การพัฒนาโปรแกรมควบคุมรถบังคับวิทยุผ่านการสื่อสารไร้สายแบบวายฟาย โดยใช้การกระจายข้อมูลแบบเกณฑ์วิธีควบคุมการส่งผ่าน

THE DEVELOPMENT OF SOFTWARE FOR CONTROLLING RC-CAR VIA Wi-Fi TRANSMISSION CONTROL PROTOCOL

นายจักรพันธ์ จันทสิงห์

นายชัชชนก จิรวัฒนาพันธ์

นายณัฐวุฒิ ภิระบรรณ์

โครงงานวิศวกรรมนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า แขนงวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ปีการศึกษา 2557

การพัฒนาโปรแกรมควบคุมรถบังคับวิทยุผ่านการสื่อสารไร้สายแบบวายฟาย โดยใช้การกระจายข้อมูลแบบเกณฑ์วิธีควบคุมการส่งผ่าน

THE DEVELOPMENT OF SOFTWARE FOR CONTROLLING RC-CAR VIA Wi-Fi USING TRANSMISSION CONTROL PROTOCOL

นายจักรพันธ์ จันทสิงห์

นายชัชชนก จิรวัฒนาพันธ์

นายณัฐวุฒิ ภิระบรรณ์

โครงงานวิศวกรรมนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า แขนงวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2557

หัวข้อโครงงานวิศวกรรมไฟฟ้า

เรื่อง การพัฒนาโปรแกรมควบคุมรถบังคับวิทยุผ่านการสื่อสารไร้สายแบบวายฟาย
โดยใช้การกระจายข้อมูลแบบเกณฑ์วิธีควบคุมการส่งผ่าน

โดย นายจักรพันธ์ จันทสิงห์
นายชัชชนก จิรวัฒนาพันธ์
นายณัฐวุฒิ ภิระบรรณ์

ภาควิชา วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ชัยณรงค์ คล้ายมณี

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ อนุมัติให้นับโครงงานวิศวกรรมไฟฟ้า เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.เวคิน ปิยรัตน์)

คณะกรรมการสอบโครงงานวิศวกรรม

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์วัชรชัย วิริยะสุทธิวงศ์)

..... กรรมการ
(อาจารย์ ดร.ประมวล ชูรัตน์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ชัยณรงค์ คล้ายมณี)

การพัฒนาโปรแกรมควบคุมรถบังคับวิทยุผ่านการสื่อสารไร้สายแบบวายฟาย
โดยใช้การกระจายข้อมูลแบบเกณฑ์วิธีควบคุมการส่งผ่าน
ปีการศึกษา 2557

