปกรณ์ HVGA LCD TFT Non-Touch Screen, Arduino MEGA-2560 Shield

โดยในการใช้งานจริงนั้นเราจำเป็นที่จะต้องใส่สเต็ปเปอร์มอเตอร์คู่กับตัวขับเคลื่อนมอเตอร์ที่เป็นตัวกลางระหว่างมอเตอร์และตัวคอนโทรลเลอร์เพื่อใช้ในการควบคุมมอเตอร์ให้หมุนไปตามขนาดและทิศทางที่เราได้ทำการป้อนให้ไป ซึ่งตัวขับเคลื่อนมอเตอร์ที่เลือกใช้คือ TB6600 Stepper Motor Driver [25] ที่แสดงดังรูปที่ 3.18

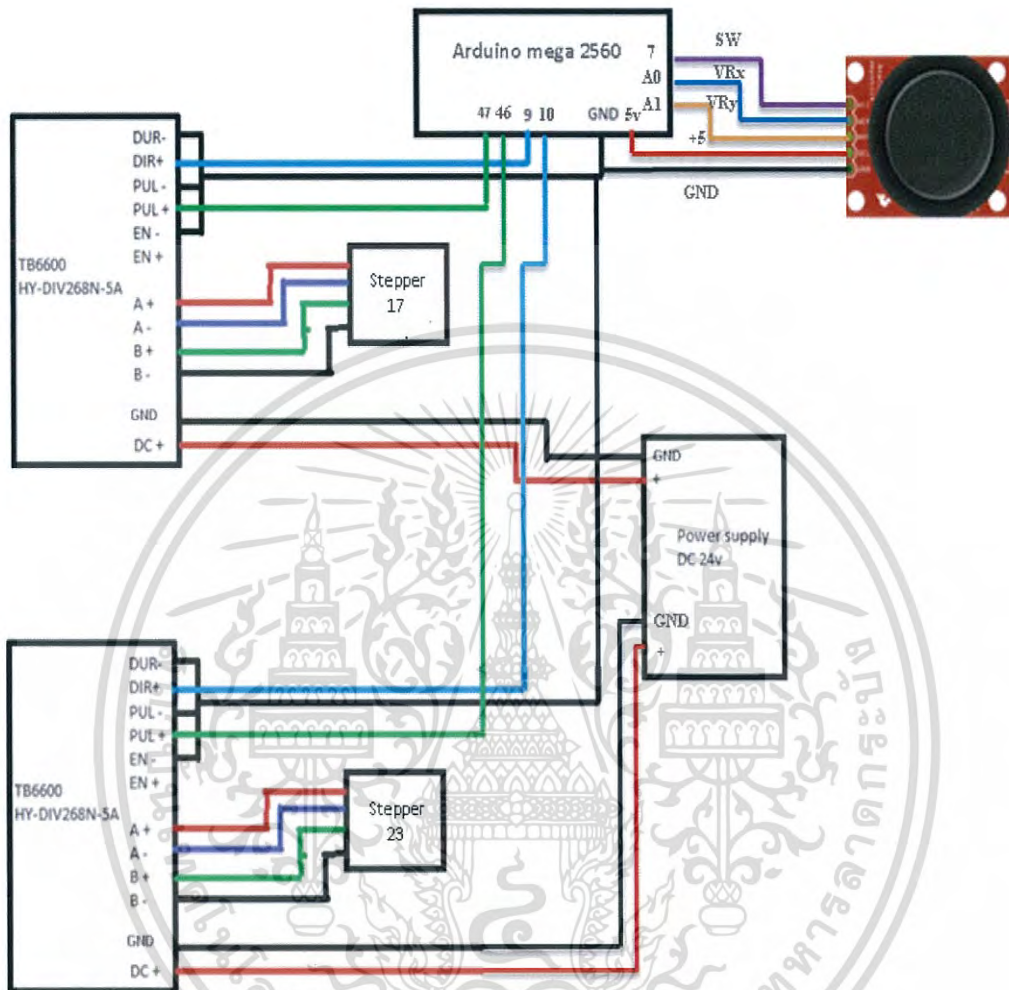


รูปที่ 3.18 TB6600 Stepper Motor Driver

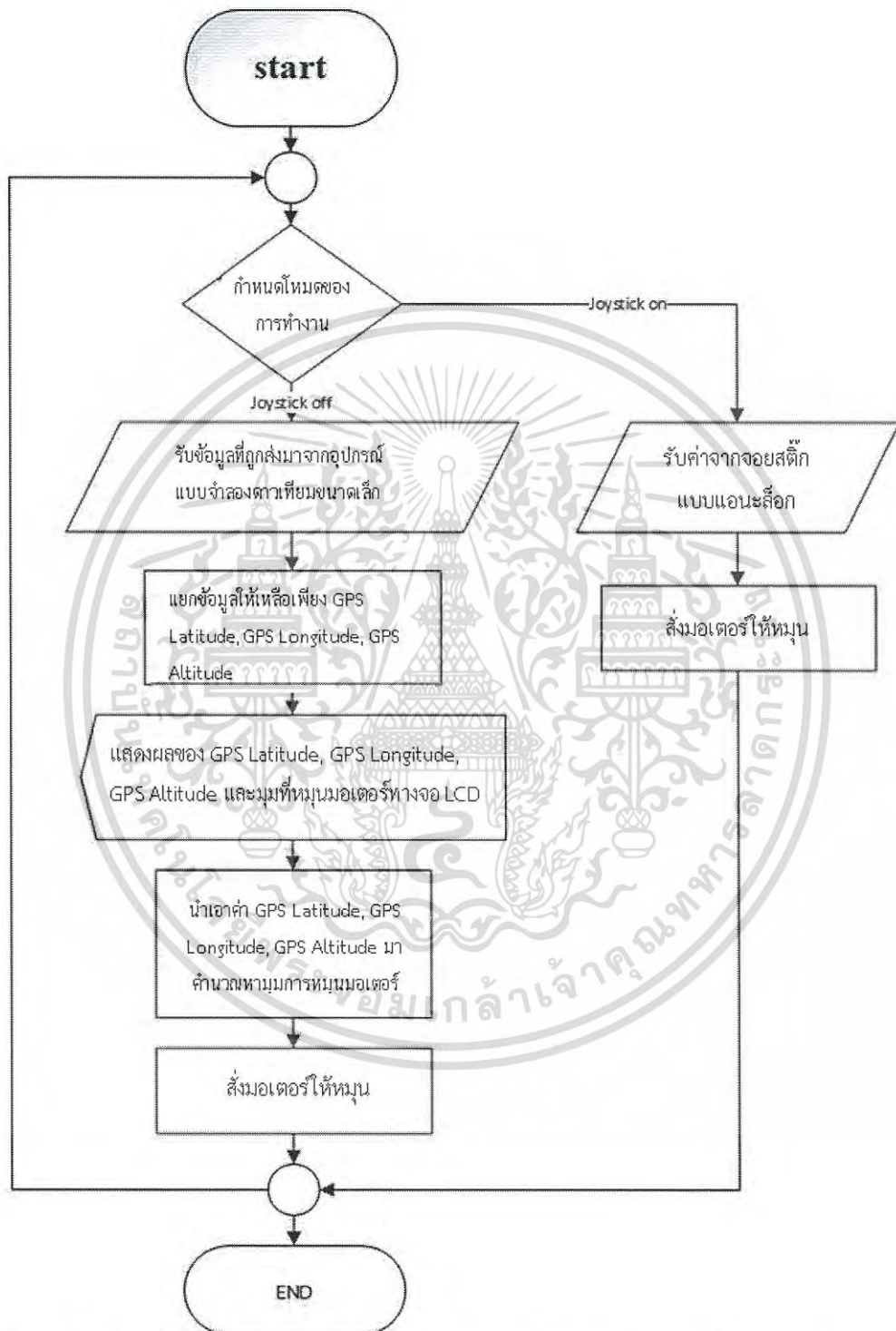
และเนื่องจากในสภาพการทำงานจริงของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็กนั้นอาจจะเกิดปัญหาเนื่องจากรับข้อมูลไม่ได้หรือค่าข้อมูลที่ได้มีความผิดพลาดอันเนื่องมาจากระยะทางที่ไกลๆ หรือมีสิ่งบดบังสัญญาณ ซึ่งจะส่งผลให้ในการติดตามทิศทางของแบบจำลองดาวเทียมขนาดเล็กมีความผิดพลาดได้ ดังนั้นทางผู้จัดทำจึงได้จัดทำฟังก์ชันการทำงานของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็กเพิ่มขึ้นมาอีกหนึ่งรูปแบบคือ การบังคับทิศทางด้วยจอยสติ๊ก โดยจอยสติ๊กที่เลือกใช้นั้นเป็นจอยสติ๊กแบบแอนะล็อก [26] โดยจะมีลักษณะรูปร่างแสดงดังรูปที่ 3.19 รวมถึงแผนภาพแสดงการเชื่อมต่อจอยสติ๊กกับระบบโดยรวมทั้งหมดแสดงดังรูปที่ 3.20 และ โพล์ชาร์ตแสดงดังรูปที่ 3.21



รูปที่ 3.19 จอยสติ๊กแบบแอนะล็อก



รูปที่ 3.20 การเชื่อมต่อสายของจอยสติ๊กแบบแอนะล็อกกับอุปกรณ์อื่น

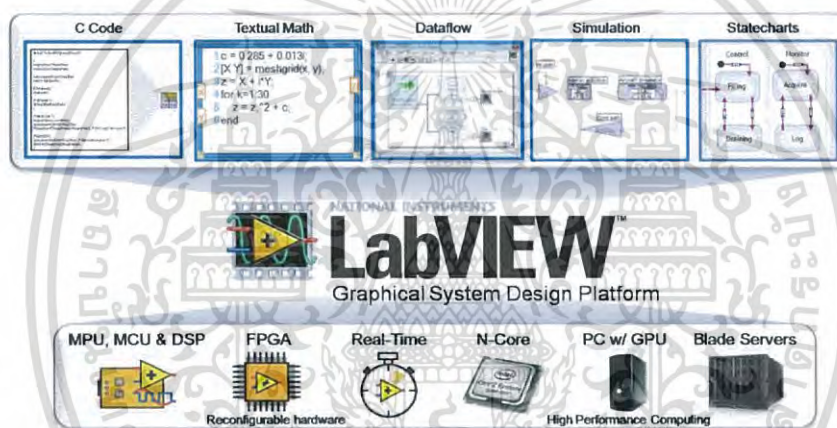


รูปที่ 3.21 โฟลว์ชาร์ตการทำงานของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การออกแบบโปรแกรมที่ใช้ในการแสดงผลข้อมูลด้วยโปรแกรม LabVIEW

LabVIEW [27] เป็นซอฟต์แวร์ระบบวิศวกรรมที่ถูกใช้ในการทดสอบการวัดและการควบคุมโดยสามารถเข้าถึงข้อมูลเชิงลึกเกี่ยวกับฮาร์ดแวร์และข้อมูลได้อย่างรวดเร็วซึ่งการออกแบบโปรแกรมโดยใช้โปรแกรม LabVIEW จะช่วยลดความยุ่งยากในการผสมรวมฮาร์ดแวร์สำหรับงานด้านวิศวกรรมเพื่อให้มีวิธีการที่สอดคล้องกันในการรับข้อมูลจาก NI และฮาร์ดแวร์ของ บริษัท อื่นๆ และ LabVIEW ยังช่วยลดความซับซ้อนของการเขียนโปรแกรมเพื่อให้สามารถมุ่งเน้นไปที่ปัญหาด้านวิศวกรรมเฉพาะ โดย LabVIEW มีความสามารถมากมายโดยแสดงได้ดัง รูปที่ 3.2 และยังสามารถเห็นผลทันทีด้วยการสร้างหน้าจอนินเทอร์เฟซของโปรแกรมได้อย่างสะดวก



