

เครือข่าย IoT เซ็นเซอร์สำหรับจักรยาน และจักรยานยนต์
IoT Sensors Network for Bicycles and Motorcycles

| | |
|--------------|---------------|
| ไชยพฤกษ์ | นครไพร |
| CHAIYAPUEK | NAKORN PRAI |
| สุรเทพ | ทนทาน |
| SURATHEP | THON THAN |
| วงศกร | ก่อมขุนทด |
| WONGSAKRON | KOMKKHUNTHOD |
| อภิชูภพ | 悱อกทองใบ |
| APHICHUPHAP | PHUETHONGBAI |
| ชีษณุพงศ์ | วาสนาสมคักษ์ |
| CHISANUPHONG | VASANASOMSAK |
| กนกรัตน์ | เมธาราภุล |
| KANOKRAT | METHAWATRAKUN |

ปริญญา妮พนธน์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
มหาวิทยาลัยกรุงเทพ
พ.ศ.2564

เครื่อข่าย IoT เซ็นเซอร์สำหรับจัดการงาน และจัดการยารยนต์

| | |
|-----------|---------------|
| ไชยพงษ์ | นครไพร |
| สุรเทพ | ทันทาน |
| วงศกร | ก่อมขุนทด |
| อภิชญาพ | ເຜື້ອກທອງໄປ |
| ชีษณุพงศ์ | วาสนาสมศักดิ์ |
| กนกรัตน์ | เมธาวราภุล |

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยกรุงเทพ
พ.ศ.2564

IoT Sensors Network for Bicycles and Motorcycles

| | |
|--------------|---------------|
| CHAIYAPUEK | NAKORNRAI |
| SURATHEP | THONTHAN |
| WONGSAKRON | KOMKKHUNTHOD |
| APHICHUPHAP | PHUETHONGBAI |
| CHISANUPHONG | VASANASOMSAK |
| KANOKRAT | METHAWATRAKUN |

A PROJECT SUMMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF ENGINEERING IN
ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING
BANGKOK UNIVERSITY

2021

**COPYRIGHT 2021
SCHOOL OF ENGINEERING
BANGKOK UNIVERSITY**

คณะวิศวกรรมศาสตร์
มหาวิทยาลัยกรุงเทพ
ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์

ชื่อนักศึกษา

เครือข่าย IoT เชิงเซอร์สำหรับจักรยาน และจักรยานยนต์
IoT Sensors Network for Bicycles and Motorcycles

รหัสประจำตัว

นาย ไชยพุกษ์ นครไพร

ชื่อนักศึกษา

1620902708

รหัสประจำตัว

นาย สุรเทพ ทนทาน

ชื่อนักศึกษา

1610901629

รหัสประจำตัว

นาย วงศกร ก่อมขุนทด

ชื่อนักศึกษา

16109000829

รหัสประจำตัว

อภิญญา พีอกทองใบ

ชื่อนักศึกษา

1610901132

รหัสประจำตัว

นาย ชิษณุพงศ์ วานานิมศักดิ์

ชื่อนักศึกษา

1640903181

รหัสประจำตัว

นางสาว กนกรัตน์ เมราวรากุล

รหัสประจำตัว

1610901017

ปริญญา

วิศวกรรมศาสตรบัณฑิต

สาขาวิชา

วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมปริญญานิพนธ์

ผศ.ดร.ปกรณ์ ยุบลโภศล

| คณะกรรมการสอบปริญญานิพนธ์ | ลายมือชื่อ |
|---------------------------|--------------|
| ผศ.ดร.ปกรณ์ ยุบลโภศล | P.U. |
| รศ.ดร.นันทิยา ชัยบุตร | |
| ดร.ศิริชัย เติมโฉกเกษม | ๖๖๖๖ ๒๔๗๘๑๖๖ |

วัน/เดือน/ปีที่สอบ 8 มิถุนายน 2565 เวลา 16.00-17.00 น.

สถานที่สอบ

ห้องสอบปริญญานิพนธ์ คณะวิศวกรรมศาสตร์ อาคาร B4 ชั้น 4 ห้อง 408

คณะวิศวกรรมศาสตร์รับรองแล้ว

.....
(ผศ.ดร.วิศาล พัฒน์ชู)
คณะบดีคณะวิศวกรรมศาสตร์
วันที่.....เดือน.....พ.ศ.

| | |
|------------------------------|---|
| หัวข้อปริญญาบัณฑิต | เครื่อข่าย IoT เชื่อมเครือสำหรับจัดการ และจัดการยานยนต์ |
| นักศึกษา | นายสุรเทพ พนทาน |
| รหัสประจำตัว | 1610901629 |
| นักศึกษา | นายวงศกร ก่อมชุนทด |
| รหัสประจำตัว | 1610900829 |
| นักศึกษา | นายอภิญญาพ เฟือกทองใบ |
| รหัสประจำตัว | 1610901132 |
| นักศึกษา | นายชัยณรงค์ วาสนาสมศักดิ์ |
| รหัสประจำตัว | 1610902320 |
| นักศึกษา | นางสาวกนกรัตน์ เมธารากุล |
| รหัสประจำตัว | 1610901017 |
| นักศึกษา | นายไชยพฤกษ์ นครไพร |
| รหัสประจำตัว | 1620902708 |
| ปริญญา | วิศวกรรมศาสตรบัณฑิต |
| สาขาวิชา | วิศวกรรมไฟฟ้า |
| ปี พ.ศ. | 2564 |
| อาจารย์ผู้ควบคุมปริญญาบัณฑิต | ผศ.ดร.ปกรณ์ ยุบลโภสล |

บทคัดย่อ

ในปัจจุบันโลกของเรามาได้เข้าสู่สังคมโลกของอินเทอร์เน็ตทุกสรรพสิ่ง หรือเรียกว่า IoT กล่าวอีกนัยหนึ่งคือโลกที่ทุกอย่างมีอุปกรณ์อิเล็กทรอนิกส์ที่สามารถเชื่อมต่อกันผ่านอินเทอร์เน็ตเป็นสิ่ง อำนวยความสะดวกที่เพิ่มมากขึ้นเพื่อให้ง่ายต่อการใช้ชีวิตในปัจจุบัน พร้อมกับมาตรการของรัฐบาลที่หัน กระตุนการดูแลสุขภาพที่เพิ่มมากขึ้น ไม่ว่าจะเป็นการขึ้นภาษีนำ้ตาล ภาษีเกลือ หรือแม้แต่น้ำมัน ทำให้ ในปัจจุบันประชาชนจำนวนหนึ่งเริ่มหันมาทำการเดินทางด้วยรถจักรยานที่เพิ่มมากขึ้นไม่ว่าจะเพื่อการ เดินทางหรือเพื่อออกกำลังกายแต่ก็สิ่งที่ปฏิเสธไม่ได้คือสภาพพื้นผิวนานของประเทศไทยที่ยังไม่อำนวย สำหรับการปั่นจักรยานไม่ว่าจะเป็น พื้นถนนไม่เรียบ ไม่มีเลนสำหรับจักรยานหรือมีแต่กลับมี รถจักรยานยนต์ไปขับ หรือแม้แต่การขับข้อนเลน ด้วยปัญหาที่เกิดขึ้นนี้ทางผู้จัดทำจึงเล็งเห็นถึงความ อัตราภัยในการขับขี่ทางผู้จัดทำจึงได้นำเสนอระบบเครือข่าย IoT เชื่อมเครือสำหรับจักรยาน และ จัดการยานยนต์เพื่อที่จะได้เข้ามาอำนวยความสะดวกสำหรับผู้ขับขี่รถจักรยานหรือแม้แต่ผู้ที่ขับขี่รถจักรยาน นายนต์ได้เช่นกัน โดยการนำระบบหน่วยตรวจจับการเคลื่อนไหวภายในเข้ามาโดยใช้เซ็นเซอร์วัดความเร่ง แบบ 3 แกนผ่านระบบการเรียนรู้ด้วยตัวเองเพื่อนำมาใช้คำนวณพุทธิกรรมขณะขับขี่รถจักรยานหรือ รถจักรยานยนต์เพื่อเพิ่มความปลอดภัยขณะขับขี่อีกทั้งระบบยังทำการส่งข้อมูลแบบตลอดเวลาขึ้นไปยัง คลาวด์เพื่อให้บุคคลทางบ้านสามารถติดตามบุตรหลานได้อย่างใกล้ชิดเพื่อความปลอดภัย

| | |
|------------------------|--|
| Project | IoT Sensors Network for Bicycles and Motorcycles |
| Student | Mr. Surtheep Thonthan |
| Student ID. | 1610901629 |
| Student | Mr. Wongsakron Komkhunthod |
| Student ID. | 1610900829 |
| Student | Mr. Aphichuphap Phuethongbai |
| Student ID. | 1610901132 |
| Student | Mr. Chitsanuphong Vasanasomsak |
| Student ID. | 1610902320 |
| Student | Miss. Kanokrat Methawatrakun |
| Student ID. | 1610901017 |
| Student | Mr. Chaiyapuek Nakornprai |
| Student ID. | 1620902708 |
| Degree | Bachelor of Engineering |
| Programme | Electrical Engineering |
| Year | 2021 |
| Project Advisor | Asst. Prof. Dr. Pakorn Ubolkosold |

ABSTRACT

Our world has now entered the Internet of Things (IoT) age. To put it another way, a world where everything is connected via the Internet offers a wide range of conveniences. Along with government initiatives aimed at encouraging higher health-care utilization. Whether it's increasing the sugar, salt, or even oil taxes. As a result, an increasing number of people are opting for bicycle commuting, whether for commuting or exercise. However, the state of Thailand's road surface, which is still not conducive to cycling, cannot be overlooked. Like unsteady path and There are no bike lanes, or simply motorbike lanes to drive, or even a reverse lane. The organizers anticipate a risk of driving as a result of this issue. With the introduction of an IMU or Inertial Measurement Unit, which measures internal movements with 3-axis accelerometers through machine learning to predict behavior while riding a bicycle or motorcycle to increase safety while driving, the organizers have presented a network of IOT sensors for electric bicycles in order to facilitate the rider.

กิตติกรรมประกาศ

ปริญญา妮พนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี เป็นพระไเดรับความเมตตากรุณาจากอาจารย์
ผู้คุมปริญญานิพนธ์คือ ผศ.ดร.ปกรณ์ ยุบลโกศล ที่ได้ให้คำชี้แนะและให้คำปรึกษาเกี่ยวกับการ
ออกแบบระบบฐานข้อมูล และส่วนการทำงานของโปรแกรมมาโดยตลอด ผู้ทำโครงการรู้สึกทราบชั้ง
ในความเคราะห์จากท่าเป็นอย่างมาก และทราบขอบเขตคุณเป็นอย่างสูง นอกจากนี้ ขอขอบคุณ
ดร.ศิริชัย เติมโชคเกษม และ รศ.ดร.นันทิยา ชัยบุตร ที่สละเวลาเป็นกรรมการคุ้มสอบและช่วยแนะนำ
ขอขอบคุณเจ้าหน้าที่ห้องเบิกจ่ายอุปกรณ์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยกรุงเทพ
ที่ให้ข้อมูลต่างๆ เกี่ยวกับฐานข้อมูล

ผู้ทำโครงการขอกราบขอบคุณ คุณพ่อ คุณแม่ และเพื่อน ๆ พี่ ๆ ที่ให้กำลังใจ และให้
โอกาสบุตรได้รับการศึกษาในระดับต่างๆ จนกระทั่งได้สำเร็จการศึกษาระดับปริญญาวิศวกรรมศาสตร์
บัณฑิตนี้ รวมทั้งคณาจารย์ทุกท่าน รวมทั้งคณะวิศวกรรมศาสตร์ที่ได้รับประสิทธิ์ประสาทวิชาความรู้
และคำสั่งสอนให้กับผู้ทำโครงการในการเรียนทุกระดับชั้น

สุดท้ายความรู้ และประโยชน์ที่ได้รับปริญญานิพนธ์ฉบับนี้ ผู้จัดทำโครงการขออภัยความดีที่
ได้ให้แก่ผู้มีพระคุณทุกท่าน

| | |
|-----------|--------------|
| สุรเทพ | ทันทาน |
| ไชยพฤกษ์ | นครไพร |
| อภิญญาพ | ເພື່ອກທອງໃບ |
| วงศกร | ກ່ອມຊຸນທດ |
| ชีชนุพงศ์ | ວາສนาສມศักดີ |
| กนกรัตน์ | เมธารากุล |

สารบัญ

| | หน้า |
|---|-------|
| บทคัดย่อภาษาไทย..... | I |
| บทคัดย่อภาษาอังกฤษ..... | II |
| กิตติกรรมประกาศ..... | III |
| สารบัญ..... | IV |
| สารบัญรูป..... | VII |
| สารบัญตาราง..... | X |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 เหตุผลและความเป็นมาของโครงการ..... | 1 |
| 1.2 วัตถุประสงค์ของโครงการ..... | 1 |
| 1.3 ขอบเขตของโครงการ..... | 2 |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับ..... | 2 |
| 1.5 งบประมาณที่ใช้ในการดำเนินโครงการ..... | 2 |
| บทที่ 2 ทฤษฎีที่เกี่ยวข้อง..... | 5 |
| 2.1 Accelerometer 3 แกน..... | 5 |
| 2.2 โมดูลจีพีเอส NEO-6M..... | 7 |
| 2.3 MAX30102..... | 10 |
| 2.4 FSR402..... | 11 |
| 2.5 ESP8266 และ NodeMCU..... | 12 |
| 2.6 Rasberry Pi..... | 13 |
| 2.7 โปรแกรม Visual Studio Code..... | 13 |
| 2.8 ภาษาเพthon (Python)..... | 14 |
| 2.9 Scikit-learn..... | 15 |
| 2.10 การทำ Machine Learning..... | 15 |
| 2.10.1 ยัลกอริทึมต้นไม้ตัดสีใจ (Decision Tree Algorithm)..... | 17 |
| 2.10.2 ยัลกอริทึมเพื่อนบ้านใกล้สุด (K-Nearest Neighbors Algorithm)..... | 18 |
| 2.10.3 ยัลกอริทึมเกาส์เชี่ยน นาอีฟเบย์ (Gaussian Naive Bayes Algorithm).... | 19 |
| 2.11 Microsoft Azure..... | 20 |
| 2.12 MQTT Protocol..... | 21 |
| 2.13 Node RED..... | 22 |

สารบัญ(ต่อ)

| | หน้า |
|--|-----------|
| บทที่ 3 ขั้นตอนการออกแบบและพัฒนา..... | 23 |
| 3.1 ขั้นตอนการดำเนินงาน..... | 23 |
| 3.2 การออกแบบระบบการทำงาน..... | 24 |
| 3.3 แผนผังระบบการทำงาน..... | 25 |
| 3.4 การออกแบบการทำ Machine Learning นำมายพัฒนาระบบการทำงาน..... | 26 |
| 3.4.1 การรับความเร่งค่าจาก Accelerometer..... | 27 |
| 3.4.2 การเทรน-ทดสอบ..... | 28 |
| 3.4.2.1 การเทรนโดยใช้ Decision Tree Algorithm..... | 28 |
| 3.4.2.2 การเทรนโดยใช้ K-Nearest Algorithm..... | 29 |
| 3.4.2.3 การเทรนโดยใช้ Gaussian Naive Bayes Algorithm..... | 29 |
| 3.4.2.4 การดึงข้อมูลโดยใช้ 3 โมเดล มาใช้..... | 30 |
| 3.4.2.5 การเลือกคำตอบที่ถูกต้องที่สุด..... | 31 |
| 3.4.2.6 การส่งข้อมูลคำตอบไปยัง MQTT Broker..... | 32 |
| 3.4.3 แผนผังการทำงานของ Machine Learning..... | 33 |
| 3.5 แผนภาพการต่อวงจรของเซ็นเซอร์จีพีเอส และFSR เข้ากับ NodeMCU..... | 34 |
| 3.6 แผนภาพการต่อวงจรของเซ็นเซอร์ MAX30102 เข้ากับ NodeMCU..... | 35 |
| 3.7 การเขียนโค้ดดึงข้อมูลจาก MQTT Broker..... | 38 |
| 3.8 การทดสอบการทำงานของชิ้นงาน..... | 35 |
| บทที่ 4 ผลการดำเนินงานในโครงงานวิศวกรรม..... | 40 |
| 4.1 ผลการทดสอบการทำงานนำมายพัฒนาระบบ..... | 40 |
| 4.1.1 อัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm)..... | 40 |
| 4.1.2 อัลกอริทึมเพื่อนบ้านใกล้สุด (K-Nearest Neighbor Algorithm หรือ KNN)..... | 41 |
| 4.1.3 อัลกอริทึมเกาส์เชี่ยน นาอีฟเบย์ (Gaussian Naive Bayes Algorithm)... | 42 |
| 4.2 กระบวนการการทำงานของระบบ..... | 43 |
| 4.3 การแสดงผลของข้อมูล..... | 43 |
| บทที่ 5 สรุปผลการดำเนินงานปริญญาบัณฑิต..... | 44 |
| 5.1 สรุปผลการดำเนินงาน..... | 44 |
| 5.2 ข้อจำกัดของโครงงาน..... | 44 |
| 5.3 ข้อเสนอแนะ..... | 44 |

สารบัญ(ต่อ)

| | หน้า |
|----------------------------|------|
| 5.4 แนวทางการพัฒนาต่อ..... | 44 |
| บรรณานุกรม..... | 45 |
| ภาคผนวก..... | 47 |
| ประวัติผู้เขียน..... | 72 |

สารบัญรูป

| รูปที่ | หน้า |
|--|------|
| 2.1 Accelerometer module ADXL345..... | 6 |
| 2.2 ทิศทางการเอียงในมุมต่าง ๆ เปรียบเทียบเหมือนแกน X ,Y ,Z..... | 7 |
| 2.3 จำลองการรับสัญญาณจากดาวเทียม..... | 9 |
| 2.4 โมดูลจีพีเอส NEO-6M..... | 9 |
| 2.5 MAX30102..... | 10 |
| 2.6 ตัวอย่างวิธีการใช้งานและหลักการทำงานของเซ็นเซอร์ MAX30120..... | 11 |
| 2.7 FSR402..... | 11 |
| 2.8 ESP8266..... | 12 |
| 2.9 NodeMCU..... | 12 |
| 2.10 บอร์ด Raspberry Pi..... | 13 |
| 2.11 หน้าจอโปรแกรม Visual Studio Code..... | 14 |
| 2.12 อัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm)..... | 17 |
| 2.13 อัลกอริทึมเพื่อนบ้านใกล้สุด (K-Nearest Neighbor algorithm)..... | 18 |
| 2.14 Microsoft Azure..... | 20 |
| 2.15 Node RED Logo..... | 22 |
| 2.16 Node.js Logo..... | 22 |
| 3.1 ขั้นตอนการทำงานของโครงงาน..... | 23 |
| 3.2 โครงสร้างของระบบ..... | 24 |
| 3.3 ผังการทำงานของระบบ..... | 25 |
| 3.4 ขั้นตอนการทำงานระบบทำนายพฤติกรรม..... | 26 |
| 3.5 การต่อวงจรเซ็นเซอร์ Accelerometer เข้ากับ Raspberry Pi..... | 27 |
| 3.6 ตัวอย่างโค้ดของ Accelerometer..... | 27 |
| 3.7 โค้ดการเทรนโมเดลแบบ Decision Tree Algorithm..... | 28 |
| 3.8 โค้ดการเทรนโมเดลแบบ K-Nearest Neighbor Algorithm (KNN)..... | 29 |
| 3.9 โค้ดการเทรนโมเดลแบบ Gaussian Naive Bayes Algorithm..... | 29 |
| 3.10 โค้ดการใช้งานโมเดล..... | 30 |
| 3.11 Code Counter ที่ใช้ในการเลือกคำตอบที่ถูกที่สุด..... | 31 |
| 3.12 ตัวอย่างการทำงานของโปรแกรม Counter..... | 32 |
| 3.13 ค่าตัวเลขที่ใช้แทนค่าอาร์ย์จากเซ็นเซอร์..... | 32 |
| 3.14 โค้ดที่ใช้สำหรับส่งข้อมูลไปยัง MQTT Broker..... | 32 |
| 3.15 ผังการทำงานของ Raspberry Pi..... | 33 |

สารบัญรูป(ต่อ)

| รูปที่ | หน้า |
|--|------|
| 3.16 การต่อวงจรของเซ็นเซอร์จีพีเอส และ FSR402 เข้ากับ NodeMCU..... | 34 |
| 3.17 โค้ดการทำงานเซ็นเซอร์จีพีเอส..... | 34 |
| 3.18 โค้ดการทำงานเซ็นเซอร์ FSR402..... | 35 |
| 3.19 การต่อวงจรของเซ็นเซอร์ MAX30102 เข้ากับ NodeMCU..... | 35 |
| 3.20 คำสั่งการทำงานของวงจร MAX30102..... | 36 |
| 3.21 คำสั่งการทำงานของวงจร MAX30102 (ต่อ)..... | 37 |
| 3.22 การเขียนโค้ดดึงข้อมูลจาก MQTT Broker..... | 38 |
| 3.23 การติดตั้งเพื่อทดสอบการทำงานในสภาพแวดล้อมจริง (1)..... | 38 |
| 3.24 การติดตั้งเพื่อทดสอบการทำงานในสภาพแวดล้อมจริง (2)..... | 39 |
| 4.1 หน้าจอแสดงผลของระบบ..... | 43 |

สารบัญตาราง

| ตารางที่ | หน้า |
|--|------|
| 1.1 แผนการดำเนินงาน..... | 4 |
| 2.1 ตัวอย่างค่า Accelerometer 3 แกน..... | 16 |
| 4.1 ตารางความถูกต้องของอัลกอริทึมต้นไม้ตัดสินใจ..... | 40 |
| 4.2 ตารางความถูกต้องของอัลกอริทึมเพื่อนบ้านใกล้สุด..... | 41 |
| 4.3 ตารางความถูกต้องของอัลกอริทึมเก้าส์เซียน นายอีฟเบย์..... | 42 |

บทที่ 1

บทนำ

บทนี้จะกล่าวถึงความเป็นมาของและเหตุผลของโครงการเครือข่ายอินเตอร์เน็ตในทุกสิ่ง (IoT) เช่นเซอร์ฟาร์บ จักรยาน และจักรยานยนต์ (IoT Sensors Network for Bicycles and Motorcycles) ระบบที่จะค่อยติดตามและทำนายพฤติกรรมจากลักษณะการทรงตัวของรถจักรยานหรือรถจักรยานยนต์ โดยขับขี่โดยใช้บอร์ดraspberrypaiy ขณะขับขี่โดยใช้บอร์ดraspberrypaiy (Raspberry Pi) ซึ่งเป็นคอมพิวเตอร์ขนาดเล็ก และคอนโทรลเลอร์บอร์ด (NodeMCU) เป็นระบบปฏิบัติการร่วม นอกเหนือนี้ยังกล่าวถึงวัตถุประสงค์ของโครงการ ขอบเขตของโครงการ ประโยชน์ที่คาดว่าจะได้รับ และงบประมาณที่คาดว่าจะใช้ในการทำโครงการในส่วนสุดท้ายจะแสดงถึงแผนการดำเนินงานทั้งหมด

1.1. เหตุผลและความเป็นมาของโครงการ

การขับขี่รถจักรยาน หรือ รถจักรยานยนต์ในประเทศไทยมักจะพบปัญหาบ่อยครั้งไม่ว่าจะเป็นถนนไม่เรียบ ไม่มีเลนจักรยาน การขับขันเลน หรืออื่น ๆ อีกมากมายจะเห็นได้ว่าจากตัวอย่างที่กล่าวมาล้วนเป็นเหตุผลที่ทำให้เกิดอุบัติเหตุในการขับขี่รถจักรยานหรือรถจักรยานยนต์ได้ทั้งนั้นนี้จึงเป็นหนึ่งในสาเหตุที่ทำให้บางครั้งผู้ปกครองจึงหลีกเลี่ยงไม่ให้บุตรหลานใช้รถจักรยานหรือรถจักรยานยนต์เนื่องจากเป็นห่วงในเรื่องของความปลอดภัย

ปัญหาของอุบัติเหตุที่เกิดขึ้นส่งผลให้เกิดความไม่สงบใจแก่ผู้ปกครองที่ไม่อยากให้บุตรหลานออกไปภายนอกบ้านด้วยรถจักรยานหรือจักรยานยนต์ ไม่ว่าจะด้วยปัญหารื่องอุบัติเหตุหรือแม้แต่การหลงทางจากปัญหาที่กล่าวมาส่งผลให้ประชากรไทยมีปัญหารื่องน้ำหนักที่เกินมาตรฐานส่งผลให้ภาครัฐผลักดันควบคุมน้ำหนักโดยการขึ้นภาษีน้ำตาล เกลือ และน้ำมัน

เพื่อแก้ไขปัญหาดังกล่าวข้างต้น ทางผู้จัดทำ IoT สำหรับจักรยาน และจักรยานยนต์ โดยการติดตั้งเซ็นเซอร์ที่จะเป็นประโยชน์ต่อการใช้รถ และระบบเครื่องคอมพิวเตอร์ที่สามารถเรียนรู้ได้ด้วยตัวเอง (Machine Learning) ในการตรวจจับพฤติกรรมของผู้ใช้รถจากลักษณะพฤติกรรมการทรงตัวของรถ ร่วมถึงการติดตามพิกัดที่อยู่ของรถ

1.2. วัตถุประสงค์ของโครงการ

- เพื่อพัฒนาระบบทามนัยพฤติกรรมขณะขับขี่จักรยานหรือจักรยานยนต์
- เพื่อเพิ่มระดับความปลอดภัยให้กับผู้ขับขี่ และความสงบใจของผู้ปกครอง
- เพื่อศึกษาระบบปฏิบัติการ Cloud Server เพื่อใช้เป็นตัวเลือกใหม่ ๆ ในอนาคต
- เพื่อศึกษาและประยุกต์ใช้โปรโตคอลสื่อสาร MQTT
- เพื่อศึกษาและประยุกต์ใช้โปรแกรม Node RED

1.3. ขอบเขตของโครงการ

ระบบติดตามพฤติกรรมขณะขับขี่รถจักรยานหรือรถจักรยานยนต์ด้วยวิธีการติด Accelerometer ที่ตัวรถแล้วรับข้อมูลจาก Accelerometer มาทำนายพฤติกรรมของลักษณะการทรงตัวของรถด้วยบอร์ด Raspberry Pi หรือที่เรียกว่า การเรียนรู้ของเครื่อง (Machine Learning) โดยการเขียนโปรแกรมด้วยภาษาไพทอน (Python) ติดตามพิกัดด้วยจีพีเอส คนนั่งเบาะด้วยเซ็นเซอร์วัดแรงกด และวัดอัตราการเต้นของหัวใจ

- เพื่อศึกษาการประยุกต์ใช้ระบบการเรียนรู้ของเครื่อง (Machine Learning)
- สามารถออกแบบและพัฒนาให้ระบบทำนายพฤติกรรมมีความถูกต้องแม่นยำในการใช้งานจริงอย่างน้อย 80%
- สามารถทำนายพฤติกรรมลักษณะการทรงตัวของรถในสถานการณ์ต่าง ๆ ได้มากกว่า 4 ประเภท
- สามารถรับ และ ส่งค่าจากเซ็นเซอร์ได้ทั้งหมด
- เซ็นเซอร์ทุกตัวสามารถทำงานพร้อมกันได้
- สามารถใช้งาน Cloud Server เพื่อเป็นตัวกลางในการบันทึกจัดเก็บ และ รับ-ส่ง ข้อมูล
- ตรวจสอบค่าสถานะได้ตลอดเวลาในทุกที่

1.4. ประโยชน์ที่คาดว่าจะได้รับ

- ได้พัฒนาระบบตรวจสอบพฤติกรรมจักรยานหรือรถจักรยานยนต์ขณะขับขี่
- ได้เรียนรู้ระบบการเรียนรู้ของเครื่อง (Machine Learning)
- ได้เรียนรู้ภาษาทางคอมพิวเตอร์ภาษาใหม่ๆ (Python)
- ได้เรียนรู้การทำงานเป็นกลุ่ม
- ได้เรียนรู้ระบบปฏิบัติการ Cloud Server (Microsoft Azure)
- ได้เรียนรู้การทำงานร่วมกันของเซ็นเซอร์

1.5. งบประมาณที่ใช้ในการดำเนินโครงการ

งบประมาณที่ใช้ดำเนินการแบ่งเป็น 2 ส่วน คือ ฮาร์ดแวร์ ได้แก่ เครื่องมือที่ใช้ในการประเมินผล และอุปกรณ์ต่อเสริมรวมถึงอุปกรณ์ติดตั้ง ซอฟต์แวร์ ได้แก่ ภาษาของโปรแกรมที่นำมาใช้ในการระบบรวมไปถึงภาษาที่นำไปใช้สำหรับเก็บข้อมูลลงในฐานข้อมูล และเว็บไซต์ที่ใช้แสดงผล

ฮาร์ดแวร์

| | | |
|-------------------------|-------|-----|
| Raspberry Pi 3 Model B+ | 2,000 | บาท |
| GY-291 ADXL345 | 50 | บาท |
| GPS Module | 100 | บาท |
| NodeMCU | 150 | บาท |
| SD Card | 795 | บาท |

| | | |
|--------------------------------------|-----|-----|
| • Pulse heart rate sensor (MAX30102) | 250 | บาท |
| • Weight Senso (FSR) | 700 | บาท |
| • สาย HDMI | 100 | บาท |

ซอฟต์แวร์

| | |
|-------------------|-----------------|
| • VSCode | ไม่มีค่าใช้จ่าย |
| • Python | ไม่มีค่าใช้จ่าย |
| • Scikit-learn | ไม่มีค่าใช้จ่าย |
| • Microsoft Azure | ไม่มีค่าใช้จ่าย |
| • Arduino IDE | ไม่มีค่าใช้จ่าย |
| • Node-Red | ไม่มีค่าใช้จ่าย |

