



การพัฒนาอุปกรณ์สวมใส่เพื่อเก็บข้อมูลการเคลื่อนไหวของวัว
โดยใช้เทคโนโลยีไอโอที

Development of Wearable Device for Logging Cow Behavior
Using IoT Technology

จิรศักดิ์ เอี่ยมสะอาด
อภิวัฒน์ จันทวงษ์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร
พ.ศ. 2561

การพัฒนาอุปกรณ์สวมใส่เพื่อเก็บข้อมูลการเคลื่อนไหวของวัว
โดยใช้เทคโนโลยีไอโอที

จิรศักดิ์ เอี่ยมสะอาด
อภิวัฒน์ จันทวงษ์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
แขนงวิชาวิศวกรรมคอมพิวเตอร์
คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร
พ.ศ. 2561

Development of Wearable Device for Logging Cow Behavior Using IoT Technology

Jirasak Iamsa-ard
Aphiwat Jantawong



The Project report Submitted in Partial Fulfillment of
The Requirement for the Degree of Bachelor of Science in Industrial Education
Department of Computer Engineering
Faculty of Industrial Education
Rajamangala University of Technology Phra Nakhon

2018

ใบรับรองโครงการ

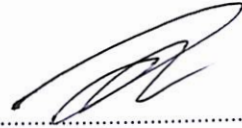
ชื่อโครงการ การพัฒนาอุปกรณ์สวมใส่เพื่อเก็บข้อมูลการเคลื่อนไหวของวัวโดยใช้เทคโนโลยี
 ไอโอที

ชื่อนักศึกษา นายจรัสศักดิ์ เอี่ยมสะอาด รหัสนักศึกษา 036050504026-9
 นายอภิวัฒน์ จันทพงษ์ รหัสนักศึกษา 036050504016-0

แขนงวิชา วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. สุชาดา เกตุดี
 อาจารย์ศรีณีย์ ฉัตรธัญญกิจ

สาขาวิชาวิศวกรรมไฟฟ้า แขนงวิชาวิศวกรรมคอมพิวเตอร์ คณะครุศาสตร์อุตสาหกรรม
มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต



(ดร.มนตรี บุญเรืองเศษ)

หัวหน้าสาขาวิชาวิศวกรรมคอมพิวเตอร์


คณะกรรมการสอบโครงการ


.....
(ผู้ช่วยศาสตราจารย์ ดร.สุชาดา เกตุดี)

ประธานกรรมการ


.....
(ดร.มนตรี บุญเรืองเศษ)

กรรมการ


.....
(ผู้ช่วยศาสตราจารย์ ดร. ว่าที่ ร.ต.พรชัย เตชะชนะเศรษฐ์)

กรรมการ


.....
(อาจารย์ศรีณีย์ ฉัตรธัญญกิจ)

กรรมการ

ชื่อโครงการ	การพัฒนาอุปกรณ์สวมใส่เพื่อเก็บข้อมูลการเคลื่อนไหวของวัวโดยใช้เทคโนโลยีไอโอที		
ชื่อนักศึกษา	นายจรัสศักดิ์ เอี่ยมสะอาด	รหัสนักศึกษา	036050504026-9
	นายอภิวัฒน์ จันทวงษ์	รหัสนักศึกษา	036050504016-0
แขนงวิชา	วิศวกรรมคอมพิวเตอร์		
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. สุชาติา เกตุดี อาจารย์ศรัณย์ ฉัตรธัญญกิจ		

บทคัดย่อ

โครงการนี้นำเสนอการพัฒนาอุปกรณ์สวมใส่เพื่อเก็บข้อมูลการเคลื่อนไหวของวัวโดยใช้เทคโนโลยีไอโอที โดยมีวัตถุประสงค์เพื่อพัฒนาอุปกรณ์เก็บข้อมูลพฤติกรรมการเคลื่อนไหวของโคนม และเพื่อระบุตำแหน่งที่อยู่ปัจจุบันของโคนม โดยโครงการนี้ใช้เซ็นเซอร์วัดความเร่ง เพื่อตรวจสอบพฤติกรรมการเดิน เซ็นเซอร์อัลตราโซนิกเพื่อตรวจสอบพฤติกรรมการนอน และเซ็นเซอร์ GPS เพื่อระบุตำแหน่งของโคนม โดยข้อมูลที่ได้มานั้นจะถูกเก็บและแสดงบนไอโอทีแพลตฟอร์มจากการทดลองพบว่า ระบบที่ได้พัฒนาสามารถแสดงผลพฤติกรรมของโคนมได้ถูกต้อง ซึ่งการทดลองได้ใช้คนแสดงท่าทางเลียนแบบโคนม นอกจากนี้ระบบที่ได้พัฒนาขึ้นสามารถแสดงตำแหน่งปัจจุบันของโคนม และแสดงข้อความเมื่อโคนมออกจากระยะที่กำหนดไว้ ซึ่งได้มีการวางแผนนำระบบที่พัฒนาแล้วไปใช้กับโคนมในสถานการณ์จริง

(มีจำนวนทั้งสิ้น 71 หน้า)

Title Development of Wearable Device for Logging Cow Behavior using IoT Technology

Students Mr.Jirasak Iamsa-ard No. 036050504026-9
Mr.Aphiwat Jantawong No. 036050504016-0

Major Computer Engineering

Advisors Asst. Prof. Dr. Suchada Katedee
Mr. Sarun Chuttunyakit

Abstract

This project presents the development of wearable device for logging cow behavior using IoT technology. This project aims to develop the device that can collect the cow's behavior and to identify the current position of cow. The accelerometer and ultrasonic sensors are used to measure walking behavior and sitting behavior, respectively. The current position of cow is measured by GPS module. The measured data will be shown and stored in IoT platform. According to the experiments, the developed systems can measure the behavior properly with human acting as a cow. Moreover, the developed system can show the current position of the cow and message when the cow is out of area. In the future, it is planned to implement the developed system with the actual cow in real situation.

(Total 71 pages)

กิตติกรรมประกาศ

การจัดทำโครงการฉบับนี้ สำเร็จลุล่วงไปได้ดีด้วยการได้รับคำแนะนำข้อมูลเกี่ยวกับการดำเนินงานจากคณะอาจารย์ที่ปรึกษา ตลอดจนความช่วยเหลือจากท่านอาจารย์ประจำสาขาวิชา ครุศาสตร์อุตสาหกรรม สาขาวิศวกรรมคอมพิวเตอร์ ขอขอบคุณคณะอาจารย์ ที่ให้คำปรึกษาและให้กำลังใจในการทำโครงการนี้ให้สำเร็จลุล่วงไปด้วยดี

โครงการฉบับนี้ได้รับทุนอุดหนุนงบประมาณจากโครงการส่งเสริมสิ่งประดิษฐ์และนวัตกรรม เพื่อคนรุ่นใหม่ ประจำปีงบประมาณ พ.ศ.2562 มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

ท้ายนี้คณะผู้จัดทำโครงการใคร่ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ซึ่งให้การสนับสนุนทางการเงิน และกำลังใจให้แก่ผู้จัดทำเสมอมาจนสำเร็จการศึกษา

คณะผู้จัดทำ



สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินงาน	2
1.5 ระยะเวลาดำเนินงาน	3
1.6 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีที่และงานวิจัยที่เกี่ยวข้อง	4
2.1 พฤติกรรมคอนม	4
2.2 บอร์ดควบคุม	6
2.3 เทคโนโลยีตัวรับรู้	8
2.4 แหล่งพลังงาน	15
2.5 เทคโนโลยี IOT	16
2.6 งานวิจัยที่เกี่ยวข้อง	18
บทที่ 3 ขั้นตอนและวิธีดำเนินการ	20
3.1 แผนการดำเนินโครงการ	20
3.2 ขั้นตอนการดำเนินการ	22
3.3 การพัฒนาระบบ	22

สารบัญ(ต่อ)

เรื่อง	หน้า
บทที่ 4 ผลการดำเนินงาน	33
4.1 การแสดงบนแผนที่	33
4.2 การระบุขอบเขตในการอยู่พื้นที่ของโคนม	34
4.3 การนอน	35
4.4 การเดิน	36
4.5 การยืน	38
บทที่ 5 สรุปผล ปัญหาและข้อเสนอแนะ	40
5.1 สรุปผลโครงการ	40
5.2 ปัญหาและแนวทางแก้ไข	40
5.3 ข้อเสนอแนะในการพัฒนาโครงการ	40
บรรณานุกรม	42
ภาคผนวก ก การติดตั้งโปรแกรม Arduino IDE	43
ภาคผนวก ข การสมัคร NETPIE	48
ภาคผนวก ค คู่มือการใช้งาน	51
ภาคผนวก ง Code Program	57
ประวัติผู้จัดทำ	70

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 แสดงระยะเวลาขั้นตอนการดำเนินโครงการ	3
ตารางที่ ง.1 Code โปรแกรมการทำงาน	58



สารบัญภาพ

ภาพที่	หน้า
ภาพที่ 2.1 ทานอนปกติ	5
ภาพที่ 2.2 ทานอนพักบนหน้าอก	5
ภาพที่ 2.3 โคลุกยกขาหลังขึ้นก่อน	6
ภาพที่ 2.4 ทำยีนปกติ	6
ภาพที่ 2.5 Module WiFi NodeMCU V2	7
ภาพที่ 2.6 ฝั่งของ Module WiFi NodeMCU V2	8
ภาพที่ 2.7 โมดูล ACC/Gyro GY-521 MPU-6050	8
ภาพที่ 2.8 หลักการของ Accelerometer	9
ภาพที่ 2.9 หลักการทำงานของ Gyroscope	9
ภาพที่ 2.10 หลักการทำงานของ I2C	10
ภาพที่ 2.11 หลักการทำงานของ SDA และ SCL ของสัญญาณ I2C	11
ภาพที่ 2.12 GY-GPS6MV2	14
ภาพที่ 2.13 Ultrasonic HC-SR04	14
ภาพที่ 2.14 หลักการทำงานของ Ultrasonic HC-SR04	15
ภาพที่ 2.15 Battery	16
ภาพที่ 2.16 แผงชาร์ตแบตเตอรี่	16
ภาพที่ 2.17 Internet of Thing	17
ภาพที่ 2.18 NETPIE	17
ภาพที่ 2.19 กราฟวงกลมโดยคิดคำนวณเป็นเปอร์เซ็นต์จากค่าเฉลี่ยเวลาที่โคแต่ละตัว แสดงพฤติกรรมต่าง ๆ ในช่วงวัน	19
ภาพที่ 3.1 ฝั่งดำเนินโครงการ	21
ภาพที่ 3.2 ฝั่งการทำงานของอุปกรณ์	23
ภาพที่ 3.3 ฝั่งการทำงานของโปรแกรมโดยรวม	24
ภาพที่ 3.4 ฝั่งการทำงานของ Module GPS	26
ภาพที่ 3.5 ฝั่งการทำงานของ MPU6050	28
ภาพที่ 3.6 ฝั่งการทำงานการคำนวณการเคลื่อนไหวของโคนม	29
ภาพที่ 3.7 ฝั่งการทำงานการคำนวณการเคลื่อนไหวของโคนม(ต่อ1)	30

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 3.8 ผังการทำงานการคำนวณการเคลื่อนไหวของโคนม(ต่อ2)	31
ภาพที่ 3.9 การแสดงผลบน NETPIE29	32
ภาพที่ 4.1 การระบุตำแหน่ง	33
ภาพที่ 4.2 การระบุขอบเขตในการอยู่พื้นที่	34
ภาพที่ 4.3 การระบุขอบการออกนอกขอบเขตในการอยู่พื้นที่	34
ภาพที่ 4.3 การนอน	35
ภาพที่ 4.4 ก่อนการแสดงผลการนอน	35
ภาพที่ 4.5 แสดงผลการนอน	36
ภาพที่ 4.6 การเดิน	36
ภาพที่ 4.7 ก่อนการแสดงผลการเดิน	37
ภาพที่ 4.8 แสดงผลการเดิน	37
ภาพที่ 4.9 การยืน	38
ภาพที่ 4.10 ก่อนการแสดงผลการยืน	38
ภาพที่ 4.11 แสดงผลการยืน	39
ภาพที่ ก.1 เว็บไซต์ดาวน์โหลด Arduino IDE	44
ภาพที่ ก.2 แสดงการกดดาวน์โหลดและเลือกที่บันทึก	44
ภาพที่ ก.3 แสดงการกดปุ่มยืนยันการติดตั้ง	45
ภาพที่ ก.4 แสดงการเลือกโปรแกรมที่จะติดตั้ง	45
ภาพที่ ก.5 แสดงการเลือกไดฟ์ที่จะติดตั้งโปรแกรม	46
ภาพที่ ก.6 แสดงเมื่อติดตั้งเสร็จเรียบร้อย	46
ภาพที่ ก.7 แสดงการตั้งค่าในโปรแกรม (1)	47
ภาพที่ ก.8 แสดงการตั้งค่าในโปรแกรม (2)	47
ภาพที่ ข.1 เว็บไซต์ NETPIE	49
ภาพที่ ข.2 กดปุ่มการสมัคร	49
ภาพที่ ข.3 กรอกข้อมูลของผู้ใช้งาน	50
ภาพที่ ข.4 Log in เข้าใช้งาน	50
ภาพที่ ค.1 เปิดสวิตซ์ที่ตัวอุปกรณ์	52
ภาพที่ ค.2 สวมใส่อุปกรณ์	52

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ ค.3 เข้าเว็บไซต์ NETPIE	53
ภาพที่ ค.4 Log in เข้าใช้งาน	53
ภาพที่ ค.5 เข้าไปที่ FREEBOARDS	54
ภาพที่ ค.6 เลือก FREEBOARD ที่ชื่อ CowBehavior	54
ภาพที่ ค.7 หน้าจอแสดงผลพฤติกรรมของโคนม	55
ภาพที่ ค.8 เลือก FEEDS ที่ชื่อ CowBehavior	55
ภาพที่ ค.9 หน้าจอแสดงผลที่บันทึกไว้	56



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

ในประเทศไทยอุตสาหกรรมโคนมเป็นส่วนสำคัญต่อภาคการเกษตร เนื่องจากผลิตภัณฑ์จากน้ำนมวัวเป็นสารอาหารสำคัญที่มนุษย์ใช้ในการดำรงชีวิต ดังนั้นเพื่อที่จะควบคุมคุณภาพของนมให้มีคุณภาพดี ฟาร์มโคนมจึงพยายามรักษาสุขภาพและอารมณ์ของโคนมให้มีความสมบูรณ์แข็งแรง อยู่เสมอ (สุธาสินี นพฤทธิ์, ฌฐพล พันธวงค์ และ กิติ์สุชาติ พสุภา, 2557) ในปัจจุบันฟาร์มโคนมต้องมีการวิเคราะห์เพื่อศึกษาพฤติกรรมโคนมโดยใช้ผู้ที่เชี่ยวชาญและมีความสามารถในการศึกษาและสังเกตพฤติกรรมโคนม แต่ในการศึกษาพฤติกรรมนั้นมิต้นทุนที่สูงและต้องใช้ผู้เชี่ยวชาญเฉพาะทาง ในปี พ.ศ. 2556 ได้มีการสังเกตพฤติกรรมของโคนมโดยอาศัยเทคนิคการสุ่มรายตัว ผลการทดลองพบว่าวิธีการดังกล่าวมีประสิทธิภาพเนื่องจากมีการรบกวนสัตว์น้อยและผู้สังเกตการณ์มีเวลาพักมากขึ้น นอกจากนี้มีการใช้เทคโนโลยีตัวรับรู้หรือเซนเซอร์ในการศึกษาพฤติกรรมการเคลื่อนไหวของโคนมเพื่อลดแรงงานของผู้สังเกตการณ์ (อรสา บุญมีรติ และ จำเริญ เทียงธรรม, 2556) และ Grothmann et al. (2014) ได้ศึกษาพฤติกรรมของโคนม โดยใช้การเปรียบเทียบจากการปรับเปลี่ยนความถี่ของการให้อาหารโดยใช้เครื่องให้อาหารอัตโนมัติ ผลการทดลองพบว่าการเปลี่ยนความถี่ในการให้อาหารไม่มีผลกระทบต่อพฤติกรรมต่าง ๆ ของโคนม

จากการศึกษาพฤติกรรมที่ได้กล่าวมาผู้จัดทำจึงมีแนวคิดการนำเทคโนโลยี Internet of Things (IoT) เข้ามาประยุกต์ใช้ โดยเทคโนโลยี Internet of Things (IoT) ได้เข้ามามีบทบาทต่อการใช้ชีวิตประจำวันมากขึ้น ทำให้การเชื่อมโยงอุปกรณ์ต่าง ๆ มีการนำเข้าสู่อินเทอร์เน็ต ทำให้การใช้งานมีความอิสระมากขึ้น เช่น การควบคุมเครื่องใช้ไฟฟ้าภายในบ้านเรือนผ่านทางอินเทอร์เน็ต หรือการควบคุมฟาร์มอัจฉริยะผ่านทางอินเทอร์เน็ต ทำให้การนำเทคโนโลยี Internet of Things (IoT) ช่วยยกระดับการใช้ชีวิตมากยิ่งขึ้น (เจษฎา ขจรฤทธิ์, ปิยนุช ชัยพรแก้ว และ หนึ่งฤทัย เอ็งฉ้วน, 2560) ดังนั้นจึงเป็นที่มาในการจัดทำโครงการขึ้นมาเพื่อที่จะเพิ่มประสิทธิภาพในการเก็บข้อมูลและลดต้นทุนในการใช้ผู้เชี่ยวชาญเฉพาะทางมาศึกษาพฤติกรรมของโคนมในการเก็บข้อมูลที่จะส่งต่อให้นักวิจัยในลำดับถัดไป

1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 เพื่อพัฒนาอุปกรณ์เก็บข้อมูลพฤติกรรมของโคนม
- 1.2.2 เพื่อเก็บข้อมูลพฤติกรรมการเคลื่อนไหวของโคนม
- 1.2.3 เพื่อระบุตำแหน่งที่อยู่ปัจจุบันของโคนม

1.3 ขอบเขตของการวิจัย

- 1.3.1 ใช้โคนมในการทดสอบ 1 ตัว ต่ออุปกรณ์ 1 ชิ้น โดยใช้กลุ่มตัวอย่าง 2 ตัว
- 1.3.2 ใช้อุปกรณ์สัญญาณอินเทอร์เน็ตไร้สาย (WIFI มาตรฐาน IEEE 802.11)
- 1.3.3 ตรวจสอบการเคลื่อนไหวของโคนมเช่น ยืน นิ่ง เดิน นอน
- 1.3.4 ตรวจสอบข้อมูลที่บันทึกได้ผ่านทาง Net pie โดยเก็บข้อมูลทุก ๆ 5 นาที แสดงผลเป็นกราฟ มีรายงานหรือบันทึกประวัติ รายสัปดาห์ ได้จากโทรศัพท์สมาร์ทโฟน และคอมพิวเตอร์
- 1.3.5 ระบุตำแหน่งที่อยู่ปัจจุบันของโคนม โดยเก็บข้อมูลทุก ๆ 1 นาที แสดงผลเป็นตำแหน่งบนแผนที่
- 1.3.6 ใช้บอร์ด NodeMCU ESP 8266 ควบคุมการทำงานของอุปกรณ์นี้

1.4 วิธีการดำเนินงาน

- 1.4.1 ศึกษาทฤษฎีเอกสารที่เกี่ยวข้อง
- 1.4.2 ศึกษาข้อมูลอุปกรณ์และโปรแกรมการทำงาน
- 1.4.3 ออกแบบอุปกรณ์และโปรแกรมควบคุมการทำงาน
- 1.4.4 สร้างอุปกรณ์
- 1.4.5 ทดสอบการใช้อุปกรณ์
- 1.4.6 สรุปผลการทดสอบ
- 1.4.7 จัดทำคู่มือการใช้งาน

1.5 ระยะเวลาดำเนินโครงการ

ตารางที่ 1.1 แสดงระยะเวลาขั้นตอนการดำเนินโครงการ

ขั้นตอนการทำงาน	พ.ศ. 2561				พ.ศ. 2562		
	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. ศึกษาทฤษฎีเอกสารที่เกี่ยวข้อง							
2. ศึกษาข้อมูลอุปกรณ์และโปรแกรมการทำงาน							
3. ออกแบบอุปกรณ์และโปรแกรมควบคุมการทำงาน							
4. สร้างอุปกรณ์							
5. ทดสอบการใช้อุปกรณ์							
6. สรุปผลการทดสอบ							
7. จัดทำคู่มือการใช้งาน							

