การสร้างแบบจำลองดาวเทียมขนาดเล็กสำหรับใช้ในการแข่งขัน : ภาคส่ง
CanSat Design for Competition : Science Vehicle

โดย

นายอธิวัตร เปียทอง

นางสาวอรจิรา สิริเศรษฐนันท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

การสร้างแบบจำลองดาวเทียมขนาดเล็กสำหรับใช้ในการแข่งขัน : ภาคส่ง
CanSat Design for Competition : Science Vehicle

โดย

นายอธิวัตร เปียทอง 57011458

นางสาวอรจิรา สิริเศรษฐนันท์ 57011491

อาจารย์ที่ปรึกษา

ดร. พีระเมศร์ โชติกวีกิจญาดา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

21/5/61

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

21/5/61

KMITL วิศวกรรมโทรคมนาคม Telecommunications

ปริญญานิพนธ์ปีการศึกษา 2560
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง การสร้างแบบจำลองดาวเทียมขนาดเล็กสำหรับใช้ในการแข่งขัน : ภาคส่ง
CANSAT DESIGN FOR COMPETITION : SCIENCE VEHICLE

ผู้จัดทำ

1.	นายอธิวัตร	เปียทอง	57011458
2.	นางสาวอรจิรา	สิริเศรษฐนันท์	57011491

..... **อาจารย์ที่ปรึกษา**
(ดร. พีระเมศร์ โชติกวีกิจญาดา)

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความอนุเคราะห์อย่างดียิ่งจากอาจารย์ที่ปรึกษา คือ ดร. พีระเมศร์ โชติกวีกิจญาดา ที่ให้คำแนะนำ คำสั่งสอน ให้ความรู้และความเข้าใจในการทำปริญญานิพนธ์ คอยดูแลเอาใจใส่ อีกทั้งยังคอยเป็นแรงผลักดันที่สำคัญตลอดระยะเวลาในการทำปริญญานิพนธ์นี้ ขอขอบพระคุณท่านอาจารย์ในความห่วงใยและความหวังดีที่ให้แก่พวกเราเป็นอย่างยิ่ง และขอขอบคุณท่านอาจารย์ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้อบรมสั่งสอนและประสิทธิ์ประสาทวิชาความรู้ให้แก่ผู้จัดทำ

สุดท้ายนี้ขอกราบขอบพระคุณบิดามารดา ที่ให้ความสนับสนุนทางด้านทุนทรัพย์ที่สำคัญคือคอยเป็นแรงผลักดัน เป็นกำลังใจในการเรียนและการทำปริญญานิพนธ์นี้จนประสบความสำเร็จตามเป้าหมายที่ตั้งไว้

อธิวัตร เปียทอง
อรจิรา สิริเศรษฐนันท์
ผู้จัดทำ

การสร้างแบบจำลองดาวเทียมขนาดเล็กสำหรับใช้ในการแข่งขัน : ภาคส่ง
CANSAT DESIGN FOR COMPETITION : SCIENCE VEHICLE

โดย นายอธิวัตร เปียทอง 57011458
นางสาวอรจิรา สิริเศรษฐนันท์ 57011491

อาจารย์ที่ปรึกษา ดร. พีระเมศร์ โชติกวีกิจญาดา

บทคัดย่อ

ปริญญานิพนธ์นี้ได้ถูกจัดทำขึ้นโดยมีจุดประสงค์เพื่อที่จะเข้าร่วมการแข่งขัน CanSat ในปี 2018 โดยแบบจำลองดาวเทียมขนาดเล็กในส่วนของภาคส่ง มีหลักการทำงาน ดังนี้ CanSat จะทำหน้าที่เก็บข้อมูลสภาพอากาศหลังจากถูกปล่อยออกจากจรวด ณ ความสูงประมาณ 700 เมตร ซึ่งข้อมูลจากอุปกรณ์เซนเซอร์ต่างๆ ประกอบด้วยอุณหภูมิ,ความดัน, ความสูง, ไจโรสโคป, ความเร่ง 3 แกน, Magnetometer, Real Time Clock, Power Voltage และ GPS จะถูกรวบรวมโดยอุปกรณ์ประมวลผล และส่งไปยังภาคพื้นดิน เมื่อแบบจำลองดาวเทียมขนาดเล็กถึงระดับความสูง 300 เมตร ทำการถ่ายภาพวิดีโอเพื่อเก็บในหน่วยความจำ

ABSTRACT

This thesis is a development of CanSat for the CanSat Competition in 2018. The CanSat will collect weather data with its sensor system and transmit the data back to a base station after it is released from a rocket at an attitude of around 700 m. The collected telemetry data consisting of air temperature, air pressure, altitude, gyroscopic data, 3-axis accelerometric data, magnetometric data, time measured by the clock, GPS data, and Voltage of the built-in battery will be collected by an Arduino device in the CanSat and sent to be processed at the ground station. The CanSat will also take a video clip at an altitude of 300 meters of the ground beneath it and stored the clip in its memory.

สารบัญ

หน้า

กิตติกรรมประกาศ I
บทคัดย่อ II
สารบัญ III
สารบัญรูป VI
สารบัญตาราง X

บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ 1
1.3 ขอบเขตของปริญญานิพนธ์ 2

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 4
2.1 แบบจำลองดาวเทียมขนาดเล็ก (CANSAT) 4
2.2 รายละเอียดการแข่งขัน CANSAT ประจำปี 2018 5
2.3 ระบบย่อยเซนเซอร์ (SENSOR SUBSYSTEM) 7
2.3.1 เซนเซอร์วัดสภาพอากาศ GY-BMEP280 7
2.3.2 เซนเซอร์ GPS GY-NEO6MV2 9
2.3.3 เซนเซอร์ POWER VOLTAGE 16
2.3.4 เซนเซอร์ความเร่ง 9 แกน GY-9250 16
2.3.5 PI CAMERA REV 1.3 17
2.4 ระบบย่อยพลังงานไฟฟ้า (ELECTRICAL POWER SUBSYSTEM) 18
2.4.1 แบตเตอรี่ LI-ION RECHARGEABLE 18
2.4.2 LM 7805 REGULATOR 19
2.5 ระบบย่อยการสื่อสารและการจัดการข้อมูล (COMMUNICATION AND DATA HANDLING SUBSYSTEM) 19
2.5.1 หน่วยประมวลผล (PROCESSOR) 19
2.5.2 หน่วยความจำ (MEMORY) 21
2.5.3 โมดูลนาฬิกาแบบเรียลไทม์ (REAL-TIME CLOCK) 22
2.5.4 สายอากาศ (ANTENNA) 22

สารบัญ (ต่อ) หน้า

2.5.5 โมดูลการสื่อสาร XBEE PRO S1 24
2.5.6 รูปแบบการสื่อสารของการรับ-ส่งข้อมูล 25
2.5.6.1 การสื่อสารแบบ I^2C 25
2.5.6.2 การสื่อสารแบบ UART 29
2.5.6.3 การสื่อสารแบบ ANALOG 30
2.6 ระบบย่อยทางกลไก (MECHANICAL SUBSYSTEM) 30
2.6.1 AERO-BRAKING HEAT SHIELD 30
2.6.2 ร่มชูชีพ (PARACHUTE) 31
บทที่ 3 การออกแบบและจัดทำแบบจำลองดาวเทียมขนาดเล็ก 36
3.1 การออกแบบและสร้างแบบจำลองดาวเทียมขนาดเล็ก 37
3.1.1 การเลือกใช้ SENSOR SUBSYSTEM ที่เหมาะสม 37
3.1.2 การเลือกใช้ ELECTRICAL POWER SUBSYSTEM ที่เหมาะสม 42
3.1.3 การเลือกใช้ COMMUNICATION AND DATA HANDLING SUBSYSTEM ที่เหมาะสม 42
3.1.4 การออกแบบ MICROSTRIP DIPOLE ANTENNA 2.4 GHZ 45
3.1.5 การออกแบบและสร้างโครงสร้างในแต่ละ LAYER 47
3.1.6 การออกแบบและสร้าง MECHANICAL SUBSYSTEM 50
3.1.7 การออกแบบและสร้าง ELECTRICAL POWER SUBSYSTEM 58
3.2 การทำงานของระบบแบบจำลองดาวเทียมขนาดเล็ก 59
3.2.1 โปรแกรมการทำงานของระบบย่อยทั้งหมดและการออกแบบ FLIGHT SOFTWARE STATE 59
3.3 การจัดเก็บผลการทดสอบ 66
3.3.1 การทดสอบการทำงานของ SENSOR SUBSYSTEM 66
3.3.2 การทดสอบการทำงานของ ELECTRICAL POWER SUBSYSTEM 68
3.3.3 การทดสอบการทำงานของ COMMUNICATION AND DATA HANDLING SUBSYSTEM 68
3.3.4 การทดสอบการทำงานของ MECHANICAL SUBSYSTEM 68

สารบัญ (ต่อ)

หน้า

บทที่ 4	**ผลการทดสอบ**	**70**
	4.1 ผลการทดสอบการทำงานของ SENSOR SUBSYSTEM	70
	4.1.1 ผลการทดสอบของ GY-BMEP280	70
	4.1.2 ผลการทดสอบของ GPS GY-NEO6MV2	72
	4.1.3 ผลการทดสอบของความสูงที่เปลี่ยนแปลงจากระดับน้ำทะเลกลาง	72
	4.1.4 ผลการทดสอบของ POWER VOLTAGE	74
	4.1.5 ผลการทดสอบของ GY-9250MPU9250	75
	4.1.6 ผลการทดสอบของ RASPBERRY PI CAMERA BOARD V1.3	87
	4.2 ผลการทดสอบการทำงานของ ELECTRICAL POWER SUBSYSTEM	88
	4.3 ผลการทดสอบการทำงานของ COMMUNICATION AND DATA HANDLING SUBSYSTEM	89
	4.3.1 ผลการทดสอบของ REAL-TIME CLOCK MODULES DS1307	89
	4.3.2 ผลการทดสอบของ MICROSTRIP DIPOLE ANTENNA	90
	4.4 ผลการทดสอบการทำงานของ MECHANICAL SUBSYSTEM	93
	4.4.1 ผลการทดสอบของ AERO-BRAKING HEAT SHIELD	93
	4.4.2 ผลการทดสอบของร่มชูชีพ (PARACHUTE)	94
	4.5 ผลการตรวจสอบความถูกต้องและความพร้อมของกลไกการทำงานตามภารกิจการแข่งขัน	96
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**99**
	5.1 สรุปผล	99
	5.2 ข้อเสนอแนะ	101
บรรณานุกรม		
ภาคผนวก	โค้ดสำหรับการทำงานของ SENSOR SUBSYSTEM	

สารบัญรูป

รูปที่		หน้า
1.1	แผนภาพโดยรวมของแบบจำลองดาวเทียมขนาดเล็ก (CANSAT)	2
1.2	บล็อกไดอะแกรมในส่วนภาคส่ง (SCIENCE VEHICLE)	3
2.1	โมดูล BME280	7
2.2	GPS GY-NEO6MV2 MODULE	9
2.3	ตัวอย่างประโยค NMEA และความหมายของข้อมูลแต่ละส่วนในประโยค	11
2.4	ตัวต้านทาน 1/4W	16
2.5	วงจรการวัดแรงดันไฟฟ้า	16
2.6	เซ็นเซอร์โมดูล GY-9250	17
2.7	RASPBERRY PI CAMERA BOARD V1.3	17
2.8	แบตเตอรี่ LI-ION 9 V (800 MAH)	18
2.9	LM7805 REGULATOR	19
2.10	ไมโครคอนโทรลเลอร์ ARDUINO PRO MINI 328	19
2.11	RASPBERRY PI ZERO	21
2.12	MICRO SD CARD SANDISK 8 GB	21
2.13	TINY REAL TIME CLOCK MODULES	22
2.14	สายอากาศแบบ HALF-WAVELENGTH DIPOLE	23
2.15	รูปแบบการแพร่กระจายคลื่นของสายอากาศไดโพล	23
2.16	การแพร่กระจายคลื่นของสายอากาศไดโพล	24
2.17	โมดูลการสื่อสารไร้สาย XBEE PRO S1	24
2.18	รูปแบบการเขียน/อ่านข้อมูลแบบ I^2C BUS	26
2.19	TIMING DIAGRAM การทางานของ I^2C BUS	27
2.20	รูปแบบการสื่อสารแบบ UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER	29
2.21	EXOMARS 2016 SCHIAPARELLI DESCENT SEQUENCE	31
2.22	ร่มชูชีพแบบ SEMI-SPHERICAL	31
2.23	ร่มชูชีพแบบ PARAPENT	32
2.24	ร่มชูชีพแบบ FLAT PARACHUTE	32
2.25	ร่มชูชีพแบบ CROSS	33
2.26	การกระทำของแรงขณะที่วัตถุกำลังเคลื่อนที่ลง	34

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.1	บล็อกไดอะแกรมการทำงานของอุปกรณ์ที่นำเสนอ	36
3.2	MICROSTRIP DIPOLE ANTENNA (ความถี่ 2.4 GHZ)	45
3.3	กราฟ S-PARAMETER	46
3.4	กราฟ VOLTAGE STANDING WAVE RATIO (VSWR)	46
3.5	แบบรูปการแพร่กระจาย (FARFIELD 3D)	47
3.6	วงจร SCHEMATIC การเชื่อมต่อโดยรวมโครงสร้างของ CANSAT	49
3.7	โครงสร้างของ CANSAT	49
3.8	การจัดองค์ประกอบภายในและภายนอกแบบจำลองดาวเทียมขนาดเล็ก	50
3.9	ขนาดเส้นผ่านศูนย์กลางและความสูงของ HEAT SHIELD ขณะใช้งานและไม่ใช้งาน	51
3.10	ขนาดเส้นผ่านศูนย์กลางและความสูงแต่ละชั้นของแบบจำลองดาวเทียมขนาดเล็ก	51
3.11	แรงที่กระทำต่อ CANSAT ขณะเคลื่อนที่	52
3.12	ค่าสัมประสิทธิ์แรงดึง (DRAG) ของรูปร่าง WEDGES, CONES และ SIMILAR	53
3.13	กราฟแสดงความสัมพันธ์ระหว่างมวลของ CANSAT ต่อขนาดรัศมีของร่มชูชีพที่ระดับความเร็วต่างๆ	54
3.14	รูปแบบของร่มชูชีพแบบ SEMI-SPHERICAL	55
3.15	แบบของชิ้นส่วนร่มแต่ละชิ้น	55
3.16	ร่มชูชีพสภาพเสร็จสมบูรณ์และมวลของร่มชูชีพ	56
3.17	โครงสร้างการปลดสลัก	56
3.18	กลไกการปล่อยใช้งาน HEAT SHIELD	57
3.19	กลไกการปล่อย HEAT SHIELD ทิ้ง	57
3.20	กลไกการปล่อยร่มชูชีพใช้งาน	58
3.21	บล็อกไดอะแกรมพลังงานไฟฟ้า	58
3.22	FLOWCHART แสดงการทำงานโดยรวม	60
3.23	สถานะในการทำงานของแต่ละขั้นตอน	62
3.24	FLOWCHART แสดงการอ่านค่าจากเซ็นเซอร์ทั้งหมด	63

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.1	ค่าสัญญาณบิตข้อมูลจากโมดูลGY-BMEP280	70
4.2	ค่าสัญญาณบิตข้อมูลจากโมดูลGY-BMEP280 (ค่าอุณหภูมิ)	71
4.3	สัญญาณบิตข้อมูลจากโมดูลGY-BMEP280 (ค่าความดัน)	71
4.4	ค่าสัญญาณบิตข้อมูลจากโมดูล GY-NEO6MV2	72
4.5	การวัดค่าของแหล่งจ่ายไฟโดยใช้ POWER VOLTAGE	74
4.6	การปรับระดับทิศทางการเคลื่อนที่ของโมดูลตามมุมในแนวแกน X	75
4.7	มุมจากการอ่านค่าจากตัวโมดูลในแนวแกน X	75
4.8	การปรับระดับทิศทางการเคลื่อนที่ของโมดูลตามมุมในแนวแกน Y	77
4.9	มุมจากการอ่านค่าจากตัวโมดูลในแนวแกน Y	77
4.10	การปรับระดับทิศทางการเคลื่อนที่ของโมดูลตามมุมในแนวแกน Z	79
4.11	มุมจากการอ่านค่าจากตัวโมดูลในแนวแกน Z	79
4.12	การเทียบทิศของโมดูลกับเข็มทิศแบบตลับและแสดงค่าเป็นองศาของทิศเหนือ	81
4.13	การเทียบทิศของโมดูลกับเข็มทิศแบบตลับและแสดงค่าเป็นองศาของทิศตะวันออก	81
4.14	การเทียบทิศของโมดูลกับเข็มทิศแบบตลับและแสดงค่าเป็นองศาของทิศใต้	82
4.15	การเทียบทิศของโมดูลกับเข็มทิศแบบตลับและแสดงค่าเป็นองศาของทิศตะวันตก	82
4.16	ค่าสัญญาณบิตจากโมดูล GY-9250MPU9250	84
4.17	ค่าสัญญาณบิตข้อมูลจากโมดูล GY-9250MPU9250 (ไบต์ที่ 4-8)	84
4.18	ค่าสัญญาณบิตข้อมูลจากโมดูล GY-9250MPU9250 (ไบต์ที่ 9-12)	85
4.19	ค่าสัญญาณบิตข้อมูลจากโมดูล GY-9250MPU9250 (ไบต์ที่ 13-17)	85
4.20	ค่าสัญญาณบิตข้อมูลจากโมดูล GY-9250MPU9250 (ไบต์ที่ 18-23)	86
4.21	กระบวนการแปลงไฟล์จากการอัดวิดีโอ	87
4.22	ค่าสัญญาณบิตจากโมดูล REAL TIME CLOCK (DS 1307)	89
4.23	ค่าสัญญาณบิตจากโมดูล REAL TIME CLOCK (DS 1307)	90
4.24	MICROSTRIP DIPOLE ANTENNA	90
4.25	กราฟแสดง S-PARAMETER	91
4.26	กราฟแสดง VOLTAGE STANDING WAVE RATIO (VSWR)	91
4.27	แบบรูปการแพร่กระจาย	92
4.28	การทดสอบการรับส่งสายอากาศ	92

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.29	กราฟความสัมพันธ์เปรียบเทียบความเร็วที่ได้จากการคำนวณทางทฤษฎีกับความเร็วที่ได้จากการทดสอบที่น้ำหนักต่างๆของ AERO-BRAKING HEAT SHIELD	94
4.30	กราฟความสัมพันธ์เปรียบเทียบความเร็วที่ได้จากการคำนวณทางทฤษฎีกับความเร็วที่ได้จากการทดสอบที่น้ำหนักต่างๆ	95
4.31	การเตรียมอุปกรณ์และการทดสอบอุปกรณ์ ณ ตาคลี จ.นครสวรรค์	98
4.32	ข้อมูลการบันทึกค่าจากการเก็บค่าสภาพอากาศ	98

สารบัญตาราง

ตารางที่		หน้า
2.1	ข้อกำหนดพื้นฐานของการแข่งขัน	5
2.2	คำอธิบาย PIN ของ GPS GY-NEO6MV2 MODULE	9
2.3	รูปแบบการจัดเรียงข้อมูลตามมาตรฐานของ NMEA	11
2.4	รูปแบบการจัดเรียงข้อมูลประเภท GGA	12
2.5	แสดงค่า FIX INDICATOR	12
2.6	รูปแบบการจัดเรียงข้อมูลประเภท GLL	13
2.7	รูปแบบการจัดเรียงข้อมูลประเภท GSA	13
2.8	รูปแบบการจัดเรียงข้อมูลประเภท GSV	14
2.9	รูปแบบการจัดเรียงข้อมูลประเภท RMC	15
2.10	รูปแบบการจัดเรียงข้อมูลประเภท VTG	15
2.11	คุณลักษณะของ RASPBERRY PI CAMERA BOARD	18
2.12	คุณลักษณะเด่นของ ARDUINO PRO MINI 328	20
2.13	คุณสมบัติทางเทคนิคของ RASPBERRY PI ZERO	21
2.14	รายละเอียดของ TINY RTC I^2C MODULES	22
2.15	คำอธิบาย PIN ของ XBEE PRO S1 ที่มีการใช้งาน	25
3.1	การเปรียบเทียบคุณสมบัติของเซ็นเซอร์วัดอุณหภูมิ	37
3.2	การเปรียบเทียบคุณสมบัติของเซ็นเซอร์วัดความดัน	38
3.3	การเปรียบเทียบคุณสมบัติของเซ็นเซอร์ GPS	39
3.4	การเปรียบเทียบคุณสมบัติของโมดูลวัดแรงดันของแหล่งจ่ายไฟ	40
3.5	การเปรียบเทียบคุณสมบัติของเซ็นเซอร์วัด ACCELEROMETER, GYROSCOPE และ MAGNETOMETER	40
3.6	การเปรียบเทียบคุณสมบัติของโมดูลกล้อง	41
3.7	คุณสมบัติจำเพาะและเหตุผลการเลือกใช้ของ ELECTRICAL POWER SUBSYSTEM	42
3.8	การเปรียบเทียบคุณสมบัติของหน่วยประมวลผล	43
3.9	การเปรียบเทียบคุณสมบัติของหน่วยความจำ	43
3.10	การเปรียบเทียบคุณสมบัติของโมดูลการนับเวลา	44
3.11	คุณสมบัติจาเพาะและเหตุผลการเลือกใช้ของANTENNA และ RADIO CONFIGURATION	44

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
4.1	เปรียบเทียบค่าที่วัดได้จากเซ็นเซอร์โมดูล GY-BMEP 280 กับค่าที่วัดจากเทอร์โมมิเตอร์ (อ้างอิง) ในสภาพแวดล้อม 3 สภาวะ	70
4.2	ค่าเปรียบเทียบระดับความสูงจากระดับน้ำทะเลกลางที่เปลี่ยนแปลงไปได้ค่าจาก GPSและที่ได้จากค่าความดัน	73
4.3	ค่าแรงดันจาก POWER VOLTAGE SENSOR และ แหล่งจ่ายไฟ	74
4.4	ค่าของมุมต่างๆที่วัดได้จากโมดูล GY-9250MPU9250 ในแนวแกน X	76
4.5	ค่าของมุมต่างๆที่วัดได้จากโมดูล GY-9250MPU9250 ในแนวแกน Y	77
4.6	ค่าของมุมต่างๆที่วัดได้จากโมดูล GY-9250MPU9250 ในแนวแกน Z	80
4.7	ทิศแสดงในหน่วยองศาของโมดูล GY-9250MPU9250 ทั้ง 4 ทิศ	83
4.8	หน่วยความจาจากการอัดวิดีโอของRASPBERRY PI CAMERA BOARD	87
4.9	รายละเอียดการใช้พลังงานของอุปกรณ์แต่ละชนิดจากการวัดด้วยมัลติมิเตอร์	88
4.10	ค่าเปรียบเทียบความเร็วที่ได้จากการคำนวณทางทฤษฎีกับความเร็วที่ได้จากการทดสอบที่น้ำหนักต่างๆ	93
4.11	ค่าเปรียบเทียบความเร็วที่ได้จากการคำนวณทางทฤษฎีกับความเร็วที่ได้จากการทดสอบที่น้ำหนักต่างๆ	95
4.12	การดำเนินงานตามเป้าหมายของการแข่งขัน	96
5.1	แผนงานโดยรวมในการสร้าง CANSAT เพื่อใช้ในการแข่งขัน	99

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

แบบจำลองดาวเทียมขนาดเล็ก (CanSat) คือ ดาวเทียมขนาดเล็กเท่าขนาดกระป๋องเครื่องดื่ม จึงมีชื่อเรียกคุ้นหูว่า “ดาวเทียมกระป๋อง” ซึ่งจะจำลองการทำงานเหมือนดาวเทียมจริง โดยติดตั้งอุปกรณ์อิเล็กทรอนิกส์ต่างๆไว้ภายใน โดยมีจุดประสงค์เพื่อใช้ในการเรียนรู้เกี่ยวกับเทคโนโลยีทางด้านอวกาศ อีกทั้งยังมีขนาดเล็กเหมาะแก่การประดิษฐ์ และไม่ได้ปล่อยเข้าสู่วงโคจรในอวกาศจริง

ในปัจจุบันเทคโนโลยีดาวเทียม เริ่มเข้ามามีบทบาทและความสำคัญต่อระบบการสื่อสารมากขึ้น เราจึงควรให้ความสำคัญนั้นๆ โดยเริ่มต้นจากการสร้างแบจำลองดาวเทียมขนาดเล็กนั่นเอง ซึ่งผู้คนเริ่มหันมาสนใจ ต้องการที่จะพัฒนาและส่งเสริมในด้านนี้มากขึ้น จึงมีการจัดการแข่งขันการสร้างดาวเทียมกระป๋องหลากหลายทั่วโลก มีตั้งแต่ระดับการแข่งขันแบบพื้นฐานไปจนถึงระดับที่ยากมากขึ้น ซึ่งในแต่ละปีจะมีกฎกติกาที่แตกต่างกันไป โดยเปิดโอกาสให้ผู้ที่สนใจได้พบกับความท้าทายในรูปแบบต่างๆในการแข่งขันนั้นๆ พื้นฐานของ CanSat จะมีระบบเซนเซอร์ต่างๆเพื่อเก็บรวบรวมข้อมูลซึ่งมีขนาดเล็ก ราคาถูก สามารถหาซื้อได้ตามท้องตลาด และสามารถหาข้อมูลการใช้งานได้ง่ายจากอินเทอร์เน็ตเพื่อนำมาพัฒนาตนเอง โดยจะปล่อยที่ความสูง 100 – 4000 เมตร จากจรวดหรือบอลลูนขึ้นอยู่กับภารกิจนั้นๆ ในระหว่างที่กางร่มชูชีพร่อนลงมา CanSat จะปฏิบัติภารกิจที่กำหนดไว้เพื่อส่งข้อมูลไปยังภาครับก่อนที่จะตกถึงพื้นดิน

CanSat เป็นโครงการอวกาศขนาดเล็กที่มุ่งเน้นการปฏิบัติจริงเพื่อการเรียนรู้และการแข่งขัน ซึ่งมีประโยชน์ที่หลากหลาย เช่น การเรียนรู้ที่จะออกแบบและสร้างแบบจำลองดาวเทียมขนาดเล็ก การสร้างตามแผนภารกิจที่มีความท้าทายในความสามารถของตนเอง การวิเคราะห์ปัญหาและการแก้ปัญหาที่เกิดขึ้นได้ นอกจากนี้จะได้เรียนรู้ทักษะในการทำงานร่วมกับผู้อื่นในกลุ่ม โดยบริหารและจัดการแบ่งหน้าที่การทำงาน เพื่อให้บรรลุเป้าหมายตามที่ต้องการ

1.2 วัตถุประสงค์

1) ออกแบบและสร้างแบบจำลองดาวเทียมขนาดเล็กตามภารกิจการแข่งขันประจำปี 2018 (ภาคส่ง)

2) ออกแบบและสร้างแบบจำลองดาวเทียมกระป๋องที่สามารถตรวจวัดสภาพอากาศบอกสถานะในการทำงาน และรายละเอียดต่างๆ ตามกติกาในการแข่งขัน

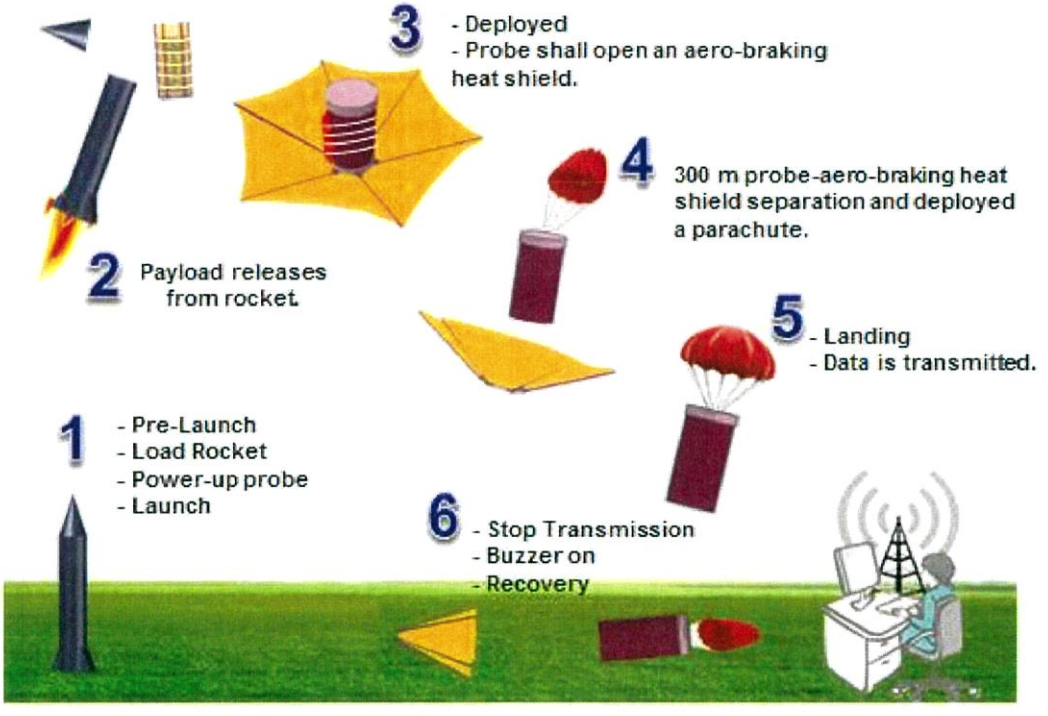
3) ออกแบบโครงสร้างและร่มชูชีพเพื่อให้สามารถเก็บกู้และนำกลับมาใช้ใหม่ได้

1.3 ขอบเขตของปริญญานิพนธ์

1) แบบจำลองดาวเทียมขนาดเล็กสามารถวัดสภาพอากาศโดยใช้เซนเซอร์ต่างๆ

2) แบบจำลองดาวเทียมขนาดเล็กสามารถส่งข้อมูลไปยังภาครับด้วยความถี่ 2.4 GHz

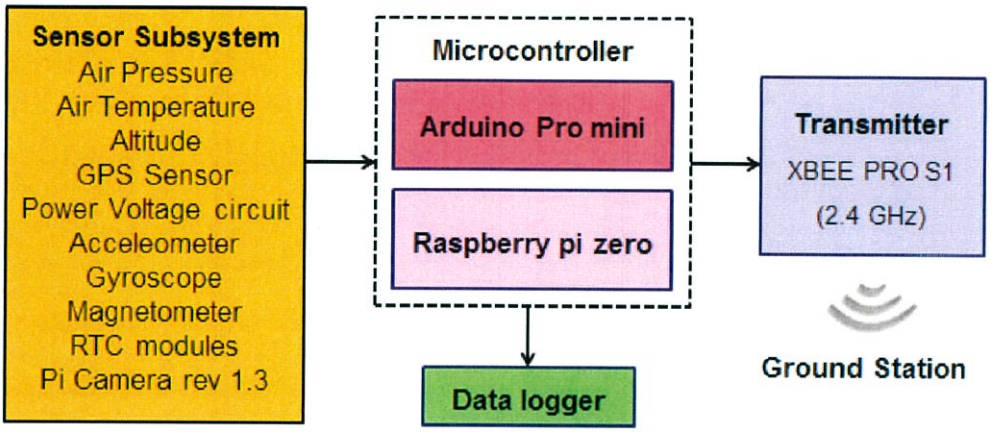
3) สามารถสำรองข้อมูลโดยเก็บไว้ในหน่วยความจำของแบบจำลองดาวเทียมขนาดเล็กเพื่อกู้คืนข้อมูลได้



รูปที่ 1.1 แผนภาพโดยรวมของแบบจำลองดาวเทียมขนาดเล็ก (CanSat)

จากรูปที่ 1.1 แสดงภาพรวมในรูปแบบของแผนภาพการทำงานแบบจำลองดาวเทียมขนาดเล็กซึ่งจะประกอบด้วย Rocket, โครงสร้างทั้งหมดของCanSat, Aero-braking heat shield และ Parachute โดยแบบจำลองดาวเทียมขนาดเล็กในส่วนของภาคส่ง (Science Vehicle) จะทำการออกแบบและสร้างแบบจำลองดาวเทียมขนาดเล็กตามภารกิจการแข่งขันประจำปี 2018 เพื่อตรวจวัดสภาพอากาศ บอกสถานะในการทำงาน รวมถึงรายละเอียดต่างๆ ตามกติกาการแข่งขัน เพื่อส่งข้อมูลไปยังภาคพื้นดิน (Ground Station)

บล็อกไดอะแกรมในส่วนของภาคส่ง โดยข้อมูลจาก Sensor Subsystem จะถูกรวบรวมโดยอุปกรณ์ประมวลผล 2 ตัว ได้แก่ Arduino Pro mini และ Raspberry pi zero จากนั้นส่งข้อมูลไปยังสถานีภาคพื้นดินผ่านโมดูลการสื่อสารไร้สายระยะไกล (XBEE PRO S1) ที่ความถี่ 2.4 GHz และเก็บข้อมูลสำรองในหน่วยความจำ ดังรูปที่ 1.2



รูปที่ 1.2 บล็อกไดอะแกรมในส่วนภาคส่ง (Science Vehicle)

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 แบบจำลองดาวเทียมขนาดเล็ก (CANSAT)

แคนแซท (CanSat) คือ ดาวเทียมขนาดเล็กเท่าขนาดกระป๋องเครื่องดื่ม ที่จำลองการทำงานเหมือนดาวเทียมจริง โดยติดตั้งอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ไว้ภายในกระป๋องเครื่องดื่มหรือวัสดุรูปทรงขนาดเท่ากระป๋องเครื่องดื่ม

การปล่อยแคนแซท จะใช้จรวดยิงขึ้นไปบนท้องฟ้า หรือใช้ Drone, บอลลูนปล่อยลงมาจากที่สูงในระดับหลายร้อยเมตร ตัวแคนแซทเมื่อถูกปล่อยออกมาแล้ว ก็จะกางร่มโดยอัตโนมัติ และระบบต่างๆ จะเริ่มทำงาน เช่น บันทึกภาพถ่ายหรือวิดีโอ ระบุตำแหน่งพิกัดของตัวแคนแซท วัดอุณหภูมิและความดันอากาศ สามารถรับส่งข้อมูลได้ทันทีผ่านอุปกรณ์ไร้สายส่งสู่อุปกรณ์รับสัญญาณภาคพื้นดิน [1]

การแข่งขัน CanSat [2] นั้นเกิดขึ้นในหลายประเทศทั่วโลกตั้งแต่ช่วงปลายทศวรรษที่ 1990 โดยปกติทีมนักเรียนจำนวนหนึ่งมารวมตัวกันเพื่อใช้ CanSats ของพวกเขาในระหว่างวันแข่งขันเพื่อชิงรางวัล กฎและวัตถุประสงค์อาจแตกต่างกันออกไป แต่ข้อกำหนดทั่วไปบางส่วนจะระบุดังนี้

1. ส่วนประกอบทั้งหมดของ CanSat ต้องอยู่ในขนาดกระป๋องเครื่องดื่มมาตรฐาน (ความสูง 115 มม. และเส้นผ่านศูนย์กลาง 66 มม.) ยกเว้นร่มชูชีพ เสาอากาศสื่อสารและเสาอากาศ GPS สามารถติดตั้งได้ภายนอก (ด้านบนหรือด้านล่างของกระป๋อง แต่ไม่ได้อยู่ที่ด้านข้าง) ขึ้นอยู่กับการออกแบบ

2. มวลของ CanSat ต้องอยู่ระหว่าง 300 ถึง 350 กรัม ซึ่ง CanSats ที่มีน้ำหนักเบาจะต้องเพิ่มความมั่นคงกับมันเพื่อให้ถึงขีดจำกัด 300 กรัมของมวลขั้นต่ำที่จำเป็น

3. ห้ามนำวัตถุระเบิด detonators, pyrotechnics และวัสดุที่เป็นอันตรายต่อการติดไฟหรือวัสดุอันตราย โดยวัสดุที่ใช้ต้องปลอดภัยสำหรับบุคลากร อุปกรณ์และสิ่งแวดล้อม

4. CanSat ต้องใช้พลังงานจากแบตเตอรี่หรือแผงเซลล์แสงอาทิตย์ และต้องสามารถเปิดเครื่องได้เป็นเวลา 4 ชั่วโมงอย่างต่อเนื่อง

5. CanSat ควรมีระบบการกู้คืน เช่นร่มชูชีพซึ่งสามารถนำกลับมาใช้ใหม่ได้หลังจากเปิดตัว ขอแนะนำให้ใช้ผ้าสีสดใสซึ่งจะช่วยในการเก็บกู้ของ CanSat หลังจากร่อนลง

2.2 รายละเอียดการแข่งขัน CANSAT ประจำปี 2018

ในแต่ละปี American Astronautical Society (AAS) และ American Institute of Aeronautics and Astronautics (AIAA) จะมีการจัดการแข่งขันการออกแบบ และสร้างอุปกรณ์ที่มีการทำงานเลียนแบบดาวเทียมที่ซึ่งต้องอาศัยหลักการและความรู้ทางด้านวิศวกรรมศาสตร์ในการออกแบบ สร้าง และพัฒนาปรับปรุง CanSat ให้เหมาะสมตามการแข่งขัน และในแต่ละปีทางการแข่งขันจะมีการเพิ่มภารกิจต่างๆ เพื่อให้กติกาการแข่งขันมีความท้าทายแก่ผู้แข่งขัน ซึ่งจะเป็นเปิดโอกาสให้กับทีมจากมหาวิทยาลัยและวิทยาลัยทั่วโลก [3]

ภารกิจประจำปี 2018 จำลอง Space probe (CanSat) เพื่อเข้าสู่ชั้นบรรยากาศ โดย Probe จะต้องบรรจุไข่ไก่ขนาดใหญ่ โดยจะต้องรักษาไม่ให้ไข่ไก่แตกระหว่างการร่อนลงสู่พื้นดินซึ่งลำดับการดำเนินงานต้องเป็นดังนี้ [4]

1) Probe จะถูกปล่อยที่ระดับความสูง 670 เมตร ถึง 725 เมตร และจากนั้นก็จะถูกปล่อยออกจากจรวด (Rocket) โดยจะต้องไม่ถูกการควบคุมทิศทางในการปล่อย

2) ในขณะที่ปล่อย Probe จะเปิด aero-braking heat shield ทันที และร่อนลงด้วยอัตราความเร็ว 50 เมตรต่อวินาที ซึ่งจะต้องรักษาความเสถียรภาพให้มั่นคง

3) ที่ระดับความสูง 300 เมตร Probe จะปล่อย aero-braking heat shield ทิ้งไปพร้อมกับปล่อยร่มชูชีพและลดความเร็วในการร่อนลงให้เหลือ 5 เมตรต่อวินาที

4) ขณะที่ร่อนลงสู่พื้นดิน Probe จะต้องปกป้องไข่ไก่เพื่อไม่ให้ไข่แตก

ซึ่งประกอบด้วยรายละเอียดการแข่งขันพื้นฐาน ดังแสดงในตารางที่ 2.1 [5]