โดย
นายจักรพันธ์ จันทสิงห์
นายชัชชนก จิรวัฒนาพันธ์
นายณัฐวุฒิ ภิระบรรณ์

อาจารย์ที่ปรึกษา
ผู้ช่วยศาสตราจารย์ชัยณรงค์ คล้ายมณี

บทคัดย่อ

ปัจจุบันเหตุการณ์ตึกถล่มเกิดขึ้นบ่อยครั้ง การเข้าไปช่วยเหลือผู้ประสบภัยทำได้ยากและเกิดอันตราย การใช้อุปกรณ์เข้ามาช่วยเพื่อลดความเสี่ยงและเพิ่มความปลอดภัยในการช่วยเหลือจึงเป็นสิ่งจำเป็น โครงงานการพัฒนาโปรแกรมควบคุมรถบังคับวิทยุผ่านการสื่อสารไร้สายแบบวายฟาย โดยใช้การกระจายข้อมูลแบบเกณฑ์วิธีควบคุมการส่งผ่าน โดยมีวัตถุประสงค์ เพื่อใช้รถบังคับวิทยุในการสำรวจพื้นที่อันตรายและเข้าถึงได้ยาก เพื่อสามารถควบคุมรถบังคับวิทยุมีความถูกต้องแม่นยำ ผ่านทางระบบ Wi-Fi และ GPS เพื่อสามารถควบคุมรถบังคับวิทยุได้ในสภาพผิวที่หลากหลาย โดยโครงงานแบ่งการทดลองเป็นสามส่วน ส่วนที่หนึ่งสามารถควบคุมรถบังคับวิทยุในการสำรวจพื้นที่อันตรายและเข้าถึงได้ยาก ผลปรากฏว่าระยะทางที่รถบังคับวิทยุสามารถรับส่งค่าได้ในพื้นที่ที่มีคลื่นรบกวนต่ำ คือ 72 เมตร และรับส่งในพื้นที่ที่มีคลื่นรบกวนสูง คือ 68.6 เมตร ในส่วนที่สองสามารถควบคุมรถบังคับวิทยุมีความถูกต้อง แม่นยำ ผ่านทางระบบ Wi-Fi และ GPS ผลปรากฏว่าเมื่อเปรียบเทียบค่าความคลาดเคลื่อนของ GPS กับระยะที่วัดได้จริง มีค่าความคลาดเคลื่อนเท่ากับ 3.05 % และในส่วนที่สาม สามารถควบคุมรถบังคับวิทยุได้ในสภาพผิวที่หลากหลาย ผลปรากฏว่าการเข้าโค้งในช่วง 45° และ 60° ในทุกสภาพพื้นผิวสามารถผ่านได้โดยไม่หลุดโค้งและจะเริ่มมีค่าความคลาดเคลื่อนที่โค้ง มุม 90° โดยค่าความคลาดเคลื่อนจะเพิ่มมากขึ้นตามจำนวนองศาที่มากขึ้น

คำสำคัญ : สัญญาณไร้สาย / เกณฑ์วิธีควบคุมการส่งผ่าน

THE DEVELOPMENT OF SOFTWARE FOR CONTROLLING RC-CAR VIA Wi-Fi USING TRANSMISSION CONTROL PROTOCOL

Academic Year 2014

By

Mr. Chakraphan Chantrasing

Mr. Chatchanok Jirawattanapan

Mr. Natthawut Phiraban

Project Advisor

Asst. Prof. Chainarong Klaimanee

ABSTRACT

Nowadays building collapse happens frequently. To help victim is difficult and dangerous. Using equipment has reduce risk and increase safety for helping victims. So that's why it is so important. The development of Radio Controller Car controls via using Wi-Fi and the TCP process of data program project has 3 main objectives, the first part is to control a Radio Controller Car in a dangerous investigation area. The result is a Radio Controller Car can receive signal in the area which has less disturbance signal and it is about 72 meters and transceivers signal in the area which has high disturbance signal it is about 68.6 meters. The second part is to control Radio Controller Car to have an accuracy in Wi-Fi and GPS system. The result is if we compare the level of GPS discrepant to a real space, the deviation is 3.05% and in the last part is to control a Radio Controller Car in many surfaces. The result is if we control a car to bend it about 45˚ and 60˚ in any surfaces, a car will not be skidded and will have more deviation at 90˚ angle. The value of deviation will be increased by a degree of an angle.

Keywords : Wireless / TCP

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ ด้วยความอนุเคราะห์และช่วยเหลือของท่านอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ชัยณรงค์ คล้ายมณี ที่ให้คำปรึกษาและคอยห่วงใยในการดำเนินงานอยู่เสมอมา ขอขอบคุณท่านอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้และขอบคุณหัวหน้าภาควิชาวิศวกรรมไฟฟ้าที่ส่งสริมและอนุเคราะห์สถานที่ทำให้การจัดทำโครงงานสำเร็จไปได้ด้วยดี กราบขอบพระคุณบิดามารดาและญาติพี่น้องที่คอยเป็นกำลังใจเสมอมา อีกทั้งให้การสนับสนุนส่งค่าเล่าเรียนมาให้จนจบการศึกษา รวมถึงผู้ที่เขียนหนังสือการเขียนโปรแกรมเพื่อควบคุมหุ่นยนต์ ซึ่งทำให้สามารถค้นคว้าหาความรู้เพื่อนำมาประยุกต์ใช้ได้อย่างสะดวก รวมไปถึงเพื่อน ๆ ทุกคนที่คอยช่วยเหลือจนกระทั่งผู้จัดทำสามารถเขียนโครงงานฉบับนี้ได้สำเร็จ

