



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบควบคุมยานพาหนะและการสื่อสารข้อมูล VEHICLE CONTROL SYSTEM AND DATA COMMUNICATION

จิรภัทร พิทักษ์สินากร

JIRAPAT PITAKSINAKORN

ภาควิชา วิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา	ระบบควบคุมยานพาหนะและการสื่อสารข้อมูล
ชื่อ-สกุลนักศึกษา	นายจิรภัทร พิทักษ์สินากร
คณะ	วิศวกรรมศาสตร์บัณฑิต ภาควิชา วิศวกรรมสารสนเทศ
ชื่อ-สกุล อาจารย์นิเทศน์	ผศ.ดร.เกล็ดดาว สัตย์เจริญ
ชื่อ-สกุล ผู้นิเทศน์งาน	คุณนิวัฒน์ รุ่งฤทธิเดช
ชื่อสถานประกอบการ	บริษัท ฮ้อปคาร์ จำกัด

บทคัดย่อ

บริษัท ฮ้อปคาร์ จำกัด เป็นผู้ให้บริการคาร์แชร์ริ่งรายแรกในประเทศไทย ให้บริการเป็นแพลตฟอร์มในการเช่าจักรยานพาหนะ ทางบริษัทมีความต้องการทำระบบอุปกรณ์ที่สามารถควบคุมการใช้งาน อ่านข้อมูลสถานะสื่อสาร ส่งข้อมูลสถานะยานพาหนะ และรับคำสั่งได้ ซึ่งพัฒนาขึ้นสำหรับรถจักรยานยนต์ Honda PCX Electric จักรยานยนต์พลังงานไฟฟ้า โดยสามารถควบคุมระบบยานพาหนะ อาทิเช่น ค้นหาตำแหน่งยานพาหนะ ล็อกปลดล็อก ตรวจสอบสถานะพลังงานแบตเตอรี่ไฟฟ้า รวมถึงตรวจสอบความผิดปกติที่เกิดขึ้นกับยานพาหนะ ผ่านทางโมบายแอปพลิเคชันที่ทำงานร่วมกับอุปกรณ์ไมโครคอนโทรลเลอร์ และใช้ CAN – BUS Protocol ในการอ่านค่าข้อมูลสถานะต่าง ๆ จากยานพาหนะ เพื่อนำมาวิเคราะห์และประยุกต์เข้ากับการใช้งาน นอกจากนี้ CAN – BUS Protocol สามารถนำไปพัฒนาต่อยอดเพื่อสร้างระบบสำหรับรถยนต์เช่นกัน

Co-operative Title	Vehicle control system and data communication
Student Intern Name	Jirapat Pitaksinakorn
Faculty	Engineering Department Information Engineering
Advisor Name	Asst.Prof.Dr. Kleddao Satcharoen
Mentor Name	Niwat Rhugritdech
Company	Hauptcar Company Limited

ABSTRACT

Hauptcar Company Limited is the first car sharing service provider in Thailand. Providing a platform for temporary vehicle rental. Recently, there are needs to develop a device that can control, read status data, communicate send vehicle status data and receive command. The project developed for Honda PCX Electric, motorcycle that used electrical power. Users can find vehicle parking spots, lock, unlock, view battery status and check the malfunction of the vehicle. The system can be accessed through mobile application that operated with microcontroller. And use CAN – BUS protocol to read vehicle status data that analyze data for use. In addition, CAN - BUS protocol. It's can be developed for used with the car's system as well.

กิตติกรรมประกาศ

ข้าพเจ้าได้รับผิดชอบและปฏิบัติหน้าที่ในบริษัท ฮอปคาร์ จำกัด ระหว่างวันที่ 5 สิงหาคม ถึงวันที่ 22 พฤศจิกายน พ.ศ.2562 ในโครงการวิชาสหกิจศึกษาที่ทางคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และบริษัทฯ ร่วมมือกันจัดตั้งขึ้นในหัวข้อโครงการ ระบบควบคุมยานพาหนะ และการสื่อสารข้อมูล ซึ่งข้าพเจ้าได้รับความรู้ ความเข้าใจ รวมถึงประสบการณ์ในการทำงานที่เป็นประโยชน์อย่างมาก อีกทั้งการดูแลช่วยเหลือต่าง ๆ ตลอดเวลาการทำงาน การปฏิบัติงานสหกิจศึกษาในครั้งนี้สำเร็จลุล่วงได้โดยได้รับความร่วมมือจากบุคคลต่าง ๆ ดังต่อไปนี้

พนักงานแผนก Managed Service

- คุณนิวัฒน์ รุ่งฤทธิ์เดช
- คุณชยานันท์ สุผา
- คุณประมะ จິงจิราโชติ
- คุณณภัทร ศรีเจริญวงศ์
- คุณวีระยุทธ แก้วเกด
- คุณกร ศิริฤกษ์รัตน

พนักงานแผนกทรัพยากรบุคคล

- คุณสุนิสา สุขชื่น

และข้าพเจ้าขอขอบคุณอาจารย์ที่ปรึกษา ผศ.ดร.เกสิดดาว สัตย์เจริญ ที่คอยให้คำแนะนำ คอยรับฟัง และช่วยเหลือปัญหาต่าง ๆ ในการทำโครงการครั้งนี้ ท้ายที่สุดข้าพเจ้าขอขอบคุณครอบครัวที่คอยให้กำลังใจที่ดีแก่ข้าพเจ้าเสมอมาทำให้ปริญญาานิพนธ์ฉบับนี้ให้สำเร็จลุล่วงไปได้ด้วยดี

จิรภัทร พิทักษ์สินากร

สารบัญ

	หน้า
บทคัดย่อ.....	II
ABSTRACT.....	III
กิตติกรรมประกาศ.....	IV
สารบัญ.....	V
สารบัญรูปภาพ.....	IX
สารบัญตาราง.....	XII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของการปฏิบัติงาน.....	1
1.3 วิธีการดำเนินงาน.....	1
1.4 ขอบเขตของงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทบทวนวรรณกรรม.....	3
2.1 พื้นฐานระบบยานพาหนะไฟฟ้า.....	3
2.1.1 ระบบยานพาหนะพลังงานไฟฟ้า.....	3
2.1.2 ประเภทของยานพาหนะไฟฟ้า.....	4
2.1.3 หน่วยควบคุมอิเล็กทรอนิกส์สำหรับยานพาหนะ (Electric control unit : ECU).....	10
2.1.4 ประเภทหน่วยควบคุมอิเล็กทรอนิกส์สำหรับยานพาหนะ (Electric control unit : ECU).....	11
2.2 โพรโตคอลการสื่อสารข้อมูลภายในยานพาหนะ.....	15
2.2.1 CAN Protocol.....	15
2.2.2 CAN – BUS Protocol.....	16

2.2.3 รูปแบบข้อมูล CAN – BUS.....	17
2.2.4 ความเร็วในการสื่อสารข้อมูล CAN – BUS Protocol.....	19
2.2.5 ประโยชน์ของระบบ CAN – BUS Protocol และการประยุกต์ใช้งาน	20
2.2.6 OBD Protocol	22
2.2.7 OBD - II Protocol	22
2.2.8 ความแตกต่างระหว่าง OBD - II Protocol กับ CAN – BUS Protocol.....	23
2.2.9 Port การเชื่อมต่อ OBD – II	24
2.2.10 รูปแบบข้อมูล OBD – II Protocol.....	25
2.3 Global positioning system (GPS).....	27
2.3.1 Global positioning system (GPS) คืออะไร ?	27
2.3.2 ส่วนประกอบของระบบ GPS.....	28
2.3.3 ประเภทของระบบ GPS.....	28
2.4 Microcontroller.....	30
2.4.1 Microcontroller คืออะไร ?	30
2.4.2 ความแตกต่างระหว่าง Microcontroller กับ Microprocessor.....	31
2.4.3 องค์ประกอบโดยทั่วไปของ Microcontroller	32
2.4.4 ตัวอย่างประเภท และการใช้งาน Microcontroller.....	34
2.5 การออกแบบ และพัฒนา Firmware	37
2.5.1 Firmware คืออะไร ?	37
2.5.2 ความแตกต่างระหว่าง Firmware กับ Software	38
2.5.3 ขั้นตอนการพัฒนา Firmware	39
2.6 Protocol การสื่อสารข้อมูล	39
2.6.1 โพรโตคอล (Protocol) คืออะไร ?.....	39

2.6.2 องค์ประกอบของ Protocol	40
2.6.3 ตัวอย่าง Protocol ที่ใช้กับอุปกรณ์ Microcontroller	41
บทที่ 3 ขั้นตอนการดำเนินงาน	45
3.1 ศึกษาข้อมูลความต้องการที่ได้รับมอบหมาย คิววิเคราะห์หา Function การใช้งานที่เหมาะสม	45
3.1.1 วิเคราะห์ความต้องการที่ได้รับมอบหมาย	45
3.1.2 แก้ปัญหาที่เกิดจาก Firmware รุ่นเก่า	46
3.1.3 ออกแบบฟังก์ชันการใช้งานใหม่ และปรับปรุงฟังก์ชันการใช้งานเก่า	46
3.2 วางแผนขั้นตอนการดำเนินงาน และกำหนดเวลาการทำงานในแต่ละส่วน	47
3.2.1 วางแผนลำดับขั้นตอนการดำเนินงาน	47
3.2.2 กำหนดช่วงเวลาการดำเนินงาน	48
3.3 ศึกษาเกี่ยวกับเครื่องมือที่ใช้ในการพัฒนาโครงการ ทดสอบการใช้	49
3.3.1 Arduino IDE	49
3.3.2 ESP32	58
3.3.3 MCP2515	61
3.3.4 รายละเอียดข้อมูล Honda PCX Electric	64
3.4 ออกแบบ และพัฒนาระบบโครงงาน	66
3.4.1 ผังลำดับงาน	66
3.4.2 ฟังก์ชันการใช้งาน	67
3.5 การใช้งานแอปพลิเคชันฮอป	68
บทที่ 4 ผลการทดลอง	72
บทที่ 5 สรุปผล	79
ปัญหาที่พบระหว่างการดำเนินงาน	80
เอกสารอ้างอิง	81



สารบัญรูปภาพ

รูป 2.1 ยานพาหนะพลังงานไฟฟ้า	3
รูป 2.2 ยานพาหนะพลังงานไฟฟ้าระบบ Mild Hybrid.....	4
รูป 2.3 ยานพาหนะพลังงานไฟฟ้าระบบ Full Hybrid.....	5
รูป 2.4 ยานพาหนะพลังงานไฟฟ้าระบบ Plug - in Hybrid (PHEV)	6
รูป 2.5 ยานพาหนะพลังงานไฟฟ้าระบบ Series Hybrid	7
รูป 2.6 ยานพาหนะพลังงานไฟฟ้าระบบ Electric Vehicle	8
รูป 2.7 ยานพาหนะพลังงานไฟฟ้าระบบ Fuel Cell	9
รูป 2.8 กล่อง ECU).....	10
รูป 2.9 กล่อง Standard ECU.....	11
รูป 2.10 กล่อง Modify ECU	12
รูป 2.11 กล่อง Piggy - Back ECU.....	13
รูป 2.12 กล่อง Standalone ECU	14
รูป 2.13 CAN Protocol.....	15
รูป 2.14 CAN – BUS Protocol Overview.....	16
รูป 2.15 CAN – BUS Protocol.....	17
รูป 2.16 CAN – BUS Data	17
รูป 2.17 CAN – BUS Message Frame.....	17
รูป 2.18 CAN – BUS Node.....	21
รูป 2.19 OBD – II and CAN - BUS.....	23
รูป 2.20 OBD - II Port Position	24
รูป 2.21 OBD – II Port	24
รูป 2.22 OBD – II Message Frame.....	25
รูป 2.23 GPS (1)	27
รูป 2.24 GPS (2).....	29
รูป 2.25 Microcontroller.....	30
รูป 2.26 Microprocessor	32

รูป 2.27 Z80.....	34
รูป 2.28 PIC.....	35
รูป 2.29 Arduino	36
รูป 2.30 Firmware	38
รูป 2.31 Software and Firmware	39
รูป 2.32 TCP / IP Protocol	42
รูป 2.33 MQTT Protocol Detail.....	43
รูป 2.34 MQTT Protocol (2).....	44
รูป 3.1 Download Arduino IDE (1).....	49
รูป 3.2 Download Arduino IDE (2).....	50
รูป 3.2 Download Arduino IDE (3).....	50
รูป 3.4 Download Arduino IDE (4).....	51
รูป 3.5 Download Arduino IDE (5).....	51
รูป 3.6 Arduino IDE.....	52
รูป 3.7 Arduino IDE Interface.....	52
รูป 3.8 Arduino IDE Preference (2).....	53
รูป 3.9 Arduino IDE Preference (2).....	53
รูป 3.10 Arduino IDE set Board Manager (1).....	54
รูป 3.11 Arduino IDE set Board Manager (2).....	54
รูป 3.12 Arduino IDE set Library (1).....	55
รูป 3.13 Arduino IDE set Library (2).....	55
รูป 3.14 Arduino IDE set Port	56
รูป 3.15 Arduino IDE Serial Monitor	56
รูป 3.16 Arduino IDE Serial Monitor set Baud Rate.....	57
รูป 3.17 ESP32.....	58
รูป 3.18 ESP32 Pin	60
รูป 3.19 MCP2515.....	61
รูป 3.20 MCP2515 Diagram	62

รูป 3.21 MCP2515 Pin	63
รูป 3.22 MCP2515 Pin Detail	63
รูป 3.23 Honda PCX Electric.....	64
รูป 3.24 Honda PCX Electric Battery.....	65
รูป 3.25 ผังลำดับงาน.....	66
รูป 3.26 องค์ประกอบของระบบ	67
รูป 3.27 วิธีการใช้งานแอปพลิเคชัน (1).....	68
รูป 3.28 วิธีการใช้งานแอปพลิเคชัน (2).....	69
รูป 3.29 วิธีการใช้งานแอปพลิเคชัน (3).....	70
รูป 3.30 วิธีการใช้งานแอปพลิเคชัน (4).....	71
รูป 3.31 วิธีการใช้งานแอปพลิเคชัน (5).....	71
รูป 4.1 ผลการทดลอง.....	72
รูป 4.2. Electric Smart Station.....	72
รูป 4.3 การจองใช้งาน (1).....	73
รูป 4.4.การจองใช้งาน (2).....	74
รูป 4.5 เปิดระบบที่จักรยานยนต์.....	75
รูป 4.6 ปลดล็อกจะจักรยานยนต์.....	76
รูป 4.7 เปิดระบบการขับขี่จักรยานยนต์.....	77
รูป 4.8 ปิดระบบจบการใช้งาน.....	78

สารบัญตาราง

ตาราง 3.1 ตารางแสดงช่วงเวลาดำเนินงาน 48



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

เนื่องจากบริษัท ฮ้อปคาร์ จำกัด ได้จัดโครงการสหกิจศึกษาาระหว่างบริษัท ฮ้อปคาร์ จำกัด กับ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยในส่วนของแผนก เทคนิคนั้น ได้ได้รับผิดชอบในการดูแลระบบการเข้าใช้งานรถยนต์ และควบคุมการทำงาน ซึ่งเป็นธุรกิจหลักของทางบริษัทในตอนนี้ รวมถึงมีการทำโครงการในด้านอื่น ๆ อาทิเช่น ระบบควบคุมรถจักรยานยนต์ ระบบสถานีชาร์จพลังงานแบตเตอรี่สำหรับยานพาหนะพลังงานไฟฟ้า ซึ่งโครงการที่ได้มีการมอบหมายงานให้นักศึกษาดูแล คือระบบควบคุมยานพาหนะและการสื่อสารข้อมูล สำหรับรถจักรยานยนต์ Honda PCX Electric นักศึกษาจึงต้องหาความรู้หลายด้านเพื่อทำโครงการ ประกอบด้วย อุปกรณ์ Microcontroller เทคโนโลยีการพัฒนาเฟิร์มแวร์ Firmware การออกแบบพัฒนา Protocol ในการส่งข้อมูล รวมถึง CAN – BUS Protocol ซึ่งเป็นระบบการสื่อสารข้อมูลระหว่างส่วนต่าง ๆ ในยานพาหนะ และยังมีความรู้บูรณาการในด้านต่าง ๆ อีกมาก เพื่อให้ นักศึกษามีความรู้และสามารถพัฒนาชิ้นงานออกมาให้ดีที่สุด ตามความที่ได้รับมอบหมาย

1.2 วัตถุประสงค์ของการปฏิบัติงาน

เนื่องจากบริษัท ฮ้อปคาร์ จำกัด เป็นบริษัทที่ให้บริการ Platform การเช่าขี่ยานพาหนะให้กับผู้ใช้งาน จึงมีความจำเป็นต้องศึกษาหาข้อมูล และพัฒนาระบบการทำงานในการควบคุมยานพาหนะให้ดีขึ้น เพื่อที่จะตอบสนองความต้องการของผู้ใช้งาน เป้าหมายคือต้องการขยายขอบเขตธุรกิจจากยานพาหนะรถยนต์ไปสู่รถจักรยานยนต์ โดยมีการเน้นไปที่การพัฒนากระบบยานพาหนะไฟฟ้าที่ตอบโจทย์เทคโนโลยีในอนาคต

1.3 วิธีการดำเนินงาน

1. วิเคราะห์ความต้องการของระบบที่ได้รับมอบหมาย และกำหนดเป้าหมายในการทำโครงการ
2. ออกแบบและวางแผนงานเบื้องต้น คิด Function และรูปแบบการใช้งาน
3. ศึกษาข้อมูลเพิ่มเติม ทั้งในด้าน Hardware ประกอบด้วย พื้นฐานด้านระบบยานยนต์ พื้นฐานด้านไฟฟ้า และอิเล็กทรอนิกส์ รวมถึงระบบ GPS ส่วนในด้าน Software ประกอบด้วย

การออกแบบ และพัฒนาเฟิร์มแวร์สำหรับควบคุมอุปกรณ์ Microcontroller การออกแบบ และพัฒนา Protocol การสื่อสารข้อมูล การออกแบบ และการพัฒนาระบบให้ทำงาน สอดคล้องกับหลายส่วนงาน รวมถึงด้านอื่น ๆ เพื่อนำมาพัฒนาโครงการ

4. ดำเนินการสร้างระบบ
5. ทดสอบระบบ
6. วิเคราะห์ปัญหาที่เกิดขึ้น ค้นหาทางแก้ไขปัญหา
7. เสนอแนะแนวทางการเพิ่มประสิทธิภาพ และพัฒนาต่อยอด
8. สรุปผลการดำเนินงาน

1.4 ขอบเขตของงาน

1. ศึกษาผลิตภัณฑ์และบริการของบริษัท เพื่อปรับตัว และเรียนรู้เกี่ยวกับงานที่ดูแลเป็นกิจวัตร
2. ศึกษาระบบซอฟต์แวร์ Software และ Server ของทางบริษัท เพื่อพัฒนาระบบให้ออกมาทำงานสอดคล้องกัน
3. เรียนรู้เกี่ยวกับข้อมูลในด้านอื่นที่ใช้งานเพื่อพัฒนาระบบออกมาให้สอดคล้องกันอย่างดี
4. ศึกษาและเรียนรู้เกี่ยวกับระบบ CAN – BUS Protocol เพื่อนำข้อมูลออกมาจากระบบของยานพาหนะไปใช้ประโยชน์
5. ออกแบบและพัฒนา Firmware ควบคุมระบบอุปกรณ์ Microcontroller
6. ออกแบบและพัฒนารูปแบบข้อมูลที่จะส่งเข้าไปใน Server
7. ออกแบบและพัฒนา Protocol ในการส่งข้อมูลเข้าไปใน Server

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. นักศึกษาได้รับความรู้ในเรื่องการพัฒนาระบบ ซึ่งเป็นการบูรณาการจากองค์ความรู้หลายด้าน
2. นักศึกษาได้รับประสบการณ์การทำงานจริง ที่เป็นระบบ มีความจริงจังเข้มข้นเหมือนการทำงานจริง และ เป็นประสบการณ์ด้านการทำงานที่ใช้ความรู้หลายด้าน ใช้ต่อยอดในอนาคต
3. บริษัทได้นวัตกรรมใหม่ที่ใช้ในการพัฒนาธุรกิจของทางบริษัทให้ดีขึ้นและหลากหลายยิ่งขึ้น

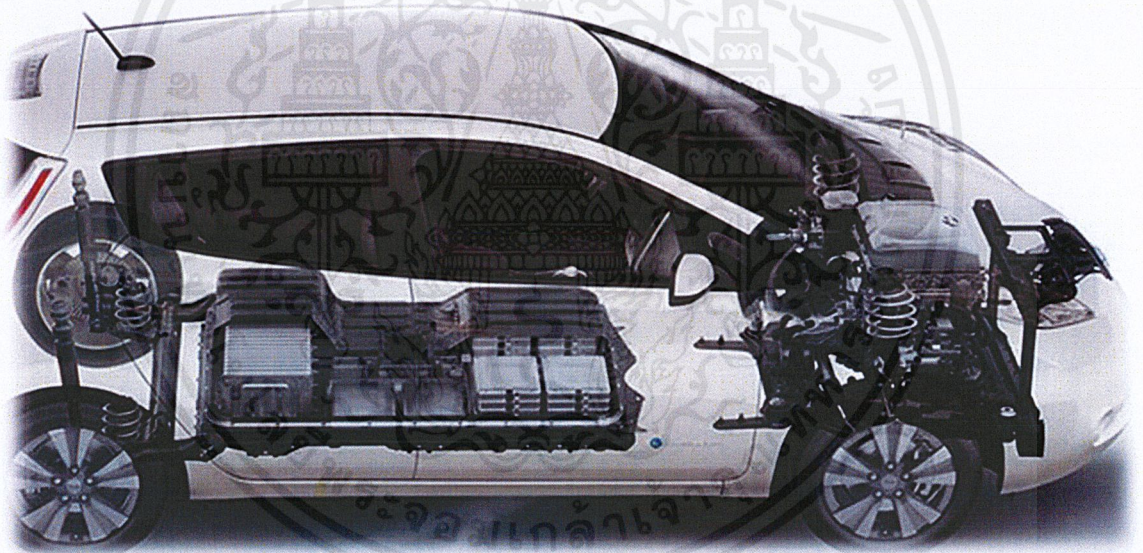
บทที่ 2

ทบทวนวรรณกรรม

2.1 พื้นฐานระบบยานพาหนะไฟฟ้า

2.1.1 ระบบยานพาหนะพลังงานไฟฟ้า

คือยานพาหนะที่ขับเคลื่อนด้วยระบบมอเตอร์ไฟฟ้า ซึ่งจะใช้ไฟฟ้าเป็นพลังงานในการขับเคลื่อนระบบ โดยจะเก็บอยู่ในแบตเตอรี่ หรืออุปกรณ์เก็บพลังงานไฟฟ้าแบบอื่น ๆ และด้วยข้อดีของมอเตอร์ไฟฟ้าที่ให้แรงบิดได้ทันที ทำให้อัตราเร่งที่เรียบ และรวดเร็ว โดยจะมีชุดอุปกรณ์หลักที่ประกอบด้วย แบตเตอรี่ที่มีความจุพลังงานไฟฟ้าสูงซึ่งเก็บไว้ในรถ และมอเตอร์ไฟฟ้าที่ทำหน้าที่ในการขับเคลื่อนโดยใช้พลังงานจากแบตเตอรี่ โดยปกติแล้วส่วนที่สำคัญก็คือปริมาณความจุของแบตเตอรี่ ที่จะเป็นตัวบอกว่ายานพาหนะจะทำงานได้นานแค่ไหน หรือส่งแรงขับเคลื่อนสูงสุดได้แค่ไหน

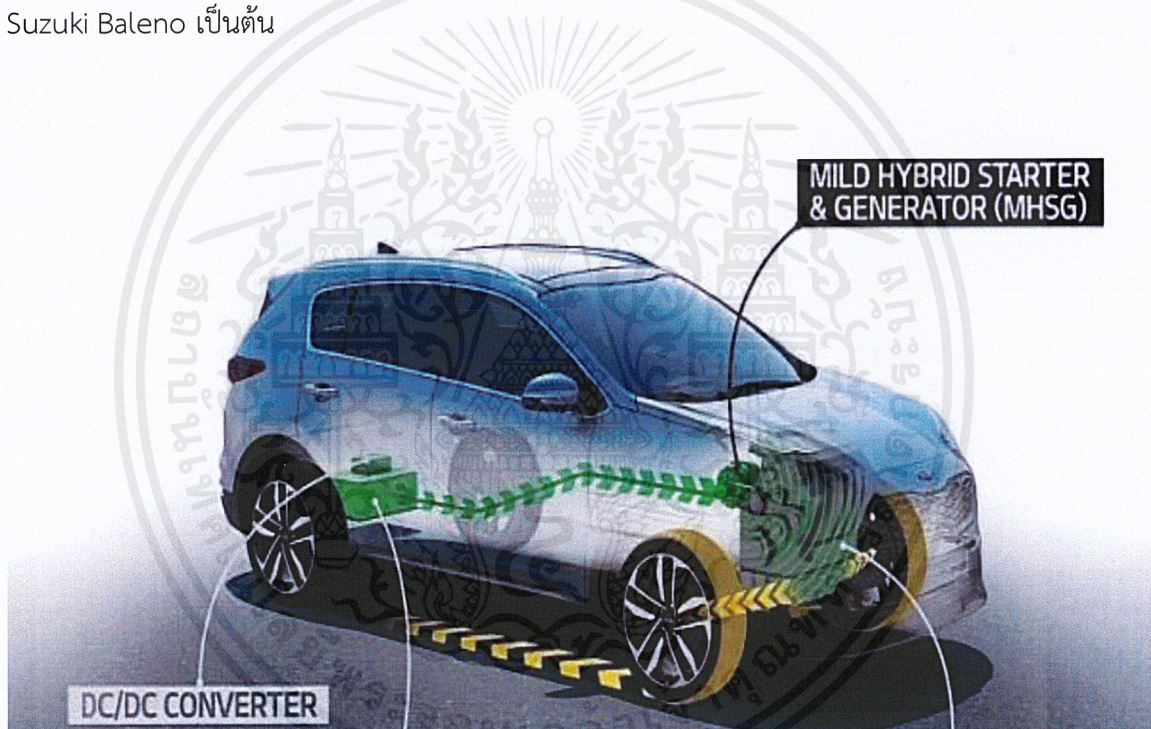


รูป 2.1 ยานพาหนะพลังงานไฟฟ้า

2.1.2 ประเภทของยานพาหนะพลังงานไฟฟ้า

1. Mild Hybrid

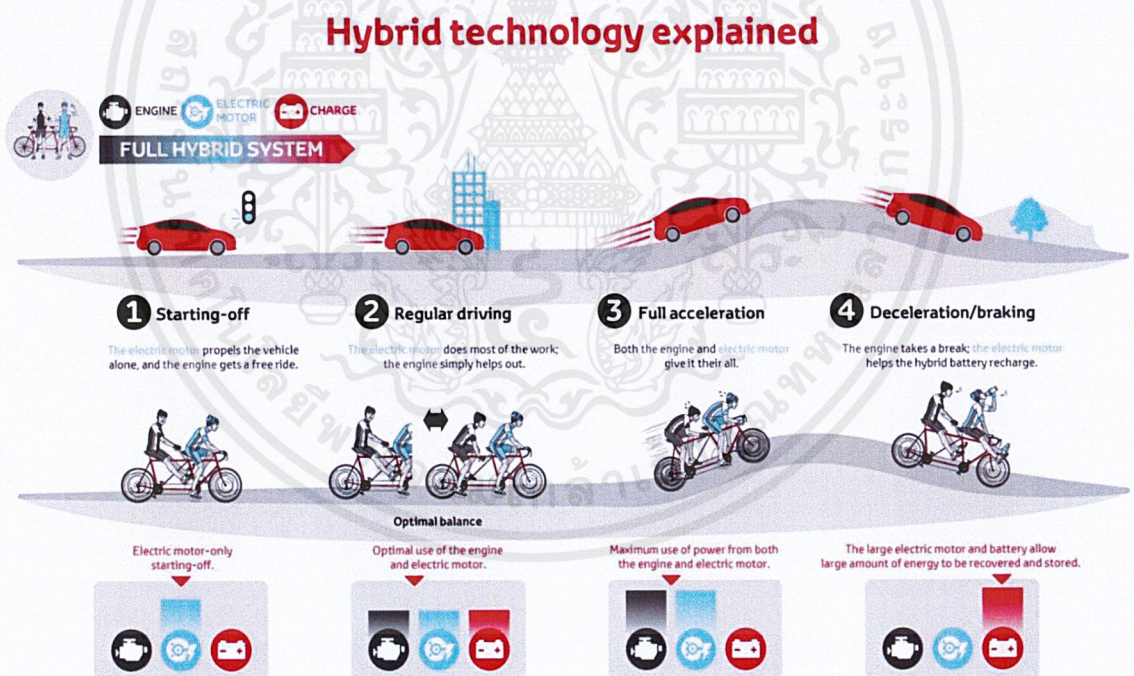
Mild Hybrid จะเป็นยานพาหนะพลังงานไฟฟ้าในระดับเริ่มต้น โดยการทำงานของระบบ Mild Hybrid นั้น จะมีมอเตอร์ไฟฟ้า และแบตเตอรี่มาช่วยการทำงานของเครื่องยนต์ ซึ่งมอเตอร์ และแบตเตอรี่ จะมีขนาดเล็ก ซึ่งทำให้จัดเก็บพลังงานไฟฟ้าได้ไม่มาก มอเตอร์ไฟฟ้าจะทำหน้าที่ในการช่วยออกตัว และการเสริมอัตราเร่งเป็นหลัก ดังนั้นตัวเครื่องยนต์จะต้องไม่ดับ ต้องทำงานอยู่ตลอดเวลา และได้ระบบมอเตอร์ไฟฟ้ามาช่วยลดการทำงานของเครื่องยนต์ในช่วงออกตัว และทำอัตราเร่งได้ระดับหนึ่ง ทำให้ช่วยลดการใช้น้ำมันได้บ้าง รถยนต์ที่ใช้ระบบนี้ก็จะมีย่าง เช่น Audi A8, Audi A6, Hyundai Tucson, Suzuki Baleno เป็นต้น