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1.6.1 ผู้ใช้สามารถทราบพฤติกรรม การเคลื่อนไหวของโคนม ได้ตลอดเวลา
- 1.6.2 นักวิจัยจะได้รับข้อมูลพฤติกรรมโคนมที่ได้มาเพื่อนำข้อมูลที่ได้ไปวิเคราะห์ต่อไป
- 1.6.3 ในอนาคตสามารถต่อยอดในการเพิ่มเซนเซอร์อุณหภูมิเพื่อนำมาวิเคราะห์อาการป่วยของโคนมได้

บทที่ 2

ทฤษฎีที่และงานวิจัยที่เกี่ยวข้อง

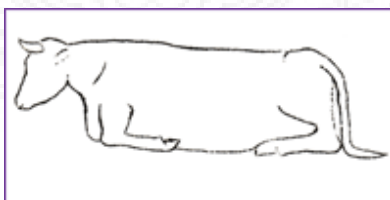
ผู้จัดทำโครงการได้ศึกษาค้นคว้า เอกสารทฤษฎีและงานวิจัยต่าง ๆ ที่เกี่ยวข้องดังนี้

- 2.1 พฤติกรรมโคนม
- 2.2 บอร์ดควบคุม
- 2.3 เทคโนโลยีตัวรับรู้
- 2.4 แหล่งพลังงาน
- 2.5 เทคโนโลยี IOT
- 2.6 งานวิจัยที่เกี่ยวข้อง

2.1 พฤติกรรมโคนม

2.2.1 การนอน ปกติโคนมจะชอบนอนพักผ่อนเพื่อสะสมพลังงาน เวลาหลังรับประทานอาหารอิ่ม โดยจะนอนในท่าขาคุ่หน้า พับไว้ใต้คอ ส่วนท้ายของลำตัว อยู่ในท่านอนตะแคงโดยยื่นขาหลังไปซ้ายหรือขวา คือท่านอนปกติ

เมื่อโคนมมีอาการผิดปกติจะสังเกตเห็นได้ เช่น ป่วยเป็นไขน้ำนม ตอนแรกโคนมจะนอนโดยใช้อกยันพื้น หัวพับไปข้างใดข้างหนึ่ง ระยะต่อมาจะซึมมาก และนอนตะแคงข้าง ขาทั้ง 4 เขยียดออกไป นอกจากนั้นจะสังเกตเห็นได้ว่าโคที่ป่วยมาก ๆ มักนอนหัวตก หูลู่ หรือคอบีบ เคลื่อนไหวไปไหนจะรู้สึกลำบาก แต่ถ้าโคที่ป่วยด้วยโรคเกี่ยวกับระบบประสาท เช่น บาดทะยัก หรือโรคในสมอง สัตว์เหล่านี้มักจะยึดเกร็งคอมากจนสังเกตเห็นได้ชัด



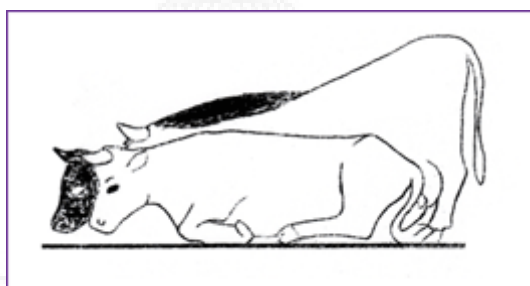
ภาพที่ 2.1 ท่านอนปกติ



ภาพที่ 2.2 ท่านอนพักบนหน้าอก

(ที่มา: https://th.pngtree.com/freepng/tummy-resting-cow_2224949.html)

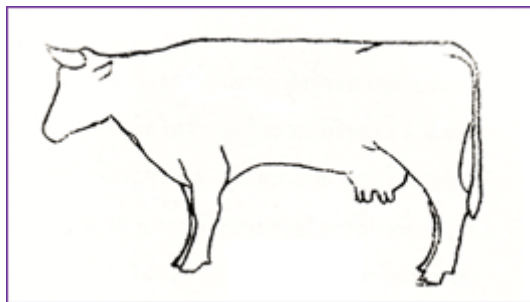
2.1.2 การลุก เมื่อโคนมจะลุก โดยจะใช้ ขาหลังทั้ง 2 ข้างยันพื้นก่อนแล้วก็ตามด้วย 2 ขาหน้า แต่ถ้าลุกต่างจากนี้ แสดงว่าอาจจะเกิดอาการเจ็บที่ขาของโคนมได้ แต่บางครั้งถ้าพบการลุกผิดปกติตั้งแต่แรกเกิด ก็อาจจะเกิดเนื่องจากความผิดปกติที่สมองหรือเป็นกรรมพันธุ์ได้



ภาพที่ 2.3 โคลุกยกขาหลังขึ้นก่อน

(ที่มา: http://www.thailivestock.com/cattle_handling)

2.1.3 การยืน ปกติโคนมเวลาขึ้นจะตัวตรง ถ้าหากยืนผิดปกติจากนี้ก็จะสังเกตได้โดยโคนมจะยืนหลังโค้งอาจจะมีความเสี่ยงที่จะป่วยได้ นอกจากนี้โคนมที่มีการขาดแร่ธาตุบางชนิด เช่น แคลเซียม อาจจะมีอาการกล้ามเนื้อสั่นเวลาขึ้น ขาหลังอ่อนไม่ค่อยมีแรง ถ้าโคขาดธาตุแมกนีเซียมมาก ๆ โคนมก็จะยืนตัวตรงไม่ได้ เข้าอ่อน สะดุดล้มอยู่เสมอ ทำเดินจะมีอาการงอแง เหมือนแบกของหนักบนหลัง



ภาพที่ 2.4 ทำยีนปกติ

(ที่มา: <https://sites.google.com/site/uraiwanpamwwy8799/kar-phathnakar-leiyng-khonm-1/kar-leiyng-kho-run-kho-saw>)

2.1.4 การเดิน ถ้าโคมีอาการเดินผิดปกติ สังเกตเห็นส่วนขาข้างที่เจ็บจะยกอย่างรวดเร็ว เวลาก้าว และโคจะไม่ ยอมใช้ขาข้างเจ็บรับน้ำหนัก ซึ่งจะสัมพันธ์กับการลุก และการยืนที่จะผิดปกติ ไปด้วยเสมอ ส่วนใหญ่มักเกิดจากการอักเสบตามข้อหรือตามกล้ามเนื้อ อาจพบข้อขาบวม ร้อน คล้ำ ดูจะเจ็บปวด หรือมีบาดแผลที่ส่วนพื้นกีบโรกิบ เช่น กรณีพื้นกีบเปื่อยจากการยืนในพื้นที่ที่มีน้ำขัง หรือขึ้นอยู่เสมอและ กรณีเป็นโรคปากและเท้าเปื่อยกรณีที่กีบแตก หรือมีเศษหิน เศษตะปูตำที่ ซอกกีบ หรือมีเนื้องอกที่ระหว่างกีบ เป็นต้น

นอกจากนี้โคอาจมีอาการเดินขาแข็ง ๆ โดยจะไม่มีบาดแผล หรือการอักเสบที่ใดให้เห็น กรณีเช่นนี้อาจเกิดจากโรคไข้ขาแข็ง (ไข้สามวัน)

2.2 บอร์ดควบคุม

2.2.1 ไอซีไมโครคอนโทรลเลอร์ เชื่อมต่อ WiFi NodeMCU V2 ESP8266

ESP8266 เป็นโมดูล WiFi ที่เขียนโปรแกรมลงไปได้ ทำให้นำไปใช้งานแทน Micro controller ได้ มีพื้นที่โปรแกรม 4MB ทำให้มีพื้นที่ในการเขียนโปรแกรมลงไปได้ ESP8266 ทำงาน โดยใช้แรงดันไฟฟ้า 3.3V - 3.6V เวลานำไปใช้งานร่วมกับอุปกรณ์เซ็นเซอร์ตัวอื่น ๆ โดยที่ใช้แรงดัน 5V กระแสที่โมดูลใช้งานสูงสุดคือ 200mA



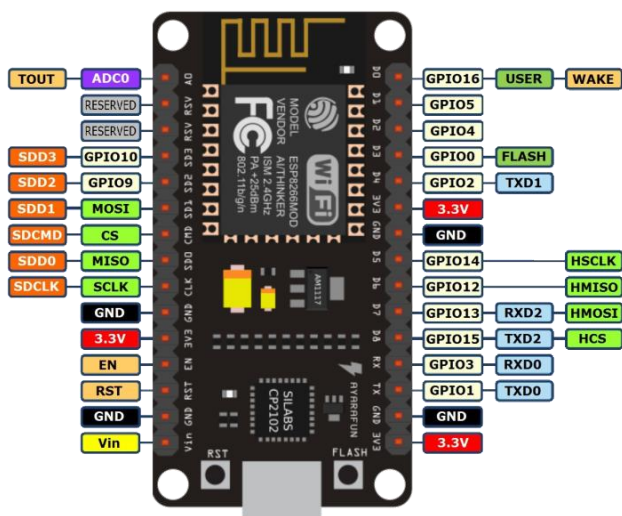
ภาพที่ 2.5 Module WiFi NodeMCU V2

(ที่มา: <https://bestbay.com.au/product/nodemcu-v1/>)

หลักการทำงานของ NodeMCU V2 ESP8266

ขาของโมดูล ESP8266 แบ่งได้ดังนี้

1. VCC: เป็นขาสำหรับจ่ายไปเข้าเพื่อให้โมดูลทำงานได้ ซึ่งแรงดันที่ใช้งานได้คือ 3.3 - 3.6V
2. GND: Ground
3. Reset: และ CH_PD (หรือ EN) เป็นขาที่ต้องต่อเข้าไฟ + เพื่อให้โมดูลสามารถทำงานได้ ทั้ง 2 ขานี้สามารถนำมาใช้รีเซ็ตโมดูลได้เหมือนกัน แต่ต่างตรงที่ขา Reset สามารถถอยไว้ได้ แต่ขา CH_PD (หรือ EN) จำเป็นต้องต่อเข้าไป + เท่านั้น เมื่อขานี้ไม่ต่อเข้าไฟ + โมดูลจะไม่ทำงานทันที
4. GPIO: เป็นขาดิจิตอลอินพุต / เอาต์พุต ทำงานที่แรงดัน 3.3V
5. GPIO15: เป็นขาที่ต้องต่อลง GND เท่านั้น เพื่อให้โมดูลทำงานได้
6. GPIO0: เป็นขาสำหรับการเลือกโหมดทำงาน หากนำขานี้ลง GND จะเข้าโหมดโปรแกรม หากถอยไว้ หรือนำเข้าไฟ + จะเข้าโหมดการทำงานปกติ
7. ADC: เป็นขานาล็อกอินพุต รับแรงดันได้สูงสุดที่ 1V ขนาด 10 บิต การนำไปใช้งานกับแรงดันที่สูงกว่าต้องใช้วงจรแบ่งแรงดันเข้าช่วย



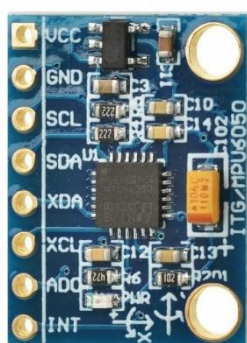
ภาพที่ 2.6 ผังของ Module WiFi NodeMCU V2

(ที่มา: https://sites.google.com/site/krukritsada/computing_science/smarthome/nodemcu)

2.3 เทคโนโลยีตัวรับรู้

2.3.1 เซนเซอร์จับความเร่ง GY-521 Serial Accelerometer/Gyro Module MPU6050

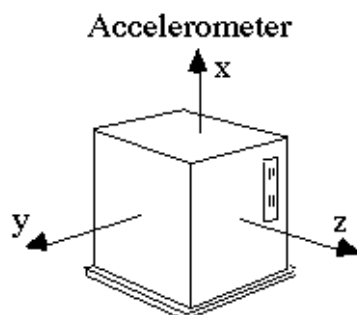
GY-521 เป็นโมดูล Accelerometers และ Gyroscope ซึ่งสามารถทำงานได้ทั้ง 2 อย่างได้พร้อมกัน ใช้ในการตรวจสอบทิศการเคลื่อนที่ และสามารถใช้ในการตรวจสอบความเร็วในการเปลี่ยนแปลงทิศทาง ของแกน XYZ ได้



ภาพที่ 2.7 โมดูล ACC/Gyro GY-521 MPU-6050

(ที่มา: <https://www.circuitointegrato.com>)

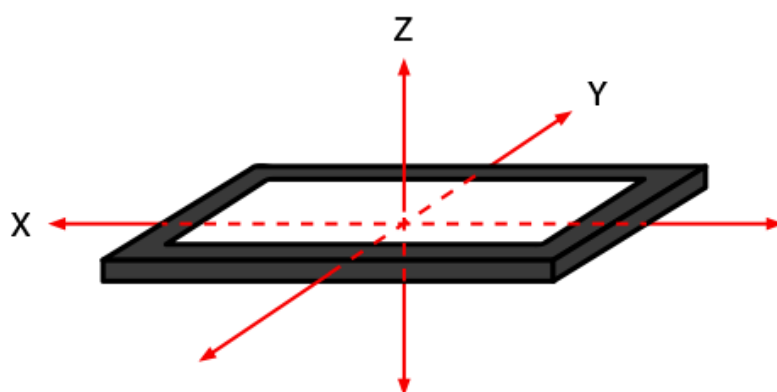
Accelerometer คือ สามารถวัดความเร่งเชิงเส้นของการเคลื่อนไหวหรือเคลื่อนที่ของวัตถุ สิ่งของ โดยจะวัดความเร่งตามแนวแกน XYZ โดยทำงานเป็น Sensor วัดความเร่งเพิ่มขึ้น หรือลดลง (ในหน่วย m/s^2)



ภาพที่ 2.8 หลักการของ Accelerometer

(ที่มา: <https://sites.google.com/site/thaimulticopter/acc>)

Gyroscope ก็คล้ายกับ Accelerometer แต่จะทำงานแตกต่างกันออกไป โดยที่ Accelerometer จะวัดความเร่งเชิงเส้นตามแนวแกน XYZ แต่สำหรับ Gyro จะเป็นความเร็วเชิงมุมของแกน XYZ



ภาพที่ 2.9 หลักการทำงานของ Gyroscope

(ที่มา: <http://www.akexorcist.com/2013/03/android-code-gyroscope.html>)

รายละเอียด GY-521 mpu6050

1. ใช้Chip: MPU6050
2. ใช้ไฟเลี้ยง: +3.3-5 v
3. การเชื่อมต่อ: I2C BUS (Inter Integrate Circuit Bus)
4. ขนาด โมดูล: 16 mm * 20 mm

ขาของ GY-521

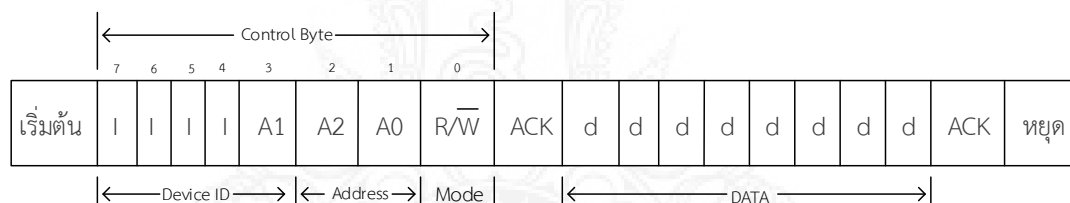
1. VCC_IN: ขารับไฟ +5 โวลต์ไปที่ Regulate 3.3 โวลต์
2. 3.3V: ขาไฟ 3.3 โวลต์

3. GND: Ground
4. SCL serial data: ขาสัญญาณนาฬิกา บนบัส I2C
5. SDA serial clock: ขาสัญญาณข้อมูล บนบัส I2C
6. XDA(AUX_SDA) : ขาสัญญาณข้อมูล บนบัส I2C (I2C Master Mode is enabled)
7. XCL(AUX_SCL): ขาสัญญาณนาฬิกา บนบัส I2C (I2C Master Mode is enabled)
8. INT: Interrupt

การเชื่อมต่ออุปกรณ์แบบ I2C (Inter Integrate Circuit Bus) ของอุปกรณ์ Gy-521

I2C Bus ย่อมมาจาก Inter Integrate Circuit Bus เป็นการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ที่ต่าง ๆ แบบอนุกรม ซิงโครนัส (Synchronous) โดยจะใช้สายสัญญาณในการเชื่อมต่อ 2 เส้น คือ SCL (serial clock) และ SDA (serial data) โดยที่เชื่อมต่อกับอุปกรณ์ตัวอื่นได้หลาย ๆ ตัว แต่ใช้เพียงแค่ 2 พอร์ต

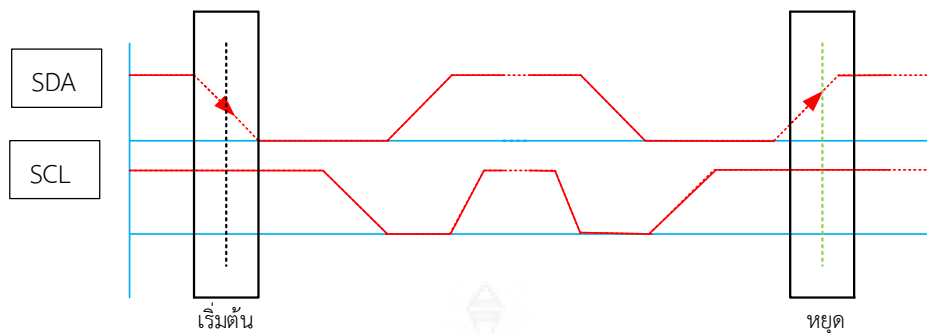
การรับ-ส่งข้อมูล I2C BUS Microcontroller



ภาพที่ 2.10 หลักการทำงาน I2C

รูปแบบการเขียน/อ่านข้อมูลแบบ I2C BUS

1. สถานะเริ่มต้น เพื่อแสดงการขอใช้บัส
 2. รหัสควบคุม ประกอบด้วย ด้วยรหัส ประจำตัวอุปกรณ์ Device ID ,Device Address ,และ Mode ในการเขียนหรืออ่านข้อมูล
 3. เมื่ออุปกรณ์ รับทราบว่า Microcontroller ต้องการ จะติดต่อด้วยก็ต้องส่ง สถานะรับรู้ หรือแจ้งให้ Microcontroller รับรู้ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง
 4. เมื่อสิ้นสุดการส่งข้อมูล Microcontroller จะต้องส่ง สถานะสิ้นสุด เพื่อบอกกับ อุปกรณ์ว่า สิ้นสุดการใช้บัส
 5. เมื่อบัสไม่ได้ถูกใช้งานหรือหยุดการทำงาน สถานะจะกลายเป็น 1 ทั้งหมด
- การกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I2C BUS



ภาพที่ 2.11 หลักการทำงาน SDAและ SCL ของสัญญาณ I2C

สถานการณ์เริ่มทำงานและการสิ้นสุดทำงานของ I2C BUS

- เมื่อต้องการส่งข้อมูล Microcontroller จะส่งสถานะเริ่มต้น ให้ SDA จาก 1 เป็น 0 และ SCL จาก 0 เป็น 1
- เมื่อสิ้นสุดการใช้บัส Microcontroller จะส่งสถานะสิ้นสุด ให้ SDA จาก 0 เป็น 1 และ SCL จาก 1 เป็น 0

การเชื่อมต่อ Address โดยการเชื่อมต่อขา Address จะมีทั้งหมดดังนี้

- 0x3B กับ 0X3C จะได้ค่า ACC_X
- 0x3D กับ 0X3E จะได้ค่า ACC_Y
- 0x3F กับ 0X4D จะได้ค่า ACC_Z
- 0x43 กับ 0X44 จะได้ค่า Gyro_X
- 0x45 กับ 0X46 จะได้ค่า Gyro_Y
- 0x47 กับ 0X48 จะได้ค่า Gyro_Z

2.3.2 module GPS NEO-6M Ublox/u-blox Ublox NEO-6MV2 GPS Module GY-GPS6MV2

GPS คือ ระบบที่ใช้ระบุตำแหน่งบนพื้นโลก ซึ่งย่อมาจากคำว่า Global Positioning System ซึ่งระบบ GPS ประกอบไปด้วย 3 ส่วนหลัก คือ