รูปที่ 3.22 ความสามารถของโปรแกรม LabVIEW

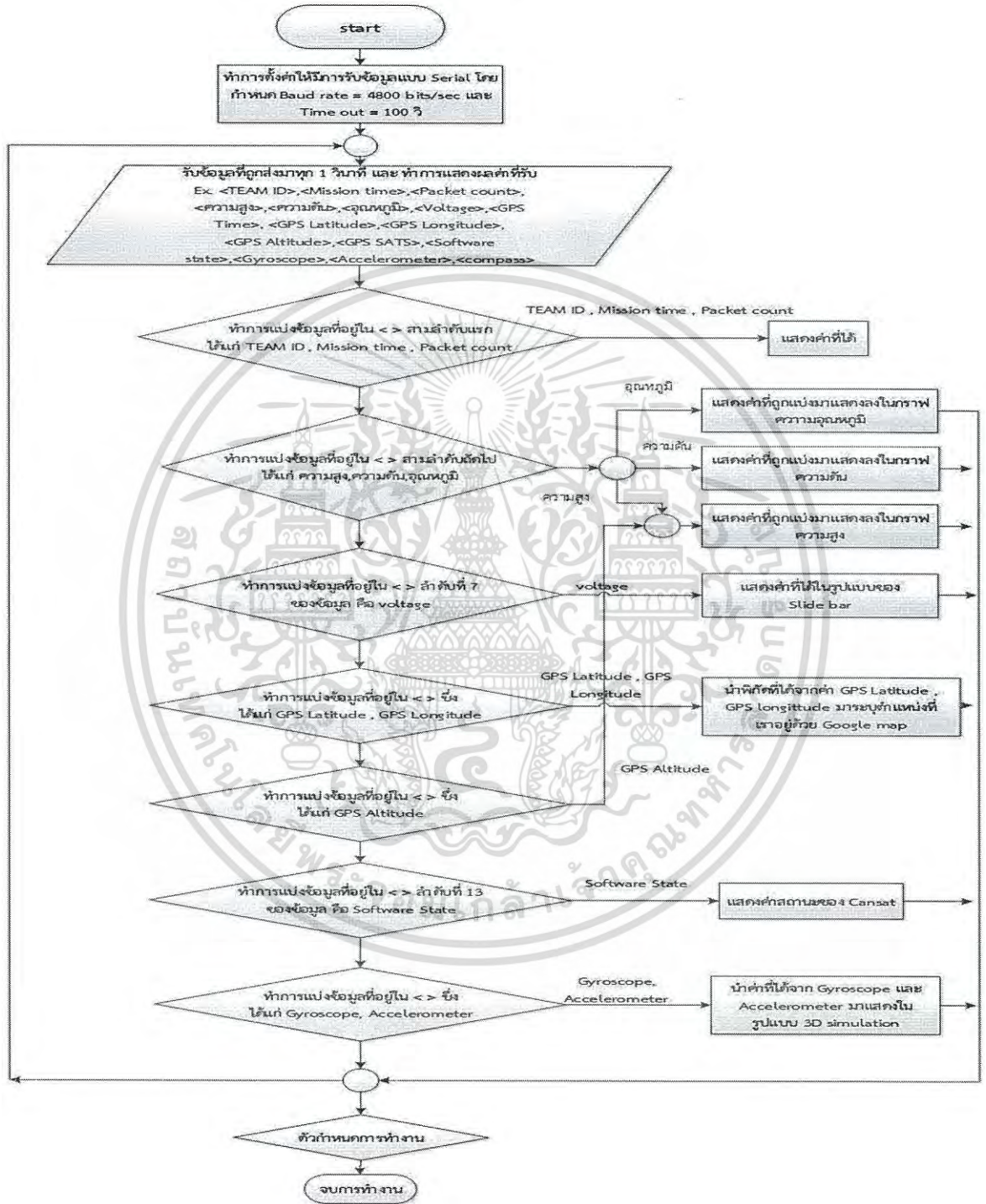
3.1.3.1 รูปแบบข้อมูล

รูปแบบของข้อมูลที่ทำกรรับจากแบบจำลองดาวเทียมขนาดเล็กนั้นจะมีรูปแบบอ้างอิงตามข้อกำหนดเป็นรูปแบบของการแข่งขัน CanSat Competition ประจำปี 2018 ซึ่งรูปแบบของข้อมูลนั้นจะแสดงดังรูปที่ 3.23

```
<TEAM ID>,<MISSION TIME>,<PACKET COUNT>,<ALTITUDE>,<PRESSURE>,<TEMP>,<VOLTAGE>,<GPS TIME>,<GPS LATITUDE>,<GPS LONGITUDE>,<GPS ALTITUDE>,<GPS SATS>,<TILT X>,<TILT Y>,<TILT Z>,<SOFTWARE STATE>,<[BONUS STATUS]>,<GPS SPEED>,<GYRO X>,< GYRO Y >,< GYRO Z >,< MAG >
```

รูปที่ 3.23 รูปแบบของข้อมูลที่ทำกรรับจากแบบจำลองดาวเทียมขนาดเล็ก

3.1.3.2 โพล์ชาร์ตการแยกข้อมูล



รูปที่ 3.24 โพล์ชาร์ตการแยกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการออกแบบโปรแกรมนี้มีหลักการในการทำงานคือรับข้อมูลจากฝั่งส่งที่มีการแยกค่าข้อมูลของแต่ละตัวด้วยตัวคั่นที่ได้ตั้งไว้ จากนั้นทำการแยกข้อมูลแต่ละตัวไปทำการพล็อตออกเป็นกราฟของแต่ละค่า นำค่าละติจูด, ลองจิจูด ไปทำการแสดงตำแหน่งในหน้าจอแสดงตำแหน่ง GPS โดยจะแสดงหลักการการทำงานเป็นโฟลว์ชาร์ตแสดงดังรูปที่ 3.24

3.1.3.3 โปรแกรมออกแบบหน้าต่างแสดงผล

เขียนโปรแกรม LabVIEW เพื่อออกแบบหน้าต่างแสดงผล [28] ให้สามารถรับข้อมูลจากแบบจำลองดาวเทียมขนาดเล็กแล้วนำมาแสดงผลในหน้าจอแสดงผลของโปรแกรมที่ใช้หลักการของโฟลว์ชาร์ตในรูปที่ 3.24 ในการแยกข้อมูลแต่ละส่วนไปแสดงผล โดยหน้าจอที่ทำการออกแบบเสร็จแล้วจะเป็นไปตามรูปที่ 3.25



รูปที่ 3.25 ส่วนประกอบต่างๆของหน้าจอแสดงผล

จากรูปที่ 3.25 นั้นแสดงส่วนประกอบต่างๆของหน้าจอแสดงผลโดยจะประกอบไปด้วยการแสดงผลหลักๆ 6 แบบ คือ

1) แบบกล่องข้อความ จะประกอบไปด้วย

1. Team ID คือ หมายเลขลำดับของทีมซึ่งจะได้จากแข่งขัน
2. Status คือ สแตตัสของสถานการณ์ทำงานของแบบจำลอง

ดาวเทียมขนาดเล็ก

3. Mission Time คือ เวลาที่จับตั้งแต่แบบจำลองดาวเทียมขนาดเล็กเริ่มทำงาน

4. Packet Count คือ ลำดับของ packet นั้นๆ

5. Serial monitor คือ ข้อความที่รับได้ทั้งหมด

2) แบบกราฟแสดงผล จะประกอบไปด้วย

1. Altitude Graph คือ กราฟความสูง

2. 3 Axes Acceleration Graph คือ กราฟความเร่งสามแกน

3. Pressure Graph คือ กราฟความดัน

4. Temperature Graph คือ กราฟอุณหภูมิ

3) แบบหน้าจอแสดงผล GPS คือ หน้าจอแสดงพิกัดตำแหน่งของแบบจำลองดาวเทียมขนาดเล็กโดยใช้ค่า GPS Latitude และ GPS Longitude ในการแสดงตำแหน่ง

4) แบบหน้าจอ 3D simulator คือ หน้าจอแสดงลักษณะการหมุนของแบบจำลองดาวเทียมขนาดเล็กโดยใช้ค่าจากเซนเซอร์ 3 Axes Acceleration ในการแสดงเป็นรูปแบบจำลอง 3D

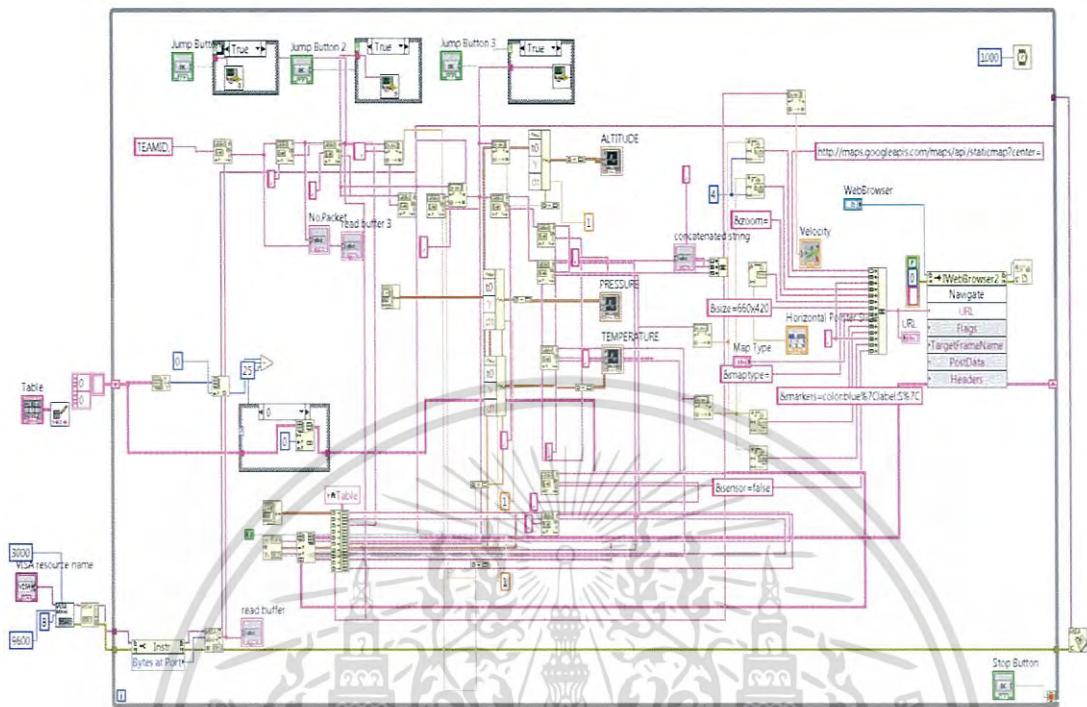
5) แบบเข็มทิศ คือ ทิศทางของแบบจำลองดาวเทียมขนาดเล็กโดยใช้ค่าจากเซนเซอร์ Magnetometer ในการชี้ทิศทาง

6) แบบ meter แสดงผล จะประกอบไปด้วย

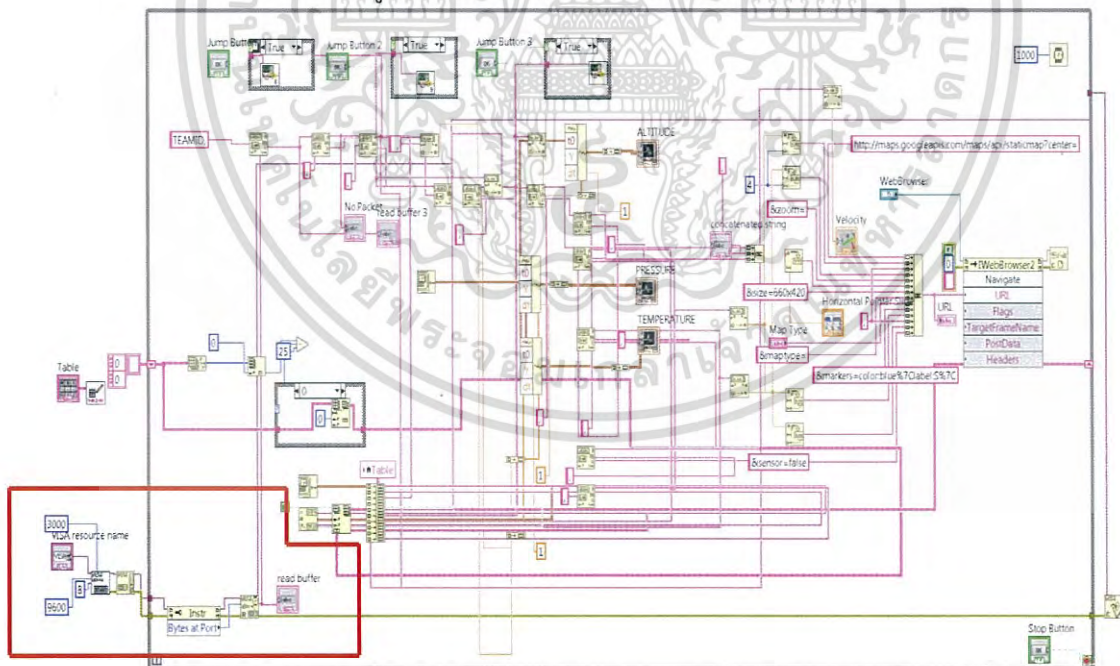
1. Voltage คือ แสดง voltage ที่เหลือของถ่านในแบบจำลองดาวเทียมขนาดเล็ก

2. Speed คือ ความเร็วของแบบจำลองดาวเทียมขนาดเล็ก

ในการออกแบบโปรแกรมที่ใช้ในการแสดงผลข้อมูลนั้นจะเขียนโดยใช้งานโปรแกรม LabVIEW ซึ่งจะมีหน้าตาในการออกแบบดังรูปที่ 3.26