รูป 2.2 ยานพาหนะพลังงานไฟฟ้าระบบ Mild Hybrid

2. Full Hybrid

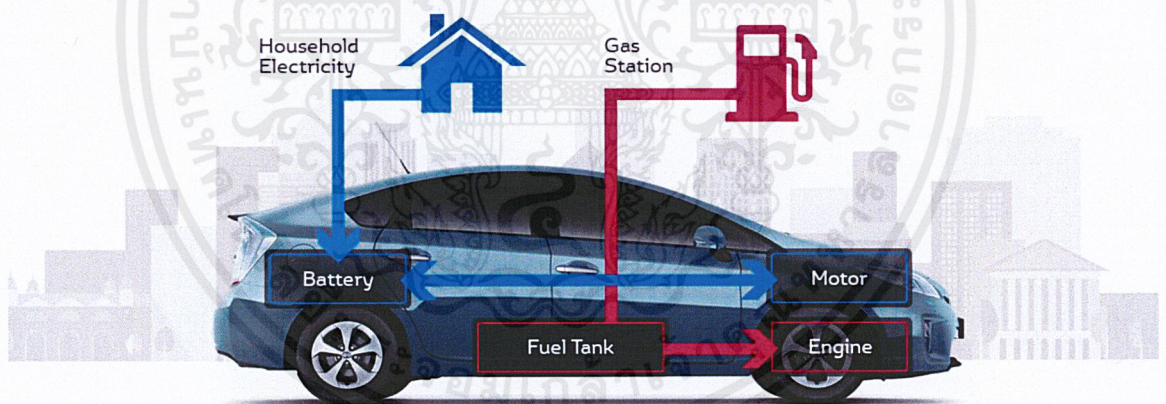
Full Hybrid จะยังคงใช้ระบบของเครื่องยนต์ดั้งเดิมในการขับเคลื่อนอยู่ แต่จะมีมอเตอร์ และ แบตเตอรี่ที่มีขนาดใหญ่กว่าของระบบ Mild Hybrid อยู่พอสมควร ซึ่งใหญ่พอที่จะพายานพาหนะเดินทางไปได้ด้วยการใช้พลังงานไฟฟ้าอย่างเดียว ๆ ราว 2 - 3 กิโลเมตร ในกรณีที่ชาร์จปริมาณแบตเตอรี่ไว้เต็ม ซึ่งมอเตอร์ไฟฟ้าจะช่วยออกตัวด้วยระบบพลังงานไฟฟ้าก่อนที่เครื่องยนต์จะติดแล้วรับหน้าที่ต่อ ช่วยทำงานพร้อมเครื่องยนต์ในช่วงที่เราต้องการอัตราเร่งสูง ซึ่งนอกจากจะได้อัตราเร่งที่ดีมากขึ้นแล้ว ยังช่วยลดการใช้ น้ำมันในสถานการณ์ต่าง ๆ ได้พอสมควร รวมถึงรถจอดหยุดนิ่ง แล้วมีปริมาณแบตเตอรี่มากพอ เครื่องยนต์จะดับ แล้วดึงพลังงานไฟฟ้าจากแบตเตอรี่มาใช้กับส่วนอื่น ๆ ในยานพาหนะ อาทิเช่น เครื่องปรับอากาศ, เครื่องเสียง, ไฟหน้ารถ เป็นต้น ทำให้ลดการใช้น้ำมันไปได้อีก มลพิษที่ถูกปล่อยออกมาก็ลดลง โดยแบตเตอรี่ที่ใช้ส่วนใหญ่จะมี 2 แบบก็คือ Lithium - ion และ Nickel – metal hydride รถยนต์ที่เป็นระบบ Full Hybrid ในปัจจุบัน ก็มีทั้ง Toyota Camry, Honda Accord เป็นต้น



รูป 2.3 ยานพาหนะพลังงานไฟฟ้าระบบ Full Hybrid

3. Plug - in Hybrid (PHEV)

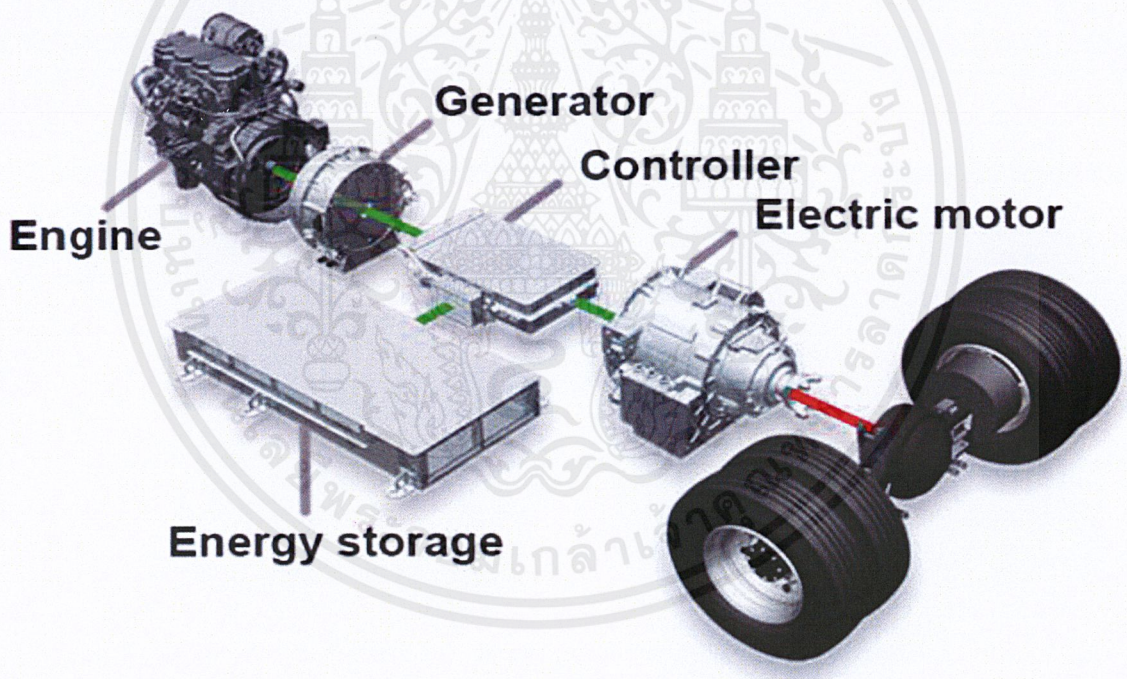
Plug - in Hybrid (PHEV) คือระบบยานพาหนะไฟฟ้าที่เพิ่มในส่วนของความจุแบตเตอรี่เข้าไป แต่ด้วยเฉพาะกำลังจากการขับเคลื่อนรถยนต์ หรือการใช้งานเครื่องยนต์ทั่วไป ไม่สามารถจะเพิ่มประจุไฟฟ้าให้เข้าไปอยู่ในแบตเตอรี่ที่มีขนาดใหญ่กว่าแบตเตอรี่ของ Full Hybrid หลายเท่าตัวได้ จึงต้องมีการใช้ไฟฟ้าจากภายนอกมาชาร์จเข้าไปโดยตรง เหมือนกับที่เราเสียบชาร์จโทรศัพท์มือถือ ซึ่งส่วนใหญ่จะมีการใช้สายชาร์จแบบเสียบปลั๊กที่อยู่ในบ้าน ใช้เวลาในการชาร์จราว 4 - 6 ชั่วโมงจนแบตเตอรี่เต็ม จะทำให้รถสามารถวิ่งไปด้วยระบบไฟฟ้าอย่างเดียว โดยที่เครื่องยนต์ไม่ต้องทำงาน ขับเคลื่อนได้ระยะทาง 20 - 50 กิโลเมตร ถ้าใช้ขับไปทำงานในระยะไม่เกินนี้ ก็แทบไม่ต้องเติมน้ำมันเพื่อใช้งาน ส่วนระบบการทำงานนั้นเหมือนกับแบบ Full Hybrid เพียงแต่ว่าในรถยนต์ PHEV วิ่งด้วยไฟฟ้าได้ไกลกว่าเท่านั้นเอง ส่วนรุ่นที่ใช้ระบบนี้ อาทิเช่น BMW 330e, Mitsubishi Outlander P-HEV, Chevrolet Volt, Mercedes-Benz C350e Plug-in Hybrid เป็นต้น



รูป 2.4 ยานพาหนะพลังงานไฟฟ้าระบบ Plug - in Hybrid (PHEV)

4. Series Hybrid

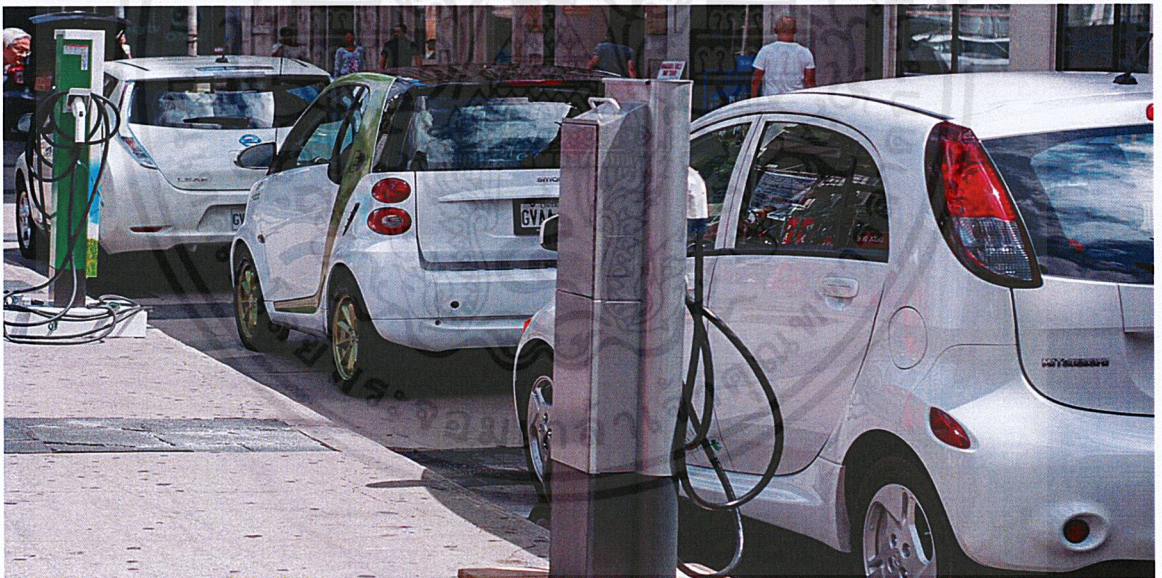
Series Hybrid คือรถยนต์ไฟฟ้าที่ใช้แต่มอเตอร์ไฟฟ้าในการขับเคลื่อนยานพาหนะเท่านั้น เพียงแต่ว่ายานพาหนะประเภทนี้จะมีแบตเตอรี่ที่ขนาดไม่ใหญ่มาก สามารถขับเคลื่อนได้ในระยะทางสั้น ๆ ดังนั้นจึงมีการติดตั้งเครื่องยนต์เข้าไปเพื่อใช้ในการผลิตไฟฟ้าเท่านั้น ไม่ได้มีการเชื่อมต่อกับระบบขับเคลื่อนอะไรทั้งสิ้น ทำหน้าที่ผลิตกระแสไฟเพื่อจ่ายไฟฟ้าเข้าสู่ที่แบตเตอรี่ และเครื่องใช้ไฟฟ้าต่าง ๆ ภายในตัวยานพาหนะ ก่อนที่แบตเตอรี่จะเอาไฟไปจ่ายเพื่อขับเคลื่อนมอเตอร์ให้รถวิ่งไปได้อีกที ดังนั้นการทำงานของเครื่องยนต์จะอยู่ในลักษณะนิ่ง จึงทำให้สามารถลดการใช้พลังงานได้มากกว่าในระบบ Full Hybrid และมีราคาที่ต้องจ่ายได้ ถูกกว่ารถยนต์ไฟฟ้าล้วนเพราะใช้แบตเตอรี่ก้อนไม่ใหญ่เท่านี้ รถยนต์ที่ใช้ระบบ Series Hybrid นี้ จะมีทั้ง Nissan Note E-Power, BMW i3 เป็นต้น



รูป 2.5 ยานพาหนะพลังงานไฟฟ้าระบบ Series Hybrid

5. Electric Vehicle (EV)

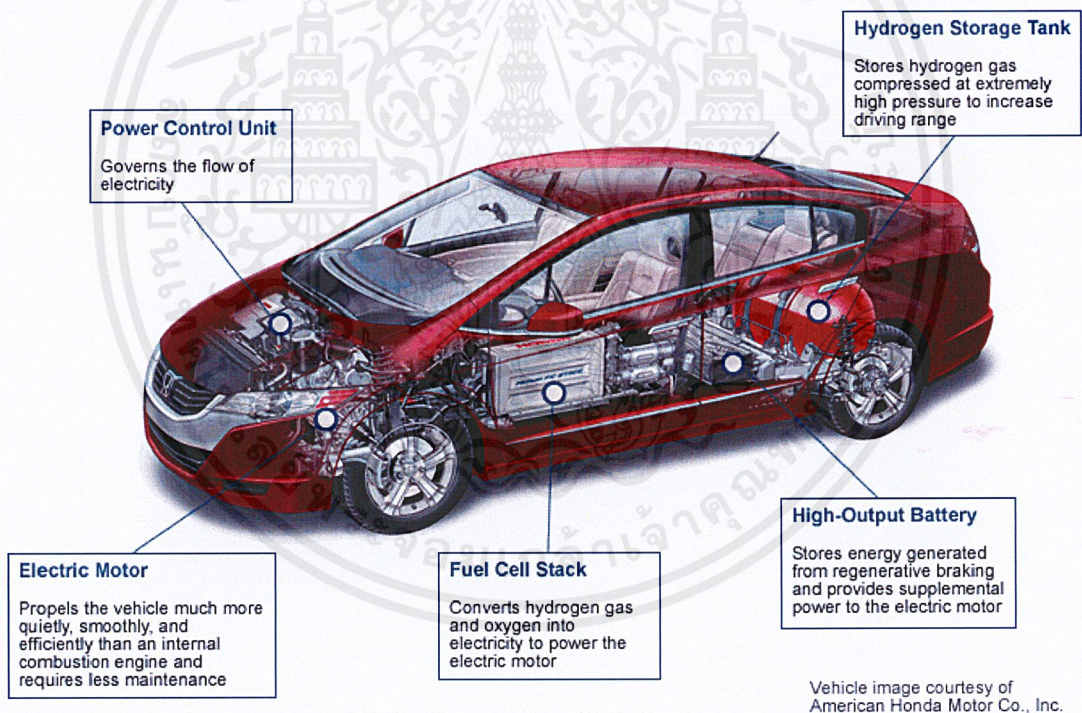
Electric Vehicle (EV) หรือยานพาหนะพลังงานไฟฟ้า ซึ่งรถยนต์ระบบนี้จะไม่มีการเครื่องยนต์ระบบสันดาปภายในเอาไว้ใช้งานเลย แหล่งพลังงานเดียวที่มีก็คือแบตเตอรี่ การชาร์จพลังงานก็ใช้จากที่ชาร์จภายในบ้าน หรือแท่นชาร์จแบบเร็วจากผู้ผลิตยานพาหนะที่ให้บริการ โดยรถยนต์ไฟฟ้าส่วนใหญ่จะมีมอเตอร์ไฟฟ้าขนาดใหญ่ และมีแบตเตอรี่ที่สามารถขับเคลื่อนไปได้อย่างน้อยเป็นระยะทาง 300 กิโลเมตรขึ้นไป โดยระบบนี้มีข้อจำกัดก็คือในเรื่องของการชาร์จไฟ ซึ่งส่วนใหญ่ในปัจจุบันยังคงใช้เวลาในการชาร์จที่นานมาก ระดับ 6 - 8 ชั่วโมงในการชาร์จปกติ หรือ 2 - 4 ชั่วโมงในการชาร์จผ่านแท่นชาร์จเร็ว ทำให้ยังมีความลังเลในการเปลี่ยนตัวเองให้มาใช้ยานพาหนะไฟฟ้านั้นเอง ปัจจุบันมีหลายค่ายที่กำลังหาทางแก้ปัญหานี้ ทั้งหาเทคโนโลยีในการชาร์จให้เร็วขึ้น หรือแม้กระทั่งการคิดค้นการเปลี่ยนแบตเตอรี่ลูกใหม่ที่ศูนย์บริการชาร์จไฟ ผู้ผลิตรถยนต์ไฟฟ้า EV นั้น นอกจาก Tesla แล้ว ก็ยังมี Nissan Leaf, Chevrolet Bolt, Renault Zoe เป็นต้น



รูป 2.6 ยานพาหนะพลังงานไฟฟ้าระบบ Electric Vehicle

6. Fuel Cell

Fuel Cell คือยานพาหนะที่ใช้พลังงานไฮโดรเจน ใช้กรรมวิธีทางวิทยาศาสตร์ในการเปลี่ยนไฮโดรเจนที่ถือว่าเป็นสารตามธรรมชาติที่สะอาด แปลงให้กลายเป็นพลังงานไฟฟ้า แล้วนำไปขับเคลื่อนมอเตอร์ไฟฟ้าให้ยานพาหนะสามารถขับเคลื่อนได้อีกที เป็นการต่อยอดมาจากยานพาหนะไฟฟ้าเพื่อแก้ไขข้อจำกัดเรื่องการเติมพลังงาน โดยใช้เวลาในการเติมให้เต็มได้ภายใน 5 นาที คล้ายกับการเติมก๊าซ CNG ในรถยนต์ แต่พารายนต์วิ่งไปได้ระดับ 300 กิโลเมตรขึ้นไป แต่ด้วยเทคโนโลยี Fuel Cell ยังไม่เสถียรมากพอที่จะใช้งานได้ทั่วไป รวมทั้งการลงทุนในสถานีเติมไฮโดรเจนก็ยังคงสูง ทำให้ในปัจจุบันจะมีการใช้รถยนต์พลังงานไฮโดรเจน อยู่ไม่กี่เมือง เช่นในญี่ปุ่น, รัฐแคลิฟอร์เนีย สหรัฐอเมริกา และในยุโรปบางประเทศ ส่วนรถยนต์ Fuel Cell ในปัจจุบันก็มีอยู่ไม่กี่รุ่น อาทิ Toyota Mirai, Honda Clarity เป็นต้น



รูป 2.7 ยานพาหนะพลังงานไฟฟ้าระบบ Fuel Cell

2.1.3 หน่วยควบคุมอิเล็กทรอนิกส์สำหรับยานพาหนะ (Electric control unit : ECU)

หน่วยควบคุมอิเล็กทรอนิกส์สำหรับยานพาหนะ (Electric control unit : ECU) หรือกล่อง ECU เป็นอุปกรณ์อิเล็กทรอนิกส์ มีพื้นฐานมาจากคอมพิวเตอร์ รูปทรงสี่เหลี่ยมขนาดประมาณฝ่ามือ หน้าที่หลักทำงานคล้ายกับสมองของยานพาหนะ โดยทำหน้าที่รับข้อมูลจากเซ็นเซอร์ต่าง ๆ รอบตัวของยานพาหนะ เพื่อนำมาประมวลผล และใช้ในการควบคุมการทำงานต่าง ๆ เช่น ใช้ในการควบคุมการการลั้งจ่ายน้ำมันเชื้อเพลิง การจุดระเบิดของเครื่องยนต์ รวมไปถึงการตรวจสอบสถานะการทำงานของอุปกรณ์อื่น ๆ ในยานพาหนะอีกด้วย



รูป 2.8 กล่อง ECU)

2.1.4 ประเภทหน่วยควบคุมอิเล็กทรอนิกส์สำหรับยานพาหนะ (Electric control unit : ECU)

แบ่งตามชนิดของกล่อง แบ่งออกเป็น 2 ชนิด

1. กล่องมาตรฐาน (Standard ECU)

กล่องมาตรฐาน (Standard ECU) คือกล่องควบคุมระดับมาตรฐานซึ่งถูกติดตั้งมาเป็นมาตรฐานในยานพาหนะทั่วไป โดยข้อดีของกล่องเดิมคือ ถูกปรับแต่งมาให้มีความสมดุล รวมถึงสามารถเข้าศูนย์บริการของทางผู้ผลิตรายานพาหนะเพื่อแก้ไขเมื่อมีปัญหาเกิดขึ้นได้



รูป 2.9 กล่อง Standard ECU

2. กล่องปรับแต่งได้ (Modify ECU)

กล่องปรับแต่งได้ (Modify ECU) คือกล่องที่ถูกเขียนโปรแกรมเสริมเพิ่มเติมขึ้นมาจากค่ามาตรฐานของโรงงาน และมีการอัปเดตโปรแกรมขึ้นมาใหม่เพื่อให้เหมาะสมกับอุปกรณ์ที่เราใส่เพิ่มเข้าไปในเครื่องยนต์



รูป 2.10 กล่อง Modify ECU

แบ่งตามชนิดการใช้งานกล่อง แบ่งออกเป็น 2 วิธี

1. กล่องพ่วง (Piggy - Back ECU)

กล่องพ่วง (Piggy - Back ECU) คือกล่องที่สามารถต่อพ่วงกับกล่องเดิมได้เลย สามารถเขียนโปรแกรมลงกล่องพ่วงเพื่อเพิ่มประสิทธิภาพของเครื่องยนต์ให้มากกว่าเดิม ยี่ห้อที่ได้รับความนิยม เช่น AEM, Unichip, Vipec, Data-Tec, KKT และ ECU=SHOP เป็นต้น



รูป 2.11 กล่อง Piggy - Back ECU

2. กล่องเดี่ยว (Standalone ECU)

กล่องเดี่ยว (Standalone ECU) คือกล่องที่ติดตั้งใช้งานแทนตัวกล่องเดิมได้เลย โดยจะมีประสิทธิภาพสูงกว่า มักจะใช้สำหรับการแข่งขันกีฬา และมีราคาที่สูงกว่ามาก เช่น F-Con, Motec, ECU=SHOP เป็นต้น



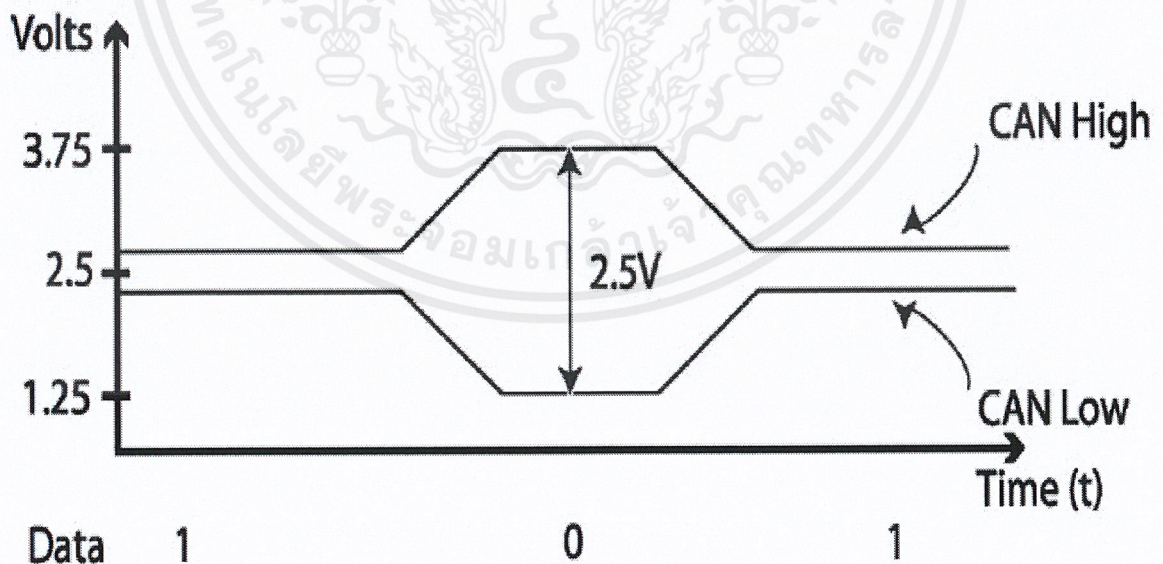
รูป 2.12 กล่อง Standalone ECU

2.2 โพรโตคอลการสื่อสารข้อมูลภายในยานพาหนะ

2.2.1 CAN Protocol

CAN Protocol (Controller Area Network Protocol) เป็นเทคโนโลยีเครือข่ายอนุกรมความเร็วสูงผ่านสองสายเชื่อมต่อ เป็นการสื่อสารแบบ Half - Duplex โดยทั่วไปจะใช้ในการสื่อสารระหว่างอุปกรณ์ต่าง ๆ ที่อยู่ในพื้นที่ขนาดเล็ก เช่นในรถยนต์ CAN Protocol เป็นโพรโตคอลแบบ CSMA - CD / ASM หรือก็คือผู้ให้บริการต้องรับรู้ถึงการเข้าถึง และตรวจจับการชนกันของข้อมูลหลายอย่างบนโพรโตคอลข้อความสำคัญ CSMA ทำให้แน่ใจว่าแต่ละ Node ต้องรอเวลาที่กำหนดก่อนที่จะส่งข้อความใด ๆ การตรวจจับการชนกันทำให้แน่ใจได้ว่าหลีกเลี่ยงการชนกัน โดยมีการเลือกข้อมูลตามลำดับความสำคัญที่กำหนดไว้

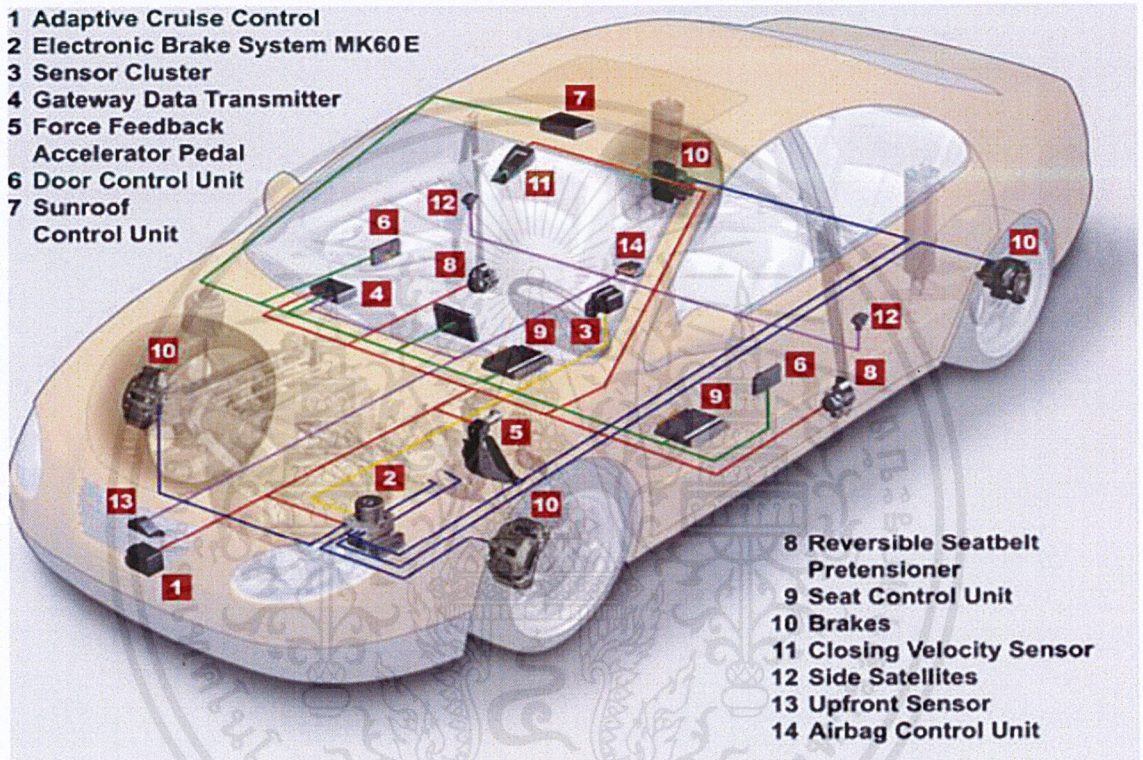
CAN Protocol สื่อสารกันโดยใช้สายไฟสองสายทำงาน เรียกสายแรกว่า CAN - High และอีกสายคือ CAN - Low โดยเมื่อทั้งสองอยู่ในสภาวะว่าง จะมีแรงดันไฟ 2.5 V แต่เมื่อต้องการส่งข้อมูล Bit ไปนั้น สาย CAN High จะมีแรงดันไฟสูงขึ้นเป็น 3.75 V ส่วน CAN Low แรงดันจะลดลงเหลือ 1.25 V นั่นคือมีความต่างศักย์อยู่ที่ 2.5 V