- 1) ส่วนอวกาศ ประกอบด้วยเครือข่ายดาวเทียมหลัก
- 2) ส่วนควบคุม ประกอบด้วยสถานีภาคพื้นดิน
- 3) ส่วนผู้ใช้งาน ผู้ใช้งานต้องมีเครื่องรับสัญญาณที่สามารถรับคลื่นและแปรรหัสจากดาวเทียมเพื่อนำมาประมวลผลให้เหมาะสมกับการใช้งานในรูปแบบต่าง ๆ

หลักการทำงานของ GPS

ดาวเทียม GPS ประกอบด้วยดาวเทียม 24 ดวง โดยแบ่งเป็น 6 รอบวงโคจร การจรจะเอียงทำมุมเอียง 55 องศา กับเส้นศูนย์สูตร (Equator) ในลักษณะสานกันคล้ายกับลูกตะกร้อแต่ละวงโคจรมีดาวเทียม 4 ดวง รัศมีวงโคจรจากพื้นโลก 20,162.81 กม. หรือ 12,600 ไมล์ ดาวเทียมแต่ละดวงใช้เวลาในการโคจรรอบโลก 12 ชั่วโมง

GPS ทำงานโดยการรับสัญญาณจากดาวเทียมแต่ละดวง โดยสัญญาณดาวเทียมนี้ประกอบไปด้วยข้อมูลที่ระบุตำแหน่งและเวลาขณะส่งสัญญาณ ตัวเครื่องรับสัญญาณ GPS จะต้องประมวลผลความแตกต่างของเวลาในการรับสัญญาณเทียบกับเวลาจริง ณ ปัจจุบันเพื่อแปรเป็นระยะทางระหว่างเครื่องรับสัญญาณกับดาวเทียมแต่ละดวง ซึ่งได้ระบุมีตำแหน่งมากับสัญญาณดังกล่าวข้างต้น

เพื่อให้เกิดความแม่นยำในการค้นหาตำแหน่งด้วยดาวเทียม ต้องมีดาวเทียมอย่างน้อย 4 ดวง เพื่อบอกตำแหน่งบนผิวโลก ซึ่งระยะห่างจากดาวเทียมทั้ง 3 กับเครื่อง GPS จะสามารถระบุตำแหน่งบนผิวโลกได้ หากพื้นโลกอยู่ในแนวระนาบแต่ในความเป็นจริงพื้นโลกมีความโค้งเนื่องจากสัณฐานของโลกมีลักษณะกลม ดังนั้นดาวเทียมดวงที่ 4 จะทำให้สามารถคำนวณเรื่องความสูงเพื่อให้ได้ตำแหน่งที่ถูกต้องมากขึ้น

นอกจากนี้ความแม่นยำของการระบุตำแหน่งนั้นขึ้นอยู่กับตำแหน่งของดาวเทียมแต่ละดวง กล่าวคือถ้าระยะห่างระหว่างดาวเทียมที่ใช้งานอยู่ห่างกันย่อมให้ค่าที่แม่นยำกว่าที่อยู่ใกล้กัน และยังมีจำนวนดาวเทียมที่รับสัญญาณได้มากก็ยิ่งให้ความแม่นยำมากขึ้น ความแปรปรวนของชั้นบรรยากาศชั้นบรรยากาศประกอบด้วยประจุไฟฟ้า ความชื้น อุณหภูมิ และความหนาแน่นที่แปรปรวนตลอดเวลา คลื่นเมื่อตกกระทบ กับวัตถุต่าง ๆ จะเกิดการหักเหทำให้สัญญาณที่ได้อ่อนลงและสิ่งแวดล้อมในบริเวณรับสัญญาณเช่นมีการบดบังจากกระจก ละอองน้ำ ใบไม้ จะมีผลต่อค่าความถูกต้องของความแม่นยำ เนื่องจากถ้าสัญญาณจากดาวเทียมมีการหักเหก็จะทำให้ค่าที่คำนวณได้จากเครื่องรับสัญญาณเพี้ยนไป และสุดท้ายก็คือประสิทธิภาพของเครื่องรับสัญญาณว่ามีความไวในการรับสัญญาณแค่ไหนและความเร็วในการประมวลผลด้วย

การวัดระยะห่างระหว่างดาวเทียมกับเครื่องรับทำได้โดยใช้สูตรคำนวณ [ระยะทาง = ความเร็ว X ระยะเวลา] วัดระยะเวลาที่คลื่นวิทยุส่งจากดาวเทียมมายังเครื่องรับ GPS คุณด้วยความเร็วของคลื่นวิทยุจะเท่ากับระยะทางที่เครื่องรับ อยู่ห่างจากดาวเทียม โดยเวลาที่วัดได้มาจากนาฬิกาของดาวเทียมที่มีความแม่นยำสูงมีความละเอียดถึงนาโนวินาที และมีการสอบทวนเสมอ ๆ กับสถานีภาคพื้นดิน

องค์ประกอบสุดท้ายคือตำแหน่งของดาวเทียมแต่ละดวงในขณะที่ส่งสัญญาณว่าอยู่ที่ใด (Almanac) มายังเครื่องรับ GPS โดยวงโคจรของดาวเทียมได้ถูกกำหนดไว้ล่วงหน้าแล้วเมื่อถูกส่ง

ชั้นสู่อวกาศ สถานีควบคุมจะคอยตรวจสอบการโคจรของดาวเทียมอยู่ตลอดเวลาเพื่อทวนสอบความถูกต้อง

โมดูล GPS GY-NEO6MV2 เป็นโมดูล U-blox รุ่น NEO-6M สามารถเชื่อมต่อได้กับไมโครคอนโทรลเลอร์หลายหลายประเภทไม่ว่าจะเป็น NodeMCU ผ่านทาง Serial UART ด้วยความเร็วที่ 9600

NMEA เป็นมาตรฐานการสื่อสารระหว่างอุปกรณ์ใช้งานกับโมดูล GPS เมื่อต้องการที่จะข้อมูลมาจากโมดูล GPS จำเป็นจะต้องรู้รูปแบบการสื่อสารแบบ NMEA สามารถระบุข้อมูลออกมาได้หลายแบบด้วยกัน ซึ่งเมื่อมีการเริ่มส่งข้อมูลมา จะเริ่มต้นการส่งด้วยเครื่องหมาย \$ แล้วจบด้วยเครื่องหมาย * หลังจากนั้นจึงเป็นการตรวจสอบค่าความผิดพลาดด้วยตัวเลข ข้อมูลที่ถูกส่งมาจะนำหน้าด้วย GPRMC GPGGA และอื่น ๆ

รูปแบบของข้อมูล GPRMC ยกตัวอย่างได้มีดังนี้

\$GPRMC , 225446 , A , 4916.45 , N , 12311.12 , W , 000.5

- 225446 = เวลามาตรฐาน UTC เวลา 22 นาฬิกา 54 นาที 46 วินาที
- A = สามารถจับสัญญาณได้ แต่ถ้าเป็นสัญลักษณ์ V = ไม่สามารถจับสัญญาณได้
- 4916.45 = 49 องศา 16.45 นาที
- N = องศาเหนือ จะได้ค่า Latitude เป็นบวก
- 12311.12 = แยกออกมาได้ 123 องศา 11.12 นาที
- W = องศาตะวันตก จะได้ค่า Longitude เป็นลบ
- 000.5 = ความเร็วที่พื้นโลก มีหน่วยเป็น Knots

การหาค่าละติจูดและลองจิจูดในรูปทศนิยมจากรูปองศาลิปดา สามารถหาได้แบบเดียวกัน
ดังนี้

$$\text{Lat/Lng} = \text{องศา} + (\text{นาที} / 60) \quad (2.1)$$

เมื่อ Lat คือ ละติจูด

เมื่อ Lng คือ ลองจิจูด

จากตัวอย่างข้างต้น สามารถหาค่าละติจูด และลองจิจูดได้ดังนี้

$$\text{Lat} = 49 + (16.45 / 60) = 49.274166$$

$$\text{Lng} = 123 + (11.12 / 60) = 123.185333$$



ภาพที่ 2.12 GY-GPS6MV2

(ที่มา: <https://www.arduinoall.com/product/149/neo-6m-ublox-u-blox-ublox-neo-6mv2-gps-module-gy-gps6mv2-พร้อมสายอากาศ>)

รายละเอียด GY-GPS6MV2

1. ใช้ไฟเลี้ยง: 3.3 V-5 V
2. ขนาด : 36mm * 24mm
3. แรงดันไฟ : 3.3 – 5 V
4. ความถี่อัปเดต : 1 Hz ถึง 5 Hz

2.3.3 โมดูลเซนเซอร์ Ultrasonic HC-SR04

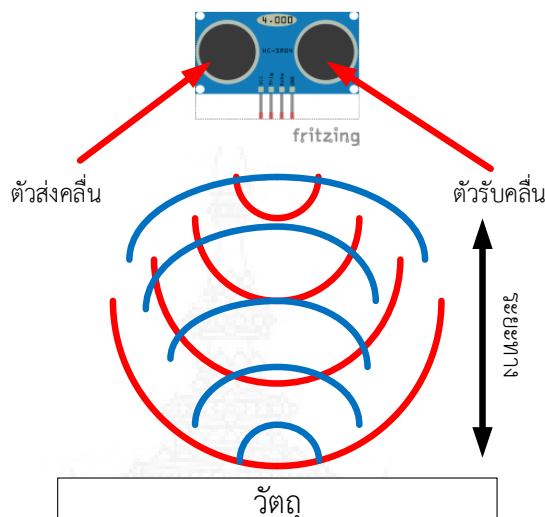
คลื่น Ultrasonic เป็นคลื่นเสียงชนิดหนึ่งที่มีความถี่สูงเกินกว่าในระดับที่มนุษย์จะได้ยินซึ่งมีความถี่สูงกว่า 20KHz ซึ่งสามารถเล็งคลื่นเสียงไปที่เป้าหมายได้เจาะจงเป็นคุณสมบัติของคลื่นอย่างหนึ่ง ยิ่งคลื่นมีความถี่สูงขึ้นความยาวคลื่นก็จะยิ่งสั้นลง คลื่นเสียงอัลตราโซนิก สามารถนำไปใช้งานได้ เช่น นำไปใช้ในเครื่องควบคุมระยะไกล หรือวัดระยะห่างจากวัตถุ เป็นต้น



ภาพที่ 2.13 Ultrasonic HC-SR04

(ที่มา: <https://www.botshop.co.za/product/hc-sr04-ultrasonic-sensor/>)

HC-SR04 เป็นอุปกรณ์ที่ใช้ในการตรวจจับวัตถุและระยะทางโดยการไม่สัมผัส สามารถวัดได้ตั้งแต่ 2-400 ซม. สามารถต่อใช้งานได้กับหลายอุปกรณ์ เช่น ไมโครคอนโทรลเลอร์ โดยจะประกอบไปด้วยตัว รับ-ส่ง อัลตราโซนิก ตัวส่งจะส่งคลื่นความถี่ 40 kHz ออกไปในอากาศด้วยความเร็วประมาณ 346 เมตรต่อวินาที และตัวรับจะคอยรับสัญญาณที่สะท้อนกลับจากวัตถุ



ภาพที่ 2.14 หลักการทำงาน Ultrasonic HC-SR04

รายละเอียดของ HC-SR04

- 1) ขา VCC สำหรับต่อแรงดันไฟเลี้ยงไม่เกิน 5V
- 2) ขา Trig เป็นขาอินพุตรับสัญญาณพัลส์ความกว้าง 10 ไมโครวินาทีเพื่อกระตุ้นการสร้างคลื่นอัลตราโซนิกความถี่ 40KHz ออกสู่อากาศจากตัวส่ง
- 3) ขา Echo เป็นขาเอาต์พุตสำหรับส่งสัญญาณพัลส์ออกจากโมดูลไปยังไมโครคอนโทรลเลอร์ เพื่อตรวจจับความกว้างของสัญญาณพัลส์และคำนวณเป็นระยะทาง
- 4) ขา GND สำหรับต่อจุดกราวด์ร่วมแรงดันและสัญญาณ

2.4 แหล่งพลังงาน

2.4.1 แบตเตอรี่ lithium polymer

แบตเตอรี่เป็นแหล่งพลังงานชนิดหนึ่ง โดยต้องชาร์จพลังงานที่เป็นกระแสไฟฟ้า DC เข้าไปในตัว ถึงจะใช้งานได้ปกติ เช่น โทรศัพท์ จำนวน mAh ที่คือส่วนแสดงความจุของ เช่น mAh คือ m = มิลลิ, A = แอมป์, h = hour (ชั่วโมง) ยกตัวอย่างเช่น 2500 mAh โดยจะใช้เป็นตัวจ่ายพลังงานไปยังอุปกรณ์เพื่อที่จะทำให้ตัวอุปกรณ์ทำงานได้ เมื่อตัว พลังงานหมด สามารถนำมาชาร์จพลังงานใหม่และนำไปใช้งานได้อีกครั้ง

ผ่านทางเครือข่ายอินเทอร์เน็ต เช่น การสั่งการทำงาน หรือเปิด-ปิด อุปกรณ์ อิเล็กทรอนิกส์ หรือ เครื่องใช้ไฟฟ้าได้ ผ่านทางระบบอินเทอร์เน็ตด้วย คอมพิวเตอร์ หรือโทรศัพท์มือถือ



ภาพที่ 2.17 Internet of Thing

(ที่มา: <https://hackernoon.com/when-iot-meets-blockchain-%EF%B8%8F-892fecdaf00c>)

2.5.2 NETPIE

NETPIE เป็น cloud platform หรือ Platform IoT ที่เอาไว้ใช้เชื่อมต่อกับอุปกรณ์ ต่างๆ เข้าด้วยกันที่ถูกออกแบบและพัฒนาขึ้นเพื่ออำนวยความสะดวกในการสื่อสารระหว่างอุปกรณ์ ในเครือข่าย IoT ช่วยให้อุปกรณ์สามารถสื่อสารกันโดยผู้พัฒนา ไม่ว่าจะอุปกรณ์นั้นจะอยู่ที่ไหน สามารถส่งข้อมูลจากอุปกรณ์ไปยัง NETPIE เพื่อบันทึกข้อมูลที่เป็นระบบ อีกทั้งCloud ได้ NETPIE เป็น platform ที่สามารถใช้งานได้ฟรี



ภาพที่ 2.18 NETPIE

(ที่มา: <https://www.nectec.or.th/innovation/innovation-software/netpie.html>)

2.6 งานวิจัยที่เกี่ยวข้อง

2.6.1 เซ็นเซอร์วัดหัวเข่าสำหรับการตรวจสอบพฤติกรรมของโค

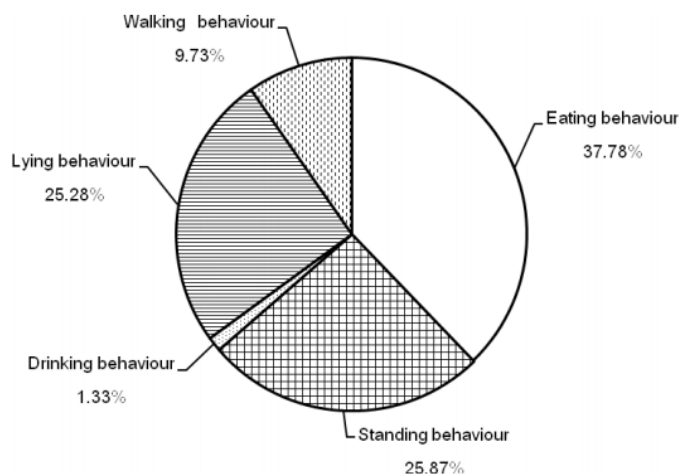
การเลี้ยงโคเป็นอาชีพทางการเกษตรที่สำคัญ อาชีพหนึ่ง ปัจจุบันการเลี้ยงโคได้มีการพัฒนาการเลี้ยงมาเป็นลักษณะฟาร์มมีการการเลี้ยงโคเป็นจำนวนมาก ซึ่งต้องใช้แรงงานคนในการเฝ้าพฤติกรรม เพื่อที่จะใช้เป็นข้อมูลในการดูแลป้องกันสุขภาพของโค แต่ความไม่เข้าใจพฤติกรรมของโคนั้นอาจเกิดปัญหาเช่นคนเลี้ยงไม่สามารถสังเกตพฤติกรรมของโคได้ตลอดเวลา และความรู้ความชำนาญในการสังเกตหรือแยกแยะความผิดปกติของโคนม รวมทั้งการใช้คนเลี้ยงจำนวนมากซึ่งจะมีผลกระทบต่อต้นทุนในการผลิตโคด้วย ดังนั้นจึงมีการมาจึงมีการพัฒนาอุปกรณ์เพื่อเฝ้าสังเกตพฤติกรรม โดยนำเอาเซนเซอร์วัดความเร่ง (Accelerometer sensor) มาติดตั้งที่ข้อขาของโคสำหรับตรวจจ็ระยะเวลาการยืนและการนอนของโค ซึ่งจะทำให้ทราบพฤติกรรมการดำรงชีวิตของโคและการสังเกตน้ำหนักของโคโดยใช้เซนเซอร์น้ำหนักซึ่งเซนเซอร์ทั้งสองจะถูกติดตั้งอยู่ที่แท่นชั่งน้ำหนักสำหรับวัดแรงกดขาทั้งสอง หากโคเริ่มมีอาการเจ็บเท้าข้างใดข้างหนึ่งจะทำให้การทรงตัวของโคไม่สมดุลทำให้สัญญาณที่ได้ผิดปกติไปและทำให้เราทราบอาการเจ็บของขาโคได้ในส่วนพฤติกรรมโคที่มีอาการปกติเมื่อปล่อยลงแปลงหญ้าโคจะแทะเล็มหญ้า ประมาณ 1-3 ชั่วโมงแล้วหยุดนิ่ง 10-20 นาทีทำแบบนี้สลับกันไปเมื่ออยู่ในคอก จะเคี้ยวเอื้องและหยุดนิ่งสลับกันไป ขณะหยุดนิ่งอาจยืนหรือนอน หูและหางยังคงกระดิกไล่แมลงตลอดเวลา ในหนึ่งวันโคจะดื่มน้ำ 3-4 ครั้ง วิธีดำเนินการทดลองได้ดำเนินการทดลองกับโคพันธุ์ บราห์มัน อายุ 1 ปี เพศเมีย จำนวน 1 ตัว สุขภาพโคของปกติและมีเงื่อนไขในการทดลองโดยตรวจจ็พฤติกรรม 3 ลักษณะคือ การเดิน การยืน การนอน ซึ่งในการทดลองนั้นเริ่มโดยการนำโคมาติดตั้งอุปกรณ์เซนเซอร์วัดหัวเข่าลักษณะการติดตั้งจะติดอยู่ขาหลังของโค โดยมีสายรัดเข้าส่วนบนของหัวเข่าและส่วนล่างของหัวเข่าในการรัดต้องรัดให้กระชับ แต่ไม่แน่นเกินไปให้ส่วนของซูดกลไกของเซนเซอร์แนบเข้าบริเวณของข้อต่อของหัวเข่า

2.6.2 การสังเกตพฤติกรรมโครีดนมในช่วงวันโดยเทคนิคการสุ่มรายตัว

การสังเกตพฤติกรรมโครีดนมในช่วงวันโดยเทคนิคการสุ่มรายตัวได้มีการนำแม่โคนมพันธุ์โฮลสไตน์ฟรีเซียน จำนวน 14 ตัว ในระบบโรงเรือนแบบปล่อยอิสระในคอกซึ่งมีแผ่นยางรองที่พื้นคอก ในส่วนของโปรแกรมการให้อาหารจะเป็นการจัดการโดยโรงรีดนมและจะใส่น้ำสะอาดและแขวนแร่ธาตุไว้ โดยใช้เทคนิคการสุ่มรายตัวโดยใช้อุปกรณ์ในการเก็บข้อมูลได้แก่ นาฬิกาแบบฟอร์มบันทึกผลเทอร์โมมิเตอร์ กระเปาะเปียก-แห้ง และเทอร์โมมิเตอร์แบบรายงานอุณหภูมิสูงสุด-ต่ำสุด จะสุ่มเก็บข้อมูลช่วง 5.30-16.30 เวลาละ 20 นาที และ พัก 10 นาที ทั้งหมด 21 ช่วง จะรวมเวลาได้ 420 นาทีต่อวัน เป็นระยะเวลา 5 วัน และผลการทดลองจะนำมาแสดงในรูปของกราฟวงกลม โดยคิดคำนวณเป็นเปอร์เซ็นต์จากค่าเฉลี่ยเวลาที่โค แต่ละตัวแสดงพฤติกรรมต่าง ๆ ในช่วงวัน เช่น การทานอาหารและน้ำ การยืน การนอน การเดิน ซึ่งจะแสดง ข้อมูลในของช่วงเวลา