รวมเป็นเงินทั้งสิ้น 4,145 บาท

ຕາງ່າງທີ 1.1 ແນວດການດຳເນີນຂາຍ

บทที่ 2

ทฤษฎีเกี่ยวข้อง

การดำเนินโครงการในครั้งนี้เป็นการรวมกันของเข็นเซอร์หลายชนิดและทำ Machine Learning เข้ามาร่วมติดกับจักรยานไฟฟ้าแล้วเชื่อมต่ออินเตอร์เน็ตส่งรับส่งข้อมูลด้วย MQTT เพื่อเพิ่มความสามารถ IoT (SMART IoT) ให้ตัวรถ และความปลอดภัยที่มากขึ้นสำหรับผู้ขับขี่โดยทางผู้จัดทำได้มีการว่างตำแหน่งในการติดตั้งเข็นเซอร์ชนิดต่าง ๆ ไว้ดังนี้ (รูปที่ 2.1) ทั้งนี้ทางผู้จัดทำจึงทำการแบ่งหัวข้อหลัก ๆ ของเข็นเซอร์แต่ละตัวออกมา ดังต่อไปนี้

- 2.1 Accelerometer 3 แกน
- 2.2 โมดูลจีพีเอส NEO-6M
- 2.3 MAX30102
- 2.4 FSR402
- 2.5 esp8266 และ NodeMCU
- 2.6 Raspberry Pi
- 2.7 Arduino IDE
- 2.8 โปรแกรม Visual Studio Code
- 2.9 ภาษาไพธอน (Python)
- 2.10 Scikit-learn
- 2.11 การทำ Machine Learning
 - 2.11.1 อัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm)
 - 2.11.1 อัลกอริทึมเพื่อนบ้านใกล้สุด (K-Nearest Neighbor Algorithm)
 - 2.11.1 อัลกอริทึมเกาส์เชี่ยน นาอีฟเบย์ (Gaussian Naive Bayes Algorithm)
- 2.12 Microsoft Azure
- 2.13 MQTT Protocol
- 2.14 NodeRED

2.1 Accelerometer 3 แกน

โมดูล Accelerometer แบบ 3 แกน เป็นโมดูลที่ใช้วัดความเร่งตามแนวแกนด้วยระบบดิจิทัลในการวัดซึ่งโมดูล Accelerometer แบบ 3 แกน ที่ทางผู้จัดทำได้เลือกที่จะนำมาใช้คือโมดูล GY-291 ADXL345 ซึ่งเป็นเซนเซอร์วัดสัญญาณค่าความเร่งของห้องสามแกน โดยจะมีทิศทางแนวแกนแสดงอยู่บนบอร์ด (รูปที่ 2.1) โมดูล ADXL345 เป็นโมดูลที่รองรับการเชื่อมต่อ (Interface) 2 แบบคือ SPI และ I2C โดยสามารถกำหนดค่าความโน้มถ่วงของระบบที่อ่านได้ตั้งแต่ $\pm 2g$, $\pm 4g$, $\pm 8g$, และ $\pm 16g$ (g คือค่าความเร่งจากแรงดึงดูดของโลก) ADXL345 เป็นระบบวัดความเร่งแบบ 3 แกนที่มีขนาดเล็ก แต่มีความละเอียดสูง และมีการใช้พลังงานที่ต่ำ ADXL345 สามารถใช้วัดได้ทั้งความเร่ง ความเร็วแบบเดนามิกที่เกิดจากการเคลื่อนไหวหรือแรงกระแทก และการเร่งความเร็วแบบคงที่ เช่นการเอียงโดย ADXL345 สามารถวัดความเปลี่ยนแปลงของศานั้นอยกว่า 1.0 องศาได้ การสื่อสารกับไมโครคอนโทรลเลอร์รุ่นใหม่ ๆ จะใช้โปรโตคอล I2C ที่ใช้เพียงขา SCL และ SDA



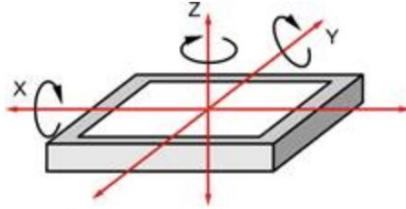
รูปที่ 2.1 Accelerometer module ADXL345

คุณสมบัติของชิปวัดค่าความเร่ง ADXL345

- ใช้ไฟเลี้ยงในช่วงกว้าง 2.0 - 3.6 V
- กินพลังงานต่ำ ($23 \mu A$ ที่ไฟเลี้ยง 2.5 V)
- สามารถเลือกความละเอียดในการวัดผ่านคำสั่งที่รับผ่านการสื่อสารแบบอนุกรมได้
- ข้อมูลที่อ่านได้มีความละเอียดสูง $0.004 g/LSB$ (13 บิต)
- สามารถนำมาเชื่อมต่อได้ทั้งแบบ SPI และ I2C

การวัดความเร่งแบบ 3 แกน

การวัดมุมเอียงของเซนเซอร์สามารถทำได้ด้วยการวัดความเร่งตามแนวแกนในแต่ละแกนตามปกติแล้วเราจะสามารถนิยามการเอียงได้ด้วยการนำมาเทียบกับแนวแกน หรือระบบพิกัดอ้างอิง โดยจะบอกความเอียงออกมากเป็นมุมที่หมุนเอียงไปตามแนวแกนนั้น ๆ ซึ่งการเอียงในแต่ละทิศทางจะมีลักษณะในการกำหนดทิศทาง (รูปที่ 2.2)



รูปที่ 2.2 ทิศทางการเอียงในมุมต่าง ๆ เปรียบเหมือนแกน X,Y,Z

ที่มา : <http://suwitkiravittaya.eng.chula.ac.th/>

จากรูปที่ 2.2 จะเห็นได้ว่ารูปแบบแนวแกนในการวัดความเร่งของโมดูล ADXL345 ตามแนวแกนทั้ง 3 ซึ่งจะสามารถแสดงผลได้แม่นในขณะที่ตัวโมดูลอยู่นิ่ง ๆ ค่าความเร่งในแต่ละแกนควรจะมีค่าออกมานะเป็นศูนย์ แต่ในความเป็นจริงค่าที่ได้ออกมาจะมีค่าไม่เท่ากับ 0 ทั้งหมด เนื่องจากยังมีแรงโน้มถ่วงของโลกอยู่กระทำอยู่ เช่น ถ้าตั้งโมดูลให้แกน Z ตั้งฉากกับพื้นโลก แกน X และ Y จะมีค่าเป็น 0 แต่ว่าแกน Z จะไม่เป็น 0 เพราะมีแรงโน้มถ่วงของโลก ดังนั้นค่าที่ได้จากแกน Z จึงมีค่าประมาณ 9.81 m/s^2

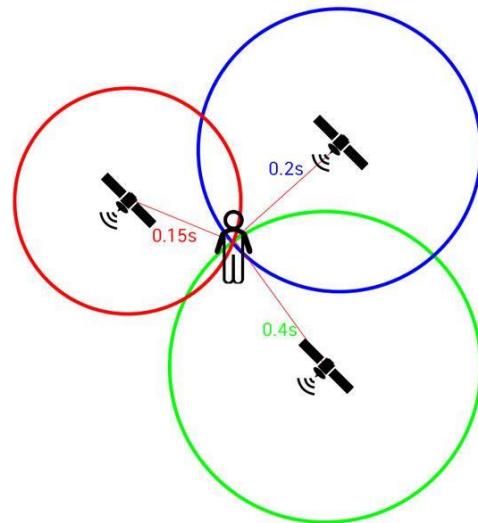
2.2 โมดูลจีพีเอส NEO-6M

ระบบระบุตำแหน่งบนพื้นโลก หรือ Global Positioning System (GPS) เป็นระบบกิจการวิทยุนำทางผ่านดาวเทียม (Radio Navigation-Satellite Service) ที่ทำการส่งคลื่นสัญญาณความถี่จากดาวเทียมลงมาสู่อุปกรณ์บนพื้นโลก ทำให้สามารถทำงานได้ทุกพื้นที่บนโลก และ ทุกสภาพอากาศ โดยจุดเริ่มต้นของโครงการนี้เริ่มต้นมาจากกระทรวงกลาโหมสหรัฐอเมริกา (United States Department of Defence) ซึ่งเริ่มโครงการเมื่อปี 1973 โดยออกแบบให้ทั้งระบบจะมีดาวเทียมไม่น้อยกว่า 24 ดวง โดยที่ดาวเทียมดวงแรกถูกยิงขึ้นไปในเดือนกุมภาพันธ์ ปี ค.ศ.1978 และถูกจำกัดการใช้แค่ในทางทหารเท่านั้น ต่อมาในยุค 1980s จึงได้มีการเปิดให้ประชาชนทั่วไปได้ใช้งานและระบบได้มีการพัฒนาอย่างต่อเนื่องในปัจจุบัน (ที่มา : <http://fitrox.lnwshop.com>) มีดาวเทียมในวงโคจร ทั้งหมด 31 ดวง และได้ปรับให้มีความแม่นยำที่คลาดเคลื่อนอยู่ที่ $30 \text{ cm.} - 5 \text{ m.}$ เพื่อความปลอดภัยทางด้านความมั่นคง

Global Positioning System (GPS) คือ ระบบระบุตำแหน่งของสหัสกรีที่ใช้สัญญาณจากดาวเทียมระบบแรกของโลก ยังมีระบบอื่นๆที่ประเทคโนโลยีต่างๆ ได้พัฒนาขึ้นมาใช้เช่น เช่น GLONASS, Galileo และ BDS เป็นต้น มีคำที่เรียกใช้รวมว่า Global Navigation Satellite System (GNSS) แปลว่า ระบบเครือข่ายดาวเทียมนำทางที่ครอบคลุมโลก เป็นคำกลางที่ใช้เรียกร่วมโครงข่ายดาวเทียมนำทางต่าง ๆ ที่มีในปัจจุบัน และโครงข่ายที่จะเกิดขึ้นในอนาคต

GNSS เป็นระบบโครงข่ายดาวเทียมที่ทำหน้าที่ส่งสัญญาณ (Broadcast) ของตัวดาวเทียมเองที่ประกอบไปด้วย ข้อมูลตำแหน่งของดาวเทียมและเวลาขณะส่ง (Time Stamp) ให้ตัวรับสัญญาณ (Receiver) โดยเวลาที่ส่งจะมีความละเอียดสูงมากถึงระดับนาโน (nano) เนื่องจากดาวเทียมสัญญาณต้องส่งด้วยคลื่นวิทยุ (Radio Wave) ที่มีความเร็วเท่ากับความเร็วแสง สัญญาณจึงมาถึงตัวรับสัญญาณ จะมีดีเลย์นานาโนวินาที-ไมโครวินาที (nanosecond-microsecond) จากทฤษฎีสัมพันธภาพของไอน์สไตน์เรื่อง ความโน้มถ่วงทำให้เวลาเดินช้าลง (Gravitational time dilation) ที่ระบุไว้ว่าเหตุการณ์ ๆ หนึ่งที่ถูกสังเกตการณ์จากผู้สังเกตการณ์สองคนที่อยู่ในพื้นที่ที่มี Gravitational potential ต่างกันจะเห็นเหตุการณ์ไม่พร้อมกัน โดยที่คนที่อยู่ในพื้นที่ที่มี Gravitational potential ต่ำกว่าเวลาจะเดินช้ากว่าในขณะที่ผู้สังเกตการณ์ในพื้นที่ที่มี Gravitational potential สูงเวลาจะเดินเร็วกว่า นั่นหมายความว่าหากดาวที่มี Gravitational potential ต่ำกว่าบนโลกเวลาจะเดินช้ากว่า ทำให้การใช้นาฬิกาที่มีความแม่นยำเพิ่มขึ้นและเที่ยงตรงมากขึ้นจะช่วยลดความคลาดเคลื่อนแบบสุ่ม (Random error) ระหว่างการนับเวลาในดาวเทียมและช่วยให้การรีเซ็ตเวลาภัยสถานีภาคพื้นดินง่ายขึ้นและเที่ยงตรงมากขึ้น ที่ดาวเทียมต้องปรับค่าเวลาใหม่ตามสถานีภาคพื้นดินก็ เพราะว่าเรื่องของการยืดหดของเวลา (Time dilation) นั่นเองที่เวลาในดาวเทียมจะค่อย ๆ คลาดเคลื่อนจากเวลางานโลกไปทีละนิด ๆ ไปเรื่อย ๆ จนทำให้เวลาผิดเพี้ยนไป ซึ่งเราจะอธิบายว่าทำไมแค่เวลาผิดเพี้ยนนิดหน่อยมันจึงร้ายแรงมาก นาฬิกาที่แม่นยำและเที่ยงตรงนี้มีชื่อว่า นาฬิกาอะตอม (Atom Clock) ซึ่งดาวเทียมทุกดวงมีนาฬิกานี้และเป็นเหตุผลที่ความแม่นยำของนาฬิกาบน ดาวเทียม GNSS นั้นสำคัญมาก ๆ เพราะว่าตัว Time Stamp ที่ส่งมาภับสัญญาณระบุตำแหน่งถูกใช้ในการคำนวณตำแหน่งบนตัวรับสัญญาณนั้นเอง หากคลาดเคลื่อนไปเพียงแค่ระดับ nanosecond ก็อาจทำให้ตำแหน่งที่คำนวณออกมาได้คลาดเคลื่อนไปเป็นเมตร ๆ ได้นั่นเอง และหากเรามีนาฬิกาที่แม่นยำระดับนี้เราจะไม่สามารถใช้ระบบระบุตำแหน่งของเรา ระหว่างขั้บรถ หรืออยู่บนรถได้เลย เพราะว่ามันจะคลาดเคลื่อน

โดยในโครงการนี้ใช้ตัวรับสัญญาณโมดูลจีพีเอส NEO-6M ซึ่งรับสัญญาณดาวเทียมจากโครงข่าย ดาวเทียมจีพีเอส มีดาวเทียม 28 ดวงสำรอง 4 ดวงใช้งานจริง 24 ดวง โดยแบ่งเป็น 6 วงโคจร แต่ละวงโคจรจะมี ดาวเทียม 4 ดวง การโคจรจะทำมุ่งเอียง 55 องศาภัยเส้นศูนย์สูตร (Equator) มีรัศมีวงโคจร 20,162.81 กิโลเมตร จากพื้นโลก ดาวเทียมแต่ละดวงใช้เวลาในการโคจรรอบโลก 12 ชั่วโมง โดยตัวโมดูลจะต้องรับสัญญาณได้อย่างน้อย 3 ดวง ขึ้นไป จึงจะสามารถระบุตำแหน่งได้ บอกอีก 1 ดวงเพื่อใช้ซิงค์ (sync) เวลาในนาฬิกาอะตอม (Atomic Clock) สามารถรับสัญญาณได้มากกว่านี้ก็ได้ เพราะจะทำให้มีความแม่นยำมากขึ้นด้วย และโปรเซสเซอร์ในโมดูลจะประมวลผลหาระยะห่างระหว่างดาวเทียมแต่ละดวงกับโมดูล โดยการนำเวลา ขณะส่ง (Time Stamp) ของดาวเทียม และค่าดีเลย์ ของสัญญาณที่มาถึงโมดูล คำนวณค่าที่มาเปรียบเทียบกับสัญญาณนาฬิกาของโมดูล



รูปที่ 2.3 จำลองการรับสัญญาณจากดาวเทียม
ที่มา : <http://fitrox.lnwshop.com>

โมดูลจีพีเอสคือชิ้นส่วนอุปกรณ์รับสัญญาณจากดาวเทียมจีพีเอส ที่สามารถร่วมกับบอร์ดแวร์ ๆ เพื่อให้สามารถรับข้อมูลจากดาวเทียมจีพีเอส อุปกรณ์ที่ใช้จีพีเอส ทั่วไปมักจะต้องมีโมดูลจีพีเอส ซึ่งจะมีฟังก์ชันจีพีเอส ขึ้นพื้นฐานอยู่แล้ว โดยประกอบไปด้วย เสาอากาศซึ่งเป็นตัวรับสัญญาณในหลายช่องสัญญาณ และทำหน้าที่ในการคำนวนในส่วนของการรับส่งข้อมูลระยะทาง เวลา ที่ส่งไปแล้วถอดรหัสข้อมูลเหล่านั้นออกมาเป็นพิกัดที่ส่งจากดาวเทียมอีกที โดยไปรษณีย์ที่อยู่ใน โมดูลจีพีเอสจะทำหน้าที่จัดการกับข้อมูลเหล่านั้น และรายงานออกมาเป็นตำแหน่ง พิกัด ความเร็ว และข้อมูลสำคัญต่าง ๆ ต่อไป ดังนั้น ขีดความสามารถของ โมดูลจีพีเอสในการคำนวนพิกัด การส่งสัญญาณ หรือการประมวลผล ความคลาดเคลื่อนของพิกัด การส่งข้อมูลที่ผิดพลาดจากตัวอุปกรณ์เอง จะทำให้เกิดการรายงานตำแหน่งที่ผิดเพี้ยนได้ โดยในจุดนี้ขึ้นอยู่กับตัวผู้ผลิต หรือบริษัทที่พัฒนา โมดูลจีพีเอสตัวนั้น ๆ ด้วย



รูปที่ 2.4 โมดูลจีพีเอส NEO-6M

โดยทั่วไปอุปกรณ์ที่ติดจีพีเอส จะที่ใช้ฟังก์ชันจีพีเอส ได้แก่ โทรศัพท์มือถือ กล้อง อุปกรณ์ติดตาม ชิปนาวูร์นำวิวิช หรืออุปกรณ์ติดตามสัตว์ เป็นต้น หรือคือ โมดูลจีพีเอสเป็นส่วนประกอบของอุปกรณ์จีพีเอส นั้นเอง โดยอุปกรณ์จีพีเอส ส่วนใหญ่มีจีพีเอสติดตาม (GPS Tracking Devices) และจีพีเอส นำทาง (GPS Navigation Devices) ในอุปกรณ์ จีพีเอสติดตาม จำเป็นต้องมีโมดูลจีพีเอส และโมดูลจีพีเอสแต่ในส่วนของ จีพีเอสนำทาง จะใช้เพียงโมดูลจีพีเอสเท่านั้น โมดูลจีพีเอสที่ใช้ในปัจจุบันมีหลากหลายแบบแต่ทุกตัวจะต้องรับส่งสัญญาณมาจากดาวเทียมจีพีเอส

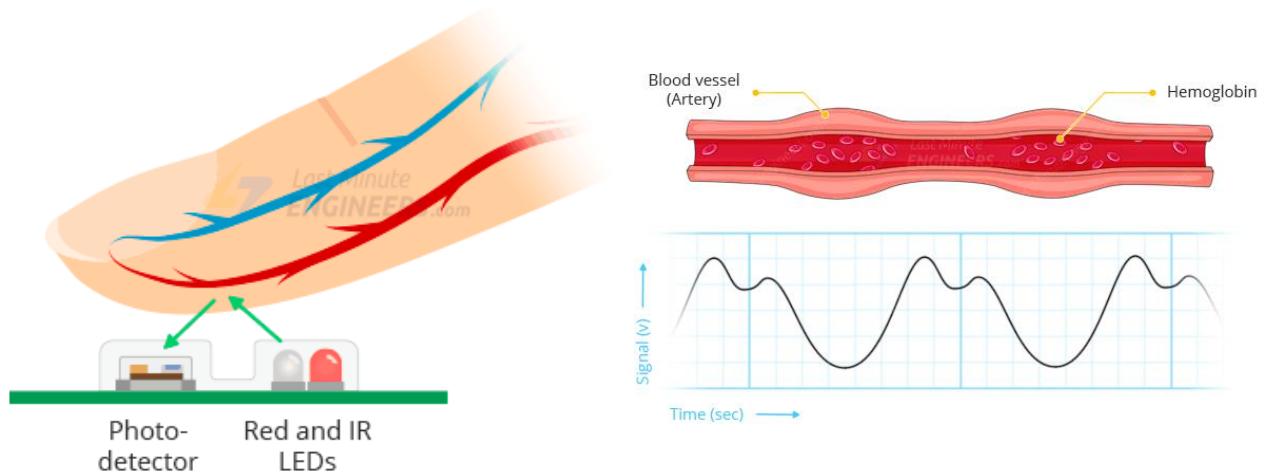
2.3 MAX30102

MAX30102 เป็นเซ็นเซอร์วัดออกซิเจนในเลือดแบบบูรณาการและเซ็นเซอร์ใบโอซิเจนเซอร์ตรวจวัดอัตราการเต้นของหัวใจร่วมกับ เอลอีดีสีแดง, เอลอีดีอินฟราเรด, เครื่องตรวจจับแสง, อุปกรณ์อปติกัล และวงจรอิเล็กทรอนิกส์สัญญาณรับกวนต่ำพร้อมการลดแสงโดยรอบ MAX30102 ใช้แหล่งจ่ายไฟ 1.8V และแหล่งจ่ายไฟ 3.3V แยกต่างหากสำหรับ LED ภายในการวัดอัตราการเต้นของหัวใจและเก็บออกซิเจนในเลือด เซ็นเซอร์วัดค่าได้ถูกต้องเมื่อสัมผัสส่วนของผิวหนังที่บาง เช่น ที่นิ้ว ติ่งหู และข้อมือ เป็นต้น ใช้โปรโตคอลการสื่อสารที่เข้ากันได้กับ I2C สามารถส่งค่าที่รวบรวมไว้ไปยัง Arduino, KL25Z และไมโครคอนโทรลเลอร์อื่น ๆ สำหรับการคำนวณอัตราการเต้นของหัวใจและออกซิเจนในเลือด นอกจากนี้ จิบปังสามารถปิดโมดูลผ่านซอฟต์แวร์ได้ด้วยโดยกระแสไฟขณะสแตนด์บายใกล้ศูนย์ และแหล่งจ่ายไฟจะยังคงอยู่ เนื่องจากประสิทธิภาพที่ยอดเยี่ยม จิปนี้จึงถูกใช้อย่างแพร่หลายในโทรศัพท์มือถือ Samsung Galaxy S series ซึ่งรวมมาตรวัดแก้วเพื่อกำจัดการรบกวนของแสงภายนอกและภายในอย่างมีประสิทธิภาพและให้ประสิทธิภาพที่เชื่อถือได้ที่สุด



รูปที่ 2.5 MAX30102

หลักการวัดอัตราการเต้นของชีพจรเซ็นเซอร์ คือ เป็นการยิงแสงแสงอินฟราเรด (IR) และ เอลอีดีสีแดง (Red LED) ไปที่ปลายนิ้วหรือส่วนผิวหนังที่บาง แล้วสะท้อนกลับมาที่ตัวตรวจจับแสง (Photodetector) เพื่อวัดค่าความเข้มแสง แล้วแปลงค่าเป็นสัญญาณไฟฟ้า ทำให้เราสามารถได้ค่าอัตราการเต้นของชีพจรถได้ เนื่องจากเส้นเลือดฝอยแดงมีอิมิโนโกรบินเป็นกลุ่ม ซึ่งเกิดจากลักษณะการปลั๊มเลือดของหัวใจ จึงทำความเข้มแสงที่สะท้อนกลับมีค่าที่ต่างกัน



รูปที่ 2.6 ตัวอย่างวิธีการใช้งานและหลักการทำงานของเซ็นเซอร์ MAX30120

<https://lastminuteengineers.com/max30102>

2.4 FSR402

เซ็นเซอร์ความดันแบบยืดหยุ่น (หรือที่เรียกว่า FSR) เป็นเซ็นเซอร์ความต้านทานความไวสูงแบบบางพิเศษ (ความหนาปกติประมาณ 0.3 มม.) เมื่อใช้แรงกดบนพื้นที่การตรวจจับความต้านทานของเซ็นเซอร์จะเปลี่ยนไป ยิ่งแรงดันมาก ความต้านทานก็จะยิ่งต่ำลง ในเวลาเดียวกันจะทำให้แรงดันไฟฟ้าออกเปลี่ยนแปลงยิ่งแรงดันมากเท่าใด แรงดันไฟฟ้าออกก็จะยิ่งมากขึ้น เซ็นเซอร์ประเภทนี้ใช้เป็นหลักในการวัดแนวโน้มการเปลี่ยนแปลงแรงดันและการกระจายแรงดันในพื้นที่ (แผ่นที่แรงดัน) เซ็นเซอร์ความดันนี้มีการใช้งานมากมาย เช่น การตรวจจับการยืดเคiskeenthunynt การวัดการเดินของมนุษย์และสัตว์ การวัดการนั่งรถเข็น พื้นรองเท้าวัดแรงดันเครื่องดนตรีอิเล็กทรอนิกส์ ถุงมือชกมวยอัจฉริยะ และอื่น ๆ



รูปที่ 2.7 FSR402

2.5 ESP8266 และ NodeMCU



รูปที่ 2.8 ESP8266

<https://www.elektor.com/>

ESP8266 (รูปที่ 2.8) เป็นชิ้นของชิปไอซีบอร์ดของโมดูลชิ่งไอซี ไม่มีพื้นที่โปรแกรม (flash memory) ในตัว ทำให้ต้องใช้อิซีภายนอก (external flash memory) ในการเก็บโปรแกรม โดยที่ใช้การเชื่อมต่อผ่านโปรโตคอล SPI ซึ่งสามารถทำให้โมดูล ESP8266 มีพื้นที่โปรแกรมมากกว่าไอซีไมโครคอนโทรลเลอร์เบอร์อื่นๆ ESP8266 ทำงานที่แรงดันไฟฟ้า 3.3V-3.6V การนำไปใช้งานร่วมกับเซ็นเซอร์อื่น ๆ ที่ใช้แรงดัน 5V ต้องใช้วงจรแบ่งแรงดันมาช่วยเพื่อไม่ให้โมดูลพังเสียหาย กระแสที่โมดูลใช้งานสูงสุดคือ 200mA ความถี่วิสตัล 40MHz ทำให้มีเวลาในการอุปกรณ์ที่ทำงานรวดเร็วตามความถี่ เช่น LCD ทำการแสดงผลข้อมูลรวดเร็วกว่าไมโครคอนโทรลเลอร์ยอดนิยมอย่าง Arduino หาก



รูปที่ 2.9 NodeMCU

<https://shopee.co.th>

NodeMCU (รูปที่ 2.9) คือแพลตฟอร์มหนึ่งที่ใช้ช่วยในการสร้างโปรเจค Internet of Things (IoT) ที่ประกอบไปด้วย Development Kit (ตัวบอร์ด) และ Firmware (ซอฟต์แวร์บอร์ด) ที่เป็น open source สามารถเขียนโปรแกรมด้วยภาษา Lua ได้ทำให้ใช้งานได้ง่ายขึ้นมาพร้อมกับโมดูล WiFi (ESP8266) ซึ่งเป็นหัวใจสำคัญในการใช้เชื่อมตอกับอินเตอร์เน็ตนั่นเองตัวโมดูล ESP8266 นั้นมีอยู่ด้วยกันหลายรุ่นมากตั้งแต่รุ่นเวอร์ชันแรก ที่เป็น ESP-01 ໄหลไปเรื่อย ๆ จนปัจจุบันมีถึง ESP-12 แล้วและที่ตั้งอยู่ใน NodeMCU version แรกนั้นก็เป็น ESP-12 แต่ใน version2 นั้นจะใช้เป็น ESP-12E แทนซึ่งการใช้งานโดยรวมก็ไม่แตกต่างกันมากนัก NodeMCU นั้นมีลักษณะคล้ายกับ Arduino ตรงที่มีพอร์ต Input Output built in มาในตัวสามารถเขียนโปรแกรมคอมโทร

อุปกรณ์ I/O ได้โดยไม่ต้องผ่านอุปกรณ์อื่น ๆ และเมื่อไม่นานมานี้มีนักพัฒนาที่สามารถทำให้ Arduino IDE ใช้งานร่วมกับ NodeMCU ได้จึงทำให้ใช้ภาษา C/C++ ในการเขียนโปรแกรมได้ทำให้เราสามารถใช้งานมันได้หลากหลายมากยิ่งขึ้น NodeMCU ตัวนี้สามารถทำอะไรได้หลายอย่างมากโดยเฉพาะเรื่อง ที่เกี่ยวข้องกับ IoT ไม่ว่าจะเป็นการทำ Web Server ขนาดเล็กการควบคุมการเปิดปิดไฟผ่าน WiFi และอื่น ๆ อีกมากมาย

2.6 Raspberry Pi

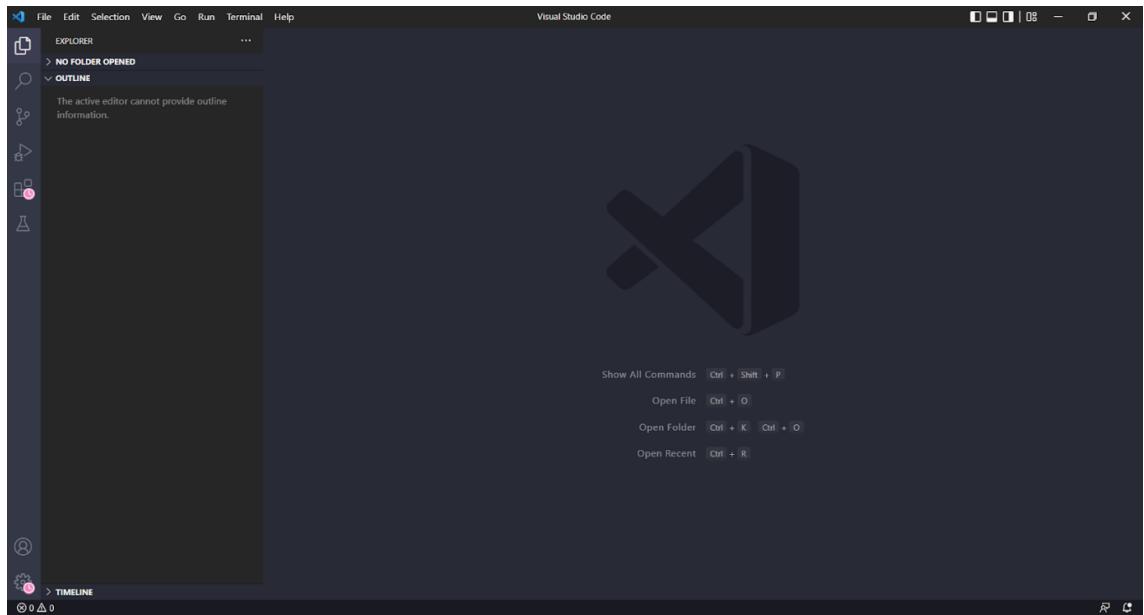
บอร์ด Raspberry Pi เป็นบอร์ดคอมพิวเตอร์ขนาดเล็ก (Single Board Computer: SBC) ที่สามารถเชื่อมต่อ กับหน้าจอแสดงผลได้โดยตรงผ่านพอร์ต HDMI และสามารถเชื่อมต่อ กับ USB Mouse หรือคีย์บอร์ด ได้อีก จุดเด่นของบอร์ด Raspberry Pi คือตัวบอร์ดมาพร้อมกับ WiFi โมดูลในตัวที่สามารถเชื่อมต่อผ่าน WiFi หรือเชื่อม ต่ออินเตอร์เน็ตโดยตรงผ่านสาย LAN ก็ได้ อีกทั้งบอร์ด Raspberry Pi ยังเป็นบอร์ดที่สามารถรองรับระบบปฏิบัติการ ระบบปฏิบัติการภาษาทางคอมพิวเตอร์ได้หลากหลายภาษาไม่ว่าจะเป็น VB.NET, C++, HTML, JavaScript, CSS, PHP และ Python เป็นต้น อีกทั้งเพื่อใช้งานได้เหมือนเครื่องคอมพิวเตอร์ขนาดเล็กบอร์ด Raspberry Pi จึงสามารถ ใช้ทำงานเอกสารจำพวก Word, Excel หรือการใช้ซอฟต์แวร์ ในเชื่อมต่ออินเตอร์เน็ตเว็บเบราว์เซอร์ ทำให้สามารถ ทำงานร่วมกับไฟล์มัลติมีเดียต่าง ๆ ได้ทั้ง อดิโอ และ วิดีโอ