รูป 2.13 CAN Protocol

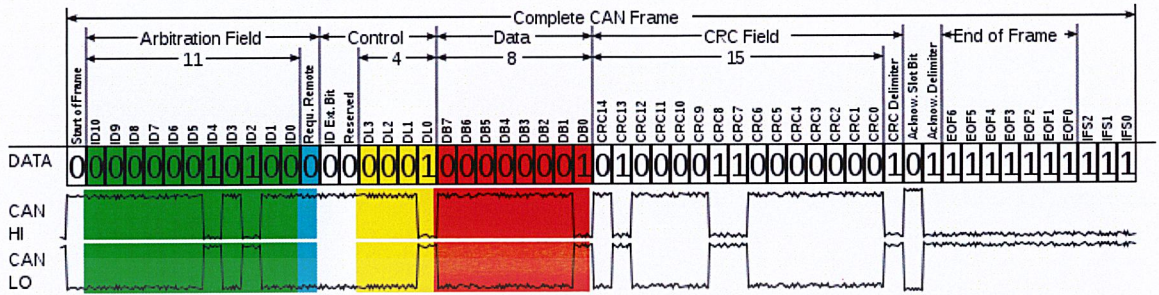
2.2.2 CAN – BUS Protocol

CAN – BUS Protocol คือการนำระบบการสื่อสารรูปแบบ CAN โปรโตคอล มาใช้ใน ระบบการสื่อสารภายในยานพาหนะหลายชนิด อาทิเช่น รถยนต์ รถจักรยานยนต์ และรถบรรทุกของ เป็นต้น โดยเป็นลักษณะของการสื่อสาร ด้วยการเข้ารหัสข้อมูล และส่งผ่านทางสายสัญญาณ 2 สาย คือ CAN - High และ CAN – Low ซึ่งส่งข้อมูลผ่านไปตลอดทั่วตัวยานพาหนะ

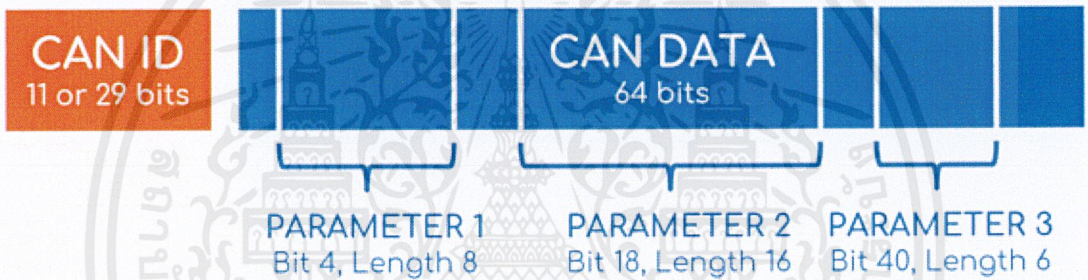


รูป 2.14 CAN – BUS Protocol Overview

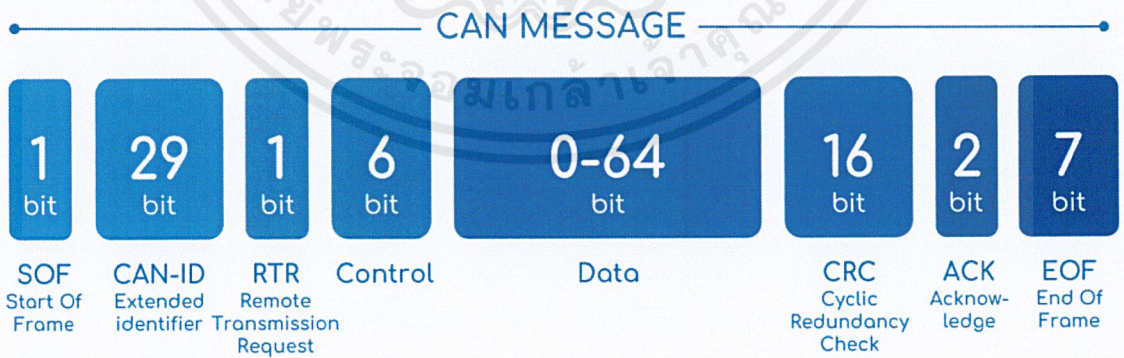
2.2.3 รูปแบบข้อมูล CAN - BUS



รูป 2.15 CAN - BUS Protocol



รูป 2.16 CAN - BUS Data



รูป 2.17 CAN - BUS Message Frame

1. SOF (Start – of - Frame)

เป็นจุดที่ ระบุการเริ่มต้นของข้อความ

2. CAN – ID or Arbitration ID

ระบุข้อความตัวบ่งชี้ชนิดของข้อความ และระบุลำดับความสำคัญของข้อความ มี Frame 2 แบบ ได้แก่ แบบมาตรฐาน ซึ่งใช้รหัสบ่งชี้ 11 Bit และแบบขยาย ซึ่งใช้รหัสบ่งชี้ 29 Bit

3. RTR (Remote transmission request)

ใช้ระบุคำขอการส่งข้อมูลระยะไกล โดยบอกได้ว่าเป็นการส่งข้อมูลด้วยตัวข้อความจากอุปกรณ์เอง หรือมีการร้องข้อความ จาก Node อื่น แยกความต่างของข้อความ

5. Control or DLC (Data length code)

ระบุขนาดความยาวมีกี่ Byte

6. Data field

รายละเอียดข้อความที่ส่งมา

7. CRC (Cyclic redundancy check)

มีรหัสการตรวจสอบความซ้ำซ้อนแบบวนซ้ำ 15 Bit และตัวค้นแบบถอยกลับใช้สำหรับตรวจจับข้อผิดพลาด

8. ACK (ACKnowledgement)

เมื่ออุปกรณ์ปลายทางที่ส่งข้อมูลได้รับข้อความแล้ว จะต้องส่ง ACK ของข้อความกลับมาในระบบ หาก Node ที่ส่งมาไม่พบ ACK บนบัส จะพยายามส่งข้อความอีกครั้ง

9. EOF (End of frame)

จบ Frame ข้อความ 1 ชุด

2.2.4 ความเร็วในการสื่อสารข้อมูล CAN – BUS Protocol

1. Class A

ความเร็วในการรับส่งข้อมูลน้อยกว่า 10 Kbps ระบบที่ใช้ได้แก่ กระบอกไฟฟ้า, เบาะไฟฟ้า, ระบบล้อต่าง ๆ, รีโมท, ระบบไฟส่องสว่าง เป็นต้น

2. Class B

ความเร็วในการสื่อสารอยู่ระหว่าง 10 Kbps – 125 Kbps ซึ่งก็ได้แก่โปรโตคอล ISO 9141 – 2 หรือ SAE J 1850 ระบบนี้จะมีความเร็วเพียงพอในการสื่อสารข้อมูลที่มีความซับซ้อน ได้แก่ ระบบเกียร์อัตโนมัติ, ระบบปรับอากาศ, ระบบอิเล็กทรอนิกส์บริเวณแผงควบคุมต่าง ๆ เป็นต้น

3. Class C

ทำความเร็วในการสื่อสารอยู่ที่ 1 Mbps หรือ ประมาณ 10 เท่าของ Class B แต่โดยทั่วไปจะใช้ที่ความเร็วประมาณ 500 Kbps ซึ่งมีความเร็วเพียงพอ สำหรับระบบ Air Bag, ABS, Stability Control, Traction Control, Power Train Control Module เป็นต้น

4. Class D

ความเร็วในการสื่อสารที่ 1 Mbps ใช้ในระบบ Onboard Entertainment เช่น Video Streaming ระบบติดต่อสื่อสารกับดาวเทียม รวมถึงระบบ Internet ที่อนาคตจะนำมาใช้ในรถยนต์

2.2.5 ประโยชน์ของระบบ CAN – BUS Protocol และการประยุกต์ใช้งาน

ประโยชน์ของ CAN – BUS Protocol

1. ประหยัดต้นทุน

เป็นเครือข่ายที่ใช้ต้นทุนในการสร้างต่ำ อุปกรณ์ต่าง ๆ ในระบบมีมีน้ำหนักที่เบา รวมถึงอุปกรณ์ในระบบมีความทนทานซึ่งช่วยให้อุปกรณ์ CAN หลายเครื่องสามารถสื่อสารกันได้ดีในระยะยาว

2. เป็นระบบกระจายสัญญาณ

เป็นการสื่อสารแบบกระจายสัญญาณออกอากาศ (Broadcast) อุปกรณ์แต่ละชิ้นในเครือข่ายมีหน่วยควบคุม CAN – BUS และยังเป็นอุปกรณ์ที่สามารถตัดสินใจในการทำงานได้ด้วยตัวเอง นอกจากนี้ อุปกรณ์ทั้งหมดในเครือข่ายจะสามารถตรวจจับข้อมูลที่ส่งทั้งหมด อุปกรณ์แต่ละตัวสามารถตัดสินใจได้ว่าข้อมูลนั้นเกี่ยวข้องกับระบบที่ดูแลอยู่หรือไม่ก่อนที่จะนำมาใช้งาน

3. การลำดับความสำคัญ

ทุกข้อมูลมีการกำหนดลำดับความสำคัญอย่างชัดเจน ดังนั้นหากสอง Node มีการส่งข้อความออกไปพร้อมกัน ข้อมูลที่มีลำดับความสำคัญสูงกว่าจะได้รับการส่งไปก่อน และข้อมูลที่มีลำดับความสำคัญต่ำกว่าจะถูกเลื่อนเวลาการส่งออกไป ทำให้ระบบมีการส่งข้อมูลที่สั้นไหล แยกแยะข้อมูลลำดับความสำคัญสูงสุด สามารถควบคุมเวลาการทำงานได้

4. ระบบตรวจสอบข้อผิดพลาด

มีระบบตรวจสอบหาข้อผิดพลาด ข้อมูลจำเพาะของ CAN ประกอบด้วยรหัส Cyclic Redundancy (CRC) ซึ่งมีไว้เพื่อทำการตรวจสอบข้อผิดพลาดในเนื้อหาของแต่ละ Frame เมื่อพบข้อผิดพลาด จะมีการหยุดการทำงาน Node ทั้งหมด ซึ่งเฟรมที่มีข้อผิดพลาดสามารถส่งสัญญาณเตือนการพบข้อผิดพลาดไปยังเครือข่าย หากตรวจพบข้อผิดพลาดมากเกินไปแต่ละ Node สามารถหยุดการส่งข้อมูลที่ผิดพลาด หรือตัดการเชื่อมต่อตัวเองออกจากเครือข่ายได้อย่างสมบูรณ์

การประยุกต์ใช้ CAN – BUS Protocol

1. ยานพาหนะประเภทอื่น ๆ

CAN – BUS Protocol สามารถใช้กับ รถบัสโดยสาร รถไฟ รถไฟฟ้าใต้ดิน รถไฟขนาดเล็ก และรถไฟทางไกล ซึ่งทำงานหลากหลายหน้าที่ควบคุมระบบ ตัวอย่างเช่น ประตู ตัวควบคุมเบรก ระบบนับจำนวนผู้โดยสาร เป็นต้น

2. การบิน

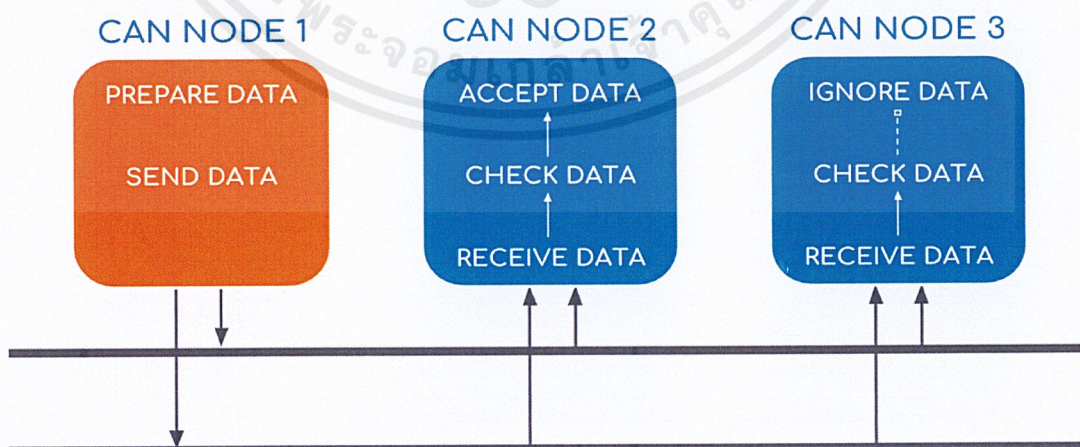
CAN – BUS Protocol ใช้กับเครื่องบินที่มีเซ็นเซอร์ตรวจจับสถานะการบิน ระบบนำทาง และการตรวจสอบข้อมูลในห้องนักบิน รวมถึงระบบการบิน และอวกาศหลายประเภทตั้งแต่การวิเคราะห์ข้อมูลในเที่ยวบิน จนถึงระบบควบคุมเครื่องยนต์อากาศยาน เช่น ระบบเชื้อเพลิงปั๊ม แอคชูเอเตอร์เชิงเส้น

3. การแพทย์

การแพทย์ใช้ CAN – BUS Protocol มาเป็นเครือข่ายแบบฝังตัวในอุปกรณ์การแพทย์ ใช้จัดการห้องผ่าตัดที่สมบูรณ์ รวมถึงควบคุมส่วนประกอบของห้องปฏิบัติการ เช่น เครื่อง X-ray และเตียงผู้ป่วย

4. ธุรกิจอื่น ๆ

ใช้งานในอุตสาหกรรมอื่น ๆ เช่น อุปกรณ์ห้องปฏิบัติการ กล้องกีฬา กล้องโทรทรรศน์ ประตูอัตโนมัติ และเครื่องชงกาแฟ เป็นต้น



รูป 2.18 CAN – BUS Node

2.2.6 OBD Protocol

OBD Protocol (On - board Diagnostics Protocol) เป็นระบบวิเคราะห์การทำงานของเครื่องยนต์ พร้อมแจ้งเตือนคนขับเพื่อบอกสาเหตุถึงความผิดปกติทั้งในส่วนของ Software และ Hardware ของระบบยานพาหนะ การวิเคราะห์และแจ้งเตือนแบบ OBD นั้น ถือเป็นมาตรฐานที่ผู้ผลิตรถยนต์ต้องปฏิบัติตาม เพื่อให้รถยนต์ทุกค่ายมีความสอดคล้องกัน โดยในอดีตนั้นรถยนต์ทุกค่ายจะต้องมีระบบ OBD (OBD Level 1) ระบบ OBD นั้นเป็นระบบวิเคราะห์เครื่องยนต์เบื้องต้น เมื่อหน่วยควบคุมอิเล็กทรอนิกส์สำหรับยานพาหนะ (ECU) ตรวจพบว่าเครื่องยนต์หรือระบบใด ๆ มีการทำงานที่ผิดปกติ จะปรากฏหลอดไฟกระพริบที่แผงหน้าปัด แต่จะไม่แจ้งสาเหตุว่าความผิดปกตินั้นว่าเกิดขึ้นเพราะอะไร ซึ่งเป็นข้อเสียของระบบ OBD

ระบบ OBD ได้เริ่มเข้ามามีบทบาทในระบบยานพาหนะตั้งแต่ต้นปี คริสต์ศักราช 1980 โดยฟังก์ชัน OBD นั้น จะถูกฝังอยู่ในหน่วยควบคุมอิเล็กทรอนิกส์สำหรับยานพาหนะ และตรวจสอบการทำงานของระบบต่าง ๆ อย่างต่อเนื่อง หลังจากนั้นสมาพันธ์ยานยนต์นานาชาติหรือ SAE จึงได้เล็งเห็นความสำคัญของระบบ OBD และได้มีการผลักดันให้ผู้ผลิตรถยนต์พัฒนาระบบ OBD ให้มีความสอดคล้องและมีความล้ำสมัยมากขึ้น จนกระทั่งกลายมาเป็น OBD - II ที่เราสามารถพบเห็นได้อย่างแพร่หลายในยานพาหนะยุคปัจจุบัน

2.2.7 OBD - II Protocol

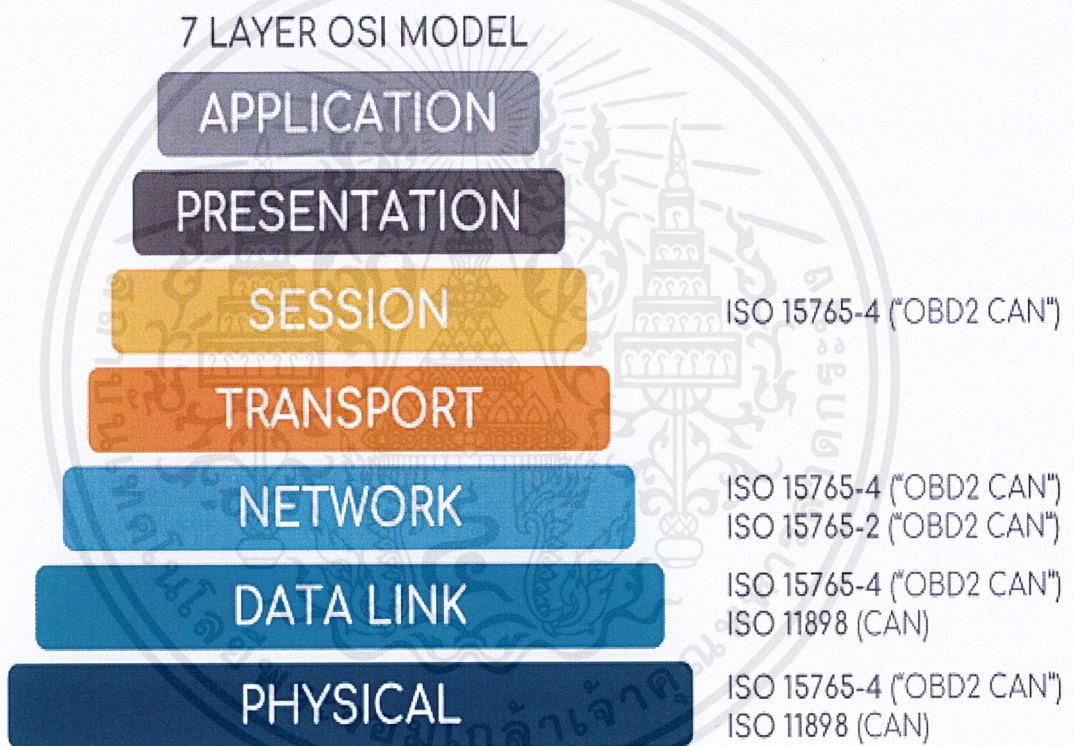
OBD - II Protocol (On - board Diagnostics Protocol version 2) ได้รับการพัฒนาต่อยอดมาจาก OBD สำหรับ OBD - II นั้น นอกจากจะสามารถแจ้งเตือนถึงความผิดปกติแล้ว ยังสามารถบ่งบอกสาเหตุของปัญหาได้อีกด้วย จึงเป็นประโยชน์สำหรับผู้ใช้นยานพาหนะโดยตรง รวมไปถึงศูนย์บริการที่ทำหน้าที่ตรวจสอบยานพาหนะ ปกติแล้วช่องเชื่อมต่อ OBD - II จะถูกติดตั้งไว้ภายในห้องโดยสารด้านล่างของ Console บริเวณคนขับ

นอกจากระบบ OBD - II จะสามารถตรวจสอบและแจ้งเตือนความผิดปกติได้แล้ว อีกหนึ่งวัตถุประสงค์ที่ SAE บังคับใช้ระบบ OBD - II กับรถยนต์ทุกประเภทนั้น ก็เกี่ยวข้องโดยตรงกับมาตรฐานมลพิษทางไอเสีย ยกตัวอย่างเช่น ถ้าหากรถยนต์คันหนึ่งมีระบบการเผาไหม้ที่ผิดปกติ หรือมีระบบกรองไอ

เสียที่ผิดปกติ จนทำให้มีปริมาณไอเสียเกินมาตรฐาน ระบบ OBD - II จะแจ้งเตือนโดยการโชว์ไฟเตือนที่แผงหน้าปัดโดยทันที เพื่อให้ผู้ขับขี่จะได้นำยานพาหนะเข้าไปตรวจสอบที่ศูนย์บริการเพื่อหาสาเหตุ

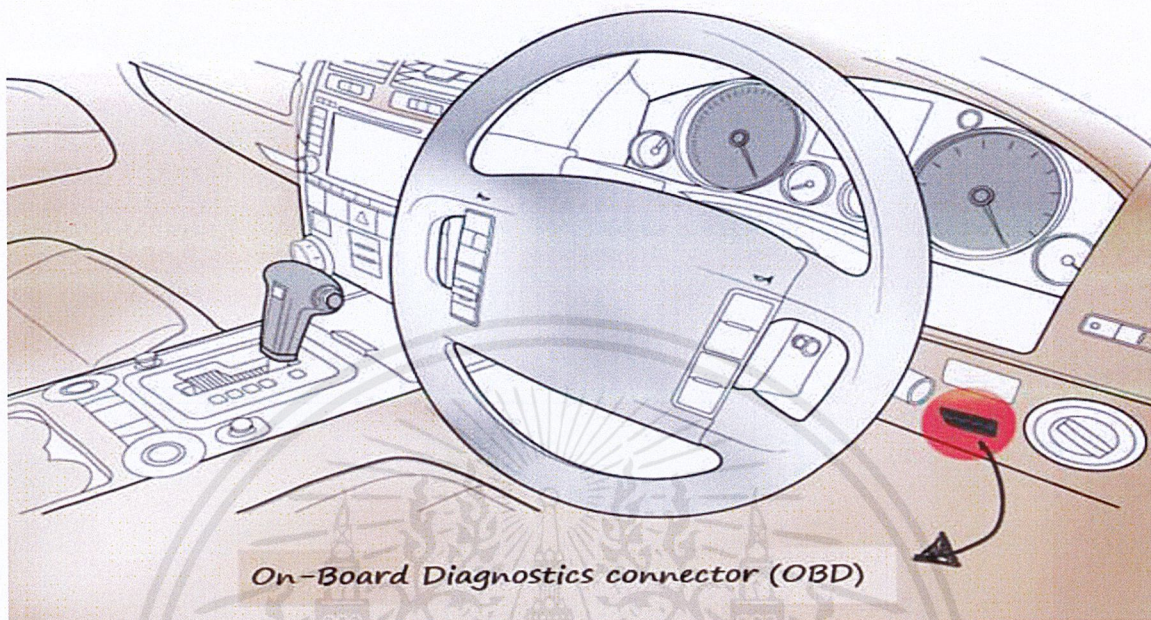
2.2.8 ความแตกต่างระหว่าง OBD - II Protocol กับ CAN – BUS Protocol

OBD - II Protocol เป็น Higher Layer ของ OSI Layer CAN – BUS Protocol ก็คือ ส่วนที่เป็นส่วนติดต่อกับผู้ใช้งาน ซึ่งเป็น Port ที่ใช้เชื่อมต่อ และกำหนดมาตรฐานที่ใช้งานซึ่งทำงานสอดคล้องกับ CAN – BUS Protocol อย่างดี

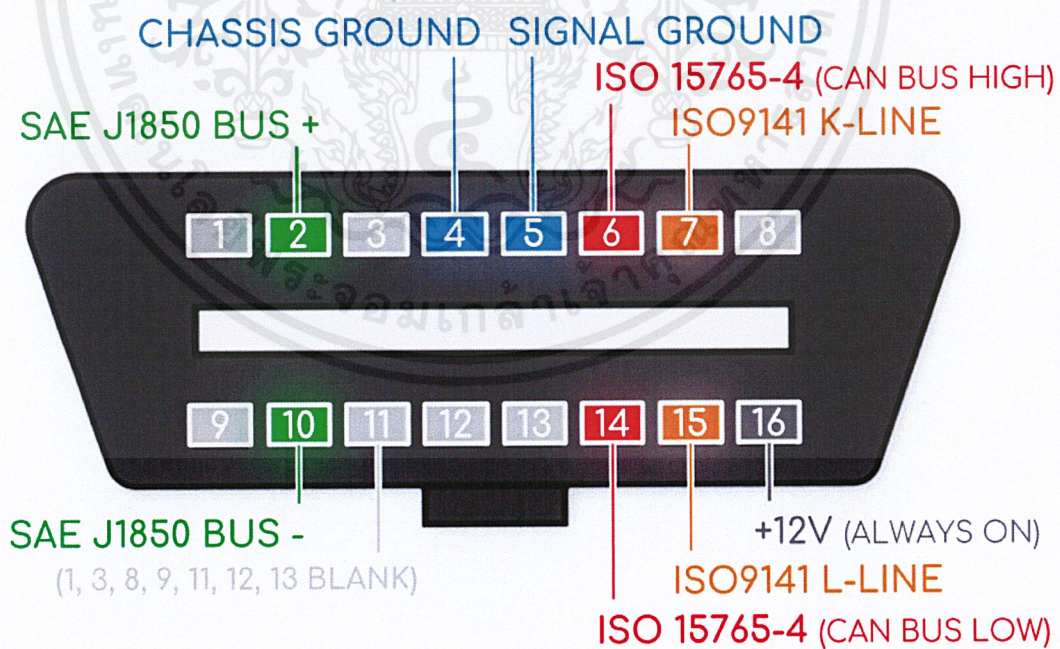


รูป 2.19 OBD – II and CAN - BUS

2.2.9 Port การเชื่อมต่อ OBD - II



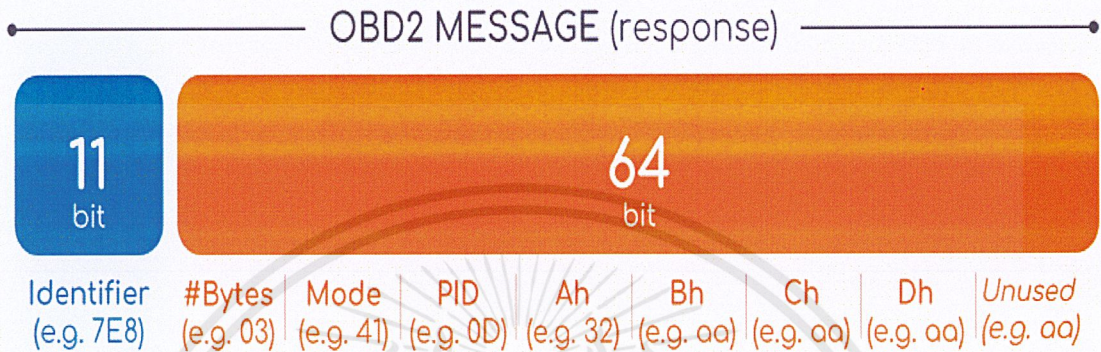
รูป 2.20 OBD - II Port Position



รูป 2.21 OBD - II Port

2.2.10 รูปแบบข้อมูล OBD – II Protocol

ชุดข้อความที่ส่งผ่าน OBD – II Protocol เราเรียกว่า Message หรือ Container ภายในมีโครงสร้าง ซึ่ง OBD - II Message นั้น แบ่งเป็น 2 ส่วนหลัก ๆ ก็คือ Identifier และ Data



รูป 2.21 OBD – II Message Frame

1. Identifier

ตัวบ่งชี้ (Identifier) คือ รหัสที่ใช้ระบุ Request Message ที่ส่งไปหาอุปกรณ์ และ Response Message ที่อุปกรณ์นั้นส่งกลับมาหาเรา เช่น

Request : 7DF 02 01 0D 55 55 55 55 55 ค่า 7DF คือ Identifiers
Data คือ ส่วนที่เก็บข้อมูลในแต่ละ Message โดยเฉพาะ

2. ข้อมูล (Data)

ข้อมูล (Data) แบ่งได้ 8 ส่วน ได้แก่ Length > Mode > PID > Ah > Bh > Ch > Dh

- Length

เป็นตัวระบุว่าในข้อความนั้นตัวข้อความคู่ไหนที่ใช้งาน โดยนับจากชุดข้อมูลถัดไป เช่น

Request: 7DF 02 01 0D 55 55 55 55 55 ; Request มี 2 Byte ที่ใช้งาน คือ 01 และ 0D

Response: 7E8 03 41 0D 32 aa aa aa aa ; Response มี 3 Byte ที่ใช้งาน คือ 41 0D และ 32