กลางวัน และจะเป็นวิธีที่รบกวนสัตว์น้อย ผู้เก็บข้อมูลหลีกเลี่ยงการรบกวนสัตว์โดย สังเกตพฤติกรรม อยู่ห่าง ๆ จำกัดการเคลื่อนไหว ไม่ส่งเสียงดังรบกวน เพื่อความสะดวกและรวดเร็วในการ บันทึก ข้อมูลพฤติกรรมโคทดลองในระหว่างการเก็บข้อมูล จึงได้กำหนดสัญลักษณ์ประกอบรหัสพฤติกรรม ของโคที่ต้องการบันทึก โดยมีรายละเอียดดังนี้

1. SE (แม่โคยืนกินอาหารที่รางอาหารรวมไปถึงยืนเลียก้อนแร่ธาตุ)
2. SI (แม่โคยืนนิ่งไม่มีการขยับขา มีการเคลื่อนไหวร่างกายของส่วนอื่น เช่น สะบัด หัวหรือหาง ผิวหนังกระดูก เป็นต้น)
3. SR (แม่โคยืนเคี้ยวเอื้องโดยมีการขย่อนอาหารมาที่ปาก)
4. SD (แม่โคยืนกินน้ำ ที่อ่างน้ำ)
5. LI (แม่โคนอนนิ่งอาจมีการเคลื่อนไหวร่างกายของส่วนอื่น เช่น สะบัดหัวหรือหาง ผิวหนังกระดูก เป็นต้น)
6. LR (แม่โคนอนเคี้ยวเอื้องโดยมีการขย่อนอาหารมา ที่ปาก)
7. W (แม่โคก้าวเท้าเดิน)



ภาพที่ 2.19 กราฟวงกลม โดยคิดคำนวณเป็นเปอร์เซ็นต์จากค่าเฉลี่ยเวลาที่โค แต่ละตัวแสดง พฤติกรรมต่าง ๆ ในช่วงวัน

(ที่มา: Grothmann et al. (2014). Influence of Different Feeding Frequencies on the Rumination and Lying Behavior of Dairy Cows. International Conference)

บทที่ 3

ขั้นตอนและวิธีดำเนินการ

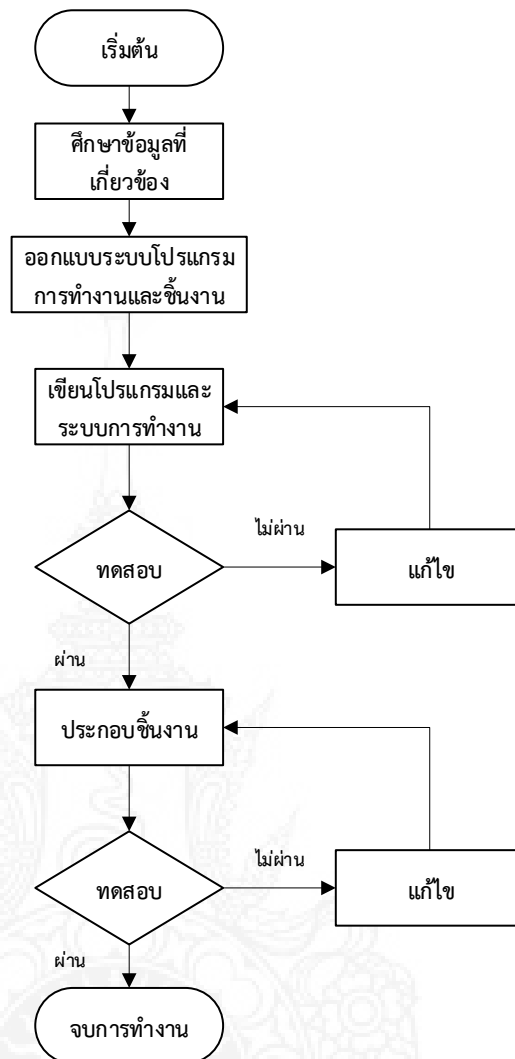
การออกแบบการพัฒนาอุปกรณ์สวมใส่เพื่อเก็บข้อมูลการเคลื่อนไหวของวัวโดยใช้เทคโนโลยีไอโอที (Development of Wearable Device for Logging Cow Behavior using IoT Technology) โดยขั้นตอนการออกแบบโครงการมีรายละเอียด ดังนี้

- 3.1 แผนการดำเนินโครงการ
- 3.2 ขั้นตอนการดำเนินการ
- 3.3 การพัฒนาระบบ

3.1 แผนการดำเนินโครงการ

จากการศึกษาทฤษฎีที่เกี่ยวข้องสามารถเขียนแผนผังแสดงขั้นตอนและวิธีการดำเนินงานต่างๆเป็นแผนภาพการทำงานได้ดังภาพที่ 3.1

เริ่มจากศึกษาข้อมูลที่เกี่ยวข้องกับพฤติกรรมของโคนมและอุปกรณ์ที่ใช้ในการทำโครงการนี้แล้วนำข้อมูลที่ศึกษามาทั้งหมดมาสรุปและเริ่มทำการออกแบบชิ้นงานและระบบโปรแกรมการทำงาน หลังจากนั้นจะมีการทดสอบชิ้นงานและปรับปรุงแก้ไข



ภาพที่ 3.1 ผังดำเนินโครงการ

3.2 ขั้นตอนการดำเนินการ

3.2.1 ศึกษาทฤษฎีที่เกี่ยวข้อง

1. ศึกษาพฤติกรรมของคอนม
 - การนอน
 - การลุก
 - ยืน
 - เดิน
2. ศึกษาหลักการทำงานของอุปกรณ์
 - ไอซีไมโครคอนโทรลเลอร์ เชื่อมต่อ WiFi
 - เซนเซอร์จับความเร่งและความเร็ว
 - โมดูล GPS
 - โมดูลเซนเซอร์ ultrasonic
 - แบตเตอรี่ lithium polymer
 - โมดูลชาร์ตแบตเตอรี่
3. ศึกษาเทคโนโลยี IOT
 - Internet of thing
 - NETPIE

3.2.2 ออกแบบและพัฒนาระบบ

1. ออกแบบตัวอุปกรณ์
2. ออกแบบระบบโปรแกรมการทำงาน

3.2.3 ทดสอบการใช้งาน

1. จำลองพฤติกรรมของคอนมโดยทดสอบเบื้องต้น
2. ปรับปรุงแก้ไข

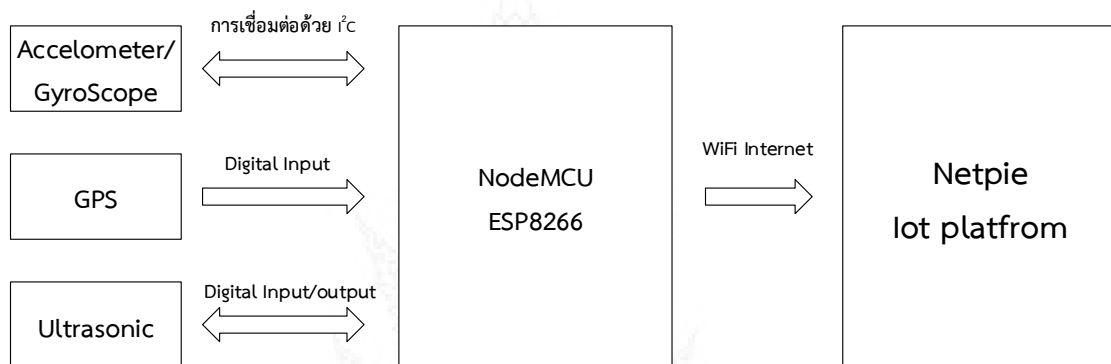
3.2.4 จัดทำเอกสารโครงการ

3.3 การพัฒนาระบบ

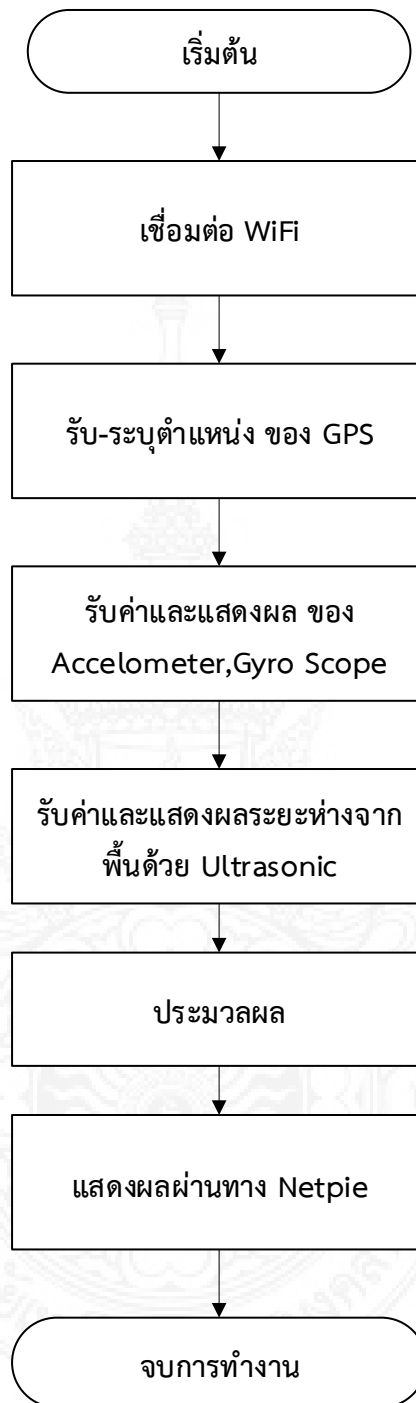
3.3.1 ภาพรวมของระบบ

โดยอุปกรณ์การทำงานจะประกอบด้วย บอร์ดควบคุมผ่าน WiFi NodeMCU Esp8266 เซ็นเซอร์จับความเร่งและความเร็ว Accelerometer/ Gyroscope mpu6050 module GPS โมดูลเซนเซอร์ Ultrasonic HC-SR04 ทำการเชื่อมต่อดังภาพที่ 3.2 หลักการทำงานของอุปกรณ์เริ่มต้นด้วยการให้ NodeMCU Esp8266 เชื่อมต่อกับสัญญาณ WIFIเพื่อที่จะไว้ทำการเชื่อมต่อกับ NETPIE ซึ่ง

เป็น IOT Platform หลังจากนั้น module GPS จะเริ่มการทำงานโดยจะรับสัญญาณจากดาวเทียม และส่งข้อมูลไปยัง NETPIE เพื่อที่จะระบุตำแหน่งของโคนมบนแผนที่ ในส่วนต่อไปจะเป็นการทำงาน ของ MPU6050 สามารถวัดความเร่งการเคลื่อนไหวหรือเคลื่อนที่ของโคนมโดยจะวัดความเร่ง Accelerometer ตามเชิงเส้น XYZ และ Gyroscope จะเป็นความเร็วเชิงมุมของ XYZ จะแสดงผล เป็นเส้นกราฟ ในส่วนของUltrasonic จะวัดระยะห่างจากพื้นดินของโคนมเพื่อที่จะระบุว่าโคนมกำลัง ยืนหรือนอนอยู่ เมื่อได้ข้อมูลจากอุปกรณ์ทั้งหมดแล้ว NodeMCU Esp8266 จะทำการส่งข้อมูล ทั้งหมดไปยัง NETPIE เพื่อที่จะแสดงผลการระบุตำแหน่ง และการวัดความเคลื่อนไหวของโคนมต่อไป ดังภาพที่ 3.3



ภาพที่ 3.2 ผังการทำงานของอุปกรณ์



ภาพที่ 3.3 ผังการทำงานโปรแกรมโดยรวม

3.3.2 การระบุตำแหน่ง

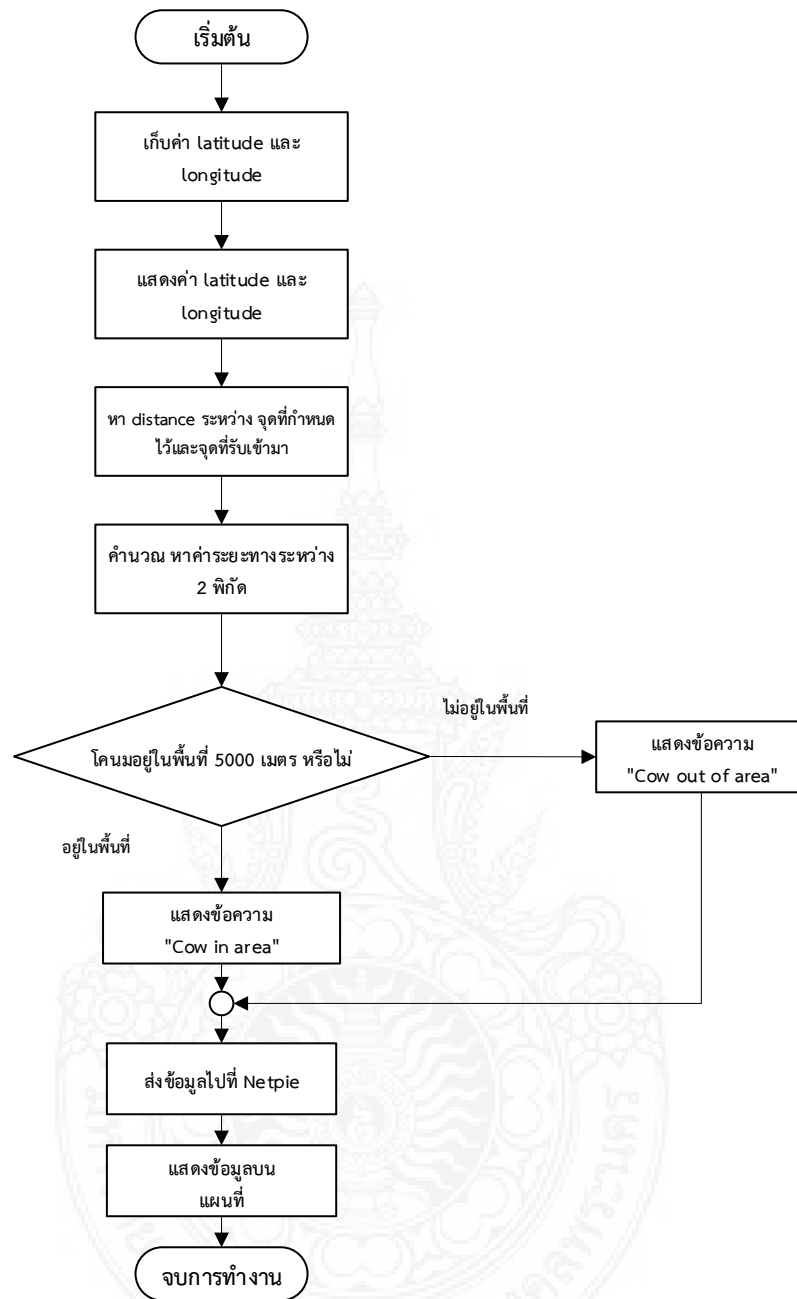
1. การแสดงบนแผนที่

หลักการทำการแสดงบนแผนที่ของ Module GPS โดยเริ่มรับสัญญาณจากดาวเทียม โดยสัญญาณดาวเทียมนี้ประกอบไปด้วยข้อมูลที่ระบุตำแหน่งเมื่อได้รับสัญญาณแล้ว จะทำการเก็บค่าตำแหน่ง Longitude และ Latitude แล้วทำการคำนวณหาค่าระยะทางระหว่างจุดที่กำหนดไว้ และจุดที่รับเข้ามาแปลงให้เป็น Radian แล้วประมวลผลโดยสูตรคำนวณ haversine formula แสดงค่าระยะห่างจากพิกัดที่กำหนดไว้และพิกัดที่รับมา

2. การระบุขอบเขตในการอยู่พื้นที่ของโคนม

หลังจากการระบุตำแหน่งแล้วจะเข้าสู่ขั้นตอนการเข้าเงื่อนไขถ้าระยะห่างจากพิกัดที่กำหนดไว้และพิกัดที่รับมาน้อยกว่า 5,000 เมตร แสดงผลว่าโคนมอยู่ในพื้นที่คือ “Cow in area” แต่ถ้าหากว่าโคนมอยู่ห่างมากกว่า 5,000 เมตร ให้แสดงผลว่าวัวอยู่นอกพื้นที่ “Cow out of area”





ภาพที่ 3.4 ผังการทำงานของ Module GPS

3.3.3 การอธิบายพฤติกรรมโคนม

1. การนอน

โดยจะเริ่มทำงานจากอุปกรณ์ Ultrasonic ส่งคลื่นไปยังพื้นดิน คลื่นจะสะท้อนกลับมาที่ตัวรับคลื่น เพื่อหาระยะห่างจากพื้นดินด้วยความกว้างของสัญญาณพัลส์ แล้วคำนวณด้วยสูตรสมการ

$$d = 0.5 \times \text{อัตราเร็วของเสียง} \times \text{Duration} \quad (3.1)$$

เมื่อ d คือ ระยะห่าง ระหว่างโคนมและพื้น

Duration คือ ระยะเวลาระหว่างการส่งของคลื่นไปกระทบกับพื้น

อัตราเร็วของเสียงมีค่าคงที่ 343 m/s

เมื่อเมื่อคำนวณออกมาแล้วค่า น้อยกว่า 15 c.m. จะเท่ากับนอน หรือถ้ามากกว่า 15 c.m. จะเท่ากับยืน

2. การเดิน

จะใช้ตัวเซนเซอร์ MPU6050 ในการจับการเคลื่อนไหวของโคนมโดยจะใช้ Accelerometer เชิงเส้นของ XYZ แล้วใช้สูตรคำนวณ

$$\text{Acc} = \sqrt{x^2 + y^2 + z^2} \quad (3.2)$$

Acc คือ อัตราเร่งรวมของทั้ง 3 แกน

X คือ อัตราเร่ง ของแกน x

Y คือ อัตราเร่ง ของแกน y

Z คือ อัตราเร่ง ของแกน z

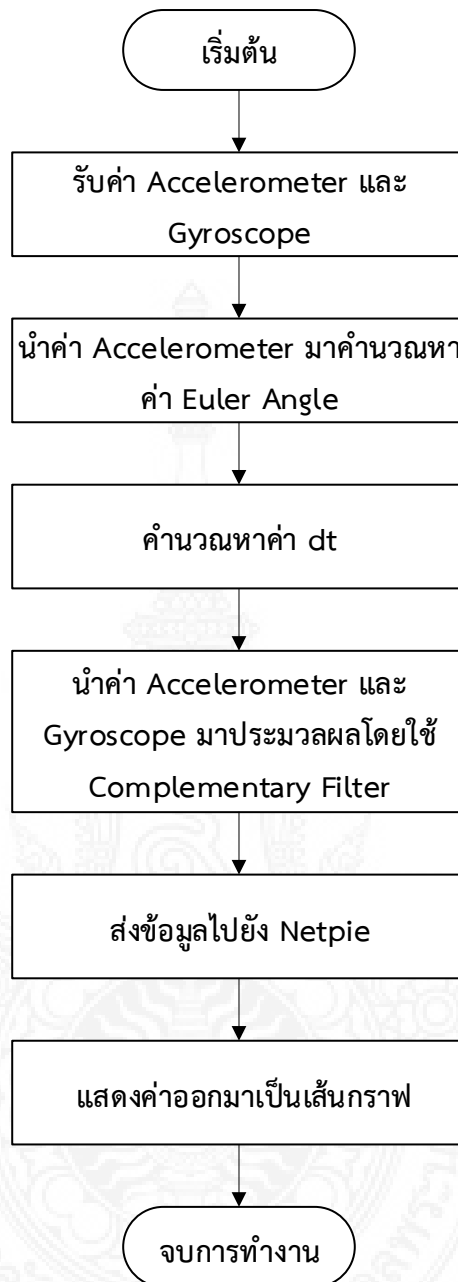
เมื่อคำนวณออกมาแล้วค่า น้อยกว่า 50 m/s². จะเท่ากับยืน หรือถ้ามากกว่า 50 m/s² จะเท่ากับเดิน