รูปที่ 2.10 บอร์ด Raspberry Pi

2.7 โปรแกรม Visual Studio Code

โปรแกรม Visual Studio Code เป็นโปรแกรมประเภท Editor ที่ไว้ช่วยในการปรับแต่งโค้ดซึ่งได้รับ การพัฒนาโดยบริษัทไมโครซอฟท์ โดยโปรแกรม Visual Studio Code เป็นโปรแกรมที่สามารถรองรับได้หลาย ภาษา ทั้ง VB.NET, C++, C#, Java, Python เป็นต้น รองรับทั้งทางไมโครซอฟต์ได้ทำการพัฒนาโปรแกรม และ ภาษาที่นิยมควบคู่เพื่อให้ใช้งานได้ร่วมกัน สามารถใช้สร้างโปรแกรมที่รันทำงานอยู่บนระบบปฏิบัติการ Windows เช่น โปรแกรมคำนวณเลข โปรแกรมฐานข้อมูล Microsoft Access, Microsoft SQL Server และยังสามารถใช้ ในการแก้ไขปัญหาต่าง ๆ ได้อย่างรวดเร็วและมีประสิทธิภาพสูง (รูปที่ 2.11)



รูปที่ 2.11 หน้าจอโปรแกรม Visual Studio Code

จากรูปที่ 2.11 หน้าจอโปรแกรม Visual Studio Code จะเห็นได้ว่าบริเวณด้านซ้ายบนจะเป็นส่วนที่ไว้ใช้สำหรับการสร้างไฟล์ และ ไว้สำหรับเปิดเมนูต่าง ๆ รวมถึงแสดงว่าในโปรเจค่มีส่วนประกอบอะไรบ้าง สำหรับบริเวณตรงกลางเป็นส่วนสำหรับใช้ในการเขียนโปรแกรม โดยโปรแกรมนี้สามารถนำมาใช้เขียนเพื่อพัฒนาเว็บไซต์ด้วยภาษา HTML, JavaScript, CSS และ PHP

2.8 ภาษาไพธอน (Python)

ภาษาไพธอน (Python) เป็นภาษาหนึ่งที่ใช้สำหรับเขียนโปรแกรมเหมือนกับ C++ หรือ JavaScript โดย Python ถือได้ว่าเป็นภาษาที่ได้รับความนิยมมากเป็นอันดับที่ 2 ของโลกรองลงมาจาก JavaScript เรียกได้ว่า Python เป็นภาษาที่มีการใช้กันอย่างกว้างขวาง มีรูปแบบภาษาเข้าใจง่ายถูกพัฒนาขึ้นมาโดยให้ใช้งานได้ในหลากหลายแพลตฟอร์มมีชุดคำสั่งเสริมหรือไลบรารีเฉพาะตัว โดยเฉพาะไลบรารีที่เกี่ยวข้องกับการประมวลผล ทำให้การเขียนโปรแกรมสำหรับการพยากรณ์ข้อมูลต่าง ๆ เพื่อแสดงคำตอบที่มีความเป็นไปได้ได้ดีกว่าภาษาอื่น ซึ่งภาษา Python สามารถทำงานได้ทั้งระบบ Linux, Windows, Unix หรือระบบปฏิบัติการอื่น ๆ และยังสามารถนำมายังภาษาโปรแกรมเพิ่มเติมการทำงานในอนาคตได้อีกด้วย นอกจากนี้ภาษา Python ยังมีการประมวลผลแบบอินเทอร์พรีเตอร์ (Interpreter) สามารถเห็นผลลัพธ์ได้ทันที แต่เมื่อข้อจำกัดคือเมื่อมีข้อผิดพลาดในการเขียนโปรแกรมเกิดขึ้น โปรแกรมจะหยุดการทำงานทันที เมื่อเทียบกับภาษาอื่น ๆ แล้วภาษา Python สามารถเรียนรู้ได้ดีกว่า

2.9 Scikit-learn

Scikit-learn เป็นไลบรารีของการเขียนโปรแกรม ถูกพัฒนาขึ้นโดยบริษัท David Cournapeau โดยมีจุดเด่น ด้านการเรียนรู้ จัดหมวดหมู่ และทำนายปัจจุบัน Scikit-learn เป็นแพ็กเกจที่รวมไลบรารีด้าน Machine Learning หรือ AI เอาไว้เป็นจำนวนมาก โดยมีส่วนที่ทำงานในด้านของ Data Model. Scikit-learn เป็นไลบรารีที่ทำงาน ข้ามฟอร์มได้อีกทั้งยังสามารถใช้งานได้ฟรี โดยในปัจจุบัน Scikit-learn มีผู้นำมารับผิดชอบต่อเป็นจำนวนมาก จึงทำให้มีการค้นหาข้อมูลได้ง่าย ผู้จัดทำจึงนำ Scikit-learn มาใช้ในการพัฒนาโครงการงานนี้ โดยเลือกใช้อัลกอริทึมมาตรฐาน ที่อยู่ใน Scikit-learn มา 3 อัลกอริทึม ได้แก่ อัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm) , อัลกอริทึมเพื่อนบ้านใกล้สุด (K-Nearest Neighbor Algorithm หรือ KNN) , อัลกอริทึมความหน้าจะเป็น (Gaussian Naive Bayes Algorithm) โดยพื้นฐานแล้วทั้ง 3 อัลกอริทึมนั้นมีหลักการที่เหมือนกันคือการทดสอบ แล้วตรวจสอบคำตอบที่ได้

2.10 การทำ Machine Learning

ขั้นตอนการสร้าง Machine learning คือการสร้างโมเดลเพื่อใช้สำหรับการทำนาย โดยการเก็บข้อมูลมาทำการฝึกสอน (Train) และทดสอบ (Test)

- Train หรือการใช้ข้อมูลสำหรับการสอนเป็นการนำข้อมูลที่มีไปสร้างสมการเพื่ออธิบายรูปแบบของข้อมูลนั้นๆ ออกมากซึ่งในที่นี้จะมีรูปแบบหลากหลายแตกต่างกันออกไปขึ้นอยู่กับรูปแบบที่เลือกใช้งาน
- Test หรือการทดสอบข้อมูล เป็นการนำข้อมูลที่มีมาป้อนให้กับสมการหรือโมเดลที่เราได้ทำการเทรนไว้แล้ว เพื่อคำนวณหาประสิทธิภาพในการทำนาย

สำหรับการ เทส-เทวน โมเดล สำหรับในครั้งนี้ทางผู้จัดทำได้นำมาใช้สำหรับตรวจสอบพฤติกรรมการทรงตัว ของรถจักรยานยนต์ในขณะขับขี่หรือจอดรถโดยจะมีการแบ่งแยกพฤติกรรมตามลักษณะการทรงตัวของรถได้ 5 ลักษณะดังนี้ 1.อุบัติเหตุ (Accident) 2.ลงเนิน (Down hill) 3.จอด (Park) 4.ทางตรง (Straight) 5.ขึ้นเนิน (Up hill) และเก็บข้อมูลตามแต่ละลักษณะอย่างละ 100 ตัวอย่าง รวมทั้งหมดเป็น 500 ตัวอย่าง ด้วยข้อมูล ประเภทไฟล์แบบ .csv เพื่อใช้ในการ เทวน-เทส โมเดล ในครั้งนี้โดยเฉพาะด้วยการใช้ Accelerometer 3 แกน คือ X, Y, Z

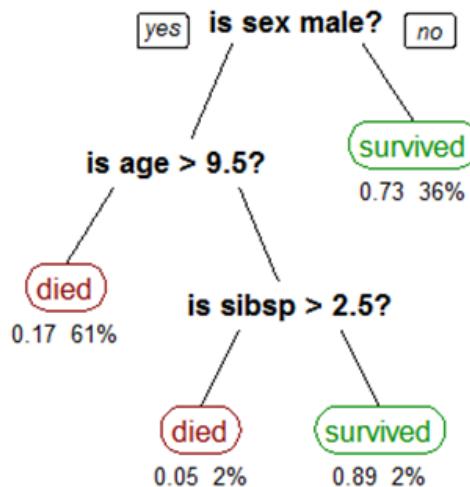
ตารางที่ 2.1 ตัวอย่างค่า Accelerometer 3 แกน

| Order | X-axis | Y-axis | Z-axis | Behavior |
|-------|--------|--------|--------|-----------|
| 1 | 0.06 | 0.04 | 9.99 | Park |
| 2 | -0.51 | 0.04 | 10.04 | Park |
| 3 | -0.51 | 0 | 10.12 | Park |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 100 | -0.04 | -2.67 | 9.69 | Park |
| 101 | -0.16 | 0.55 | 9.92 | Straight |
| 102 | -1.18 | 2.57 | 9.89 | Straight |
| 103 | -0.2 | 1.26 | 10.41 | Straight |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 200 | -0.2 | 1.26 | 10.41 | Straight |
| 201 | 2.9 | 0.08 | 6.94 | Up hill |
| 202 | 2.98 | 0 | 7.73 | Up hill |
| 203 | 4.43 | 0.31 | 7.02 | Up hill |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 300 | 1.96 | -0.39 | 8.22 | Up hill |
| 301 | -6.28 | -1.06 | 7.45 | Down hill |
| 302 | -6.08 | -2.59 | 6.98 | Down hill |
| 303 | -6.83 | -3.02 | 7.41 | Down hill |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 400 | -1.73 | 2.39 | 9.38 | Down hill |
| 401 | 0.04 | 9.89 | -3.06 | Accident |
| 402 | 0.08 | 10.16 | -3.18 | Accident |
| 403 | 0.31 | 9.65 | -3.45 | Accident |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 500 | 0.08 | 10.16 | -3.18 | Accident |

จากตารางที่ 2.1 เป็นตัวอย่างข้อมูลที่เก็บมาจากการทรงตัวของรถจักรยานยนต์ในขณะนั้น ๆ

2.10.1 อัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm)

อัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm) เป็นอัลกอริทึมที่ช่วยในการวิเคราะห์ข้อมูลความเป็นไปได้ที่แตกแขนงออกไปเพื่อเน้นรากของต้นไม้จนนำไปสู่ข้อสรุปที่จะใช้ในการตัดสินใจ โดยอัลกอริทึมต้นไม้ตัดสินใจ มีประโยชน์ในการแยกย่อยข้อมูลออกไปสรุปผลของการตัดสินใจเพื่อเปลี่ยนจากข้อมูลที่มีความซับซ้อนให้ง่ายต่อความเข้าใจมากขึ้น ปัจจุบันอัลกอริทึมต้นไม้ตัดสินใจเป็นที่นิยมมากขึ้นสำหรับการใช้งานที่หลากหลายไม่ว่าจะเป็นในด้าน การแพทย์ ธุรกิจ การเขียนโปรแกรม หรือการสร้างเครื่องมือที่เรียนรู้ได้ด้วยตัวเอง การสร้างระบบผู้เชี่ยวชาญ ฯลฯ



รูปที่ 2.12 อัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm)

ที่มา : <https://towardsdatascience.com>

จากรูปที่ 2.12 จะเห็นได้ว่าการแต่งแขนงของข้อมูลโดยเริ่มจากการพิจารณาข้อมูลเริ่มแรกซึ่งจะ เรียกว่า โหนดแรก (Root node) ถ้าข้อมูลที่ได้ตรงเงื่อนไขนั้น การตัดสินใจก็จะวิ่งไปทางขวาของ Root node ที่เรียกว่า ปมลูก (Child node) ถ้าข้อมูลที่ได้ตรงตามเงื่อนไขของ Child node ก็จะถือว่าสิ้นสุด หรือก็คือโหนดใบ (Leaf node) จากจุด Root node ข้อมูลที่ไม่ตรงกับเงื่อนไขก็จะวิ่งไปทางซ้ายไปยังจุด Child node อีกอันซึ่งจะทำการติดเครื่องหมายคำนามไว้ ทำแบบนี้ไปเรื่อย ๆ จนกว่าจะได้คำตอบที่ต้องการ

$$G_i = 1 - \sum_{k=1}^n p_{i,k}^2$$

สมการที่ : 1

สมการที่ใช้ในการคำนวณเป็นดังนี้โดย $P_{i,k}$ คือสัดส่วนที่แสดงจำนวนของการข้อมูลที่อยู่ใน Node ที่ i ที่อยู่ในคลาส k ภาระ

ในส่วนของการทำงานของтренเมตออกเป็น 2 ส่วนโดยเลือกคลาส k และเงื่อนไข tk เช่น petal length น้อยกว่าเท่ากับ 2.45 โดยคันหาคู่ของ k และ tk ที่จะได้ Node ที่ "บริสุทธิ์" ที่สุด นั่นคือมีค่า Gini ต่ำที่สุด นั่นเอง เราสามารถแสดง Cost function ที่สอดคล้องกับเงื่อนไขนี้ได้ดังนี้

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

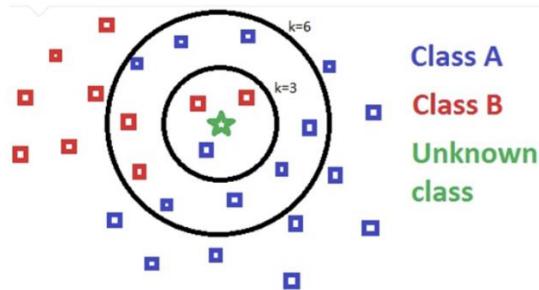
สมการที่ : 2

- Gleft/right คือ Gini ของข้อมูลชุดซ้ายและขวาที่ถูกแบ่ง
- mleft/right คือจำนวนรายการข้อมูลในชุดซ้ายและขวา
- m คือจำนวนรายการข้อมูลทั้งหมดใน Node นั้น

จากนั้นแยกข้อมูลแต่ละชุดย่อยออกเป็นสองชุดและทำขั้วไปแบบนี้เรื่อยๆ จนกระทั่งถึงความลึก max-dept ที่กำหนด หรือจนกระทั่งไม่พบค่า k และ tk ที่จะลดความไม่บริสุทธิ์ได้อีกต่อไป

2.10.2 อัลกอริทึมเพื่อบ้านใกล้สุด (K-Nearest Neighbors Algorithm)

อัลกอริทึมเพื่อบ้านใกล้สุด (K-Nearest Neighbors Algorithm หรือ KNN) เป็นอัลกอริทึมที่ช่วยในการแบ่งคลาสสำหรับใช้ในการจัดหมวดหมู่ของข้อมูล (Classification) โดยจะทำการเปรียบเทียบข้อมูลที่ใช้ทดสอบแล้วนำไปตรวจสอบกับข้อมูลอื่นว่ามีความเหมือนหรือต่างกันมากน้อยขนาดไหน หากข้อมูลที่ใช้ในการทดสอบนั้นอยู่ใกล้ข้อมูลใดมากที่สุด ระบบจะให้คำตอบเป็นเหมือนคำตอบของข้อมูลที่อยู่ใกล้ที่สุดนั้นลักษณะการทำงานแบบมีเม็ดให้ใช้ข้อมูลชุดเรียนรู้ (Training data) ในการสร้างแบบจำลองแต่จะใช้ข้อมูลนี้มาเป็นตัวแบบจำลอง而已



รูปที่ 2.13 อัลกอริทึมเพื่อบ้านใกล้สุด (K-Nearest Neighbor algorithm)

ที่มา : <https://kongruksiam.medium.com>

จากรูปที่ 2.13 คือการพิจารณาข้อมูลมีความหน้าจะเป็นที่จะเป็นว่าจะอยู่ในคลาสไหนมากที่สุดโดยจากรูปจะเห็นได้ว่ามีจุดสีน้ำเงิน (Class A), จุดที่แดง (Class B) มีการกระจายตัวอยู่เป็นจำนวนมากโดย อัลกอริทึมเพื่อบ้านใกล้สุดจะทำการกำหนดระยะเป็นรัศมีวงกลมโดยมีข้อมูลที่ต้องการทราบค่าเป็นจุดศูนย์กลางออกไปเป็นชั้น ๆ

จากรูปจะเห็นได้ว่าในวงกลมชั้นแรกข้อมูลที่ต้องการทราบอยู่ในลักษณะ B จำนวน 2 ค่า ในขณะที่คลาส A จำนวน 1 ค่า ดังนั้นคำตอบที่ได้ก็คือข้อมูลที่ต้องการทราบค่าเป็นคลาส B แต่ในกรณีที่ภายนอกนี้มีรัศมีที่เล็กที่สุดไม่มีจุดใด ๆ อยู่ ใกล้เลยระบบจึงทำการคำนวณข้อมูลในรัศมีที่ใหญ่ขึ้น

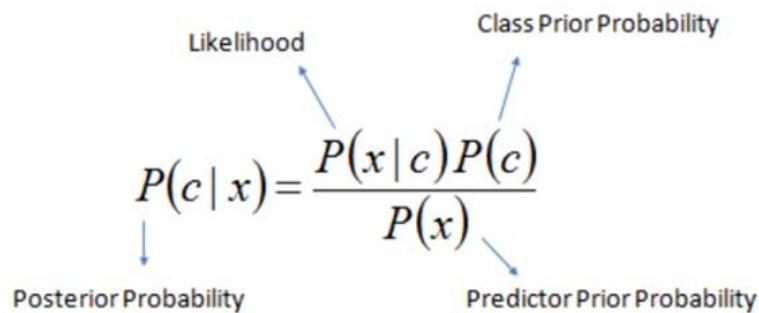
$$dist = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$$

สมการที่ : 3

สมการที่ใช้ในการคำนวนเพื่อวัดระยะห่างความคล้ายคลึงของข้อมูลโดยจะเรียงจากน้อยไปมากหลังจากนั้นจึงจะดูว่าจะกำหนดค่า k เป็นเท่าไร และจึงนำลำดับที่เรียนมาเทียบกับคลาสที่ใกล้ที่สุด

2.10.3 อัลกอริทึมเกาส์เชี่ยน นาอีฟเบย์ (Gaussian Naive Bayes Algorithm)

อัลกอริทึมเกาส์เชี่ยน นาอีฟเบย์ (Gaussian Naive Bayes algorithm) หรือ GaussianNB เป็นอัลกอริทึมที่ใช้การจัดหมวดหมู่โดยใช้หลักความหน้าจะเป็นเข้ามาช่วยในการคำนวน



$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \cdots \times P(x_n | c) \times P(c)$$

สมการที่ : 4

- C คือ Class (ห้อง)
- X คือ Attribute (คุณลักษณะ)
- P คือ Probability (ความหน้าจะเป็น)

$P(c|x)$ Posterior probability คือ ความน่าจะเป็นที่ข้อมูลที่มีแอตทริบิวต์เป็น x จะมีคลาส C

$P(x|c)$ Likelihood คือ ความน่าจะเป็นที่ข้อมูลที่มีคลาส C และมีแอตทริบิวต์ x

$P(c)$ Prior probability คือ จำนวน Class ที่อาจจะเกิดขึ้นจำนวน Class ทั้งหมดหรือความน่าจะเป็นของ Class C

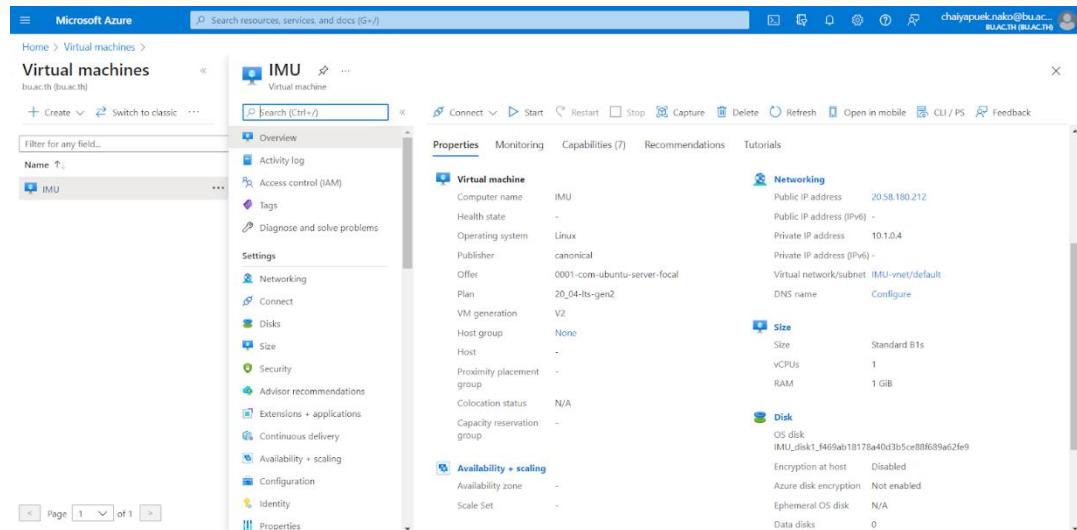
$P(x)$ Predictor Prior probability คือ จำนวน Attribute ทั้งหมด

2.11 Microsoft Azure

Microsoft Azure คือการใช้คลาวด์ให้ทำงานร่วมกันกับระบบโครงสร้างพื้นฐานอย่าง Windows Server และ SQL Server ให้สามารถเข้ามาร่วมต่อเข้ากับ Microsoft Azure ผ่านทาง Windows Admin Center ได้โดยตรงทำให้สามารถใช้เซิร์ฟเวอร์เดิมบนคลาวด์ต่อไปได้โดยใช้ระบบควบคุมเพียงตัวเดียวในการพร้อมทั้งยังใช้ระบบเพิ่มเติมของ Azure ได้ เช่น

- สำรองข้อมูลขึ้นคลาวด์ผ่านบริการ Azure Backup
- เตรียมพร้อมเหตุฉุกเฉินให้สามารถย้ายงานขึ้นคลาวด์ได้ ด้วย Azure Site Recovery
- ซิงก์ไฟล์ขึ้นคลาวด์ด้วย Azure File Sync
- ตรวจสอบความพร้อมของระบบด้วย Azure Monitor

Microsoft Azure คือซอฟต์แวร์ฐานข้อมูลที่ได้รับความไว้วางใจในระดับองค์กรมาเป็นเวลานาน และเป็นระบบฐานข้อมูลของแอพพลิเคชันธุรกิจจำนวนมาก จากความสามารถในการรองรับมาตรฐาน SQL ที่มีความครบถ้วน และยังสามารถเพิ่มประสิทธิภาพในเวอร์ชันใหม่ ๆ ได้ด้วยกระบวนการในการปรับแต่งประสิทธิภาพอัตโนมัติด้วย Machine Learning และมีเครื่องควบคุมความปลอดภัยในตัว



รูปที่ 2.14 Microsoft Azure

2.12 MQTT Protocol

MQTT (Message Queue Telemetry Transport) คือโปรโตคอลในการส่งข้อมูลที่พัฒนามาเพื่อใช้ในระบบ IoT มันทำงานแบบบрокเกอร์-ไคลเอนต์เน็ตเวิร์ก (Broker and Clients Network) ถูกออกแบบให้สามารถส่งข้อมูลแบบเรียลไทม์ (Real-Time) ในปริมาณข้อมูลที่น้อย ทำให้ใช้พลังงานต่ำมั่นคงพัฒนามาจาก TCP/IP ที่มีการส่งข้อมูลแบบ One-To-One ทำให้สิ้นเปลืองทรัพยากรณ์มากซึ่งไม่เหมาะสมกับระบบ IoT เนื่อง จากในระบบ IoT มีการส่งข้อมูลตลอดเวลา และ 1 อุปกรณ์อาจรับหรือส่งข้อมูลไปยังหลายอุปกรณ์ หรือการ ส่งข้อมูลแบบ One-To-All โดยอุปกรณ์ทุกตัวที่ทำการติดตาม (Subscriber) ไปยังหัวข้อ (Topic) ใด ๆ บน Broker จะได้รับข้อมูลที่ Publisher ส่งให้ Topic นั้น ๆ บน Broker ทั้งหมด โดยมั่นคงคิดค้นขึ้นมาในปี 1999 โดย Andy Stanford-Clark จาก IBM และ Arlen Nipper จาก Cirrus Link โดยถูกใช้เพื่อตรวจสอบท่อส่งน้ำมันในทะเลรายโดยเป้าหมายคือ เป็นโปรโตคอลที่มีประสิทธิภาพสูง ส่งข้อมูลขนาดไม่มาก ใช้พลังงานต่ำ เนื่องจากมันต้องเชื่อมต่อผ่านดาวเทียมซึ่งมีราคากาражสูงมากในขณะนั้น

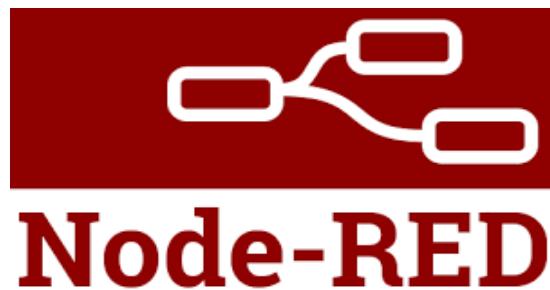
MQTT ประกอบไปด้วย

- Broker (Server) คือตัวกลางในการรับข้อมูลจาก Publisher และส่งข้อมูลให้กับ Subscribe/Clients (Subscriber / Publisher)
- Publisher คือตัวส่งข้อมูลให้กับ Topic ที่อยู่ใน Broker เรียกว่าการ Publish
- Subscriber คือตัวรับข้อมูลจาก Topic ที่อยู่ใน Broker เรียกว่าการ Subscribe
- Topic คือหัวเรื่องที่เราต้องการรับส่งข้อมูล ระหว่าง Publisher กับ Subscriber

หลักการทำงานของ MQTT คือการการรับส่งข้อมูลระหว่าง Server (Broker) และ Clients (Publisher/Subscriber) โดยการประกาศหัวข้อการรับส่งข้อมูลเรียกว่า Topic ไว้ใน Broker จากนั้น Publisher จะส่งข้อมูลไปยัง Topic นั้น ๆ และ Subscriber ก็จะได้รับข้อมูลทั้งหมดใน Topic นั้น ๆ เช่นกัน

2.13 Node RED

Node RED เป็นเครื่องมือที่ออกแบบมาเพื่อใช้กับงาน IoT ในการแสดงข้อมูลในรูปแบบแดชบอร์ด (Dashboard) ทำงานเป็นเว็บเซิร์ฟเวอร์ ถูกพัฒนาขึ้นมาโดย IBM's Emerging Technology Services ใช้การเขียนโปรแกรมแบบฟลว์ (Flow-based Programming) ซึ่งทำให้ง่ายกับเขียนโปรแกรมง่าย ช่วยลดการเสียเวลา และความยุ่งยากในการเขียนโปรแกรมแบบปกติได้มาก



รูปที่ 2.15 Node RED Logo

<http://domoticx.com/node-red-plugin-dashboard/>

การใช้ Node RED เป็นเครื่องมือมีข้อดีคือ สามารถใช้งานได้ง่ายไม่จำเป็นที่จะต้องเป็นคนที่เก่งในด้านข้อมูลที่เขียนโค้ด เป็นเท่านั้น เพราะ NodeRED นั้นมีเครื่องมือที่หลากหลายใช้เลือกใช้งานในหลายๆ อุปกรณ์ ไม่จำเป็นที่จะต้องเขียนโค้ดแม้แต่น้อยเพียงแค่ผู้ใช้งานสามารถเชื่อมงจรของเครื่องมือต่างๆ เข้าด้วยกันได้อย่างถูกต้องก็เพียงพอสำหรับการใช้งานเบื้องต้นได้แล้ว

ในส่วนของการติดตั้งก่อนที่จะทำการติดตั้ง Node RED เป็นเครื่องมือที่จัดเหตุการณ์ขึ้นอยู่กับ Node.js ดังนั้น จะต้องทำการติดตั้ง Node.js ก่อนจึงสามารถใช้งานได้ โดยในส่วนของ Node.js คือ สภาพแวดล้อมที่ใช้สำหรับในการทำงานของภาษา JavaScript นอกเว็บบรัวร์เซอร์ โดยที่ทำงานด้วย V8 engine หมายความว่าสามารถใช้ Node.js ในการพัฒนา แอพพลิเคชันต่างๆ ได้แบบ Command line, Desktop หรือแม้แต่ในส่วนของเว็บเซิร์ฟเวอร์ เอง โดยที่ Node.js จะมี APIs ทำให้ที่สามารถมาใช้งานร่วมกับระบบปฏิบัติการอื่นได้ เช่น การรับค่าและการแสดงผล การอ่านเขียนไฟล์ และการทำงานกับเน็ตเวิร์ค