- Mode

Mode มีค่าแตกต่างกันไปตามมาตรฐานที่ใช้ โดยจะเป็นตัวระบุว่าใน Request / Response นั้น ๆ ว่ากำลังจะทำอะไร จะดูค่า หรือเช็คข้อผิดพลาด เป็นต้น

- Parameter ID

Parameter ID (PID) โดย 1 mode มีหลาย PIDs แต่ละ PID จะเป็นตัวบอกค่าที่ติงมานั้น หมายถึงค่าอะไร เช่น ความเร็วรอบ MAF / MAP ฯลฯ ตัวอย่าง

Response: 7E8 03 41 0D 32 aa aa aa aa

ค่า PID 0D หมายถึง Vehicle speed

- A - B - C - D

เป็นพื้นที่เก็บข้อมูลดิบ อยู่ในรูปแบบเลขฐาน 16 จะต้องแปลงให้เป็นเลขฐาน 10 และเข้าสู่ตรรกะบางอย่างก่อนจึงจะสามารถนำมาใช้งานได้ ตัวอย่าง

Response: 7E8 03 41 0D 32 aa aa aa aa

ค่า PID 0D หมายถึง Vehicle speed

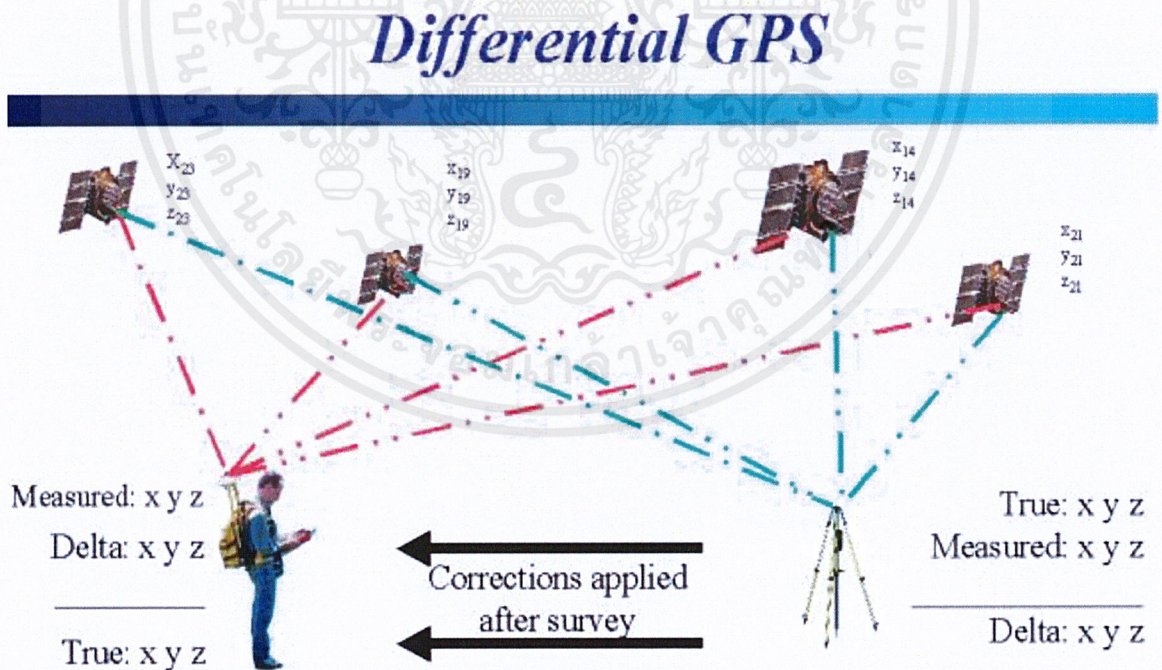
ค่า 32 คือค่าความเร็ว ตามตารางไม่ต้องบวกลบคูณหาร แปลงให้เป็นฐาน 10 = 50 km/h

2.3 Global positioning system (GPS)

2.3.1 Global positioning system (GPS) คืออะไร ?

Global positioning system (GPS) คือ ระบบการระบุตำแหน่งบนพื้นโลกที่มีการทำงานร่วมกับดาวเทียมบอกตำแหน่งทั้งหมดจำนวน 24 ดวง สามารถบอกระดับพิกัดตำแหน่ง (X, Y, Z) ความเร็ว และเวลาได้ โดยดาวเทียมต่าง ๆ นี้จะโคจรรอบโลกเพื่อส่งสัญญาณให้กลับมายังจุดรับสัญญาณบริเวณต่าง ๆ ที่อยู่ในโลกของเรา เช่น มือถือ เร็, รถยนต์ และมีการคำนวณระยะห่างจากดาวเทียมอย่างน้อย 3 ดวง กับจุดรับสัญญาณ ดาวเทียมเหล่านี้เป็นดาวเทียมที่มีวงโคจรระดับกลาง โคจร ความสูงประมาณ 20,000 กิโลเมตร จากพื้นผิวโลก

GPS ถูกพัฒนาขึ้นตั้งแต่ปี คริสต์ศักราช 1957 สำหรับระบบการระบุตำแหน่งบนพื้นโลก จะทำงานควบคู่ไปกับดาวเทียม เอาไว้ใช้ในการระบุตำแหน่งบนพื้นผิวโลกของเรา ตัวเครื่องที่ทำการรับสัญญาณ ต้องมีการประมวลผลเรื่องความแตกต่างของเวลาดำเนินการรับสัญญาณเทียบกับเวลาจริง ส่งผลให้สามารถระบุตำแหน่งต่าง ๆ บนพื้นผิวโลกของเราได้อย่างชัดเจน และเป็นสิ่งที่เรานำมาใช้งานกันอยู่



รูป 2.22 GPS (1)

2.3.2 ส่วนประกอบของระบบ GPS

1. ส่วนอวกาศ

มีดาวเทียมเครือข่าย 3 ค่ายหลัก ๆ คือ สหรัฐอเมริกา ยุโรป และรัสเซีย ทำหน้าที่ในการตรวจสอบตำแหน่งตามสัญญาณที่สื่อสารกับบนผิวโลก

2. ส่วนควบคุม

จะมีสถานีภาคพื้นดินอยู่ที่ Falcon Air Force Base ในสหรัฐอเมริกา พร้อมศูนย์ควบคุม ย่อยอีกทั้งหมด 5 จุด กระจายออกไปยังพื้นที่ต่างๆ ทั่วโลกให้ได้ใช้งานกัน

3. ส่วนผู้ใช้งาน

คนที่ใช้งานระบบ GPS ต้องมีเครื่องรับสัญญาณที่รับคลื่นพร้อมมีการแปรรหัสจากดาวเทียมสำหรับนำมาประมวลผลให้เหมาะกับการใช้งานในรูปแบบต่างๆ ได้นั่นเอง

2.3.3 ประเภทของระบบ GPS

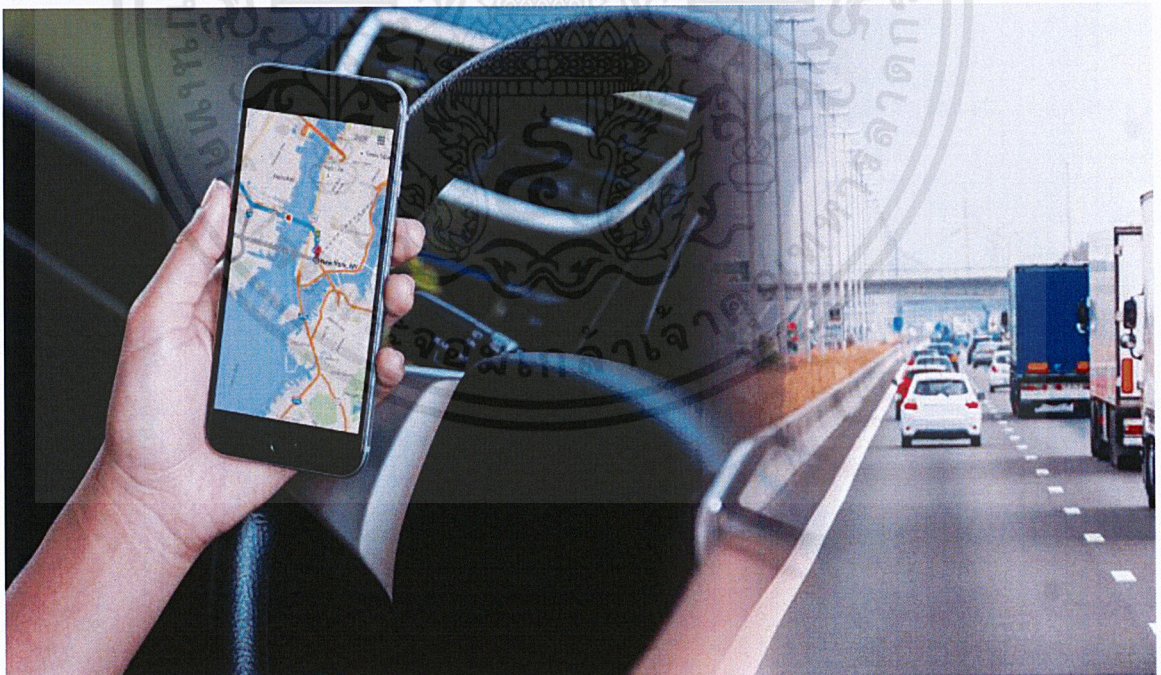
1. GPS นำทาง

เป็นอุปกรณ์และระบบนำทางที่เราคุ้นเคยกันมากเนื่องจากจะใช้งานกับรถยนต์ทั่วไปสำหรับบอกแผนที่เส้นทางต่าง ๆ ที่เราต้องการเดินทางไปแต่ไม่แน่ใจว่าต้องเดินทางผ่านถนนเส้นไหน เลี้ยวเข้าซอยอะไร การทำงานของระบบการระบุตำแหน่งบนพื้นโลก คือ ต้องมีการบอกพิกัดตำแหน่งปลายทางที่เราต้องการไปเมื่อบอก พิกัดเรียบร้อยแล้ว GPS จะทำการคำนวณเส้นทางที่เหมาะสมพร้อมระยะทางจากจุดที่คุณอยู่และบอกเวลาได้ด้วยว่าต้องใช้ประมาณกี่นาทีจึงจะถึงจุดหมายปลายทาง

2. GPS ติดตาม

อุปกรณ์และระบบติดตามรถ ยานพาหนะและสัตว์เลี้ยง สำหรับ GPS ประเภทนี้จะมีลักษณะ ทำนองติดตามการเดินทางพร้อมบอกพิกัดกับตำแหน่งของตัวเครื่อง GPS ได้ อีกทั้งเรายังสามารถแยกย่อย ออกได้อีก 2 แบบ คืออุปกรณ์ติดตามรถแบบออฟไลน์ เอาไว้ใช้เพื่อตรวจสอบประวัติการเดินทาง แต่จะตรวจสอบเรื่องของตำแหน่งที่ตั้งของที่อยู๋เครื่อง GPS นั้นไม่ได้ อีกแบบคือ อุปกรณ์ติดตามรถกึ่ง ออฟไลน์ อุปกรณ์ตัวนี้ทำงานร่วมกับมือถือนั้นทำให้สามารถดูในส่วนของประวัติการเดินทางได้ รวมถึง ตำแหน่งที่ตั้งปัจจุบันของตัวอุปกรณ์ GPS นั้นได้ด้วยว่าตอนนี้อยู่ตรงไหนแล้ว

การทำงานของระบบ GPS ทั่วไปจะมีระยะเวลาในการทำงานเบื้องต้นอยู่ราว 30 - 40 วินาที เพื่อเริ่มต้นระบุตำแหน่งตามความต้องการของเรา มีการใช้ข้อมูลการโคจรจากดาวเทียมเพื่อคำนวณ ตำแหน่งในปัจจุบันออกมา อัตราการส่งสัญญาณอยู่ที่ 50 bps อย่างไรก็ตามการดาวน์โหลดข้อมูลเกี่ยวกับ วงโคจรดังกล่าวนี้อาจใช้เวลาานกว่านั้นได้



รูป 2.23 GPS (2)

2.4 Microcontroller

2.4.1 Microcontroller คืออะไร ?

Microcontroller มาจากคำ 2 คำ คือ Micro หมายถึงขนาดเล็ก และ Controller หมายถึงตัวควบคุม หรืออุปกรณ์ควบคุม ดังนั้น Microcontroller จึงหมายถึงอุปกรณ์ควบคุมขนาดเล็ก แต่ในตัวอุปกรณ์ควบคุมขนาดเล็กที่มีความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์

ลักษณะของ Microcontroller คืออุปกรณ์สารกึ่งตัวนำ (Semi - Conductor) ที่เป็นอุปกรณ์อิเล็กทรอนิกส์มีโครงสร้างภายนอกเป็นวงจรรวม (Integrate Circuit : IC) ส่วนโครงสร้างภายในคล้ายคลึงกับ Microcomputer ประกอบด้วยหน่วยประมวลผล (Processor unit) หน่วยความจำ (Memory unit) และหน่วยเชื่อมต่อกับอุปกรณ์ภายนอก (I/O Port unit) มีหลายตระกูลให้เลือกใช้งาน อาทิเช่น MCS-51, PIC, AVR, ARM, Basic Stamp, PSoC และ Arduino เป็นต้น โดยจะสามารถออกแบบ Software ควบคุมด้วยภาษา Assembly, Basic, C หรือภาษาอื่น ๆ ที่บริษัทผู้ผลิต Microcontroller แต่ละตระกูลได้พัฒนาขึ้นเพื่อใช้งานร่วมกันเพื่อควบคุมการทำงาน โดยการเขียนข้อมูลลงไปหน่วยความจำ

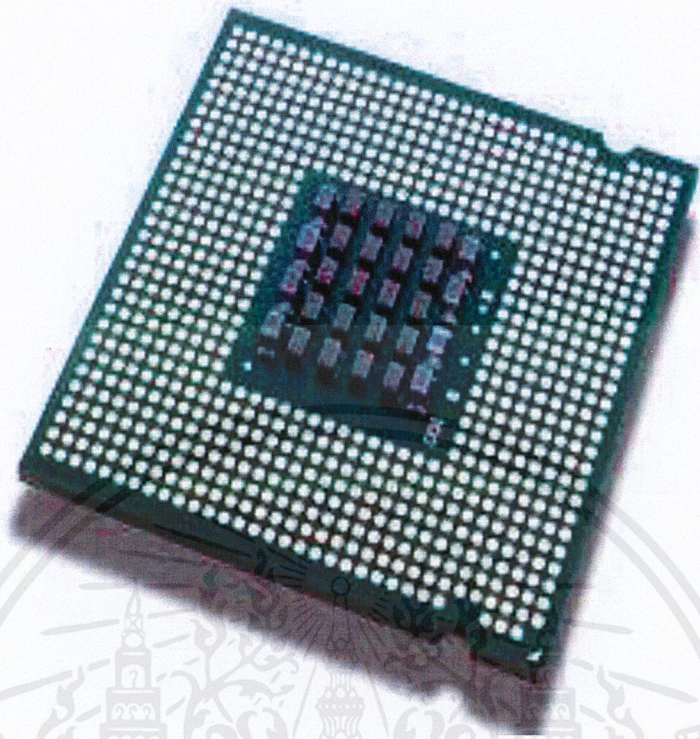


รูป 2.24 Microcontroller

2.4.2 ความแตกต่างระหว่าง Microcontroller กับ Microprocessor

Microprocessor ที่ใช้อยู่ในปัจจุบันจะไม่มีหน่วยความจำชั่วคราว (RAM) หน่วยความจำถาวร (ROM) และ Port เชื่อมต่อภายนอกอยู่ในตัว (I/O Port unit) ทำให้ต้องต่อหน่วยความจำภายนอกเพิ่มเพื่อเก็บข้อมูล Software และต้องใช้ ICs ขยาย Port เชื่อมต่อภายนอกเพิ่มเติม สามารถเพิ่มหน่วยความจำได้ตลอด ส่วน Microcontroller จะมีวงจรรีเลย์ทริกส์พื้นฐานประกอบอยู่ภายใน Chip เช่น หน่วยความจำชั่วคราว หน่วยความจำถาวร และ Port เชื่อมต่อภายนอกอยู่ในตัว ดังนั้นในระบบ Microcontroller จึงมีขนาดเล็กกว่า และยังมีราคาต่ำกว่าระบบ Microprocessor

1. Microprocessor ประกอบด้วยชิปซิลิกอนที่มีหน่วยคำนวณเชิงตรรกะ (ALU), หน่วยควบคุม (CU) และหน่วย Register ในทางกลับกัน ไมโครคอนโทรลเลอร์จะรวมคุณสมบัติของไมโครโพรเซสเซอร์ พร้อมกับหน่วยความจำชั่วคราว หน่วยความจำถาวร พอร์ตเชื่อมต่อภายนอก และอื่น ๆ
2. Microprocessor ต้องการกลุ่มของ Chip อื่น ๆ มาต่อขยายร่วมด้วย เช่น ตัวจับเวลาตัวควบคุมการซัดจังหวะ (Clock) Software ควบคุม และหน่วยความจำ (Memory) Microcontroller ไม่ต้องการ Hardware อื่น ๆ
3. Port การเชื่อมต่อภายนอก Microcontroller มีในตัว ในขณะที่ Microprocessor ไม่มี Port การเชื่อมต่อภายนอกในตัว
4. Microprocessor เน้นในเรื่องประสิทธิภาพการทำงาน ดังนั้นจึงมีจุดมุ่งหมายสำหรับตลาดระดับสูง (Hi - End) ส่วนทางด้าน Microcontroller ตั้งเป้าไปที่ตลาดระบบฝังตัว (Embedded) ทำงานโดยเฉพาะที่เขียน Software ควบคุมกำหนดไว้
5. Microcontroller ประหยัดพลังงานมากกว่า Microprocessor



รูป 2.25 Microprocessor

2.4.3 องค์ประกอบโดยทั่วไปของ Microcontroller

1. หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)

หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit) ทำหน้าที่เป็นศูนย์กลางควบคุมการทำงานของระบบทั้งหมด โดยนำข้อมูลจากหน่วยรับข้อมูล (Input port) นำมาทำงานประมวลผลข้อมูลตามคำสั่งของ Software และส่งผลลัพธ์ออกไปแสดงผล

2. หน่วยความจำ (Memory)

หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับ เก็บโปรแกรมหลัก (Program Memory) เสมือนฮาร์ดดิสก์ (Hard disk) ของเครื่องคอมพิวเตอร์ ซึ่งเป็นหน่วยความจำถาวร (ROM) คือข้อมูลใด ๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) หากไม่มีไฟเลี้ยงข้อมูลก็จะหายไปคล้ายกับหน่วยความจำชั่วคราว

(RAM) ในเครื่องคอมพิวเตอร์ทั่ว ๆ ไป แต่สำหรับ Microcontroller สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำชั่วคราว (RAM) ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และมีอีพรอม (EEPROM : Electrically Erasable Programmable Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port)

ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะใช้การกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณ หรือบัส (BUS)

ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่างหน่วยประมวลผล หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณจำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

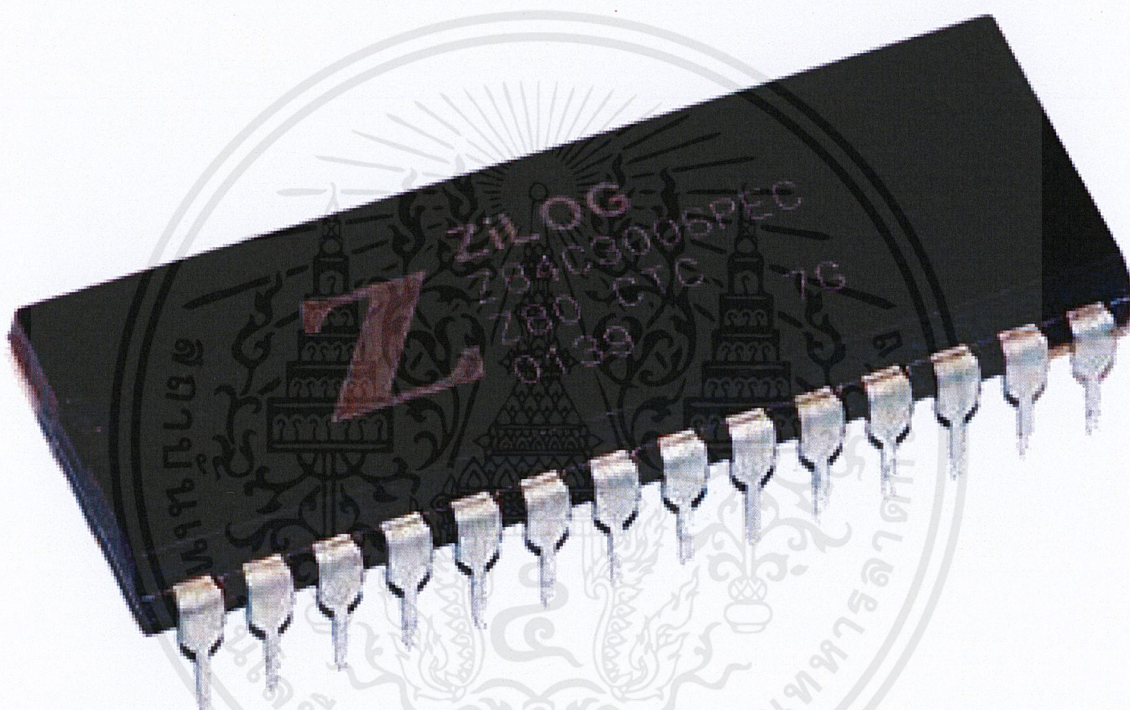
5. วงจรกำเนิดสัญญาณนาฬิกา (Clock)

วงจรกำเนิดสัญญาณนาฬิกา (Clock) เป็นองค์ประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับกำหนัดจังหวะ หากสัญญาณนาฬิกาที่มีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วย

2.4.4 ตัวอย่างประเภท และการใช้งาน Microcontroller

1. Z80

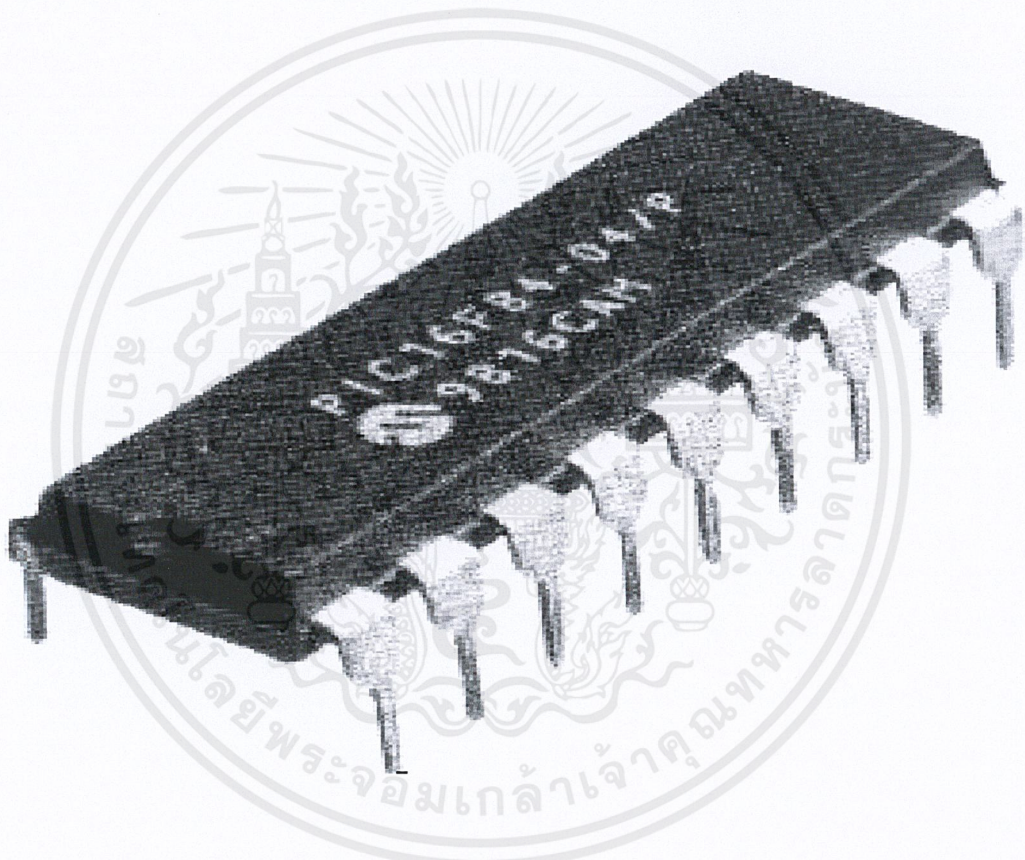
Z80 เป็น Microcontroller ที่นิยมใช้กัน เริ่มตั้งแต่ตัวแรกที่เป็นลักษณะของหน่วยประมวลผล (Processing unit : CPU) ไม่ถึงขั้นเรียกว่า Microcontroller คือ ทรานซิสเตอร์ Z80 เป็นลักษณะของหน่วยประมวลผลเล็ก ๆ ที่ต้องอาศัย Port การเชื่อมต่อภายนอก (I/O Port unit) ต่าง ๆ เพิ่มเติมเข้ามามาก จึงทำให้บอร์ดมีขนาดค่อนข้างใหญ่ จัดได้ว่าเป็นการเริ่มต้นของยุคสมัยนั้น



รูป 2.27 Z80

2. PIC

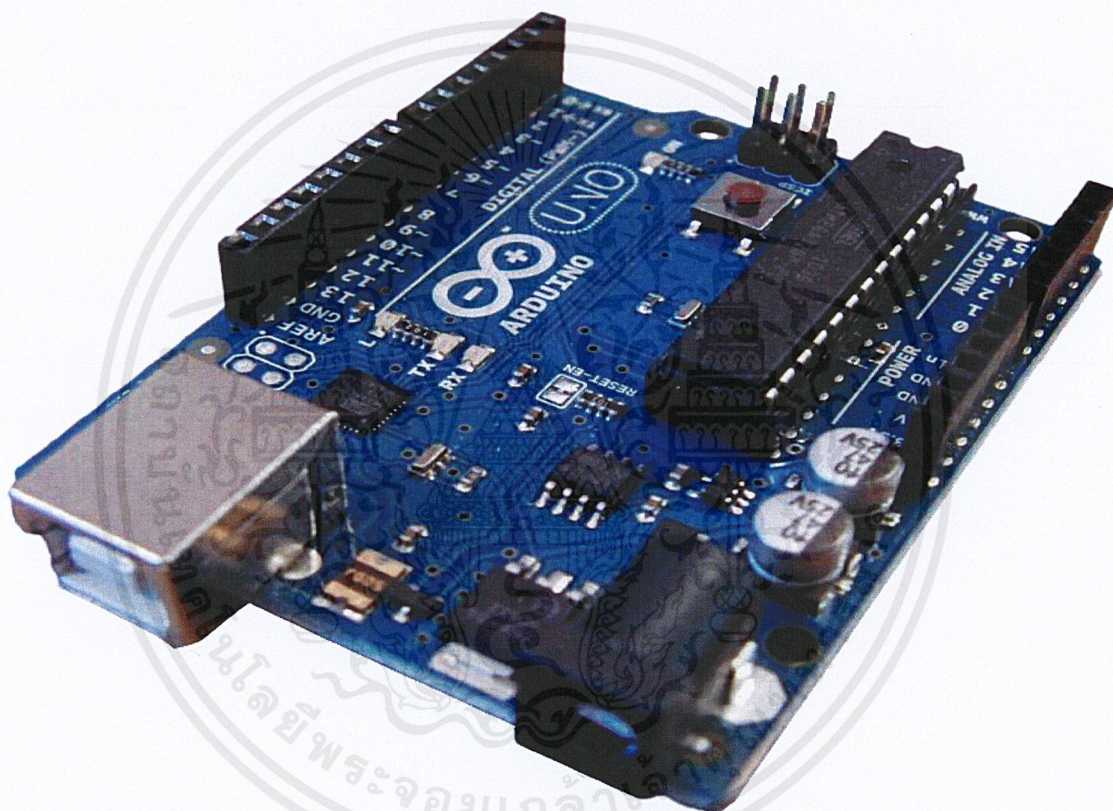
PIC เป็น Microcontroller และผลิตออกมา PIC เป็น Microcontroller ยุคต่อมาที่ได้รับความนิยมสูงอีกตระกูลหนึ่งตั้งแต่อดีตจนถึงปัจจุบัน คำว่า PIC ย่อมาจากคำว่า (Peripheral Interface Controller) Microcontroller ตระกูลนี้มีการพัฒนาเทคโนโลยีขึ้นในทุกด้าน ทำให้ได้รับความนิยมกว่าเป็น Microcontroller ยุคเก่า เพราะในเรื่องของอุปกรณ์ต่อพ่วง (Module) ในยุคเก่าที่มีน้อย ประกอบกับมีหน่วยความจำ EEPROM ในตัว จึงทำให้ง่ายต่อการบันทึกและจัดเก็บข้อมูล และอุปกรณ์ต่อพ่วงต่าง ๆ ได้มีการ latch ในตัว IC อยู่แล้ว จึงสามารถต่อออกมาใช้งานภายนอกได้โดยตรง