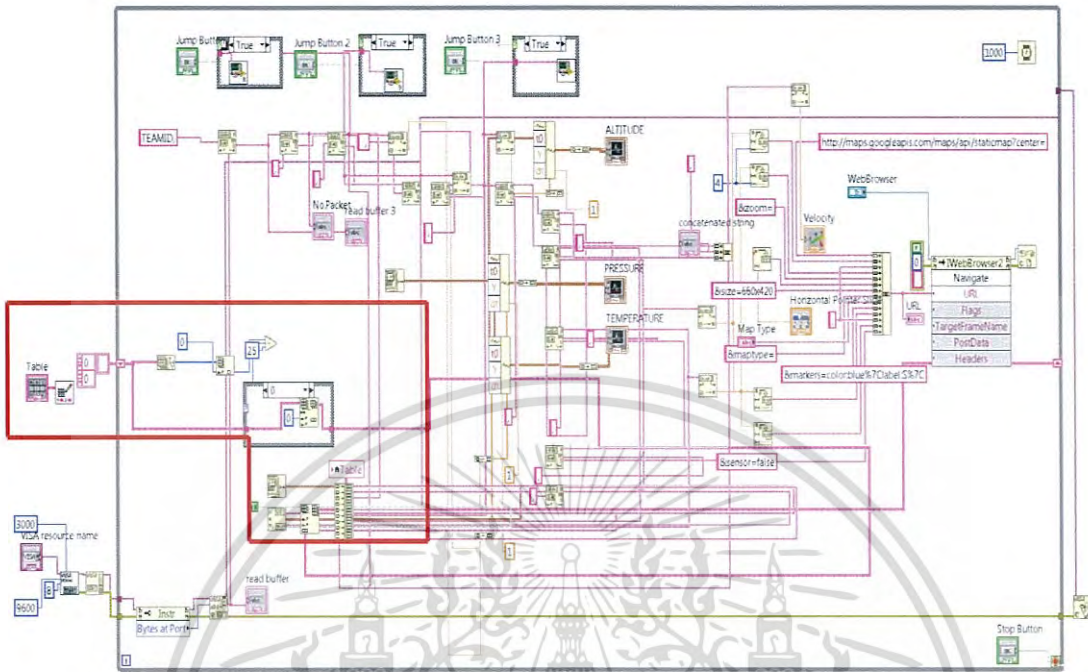


รูปที่ 3.26 โปรแกรมที่ทำการออกแบบ

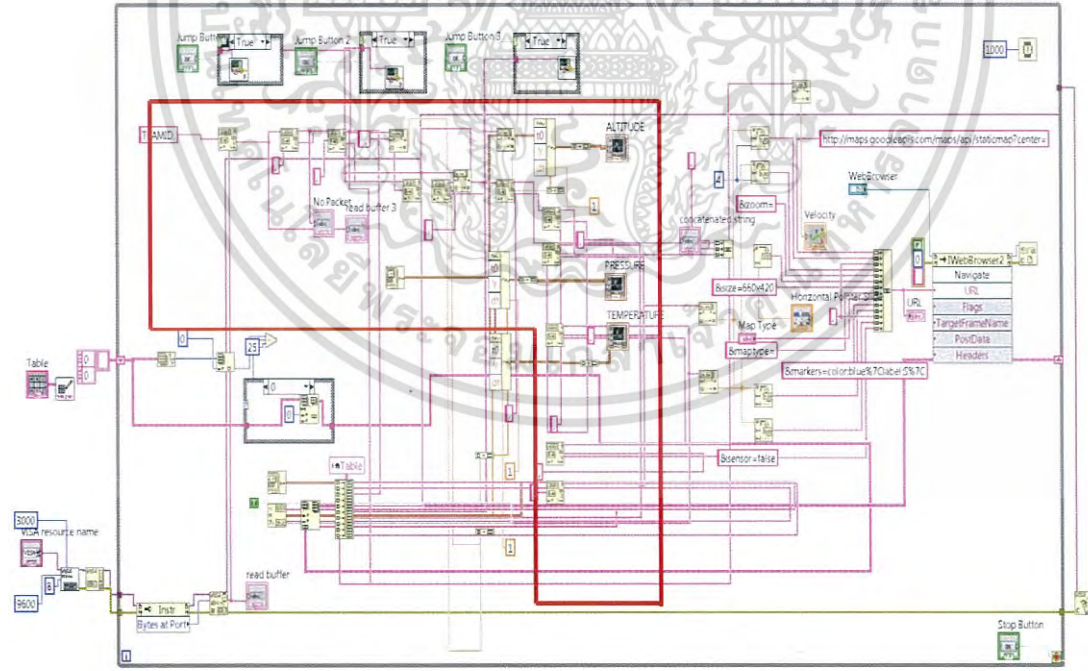


รูปที่ 3.27 โปรแกรมที่ทำงานในส่วนของการรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

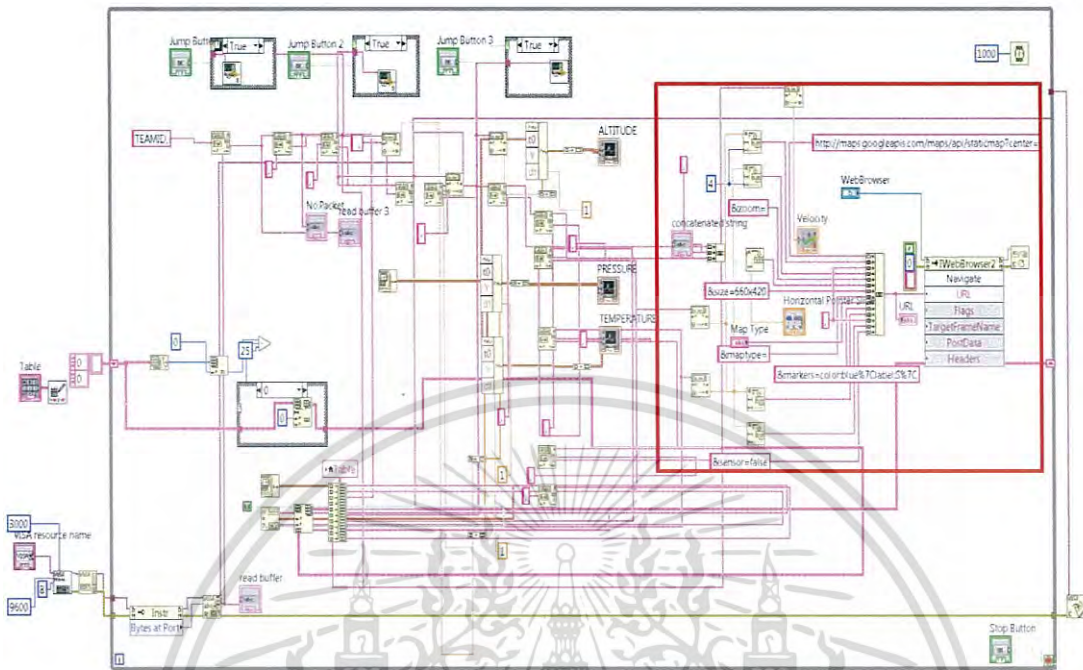


รูปที่ 3.28 โปรแกรมที่ทำงานในส่วนของตาราง



รูปที่ 3.29 โปรแกรมที่ทำงานในส่วนของกรแยกข้อมูลและแสดงผลกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.30 โปรแกรมที่ทำงานในส่วนของแสดงผล GPS

โดยจากรูปที่ 3.27 ในกรอบสี่เหลี่ยมจะเป็นส่วนของโปรแกรมที่ทำหน้าที่ในการรับข้อมูลจากฝั่งส่งแล้วแสดงในหน้าจอแสดงผลว่าข้อความที่รับมานั้นคืออะไรบ้าง จากรูปที่ 3.28 ในกรอบสี่เหลี่ยมจะเป็นส่วนของโปรแกรมที่ทำหน้าที่ของการใส่ค่าข้อมูลที่รับมาได้ลงในตารางในหน้าจอแสดงผล จากรูปที่ 3.29 ในกรอบสี่เหลี่ยมจะเป็นส่วนของโปรแกรมที่ทำหน้าที่ในการแยกข้อมูลจากฝั่งส่งที่ได้รับแยกออกจากกันด้วยเครื่องหมาย $>$, $<$ จากนั้นก็นำค่าไปพลอตกราฟในหน้าจอแสดงผล และจากรูปที่ 3.30 ในกรอบสี่เหลี่ยมจะเป็นส่วนของโปรแกรมที่ทำหน้าที่ในส่วนของการแสดงผลตำแหน่งพิกัด GPS โดยรับค่าละติจูดและลองจิจูด ไปใส่ใน URL ของ google map แล้วทำการแสดงตำแหน่งที่หน้าจอแสดงผล

โดยอุปกรณ์แต่ละตัวที่ใช้ในการออกแบบจะประกอบไปด้วย

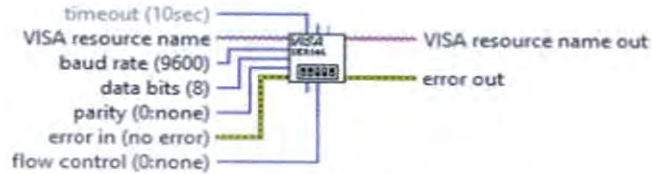
1) VISA resource name เป็นตัวกำหนดพอร์ตของอุปกรณ์ที่เชื่อมต่อแสดงดังรูปที่ 3.31

VISA resource name



รูปที่ 3.31 VISA resource name

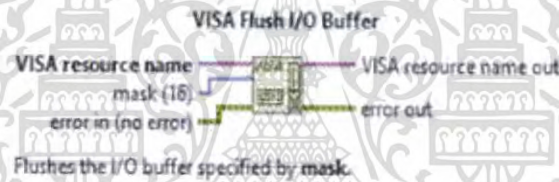
2) VISA Configure Serial Port เป็นเครื่องมือที่ใช้ในการส่งแบบ serial โดยเริ่มจากเชื่อมต่อสายจากพอร์ต VISA resource name เพื่อกำหนดอุปกรณ์ที่จะเชื่อมต่อ แสดงดังรูปที่ 3.32



รูปที่ 3.32 VISA Configure Serial Port

รูปที่ 3.33

3) VISA Flush I/O Buffer เป็นเครื่องมือที่ใช้ในการล้าง Buffer แสดงดัง



รูปที่ 3.33 VISA Flush I/O Buffer

4) Table เป็นเครื่องมือตารางที่ใช้สำหรับเก็บค่า แสดงดังรูปที่ 3.34



รูปที่ 3.34 Table

5) Array size ทำหน้าที่หาค่าขนาดของ array แสดงดังรูปที่ 3.35

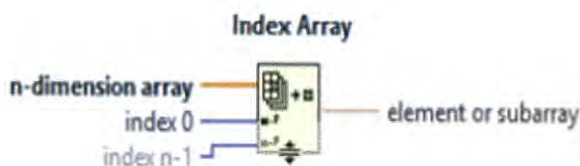
Array Size



รูปที่ 3.35 Array size

6) Index Array ทำหน้าที่แปลงมิติของarray แบ่งเป็นส่วนย่อยๆ แสดงดัง

รูปที่ 3.36



รูปที่ 3.36 Index Array

7) Wait (ms) เป็นอุปกรณ์ที่ใช้ในการกำหนดค่าหน่วยเวลาโดยค่าที่ใส่เข้าไปนั้นจะมีหน่วยเป็น milliseconds แสดงดังรูปที่ 3.37



รูปที่ 3.37 Wait (ms)

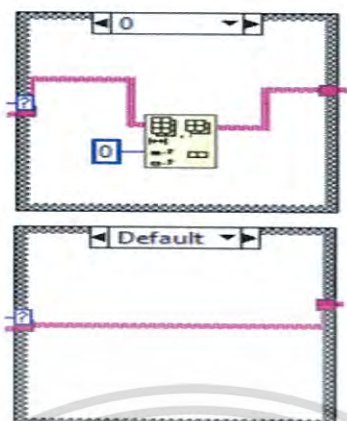
3.38

8) read buffer เป็นอุปกรณ์ที่ใช้ในการอ่านค่าข้อมูลที่ได้รับ แสดงดังรูปที่



รูปที่ 3.38 read buffer

9) Case Structure เป็นอุปกรณ์ที่ใช้ในการกำหนดเงื่อนไขที่เราต้องการ โดยเงื่อนไขนั้นจะขึ้นอยู่กับ input หาก input ไหนตรงกับเงื่อนไขก็จะทำตามรูปแบบที่กำหนดไว้ โดยแสดงดังรูปที่ 3.39



รูปที่ 3.39 Case Structure

10) Fract/Exp String To Number เป็นอุปกรณ์ที่ใช้ในการเปลี่ยนค่า string ให้กลายเป็นตัวเลขเพื่อนำเอาค่าที่ได้ไปใช้ในการคำนวณ แสดงดังรูปที่ 3.40

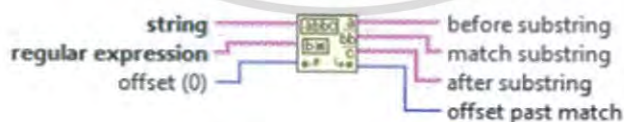
Fract/Exp String To Number



รูปที่ 3.40 Fract/Exp String To Number

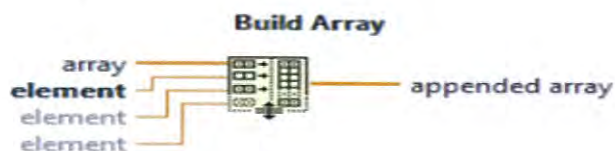
11) Match Pattern เป็นอุปกรณ์ที่ใช้ในการแยกข้อมูลที่ส่งมาผ่านพอร์ต String โดยการกำหนดตัวอักษรหรือเครื่องหมายที่ใช้ในการแยกผ่านพอร์ต regular expression โดยได้ใช้เป็นเครื่องหมาย , ในการแยกข้อมูลโดย แสดงดังรูปที่ 3.41