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	**ข**
บทคัดย่อภาษาอังกฤษ	**ค**
กิตติกรรมประกาศ	**ง**
สารบัญ	**จ**
บทที่ 1 บทนำ	**2**
1.1 ความเป็นมาและความสำคัญของโครงงาน	2
1.2 วัตถุประสงค์	2
1.3 ขอบเขตโครงงานวิศวกรรม	2
1.4 ประโยชน์ที่จะได้รับ	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	**4**
2.1 ฮาร์ดแวร์ของรถบังคับวิทยุ	4
2.2 ซอฟต์แวร์ของรถบังคับวิทยุ	16
2.3 ส่วนของการเชื่อมต่อ	17
บทที่ 3 การออกแบบหลักการทำงาน	**31**
3.1 ส่วนฮาร์ดแวร์	32
3.2 หลักการทำงานด้านซอฟต์แวร์	38
3.3 การออกแบบในส่วนแสดงผล	39
บทที่ 4 การทดลองและผลการทดลองโครงงาน	**42**
4.1 การทดลองหาระยะการรับส่งสัญญาณ	42
4.2 การทดลองในส่วนของ GPS	43
4.3 การทดลองในส่วนของการควบคุมรถบนสภาพพื้นผิวต่าง ๆ	44

สารบัญ (ต่อ)

บทที่ 5 สรุปผลและข้อเสนอแนะ **49**

5.1 สรุปผลการทดลอง 49

5.2 ปัญหาและข้อเสนอแนะในโครงงาน 50

เอกสารอ้างอิง **51**

ภาคผนวก **52**

ประวัติย่อนิสิตผู้จัดทำโครงงาน **82**

สารบัญตาราง

ตารางที่

2.1 การเชื่อมต่อ Motor Driver (EVO24V9.1) 14
2.2 ตารางค่าความแรงของสัญญาณที่เชื่อมต่อระหว่างภาครับและภาคส่ง (Link Margin) 23
2.3 ตารางแสดงอัตราการลดทอนของสิ่งกีดขวางชนิดต่าง ๆ 24
4.1 แสดงระยะทางที่ไกลที่สุดที่รถสามารถรับสัญญาณได้ 42
4.2 แสดงระยะทางจากจุดเริ่มต้นไปยังจุดอ้างอิง 44
4.3 แสดงการทดลองหาระยะความคลาดเคลื่อนของการเข้าโค้งแต่ละมุมบนพื้นกระเบื้อง 46
4.4 แสดงการทดลองหาระยะความคลาดเคลื่อนของการเข้าโค้งแต่ละมุมบนพื้นหินกรวด 47
4.5 แสดงการทดลองหาระยะความคลาดเคลื่อนของการเข้าโค้งแต่ละมุมบนพื้นหญ้า 47