รูป 2.28 PIC

3. Arduino

Arduino เป็น Microcontroller ซึ่งถูกสร้างมาจากหน่วยควบคุมตระกูล ARM ของ ATMEL มีข้อดีคือการเป็น Open Source ที่สามารถนำไปพัฒนาต่อเป็นอุปกรณ์ หรือระบบต่าง ๆ ได้ จึงทำให้การพัฒนา Firmware ทำได้ง่ายขึ้น และยังมีการพัฒนา Software ที่ใช้ในการควบคุมตัวบอร์ด ของ Arduino มีลักษณะเป็นภาษา C++ ที่โปรแกรมเมอร์มีความคุ้นเคยในการใช้งาน ตัวบอร์ดสามารถนำอุปกรณ์ต่อพ่วงมาต่อเพิ่มได้อีกมากมาย



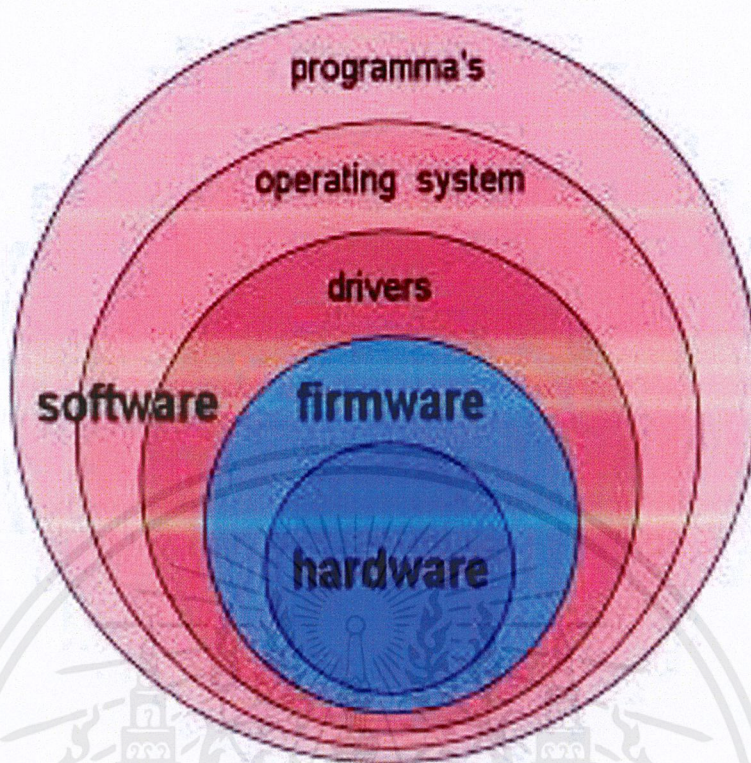
รูป 2.29 Arduino

2.5 การออกแบบ และพัฒนา Firmware

2.5.1 Firmware คืออะไร ?

Firmware เป็น Software พื้นฐานของอุปกรณ์ต่าง ๆ เปรียบเสมือนกับระบบปฏิบัติการ Windows บนเครื่องคอมพิวเตอร์ Firmware เป็น Software ที่เขียนขึ้นมาเพื่อควบคุมการทำงานของอุปกรณ์อิเล็กทรอนิกส์ อาทิเช่น โทรศัพท์มือถือ กล้องดิจิทัล เครื่องเล่นดีวีดี กล้องดิจิทัล เครื่องพิมพ์ และ Microcontroller เป็นต้น เพื่อควบคุมการทำงานคำสั่งต่าง ๆ ซึ่งจะควบคุมการทำงานโดย Chip ซึ่งใน Chip นั้นจะมี Software ที่ควบคุมการทำงานใช้เก็บค่าคำสั่งต่าง ๆ Firmware ถูกออกแบบมาให้ทำงานได้กับอุปกรณ์ที่มีหน่วยความจำอ่านได้อย่างเดียว และเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยงระบบ (Non-volatile memory) อาทิเช่น ROM, EPROM, EEPROM เป็นต้น

แม้ Firmware จะฝังมาที่อุปกรณ์ตั้งแต่แรก แต่ก็ยังสามารถเปลี่ยนหรืออัปเดตได้โดยไม่ต้องเปลี่ยน Hardware ด้วยพร้อมกันได้ อย่างไรก็ตาม อุปกรณ์ไอทีส่วนใหญ่ก็ไม่สามารถทำงานได้ตามปกติถ้าไม่มี Firmware รุ่นล่าสุด การอัปเดต Firmware ส่วนใหญ่นั้นมักมีวัตถุประสงค์เพื่อแก้ปัญหาหรือเพิ่มฟีเจอร์ใหม่ รวมทั้งยกระดับความปลอดภัยของอุปกรณ์ ซึ่งอุปกรณ์ที่เชื่อมต่ออินเทอร์เน็ตได้บางตัวสามารถตรวจสอบ Firmware ใหม่ได้เป็นประจำเพื่อดาวน์โหลด และติดตั้งเองแบบอัตโนมัติ ขณะที่อุปกรณ์อื่นนั้นผู้ผลิตต้องการให้ผู้ใช้เยี่ยมชมเว็บไซต์ของผู้ผลิตด้วยตัวเอง เพื่อดาวน์โหลดตัวอัปเดต Firmware มาติดตั้งบนอุปกรณ์ด้วยตัวเองเช่นกัน

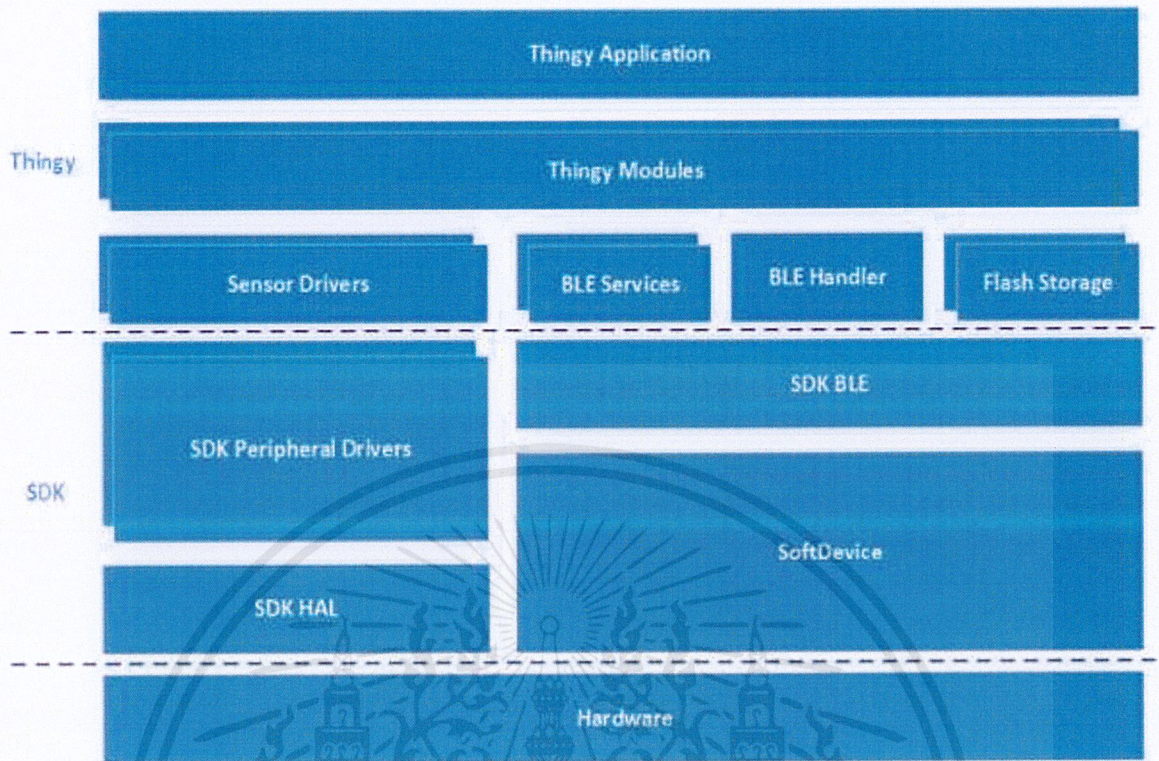


รูป 2.30 Firmware

2.5.2 ความแตกต่างระหว่าง Firmware กับ Software

เราสามารถจัดว่า Firmware เป็น Software ประเภทหนึ่ง มากกว่าที่จะคิดว่าแยกส่วน ไม่เกี่ยวข้องกัน แต่เวลาเราพูดถึง Software นั้น มักใช้จำกัดความถึงแอปพลิเคชันที่มีขนาดใหญ่ประมาณหนึ่ง และมีประโยชน์ใช้งานค่อนข้างกว้างกว่า โดยขนาดไฟล์มีตั้งแต่หลายร้อยกิโลไบต์ไปจนถึงหลายกิกะไบต์ ขณะที่ Firmware นั้นมักมีขนาดค่อนข้างเล็กประมาณไม่กี่กิโลไบต์

Software นั้นข้อมูลจะถูกจัดเก็บในหน่วยความจำ (Memory) หรือพื้นที่ที่ผู้ใช้สามารถเข้าถึง และแก้ไขข้อมูลได้ง่าย แต่ Firmware มักถูกเขียนอยู่ในหน่วยความจำ (Memory) ที่ค่อนข้างถาวรหรือปรับเปลี่ยนได้ยาก ด้วยเจตนาที่ต้องการให้อุปกรณ์สามารถทำงานได้อย่างต่อเนื่อง โดยไม่เสี่ยงต่อการถูกรบกวนจากภายนอก เมื่อก่อนนี้นิยมใส่เฟิร์มแวร์ลงใน Chip ที่เรียกว่า EPROM แต่ปัจจุบันนิยมหันมาใช้ EEPROM ที่เปิดโอกาสให้ผู้ใช้แก้ไขหรืออัปเดตได้มากกว่า



รูป 2.31 Software and Firmware

2.5.3 ขั้นตอนการพัฒนา Firmware

1. กำหนดเป้าหมายที่ต้องการ และข้อกำหนดที่ชัดเจนในการทำงานแต่ละส่วน
2. แยกระดับชั้นของโครงการ โครงสร้างแต่ละส่วนแยกกันชัดเจน
3. กำหนดระยะเวลาการทำงานที่โครงการที่เหมาะสม คำนวณระยะเวลาเพื่อสำหรับทดสอบ และแก้ไขระบบโครงการ อพเทคการทำงานทันทีเมื่อมีการเปลี่ยนแปลงรายละเอียด
4. ออกแบบตัวทดสอบจำลองเพื่อทดสอบระบบโครงการโดยต้นทุนต่ำ
5. วางแผนสำหรับการเปลี่ยนแปลงรายละเอียด ออกแบบและพัฒนารองรับการอพเทคในอนาคต

2.6 Protocol การสื่อสารข้อมูล

2.6.1 โพรโตคอล (Protocol) คืออะไร ?

Protocol คือ ข้อกำหนดหรือข้อตกลงในการสื่อสารระหว่างคอมพิวเตอร์ หรืออุปกรณ์ที่ทำงานในรูปแบบที่ใกล้เคียงกับคอมพิวเตอร์ เช่น Microcontroller ซึ่งทำให้สามารถเกิดการสื่อสารระหว่างสองฝ่ายได้ คล้าย ๆ ภาษามนุษย์ที่ ถ้าจะสื่อสารกันได้ก็จะต้องใช้ภาษาเดียวกัน

Protocol จะเป็นตัวกำหนดลักษณะ หรือองค์ประกอบต่าง ๆ ที่ใช้ในการสื่อสาร ไม่ว่าจะเป็นรูปแบบการแทนข้อมูล วิธีการในการรับส่งข้อมูล รูปแบบสัญญาณในสื่อสาร อุปกรณ์ที่สามารถใช้ใน ระบบ สื่อกลางในการส่งข้อมูล การอ้างอิงตำแหน่งข้อมูล การตรวจสอบความผิดพลาดของข้อมูล รวมถึง ความเร็วในการสื่อสาร

ในบางกรณีถ้าอุปกรณ์สองชิ้น มีการใช้โปรโตคอลที่ต่างกันและต้องการนำมาเชื่อมต่อกัน จะต้องมีตัวกลางในการแปลง Protocol กลับไปกลับมาซึ่งนิยมเรียกว่า Gateway ถ้าเทียบกับภาษามนุษย์ ก็คือล่าม มีรูปแบบทั้งที่เป็นเครื่อง Server แยกต่างหากสำหรับทำหน้าที่นี้โดยเฉพาะ หรืออาจจะเป็น Software หรือ Driver ที่สามารถติดตั้งในเครื่องอุปกรณ์นั้น ๆ ได้เลย

2.6.2 องค์ประกอบของ Protocol

1. Syntax

Syntax หมายถึง Format หรือ Structure ของข้อมูล เช่น กำหนดว่าใน 8 Bit แรกจะหมายถึง Address ของผู้ส่งอีก 8 Bit ถัดมา หมายถึง Address ของผู้รับ ส่วนที่เหลือจึงจะเป็นข้อมูลจริง ๆ ถ้าไม่มีการกำหนด Syntax แล้ว Entity จะไม่สามารถทราบได้เลยว่าได้รับอะไรมา

2. Semantics

Semantics หมายถึง ความหมายของข้อมูลที่ได้รับมา เช่น เมื่อได้รับข้อมูลแล้ว Entity รู้ Syntax แล้ว แต่จะยังไม่รู้ว่าบิตแต่ละบิตนั้นทำอะไรได้บ้าง ดังนั้นจึงต้องมาทำการแปลความหมายของข้อความ เหล่านั้นเสียก่อน เช่น เมื่อทราบแอดเดรสของผู้รับแล้ว Entity จะสามารถทำการหาเส้นทางการสื่อสารได้

3. Timing

Timing เป็นข้อกำหนดของเวลาในการรับส่งข้อมูล ใช้สำหรับกำหนดลำดับ เนื่องจาก Entity แต่ละตัวนั้นมาด้วยความเร็วในการรับส่งที่ไม่เท่ากัน

2.6.3 ตัวอย่าง Protocol ที่ใช้กับอุปกรณ์ Microcontroller

1. TCP / IP Protocol

TCP / IP Protocol การที่เครื่องคอมพิวเตอร์ที่ถูกเชื่อมโยงกันไว้ในระบบ จะสามารถติดต่อสื่อสารกันได้นั้น จำเป็นจะต้องมีภาษาสื่อสารที่เรียกว่า Protocol ซึ่งในระบบอินเทอร์เน็ตจะใช้ภาษาสื่อสารมาตรฐานที่ชื่อว่า TCP/IP เป็นภาษาหลัก ดังนั้นหากเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นเครื่องระดับ Microcomputer Minicomputer หรือ Mainframe computer ก็สามารถเชื่อมโยงเข้าสู่อินเทอร์เน็ตได้

TCP ย่อมาจากคำว่า Transmission Control Protocol

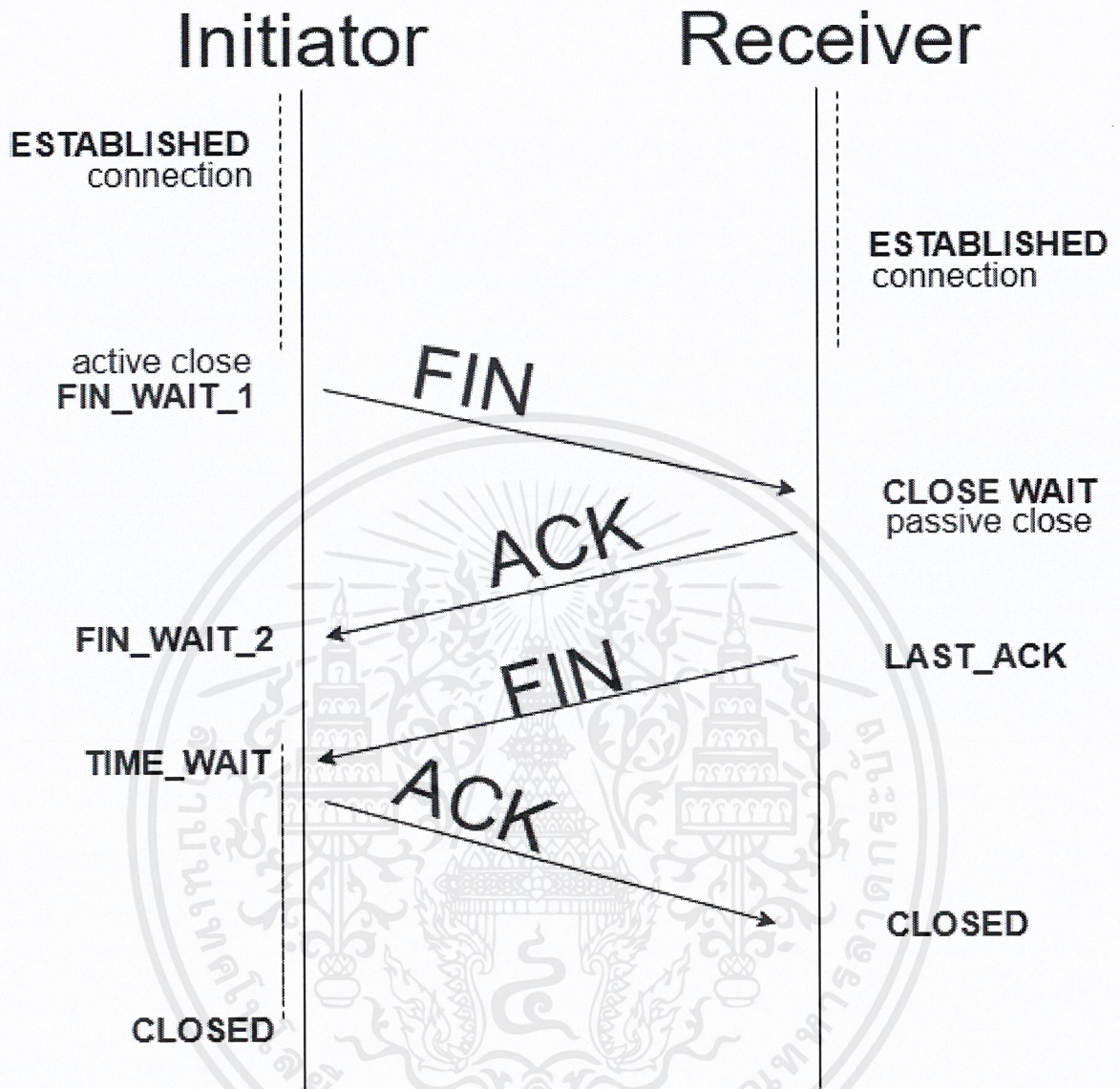
IP ย่อมาจากคำว่า Internet Protocol

การทำงานนั้นแบ่งออกได้เป็น 3 รูปแบบ

Establish : คือการสร้าง Connection

Transfer : คือการแลกเปลี่ยนข้อมูล

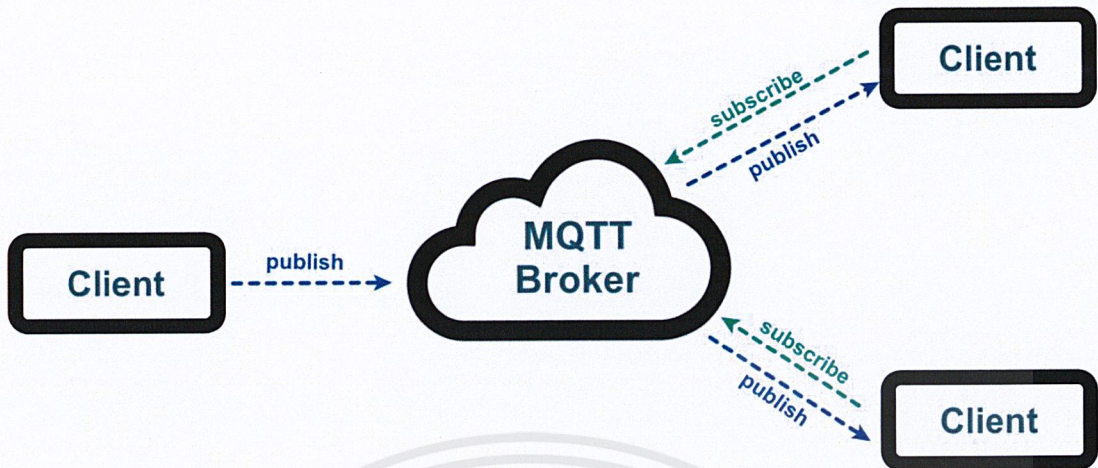
Terminate : ปิด Connection



รูป 2.32 TCP / IP Protocol

2. MQTT Protocol

Message Queuing Telemetry Transport Protocol (MQTT Protocol) ถูกออกแบบมาเพื่อการเชื่อมต่อแบบ M2M (Machine - to - Machine) คือการที่อุปกรณ์เชื่อมต่อกับอุปกรณ์ สนับสนุนเทคโนโลยี IoT (Internet of Things) คือเทคโนโลยีที่อินเทอร์เน็ตเชื่อมต่อกับอุปกรณ์ต่าง ๆ เช่น โทรศัพท์มือถือ รถยนต์ โทรทัศน์ ตู้เย็น เข้ากับอินเทอร์เน็ตทำให้สามารถเชื่อมโยงสื่อสารกับอุปกรณ์ต่าง ๆ ได้ โดยผ่านเครือข่ายอินเทอร์เน็ต ซึ่งจะทำให้มนุษย์สามารถ ควบคุมอุปกรณ์ต่าง ๆ จากที่อื่นได้ เช่น การสั่งปิดเปิดไฟในบ้านจากที่อื่น ๆ



รูป 2.33 MQTT Protocol Detail

องค์ประกอบของ MQTT Protocol

1. Client

Client หมายถึง Publisher หรือ Subscriber ที่เชื่อมต่อแบบรวมศูนย์ไปยัง Broker ซึ่งสามารถเชื่อมต่อได้ทั้งแบบ Persistent ที่ทำการสร้าง Session ค้างไว้เปิดตลอดเวลาเพื่อติดต่อกับ Broker ซึ่งตรงกันข้ามกับ Client ที่เชื่อมต่อแบบ transient ซึ่ง Broker ไม่สามารถติดตามสถานะได้

2. Publisher

ทำหน้าที่ในการส่งข้อมูลออกไปสู่ระบบตาม Topic ที่กำหนด

3. Subscriber

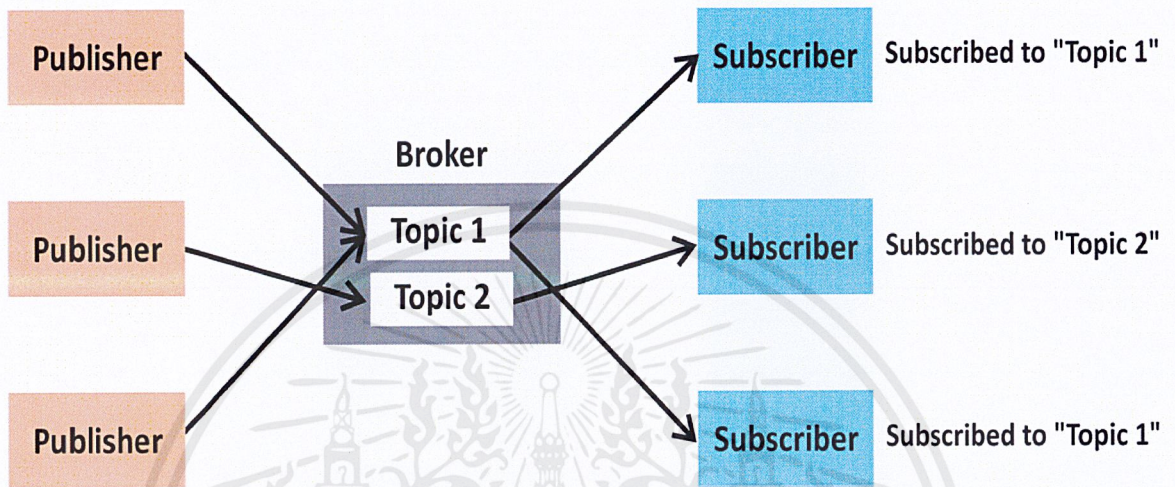
Subscriber จำทำหน้าที่คอยดูการเปลี่ยนแปลงของ Message ที่อ้างอิงด้วย Topic เช่นถ้ามีหัวข้อที่น่าสนใจ และมีการเปลี่ยนแปลงก็จะทำการดึงข้อมูลนั้น ๆ มาใช้งาน

4. Broker

Broker เป็น Software ที่ทำหน้าที่รับข้อความทั้งหมดที่ได้จาก Publisher แล้วจึงส่งต่อไปให้ Subscriber ตามแต่ Topic ที่ client ได้ทำการ Subscribe ไว้

5. Topic

Topic เป็นเหมือน Address หรือ Endpoint บน Broker ที่ Client ทำการเชื่อมต่อเพื่อรับส่งข้อความระหว่างกันนั่นเองค่ะ



รูป 2.34 MQTT Protocol (2)

ข้อดีคือ MQTT คือเหมาะกับการนำไปใช้กับระบบ Cloud ที่ให้บริการแบบรวมศูนย์ เพราะถูกออกแบบให้เหมาะกับการกระจายข้อมูลแบบ Many - to - Many ตัวอย่างแอปพลิเคชันที่นำ MQTT ไปใช้อย่างแพร่หลายคงจะหนีไม่พ้น IoT Platform ที่มีอยู่ในท้องตลาดมากมาย แต่ก่อนหน้านี้ MQTT ก็ได้ถูกนำไปใช้กับ Facebook Messenger แล้วด้วยเหตุนี้เองจึงทำให้เป็นตัวเลือกยอดนิยมในการให้บริการโซลูชันด้าน IoT บนระบบ Cloud

บทที่ 3

ขั้นตอนการดำเนินงาน

วิธีการดำเนินงานเบื้องต้น

1. ศึกษาข้อมูลความต้องการที่ได้รับมอบหมาย คิดวิเคราะห์หา Function การใช้งานที่เหมาะสม
2. วางแผนขั้นตอนการดำเนินงาน และกำหนดเวลาการทำงานในแต่ละส่วน
3. ศึกษาเกี่ยวกับเครื่องมือที่ใช้ในการพัฒนาโครงการ จัดหาอุปกรณ์ ทดสอบการใช้งาน และปรับเปลี่ยนแก้ไขตามความเหมาะสม
4. ออกแบบ และพัฒนาระบบโครงการ
5. การใช้งานแอปพลิเคชันฮอป

3.1 ศึกษาข้อมูลความต้องการที่ได้รับมอบหมาย คิดวิเคราะห์หา Function การใช้งานที่เหมาะสม

3.1.1 วิเคราะห์ความต้องการที่ได้รับมอบหมาย

1. สามารถอ่านค่าข้อมูลจากระบบ CAN - BUS Protocol ออกมาได้
2. สามารถวิเคราะห์ข้อมูลทุกข้อมูล หาข้อมูลที่สนใจจากระบบ CAN - BUS Protocol ออกมาได้
3. ส่งข้อมูล CAN - BUS Protocol ออกมาใช้ประโยชน์ได้
4. พัฒนา Firmware รุ่นใหม่ ใหม่ให้กับระบบควบคุมยานพาหนะ และการสื่อสารข้อมูลได้
5. แก้ไขปรับปรุงข้อผิดพลาดจาก Firmware รุ่นเก่า
6. เพิ่มเติมฟังก์ชันที่เกี่ยวกับ ระบบ CAN - BUS Protocol เข้าไปในระบบควบคุมยานพาหนะ และการสื่อสารข้อมูลได้
7. ออกแบบ Protocol การสื่อสารส่งข้อมูลรุ่นใหม่

3.1.2 แก้ปัญหาที่เกิดจาก Firmware รุ่นเก่า

1. ลดปัญหาระยะเวลาการรอการตอบสนองของอุปกรณ์ Hardware บนยานพาหนะที่กินเวลานาน
2. สัญญาณรบกวนในระบบ GPS
3. ระยะเวลาการรอรอบของการส่งข้อมูลสถานะของยานาหนะขึ้น Server
4. ความเร็วในการส่งข้อมูล และการส่งข้อมูลเมื่อมีการเกิดการกระทำต่าง ๆ ต่อระบบกินระยะเวลานาน
5. โค้ดของ Firmware รุ่นเก่าไม่เป็นระเบียบ

3.1.3 ออกแบบฟังก์ชันการใช้งานใหม่ และปรับปรุงฟังก์ชันการใช้งานเก่า

1. อ่านข้อมูลยานพาหนะผ่านระบบ CAN – BUS Protocol
2. เลือกข้อมูล CAN – BUS ที่สนใจ และแปลงข้อมูลให้สามารถอ่านได้เข้าใจ
3. เปลี่ยน Protocol การสื่อสาร
4. ลดระยะเวลาการทำงานของระบบโดยรวม
5. จัดระเบียบโค้ดของ Firmware ใหม่