Match Pattern



รูปที่ 3.41 Match Pattern

12) Build Array เป็นอุปกรณ์ที่ใช้ในการรวม Array แต่ละ element เข้าด้วยกันโดยแต่ละ element ต้องมีขนาดเท่ากัน แสดงดังรูปที่ 3.42



รูปที่ 3.42 Build Array

13) Get Date/Time In Seconds เป็นอุปกรณ์ที่ใช้ในการสร้างข้อมูลที่เป็นวันและเวลาปัจจุบัน แสดงดังรูปที่ 3.43

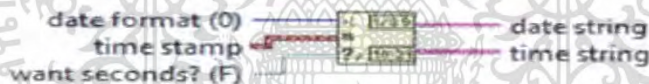
Get Date/Time In Seconds



รูปที่ 3.43 Get Date/Time In Seconds

14) Get Date/Time String เป็นอุปกรณ์ที่ใช้ในการแยกวันและเวลาออกมาเป็นข้อมูลชนิด String แสดงดังรูปที่ 3.44

Get Date/Time String



รูปที่ 3.44 Get Date/Time String

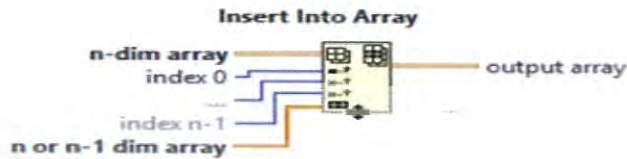
15) Build Waveform เป็นอุปกรณ์ที่ใช้ในการกำหนดแกนโดย t0 พอร์ต เป็นการกำหนดแกน x และ y พอร์ต กำหนดแกน y โดยที่พอร์ต dt จะกำหนดเวลาแกน y โดยที่พอร์ต dt จะกำหนดเวลาที่ใช้ในการแสดงค่า แสดงดังรูปที่ 3.45

Build Waveform



รูปที่ 3.45 Build Waveform

16) Insert Into Array เป็นอุปกรณ์ที่ใช้ในการรวม Array เข้าด้วยกัน แสดงดังรูปที่ 3.46



รูปที่ 3.46 Insert Into Array

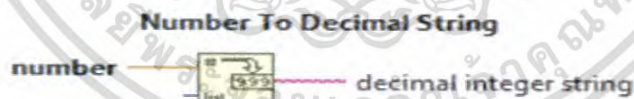
17) Waveform Chart เป็นอุปกรณ์ที่ใช้ในการแสดงข้อมูลออกมาเป็นกราฟแสดงดังรูปที่ 3.47

Waveform Chart



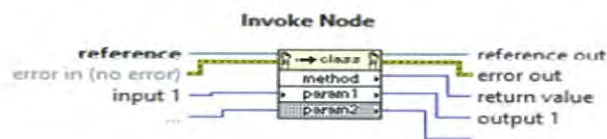
รูปที่ 3.47 Waveform Chart

18) Number To Decimal String เป็นอุปกรณ์ที่ใช้ในการแปลงตัวเลขให้กลายเป็นข้อมูลทศนิยมแบบ string แสดงดังรูปที่ 3.48



รูปที่ 3.48 Number To Decimal String

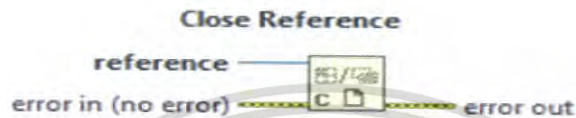
19) Invoke Node เป็นอุปกรณ์ที่ใช้ในการเรียกใช้วิธีการหรือการกระทำจาก reference ส่วนมากวิธีการจะมีพารามิเตอร์ที่มีความเกี่ยวข้องกัน แสดงดังรูปที่ 3.49



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

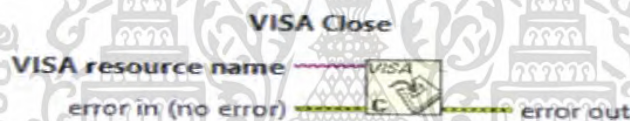
รูปที่ 3.49 Invoke Node

20) Close Reference เป็นอุปกรณ์ที่ใช้ในการปิดการทำงานของอุปกรณ์ที่เกี่ยวข้องกับตัว reference แสดงดังรูปที่ 3.50



รูปที่ 3.50 Close Reference

21) VISA Close เป็นอุปกรณ์ที่ใช้ในการปิดการทำงานของอุปกรณ์ที่เกี่ยวข้องกับตัว VISA resource name แสดงดังรูปที่ 3.51



รูปที่ 3.51 VISA Close

22) While Loop เป็นอุปกรณ์ที่ใช้ในการกำหนดการทำงานโดยจะทำงานจนกว่าค่าผลลัพธ์ที่ได้จะ false หรือทำการกำหนดเองโดยใช้ตัว control แสดงดังรูปที่ 3.52

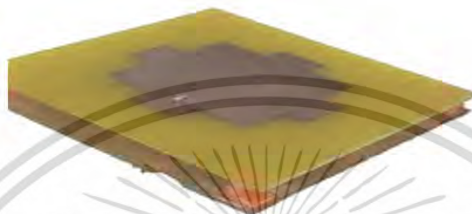


รูปที่ 3.52 While Loop

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 สายอากาศ stack microstrip

สายอากาศที่นำค่าจากการคำนวณไปทำการออกแบบด้วยโปรแกรมและนำค่าจากการจำลองค่าความยาวและระยะห่างแต่ละพารามิเตอร์ในโปรแกรมมาทำการสร้างเป็นสายอากาศขึ้นมา โดยแสดงดังรูปที่ 3.53



รูปที่ 3.53 สายอากาศ stack microstrip

3.2.2 เครื่องสเปกตรัมอนาไลเซอร์(Spectrum Analyzer)

สเปกตรัมอนาไลเซอร์ หรือ Spectrum Analyzer [29] เป็นอุปกรณ์ที่มีความสามารถในการวัดสัญญาณในย่านความถี่สูงมากๆ เช่นความถี่จากเสาอากาศมักใช้ในงานสื่อสารแบบไร้สาย สเปกตรัมอนาไลเซอร์สามารถวัดและแสดงค่าในรูปแบบความถี่(Frequency domain) และแอมพลิจูดของสัญญาณได้โดยใช้ในการวัดค่าคุณสมบัติที่จำเป็นของสายอากาศเพื่อดูประสิทธิภาพมีลักษณะดังแสดงในรูปที่ 3.54



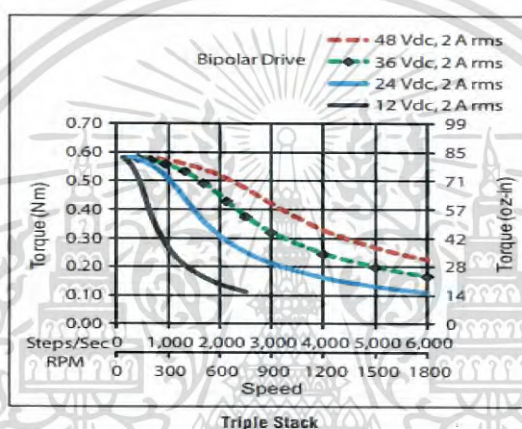
รูปที่ 3.54 เครื่องสเปกตรัมอนาไลเซอร์ [29]

3.2.3 Stepper Motor NEMA 17

มอเตอร์ตัวแรกที่เลือกใช้คือ Stepper Motor NEMA 17 [20] (ตัวที่อยู่ด้านบนของรูปที่ 3.39) จะนำไปใช้ในการหมุนตามแนวแกนตั้ง และมีคุณสมบัติต่างๆแสดงดังตารางที่ 3.3 โดยจะมีประสิทธิภาพของ Stepper Motor NEMA 17 เมื่อใช้งานต่อเนื่องเทียบกันแต่ละแรงดันไฟฟ้าแสดงในรูปที่ 3.55

ตารางที่ 3.3 คุณสมบัติของ Stepper Motor NEMA 17

Description	Length	Mounted Rated Current	Mounted Holding Torque		Detent Torque		Rotor Inertia		Motor Weight	
			Nm Typ.	oz-in Typ.	mNm	oz-in	g cm ²	oz-in ²	Kg	lbs
(Stack)	"L" Max	Amps								
Triple	62.8 mm (2.47 in)	2	0.83	120	30	4.2	123	0.67	0.6	1.3



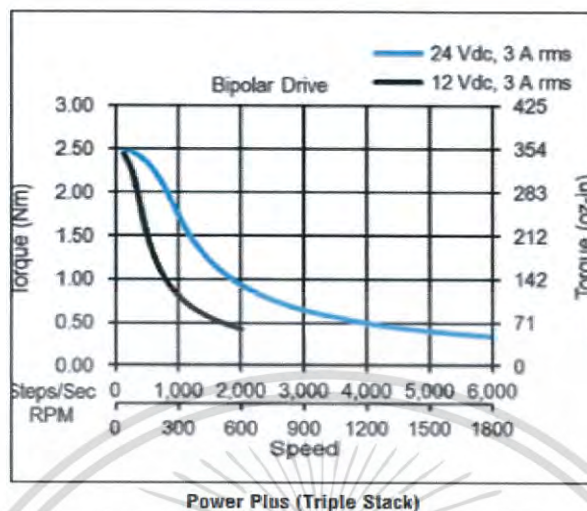
รูปที่ 3.55 ประสิทธิภาพของ Stepper Motor NEMA 17 เมื่อใช้งานต่อเนื่องเทียบกับกันแต่ละแรงดันไฟฟ้า [20]

3.2.4 Stepper Motor NEMA 23

มอเตอร์ตัวที่สองนั้น Stepper Motor NEMA 23 [20] (ตัวที่อยู่ด้านล่างของรูปที่ 3.39) จะนำไปใช้ในการหมุนตามแนวแกนนอน และมีคุณสมบัติต่างๆแสดงในตารางที่ 3.4 โดยจะมีประสิทธิภาพของ Stepper Motor NEMA 23 เมื่อใช้งานต่อเนื่องเทียบกับกันแต่ละแรงดันไฟฟ้าแสดงดังในรูปที่ 3.56

ตารางที่ 3.4 คุณสมบัติของ Stepper Motor NEMA 23

Description	Length	Mounted Rated Current	Mounted Holding Torque		Detent Torque		Rotor Inertia		Motor Weight	
			Nm Typ.	z-in Typ.	mNm	oz-in	g cm ²	oz-in ²	kg	lbs
(Stack)	"L" Max	Amps								
Power Plus	77.0 mm (3.03 in)	3	3.30	470	150	21	390	2.1	1.1	2.4



รูปที่ 3.56 ประสิทธิภาพของ Stepper Motor NEMA 23 เมื่อใช้งานต่อเนื่องเทียบกันแต่ละแรงดันไฟฟ้า [20]

3.2.5 TB6600 Stepper Motor Driver

Stepping Motor Drive [25] เป็นอุปกรณ์ที่คอยขับสเต็ปเปอร์มอเตอร์ให้สามารถหมุนไปตามตำแหน่งและทิศทางตามที่ต้องการซึ่งเป็นอุปกรณ์ที่ทำหน้าที่เชื่อมต่อระหว่างสเต็ปเปอร์มอเตอร์กับอุปกรณ์ไมโครคอนโทรลเลอร์ มีลักษณะอุปกรณ์ตามรูปที่ 3.57



รูปที่ 3.57 TB6600 Stepper Motor Driver

TB6600 Stepper Motor Driver มีคุณสมบัติต่างๆดังแสดงดังตารางที่ 3.5

ตารางที่ 3.5 คุณสมบัติต่างๆของ TB6600 Stepper Motor Driver

คุณสมบัติ	ค่าของคุณสมบัติ
Input Current	0-5 A
Output Current	0.5 - 4.0A
Control Signal	3.3 - 24V
Power (MAX)	160 W
Micro Step	1, 2/A, 2/B, 4, 8, 16, 32
Weight	0.2 kg

3.2.6 3.2" HVGA LCD TFT Non-Touch Screen, Arduino MEGA-2560 Shield

HVGA LCD TFT Non-Touch Screen, Arduino MEGA-2560 Shield [24] เป็นหน้าจอ LCD ที่ความละเอียดอยู่ที่ 490 x 320 และมีทั้งหมด 262K สี โดยอุปกรณ์มีลักษณะดังรูปที่ 3.58 และมีคุณสมบัติดังตารางที่ 3.6



รูปที่ 3.58 อุปกรณ์ HVGA LCD TFT Non-Touch Screen, Arduino MEGA-2560 Shield [24]

ตารางที่ 3.6 คุณสมบัติของอุปกรณ์ HVGA LCD TFT Non-Touch Screen, Arduino MEGA-2560 Shield