สารบัญรูป

รูปที่

1.1 อาคาร U Place Condotel ที่ถล่มลงมาขณะยังก่อสร้างไม่เสร็จ 2

1.2 พื้นที่อันรายและคนเข้าถึงได้ยาก 2

2.1 องค์ประกอบของระบบดาวเทียม GPS 5

2.2 USB GPS Receiver 6

2.3 ลักษณะของบอร์ด ET-Easy Mega1280 9

2.4 โครงสร้างของบอร์ด ET-Easy Mega1280 (Duino Mega) 10

2.5 แสดงหน้าต่างโปรแกรม Aduino Version 1.6.4 12

2.6 การป้อนสัญญาณความกว้างของพัลส์ 13

2.7 วงจรขับมอเตอร์รุ่น EVO24V9.1 13

2.8 การต่อใช้งาน EVO24V9.1 14

2.9 ด้านหลังของ D-Link IP Camera wireless 300 Mbps (DCS-942L) 15

2.10 แสดงหน้าต่างของโปรแกรม Microsoft Visual Basic 2010 16

2.11 Access Point มาตรฐาน IEEE 802.11g 17

2.12 การเชื่อมต่อแบบ Client/ Server (Infrastructure Mode) 20

2.13 ขั้นตอนการ Encapsulation และ Demultiplexing 29

3.1 บล็อกไดอะแกรมทั้ง 3 ส่วน 31

3.2 บล็อกไดอะแกรม 32

3.3 Access Point Rounter 33

3.4 การต่อวงจร Rounter 33

3.5 ตัวอย่าง Arduino Board 34

3.6 รูปแบบการเชื่อมต่อของ Arduino Board 34

3.7 บล็อกไดอะแกรมของ DC Motor Drive Board 35

3.8 DC Motor Driver Board 35

3.9 บล็อกไดอะแกรมของ IP Camera 35

3.10 วงจรIP Camera 36

3.11 บล็อกไดอะแกรมของ GPS Module 36

3.12 การต่อวงจร GPS 36

สารบัญรูป (ต่อ)

3.13 รูป LIPO 3s. Battery 37

3.14 วงจรแปลงกระแส 37

3.15 รูปแบบการต่อวงจรแปลงกระแส 38

3.16 บล็อกไดอะแกรมของซอฟต์แวร์ 38

3.17 การออกแบบหน้า User Interface 40

3.19 หน้า User Interface 41

4.1 แสดงตำแหน่งและระยะอ้างอิง 43

4.2 แผนผังทดสอบการเลี้ยวของรถบังคับวิทยุ 45

4.3 แสดงระยะห่างระหว่างขอบเส้นถึงขอบล้อ 45

4.4 การทดลองขับเคลื่อนรถบนสภาพพื้นผิวที่ต่างกัน 46

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

ปัจจุบันเหตุการณ์แผ่นดินไหวหรือตึกถล่มเป็นสิ่งเกิดขึ้นได้ตลอดเวลา โดยปัญหาดังกล่าวทำให้เกิดความเสียหายต่อทรัพย์สินและตัวบุคคล โดยเฉพาะอย่างยิ่งปัญหาการเข้าไปช่วยเหลือผู้ที่ติดอยู่ในซากหรือผู้รอดชีวิตเป็นสิ่งที่ทำได้ยาก โดยส่วนใหญ่การเข้าไปช่วยเหลือจะทำโดยใช้คนเข้าไปในพื้นที่เพื่อสำรวจและช่วยเหลือ เนื่องจากเป็นวิธีการที่สะดวกและรวดเร็ว ดังแสดงในรูปที่ 1.1

รูปที่ 1.1 อาคาร U Place Condotel ที่ถล่มลงมาขณะยังก่อสร้างไม่เสร็จ
ที่มา : www.bangkokbiznews.com/news/detail/599379

แต่อย่างไรก็ตาม บางเหตุการณ์นั้นอันตรายและคนเข้าไปได้ยาก เนื่องจากบางพื้นที่เป็นพื้นที่เสี่ยงภัยและอันตรายต่อผู้จะเข้าไปสำรวจหรือช่วยเหลือผู้รอดชีวิต ปัจจุบันจึงได้มีแนวคิดในการนำหุ่นยนต์ โดรน หรืออุปกรณ์อัตโนมัติต่าง ๆ ที่ใช้แทนคนมาใช้ในการเข้าไปค้นหาคนที่ติดอยู่ในซากหรือสำรวจเพื่อให้การช่วยเหลือเป็นไปได้ง่ายขึ้น แต่เนื่องจาก หุ่นยนต์และโดรน มีข้อจำกัดทางด้านการบำรุงรักษา มีกลไกที่ซับซ้อน ใช้ความชำนาญสูงในการบังคับ อีกทั้งการเคลื่อนที่ภายในบริเวณแคบก็เป็นไปด้วยความยากลำบาก และยังไม่ได้เป็นที่นิยมในปัจจุบัน เนื่องจาก