3.2 วางแผนขั้นตอนการดำเนินงาน และกำหนดเวลาการทำงานในแต่ละส่วน

3.2.1 วางแผนลำดับขั้นตอนการดำเนินงาน

1. ทำความเข้าใจระบบ CAN – BUS Protocol
2. วิเคราะห์ข้อมูล CAN – BUS ที่สนใจ และอ่านค่าออกมาใช้งานได้
3. ทำความเข้าใจกับ Firmware รุ่นเก่า ระบบการทำงานเก่า วิเคราะห์ปัญหาและหาทางแก้ไข
4. พัฒนา Firmware รุ่นใหม่
5. พัฒนา Protocol การสื่อสารรุ่นใหม่
6. ประสานการทำงานกับ Hardware และ Software
7. ทดสอบระบบทั้งหมด ตรวจสอบปัญหา และวิเคราะห์ทางแก้ไข
8. ทดสอบการติดตั้งใช้งานจริง
9. สรุปผล และเสนอแนวทางการพัฒนาต่อยอด

3.2.2 กำหนดช่วงเวลาการดำเนินงาน

หัวข้อ	ส.ค.		ก.ย.		ต.ค.		พ.ย.	
	1	2	1	2	1	2	1	2
1. ทำความเข้าใจระบบ CAN – BUS Protocol								
2. วิเคราะห์ข้อมูล CAN – BUS ที่สนใจ และอ่านค่าออกมาใช้งานได้								
3. ทำความเข้าใจกับ Firmware รุ่นเก่า ระบบการทำงานเก่า วิเคราะห์ปัญหาและหาทางแก้ไข								
4. พัฒนา Firmware รุ่นใหม่								
5. พัฒนา Protocol การสื่อสารรุ่นใหม่								
6. ประสานการทำงานกับ Hardware และ Software								
7. ทดสอบระบบทั้งหมด ตรวจสอบปัญหา และวิเคราะห์ทางแก้ไข								
8. ทดสอบการติดตั้งใช้งานจริง								
9. สรุปผล และเสนอแนวทางพัฒนาต่อยอด								

ตาราง 3.1 ตารางแสดงช่วงเวลาดำเนินงาน

3.3 ศึกษาเกี่ยวกับเครื่องมือที่ใช้ในการพัฒนาโครงการ ทดสอบการใช้

3.3.1 Arduino IDE

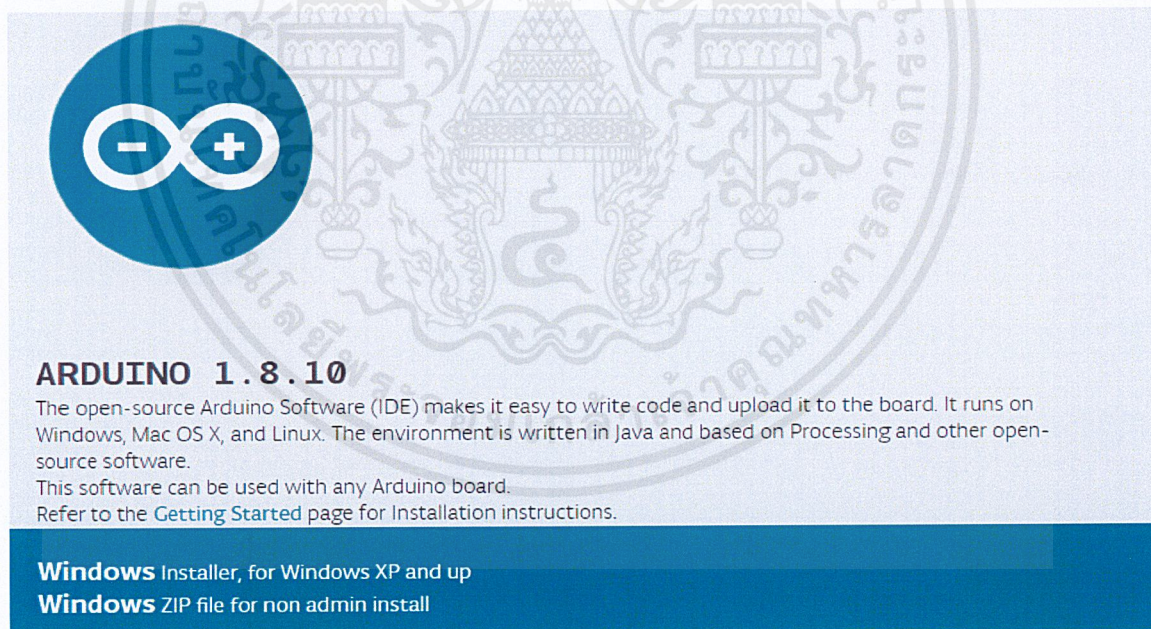
Arduino IDE คือ Software ที่เป็น Compiler ในการเขียนโปรแกรม และ Compile ลงบอร์ด Microcontroller โดยใช้พัฒนา Firmware ด้วย ภาษา C++

IDE (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนา หรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือนักพัฒนา เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่าง ๆ เร็วมากขึ้น

การดาวน์โหลด และติดตั้งโปรแกรม

1. การดาวน์โหลดโปรแกรม

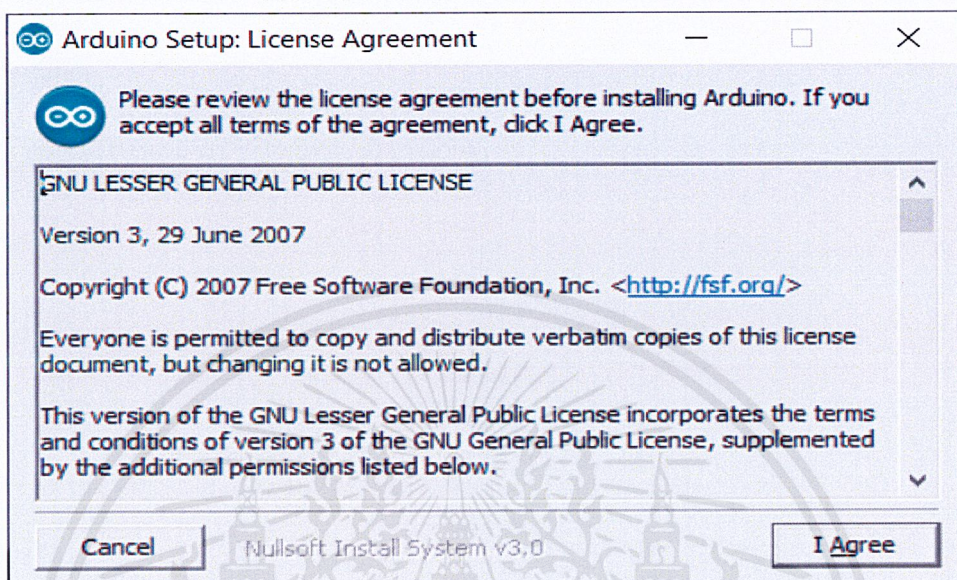
- เชื่อมต่อไปที่ลิงค์ <https://www.arduino.cc/en/Main/Software>
- เลือก Download the Arduino IDE ตามระบบปฏิบัติการที่ใช้



รูป 3.1 Download Arduino IDE (1)

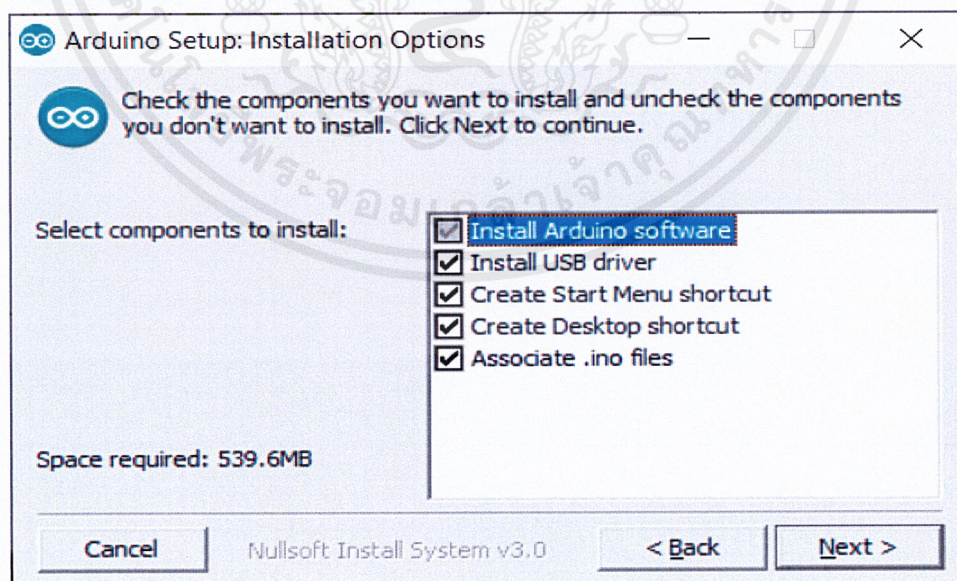
2. การติดตั้งโปรแกรม

- กดยินยอมรับข้อตกลงการใช้งาน



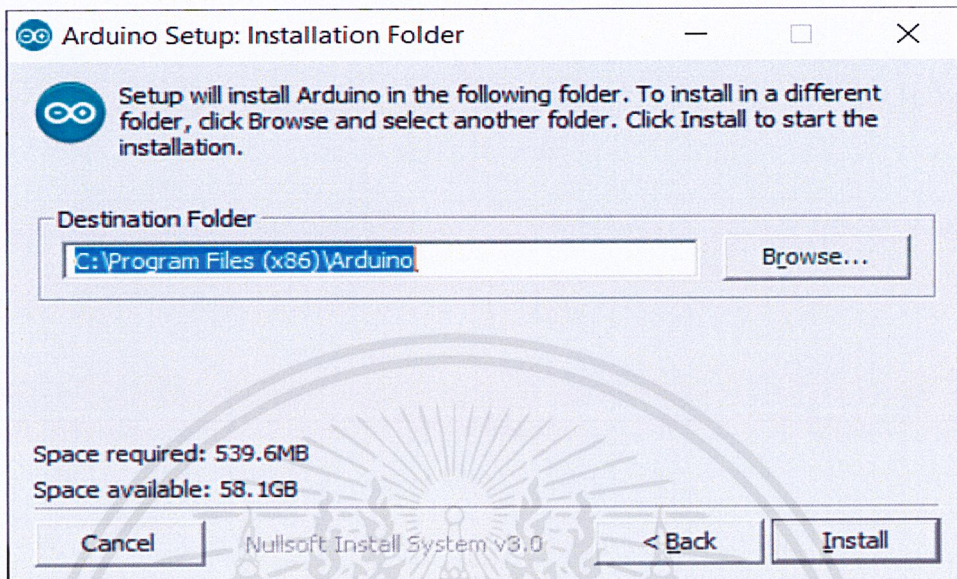
รูป 3.2 Download Arduino IDE (2)

- เลือกจัดการองค์ประกอบการลงโปรแกรม



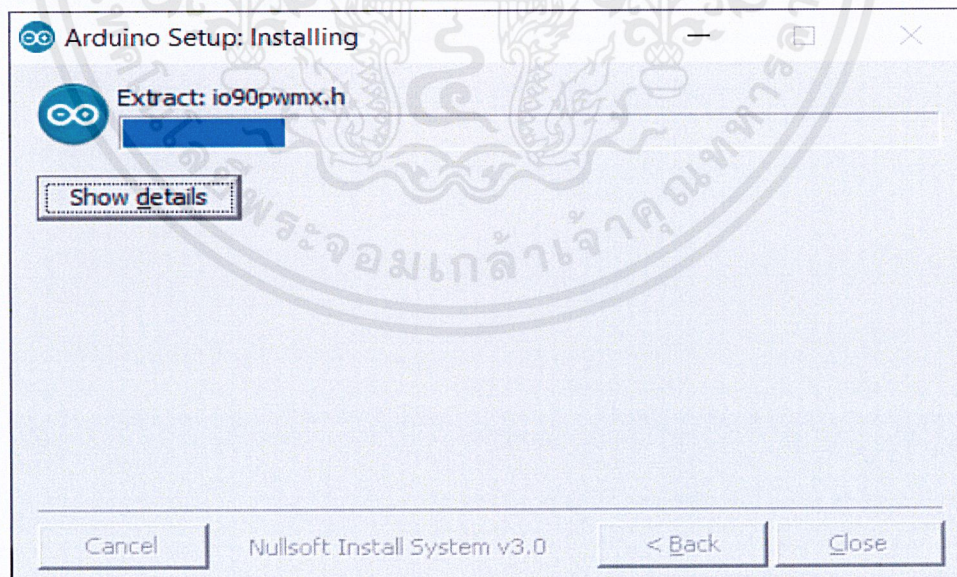
รูป 3.3 Download Arduino IDE (3)

- เลือกตำแหน่งติดตั้งโปรแกรม



รูป 3.4 Download Arduino IDE (4)

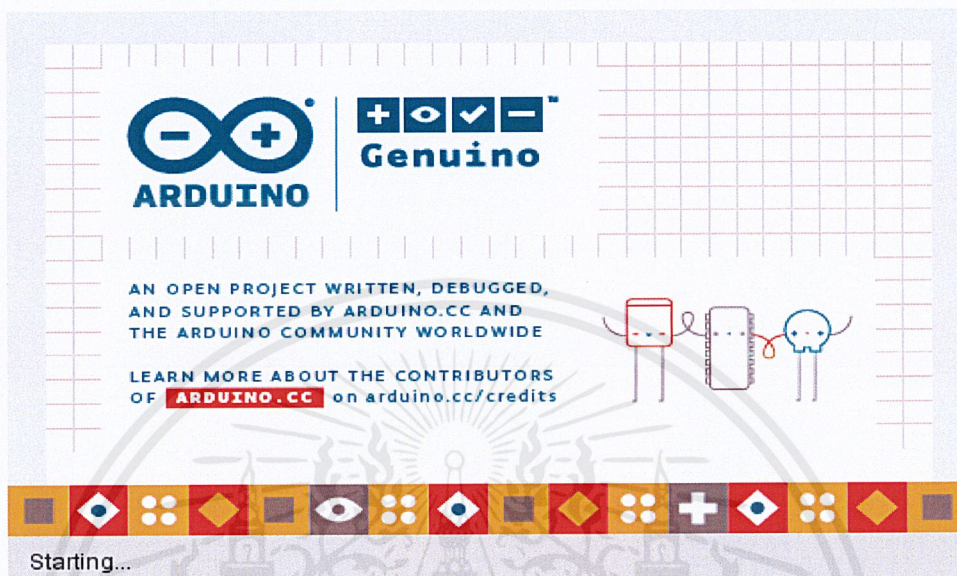
- ติดตั้งโปรแกรมให้สมบูรณ์



รูป 3.5 Download Arduino IDE (5)

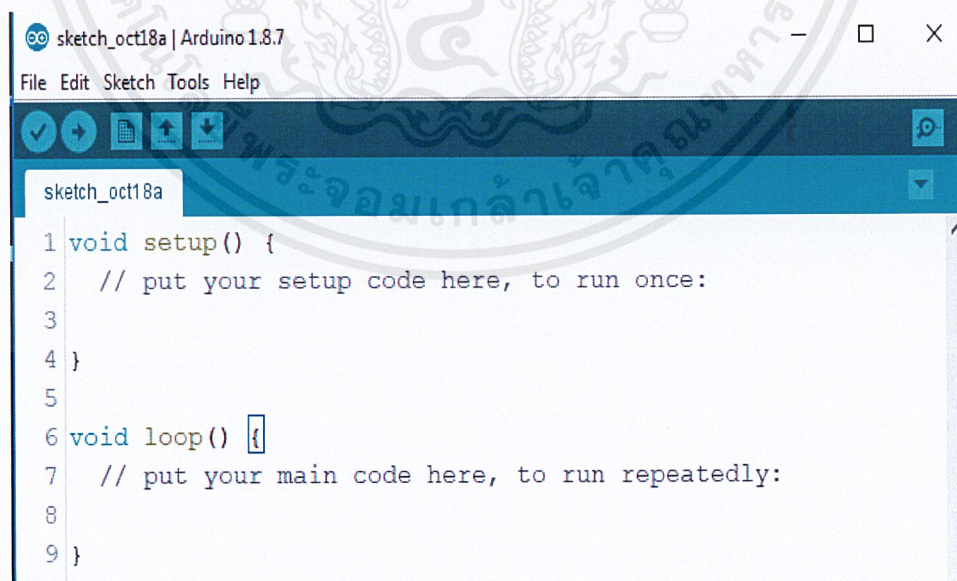
การตั้งค่าแพคเกจ

- เข้าสู่โปรแกรม



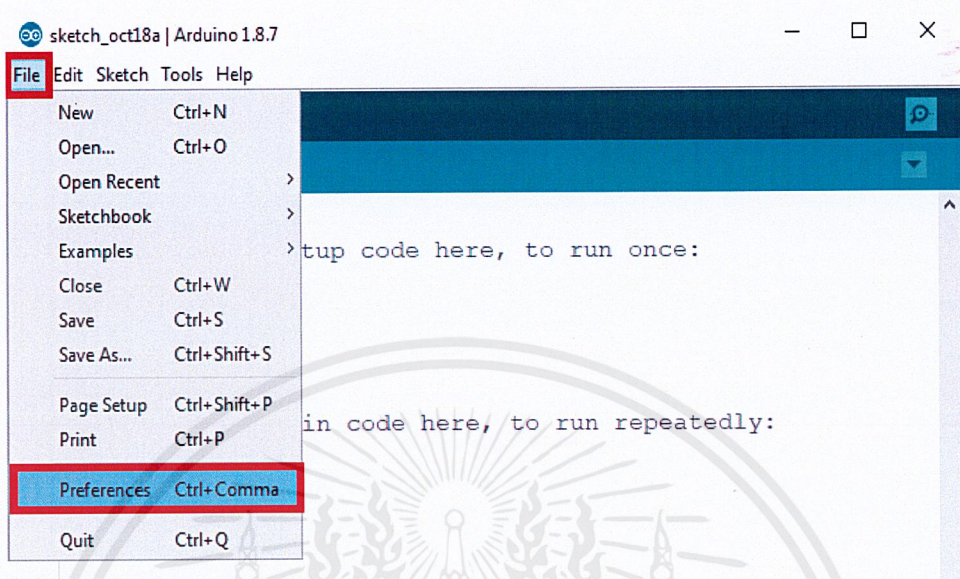
รูป 3.6 Arduino IDE

- อินเทอร์เฟซพื้นฐาน



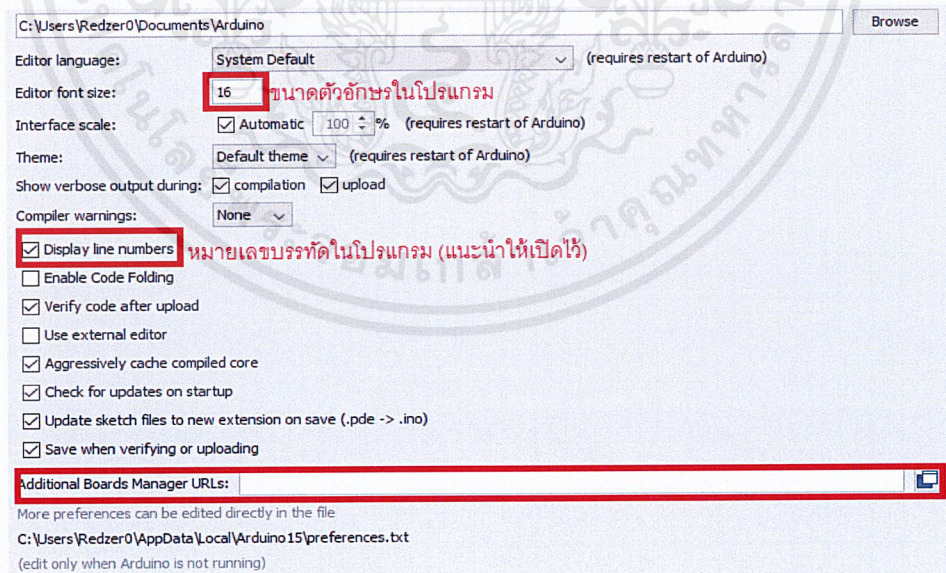
รูป 3.7 Arduino IDE Interface

- เลือก Preference



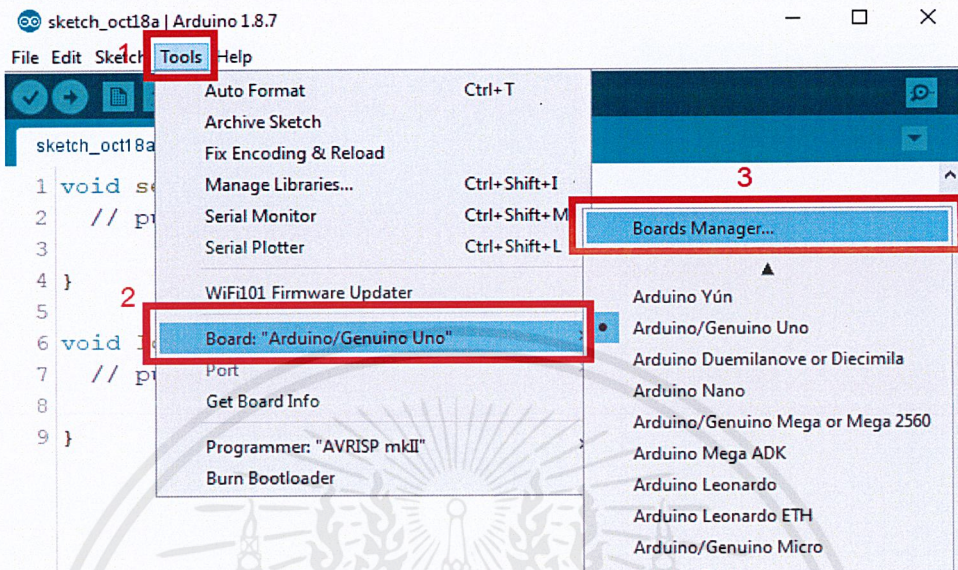
รูป 3.8 Arduino IDE Preference (1)

- Preference (ใส่ลิงค์ดาวน์โหลดในช่อง Additional Board Manager URLs:)



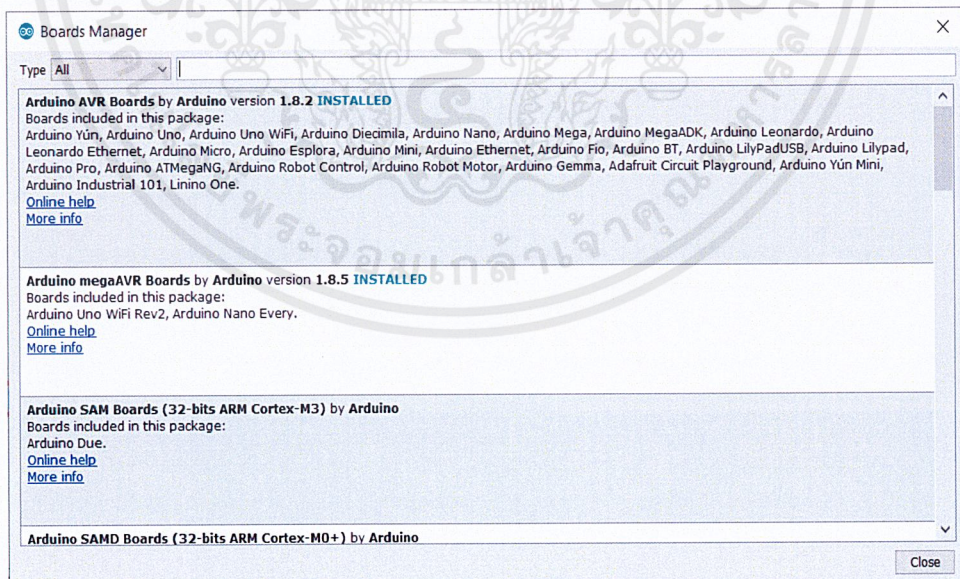
รูป 3.9 Arduino IDE Preference (2)

- เลือก Board Manager



รูป 3.10 Arduino IDE set Board Manager (1)

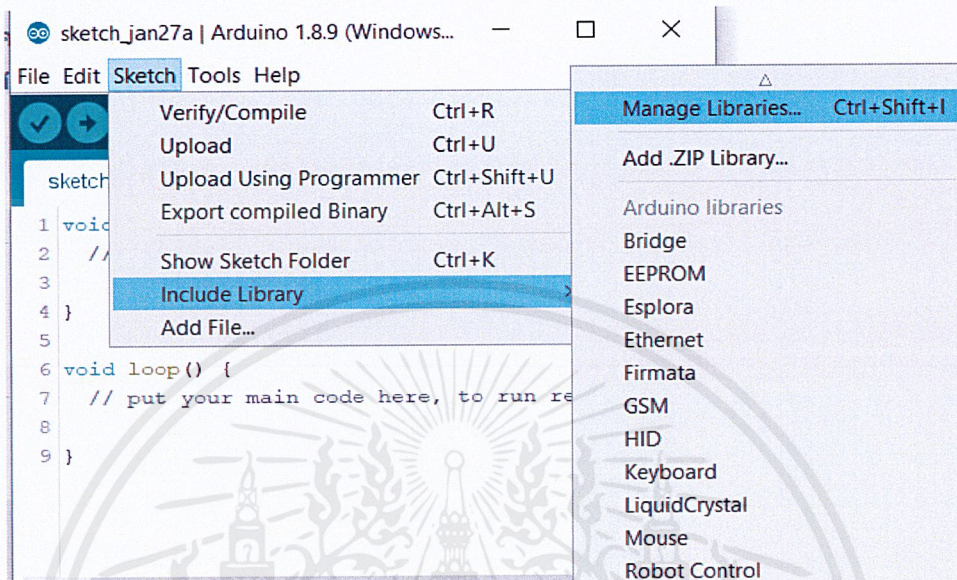
- เลือกบอร์ดที่เราใช้งาน



รูป 3.11 Arduino IDE set Board Manager (2)

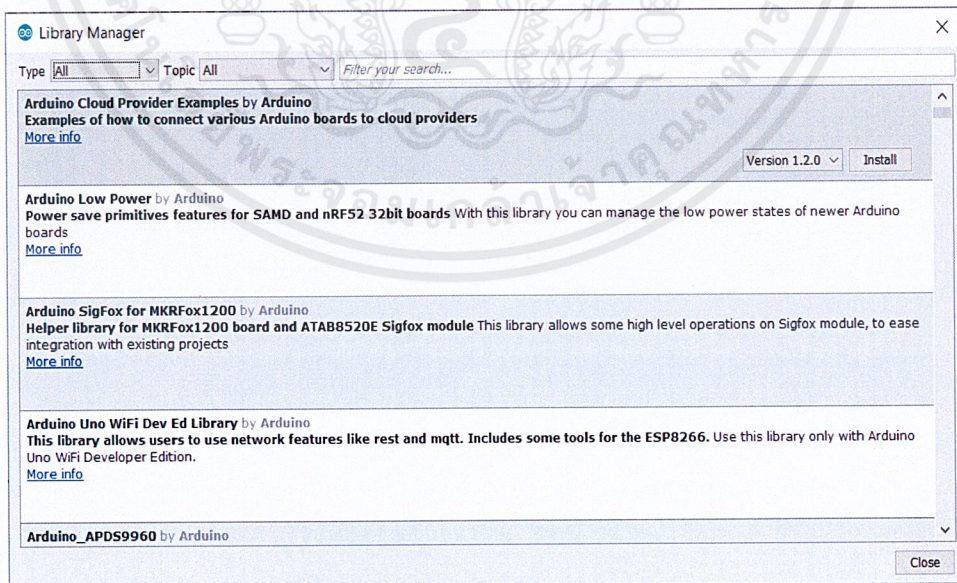
การติดตั้งไลบรารีเสริม

- เลือก Manage Library



รูป 3.12 Arduino IDE set Library (1)

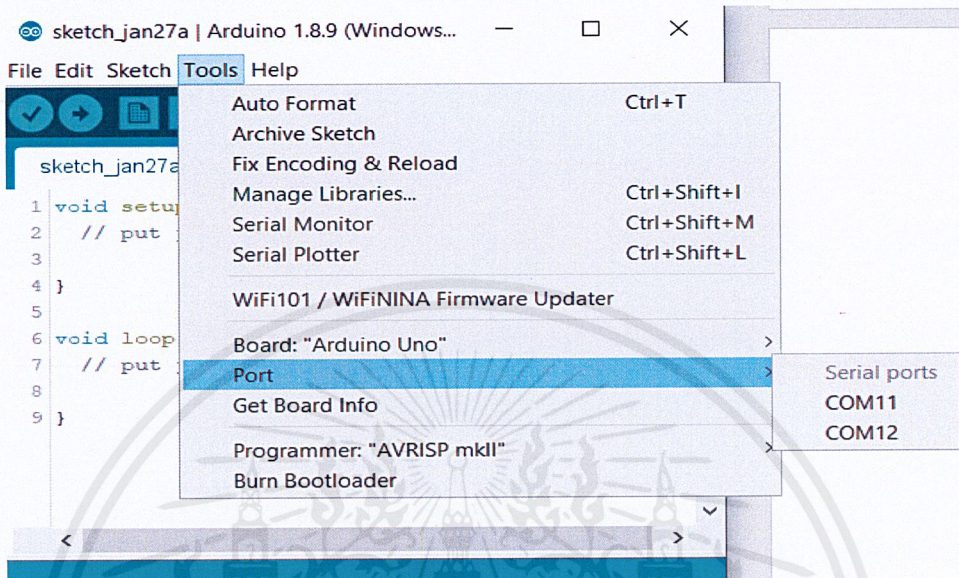
- เลือกไลบรารีที่เราใช้งาน



รูป 3.13 Arduino IDE set Library (2)