คุณสมบัติ	ค่าของคุณสมบัติ
เลขของอุปกรณ์ IC	ILI9481
ขนาดของจอแสดงผล	3.2 นิ้ว
ชนิดของ Backlight	Highlighting White 6 LED
ความละเอียดของหน้าจอ	480 * 320 Pixel (HVGA)
ปริมาณการใช้กระแสไฟ	80~110 mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ	ค่าของคุณสมบัติ
ชนิดของหน้าจอแสดงผล	TFT (IPS Whole Perspective)
แรงดันที่ใช้งาน	5V / 3.3V
ขนาดของอุปกรณ์	Size : 89.92(W) * 54.25(L) (mm)

อุปกรณ์ที่นำมาเชื่อมต่อกับจอ HVGA LCD TFT Non-Touch Screen ได้แก่ Arduino Mega 2560 ทำให้ไม่ต้องต่อกับบอร์ดนี้เพราะเนื่องจากตัวอุปกรณ์ มี HVGA LCD TFT Non-Touch Screen มี shield ที่ติดมากับอุปกรณ์ที่ใช้สำหรับต่อกับบอร์ด Arduino MEGA-2560 ได้โดยตรงทำให้สะดวกในการเชื่อมต่อกับ บอร์ด Arduino MEGA-2560

3.2.7 Arduino Mega 2560



รูปที่ 3.59 บอร์ด Arduino Mega 2560 [30]

Arduino Mega 2560 [30] บอร์ดในตระกูลของ Arduino ซึ่งจะมีคุณสมบัติต่างๆ เพิ่มขึ้นจาก Arduino Uno R3 โดยจะใช้ชิพ ATmega2560 ที่มีหน่วยความจำแฟลช 256 KB แรม 8 KB และใช้ไฟเลี้ยงอยู่ที่ 7 ถึง 12 V แรงดันของระบบอยู่ที่ 5 V มี Digital Input / Output มากถึง 54 ขา (เป็น PWM ได้ 14 ขา) มี Analog Input 16 ขา Serial UART 4 ชุด I2C 1 ชุด SPI 1 ชุด เขียนโปรแกรมบน Arduino IDE และยังสามารถโปรแกรมข้อมูลสู่บอร์ดผ่านสาย USB ได้อีกด้วย โดยมีคุณสมบัติโดยย่อตามตารางที่ 3.7 และมีลักษณะแสดงดังรูปที่ 3.59

ตารางที่ 3.7 คุณสมบัติของบอร์ด Arduino Mega 2560

คุณสมบัติ	ค่าของคุณสมบัติ
Microcontroller chip	ATmega2560
Operating Voltage	5V
Input Voltage	7-12V
Digital I/O Pins	54
Analog Input Pins	16
DC Current per I/O Pin	40 mA
Flash Memory	256 KB โดยที่อีก 8 KB ใช้สำหรับ bootloader
Clock Speed	16 MHz

3.2.8 Xbee Pro 60mW U.FL Connection - Series 1 [31]

Xbee เป็นอุปกรณ์ที่มี Microcontroller และ RF IC อยู่ภายใน ทำหน้าที่เป็นอุปกรณ์ transceiver (อุปกรณ์รับ-ส่งสัญญาณ) แบบ Half Duplex ย่านความถี่ 2.4 GHz ตามมาตรฐานโปรโตคอล ZigBee / IEEE 802.15 ใช้พลังงานต่ำ มี interface ที่ใช้รับและส่งข้อมูลกับ Xbee เป็น UART (TTL) โดยตัว Xbee ที่เลือกใช้คือ Xbee Pro 60mW U.FL Connection - Series 1 (802.15.4) โดยจะมีคุณสมบัติตามตารางที่ 3.8 และมีลักษณะแสดงดังรูปที่ 3.60



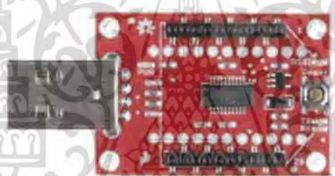
รูปที่ 3.60 Xbee Pro 60mW U.FL Connection - Series 1 (802.15.4) [31]

ตารางที่ 3.8 คุณสมบัติของ Xbee Pro 60mW U.FL Connection - Series 1

คุณสมบัติ	ค่าของคุณสมบัติ
Operating Voltage	3.3 v
Current rate	215mA
Max data rate	250kbps
Power output	60mW
Max transmission range	1 mile (1500m)
Pins	6 10-bit ADC input pins and 8 digital IO pins

3.2.9 Xbee USB Dongle

Xbee usb dongle [32] ทำหน้าที่เป็นตัวเชื่อมระหว่าง xbee กับ คอมพิวเตอร์แสดงดังรูป ที่ 3.61



รูปที่ 3.62 xbee usb dongle [32]

3.2.10 ออสซิลโลสโคป

ออสซิลโลสโคป (Cathode ray oscilloscope ; CRO) ที่แสดงดังรูปที่ 3.62 หมายถึงออสซิลโลสโคปใช้หลอดรังสีแคโทด สโคปเป็นเครื่องมือวัดทาง อิเล็กทรอนิกส์ที่สำคัญอีกชนิดหนึ่งที่ใช้ในการวัดแสดงรูปคลื่นสัญญาณต่างๆ ออกมาเป็นภาพ ปรากฏบนจอหลอดภาพให้เห็นได้ เช่น การวัดสัญญาณกระแสไฟฟ้าหรือแรงดันไฟฟ้า(ที่เป็นไฟ AC หรือ DC) การวัดความถี่ของสัญญาณ การวัดเฟสของสัญญาณ และรวมถึงการวัดสัญญาณพัลส์การอ่านค่าแอมพลิจูดของสัญญาณจะเป็น พีค-ทูปิก หรือค่าพีคและค่าเวลาเป็นวินาที



รูปที่ 3.62 ออสซิลโลสโคป [33]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การจัดเก็บผลการทดลอง

ในการเก็บบันทึกผลการทดลองนั้นจะแบ่งการทดลองเป็นสามส่วนดังนี้

3.3.1 การทดสอบการทำงานของสายอากาศ

การทดสอบการทำงานของสายอากาศนั้นจะทำการทดสอบโดยใช้เครื่องสเปคตรัม-นาไลเซอร์ ซึ่งเป็นการวัดค่าคุณสมบัติต่างๆของสายอากาศว่ามีค่าเท่าไรโดยประกอบด้วย ค่าS11, ค่าVSWR, รูปแบบแพทเทิร์นการแพร่กระจายคลื่น

3.3.2 การทดสอบการทำงานของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก

การทดสอบการทำงานของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็กนั้นทำได้โดยการป้อนข้อมูลให้มอเตอร์หมุนแล้วทดสอบดูว่ามอเตอร์นั้นหมุนไปเท่าไรโดยเปรียบเทียบกับกระดาษที่ทำการเขียนแกนของมุมลงไป

3.3.3 การทดสอบโปรแกรม

การทดสอบการทำงานในส่วนของโปรแกรมนั้นทำได้โดยการรับค่าที่ส่งมาจากฝั่งส่งแล้วทำการแสดงผลที่หน้าจอแสดงผลเพื่อดูว่าการทำงานของโปรแกรมแสดงค่าออกมาได้อย่างที่ควรจะเป็นหรือไม่ จากนั้นก็ทำการทดสอบที่ระยะห่างต่างๆ เพื่อดูประสิทธิภาพเรื่องระยะทางในการรับ-ส่งข้อมูล

บทที่ 4

ผลการทดลอง

4.1 ผลการทดสอบการทำงานของสายอากาศ

เมื่อนำสายอากาศไปทำการสร้างลงบนแผ่นวงจรพิมพ์ FR4 สายอากาศจะมีขนาด 82×82 มม. จากนั้นได้ทำการบัดกรีหัวต่อ SMA 50 โอห์ม แสดงดังรูปที่ 4.1 จากนั้นนำสายอากาศไปทำการวัดทดสอบเพื่อหาค่าการสูญเสียย้อนกลับ (Return Loss) ของสายอากาศที่ความถี่ 2.45 GHz ด้วยเครื่องวิเคราะห์ที่โครงข่ายงาน Field Fox RF Analyzer รุ่น N9912A ผลที่ได้จะแสดงดังรูปที่ 4.2 และผลการวัดค่าการสูญเสียย้อนกลับ (Return Loss) ช่วงแถบความถี่ 2.38-2.56 GHz ซึ่งมีค่าน้อยกว่า -10 dB แสดงดังรูปที่ 4.3 ทำให้เราทราบว่าความถี่ที่วัดได้ครอบคลุมอยู่ในย่านการใช้งานที่ WLAN 2.45 GHz

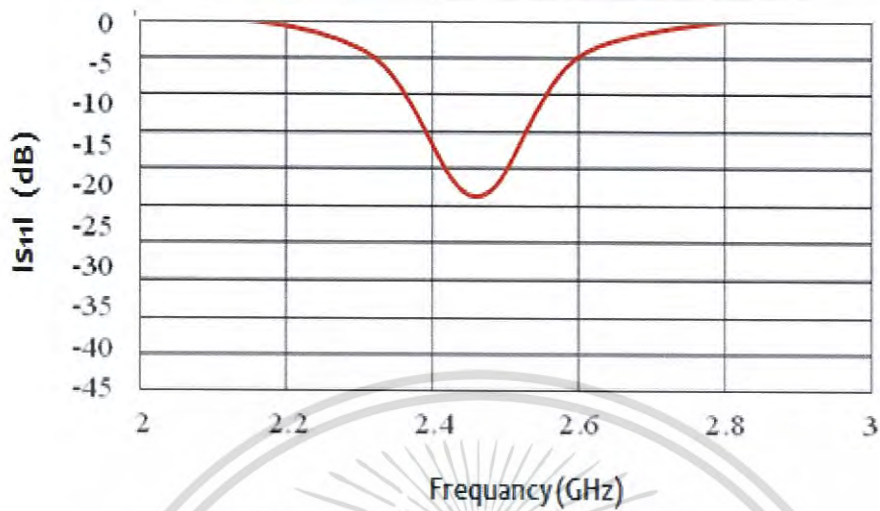


รูปที่ 4.1 สายอากาศสร้างลงบนแผ่นวงจรพิมพ์ FR4

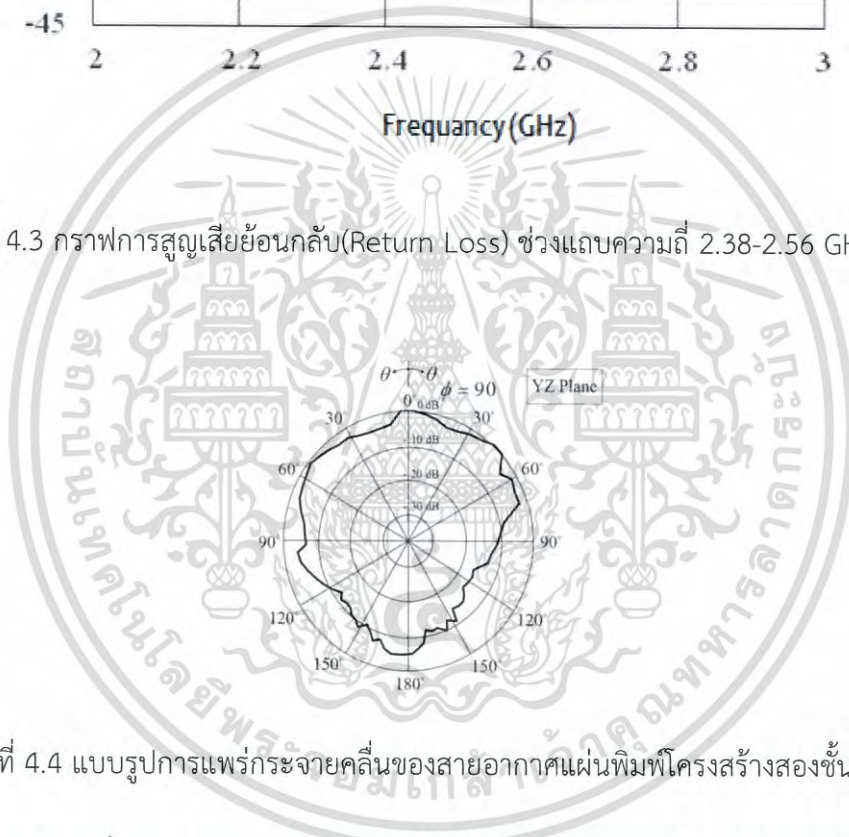


รูปที่ 4.2 ผลการวัดค่าการสูญเสียย้อนกลับ(Return Loss) ช่วงแถบความถี่ 2.38-2.56 GHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 กราฟการสูญเสียย้อนกลับ(Return Loss) ช่วงแถบความถี่ 2.38-2.56 GHz