รูปที่ 2.16 Node.js Logo

https://cdnlogo.com/logo/node-js_36466

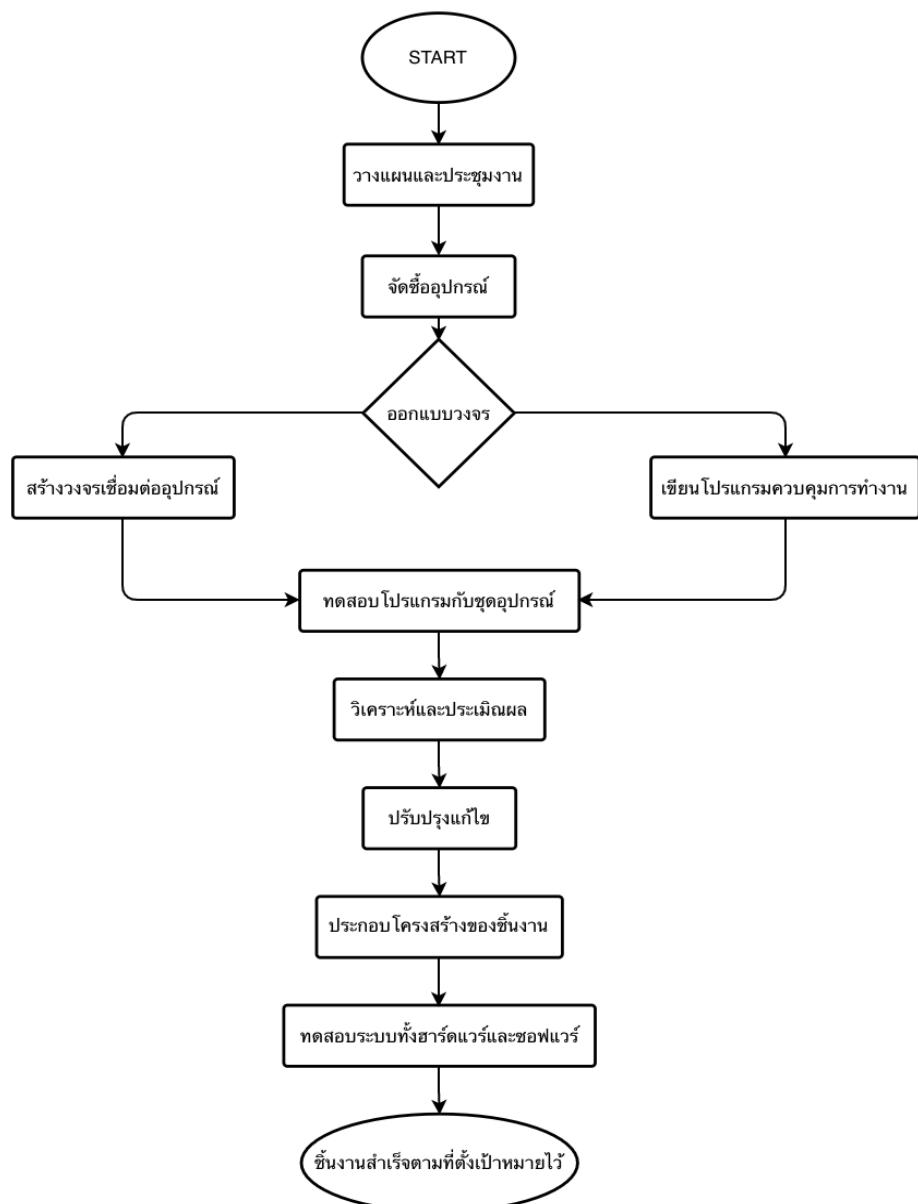
บทที่ 3

ขั้นตอนการออกแบบและพัฒนา

บทนี้จะกล่าวถึงขั้นตอนการออกแบบและพัฒนาในแต่ละส่วนของโครงการเครือข่ายเซ็นเซอร์สำหรับจัดการยานและจัดการยานยนต์ โดยได้ออกแบบและพัฒนาโดยแบ่งเป็นการออกแบบการประเมินผล การส่งข้อมูล

3.1 ขั้นตอนการดำเนินงาน

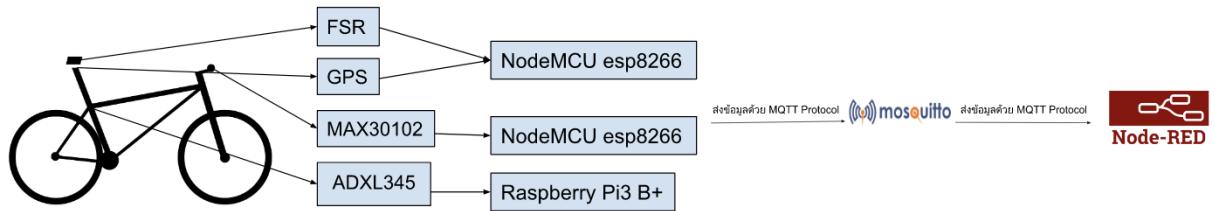
ขั้นตอนการทำงานของ เครือข่าย IoT เซ็นเซอร์สำหรับจัดการยาน และจัดการยานยนต์ แบ่งออกเป็น 3 ขั้นตอนได้ดังนี้



รูปที่ 3.1 ขั้นตอนการทำงานของโครงการ

3.2 ออกแบบระบบการทำงาน

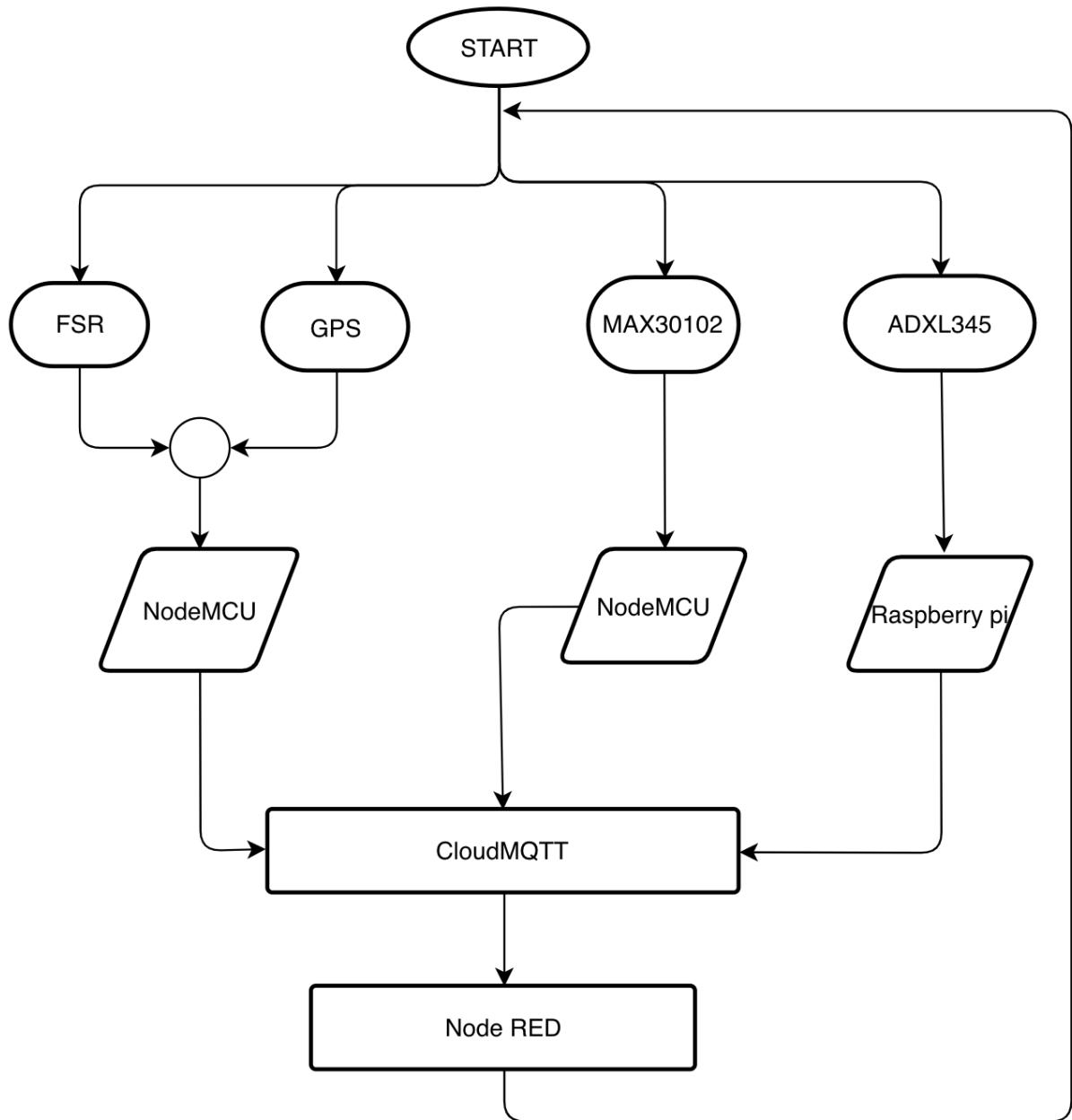
ลำดับการทำงานของระบบเซ็นเซอร์เมื่อทำการติดตั้งบนรถจักรยาน



รูปที่ 3.2 โครงสร้างของระบบ

จากรูปที่ 3.2 เป็นโครงสร้างของระบบและแผนผังการติดตั้งระบบเซ็นเซอร์บนจักรยานโดยใช้ บอร์ด Raspberry Pi ในการรับข้อมูลจาก Accelerometer ประมวลผลในเรื่องของการนำทางพุติกรรมการทรงตัวของ และใช้ NodeMCU 2 ตัว ในการรับข้อมูลจากเซ็นเซอร์โดยโมดูลจีพีเอส กับ FSR Sensor จะส่งข้อมูลมาที่ NodeMCU เดียวกัน และ MAX30102 แยกไป NodeMCU อีกตัว เมื่อบอร์ดคอนโทรลเลอร์รับข้อมูลมาประมวลผล เรียบร้อยแล้ว ข้อมูลที่ประมวลเสร็จจะถูก Publish ไปยังคลาวด์ของ MQTT mosquitto ซึ่งเป็นคลาวด์เซิร์ฟเวอร์ที่ เป็น Blocker ของโปรโตคอล MQTT หลังจากนั้นข้อมูลจะถูก Subscribe ด้วยโปรแกรม Node RED ที่ติดตั้งอยู่ บนคลาวด์เซิร์ฟเวอร์ของ Microsoft Azure เพื่อนำเสนอข้อมูลในรูปแบบ Dashboard

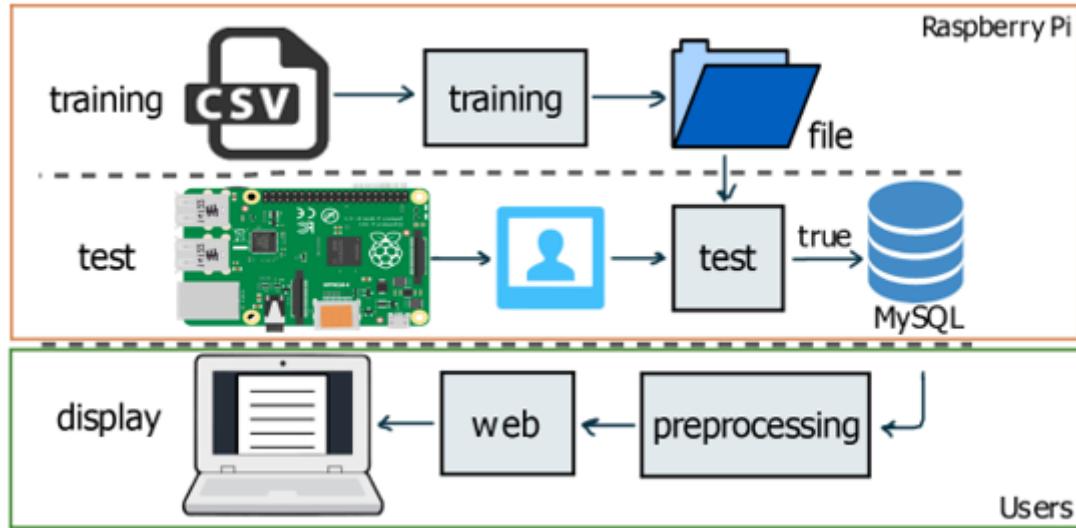
3.3 แผนผังการทำงานของระบบ



รูปที่ 3.3 ผังการทำงานของระบบ

จากรูปที่ 3.3 เป็นการอธิบายถึงเค้าโครงการทำงานของระบบในรูปแบบของผังการทำงาน โดยมีการทำงานอยู่ 4 ส่วน คือมีการเก็บข้อมูลจากเซ็นเซอร์โดยจะมีการแบ่งเป็น 2 ประเภทตามบอร์ดคือ NodeMCU กับ Raspberry Pi เพื่อทำการประมวลผลแล้วส่งข้อมูลด้วยสัญญาณ WiFi โดยใช้โปรโตคอล MQTT ไปยัง CloudMQTT และส่งไปที่ Node RED

3.4 การออกแบบการทำ Machine Learning ทำนายพฤติกรรมการทรงตัวของรถ



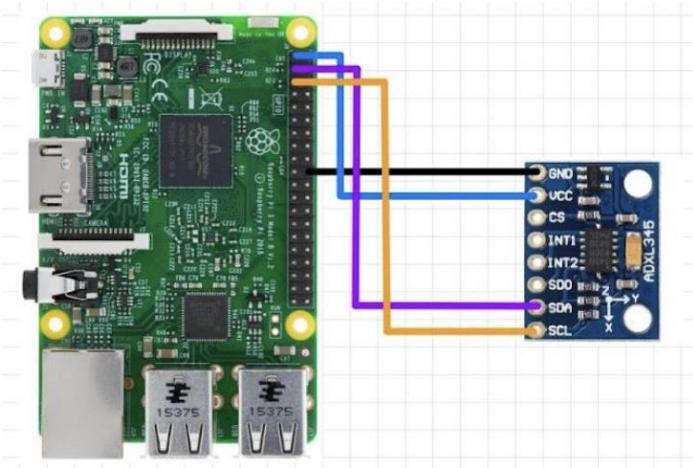
รูปที่ 3.4 ขั้นตอนการทำงานระบบทำนายพฤติกรรม
ที่มา : <https://www.edn.com>

จากรูปที่ 3.4 คือแผนผังการทำงานของระบบทำนายพฤติกรรมการทรงตัวของรถ ซึ่งจะแบ่งออกได้เป็น 3 ส่วน ดังนี้

ส่วนที่ 1 คือการเทรน เป็นการนำข้อมูลจาก Accelerometer ที่บันทึกเป็นไฟล์นามสกุล .CSV พร้อมกับ คำตอบที่สร้างขึ้นมาใส่ในอัลกอริทึม เพื่อให้อัลกอริทึมนั้นรู้จักข้อมูลและคำตอบ แล้วอัลกอริทึมนั้นจะสร้าง ความสัมพันธ์ข้อมูลและคำตอบบนนั้นออกมาในรูปแบบสมการของอัลกอริทึมนั้น

ส่วนที่ 2 คือการทดสอบ เป็นการนำข้อมูลที่ไม่มีคำตอบมาใส่อัลกอริทึมเดิมที่ใช้ในส่วนที่ 1 เพื่อให้อัลกอริทึมใช้ สมการที่ได้จากการเทรน ให้ทำการทำนายข้อมูลที่ได้ออกมาเป็นคำตอบ โดยความแม่นยำของคำตอบที่ได้นั้นจะ ขึ้นอยู่กับการเลือกใช้อัลกอริทึม และคุณภาพของข้อมูลที่ใช้ในการเทรน ซึ่งความแม่นยำของการทดสอบ จะ เปรียบเทียบกับอุปกรณ์เป็นเปอร์เซ็นต์กับคำตอบจริงของข้อมูล

ส่วนที่ 3 คือการส่งผลข้อมูลตอบการทำนายพฤติกรรมการทรงตัวของรถด้วย WiFi โดยการใช้ MQTT Protocol ส่งข้อไปที่ MQTT Broker และใช้ NodeRED ดึงข้อมูลเพื่อแสดงผลในรูปแบบของ Dashboard



รูปที่ 3.5 การต่อวงจรเซ็นเซอร์ Accelerometer เข้ากับ Raspberry Pi

จากรูปที่ 3.5 การต่อวงจรของ Accelerometer กับ Raspberry Pi โดยการต่อไปนี้คือการรับ-ส่ง ข้อมูลแบบ I2C

3.4.1 การรับค่าความเร่งจาก Accelerometer

ในขั้นนี้เป็นการเขียนโค้ดคือรับค่าความเร่ง 3 แกนโดยมีหน้าที่ในการส่งต่อข้อมูลไปยัง Raspberry Pi โดยเลือกใช้protocolรับ-ส่งข้อมูลแบบ I2C

```

File Edit Selection View Go Run Terminal Help
EXPLORER C:\Users\sabrat>gitlab>Downloads>Counter_ADXL.py ...
NO FOLDER OPENED
D:\tree_save.py D:\GaussianNB_save.py
D:\Counter_ADXL.py 3 KNN_SAVE.py
D:\board.I2C() # uses board.SCL and board.SDA
# For ADXL343
accelerometer = adafruit_adxl34x.ADXL343(i2c)
# load the model from disk
model_tree = pickle.load(open('model_IMU_tree.sav', 'rb'))
model_GaussianNB = pickle.load(open('model_IMU_GaussianNB.sav', 'rb'))
model_KNN = pickle.load(open('model_IMU_KNN.sav', 'rb'))
while True:
    array = accelerometer.acceleration
    d = [array[0], array[1], array[2]] # read from IMU sensor
    x_IMU = d
    y_tree = model_tree.predict(x_IMU)
    y_GaussianNB = model_GaussianNB.predict(x_IMU)
    y_KNN = model_KNN.predict(x_IMU)
    CT1 = 0 # 21 = Park
    CT2 = 1 # 22 = Straight
    CT3 = 2 # 23 = Up hill
    CT4 = 3 # 24 = Down hill
    CT5 = 4 # 25 = Accident
    CG1 = 5 # 21 = Park
    CG2 = 6 # 22 = Straight
    CG3 = 7 # 23 = Up hill
    CG4 = 8 # 24 = Down hill
    CG5 = 9 # 25 = Accident
    CK1 = 10 # 21 = Park
    CK2 = 11 # 22 = Straight
    CK3 = 12 # 23 = Up hill
    CK4 = 13 # 24 = Down hill
    CK5 = 14 # 25 = Accident
    y_number = y_tree
    y_ans = y_KNN
    print(y_number, y_ans)
    time.sleep(0.1)

```

รูปที่ 3.6 ตัวอย่าง โค้ดของ Accelerometer

จากรูปที่ 3.6 คือตัวอย่างโค้ดส่วนหนึ่งของ Accelerometer โดยการรับค่าแบบ I2C และจึงทำการรับค่าความเร่งทั้ง 3 แกนออกมาในรูปแบบอารเรย์ (Array)

3.4.2 การเทรน-เทส โมเดล

ในขั้นตอนนี้คือการนำข้อมูลที่เก็บไว้ก่อนหน้านี้ดังตารางที่ 2.1 มาใช้สำหรับการฝึกโมเดลทั้ง 3 แบบ คือ Decision Tree Algorithm, K-Nearest Neighbor Algorithm หรือ KNN และ Gaussian Naive Bayes Algorithm จากนั้นทำการตรวจสอบความถูกต้องตามหัวข้อที่ 2.10.1, 2.10.2, 2.10.3

3.4.2.1 การเทรนโมเดล Decision Tree Algoithm

```

12  data = pd.read_csv('ADXL_IMU.csv')
13
14  data[['X-axis', 'Y-axis']]
15  data[['X-axis', 'Y-axis']].shape
16
17  Straight = data['Behavior']=='Straight'
18
19  x = data.drop(['Behavior'], axis=1)
20  y = data[['Behavior']]
21
22  x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4)
23
24  model = tree.DecisionTreeClassifier()
25
26  model.fit(x_train, y_train)
27  y_predict = model.predict(x_test)
28
29  def plot_confusion_matrix(cm, classes,
30                           normalize=False,
31                           title='Confusion matrix',
32                           cmap=plt.cm.Blues):
33      """
34          This function prints and plots the confusion matrix.
35          Normalization can be applied by setting `normalize=True`.
36      """
37      plt.imshow(cm, interpolation='nearest', cmap=cmap)
38      plt.title(title)
39      plt.colorbar()
40      tick_marks = np.arange(len(classes))
41      plt.xticks(tick_marks, classes, rotation=45)
42      plt.yticks(tick_marks, classes)
43
44      if normalize:
45          cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
46          print("Normalized confusion matrix")
47      else:
48          print('Confusion matrix, without normalization')
49
50      print(cm)
51
52      thresh = cm.max() / 2.
53      for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
54          plt.text(j, i, cm[i, j],
55                   horizontalalignment="center",
56                   color="white" if cm[i, j] > thresh else "black")
57

```

รูปที่ 3.7 โค้ดการเทรนโมเดลแบบ Decision Tree Algorithm

จากรูปที่ 3.7 คือตัวอย่าง โค้ดของการเทรนโมเดลแบบ Decision Tree Algorithm ซึ่งจะมีการเทรนที่ 60% เทสที่ 40% โดยมีอัตราความถูกต้องในการทำนายอยู่ที่ 97.5%

3.4.2.2 การเทรนโมเดล K-Nearest Neighbor Algolithm

```
#Read Data
df=pd.read_csv('ADXL_IMU.csv')
# data
x=df.drop("Behavior",axis=1).values
# outcome data
y=df['Behavior'].values

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4)

knn=KNeighborsClassifier(n_neighbors=8)
#train
knn.fit(x_train,y_train)

#prediction
y_pred=knn.predict(x_test)

print(pd.crosstab(y_test,y_pred,rownames=['Actually'],colnames=['Prediction'],margins=True))

accuracy_score(y_test, y_pred)
print('Prediction accuracy : {}'.format(accuracy_score(y_test, y_pred)))
```

รูปที่ 3.8 โค้ดการเทรนโมเดลแบบ K-Nearest Neighbor Algorithm (KNN)

จากรูปที่ 3.8 คือตัวอย่าง โค้ดของการเทรนโมเดลแบบ K-Nearest Neighbor Algorithm ซึ่งจะมีการเทรนที่ 60% เทสที่ 40% โดยมีอัตราความถูกต้องในการทำนายอยู่ที่ 94%

3.4.2.3 การเทรนโมเดล Gaussian Naive Bayes Algolithm

```
17 #Read Data
18 df=pd.read_csv('ADXL_IMU.csv')
19 # data
20 x=df.drop("Behavior",axis=1).values
21 # outcome data
22 y=df['Behavior'].values
23
24 # train , test
25 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4)
26
27 #model
28 model=GaussianNB()
29 #train
30 model.fit(x_train,y_train)
31
32 #prediction
33 y_pred=model.predict(x_test)
34
35 print(pd.crosstab(y_test,y_pred,rownames=['Actually'],colnames=['Prediction'],margins=True))
36
37 accuracy_score(y_test, y_pred)
38 print('Prediction accuracy : {}'.format(accuracy_score(y_test, y_pred)))
39
```

รูปที่ 3.9 โค้ดการเทรนโมเดลแบบ Gaussian Naive Bayes Algorithm

จากรูปที่ 3.9 คือตัวอย่าง โค้ดของการเทรนโมเดลแบบ Gaussian Naive Bayes Algorithm ซึ่งจะมีการเทรนที่ 60% เทสที่ 40% โดยมีอัตราความถูกต้องในการทำนายอยู่ที่ 96.5%

3.4.2.4 การดึงข้อมูลโนมเดลทั้ง 3 โนมเดล มาใช้

```

11
12 # For ADXL343
13 accelerometer = adafruit_adxl34x.ADXL343(i2c)
14
15 # load the model from disk
16 model_tree = pickle.load(open('model_IMU_tree.sav', 'rb'))
17 model_GaussianNB = pickle.load(open('model_IMU_GaussianNB.sav', 'rb'))
18 model_KNN = pickle.load(open('model_IMU_KNN.sav', 'rb'))
19
20 while True:
21
22     array = accelerometer.acceleration
23
24     d = [[array[0], array[1], array[2]]] # read from IMU sensor
25     x_IMU = d
26
27     y_tree = model_tree.predict(x_IMU)
28     y_GaussianNB = model_GaussianNB.predict(x_IMU)
29     y_KNN = model_KNN.predict(x_IMU)
30

```

รูปที่ 3.10 โค้ดการใช้งานโนมเดล

จากรูปที่ 3.10 คือตัวอย่างโค้ดในการดึงข้อมูลจากโนมเดลที่ทำได้ทำการ tren ข้อมูลทั้ง 3 แบบไว้เพื่อนำมาใช้งานในการทำนายพฤติกรรมในขณะขับขี่จักรยานหรือจักรยานยนต์

3.4.2.5 การเลือกคำตอบที่ถูกต้องที่สุด

```

31     CT1 = 0 # 21 = Park
32     CT2 = 1 # 22 = Straight
33     CT3 = 2 # 23 = Up hill
34     CT4 = 3 # 24 = Down hill
35     CT5 = 4 # 25 = Accident
36
37     CG1 = 5 # 21 = Park
38     CG2 = 6 # 22 = Straight
39     CG3 = 7 # 23 = Up hill
40     CG4 = 8 # 24 = Down hill
41     CG5 = 9 # 25 = Accident
42
43     CK1 = 10 # 21 = Park
44     CK2 = 11 # 22 = Straight
45     CK3 = 12 # 23 = Up hill
46     CK4 = 13 # 24 = Down hill
47     CK5 = 14 # 25 = Accident
48
49     if(y_tree == ['Park']):
50         CT1 = CT1+21
51     if(y_tree == ['Straight']):
52         CT2 = CT2+21
53     if(y_tree == ['up hill']):
54         CT3 = CT3+21
55     if(y_tree == ['Down hill']):
56         CT4 = CT4+21
57     if(y_tree == ['Accident']):
58         CT5 = CT5+21
59
60     if(y_GaussianNB == ['Park']):
61         CG1 = CG1+16
62     if(y_GaussianNB == ['Straight']):
63         CG2 = CG2+16
64     if(y_GaussianNB == ['Up hill']):
65         CG3 = CG3+16
66     if(y_GaussianNB == ['Down hill']):
67         CG4 = CG4+16
68     if(y_GaussianNB == ['Accident']):
69         CG5 = CG5+16
70
71     if(y_KNN == ['Park']):
72         CK1 = CK1+11
73     if(y_KNN == ['Straight']):
74         CK2 = CK2+11
75     if(y_KNN== ['Up hill']):
76         CK3 = CK3+11
77     if(y_KNN== ['Down hill']):
78         CK4 = CK4+11
79     if(y_KNN == ['Accident']):
80         CK5 = CK5+11
81
82     y_Number = Counter([CT1, CT2, CT3, CT4, CT5, CG1, CG2, CG3, CG4, CG5, CK1, CK2, CK3, CK4, CK5])
83     #print(y_Number)
84
85     y_Ans = y_Number.most_common(1)[0][0]
86     print(y_Ans)
87

```

รูปที่ 3.11 Code Counter ที่ใช้ในการเลือกคำตอบที่ถูกที่สุด

จากรูปที่ 3.11 เป็น Code Counter ที่ใช้สำหรับในการเลือกคำตอบที่ถูกต้องที่สุดสำหรับทั้ง 3 โมเดล หรือเป็นโปรแกรมในการเลือกคำตอบที่เหมือนกันมากที่สุดจากทั้ง 3 โมเดล ไม่ก็เป็น 2 ใน 3 ของคำตอบที่มีเปอร์เซ็นความต้องสูงสุด

```

Counter({22: 2, 0: 1, 2: 1, 3: 1, 4: 1, 21: 1, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 12: 1, 13: 1, 14: 1})
[(22, 2)]
22
Straight

```

รูปที่ 3.12 ตัวอย่างการทำงานของโปรแกรม Counter

จากรูปที่ 3.12 คือการนำคำตอบจากทั้ง 3 โมเดล ที่ได้ทำการแทนไว้แล้วด้วยการทดสอบโดยการค่าตัวเลขเข้าไปแทนค่าอาเรย์ (Array) ที่รับมาจาก Accelerometer ดังรูปที่ 3.13

```

d = [[-0.49, -1.16, 9.18]]
x_new = d

```

รูปที่ 3.13 ค่าตัวเลขที่ใช้แทนค่าอาเรย์จากเซ็นเซอร์

3.4.2.6 การส่งข้อมูลคำตอบไปยัง MQTT Broker

จากรูปที่ 3.13 จะเห็นได้ว่าคำตอบของการนำคำตอบของ MQTT Broker ของ Eclipse Mosquitto ต้องมีดูเมนเนม “test.mosquitto.org” ซึ่งเป็นดูเมนเนมของคลาวด์เซิร์ฟเวอร์ของ MQTT Broker นี้