ตารางที่ 2.1 ข้อกำหนดพื้นฐานของการแข่งขัน

ข้อที่	ข้อกำหนด
1	มวลโดยรวมทั้งหมดของ CanSat (probe) ต้องไม่เกิน 500 กรัม
2	Probe จะต้องเก็บรักษา heat shield ในการควบคุมอัตราเร็ว
3	Probe จะต้องไม่คะมำหรือคว่ำในส่วนใดส่วนหนึ่งของการร่อนลง
4	Probe และ aero-braking heat shield จะบรรจุลงกระป๋องทรงกระบอกที่มีเส้นผ่าศูนย์กลาง 125 มิลลิเมตร และสูงไม่เกิน 310 มิลลิเมตร ได้พอดี
5	Probe จะต้องบรรจุไข่ไก่ขนาดใหญ่และปกป้องมันจากการถูกทำลายตั้งแต่ปล่อยจนกระทั่งร่อนถึงพื้นดิน
6	Probeจะต้องเอื้อต่อไข่ไก่ขนาดใหญ่ที่มีมวล54ถึง68 กรัม และเส้นผ่าศูนย์กลางที่มากกว่า 50 มิลลิเมตร และความสูงที่มากกว่า 70 มิลลิเมตร
7	Aero-braking heat shield จะต้องไม่มีขอบแหลมที่จะติดค้างในส่วนของจรวด

ข้อที่	ข้อกำหนด
8	Aero-braking heat shield จะต้องเป็นสีสะท้อนแสง เช่น สีชมพูหรือสีส้ม
9	Aero-braking heat shield จะต้องถูกปล่อยออกจาก probe ที่ระดับความสูง 300 เมตร
10	Probe จะปล่อยร่มชูชีพที่ระดับความสูง 300 เมตร
11	องค์ประกอบทางอิเล็กทรอนิกส์ทั้งหมดจะถูกปิดล้อมและมีที่ปิดกั้นจากสิ่งแวดล้อมกับการยกเว้นของเซนเซอร์
12	อุปกรณ์อิเล็กทรอนิกส์ทั้งหมดจะต้องติดตั้งอย่างมั่นคง เช่น ยึดด้วยสกรูหรือกาวที่มีประสิทธิภาพสูง
13	กลไกทั้งหมดจะต้องสามารถทำงานได้ภายใต้แรงกระทำทั้งหมดได้
14	กลไกต้องไม่ใช้วัตถุที่ทำให้เกิดการระเบิด ติดไฟ หรือสารเคมี
15	กลไกที่ใช้ความร้อน (เช่นลวดนิโครม) ไม่ควรสัมผัสกับสภาพแวดล้อมภายนอกเพื่อลดความเสี่ยงที่อาจเกิดขึ้นจากไฟ
16	ระหว่างการร่อนลง probe จะเก็บความดันอากาศ ,อุณหภูมิจากภายนอก, ตำแหน่ง GPS และระดับแรงดันของแบตเตอรี่ต่อหนึ่งวินาที และแท็กข้อมูลเวลาด้วย mission time
17	ระหว่างการร่อนลง probe จะส่ง telemetry ทั้งหมดได้อย่างต่อเนื่อง
18	XBEE จะถูกใช้สำหรับการส่ง telemetry ที่ความถี่ 2.4 GHz Series 1 และ 2 หรือที่ความถี่ 900 MHz
19	ข้อมูลจะต้องส่งไปยังภาครับด้วยอัตรา 1 วินาทีต่อ 1 ชุด
20	ต้นทุนของ CanSat จะต้องไม่เกิน 1,000 เหรียญ โดยไม่รวมภาคพื้นดิน
21	Telemetry ทั้งหมดจะต้องแสดงแบบเรียลไทม์ขณะที่ร่อนลงสู่พื้นดิน
22	Telemetry ทั้งหมดจะต้องแสดงผลในหน่วยทางวิศวกรรม เช่น เมตร, เมตรต่อวินาที และ องศาเซลเซียส เป็นต้น
23	ต้องมีสัญญาณเสียงเมื่อร่อนลงสู่พื้นดินและหยุดการทำงาน
24	ไม่อนุญาตให้ใช้เลเซอร์
25	Probe จะต้องประกอบด้วยสวิตช์ไฟที่สามารถเข้าถึงได้ง่าย
26	Probe จะต้องประกอบด้วยไฟแสดงสถานะ เช่นไฟ LED หรือเครื่องสร้างเสียง
27	อัตราเร็วในการร่อนลงของ probe และ aero-braking heat shield 10-30 เมตรต่อวินาที
28	อัตราเร็วในการร่อนลงของ probe ที่ปล่อย aero-braking heat shield และร่มชูชีพ 5 เมตรต่อวินาที
29	ต้องใช้เซนเซอร์วัดความเอียงเพื่อยืนยันความเสถียรของ probe ระหว่างร่อนลง

2.3 ระบบย่อยเซนเซอร์ (SENSOR SUBSYSTEM)

Sensor Subsystem ซึ่งเป็นระบบย่อยๆของเซ็นเซอร์ประกอบรวมเป็นวงจรทางอิเล็กทรอนิกส์ทั้งหมด ซึ่งประกอบด้วย เซ็นเซอร์วัดสภาพอากาศ (ความดันอากาศ, อุณหภูมิอากาศ และความสูง), เซ็นเซอร์ GPS, Power voltage, เซ็นเซอร์ความเร่ง 9 แกน และ Pi camera rev1.3

2.3.1 เซ็นเซอร์วัดสภาพอากาศ GY-BMEP280

ใช้ Sensor Module ในการตรวจวัดสภาพอากาศ ซึ่งเป็นที่นิยมกันอย่างแพร่หลาย และสามารถหาใช้งานได้ง่ายและราคาไม่แพงมากจนเกินไปซึ่ง Sensor คืออุปกรณ์ตรวจจับสัญญาณหรือปริมาณทางฟิสิกส์ต่างๆ เช่น อุณหภูมิ ความดัน ความชื้น เป็นต้น

2.3.1.1 เซ็นเซอร์ตรวจวัดอุณหภูมิและความดัน

เซ็นเซอร์ตรวจวัดอุณหภูมิ (Temperature Sensor) ซึ่งเป็นอุปกรณ์ที่ทำหน้าที่เปลี่ยนระดับอุณหภูมิ เช่น ความร้อน-เย็น เป็นระดับแรงดันไฟฟ้า ซึ่งจะทำหน้าที่เป็นส่วนตรวจวัดค่าอุณหภูมิ

เซ็นเซอร์วัดความดัน (Pressure Sensor) ซึ่งเป็นอุปกรณ์ที่สามารถวัดที่ควบคุมความดันที่มีหน่วยในการแสดงผลได้หลากหลายในตัวเดียว เช่น bar, kPa, psi, mmHg เป็นต้น และมีสัญญาณ output แบบต่างๆให้เลือกใช้งานได้ เช่น NPN, PNP, 4-20 mA เป็นต้น โดยทำการศึกษาและเลือกใช้เซ็นเซอร์โมดูล ดังต่อไปนี้

BME/BMP280 [6]

โมดูล BME280 สำหรับวัดอุณหภูมิ (Temperature) ความชื้นสัมพัทธ์ (Humidity) และความกดอากาศ (Pressure) ใช้ชิพ BME280 สื่อสารผ่านบัส I^2C ใช้ไฟเลี้ยง 5 โวลต์ สามารถใช้ต่อกับบอร์ด arduino ได้ สามารถแสดงได้ดังรูปที่ 2.1

รูปที่ 2.1 โมดูล BME280 [6]

โมดูล BME280 ใช้ไฟเลี้ยง 5 โวลต์ เชื่อมต่อเข้ากับไมโครคอนโทรลเลอร์ผ่านทางอินเตอร์เฟส I^2C ที่ขาสัญญาณ SDL กับ SDA ซึ่งมีวงจร Voltage Divider ที่สามารถต่อกับระดับแรงดัน 5 โวลต์ได้

ข้อมูลเชิงเทคนิค (Technical details)

- Supply Voltage: 1.8 - 5V DC
- Interface: I^2C (up to 3.4 MHz)
- Temperature: -40 to +85°C
- Humidity: 0 -100 %
- Pressure: 300-1100 hPa
- Humidity: 0.008 %
- Pressure: 0.18 Pa
- Humidity: ± 3 %
- Pressure: ± 1 Pa
- Temperature: ± 1 °C
- โมดูลนี้มีขาเชื่อมต่อ 4 ขา ตามลำดับดังนี้ VCC (+3.3V), GND, SCL, SDA

2.3.1.2 การคำนวณความสูงจากค่าความดัน

ใช้ค่าความดันจากโมดูล BME280 เพื่อนำมาคำนวณหาระดับความสูงจากระดับน้ำทะเล ซึ่งสามารถคำนวณด้วยวิธีดังนี้

- ความกดอากาศที่ระดับน้ำทะเลปานกลาง(P_0) มีค่าเท่ากับ 1013.25 hPa
- จาก 760 mmHg เท่ากับ 101325 Pascal มีค่าเท่ากับ 1013.35 hPa

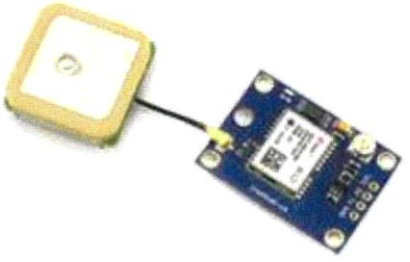
ดังนั้น ความสูงจากระดับน้ำทะเล สามารถคำนวณได้ดังสมการที่ 2.1 [6]

$$h = 44330 \times \left[1 - \left(\frac{P}{P_0} \right)^{\frac{1}{5.255}} \right] \qquad (2.1)$$

และ P_0 = 1013.35 hPa

2.3.2 เซ็นเซอร์ GPS GY-NEO6MV2

GPS GY-NEO6MV2 Module แสดงดังรูปที่ 2.2 [7] เป็นโมดูล GPS ที่มีความไวในการรับถึง - 162 dBm และสามารถเชื่อมต่อได้ถึง 50 ช่องสัญญาณ โดยใช้แรงดันไฟฟ้า 3.3 - 5V DC เชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่าน UART ความเร็ว 9600 ทำงานได้ในช่วงอุณหภูมิ -40 ถึง 85°C และมีทั้งหมด 4 pins แสดงคำอธิบายแต่ละ pin ดังตารางที่ 2.2



รูปที่ 2.2 GPS GY-NEO6MV2 Module [7]

ตารางที่ 2.2 คำอธิบาย PIN ของ GPS GY-NEO6MV2 Module

PIN	คำอธิบาย
Vcc	ไฟเลี้ยง 3.3-5.0 V DC
Rx	Receive
Tx	Transmit
GND	Ground

2.3.2.1 ความรู้ทั่วไปเกี่ยวกับ GPS

GPS ย่อมาจากคำว่า Global Positioning System คือ ระบบระบุตำแหน่งบนพื้นโลก โดยหลักการทำงานจะมี ดาวเทียม GPS (Navstar) ประกอบด้วย ดาวเทียม 24 ดวง โดยแบ่งเป็น 6 รอบวงโคจร การโคจรจะเอียงทำมุม 55 องศา กับเส้น ศูนย์สูตร (Equator) ในลักษณะสานกันคล้าย ลูกตะกร้อแต่ละวงโคจรมีดาวเทียม 4 ดวง รัศมีวงโคจรจากพื้นโลก 20,162.81 กิโลเมตร หรือ 12,600 ไมล์ ดาวเทียมแต่ละดวงใช้ เวลาในการโคจรรอบโลก 12 ชั่วโมง

GPS ทำงานโดยการรับสัญญาณจากดาวเทียมแต่ละดวง โดยสัญญาณดาวเทียมนี้ประกอบไปด้วยข้อมูลที่ระบุตำแหน่งและเวลาขณะส่งสัญญาณ ตัวเครื่องรับสัญญาณ GPS จะต้องประมวลผลความแตกต่างของเวลาในการรับสัญญาณเทียบกับเวลาจริง ณ ปัจจุบันเพื่อ

แปรเป็นระยะทางระหว่างเครื่องรับสัญญาณกับดาวเทียมแต่ละดวง ซึ่งได้ระบุมีตำแหน่งของมันมากับสัญญาณดังกล่าว

เพื่อให้เกิดความแม่นยำในการค้นหาตำแหน่งด้วยดาวเทียม ต้องมีดาวเทียมอย่างน้อย 4 ดวง เพื่อบอกตำแหน่งบนผิวโลก ซึ่งระยะห่างจากดาวเทียมทั้ง 3 กับเครื่อง GPS ซึ่งจะสามารถระบุตำแหน่งบนผิวโลกได้ หากพื้นโลกอยู่ในแนวระนาบแต่ในความเป็นจริงพื้นโลกมีความโค้งเนื่องจากสัณฐานของโลกมีลักษณะกลม ดังนั้นดาวเทียมดวงที่ 4 จะทำให้สามารถคำนวณเรื่องความสูงเพื่อทำให้ได้ตำแหน่งที่ถูกต้องมากขึ้น [8]

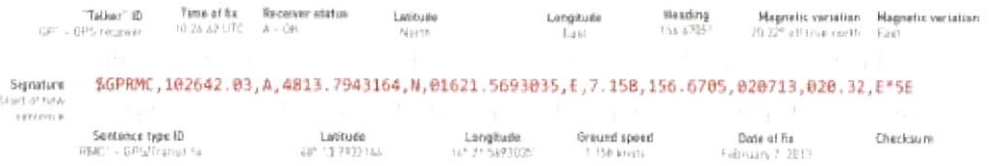
2.3.2.2 โปรโตคอล NMEA ของ GPS [9]

NMEA ย่อมาจาก Nation Maritime Electronics Association ซึ่งเป็นสมาคมที่มุ่งเน้นศึกษาและพัฒนาอุปกรณ์อิเล็กทรอนิกส์ เพื่อการเชื่อมต่อและทำงานร่วมกันของอุปกรณ์ โดยอุปกรณ์เหล่านี้ เมื่อเชื่อมต่อและทำงานร่วมกันต้องสามารถเข้าใจ กันได้หรือสื่อสารโดยใช้ภาษาเดียวกัน NMEA จึงพัฒนามาตรฐานในการสื่อสารข้อมูลระหว่างอุปกรณ์ดังกล่าว เรียกว่า NMEA Standard ซึ่งระบุข้อมูลการเชื่อมต่อทางไฟฟ้าและรูปแบบของข้อมูล

NMEA Standard เป็นโปรโตคอลภาษาที่ได้มีการพัฒนาเวอร์ชัน โดยเริ่มตั้งแต่ NMEA-0180, NMEA-0182 จนถึง NMEA-0183 โดยที่การใช้งานของ NMEA-0180 และ NMEA-0182 ค่อนข้างมีข้อจำกัดและจะเน้นทางการสื่อสารระหว่าง Loran C กับ Autopilot ดังนั้น จึงได้มีการพัฒนาให้สามารถที่ใช้งานได้อย่างกว้างขวางขึ้น โดยครอบคลุมถึงอุปกรณ์อิเล็กทรอนิกส์ในการเดินเรือจนกลายมาเป็น NMEA-0183 ใช้อักษร ASCII และการสื่อสารข้อมูลแบบอนุกรมในการส่งข้อมูล จากอุปกรณ์ตัวหนึ่งไปยังอุปกรณ์รับตัวหนึ่งหรือหลายๆตัว

มาตรฐาน NMEA-0183 ซึ่งในมาตรฐานนี้ ตัวอักขระที่ใช้คือ ASCII Text ซึ่งสามารถพิมพ์ได้ (รวมไปถึง Carriage Return and Line Feed) NMEA-0183 นั้นส่งข้อมูลด้วยอัตรา 4800 baud

รูปแบบการจัดเรียงข้อมูลของจีพีเอสเป็นไปตามมาตรฐานของ National Marine Electronics Association หรือ NMEA โดยจะขึ้นต้นด้วยอักขระ $GPxxx ซึ่งมีทั้งหมด 6 รูปแบบด้วยกัน คือ GGA, GLL, GSA, GSV, RMC, VTG รายละเอียดตามที่ได้ แสดงในตารางที่ 2.3 ทั้งนี้แต่ละรูปแบบต่างมีการจัดเรียงข้อมูลภายในที่แตกต่างกันออกไป เช่น



รูปที่ 2.3 ตัวอย่างประโยค NMEA และความหมายของข้อมูลแต่ละส่วนในประโยค [10]

รูปที่ 2.3 [10] แสดงเป็นรูปแบบข้อมูล RMC หรือ Recommended minimum specific GNSS data ตามด้วยข้อมูลเวลา UTC พิกัดตำแหน่งละติจูด, ตำแหน่งซีกโลก, พิกัดลองจิจูด, ตำแหน่งซีกโลก, และข้อมูลอื่น ๆ ซึ่งการที่จะเขียนโปรแกรมเพื่อดึงข้อมูล ณ ตำแหน่งที่ต้องการจากข้อมูลที่ส่งมาเป็นประโยค

ตารางที่ 2.3 รูปแบบการจัดเรียงข้อมูลตามมาตรฐานของ NMEA

NMEA Record	Description
GGA	Global positioning system fixed data
GLL	Geographic position - latitude/longitude
GSA	GNSS DOP and active satellites
GSV	GNSS satellites in view
RMC	Recommended minimum specific GNSS data
VTG	Course over ground and ground speed

<u>แบบที่ 1</u> GGA (Global positioning system fixed data) รูปแบบที่แสดงว่าข้อมูลของ GPS เพียงพอที่จะแสดงพิกัดได้สามมิติ (3D) ซึ่งดาวเทียมที่รับได้ต้องมากถึง 4 ดวงขึ้นไป ภาษาอังกฤษเรียกว่า Fix data ซึ่งมีรูปแบบการจัดเรียงและตัวอย่าง แสดงดังตารางที่ 2.4 [11]

ตารางที่ 2.4 รูปแบบการจัดเรียงข้อมูลประเภท GGA

Name	Example	Units	Description
Message ID	$GPGGA	-	GGA protocol header
UTC Position	161229.487	-	UTC Time : 161229.487
Latitude	3723.2475	-	37 องศา 23.2475 ลิปดา
N/S Indicator	N	-	N=north or S=south
Longitude	12158.3416	-	121 องศา 58.3416 ลิปดา
E/W Indicator	w	-	E=east or W=west
Position Fix Indicator	1	-	แสดงดังตารางที่ 2.7
Satellites Used	08	-	จำนวนดาวเทียมที่ GPS รับได้ (Range 0 to 12)
HDOP	1.1	-	Horizontal Dilution of Precision
MSL Altitude, Units	9,M	meters	ค่าความสูงเหนือระดับน้ำทะเลปานกลาง,หน่วย
Geoid Separation, Units	15.2,M	meters	ความสูงเหนือทรงรี WGS84,หน่วย
Age of Diff. Corr	-	second	เวลาเป็นวินาทีที่นับจากที่ได้รับค่า fix รูปแบบ DGPS
Diff. Ref. Station ID	0000	-	แสดงหมายเลขสถานีของ DGPS
Checksum	*18	-	Checksum Data

ตารางที่ 2.5 แสดงค่า Fix Indicator

Value	Description
0	Fix not available or invalid
1	GPS SPS Mode, fix valid
2	Differential GPS, SPS Mode, fix valid
3 to 5	Not supported
6	Dead Reckoning Mode, fix valid

แบบที่ 2 GLL (Geographic position - latitude/longitude) เป็นข้อมูลการครอบครองจากข้อมูล Loran และหน่วยเก่าบางส่วนอาจไม่สามารถส่งข้อมูลเวลาและข้อมูลที่ใช้งานได้ ซึ่งมีรูปแบบการจัดเรียงและตัวอย่าง แสดงดังตารางที่ 2.6 [11]

ตารางที่ 2.6 รูปแบบการจัดเรียงข้อมูลประเภท GLL

Name	Example	Units	Description
Message ID	$GPGLL	-	GLL protocol header
Latitude, N/S Indicator	4916.46,N	-	Latitude 49 deg. 16.45 min. North
Longitude, N/S Indicator	12311.12,W	-	Longitude 123 deg. 11.12 min. West
UTC Position	225444	-	Fix taken at 22:54:44 UTC
Status	A	-	Data Active or V (void)
Checksum	*ID	-	Checksum Data

แบบที่ 3 GSA (GNSS DOP and active satellites) รูปแบบที่แสดงรายละเอียดของข้อมูล Fix จำนวนดาวเทียมที่ใช้งานได้ รวมถึงค่าความคลาดเคลื่อน DOP (dilution of precision) ซึ่งตัวเลขน้อยๆจะเป็นค่าที่ดีมีความถูกต้องสูง ซึ่งมีรูปแบบการจัดเรียงและตัวอย่าง แสดงดังตารางที่ 2.7 [11]

ตารางที่ 2.7 รูปแบบการจัดเรียงข้อมูลประเภท GSA

Name	Example	Units	Description
Message ID	$GPGSA	-	GSA protocol header
Status	A	-	Auto selection of 2D or 3D fix
value include	3	-	3D fix – value include
PRNs of satellites	10,05,28,27,09,02	-	PRNs of satellites used for fix
PDOP	2.5	-	Dilution of precision
HDOP	1.8	-	Horizontal dilution of precision
VDOP	1.8	-	Vertial dilution of precision
Checksum	*35	-	checksum data

<u>แบบที่ 4</u> GSV (GNSS satellites in view) ซึ่งเป็นรูปแบบที่แสดงรายละเอียดของ GPS แต่ละดวงเช่นระดับความสูง (Elevation) อะซิมัทและ SNR (Signal to Noise Ratio) ซึ่ง เทียบได้กับความแรงของสัญญาณ SNR มีค่าตั้งแต่ 0 ถึง 99 ซึ่งค่ามากเป็นค่าที่ดี ในบางขณะเครื่อง GPS อาจจะรับสัญญาณได้เต็มที่ทั้งหมด 12 ดวง การส่งข้อความจะมากไปถ้าต้องแสดงในบรรทัดเดียว สมาคม NMEA จึงออกแบบให้รูปแบบ GSV สามารถที่จะแสดงข้อมูลดาวเทียมได้เต็มที่ประโยคหรือบรรทัดละ 4 ดวงเท่านั้น ดังนั้นถ้ารับสัญญาณดาวเทียมมาได้ทั้ง 12 ดวงจะได้รับประโยคทั้งหมด 3 บรรทัด ซึ่งมีรูปแบบการจัดเรียงและตัวอย่าง แสดงดังตารางที่ 2.8 [11]

ตารางที่ 2.8 รูปแบบการจัดเรียงข้อมูลประเภท GSV

Name	Example	Units	Description
Message ID	$GPGSV	-	GSV protocol header
Number of sentences	2	-	Number of sentences for full data
Sentence	1	-	Sentence 1 of 2
Satellites Used	08	-	จำนวนดาวเทียมที่ GPS รับได้
PRN number	01	-	Satellite PRN number
Elevation	40	degrees	Elevation
Azimuth	083	degrees	Azimuth
SNR	46	-	SNR - higher is better
Checksum	*75	-	Checksum data

<u>แบบที่ 5</u> RMC (Recommended minimum specific GNSS data) รูปแบบที่แสดงรายละเอียดของ GPS เรื่องความเร็ว (velocity) ค่าพิกัด เวลา ตลอดจน ทิศทางซึ่งมีรูปแบบการจัดเรียงและตัวอย่าง แสดงดังตารางที่ 2.9 [11]

ตารางที่ 2.9 รูปแบบการจัดเรียงข้อมูลประเภท RMC

Name	Example	Units	Description
Message ID	$GPRMC	-	RMC protocol header
UTC Time	103740	-	UTC Time : 103740
Status	A	-	A=data valid or V=data not valid
Latitude	2503.6319	-	25 องศา 03.6319 ลิปดา
N/S Indicator	N	-	N=north or S=south
Longitude	12136.0099	-	121 องศา 36.0099 ลิปดา
E/W Indicator	E	-	E=east or W=west
Speed over ground	2.69	knots	True
Course over ground	79.65	degrees	-
Date	100106	-	10/01/06
Magnetic variation	-	degrees	-
Variation sense	-	-	E=east or W=west
Mode	A	-	A=autonomous, D=DGPS, E=DR
Checksum	*53	-	Checksum Data

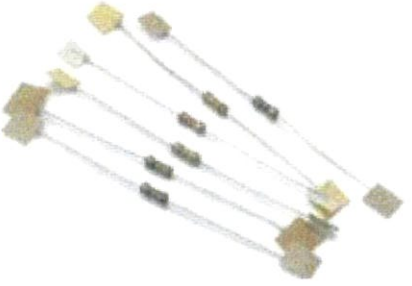
<u>แบบที่ 6</u> VTG (Course over ground and ground speed) ความเร็วดีขึ้น ตัวรับสัญญาณ GPS อาจใช้คำนำหน้า LC แทน GP ถ้ามีการจำลองเอาต์พุต Loran ซึ่งมีรูปแบบการจัดเรียงและตัวอย่าง แสดงดังตารางที่ 2.10 [11]

ตารางที่ 2.10 รูปแบบการจัดเรียงข้อมูลประเภท VTG

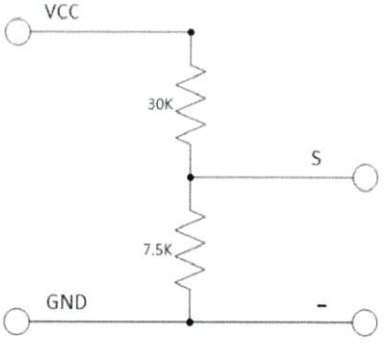
Name	Example	Units	Description
Message ID	$GPVTG	-	VTG protocol header
True track	054.7,T	degrees	True track made good
Magnetic track	034.4,M	-	Magnetic track made good
Ground speed	005.5,N	Knots	Ground speed

2.3.3 เซ็นเซอร์ POWER VOLTAGE

โดยการใช้ตัวแบ่งแรงดันไฟฟ้าที่เป็นพื้นฐานการแบ่งแรงดันไฟฟ้า 5 : 1 โดยใช้ตัวต้านทาน 30K และ 7.5K Ohm 1/4W แสดงดังรูปที่ 2.4 และมีข้อจำกัดด้านแรงดันไฟฟ้าที่น้อยกว่า 25 โวลต์ และมีวงจรการวัดแรงดันไฟฟ้า แสดงดังรูปที่ 2.5 [12]



รูปที่ 2.4 ตัวต้านทาน 1/4W [12]



รูปที่ 2.5 วงจรการวัดแรงดันไฟฟ้า [12]

โดยเลือกใช้หลักการ divider แรงดันไฟฟ้าแบบใช้ตัวต้านทาน 2 ตัว แทนการใช้ Voltage sensor ที่เป็นโมดูลสำเร็จ เพราะประหยัดมากกว่าและสามารถสร้างเองได้ง่าย

จุดประสงค์ของการสร้าง Power Voltage ลงในแบบจำลองดาวเทียมขนาดเล็ก เพราะว่าเราต้องการวัดแรงดันไฟฟ้าจากแบตเตอรี่ที่ใช้งาน 9 V ว่าเหลือเท่าไหร่ในขณะนั้นๆ

2.3.4 เซ็นเซอร์ความเร่ง 9 แกน GY-9250

เซ็นเซอร์โมดูล GY-9250 MPU 9250 แสดงดังรูปที่ 2.6 [13] เป็นเซ็นเซอร์ที่สามารถวัดค่าข้อมูลได้ครบตามที่เราต้องการ นั่นคือ สามารถวัดได้ทั้ง accelerometers, gyroscope และ magnetometer ในตัวเดียวกัน ซึ่งช่วยในการลดจำนวนโมดูลภายใน CanSat และลดมวลน้ำหนักของอุปกรณ์อิเล็กทรอนิกส์อีกด้วย โดยโมดูลนี้ใช้การเชื่อมต่อส่งข้อมูลผ่าน Bus I^2C โดยใช้ในการหาค่าของการเคลื่อนไหวทั้ง 9 แกน และทิศทางของการเคลื่อนที่ของโมด

รูปที่ 2.6 เซนเซอร์โมดูล GY-9250 [13]

ข้อมูลเชิงเทคนิค (Technical details) [13]

- ใช้ชิป: MPU-9250MPU9250MPU 9250
- แหล่งจ่ายไฟ: 3-5 โวลต์
- การสื่อสาร: มาตรฐานI^2C/SPIโปรโตคอลการสื่อสาร

(ชิป16 bit แปลงเอาท์พุทข้อมูล16)

- Gyro ช่วง: ± 250 500 1000 2000 °/s
- เร่งช่วง: ± 2 ± 4 ± 8 ± 16กรัม
- สนามแม่เหล็กช่วง: ± 4800uT
- โมดูลขนาด15มิลลิเมตร x 25มิลลิเมตร

2.3.5 PI CAMERA REV 1.3

รูปที่ 2.7 Raspberry Pi Camera Board v1.3 [14]

Raspberry Pi Camera Board [14] สามารถเสียบเข้ากับช่อง CSI บนRaspberry Pi โดยตรงได้เลย และสามารถถ่ายภาพความละเอียด 5MP ที่คมชัดหรือบันทึกวิดีโอ 1080p HD ที่ความเร็ว 30 เฟรมต่อวินาทีของเวอร์ชันล่าสุด 1.3 แสดงดังรูปที่ 2.7

Raspberry Pi มีเซ็นเซอร์ Omnivision 5647 ขนาด 5MP ในโมดูลโฟกัสคงที่ โมดูลนี้ยึดติดกับ Raspberry Pi โดยใช้สายริบบ้อน 15 pin เพื่อเชื่อมต่อกับกล้องถ่ายภาพ MIPI 15 pin ซึ่งออกแบบมาโดยเฉพาะสำหรับการเชื่อมต่อกับกล้อง บัส CSI มีอัตราข้อมูลที่สูงมากและมีเฉพาะข้อมูลพิกเซลที่ประมวลผลโดย BCM2835 เท่านั้น

บอร์ดมีขนาดเล็กทำให้เหมาะสำหรับมือถือหรือแอพพลิเคชั่นอื่น ๆ ที่มีขนาดและน้ำหนักที่สำคัญตัวเซนเซอร์มีความละเอียด 5 ล้านพิกเซลและมีเลนส์โฟกัสคงที่บนตัวกล้อง ในแง่ของภาพนิ่งกล้องจะสามารถถ่ายภาพนิ่งได้และยังสามารถรองรับการบันทึกวิดีโอได้อีกด้วย

ตารางที่ 2.11 คุณลักษณะของ Raspberry Pi Camera Board [14]

คุณลักษณะ	Raspberry Pi Camera Board
การใช้งานร่วม	ใช้ร่วมกับทั้งรุ่น A และรุ่น B Raspberry Pi
ระบบโมดูล	โมดูลกล้อง 530 Omnivision 5647
ความละเอียดภาพนิ่ง	2592 x 1944 พิกเซล
วิดีโอ	รองรับ 1080p ที่ 30fps, 720p ที่ 60fps และ 640x480p 60/90 การบันทึก
ขนาด	20 x 25 x 9 มิลลิเมตร

2.4 ระบบย่อยพลังงานไฟฟ้า (ELECTRICAL POWER SUBSYSTEM)

2.4.1 แบตเตอรี่ LI-ION RECHARGEABLE

แบตเตอรี่ใช้เป็นแหล่งจ่ายแรงดันไฟฟ้า 9 โวลต์ ซึ่งเป็นแบตเตอรี่ Li-ion มีความจุ 800 mAh แสดงดังรูปที่ 2.8

รูปที่ 2.8 แบตเตอรี่ Li-ion 9 V (800 mAh)

2.4.2 LM 7805 REGULATOR

LM7805 ทำหน้าที่แปลงไฟจาก 9 Volt เป็น 5 Volt สามารถจ่ายกระแสไฟให้โหลดได้สูงสุด 1 แอมป์ แสดงดังรูปที่ 2.9

รูปที่ 2.9 LM7805 Regulator

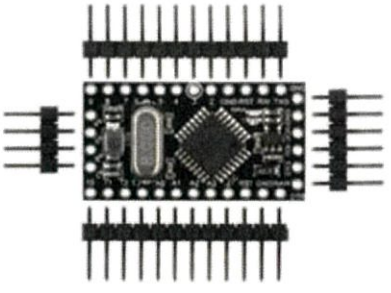
2.5 ระบบย่อยการสื่อสารและการจัดการข้อมูล (COMMUNICATION AND DATA HANDLING SUBSYSTEM)

2.5.1 หน่วยประมวลผล (PROCESSOR)

ระบบการประมวลผลมีหน้าที่ในการประมวลผลและจัดการข้อมูล แล้วส่งผ่านกระจายคำสั่งที่ได้รับมาจากสถานีภาคพื้นดินให้กับอุปกรณ์และเพย์โหลดที่เกี่ยวข้อง แล้วบันทึกข้อมูลที่ได้รับจากเพย์โหลด รวมทั้งควบคุมการทำงานของอุปกรณ์และเพย์โหลด โดยอุปกรณ์ที่ใช้ในระบบประมวลผลสัญญาณ ประกอบด้วย Arduino Pro mini 328 ใช้กับ Sensor Subsystem และ Raspberry Pi ใช้กับ Raspberry Pi Camera Board v1.3

2.5.1.1 Arduino Pro mini 328

Arduino Pro mini 328 เป็นบอร์ดไมโครโทรลเลอร์ (MCU) ที่ใช้ชิปประมวลผล คือ ATmega328 เป็นหน่วยประมวลผลหลักมี 14 ดิจิตอลอินพุท/เอาท์พุท เนื่องจากไม่มีฟังก์ชันการติดต่อ USB ทำให้มีขนาดเล็ก แสดงดังรูปที่ 2.10 [15] และมีคุณลักษณะเด่นของ Arduino Pro Mini 328 แสดงดังตารางที่ 2.12



รูปที่ 2.10 ไมโครคอนโทรลเลอร์ Arduino Pro mini 328 [15]

ตารางที่ 2.12 คุณลักษณะเด่นของ Arduino Pro Mini 328

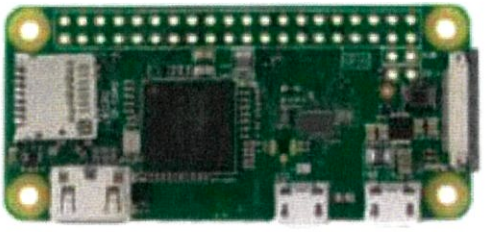
คุณสมบัติ	รายละเอียด
ไมโครคอนโทรลเลอร์	ATmega328
แรงดันไฟฟ้า	3.3 V
แรงดันไฟฟ้าอินพุต	3.3-9 V
ดิจิตอลอินพุท/เอาท์พุต	14 (สามารถใช้เป็น PWM เอาท์พุตได้ 6 pins)
อนาล็อกอินพุต	8 pins
Flash Memory	32 KB
SRAM	2 KB
E2PROM	1 KB
Clock speed	8 MHz

2.5.1.2 Raspberry Pi

Raspberry Pi [16] คือ บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานสามารถเล่นไฟล์วีดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย

บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มีทั้ง CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆได้ [17]

Raspberry Pi ได้ถูกสร้างขึ้นในปี 2549 ที่มหาวิทยาลัยเคมบริดจ์ ประเทศอังกฤษ โดยจะเลือกใช้ Raspberry Pi zero แสดงดังรูปที่ 2.11 ที่มีขนาดเล็กกว่า Raspberry Pi แบบทั่วไป เพราะว่าจะต้องจำกัดขนาดและน้ำหนักเพื่อให้สามารถบรรจุลงแบบจำลองดาวเทียมขนาดเล็กได้



รูปที่ 2.11 Raspberry pi zero [18]

บอร์ด Raspberry Pi zero เป็นบอร์ด Raspberry Pi ที่มีขนาดเล็กเป็นครึ่งหนึ่งของบอร์ด Raspberry Pi แบบทั่วไป มีช่องใส่ micro SD card พอร์ต mini HDMพอร์ต micro USB สำหรับจ่ายไฟและ micro USB OTG / Host สำหรับต่อ อุปกรณ์ ขา GPIO มี 40 ขา และพอร์ต CSI Camera Interface สำหรับต่อกล้อง Raspberry Pi Camera [18] และคุณสมบัติของ Raspberry pi zero แสดงดังตาราง ที่ 2.13

ตารางที่ 2.13 คุณสมบัติทางเทคนิคของ Raspberry Pi zero

คุณสมบัติ	รายละเอียด
ระบบชิพ	Broadcom BCM2835 ARM11
ความเร็ว	1 GHz (ปกติรันที่ 700 MHz)
หน่วยความจำ	RAM LPDDR2 ขนาด 512 MB
ขนาด	65x30x5 มิลลิเมตร
แรงดันเอาต์พุต	3.3 V-5.0 V

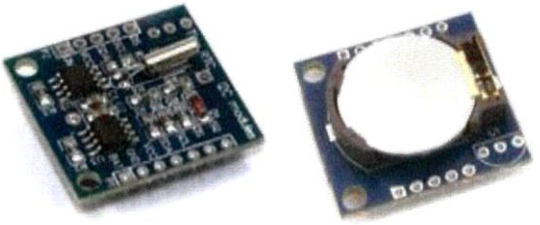
2.5.2 หน่วยความจำ (Memory)

หน่วยความจำ ใช้ Micro SD card ที่มีขนาดเล็กใช้สำหรับบันทึกข้อมูลขนาด 8 GB แสดงดังในรูปที่ 2.12

รูปที่ 2.12 Micro SD card SanDisk 8 GB

2.5.3 โมดูลนาฬิกาแบบเรียลไทม์ (REAL-TIME CLOCK)

Tiny RTC modules [19] แสดงดังรูปที่ 2.13 เป็นโมดูลนาฬิกาแบบเรียวไทม์ โดยใช้ IC เบอร์ DS 1307 เก็บข้อมูลวันเวลา แสดงผลเวลาได้ทั้งแบบ 12 และ แบบ 24 ชั่วโมง และหน่วยความจำ (memory) ใช้เบอร์ 24C32



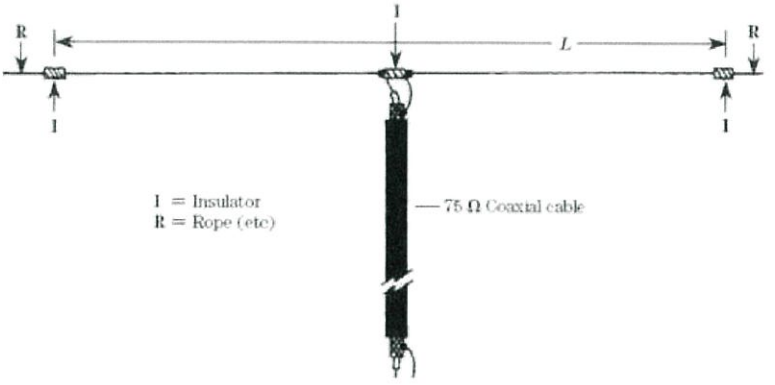
รูปที่ 2.13 Tiny Real Time Clock Modules (ด้านหน้าและด้านหลัง ตามลำดับ) [19]

ตารางที่ 2.14 รายละเอียดของ Tiny RTC I^2C modules [19]

คุณสมบัติ	Tiny RTC I^2C modules
ชิพ Real time clock	DS1307
หน่วยความจำ	24C32 32K EEPROM
แบตเตอรี่	ลิเธียม LIR2032 แบบชาร์จไฟและวงจรชาร์จ
ขนาด	27 × 28 × 8.4 มิลลิเมตร
การเชื่อมต่อ	I^2C

2.5.4 สายอากาศ (ANTENNA)