อุปกรณ์ดังกล่าวมีราคาแพง ในขณะที่รถบังคับวิทยุนั้นมีมานานแล้วและใช้กันอย่างแพร่หลาย เพียงแต่การดัดแปลงปรับปรุงให้เข้ากับรูปแบบของการเข้าไปสำรวจนั้นยังไม่มี ดังนั้นจึงเป็นสิ่งที่น่าสนใจในการประยุกต์ใช้รถบังคับวิทยุเข้าไปสำรวจพื้นที่ โดยมีการนำความรู้ทางด้าน Wi-Fi และ GPS ที่มีคุณสมบัติในการติดต่อสื่อสารจากระยะไกลและสามารถระบุตำแหน่งพิกัดบนพื้นโลกได้ โครงงานนี้จึงได้นำความสามารถของรถวิทยุบังคับและ Wi-Fi กับ GPS มาประยุกต์เข้าด้วยกันเพื่อจะแก้ไขปัญหาที่ได้กล่าวมาข้างต้น

รูปที่ 1.2 พื้นที่อันรายและคนเข้าถึงได้ยาก
ที่มา : www.manager.co.th/Around/ViewNews.aspx?

1.2 วัตถุประสงค์

1.2.1. เพื่อใช้รถบังคับวิทยุ ในการสำรวจพื้นที่อันตรายและเข้าถึงได้ยาก
1.2.2. เพื่อสามารถควบคุมรถบังคับวิทยุมีความถูกต้อง แม่นยำ ผ่านทางระบบ Wi-Fi และ GPS
1.2.3. เพื่อสามารถควบคุมรถบังคับวิทยุได้ในสภาพผิวที่หลากหลาย

1.3 ขอบเขตโครงงานวิศวกรรม

1.3.1. สามารถเชื่อมต่อระหว่างบอร์ดกับคอมพิวเตอร์แบบ TCP ได้
1.3.2. สามารถระบุตำแหน่งของรถบังคับได้
1.3.3. สามารถรับส่งข้อมูลระหว่างบอร์ด Arduino กับ Computer ได้อย่างต่อเนื่อง ผ่านระบบ Wi-Fi โดยใช้ TCP

1.4 ประโยชน์ที่จะได้รับ

1.4.1. รถบังคับวิทยุสามารถเข้าสำรวจในพื้นที่อันตรายหรือพื้นที่แคบได้

1.4.2. รถบังคับวิทยุสามารถขับเคลื่อนและสำรวจผ่าน Wi-Fi และ GPS ถูกต้องแม่นยำ

1.4.3. รถบังคับวิทยุที่สามารถเคลื่อนที่ได้หลากหลายสภาพผิว

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

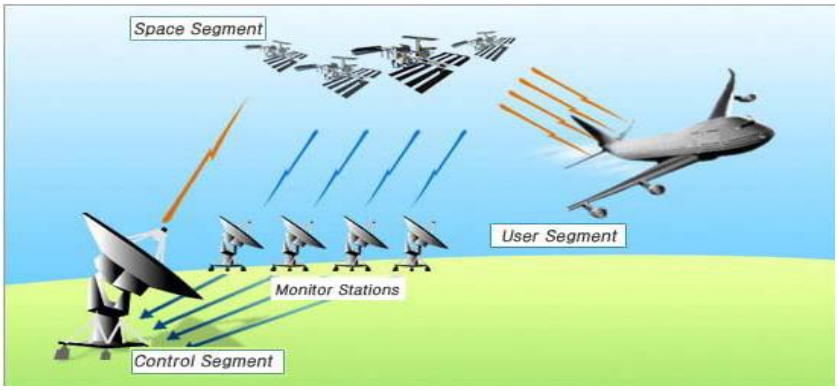
ในการออกแบบการทำงานของโปรแกรม และรถบังคับวิทยุเพื่อใช้ในการสำรวจสถานที่นั้น จำเป็นจะต้องใช้หลักการและทฤษฎีต่าง ๆ ที่เกี่ยวข้องมาประยุกต์ใช้สร้างโปรแกรมควบคุมการทำงานของรถบังคับวิทยุเพื่อให้มีประสิทธิภาพดียิ่งขึ้น มีส่วนประกอบหลัก 3 ส่วน นั่นคือ ส่วนฮาร์ดแวร์ของรถบังคับวิทยุ ส่วนซอฟต์แวร์ของรถบังคับวิทยุ และส่วนของการเชื่อมต่อ ซึ่งมีรายละเอียดดังต่อไปนี้