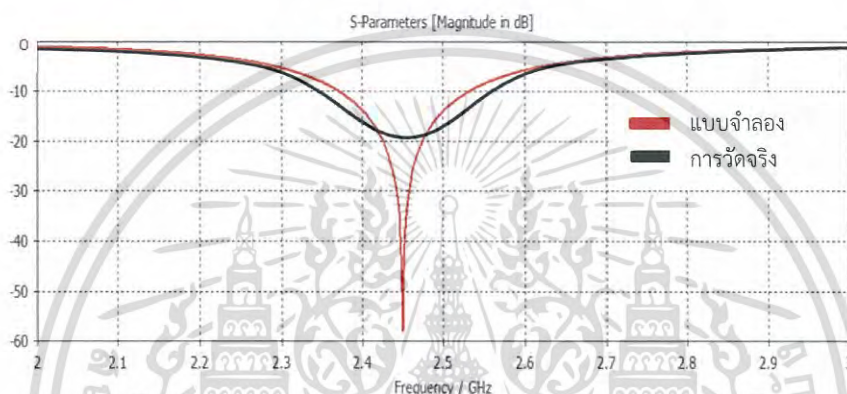


รูปที่ 4.4 แบบรูปการแพร่กระจายคลื่นของสายอากาศแผ่นพิมพ์โครงสร้างสองชั้น

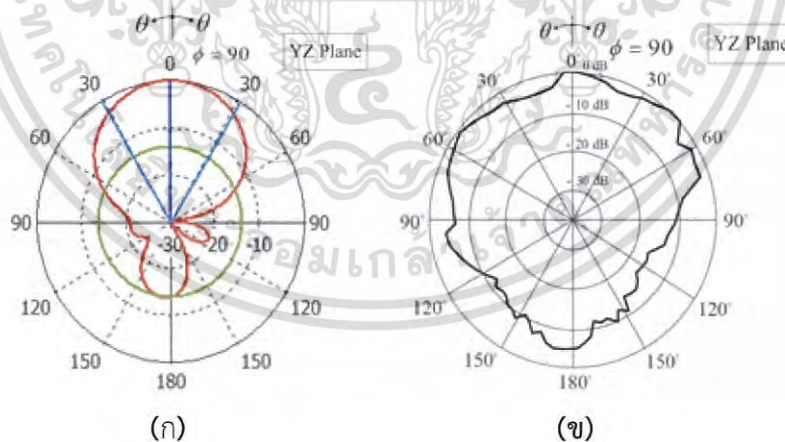
จากรูปที่ 4.4 แสดงแบบรูปการแพร่กระจายคลื่นของสายอากาศแผ่นพิมพ์โครงสร้างสองชั้นตลอดช่วงความถี่ 2.38-2.56 GHz ในระนาบ y-z plane ที่ความถี่กลาง 2.45 GHz สายอากาศที่สร้างขึ้นนี้มีแบบรูปการแพร่กระจายคลื่นแบบชี้ทิศทางเดียวและมีอัตราขยายของสายอากาศที่ 6.5 dBi

4.1.1 เปรียบเทียบผลการจำลองด้วยโปรแกรมกับการสร้างจากแบบจำลอง

หลังจากสร้างและทดสอบสายอากาศค่าการสูญเสียย้อนกลับ (Return Loss) ของสายอากาศในช่วงความถี่ 2.38 - 2.53 GHz นำผลที่ได้ไปเปรียบเทียบกับผลจากแบบจำลองสายอากาศด้วยโปรแกรม แสดงดังรูปที่ 4.5 และเปรียบเทียบแบบรูปการแพร่กระจายคลื่นของสายอากาศแผ่นพิมพ์โครงสร้างสองชั้นที่ความถี่กลาง 2.45 GHz แสดงดังรูปที่ 4.6 และเปรียบเทียบอัตราการขยายของสารอากาศที่ได้แสดงดังตารางที่ 4.1



รูปที่ 4.5 เปรียบเทียบค่าการสูญเสียย้อนกลับ(Return Loss) ที่ความถี่กลาง 2.45 GHz



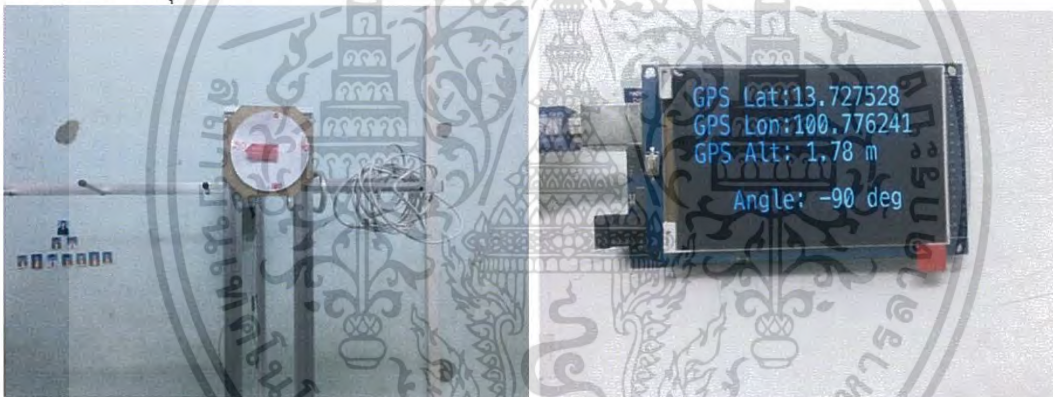
รูปที่ 4.6 แบบรูปการแพร่กระจายคลื่นของสายอากาศแผ่นพิมพ์โครงสร้างสองชั้น (ก) แบบรูปการแพร่กระจายคลื่นจากแบบจำลอง (ข) แบบรูปการแพร่กระจายคลื่นจากการสร้าง

ตารางที่ 4.1 เปรียบเทียบค่าอัตราขยายของสายอากาศที่ได้จากแบบจำลองและจากการสร้าง

ความถี่ (GHz)	อัตราขยายของสายอากาศ (dBi)	
	ผลจากแบบจำลอง	ผลจากการวัดค่า
2.45	8.31	6.5

4.2 ผลการทดสอบการทำงานของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก

จากรูปที่ 4.7 และรูปที่ 4.8 คือการป้อนข้อมูลใส่มอเตอร์ให้หมุนไป 90 องศา ในด้านทวนเข็มนาฬิกาและ หมุนไป 90 องศาในด้านตามเข็มนาฬิกา ตามลำดับ พร้อมกับแสดงผลข้อมูลผ่านทางจอ LCD แล้วทดสอบดูว่ามอเตอร์นั้นหมุนไปเท่าไรโดยการเทียบกับกระดาษที่ทำการเขียนแกนของมุมลงไป



รูปที่ 4.7 ทดสอบการหมุนของมอเตอร์



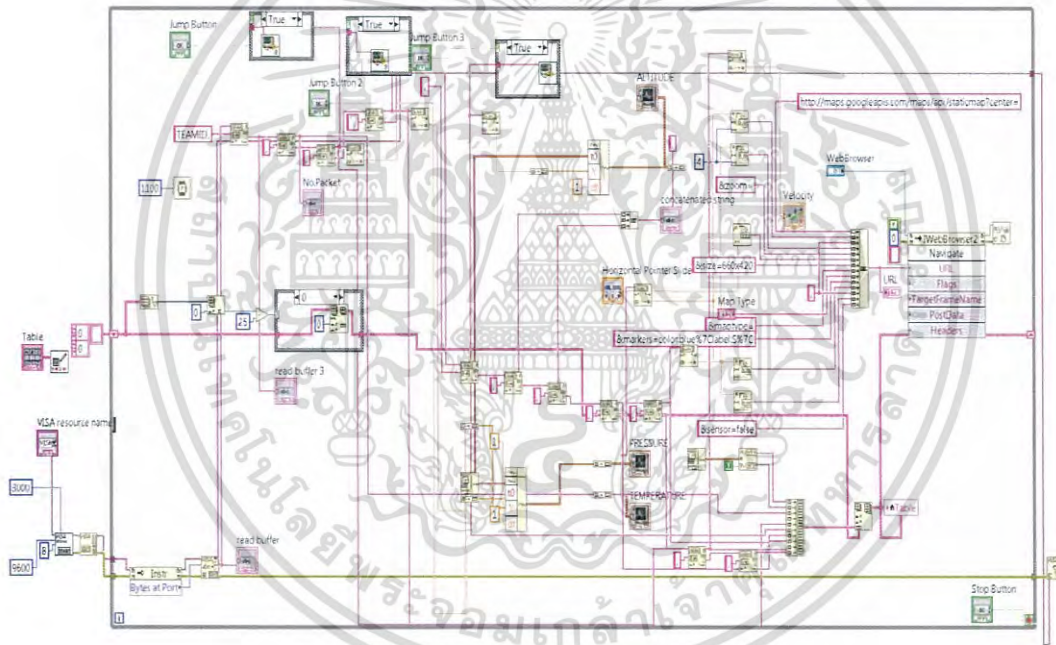
รูปที่ 4.8 ทดสอบการหมุนของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

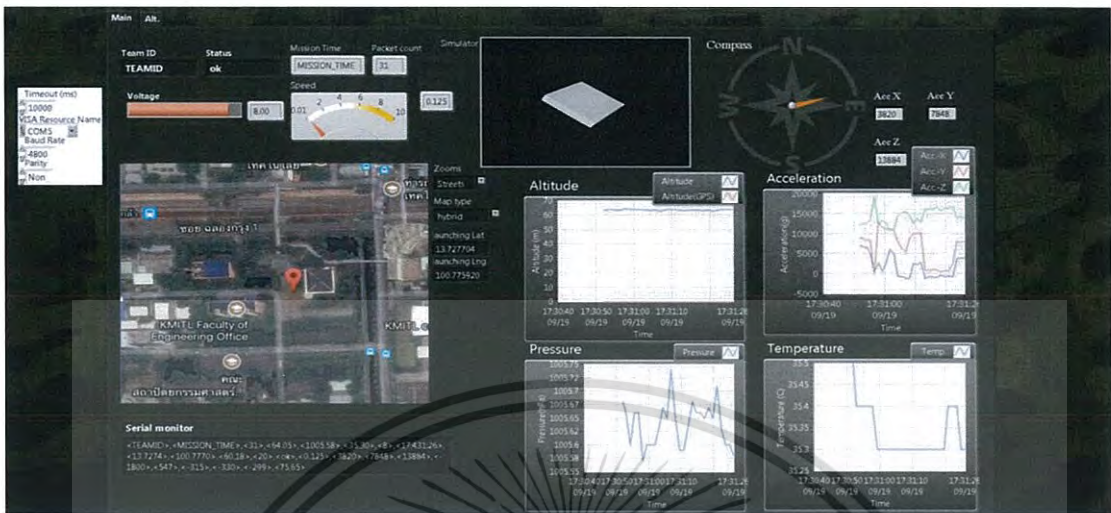
ผลที่ได้จากการทดสอบมอเตอร์คือมอเตอร์หมุนไปตามองศาที่ได้ทำการโปรแกรมไว้แต่จะมีผิดพลาดเล็กน้อยประมาณ ± 2 องศา เนื่องจากมอเตอร์ที่ใช้ในงานร่วมกับตัวควบคุมมอเตอร์ซึ่งทำให้มุมของการหมุนมอเตอร์เวลาชกับตัวควบคุมมอเตอร์นั้นเหลื่อมล้ำกันไปทีละเล็กละน้อยเมื่อผ่านการใช้งานไปสักระยะหนึ่ง

4.3 ผลการทดสอบการทำงานของโปรแกรม

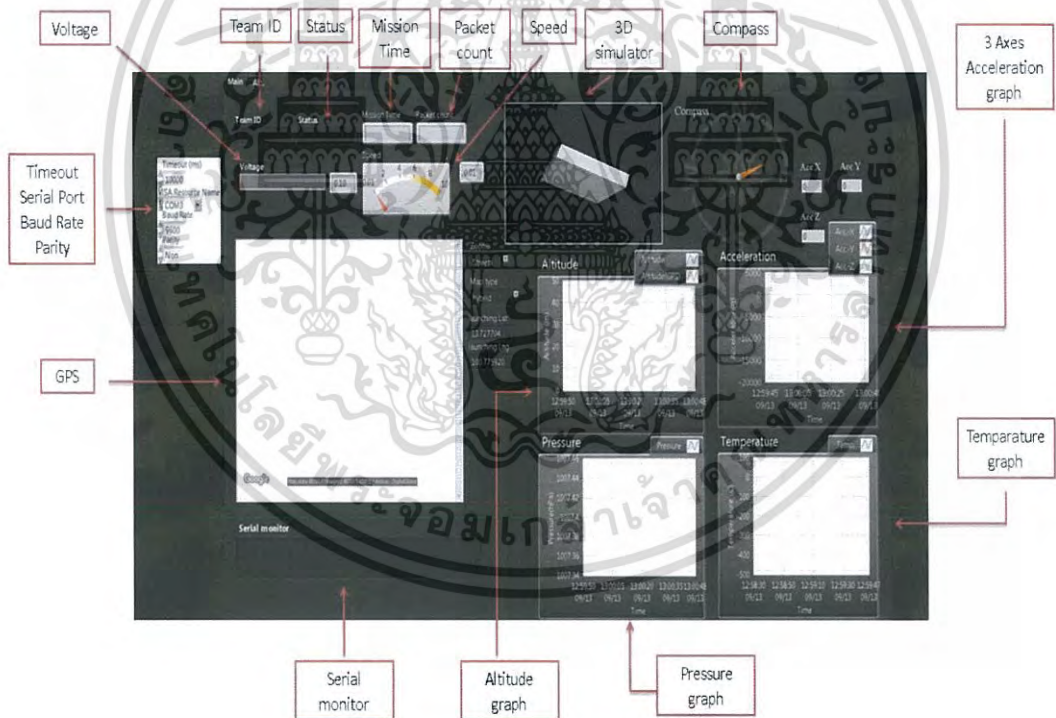
ในการทดสอบการทำงานของโปรแกรมนั้นทำได้โดยการรับค่าจากภาคส่งแล้วทำการแสดงผลในหน้าจอแสดงผลโดยจะมีบล็อกโคแธแกรมของโปรแกรมที่ทำการออกแบบดัง รูปที่ 4.9 และหน้าจอแสดงผลดังแสดงใน รูปที่ 4.10 โดยจะแสดงส่วนประกอบของหน้าจอแสดงผลดังรูปที่ 4.11