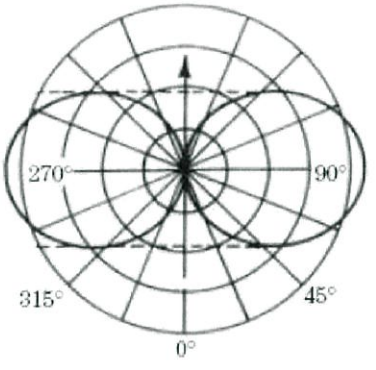
สายอากาศไดโพล [20] เป็นสายอากาศที่มีโครงสร้างง่ายที่สุด มีส่วนประกอบเป็นเส้นลวดสองเส้นที่มีความยาว L โดยสายอากาศไดโพล บางครั้งจะเรียกว่า สายอากาศแบบ Hertz เพราะว่าสายอากาศแบบนี้ถูก ค้นพบโดย Heinrich Rudolph Hertz เมื่อประมาณปี 1886 สายอากาศแบบ half-wavelength dipole เป็นสายอากาศแบบ สมดุล (balanced) ประกอบด้วยส่วนที่แพร่กระจายคลื่น 2 ส่วน แสดงดังรูปที่ 2.14 [20] ซึ่งสมดุล (balanced) หมายถึง ปลายสายทั้งสองของมีกระแสไหลเท่ากัน



รูปที่ 2.14 สายอากาศแบบ half-wavelength dipole [20]

แต่ละส่วนจะมีความยาว 1/4 ของความยาวคลื่น (quarter-wavelength) เมื่อรวมทั้งสองข้างก็จะเท่ากับ 1/2 ความยาวคลื่น (half-wavelength)

รูปแบบการแพร่กระจายคลื่นของสายอากาศไดโพล แบบ half-wavelength เมื่อทำการวางสายอากาศในแนวนอน



รูปที่ 2.15 รูปแบบการแพร่กระจายคลื่นของสายอากาศไดโพล [20]

จากรูปที่ 2.15 [20] ซึ่งก็จะเป็นว่าสายอากาศไดโพล แบบ half-wavelength จะมีการแพร่กระจายคลื่นออกเป็นสองส่วนหลัก หรือก็คือสายอากาศแบบ 2 ทิศทางนั่นเอง (bidirectional) คลื่นจะแพร่ออกทางด้านข้างของตัวนำ ส่วนหัวและท้าย จะไม่มีการแพร่ออกมาหรือออกมาน้อยมาก แต่ถ้าเราจับสายอากาศมาวางในแนวดิ่ง การแพร่กระจายคลื่นก็จะออกมาในรูป คล้าย ๆ ขนมโดนัท (doughnut) แสดงดังรูปที่ 2.16 [20] ซึ่งเป็นการแพร่กระจายคลื่นใน

สุญญากาศ แต่เมื่อใช้งานจริง การติดตั้งใกล้กับพื้นโลก (earth's surface) ซึ่งทำให้รูปแบบการแพร่กระจายคลื่นผิดเพี้ยนไป

รูปที่ 2.16 การแพร่กระจายคลื่นของสายอากาศไดโพล [20]

2.5.5 โมดูลการสื่อสาร XBEE PRO S1

ระบบสื่อสารเป็นส่วนประกอบหลักของดาวเทียม โดยทั่วไปแล้วส่วนประกอบทางด้านการสื่อสารเป็นเพียงชุดอุปกรณ์เล็กๆในตัวดาวเทียมเท่านั้น ซึ่งดาวเทียม CanSat นี้จะใช้โมดูลสื่อสาร XBEE PRO S1 ในการสื่อสารแบบไร้สาย

XBEE PRO S1 เป็นโมดูลสื่อสารไร้สายที่มีคุณลักษณะ ดังนี้ [21]

- ความถี่ในการทำงาน 2.4 GHz
- ระยะทำการในร่ม สูงสุดประมาณ 100 เมตร
- กำลังส่ง 60 mW (18 dBm)
- ความไวในการรับสัญญาณ -100 dBm (1% Packet error rate)
- ขนาด 2.438 x 3.294 cm
- ไฟเลี้ยง 2.8 ถึง 3.4 V
- กระแสไฟฟ้า เมื่อส่งข้อมูล 215 mA รับข้อมูล 55 mA
- อุณหภูมิ ใช้งาน -40 ถึง 85 °C

รูปที่ 2.17 โมดูลการสื่อสารไร้สาย XBEE PRO S1 [21]

โมดูลการสื่อสารไร้สาย XBEE PRO S1 แสดงดังรูปที่ 2.17 [22] และฟังก์ชันการทำงานของ PIN ในแต่ละขา ซึ่งจะแสดงคำอธิบาย PIN เฉพาะขาที่มีการใช้งานร่วมกับ arduino pro mini แสดงดังตารางที่ 2.15

ตารางที่ 2.15 คำอธิบาย PIN ของ XBEE PRO S1 ที่มีการใช้งาน

PIN	คำอธิบาย
RX	Receiver
TX	Transmit
GND	Ground
5V	ไฟเลี้ยง 5V

2.5.6 รูปแบบการสื่อสารของการรับ-ส่งข้อมูล

การสื่อสารของอุปกรณ์ในแต่ละโมดูลแต่ละตัวของเซนเซอร์ซึ่งก็แตกต่างกันไปขึ้นอยู่กับอุปกรณ์นั้นๆ ซึ่งการสื่อสารของเซนเซอร์แต่ละตัวเป็นดังนี้

การสื่อสารแบบ I^2C ได้แก่ โมดูล BME280 , โมดูล GY-9250 MPU 9250 และโมดูล Tiny Real Time Clock DS1307

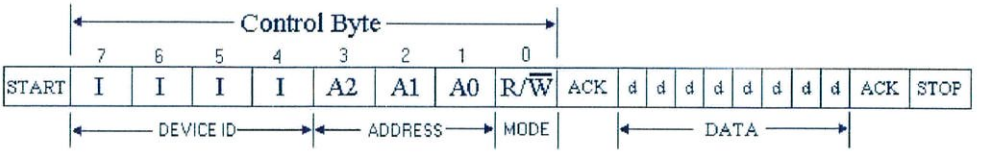
การสื่อสารแบบ UART ได้แก่ โมดูล GY-NEO6MV2 และ โมดูล XBEE PRO S1

การสื่อสารแบบ Analog ได้แก่ power voltage

2.5.6.1 การสื่อสารแบบ I^2C

I^2C ย่อมาจากคำว่า Inter Integrate Circuit Bus (IIC) นิยมเรียกสั้นๆ ว่า I^2C BUS เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous)เพื่อใช้ติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ serial data (SDA) และสาย serial clock (SCL) ซึ่งสามารถเชื่อมต่ออุปกรณ์จำนวนหลายๆ ตัวเข้าด้วยกันได้

โดยการรับ-ส่งข้อมูลแบบ I^2C BUS ซึ่งมีรูปแบบการเขียน/อ่านข้อมูลแสดงดังรูปที่ 2.18 [23]



รูปที่ 2.18 รูปแบบการเขียน/อ่านข้อมูลแบบ I^2C BUS

ไมโครคอนโทรลเลอร์จะทำการส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัส ตามด้วยรหัสควบคุม (Control Byte) ซึ่งประกอบ ด้วยรหัสประจำตัวอุปกรณ์ Device ID,Device Address และ Mode ในการเขียนหรืออ่านข้อมูล เมื่ออุปกรณ์รับทราบว่า MCU ต้องการจะติดต่อด้วยก็จะต้องส่งสถานะรับรู้ไป (Acknowledge) หรือแจ้งให้ MCU รับรู้ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง และเมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่ง สถานะสิ้นสุด (STOP Conditions) เพื่อบอกกับอุปกรณ์ว่า สิ้นสุดการใช้บัส

สภาวะที่เกิดขึ้นบนบัส I^2C [24]

มีทั้งหมด 5 สภาวะ ดังนี้

1) บัสว่าง (Bus not busy) สภาวะนี้เกิดขึ้นเมื่อ สถานะลอจิกบนสาย SDA และ SCL มีลอจิกสูงทั้งคู่ นั้นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้

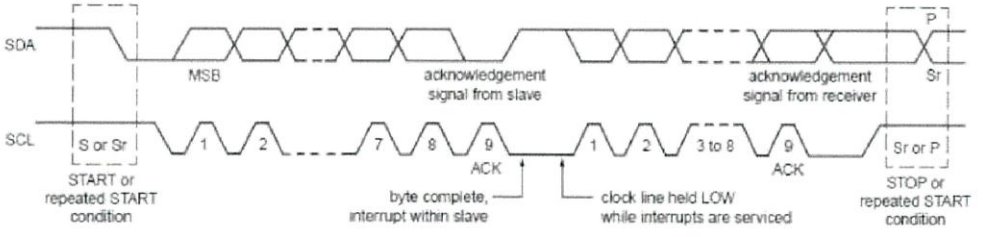
2) เริ่มต้นการถ่ายทอดข้อมูล (Start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะนี้ว่า สภาวะเริ่มต้น (START)

3) ข้อมูลดำรงอยู่บนบัส (Data valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL มีลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับข้อมูลใน จังหวะนั้นว่า เป็น"0" หรือ "1" ข้อมูลอาจเกิดความเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุด หรือ สภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดเกิดความผิดพลาดเกิดขึ้น

4) รับรู้ข้อมูล (Acknowledge) เกิดขึ้นหลังจากการถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิตเรียกว่า บิตรับรู้ (Acknowledge bit) มีสถานะเป็นลอจิกสูง หลังการส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์Master จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา อุปการณ์ Slave ที่ถูกอ้างอิงในการติดต่อหรือ กำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้ที่มีสถานะลอจิกต่ำ ได้รับข้อมูลเรียบร้อย

5) หยุดการถ่ายทอดข้อมูล (Stop dara transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูงเรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (STOP)

ลำดับการทำงานของการสื่อสารบนบัส I^2C แสดงดังรูปที่ 2.19 [24]



รูปที่ 2.19 Timing Diagram การทำงานของ I^2C BUS

1) ในสภาวะที่ยังไม่เริ่มทำงาน อุปกรณ์ทุกตัวจะมีลอจิกสูง

2) เมื่อเริ่มจะส่งข้อมูล SDA จะลดสถานะลอจิกจากสูงไปต่ำในขณะที่ SCL ยังอยู่ และสถานะลอจิกสูงโดย SCL จะค้างที่ลอจิกสูงนานถึง 0.6 ไมโครวินาที เมื่อผ่านช่วงเวลานี้ไป Master จะเริ่มสั่งการ

3) SCL จะมีสถานะลอจิกต่ำ จากนั้นระหว่างที่รอสัญญาณขาขึ้นของ Clock แรก Masterจะส่งค่าบิตแรกไปพร้อมกับสัญญาณ Clock และอุปกรณ์ที่เป็น Slave บนบัสจะเริ่มอ่านค่าในจังหวะที่ SCL เป็นลอจิกสูง

4) จากนั้นส่งค่า Address ของอุปกรณ์ที่ต้องการจะติดต่อด้วย ไปรวมทั้งสิ้น 7 บิตและตามด้วยบิตที่ 8 ซึ่งจะระบุว่าจะสั่งให้ Slave รอคำสั่ง (Write ระบุโดย SDA เป็นลอจิกสูง) หรือจะรออ่านค่าที่ส่งมาจาก Slave (Read ระบุค่าโดย SDA เป็นลอจิกต่ำ)

5) บิตที่ 9 จะเป็นการตอบรับจาก Slave ที่มี Address ตรงกับที่ Master ส่งไป ถ้า มี Slave ตอบรับ (Acknowledge) โดยการดึงสัญญาณ SDA ลงลอจิกต่ำ ได้ว่ามี Slave พร้อมจะสื่อสารด้วย แต่ถ้าไม่มี คือ SDA ค้างที่ลอจิกสูงได้ว่าไม่มี Slave พร้อมจะสื่อสารด้วย (Not Acknowledge)

6) หลังจากมีการ Acknowledge จะเป็นช่วงเวลาที่ SCL มีลอจิกต่ำเป็นเวลาสั้นๆ และ SDA จะถูกปล่อยวาง ก่อนที่ Slave ที่ติดต่อด้วยจะเริ่มส่ง ข้อมูลบิตแรกมาที่ SDA

7) จากนั้นสัญญาณ SCL จาก Master จะปล่อยออกมาเพื่ออ่านค่าสัญญาณบนสาย SDA ในจังหวะที่ SCL เป็น H และรับค่าเช่นนี้ไปเรื่อยๆ จน Data ครบ 8 บิตแรก ตามจังหวะสัญญาณ SCL

8) เมื่อข้อมูล (Data) ครบ 8 บิตแรกแล้ว Master จะให้สัญญาณ Acknowledge โดยดึงสัญญาณเป็น L เพื่อแสดงให้ Slave รู้ว่า Master จะรอ Data อีก 8 บิต ต่อไป (ในกรณีที่ข้อมูลที่ Slave ต้องการส่งเป็นแบบ 16 บิต)

9) จะกลับเข้าการทำงานในข้อ 6 อีกครั้งและทำการอ่านข้อมูลจนครบบิตที่ 9 ถึง 16

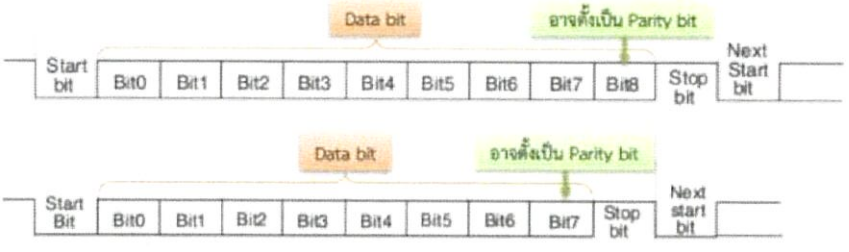
10) จากนั้นเมื่อถึงจังหวะการ Acknowledge ในบิตที่ 9 คราวนี้ Slave ไม่ต้องการส่งข้อมูลต่อแล้ว เพราะครบ 16 บิตแล้ว Slave จะ ไม่ตอบสนองแล้ว สาย SDA จะถูกปล่อยให้เป็น H

11) จากนั้น Master จะรู้แล้วว่าข้อมูลมาครบถ้วน 16 บิต Master ก็จะสั่งหยุดโดยการส่งสัญญาณ ขาขึ้นในสาย SDA พร้อมๆกับ SCL เป็น H ก็เป็นอันรู้กันว่าคือ Stop bit หมายถึงส่งข้อมูลครบ Byte แรก ซึ่งเป็น Address ของ Slaveตัวที่ Master ต้องการคุย

2.5.6.2 การสื่อสารแบบ UART

โดย UART ซึ่งย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter เป็นการเชื่อมต่อและสื่อสารข้อมูลอนุกรมกับอุปกรณ์ต่างๆ เช่น คอมพิวเตอร์, RFID, GPS, GSM Module, Wifi Module เป็นต้น [25]

ข้อดีของการใช้ Asynchronous คือสามารถสื่อสารแบบ Full duplex กล่าวคือสามารถรับและส่งข้อมูลระหว่าง Receiver และ Transmitter ได้ในเวลาเดียวกัน นอกจากนี้ ไม่ต้องใช้สายสัญญาณ Clock เพื่อกำหนดจังหวะการรับส่งข้อมูล แต่มีการกำหนดรูปแบบ Format หรือ Protocol การรับส่งข้อมูลขึ้นมาแทน และอาศัยการกำหนดความเร็วของการรับส่งข้อมูลให้เท่ากัน



รูปที่ 2.20 รูปแบบการสื่อสารแบบ Universal Asynchronous Receiver Transmitter [25]

จากรูป 2.20 แสดงรูปแบบการสื่อสารของ UART โดยจะต้องกำหนดคุณสมบัติเหล่านี้ให้เหมือนกันทั้งฝั่ง Receiver และ Transmitter ซึ่งประกอบด้วย

- Start Bit เป็นสถานะ Low
- ผู้ใช้งานสามารถกำหนดจำนวนข้อมูลของ Data Bit ให้เป็น 8 หรือ 9
- ผู้ใช้งานสามารถกำหนดชนิดของ Parity Bit แบบ Odd, Even,None
- Stop Bit มีจำนวน 0.5, 1, 1.5 หรือ 2 บิต

นอกจากนี้ผู้ใช้ควรศึกษาระดับแรงดันไฟฟ้าของสัญญาณการสื่อสารแบบ UART โดยจะแบ่งได้ดังนี้

1) TTL เป็นระดับสัญญาณแบบดิจิตอลทั่วไปที่ใช้กับ Microcontroller โดยสถานะโลจิก “0” จะมีแรงดันเท่ากับ 0 โวลต์ และสถานะโลจิก “1” จะมีแรงดันเท่ากับ 3.3 หรือ 5 โวลต์

2) RS232 เป็นระดับสัญญาณที่ใช้กับ Computer โดยสถานะโลจิก “0” จะมีแรงดันเท่ากับ -5 หรือ -13 V และสถานะโลจิก “1” จะมีแรงดันเท่ากับ +5

2.5.6.3 การสื่อสารแบบ Analog

การสื่อสารแอนะล็อกเป็นระบบที่ออกแบบให้ส่งข้อมูลสัญญาณแอนะล็อก เช่น สัญญาณเสียง แต่ได้มีการพัฒนาจนประยุกต์ให้สามารถส่งข่าวสารได้ด้วยในปัจจุบัน ปัญหาสำคัญสำหรับการสื่อสารแอนะล็อกก็คือเรื่องสัญญาณรบกวน แต่เนื่องจากสัญญาณในธรรมชาติทั้งหมดเป็นสัญญาณแอนะล็อก จึงยังคงเห็นการพัฒนาของการสื่อสารแบบแอนะล็อกในปัจจุบัน เช่น การมอดูเลตแอมพลิจูด (Amplitude Modulation หรือ AM) และการมอดูเลตความถี่ (Frequency Modulation หรือ FM) ซึ่งการมอดูเลต (Modulation) เป็นการผสมสัญญาณของข้อมูลเข้าไปกับสัญญาณอีกสัญญาณหนึ่งเรียกว่า คลื่นพาห์ ซึ่งสัญญาณนี้มีความถี่ที่เหมาะกับช่องสัญญาณนั้นๆ เพื่อให้ข้อมูลที่ส่งเข้าไปในช่องสัญญาณเดินทางได้ไกลมากขึ้น การเลือกวิธีมอดูเลตขึ้นอยู่กับปัจจัยหลายประการ เช่น ชนิดของสัญญาณแบนด์วิดท์ ประสิทธิภาพของระบบที่ต้องการ และความต้านทานต่อสัญญาณรบกวน เป็นต้น

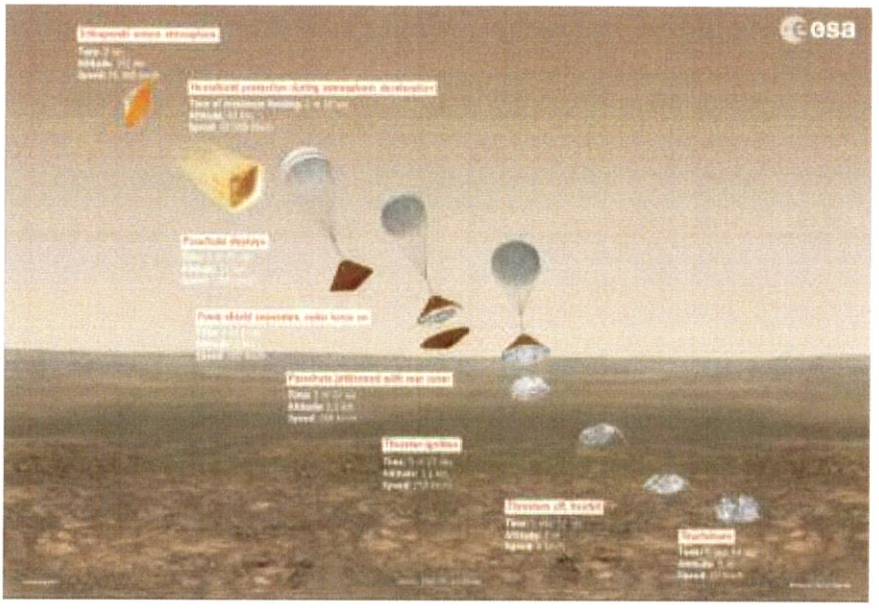
สัญญาณอนาล็อก (Analog Signal) เป็นสัญญาณแบบต่อเนื่อง มีลักษณะเป็นคลื่นไซน์ (sine wave) โดยที่แต่ละคลื่นจะมีความถี่และความเข้มของสัญญาณที่ต่างกัน เมื่อนำสัญญาณข้อมูลเหล่านี้ผ่านอุปกรณ์รับสัญญาณและแปลงสัญญาณก็จะ ได้ข้อมูลที่ต้องการ ตัวอย่างของการส่งข้อมูลที่มีสัญญาณแบบอนาล็อก คือ การส่งผ่านระบบโทรศัพท์ สัญญาณอนาล็อกเป็นสัญญาณที่มักเกิดขึ้นในธรรมชาติเป็นสัญญาณที่มีความต่อเนื่อง ไม่ได้มีการเปลี่ยนแปลงอย่างรวดเร็ว สัญญาณแบบนี้ เช่น เสียงพูด เสียงดนตรี เป็นต้น สัญญาณ อนาล็อกสามารถถูกรบกวนได้ง่ายจากสัญญาณรบกวน (Noise) หากมีสัญญาณรบกวนปะปนมากับ สัญญาณอนาล็อกแล้ว จะส่งผลให้การส่งข้อมูลช้าลง และทำให้การจำแนกหรือตัดสัญญาณรบกวน ออกจากข้อมูลต้นฉบับทำได้ยาก [26]

2.6 ระบบย่อยทางกลไก (MECHANICAL SUBSYSTEM)

2.6.1 AERO-BRAKING HEAT SHIELD

Heat shields ส่วนใหญ่นั้นทำมาจากเซรามิคและพลาสติกที่ติดตั้งอยู่บนเปลือกนอกของกระสวยอวกาศเพื่อไม่ให้เกิดความร้อนสูงขึ้นและถูกเผาหลังจากเข้าสู่ชั้นบรรยากาศ เมื่อภารกิจอวกาศมีแนวโน้มที่จะใหญ่ขึ้นและยาวนานขึ้นเรื่อย ๆ ความร้อนก็จะยิ่งมากขึ้นและ NASA คาดว่าเมื่อถึงเวลาเที่ยวบินที่ได้รับการจัดการอย่างถูกต้องก็จะเริ่มขึ้นใน Red Planet ดังนั้น Heat shields ก็จะขนาดเล็กเกินไปสำหรับการออกแบบที่มีขนาดใหญ่

ดังนั้น NASA จึงได้ทำการทดสอบ aero-braking heat shield ใหม่ชื่อ Adaptive Deployable Entry และ Placement Technology (ADEPT) โดยใช้เส้นใยคาร์บอนที่สามารถใช้งานได้และพับเก็บได้เหมือนร่ม [27]



รูปที่ 2.21 ExoMars 2016 Schiaparelli descent sequence [28]

2.6.2 ร่มชูชีพ (PARACHUTE)

2.6.2.1 ชนิดของร่มชูชีพ [29]

1) ร่มชูชีพแบบ Semi-Spherical

รูปที่ 2.22 ร่มชูชีพแบบ Semi-Spherical

ร่มชูชีพแบบ Simi-Spherical เป็นร่มชูชีพที่มีลักษณะครึ่งทรงกลม แสดงดังรูปที่ 2.22 ซึ่งเป็นร่มที่มีประสิทธิภาพดีเยี่ยม แต่ยากในสร้างให้ได้รูปทรงตรงตามต้องการ และมีค่าสัมประสิทธิ์ (Cd) เท่ากับ 1.5

2) ร่มชูชีพแบบ parapent

รูปที่ 2.23 ร่มชูชีพแบบ parapent

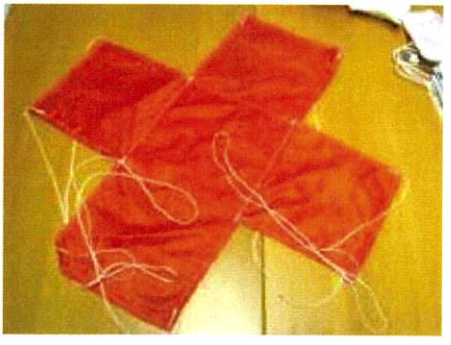
ร่มชูชีพแบบ parapent ร่มชีชีพที่มีลักษณะคล้ายสี่เหลี่ยมผืนผ้า มีจุดเด่นคือสามารถควบคุมทิศทางได้ แต่มีโครงสร้างในการสร้างที่ซับซ้อน นิยมนำมาใช้ในการโดดร่มโดนมีนักกระโดดร่มควบคุมทิศทาง แสดงดังรูปที่ 2.23 มีค่าสัมประสิทธิ์ (Cd) ขึ้นอยู่กับการออกแบบ

3) ร่มชูชีพแบบ Flat parachute

รูปที่ 2.24 ร่มชูชีพแบบ Flat parachute

ร่มชูชีพแบบ Flat parachute เป็นร่มชูชีพที่มีลักษณะคล้ายแบบ Semi-Spherical แต่เกิดจากการ ประกอบกันขึ้นมาจากผ้ารูปสามเหลี่ยมมาประกอบกันหลายรูปจนเป็นวงกลม แต่มีเสถียรภาพน้อยกว่าร่มชูชีพ แบบ Semi-Sphericalแสดงดังรูปที่ 2.24 ซึ่งมีค่าสัมประสิทธิ์ (Cd) เท่ากับ 0.75

4) ร่มชูชีพแบบ Cross



รูปที่ 2.25 ร่มชูชีพแบบ Cross

ร่มชูชีพแบบ Cross เป็นร่มชูชีพที่มีลักษณะคล้ายเครื่องหมายบวก มีข้อดีที่ออกแบบง่าย ไม่ซับซ้อน แต่ไม่เหมาะกับวัตถุขนาดเล็ก แสดงดังรูปที่ 2.25 ซึ่งมีค่าสัมประสิทธิ์ (Cd) เท่ากับ 0.8

สรุป เราจึงเลือกร่มชูชีพแบบ Semi-Spherical ที่มีลักษณะครึ่งทรงกลม แสดงดังรูปที่ 2.15 เนื่องจากมีประสิทธิภาพการใช้งานที่เหมาะสมกับ CanSat มากที่สุด

2.6.2.2 วัสดุที่เลือกใช้ในการทำร่มชูชีพ

โดยปกติแล้วความแข็งแรง และความคงทนของร่มชูชีพนั้นจะขึ้นอยู่กับวัสดุที่ใช้ในการทำตัวร่ม ซึ่งผ้าที่นำมาทำตัวร่มชูชีพนั้นควรมีน้ำหนักเบา แข็งแรง และกันน้ำได้ในระดับหนึ่ง ซึ่งโดยปกติจะใช้ผ้าร่มชนิด Ripstop ที่เป็นเนื้อผ้าชนิดเดียวกันที่เราใช้ทำร่มกันแดด ว่าว และเสื้อชูชีพ

2.6.2.3 ความเร็วที่ CanSat เคลื่อนที่

ในการสร้างร่มชูชีพนั้นมีจุดประสงค์เพื่อทำให้ CanSat ตกกระทบพื้นโลกโดยไม่เสียหายจากแรงดึงดูดของโลก (Gravity Force (F_G)) ดังนั้นความเร็วในการเคลื่อนที่ของ CanSat จึงเป็นปัจจัยหนึ่งในการออกแบบขนาดของร่มชูชีพ ซึ่งการที่จะทำให้ความเร็วของร่มชูชีพคงที่ได้นั้น จะต้องออกแบบขนาดของร่มให้เกิดแรงในทิศตรงข้ามกัน เรียกแรงนั้นว่า Drag Force (F_D) แสดงดังสมการที่ 2.2 [30]

$$F_D = \frac{1}{2}\rho C_d A v^2 \qquad (2.2)$$

เมื่อ F_D คือ แรงดึงขึ้นที่เกิดจากร่มทำให้ความเร่งเข้าใกล้ศูนย์

ρ คือ ความหนาแน่นของอากาศ = 1.225 kg /m^3

C_d คือ ค่าสัมประสิทธิ์แรงต้านของร่ม

A คือ ค่าพื้นที่หน้าตัดของตัวร่ม

v คือ ค่าความเร็วของร่มชูชีพ

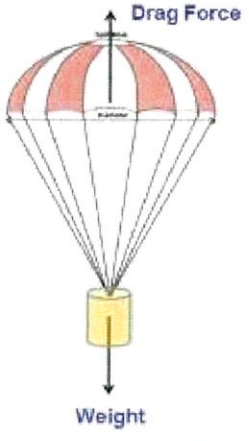
เนื่องจาก CanSat เคลื่อนที่ลงมาตามแรงดึงดูดของโลก (Gravity Force (F_G)) และสามารถหาค่าแรง F_G แสดงดังสมการที่ 2.3 [30]

$$F_G = mg \qquad (2.3)$$

เมื่อ m คือ มวลของวัตถุ

g คือ ค่าความเร่งเนื่องจากแรงโน้มถ่วง มีค่าเท่ากับ 9.81 m/s^2

เนื่องจากต้องการให้ความเร็วของ CanSat ขณะตกลงมานั้นคงที่และน้อยว่า 3 เมตรต่อวินาที ที่เป็นค่าความเร็วที่ทำให้ CanSat ได้รับความเสียหายน้อย ดังนั้นจึงต้องมีการออกแบบให้ความเร่งขณะที่เคลื่อนที่ลงนั้นมีค่าเท่ากับ 0 m/s^2 ซึ่งจะได้แรงที่เคลื่อนที่ลงเนื่องจากแรงดึงดูดของโลกนั้นจะมีค่าเท่ากับแรงดึงที่เคลื่อนที่ขึ้นเนื่องจากร่มชูชีพดังรูปที่ 2.26 และแสดงดังสมการที่ 2.4 และแสดงตัวแปรต่างๆ ดังสมการที่ 2.5 [30]



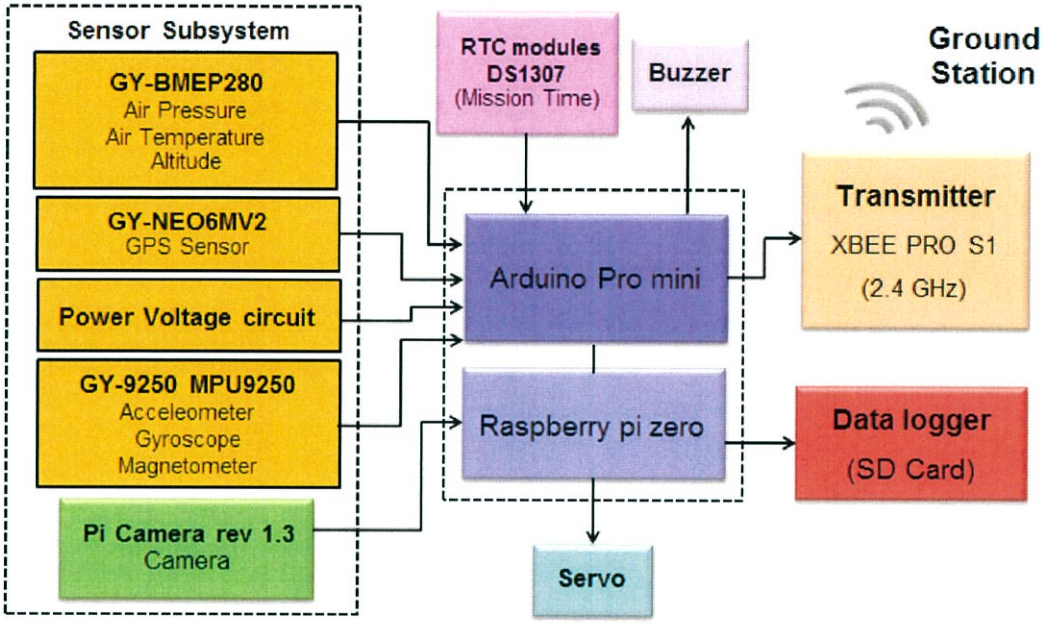
รูปที่ 2.26 การกระทำของแรงขณะที่วัตถุกำลังเคลื่อนที่ลง

$$F_G = F_D \tag{2.4}$$

$$mg = \frac{1}{2}\rho C_d A v^2 \tag{2.5}$$

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์



รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของอุปกรณ์ที่นำเสนอ

ปริญญานิพนธ์นี้นำเสนอการออกแบบและสร้างแบบจำลองดาวเทียมขนาดเล็ก (CanSat) เพื่อใช้ในการแข่งขัน CanSat Competition ประจำปี 2018 โดยหลักการทำงาน คือ จะวัดเก็บค่าสภาพอากาศ (ได้แก่ อุณหภูมิ ความดัน และความสูงจากระดับน้ำทะเลกลาง) เก็บค่าจาก GPS ที่ระบุถึงตำแหน่ง (ได้แก่ เวลา ละติจูด ลองจิจูด ความสูง เลขดาวเทียม) ค่าจากความเร่ง 9 แกน (ได้แก่ accelerometers, gyroscope และ magnetometer) ค่าระดับแรงดันจากแบตเตอรี่ และค่าการนับเวลาภารกิจจาก Real time clock ณ เวลาขณะนั้นๆของระบบอิเล็กทรอนิกส์ทั้งหมด และจะทำการประมวลผลด้วย Arduino pro mini อีกทั้งยังสามารถถ่ายวิดีโอบรรยากาศขณะร่อนลงด้วย pi camera rev 1.3 แล้วถูกประมวลผลด้วย Raspberry pi zero จากนั้นข้อมูลที่เก็บมานั้นจะทำการจัดเก็บลง SD Card on Board pi zero และส่งข้อมูลทั้งหมดไปแสดงผลยังภาครับ (Ground Station) ซึ่งปริญญานิพนธ์นี้จะกล่าวถึงเฉพาะในส่วนของภาคส่งเท่านั้น โดยบล็อกไดอะแกรมการทำงานของ CanSat แสดงดังรูปที่ 3.1

3.1 การออกแบบและสร้างแบบจำลองดาวเทียมขนาดเล็ก

การแข่งขัน CanSat Competition มีปัจจัยพื้นฐานที่มีผลต่อการสร้างคือ CanSat จะต้องมีน้ำหนักรวมไม่เกิน 500 กรัม และมีต้นทุนในการสร้างไม่เกิน 1000 เหรียญ ดั้งนั้น เราจึงต้องพิจารณาในการเลือกโมดูล เซ็นเซอร์ และอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ที่เหมาะสมมากที่สุด จึงได้นำโมดูลและเซนเซอร์ที่หาซื้อได้ง่ายมาเปรียบเทียบ รวมทั้งพิจารณา คุณสมบัติ ขนาด น้ำหนัก และราคาที่พอเหมาะ

3.1.1 การเลือกใช้ SENSOR SUBSYSTEM ที่เหมาะสม

โดยพิจารณาคุณสมบัติ ข้อมูลจำเพาะ ข้อดี และข้อเสียในแต่ละตัวของเซ็นเซอร์วัดอุณหภูมิ แสดงดังตารางที่ 3.1, เซนเซอร์วัดความดัน แสดงดังตารางที่ 3.2 เซ็นเซอร์ GPS แสดงดังตารางที่ 3.3, การวัดแรงดันของแหล่งจ่ายไฟ แสดงดังตารางที่ 3.4, เซ็นเซอร์วัด accelerometer, gyroscope และ magnetometer แสดงดังตารางที่ 3.5, และโมดูลกล้อง แสดงดังตารางที่ 3.6 ตามลำดับ

ตารางที่ 3.1 การเปรียบเทียบคุณสมบัติของเซ็นเซอร์วัดอุณหภูมิ

ชื่อ	ข้อมูลจำเพาะ	ข้อดี	ข้อเสีย
DHT22	ทำงานที่แรงดัน 3.3 - 6 V DC ทำงานที่กระแส 1-1.5 mA ทำงานได้ในช่วง : อุณหภูมิ -40 ถึง 80°C ±0.5 °C ความชื้น 0 - 100% ±2 %	-วัดได้ทั้งอุณหภูมิและความชื้น - ความแม่นยำสูง	- ขนาดใหญ่
DS18B20	ทำงานที่แรงดัน 3.0 - 5.5 VDC ทำงานได้ในช่วง : อุณหภูมิ -55 ถึง 125°C±0.5 °C	-ขนาดเล็ก -ราคาถูก	-ไม่รองรับการเชื่อมต่อแบบ I^2C -วัดได้เพียงอุณหภูมิ

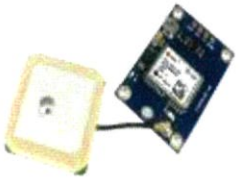
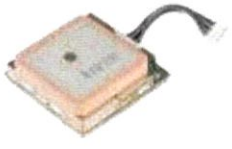
ชื่อ	ข้อมูลจำเพาะ	ข้อดี	ข้อเสีย
DS1621	ทำงานที่แรงดัน 2.7 - 5.5 VDC รองรับการเชื่อมต่อแบบ I^2C ทำงานได้ในช่วง : อุณหภูมิ -55 ถึง125°C ±0.5 °C	-ราคาถูก -รองรับการเชื่อมต่อแบบ I^2C	-ใช้สายเชื่อมเยอะเนื่องการขาของอุปกรณ์เยอะ
TMP36	ทำงานที่แรงดัน 2.7 - 5.5 VDC ทำงานได้ในช่วง : อุณหภูมิ -55 ถึง125°C ±0.5 °C	-ราคาถูก -ขนาดเล็ก	-ไม่รองรับการเชื่อมต่อ แบบ I^2C -วัดได้เพียงอุณหภูมิ
BME/BMP280	ทำงานที่แรงดัน 1.8 - 5 VDC รองรับการเชื่อมต่อ : I^2C (up to 3.4 MHz) ทำงานได้ในช่วง : อุณหภูมิ -40 ถึง 85°C ±1 °C ความชื้น 0 -100% ±3 % ความดัน 300 ถึง 1100 hPa±1%	-วัดได้ทั้งอุณหภูมิ ความดัน และความชื้น -ขนาดเล็ก -รองรับการเชื่อมต่อแบบ I^2C	-ราคาสูง

ตารางที่ 3.2 การเปรียบเทียบคุณสมบัติของเซ็นเซอร์วัดความดัน

ชื่อ	ข้อมูลจำเพาะ	ข้อดี	ข้อเสีย
BME085	ทำงานที่แรงดัน 1.8 - 3.6 VDC รองรับการเชื่อมต่อแบบ I^2C ทำงานได้ในช่วง : ความดัน 300 ถึง 1100 hPa	- ขนาดใหญ่	- วัดได้แค่ความดัน
BME180	ทำงานที่แรงดัน 3 - 5 VDC รองรับการเชื่อมต่อแบบ I^2C ทำงานได้ในช่วง : 300-1100 hPa อุณหภูมิ -40 ถึง +85°C	- วัดได้ทั้งค่าความดันและอุณหภูมิ	- ขนาดใหญ่

ชื่อ	ข้อมูลจำเพาะ	ข้อดี	ข้อเสีย
BME/BMP280	ทำงานที่แรงดัน 1.8 - 5 VDC รองรับการเชื่อมต่อ: I^2C (up to 3.4 MHz) ทำงานได้ในช่วง: อุณหภูมิ -40 ถึง 85°C ±1 °C ความชื้น 0 -100% ±3 % ความดัน 300 ถึง 1100 hPa±1%	-วัดได้ทั้งอุณหภูมิ ความดัน และ ความชื้น -ขนาดเล็ก -รองรับกาเชื่อมต่อแบบ I^2C และ SPI	- ราคาสูง