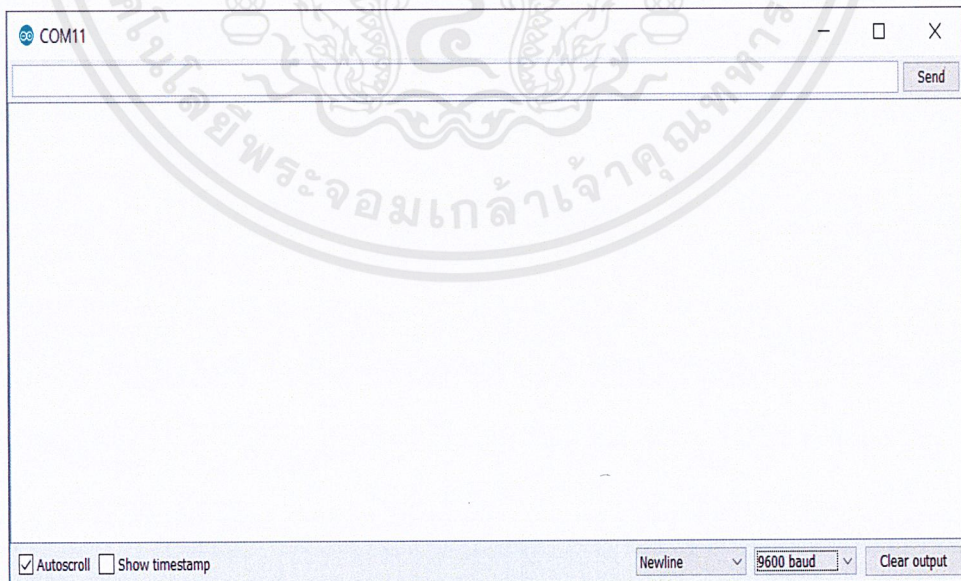
การตั้งค่าพื้นฐานการใช้งานพื้นฐานอื่น ๆ

- เลือกพอร์ตการใช้งาน



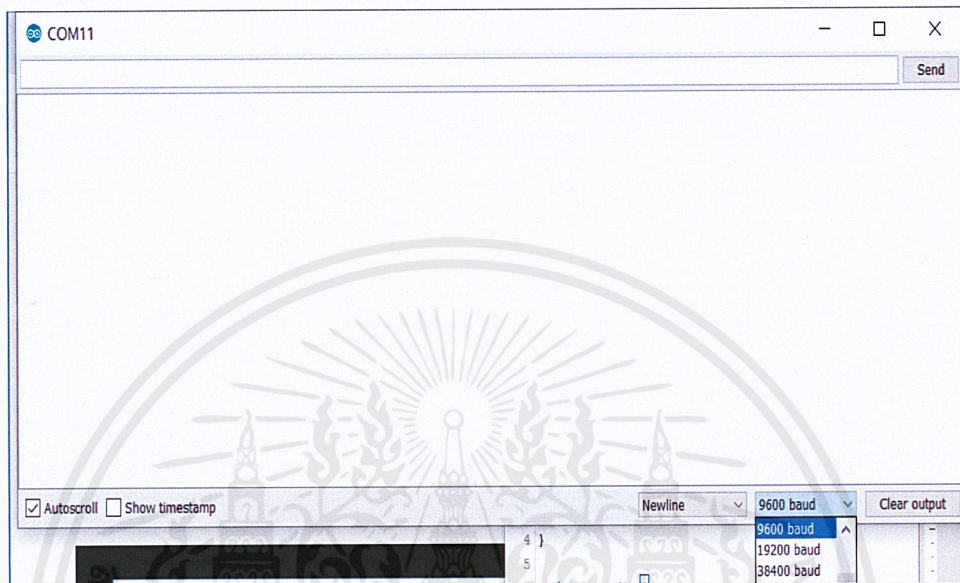
รูป 3.14 Arduino IDE set Port

- Serial Monitor



รูป 3.15 Arduino IDE Serial Monitor

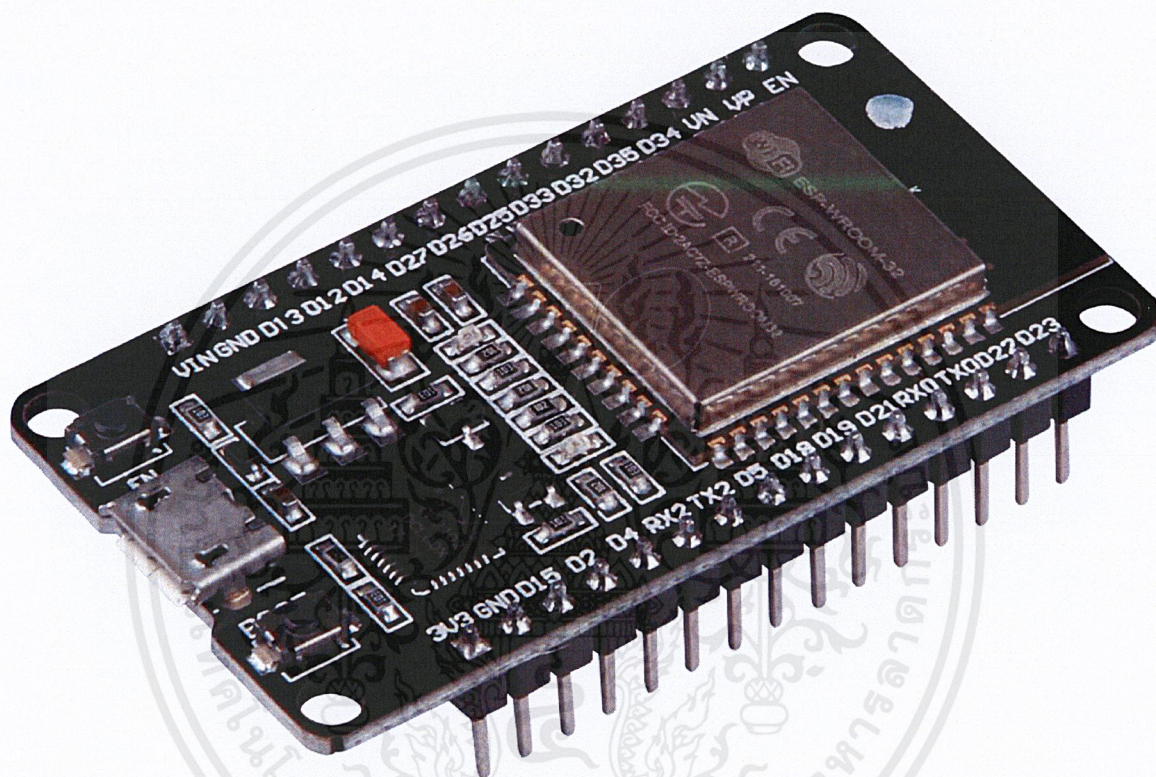
- การเลือก Baud Rate



รูป 3.16 Arduino IDE Serial Monitor set Baud Rate

3.3.2 ESP32

ESP32 เป็นอุปกรณ์วงจรรวม (Integrate Circuit : IC) บนไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ WiFi และ Bluetooth 4.2 รูปแบบ Bluetooth Low Energy (BLE) ในตัว มีองค์ประกอบต่าง ๆ อีกมากมาย รองรับการใช้งานที่หลากหลาย รวมถึงมีไลบรารีที่รองรับเป็นจำนวนมาก พัฒนาโดยบริษัท Espressif จากประเทศจีน



รูป 3.17 ESP32

รายละเอียดสเปกอุปกรณ์

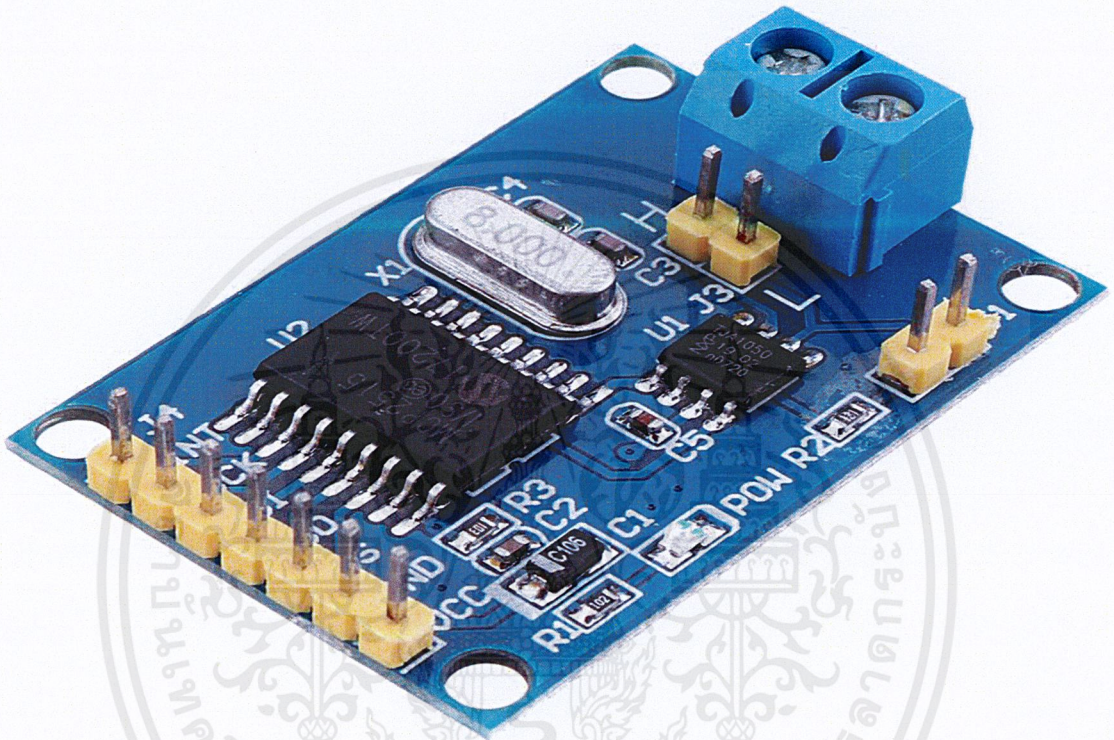
1. หน่วยประมวลผลใช้สถาปัตยกรรม Tensilica LX6 แบบ 2 แกนสมอง สัญญาณนาฬิกา 240MHz
2. หน่วยความจำชั่วคราวภายในตัว 512KB
3. รองรับการเชื่อมต่อหน่วยความจำถาวรภายนอกสูงสุด 16MB
4. มาพร้อมกับ WiFi มาตรฐาน 802.11 b/g/n รองรับการใช้งานทั้งในโหมด Station softAP และ Wi-Fi direct
5. มีบลูทูธในตัว รองรับการใช้งานในโหมด 2.0 และโหมด 4.0 BLE
6. ใช้แรงดันไฟฟ้าในการทำงาน 2.6V ถึง 3V
7. โหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5 uA
8. ทำงานได้ที่อุณหภูมิ - 40°C ถึง 125°C
9. รองรับการเชื่อมต่อกับ Memory Card
10. รองรับการเข้ารหัส WiFi แบบ WEP และ WPA/WPA2 PSK/Enterprise
11. มีวงจรเข้ารหัส AES / SHA2 / Elliptical Curve Cryptography / RSA-4096 ในตัว

อุปกรณ์เสริมติดตั้งภายใน

- วงจรกรองสัญญาณรบกวนในวงจรขยายสัญญาณ
- เซ็นเซอร์แม่เหล็ก
- เซ็นเซอร์สัมผัส (Capacitive Touch) รองรับ 10 ช่อง
- รองรับการเชื่อมต่อคลิสตอล 32.768 Khz สำหรับใช้กับส่วนวงจรนับเวลาโดยเฉพาะ

3.3.3 MCP2515

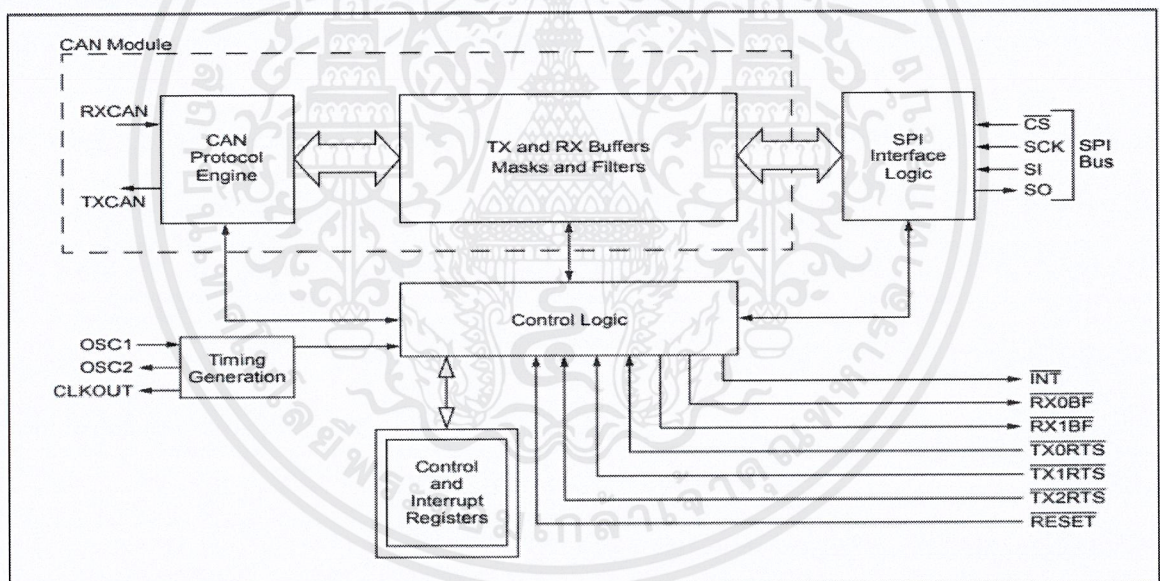
MCP2515 เป็นอุปกรณ์ต่อขยายสำหรับ Microcontroller ที่มีความสามารถในการอ่านข้อมูลจากระบบ CAN – BUS Protocol โดยมี Bus Controller พร้อม Interface SPI และ TJA1050 CAN - BUS Transceiver



รูป 3.19 MCP2515

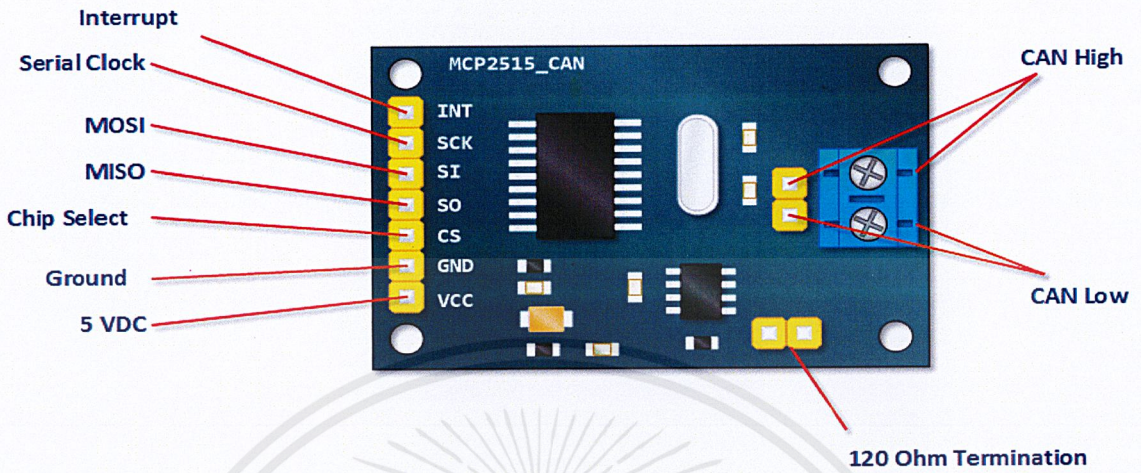
รายละเอียดสเปกอุปกรณ์

1. CAN CONTROLLER เบอร์ MCP2515 ของ MICROCHIPS
2. CAN DRIVER เบอร์ TJA1050 ของ NXP
3. เชื่อมต่อกับไมโครคอนโทรลเลอร์ภายนอกผ่านทางอินเตอร์เฟส SPI ได้ทั้งระบบ 5V และ 3.3V ด้วยความเร็วสูงสุด 10 MHz
4. รองรับมาตรฐาน CAN 2.0B, CAN ISO-11898
5. ระยะทางการเชื่อมต่อสัญญาณสื่อสารระบบ CAN – BUS โปรโตคอล ความเร็ว 62.5 Kbps คือ 1,000 เมตร และความเร็ว 1 Mbps คือ 30 เมตร
7. ใช้ไฟเลี้ยงวงจร 5 V



รูป 3.20 MCP2515 Diagram

ขบวนการเชื่อมต่อขยายการทำงาน



รูป 3.21 MCP2515 Pin

Name	PDIP/ SOIC Pin #	TSSOP Pin #	QFN Pin #	I/O/P Type	Description	Alternate Pin Function
TXCAN	1	1	19	O	Transmit output pin to CAN bus	—
RXCAN	2	2	20	I	Receive input pin from CAN bus	—
CLKOUT	3	3	1	O	Clock output pin with programmable prescaler	Start-of-Frame signal
TX0RTS	4	4	2	I	Transmit buffer TXB0 Request-to-Send; 100 kΩ internal pull-up to V _{DD}	General purpose digital input, 100 kΩ internal pull-up to V _{DD}
TX1RTS	5	5	3	I	Transmit buffer TXB1 Request-to-Send; 100 kΩ internal pull-up to V _{DD}	General purpose digital input, 100 kΩ internal pull-up to V _{DD}
TX2RTS	6	7	5	I	Transmit buffer TXB2 Request-to-Send; 100 kΩ internal pull-up to V _{DD}	General purpose digital input, 100 kΩ internal pull-up to V _{DD}
OSC2	7	8	6	O	Oscillator output	—
OSC1	8	9	7	I	Oscillator input	External clock input
V _{SS}	9	10	8	P	Ground reference for logic and I/O pins	—
RX1BF	10	11	9	O	Receive buffer RXB1 interrupt pin or general purpose digital output	General purpose digital output
RX0BF	11	12	10	O	Receive buffer RXB0 interrupt pin or general purpose digital output	General purpose digital output
INT	12	13	11	O	Interrupt output pin	—
SCK	13	14	12	I	Clock input pin for SPI interface	—
SI	14	16	14	I	Data input pin for SPI interface	—
SO	15	17	15	O	Data output pin for SPI interface	—
CS	16	18	16	I	Chip select input pin for SPI interface	—
RESET	17	19	17	I	Active-low device Reset input	—
V _{DD}	18	20	18	P	Positive supply for logic and I/O pins	—
NC	—	6,15	4,13	—	No internal connection	—
EP	—	—	21	—	Exposed Thermal Pad, connect to V _{SS}	—

รูป 3.22 MCP2515 Pin Detail

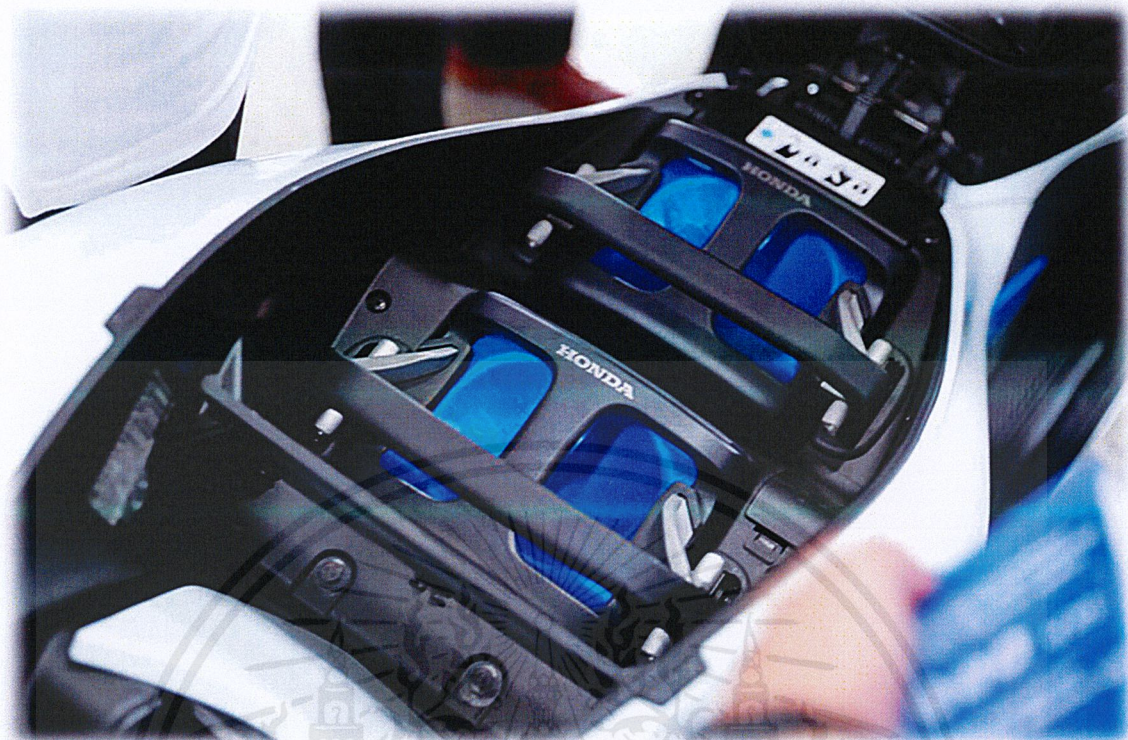
3.3.4 รายละเอียดข้อมูล Honda PCX Electric



รูป 3.23 Honda PCX Electric

รถจักรยานยนต์พลังงานไฟฟ้า โดยเปลี่ยนระบบสายพานแบบมอเตอร์ไซค์ปกติมาเป็นระบบมอเตอร์ไฟฟ้า ที่ขับเคลื่อนด้วยพลังงานไฟฟ้าแบบ 100 เปอร์เซ็นต์ ซึ่งส่งผลให้ทุกการขับขี่ของ ไร้การปล่อยก๊าซคาร์บอนไดออกไซด์ และเนื่องจากไม่มีการใช้เครื่องยนต์สันดาปภายในแล้ว ระบบมอเตอร์ไฟฟ้าก็ช่วยส่งผลให้เครื่องยนต์เงียบไร้เสียง

ตัวจักรยานยนต์มีน้ำหนักโดยรวมอยู่ที่ 144 กิโลกรัม ขนาดยางหน้าเท่ากับ 100/80 และยางหลัง 120/70 ขับเคลื่อนด้วยระบบสายพาน V - Matic แบตเตอรี่สำหรับการสตาร์ทเครื่องและการส่องสว่าง นั้น มีขนาดความจุอยู่ที่ 12V ระบบไฟเป็นแบบ LED รอบคัน ในการชาร์จไฟบ้านหนึ่งครั้งจะสามารถวิ่งได้ที่ระยะทางประมาณ 60 กิโลเมตร โดยใช้เวลาชาร์จจากแบตเตอรี่ 0 เปอร์เซ็นต์ จนถึง 100เปอร์เซ็นต์เร็วที่สุดประมาณ 4 ชั่วโมง

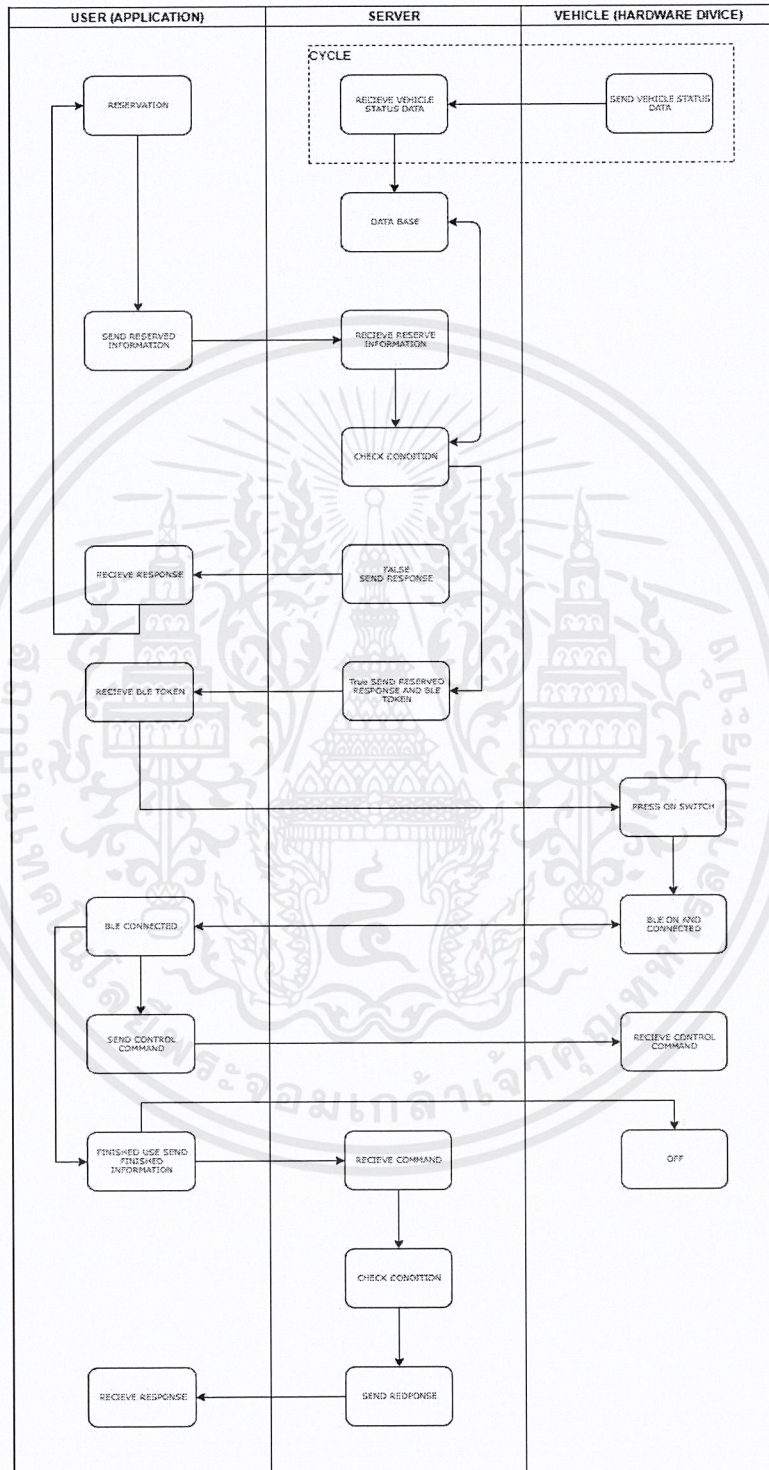


รูป 3.24 Honda PCX Electric Battery

ระบบแบตเตอรี่มีการพัฒนาโดยเฉพาะในชื่อเทคโนโลยี “Honda Mobile Power Pack” ซึ่งเป็นแบตเตอรี่ Li - on จำนวน 2 ก้อนที่จัดเก็บไว้ด้านในตัวเครื่อง ความจุขนาด 50.4 V รีดแรงม้าออกมาได้สูงสุดที่ 5.7 PS ที่ 5,000 rpm และแรงบิดสูงสุดอยู่ที่ 18 nm ที่ 500 rpm ง่าย สามารถถอดแบตเตอรี่ออกมาชาร์จด้านนอกได้ สามารถสับเปลี่ยนก้อนได้อย่างง่าย รวมถึงรองรับการเสียบชาร์จโดยตรงเข้าสู่ตัวมอเตอร์ไซค์ได้อีกด้วย หมดกังวลเรื่องปัญหาแบตเตอรี่หมดไปได้เลย

3.4 ออกแบบ และพัฒนาระบบโครงงาน

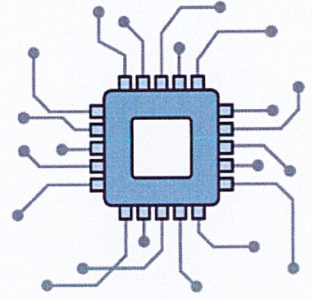
3.4.1 ฝั่งลำดับงาน



รูป 3.25 ฝั่งลำดับงาน



Vehicle



Microcontroller



Smart Phone



Server

รูป 3.26 องค์ประกอบของระบบ

3.4.2 ฟังก์ชันการใช้งาน

1. ล็อก และปลดล็อกระบบจักรยานยนต์ ผ่านการส่งคำสั่งด้วย Bluetooth แบบ BLE
2. ตรวจสอบตำแหน่งจักรยานยนต์ผ่านระบบ GPS 3G
3. ตรวจสอบหมายเลขแบตเตอรี่ ผ่านระบบ CAN – BUS Protocol
3. ตรวจสอบสถานะพลังงานแบตเตอรี่ CAN – BUS Protocol
4. ตรวจสอบสถานะการชาร์จพลังงานแบตเตอรี่ CAN – BUS Protocol
5. ตรวจสอบระยะทางการเดินทางโดยรวม CAN – BUS Protocol
6. ส่งข้อมูลสถานะจักรยานยนต์ และการกระทำต่าง ๆ ผ่าน โพรโทคอลการสื่อสารด้วยสัญญาณ 3G