3. การยืน

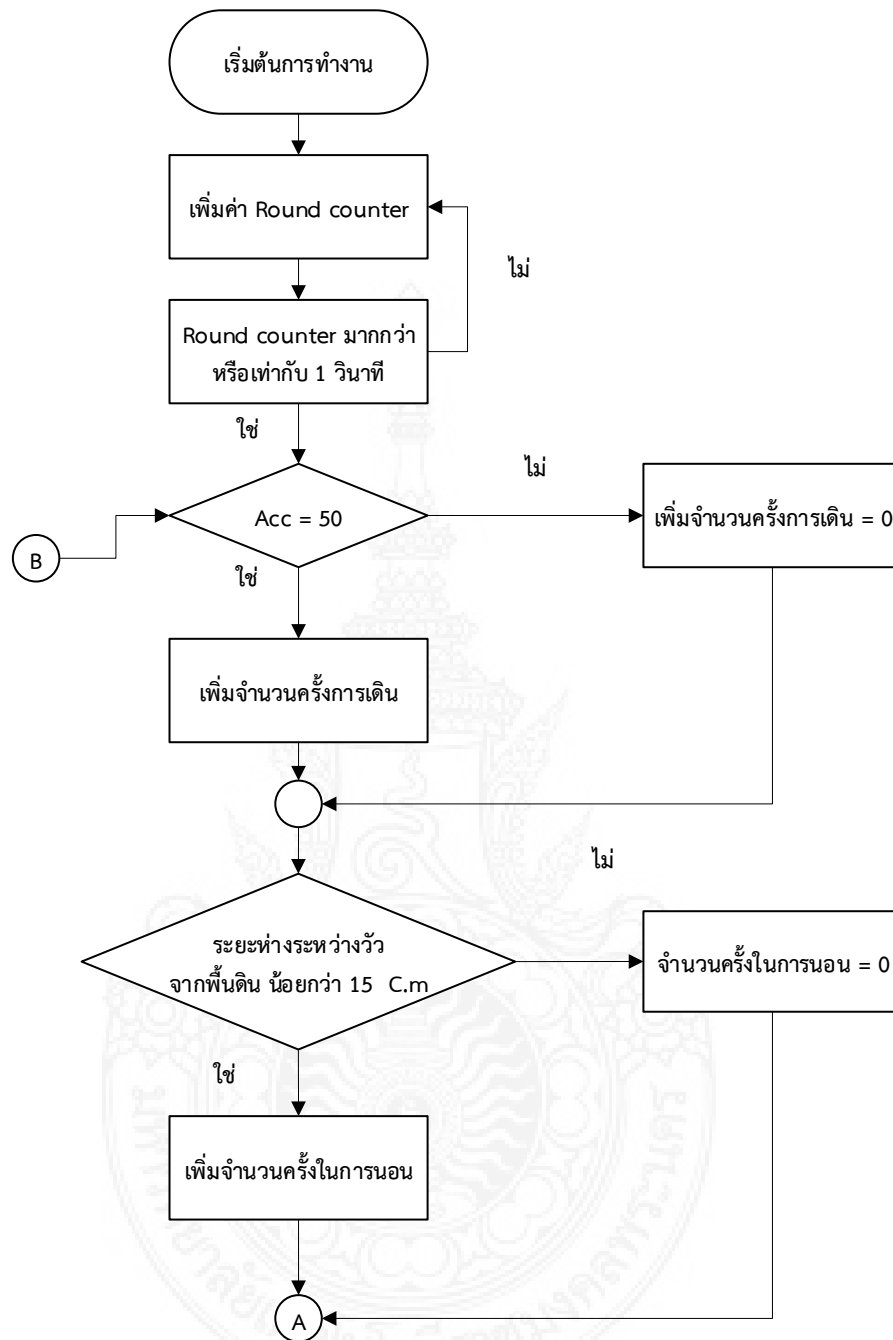
ส่วนในการยืนนั้นได้ทำการนำผลลัพธ์ของทั้ง 2 พฤติกรรมมาเทียบเพื่อระบุว่าโคนมกำลังยืนอยู่หรือไม่ โดย ค่าของการนอน มากกว่า 15 c.m. และ <40 m/s² จะเท่ากับว่าโคนมกำลังยืนซึ่งหลักการทำงาน สูตรการคำนวณ ที่แสดงหรือระบุการเคลื่อนไหวของโคนมดังภาพที่ 3.5 จะบอกหลักการทำงานโดยรวมของการทำงาน mpu6050 และภาพที่ 3.6 ซึ่งจะบอกหลักการคำนวณเป็นขั้นตอน

4. การส่งข้อมูลไปยัง NETPIE

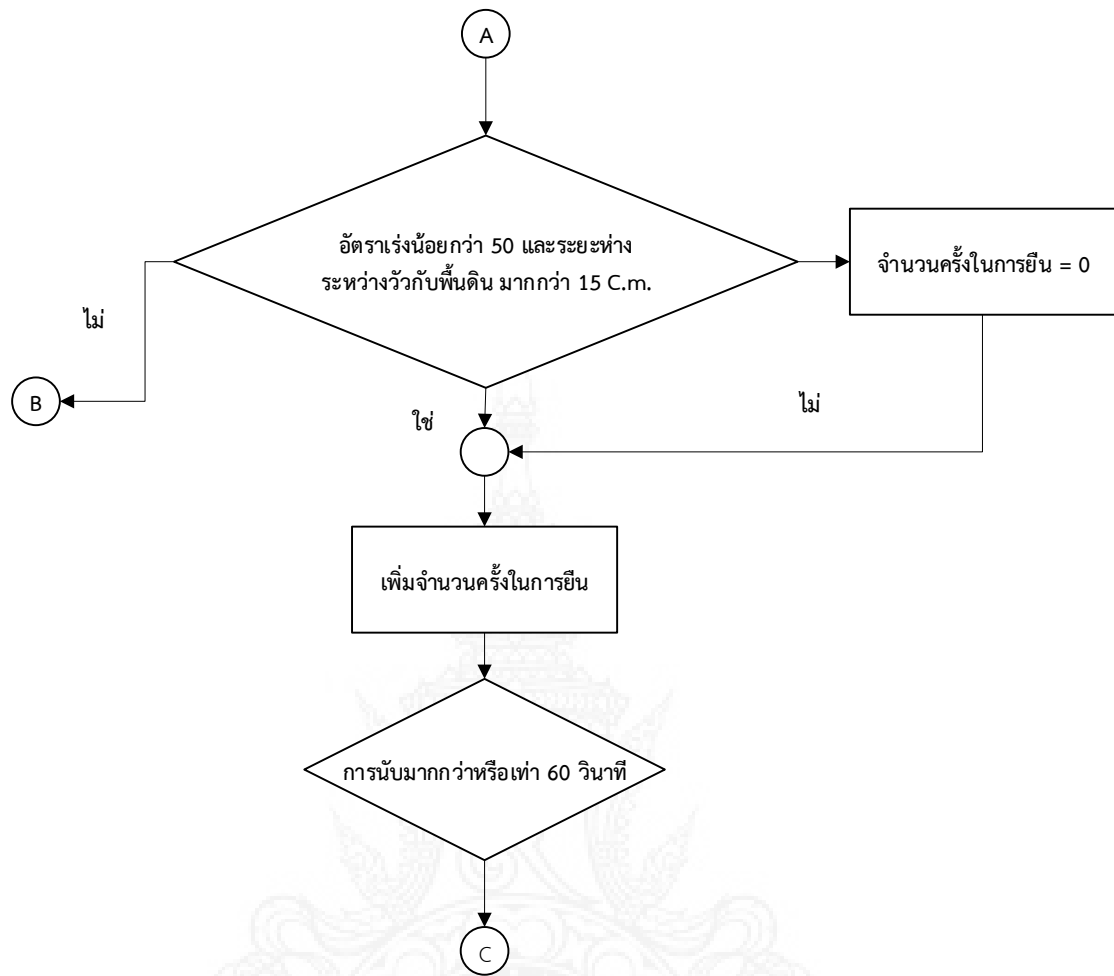
การส่งข้อมูลไปยัง NETPIE เมื่อค่าต่าง ๆ ที่ได้มาจากอุปกรณ์ทั้งสามแล้ว NodeMCUESP8266 จะทำการประมวลผลแล้วส่งข้อมูลไปยัง NETPIE ด้วยสัญญาณ Internet WiFi โดยใช้ Library Microgear ของโปรแกรม Arduino IDE แล้วแสดงผลมายังหน้าจอ



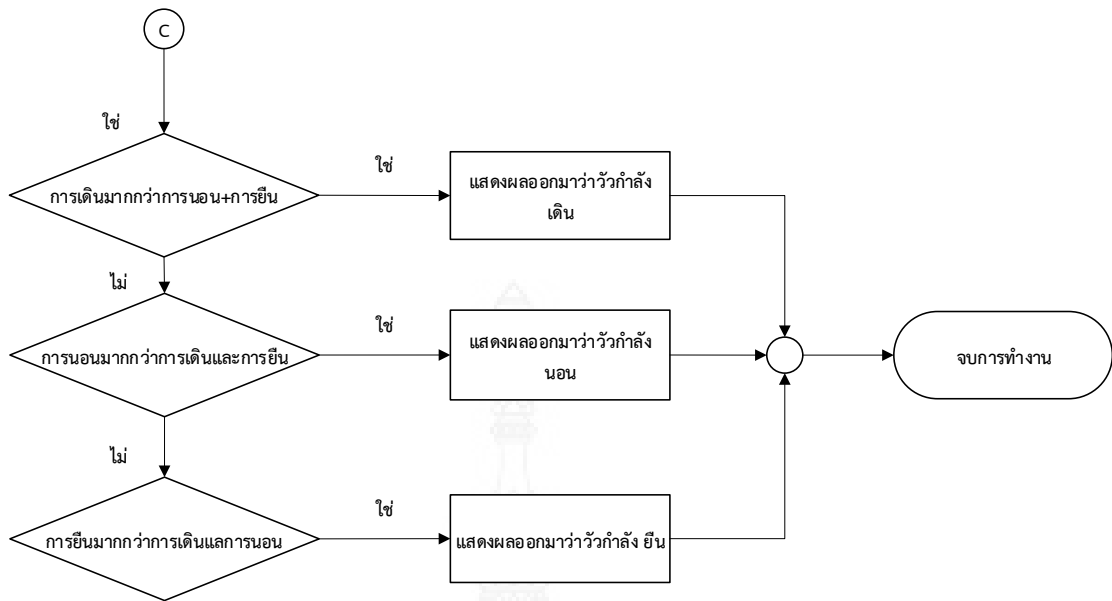
ภาพที่ 3.5 ผังการทำงานของ MPU6050



ภาพที่ 3.6 ผังการทำงานการคำนวณการเคลื่อนไหวของโคนม



ภาพที่ 3.7 ผังการทำงานการคำนวณการเคลื่อนไหวของโคนม (ต่อ1)

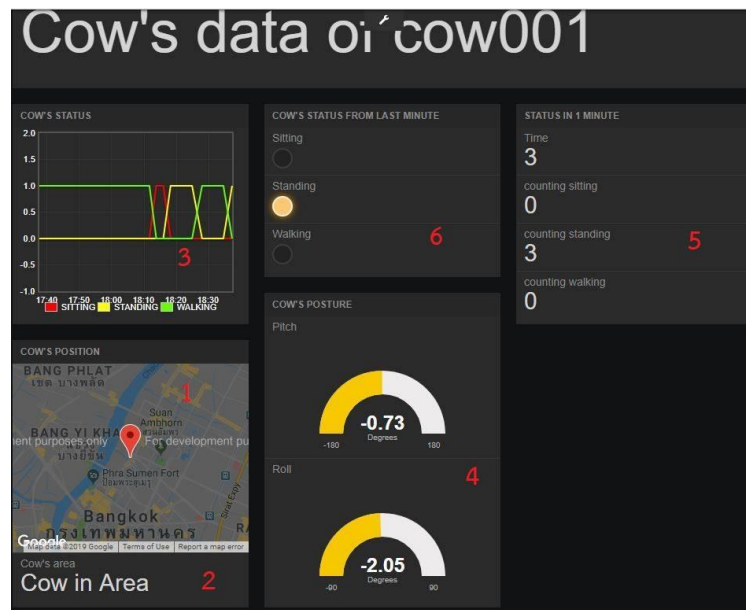


ภาพที่ 3.8 ผังการทำงานการคำนวณการเคลื่อนไหวของโคนม (ต่อ2)

3.3.4 การแสดงผล IOT Platform

ในส่วนของการแสดงผลเมื่อ Node MCUESP8266 ส่งข้อมูลมายัง NETPIE แล้วจะเริ่มทำการเก็บข้อมูลไว้แล้วแสดงผลทุก ๆ 1 นาที ไว้ดังนี้

1. ระบุตำแหน่งของโคนมโดยจะแสดงบน Google map
2. การระบุขอบเขตในการอยู่พื้นที่ของโคนมว่าอยู่ในพื้นที่หรือไม่
3. การแสดงสถานะพฤติกรรมโคนมโดยจะแสดงเป็นเส้นกราฟและจะประมวลผลว่าโคนมกำลัง นอน ยืน หรือเดิน
4. การแสดงค่าการเคลื่อนไหวมุม Pitch กับมุม Roll
5. การแสดงการนับจำนวนครั้งในการเคลื่อนไหวภายใน 1 นาที โดยจะนับทุก 1 วินาที จะมีทั้งหมดสามพฤติกรรม การนอน การเดิน การยืน
6. การระบุพฤติกรรมของโคนมหลังประมวลผลภายใน 1 นาที



ภาพที่ 3.9 การแสดงผลบน NETPIE



บทที่ 4

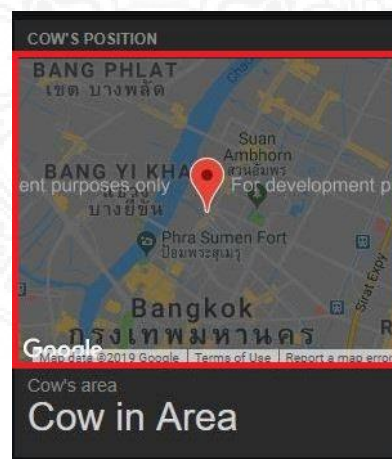
ผลการดำเนินงาน

ขั้นตอนการดำเนินงานการพัฒนาอุปกรณ์สวมใส่เพื่อเก็บข้อมูลการเคลื่อนไหวของวัวโดยใช้เทคโนโลยีไอโอที ในบทที่ 3 ได้มีการทดสอบที่มหาวิทยาลัยเทคโนโลยีราชมงคลพระนครโดยใช้โดยทหารใช้มนุษย์ในการเลียนแบบพฤติกรรมของโคนมในการทดสอบเบื้องต้นทำให้ผลทดสอบในออกมาดังนี้

- 4.1 การแสดงบนแผนที่
- 4.2 การระบุขอบเขตในการอยู่พื้นที่ของโคนม
- 4.3 การนอน
- 4.4 การเดิน
- 4.5 การยืน

4.1 การแสดงบนแผนที่

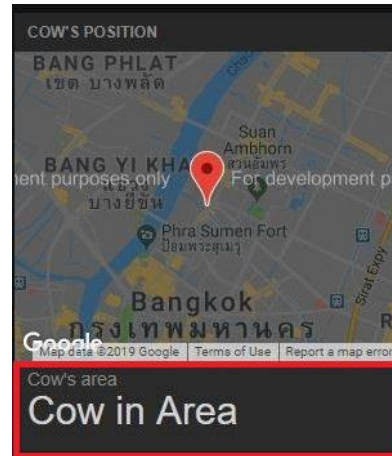
การระบุตำแหน่งของบนแผนที่สรุปได้ดังนี้เมื่อ Module GPS ระบุตำแหน่งพิกัดของ Latitude และ Longitude คือ 13.770538, 100.504448 ผลที่ได้จะแสดงตำแหน่งปัจจุบันบน Google ดังภาพที่ 4.1



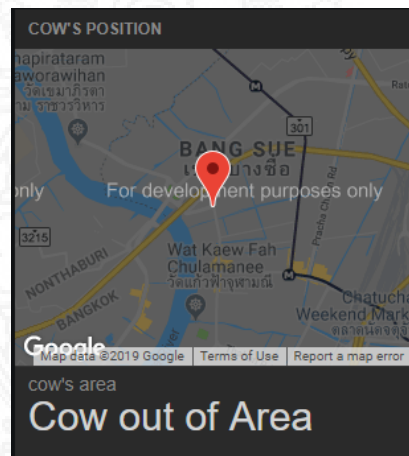
ภาพที่ 4.1 การระบุตำแหน่ง

4.2 การระบุขอบเขตในการอยู่พื้นที่ของโคนม

หลังจากที่ระบุตำแหน่งแล้วเมื่ออยู่ในขอบเขตของพื้นที่ที่กำหนดไว้ในโปรแกรม จะแสดงข้อความ “ Cow in Area “ ดังภาพที่ 4.2 ถ้าไม่ได้อยู่ในขอบเขตพื้นที่ที่จะแสดงผลทางข้อความว่า “Cow out of Area” ดังภาพที่ 4.3



ภาพที่ 4.2 การระบุขอบเขตในการอยู่พื้นที่



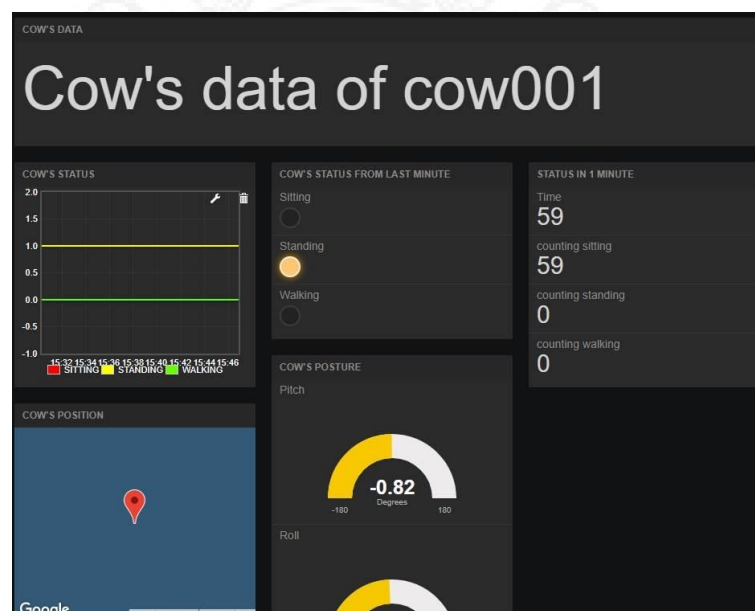
ภาพที่ 4.3 การระบุขอบการออกนอกขอบเขตในการอยู่พื้นที่

4.3 การนอน

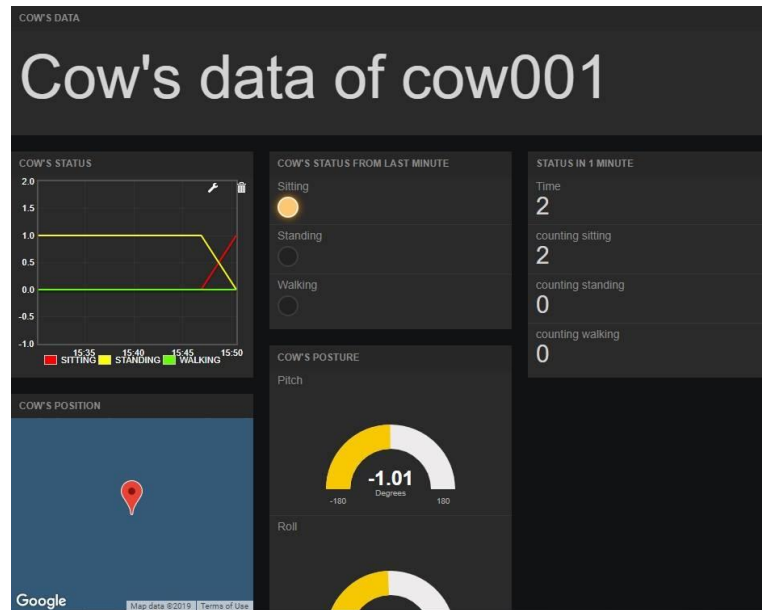
เมื่อทำการเลียนแบบพฤติกรรมการนอนดังภาพที่ 4.3 ค่าที่รับจาก Ultrasonic และ Mpu6050 นำมาประมวลผลภายใน 1 นาที ผลที่ได้คือจำนวน Counting Sitting มีค่าเท่ากับ 59 ดังภาพที่ 4.4 มีค่ามากที่สุดและการวัดระยะห่างที่วัดได้น้อยกว่า 15 เซนติเมตร จึงระบุว่ากำลังนอน ดังภาพที่ 4.5