ตารางที่ 3.3 การเปรียบเทียบคุณสมบัติของเซ็นเซอร์ GPS

ชื่อ	ข้อมูลจำเพาะ	ข้อดี	ข้อเสีย
GY-NEO6MV2 	ความไวในการรับข้อมูล -162 dBm ช่องสัญญาณ 50 ช่อง ทำงานที่แรงดัน 3 ถึง 5 VDC ทำงานที่อุณหภูมิ -40 ถึง 85°C น้ำหนัก 15 กรัม	-มีขนาดเล็ก -ราคาถูกกว่า EM-506	-น้ำหนักมาก
EM-506 	ความไวในการรับข้อมูล-163 dBm ช่องสัญญาณ 48 ช่อง ทำงานที่แรงดัน 5 VDC ทำงานที่กระแส 55 mA น้ำหนัก 16 กรัม	-ขนาดเล็ก	-ราคาแพง -น้ำหนักมาก

ตารางที่ 3.4 การเปรียบเทียบคุณสมบัติของโมดูลวัดแรงดันของแหล่งจ่ายไฟ

ชื่อ	ข้อมูลจำเพาะ	ข้อดี	ข้อเสีย
Voltage Sensor Module	ทำงานที่แรงดัน < 25 VDC รองรับการเชื่อมต่อแบบ analog ความแม่นยำ ±1% Sensitivity: 100 mV/A	- ง่ายในการใช้งาน	- ขนาดใหญ่ - น้ำหนักมาก
Voltage Divider circuit	ทำงานที่แรงดัน < 25 VDC รองรับการเชื่อมต่อแบบ analog กำลังวัตต์ ¼ ความแม่นยำ ±1%	- น้ำหนักเบามาก - ราคาถูก - หาซื้อง่าย	- ทนความร้อนได้น้อยกว่า

ตารางที่ 3.5 การเปรียบเทียบคุณสมบัติของเซ็นเซอร์วัด accelerometer, gyroscope และ magnetometer

ชื่อ	ข้อมูลจำเพาะ	ข้อดี	ข้อเสีย
GY-87	ทำงานที่แรงดัน 3.3 - 5.0 VDC รองรับการเชื่อมต่อแบบ I^2C ทำงานได้ในช่วง : อุณหภูมิ -40 ถึง +85 °C	- ขนาดเล็กกว่า	- วัดได้แค่ความเร่ง 3 แกนและ gyroscope
GY-9250	ทำงานที่แรงดัน 3.3 - 5.0 VDC รองรับการเชื่อมต่อแบบ I^2C ทำงานได้ในช่วง : อุณหภูมิ -40 ถึง +105 °C	- วัดได้หลากหลายค่า (ความเร่ง 9 แกน)	- มีขนาดใหญ่กว่า GY-87

ตารางที่ 3.6 การเปรียบเทียบคุณสมบัติของโมดูลกล้อง

ชื่อ	ข้อมูลจำเพาะ	ข้อดี	ข้อเสีย
Serial TTL Camera VC0706	ทำงานที่แรงดัน +5 VDC ความละเอียด: 640x480 อัตราเฟรม: 30fps Dynamic Range: 60DB Max analog gain: 16DB	- มีความละเอียดที่ค่อนข้างสูง	- น้ำหนักมากกว่า - ขนาดใหญ่ - ราคาแพง
Pi Camera rev 1.3	ความละเอียดในการถ่ายภาพ: 5 megapixel ความละเอียดในการถ่ายวิดีโอ: 1080p HD อัตราเฟรม: 30,60,90 fps	- มีขนาดเล็ก - น้ำหนักเบากว่า - support ในการถ่ายวิดีโอ - ราคาถูกกว่า	- ใช้งานยากกว่า

จากตารางที่ 3.1 การเปรียบเทียบคุณสมบัติของเซ็นเซอร์วัดอุณหภูมิ และ ตารางที่ 3.2 การเปรียบเทียบคุณสมบัติของเซ็นเซอร์วัดความดัน พบว่า BME280 เป็นเซ็นเซอร์ที่สามารถวัดได้ทั้ง อุณหภูมิ, ความดัน และความชื้น ทำให้ประหยัดจำนวนของเซ็นเซอร์โดยรวมอยู่ในตัวเดียว อีกทั้งยังมีขนาดที่เล็ก น้ำหนักเบา รองรับการเชื่อมต่อแบบ I^2C และ SPI

จากตารางที่ 3.3 การเปรียบเทียบคุณสมบัติของโมดูล GPS พบว่า GY-NEO6MV2 และ EM-506 มีคุณสมบัติที่ใกล้เคียงกันแต่ GY-NEO6MV2 มีราคาที่ถูกกว่า

จากตารางที่ 3.4 การเปรียบเทียบคุณสมบัติของโมดูลวัดแรงดันของแหล่งจ่ายไฟ พบว่า Voltage Divider circuit เป็นสิ่งที่เราสามารถสร้างเองได้เพียงใช้ตัวต้านทาน 2 ตัว มีราคาถูก ขนาดเล็ก และหาซื้อง่ายกว่าด้วย

จากตารางที่ 3.5 การเปรียบเทียบคุณสมบัติของเซ็นเซอร์วัด accelerometer, gyroscope และ magnetometer พบว่า เลือกใช้ GY-9250 เพราะว่าสามารถวัดได้หลากหลายครบถ้วนตามที่ต้องการ แม้ GY-9250 จะมีขนาดที่ใหญ่กว่า แต่น้ำหนักไม่ต่างกันมากนัก

จากตารางที่ 3.6 การเปรียบเทียบคุณสมบัติของโมดูลกล้อง พบว่า Pi Camera rev 1.3 สามารถถ่ายวีดีโอได้ที่ความละเอียดที่ต้องการ มีราคาถูกกว่า น้ำหนักเบาและขนาดเล็กกว่า

3.1.2 การเลือกใช้ ELECTRICAL POWER SUBSYSTEM ที่เหมาะสม

พิจารณาคุณสมบัติ ข้อมูลจำเพาะของแหล่งพลังงานที่จะนำมาใช้งานเพื่อให้เพียงพอต่อการใช้ และ ตัวปรับแรงดันไฟ 5 V เพื่อหล่อเลี้ยงอุปกรณ์อิเล็กทรอนิกส์ให้พอเหมาะในแต่ละอุปกรณ์ แสดงดังตารางที่ 3.7

ตารางที่ 3.7 คุณสมบัติจำเพาะและเหตุผลการเลือกใช้ของ Electrical Power Subsystem

ชื่อ	ข้อมูลจำเพาะ	เหตุผลในการเลือกใช้
BESTON (NIMH) Rechargeable 9 V 	โวลต์: 9 VDC กระแส: 800 mAh ชนิด: Nickel-Metal Hydride ประเภท: Rechargeable	- มีกระแสที่สูงเพียงพอต่อการใช้งาน - ราคาถูก - สามารถใช้งานซ้ำได้อีกด้วย
LM7805 	- สามารถแปลงไฟจากโวลต์ที่สูงกว่าให้เหลือ 5 V - จ่ายกระแสไฟให้โหลด ได้สูงสุด 1 แอมป์	- สามารถทนต่อแรงดันและอุณหภูมิความร้อนได้มากกว่า - น้ำหนักเบา - ใช้งานได้ง่าย

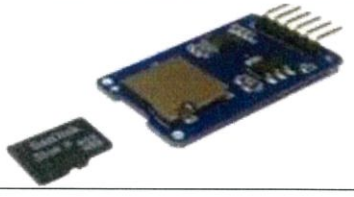
3.1.3 การเลือกใช้ COMMUNICATION AND DATA HANDLING SUBSYSTEM ที่เหมาะสม

โดยพิจารณาคุณสมบัติ ข้อมูลจำเพาะของหน่วยประมวลผล แสดงดังตารางที่ 3.8, หน่วยความจำ (SD Card) แสดงดังตารางที่ 3.9, โมดูลการนับเวลา (Real Time Clock) แสดงดังตารางที่ 3.10 ตามลำดับ

ตารางที่ 3.8 การเปรียบเทียบคุณสมบัติของหน่วยประมวลผล

ลักษณะจำเพาะ	ชื่อโมดูล		
	ATmega328 (Arduino pro mini)	ATmega32U4 (Leonardo Pro Micro)	Raspberry Pi Zero
Clock Speed	8 MHz	16 MHz	1 GHz (Run 700MHz)
Digital I/O Pins	14 pins	12 pins	17 GPIO
Analog Input Pins	6 pins	4 pins	✗
UART	✓	✓	✓
I2C	✓	✓	✓
SPI	✓	✓	✓
Flash Memory	32 KB	32 KB	512 MB
SRAM	2 KB	2.5 KB	512 MB

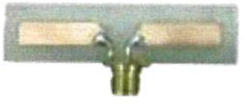
ตารางที่ 3.9 การเปรียบเทียบคุณสมบัติของหน่วยความจำ

ชื่อ	การเชื่อมต่อ (interface)	พื้นที่หน่วยความจำ
SD CARD on Board Pi ZERO	อยู่บนบอร์ด Raspberry pi zero	16GB และที่สามารถขยายมากกว่านี้ได้
Arduino SD card Module 	การเชื่อมต่อแบบ SPI	รองรับหน่วยความจำได้มากกว่า 16GB

ตารางที่ 3.10 การเปรียบเทียบคุณสมบัติของโมดูลการนับเวลา

ชื่อ	ข้อมูลจำเพาะ	ข้อดี	ข้อเสีย
DS3231	ทำงานที่แรงดัน 3.3 - 5.5 VDC รองรับการเชื่อมต่อแบบ I^2C ทำงานได้ในช่วง : อุณหภูมิ -40 ถึง +85°C	- ขนาดใหญ่	- น้ำหนักมากกว่า - ราคาสูงกว่า
Tiny RTC DS1307	ทำงานที่แรงดัน 3.3 – 5.0 VDC รองรับการเชื่อมต่อแบบ I^2C ทำงานได้ในช่วง : อุณหภูมิ -40 ถึง +85°C	- ราคาถูกกว่า - น้ำหนักเบากว่า	- ขนาดเล็ก

ตารางที่ 3.11คุณสมบัติจำเพาะและเหตุผลการเลือกใช้ของAntenna และ Radio Configuration

ชื่อ	ข้อมูลจำเพาะ	เหตุผลในการเลือกใช้
2.4 GHz Microstrip dipole Antenna 	Type: Omni-Directional ความถี่ต่ำสุด : 2106 MHz ความถี่สูงสุด: 2658 MHz Impedance: 50 Ohm Typical Gain: 2 dBi	- สามารถสร้างเองได้ตามรูปแบบที่ต้องการ - ราคาถูก - น้ำหนักเบา
XBEE PRO S1	โวลต์ : 2.8 -3.4 VDC RF Data rate : 250 kbps กำลังส่ง : 60 mW	- มีกำลังส่งสูง - น้ำหนักเบา ขนาดเล็ก - มีช่วงการส่งที่กว้าง

จากตารางที่ 3.8 การเปรียบเทียบคุณสมบัติของหน่วยประมวลผลได้ทำการเลือกใช้ Arduino pro mini ในการประมวลผลเซนเซอร์เป็นส่วนใหญ่ เพราะว่าบอร์ดจะใช้แรงดันไฟฟ้าต่ำที่ 3.3V ,อินพุต DC 3.3V ถึง 12V ราคาถูกและมีน้ำหนักที่เบา และเลือก Raspberry pi zero ใช้ประมวลผลกับโมดูลกล้องและการควบคุมเซอร์โว เพราะว่าง่ายต่อการเชื่อมต่อไปยังโมดูลกล้องและหน่วยความจำ

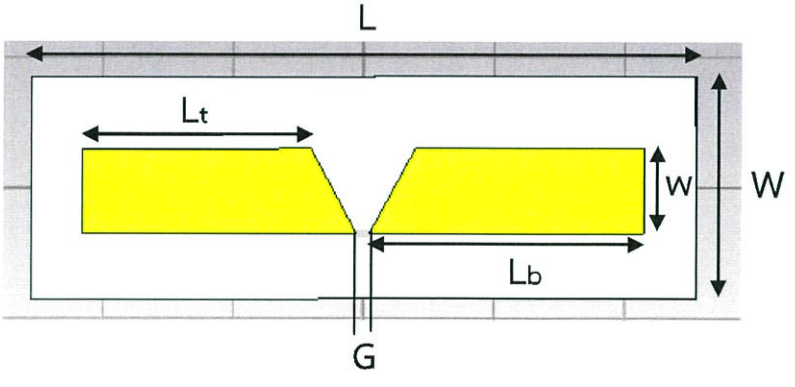
จากตารางที่ 3.9 การเปรียบเทียบคุณสมบัติของหน่วยความจำ เลือกใช้หน่วยความจำบนบอร์ด Raspberry pi zero ซึ่งมีความเร็วสูง มีหน่วยความจำที่มากกว่า และง่ายต่อการเชื่อมต่อไปยังหน่วยความจำ

จากตารางที่ 3.10 การเปรียบเทียบคุณสมบัติของโมดูลการนับเวลา พบว่า DS 3231 และ Tiny RTC DS1307 มีคุณลักษณะคล้ายคลึงกันมาก จึงเลือกใช้ Tiny RTC DS1307 เพราะว่าราคาถูกกว่า และน้ำหนักเบากว่า

3.1.4 การออกแบบ Microstrip dipole antenna 2.4 GHz

พารามิเตอร์ส่วนต่างๆของสายอากาศที่ต้องพิจารณา มีดังนี้

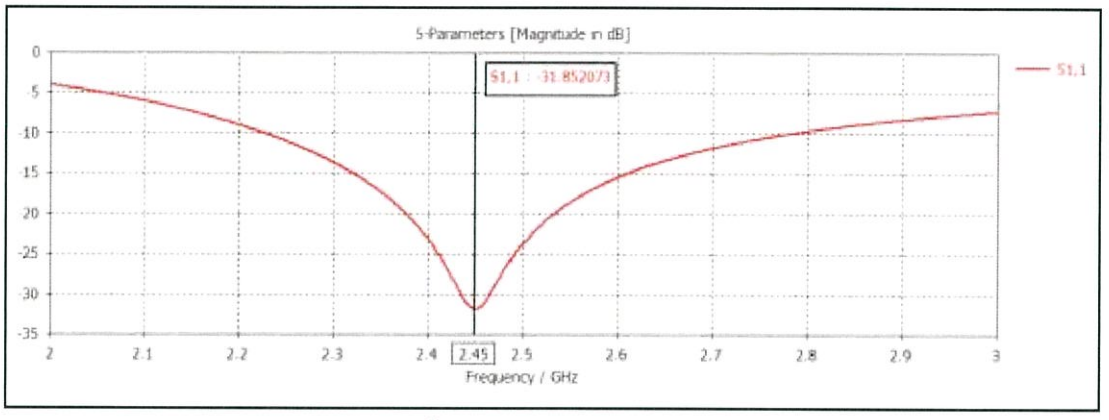
- ความกว้างของสายอากาศและ substrate
- ความยาวของสายอากาศและ substrate
- ความหนาของสายอากาศและ substrate
- Feeding gap ของสายอากาศ



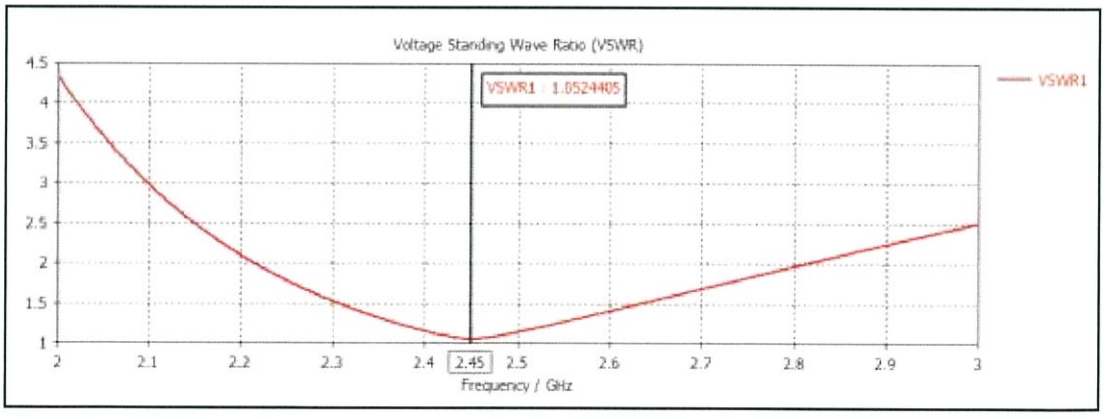
รูปที่ 3.2 Microstrip dipole antenna (ความถี่ 2.4 GHz)

จากรูปที่ 3.2 Microstrip dipole antenna 2.4 GHz จะประกอบด้วยตัวแปรต่างๆ ในการกำหนดขนาดของสายอากาศทั้ง Substate และ ขนาดของ Microstrip dipole ดังนี้ L = 5.09 cm , W = 1.7 cm, Lt = 1.75 cm, Lb = 2.1 cm, w = 0.65 cm และ G = 0.15 cm

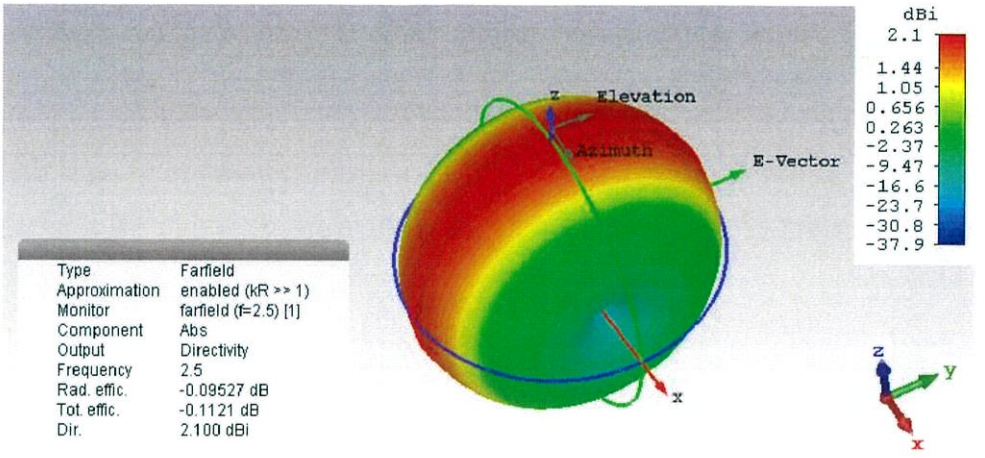
ในการทำการออกแบบ Microstrip dipole antenna 2.4 GHz โดยใช้โปรแกรม CST STUDIO SUITE และทำการ simulation แมตซ์ที่ 50 โอห์ม จะได้ กราฟ S-parameter ดังรูปที่ 3.3 กราฟ Voltage Standing Wave Ratio (VSWR) แสดงดังรูปที่ 3.4 และแบบรูปการแพร่กระจาย (farfield 3D) แสดงดังรูปที่ 3.5



รูปที่ 3.3 กราฟ S-parameter



รูปที่ 3.4 กราฟ Voltage Standing Wave Ratio (VSWR)



รูปที่ 3.5 แบบรูปการแพร่กระจาย (farfield 3D)

3.1.5 การออกแบบและสร้างโครงสร้างในแต่ละ LAYER

สำหรับปริญญานิพนธ์นี้ ได้ทำการออกแบบระบบย่อยต่างๆตามกติกาการแข่งขัน พื้นฐานของการแข่งขัน CanSat Competition 2018 โดยการออกแบบ CanSat แบ่งเป็น 3 ส่วนหลัก ดังนี้

1. **Sensor Subsystem** เป็นระบบย่อยของเซ็นเซอร์ทั้งหมด ประกอบด้วย
   - **เซ็นเซอร์ GY-BMEP280** ใช้เพื่อวัดความดันอากาศ อุณหภูมิ และใช้ค่าความดันเพื่อคำนวณความสูง
   - **เซ็นเซอร์ GY-NEO6MV2** เป็น GPS ใช้ระบุตำแหน่งและพิกัดของ CanSat
   - **เซ็นเซอร์ Power Voltage** ใช้วัดแรงดันที่เหลือของแบตเตอรี่ที่ใช้เป็นแหล่งจ่าย
   - **เซ็นเซอร์ GY-9250MPU9250** ใช้วัดค่าความเร่ง 9 แกน โดยจะใช้วัดค่าAceleometer, Gyroscope และ Magnetometer เพื่อดูการเอียง, การหมุน, การเคลื่อนไหวของ CanSat รวมทั้งค่าเข็มทิศแสดงทิศอีกด้วย
   - **เซ็นเซอร์ MPXV7002DP** เป็น Airspeed Sensor ใช้วัดความเร็วลม
   - **Pi Camera rev 1.3** เป็นโมดูลกล้องใช้ถ่ายวิดีโอบรรยากาศกลางอากาศขณะที่ CanSat กำลังร่อนลงสู่พื้นดิน

2. **Electrical Power Subsystem** ซึ่งประกอบด้วย

   - **BESTON Rechargeable Battery 9 V (800 mAh)** ใช้เป็นแหล่งจ่ายไฟเพื่อไปหล่อเลี้ยงส่วนประกอบทั้งหมดภายใน CanSat
   - **LM 7805** ใช้ในการลดแรงดันให้เหลือ 5 V
   - **Switch** โดยใช้เปิด-ปิดในการปล่อยหรือหยุดการไหลของกระแสจากแบตเตอรี่ไปยังส่วนประกอบต่างๆ

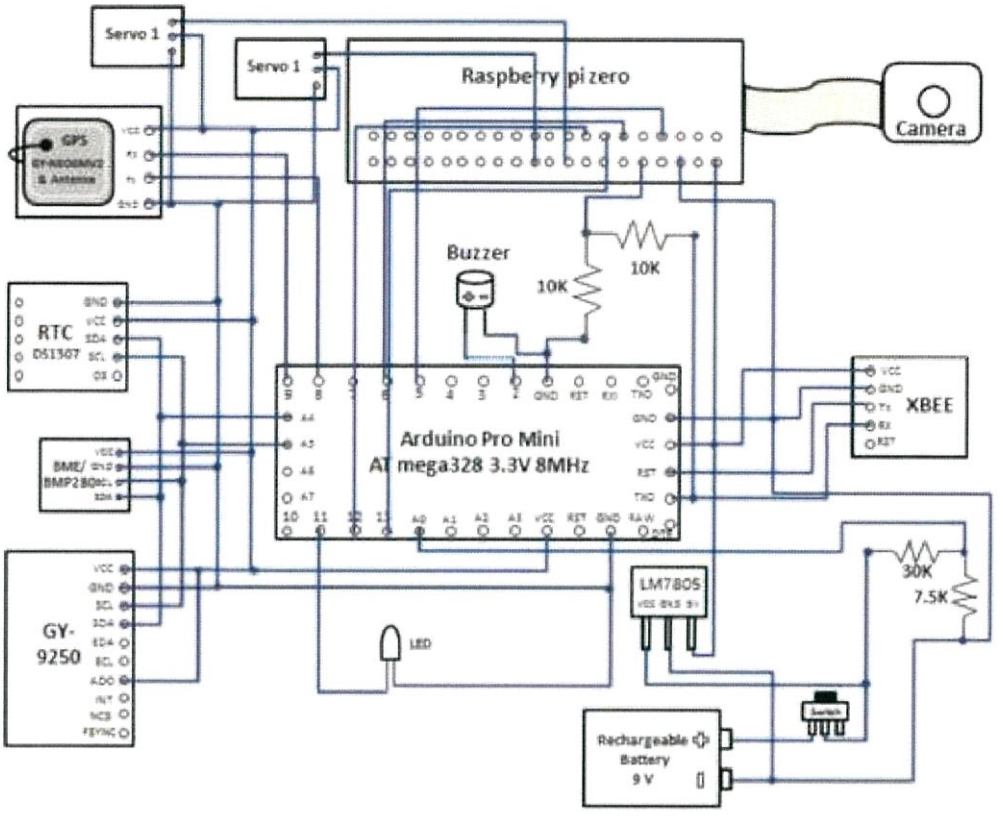
3. **Communication and Data Handling Subsystem** ซึ่งประกอบด้วย

   - **ตัวประมวลผล (Processor)** ประกอบด้วย 2 ตัว ดังนี้

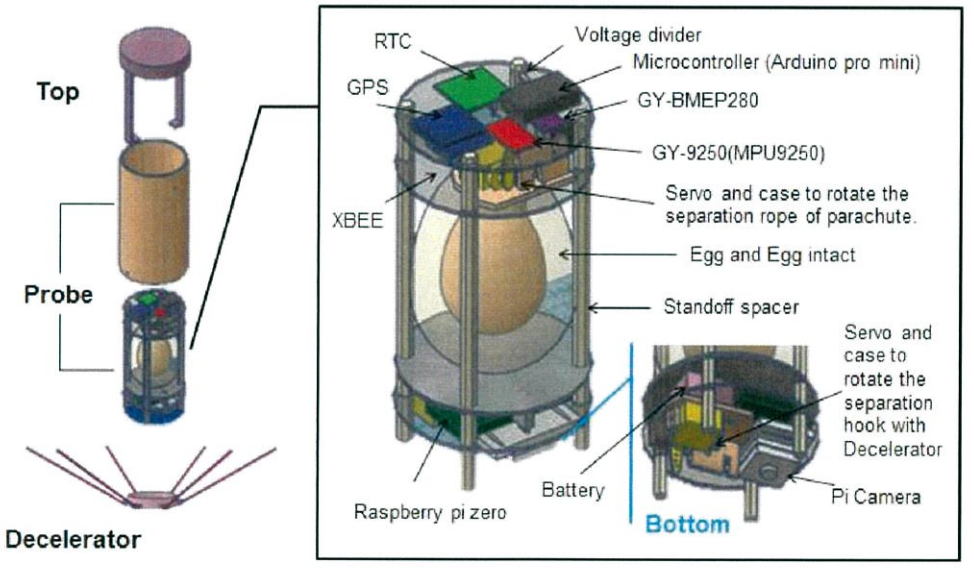
     1. โมดูล Arduino pro mini ใช้เป็นตัวประมวลของระบบย่อยของเซนเซอร์ทั้งหมด
     2. Raspberry pi zero ใช้เป็นตัวประมวลผลของโมดูลกล้อง และเก็บข้อมูลทั้งหมดลงหน่วยความจำบนบอร์ดของ Raspberry pi zero อีกด้วย

   - **Real Time Clock Modules DS1307** ใช้แสดงสถานะเวลาของภารกิจ (Mission Time)
   - **XBEE PRO S1 and Microstrip dipole antenna 2.4 GHz** เป็นตัวสื่อสารเพื่อส่งข้อมูลไปยังภาคพื้นดิน

โดยองค์ประกอบทั้ง 3 ส่วนนี้ซึ่งจะแบ่งออกเป็น 4 ชั้น โดยชั้นที่ 1 (ชั้นบนสุด) จะประกอบด้วย GY-BMEP280, GY-NEO6MV2, GY-9250MPU9250 และ Real Time Clock Modules DS1307 เพื่อรับสภาพอากาศ ตำแหน่งของแบบจำลองดาวเทียม และ หลีกเลี่ยงความร้อนจากถ่านที่มีผลต่ออุณหภูมิที่ต้องการวัด และ Arduino pro mini เพื่อง่ายต่อการเชื่อมต่อกับโมดูลในแต่ละตัว ชั้นที่ 2 จะเป็นชั้นของการปกป้องไข่ไก่ ชั้นที่ 3 ประกอบด้วย Servo, XBEE PRO S1, Buzzer, LM7805 และ Power Voltage Circuit ชั้นที่ 4 ประกอบด้วยสองด้าน ด้านบนประกอบด้วย Raspberry pi zero และแบตเตอรี่ ด้านใต้ ประกอบด้วย Servo และ Pi camera แสดงดังรูปที่ 3.7 ซึ่งวงจรการเชื่อมต่อรวมทั้ง 3 ชั้น แสดงดังรูปที่ 3.6



รูปที่ 3.6 วงจร Schematic การเชื่อมต่อโดยรวมโครงสร้างของ CanSat

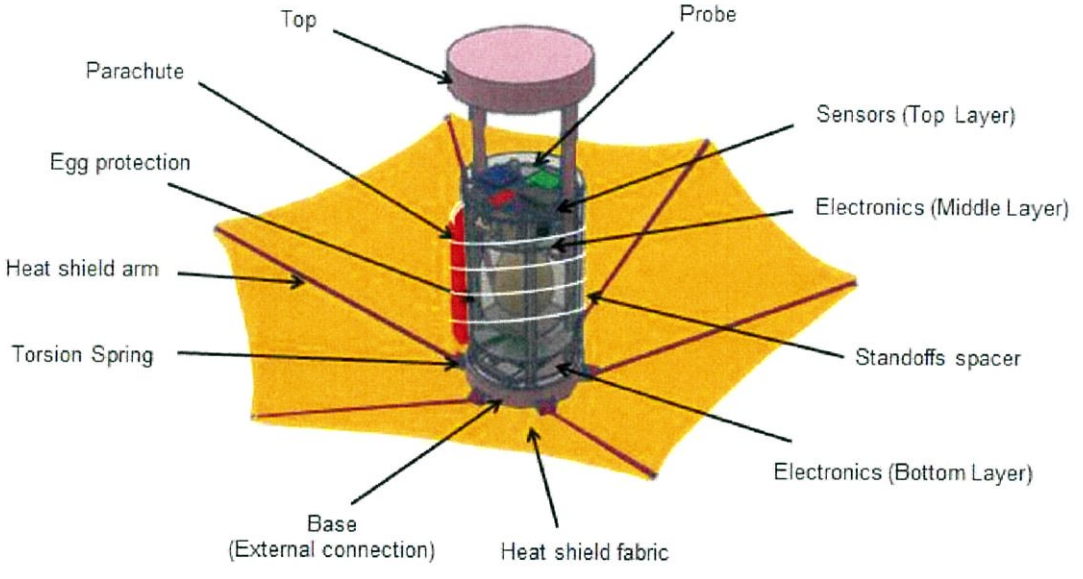


รูปที่ 3.7 โครงสร้างของ CanSat

3.1.6 การออกแบบและสร้าง MECHANICAL SUBSYSTEM

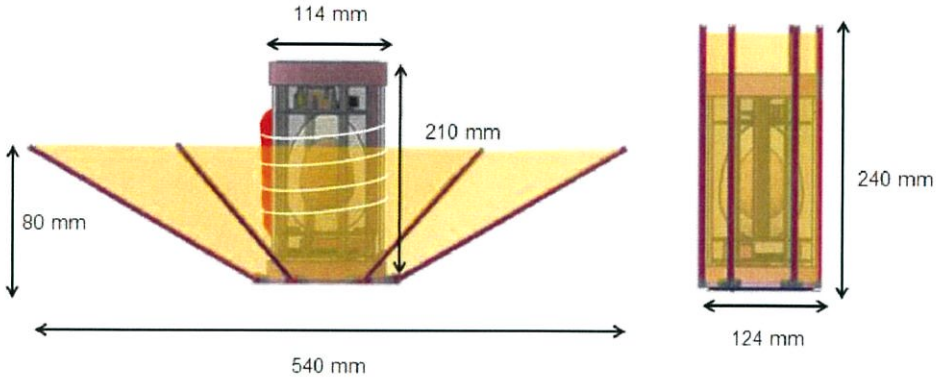
โครงสร้างและองค์ประกอบโดยรวมทั้งหมดจะแสดงดังรูปที่ 3.8 ซึ่งในส่วนของการออกแบบโครงสร้างจะทำการออกแบบด้วยโปรแกรม Tinkercad ซึ่งจากกติกาการแข่งขัน CANSAT จะต้องประกอบไปด้วย 2 ส่วนหลักๆ คือ ส่วนที่ 1 คือ Aero-braking Heat Shield เพื่อลดความเร็วเมื่อดีดตัวออกจากจรวด และส่วนที่ 2 คือ ส่วน Probe เพื่อเก็บอุปกรณ์อิเล็กทรอนิกส์ของระบบย่อยต่างๆและโครงสร้างการป้องกันไข่ไก่ โดยทั้ง 2 ส่วนนี้รวมกันจะต้องมีขนาดเส้นผ่านศูนย์กลางไม่เกิน 125 มิลลิเมตร สูงไม่เกิน 310 มิลลิเมตร และน้ำหนักไม่เกิน 500 กรัม เพื่อให้สามารถบรรจุลงจรวดได้อย่างเหมาะสมและพอดี

ซึ่งระบบย่อย Mechanical จะประกอบด้วย โครงสร้าง Aero-Braking Heat Shield และ ร่มชูชีพ (parachute) ในการช่วยลดความเร็วของการร่อนลงของ CanSat

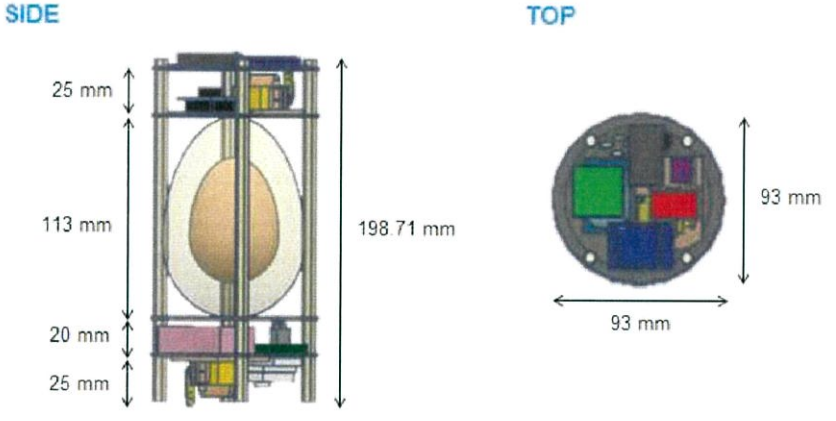


รูปที่ 3.8 การจัดองค์ประกอบภายในและภายนอกแบบจำลองดาวเทียมขนาดเล็ก

โครงสร้าง Probe ทั้งหมดรวมทั้ง Aero-braking Heat Shield โดยเราได้ทำการออกแบบและกำหนดขนาดเส้นผ่านศูนย์กลางและความสูงในแต่ละส่วน แสดงดังรูปที่ 3.9 และโครงสร้างระบบภายใน แสดงดังรูปที่ 3.10



รูปที่ 3.9 ขนาดเส้นผ่านศูนย์กลางและความสูงของ Heat Shield ขณะใช้งานและไม่ใช้งาน

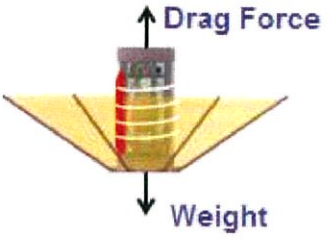


รูปที่ 3.10 ขนาดเส้นผ่านศูนย์กลางและความสูงแต่ละชั้นของแบบจำลองดาวเทียมขนาดเล็ก

การออกแบบและสร้าง Aero-braking Heat Shield

การเคลื่อนที่ของวัตถุภายใต้แรงโน้มถ่วงพร อมกับแรงต้านทานการเคลื่อนที่อัน เนื่องมาจากของเหลวที่มีความหนืด โดยวัตถุที่ใช้คือแบบจำลองดาวเทียมขนาดเล็ก เมื่อเราปล่อย แบบจำลองดาวเทียมขนาดเล็กลงไปนั้นมีแรงที่มากระทำ เนื่องจาก CanSat เคลื่อนที่ลงมาตามแรง ดึงดูดของโลก (Gravity Force (F_G)) และสามารถหาค่าแรง F_G แสดงดังสมการที่ 3.1

$$F_G = mg \tag{3.1}$$



รูปที่ 3.11 แรงที่กระทำต่อ CanSat ขณะเคลื่อนที่

การเคลื่อนที่การตกของวัตถุกับความต้านทานอากาศ ดังรูปที่ 3.11 แสดงดังสมการที่ 3.2 และ 3.3

$$F = ma \tag{3.2}$$

$$a = \frac{F}{m} = \frac{(D - W)}{m} \tag{3.3}$$

สูตรในการคำนวณความเร็วปลาย (Terminal velocity) แสดงดังสมการที่ 3.4

$$V_t = \sqrt{\frac{2mg}{AC_d\rho}} \tag{3.4}$$

เมื่อ V_t คือ ความเร็วปลาย (Terminal velocity)

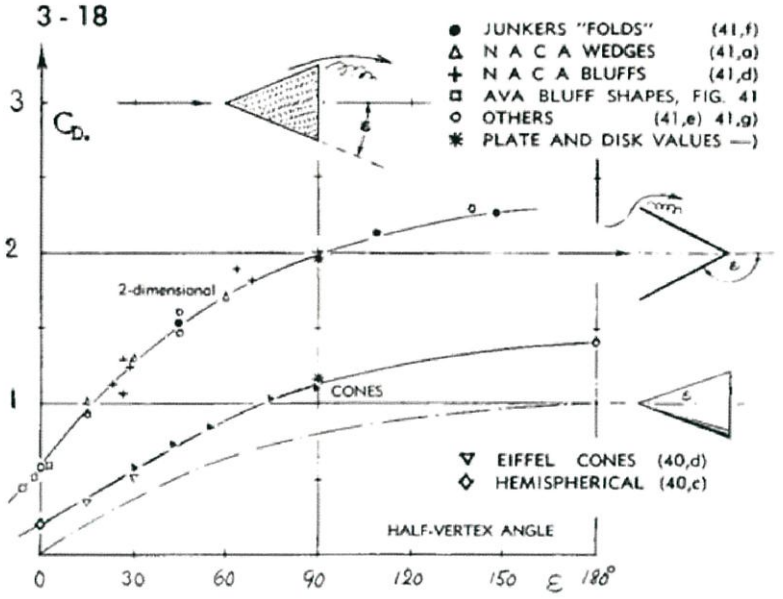
m คือ มวลของ CanSat ในหน่วยกิโลกรัม เท่ากับ 0.5 กิโลกรัม

g คือ ความเร่งเนื่องจากแรงโน้มถ่วงมีค่าเท่ากับ 9.81 m/s^2

ρ คือ ความหนาแน่นของอากาศเท่ากับ 1.12 kg/m^3

A คือ พื้นที่ผิวสัมผัสกับอากาศเท่ากับ 0.104 m^2
(ร่มชูชีพแบบ Semi-Spherical มีค่าเท่ากับ 1.5)

C_d คือ ค่าสัมประสิทธิ์แรงดึง (drag)



รูปที่ 3.12 ค่าสัมประสิทธิ์แรงดึง (drag) ของรูปร่าง wedges, cones และ similar [31]

จากรูปที่ 3.12 แสดงค่าสัมประสิทธิ์แรงดึง (Drag coefficients) ของทั้งรูปร่างทั้ง 3 รูปแบบ ซึ่งเป็นฟังก์ชันของ half-vertex angle (θ) โดยลักษณะของ heat shield เป็นลักษณะของ cone ที่มีค่ามุม half-vertex (θ) มีค่าเท่ากับ 60 องศา เมื่ออ่านค่าจากกราฟจะได้ประมาณ 0.8 หรือคำนวณจากสูตร

$$C_d = 0.0112(\theta)+0.162 \qquad (3.5)$$

$$C_d = 0.834$$

เมื่อแทนค่า C_d ในสมการความเร็วปลาย จะได้ความเร็วของแบบจำลองดาวเทียมขนาดเล็กเท่ากับ 10.02 m/s