2.1 ฮาร์ดแวร์ของรถบังคับวิทยุ

ในส่วนของฮาร์ดแวร์นี้จะกล่าวถึงส่วนประกอบของฮาร์ดแวร์ในรถบังคับวิทยุ ประกอบด้วย ระบบกำหนดตำแหน่งบนโลก (GPS), ไมโครคอนโทรลเลอร์ Arduino Mega1280, Servo Motor, ชุดขับมอเตอร์ (Motor Driver) และกล้องบันทึกภาพวีดีโอพร้อมเสียงระบบ High Definition

2.1.1. ระบบกำหนดตำแหน่งบนโลก (GPS)

GPS ประกอบไปด้วย 3 ส่วนหลัก คือส่วนอวกาศ ประกอบด้วยเครือข่ายดาวเทียมอย่างน้อยจำนวน 3 ดวง ส่วนควบคุม ได้แก่สถานีภาคพื้นดินที่ทำหน้าที่ควบคุมระบบ กระจายอยู่ตามส่วนต่าง ๆ ของโลก เพื่อทำหน้าที่ปรับปรุงข้อมูลดาวเทียมให้มีความถูกต้องและทันสมัยอยู่ตลอดเวลา โดยแบ่งออกเป็นสถานีควบคุมหลัก สถานีติดตามดาวเทียม ทำหน้าที่รังวัดติดตามดาวเทียมตลอดเวลา สถานีรับส่งสัญญาณ และส่วนผู้ใช้งาน นั่นคือผู้ใช้งานต้องมีเครื่องรับสัญญาณที่สามารถรับคลื่นและแปรรหัสจากดาวเทียมเพื่อนำมาประมวลผลให้เหมาะสมกับการใช้งานในรูปแบบต่าง ๆ



รูปที่ 2.1 องค์ประกอบของระบบดาวเทียม GPS

ที่มา : www.sahavicha.com/?name=knowledge&file=readknowledge&id=2691

GPS ทำงานโดยการรับสัญญาณจากดาวเทียมแต่ละดวง โดยสัญญาณดาวเทียมนี้ประกอบไปด้วยข้อมูลที่ระบุตำแหน่งและเวลาขณะส่งสัญญาณ ตัวเครื่องรับสัญญาณ GPS จะต้องประมวลผลความแตกต่างของเวลาในการรับสัญญาณเทียบกับเวลาจริง ณ ปัจจุบันเพื่อแปรเป็นระยะทางระหว่างเครื่องรับสัญญาณกับดาวเทียมแต่ละดวง ซึ่งได้ระบุมีตำแหน่งของมันมากับสัญญาณดังกล่าวข้างต้น

เพื่อให้เกิดความแม่นยำในการค้นหาตำแหน่งด้วยดาวเทียม ต้องมีดาวเทียมอย่างน้อย 4 ดวง เพื่อบอกตำแหน่งบนผิวโลก ซึ่งระยะห่างจากดาวเทียมทั้ง 4 กับเครื่อง GPS จะสามารถระบุตำแหน่งบนผิวโลกได้ หากพื้นโลกอยู่ในแนวระนาบแต่ในความเป็นจริงพื้นโลกมีความโค้งเนื่องจากสัณฐานของโลกมีลักษณะกลม ดังนั้นดาวเทียมดวงที่ 4 จะทำให้สามารถคำนวณเรื่องความสูงเพื่อทำให้ได้ตำแหน่งที่ถูกต้องมากขึ้น ซึ่งจะมีความแม่นยำในระดับบวก/ลบ 10 เมตร