ภาพที่ 4.3 การนอน



ภาพที่ 4.4 ก่อนการแสดงผลการนอน



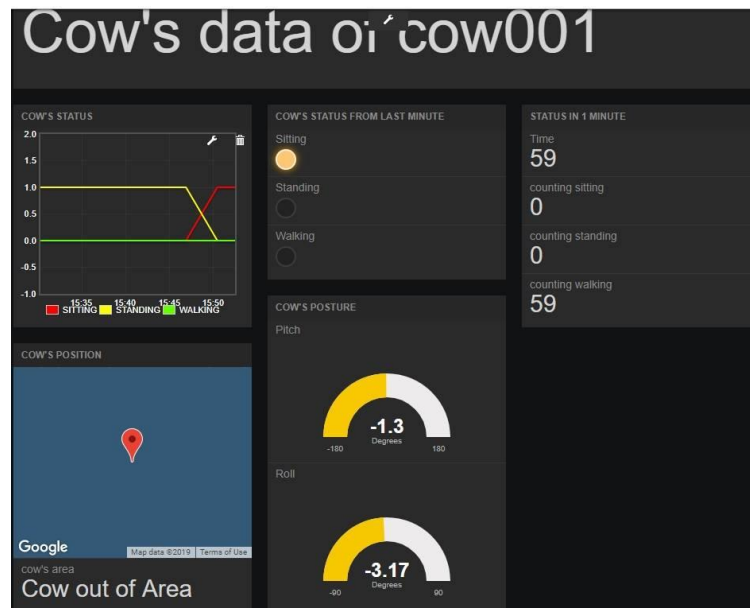
ภาพที่ 4.5 แสดงผลการนอน

4.4 การเดิน

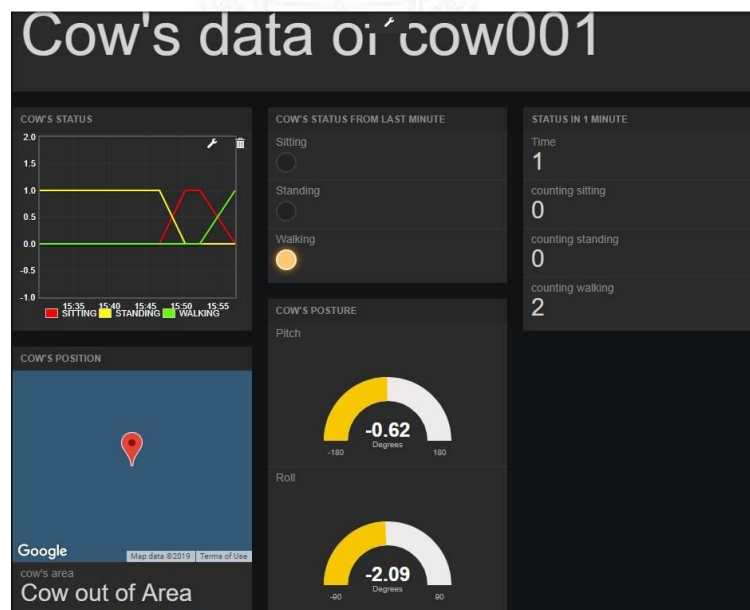
เมื่อทำการเลียนแบบพฤติกรรมกรรมการเดินดังภาพที่ 4.6 ค่าที่รับจาก Ultrasonic และ Mpu6050 นำมาประมวลผลภายใน 1 นาที ผลที่ได้คือจำนวน Counting Walking มีค่าเท่ากับ 59 ภาพที่ 4.7 มีค่ามากที่สุดและการวัดระยะห่างที่วัดได้มากกว่า 15 เซนติเมตร ผลที่ได้จะระบุที่กำลังเดินดังภาพที่ 4.8



ภาพที่ 4.6 การเดิน



ภาพที่ 4.7 ก่อนการแสดงผลการเดิน



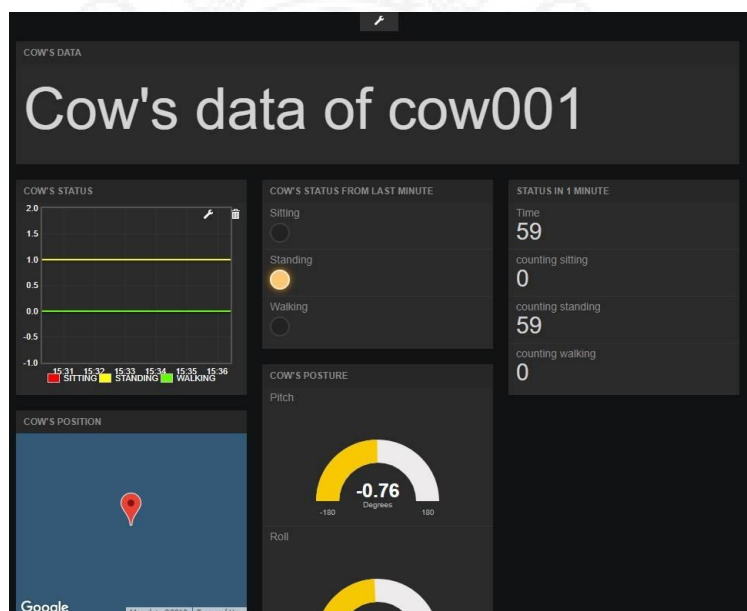
ภาพที่ 4.8 แสดงผลการเดิน

4.5 การยืน

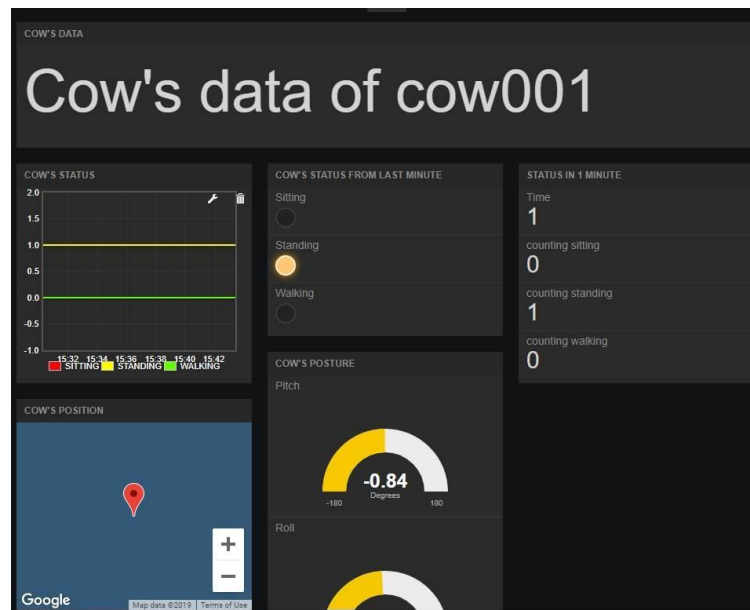
เมื่อทำการเลียนแบบพฤติกรรมกรรมการเดินดังภาพที่ 4.9 ค่าที่รับจาก Ultrasonic และ Mpu6050 นำมาประมวลผลภายใน 1 นาที ผลที่ได้คือจำนวน Counting Standing มีค่าเท่ากับ 59 ภาพที่ 4.10 มีค่ามากที่สุดและการวัดระยะห่างที่วัดได้มากกว่า 15 เซนติเมตร ผลที่ได้จะระบุว่าการยืนดังภาพที่ 4.11



ภาพที่ 4.9 การยืน



ภาพที่ 4.10 ก่อนการแสดงผลการยืน



ภาพที่ 4.11 แสดงผลการยื่น



บทที่ 5

สรุปผล ปัญหาและข้อเสนอแนะ

การพัฒนาอุปกรณ์สวมใส่เพื่อเก็บข้อมูลการเคลื่อนไหวของวัวโดยใช้เทคโนโลยีไอโอที (Development of Wearable Device for Logging Cow Behavior using IoT Technology) โดยสรุปปัญหาและข้อเสนอแนะโครงการ ดังนี้

5.1 สรุปผลโครงการ

5.2 ปัญหาและแนวทางการแก้ไข

5.3 ข้อเสนอแนะในการพัฒนาโครงการ

5.1 สรุปผลโครงการ

โครงการปริญญาโทพัฒนาอุปกรณ์สวมใส่เพื่อเก็บข้อมูลการเคลื่อนไหวของวัวโดยใช้เทคโนโลยีไอโอที (Development of Wearable Device for Logging Cow Behavior using IoT Technology) สามารถบันทึกข้อมูลการเคลื่อนไหวและตำแหน่งของโคนมส่งไปยัง Netpie ที่เป็น IoT Platform โดยใช้เซ็นเซอร์จับการเคลื่อนไหว Accelerometer และ Gyro Scope และใช้โมดูล GPS ในการระบุตำแหน่งที่อยู่ของโคนม เพื่อเป็นการลดภาระของผู้เชี่ยวชาญที่ต้องใช้ทั้งจำนวน คน เวลา ในการสังเกตพฤติกรรม อีกทั้งยังเป็นการไม่รบกวนใช้ชีวิตประจำวันของโคนม เพื่อให้โคนมสุขภาพ และ อารมณ์ที่ดีจึงได้นำมโคที่มีคุณภาพ มาบริโภคต่อไป

การพัฒนาอุปกรณ์สวมใส่เพื่อเก็บข้อมูลการเคลื่อนไหวของวัวโดยใช้เทคโนโลยีไอโอที (Development of Wearable Device for Logging Cow Behavior using IoT Technology) เริ่มจากศึกษาข้อมูลที่เกี่ยวข้องกับพฤติกรรมของโคนมและอุปกรณ์ที่ใช้ในการทำโครงการนี้แล้วนำข้อมูลที่ศึกษามาทั้งหมดมาสรุปและเริ่มทำการออกแบบชิ้นงานและระบบโปรแกรมการทำงาน หลังจากนั้นจะมีการทดสอบชิ้นงานและปรับปรุงแก้ไข และพัฒนาจนเสร็จสิ้น

การทดสอบโดยใช้ module GPS ในการระบุตำแหน่งและขอบเขตการอยู่ในพื้นที่ Mpu6050 ใช้ในการวัดความเร่ง Accelerometer ตามเชิงเส้น XYZ และ Gyroscope จะเป็นการวัดความเร็วเชิงมุมของ XYZ และ ultrasonic ใช้ในการวัดระยะห่างระหว่างพื้นกับโคนม เมื่อนำค่าที่ได้มาทั้งหมดมาประมวลผลออกมาเป็นการระบุพฤติกรรมของโคนม

5.2 ปัญหาและแนวทางการแก้ไข

ปัญหาที่เกิดขึ้นระหว่างการดำเนินงาน เป็นปัญหาที่จำเป็นต้องได้รับการแก้ไข โดยจะมีลักษณะของปัญหาและวิธีการแก้ไขดังต่อไปนี้

5.2.1 การจับเคลื่อนไหวของ Accelerometer และ Gyro Scope ยังมีความคาดเคลื่อน ต้องปรับปรุงในส่วนของอุปกรณ์และโปรแกรม

5.2.2 สัญญาณระบุตำแหน่งของ GPS ยังมีบางช่วงเวลาที่ยับสัญญาณ จึงไม่สามารถระบุตำแหน่งได้

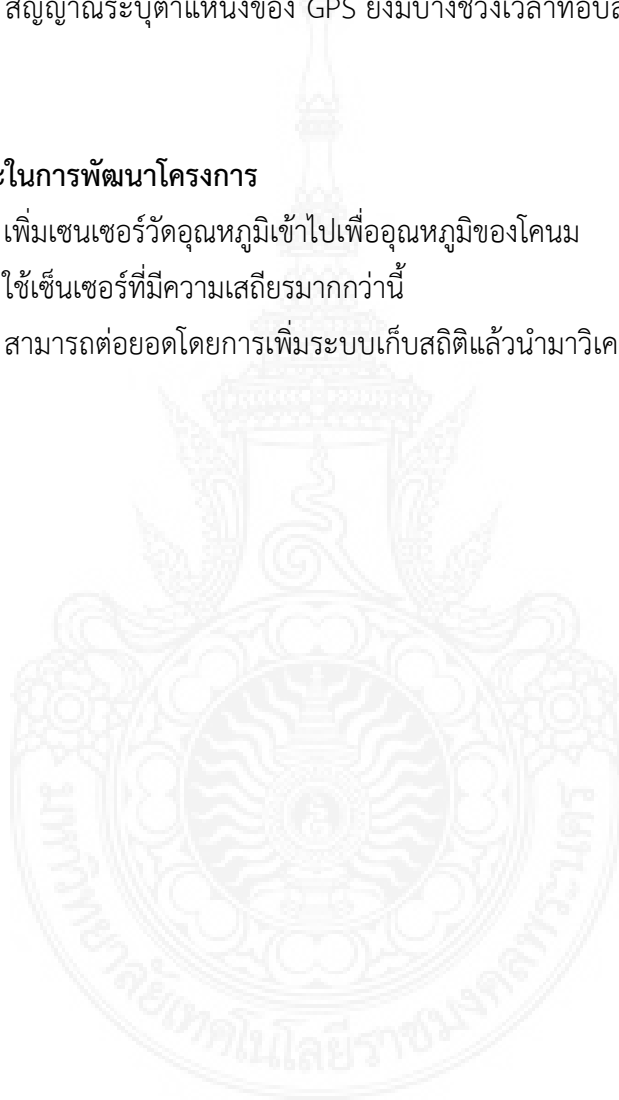
5.3 ข้อเสนอแนะในการพัฒนาโครงการ

5.3.1 เพิ่มเซนเซอร์วัดอุณหภูมิเข้าไปเพื่ออุณหภูมิของโคนม

5.3.2 ใช้เซ็นเซอร์ที่มีความเสถียรมากกว่านี้

5.3.3 สามารถต่อยอดโดยการเพิ่มระบบเก็บสถิติแล้วนำมาวิเคราะห์เพื่อวินิจฉัยพฤติกรรม

โคนม



บรรณานุกรม

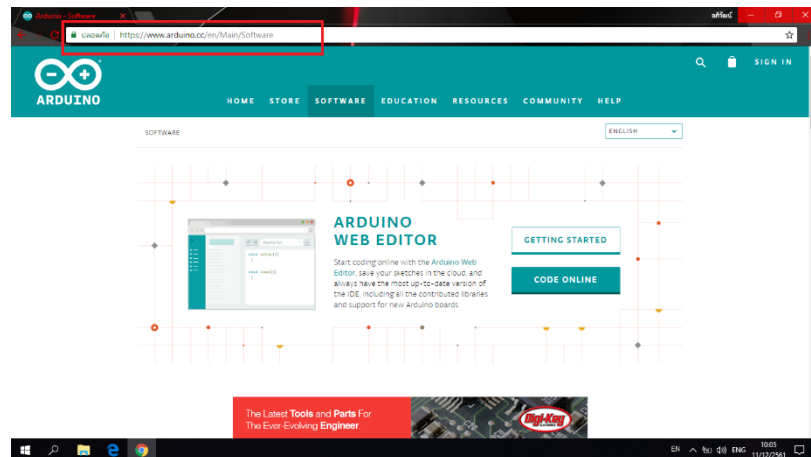
- จชฎา ขจรฤทธิ์, ปิยนุช ชัยพรแก้ว และ หนึ่งฤทัย เอ็งฉ้วน. (2560). การประยุกต์ใช้เทคโนโลยี Internet of Things ในการควบคุมระบบส่องสว่างสำหรับบ้านอัจฉริยะ. **วารสารวิทยาการและเทคโนโลยีสารสนเทศ**, 7(1), หน้า 1.
- สุธาสินี นพฤทธิ์, อนุพล พันธุ์วงศ์ และ กิติ์สุชาติ พสุภา. (2557). การวิเคราะห์พฤติกรรมโคนมจากภาพวิดีโอ. **วารสารเทคโนโลยีสารสนเทศลาดกระบัง**, 3(2), หน้า 1.
- อรสา บุญมีรติ และ จำเริญ เทียงธรรม. (2556). การสังเกตพฤติกรรมโครีดนมในช่วงวันโดยเทคนิคการสุ่มรายตัว. **วารสารวิทยาศาสตร์เกษตร**, 44(1), หน้า 203.
- Grothmann et al. (2014). Influence of Different Feeding Frequencies on the Rumination and Lying Behavior of Dairy Cows. International Conference
- เนติยะ ยอดเนตร, “การสังเกตพฤติกรรมโคป่วย,” [Online]. Available: http://www.thaillivestock.com/cattle_handling/อาการหรือลักษณะที่สังเกตได้ง่ายๆ-มีดังนี้. [Accessed: 6 ธันวาคม 2561]
- สนธยา นงนุช, “ESP8266 ตอนที่ 1 รู้จักกับ ESP และรุ่นที่นิยมใช้งาน,” [Online]. Available: www.ioxhop.com/article/13/esp8266-ตอนที่1-รู้จักกับ-esp-และรุ่นที่นิยมใช้งาน. [Accessed: 6 ธันวาคม 2561]
- Yatinan Thanwongsa, “Accelerometer 198210 LINEAR CIRCUIT ANALYSIS,” [Online]. Available: <http://198210-accelerometer2015.blogspot.com/2015/09/>. [Accessed: 6 ธันวาคม 2561]
- Thaimicrotron.com "การเชื่อมต่ออุปกรณ์แบบ I2C (I²C)," [Online]. Available: www.thaimicrotron.com/CCS-628/Reference/I2CBUS.htm. [Accessed: 25 มกราคม 2562]
- สมเกียรติ กิจวงศ์วัฒน์, “[Android Code] การใช้งาน Gyroscope,” [Online]. Available: <http://www.akexorcist.com/2013/03/android-code-gyroscope.html>. [Accessed: 6 ธันวาคม 2561]
- บริษัท โกลบอลไฟว์ จำกัด, "ความรู้ทั่วไปเกี่ยวกับ GPS," [Online]. Available: www.global5thailand.com/thai/gps.htm. [Accessed: 6 ธันวาคม 2561]
- IOXhop, "การใช้งานโมดูล GPS Ublox NEO-6M," [Online]. Available: www.ioxhop.com/aritclt/54/การใช้งานโมดูล-gps-ublox-neo-6m. [Accessed: 25 มกราคม 2562]
- บริษัท คอมโพแม็กซ์ จำกัด, "Ultrasonic คือ..," [Online]. Available: <http://www.compomax.co.th/product/ultrasonic-theory>. [Accessed: 25 มกราคม 2562]

ภาคผนวก ก
การติดตั้งโปรแกรม Arduino IDE



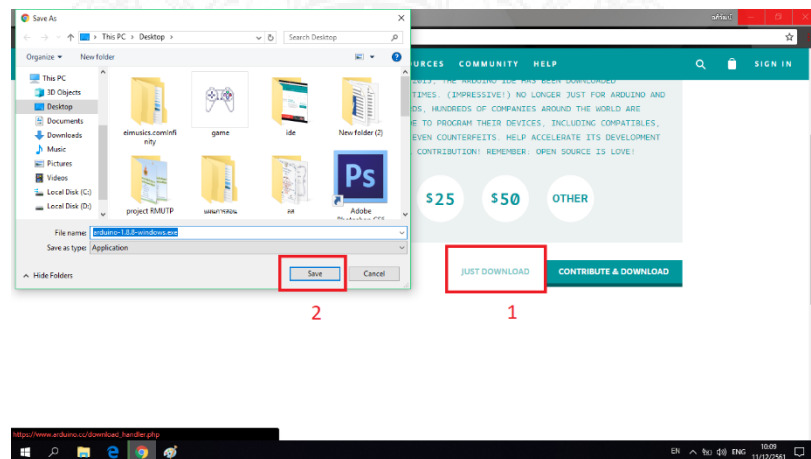
การติดตั้งโปรแกรม Arduino IDE

1) เข้าไปที่เว็บไซต์ <https://www.arduino.cc/en/main/software> เพื่อทำการดาวน์โหลดโปรแกรม โดยให้เลือก Windows Installer, for Windows XP and up ที่เป็นของระบบปฏิบัติการวินโดวส์