การออกแบบร่มชูชีพ

จากสมการที่ 2.5 ทำให้สามารถหาค่าขนาดรัศมีของร่มชูชีพได้ โดยให้ความเร็ว มีค่าเท่ากับ 5 m/s มวลมีค่าเท่ากับ 500 กรัม หรือ 0.5 กิโลกรัม ซึ่งเป็นค่าที่มวลมากที่สุดที่กำหนดตามกติกา จะได้พื้นที่หน้าตัดของตัวร่ม (A) เท่ากับ

$$A = \frac{2mg}{\rho C_d v^2} \qquad (3.6)$$

เมื่อพื้นที่หน้าตัดของร่มเป็นวงกลมจะได้

$$A = \pi r^2 \tag{3.7}$$

$$\pi r^2 = \frac{2mg}{\rho C_d v^2} \tag{3.8}$$

จะได้ค่ารัศมีเท่ากับ

$$r = \sqrt{\frac{2mg}{\rho C_d v^2 \pi}} \tag{3.9}$$

เมื่อ r คือ รัศมีของร่มชูชีพในหน่วยเมตร

m คือ มวลของ CanSat ในหน่วยกิโลกรัม เท่ากับ 0.5 กิโลกรัม

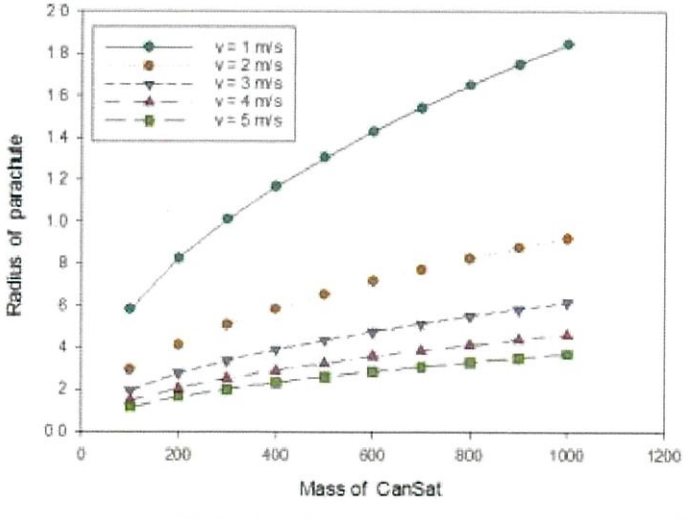
g คือ ความเร่งเนื่องจากแรงโน้มถ่วงมีค่าเท่ากับ 9.81 m/s^2

ρ คือ ความหนาแน่นของอากาศ = 1.225 kg/m^3

C_d คือ ค่าสัมประสิทธิ์แรงต้านของร่ม ขึ้นอยู่กับชนิดของร่มชูชีพ (ร่มชูชีพแบบ Semi-Spherical มีค่าเท่ากับ 1.5)

v คือ ค่าความเร็วของร่มชูชีพ

จากข้อกำหนดในการแข่งขันน้ำหนักของ CanSat ต้องมีค่าไม่เกิน 500 กรัม เมื่อทำการแทนค่าที่น้ำหนักต่างๆ จะทำให้หาค่ารัศมีของร่มชูชีพได้ดังรูปที่ 3.13



รูปที่ 3.13 กราฟแสดงความสัมพันธ์ระหว่างมวลของ CanSat ต่อขนาดรัศมีของร่มชูชีพที่ระดับความเร็วต่างๆ

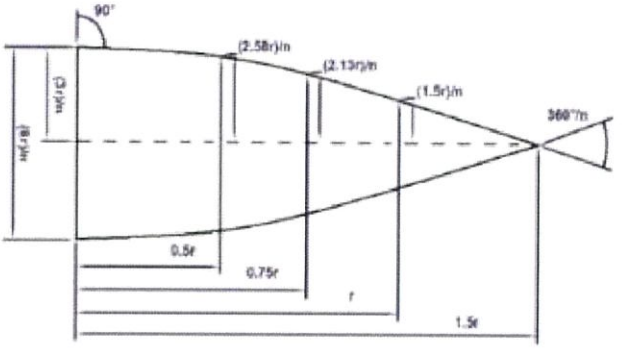
จากการแสดงให้เห็นความสัมพันธ์ระหว่างมวลของ CanSat ต่อขนาดรัศมีของร่มชูชีพที่ระดับความเร็วต่างๆ จะพบว่าที่ความเร็ว 5 m/s และมวล 500 กรัม ร่มชูชีพควรมีรัศมีไม่ต่ำกว่า 0.5 เมตร แสดงผลการคำนวณ ดังนี้

$$r = \sqrt{\frac{2 \times 0.5 \times 9.81}{1.225 \times 1.5 \times 5^2 \times \pi}}$$

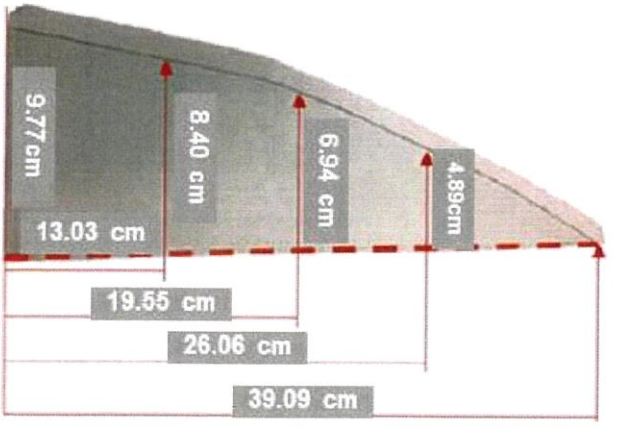
$$r = 0.26m$$

เมื่อ n คือ จำนวนชิ้นส่วนของร่ม

r คือ รัศมีของร่ม



รูปที่ 3.14 รูปแบบของร่มชูชีพแบบ Semi-Spherical [30]



รูปที่ 3.15 แบบของชิ้นส่วนร่มแต่ละชิ้น [30]

รูปที่ 3.16 ร่มชูชีพสภาพเสร็จสมบูรณ์และมวลของร่มชูชีพ

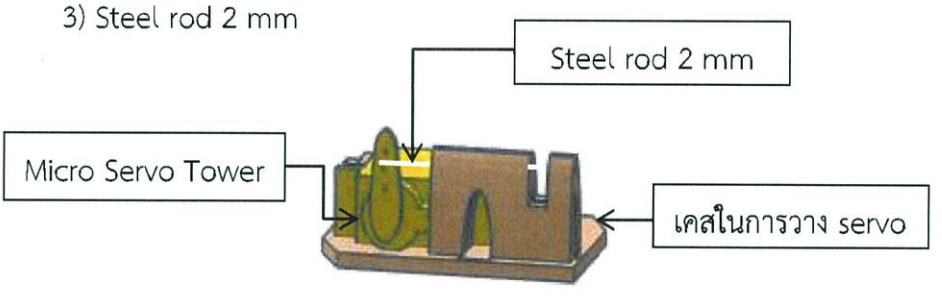
การออกแบบและติดตั้งกลไกการทำงานของ Servo motor

ในส่วนของโครงสร้างการปลดสลักเพื่อใช้งานแมคคานิก แบ่งเป็น 3 องค์ประกอบหลักๆ แสดงดังรูปที่ 3.17 คือ

1) Micro Servo Tower Pro SG90

2) เคสในการวาง servo

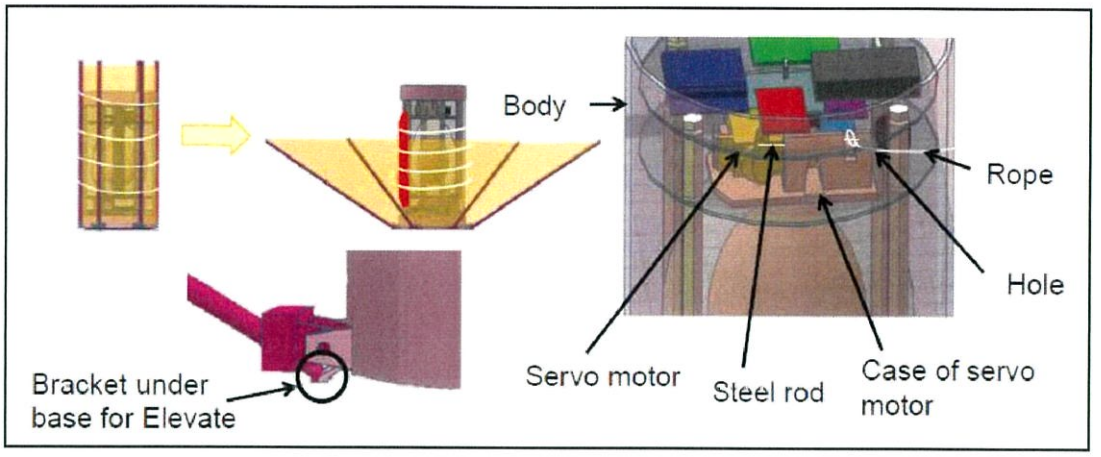
3) Steel rod 2 mm



รูปที่ 3.17 โครงสร้างการปลดสลัก

โดยในส่วนของโครงสร้างนี้จะแบ่งเป็น 2 ส่วน ดังนี้

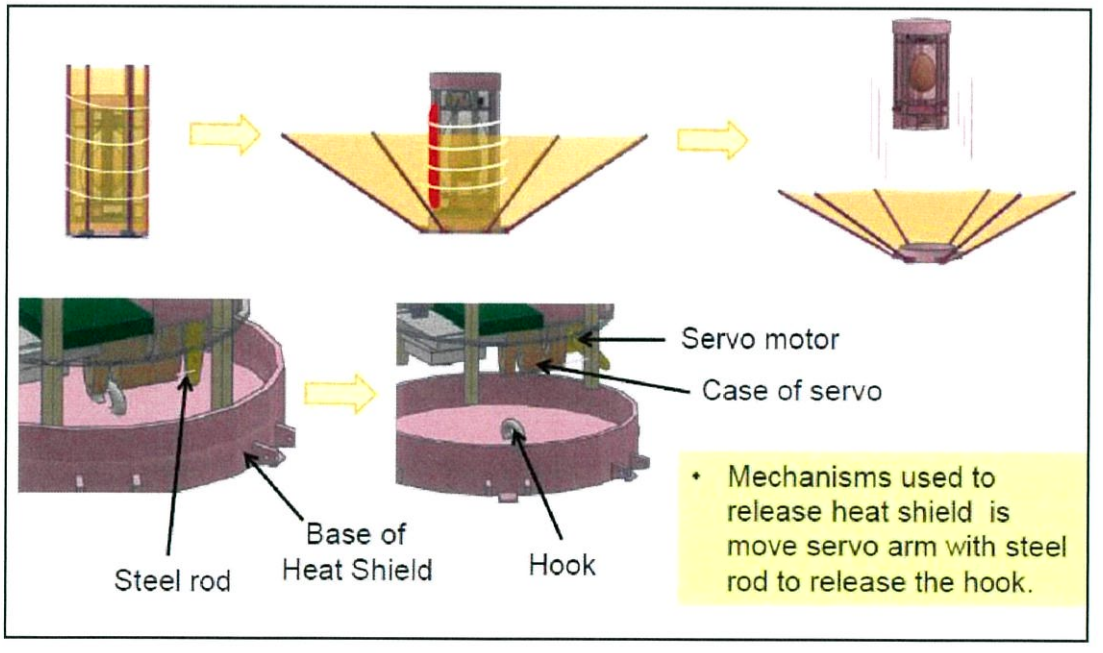
ส่วนที่ 1 โครงสร้างการปล่อยเชือกที่พันรอบ heat shield โดยปล่อยใช้ heat shield ที่ระดับความสูง 700 เมตร



รูปที่ 3.18 กลไกการปล่อยใช้งาน heat shield

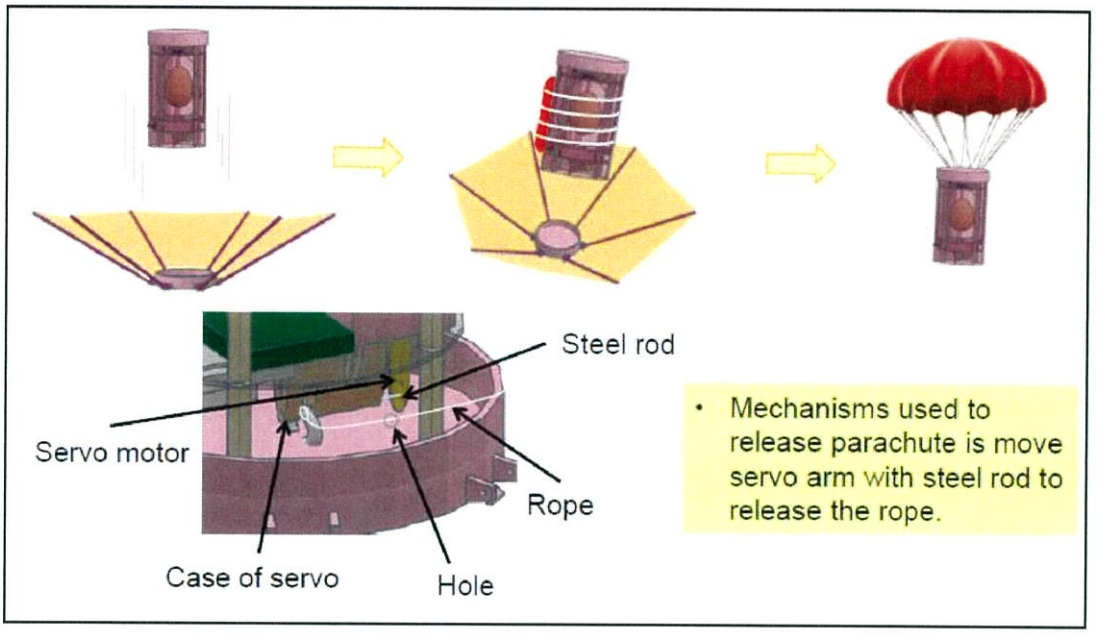
กลไกที่ใช้ในการเปลี่ยนสถานะไปใช้งานโดยการเลื่อนแขน servo ด้วย steel rod เพื่อปล่อยเชือกใช้ปีกของ heat shield และแขนของ heat shield ซึ่งพับเก็บไว้กับลำตัวจะต้องปล่อยใช้อัตโนมัติด้วยสปริงและที่ค้ำภายใต้ฐาน heat shield ดังแสดงในรูปที่ 3.18

ส่วนที่ 2 โครงสร้างการปล่อย aero-braking heat shield ทิ้งและปล่อยใช้ร่มชูชีพที่ระดับความสูง 300 เมตร



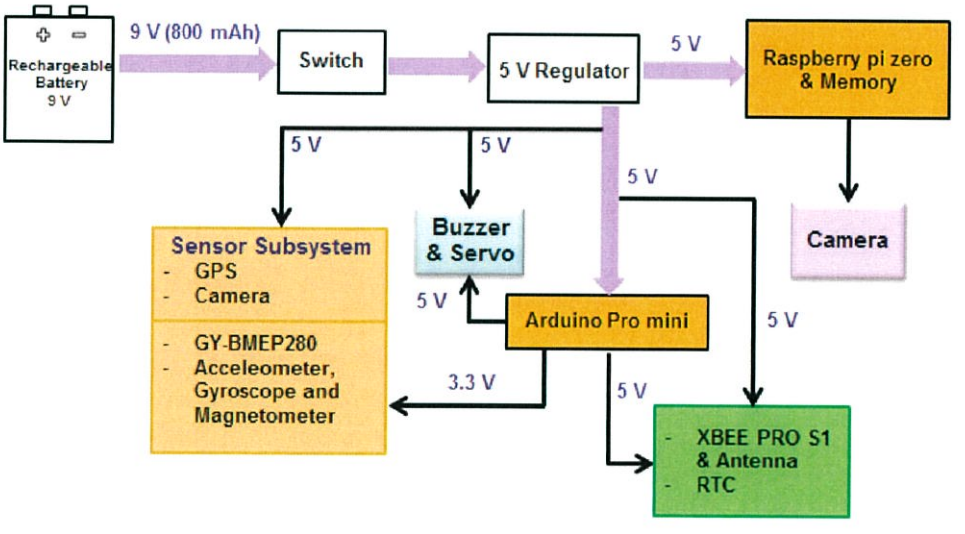
รูปที่ 3.19 กลไกการปล่อย heat shield

จากรูปที่ 3.19 กลไกที่ใช้ในการปล่อย heat shield ทิ้ง คือแขนของ servo เคลื่อนที่ซึ่งติดกับ steel rod เพื่อปล่อย hook ที่ติดอยู่กับ heat shield และปล่อยเชือกที่พันเก็บร่มชูชีพไว้เพื่อใช้งานร่มชูชีพ แสดงดังรูปที่ 3.20



รูปที่ 3.20 กลไกการปล่อยร่มชูชีพใช้งาน

3.1.7 การออกแบบและสร้าง ELECTRICAL POWER SUBSYSTEM



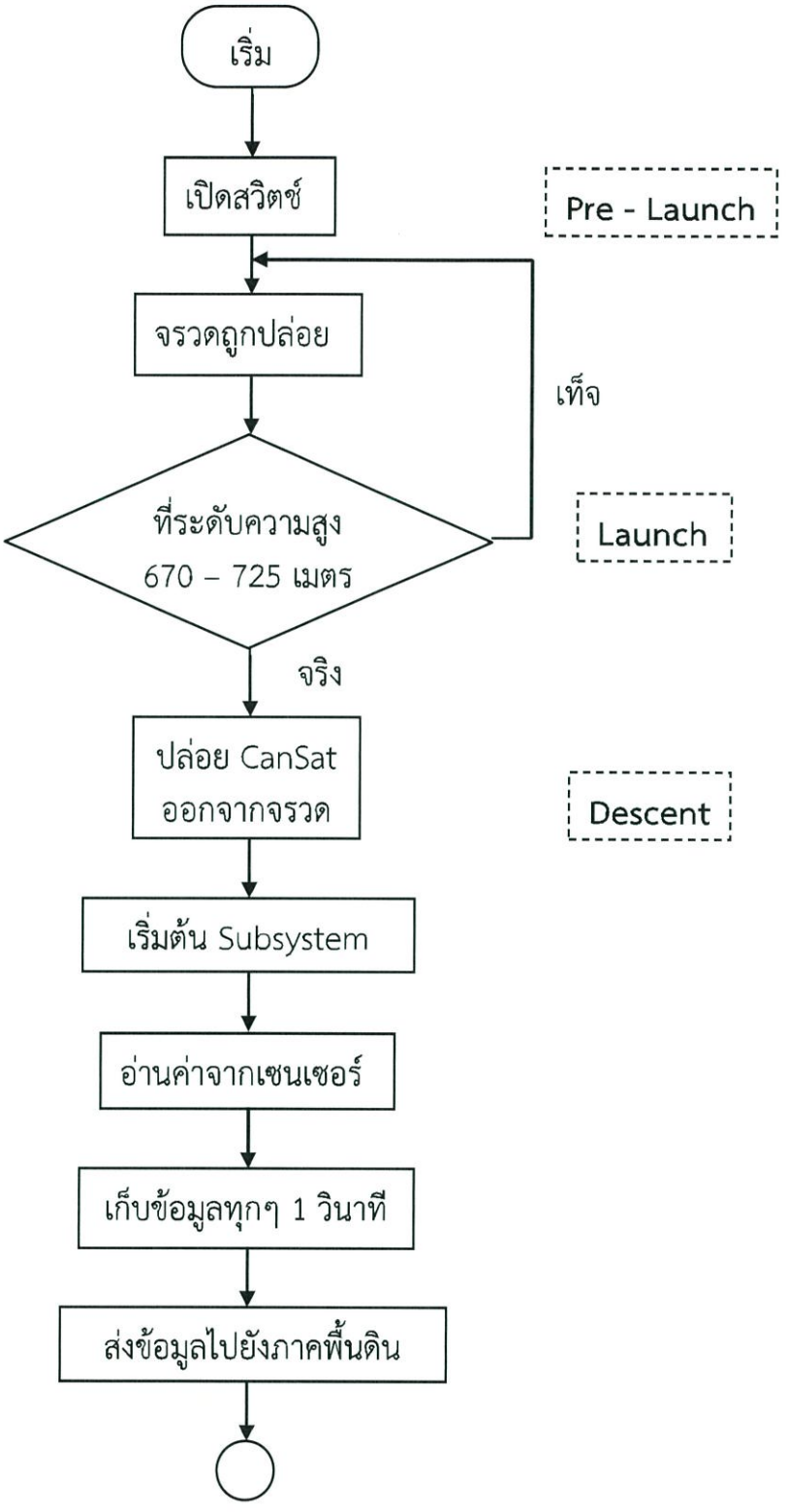
รูปที่ 3.21 บล็อกไดอะแกรมพลังงานไฟฟ้า

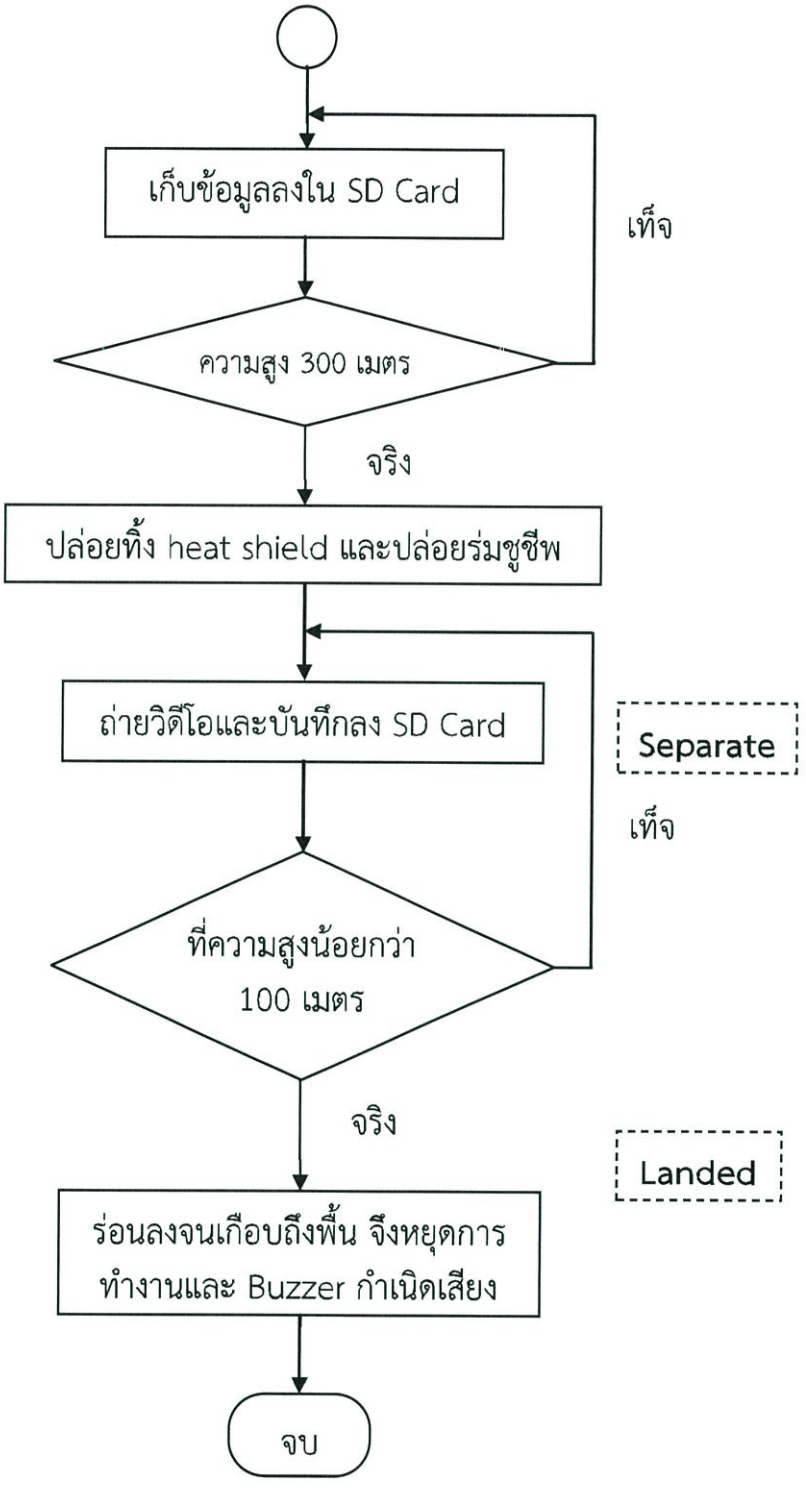
จากรูปที่ 3.21 แสดงบล็อกไดอะแกรมการใช้พลังงานไฟฟ้าในแต่ละอุปกรณ์ โดยใช้แบตเตอรี่ 9 โวลต์ 800 mAh ใช้ LM 7805 แบ่งแรงดันไฟฟ้าให้เหลือ 5 โวลต์เพื่อใช้เลี้ยงอุปกรณ์อิเล็กทรอนิกส์ต่างๆ และแรงดันไฟฟ้า 3.3 โวลต์ ใช้เลี้ยงเซนเซอร์บางตัว เพื่อลดพลังงานในการใช้งานภายในวงจร

3.2 การทำงานของระบบแบบจำลองดาวเทียมขนาดเล็ก

3.2.1 โปรแกรมการทำงานของระบบย่อยทั้งหมดและการออกแบบ FLIGHT SOFTWARE STATE

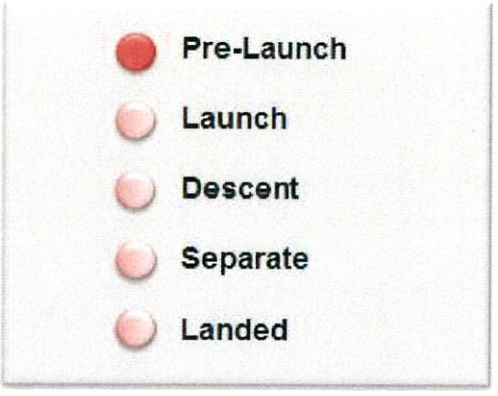
การออกแบบโปรแกรมการทำงานของอุปกรณ์และเซ็นเซอร์ต่างๆภายใน CanSat นั้น จะเริ่มเมื่อทำการเปิดสวิตช์ ทำการปล่อยจรวดพอถึงที่ระดับความสูง 670-725 เมตร จรวดจะปล่อย CanSat ออกจากจรวด และระบบเซ็นเซอร์ทั้งหมดจะทำการเก็บค่า โดยเก็บข้อมูลและส่งข้อมูลไปยังภาคพื้นดินในทุกๆวินาที ในขณะนั้นก็เก็บข้อมูลลงใน SD Card ด้วย พอถึงระดับความสูง 300 เมตร จะปล่อย heat shield ทิ้งออกและปล่อยร่มชูชีพ ทำการถ่ายวิดีโอ ณ ขณะนั้น ต่อจากนั้นร่อนลงจนถึงใกล้พื้นดิน จึงหยุดระบบการทำงานทั้งหมด และให้ Buzzer กำเนิดเสียงเตือนเพื่อร้องให้รับรู้ว่า CanSat ตกสู่พื้นดินแล้ว ดัง Flowchart โดยรวมที่แสดงในรูปที่ 3.22


เริ่ม
เปิดสวิตช์
Pre - Launch
จรวดถูกปล่อย
เท็จ
ที่ระดับความสูง
670 – 725 เมตร
Launch
จริง
ปล่อย CanSat
ออกจากจรวด
Descent
เริ่มต้น Subsystem
อ่านค่าจากเซนเซอร์
เก็บข้อมูลทุกๆ 1 วินาที
ส่งข้อมูลไปยังภาคพื้นดิน



รูปที่ 3.22 Flowchart แสดงการทำงานโดยรวม

จากรูปที่ 3.22 จะกำหนดถึงการบอกสถานะในการทำงานในขณะนั้นๆโดยในแต่ละช่วงเวลานั้นมีการปล่อยตัวของอุปกรณ์ต่างๆ โดยแบ่งออกเป็น 5 สถานะ ดังแสดงในรูปที่ 3.23



รูปที่ 3.23 สถานะในการทำงานของแต่ละขั้นตอน

จากรูปที่ 3.23 เป็นการแสดงสถานะ ณ ขณะนั้นๆว่า CanSat อยู่ในกระบวนการใดของการร่อนลงสู่พื้นดิน แบ่งเป็น 5 สถานะของการทำงาน ดังนี้

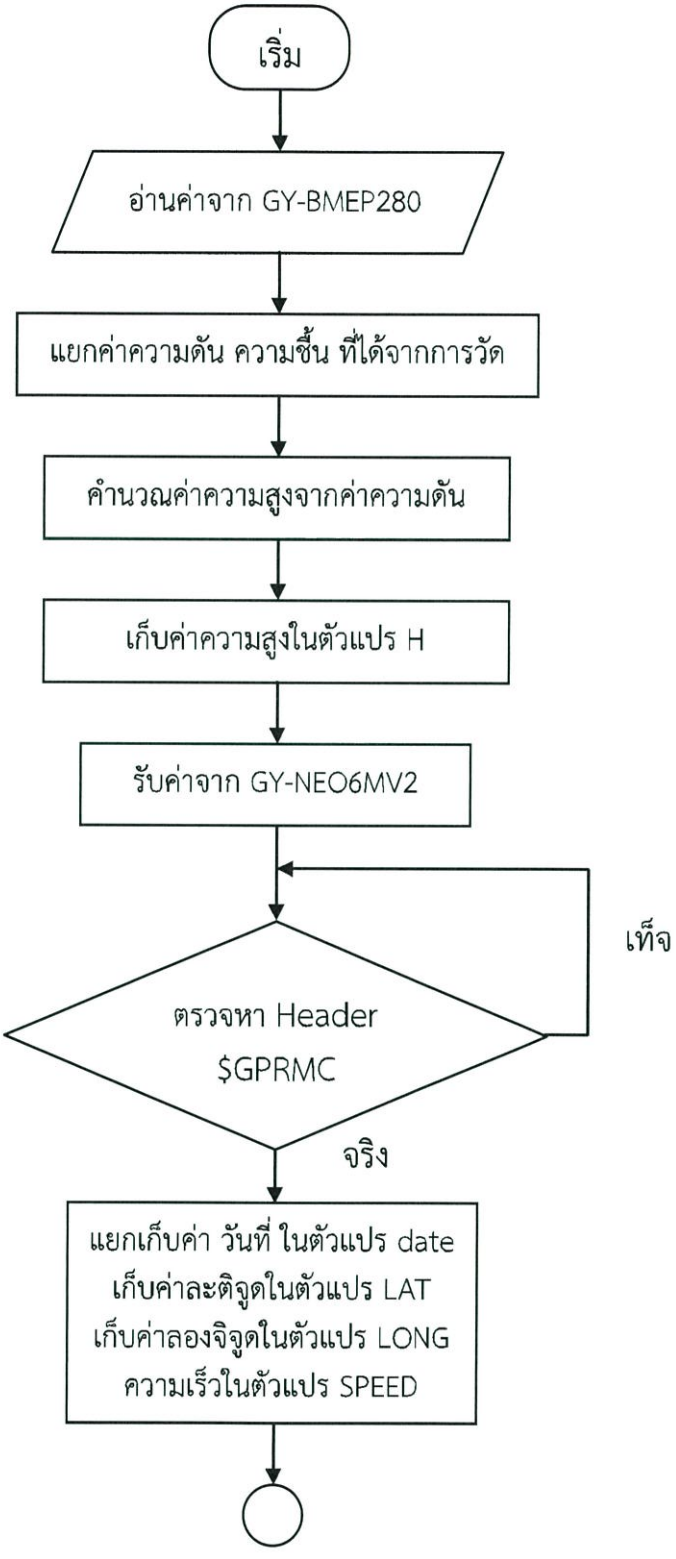
Pre – Launch เป็นสถานะเริ่มต้นในขณะที่เปิดสวิตช์พร้อมเริ่มการใช้งานก่อนการปล่อยจรวดขึ้นสู่ท้องฟ้า

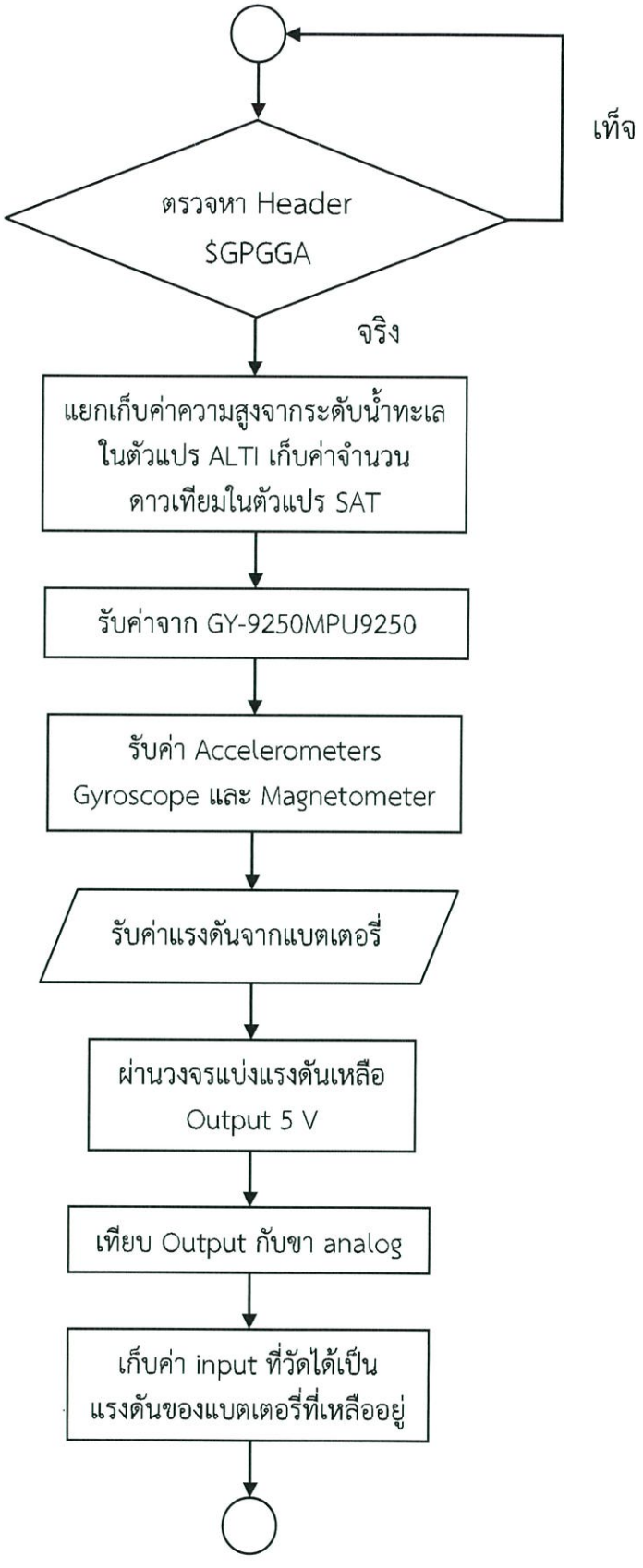
Launch เป็นสถานะในเวลา ณ ขณะที่จรวดถูกปล่อยและกำลังขึ้นสู่ท้องฟ้าจนถึงที่ระดับความสูง 670 – 725 เมตร

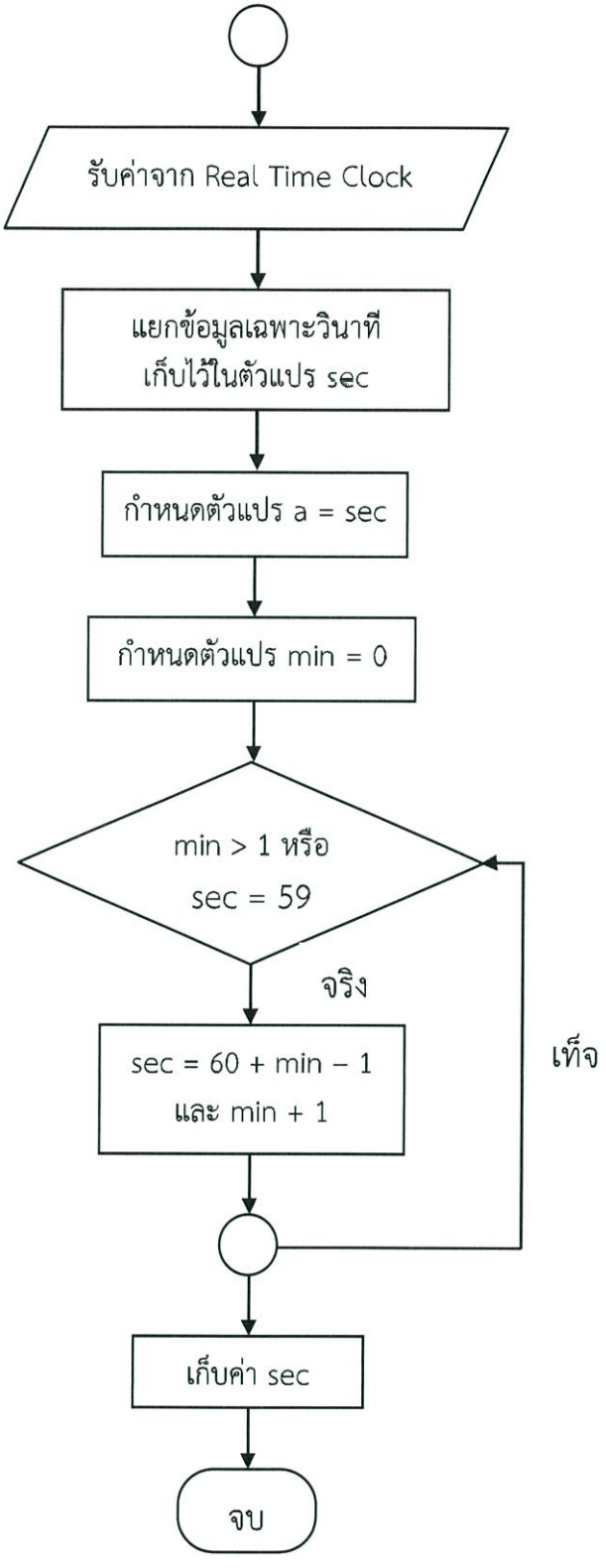
Descent เป็นสถานะเมื่อถึงจุดสูงสุด แล้วต้องปล่อย CanSat ออกจากจรวดและกำลังร่อนลงมา

Separate เป็นสถานะที่บ่งบอกว่าที่ความสูง 300 เมตร จะมีการปล่อย heat shield แยกออกจาก CanSat และ ณ ขณะเวลาเดียวกันนั้นก็มีการปล่อยร่มชูชีพเพื่อใช้งาน และการทำงานของโมดูลกล้องในการถ่ายวิดีโอขณะที่ปล่อย heat shield ไปพร้อมๆกัน

Landed เป็นสถานะแจ้งขณะที่แบบจำลองดาวเทียมขนาดเล็กตกลงสู่พื้นดิน


เริ่ม
อ่านค่าจาก GY-BMEP280
แยกค่าความดัน ความชื้น ที่ได้จากการวัด
คำนวณค่าความสูงจากค่าความดัน
เก็บค่าความสูงในตัวแปร H
รับค่าจาก GY-NEO6MV2
ตรวจหา Header
$GPRMC
เท็จ
จริง
แยกเก็บค่า วันที่ ในตัวแปร date
เก็บค่าละติจูดในตัวแปร LAT
เก็บค่าลองจิจูดในตัวแปร LONG
ความเร็วในตัวแปร SPEED