รูปที่ 4.9 บล็อกโคแธแกรมที่ออกแบบ



รูปที่ 4.10 หน้าจอแสดงผล



รูปที่ 4.11 ส่วนประกอบต่างๆของหน้าจอแสดงผล

โดยข้อมูลที่รับจากภาคส่งนั้นจะมีรูปแบบแสดงดังรูปที่ 4.12 ซึ่งเป็นรูปแบบที่บังคับเพื่อใช้ในการแข่งขันและจากนั้นทำการทดลองจับบิตของข้อมูลที่รับได้จากแบบจำลองดาวเทียมขนาดเล็กโดยใช้อุปกรณ์ Oscilloscope เพื่อตรวจสอบว่าบิตข้อมูลที่รับได้นั้นตรงกับข้อมูลที่ส่งมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือไม่ซึ่งแสดงบิตข้อมูลที่จับได้ในรูปที่ 4.13 โดยผลลัพธ์ที่ได้คือบิตข้อมูลที่ได้มีการเรียงลำดับตามรูปแบบข้อมูลตามข้อที่ 4.12

```
<TEAM ID>,<MISSION TIME>,<PACKET COUNT>,<ALTITUDE>,<PRESSURE>,<TEMP>,<VOLTAGE>,<GPS TIME>,<GPS LATITUDE>,<GPS LONGITUDE>,<GPS ALTITUDE>,<GPS SATS>,<TILT X>,<TILT Y>,<TILT Z>,<SOFTWARE STATE>,<[BONUS STATUS]>,<GPS SPEED>,<GYRO X>,< GYRO Y >,< GYRO Z >,< MAG >
```

รูปที่ 4.12 รูปแบบของข้อมูลที่ทำกรรับจากแบบจำลองดาวเทียมขนาดเล็ก



รูปที่ 4.13 ค่าสัญญาณบิตข้อมูลเมื่อดูจากเครื่องออสซิลโลสโคป

โดยจากรูปที่ 4.13 คือค่าสัญญาณบิตข้อมูลที่วัดจากเครื่องออสซิลโลสโคป ที่ส่งแบบ UART ซึ่งจะต้องทำการกลับบิตก่อนที่จะเทียบกับข้อมูลโดยจะแสดงข้อมูลเริ่มจากบิตสตอป บิตสตาร์ท ตามด้วยข้อมูลและจะคั่นทุกไบต์ด้วยบิตสตอป บิตสตาร์ท เป็นอย่างนี้ทุกไบต์จนถึงข้อมูลตัวสุดท้าย



รูปที่ 4.14 เปรียบเทียบตำแหน่ง GPS จากโปรแกรมแสดงผลกับโทรศัพท์

จากรูปที่ 4.14 คือการทดสอบการรับค่า GPS Latitude และ GPS Longitude มาเพื่อแสดงพิกัดตำแหน่งของแบบจำลองดาวเทียมขนาดเล็กในหน้าจอแสดงผลโดยผลที่ได้ นั่นคือรูปทางซ้ายมือในรูปที่ 4.14 และนำมาทำการเปรียบเทียบกับตำแหน่งพิกัดที่แสดงผลจากโทรศัพท์ที่ได้ตั้งรูปขวามือของรูปที่ 4.14 ผลลัพธ์ที่ได้ นั่นคือตำแหน่งของทั้ง 2 มีค่าใกล้เคียงกัน โดยแต่ละค่าที่ทำการทดสอบนั้นแสดงได้ดังตารางที่ 4.2

ตารางที่ 4.2 ค่าตำแหน่ง GPS จากโปรแกรมแสดงผลเทียบกับโทรศัพท์

ตำแหน่ง GPS จากโปรแกรมแสดงผล		ตำแหน่ง GPS จากโทรศัพท์	
GPS Latitude	GPS Longitude	GPS Latitude	GPS Longitude
13.727327	100.776963	13.727328	100.776967
13.727320	100.776956	13.727330	100.776965
13.727340	100.776982	13.727330	100.776966
13.727334	100.776978	13.727335	100.776962
13.727338	100.776980	13.727334	100.776970

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

จากการปฏิญานิพนธ์เรื่อง การสร้างแบบจำลองดาวเทียมขนาดเล็กสำหรับการแข่งขัน (2): ภาครับ ซึ่งแบ่งการทำงานออกเป็น 3 ส่วนหลักๆคือ ส่วนของการทำสายอากาศ ส่วนของโปรแกรม และส่วนของการออกแบบและสร้างระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก โดยได้ทำการศึกษาหลักการต่างๆเพื่อนำมาประยุกต์ใช้ในการทำโครงการซึ่งสามารถสรุปผลการดำเนินงาน ได้ดังนี้

1) สามารถทำการสร้างสายอากาศ Stack microstrip ที่ความถี่กลาง 2.45 GHz ซึ่งนำมาใช้งานกับ xbee ได้

2) สามารถเขียนโปรแกรมเพื่อเก็บค่าสภาพอากาศ (ได้แก่ อุณหภูมิ ความดัน และความสูงจากระดับน้ำทะเลกลางโดยคำนวณจากความดัน) เก็บค่าจาก GPS ที่ระบุถึงตำแหน่ง (ได้แก่ เวลา, ละติจูด, ลองจิจูด, ความสูง และ เลขดาวเทียม) ค่าจากความเร่ง 9 แกน (ได้แก่ accelerometers, gyroscope และ magnetometer) ค่าระดับแรงดันจากแบตเตอรี่ และค่าการนับเวลาภารกิจเป็นวินาทีโดยใช้ Real time clock

3) สามารถเขียนโปรแกรมนำข้อมูลที่รับได้มาแยกแสดงผลออกเป็นแบบกล่องข้อความ (ได้แก่ Team ID, Status, Mission Time, Packet Count และ Serial Monitor) แบบกราฟแสดงผล (ได้แก่ Altitude Graph, 3 Axes Acceleration Graph, Pressure Graph และ Temperature Graph) แบบหน้าจอแสดงผล GPS แบบหน้าจอ 3D simulator แบบเข็มทิศ และแบบ meter (ได้แก่ Voltage และ Speed)

4) สามารถออกแบบและสร้างระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก

5) สามารถสร้างระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็กให้มีความทำงานเป็นสองโหมดคือ โหมดติดตามแบบอัตโนมัติ และโหมดบังคับ manual ด้วยจอยสติ๊ก

5.2 ข้อเสนอแนะ

- สายอากาศ Stack microstrip ที่สร้างขึ้นมามีกำลังขยายน้อย ทำให้การรับข้อมูลในทางภาครับสำหรับระยะทางที่ไกลจากแบบจำลองดาวเทียมขนาดเล็กที่มากกว่า 500 เมตร มีปัญหาการรับข้อมูลไม่ได้บ้าง

- เนื่องจากความถี่ที่เลือกในย่านความถี่ 2.4 GHz เป็นสัญญาณความถี่สูงทำให้ระยะทางที่ใช้ในการส่งได้ไม่ไกลเท่ากับความถี่ต่ำ จึงเป็นเหตุให้ต้องสร้างสายอากาศที่มีอัตราขยายที่สูง
- การทำงานของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็กในโหมดติดตามแบบอัตโนมัตินั้นยังมีการตอบสนองช้ากว่าที่ควร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ดาราศาสตร์ศึกษา (Astronomy Education). “CANSAT คืออะไร?”
<http://www.astroeducation.com/what-is-cansat>.
- [2] Yothin Boonnarat. “ระบบการสื่อสารผ่านดาวเทียม (Satellite Communication System).” <http://rsu-itm640.blogspot.com/2010/08/satellite-communication-system.html>.
- [3] Phakpoom Laotrakul. “ดาวเทียม (Satellites).”
<http://www.gistda.or.th/main/th/node/962>.
- [4] CanSat Competition. “Introduction.”
<http://www.cansatcompetition.com/index.html>.
- [5] CanSat Competition. “Mission.”
<http://cansatcompetition.com/mission.html>.
- [6] CanSat Competition. “Mission guide.”
http://www.cansatcompetition.com/docs/mission_guide_2018_r20171020_r1.4.pdf.
- [7] สำนักงานคณะกรรมการวิจัยแห่งชาติ. “ทฤษฎีพื้นฐานและการประยุกต์ใช้งานสายอากาศ.”
http://doi.nrct.go.th/ListDoi/Download/144454/b6c19ab11332f5d090bc4e08e75b00e2?Resolve_DOI=10.14457/KKU.res.2012.22.
- [8] วิกิพีเดีย. “ความยาวคลื่น.”
<https://th.wikipedia.org/wiki/%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%A2%E0%B8%B2%E0%B8%A7%E0%B8%84%E0%B8%A5%E0%B8%B7%E0%B9%88%E0%B8%99>.
- [9] Artificial motion. “STEPPING MOTOR.”
https://www.nectec.or.th/schoolnet/library/webcontest2003/100team/dlnes137/am/step_motor.html
- [10] Arduitrronics. “Arduino and Motor Control.”
<https://www.arduitronics.com/article/25/arduino-and-motor-control-part-3-stepper-motor>
- [11] วิชาการ.คอม. “จอยสติคแบบอนาล็อก.”
<http://www.vcharkarn.com/blog/41099/13064>
- [12] พูนศักดิ์ พรเพิ่มพูน. “ระบบไร้สายโดยใช้ Zigbee เพื่อควบคุมและติดตามสถานะเครื่องจักรและเซ็นเซอร์ในโรงงานผ่านเครือข่ายอินเทอร์เน็ต.” วิทยานิพนธ์ปริญญาวิทยาศาสตร

มหบัณฑิต, สาขาวิชาวิทยาการคอมพิวเตอร์และระบบสารสนเทศ คณะสถิติประยุกต์, สถาบันบัณฑิตพัฒนบริหารศาสตร์, 2556.

- [13] researchgate. “IEEE820.15.4/ZigBee protocol stack architecture.”
https://www.researchgate.net/figure/226297449_fig2_Fig-22-IEEE820154ZigBee-protocol-stack-architecture.
- [14] Internetworking ICT KKU , Sec 3. “ย่านความถี่ใช้งานตามมาตรฐาน.”
<https://sites.google.com/site/322462wirelesspan/wireless-personal-area-network/thekhnoiyi-thi-keiywkhxng-kab-wireless-pan/zigbee>.
- [15] Wikipedia. “IEEE 802.15.4.”
https://en.wikipedia.org/wiki/IEEE_802.15.4.
- [16] Wikimedia commons. “UART XBee Data Format.”
https://commons.wikimedia.org/wiki/File:UART_XBee_Data_format.jpg.
- [17] Digi International. “AT commands.”
<http://docs.digi.com/display/RFKitsCommon/Command+mode>.
- [18] Digi International. “API frames.”
<http://docs.digi.com/display/XCTU/Concepts+and+terminology>.
- [19] มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร. “สายอากาศไมโครสตริปแบบชั้นที่มีโพลาไรซ์วงกลม สำหรับเครือข่าย 2.4 GHz.” /www.ectihailand.org/assets/papers/1660_pub_49.pdf.
- [20] pbcllinear. “Stepper Motor NEMA 17, Stepper Motor NEMA 23.”
<http://www.pbcllinear.com/Download/DataSheet/Stepper-Motor-Support-Documnet.pdf>.
- [21] NMEC. “Aluminium profile.”
<http://www.nmec.co.th/aluminium-profile>.
- [22] iGiSMAP. “Formula to Find Bearing or Heading angle between two points: Latitude Longitude.” <http://www.igismap.com/formula-to-find-bearing-or-heading-angle-between-two-points-latitude-longitude>.
- [23] Andrew Hedges. “Finding distances based on Latitude and Longitude.”
<https://andrew.hedges.name/experiments/haversine>.
- [24] QQonlineTrading. “3.2" HVGA LCD TFT Non-Touch Screen, Arduino MEGA-2560 Shield Compatible.” <http://qqtrading.com.my/32-inch-lcd-tft-screen-for-arduino-mega-2560-shield-compatible>.