3.5 การใช้งานแอปพลิเคชันฮ้อป

การสมัครสมาชิกและแอดข้อมูลบัตร

เพื่อความรวดเร็ว โปรดเตรียมบัตรประชาชนและใบขับขี่ไว้ให้พร้อม

-
- 1 กดเลือกสื่ออื่นผ่าน Facebook หรือ กดปุ่ม Register ที่มุมขวาล่าง
 - 2 กดรูปประตู่ ที่มุมขวาล่าง จากนั้นกดส่งข้อความ "Register Now"
 - 3 หากใช้งานครั้งแรกต้องสมัครสมาชิกด้วยการใส่เบอร์โทรศัพท์
 - 4 ทำการอัปโหลดบัตรประชาชนและใบขับขี่
 - 5 เตรียมใบขับขี่สำหรับเซลฟ์พร้อม กับใบหน้าของคุณ
 - 6 อ่านข้อกำหนดและเงื่อนไขและยอมรับ
 - 7 เชื่อมโยงเงินลงในช่องว่าง
 - 8 รอเจ้าหน้าที่ตรวจสอบเอกสารและอนุมัติภายใน 24 ชั่วโมง

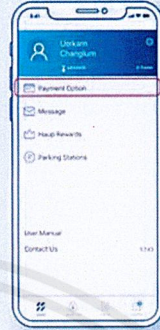
รูป 3.27 วิธีการใช้งานแอปพลิเคชัน (1)

☰ การเพิ่มข้อมูลบัตรเดบิต หรือบัตรเครดิต

ก่อนจะทำการจองรถ ต้องใส่ข้อมูลบัตรเดบิตหรือบัตรเครดิตก่อน



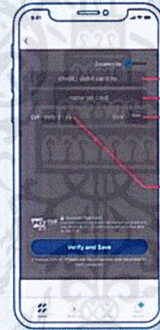
1 กดเมนูสามขีดมุมซ้ายบน



2 เลือกเมนู Payment Option



3 เลือก "Add Payment Method"



หมายเลขบัตรเครดิต/เดบิต
ชื่อผู้ถือบัตร
รหัสความปลอดภัย
วันที่ 3 หลัก ใช้สำหรับยืนยันตัวตนบัตร
CVV
เดือน/ปี ที่หมดอายุของบัตร

4 กรอกข้อมูลบนบัตร

รูป 3.28 วิธีการใช้งานแอปพลิเคชัน (2)



การจองรถ การเปลี่ยนเวลาจองรถ และคืนรถจุดอื่น



1 เปิดแอป Haupcar หรือ คลิ๊ก



2 เลือกเวลาที่ต้องการใช้รถ และเวลาที่ต้องการคืนรถ



3 ค้นหาจุดรับรถที่ต้องการ โดยกด "Search location" หรือเลือกจากแผนที่ที่อยู่บนแผนที่

กรณีที่ต้องการคืนรถจุดอื่น หรือ Dropoff/One way



4 เลือกรุ่นรถที่ต้องการ และกด "Reserve"



ปิดโหมด "Select stop station"

เปิดโหมด "Select stop station"



หลังจากเปิดโหมด "Select stop station" จะขึ้นจุดที่สามารถส่งรถคืนได้

การคำนวณราคา ระบบจะคิดค่าบริการ 2 ส่วนคือ ค่าชั่วโมง+ค่าระยะทาง



5 เลือกแพ็คเกจราคา

1 ค่าชั่วโมง คิดตามเวลาที่จอง

* หากจองตั้งแต่ชั่วโมงที่ 8 ขึ้นไป ระบบจะคิดเป็นรายชั่วโมงโดยอัตโนมัติ

แพ็คเกจ	ระยะทาง	ค่าระยะทาง
L	30 กม./ชม. หรือ 240 กม./วัน	35 บาท/กม.
M	15 กม./ชม. หรือ 120 กม./วัน	47 บาท/กม.
S	10 กม./ชม. หรือ 80 กม./วัน	53 บาท/กม.

* หากยืนยันการจองจากแพ็คเกจระยะทางที่เลือกไว้ ระบบจะคิดค่าระยะทางส่วนที่เกินจนราคาแพ็คเกจ
* ระยะทางเหมารายวัน คำนวณจากระยะทางเหมารายชั่วโมงคูณ 8 (ค่าชั่วโมงสูงสุดต่อหนึ่งวัน)



6 กดยืนยันการจอง "CONFIRM RESERVATION"

- กดเลือกอีซีโอปเกรด
- เลือกयोगทางชำระเงิน
- รากรวมก่อนหักเงิน ไร่รวมค่าระยะทางส่วนเกิน

รูป 3.29 วิธีการใช้งานแอปพลิเคชัน (3)

🔒 การรับรถ และกดปลดล็อกรถ



- 1 ถึงเวลาที่จองรถแล้ว ไปที่รถได้เลย เปิดแอป Haupcar และกดเมนู Lock/Unlock
- 2 กดปุ่ม "Unlock" เมื่อต้องการปลดล็อกรถ และกดปุ่ม "Lock" เมื่อต้องการล็อกรถ

รูป 3.30 วิธีการใช้งานแอปพลิเคชัน (4)

📍 การคืนรถ



- 1 นำรถกลับไปจอดคืนที่จุดเดิม หรือจุดอื่นที่คุณเลือกไว้ตอนจองรถ กด "Lock" เพื่อล็อกรถให้เรียบร้อย
- 2 กด "Finish your trip" เพื่อสิ้นสุดการใช้งาน

รูป 3.31 วิธีการใช้งานแอปพลิเคชัน (5)

บทที่ 4

ผลการทดลอง

การใช้งานระบบรถจักรยานยนต์ Honda PCX Electric

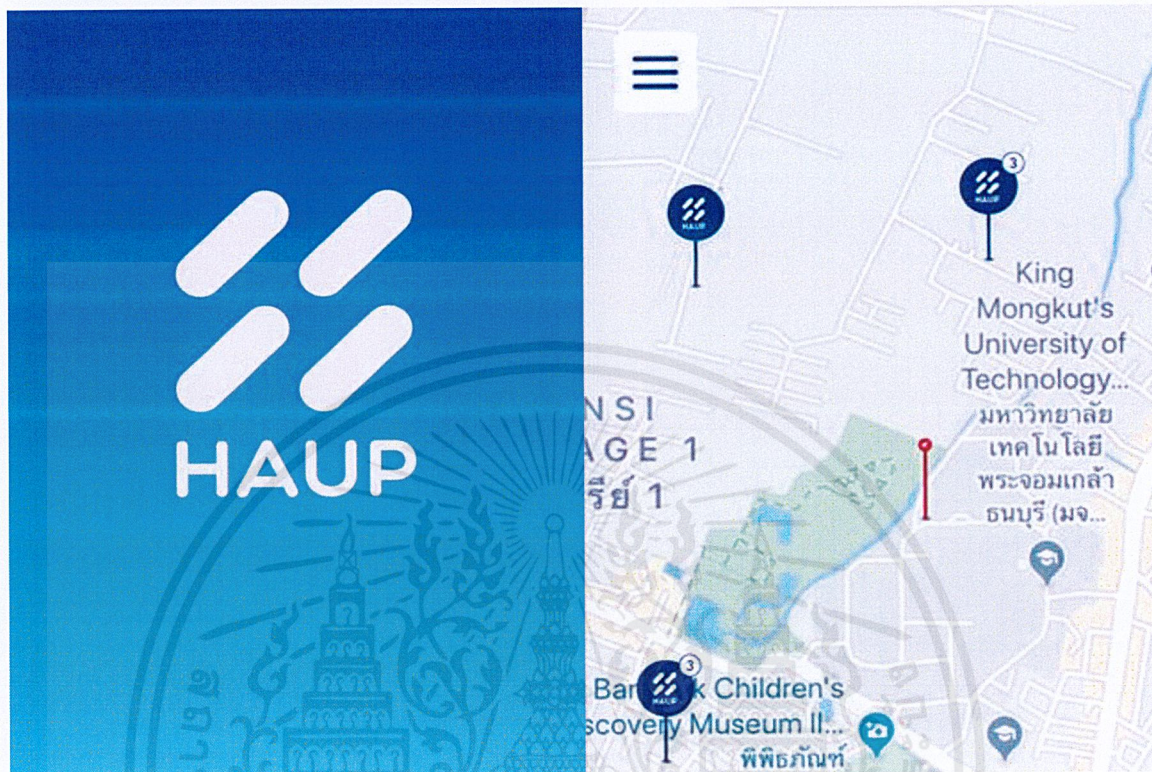


รูป 4.1 ผลการทดลอง



รูป 4.2 Electric Smart Station

1. เปิดแอปพลิเคชัน และเลือกสถานที่จุดจอดที่เราต้องการใช้งานจักรยานยนต์



รูป 4.3 การจองใช้งาน (1)

2. เลือกจักรยานยนต์คันที่ต้องการ พร้อมทั้งเลือกวันเวลาการใช้งานตามต้องการ และชำระค่าบริการ

KMUTT-Building3 Cancel **Honda PCX Electric** KMUTT-Building3
ทะเบียน 9กท-9281 แบทเตอร์รี่ 100%

START 14:20 18 Nov 2019 → END 17:20 18 Nov 2019
เวลาจอง - คับรถ 2019-11-18 → 2019-11-18
14:20 17:20

ต้องการคืนรถที่จุดอื่น

Honda PCX Electric

ส่วนลด
โปรโมชั่นโค้ด

เครดิต

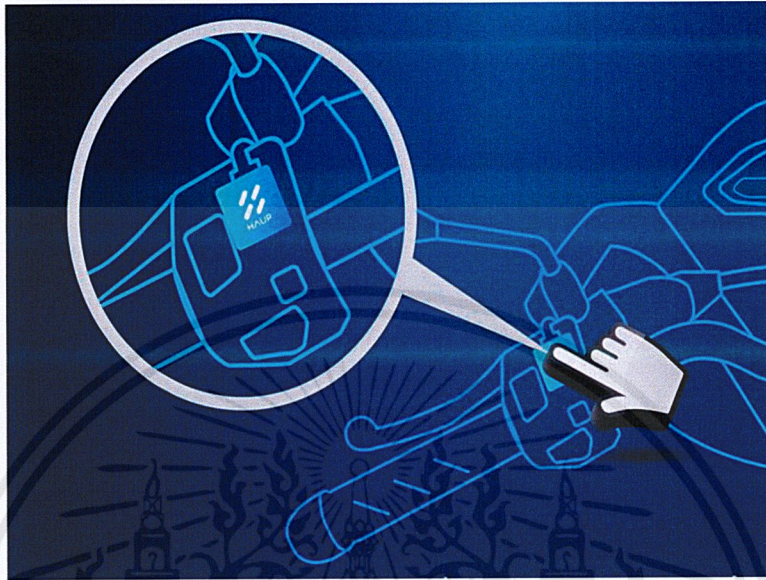
การคำนวณราคา
3 hour rate **฿90.00**
Estimated price **฿90.00**

รวม **90.00฿**

คั่นที่ 1
ทะเบียน 9กท-9266
แบทเตอร์รี่ 100%

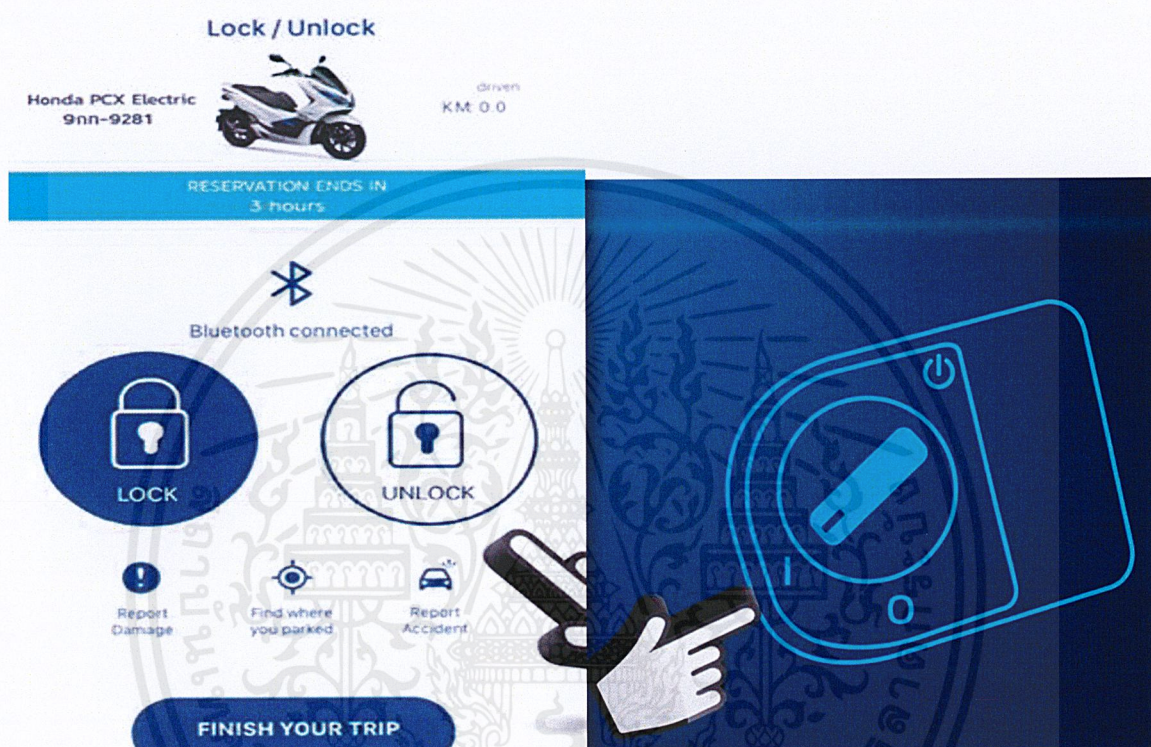
รูป 4.4 การจองใช้งาน (2)

3. กดสวิตช์ด้านบนแฮนด์จับซ้ายมือของตัวจักรยานยนต์ เพื่อเป็นการปลุกระบบอุปกรณ์ที่จักรยานยนต์



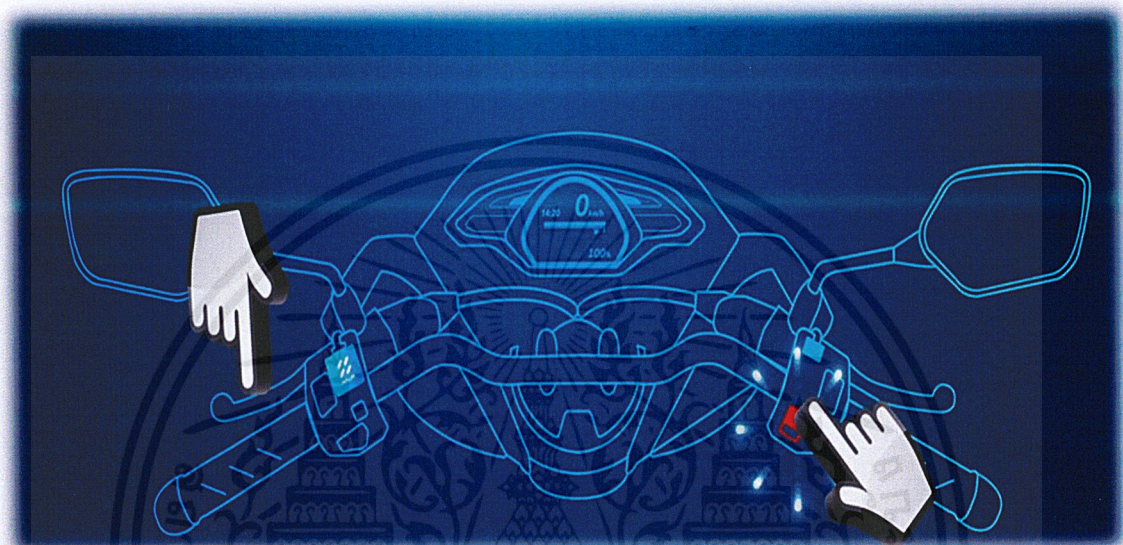
รูป 4.5 เป็ตรบบที่จักรยานยนต์

4. ระบบจะเชื่อมต่อบลูทูธอัตโนมัติ ดูจากสัญญาณที่แอปพลิเคชัน เลือก "Unlock" ที่แอปพลิเคชัน และ ปิดสวิตช์เปิดตามเข็มนาฬิกาที่ตัวจักรยานยนต์ เพื่อทำการเปิดระบบการใช้งานจักรยานยนต์



รูป 4.6 ปลดล็อกจักรยานยนต์

7. เมื่อต้องการเปิดการทำงานของจักรยานยนต์ กดสวิทช์สตาร์ทการทำงานของยานพาหนะที่ตัวยานพาหนะ พร้อมกำเบรกด้านซ้ายของเราไว้ เป็นการเปิดการทำงานของจักรยานยนต์



รูป 4.7 เปิดระบบการขับขี่จักรยานยนต์

8. เมื่อต้องการเลิกใช้งานจักรยานยนต์ ปิดสวิตช์ปิดทวนเชื่อมนาฬิกาที่ตัวจักรยานยนต์ เพื่อปิดการทำงานของระบบตัวรถจักรยานยนต์ และเลิก “Finish Your Trip” ที่แอปพลิเคชัน เป็นอันเสร็จสิ้น



รูป 4.8 ปิดระบบจบการใช้งาน

บทที่ 5

สรุปผล

ระบบระบบควบคุมยานพาหนะและการสื่อสารข้อมูล เป็นโครงการที่ทำการขึ้นมากเพื่อควบคุมระบบยานพาหนะ การสื่อสารส่งคำสั่งควบคุมการทำงาน และการดูแลระบบการทำงานป้อนข้อมูลสถานะของยานพาหนะที่สามารถอ่านค่าออกมาได้ การทำงานของระบบอุปกรณ์นี้ เป็นการทำงานของอุปกรณ์ที่จะสามารถช่วยให้ผู้ใช้งานยานพาหนะมีความสะดวกสบายในการใช้งานยานพาหนะมากขึ้น ไม่ต้องพกกุญแจติดตัว สามารถควบคุมการใช้งานยานพาหนะได้อย่างเต็มที่ผ่านโทรศัพท์มือถือของผู้ใช้งาน ซึ่งเป็นการเปลี่ยนวิถีชีวิต และสังคมของประเทศไทยให้มุ่งไปสู่สังคมเทคโนโลยีอนาคต โดยเฉพาะการอ่านค่าข้อมูลต่าง ๆ ผ่านทางระบบโปรโตคอลสื่อสารภายในยานพาหนะ ซึ่งทำให้เข้าถึงข้อมูลในระดับที่ผู้ใช้งานทั่วไปไม่อาจเข้าถึงได้ ซึ่งเป็นประโยชน์ต่อการที่สามารถวิเคราะห์ข้อมูลเชิงลึกของยานพาหนะ และพฤติกรรมการใช้งานของผู้ใช้งานได้ การพัฒนาระบบนี้จำเป็นต้องมีการพัฒนาเฟิร์มแวร์ ฮาร์ดแวร์ ซอฟต์แวร์ โปรโตคอลการสื่อสาร และการวิเคราะห์ข้อมูล ซึ่งเป็นส่วนสำคัญในการพัฒนาโครงการให้สำเร็จลุล่วง ระบบนี้สามารถนำไปต่อยอด และประยุกต์ใช้ในหลายด้านในอนาคต

ปัญหาที่พบระหว่างการดำเนินงาน

1. ระบบ CAN – BUS Protocol เป็นระบบที่หาข้อมูลได้น้อย และมีความแตกต่างกันในหลายแหล่งข้อมูล จึงทำให้เกิดปัญหาความเข้ากันได้กับระบบ Firmware พื้นฐาน

- ปรับแก้ระบบ กับ CAN – BUS Protocol ให้สอดคล้องกัน เลือกหาอุปกรณ์ Hardware ที่เหมาะสม ศักยภาพ และรูปแบบของกระแสไฟฟ้าภายในจักรยานยนต์ เพื่อให้ทำงานเข้ากับอุปกรณ์ควบคุมระบบ

2. ระบบมีการรวบรวมสัญญาณจากภายในยานพาหนะ อุปกรณ์ควบคุมระบบ และระบบ GPS

- เปลี่ยนรูปแบบการติดตั้งอุปกรณ์ ทิศทาง และระยะห่างของแต่ละส่วนในระบบอุปกรณ์ควบคุม ปรับระบบให้เน้นที่การสื่อสารตามสายเป็นหลัก

3. ระบบมีปัญหาเรื่องความเร็ว และความเสถียรในการส่งข้อมูลไปที่ Server

- แก้ปัญหาด้วยการพัฒนา Protocol ในการสื่อสารให้สอดคล้องกับการทำงานของระบบ เน้นกันไปที่การส่งข้อมูล 1 Frame ให้ได้ข้อมูลได้จำนวนมากที่สุด และลดจำนวนรอบการส่งเพื่อลดการชนกันของลำดับการส่งข้อมูล

เอกสารอ้างอิง

[1] CSS electronics, CAN BUS EXPLAINED Retrieved from

<https://www.csselectronics.com/screen/page/simple-intro-to-can-bus/language/en>

[2] Kvaser, CAN BUS Tutorial course , Retrieved from

<https://www.kvaser.com/can-protocol-tutorial/> rity Configuration Guide, Release 12.2

[3] Texas Instruments, Introduction to the Controller Area Network (CAN)

[4] CSS electronics, OBD2 EXPLAINED, Retrieved from

<https://www.csselectronics.com/screen/page/simple-intro-obd2-explained/language/en>

[5] Microchip, MCP2515 Retrieved from

<http://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf>


[6] Espressif, ESP32 Series Retrieved from

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[6] Arduino, Arduino references Retrieved from

<https://www.arduino.cc/reference/en/>

โปสเตอร์นำเสนอผลงาน



สหกิจศึกษา ปีการศึกษา 2562

ระบบควบคุมยานพาหนะและการสื่อสารข้อมูล

VEHICLE CONTROL SYSTEM AND DATA COMMUNICATION

นายจิรภัทร พิทักษ์สินการ อาจารย์ที่ปรึกษา ผศ.ดร.เกสิทธิ์ดาว สัตย์เจริญ
สาขาวิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

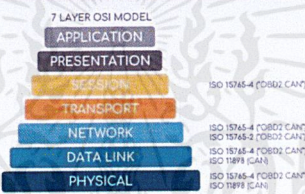
บทคัดย่อ

ระบบควบคุมยานพาหนะและการสื่อสารข้อมูล เป็นโครงการที่จัดทำระบบควบคุมการทำงานของอุปกรณ์ภายในยานพาหนะ และอ่านข้อมูลสถานะสำหรับรถจักรยานยนต์ Honda PCX Electric ซึ่งเป็นการจัดการยานยนต์ที่พลังงานไฟฟ้า โดยมีการใช้งานผ่านแอปพลิเคชันเพื่อทำงานร่วมกับอุปกรณ์ไมโครคอนโทรลเลอร์ ผู้ใช้สามารถตรวจหาตำแหน่งจุดจอด ล็อก และปลดล็อกยานพาหนะ ตรวจสอบสถานะพลังงานแบตเตอรี่ รวมถึงตรวจสอบความผิดปกติที่เกิดขึ้นกับยานพาหนะ โดยมีการอ่านค่าสถานะของยานพาหนะผ่านทางระบบ CAN - BUS protocol ซึ่งสามารถใช้พัฒนาโครงการให้นำไปใช้ระบบของรถยนต์ได้เช่นกัน

Vehicle control system and data communication. Is a project that created to control device and read its status. The project developed for Honda PCX Electric motorcycle. The system can be accessed through mobile application to communicate with microcontroller. Users can find parking spots, lock and unlock vehicles, view battery status and check the malfunction of the vehicle. The vehicle data is read via CAN - BUS protocol. It's can be developed for used with the car's system as well.

หลักการของ OBD - II protocol


OBD (On-board Diagnostics) เป็นระบบวิเคราะห์การทำงานของเครื่องยนต์พร้อมแจ้งเตือนคนขับเพื่อบอกสาเหตุถึงความผิดปกติที่เกิดขึ้นในส่วนของซอฟต์แวร์และฮาร์ดแวร์ โดยระบบ OBD - II เป็น Higher Layer Protocol ของ OSI Layer ใน CAN - BUS protocol



ภาพที่ 2 ระบบ OBD - II protocol

วิธีการใช้งานระบบ

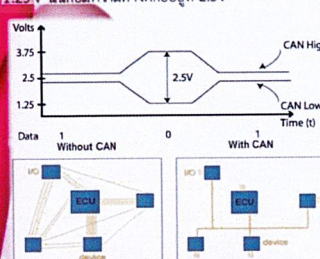
- ส่งข้อมูลการจราจรจากแอปพลิเคชันไปที่เซิร์ฟเวอร์
- เซิร์ฟเวอร์ส่งข้อมูล Token ระบุตัวตนผู้ใช้งานที่อุปกรณ์
- เซิร์ฟเวอร์ส่งข้อมูล Token ระบุตัวตนผู้ใช้งานที่แอปพลิเคชัน
- แอปพลิเคชันสามารถส่งคำสั่งควบคุมอุปกรณ์ได้



หลักการของ CAN - BUS protocol

CAN หรือ Controller Area Network เป็นเทคโนโลยีเครือข่ายอนุกรมความเร็วสูงสองสายแบบ Half - Duplex มักถูกใช้โดยทั่วไปในการสื่อสารระหว่างอุปกรณ์ต่าง ๆ ในพื้นที่ที่มีครีมีต่ำ เช่น ในยานพาหนะ ซึ่งสามารถใช้ในการจัดการการส่งข้อมูลระหว่างส่วนต่าง ๆ ในระบบยานพาหนะได้


CAN - BUS protocol สื่อสารกันโดยใช้สายไฟสองสาย เรียกว่า CAN High และอีกสายคือ CAN Low โดยเมื่อทั้งสองอยู่ในสภาวะว่าง จะมีแรงดันไฟ 2.5V แต่เมื่อต้องการส่งข้อมูล Bit ไปนั้น สาย CAN High จะมีแรงดันไฟสูงขึ้นเป็น 3.75V ส่วน CAN Low แรงดันจะลดลงไปเหลือ 1.25V นั่นคือมีความต่างศักย์อยู่ที่ 2.5V



ภาพที่ 1 CAN - BUS protocol

ขั้นตอนการใช้งาน

- เลือกจองจักรยานยนต์ สถานที่ใช้งาน เวลาการใช้งาน
- กดปุ่มเปิดระบบที่จักรยานยนต์ เชื่อมต่อบลูทูธ ปลดล็อก
- สตาร์ทจักรยานยนต์เพื่อใช้งาน
- จบการใช้งาน



ภาพที่ 4 ตัวอย่างการใช้งานระบบ

สรุปผลการโครงการ

ผลที่ได้จากการทำโครงการระบบควบคุมยานพาหนะและการสื่อสารข้อมูล สรุปได้เป็น 3 ข้อ

- ได้พัฒนาโครงการที่เป็นระบบเทคโนโลยีสมัยใหม่ การควบคุมระบบยานพาหนะที่ง่ายขึ้นผ่านทางแพลตฟอร์มของโครงการนี้ โดยมุ่งเน้นไปที่ระบบยานพาหนะพลังงานไฟฟ้า ที่จะเป็นที่นิยมสำหรับสังคมอนาคต
- ได้ความรู้ในการพัฒนาระบบ การทำงานประสานกันของฮาร์ดแวร์ และซอฟต์แวร์ การพัฒนาระบบการส่งข้อมูล
- ได้ความรู้ประสบการณ์ในหลายด้าน มีการพัฒนาความรู้พื้นฐานในด้าน เครื่องกล, ไฟฟ้า, อิเล็กทรอนิกส์ และอื่น ๆ

กิตติกรรมประกาศ

ขอขอบคุณบริษัท อีออปาร์ จำกัด ที่ได้มอบหมายให้ทำโครงการสหกิจศึกษาร่วมกับทางบริษัท ร่วมกับทางแผนกเทคนิค ในตำแหน่งงาน Hardware Engineer ซึ่งทำให้ได้รับความรู้ และประสบการณ์การทำงานที่ดีที่สุด คอยแนะนำให้คำปรึกษา และดูแลอย่างใกล้ชิด สอนในเรื่องต่าง ๆ มากมาย นอกจากนี้ยังขอขอบคุณอาจารย์ที่ปรึกษา ผศ.ดร.เกสิทธิ์ดาว สัตย์เจริญ ที่คอยช่วยในการประสานงาน ดูแล และให้คำปรึกษาในการพัฒนาโครงการงานให้ออกมาสำเร็จได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้