เท็จ
ตรวจหา Header
$GPGGA
จริง
แยกเก็บค่าความสูงจากระดับน้ำทะเล
ในตัวแปร ALTI เก็บค่าจำนวน
ดาวเทียมในตัวแปร SAT
รับค่าจาก GY-9250MPU9250
รับค่า Accelerometers
Gyroscope และ Magnetometer
รับค่าแรงดันจากแบตเตอรี่
ผ่านวงจรแบ่งแรงดันเหลือ
Output 5 V
เทียบ Output กับขา analog
เก็บค่า input ที่วัดได้เป็น
แรงดันของแบตเตอรี่ที่เหลืออยู่



รูปที่ 3.24 Flowchart แสดงการอ่านค่าจากเซนเซอร์ทั้งหมด

Flowchart แสดงการทำงานโดยละเอียดของการอ่านค่าจากเซ็นเซอร์ทั้งหมด โดยจะแสดงในรูปที่ 3.23 ซึ่งจะประกอบไปด้วยการทำงานของ GY-BMEP280, GY-NEO6MV2, GY-9250MPU9250, Power voltage circuit และ Real Time Clock Modules DS1307 ตามลำดับ โดยเริ่มจากการรับค่าจากเซ็นเซอร์ BME 280 ซึ่งประกอบด้วย ค่าความดัน ความชื้น และอุณหภูมิ จากนั้นทำการแยกข้อมูลเลือกส่งค่าและจัดเก็บค่าอุณหภูมิลง SD Card แยกค่าความดันที่ได้นำไปหาความสูงจากระดับน้ำทะเลกลาง และส่งค่าค่าความดัน และความสูงจากระดับน้ำทะเลกลางและจัดเก็บใน SD Card จากนั้นตามด้วยการรับค่าและอ่านค่าจากโมดูล GY-NEO6MV2 โดยต้องการข้อมูล วัน เวลา ละติจูด ลองจิจูด และความเร็ว จากในส่วนของheader $GPRMC และข้อมูลความสูงจากระดับน้ำทะเลกลางจากheader $GPGGA ก่อนที่จะจัดเก็บค่าลง SD Card ต่อมาอ่านค่าจาก GY-9250MPU9250 ซึ่งจะรับค่า Accelerometers, Gyroscope และ Magnetometer แล้วรับค่าแรงดันจากแบตเตอรี่ นำมาผ่านวงจรแบ่งแรงดันให้เหลือ 5 V และทำการเทียบ Output กับขา analog จะได้เป็นค่าแรงดันของแบตเตอรี่ที่เหลืออยู่ จากนั้นรับค่าจาก Real Time Clock Sensor โดยแยกเก็บค่าเฉพาะวินาที เพื่อนำเป็นการนับเวลาในภารกิจการแข่งขัน และจัดเก็บลง SD Card

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดสอบการทำงานของ SENSOR SUBSYSTEM

เซ็นเซอร์วัดสภาพอากาศ GY-BMEP280

ทำการทดสอบของอุณหภูมิ โดยทำการวัดค่าอุณหภูมิของสภาพอากาศทั้ง 3 ครั้งที่อุณหภูมิแตกต่างกันคือที่อุณหภูมิภายนอก อาคารอุณหภูมิห้องปรับอากาศ และอุณหภูมิภายในตู้เย็นตามลำดับแล้วทำการหาค่าเฉลี่ยเพื่อทำการเปรียบเทียบค่าอุณหภูมิที่วัดได้จากเทอร์โมมิเตอร์และทำการบันทึกผล

เซ็นเซอร์ GPS GY-NEO6MV2

ทำทดสอบของเซ็นเซอร์ GPS GY-NEO6MV2 โดยทำการรับข้อมูลที่ได้จากเซ็นเซอร์ GPS GY-NEO6MV2 และวัดค่าสัญญาณบิตข้อมูลที่ได้ของขา Rx ของอุปกรณ์จากออสซิลโลสโคป และทำการถอดรหัสข้อมูลที่ได้

การทดสอบการทำงานของความสูงที่เปลี่ยนแปลงจากระดับน้ำทะเลกลาง

โดยทำการทดสอบจากดาดฟ้าตึกสูง 13 เมตร นำ CanSat ผูกกับเชือกแล้วอ่านค่าความสูงจากระดับน้ำทะเลกลางที่ได้จากความดันและ GPS ในตำแหน่งนั้น แล้วบันทึกค่า จากนั้นปล่อยเชือกทีละ 1 เมตรแล้วทำการบันทึกค่าความสูงจากระดับน้ำทะเลกลางที่ได้จากโมดูล BME280 และ GPS GY-NEO6MV2

เซ็นเซอร์ POWER VOLTAGE

ทำการทดสอบการอ่านค่าของ Power Voltage เทียบกับค่าจากแหล่งจ่ายไฟที่ป้อนเข้ามาโดยทำการวัดที่ค่าแรงดันแตกต่างกัน 4 ค่า ดังนี้ 3V, 5V, 7V และ 9V ตามลำดับ โดยทำการหาค่าเฉลี่ยและเปอร์เซ็นต์ความคลาดเคลื่อน

เซ็นเซอร์ความเร่ง 9 แกน GY-9250

ทำการทดสอบ GY-9250 โดยทำการวัดค่า Accelerometers, Gyroscope และ Magnetometer ซึ่ง Accelerometers หรือการเคลื่อนไหวทั้ง 3 แกน (x,y,z) โดยจะวัดเทียบกับการสังเกตการปรับเปลี่ยนทิศทางการเคลื่อนที่ของโมดูลตามมุมต่างๆที่ทดสอบในแนวแกน x,y,z โดยนำค่าที่ได้จากการทดสอบโดยทำการสุ่ม 10 ค่าของมุมนั้นๆนำมาเฉลี่ยรวมทั้ง 3 แกน พร้อมทั้งหาเปอร์เซ็นต์ความคลาดเคลื่อนและวัดสัญญาณบิตข้อมูลของค่า Accelerometers และ Gyroscope โดยแสดงข้อมูล ความเร่งในแนวแกน x (High), ความเร่งในแนวแกน x (Low), ความเร่งในแนวแกน y (High), ความเร่งในแนวแกน y (Low), ความเร่งในแนวแกน z (High), ความเร่งในแนวแกน z (Low), Gyroscope แกน x (High) , Gyroscope แกน x (Low), Gyroscope แกน y (High), Gyroscope แกน y (Low), Gyroscope แกน z (High) และ Gyroscope แกน z (Low)

การวัดค่า Magnetometer หรือ Compass (สนามแม่เหล็กโลกหรือเข็มทิศ) ทั้ง 4 ทิศ ได้แก่ ทิศเหนือ ทิศตะวันออก ทิศใต้ และทิศตะวันตก ตามลำดับ โดยเทียบกับเข็มทิศแบบตลับและแสดงค่าบนจอที่อ่านได้จากโมดูลทั้ง 4 ทิศ พร้อมทั้งหาเปอร์เซ็นต์ความคลาดเคลื่อน

Raspberry Pi Camera Board v1.3

ทำการอัดวิดีโอที่ความละเอียดต่างๆในอัตราเฟรม 30 และ 60 โดยเปรียบเทียบหน่วยความจำที่ใช้ในการบันทึกวิดีโอกับหน่วยความจำที่บรรจุในแบบจำลองดาวเทียมขนาดเล็กเพื่อให้เพียงพอต่อการใช้งานในการอัดวิดีโอ 1-2 นาที

3.3.2 การทดสอบการทำงานของ ELECTRICAL POWER SUBSYSTEM

วัดการใช้กำลังและกระแสไฟฟ้าของอุปกรณ์แต่ละชนิดและพลังงานโดยรวมที่ใช้ในการทำงานทั้งหมด พร้อมทั้งเวลาในการใช้งานของแบตเตอรี่ 9 โวลต์ 1 ก้อน

3.3.3 การทดสอบการทำงานของ COMMUNICATION AND DATA HANDLING SUBSYSTEM

Real Time Clock Modules DS1307

ทำการวัดสัญญาณบิตข้อมูลจาก Tiny Real Time Clock Modules โดยแสดงข้อมูลขา SCLและ SDA ของ IC เบอร์ DS 1307 จะแสดงข้อมูลเรียงตามลำดับ ตั้งแต่ 7 bit DS1307 address คือ 1101000, R/ W (Read/Write) , A (Acknowledge) , register address จากนั้นตามด้วยข้อมูล second , minutes, hours, day, date, month, year โดยจะเลือกใช้ข้อมูล second เพียงนับเวลาเป็นวินาทีเท่านั้น

Microstrip dipole antenna 2.4 GHz

ทำการทดสอบคุณสมบัติการใช้งานของสายอากาศ Microstrip dipole 2.4 GHz ที่ทำการออกแบบและประดิษฐ์ขึ้น โดยทำการวัดค่า S_{11}, Bandwidth, ความถี่กลาง, VSWR, รูปแบบการแพร่กระจาย และวัดค่า Gain ของสายอากาศ รวมถึงการใช้งานร่วมกับ XBEE PRO S1 ในการส่งข้อมูลไปยังภาครับ

3.3.4 การทดสอบการทำงานของ MECHANICAL SUBSYSTEM

Aero-braking heat shield

ทดสอบการ drop test โดยทำการเตรียมถุงทรายถุงละ 100 กรัม ใส่ในภาชนะที่ติดกับ Aero-braking heat shield ที่ระดับความสูงประมาณ 50 เมตร ทำการปล่อย Aero-braking heat shield ร่อนลงมาจากตึกแล้วทำการจับเวลา แล้วคำนวณหาความเร็วของ Aero-braking heat shield จากสูตร v=s/t ทำการทดสอบ 3 ครั้ง แล้วบันทึกผลเพื่อหาค่าเฉลี่ยจากนั้นทำการทดสอบเช่นเดิมแต่เพิ่มถุงทรายทีละ 100 กรัม จนครบ 400 กรัม แล้วเปรียบเทียบค่าที่คำนวณได้ตามทฤษฎี

ร่มชูชีพ (Parachute)

โดยทำการเตรียมถุงทรายถุงละ 100 กรัม ใส่ในภาชนะที่ติดกับร่มชูชีพ แล้วปล่อยที่ระดับความสูง 55 เมตร จากนั้นจับเวลาที่ร่มชูชีพและภาชนะที่บรรจุถุงทรายใช้เคลื่อนที่ลงถึงพื้นดิน แล้วคำนวณหาความเร็วของร่มชูชีพและภาชนะที่บรรจุถุงทรายหนัก 100 กรัม จากสูตร $v = s/t$ ทำการทดสอบซ้ำ 3 ครั้ง แล้วบันทึกผลเพื่อหาค่าเฉลี่ย จากนั้นทำการทดสอบเช่นเดิมแต่เพิ่มถุงทรายทีละ100 กรัม จนครบ 700 กรัม แล้วบันทึกผลการทดลองเปรียบเทียบกับค่าคำนวณที่ได้ตามทฤษฎี

บทที่ 4

ผลการทดลอง

4.1 ผลการทดสอบการทำงานของ SENSOR SUBSYSTEM

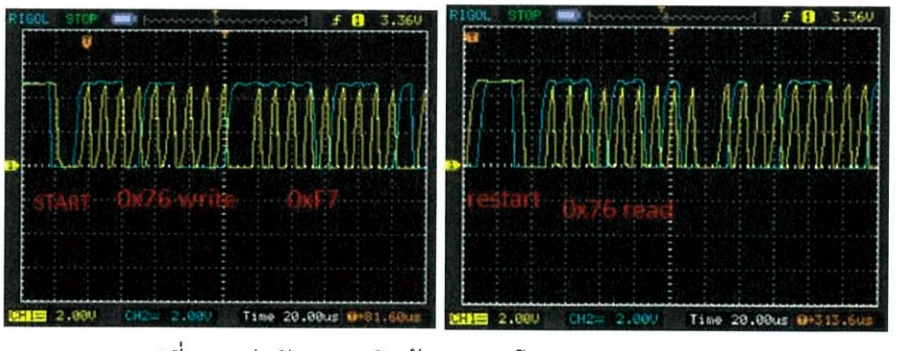
4.1.1 ผลการทดสอบของ GY-BMEP280

โดยจากการวัดค่าอุณหภูมิของสภาพอากาศทั้ง 3 ครั้งที่อุณหภูมิแตกต่างกันคือที่อุณหภูมิภายนอกอาคาร อุณหภูมิห้องปรับอากาศ และอุณหภูมิภายในตู้เย็น ตามลำดับ ซึ่งเมื่อนำไปหาค่าความคลาดเคลื่อนจะได้ดังตารางที่ 4.1

ตารางที่ 4.1 เปรียบเทียบค่าที่วัดได้จากเซ็นเซอร์โมดูล GY-BMEP 280 กับค่าที่วัดจาเทอร์โมมิเตอร์ (อ้างอิง) ในสภาพแวดล้อม 3 สภาวะ

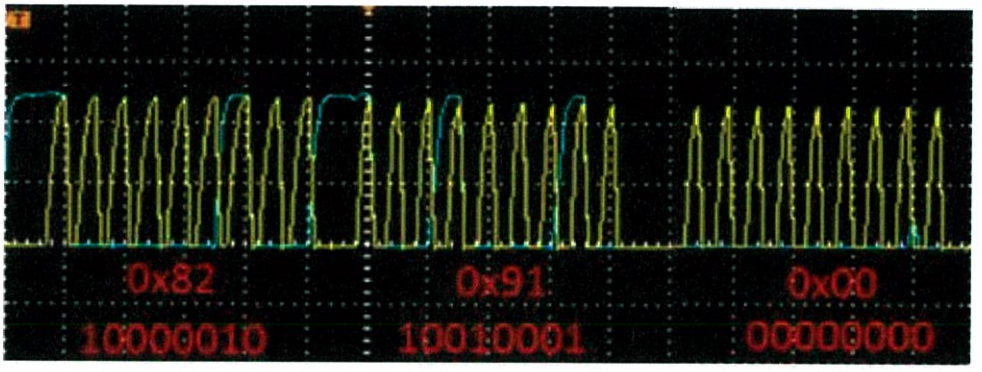
สภาพแวดล้อม	ค่าที่วัดจาก เทอร์โมมิเตอร์	ค่าที่วัดได้จากเซนเซอร์ โมดูล GY-BMEP 280	ค่าความผิดพลาด (%)
อุณหภูมิภายนอกอาคาร	33.00°C	33.10°C	0.30
อุณหภูมิห้องปรับอากาศ	27.00°C	27.28°C	1.04
อุณหภูมิภายในตู้เย็น	6.00°C	6.11°C	1.83
เฉลี่ย			1.06

จากตารางที่ 4.1 ซึ่งเป็นการเปรียบเทียบค่าที่วัดได้จากเซ็นเซอร์โมดูลกับค่าที่วัดจากเทอร์โมมิเตอร์ (อ้างอิง) ในสภาพแวดล้อม 3 สภาวะ พบว่าค่าที่ได้จากอุปกรณ์จะมีค่าความคลาดเคลื่อนเกิดขึ้นจริงเฉลี่ย 1.06 %



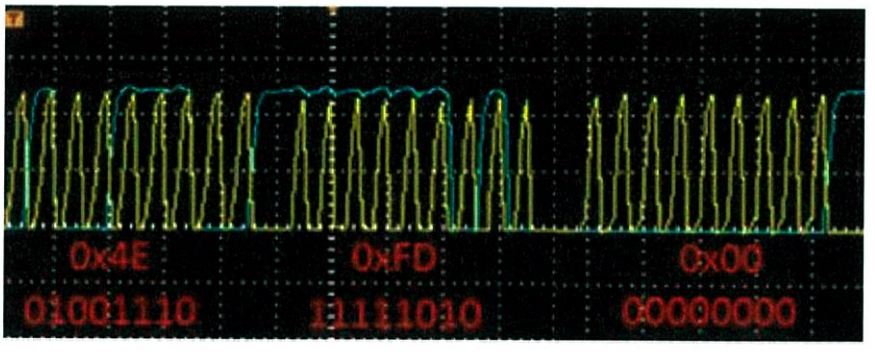
รูปที่ 4.1 ค่าสัญญาณบิตข้อมูลจากโมดูลGY-BMEP280

จากรูปที่ 4.1 ค่าสัญญาณบิตข้อมูลจากโมดูล GY-BMEP 280 แสดงข้อมูลเป็นดังนี้ CH1:SCL , CH2:SDA , Address BME280 = 0x76 และ Register 0xF7 - 0xF9 เป็นค่าความดัน , Register 0xFA - 0xFC เป็นค่าอุณหภูมิ และตามด้วย Register 0xFD – 0xFE เป็นค่าความชื้น



รูปที่ 4.2 ค่าสัญญาณบิตข้อมูลจากโมดูลGY-BMEP280 (ค่าอุณหภูมิ)

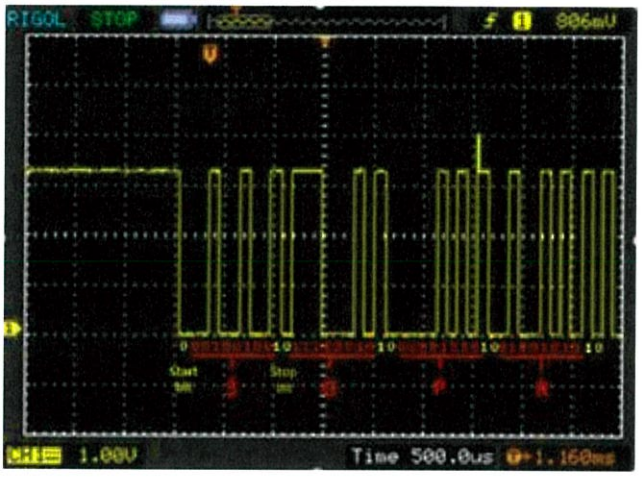
จากรูปที่ 4.2 แสดงค่าสัญญาณบิตที่วัดจาก Oscilloscope ซึ่งมีค่าอุณหภูมิ 82910 และค่าสัญญาณบิตที่วัดจาก Oscilloscope ของค่าความดันมีค่าเท่ากับ 4EFD0 แสดงดังรูปที่ 4.3



รูปที่ 4.3 สัญญาณบิตข้อมูลจากโมดูลGY-BMEP280 (ค่าความดัน)

4.1.2 ผลการทดสอบของ GPS GY-NEO6MV2

จากการทดสอบรับข้อมูลของเซ็นเซอร์ GPS GY-NEO6MV2 โดยทำการรับข้อมูลที่ได้จากเซ็นเซอร์ GPS GY-NEO6MV2 และวัดค่าสัญญาณบิตข้อมูลที่ได้ของขา Rx ของอุปกรณ์จากออสซิลโลสโคปและทำการถอดรหัสข้อมูลที่ได้ จะได้ดังตัวอย่างรูปที่ 4.4



รูปที่ 4.4 ค่าสัญญาณบิตข้อมูลจากโมดูล GY-NEO6MV2

ค่าบิตข้อมูลจากโมดูล GY-NEO6MV2 จะส่งมายัง Microcontroller อันซึ่งจะอยู่ในรูปแบบการสื่อสารแบบ UART โดยข้อมูลที่ส่งจะอยู่ในรูปรหัส ASCII 8 bit แล้วต่อด้วย Stop bit 1 bit เช่น ส่ง Protocol $GPRMC ดังแสดงในรูปที่ 4.4

4.1.3 ผลการทดสอบของความสูงที่เปลี่ยนแปลงจากระดับน้ำทะเลกลาง

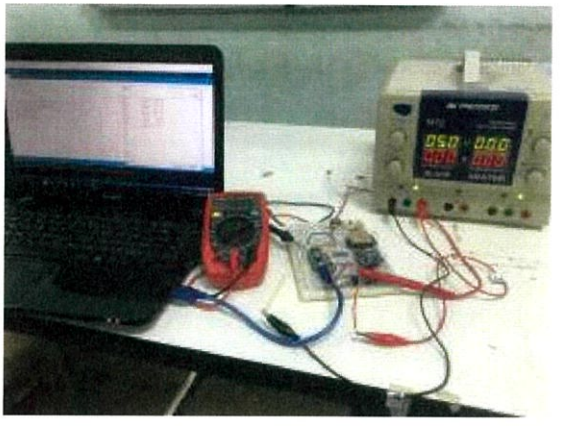
โดยจากการทดสอบจากดาดฟ้าตึกสูง 13 เมตร เมื่อนำ CanSat ผูกกับเชือกแล้วอ่านค่าความสูงจากระดับน้ำทะเลกลางที่ได้จากความดันและ GPS ในตำแหน่งนั้นแล้วบันทึกค่า จากนั้นปล่อยเชือกทีละ 1 เมตรแล้วทำการบันทึกค่าความสูงจากระดับน้ำทะเลกลางที่ได้จาก GY-BMEP 280 และ GPS GY-NEO6MV2 เปรียบเทียบกับความสูงที่เปลี่ยนไปทีละ 1 เมตร จะได้ผลดังตารางที่ 4.2

ตารางที่ 4.2 ค่าเปรียบเทียบระดับความสูงจากระดับน้ำทะเลกลางที่เปลี่ยนแปลงไปได้ค่าจาก GPS และที่ได้จากค่าความดัน

ความยาวเชือกจากดาดฟ้า (เมตร)	ความสูงที่เปลี่ยนแปลง(เมตร)	
	ค่าจาก GPS (เมตร)	ค่าจากความดัน (เมตร)
0	-	-
1	1.50	0.34
2	0.90	1.00
3	0.40	0.51
4	3.90	1.25
5	5.30	1.34
6	0.40	0.92
7	6.20	1.18
8	3.70	1.08
9	3.40	0.92
10	2.70	1.09
ความยาวที่เปลี่ยนแปลงเฉลี่ย	**2.84**	**0.96**

จากตารางที่ 4.2 ค่าเปรียบเทียบระดับความสูงจากระดับน้ำทะเลกลางที่เปลี่ยนแปลงไปได้ค่าจาก GPS และที่ได้จากค่าความดัน พบว่าค่าความสูงจากระดับน้ำทะเลกลางที่ได้จากค่าความดันมีค่าใกล้เคียงกว่าความสูงจากระดับน้ำทะเลกลางที่ได้จากค่า GPS และมีความคลาดเคลื่อนเท่ากับ 3.70 เปอร์เซ็นต์

4.1.4 ผลการทดสอบของ POWER VOLTAGE



รูปที่ 4.5 การวัดค่าของแหล่งจ่ายไฟโดยใช้ Power Voltage

จากรูปที่ 4.5 ทดสอบการอ่านค่าของ Power Voltage Circuit เทียบกับค่าจากแหล่งจ่ายไฟที่ป้อนเข้ามา โดยทำการวัดที่ค่าแรงดันแตกต่างกัน 4 ค่า ดังนี้ 3V, 5V, 7V และ 9V ตามลำดับ แสดงดังตารางที่ 4.3

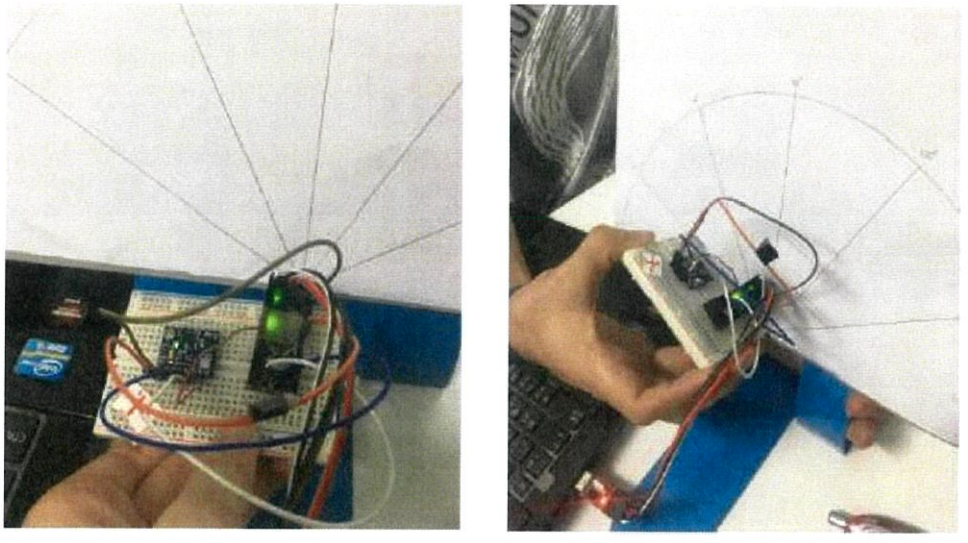
ตารางที่ 4.3 ค่าแรงดันจาก Power Voltage Sensor และ แหล่งจ่ายไฟ

ค่าที่	ค่าที่วัดได้จาก Power Voltage Sensor			
	3 V	5 V	7 V	9 V
1	2.63	4.75	6.90	9.07
2	2.68	4.80	6.95	9.05
3	2.70	4.72	6.87	9.07
4	2.65	4.80	6.92	9.12
5	2.70	4.82	6.95	9.09
ค่าเฉลี่ย	2.67	4.78	6.92	9.08
% คลาดเคลื่อน	10.93	4.40	1.14	0.89

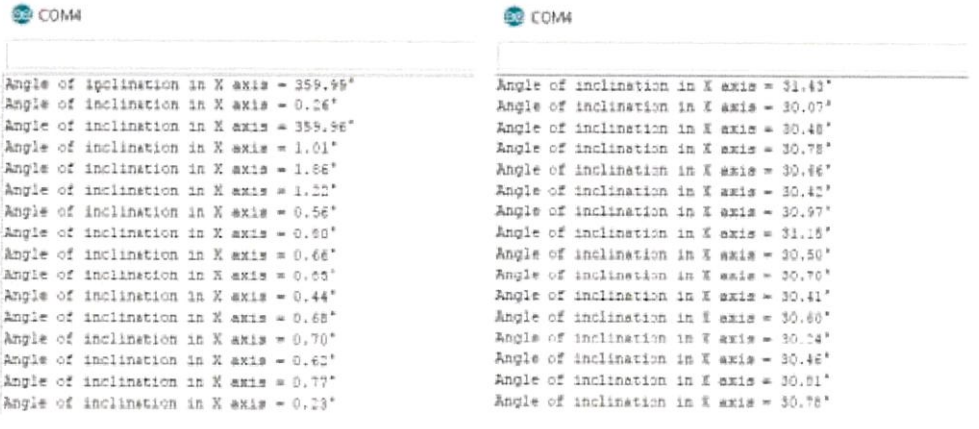
จากตารางที่ 4.3 แสดงข้อมูลจากการทดสอบการวัดแรงดันของ Power Voltage Circuit เทียบกับค่าแรงดันจากแหล่งจ่ายไฟที่ระดับแรงดัน 3V, 5V, 7V และ 9V ทำการหาค่าเฉลี่ยและเปอร์เซ็นต์ความคลาดเคลื่อน โดยหาเปอร์เซ็นต์ความคลาดเคลื่อนของ Power Voltage Circuit มีค่าเท่ากับ 4.34 %

4.1.5 ผลการทดสอบของ GY-9250MPU9250

โดยทำการวัดค่า Accelerometers หรือการเคลื่อนไหวทั้ง 3 แกน (x,y,z) กับการสังเกตปรับเปลี่ยนทิศทางการเคลื่อนที่ของโมดูลตามมุมต่างๆที่ทำการทดสอบในแนวแกน x,y,z แสดงดังรูปที่ 4.6, 4.8 และ 4.10 ตามลำดับ โดยนำค่าที่ได้จากการทดสอบโดยทำการสุ่ม 10 ค่า ของมุมนั้นๆ นำมาเฉลี่ยรวมทั้งในแนวแกน x,y,z แสดงดังตารางที่ 4.5, 4.6 และ 4.7 ตามลำดับ



รูปที่ 4.6 การปรับระดับทิศทางการเคลื่อนที่ของโมดูลตามมุมในแนวแกน x (มุม 0 และ 30 องศา ตามลำดับ)



รูปที่ 4.7 มุมจากการอ่านค่าจากตัวโมดูลในแนวแกน x (มุม 0 และ 30 องศา ตามลำดับ)

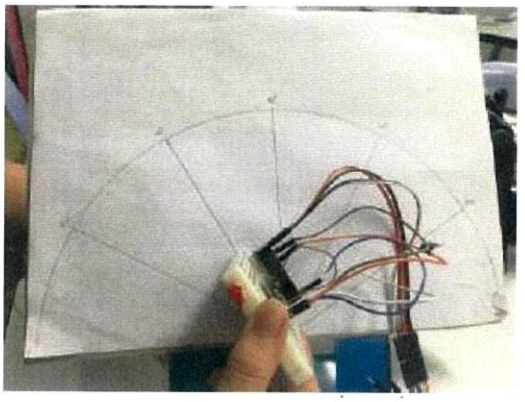
จากรูปที่ 4.7 เป็นค่ามุมที่อ่านได้จากการปรับทิศทางการเคลื่อนที่ของโมดูลให้สัมพันธ์กันในแนวแกน x ดังแสดงในรูปที่ 4.4 (ที่มุม 0 และ 30 องศา ตามลำดับ)

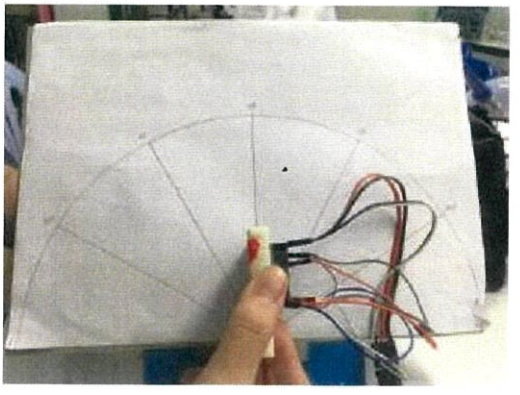
ตารางที่ 4.4 ค่าของมุมต่างๆที่วัดได้จากโมดูลGY-9250MPU9250 ในแนวแกน x

ค่าสุ่มที่	มุม (องศา)						
	0	30	60	90	120	150	180
1	359.99	30.66	60.30	90.97	119.06	150.70	180.94
2	359.96	30.36	59.68	90.74	119.57	150.51	181.23
3	1.01	31.15	58.91	90.95	118.80	150.35	179.83
4	1.22	30.42	60.09	90.44	119.52	150.47	179.73
5	0.56	30.97	60.21	90.62	119.01	150.72	179.75
6	0.80	30.46	60.70	90.32	119.57	150.45	180.06
7	0.66	30.36	60.78	89.91	118.84	150.94	180.29
8	0.44	30.51	60.53	89.84	119.72	150.64	180.30
9	0.56	30.52	60.30	89.99	119.20	150.68	180.16
10	0.96	29.56	60.10	90.20	120.05	150.58	180.01
เฉลี่ยรวม	0.616	30.497	60.16	90.398	119.334	150.604	180.23
เปอร์เซ็นต์ความคลาดเคลื่อน(%)	0.17	1.66	0.27	0.44	0.56	0.40	0.13

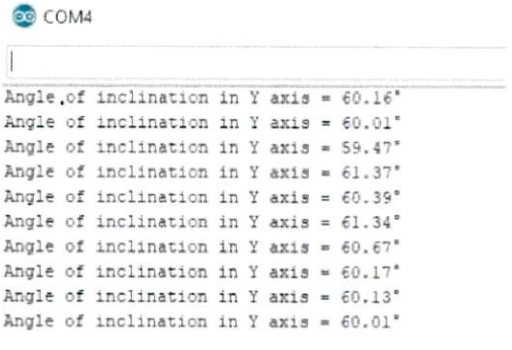
จากตารางที่ 4.4 แสดงค่าสุ่ม 10 ค่าของค่ามุมการเคลื่อนที่ของโมดูลแต่ละมุมในแนวแกน x ทั้งหมด 7 มุม นั่นคือ 0,30,60,90,120,150 และ 180 องศา ตามลำดับ ทำการเก็บค่าการปรับการเคลื่อนที่ของโมดูลทีละ 30 องศา จากนั้นนำค่า 10 ค่าของแต่ละองศามาเฉลี่ยรวมพร้อมทั้งหาเปอร์เซ็นต์ความคลาดเคลื่อน ดังสรุปค่ามุมที่วัดได้ในแนวแกน x ได้ดังนี้

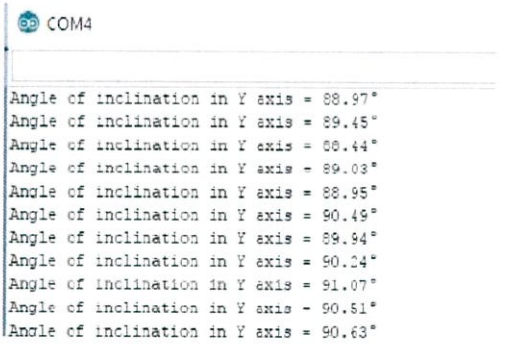
ที่มุม 0 องศา เฉลี่ยรวม 0.616 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.17 %
ที่มุม 30 องศา เฉลี่ยรวม 30.497 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 1.66 %
ที่มุม 60 องศา เฉลี่ยรวม 60.16 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.27 %
ที่มุม 90 องศา เฉลี่ยรวม 90.398 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.44 %
ที่มุม 120 องศา เฉลี่ยรวม 119.334 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.56 %
ที่มุม 150 องศา เฉลี่ยรวม 150.604 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.40 %
ที่มุม 180 องศา เฉลี่ยรวม 180.230 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.13 %





รูปที่ 4.8 การปรับระดับทิศทางการเคลื่อนที่ของโมดูลตามมุมในแนวแกน y
(มุม 60 และ 90 องศา ตามลำดับ)





รูปที่ 4.9 มุมจากการอ่านค่าจากตัวโมดูลในแนวแกน y (มุม 60 และ 90 องศา ตามลำดับ)

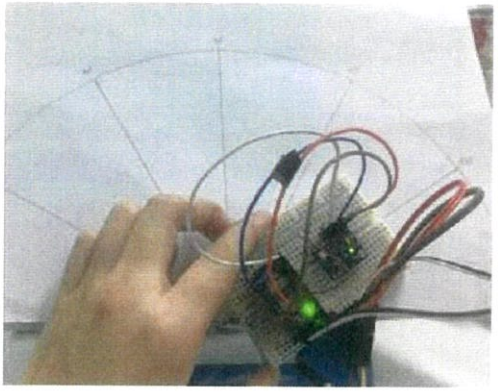
ตารางที่ 4.5 ค่าของมุมต่างๆที่วัดได้จากโมดูล GY-9250MPU9250 ในแนวแกน y

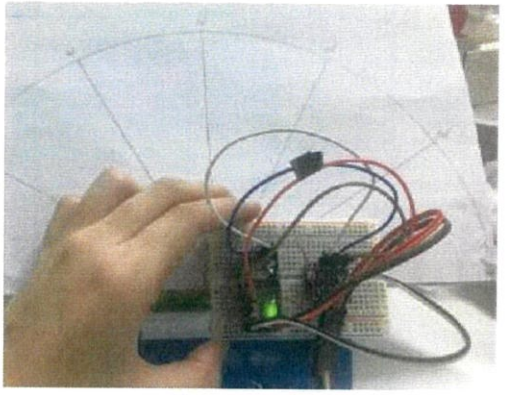
ค่าสุ่มที่	มุม (องศา)						
	0	30	60	90	120	150	180
1	0.88	29.45	60.16	89.45	120.25	149.40	176.22
2	0.96	29.92	60.01	88.44	120.53	150.29	180.60
3	1.23	30.78	59.47	89.03	120.87	151.08	180.70
4	1.23	29.72	60.67	90.49	119.90	151.78	180.80
5	1.17	29.53	60.17	89.94	121.01	150.77	180.85
6	1.03	29.91	60.13	90.27	120.04	151.06	180.43

ค่าสุ่มที่	มุม (องศา)						
	0	30	60	90	120	150	180
7	0.85	30.04	60.01	90.01	120.18	150.46	178.82
8	1.05	30.14	60.61	90.99	120.64	151.42	179.41
9	0.92	30.10	59.99	90.70	120.35	151.19	180.26
10	1.07	29.07	59.93	90.25	120.65	151.02	179.90
เฉลี่ยรวม	1.039	29.866	60.115	89.957	120.442	150.847	179.799
เปอร์เซ็นต์ความคลาดเคลื่อน(%)	0.29	0.45	0.19	0.05	0.37	0.56	0.11

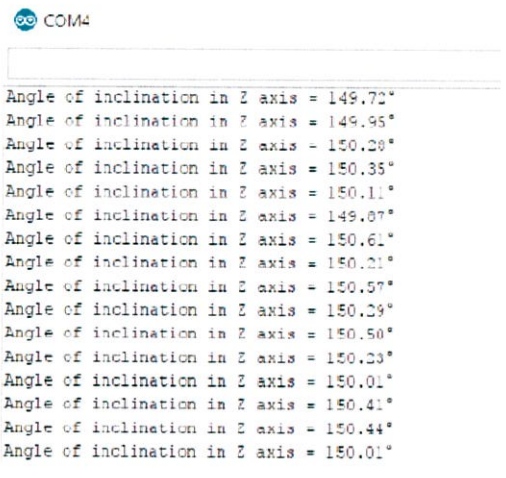
จากตารางที่ 4.5 แสดงค่าสุ่ม 10 ค่าของค่ามุมการเคลื่อนที่ของโมดูลแต่ละมุมในแนวแกน y ทั้งหมด 7 มุม นั่นคือ 0,30,60,90,120,150 และ 180 องศา ตามลำดับ ทำการเก็บค่าการปรับการเคลื่อนที่ของโมดูลทีละ 30 องศา จากนั้นนำค่า 10 ค่าของแต่ละองศามาเฉลี่ยรวม พร้อมทั้งหาเปอร์เซ็นต์ความคลาดเคลื่อน ดังสรุปค่ามุมที่วัดได้ในแนวแกน y ได้ดังนี้

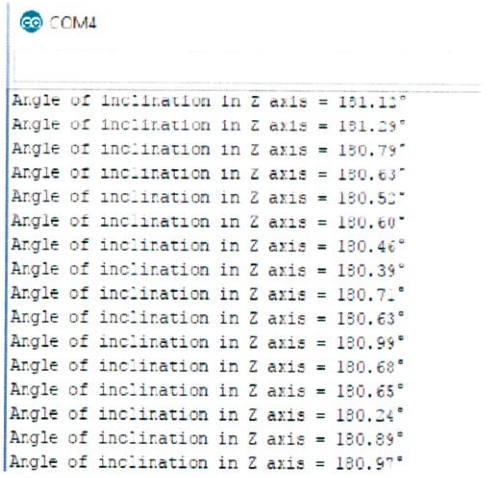
ที่มุม 0 องศา เฉลี่ยรวม 1.039 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.29 %
ที่มุม 30 องศา เฉลี่ยรวม 29.866 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.45 %
ที่มุม 60 องศา เฉลี่ยรวม 60.115 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.19 %
ที่มุม 90 องศา เฉลี่ยรวม 89.957 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.05 %
ที่มุม 120 องศา เฉลี่ยรวม 120.442 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.37 %
ที่มุม 150 องศา เฉลี่ยรวม 150.847 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.56 %
ที่มุม 180 องศา เฉลี่ยรวม 179.799 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.11 %





รูปที่ 4.10 การปรับระดับทิศทางการเคลื่อนที่ของโมดูลตามมุมในแนวแกน z
(มุม 150 และ180 องศา ตามลำดับ)





รูปที่ 4.11 มุมจากการอ่านค่าจากตัวโมดูลในแนวแกน z (มุม 150 และ 180 องศา ตามลำดับ)