- [25] Dfrobot. “TB6600 Stepper Motor Driver SKU: DRI0043.”
https://www.dfrobot.com/wiki/index.php/TB6600_Stepper_Motor_Driver_SKU:_DRI0043.
- [26] Art of Circuits. “Dual Axis XY Joystick Module.”
<http://artofcircuits.com/product/dual-axis-xy-joystick-module>.
- [27] LCDTVTHAILAND. “LabVIEW คืออะไร? ประโยชน์และการใช้งาน LabView.”
<https://www.lcdtvthailand.com/webboard/index.php?topic=392759.0>
- [28] National Instruments. “3D Picture Control.”
<https://forums.ni.com/t5/UAV-Interest-Group/3D-Picture-Control-to-show-orientation-Roll-Pitch-and-Yaw/gpm-p/3531888>.
- [29] RIGOL. “Spectrum Analyzers.”
<http://www.rigol.eu/products/spectrum-analyzers/>
- [30] ThaiEasyElec. “Arduino Mega 2560.”
<http://www.thaieasyelec.com/products/development-boards/arduino-mega-2560-detail.html>.
- [31] sparkfun. “XBee Pro 60mW U.FL Connection - Series 1 (802.15.4).”
<https://www.sparkfun.com/products/8710>.
- [32] sparkfun. “SparkFun XBee Explorer Dongle.”
<https://www.sparkfun.com/products/11697>.
- [33] พิสิทธ์ราชมงคล. “ออสซิลโลสโคป.”
<http://www.rmutphysics.com/charud/virtualexperiment/labphysics2/meter>.

ภาคผนวก

โค้ดการทำงานของระบบสายอากาศสำหรับติดตามแบบจำลองดาวเทียมขนาดเล็ก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดสำหรับ Arduino UNO

```

#include <AccelStepper.h> // เรียกใช้ library การขับเคลื่อนมอเตอร์
#include <Bounce2.h> // เรียกใช้ library ควบคุม button
const int pinSwEnable = 7;
unsigned long debounceDelay = 10;
const int jX = A0; // กำหนดขาแอนะล็อกสำหรับอ่านค่า X
const int pulX = 12; // กำหนดขา pulse สำหรับขับเคลื่อนมอเตอร์ X
const int dirX = 11; // กำหนดขา direction สำหรับขับเคลื่อนมอเตอร์ X
long speedX, valX, mapX; // ตัวแปรการจัดการเคลื่อนไหวของมอเตอร์ X

const int jY = A1; // กำหนดขาแอนะล็อกสำหรับอ่านค่า Y
const int pulY = 8; // กำหนดขา pulse สำหรับขับเคลื่อนมอเตอร์ Y
const int dirY = 9; // กำหนดขา direction สำหรับขับเคลื่อนมอเตอร์ Y
long speedY, valY, mapY; // ตัวแปรการจัดการเคลื่อนไหวของมอเตอร์ Y
long sw;
// ตัวแปรที่ใช้สำหรับ library AccelStepper
const int maxSpeed = 1000; // ความเร็วสูงสุดของมอเตอร์
const int minSpeed = 0; // ความเร็วต่ำสุดของมอเตอร์
const float accelerazione = 1000.0; // จำนวน step ต่อวินาที ในการเร่งความเร็ว

const int treshold = 30;
long tresholdUp, tresholdDown;
boolean abilitato, moveX, moveY, enable; // ตัวแปรการจัดการการเคลื่อนไหว

Bounce btnEnable = Bounce(); // เรียกใช้ button จาก library Bounce

AccelStepper motorX(AccelStepper::DRIVER, pulX, dirX);
AccelStepper motorY(AccelStepper::DRIVER, pulY, dirY);
int pos;
int stringplace = 0;
String timeUp;
char inByte;
char *p, *i;
char text[200];
#define MAX_STRING_LEN 200
double pi = 3.14159265359; // ค่า pi

```

```

double lat1 = 13.727528*pi/180;
double lon1 = 100.776277*pi/180;
double B = lon1*180/pi;
double lat2 = 13.727528*pi/180;
double lon2 = 100.776309*pi/180;
double dlon = lon2 - lon1;
double dlat = lat2 - lat1;
double C,D;

void setup() {
  Serial.begin(4800);
  speedX = speedY = 0;
  enable = true; //status ของ button

  pinMode(pinSwEnable, INPUT_PULLUP);
  btnEnable.attach(pinSwEnable);
  btnEnable.interval(debounceDelay);
  tresholdDown = (maxSpeed / 2) - treshold;
  tresholdUp = (maxSpeed / 2) + treshold;
  motorX.setMaxSpeed(maxSpeed); //ตั้ง maxspeed ของ motor x
  motorX.setSpeed(minSpeed); //ตั้ง speed ของ motor x
  motorX.setAcceleration(accelerazione); //ตั้ง ค่าความเร่ง ของ motor x
  motorY.setMaxSpeed(maxSpeed); //ตั้ง maxspeed ของ motor Y
  motorY.setSpeed(minSpeed); //ตั้ง speed ของ motor Y
  motorY.setAcceleration(accelerazione); //ตั้ง ค่าความเร่ง ของ motor Y
}

void loop() {
  checkEnable();
  if (enable == 1) //เช็คค่า button on หรือ off

  if(Serial.available()) { //เข้ารูปเมื่อมีข้อมูลเข้ามาทาง serial port
    if(Serial.available()>0){
      String inByte= Serial.readStringUntil("\n");
      Serial.print("Receive:");

```

```

Serial.println(inByte);
Serial.print("inByte length =");
Serial.println(inByte.length());
inByte.toCharArray(text,200);
String text10 = subStr(text,">,<", 10);           // GPS Lat to LCD
String text11 = subStr(text,">,<", 11);           // GPS Lon to LCD
String text12 = subStr(text,">,<", 12);           // GPS Alt to LCD

String sum = "<"+String(text10)+">,<"+ String(text11)+">,<"+String(text12)+">";
Serial.println(sum);
C = text10.toDouble();
D = text11.toDouble();
long E = D-B;
double X = cos(C*pi/180) * sin((E)*pi/180);
double Y = cos(lat1) * sin(C*pi/180) -sin(lat1) * cos(C*pi/180) * cos((E)*pi/180);
double beta1 = (atan2(X,Y)*180)/pi;

Serial.print("angle23 = ");
Serial.println(angle23,6);

motorX.moveTo(angle23*5.556);
motorX.runToPosition();
Serial.print("Move to ");
Serial.println(angle23);

}
}

if(enable==0){
valX = analogRead(jX);
valY = analogRead(jY);
mapX = map(valX, 0, 1023, minSpeed, maxSpeed);
mapY = map(valY, 0, 1023, minSpeed, maxSpeed);
pilotaMotori(mapX, mapY);
delay(2);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void pilotaMotori(long mapX, long mapY) {

    if (mapX <= tresholdDown) {
        //x Counterclockwise
        speedX = -map(mapX, tresholdDown, minSpeed, minSpeed, maxSpeed);
        moveX = true;
    } else if (mapX >= tresholdUp) {
        //x clockwise
        speedX = map(mapX, maxSpeed, tresholdUp, maxSpeed, minSpeed);
        moveX = true;
    } else {
        //motor x stop
        speedX = 0;
        moveX = false;
    }

    if (mapY <= tresholdDown) {
        //motor y up
        speedY = -map(mapY, tresholdDown, minSpeed, minSpeed, maxSpeed);
        moveY = true;
    } else if (mapY >= tresholdUp) {
        //motor y down
        speedY = map(mapY, maxSpeed, tresholdUp, maxSpeed, minSpeed);
        moveY = true;
    } else {
        // motor y stop
        speedY = 0;
        moveY = false;
    }

    if (moveX) {
        motorX.setSpeed(speedX);
        motorX.run();
        Serial.print("speed X =");
        Serial.println(speedX);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    motorX.stop();
}
if (moveY) {
    motorY.setSpeed(speedY);
    motorY.run();
    Serial.print("speed Y =");
    Serial.println(speedY);
} else {
    motorY.stop();
}
}
void checkEnable() { //function สำหรับการเช็คสถานะของ button

    btnEnable.update();

    if (btnEnable.fell()) {
        enable = !enable;
    }

}

char* subStr (char* str, char *delim, int index) { //function สำหรับการแยกข้อมูล
    char *act, *sub, *ptr;
    static char copy[MAX_STRING_LEN];
    int i;

    // Since strtok consumes the first arg, make a copy
    strcpy(copy, str);

    for (i = 1, act = copy; i <= index; i++, act = NULL) {
        //Serial.print(".");
        sub = strtok_r(act, delim, &ptr);
        if (sub == NULL) break;
    }
    return sub;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดสำหรับ Arduino mega 2560

```
#include <UTFT.h> //เรียกใช้ library ของจอ LCD
#include <AccelStepper.h> //เรียกใช้ library ของ ควบคุม button
extern uint8_t SmallFont[];
extern uint8_t BigFont[];
extern uint8_t Grotesk24x48[];
extern uint8_t SevenSegNumFont[];
UTFT myGLCD(CTE32HR,38,39,40,41);
double pi = 3.14159265359;
double lat1 = 13.727528*pi/180;
double lon1 = 100.776277*pi/180;
double lat2 = 13.727528*pi/180;
double lon2 = 100.776309*pi/180;
double dlon = lon2 - lon1;
double dlat = lat2 - lat1;
double C,D;
float calibrate = 1.8;
int R = 6373;
int pos;
int stringplace = 0;
String timeUp;
String nmea[15];
char inByte;
char *p, *i;
char text[200];
#define MAX_STRING_LEN 200
const int pulY = 47; //กำหนดค่า pulse สำหรับขับเคลื่อนมอเตอร์ Y
const int dirY = 9; //กำหนดค่า direction สำหรับขับเคลื่อนมอเตอร์ Y
const int maxSpeed = 1000;
const int minSpeed = 0; //
const float accelerazione = 1000.0;
AccelStepper motorY(AccelStepper::DRIVER, pulY, dirY);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void setup() {

  Serial.begin(4800);
  Serial1.begin(4800);

  motorY.setMaxSpeed(maxSpeed);
  motorY.setSpeed(minSpeed);
  motorY.setAcceleration(accelerazione);

  myGLCD.InitLCD();
  myGLCD.clrScr();
  myGLCD.setColor(0, 147, 159);
  myGLCD.setBackColor(VGA_BLACK);
  myGLCD.setFont(Grotesk24x48);
  String Lat = "GPS Lat: " ;
  myGLCD.print(Lat,LEFT,16);
  String Lon = "GPS Lon: " ;
  myGLCD.print(Lon, LEFT, 70);
  String Alt = "GPS Alt: " ;
  myGLCD.print(Alt, LEFT, 124);
}

void loop() {

  if(Serial.available()) {
    if(Serial.available()>0){
      String inByte= Serial.readStringUntil('\n');
      Serial.print("Receive:");
      Serial.println(inByte);
      Serial.print("inByte length =");
      Serial.println(inByte.length());
      inByte.toCharArray(text,200);
      String text10 = subStr(text,">,<", 10);           // GPS Lat to LCD
      String text11 = subStr(text,">,<", 11);           // GPS Lon to LCD
      String text12 = subStr(text,">,<", 12);
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

String Lat = String(text10);
myGLCD.print(Lat,RIGHT,16);           // print GPS Lat on LCD
String Lon = " " + String(text11);
myGLCD.print(Lon, RIGHT, 70);         // print GPS Lon on LCD
String Alt = " " + String(text12) + " m ";
myGLCD.print(Alt, RIGHT, 124);        // print GPS Alt on LCD

double a = (sin(dlat/2))*(sin(dlat/2)) + cos(lat1) * cos(lat2) *
            sin(dlon/2)*(sin(dlon/2));
double c = 2 * atan2( sqrt(a), sqrt(1-a) );
double d = R * c *1000;
double alt =5 - calibrate;
double angle17 = 90-(atan(alt/d)*180/pi);

motorY.moveTo(angle17*5.557);
motorY.runToPosition();
Serial.print("d = ");
Serial.println(d,10);
Serial.print("angle17 = ");
Serial.println(angle17,10);

String mo = "move to: " +String(angle17,3) + " deg " ;
myGLCD.print(mo, CENTER, 180);
String track = "----Auto tracking Mode----";
myGLCD.print(track, CENTER, 256);
}
}

else{
String line1 = " ";
myGLCD.print(line1,CENTER,16);
String line2 = " ";
myGLCD.print(line2, CENTER, 70);
String line3 = " ";
myGLCD.print(line3, CENTER, 124);
String line4 = " ";
myGLCD.print(line4, CENTER, 256);
}
}

```

```

String man = "-----Manual Mode-----";
myGLCD.print(man, CENTER, 180);
}
}

```

```

char* subStr (char* str, char *delim, int index) {
char *act, *sub, *ptr;
static char copy[MAX_STRING_LEN];
int i;

// Since strtok consumes the first arg, make a copy
strcpy(copy, str);

for (i = 1, act = copy; i <= index; i++, act = NULL) {
//Serial.print(".");
sub = strtok_r(act, delim, &ptr);
if (sub == NULL) break;
}
return sub;
}

```