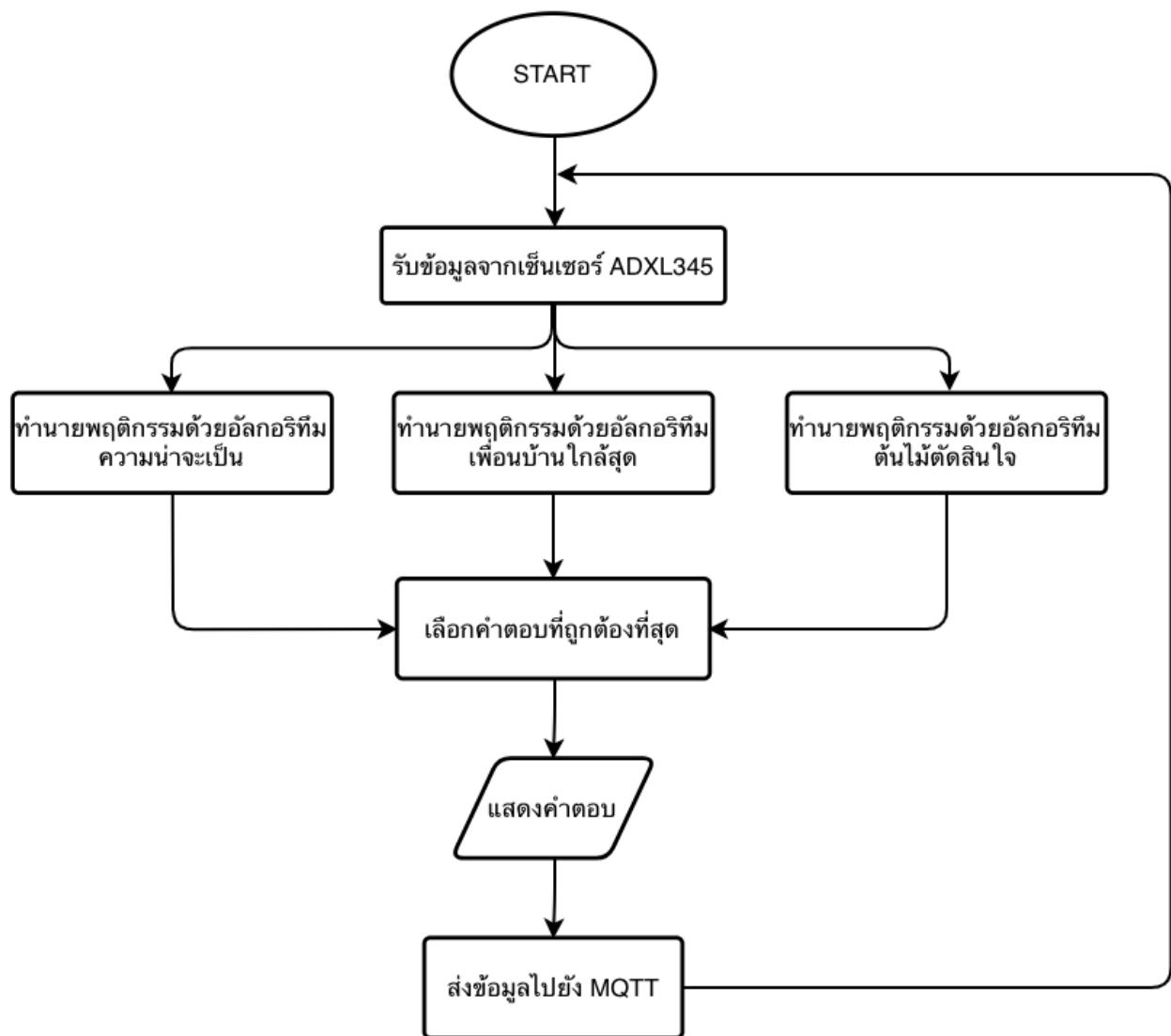
```

88     if(y_Ans == 21):
89         print('Park')
90         pulish.single("academics/IMU_AXDL345",'Park', hostname = "test.mosquitto.org")
91
92     if(y_Ans == 22):
93         print('Straight')
94         pulish.single("academics/IMU_AXDL345",'straight', hostname = "test.mosquitto.org")
95
96     if(y_Ans == 23):
97         print('Up hill')
98         pulish.single("academics/IMU_AXDL345",'Up hill', hostname = "test.mosquitto.org")
99
100    if(y_Ans == 24):
101        print('Down hill')
102        pulish.single("academics/IMU_AXDL345",'Down hill', hostname = "test.mosquitto.org")
103
104    if(y_Ans == 25):
105        print('Accident')
106        pulish.single("academics/IMU_AXDL345",'Accident', hostname = "test.mosquitto.org")
107
108    print("Done")

```

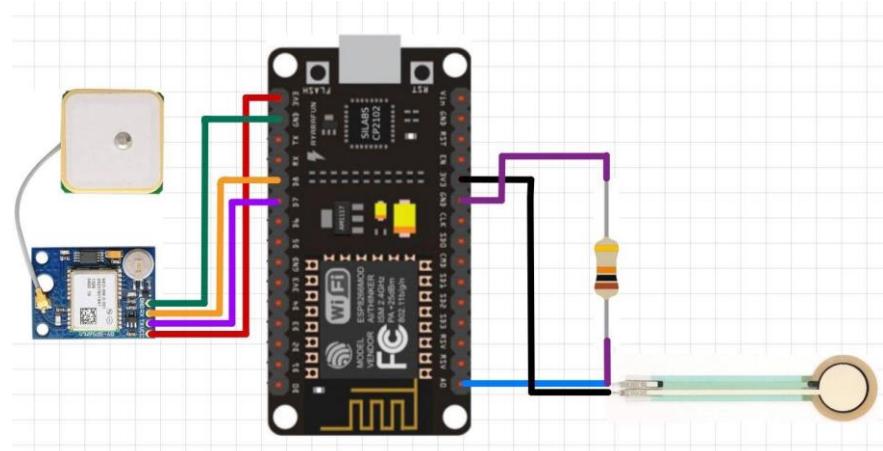
รูปที่ 3.14 โค้ดที่ใช้สำหรับส่งข้อมูลไปยัง MQTT Broker

3.4.3 ผังการทำงานของ Machine Learning



รูปที่ 3.15 ผังการทำงานของ Raspberry Pi

3.5 แผนภาพการต่อวงจรของเซ็นเซอร์จีพีเอส และ FSR เข้ากับ NodeMCU



รูปที่ 3.16 การต่อวงจรของเซ็นเซอร์จีพีเอส และ FSR402 เข้ากับ NodeMCU

จากรูปที่ 3.16 แผนภาพการต่อวงจร (circuit diagram) เซ็นเซอร์จีพีเอส และ FSR402 กับ NodeMCU โดยการต่อวงจรเซ็นเซอร์ GPS ด้วยสายส่งสัญญาณแบบ UART ในส่วนของวงจร FSR402 จะเป็นการส่งสัญญาณแบบอนาล็อกที่ขา A0

```
void GPS() {
    Serial.println("debug");
    if (gps.location.isUpdated()) {
        double lat = gps.location.lat();
        double lng = gps.location.lng();

        char msg[100];
        char val[100];
        dtostrf(gps.location.lat(), 3, 6, val);
        sprintf(msg, "%s", val);
        Serial.println(gps.location.lat());
        client.publish("GPSlat", msg);

        char msg1[100];
        char val1[100];
        dtostrf(gps.location.lng(), 3, 6, val1);
        sprintf(msg1, "%s", val1);
        Serial.println(gps.location.lng());
        client.publish("GPSlng", msg1);

        snprintf(buffer, sizeof(buffer),
                 "%.6f %.6f",
                 lat, lng);
    }
}
```

รูปที่ 3.17 โค้ดการทำงานเซ็นเซอร์จีพีเอส

```

void FSR(){
    fsrReading = analogRead(fsrPin);

    Serial.print("Analog reading = ");
    Serial.print(fsrReading);      // the raw analog reading

    char msg[30];
    char val[10];
    dtostrf(fsrReading, 5, 0, val);
    sprintf(msg, "%s", val);
    client.publish("ForceSensingResistance", msg);

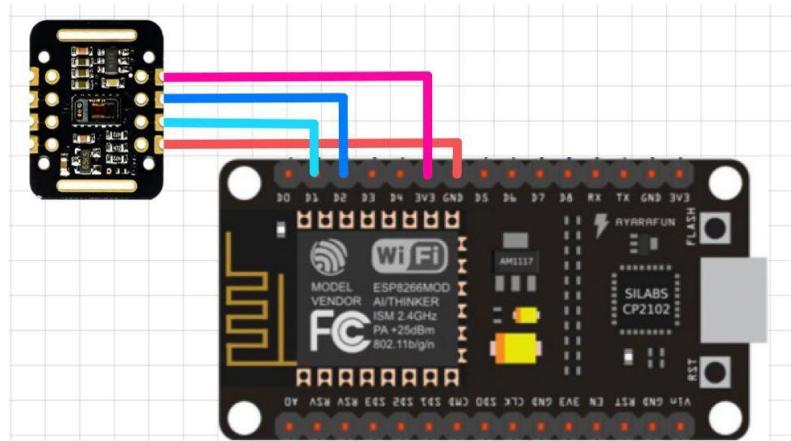
    if (fsrReading < 10) {
        client.publish("FSR1", "ไม่มคนนั่งเบาะ");
    } else if (fsrReading > 500) {
        client.publish("FSR1", "มีคนนั่งเบาะ");
    }
}

```

รูปที่ 3.18 โค้ดการทำงานเซ็นเซอร์ FSR402

จากรูปที่ 3.17 - 3.18 เป็นโค้ดที่เป็นคำสั่งควบคุมการทำงานของเซ็นเซอร์จีพีเอส และ FSR402 โดยในส่วนของเซ็นเซอร์จีพีเอส จะเขียนโค้ดให้แสดงค่าละติจูด (latitude), ลองจิจูด (longitude) และส่วนของ FSR402 จะเขียนโค้ดให้แสดงค่าแรงกดทับที่วัดได้จากเซ็นเซอร์ และบอกสถานะว่ามีคนนั่งเบาะหรือไม่

3.6 แผนภาพการต่อวงจรของเซ็นเซอร์ MAX30102 เข้ากับ NodeMCU



รูปที่ 3.19 การต่อวงจรของเซ็นเซอร์ MAX30102 เข้ากับ NodeMCU

จากรูปที่ 3.19 คือแผนภาพการต่อวงจร (circuit diagram) ของ MAX30102 กับ NodeMCU โดยจะใช้โปรโตคอลส่งสัญญาณแบบ I2C

```

void loop()
{
    if (!client.connected()){
        reconnect();
    }
    long irValue = particleSensor.getIR();

    if (checkForBeat(irValue) == true)
    {
        //We sensed a beat!
        long delta = millis() - lastBeat;
        lastBeat = millis();

        beatsPerMinute = 60 / (delta / 1000.0);

        if (beatsPerMinute < 255 && beatsPerMinute > 20)
        {
            rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the array
            rateSpot %= RATE_SIZE; //Wrap variable

            //Take average of readings
            beatAvg = 0;
            for (byte x = 0 ; x < RATE_SIZE ; x++)
                beatAvg += rates[x];
            beatAvg /= RATE_SIZE;
        }
    }

    Serial.print(irValue);
    beatAvg = 0;
    for (byte x = 0 ; x < RATE_SIZE ; x++)
        beatAvg += rates[x];
    beatAvg /= RATE_SIZE;
}

Serial.print(irValue);
char msg[30];
char val[10];
dtostrf(irValue, 5, 0, val);
sprintf(msg, "%s", val);
client.publish("irValue", msg);

Serial.print("\t");

Serial.print(beatsPerMinute);
char msg3[40];
char val3[20];
dtostrf(beatsPerMinute, 5, 2, val3);
sprintf(msg3, "%s", val3);
if (irValue < 50000){
    client.publish("beatsPerMinute", "0");
}
else if (irValue > 50000){
    client.publish("beatsPerMinute", msg3);
}

Serial.print("\t");
Serial.print(beatAvg);

```

รูปที่ 3.20 คำสั่งการทำางานของวงจร MAX30102

```

        client.publish("beatsPerMinute", "0");
    }
    else if (irValue > 50000){
        client.publish("beatsPerMinute", msg3);
    }

    Serial.print("\t");

    Serial.print(beatAvg);
    char msg2[50];
    char val2[30];
    dtostrf(beatAvg, 2, 0, val2);
    sprintf(msg2, "%s", val2);
    if (irValue < 50000){
        client.publish("beatAvg", "0");
    }
    else if (irValue > 50000){
        client.publish("beatAvg", msg2);
    }
    if (irValue < 50000){
        Serial.print(" No finger?");
        client.publish("state sensor", "No finger?");
    }
    else if (irValue > 50000){
        Serial.print("have finger");
        client.publish("state sensor", "have finger");
    }

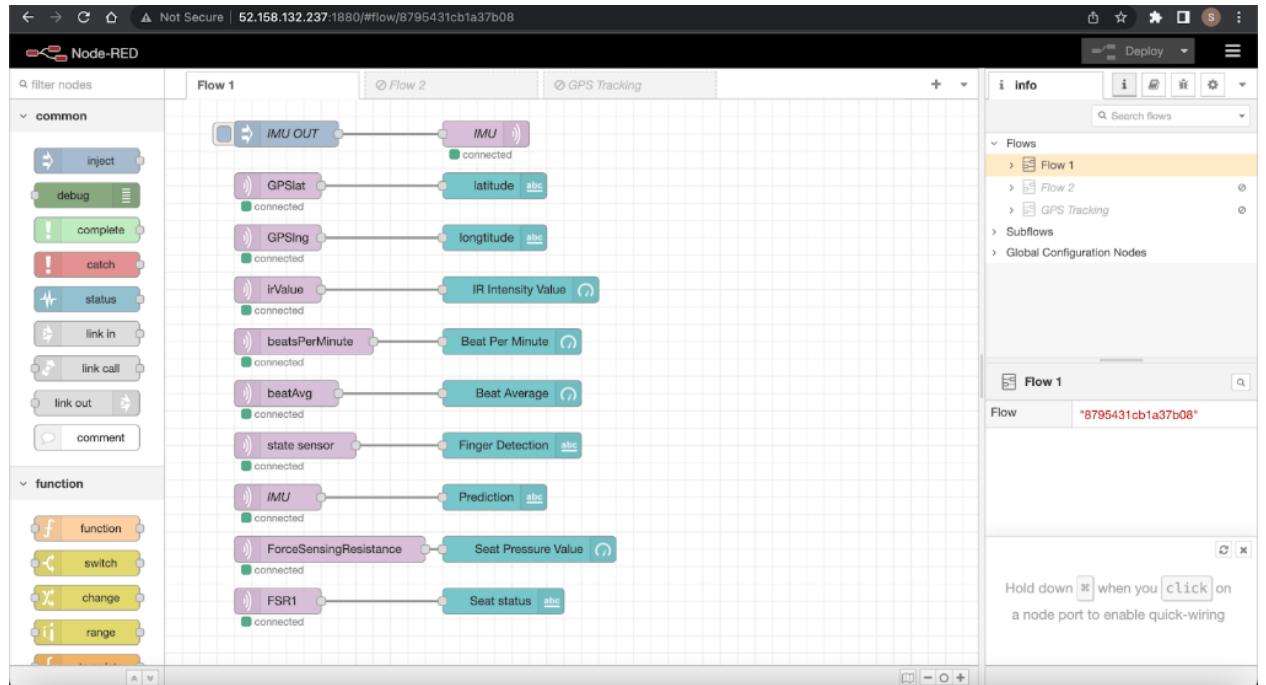
    Serial.println();
}

```

รูปที่ 3.21 คำสั่งการทำงานของวงจร MAX30102 (ต่อ)

จากรูปที่ 3.20 เป็นโค้ดคำสั่งในการทำงานของเซ็นเซอร์ MAX30102 ซึ่งจะเป็นการวัดอัตราการเต้นของหัวใจ แล้วนำค่าที่ได้มาคำนวณเป็นค่าเฉลี่ยการเต้นของหัวใจ โดยข้อมูลที่แสดงมีความเข้มแสง Red LED กับ IR LED อัตราการเต้นของหัวใจ และค่าเฉลี่ยอัตราการเต้นของหัวใจ สถานการณ์เซ็นเซอร์ว่ามีอะไรสัมผัสหน้าเซ็นเซอร์ ซึ่งสามารถเช็คได้จากค่าความเข้มของ Red LED กับ IR LED ที่สะท้อนกลับไปที่ตัวโฟโตไดโตกเตอร์ (Photodetector)

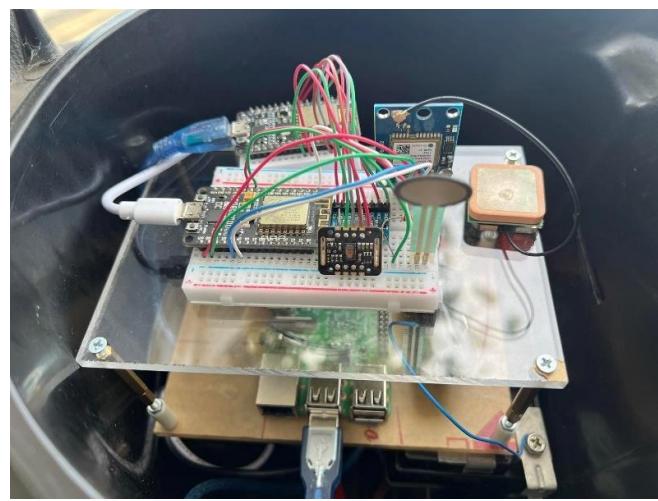
3.7 การเขียนโค้ดดึงข้อมูลจาก MQTT Broker



รูปที่ 3.22 การเขียนโค้ดดึงข้อมูลจาก MQTT Broker

จากที่ 3.21 เป็นการเขียนโค้ด subscribe ข้อมูลจาก MQTT Broker เพื่อนำมาในรูปแบบของ DashBoard

3.8 การทดสอบการทำงานของชิ้นงาน



รูปที่ 3.23 การติดตั้งเพื่อทดสอบการทำงานในสภาพแวดล้อมจริง (1)



รูปที่ 3.24 การติดตั้งเพื่อทดสอบการทำงานในสภาพแวดล้อมจริง (2)

บทที่ 4

ผลการดำเนินงานในโครงการวิศวกรรม

บทนี้จะกล่าวถึงผลการดำเนินงานที่ได้จัดทำไปแล้วโดยจะอธิบายถึงในส่วนของการทำงานพฤติกรรมของรถจักรยานหรือรถจักรยานยนต์ และในส่วนของการแสดงผลผ่าน Dashboard

4.1 ผลการทดสอบการทำงานพฤติกรรม และความถูกต้อง

สำหรับการทำงานพฤติกรรมขณะขับขี่รถจักรยานหรือรถจักรยานยนต์ ทางผู้จัดทำได้นำอัลกอริทึมทั้ง 3 ชนิดมาใช้ เพื่อให้ได้ค่าในการทำงานที่ถูกต้องมากที่สุดโดยใช้ข้อมูลจากตารางที่ 2.1 ดังนี้

4.1.1 อัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm)

ในทั้ง 3 อัลกอริทึมที่เลือกนำมาใช้สามารถกล่าวได้ว่าอัลกอริทึมต้นไม้ตัดสินใจถือเป็นอัลกอริทึมที่มีอัตราความถูกต้องของข้อมูลมากที่สุดโดยมีเปอร์เซ็นต์ความถูกต้องมากถึง 97.5 % โดยในการทำงานของระบบ จะทำการแบ่งข้อมูลออกเป็น 2 กลุ่ม จากข้อมูลทั้ง 500 ค่าคือ เทคนที่ 60% และ เทสที่ 40% กล่าวคือ เทคน 300 ค่า และ เทส 200 ค่าด้วยการสุ่มของระบบ

ตารางที่ 4.1 ตารางความถูกต้องของอัลกอริทึมต้นไม้ตัดสินใจ

| Predict Actual \ Predict Actual | Accident | Down hill | Park | Straight | Up hill | Total |
|---------------------------------------|----------|-----------|------|----------|---------|-------|
| Accident | 37 | 0 | 3 | 0 | 0 | 40 |
| Down hill | 0 | 39 | 0 | 0 | 0 | 39 |
| Park | 0 | 0 | 40 | 2 | 0 | 42 |
| Straight | 0 | 0 | 0 | 42 | 0 | 42 |
| Up hill | 0 | 0 | 0 | 0 | 37 | 37 |
| Total | 37 | 39 | 43 | 44 | 37 | 200 |

จากตารางที่ 4.1 จะแสดงถึงผลของการทำงานพฤติกรรมรถจักรยานยนต์ดังต่อไปนี้คือ 1.อุบัติเหตุ (Accident) 2.ลงเนิน (Down hill) 3.จอด (Park) 4.ทางตรง (Straight) 5.ขึ้นเนิน (Up hill) จากคำตอบทั้ง 5 ประเภทจะได้คำตอบที่ถูกต้องทั้งหมดอยู่ที่ $37 + 39 + 40 + 42 + 37$ รวมเป็น 195 คำตอบ และได้คำตอบที่ผิดพลาดทั้งหมด $3 + 2$

รวมเป็น 5 คำตอบซึ่งได้มีการสุ่ม เทส ทั้งหมด 200 ค่า สามารถคำนวณอัตราความถูกต้องได้ $(195/200) * 100\% = 97.5\%$ จะเห็นได้ว่าความถูกต้องของอัลกอริทึมต้นไม่มีตัดสินใจ ถือว่ามีความหน้าเชื่อถือ

4.1.2 อัลกอริทึมเพื่อนบ้านใกล้สุด (K-Nearest Neighbor Algorithm หรือ KNN)

สำหรับอัลกอริทึมที่ 2 ที่ทางผู้จัดทำได้เลือกใช้ก็คืออัลกอริทึมเพื่อนบ้านใกล้สุดสำหรับอัลกอริทึมนี้จัดได้ว่าเป็นอัลกอริทึมที่มีความถูกต้องน้อยที่สุด โดยมีอัตราความถูกต้องอยู่ที่ 94 % โดยในการทำงานระบบจะทำการแบ่งข้อมูลออกเป็น 2 กลุ่ม จากข้อมูลทั้ง 500 ค่า คือ เทคนิค 60% และ เทสที่ 40% กล่าวคือ เทคนิค 300 ค่าและ เทส 200 ค่าด้วยการสุ่มของระบบ

ตารางที่ 4.2 ตารางความถูกต้องของอัลกอริทึมเพื่อนบ้านใกล้สุด

| Predict \ Actual | Accident | Down hill | Park | Straight | Up hill | Total |
|------------------|----------|-----------|------|----------|---------|-------|
| Accident | 49 | 0 | 0 | 0 | 0 | 49 |
| Down hill | 0 | 39 | 0 | 0 | 0 | 39 |
| Park | 0 | 0 | 38 | 1 | 0 | 39 |
| Straight | 0 | 0 | 11 | 26 | 0 | 37 |
| Up hill | 0 | 0 | 0 | 0 | 36 | 36 |
| Total | 49 | 39 | 49 | 27 | 36 | 200 |

จากตารางที่ 4.2 จะเห็นได้ถึงผลของการทำงานพฤติกรรมรถจักรยานยนต์ดังนี้คือ 1.อุบัติเหตุ (Accident) 2.ลงเนิน (Down hill) 3.จอด (Park) 4.ทางตรง (Straight) 5.ขึ้นเนิน (Up hill) จากคำตอบทั้ง 5 ประเภท จะได้คำตอบที่ถูกต้องทั้งหมดอยู่ที่ $49 + 39 + 38 + 26 + 36 = 188$ คำตอบ และได้คำตอบที่ผิดพลาดทั้งหมด $1 + 11 = 12$ คำตอบซึ่งได้มีการสุ่มเทส ทั้งหมด 200 ค่า สามารถคำนวณอัตราความถูกต้องได้ $(188/200) * 100\% = 94\%$ จะเห็นได้ว่าความถูกต้องของอัลกอริทึมเพื่อนบ้านใกล้สุด ถือว่ามีความหน้าเชื่อถือ

4.1.3 อัลกอริทึมเก้าส์เชี่ยน นาอีฟเบร์ (Gaussian Naive Bayes Algorithm)

อัลกอริทึมเก้าส์เชี่ยน นาอีฟเบร์ สำหรับอัลกอริทึมนี้ถือว่าเป็น อัลกอริทึมที่มีอัตราความถูกต้องมาก เป็นอันดับที่ 2 จากทั้ง 3 อัลกอริทึมโดยมาอัตราความถูกต้องอยู่ที่ 96.5 % โดยในการทำงานระบบจะทำการแบ่งข้อมูลออกเป็น 2 กลุ่ม จากข้อมูลทั้ง 500 ค่าคือ เทคนที่ 60% และ เทสที่ 40% กล่าวคือ เทคน 300 ค่าและ เทส 200 ค่าด้วยการสุ่มของระบบ

ตารางที่ 4.3 ตารางความถูกต้องของอัลกอริทึมเก้าส์เชี่ยน นาอีฟเบร์

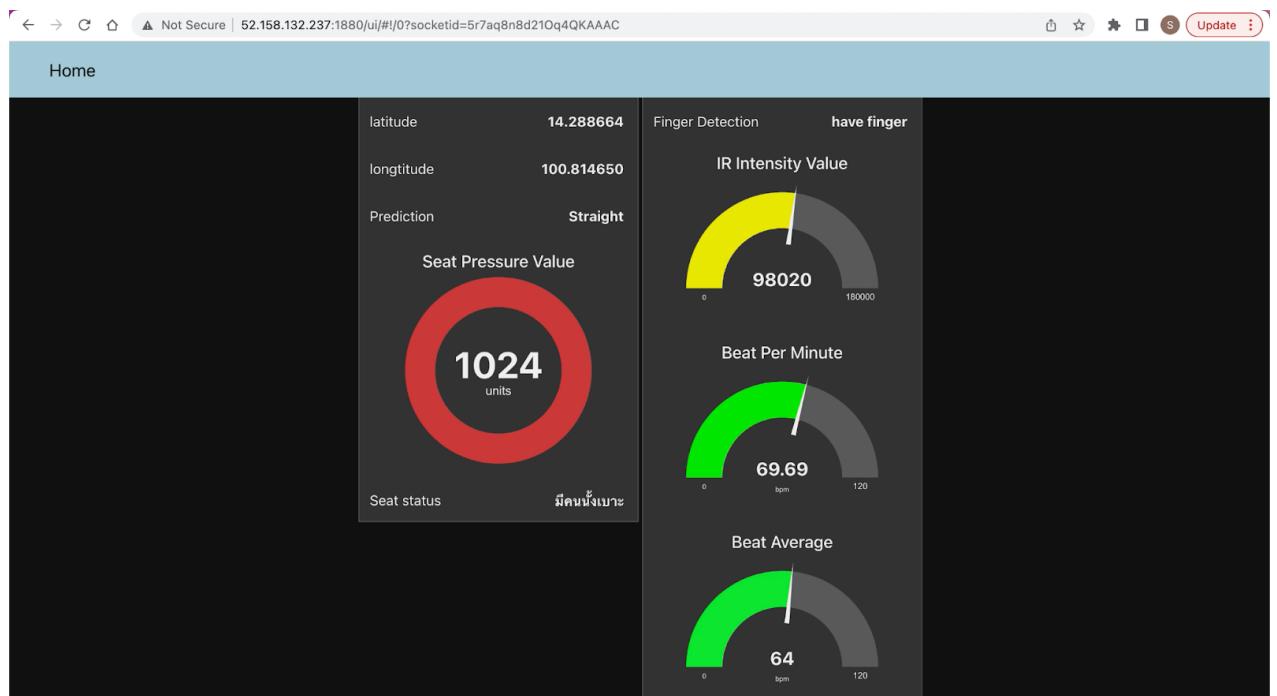
| Predict Actual | Accident | Down hill | Park | Straight | Up hill | Total |
|-------------------|----------|-----------|------|----------|---------|-------|
| Accident | 37 | 0 | 0 | 0 | 0 | 37 |
| Down hill | 0 | 37 | 0 | 0 | 0 | 37 |
| Park | 0 | 0 | 46 | 3 | 0 | 49 |
| Straight | 0 | 0 | 4 | 33 | 0 | 37 |
| Up hill | 0 | 0 | 0 | 0 | 40 | 40 |
| Total | 37 | 37 | 50 | 36 | 40 | 200 |

จากตารางที่ 2.4 จะเห็นได้ถึงผลของการทำนายพฤติกรรมรถจักรยานยนต์ดังนี้คือ 1.อุบัติเหตุ (Accident) 2.ลงเนิน (Down hill) 3.จอด (Park) 4.ทางตรง (Straight) 5.ขึ้นเนิน (Up hill) จากคำตอบทั้ง 5 ประเภทจะได้คำตอบที่ถูกต้องทั้งหมดอยู่ที่ $37 + 37 + 46 + 33 + 40 = 193$ คำตอบ และได้คำตอบที่ผิดพลาดทั้งหมด $3 + 4 = 7$ คำตอบซึ่งได้มีการสุ่มเทส ทั้งหมด 200 ค่าสามารถคำนวณอัตราความถูกต้องได้ $(193/200) * 100\% = 96.5\%$ จะเห็นได้ว่าความถูกต้องของอัลกอริทึมเก้าส์เชี่ยน นาอีฟเบร์ ถือว่ามีความหน้าเชื่อถือ

4.2 กระบวนการทำงานของระบบ

ระบบเซ็นเซอร์จะทำงานตลอดเวลาเมื่อมีไฟเลี้ยงบอร์ด และส่งข้อมูลไปที่ MQTT Broker ได้ตลอดเวลาที่บอร์ดเชื่อมต่อกับสัญญาณ WiFi ได้ โดย Machine Learning จะทำนายพฤติกรรมต่าง ๆ จากลักษณะการทรงตัวของรถด้วยการรับข้อมูลมาจากเซ็นเซอร์ Accelerometer, เซ็นเซอร์จีพีเอส จะจับสัญญาณดาวเทียมเพื่อประมวลผลหาพิกัด ละติจูด (Latitude) ลองจิจูด (Longitude), เซ็นเซอร์ FSR จะบอกสถานะว่ามีคนนั่งเบาะหรือไม่ จากการตรวจจับแรงกดที่เบาะ และเซ็นเซอร์ MAX30102 จะทำงานตลอดเวลาสามารถใช้นี้ไว้แตะที่เซ็นเซอร์ เพื่อวัดอัตราการเต้นของชีพจร แล้วดูข้อมูลผ่านทาง Dashboard ของ Node RED ได้โดยการใส่ออปเปอร์เดรส (IP address) ที่เว็บเบราว์เซอร์

4.3 การแสดงผลของข้อมูล



รูปที่ 4.1 หน้าจอแสดงผลของระบบ

จากรูปที่ 4.1 เป็น Dashboard ของ Node RED ที่แสดงข้อมูลจากเซ็นเซอร์แบบเรียลไทม์ซึ่งจะบอกค่า Latitude, Longitude, การทำนาย (Prediction) ลักษณะการทรงตัวของรถ, ค่าแรงกดที่เบาะ (Seat Pressure Value), ค่าแสดงสถานะ (Seat Status) ของเบาะว่ามีคนนั่งอยู่หรือไม่ และการใช้งานเซ็นเซอร์วัดอัตราการเต้นของชีพจร โดยเมื่อที่วัดถูกสัมผัสที่เซ็นเซอร์ จะมีข้อความว่า “have finger” เกี่ยวกับ IP intensity Value หมายถึงค่าความเข้มแสง RED LED ที่ Photodetector ตรวจจับได้ ค่ามีค่ามากขึ้นเมื่อมีวัตถุมาสัมผัสที่เซ็นเซอร์ และจะสามารถประมวลเป็นค่าอัตราการเต้นของชีพจร (Beat Per Minute) และค่าเฉลี่ยอัตราการเต้นของชีพจร (Beat Average)