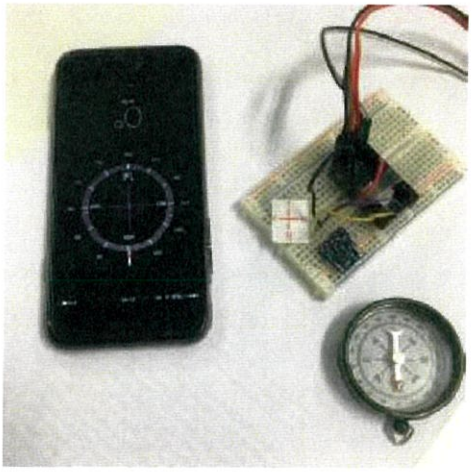
ตารางที่ 4.6 ค่าของมุมต่างๆที่วัดได้จากโมดูล GY-9250MPU9250 ในแนวแกน z

ค่าสุ่มที่	มุม (องศา)						
	0	30	60	90	120	150	180
1	0.91	29.38	61.38	90.24	120.65	149.72	181.12
2	0.54	30.18	60.08	90.01	121.27	149.95	180.63
3	359.87	30.28	60.04	90.10	120.44	150.01	180.68
4	359.12	30.38	60.11	90.28	120.38	150.41	180.24
5	358.40	30.36	60.60	89.85	120.30	150.44	180.81
6	359.87	30.19	60.30	90.21	120.12	149.97	180.30
7	359.69	30.21	59.98	89.63	120.35	149.93	180.85
8	359.54	29.84	59.97	90.12	120.10	150.18	181.02
9	0.06	29.93	60.20	90.23	120.44	150.04	180.29
10	359.92	30.18	60.44	90.06	120.35	150.31	180.39
เฉลี่ยรวม	0.208	30.093	60.31	90.073	120.44	150.096	180.633
เปอร์เซ็นต์ความคลาดเคลื่อน (%)	0.58	0.31	0.52	0.08	0.37	0.06	0.35

จากตารางที่ 4.6 แสดงค่าสุ่ม 10 ค่าของค่ามุมการเคลื่อนที่ของโมดูลแต่ละมุมในแนวแกน z ทั้งหมด 7 มุม นั่นคือ 0,30,60,90,120,150 และ 180 องศา ตามลำดับ ทำการเก็บค่าการปรับการเคลื่อนที่ของโมดูลทีละ 30 องศา จากนั้นนำค่า 10 ค่าของแต่ละองศามาเฉลี่ยรวม พร้อมทั้งหาเปอร์เซ็นต์ความคลาดเคลื่อน ดังสรุปค่ามุมที่วัดได้ในแนวแกน z ได้ดังนี้

ที่มุม 0 องศา เฉลี่ยรวม 0.208 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.58 %
ที่มุม 30 องศา เฉลี่ยรวม 30.093 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.31 %
ที่มุม 60 องศา เฉลี่ยรวม 60.31 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.52 %
ที่มุม 90 องศา เฉลี่ยรวม 90.073 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.08 %
ที่มุม 120 องศา เฉลี่ยรวม 120.44 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.37 %
ที่มุม 150 องศา เฉลี่ยรวม 150.096 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.06 %
ที่มุม 180 องศา เฉลี่ยรวม 180.633 องศา มีเปอร์เซ็นต์ความคลาดเคลื่อน 0.35 %

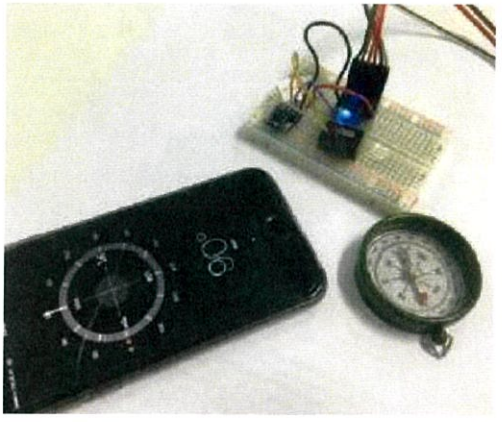
ทำการวัดค่า Magnetometer หรือ Compass (สนามแม่เหล็กโลกหรือเข็มทิศ) ของทั้ง 4 ทิศ ได้แก่ ทิศเหนือ ทิศตะวันออก ทิศใต้ และทิศตะวันตก ตามลำดับ โดยเทียบกับเข็มทิศแบบตลับ และแสดงค่าบนจอที่อ่านได้จากโมดูลทั้ง 4 ทิศ ตามลำดับ แสดงดังรูปที่ 4.12, 4.13, 4.14 และ 4.15 ตามลำดับ



COM5

mag:	82	-32	-183	heading:	338.69
mag:	67	-16	-188	heading:	344.96
mag:	63	-17	-188	heading:	344.90
mag:	62	-16	-187	heading:	345.53
mag:	60	-15	-194	heading:	345.96
mag:	51	-7	-195	heading:	352.18
mag:	51	-7	-194	heading:	352.16
mag:	51	-8	-196	heading:	351.09
mag:	48	-7	-195	heading:	351.70
mag:	46	-5	-198	heading:	353.80
mag:	45	-4	-192	heading:	354.92
mag:	41	-3	-196	heading:	355.82
mag:	42	-1	-193	heading:	358.64
mag:	38	0	-197	heading:	0.00

รูปที่ 4.12 การเทียบทิศของโมดูลกับเข็มทิศแบบตลับและแสดงค่าเป็นองศาของทิศเหนือ



COM5

mag:	0	53	-168	heading:	90.00
mag:	0	54	-169	heading:	90.00
mag:	0	54	-172	heading:	90.00
mag:	0	52	-169	heading:	90.00
mag:	4	54	-156	heading:	85.76
mag:	0	48	-178	heading:	90.00
mag:	-1	49	-177	heading:	91.17
mag:	-2	49	-182	heading:	92.34
mag:	1	49	-184	heading:	88.83
mag:	1	49	-172	heading:	88.83
mag:	2	49	-177	heading:	87.66
mag:	2	48	-178	heading:	87.61
mag:	0	49	-176	heading:	90.00

รูปที่ 4.13 การเทียบทิศของโมดูลกับเข็มทิศแบบตลับและแสดงค่าเป็นองศาของทิศตะวันออก

COM5

mag:	-153	-3	-182	heading:	181.12
mag:	-152	-3	-182	heading:	181.13
mag:	-153	-3	-190	heading:	181.12
mag:	-153	-3	-185	heading:	181.12
mag:	-152	-3	-186	heading:	181.13
mag:	-152	-4	-192	heading:	181.51
mag:	-153	-4	-171	heading:	181.50
mag:	-152	-3	-188	heading:	181.13
mag:	-152	-4	-187	heading:	181.51
mag:	-152	-4	-190	heading:	181.51
mag:	-153	-4	-187	heading:	181.50
mag:	-153	-4	-189	heading:	181.50

รูปที่ 4.14 การเทียบทิศของโมดูลกับเข็มทิศแบบตลับและแสดงค่าเป็นองศาของทิศใต้

COM5

mag:	17	-338	-138	heading:	272.88
mag:	9	-342	-139	heading:	271.51
mag:	6	-341	-135	heading:	271.01
mag:	-1	-346	-142	heading:	269.83
mag:	-2	-344	-130	heading:	269.67
mag:	-1	-345	-140	heading:	269.83
mag:	0	-346	-130	heading:	270.00
mag:	2	-344	-131	heading:	270.33
mag:	2	-345	-136	heading:	270.33
mag:	3	-344	-132	heading:	270.50
mag:	4	-344	-129	heading:	270.67
mag:	3	-344	-128	heading:	270.50

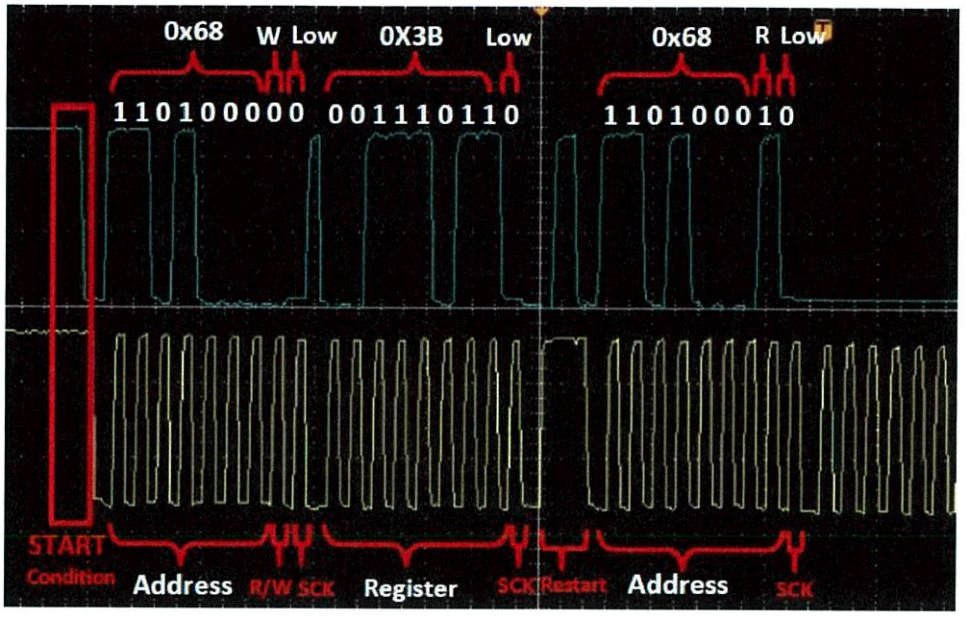
รูปที่ 4.15 การเทียบทิศของโมดูลกับเข็มทิศแบบตลับและแสดงค่าเป็นองศาของทิศตะวันตก

ตารางที่ 4.7 ทิศแสดงในหน่วยองศาของโมดูล GY-9250MPU9250 ทั้ง 4 ทิศ

ค่าสุ่มที่	ทิศ (องศา)			
	ทิศเหนือ (0 องศา)	ทิศตะวันออก (90 องศา)	ทิศใต้ (180 องศา)	ทิศตะวันตก (270 องศา)
1	358.64	90.00	181.12	271.51
2	0.00	90.00	181.13	271.01
3	0.00	90.00	181.50	269.83
4	1.51	90.00	181.13	269.67
5	358.53	91.17	181.51	269.83
6	0.00	92.34	181.13	270.00
7	1.47	90.00	181.14	270.33
8	0.00	91.17	180.75	270.50
9	0.00	91.15	180.38	270.66
10	1.47	92.34	180.13	270.33
ค่าเฉลี่ยรวม	0.162	90.817	180.992	270.367
เปอร์เซ็นต์ความคลาดเคลื่อน(%)	0.05	0.91	0.55	0.14

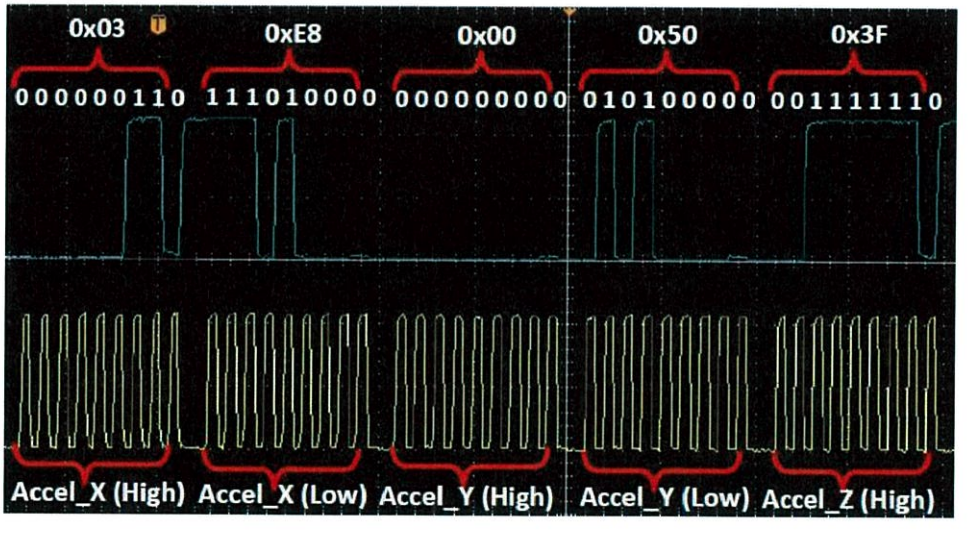
จากตารางที่ 4.7 แสดงค่าสุ่ม 10 ค่าของทิศที่แสดงในหน่วยองศาทั้งหมด 4 ทิศ นั่นคือ ทิศเหนือ (0 องศา), ทิศตะวันออก (90 องศา), ทิศใต้ (180 องศา) และ ทิศตะวันตก (270 องศา) ตามลำดับ ทำการเก็บค่าในแต่ละทิศในการหันของโมดูล จากนั้นนำค่า 10 ค่าของแต่ละทิศมาเฉลี่ยรวม พร้อมทั้งหาเปอร์เซ็นต์ความคลาดเคลื่อน สรุปทิศ (หน่วยองศา) ของโมดูล GY-9250MPU9250 ทั้ง 4 ทิศ ได้ดังนี้

ทิศเหนือ 0 องศา	เฉลี่ยรวม	0.16	องศา	มีเปอร์เซ็นต์ความคลาดเคลื่อน	0.05	%
ทิศตะวันออก90 องศา	เฉลี่ยรวม	90.81	องศา	มีเปอร์เซ็นต์ความคลาดเคลื่อน	0.91	%
ทิศใต้ 180 องศา	เฉลี่ยรวม	180.99	องศา	มีเปอร์เซ็นต์ความคลาดเคลื่อน	0.55	%
ทิศตะวันตก270 องศา	เฉลี่ยรวม	270.36	องศา	มีเปอร์เซ็นต์ความคลาดเคลื่อน	0.14	%

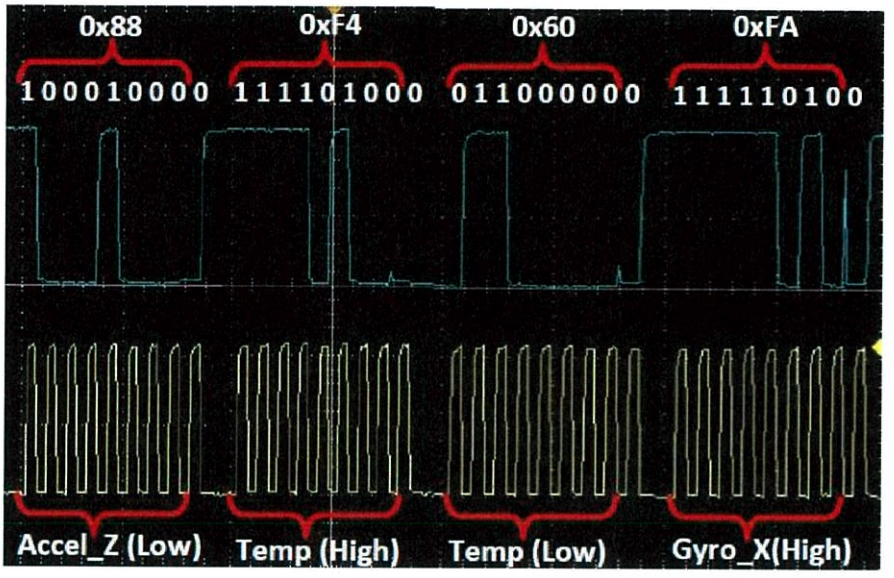


รูปที่ 4.16 ค่าสัญญาณบิตจากโมดูล GY-9250MPU9250

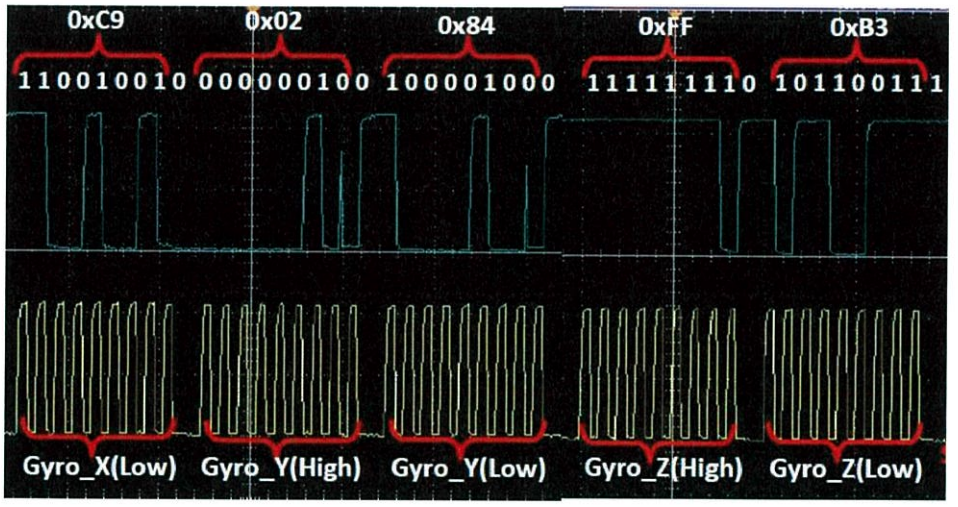
จากรูปที่ 4.16 ค่าสัญญาณบิตข้อมูลจากโมดูล GY-9250MPU9250 แสดงข้อมูลเป็นดังนี้ CH1 คือ SCL (Serial Clock) CH2 คือ SDA (Signals serial data) และ Slave I^2C Address ของชิป MPU9250 คือ 0x68 และ Register 0x3B จากนั้น Restart และเริ่ม Address 0x68 ใหม่อีกครั้ง แล้วตามด้วยข้อมูลตั้งแต่ไบต์ที่ 4 – 23 แสดงดังรูปที่ 4.17, 4.18, 4.19, 4.20 และ 4.21



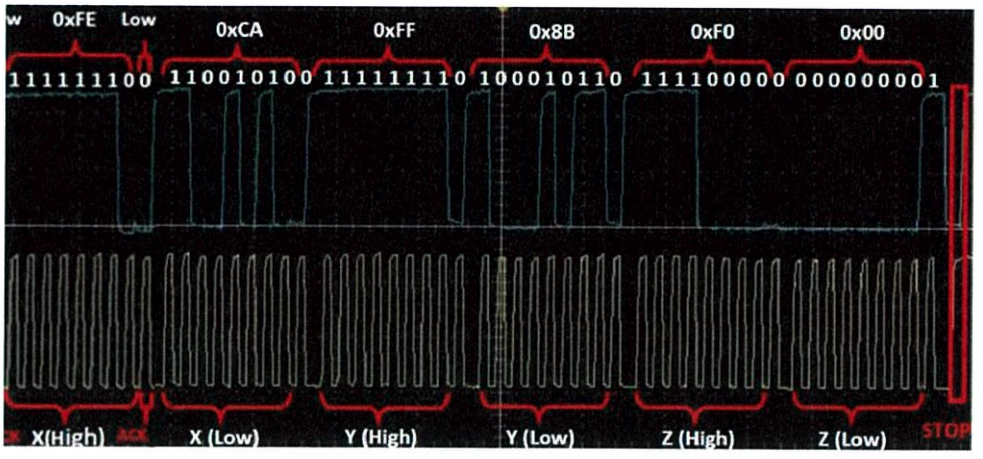
รูปที่ 4.17 ค่าสัญญาณบิตข้อมูลจากโมดูล GY-9250MPU9250 (ไบต์ที่ 4-8)



รูปที่ 4.18 ค่าสัญญาณบิตข้อมูลจากโมดูล GY-9250MPU9250 (ไบต์ที่ 9-12)



รูปที่ 4.19 ค่าสัญญาณบิตข้อมูลจากโมดูล GY-9250MPU9250 (ไบต์ที่ 13-17)



รูปที่ 4.20 ค่าสัญญาณบิตข้อมูลจากโมดูล GY-9250MPU9250 (ไบต์ที่ 18-23)

ค่าสัญญาณบิตข้อมูลที่ได้ตั้งแต่ไบต์ที่ 4-23 แสดงข้อมูล ความเร่งในแนวแกน x (High), ความเร่งในแนวแกน x (Low), ความเร่งในแนวแกน y (High), ความเร่งในแนวแกน y (Low), ความเร่งในแนวแกน z (High), ความเร่งในแนวแกน z (Low), อุณหภูมิ (High), อุณหภูมิ (Low), Gyroscope แกน x (High) , Gyroscope แกน x (Low), Gyroscope แกน y (High), Gyroscope แกน y (Low), Gyroscope แกน z (High) และ Gyroscope แกน z (Low) ตามลำดับ และ Magnetometer x (High), Magnetometer x (Low), Magnetometer y (High), Magnetometer y (Low), Magnetometer z (High), Magnetometer z (Low) และตามด้วยบิต STOP ตามลำดับ

การหาค่าของอุปกรณ์ในแต่ละแกนสามารถหาค่าได้โดยนำค่าที่อ่านได้มาผ่านการ AND GATE กับแอดเดรสของค่าที่อ่านได้ในแต่ละแกน

ค่าที่แสดงผล = ค่าที่อ่านได้ && แอดเดรสของค่าที่อ่าน

ยกตัวอย่างเช่น การแสดงค่าความเร่งแกน x , y และ z

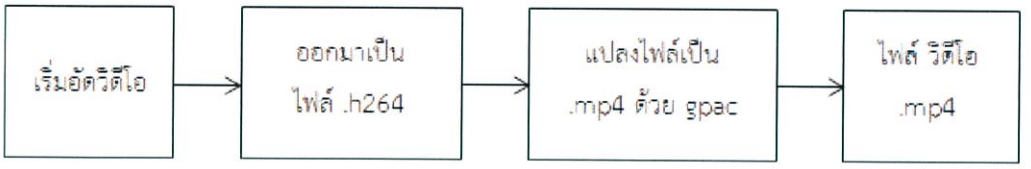
ค่า แกนความเร่ง x = 0x03E8 . 0x3B3C = 0x0328 = 808 degrees/s

ค่า แกนความเร่ง y = 0x0050 . 0x3D3F = 0x0010 = 16 degrees/s

ค่า แกนความเร่ง z = 0x3F88 . 0x3F40 = 0x3F00 = 16,128 degrees/s

4.1.6 ผลการทดสอบของ RASPBERRY PI CAMERA BOARD V1.3

ทำการอัดวิดีโอที่ความละเอียดต่างๆในอัตราเฟรม 30 และ 60 โดยทำการเปรียบเทียบหน่วยความจำที่ใช้ในการบันทึกวิดีโอกับหน่วยความจำที่บรรจุในดาวเทียมขนาดเล็กเพื่อให้เพียงพอต่อการใช้งานในการอัดวิดีโอ 1-2 นาที โดยหลักการทำงาน เป็นดังรูปที่ 4.21



รูปที่ 4.21 กระบวนการแปลงไฟล์จากการอัดวิดีโอ

จากรูปที่ 4.21 แสดงกระบวนการแปลงไฟล์จาก .h264 เป็นไฟล์ .mp4 เพราะว่าไฟล์ .mp4 ซึ่งสามารถเปิดได้ในอุปกรณ์ที่หลากหลายมากกว่า และแสดงหน่วยความจำที่ความละเอียดต่างๆ คือ 640×480 , 1280×720 และ 1920×1080 ในแต่ละอัตราเฟรมที่ 30 และ 60 ตามลำดับ โดยการอัดที่เวลา 1นาที และ 2 นาที แสดงดังตารางที่ 4.8

ตารางที่ 4.8 หน่วยความจำจากการอัดวิดีโอของ Raspberry Pi Camera Board

ความละเอียด	30 เฟรมต่อวินาที		60 เฟรมต่อวินาที	
	1 นาที	2 นาที	1 นาที	2 นาที
640×480	13,195 KB	25,966 KB	36,992 KB	80,012 KB
1280×720	37,451 KB	85,631 KB	123,863 KB	232,122 KB
1920×1080	120,604 KB	224,420 KB	-	-

จากตารางที่ 4.8 สรุปได้ว่า เมื่อถ่ายวิดีโอที่มีความละเอียดมากขึ้นหน่วยจำในการบันทึกก็ยิ่งมากขึ้น อีกทั้งอัตราเฟรมส่งผลต่อหน่วยความจำ ถ้าอัตราเฟรมยิ่งมาก ภาพก็จะสมูทมากขึ้นและใช้หน่วยความจำที่เพิ่มมากขึ้นไปอีก จึงทดสอบถ่ายวิดีโอที่ 1นาที และ 2 นาที ตามลำดับ

4.2 ผลการทดสอบการทำงานของ ELECTRICAL POWER SUBSYSTEM

ทำการทดสอบโดยวัดการใช้กำลังและกระแสไฟฟ้าของอุปกรณ์แต่ละชนิดและพลังงานโดยรวมที่ใช้ในการทำงานทั้งหมด พร้อมทั้งเวลาในการใช้งานของแบตเตอรี่ 9 โวลต์ 1 ก้อน

ตารางที่ 4.9 รายละเอียดการใช้พลังงานของอุปกรณ์แต่ละชนิดจากการวัดด้วยมัลติมิเตอร์

อุปกรณ์	ความต่างศักย์(V)	กระแสไฟฟ้า(mA)	กำลัง(mW)
BME280	3.3	0.008	0.0246
GPS	5	45.2	226
Voltage Divider Circuit	5	0.14	0.7
MPU 9250	3.3	3.65	12.045
Pi Camera	5	103.4	517
Arduino Pro mini	5	12.8	64
Raspberry Pi Zero	5	64.2	321
XBEE Pro S1	5	81.7	408.5
RTC	5	1.5	7.5
Buzzer	5	2.9	14.5
Servo	5	3.56	17.8
รวม	-	**319.06**	**1,589.07**

จากตารางที่ 4.9 แสดงผลรวมการวัดค่ากระแสไฟฟ้าทั้งหมดของแบบจำลองดาวเทียมกระป๋องขนาดเล็กโดยใช้กระแสรวมทั้งหมด เท่ากับ 319.06 mA และใช้กำลังรวมทั้งหมด เท่ากับ 1,589.07 มิลลิวัตต์

BESTON ถ่านชาร์จ 9V 800 mAh (NiMh) ที่เราใช้ซึ่งมีความจุ 800 mA เนื่องจากแบบจำลองดาวเทียมขนาดเล็กใช้กระแสไฟฟ้ารวม 319.06 mA ต่อชั่วโมงเพราะฉะนั้นถ่านหนึ่งก้อนจะสามารถใช้งานได้เป็นระยะเวลา 2.51 ชั่วโมง หรือประมาณ 2 ชั่วโมง 30 นาที

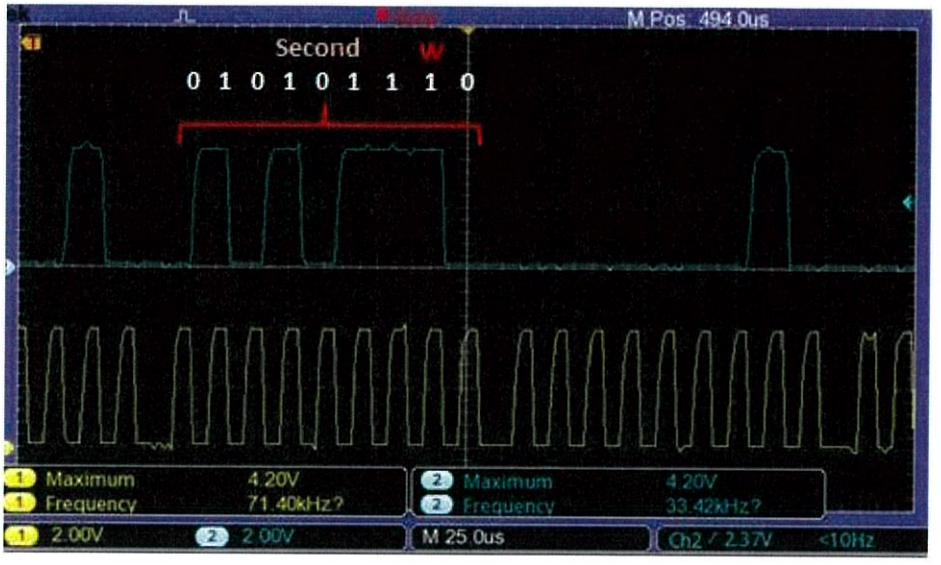
4.3 ผลการทดสอบการทำงานของ COMMUNICATION AND DATA HANDLING SUBSYSTEM

4.3.1 ผลการทดสอบของ REAL-TIME CLOCK MODULES DS1307

ทำการวัดสัญญาณบิตข้อมูลจาก Tiny Real Time Clock Modules แสดงข้อมูลขา SCLและ SDA ของ IC เบอร์ DS 1307 จะแสดงข้อมูลเรียงตามลำดับ ดังนี้ 7 bit DS1307 address คือ 1101000, R/ W (Read/Write) , A (Acknowledge) , register address จากนั้นตามด้วย ข้อมูลsecond , minutes, hours, day, date, month, year โดยจะเลือกใช้ข้อมูล second เพียง นับเวลาเป็นวินาทีเท่านั้น

รูปที่ 4.22 ค่าสัญญาณบิตจากโมดูล Real Time Clock (DS 1307)

จากรูปที่ 4.22 แสดงค่าสัญญาณบิตจากโมดูล Real Time Clock โดยวัดสัญญาณ กำหนดดังนี้ CH 1 คือ SCL (Serial Clock) CH2 คือ SDA (Signals serial data) แสดงบิตข้อมูล เริ่มต้น START, address ของ DS 1307 คือ 1101000 (0x68), Register คือ 0000100 (0x04)



รูปที่ 4.23 ค่าสัญญาณบิตจากโมดูล Real Time Clock (DS 1307)

จากรูปที่ 4.23 แสดงค่าสัญญาณบิตจากโมดูล Real Time Clock แสดงข้อมูลวินาที (second) เป็น 0101011 (0x2B) โดยสามารถหาค่าได้โดยนำค่าที่อ่านได้มาผ่านการ AND GATE กับแอดเดรสของค่าที่อ่านได้

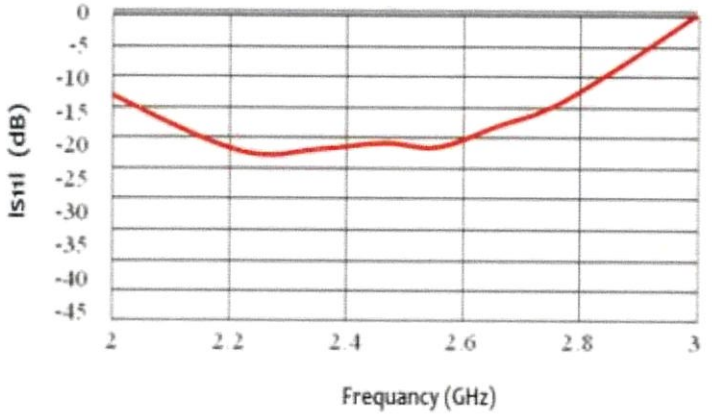
ค่าที่แสดงผล = ค่าที่อ่านได้ && แอดเดรสของค่าที่อ่าน

จะได้ 0101011 && 1101000 = 0101000 = 28 วินาที

4.3.2 ผลการทดสอบของ MICROSTRIP DIPOLE ANTENNA

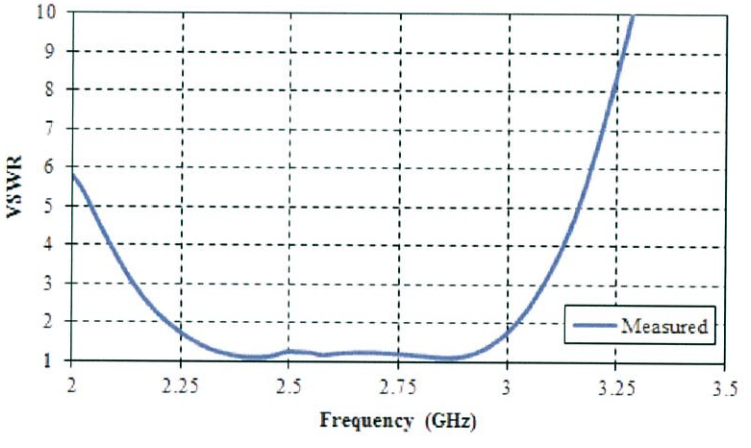
รูปที่ 4.24 Microstrip dipole antenna

ทำการทดสอบสายอากาศที่ความถี่ 2.4 GHz ที่ได้ออกแบบและสร้างขึ้นมา ดังรูปที่ 4.24 ตามขนาดที่ได้ออกแบบไว้ โดยการวัดคุณสมบัติพื้นฐานของสายอากาศ กราฟแสดงค่า S-parameter (S_{11}) แสดงดังรูปที่ 4.25 และ กราฟแสดงค่า Voltage Standing Wave Ratio (VSWR) แสดงดังรูปที่ 4.26



รูปที่ 4.25 กราฟแสดง S-parameter

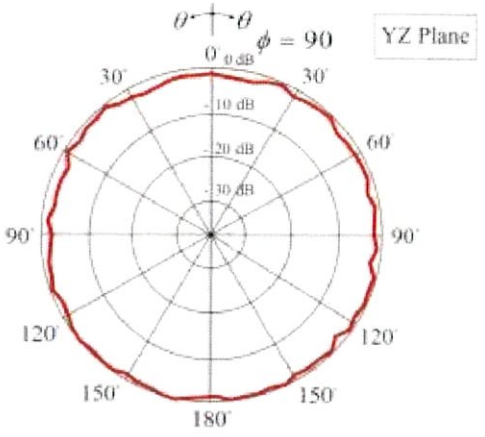
รูปที่ 4.25 กราฟ S-parameter จะได้ค่าช่วงความถี่ที่ใช้งาน เท่ากับ 2.108 – 2.658 GHz มีความถี่กลาง 2.45 GHz และมีค่าการสูญเสียเนื่องจากการย้อนกลับ (Return loss) มีค่าเท่ากับ -21.31 dB ซึ่งพบว่า อยู่ในมาตรฐานที่นำมาใช้งานจริงได้ และสามารถนำมาใช้งานร่วมกับโมดูล XBEE PRO S1 ได้



รูปที่ 4.26 กราฟแสดง Voltage Standing Wave Ratio (VSWR)

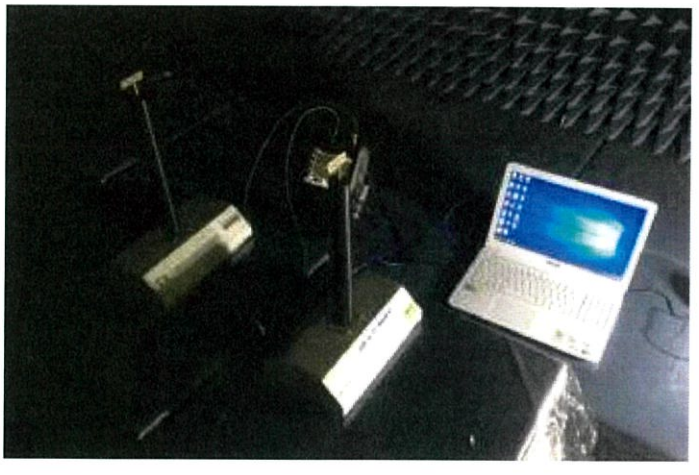
รูปที่ 4.26 เป็นกราฟแสดงค่า Voltage Standing Wave Ratio (VSWR) มีค่าเท่ากับ 1.133 : 1 หมายความว่า แรงดันไฟฟ้ากระแสสลับเนื่องจากคลื่นนิ่งไปตามเส้นสายส่งขึ้นสูงถึงค่าสูงสุดเป็น 1.133 เท่าของแรงดันไฟฟ้า AC ต่ำสุดไปตามแนวสายส่งนั้น ซึ่งมาตรฐานที่สามารถใช้งานได้ต้องไม่เกิน 2 : 1 ถ้ายิ่งมีค่ามาก ก็จะยิ่งมีการสูญเสียและส่งผลกระทบต่อการรับ-ส่งมาก และ

แสดงแบบรูปการแพร่กระจาย ดังรูปที่ 4.27 ซึ่งจะได้การแพร่กระจายเป็นแบบรอบทิศทาง (Omnidirectional)



รูปที่ 4.27 แบบรูปการแพร่กระจาย

จากการวัดแบบรูปการแพร่กระจายของคลื่น จะทำการวัดกำลังในการส่งของสายอากาศ ระหว่าง 2 สายอากาศ ขณะการรับส่งของสายอากาศทั้ง 2 โดยใช้สายอากาศฝั่งส่งและฝั่งรับที่เป็นคู่เหมือน เพื่อให้ง่ายและแม่นยำต่อการหาอัตราขยายของสายอากาศ แสดงได้ดังรูปที่ 4.28 และสามารถวัดกำลังของการส่งได้เท่ากับ -27.54 dB ที่ระยะ 0.365 เมตร สามารถนำมาคำนวณหาอัตราขยายของสายอากาศที่จะใช้งานได้จากการคำนวณที่ 4.1



รูปที่ 4.28 การทดสอบการรับส่งสายอากาศ

$$(G_{ot}) = (G_{or}) = 0.5\left(20\log\left(\frac{4\pi R}{\lambda}\right) + 10\log\left(\frac{P_r}{P_t}\right)\right) \quad (4.1)$$

$$\lambda = \frac{c}{f} \quad (4.2)$$

จากการนำค่ากำลังของการส่งได้เท่ากับ -27.54 dB ที่ระยะ 0.365 เมตร ที่วัดได้จากสายอากาศมาคำนวณแทนค่าลงในสูตรที่ 4.1 และ 4.2 โดยแทนค่า c = 3×10^8 , f = 2.45×10^9, R = 0.365, P_r/P_t = -27.48 dB จะได้อัตราขยาย (Gain) ของสายอากาศคู่เหมือนทั้ง 2 มีค่าเท่ากับ 1.995 dBi

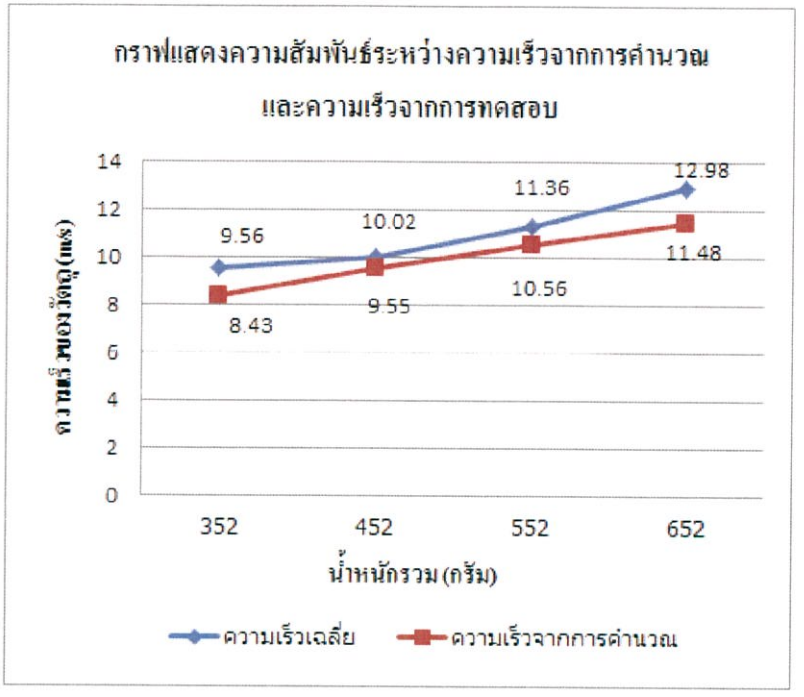
4.4 ผลการทดสอบการทำงานของ MECHANICAL SUBSYSTEM