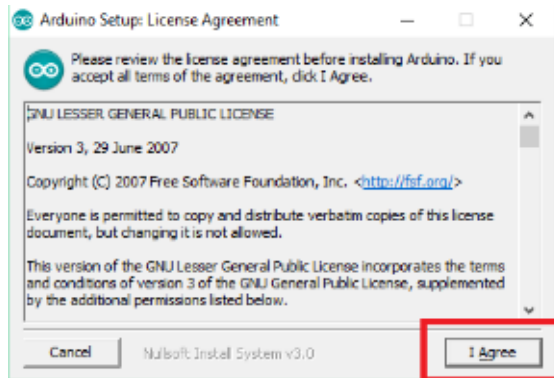
ภาพที่ ก.1 เว็บไซต์ดาวน์โหลด Arduino IDE

2) กดที่ปุ่ม just Download หลังจากนั้นจะมีหน้าต่างขึ้นมาให้เลือกตำแหน่งที่จะดาวน์โหลด แล้วกดปุ่ม save เพื่อทำการดาวน์โหลด



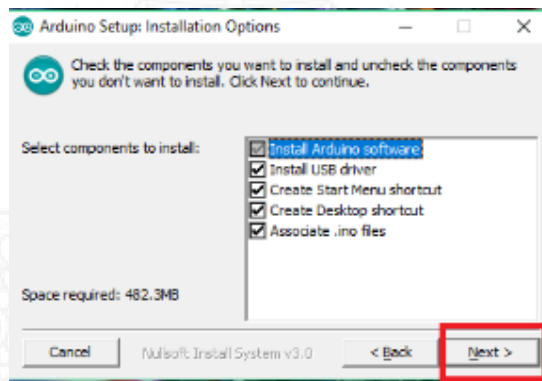
ภาพที่ ก.2 แสดงการกดดาวน์โหลดและเลือกที่บันทึก

3) ให้กดปุ่ม I Agree เพื่อเป็นการยอมรับข้อตกลงของโปรแกรม



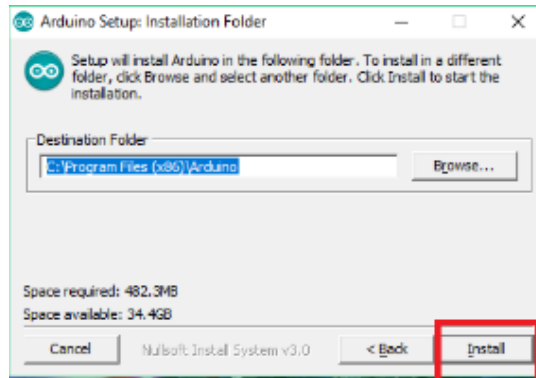
ภาพที่ ก.3 แสดงการกดปุ่มยืนยันการติดตั้ง

4) ให้กดปุ่ม NEXT เพื่อไปยังขั้นตอนถัดไป



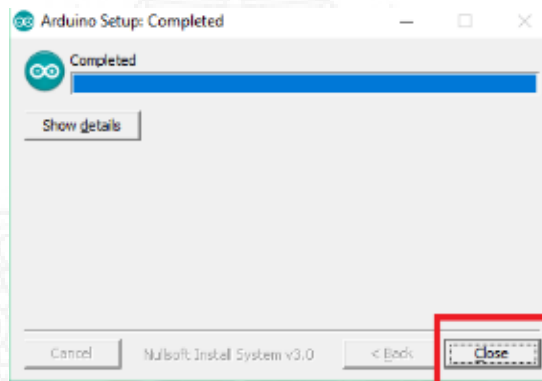
ภาพที่ ก.4 แสดงการเลือกโปรแกรมที่จะติดตั้ง

5) กดปุ่ม Install เพื่อทำการติดตั้งโปรแกรม



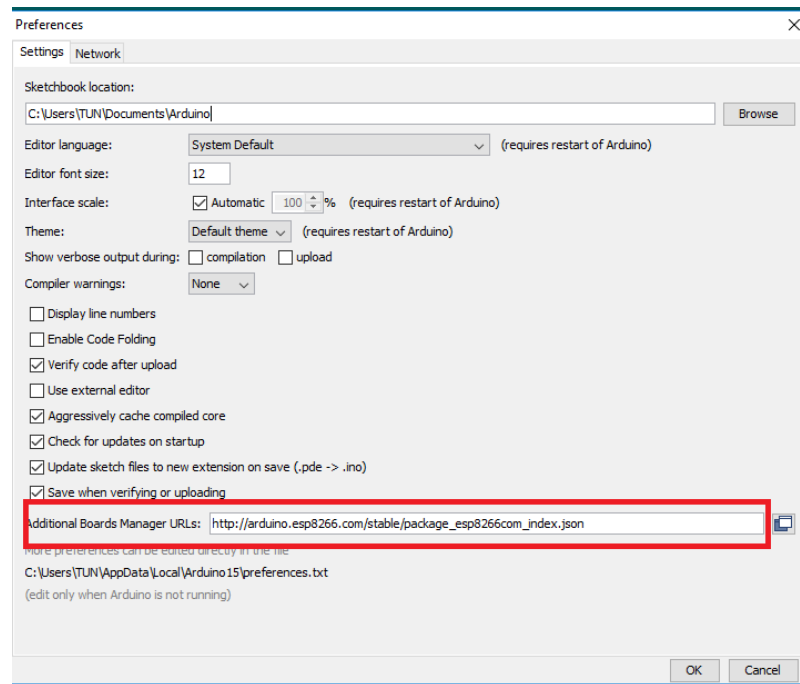
ภาพที่ ก.5 แสดงการเลือกไดรฟ์ที่จะติดตั้งโปรแกรม

6) กดปุ่ม Close เมื่อโปรแกรมติดตั้งเสร็จ



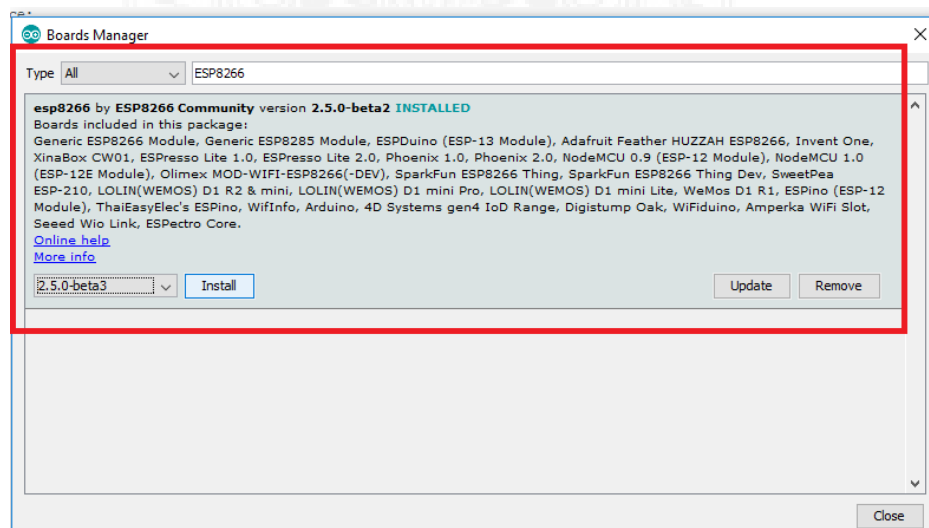
ภาพที่ ก.6 แสดงเมื่อติดตั้งเสร็จเรียบร้อย

7) เมื่อเปิดโปรแกรมขึ้นมาให้ไปที่ เมนู File > Preferences แล้วก็อปข้อความลิงค์นี้
http://arduino.esp8266.com/stable/package_esp8266com_index.json ใส่ลงไปในช่วง
 Additional Boards Managers URLs หลังจากนั้นให้กดปุ่ม OK



ภาพที่ ก.7 แสดงการตั้งค่าในโปรแกรม(1)

8) ให้ไปที่เมนู Tool > Boards : > Boards Manager... ให้พิมพ์ ESP8266 ลงไปในช่องว่าง
 หลังจากนั้นให้กดปุ่ม Install เพื่อติดตั้ง Boards ESP8266



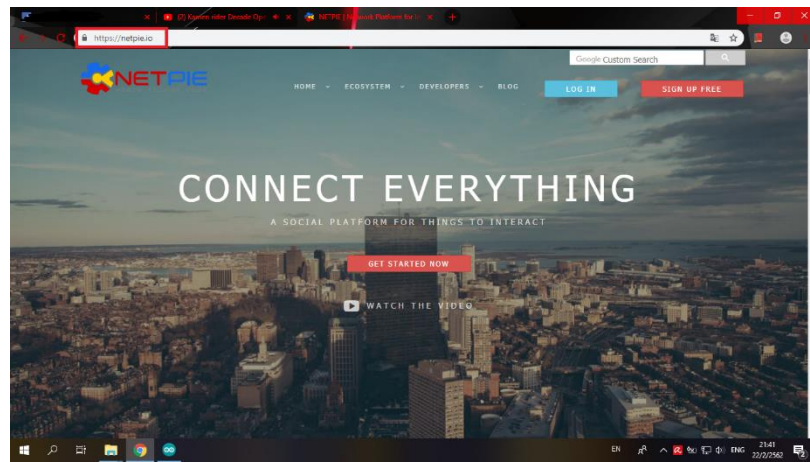
ภาพที่ ก.8 แสดงการตั้งค่าในโปรแกรม(2)

ภาคผนวก ข
การสมัคร NETPIE



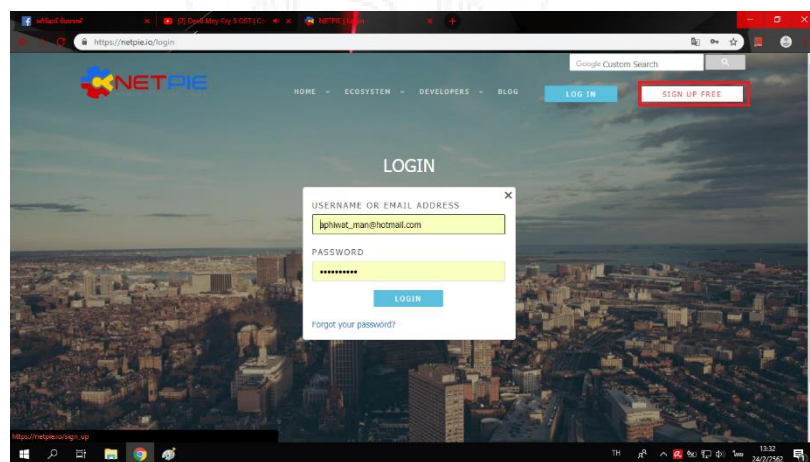
ขั้นตอนการสมัคร NETPIE

- 1) เข้าไปที่เว็บไซต์ <https://netpie.io/login>



ภาพที่ ข.1 เว็บไซต์ NETPIE

- 2) เข้าไปที่ Sing up free



ภาพที่ ข.2 กดปุ่มการสมัคร

3) กรอกข้อมูลการสมัครให้ครบ

CREATE AN ACCOUNT

EMAIL
required

NAME
required

ORGANIZATION
required

COUNTRY CODE
Thailand (+66)

MOBILE PHONE NUMBER* (NO COUNTRY CODE)
required and number only

I agree to the Privacy Statement and Terms of Use

SIGN UP

* Password will be sent to your mobile phone number.

ภาพที่ ข.3 กรอกข้อมูลของผู้ใช้งาน

4) กรอก Email และ Password แล้ว Login

LOGIN

USERNAME OR EMAIL ADDRESS
aphwat_rnan@hotmail.com

PASSWORD

LOGIN

Forgot your password?

ภาพที่ ข.4 Log in เข้าใช้งาน

ภาคผนวก ค
คู่มือการใช้งาน



1. ส่วนของการใช้อุปกรณ์

1) เปิดสวิตช์ที่ตัวอุปกรณ์



ภาพที่ ค.1 เปิดสวิตช์ที่ตัวอุปกรณ์

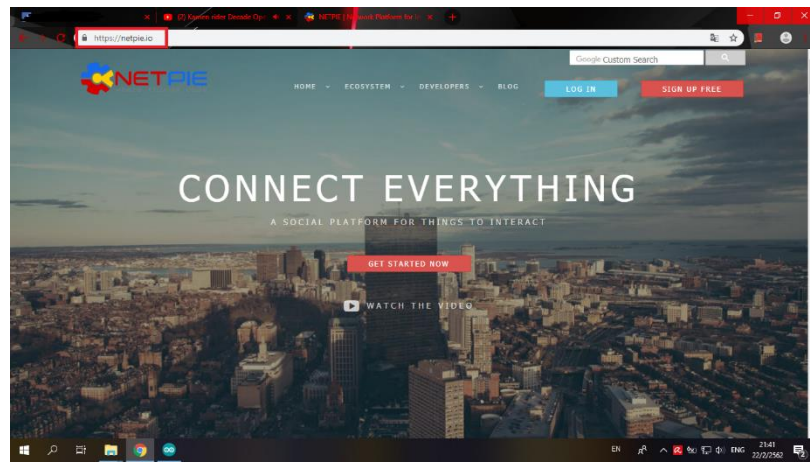
2) สวมใส่อุปกรณ์



ภาพที่ ค.2 สวมใส่อุปกรณ์

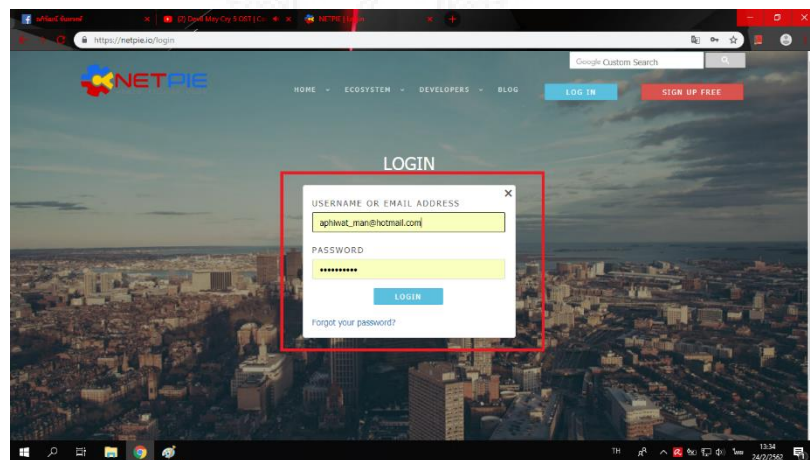
2. ส่วนเข้าไปดูการแสดงผลและการบันทึกข้อมูลบนNETPIE

3) เมื่อต้องการดูการแสดงผลบน NETPIE ให้เข้าไปที่เว็บไซต์ <https://netpie.io/login>



ภาพที่ ค.3 เข้าเว็บไซต์ NETPIE

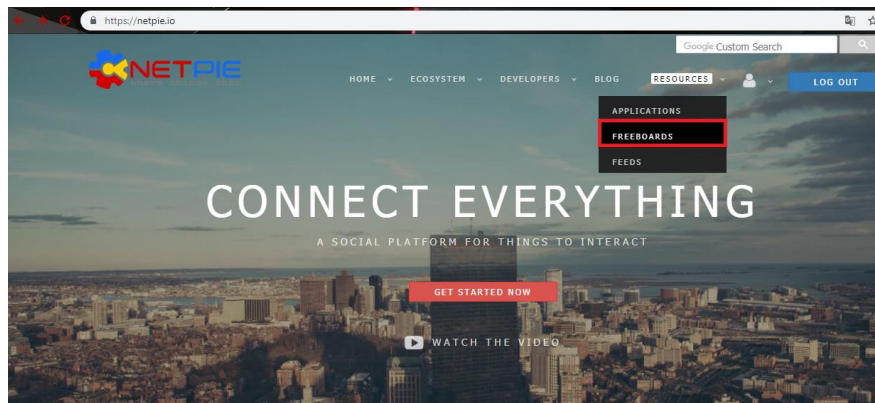
4) กรอก Email และ Password แล้ว Login



ภาพที่ ค.4 Log in เข้าใช้งาน

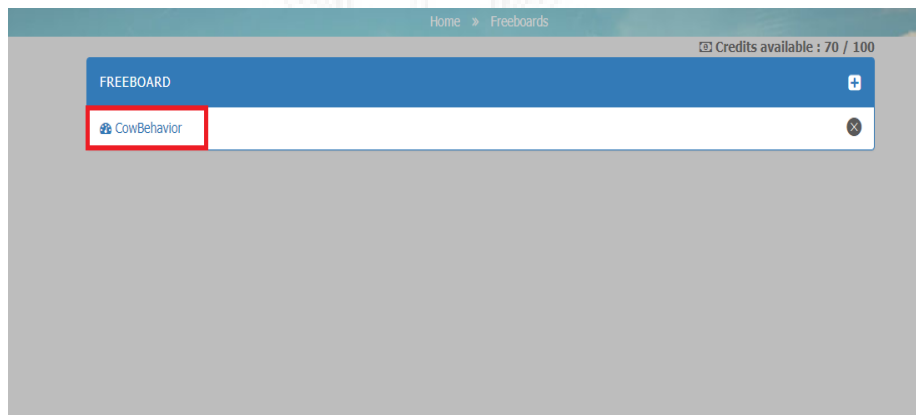
3. ส่วนของการดูการแสดงผลแบบ Real Time บน FEEDBOARD

5) เข้าไปที่ FREEBOARDS



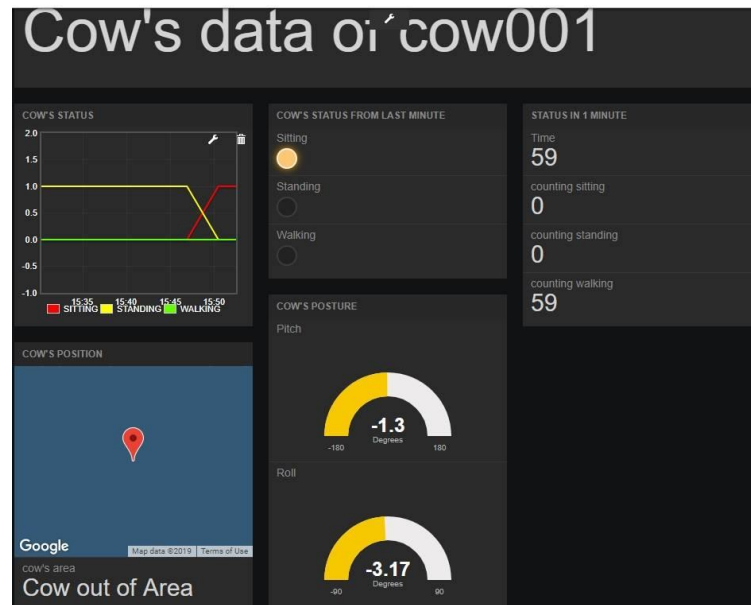
ภาพที่ ค.5 เข้าไปที่ FREEBOARDS

6) เข้าไปที่ CowBehavior



ภาพที่ ค.6 เลือก FREEBOARD ที่ชื่อ CowBehavior

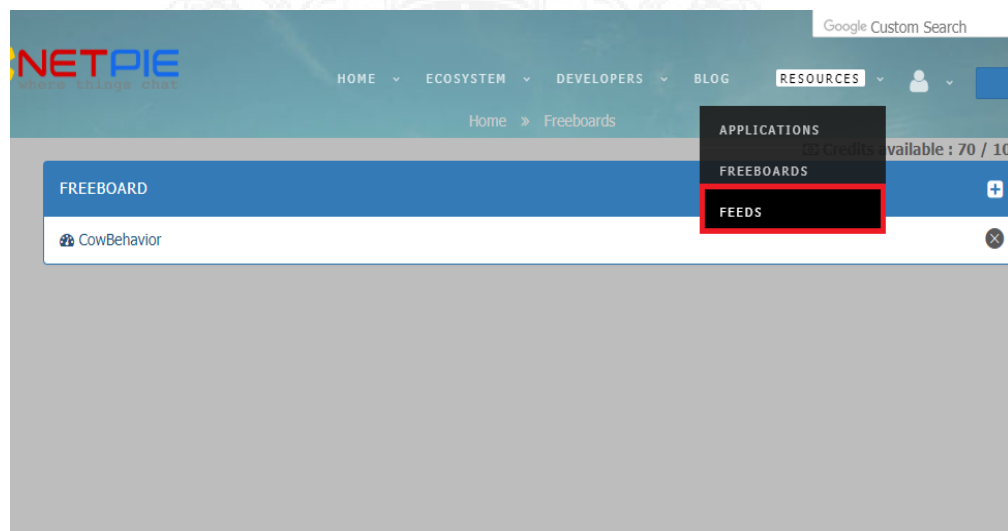
7) การแสดงผลแบบ Real Time



ภาพที่ ค.7 หน้าจอแสดงผลพฤติกรรมของโคนม

4. ส่วนของการดูการแสดงผลที่บันทึกไว้บน FEED

8) เข้าไปที่ FEED



ภาพที่ ค.8 เลือก FEEDS ที่ชื่อ CowBehavior

9) ดูการแสดงผลที่บันทึกไว้

NETPI

HOME ECOSYSTEM DEVELOPERS BLOG RESOURCES LOG OUT

Home Feeds Feed Configuration

cow001 General Info Permission Data Display Credits available : 70 / 100

DESCRIPTION : ...Unspecified... TAGS : ...Unspecified... LOCATION : ...Unspecified... EDIT