บทที่ 5

สรุปผลการดำเนินงานปริญญาอินพนร

บทนี้จะกล่าวภาพรวมของโครงการเพื่อสรุปให้ทักษะและกระบวนการวิเคราะห์โจทย์ปัญหา การออกแบบระบบการทำงานของชิ้นงาน และประสิทธิภาพของผลงาน ที่ได้ทำมาตลอด 1 ปีการศึกษา

5.1 สรุปผลการดำเนินงาน

โครงการวิศวกรรมเรื่อง “IoT Sensors Network for Bicycles and Motorcycles” ผลงานทำงานของระบบที่พัฒนาขึ้นมาใหม่ ใช้งานได้จริง เช่นเซอร์ทุกตัวที่เลือกใช้เป็นประโยชน์ต่อรถได้เป็นอย่างดี สามารถส่งข้อเพื่อทำการรายงานผลและดูแบบเรียลไทม์ได้ ระบบโดยร่วมสามารถแก้ไขได้อย่างคลอดคล้องและแก้ไขได้กับปัญหาที่เจอนในระดับหนึ่ง การเขียนโค้ดสร้างทำงานของระบบ Machine Learning เพื่อทำนายพฤติกรรมการทรงตัวของรถ โดยการใช้อัลกอริทึม 3 ตัว ให้ความถูกต้องในการทำนายเมื่อคิดเป็นเปอร์เซ็นต์ได้ดังนี้ 97.5% , 96.5% และ 94% ถือเป็นเปอร์เซ็นต์ที่ถือว่าสูง และใกล้เคียงกับทุก เป็นตัวบ่งชี้ได้อย่างดีว่ากระบวนการที่ออกแบบระบบโค้ดนั้นจะสามารถทำงานได้ดี สามารถนำไปใช้งานจริง หรือเป็นแนวทางในการพัฒนา Machine Learning ให้ฉลาด และมีประสิทธิภาพที่ดียิ่งขึ้น เพื่อเพิ่มความสะดวกสบาย และความมั่นใจให้กับผู้ใช้งาน

5.2 ข้อจำกัดของโครงการ

- 1 เซ็นเซอร์ IMU มีแค่ Accelerometer ซึ่งขาด gyroscope ทำให้มีข้อจำกัดในการพัฒนา AI
- 2 การทำนายพฤติกรรมขณะขับขี่รถจักรยานหรือรถจักรยานยนต์ ยังมีความถูกต้องไม่ถึง 100 %
- 3 การ starters การทำงาน แล้วออโต้รัน ในส่วนของ Raspberry Pi 3 B+ ใช้เวลาค่อนข้างนาน
- 4 NodeMCU esp8266 มีข้อจำกัดในการเชื่อมต่อ WiFi ที่ต้องเชื่อมต่อตามโค้ดที่เขียนไว้
- 5 เซ็นเซอร์ MAX30102 วัดอัตราการเต้นของหัวใจ ไม่ค่อยเสถียร

5.3 เสนอแนะ

- 1 บอร์ดคอนโทรลเลอร์ที่ใช้ทำ Machine Learning ควรใช้スペคที่สูงกว่านี้ เพื่อเพิ่มประสิทธิภาพของ AI
- 2 Machine Learning ควรทำนายพฤติกรรมได้หลากหลายมากกว่านี้ เพื่อให้ตรงกับสถานการณ์ได้ละเอียดขึ้น
- 3 ควรติดตั้งอุปกรณ์ให้เหมาะสมสมก่วนนี้

5.4 แนวทางการพัฒนาต่อ

- 1 สามารถนำไปพัฒนาเป็นระบบ Machine Learning ที่ทำนายได้แม่นยำและหลากหลายสถานการณ์
- 2 สามารถนำไปพัฒนาต่อเป็นการแจ้งเตือนหากเกิดอุบัติเหตุกับผู้ขับขี่ได้
- 3 สามารถนำไปพัฒนาต่อเป็นระบบติดตามรถในยามที่รถหายหรือลูกขโมย

บรรณานุกรม

- [1] การใช้งานเซ็นเซอร์วัดความเร่งแบบสามแกน - Suwit Kiravittaya. [Online]. ประเทศไทย. 2564.
แหล่งข้อมูลเข้าถึง : <http://suwitkiravittaya.eng.chula.ac.th>
- [2] [Tutorial] การใช้งาน GPS Module (Ublox NEO-6M). [Online]. ประเทศไทย. 2564.
แหล่งข้อมูลเข้าถึง : <http://fitrox.lnwshop.com/article/59/>
- [3] Interfacing MAX30102 Pulse Oximeter and Heart Rate Sensor with Arduino. [Online].
Last minute Engineer. 2561. แหล่งข้อมูลเข้าถึง : <https://lastminuteengineer.com>
- [4] FSR402 เซ็นเซอร์วัดแรงกด ชั้งน้ำหนัก Force Sensing Resistor [Online]. ประเทศไทย. 2563.
แหล่งข้อมูลเข้าถึง : <https://www.arduino4.com/product/685/fsr402>
- [5] Arduino ESP8266 (NodeMCU) - มหาวิทยาลัยธรรมศาสตร์ [Online]. ประเทศไทย. 2555. แหล่งที่เข้าถึง :
<http://narong.ece.engr.tu.ac.th/ei444/document>
- [6] โปรโตคอล MQTT- EnConLab [Online]. ประเทศไทย. 2555. แหล่งที่เข้าถึง :
<https://www.enconlab.com/chiller/index.php/>
- [7] เทคนิคสร้าง Node เพื่อใช้งานกับ Node-RED ให้适合自己之需求 [Online]. ประเทศไทย. 2556.
แหล่งที่เข้าถึง : <https://nickuntitled.com/2021/03/27/customize-node-red>
- [8] Te. เริ่มต้นเขียนโปรแกรมด้วยภาษา Python กันเถอะ. [Online]. ประเทศไทย. 2552. แหล่งที่เข้าถึง :
<http://www.ayarafun.com/2009/08/basic-python-programming-part1>
- [9] Nonthakon Jitchiranant. Machine Learning ด้วย Python : การทำ Decision Tree เพื่อช่วยในการตัดสินใจด้วย scikit-learn. [Online]. ประเทศไทย. 2563. แหล่งที่เข้าถึง :
<https://nonthakon.medium.com/machine-learning-%>
- [10] Mindphp. รู้จักกับ Visual Studio Code (วิชวล สตูดิโอ โคด) โปรแกรมฟรีจากค่ายไมโครซอฟท์. [Online]. ประเทศไทย. 2560. แหล่งที่เข้าถึง :
<https://silllovely.wordpress.com/2013/06/11/%E9%80%A1%E9%80%A1Visual-Studio-Code-%E0%9C%A4%E0%9C%A4/>
- [11] Microsoft. Getting Started. [Online]. U.S.A. 2565. แหล่งที่เข้าถึง :
<https://code.visualstudio.com/docs>
- [12] Achieve.Plus. รู้จักใช้ Scikit-learn เมื่อตนมีโปรเจกต์. [Online]. ประเทศไทย. 2563. แหล่งที่เข้าถึง :
https://docs.opencv.org/3.4.2/d7/d8b/tutorial_py_face_detection.html
- [13] Prashant Gupta. Decision Trees in Machine Learning. [Online]. U.S.A. 2560. แหล่งที่เข้าถึง :
<https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>
- [14] ชิตพงษ์ กิตตินราดร. Decision Tree. [Online]. ประเทศไทย. 2563. แหล่งที่เข้าถึง :
<https://guopai.github.io/ml-blog09.html?fbclid>

- [15] Kong Ruksiam. **สรุป Machine Learning(EP.4)**. [Online]. ประเทศไทย. 2563. แหล่งที่เข้าถึง : <https://kongruksiam.medium.com>
- [16] Kong Ruksiam. **สรุป Machine Learning(EP.5)**. [Online]. ประเทศไทย. 2563. แหล่งที่เข้าถึง : <https://kongruksiam.medium.com>
- [17] Dexter. **Run a Program On Your Raspberry Pi At Startup**. [Online]. ประเทศไทย. 2560. แหล่งที่เข้าถึง : <https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/>
- [18] 3 เหตุผลที่คุณควรเลือก Azure เป็นโครงสร้างคลาวด์ขององค์กร. [Online]. ประเทศไทย. 2559. แหล่งที่เข้าถึง : <https://www.vstecs.co.th>
- [19] เทคนิคสร้าง Node เพื่อใช้งานกับ Node-RED ให้适合自己 的需求 [Online]. ประเทศไทย. 2556. แหล่งที่เข้าถึง : <https://nickuntitled.com/2021/03/27/customize-node-red>

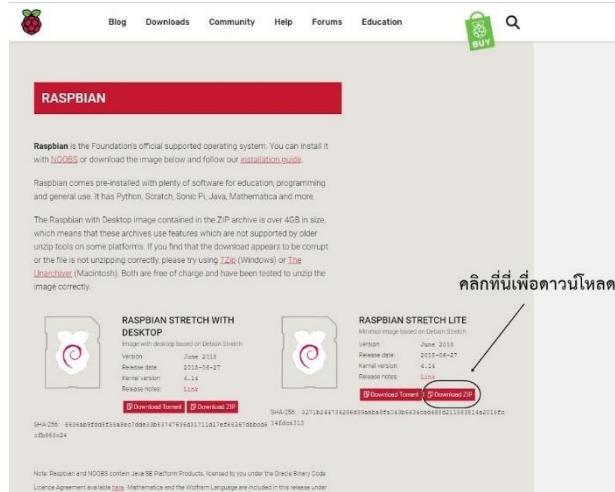
ภาคผนวก

ภาคผนวก ก.

คู่มือการลงระบบปฏิบัติการใน Raspberry Pi

- ขั้นตอนที่ 1** ติดตั้ง RASPBIAN STRETCH LITE เวอร์ชัน 4.14 หรือใหม่กว่า เพื่อนำมาใช้เป็นระบบปฏิบัติการของ Raspberry Pi โดย RASPBIAN STRETCH LITE ซึ่งสามารถโหลดได้จากเว็บไซต์ <https://www.raspberrypi.org/downloads/Raspbian/> ดังแสดงในรูปที่ ก.1

RASPBIAN STRETCH LITE

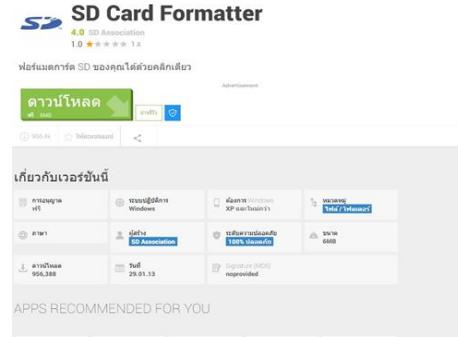


รูปที่ ก.1 หน้าเว็บไซต์ดาวน์โหลดระบบปฏิบัติการ RASPBIAN STRETCH LITE

ไฟล์ดาวน์โหลดจะมีให้เลือกเป็น Download Torrent และแบบ Download ZIP ในกรณีที่ทำการดาวน์โหลดเป็น ZIP File เมื่อเสร็จสิ้นการดาวน์โหลด ให้ทำการแตกไฟล์ออก จะได้เป็นไฟล์ Disc Image File หลังจากนั้นเราต้องทำการนำไฟล์ลง Flash Drive

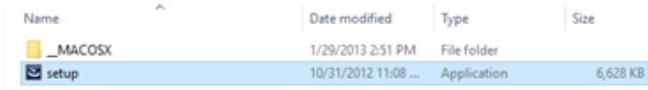
- ขั้นตอนที่ 2** ติดตั้ง SD Card Formatter เวอร์ชัน 16.0.0.400 หรือใหม่กว่า เพื่อนำมาล้างข้อมูลใน Micro SD Card ที่จะนำมาลงระบบปฏิบัติการ ซึ่งสามารถโหลดได้จากเว็บไซต์ <https://sd-card-formatter.th.uptodown.com/windows/download> ดังแสดงในรูปที่ ก.2

SD Card Formatter



รูปที่ ก.2 หน้าเว็บไซต์ดาวน์โหลด SD Card Formatter

- ขั้นตอนที่ 2.1 ไฟล์ที่ดาวน์โหลดจะเป็น ZIP File เมื่อดาวน์โหลดเสร็จสิ้น ให้ทำการแตกไฟล์แล้วทำการลงโปรแกรม SD Formatter ดังแสดงในรูปที่ ก.3



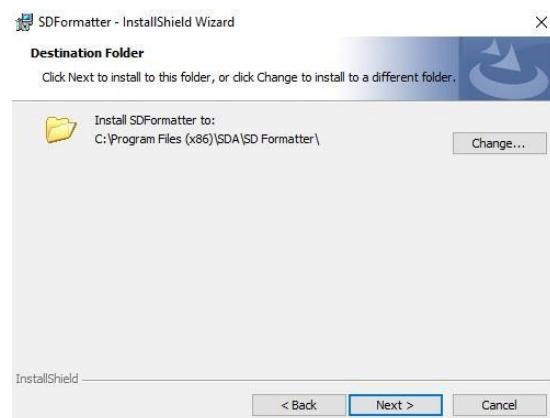
รูปที่ ก.3 หน้าต่างหลังทำการแตกไฟล์

- ขั้นตอนที่ 2.2 ลงโปรแกรม SD Formatter ทำการคลิกที่ไฟล์ Setup และกด Next ดังแสดงในรูปที่ ก.4



รูปที่ ก.4 หน้าต่างแสดงการติดตั้งโปรแกรม SD Formatter

- ขั้นตอนที่ 2.3 เลือกที่อยู่สำหรับจัดเก็บโปรแกรม SD Card Formatter จากนั้นกด Next ดังแสดงในรูปที่ ก.5



รูปที่ ก.5 หน้าต่างแสดงการติดตั้งโปรแกรม SD Formatter

ขั้นตอนที่ 2.4 หลังจากนั้นรอให้โปรแกรม SD Card Formatter ทำการติดตั้ง ดังแสดงในรูปที่ ก.6

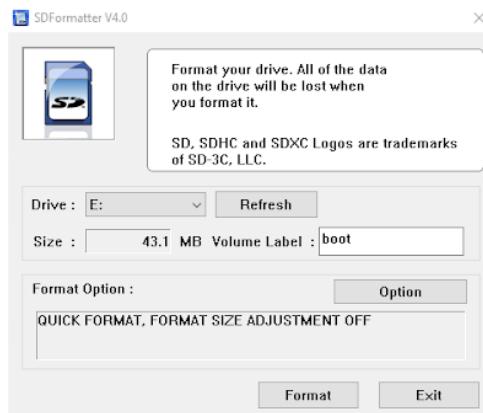


รูปที่ ก.6 หน้าต่างแสดงการติดตั้งโปรแกรม SD Formatter

ขั้นตอนที่ 2.5 เมื่อทำการติดตั้งเสร็จแล้วจะได้หน้าต่างให้กด Finish หลังจากนี้ก็สามารถใช้งานได้ดังแสดงในรูปที่ ก.7

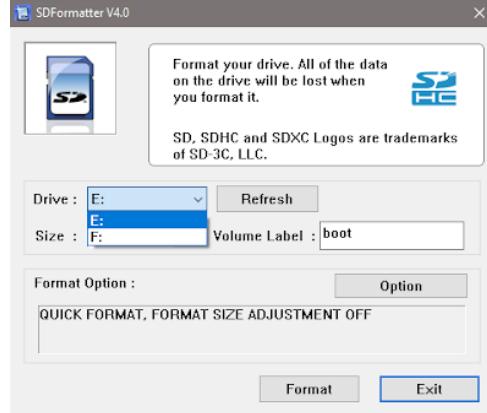


รูปที่ ก.7 หน้าต่างแสดงการติดตั้งโปรแกรม SD Formatter



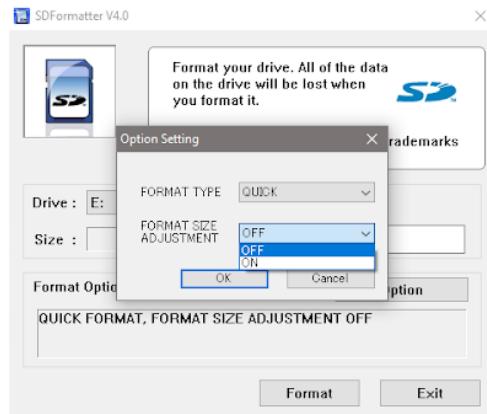
รูปที่ 8 หน้าตากของโปรแกรม SD Formatter

ขั้นตอนที่ 2.6 ใส่ Micro SD Card ที่ต้องการจะ Format ลงไป เลือก Drive ที่จะทำการ Format ดังแสดงในรูปที่ ก.9



รูปที่ ก.9 หน้าตาแสดงการ Format Sd Card

ขั้นตอนที่ 2.7 จากนั้นเลือก Option ที่ต้องการจะ Format ที่สำคัญอย่าลืมเลือก Format Size Adjustment เป็น ON และกด OK จากนั้นกด Format แล้วเลือก OK ต่ออีกสองครั้ง ดังแสดงในรูปที่ ก.10



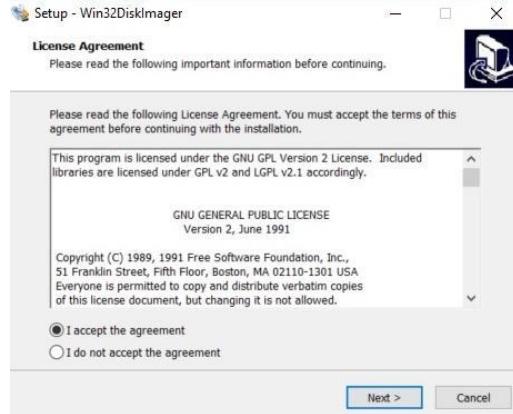
รูปที่ ก.10 หน้าตาแสดงการ Format SD Card

ขั้นตอนที่ 3 ติดตั้งโปรแกรม Win32 Disk Imager เวอร์ชัน 1.0.0 หรือใหม่กว่า เพื่อนำมาเขียนไฟล์ระบบปฏิบัติการที่เป็นไฟล์ Disc Image File ลง Micro SD Card ซึ่งสามารถดาวน์โหลดได้ที่เว็บไซต์ SOURCEFORGE



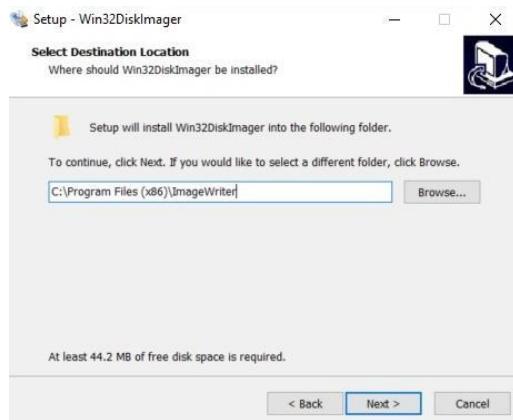
รูปที่ ก.11 หน้าเว็บไซต์ดาวน์โหลดโปรแกรม Win32 Disk Imager

- ขั้นตอนที่ 3.1 เมื่อดาวน์โหลดโปรแกรม Win32 Disk Imager จะได้เป็นไฟล์ .exe จากนั้นทำการรันโปรแกรม โดยเลือก I accept the agreement แล้วกด Next ดังรูปที่ ก.12



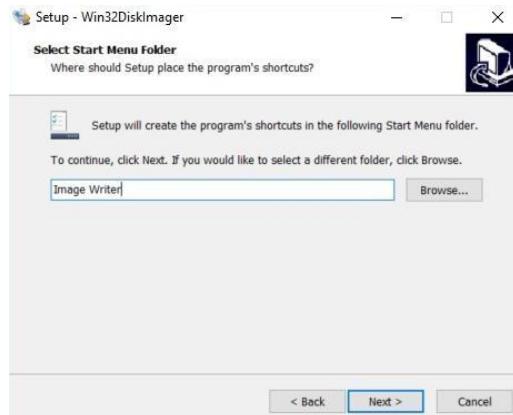
รูปที่ ก.12 หน้าต่างแสดงการติดตั้งโปรแกรม Win32 Disk Imager

- ขั้นตอนที่ 3.2 เลือกพื้นที่ในการจัดเก็บไฟล์ Win32 Disk Imager และกด Next ดังรูปที่ ก.13



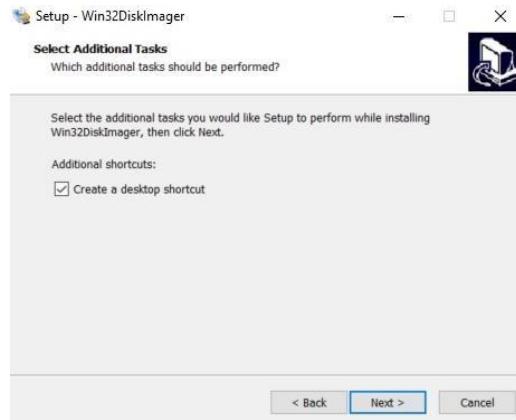
รูปที่ ก.13 หน้าต่างแสดงการเลือกพื้นที่ในการจัดเก็บไฟล์

- ขั้นตอนที่ 3.3 เป็นการตั้ง Shortcuts เสร็จเรียบร้อย ให้ทำการกด Next ดังรูปที่ ก.14



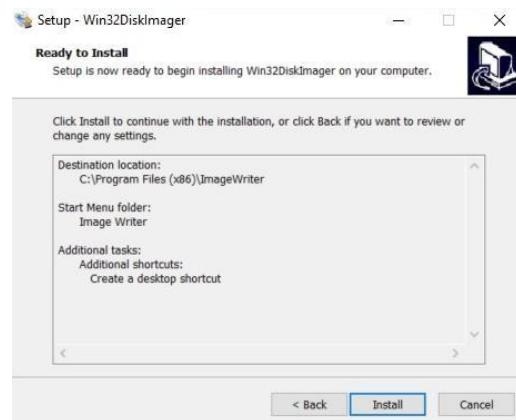
รูปที่ ก.14 หน้าต่างแสดงการติดตั้งโปรแกรม Win32 Disk Imager

- ขั้นตอนที่ 3.4 เลือก Create a Desktop Shortcuts เพื่อให้โปรแกรมอยู่หน้า Desktop เมื่อตั้งเสร็จก็กด Next ดังรูปที่ ก.15



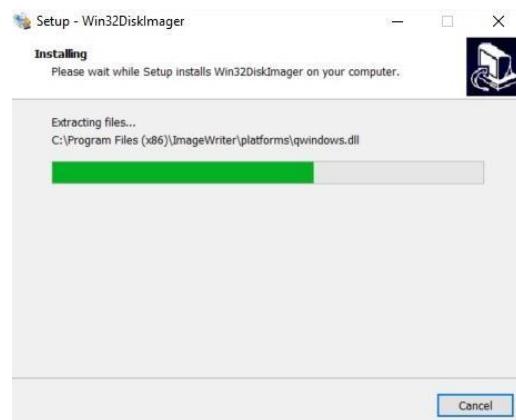
รูปที่ ก.15 หน้าต่างแสดงการเลือกให้โปรแกรมอยู่หน้า Desktop

- ขั้นตอนที่ 3.5 เลือก Install เพื่อเป็นการติดตั้งโปรแกรม ดังรูปที่ ก.16



รูปที่ ก.16 หน้าต่างแสดงการติดตั้งโปรแกรม Win32 Disk Imager

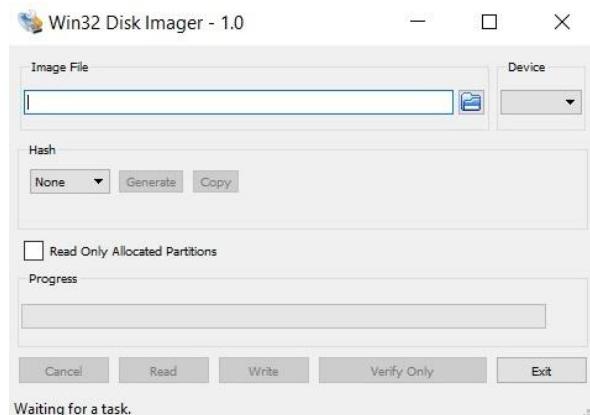
- ขั้นตอนที่ 3.6 โปรแกรมกำลังทำการติดตั้ง เมื่อติดตั้งเสร็จแล้วกด Finnish ดังรูปที่ ก.17 และ ก.18



รูปที่ ก.17 หน้าต่างแสดงการติดตั้งโปรแกรม Win32 Disk Image

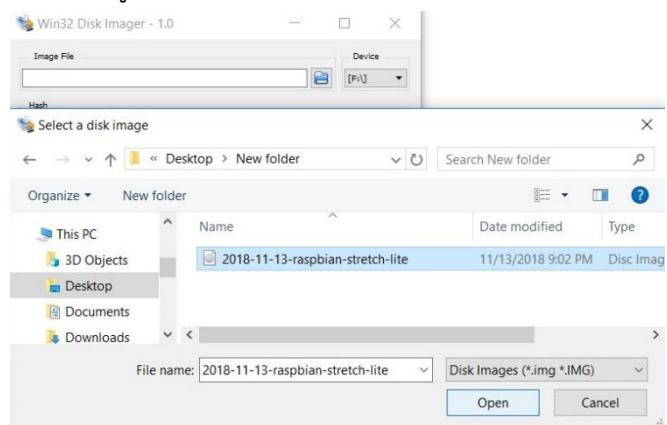


รูปที่ ก.18 โปรแกรมทำการติดตั้งเสร็จเรียบร้อยแล้ว



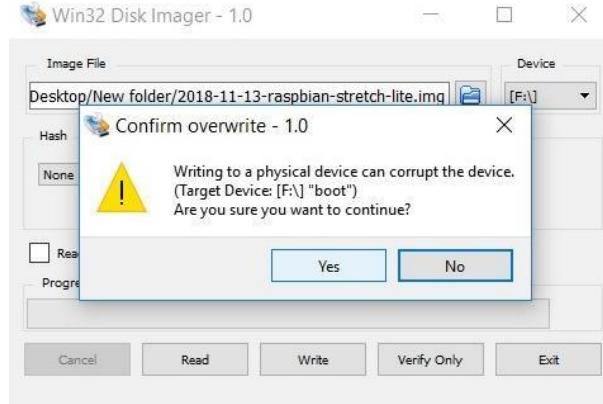
รูปที่ ก.19 หน้าตาของโปรแกรม Win32 Disk Imager

ขั้นตอนที่ 3.7 วิธีการลงระบบปฏิบัติการ Raspbian Stretch Lite ให้เลือกไฟล์ Raspbian Stretch Lite ที่โหลดได้ไว้มาในขั้นตอนที่ 2 ดังรูปที่ ก.20



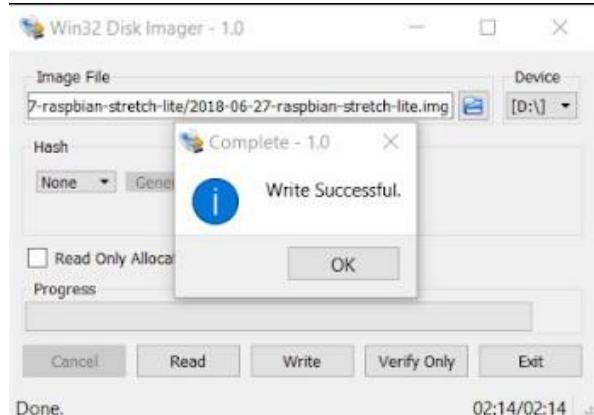
รูปที่ ก.20 หน้าต่างแสดงการเลือกไฟล์ Raspbian Stretch Lite ที่โหลดไว้

- ขั้นตอนที่ 3.8 จากนั้นกด Write เพื่อทำการเขียนไฟล์เข้า SD Card โปรแกรมจะแสดงหน้าต่างขึ้นมาให้รายึนยันว่าจะเขียนไฟล์ลงไปหรือไม่ ให้เลือก Yes เพื่อยืนยันการเขียนไฟล์ ดังรูปที่ ก.21



รูปที่ ก.21 หน้าต่างแสดงการเขียนไฟล์เข้า SD Card

- ขั้นตอนที่ 3.9 เมื่อโปรแกรมทำการลงเสร็จสิ้นแล้ว จะมีหน้าต่างแสดงขึ้นว่า Write Successful ถือเป็นการลงเสร็จสิ้นแล้ว ให้ถอน Micro SD Card ออกจากคอมพิวเตอร์แล้วนำไปใส่ที่ Raspberry Pi ดังรูปที่ ก.22



รูปที่ ก.22 หน้าต่างแสดงว่าโปรแกรมติดตั้งเสร็จเรียบร้อยแล้ว

- ขั้นตอนที่ 3.10 หลังจากที่ทำการลงระบบปฏิบัติการเรียบร้อยแล้ว จากนั้นนำ SD Card มาเสียบที่เครื่อง Raspberry Pi จากนั้นทำการต่ออุปกรณ์เสริม เช่น หน้าจอแสดงผล, เม้าส์ และคีย์บอร์ด จากนั้นทำการเสียบปลั๊กเป็นการเปิดเครื่อง