4.4.1 ผลการทดสอบของ AERO-BRAKING HEAT SHIELD

ทำการทดสอบปล่อยถุงทรายหนัก 100 กรัม จนถึง 400 กรัม ไปพร้อมกับ Aero-braking heat shield และกระป๋องที่มีน้ำหนักรวมกันเท่ากับ 252 กรัม เพื่อหาความเร็วที่ตกลงสู่พื้นดินได้ตามต้องการ จะได้ผลการทดสอบดังตารางที่ 4.10

ตารางที่ 4.10 ค่าเปรียบเทียบความเร็วที่ได้จากการคำนวณทางทฤษฎีกับความเร็วที่ได้จากการทดสอบที่น้ำหนักต่างๆ

น้ำหนัก Aero-braking heat shield และกระป๋อง (252 กรัม) และทราย (กรัม)	ความเร็วของวัตถุ (เมตรต่อวินาที)				
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	คำนวณ
352	10	9.47	9.23	9.56	8.43
452	9.47	10	10.59	10.02	9.55
552	12	11.18	10.91	11.36	10.56
652	12.41	12.68	13.85	12.98	11.48



รูปที่ 4.29 กราฟความสัมพันธ์เปรียบเทียบความเร็วที่ได้จากการคำนวณทางทฤษฎีกับความเร็วที่ได้จากการทดสอบที่น้ำหนักต่างๆของ Aero-braking heat shield

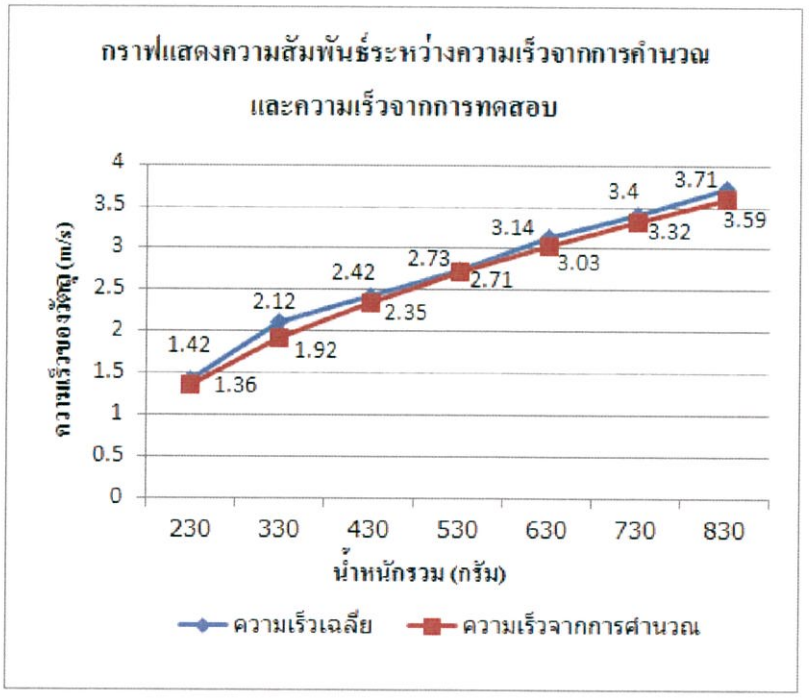
จากรูปที่ 4.29 แสดงกราฟความสัมพันธ์เปรียบเทียบความเร็วที่ได้จากการคำนวณทางทฤษฎีกับความเร็วที่ได้จากการทดสอบที่น้ำหนักต่างๆพบว่า ถ้ามวลยิ่งเพิ่มมากขึ้น ความเร็วของ CANSAT จะยิ่งมีความเร็วเพิ่มขึ้นด้วย และเปรียบเทียบเส้นกราฟระหว่างความเร็วจากการคำนวณกับความเร็วเฉลี่ยที่ได้จากการทดลอง พบว่าความเร็วเฉลี่ยที่ได้จากการทดลองมีค่าสูงกว่าค่าจากการคำนวณซึ่งมีค่าความคลาดเคลื่อนเท่ากับ 9.74 เปอร์เซ็นต์ อันเนื่องมาจากความเร็วลม ทิศทางลม วัสดุต่างๆ ชนิดของสปริงและปัจจัยอื่นๆที่ส่งผลต่อความเร็ว

4.4.2 ผลการทดสอบของร่มชูชีพ (Parachute)

โดยจากการทดสอบปล่อยถุงทรายที่หนัก 100 กรัม จนถึง 700 กรัม ไปกับร่มชูชีพที่ทำการทดสอบเพื่อหาความเร็วที่ได้ แล้วนำไปเปรียบเทียบกับค่าที่คำนวณตอนทำการออกแบบ จะได้ผลการทดสอบดังตารางที่ 4.11

ตารางที่ 4.11 ค่าเปรียบเทียบความเร็วที่ได้จากการคำนวณทางทฤษฎีกับความเร็วที่ได้จากการทดสอบที่น้ำหนักต่างๆ

น้ำหนักของร่มชูชีพ ภาชนะ (30 กรัม) และทราย (กรัม)	ความเร็วของวัตถุ (เมตรต่อวินาที)				
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	คำนวณ
230	1.36	1.42	1.47	1.42	1.36
330	2.12	2.24	2.01	2.12	1.92
430	2.35	2.46	2.46	2.42	2.35
530	2.63	2.84	2.72	2.73	2.71
563	3.13	3.12	3.17	3.14	3.03
730	3.37	3.48	3.35	3.40	3.32
830	3.70	3.76	3.66	3.71	3.59



รูปที่ 4.30 กราฟความสัมพันธ์เปรียบเทียบความเร็วที่ได้จากการคำนวณทางทฤษฎีกับความเร็วที่ได้จากการทดสอบที่น้ำหนักต่างๆ

จากรูปที่ 4.30 แสดงกราฟความสัมพันธ์เปรียบเทียบความเร็วที่ได้จากการคำนวณทางทฤษฎีกับความเร็วที่ได้จากการทดสอบที่น้ำหนักต่างๆพบว่า ถ้ามวลยิ่งเพิ่มมากขึ้น ความเร็วของ CANSAT จะยิ่งมีความเร็วเพิ่มขึ้นด้วย และเปรียบเทียบเส้นกราฟระหว่างความเร็วจากการคำนวณกับความเร็วเฉลี่ยที่ได้จากการทดลอง พบว่าความเร็วเฉลี่ยที่ได้จากการทดลองมีค่าสูงกว่าค่าจากการคำนวณซึ่งมีค่าความคลาดเคลื่อนเท่ากับ 3.97 เปอร์เซ็นต์ อันเนื่องมาจากความเร็วลม ทิศทางลม และปัจจัยอื่นๆที่ส่งผลต่อความเร็ว

4.5 ผลการตรวจสอบความถูกต้องและความพร้อมของกลไกการทำงานตามภารกิจการแข่งขัน

ภารกิจการแข่งขัน CanSat ประจำปี 2018 ได้มีข้อกำหนดพื้นฐานสำหรับการแข่งขัน และมีการดำเนินการสร้างแบบจำลองดาวเทียมขนาดเล็กได้สำเร็จตามเป้าหมายไว้ดังตารางที่ 4.12

ตารางที่ 4.12 การดำเนินงานตามเป้าหมายของการแข่งขัน

ข้อกำหนดของการแข่งขัน	บรรลุเป้าหมาย
1. Probe จะต้องไม่คะมำหรือคว่ำในส่วนใดส่วนหนึ่งของการร่อนลง	✓
2. Probe และ aero-braking heat shield จะบรรจุลงกระป๋องทรงกระบอก ที่มีเส้นผ่าศูนย์กลาง 125 มิลลิเมตร และสูงไม่เกิน 310 มิลลิเมตร ได้พอดี	✓
3. Probe จะต้องบรรจุไข่ไก่ขนาดใหญ่และปกป้องมันจากการถูกทำลายตั้งแต่ปล่อยจนกระทั่งร่อนถึงพื้นดิน	✓
4. Aero-braking heat shield จะต้องไม่มีขอบแหลมที่จะทำให้มันติดค้างในส่วนของจรวด	✓
5. องค์ประกอบทางอิเล็กทรอนิกส์ทั้งหมดจะถูกปิดล้อมและมีที่ปิดกั้นจากสิ่งแวดล้อมกับการยกเว้นของเซนเซอร์บางตัว	✓
6. อุปกรณ์อิเล็กทรอนิกส์ทั้งหมดจะต้องติดตั้งอย่างมั่นคง เช่น ยึดด้วยสกรูหรือกาวที่มีประสิทธิภาพสูง	✓
7. กลไกทั้งหมดจะต้องสามารถทำงานได้ภายใต้แรงกระทำทั้งหมดได้	✓
8. กลไกต้องไม่ใช้วัตถุที่ทำให้เกิดการระเบิด ติดไฟ หรือสารเคมี	✓

ข้อกำหนดของการแข่งขัน	บรรลุเป้าหมาย
9. กลไกที่ใช้ความร้อน (เช่นลวดนิโครม) ไม่ควรสัมผัสกับสภาพแวดล้อมภายนอก เพื่อลดความเสี่ยงที่อาจเกิดขึ้นจากไฟ	✓
10. ระหว่างการร่อนลง probe จะเก็บความดันอากาศ ,อุณหภูมิจากภายนอก, ตำแหน่ง GPS และระดับแรงดันของแบตเตอรี่ต่อหนึ่งวินาที และแท็กข้อมูลเวลาด้วย mission time	✓
11. ระหว่างการร่อนลง probe จะส่ง telemetry ทั้งหมดได้อย่างต่อเนื่อง	✓
12. ข้อมูลจะต้องส่งไปยังภาครับด้วยอัตรา 1 วินาทีต่อ 1 ชุด	✓
13. XBEE จะถูกใช้สำหรับการส่ง telemetry ที่ความถี่ 2.4 GHz Series 1 และ 2 หรือที่ความถี่ 900 MHz	✓
14. ต้นทุนของ CanSat จะต้องไม่เกิน 1,000 เหรียญ โดยไม่รวมภาคพื้นดิน	✓
15. ทั้ง heat shield และ probe จะต้องมีป้ายติดที่ประกอบด้วยข้อมูลการติดต่อของทีม นั่นคืออีเมลล์	✓
16. ไม่อนุญาตให้ใช้เลเซอร์	✓
17. Probe จะต้องประกอบด้วยสวิตช์ไฟที่สามารถเข้าถึงได้ง่าย	✓
18. Probe จะต้องประกอบด้วยไฟแสดงสถานะ เช่น ไฟ LED หรือเครื่องส่งสัญญาณเสียง	✓
19. อัตราเร็วในการร่อนลงของ probe และ aero-braking heat shield 10-30 เมตรต่อวินาที	✓
20. อัตราเร็วในการร่อนลงของ probe ที่ปล่อย aero-braking heat shield และร่มชูชีพ 5 เมตรต่อวินาที	✓
21. ต้องมี audio beacon ส่งเสียงร้องหลังจากร่อนลงสู่พื้นดิน	✓
22. CanSat ต้องใช้แบตเตอรี่อัลคาไลน์ , Ni-Cad , Ni-MH หรือลิเธียม	✓
23. ต้องใช้เซ็นเซอร์วัดความเอียงเพื่อยืนยันความเสถียรของ probe ระหว่างการร่อนลงของ CanSat	✓

จากการทำการรวมระบบทั้งหมดและทำการทดสอบของดาวเทียมกระป๋องขนาดเล็กแล้ว ณ สถานีเรดาร์ฝนหลวงตาคลี อ.ตาคลี จ.นครสวรรค์ ซึ่งได้ผลลัพธ์ตามที่คาดการณ์ไว้ที่สามารถรับส่งข้อมูลได้ถึงระยะความสูง 1000 เมตร และทำการทดสอบทั้งหมดในการทำงานจริงในสภาพแวดล้อมจริง แสดงดังรูปที่ 4.31 และตารางแสดงการบันทึกค่าข้อมูล แสดงดังรูปที่ 4.32

รูปที่ 4.31 การเตรียมอุปกรณ์และการทดสอบอุปกรณ์ ณ ตาคลี จ.นครสวรรค์

	A	B	C	D	E	F	G	H	I	J	K
1	Team ID	Packet cou	Altitude	Pressure	Temperatu	Voltage	GPS time	GPS Lat	GPS Lon	Speed	Status
2	3806	88	196.48	965.39	21.2	7.96	0:45:06	15.25274	100.3286	51.41	Launch
3	3806	89	196.05	965.44	21.3	8.01	0:45:07	15.25274	100.3284	52.74	Descent
4	3806	90	195.44	965.51	21.3	7.96	0:45:08	15.25275	100.3283	54.6	Descent
5	3806	91	195.27	965.53	21.3	8.01	0:45:09	15.25275	100.3282	50.86	Descent
6	3806	92	195.44	965.51	21.3	7.98	0:45:10	15.25275	100.328	51.17	Launch
7	3806	93	195.27	965.53	21.3	8.03	0:45:11	15.25276	100.3279	55.02	Descent
8	3806	94	195.09	965.55	21.3	7.96	0:45:12	15.25277	100.3278	54.39	Descent
9	3806	95	195.01	965.56	21.4	8.01	0:45:13	15.25277	100.3276	49.82	Descent
10	3806	96	195.27	965.53	21.3	7.96	0:45:14	15.25277	100.3275	50.99	Launch
11	3806	97	195.53	965.5	21.4	8.01	0:45:15	15.25277	100.3274	54.99	Launch
12	3806	98	195.09	965.55	21.4	7.96	0:45:16	15.25278	100.3272	53.04	Descent
13	3806	99	196.39	965.4	21.3	8.01	0:45:17	15.25278	100.3271	100	195.53
14	3806	101	196.22	965.42	21.3	8.01	0:45:19	15.25278	100.3268	54.69	Launch
15	3806	102	195.87	965.46	21.4	7.98	0:45:20	15.25278	100.3267	50.97	Descent
16	3806	103	195.27	965.53	21.4	8.01	0:45:21	15.25278	100.3265	50.32	Descent
17	3806	104	198.47	965.16	21.3	7.96	0:45:22	15.25278	100.3264	53.95	Launch
18	3806	105	196.57	965.38	21.3	8.01	0:45:23	15.25278	100.3263	52.21	Descent
19	3806	106	196.91	965.34	21.3	7.96	0:45:24	15.25278	100.3261	47.54	Launch
20	3806	107	197.69	965.25	21.3	8.01	0:45:25	15.25278	100.326	51.74	Launch
21	3806	108	197.69	965.25	21.3	7.96	0:45:26	15.25278	100.3259	57.21	Launch
22	3806	109	198.91	965.11	21.2	7.98	0:45:27	15.25278	100.3257	52.74	Launch
23	3806	110	198.21	965.19	21.3	7.96	0:45:28	15.25279	100.3256	49.8	Descent
24	3806	111	200.38	964.94	21.3	8.01	0:45:29	15.25279	100.3255	54.5	Launch

รูปที่ 4.32 ข้อมูลการบันทึกค่าจากการเก็บค่าสภาพอากาศ

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

จากปริญญานิพนธ์เรื่องการออกแบบและสร้างดาวเทียมกระป๋องสำหรับใช้ในการแข่งขัน (1) : ภาคส่ง ซึ่งได้ทำการศึกษาองค์ประกอบและหลักการทำงานของ CanSat เพื่อนำมาประยุกต์ใช้ในการสร้าง CanSat เพื่อใช้ในการแข่งขัน CanSat Competition ประจำปี 2018 สามารถสรุปผลการดำเนินงาน ได้ดังนี้

1) สามารถสร้างอุปกรณ์อิเล็กทรอนิกส์ ของ Sensor Subsystem ทั้งหมดเพื่อวัดเก็บค่าสภาพอากาศ ซึ่งได้แก่ อุณหภูมิ ความดัน และความสูงจากระดับน้ำทะเลกลาง , ค่าจาก GPS ที่ระบุถึงตำแหน่ง ซึ่งได้แก่ เวลา ละติจูด ลองจิจูด ความสูง และเลขดาวเทียม ,ค่าจากความเร่ง 9 แกน ได้แก่ accelerometers, gyroscope และ magnetometer, ค่าระดับแรงดันจากแบตเตอรี่ และค่าการนับเวลาภารกิจเป็นวินาทีโดยใช้ Real time clock และเก็บวิดีโอจากโมดูลกล้อง

2) สามารถเก็บค่าอุณหภูมิ, ความดัน, GPS, accelerometers, gyroscope, magnetometer, ค่าระดับแรงดันจากแบตเตอรี่ และค่าการนับเวลาทีละวินาทีจากโมดูลเซนเซอร์ต่างๆ โดยทำการประมวลผลด้วย Arduino pro mini และ Raspberry pi zero ทำการจัดเรียงฟอร์มแมต Telemetry สำหรับการส่งข้อมูลตามภารกิจการแข่งขัน ได้ดังนี้

<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 >,< HEAD >**

- ข้อมูลในการส่ง(บังคับ) ตามภารกิจการแข่งขัน CanSat 2018
- ข้อมูลที่เพิ่มเติมเข้ามานอกเหนือจากข้อบังคับการแข่งขัน

ตัวอย่างรูปแบบการส่งข้อมูลทั้งหมดให้ภาครับ

<TEAM ID>,<1>,<1>,<1.20>,<100985.00>,<28.65>,<5.20>,<11/1/2017>,<13.7275321>, <100.7759516>,<14.20>,<4>,<2>,<-580>,<-92>,<17020>,<[recorded]>,**<3.20>,<59>, <-37>,<7>,<226>**

3) สามารถส่งข้อมูลไปยังส่วนของภาครับ (ภาคพื้นดิน) โดยใช้โมดูล XBEE PRO S1 และเก็บข้อมูลลง SD Card on board pi zero ใน CanSat ได้

4) สามารถสร้างสายอากาศ Microstrip dipole 2.45 GHz เพื่อใช้กับโมดูล XBEE PRO S1 เพื่อให้สามารถส่งข้อมูลได้ไกลมากขึ้น

5) สามารถสร้างร่มชูชีพที่ใช้กู้ CanSat และลดความเร็วในการร่อนลง

6) สามารถสร้างเพย์โหลดและโครงสร้างอิเล็กทรอนิกส์ทั้งหมดได้

7) สร้าง Aero-braking heat shield เพื่อลดความเร็ว

8) พัฒนาโครงสร้างของ CanSat เพื่อความเหมาะสมในการใช้งาน

9) สร้างเพย์โหลดตามภารกิจการแข่งขัน

10) ทดสอบเก็บค่าข้อมูลที่ระดับความสูงต่างๆโดยใช้บอลลูนและส่งมายังภาครับ

11) ทำการสร้างกลไกการปลดร่มชูชีพอัตโนมัติและกลไกการปล่อย Heat Shield

12) ทำการทดสอบความเร็วของการร่อนลงของ CanSat

ตารางที่ 5.1 แผนงานโดยรวมในการสร้าง CanSat เพื่อใช้ในการแข่งขัน

รายการ	**ทดสอบ**
GY-BMEP280	✓
GPS (GY-NEO6MV2)	✓
GY-9250	✓
Tiny RTC DS1307	✓
Voltage Divider circuit	✓
Pi Camera rev 1.3	✓
Arduino pro mini	✓
Raspberry Pi Zero	✓
XBEE PRO Series 1	✓
LM7805	✓
Audio Beacon	✓
Micro SD card on board pi zero	✓
วงจรรวม	✓

รายการ	ทดสอบ
Servo motor	✓
โครงสร้างและเพย์โหลดของ CanSat	✓
ทดสอบ Droptest	✓

5.2 ข้อเสนอแนะ

- เพย์โหลดควรมีความแข็งแรงทนต่อแรงกระทำ และน้ำหนักเบากว่านี้
- ควรเก็บข้อมูลที่ระดับความสูงที่ใกล้เคียงกับการแข่งขันในแนวดิ่ง และทดสอบการส่งในพื้นที่ที่ไม่มีสัญญาณรบกวน

บรรณานุกรม

[1] Beng Parilat. "Cansat คืออะไร."
https://www.nstda.or.th/jaxa-thailand/what-is-cansat/.

[2] esa. "WHAT IS A CANSAT."
http://www.esa.int/Education/CanSat/What_is_a_CanSat.

[3] CanSat Competition. "Introduction."
http://www.cansatcompetition.com/Index.html.

[4] CanSat Competition. "Mission."
http://cansatcompetition.com/mission.html.

[5] CanSat Competition. "Mission Guide."
http://cansatcompetition.com/docs/mission_guide_2018_r20171020_r1.4.pdf.

[6] Bosch Sensortec. "BME280 Combined humidity and pressure sensor."
https://cdn-shop.adafruit.com/datasheets/BST-BME280_DS001-10.pdf.

[7] Electrotec. "Tutorial Arduino #7 - Uso del modulo GPS6MV2."
http://electrotec.pe/blog/GPS.

[8] Global5thailand. "ความรู้ทั่วไปเกี่ยวกับ GPS."
http://www.global5thailand.com/thai/gps.htm.

[9] Wichet Darakai. "GPS : โปรโตคอล NMEA ของ GPS."
http://darakai.blogspot.com/2011/11/nmea-gps.html.

[10] Arduitronics. "Iteadstudio GPS Shield."
https://www.arduitronics.com/article/40/iteadstudio-gps-shield-part-1.

[11] Dale DePriest. "NMEA data."
http://www.gpsinformation.org/dale/nmea.htm.

[12] Henry's Bench. "Arduino 25V Voltage Sensor Module User Manual."
http://henrysbench.capnfatz.com/henrys-bench/arduino-voltage-measurements/arduino-25v-voltage-sensor-module-user-manual/.

[13] AliExpress. "GY-9250 MPU 9250 MPU-9250ทัศนคติ+ Gyro +เร่ง+ Magnetometer เซนเซอร์โมดูลMPU9250." https://th.aliexpress.com/store/product/1-pcs-SPI-IIC-GY-9250-MPU-9250-MPU-9250-9-Axis-Attitude-Gyro-Accelerator-Magnetometer/1089303_32789580858.html.

[14] Pi Supply. "Raspberry Pi Camera Board v1.3 (5MP, 1080p)." https://www.pi-supply.com/product/raspberry-pi-camera-board-v1-3-5mp-1080p/.

[15] AliExpress. "โปรมินิ ATmega168 3.3โวลต์ (8เมกะเฮิร์ตซ์)." https://th.aliexpress.com/store/product/Pro-Mini-ATmega168-3-3V-8MHz-Compatible-for-Arduino-ProMini/1950989_32579105597.html.

[16] ThaiEasyElec. "Raspberry Pi ." http://www.thaieasyelec.com/article-wiki/embedded-electronics-application/.

[17] K Lock Pro. "มารู้จักคอมพิวเตอร์จิ๋ว Raspberry Pi." http://www.vcharkarn.com/maker/501923.

[18] RASPBERRY PI FOUNDATION. "RASPBERRY PI ZERO." https://www.raspberrypi.org/products/raspberry-pi-zero/.

[19] Arduino All. "Tiny RTC I2C modules 24C32 memory DS1307 clock for arduino พร้อมแบตเตอรี่."https://www.arduinoall.com/product/158/tiny-rtc-i2cmodules24c32memoryds1307-clock-for-arduino.

[20] CW Freak. "HIGH-FREQUENCY DIPOLE." http://www.hs8jyx.com/html/dipole.html.

[21] บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์. "XBee-PRO โมดูลสื่อสารข้อมูลไร้สาย 2.4 GHz." http://www.es.co.th/Schemetic/PDF/XBEE-PRO.PDF.

[22] Seeed Studio. "XBee Pro Chip Antenna - S1 (802.15.4)." https://www.seeedstudio.com/XBee-Pro-Chip-Antenna-S1%28802.15.4%29-p-1111.html.

[23] Thaimicrotron. "การเชื่อมต่ออุปกรณ์แบบ I2C." http://www.thaimicrotron.com/CCS-628/Referrence/I2CBUS.htm.

[24] Arduitronics. "I2C Communication: Case study of GY-30." https://www.arduitronics.com/article/47/i2c-communication-case study-of-gy-30-ambient-light-sensor.

[25] Chaiyapron. "การใช้งานพอร์ตสื่อสาร UART." http://aimagin.com/blog/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%9E%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%95%E0%B8%AA%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B8%AA%E0%B8%B2%E0%B8%A3-uart/?lang=th.

[26] LinkedIn Learning. "การสื่อสารแบบ Analog." https://www.slideshare.net/ChainarongMaharak/analog-69579920.

[27] NASA. "NASA Tests Foldable Heat Shields For Future Martian Spacecrafts." http://wonderfulengineering.com/nasa-tests-foldable-heat-shields-for-future-martian-spacecrafts/.

[28] ESA. "EXOMARS 2016 SCHIAPARELLI DESCENT SEQUENCE." http://www.esa.int/spaceinimages/Images/2016/02/ExoMars_2016_Schiaparelli_descent_sequence.

[29] John London. "Types of Parachutes." http://ourpastimes.com/types parachutes 8231828.html.

[30] Randy Culp. "Parachute Descent Calculations." http://www.rocketmime.com/rockets/descent.html#Velocity.

[31] Dr.-Ing S.F. HOERNER. "Fluid-dynamic drag" http://dl.kashti.ir/ENBOOKS/NEW/FDD.pdf.

ภาคผนวก ก

โค้ดสำหรับการทำงานของ Sensor Subsystem

Code Arduino

```
/*                                          //การเชื่อมของ Microcontrolle และ Sensor.
  PIN 2 BUZZER
  PIN 11 LED
  PIN 12 Servo 1 >> PI ( GPIO 27 )
  PIN 13 Servo 2 >> PI ( GPIO 22 )
  PIN 5 Switch Record VDO >> PI ( GPIO 4 )
  PIN 6 Reset Packet >> PI ( GPIO  17 )
  PIN A0 Voltage Sensor
  */                                        //ส่วนของการเรียกใช้ฟังค์ชัน Library
#include "Wire.h"
#include "Seeed_BME280.h"
#include "TinyGPS++.h"
#include "SoftwareSerial.h"
#include "MPU9250.h"
/************BME 280 ***********/
    BME280 bme280;                          //เรียกใช้งาน Library BME280
    float P0,Hoffset;                       //ตั้งตัวแปร P0 คือ ความดันสัมพัทธ์
                                            //Hoffset คือ ค่าเริ่มต้นของ H
/************GPS***************/
    static const uint32_t GPSBaud = 9600;   //กำหนด Baud rate ของ GPS
    TinyGPSPlus gps;
    SoftwareSerial ss(8, 9);                //กำหนด ขา input และ output
/************MPU9250***************/
    MPU9250 accelgyro(0x69);                //กำหนด I2C Address ของMPU9250
    int16_t ax, ay, az;                     //กำหนดตัวแปนเก็บค่า ความเร่ง 3 แกน
    int16_t gx, gy, gz;                     //กำหนดตัวแปนเก็บค่า ไจโรสโคป 3 แกน
    int16_t mx, my, mz;                     //กำหนดตัวแปนเก็บค่า สนามแม่เหล็ก 3 แกน
/**********RTC SENSOR***************/
    #define DS3231_I2C_ADDRESS 0x68         //กำหนด I2C Address ของ DS3231
    int c =0;

                                            // ฟังค์ชันเปลี่ยนเลขฐานสิบเป็นเลขBCD
    byte decToBcd(byte val)
    {
      return( (val/10*16) + (val%10) );
    }
```

```
                                                    // ฟังค์ชันเปลี่ยนเลขBCDเป็นเลขฐานสิบ
    byte bcdToDec(byte val)
    {
      return( (val/16*10) + (val%16) );
    }
/************Voltage Senser***************/
    float R1 = 300000.00;                          //กำหนดค่า R1
    float R2 = 75000.00;                           //กำหนดค่า R2
/************Soft State***************/          //สถานะการทำงานของCANSAT
    String Status[] = {"Pre-Launch","Launch","Descent","Separate","Landed"};
    String ID = "TEAM ID";
    int   No = 0, numstate, numbonus, down , pao;
    String bonus[]= {"[Non record]","[Recording]","[Recorded]"};        //สถานะการทำงาน
ของ VDO
    const int buttonPin = 6;                       //PIN Reset
    int buttonState = 0;                           //กำหนดเป็นสถานะ LOW

void setup()
{
  Wire.begin();
  Serial.begin(4800);                              //กำหนด Baud Rate 4800
  ss.begin(GPSBaud);
  setDS3231time(0);                                //ตั้งค่า นับ RTC ให้เริ่มต้นที่ 0
  bme280.init();                                   //เร่มเก็บข้อมูลจาก BME280
  accelgyro.initialize();                          //เร่มเก็บข้อมูลจาก MPU9250
  Hoffset =  bme280.calcAltitude(bme280.getPressure());          //เก็บความสูงค่แรก
  pinMode(buttonPin, INPUT);                       //ตั้ง PIN 6 รับ input จาก pi เพื่อ reset
   pinMode(11, OUTPUT);                            //ตั้ง PIN output LED
   pinMode(12, OUTPUT);                            //ตั้ง PIN output servo 1  >> GPIO 27
   pinMode(13, OUTPUT);                            //ตั้ง PIN output servo 1  >> GPIO 22
   pinMode(5, OUTPUT);                             //ตั้ง PIN output Record Video
   pinMode(2, OUTPUT);                             //buzzer
  if(Hoffset>2000)
    {
      delay(1000);
      Hoffset =  bme280.calcAltitude(bme280.getPressure());
```

```
    }
}

void loop()
{                                                   //การทำงานของนับวินาทีจาก RTC
    byte second;
    readDS3231time(&second);                        //รับข้อมูลวินาที
    int Ta= second ;
    if (c>=1||Ta==59)                               //การทดให้นับต่อไปเรื่อยๆ
    {
      Ta=60+c-1;
      c++;
      delay(180);                                           //ดีเลย์ในการรัน
    }
    int Vout = analogRead(A0);                      // รับค่าจากขาอนาล็อก
    double Vin =((5.00*Vout)/(1024.00))/(R2/(R1+R2));   //ใช้หลักการ Voltage divider
/************BME***************/
    float P =  bme280.getPressure();                //เก็บค่าความดัน
    float H =  bme280.calcAltitude(P)-Hoffset; //เก็บค่าความสูงจากความดัน
    float T =  bme280.getTemperature();             //เก็บค่าอุณหภูมิ
/***************** GPS ******************/         //เก็บค่าจาก GY-NEO6MV2
    int   sat = gps.satellites.value();
    int   mm = gps.date.month();
    int   dd = gps.date.day();
    int   yy = gps.date.year();
    int   hh = gps.time.hour();
    int   mi = gps.time.minute();
    int   sec = gps.time.second();
    float LAT = gps.location.lat();
    float LONG = gps.location.lng();
    float ALTI = gps.altitude.meters();
    float SPEED = gps.speed.kmph();
    smartDelay(550);                                //ดีเลย์สำหรับเข้ารหัส Library GPS Sensor

// ส่วนของการคำนวนหาทิศทางของ CANSAT
    accelgyro.getMotion9(&ax, &ay, &az, &gx, &gy, &gz, &mx, &my, &mz);
```

```
  float heading = atan2(my, mx);
  if(heading < 0)
     heading += (2 * M_PI);
  double vector = (heading* 180/M_PI)-30;
  if(vector<0)
     vector = 360+vector;

//การกำหนดขั้นตอนการงานในแต่ละขั้นตอนของสถานะ CANSAT
  if(No<700)                          //ที่ความสูงน้อยกว่า 700 เมตร
   { numstate = 0;                    //สถานะ CANSAT
     numbonus = 0;                    //สถานะ กล้อง
     //digitalWrite(2, HIGH);         //กาง Heat Shield
   }
  if(No>=300&&No<750)                 //ที่ความสูง300-700 เมตร ขาลง และขาขึ้น
   { numstate = 1;                    //สถานะ CANSAT
     if(No>=700)
      {
      down = 1;                       //กำหนดแสดงสถานะการตก
      }
   }
  if(No>=300&&No<700&&down==1)        //ที่ความสูง300-700 เมตร ขาลง
   { numstate = 2;                    //สถานะ CANSAT
     digitalWrite(12, HIGH);          //ปล่อย Heat Shield
   }
  if(No>=5&&No<300&&down==1)                  //ที่ความสูง5-300 เมตร ขาลง
   { numstate = 3;                    //สถานะ CANSAT
     numbonus = 1;                    //สถานะ กล้อง
     digitalWrite(5, HIGH);           //เริ่มอัด VDO
    digitalWrite(13, HIGH);           //ปล่อยลมชูชีพ
   }
  if(No<=5&&down==1)                  //ที่ความสูง1-5 เมตร ขาลง
   {
   numstate = 4;                      //สถานะ CANSAT
   numbonus = 2;                      //สถานะกล้อง
   digitalWrite(2, (No%2));           //BUZZER ทำงาน
   digitalWrite(5, LOW);              //หยุดถ่าย VDO
```

```
    }

    String S2 = String(hh)+":"+String(mi)+":"+String(sec);          //เก็บเวลาจาก GPS
    //if(No==Ta&&No>=Ta)
    {                                                    //รูปแบบการเรียงข้อมูล
String S = "<"+String(ID)+">,<"+ String(Ta)+">,<"+String(No)+">,<"+ String(H)+">,<"+
String(P)+">,<"+
String(T)+">,<"+String(Vin)+">,<"+String(S2)+">,<"+String(LAT,6)+">,<"+String(LONG,6)+">
,<"+String(ALTI)+">,<"+String(sat)+">,<"+String(ax)+">,<"+String(ay)+">,<"+String(az)+">,<
"+String(Status[numstate])+">,<"+String(bonus[numbonus])+">,<"+String(SPEED)+">,<"+
String(gx)+">,<"+String(gy)+">,<"+ String(gz)+">,<"+ String(vector)+">";
        Serial.println(S);                               //แสดงผลข้อมูล
        digitalWrite(11, (No%2));                        //LEDแสดงสถานะการทำงาน
        No = No+1;
        delay(200);
    }
buttonState = digitalRead(buttonPin);                    //ตั้งเงื่อนไข Reset เมื่อรับค่า pi
if (buttonState == HIGH)
  {
delay(1000);
No = 0;
    setDS3231time(0);
  }
}
static void smartDelay(unsigned long ms)                 //ฟังค์ชันเข้ารหัสดีเลย์ GPS
 {
   unsigned long start = millis();
   do
   {
     while (ss.available())
         gps.encode(ss.read());
   } while (millis() - start < ms);
 }
void setDS3231time(byte second)                          //ฟังค์ชั่นเรียกค่า RTC
```

```
{                                                    //ตั้งเวลา RTC
   Wire.beginTransmission(DS3231_I2C_ADDRESS);
   Wire.write(0);// set next input to start at the seconds register
   Wire.write(decToBcd(second));// set seconds
   Wire.endTransmission();
}
void readDS3231time(byte *second)
{
   Wire.beginTransmission(DS3231_I2C_ADDRESS);
   Wire.write(0); // set DS3231 register pointer to 00h
   Wire.endTransmission();
   Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
   // request seven bytes of data from DS3231 starting from register 00h
   *second = bcdToDec(Wire.read() & 0x7f);
}
```

Code python

```
#python code
#GPIO 4 Input for recording VDO.
#GPIO 17 Output for resetting Arduino
#Arduino PIN 11 servo1 = 22 Input
#Arduino PIN 12 servo2 = 23 Input
#servo1 = 24 Output
#servo2 = 25 Output
#Rx 1K ohm
import serial                          #เรียกใช้ Library serial
import picamera                        #เรียกใช้ Library picamera
from time import sleep                 #เรียกใช้ Library time
from subprocess import call            #เรียกใช้ Library subprocess
import RPi.GPIO as GPIO                #เรียกใช้ Library GPIO

#ตั้งค่า GPIO
GPIO.setmode(GPIO.BCM)        # กำหนดขา GPIO
button = 4
reset = 17
```

```
Inservo1 = 22    #Input
Inservo2 = 23    #Input
Outservo1 = 24 #Output
Outservo2 = 25 #Output

GPIO.setup(button, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)    #ตั้งค่า button เป็น
INPUT (GPIO4)
GPIO.setup(reset, GPIO.OUT)                                #ตั้งค่าresetเป็น OUTPUT(GPIO17)
GPIO.setup(Inservo1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) #ตั้งค่าInservo1เป็น
INPUT(GPIO22)
GPIO.setup(Inservo2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) #ตั้งค่าInservo2เป็น
INPUT(GPIO23)
GPIO.setup(Outservo1, GPIO.OUT)                    #ตั้งค่าOutservo1เป็น INPUT(GPIO24)
GPIO.setup(Outservo2, GPIO.OUT)                    #ตั้งค่าOutservo2เป็น INPUT(GPIO25)

b=1
c=1                                                #สำหรับดูสถานะของ servo
name = "/home/pi/Desktop/cookie/"+"3"+ ".txt"      #สำหรับกำหนดชื่อไฟล์
count = 1
old =1

ser = serial.Serial('/dev/ttyAMA0', 4800,timeout=0.8) #กำหนดการเชื่อมต่ากับ Aduino
#เริ่ม Reset Arduino
GPIO.output(reset, 0)
GPIO.output(reset, 1)
sleep(1.5)
GPIO.output(reset, 0)
GPIO.output(Outservo1,0)
GPIO.output(Outservo2,0)

while True:                                        #กำหนดลูป
   packet = ser.readline()                         #รับข้อมูลเก็บในตัวแปร packet
   print (packet)                                  #แสดงข้อมูลในตัวแปร packet
   count=count+1
   with open( name, "a") as text_file:             #บันทึกไฟล์
            text_file.write("%s\n" % (packet))
```

```
c1 = GPIO.input(Inservo1)
if c1 and c==1:                          #เงื่อนไขการทำงาน Servo1
   for i in range(1,70):
      GPIO.output(Outservo1,1)
      GPIO.output(Outservo1,0)
   c=2
GPIO.output(Outservo1,0)
c2 = GPIO.input(Inservo2)
if c2 and c==2:                          #เงื่อนไขการทำงาน Servo2
   for i in range(1,70):
      GPIO.output(Outservo2,1)
      GPIO.output(Outservo2,0)
   c=3
GPIO.output(Outservo2,0)
a = GPIO.input(button)
if a and b==1:                           #เงื่อนไขการถ่าย VDO
   print "Start record video."
   command = "rm /home/pi/Desktop/cookie/testVideo.mp4"   #ลบไฟล์ VDO เก่า
   #Execute our command
   call([command], shell=True)
   with picamera.PiCamera() as camera:              #ตั้งค่า Pi Camera
      camera.resolution = (640,480)                 #กำหนด resolution
      camera.framerate = 30                         #กำหนด framerate
                                                    # เริ่มถ่าย VDO
      camera.start_recording("/home/pi/Desktop/cookie/Video.h264")

      while (a):                                    #การเก็บข้อมูลขณะอัด VDO
            packet = ser.readline()                 #รับข้อมูลเก็บในตัวแปร packet
            print (packet)
     print("recording")
            count=count+1
            with open( name, "a") as text_file:
                  text_file.write("%s\n" % (packet))    #บันทึกข้อมูลลงไฟล์
            sleep(0.8)
     a = GPIO.input(button)                         #ใช้ a เป็นตัวกำหนดเงื่อนไข
      print("recorded")
```

```
            camera.stop_recording()                    #หยุดถ่าย VDO
            command = "MP4Box -add /home/pi/Desktop/cookie/Video.h264 -fps 30
/home/pi/Desktop/cookie/testVideo.mp4"                 #แปลงไฟล์เป็น .MP4
            call([command], shell=True)
            print("recorded")
        print(c)                                       #สถานะ Servo
        sleep(1)
```