Walking
Type : number
Unit : N/A
07:00:00 WALKING

Sitting
Type : number
Unit : N/A
07:00:00 SITTING

Standing
Type : number
Unit : N/A
07:00:00 STANDING

ภาพที่ ค.9 หน้าจอแสดงผลที่บันทึกไว้



ภาคผนวก ง
Code Program



ตาราง ง.1 Code โปรแกรมการทำงาน

ลำดับ	Code
1	#include <ESP8266WiFi.h>
2	#include <MicroGear.h>
3	#include "Wire.h"
4	#include "TinyGPS++.h"
5	
6	const char* ssid = "XXXXXXXXXX"; //ชื่อ WiFi
7	const char* password = "XXXXXXXXXX"; //รหัส WiFi
8	
9	#define APPID "ProjectCow"
10	#define KEY "mG093FgToXH1Hjc"
11	#define SECRET "KezRn4qQEc12nMKDUA7oTEwks"
12	
13	#define ALIAS "CowBehavior"
14	#define FEEDID "cow001"
15	
16	//#define INTERVAL 60000
17	//#define T_INCREMENT 200
18	//#define T_RECONNECT 5000
19	#define BAUD_RATE 9600
20	
21	TinyGPSPlus gps;
22	const uint8_t MPU = 0x68;
23	// I2C Address I2C
24	const float Acc_Scale = 16384.0;
25	// Scale Acc +-2g
26	const float Gyro_Scale = 131.1;
27	// Scale Gyro +-250 °/s
28	const float Alpla_Cp = 0.02;
29	// ค่าคงที่ของ Complementary filter

```

30 int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;
31 // เก็บค่าดิบความเร่งเชิงเส้นและความเร่งเชิงมุมแกน x, y, z
32 float AcX_Scale, AcY_Scale, AcZ_Scale, GyX_Scale, GyY_Scale, GyZ_Scale;
33 // เก็บค่าความเร่งเชิงเส้นและความเร่งเชิงมุมที่ผ่านที่ผ่านการหาร scale แล้ว โดย
34 หน่วยที่ได้จะเป็น g และ °/s ตามลำดับ
35 float AcX_Conv, AcY_Conv , AcZ_Conv;
36 // เก็บค่าความเร่ง3แกนโดย นำความเร่งเชิงเส้น*กับความเร่งโน้มถ่วง จะได้หน่วย
37 m/s2
38 float Euler_Angle[2];
39 // เก็บตัวแปรจากการหา euler angle โดยมีมุม roll และ pitch
40 float Comp_Angle[3];
41 // เก็บค่ามุมที่ผ่าน complementary filter แล้ว
42 float Lati2, Longi2;
43 const float Lati_Org = 13.770695;
44 // ตั้งค่าพิกัด Latitude ที่เป็นจุดกึ่งกลาง
45 const float Longi_Org = 100.503710;
46 // ตั้งค่าพิกัด Longitude ที่เป็นจุดกึ่งกลาง
47 const float Earth_Radius = 6371000.0;
48 // เก็บค่ารัศมีของโลก หน่วยเป็น metre
49 const float Earth_Gravity = 9.807;
50 // เก็บค่าแรงโน้มถ่วงของโลก หน่วยเป็น m/s2
51 float Distance;
52 bool cow_area;
53 String values = "";
54 // เก็บค่าทั้งหมดที่คำนวณมาแล้ว
55
56 long time_prev ;
57 // เวลาในลูปรอบก่อนหน้า
58 float dt;
59 // เก็บค่าการคำนวณหา dt
60

```



```
61 int round_counter = 0;
62 int walkingCheck[60];
63 int sittingCheck[60];
64 int standingCheck[60];
65 int count = 0;
66 int sumw = 0;
67 int sumsi = 0;
68 int sumst = 0;
69
70 int sta_walk = 0, sta_sit = 0, sta_stand = 0;
71 String sta_cow = "";
72
73 #define Echo 14
74 #define Trig 12
75 unsigned long duration;
76 float distance_z;
77
78 WiFiClient client;
79
80 int timer_feed = 0;
81 int timer_freeboard = 0;
82
83 MicroGear microgear(client);
84
85 // when the other thing send a msg to this board
86 void onMsghandler(char *topic, uint8_t* msg, unsigned int msglen) {
87     Serial.print("Incoming message --> ");
88     msg[msglen] = '\0';
89     Serial.println((char *)msg);
90 }
91
```

```
92 void onConnected(char *attribute, uint8_t* msg, unsigned int msglen) {
93     Serial.println("Connected to NETPIE...");
94     microgear.setAlias(ALIAS);
95 }
96
97 void setup() {
98     Wire.begin(4, 5); // GPIO4=SDA, GPIO5=SCL
99     Wire.beginTransmission(MPU);
100    Wire.write(0x6B);
101    Wire.write(0);
102    Wire.endTransmission(true);
103    pinMode(Trig, OUTPUT);
104    pinMode(Echo, INPUT);
105    microgear.on(MESSAGE, onMsghandler);
106    microgear.on(CONNECTED, onConnected);
107
108    Serial.begin(BAUD_RATE);
109    Serial.println("Starting...");
110
111    if (WiFi.begin(ssid, password)) {
112        while (WiFi.status() != WL_CONNECTED) {
113            delay(500);
114            Serial.print(".");
115        }
116    }
117
118    Serial.println("WiFi connected");
119    Serial.println("IP address: ");
120    Serial.println(WiFi.localIP());
121
122    microgear.init(KEY, SECRET, ALIAS);
```

```
123  microgear.connect(APPID);
124  }
125
126  void loop() {
127    GPS();
128    float acc = sta_xy();
129    float distance_z = sta_z();
130    round_counter++;
131    if (round_counter % 100 == 0) {
132      count = count >= 60 ? 0 : count + 1;
133      if (acc >= 50 ) {
134        walkingCheck[count] = 1;
135      }
136      else {
137        walkingCheck[count] = 0;
138      }
139      if (distance_z < 15 ) {
140        sittingCheck[count] = 1;
141      }
142      else {
143        sittingCheck[count] = 0;
144      }
145      if (acc < 50 && distance_z > 15 ) {
146        standingCheck[count] = 1;
147      }
148      else {
149        standingCheck[count] = 0;
150      }
151      sumw = 0;
152      sumsi = 0;
153      sumst = 0;
```

```
154     for (int i = 0; i < 60; i++) {
155         sumw += walkingCheck[i];
156         sumsi += sittingCheck[i];
157         sumst += standingCheck[i];
158     }
159     Serial.println(sumw);
160     Serial.println(sumsi);
161     Serial.println(sumst);
162     if (count >= 60) {
163         if (sumw > sumsi && sumw > sumst) {
164             //Serial.println("Walking");
165             sta_walk = 1;
166             sta_sit = 0;
167             sta_stand = 0;
168         }
169         else if (sumsi > sumw && sumsi > sumst) {
170             //Serial.println("Sitting");
171             sta_walk = 0;
172             sta_sit = 1;
173             sta_stand = 0;
174         }
175         else if (sumst > sumw && sumst > sumsi) {
176             //Serial.println("Standing");
177             sta_walk = 0;
178             sta_sit = 0;
179             sta_stand = 1;
180         }
181         sta_cow = "{"Walking\":";
182         sta_cow += sta_walk;
183         sta_cow += ", \"Sitting\":";
184         sta_cow += sta_sit;
```

```
185     sta_cow += ", \"Standing\":";
186     sta_cow += sta_stand;
187     sta_cow += "}";
188     for (int i = 0; i < 60; i++) {
189         walkingCheck[i] = 0;
190         sittingCheck[i] = 0;
191         standingCheck[i] = 0;
192     }
193 }
194
195 }
196 if (microgear.connected()) {
197     //Serial.print("*");
198     microgear.loop();
199     if (count >= 60) {
200         Serial.println("Sending -->Cow_Sta");
201         microgear.writeFeed(FEEDID, sta_cow);
202         count = 0;
203     }
204     if (round_counter >= 100) {
205         Serial.println("Sending -->GPS & Cow Behavior");
206         microgear.chat("/GPS_lat", String(Lati2, 6));
207         microgear.chat("/GPS_long", String(Longi2, 6));
208         microgear.chat("/AcX", String(AcX_Conv));
209         microgear.chat("/AcY", String(AcY_Conv));
210         microgear.chat("/AcZ", String(AcZ_Conv));
211         microgear.chat("/Roll", String(Comp_Angle[0]));
212         microgear.chat("/Pitch", String(Comp_Angle[1]));
213         if (cow_area) {
214             microgear.chat("/cow_area", String("Cow in Area"));
215         }
```

```

216     else {
217         microgear.chat("/cow_area", String("Cow out of Area"));
218     }
219     round_counter = 0;
220 }
221 }
222 else {
223     Serial.println("connection lost, reconnect...");
224     microgear.connect(APPID);
225 }
226 delay(10);
227 }
228 void GPS() {
229     while (Serial.available()) {
230         gps.encode(Serial.read());
231     }
232     Lati2 = String(gps.location.lat(), 6).toFloat();
233     // เก็บค่า latitude ลงในตัวแปร lati
234     Longi2 = String(gps.location.lng(), 6).toFloat();
235     // เก็บค่า longitude ลงในตัวแปร longi
236     //Serial.println("Latitude :"+String(Lati2,6)+" Longitude: "+String(Longi2,6));
237     // แสดงค่า latitude และ longitude
238     //Serial.println("Latitude Origin: "+String(Lati_Org,6)+" Longitude:
239 "+String(Longi_Org,6));
240
241     //หา distance ระหว่าง จุดที่กำหนดไว้และจุดที่รับเข้ามา
242     float Lati1_Rad = radians(Lati_Org);
243     // แปลง latitude ของจุดที่กำหนดไว้ ให้เป็น Radian
244     float Lati2_Rad = radians(Lati2);
245     // แปลง latitude ของจุดที่รับมา ให้เป็น Radian
246

```

```

247 float Lati = radians(Lati2 - Lati_Org);
248 // นำ latitude ของพิกัดที่กำหนดไว้และพิกัดที่รับมามาลบกันแล้วแปลงเป็น Radian
249 float Longi = radians(Longi2 - Longi_Org);
250 // นำ longitude ของพิกัดที่กำหนดไว้และพิกัดที่รับมามาลบกันแล้วแปลงเป็น Radian
251
252 //นำเข้าสู่สูตรคำนวณ haversine formula
253 float a = (sin(Lati / 2) * sin(Lati / 2)) + cos(Lati1_Rad) * cos(Lati2_Rad) *
254 sin(Longi) * sin(Longi);
255 float c = 2 * atan2(sqrt(a), sqrt(1 - a));
256 Distance = Earth_Radius * c ;
257 //Serial.println("Distance: "+String(Distance)+" Metre");
258 // แสดงค่าระยะห่างจากพิกัดที่กำหนดไว้และพิกัดที่รับมา
259
260 //เข้าใจเอาไว้ว่า ถ้าระยะห่างจากพิกัดที่กำหนดไว้และพิกัดที่รับมาน้อยกว่า 5000
261 เมตร แสดงผลว่าวัวอยู่ในพื้นที่แต่หากวัวอยู่ห่างมากกว่านั้นให้แสดงผลว่าวัวอยู่นอกพื้นที่
262 if (Distance < 5000) {
263     cow_area = true;
264 }
265 else {
266     cow_area = false;
267 }
268 }
269
270 float sta_xy() {
271     //รับค่า Accelerometer
272     Wire.beginTransmission(MPU);
273     // เริ่มการเชื่อมต่อไปที่ Address I2C
274     Wire.write(0x3B);
275     // ส่งบิตนำไปที่ Address 0x3B
276     Wire.endTransmission(false);
277

```

```

278 Wire.requestFrom(MPU, 6, true);
279 // เริ่มรับข้อมูลที่ส่งมา 6 Address
280 AcX = Wire.read() << 8 | Wire.read();
281 // รับข้อมูลจาก 0x3B และ 0x3C มาต่อกัน แล้วเก็บในตัวแปร AcX
282 AcY = Wire.read() << 8 | Wire.read();
283 // รับข้อมูลจาก 0x3D และ 0x3E มาต่อกัน แล้วเก็บในตัวแปร AcY
284 AcZ = Wire.read() << 8 | Wire.read();
285 // รับข้อมูลจาก 0x3F และ 0x40 มาต่อกัน แล้วเก็บในตัวแปร AcZ
286
287 //นำค่าดิบของ Accelerometer คำนวณเพื่อให้ได้ค่าที่ใช้งานได้จริง มีหน่วยเป็น g (g
288 unit = 9.807)
289 AcX_Scale = AcX / Acc_Scale;
290 AcY_Scale = AcY / Acc_Scale;
291 AcZ_Scale = AcZ / Acc_Scale;
292
293 //แปลงค่าความเร่งเชิงเส้นจากหน่วย G เป็น หน่วย m/s2
294 AcX_Conv = AcX_Scale * Earth_Gravity;
295 // ค่าเร่งเชิงเส้นแกน X หน่วย m/s2
296 AcY_Conv = AcY_Scale * Earth_Gravity;
297 // ค่าเร่งเชิงเส้นแกน Y หน่วย m/s2
298 AcZ_Conv = (AcZ_Scale - 1) * Earth_Gravity;
299 // ค่าเร่งเชิงเส้นแกน Z หน่วย m/s2 โดยต้อง ลบหนึ่งก่อนเนื่องจากแกน Z มีความเร่ง
300 โน้มถ่วงรวมอยู่ด้วย
301
302 //นำค่า Accelerometer มาคำนวณหา Euler Angle (Pitch, Roll)
303 Euler_Angle[0] = radians(atan(AcY_Scale / sqrt(pow((AcX_Scale), 2) +
304 pow((AcZ_Scale), 2)))); // หา Euler Angle มุม Roll แปลงจาก radian
305 เป็นองศาและเก็บในตัวแปร Euler_Angle[1]
306 Euler_Angle[1] = radians(atan((-1 * AcX_Scale) / sqrt(pow((AcX_Scale), 2)
307 + pow((AcZ_Scale), 2)))); // หา Euler Angle มุม Pitch แปลงจาก radian
308 เป็นองศาและเก็บในตัวแปร Euler_Angle[0]

```


308	//รับค่า Gyroscope
310	Wire.beginTransmission(MPU);
311	// เริ่มการเชื่อมต่อไปที่ Address I2C
312	Wire.write(0x43);
313	// ส่งบิตนำไปที่ Address 0x43
314	Wire.endTransmission(false);
315	Wire.requestFrom(MPU, 6, true);
316	// เริ่มรับข้อมูลที่ส่งมา 6 Address
317	GyX = Wire.read() << 8 Wire.read();
318	// รับข้อมูลจาก 0x43 และ 0x44 มาต่อกัน แล้วเก็บในตัวแปร GyX
319	GyY = Wire.read() << 8 Wire.read();
320	//รับข้อมูลจาก 0x45 และ 0x46 มาต่อกัน แล้วเก็บในตัวแปร GyY
321	GyZ = Wire.read() << 8 Wire.read();
322	//รับข้อมูลจาก 0x47 และ 0x48 มาต่อกัน แล้วเก็บในตัวแปร GyZ
323	
324	//นำค่าดิบของ Gyroscope มาคำนวณเพื่อให้ได้ค่าที่ใช้งานได้จริง มีหน่วยเป็น %s
325	(องศา/วินาที)
326	GyX_Scale = GyX / Gyro_Scale;
327	GyY_Scale = GyY / Gyro_Scale;
328	GyZ_Scale = GyZ / Gyro_Scale;
329	
340	//คำนวณหาค่า dt
341	dt = (millis() - time_prev) / 1000.0;
342	time_prev = millis();
343	
344	//นำค่า Accelerometer และ Gyroscope มาหามุมโดยใช้ Complementary
345	Filter
346	Comp_Angle[0] = (1 - Alpla_Cp) * (Comp_Angle[0] + GyX_Scale * dt) +
347	Alpla_Cp * Euler_Angle[0] ; //มุม Roll ที่ผ่าน Complementary Filter
348	และเก็บในตัวแปร Comp_Angle[0]
349	

```

351   Comp_Angle[1] = (1 - Alpla_Cp) * (Comp_Angle[1] + GyY_Scale * dt) +
352   Alpla_Cp * Euler_Angle[1] ;           //มุม Pitch ที่ผ่าน Complementary Filter
353   และเก็บในตัวแปร Comp_Angle[1]
354   //แปลงค่า Acc และ Euler angle เป็น String และแสดงค่าออกมา
355   //values = String(Comp_Angle[1]) + ", " + String(Comp_Angle[0]) + ", " +
356   String(AcX_Conv) + ", " + String(AcY_Conv) + ", " + String(AcZ_Conv);
357   //values = String(AcX_Conv) + "," + String(AcY_Conv) + "," +
358   String(AcZ_Conv);
359   //Serial.println(values);
360
361   float acc = sqrt(pow(AcX_Conv * dt * 1000, 2) + pow(AcY_Conv * dt *
362   1000, 2) + pow(AcZ_Conv * dt * 1000, 2));
363   return acc;
364   }
365   float sta_z() {
366   //ส่งสัญญาณ PULSE ไปที่ขา Trig ทุกๆ 10 microsec
367   digitalWrite(Trig, LOW);
368   delayMicroseconds(2);
369   digitalWrite(Trig, HIGH);
370   delayMicroseconds(10);
371   digitalWrite(Trig, LOW);
372   //ตรวจสอบความกว้างของสัญญาณ PULSE ที่ขา Echo
373   duration = pulseIn(Echo, HIGH, 10000);
374   if (duration != 0) {
375       distance_z = duration * 0.0343 / 2;
376   }
377   return distance_z;
378   }

```

ประวัติผู้จัดทำ



ชื่อ-สกุล นายจิรศักดิ์ เอี่ยมสะอาด
 สาขา วิศวกรรมคอมพิวเตอร์
 วัน/เดือน/ปีเกิด 01 มีนาคม 2537
 สถานที่เกิด ภูเก็ต
 ที่อยู่ 50/6 หมู่ 1 ซอย สุขนิรันดร์ ตำบล วิจิต ถนน เจ้าฟ้า อำเภอ เมือง จังหวัด ภูเก็ต 83000

ประวัติการศึกษา

ระดับประกาศนียบัตรวิชาชีพ

วิทยาลัยเทคนิคภูเก็ต แผนก อิเล็กทรอนิกส์

ปีการศึกษาที่จบ พ.ศ.2555

ระดับประกาศนียบัตรวิชาชีพชั้นสูง

วิทยาลัยเทคนิคภูเก็ต แผนก อิเล็กทรอนิกส์ สาขา เทคนิคคอมพิวเตอร์

ปีการศึกษาที่จบ พ.ศ.2557

ประวัติผู้จัดทำ



ชื่อ-สกุล	นายอภิวัฒน์ จันทวงษ์
สาขา	วิศวกรรมคอมพิวเตอร์
วัน/เดือน/ปีเกิด	05 กุมภาพันธ์ 2540
สถานที่เกิด	โรงพยาบาลสมุทรสาคร
ที่อยู่	94/63 หมู่ 7 ซอย กิโลเมตร 2 ตำบล ท่าทราย ถนน เศรษฐกิจ อำเภอ เมือง จังหวัด สมุทรสาคร 74000

ประวัติการศึกษา

ระดับประกาศนียบัตรวิชาชีพ

วิทยาลัยเทคนิคสมุทรสาคร แผนก อิเล็กทรอนิกส์

ปีการศึกษาที่จบ พ.ศ.2557

ระดับประกาศนียบัตรวิชาชีพชั้นสูง

วิทยาลัยเทคนิคสมุทรสาคร แผนก อิเล็กทรอนิกส์ สาขา วิศวกรรมคอมพิวเตอร์

ปีการศึกษาที่จบ พ.ศ.2559