คู่มือการติดตั้ง Visual Studio Code

ขั้นตอนที่ 1 ตั้ง username และ password ดังรูปที่ ก.23



```

Starting Login Service...
Starting dhcpcd on all interfaces...
Starting Avahi mDNS/DNS-SD Stack...
[OK] Started triggerhappy global hotkey daemon.
[OK] Started Disable WiFi if country not set.
[OK] Started Save/Restore Sound Card State.
[OK] Started System Logging Service.
[OK] Started Login Service.
[OK] Started Avahi mDNS/DNS-SD Stack.
    Starting Load/Save RF Kill Switch Status...
[OK] Started Load/Save RF Kill Switch Status.
[OK] Started LSB: Autogenerate and use a swap file.
[OK] Started LSB: Switch to ondemand cpu governor (unless shift key is pressed).
[OK] Started Configure Bluetooth Modems connected by UART.
    Starting Bluetooth service...
[OK] Started Bluetooth service.
[OK] Reached target Bluetooth.
    Starting Hostname Service...
[OK] Started Hostname Service.
[OK] Started dhcpcd on all interfaces.
[OK] Reached target Network.
    Starting OpenBSD Secure Shell server...
    Starting /etc/rc.local Compatibility...
    Starting Permit User Sessions...
my IP address is 192.168.1.45
[OK] Started /etc/rc.local Compatibility.
[OK] Started Permit User Sessions.
    Starting Hold until boot process finishes up...
    Starting Terminate Plymouth Boot Screen...

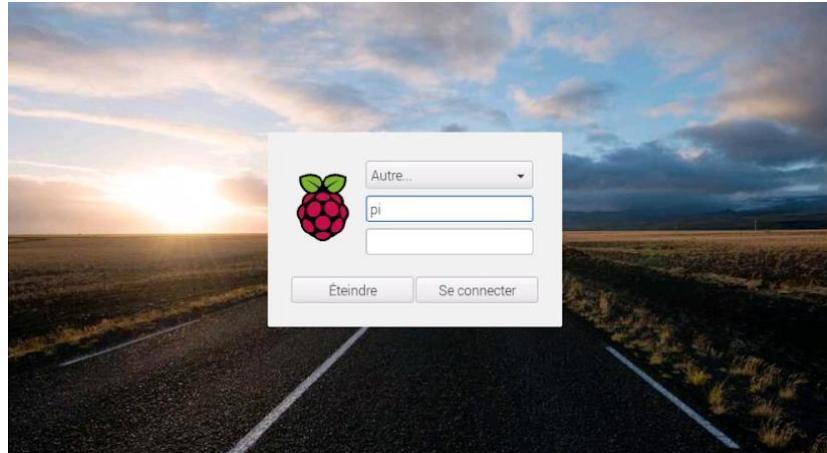
Raspbian GNU/Linux 9 raspberrypi ttym1
raspberrypi login:

```

รูปที่ ก.23 หน้าต่างแสดงการตั้ง username และ password

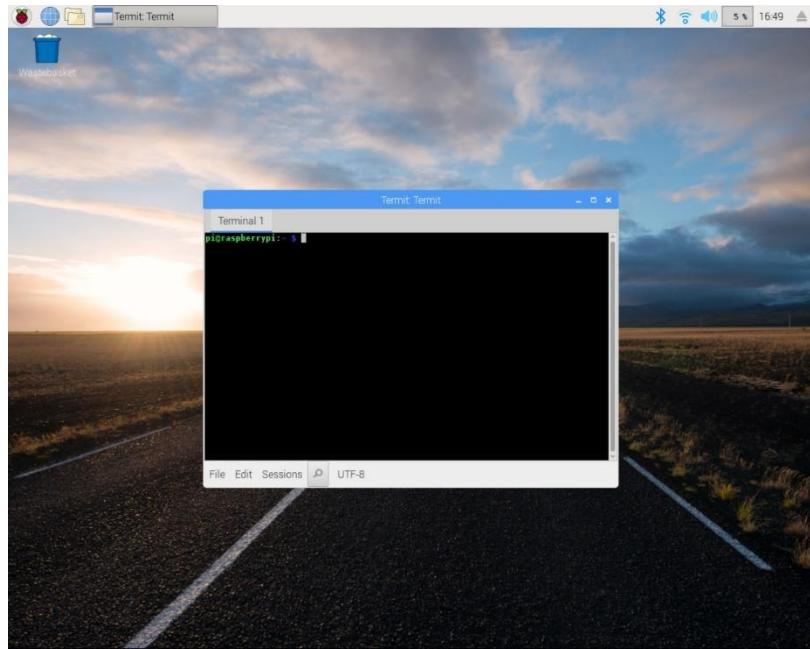
ที่มา : <https://www.behind-the-scenes.co.za/raspberry-pi-first-time-Setup/>

ขั้นตอนที่ 2 เข้าสู่ระบบการติดตั้ง Visual Studio Code โดยใช้ Username และ Password ที่ตั้งไว้ใน
ดังแสดงในรูปที่ ก.24



รูปที่ ก.24 หน้าต่าง login

ขั้นตอน 3 ทำการเปิดหน้าต่าง Terminal ดังแสดงในรูปที่ ก.25



รูปที่ ก.25 หน้าต่าง Terminal

ขั้นตอนที่ 4 ทำการอัพเดตระบบปฏิบัติการโดยใช้คำสั่ง

```
$ sudo apt update
```

ขั้นตอนที่ 5 ทำการอัพเกรดรับระบบปฏิบัติการโดยใช้คำสั่ง

```
$ sudo apt upgrade
```

ขั้นตอนที่ 6 ทำการติดตั้งระบบปฏิบัติการ Visual Studio Code โดยใช้คำสั่ง

```
$ sudo apt install code
```

ขั้นตอนที่ 7 ทำการอัพเดตระบบปฏิบัติการเพื่อตรวจสอบความเข้ากันของรุ่นโดยใช้คำสั่ง

```
$ sudo apt update
```

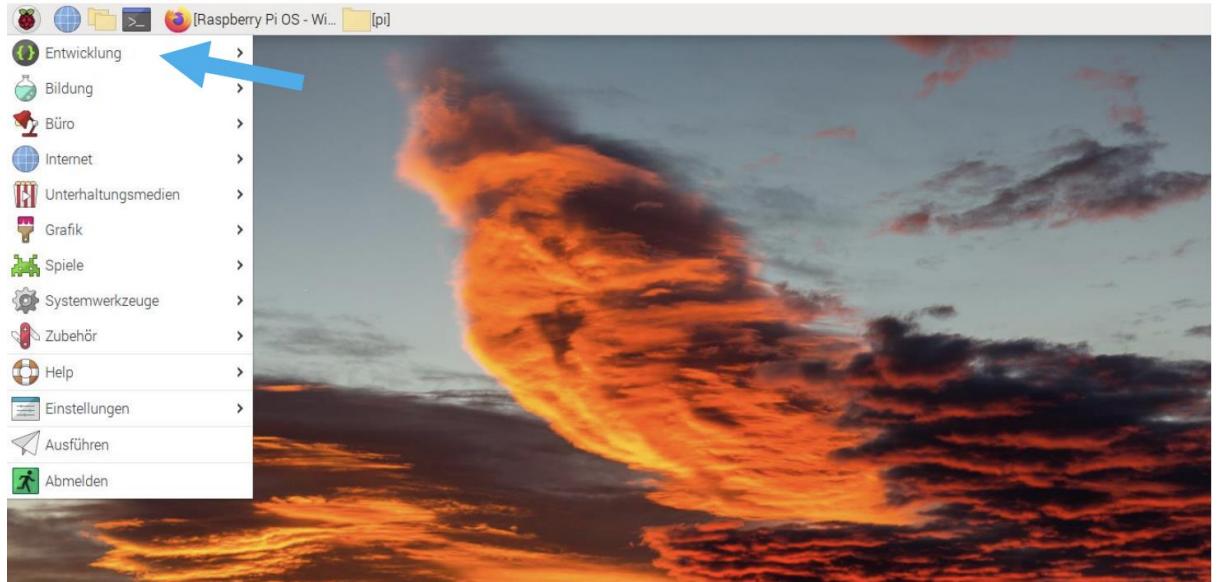
ขั้นตอนที่ 8 ทำการอัพเกรดรับระบบปฏิบัติการ Visual Studio Code โดยใช้คำสั่ง

```
$ sudo apt upgrade code
```

คู่มือการติดตั้ง scikit-learn ใน Raspberry Pi

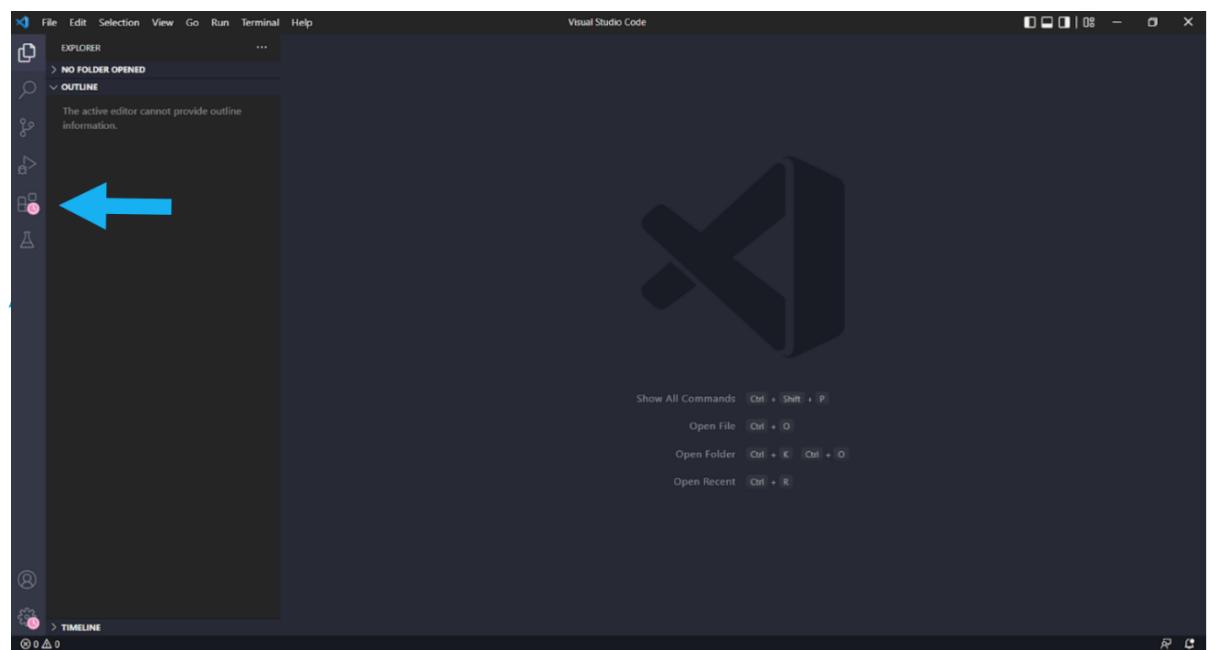
สำหรับการติดตั้ง scikit-learn เราจะต้องทำการติดตั้งภาษา Python ก่อนซึ่งจะเป็นเรื่องที่ง่ายเมื่อทำการติดตั้ง Visual Studio Code ไปแล้ว

ขั้นตอนที่ 1 เปิดโปรแกรม Visual Studio Code บน Raspberry Pi



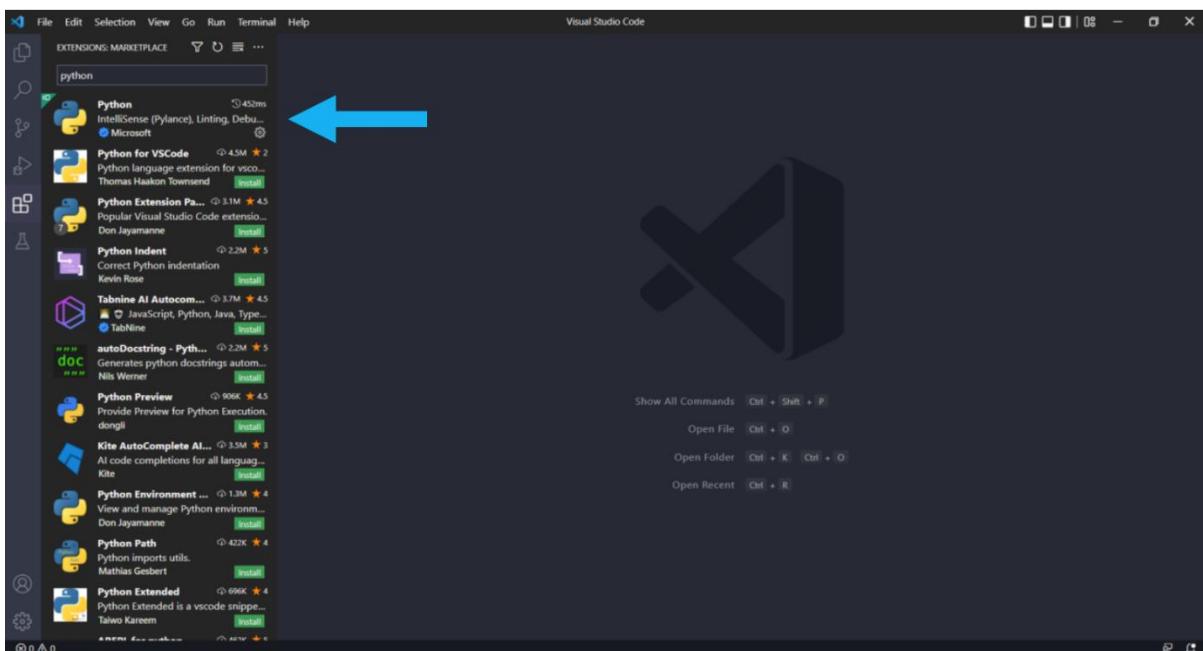
รูปที่ ก.26 ที่อยู่ของโปรแกรม Visual Studio Code

ขั้นตอนที่ 2 เริ่มทำการติดตั้งภาษา Python



รูปที่ ก.27 ติดตั้งตำแหน่ง Extensions หรือ ctrl + shift + x

ขั้นตอนที่ 3 ค้นหาคำว่า Python



รูปที่ ก.28 ค้นหา Python

ขั้นตอนที่ 4 ทำการกด Install เพื่อทำการติดตั้งภาษา Python

ขั้นตอนที่ 5 จากนั้นทำการเปิดหน้าต่าง Terminal (รูปที่ ก.25)

ขั้นตอนที่ 6 ตรวจสอบรุ่นของ Python ว่าเหมาะสมกับการใช้งานหรือไม่

```
$ python3 - -version
```

ขั้นตอนที่ 7 ทำการอัพเดตระบบปฏิบัติการโดยใช้คำสั่ง

```
$ sudo apt-get update
```

ขั้นตอนที่ 8 ทำการติดตั้งเตรียมสำหรับ scikit-learn

```
$ sudo apt-get install gfortran libatlas-base-dev libopenblas-dev liblapack-dev
```

ขั้นตอนที่ 9 ในการติดตั้งด้วยคำสั่ง pip

โดยตรงอาจให้เวลานานซึ่งในระหว่างติดตั้งอาจจะทำให้เกิดปัญหาได้เราจึงทำการติดตั้งโดยดึงข้อมูลจาก piwheels.org แทน

```
$ pip3 install scikit-learn --index-url https://piwheels.org/simple
```

ภาคผนวก ข.

ตัวอย่าง โค้ด และ หลักการทำงาน

```

1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn import tree
4 from sklearn.metrics import accuracy_score
5 from sklearn.metrics import plot_confusion_matrix
6 import matplotlib.pyplot as plt
7 import pickle
8 import itertools
9 import numpy as np
10 from sklearn import svm, datasets
11
12 data = pd.read_csv('ADXL_IMU.csv')
13
14 data[['X-axis', 'Y-axis']]
15 data[['X-axis', 'Y-axis']].shape
16
17 Straight = data['Behavior']=='Straight'
18
19 x = data.drop(['Behavior'], axis=1)
20 y = data[['Behavior']]
21
22 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4)
23
24 model = tree.DecisionTreeClassifier()
25
26 model.fit(x_train, y_train)
27 y_predict = model.predict(x_test)
28
29 def plot_confusion_matrix(cm, classes,
30                           normalize=False,
31                           title='Confusion matrix',
32                           cmap=plt.cm.Blues):
33     """
34     This function prints and plots the confusion matrix.
35     Normalization can be applied by setting `normalize=True`.
36     """
37     plt.imshow(cm, interpolation='nearest', cmap=cmap)
38     plt.title(title)
39     plt.colorbar()
40     tick_marks = np.arange(len(classes))
41     plt.xticks(tick_marks, classes, rotation=45)
42     plt.yticks(tick_marks, classes)
43
44     if normalize:
45         cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
46         print("Normalized confusion matrix")
47     else:
48         print('Confusion matrix, without normalization')
49
50     print(cm)
51
52     thresh = cm.max() / 2.
53     for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
54         plt.text(j, i, cm[i, j],
55                  horizontalalignment="center",
56                  color="white" if cm[i, j] > thresh else "black")

```

รูปที่ ข.1 โค้ดการทำงานของโมเดลอัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm)

```

57     plt.tight_layout()
58     plt.ylabel('True label')
59     plt.xlabel('Predicted label')
60
61     # Compute confusion matrix
62     cnf_matrix = confusion_matrix(y_test, y_predict)
63     np.set_printoptions(precision=2)
64
65     # Plot non-normalized confusion matrix
66     plt.figure()
67     plot_confusion_matrix(cnf_matrix, classes=Straight,
68                            title='Confusion matrix, without normalization')
69
70     # Plot normalized confusion matrix
71     plt.figure()
72     plot_confusion_matrix(cnf_matrix, classes=Straight, normalize=True,
73                           title='Normalized confusion matrix')
74
75     plt.show()
76
77
78     accuracy_score(y_test, y_predict)
79     print('Prediction accuracy : {}'.format(accuracy_score(y_test, y_predict)))
80
81
82     filename = 'model_IMU_tree.sav'
83     pickle.dump(model, open(filename, 'wb'))
84
85     print('{} has been saved'.format(filename))

```

รูปที่ ข.1 โค้ดการทำงานของโมเดลอัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm)(ต่อ)

จากรูปที่ ข.1 คือคำสั่งทั้งหมดที่ใช้ในการเทรน-เทส โมเดล ของอัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm) โดยเริ่มจากระบบจะรับข้อมูลตัวอย่างจากไฟล์ .csv โดยใช้คำสั่งในการอ่านข้อมูลในบรรทัดที่ 12 จากนั้นจึงทำการแบ่งข้อมูลออกเป็น 2 ส่วนเพื่อไว้ใช้สำหรับการเทรน-เทส โดยระบบจะทำการแบ่งข้อมูลทั้ง 500 ค่า เพื่อไว้ใช้เทรน 300 ค่าหรือ 60% และไว้สำหรับเทส อีก 200 ค่าหรือ 40% ด้วยการสุ่มของระบบจากนั้นในบรรทัดที่ 24 จะเป็นการเรียกใช้อัลกอริทึมของอัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm) ในบรรทัดที่ 29 - 76 คือการแสดงคำตอบที่ใช้ในการเทรน-เทส ออกแบบในรูปแบบของเมทริกซ์ และจึงทำการสรุปอัตราความถูกต้องของระบบด้วยคำสั่งบรรทัดที่ 79 – 80 เสร็จแล้วจึงจะทำการเซฟ ข้อมูลโดยการหาไฟล์ชื่อว่า model_IMU_tree.sav ถ้าเจอก็จะทำการเซฟโมเดลลงไปไฟล์นั้น แต่หากมาเจอระบบก็จะทำการสร้างไฟล์ชื่อนั้นอีกครั้งก่อนจะทำการเซฟ โมเดลลงไป

```

1  from sklearn.model_selection import train_test_split
2  from sklearn.neighbors import KNeighborsClassifier
3  from sklearn.metrics import accuracy_score
4  from sklearn.metrics import classification_report,confusion_matrix
5  import numpy as np
6  import pandas as pd
7  import matplotlib.pyplot as plt
8  import seaborn as sns
9  from pandas.plotting import parallel_coordinates
10 from sklearn import metrics
11 from sklearn.naive_bayes import GaussianNB
12 from sklearn.discriminant_analysis import LinearDiscriminantAnalysis, QuadraticDiscriminantAnalysis
13 from sklearn.neighbors import KNeighborsClassifier
14 from sklearn.svm import SVC
15 from sklearn.linear_model import LogisticRegression
16 import pickle
17
18
19 #Read Data
20 df=pd.read_csv('ADXL_IMU.csv')
21 # data
22 x=df.drop("Behavior",axis=1).values
23 # outcome data
24 y=df['Behavior'].values
25
26 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4)
27
28 knn=KNeighborsClassifier(n_neighbors=8)
29 #train
30 knn.fit(x_train,y_train)
31
32 #prediction
33 y_pred=knn.predict(x_test)
34
35 print(pd.crosstab(y_test,y_pred,rownames=['Actually'],colnames=['Prediction'],margins=True))
36
37 accuracy_score(y_test, y_pred)
38 print('Prediction accuracy : {}'.format(accuracy_score(y_test, y_pred)))
39
40 filename = 'model_IMU_KNN.sav'
41 pickle.dump(knn, open(filename, 'wb'))
42
43 print('{} has been saved'.format(filename))
44

```

รูปที่ ข.2 โค้ดการทำงานของโมเดลอัลกอริทึมเพื่อนบ้านใกล้สุด (K-Nearest Neighbor Algorithm หรือ KNN)

จากรูปที่ ข.2 คือคำสั่งทั้งหมดที่ใช้ในการเทรน-เทส โมเดล ของอัลกอริทึมเพื่อนบ้านใกล้สุด (K-Nearest Neighbor Algorithm หรือ KNN) โดยเริ่มจากระบบจะรับข้อมูลตัวอย่างจากไฟล์ .csv โดยใช้คำสั่งในการอ่านข้อมูลตำแหน่งบรรทัดที่ 20 จากนั้นจึงทำการแบ่งข้อมูลออกเป็น 2 ส่วนเพื่อไว้ใช้สำหรับการเทรน-เทส โดยระบบจะทำการแบ่งข้อมูลทั้ง 500 ค่าเพื่อไว้ใช้เทรน 300 ค่าหรือ 60% และไว้สำหรับทดสอบ 200 ค่าหรือ 40% ด้วยการสุ่มของระบบจากนั้นในบรรทัดที่ 28 จะเป็นการเรียกใช้อัลกอริทึมของเพื่อนบ้านใกล้สุด (K-Nearest Neighbor Algoithm หรือ KNN) ในบรรทัดที่ 35 จะเป็นการแสดงผลคำตอบที่ใช้ในการเทรน-เทส ออกมาในรูปแบบของเมทริกซ์ แล้วจึงทำการสรุปอัตราความถูกต้องของระบบด้วยคำสั่งบรรทัดที่ 37 – 38 เสร็จแล้วจึงจะทำการเซฟข้อมูลโดยการหาไฟล์ชื่อว่า model_IMU_KNN.sav ถ้าเจอก็จะทำการเซฟโมเดลลงไปไฟล์นั้น แต่หากไม่เจอระบบก็จะทำการสร้างไฟล์ชื่อนั้นอีกครั้งก่อนทำการเซฟโมเดลลงไป

```

1  from sklearn.model_selection import train_test_split
2  from sklearn.naive_bayes import GaussianNB
3  from sklearn.metrics import accuracy_score
4  from sklearn.metrics import classification_report,confusion_matrix
5  import numpy as np
6  import pandas as pd
7  import matplotlib.pyplot as plt
8  import seaborn as sns
9  from pandas.plotting import parallel_coordinates
10 from sklearn import metrics
11 from sklearn.naive_bayes import GaussianNB
12 from sklearn.discriminant_analysis import LinearDiscriminantAnalysis, QuadraticDiscriminantAnalysis
13 from sklearn.svm import SVC
14 from sklearn.linear_model import LogisticRegression
15 import pickle
16
17 #Read Data
18 df=pd.read_csv('ADXL_IMU.csv')
19 # data
20 x=df.drop("Behavior",axis=1).values
21 # outcome data
22 y=df['Behavior'].values
23
24 # train , test
25 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4)
26
27 #model
28 model=GaussianNB()
29 #train
30 model.fit(x_train,y_train)
31
32 #prediction
33 y_pred=model.predict(x_test)
34
35 print(pd.crosstab(y_test,y_pred,rownames=['Actually'],colnames=['Prediction'],margins=True))
36
37 accuracy_score(y_test, y_pred)
38 print('Prediction accuracy : {}'.format(accuracy_score(y_test, y_pred)))
39
40 filename = 'model_IMU_GaussianNB.sav'
41 pickle.dump(model, open(filename, 'wb'))
42
43 print('{} has been saved'.format(filename))

```

รูปที่ ข.3 โค้ดการทำงานของโมเดลอัลกอริทึมเก้าส์เซียน นาอีฟเบย์ (Gaussian Naive Bayes Algorithm)

จากรูปที่ ข.3 คือคำสั่งทั้งหมดที่ใช้ในการเทรน-เทส โมเดล ของอัลกอริทึมเก้าส์เซียน นาอีฟเบย์ (Gaussian Naive Bayes Algorithm) โดยเริ่มจากระบบจะรับข้อมูลตัวอย่างจากไฟล์ .csv โดยใช้คำสั่งในการอ่านข้อมูลในตำแหน่งบรรทัดที่ 18 จากนั้นจึงทำการแบ่งข้อมูลออกเป็น 2 ส่วนเพื่อไว้ใช้สำหรับการเทรน-เทส โดยระบบจะทำการแบ่งข้อมูลทั้ง 500 ค่าเพื่อไว้ใช้ เทรน 300 ค่าหรือ 60% และไว้สำหรับ เทส อีก 200 ค่าหรือ 40% ด้วยการสุ่มของระบบจากนั้นในบรรทัดที่ 28 จะเป็นการเรียกใช้อัลกอริทึมของความหน้าจะเป็น (Gaussian Naive Bayes Algorithm) ในส่วนของบรรทัดที่ 35 จะเป็นการแสดงผลคำตอบที่ใช้ในการเทรน-เทส ออกมาในรูปแบบของเมทริกซ์ แล้วจึงทำการสรุปอัตราความถูกต้องของระบบด้วยคำสั่งบรรทัดที่ 37-38 เสร็จแล้วจึงจะทำการเซฟข้อมูลโดยการหาไฟล์ชื่อว่า model_IMU_GaussianNB.sav ถ้าเจอก็จะทำการเซฟโมเดลลงไปไฟล์นั้น แต่หากไม่เจอระบบก็จะทำการสร้างไฟล์ชื่อนั้นอีกมาก่อนจะทำการเซฟโมเดลลงไป

- ❑ model_IMU_KNN.sav
- ❑ model_IMU_tree.sav
- ❑ model_IMU_GaussianNB.sav

รูปที่ ข.4 โมเดลที่ทำการเซฟไว้

จากรูปที่ ข.4 จะแสดงถึงโมเดลทั้ง 3 หลังจากที่ได้ทำการเซฟไว้โดยทั้ง 3 นี้จะมีส่วนช่วยในการลดปริมาณ โค้ดที่ใช้งานได้และช่วยให้สามารถนำไปใช้ในคำสั่งอื่น ๆ ได้อีกด้วยเพียงแค่ทำการเรียกเปิดโมเดล ที่เราต้องการขึ้นมาและทำการเซฟโมเดลแบบนี้ยังช่วยลดปัญหาความผิดพลาดของโค้ดที่อาจจะเกิดขึ้นได้อีกด้วย

```

1 import time
2 import board
3 import adafruit_adxl34x
4 import pickle
5 import paho.mqtt.publish as publish
6 from collections import Counter
7 import numpy as np
8
9
10 i2c = board.I2C() # uses board.SCL and board.SDA
11
12 # For ADXL343
13 accelerometer = adafruit_adxl34x.ADXL343(i2c)
14
15 # load the model from disk
16 model_tree = pickle.load(open('model_IMU_tree.sav', 'rb'))
17 model_GaussianNB = pickle.load(open('model_IMU_GaussianNB.sav', 'rb'))
18 model_KNN = pickle.load(open('model_IMU_KNN.sav', 'rb'))
19
20 while True:
21
22     array = accelerometer.acceleration
23
24     d = [[array[0], array[1], array[2]]] # read from IMU sensor
25     x_IMU = d
26
27     y_tree = model_tree.predict(x_IMU)
28     y_GaussianNB = model_GaussianNB.predict(x_IMU)
29     y_KNN = model_KNN.predict(x_IMU)
30
31     CT1 = 0 # 21 = Park
32     CT2 = 1 # 22 = Straight
33     CT3 = 2 # 23 = Up hill
34     CT4 = 3 # 24 = Down hill
35     CT5 = 4 # 25 = Accident
36
37     CG1 = 5 # 21 = Park
38     CG2 = 6 # 22 = Straight
39     CG3 = 7 # 23 = Up hill
40     CG4 = 8 # 24 = Down hill
41     CG5 = 9 # 25 = Accident
42

```

รูปที่ ๑.๕ คำสั่งการทำงานของระบบทำงานและการส่งข้อมูล

```

43     CK1 = 10 # 21 = Park
44     CK2 = 11 # 22 = Straight
45     CK3 = 12 # 23 = Up hill
46     CK4 = 13 # 24 = Down hill
47     CK5 = 14 # 25 = Accident
48
49     if(y_tree == ['Park']):
50         CT1 = CT1+21
51     if(y_tree == ['Straight']):
52         CT2 = CT2+21
53     if(y_tree == ['Up hill']):
54         CT3 = CT3+21
55     if(y_tree == ['Down hill']):
56         CT4 = CT4+21
57     if(y_tree == ['Accident']):
58         CT5 = CT5+21
59
60     if(y_GaussianNB == ['Park']):
61         CG1 = CG1+16
62     if(y_GaussianNB == ['Straight']):
63         CG2 = CG2+16
64     if(y_GaussianNB == ['Up hill']):
65         CG3 = CG3+16
66     if(y_GaussianNB == ['Down hill']):
67         CG4 = CG4+16
68     if(y_GaussianNB == ['Accident']):
69         CG5 = CG5+16
70
71     if(y_KNN == ['Park']):
72         CK1 = CK1+11
73     if(y_KNN == ['Straight']):
74         CK2 = CK2+11
75     if(y_KNN== ['Up hill']):
76         CK3 = CK3+11
77     if(y_KNN== ['Down hill']):
78         CK4 = CK4+11
79     if(y_KNN == ['Accident']):
80         CK5 = CK5+11
81
82     y_Number = Counter([CT1, CT2, CT3, CT4, CT5, CG1, CG2, CG3, CG4, CG5, CK1, CK2, CK3, CK4, CK5])
83     #print(y_Number)
84
85     y_Ans = y_Number.most_common(1)[0][0]
86     print(y_Ans)

```

รูปที่ ๗.๕ คำสั่งการทำงานของระบบที่นายและภารส่งข้อมูล(ต่อ)

```

88     if(y_Ans == 21):
89         print('Park')
90         pulish.single("academics/IMU_AXDL345",'Park', hostname = "test.mosquitto.org")
91
92     if(y_Ans == 22):
93         print('Straight')
94         pulish.single("academics/IMU_AXDL345",'Straight', hostname = "test.mosquitto.org")
95
96     if(y_Ans == 23):
97         print('Up hill')
98         pulish.single("academics/IMU_AXDL345",'Up hill', hostname = "test.mosquitto.org")
99
100    if(y_Ans == 24):
101        print('Down hill')
102        pulish.single("academics/IMU_AXDL345",'Down hill', hostname = "test.mosquitto.org")
103
104    if(y_Ans == 25):
105        print('Accident')
106        pulish.single("academics/IMU_AXDL345",'Accident', hostname = "test.mosquitto.org")
107
108    print("Done")
109
110    time.sleep(1.0)

```

รูปที่ ข.5 คำสั่งการทำงานของระบบนำทางและการส่งข้อมูล(ต่อ)

จากรูปที่ ข.5 คือคำสั่งหลัก ๆ ที่ใช้ในการนำทางพุติกรรมการเรียกใช้โมเดล, การเลือกคำตอบที่ถูกต้องที่สุด และ การส่งข้อมูล โดยจะเริ่มที่บรรทัดที่ 10 – 13 คือการกำหนดให้มีการรับค่าจากเซ็นเซอร์ ADXL345 ในรูปแบบการรับส่งข้อมูลแบบ I2C บรรทัดที่ 16 – 18 คือการเรียกใช้ model ทั้ง 3 ที่เราได้ทำการเชฟไว้ในบรรทัดที่ 22 – 25 คือการกำหนดค่าที่รับจากเซ็นเซอร์ ADXL345 ให้มีค่าเป็นอาเรย์ ในส่วนของบรรทัดที่ 27 – 29 คือการกำหนดตัวแปรเพื่อเรียกใช้โมเดล ในการนำทางพุติกรรมโดยจะรับค่าอาเรย์ ที่มาจากการเซฟ ADXL345 ต่อไปจะเป็นการเลือกคำตอบที่ถูกต้องที่สุด หรือ ก็คือมีความเหมือนกันของคำตอบ 2 ใน 3 โดยคำสั่งที่ใช้ในการควบคุมนี้จะอยู่ตั้งแต่บรรทัดที่ 31 – 86 โดยจะทำการแบ่งเป็น 3 กลุ่มดังนี้ CT = อัลกอริทึมไม้ตัดสินใจ (Decision Tree Algorithm), CK = อัลกอริทึมเพื่อบ้านใกล้สุด (K-Nearest Neighbor Algorithm หรือ KNN), CG = ความหน้าจะเป็น (Gaussian Naive Bayes) จากนั้นจึงเข้าสู่คำสั่ง if ตามที่แสดง ในส่วนของบรรทัดที่ 82 คำสั่ง Counter จะมีหน้าที่ในการนับจำนวนของตัวเลขแล้วแบ่งประเภทของตัวเลขที่ซ้ำกัน และบรรทัดที่ 85 จะมีหน้าที่ในการคัดเลือกตัวเลขที่มีการซ้ำกันมากที่สุดอ กมา โดยตัวเลขที่จะแสดงจะมี 5 ค่าดังนี้ 21 = Park, 22 = Straight, 23 = Up hill, 24 = Down hill, 25 = Accident

ในส่วนสุดท้ายคือการส่งข้อมูลที่นำทางออกมายังบอร์ดด้วยระบบ MQTT โดยจะทำการแปลงค่าตัวเลขที่ซ้ำกันมากที่สุดอ กมาเป็นคำตอบตามที่ได้อธิบายไปก่อนหน้าด้วยคำสั่งในบรรทัดที่ 88 – 108 ด้วยคำสั่ง if เมื่อทำการส่งข้อมูลไปยัง MQTT และระบบจะทำการ Print “Done” เพื่อแสดงว่าส่งข้อมูลแล้วจะทำการหน่วงเวลา 1 วินาทีก่อนจะกลับไปเริ่มการทำงานซ้ำจากบรรทัดที่ 20

ภาคผนวก ค.

การติดตั้ง Node Red บน Microsoft Azure

- ขั้นตอนที่ 1** ติดตั้งระบบปฏิบัติการคอมพิวเตอร์โดยใช้ระบบ Azure services โดยคลิกที่ Virtual machines แล้วเลือกระบบปฏิบัติการตามที่ต้องการ โดยเลือกติดตั้งระบบปฏิบัติการ Linux เพื่อที่จะนำมาใช้ติดตั้ง NodeRED

รูปที่ ค.1 หน้าเว็บ Azure services

- ขั้นตอนที่ 2** ติดตั้ง Node.js ให้ตรงกับระบบปฏิบัติการที่ได้ติดตั้งที่ Virtual machines เปิด cloud shell ให้เลือกแบบ Bash รันคำสั่งเพื่อ connect กับ ระบบปฏิบัติการที่ติดตั้งไว้ ssh ตามด้วยชื่อที่ตั้งไว้ในระบบปฏิบัติการด้วยเลข IP address เช่น (ssh surathee@52.158.132.237) จากนั้นทำการใส่รหัสตามที่ได้ตั้งไว้

```

Command Prompt - ssh surathep@52.158.132.237
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saibeat>ssh surathep@52.158.132.237
surathep@52.158.132.237's password: 

```

รูปที่ ค.2 การเปิดการใช้งาน

- | | |
|--------------|---|
| ขั้นตอนที่ 3 | หลังจากเปิดการทำงานของระบบแล้วจึงทำการติดตั้ง Node.js ด้วยคำสั่ง |
| ขั้นตอนที่ 4 | ทำการอัพเดทระบบด้วยคำสั่ง |
| | <code>\$ sudo apt-get update</code> |
| ขั้นตอนที่ 5 | ทำการติดตั้ง Node.js ด้วยคำสั่ง |
| | <code>\$ sudo apt-get install nodejs</code> |
| ขั้นตอนที่ 6 | ทำการติดตั้ง npm ด้วยคำสั่ง |
| | <code>\$ sudo apt-get install npm</code> |
| ขั้นตอนที่ 7 | ทำการติดตั้ง Node-Red โดยนำเข้าข้อมูลจากเว็บไซต์ |
| | <code>\$ curl -sL https://deb.nodesource.com/setup_12.x sudo -E bash -</code> |
| ขั้นตอนที่ 8 | ทำการติดตั้ง NodeRED ด้วยคำสั่ง |
| | <code>\$ sudo apt-get install -y node-red build-essential</code> |
| ขั้นตอนที่ 8 | ทำการติดตั้ง และ ปรับ npm ให้เหมาะสมกับรุ่นของ NodeRED ที่ติดตั้ง |
| | <code>\$ sudo npm install -g --unsafe-perm node-red</code> |
| ขั้นตอนที่ 9 | หลังจากที่ติดตั้ง NodeRED เสร็จเรียบร้อย สามารถเข้าการติดตั้งได้โดยการใช้คำสั่งแล้ว |
| | <code>\$ node-red</code> |

```

surathep@Azure:~$ ssh surathep@52.158.132.237
surathep@52.158.132.237's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-1025-azure x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Thu Jun  9 09:15:34 UTC 2022

System load:  0.01      Processes:          119
Usage of /:   11.9% of 28.90GB  Users logged in:    1
Memory usage: 11%
              IPv4 address for eth0: 10.0.0.4
Swap usage:   0%

* Super-optimized for small spaces - read how we shrank the memory
footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

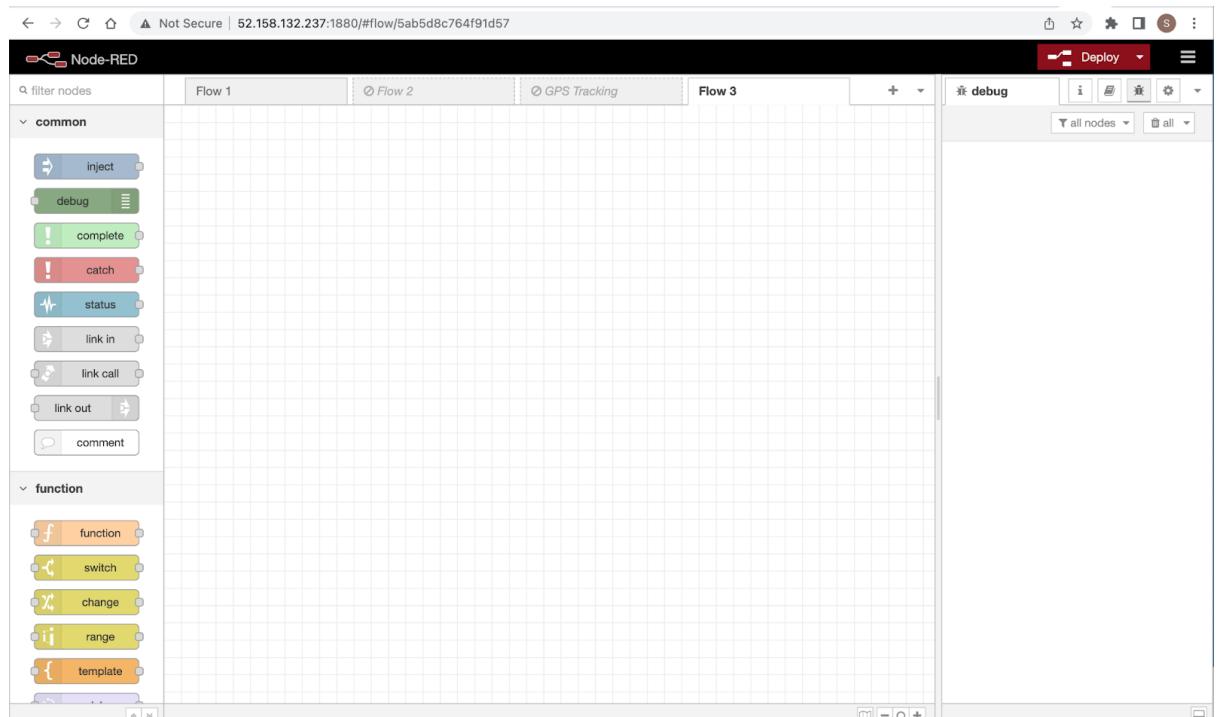
50 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Thu Jun  9 08:54:17 2022 from 20.212.208.108
surathep@surathep:~$ node-red
9 Jun 09:15:39 - [info] Welcome to Node-RED
=====
9 Jun 09:15:39 - [info] Node-RED version: v2.2.2
9 Jun 09:15:39 - [info] Node.js version: v12.22.12
9 Jun 09:15:39 - [info] Linux 5.13.0-1025-azure x64 LE
9 Jun 09:15:39 - [info] Loading palette nodes

```

รูปที่ ค.3 เมื่อทำการเปิดใช้งาน NodeRED แล้ว

ขั้นตอนที่ 10 หลังจากนั้นสามารถใช้งาน NodeRED ได้เมื่อൺปกติโดยใช้เลข IP address ของ Virtual machines ของตนเองตามด้วยเลข Port ของ :1880 เช่น (52.158.132.237:1880)



รูปที่ ค.4 หน้าจอ NodeRED

เทคนิคหรือเทคโนโลยีที่ใช้

1. raspberrypi Model B+ (raspberrypi Model B+)



รูปที่ 2 บอร์ด raspberrypi Model B+

ที่มา : [https://www.thaieeasyelec.com](https://www.thaieasyelec.com)

- Raspberry Pi คือ บอร์ดคอมพิวเตอร์ขนาดเล็ก ที่ถูกพัฒนาขึ้นโดย Raspberry Pi Foundation มีคุณสมบัติเด่น คือ ติดต่อและควบคุมอุปกรณ์อิเล็กทรอนิกส์ได้ โดยเป็นคอมพิวเตอร์ที่มีความสามารถในการใช้งานทั่วไปได้ เช่น ใช้เพื่อทำงานเอกสาร, ดูหนัง พิ้งเพลง, ใช้เพื่อการคำนวณต่างๆ เป็นคอมพิวเตอร์ที่มีขนาดเพียง 85 mm x 49 mm มีราคาถูก เพราะผู้พัฒนามีเจตนาสร้างขึ้นมาเพื่อให้เป็นสื่อการเรียนการสอนทางด้านคอมพิวเตอร์ และเพื่อให้กลุ่มประเทศที่กำลังพัฒนาสามารถมีคอมพิวเตอร์ใช้ได้ทั่วถึงขึ้น
- Model B+ ใช้ชิป BCM2835 ของ Broadcom ในตระกูล Application Processor ของ ARM บนสถาปัตยกรรม ARM11 มีความเร็วในการทำงานที่ 700MHz
- ขนาดความกว้างและความยาว คือ 85 mm x 49 mm ใช้แหล่งจ่ายไฟผ่านคอนเนคเตอร์ microUSB สามารถรันระบบปฏิบัติการ Raspbian หรือระบบปฏิบัติการอื่นๆ
- ขาสัญญาณบนพอร์ต GPIO (General Purpose Input-Output) 40 ขา การรับ/ส่งสัญญาณ input/output ผ่าน GPIO จำเป็นต้องเขียนโปรแกรมสั่งงาน โดยภาษาที่นิยมใช้กันทั่วไป ได้แก่ ภาษา Python แต่นอกจากภาษา Python แล้ว ก็ยังมีภาษาอื่นๆ ให้เลือกใช้กันอีก เช่น C/C++, Shell Script และภาษาอื่นๆ
- ช่องเสียบ USB 4 ช่อง โดยใช้ชิป LAN9514 เป็นมาตรฐาน USB 2.0
- มีช่อง Standard Full-Size SD Card

หมายเหตุ สาเหตุหลักที่ใช้บอร์ด raspberrypi Model B+ เนื่องจากมีความเร็วกว่า Model A+ มีขนาดเล็กกว่าคอมพิวเตอร์ตั้งโต๊ะ จึงสามารถนำไปติดตั้งได้ทุกที่ ดังแสดงในรูปที่ 2

เครื่องมือที่ใช้ในการพัฒนา

การพัฒนาโปรแกรมบน Raspberry Pi ใช้เครื่องมือดังต่อไปนี้

1. Python เวอร์ชัน 3.9.13

ภาษาไพธอนเป็นภาษาที่มีรูปแบบเข้าใจง่าย สามารถใช้งานได้หลายแพลตฟอร์ม มีชุดคำสั่งเสริมหรือไลบรารี (library) ค่อนข้างเยอะ โดยเฉพาะอย่างยิ่งไลบรารีที่เกี่ยวกับการประมวลผลการเรียนรู้ด้วยต้นเอง ทำให้การเขียนโปรแกรมสำหรับการทำนายพฤติกรรมต่างๆ ทำได้ง่ายกว่าภาษาอื่นพร้อมทั้งสามารถนำมาพัฒนาโปรแกรมเพิ่มเติมเพื่อต่อยอดระบบได้ในอนาคต นอกจากนี้ภาษาไพธอนยังมีการประมวลผลแบบอินเทอร์เพตเตอร์ (interpreter) สามารถเห็นผลลัพธ์ได้ในทันที แต่ข้อจำกัดคือเมื่อมีข้อผิดพลาดในโปรแกรมเกิดขึ้น โปรแกรมจะหยุดการทำงานทันที แต่เมื่อเทียบกับภาษาอื่นๆ แล้วภาษาไพธอนสามารถเรียนรู้ได้เร็วกว่า

2. Scikit-learn เวอร์ชัน 1.0.2

Scikit-learn เป็นไลบรารีของการเขียนโปรแกรมที่มีจุดเด่นด้านการเรียนรู้ จัดหมวดหมู่ และทำนาย ด้วยการที่ Scikit-learn เป็นแพ๊กเกจที่มีการรวมรวมไลบรารีด้าน Machine Learning หรือ AI เอาไว้เป็นจำนวนมาก ทำให้ง่ายต่อการเลือกใช้งานโดยเฉพาะ ระบบการเรียนรู้ด้วยต้นเองซึ่งในปัจจุบันได้มีผู้พัฒนานำมาต่อยอดอีกเป็นจำนวนมาก จึงทำให้ในปัจจุบัน Scikit-learn มีฟังก์ชันมากมายที่ง่ายต่อการนำมาใช้งาน อีกทั้ง Scikit-learn ยังสามารถใช้งาน และ อัปเดตได้ฟรี

ที่กล่าวมานี้เหตุผลที่ใช้ภาษา Python, Scikit-learn เพราะว่าสามารถเรียนรู้ภาษาได้ง่ายและ Scikit-learn ยังมีไลบรารี (library) รองรับการเรียนรู้ด้วยต้นเอง การอ่าน และการประมวลผลอีกจำนวนมากที่มีการใช้งานกันอย่างแพร่หลาย ดังแสดงในรูปที่ ข.1 - ข.3 ซึ่งเป็นตัวอย่างของระบบการเรียนรู้ด้วยต้นเองของ Scikit-learn

ปัญหาและอุปสรรค

- raspberrypi 모델비플러스 (Raspberry Pi Model B+)

raspberrypi ในช่วงแรกที่ใช้ไม่สามารถใช้งานได้ เนื่องจากบอร์ดไม่อ่าน sd card หลังจากการรันนั้น จึงทำการเปลี่ยน sd card และลงโปรแกรมใหม่ทำให้ใช้งานได้ตามปกติ

- Python

Python ในตอนแรกเราลงใช้งานไม่ได้ เวอร์ชันที่ลงไม่รองรับการทำงาน หลังจากนั้นจึงได้อัปเดตเวอร์ชันใหม่ เพื่อให้รองรับการทำงาน

- Scikit-learn

Scikit-learn เนื่องจากในตอนแรกเราลงเวอร์ชันที่ไม่รองรับการทำงานจึงต้องทำการลงเวอร์ชันใหม่เพื่อให้รองรับการทำงาน

แนวทางในการพัฒนาและประยุกต์ใช้ร่วมกับงานอื่นๆ ในขั้นตอนไปนี้

โครงการนี้ใช้rasเบอร์รี่พาย โมเดลบีพลัสที่มี Accelerometer แบบ 3 แกนติดตั้งอยู่โดยระบบที่ใช้งาน จะพยายามทำนายพฤติกรรมตลอดเวลาทำให้สามารถนำมาระบุกต์ใช้งานร่วมกับระบบการทรงตัวในรูปแบบต่างๆ หรือการแจ้งเตือนเมื่อเกิดอุบัติเหตุได้ อีกทั้งหลักการทำงานที่กล่าวมานี้ไม่จำเป็นที่จะต้องใช้ในการทำนายพฤติกรรมในขณะขับขี่รถจักรยานหรือรถจักรยานยนต์เพียงเท่านั้น เพราะที่กล่าวมาเป็นเพียงแค่หลักการทำงานหากทำการเปลี่ยนแปลงค่าที่บันทึกที่ใช้สำหรับการเทรน-เทส เป็นความกว้างยาวของกลีบดอกไม้แทนระบบที่เด็กจะสามารถเปลี่ยนเป็นการทำนายชนิดของดอกไม้ได้ หรือ อาจนำไปต่อยอดจนไปถึงการใช้เป็นระบบไร้คนขับที่เรียกว่าปัญญาประดิษฐ์ (Artificial Intelligence) หรือ AI ได้เช่นกัน

ข้อสรุปและข้อเสนอแนะ

การพัฒนาเครื่องข่าย IoT เชื่อมเข้ากับระบบจักรยาน และจักรยานยนต์โดยหลักการทำงานใช้อัลกอริทึมต้นไม้ตัดสินใจ (Decision Tree Algorithm), อัลกอริทึมเพื่อนบ้านใกล้สุด (K-Nearest Neighbor Algorithm หรือ KNN), อัลกอริทึมเกาส์เซียน นาϊฟเบย์ (Gaussian Naive Bayes Algorithm) ซึ่งจะทำการสอนให้โปรแกรมรู้จักการทำนายพฤติกรรม

ประวัติผู้เขียน



ชื่อผู้เขียน นาย นายไชยพุกษ์ นครไพร
วัน เดือน ปีเกิด วันที่ 28 พฤษภาคม พ.ศ. 2541
สถานที่เกิด กรุงเทพมหานคร
ที่อยู่ 19/225 หมู่ 1
หมู่บ้านฟ้ารีนพาร์ค
ตำบลคุคต อำเภอลำลูกกา
จังหวัดปทุมธานี 12130

ประวัติการศึกษา

- สำเร็จการศึกษา มัธยมศึกษาปีที่ 3 จาก โรงเรียนสายปัญญาธงสิต
- สำเร็จการศึกษา มัธยมศึกษาปีที่ 6 จาก โรงเรียนสายปัญญาธงสิต

ประวัติผู้เขียน



ชื่อผู้เขียน นาย สุรเทพ ทันทาน
วัน เดือน ปีเกิด วันที่ 30 กันยายน พ.ศ. 2542
สถานที่เกิด อุบลราชธานี
ที่อยู่ 100/157 หมู่ 6
ตำบลพันท้ายนรสิงห์
อำเภอเมืองสุพรรณบุรี
จังหวัดสุพรรณบุรี 74000

ประวัติการศึกษา

- สำเร็จการศึกษา มัธยมศึกษาปีที่ 6 จาก
โรงเรียนสวนกุหลาบวิทยาลัย

ประวัติผู้เขียน



ชื่อผู้เขียน นางสาว กนกรัตน์ เมธารากุล
วัน เดือน ปีเกิด วันที่ 27 มีนาคม พ.ศ 2543
สถานที่เกิด จังหวัดบุรี
ที่อยู่ 127/3 หมู่ 5 ตำบลพวา
อำเภอแก่งหางเม瓦
จังหวัดจันทบุรี 22160

ประวัติการศึกษา

- สำเร็จการศึกษา ชั้นมัธยมศึกษาปีที่ 3
จากโรงเรียนล้มแบร์ตพิชญาลัย
- สำเร็จการศึกษา ชั้นมัธยมศึกษาปีที่ 6
จาก โรงเรียนสตรีมารดาพิทักษ์

ประวัติผู้เขียน



ชื่อผู้เขียน นาย วงศกร ก้อมขุนทด
วัน เดือน ปีเกิด วันที่ 10 มกราคม พ.ศ.2543
สถานที่เกิด สาระบุรี
ที่อยู่ 21/1 หมู่ที่ 3 ตำบลบ้านเก่า
อำเภอค่านขุนทด
จังหวัดนครราชสีมา 30210

ประวัติการศึกษา

- สำเร็จการศึกษา มัธยมศึกษาปีที่ 3 จาก โรงเรียนพงษ์ศิริวิทยา
- สำเร็จการศึกษา มัธยมศึกษาปีที่ 6 จาก โรงเรียนมัธยมด่านขุนทด

ประวัติผู้เขียน



ชื่อผู้เขียน นาย อภิญญา พีอกทองใบ
วัน เดือน ปีเกิด วันที่ 13 พฤษภาคม พ.ศ 2542
สถานที่เกิด พิจิตร
ที่อยู่ 108/6 หมู่ 6 ตำบลหัวยแก้ว
อำเภอปีนරัง
จังหวัดพิจิตร 66130

ประวัติการศึกษา

- สำเร็จการศึกษา มัธยมศึกษาปีที่ 3 จาก โรงเรียนบ้านวังพร้าว
- สำเร็จการศึกษา มัธยมศึกษาปีที่ 6 จาก โรงเรียนโพธิ์ทรงงาม

ประวัติผู้เขียน



ชื่อผู้เขียน นาย ชัยณรงค์ วาสนาสมศักดิ์
วัน เดือน ปีเกิด วันที่ วันที่ 18 มีนาคม 2539
สถานที่เกิด ชลบุรี
ที่อยู่ 326/215 หมู่ 8 ตำบลสุรศักดิ์
อำเภอศรี ราช จังหวัดชลบุรี

ประวัติการศึกษา

- สำเร็จการศึกษา มัธยมศึกษาปีที่ 3 จาก โรงเรียนกสินธาราชน์ปีเตอร์
- สำเร็จการศึกษา มัธยมศึกษาปีที่ 6 จาก โรงเรียนกสินธาราชน์ปีเตอร์

มหาวิทยาลัยกรุงเทพ

ข้อตกลงว่าด้วยการอนุญาตให้ใช้สิทธิในปริญญาอินพนธ์

วันที่ 9 เดือน มิถุนายน พ.ศ. 2565

| | | | |
|----------|-------------------|---------------|---------------------------|
| ข้าพเจ้า | 1. นายไชยพฤกษ์ | นครไพร | รหัสประจำตัว 1620902708 , |
| | 2. นายนายสุรเทพ | ทนทาน | รหัสประจำตัว 1610901629 , |
| | 3. นางสาวกนกรัตน์ | เมธารากุล | รหัสประจำตัว 1610901017 , |
| | 4. นายวงศกร | ก่อมขุนทด | รหัสประจำตัว 1610900829 , |
| | 5. นายอภิญญาพ | ເຝືອກທອງໃບ | รหัสประจำตัว 1610901132 , |
| | 6. นายชีษณุพงศ์ | วาสนาสมศักดิ์ | รหัสประจำตัว 1610902320 |

เป็นนักศึกษาของมหาวิทยาลัยกรุงเทพ

ระดับปริญญา ตรี โท เอก

หลักสูตร ภาษาไทย สาขาวิชา คณะ วิศวกรรมศาสตร์ ชั้น
ต่อไปนี้เรียกว่า “ผู้อนุญาตให้ใช้สิทธิ” ฝ่ายหนึ่ง และ

มหาวิทยาลัยกรุงเทพ ตั้งอยู่เลขที่ 9/1 หมู่ 5 ตำบลคลองหนึ่ง อำเภอคลองหลวง จังหวัดปทุมธานี 12120 ชั้น
ต่อไปนี้เรียกว่า “ผู้ได้รับอนุญาตให้ใช้สิทธิ” อีกฝ่ายหนึ่ง

ผู้อนุญาตให้ใช้สิทธิ และ ผู้ได้รับอนุญาตให้ใช้สิทธิ ตกลงทำสัญญากันโดยมีข้อความดังต่อไปนี้

ข้อ 1. ผู้อนุญาตให้ใช้สิทธิขอรับรองว่าเป็นผู้สร้างสรรค์และมีสิทธิแต่เพียงผู้เดียวในงานปริญญาอินพนธ์

หัวข้อ ชั้นถือเป็นส่วนหนึ่งของ
การศึกษาตามหลักสูตร ปริญญาวิศวกรรมศาสตรบัณฑิต...สาขาวิศวกรรม ของ
มหาวิทยาลัยกรุงเทพ (ต่อไปนี้เรียกว่า “ปริญญาอินพนธ์”)

ข้อ 2. ผู้อนุญาตให้ใช้สิทธิตกลงยินยอมให้ผู้ได้รับอนุญาตให้ใช้สิทธิ ใช้สิทธิโดยปราศจากค่าตอบแทนและไม่มีกำหนด
ระยะเวลา ในกรณีนำปริญญาอินพนธ์ ชั้นรวมถึงแต่ไม่จำกัดเพียงการทำซ้ำ ดัดแปลง เผยแพร่ต่อสาธารณะชน ให้เช่า^{ให้เช่า}
ต้นฉบับหรือสำเนางาน ให้ประโยชน์ในปริญญาอินพนธ์ แก่ผู้อื่น อนุญาตให้ผู้อื่นใช้สิทธิโดยจะกำหนดเงื่อนไขอย่างหนึ่ง^{อย่าง}
อย่างใดหรือไม่ก็ได้ ไม่ว่าทั้งหมดหรือเพียงบางส่วน หรือการกระทำอื่นใดในลักษณะทำงานเดียวกัน ทั้งนี้ ไม่ตัดสิทธิ
แก่ผู้อนุญาตให้ใช้สิทธิในการนำปริญญาอินพนธ์ ไปแสวงหาประโยชน์ใด ๆ ตราบทে่าที่ไม่ขัดกับการใช้สิทธิของผู้ได้รับ
อนุญาตให้ใช้สิทธิ

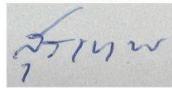
ข้อ 3. หากกรณีมีข้อขัดแย้งในปัญหาลิขสิทธิ์ในปริญญาอินพนธ์ระหว่างผู้อนุญาตให้ใช้สิทธิกับบุคคลภายนอกก็ตี หรือ
ระหว่างผู้ได้รับอนุญาตให้ใช้สิทธิกับบุคคลภายนอกก็ตี หรือมีเหตุขัดข้องอื่น ๆ เกี่ยวกับลิขสิทธิ์ อันเป็นเหตุให้ผู้รับ

อนุญาตให้ใช้สิทธิไม่สามารถนำ้งานนั้นออกทำซ้ำ เผยแพร่ หรือโฆษณาได้ ผู้อนุญาตให้ใช้สิทธิยินยอมรับผิดและชดใช้ค่าเสียหายแก่ผู้ได้รับอนุญาตให้ใช้สิทธิในความเสียหายต่าง ๆ ที่เกิดขึ้นแก่ผู้ได้รับอนุญาตให้ใช้สิทธิทั้งสิ้น

สัญญานี้ทำขึ้นสองฉบับ มีข้อความถูกต้องตรงกัน คู่สัญญาได้อ่านและเข้าใจข้อความในสัญญานี้โดยตลอดแล้ว จึงได้ลงลายมือชื่อไว้ต่อหน้าพยานเป็นสำคัญ และเก็บไว้ฝ่ายละฉบับ

ลงชื่อ บุญธรรม นครไพร ผู้อนุญาตให้ใช้สิทธิ

(นายไชยพฤกษ์ นครไพร)



ลงชื่อ ผู้อนุญาตให้ใช้สิทธิ

(นายสุรเทพ พนพา)

ลงชื่อ Kanokrat Methavarakun ผู้อนุญาตให้ใช้สิทธิ

(นางสาวกนกรัตน์ เมธารากุล)

ลงชื่อ วงศกร ก่อมขุนทด ผู้อนุญาตให้ใช้สิทธิ

(นายวงศกร ก่อมขุนทด)

ลงชื่อ อภิญญา เพ็อกทองใบ ผู้อนุญาตให้ใช้สิทธิ

(นายอภิญญา เพ็อกทองใบ)

ลงชื่อ ธนกร กิตติผลิตา ผู้อนุญาตให้ใช้สิทธิ

(นายธนกร กิตติผลิตา)

ลงชื่อ อิงค์ ผู้ได้รับอนุญาตให้ใช้สิทธิ

(อัจฉริยา ฤทธิศร)

ลงชื่อ พยาน

(ปกรณ์ ยุบลโภศล)

ลงชื่อ พยาน

()