ความแม่นยำของการระบุตำแหน่งนั้นขึ้นอยู่กับตำแหน่งของดาวเทียมแต่ละดวง กล่าวคือถ้าระยะห่างระหว่างดาวเทียมที่ใช้งานอยู่ห่างกันย่อมให้ค่าที่แม่นยำกว่าที่อยู่ใกล้กัน และยิ่งมีจำนวนดาวเทียมที่รับสัญญาณได้มาก ก็ยิ่งให้ความแม่นยำมากขึ้น ความแปรปรวนของชั้นบรรยากาศชั้นบรรยากาศประกอบด้วยประจุไฟฟ้า ความชื้น อุณหภูมิ และความหนาแน่นที่แปรปรวนตลอดเวลา คลื่นเมื่อตกกระทบกับวัตถุต่าง ๆ จะเกิดการหักเหทำให้สัญญาณที่ได้อ่อนลง และสิ่งแวดล้อมในบริเวณรับสัญญาณเช่นมีการบดบังจากกระจก ละอองน้ำ ใบไม้ จะมีผลต่อค่าความถูกต้องของความแม่นยำ เนื่องจากถ้าสัญญาณจากดาวเทียมมีการหักเหก็จะทำให้ค่าที่คำนวณได้จากเครื่องรับสัญญาณเพี้ยนไป และสุดท้ายก็คือประสิทธิภาพของเครื่องรับสัญญาณว่ามีความไวในการรับสัญญาณมากน้อยแค่ไหนและความเร็วในการประมวลผลด้วย

รูปที่ 2.2 USB GPS Receiver
ที่มา : goo.gl/xvOCRN

USB GPS Receiver เป็นเครื่องรับสัญญาณดาวเทียม GPS แบบ USB เหมาะสำหรับการใช้งาน GPS บน Notebook หรือ Netbook เพื่อให้สามารถดูแผนที่ได้อย่างสบายสายตามากขึ้น และเป็นการประหยัดเงิน ไม่ต้องซื้อเครื่อง GPS Navigator ในราคาที่แพง การใช้งานเพียงต่อสายเข้ากับ Notebook หรือ Netbook ทางช่อง USB ก็จะทำให้สามารถรู้ตำแหน่งพิกัด รวมถึงระดับความสูง และความเร็ว ซึ่งสามารถนำไปใช้ร่วมกับโปรแกรมแผนที่นำทางเพื่อเป็นเครื่องนำทางในรถยนต์ในเรือ หรือชี้ตำแหน่งในโปรแกรม Google Earth ได้ และสามารถนำไปประยุกต์ใช้งานต่าง ๆ ได้หลากหลาย เช่น เดินป่า หรือสำรวจพื้นที่ เป็นต้น

2.1.1.1. เทคนิคการหาตำแหน่งของระบบระบุตำแหน่งบนพื้นโลก

การหาตำแหน่งมาจากแนวความคิดง่าย ๆ ที่ว่า ถ้าเรารู้ตำแหน่งของดาวเทียม และเรารู้ระยะทางจากดาวเทียมถึงเครื่องรับ เราจะสามารถหาตำแหน่งของเครื่องรับสัญญาณได้ เช่น ถ้าลองพิจารณาใน 2 มิติ แล้วทั้งตำแหน่งที่กำหนดให้ 2 จุด และระยะจากจุดทั้ง 2 ถึงจุดที่ต้องการหา (x,y) เราสามารถใช้วงเวียนเขียนเส้น โดยมีจุดที่กำหนดให้เป็นศูนย์กลาง รัศมีวงเวียนเท่ากับระยะทางที่รู้ เส้นวงกลมที่ได้จะตัดกัน 2 จุด โดยหนึ่งจุดเป็นคำตอบที่ถูกต้อง ซึ่งสมการอย่างง่ายเขียนได้ดังนี้

ระยะจากจุดที่ 1 (X_1, Y_1) $$D_1 = \sqrt{(X_1 - x)^2 + (Y_1 - y)^2} \quad (2.1)$$

ระยะจากจุดที่ 2 (X_2, Y_2) $$D_2 = \sqrt{(X_2 - x)^2 + (Y_2 - y)^2} \quad (2.2)$$

ถ้าเป็นสามมิติก็สามารถทำได้ในลักษณะเดียวกัน โดยมีจุดที่กำหนดให้ 3 จุด ในทำนองเดียวกัน ซึ่งสมการอย่างง่ายเขียนได้ดังนี้

ระยะจากจุดที่ 1 $$D_1 = \sqrt{(X_1 - x)^2 + (Y_1 - y)^2 + (Z_1 - z)^2} + \text{C dt} \quad (2.3)$$

ระยะจากจุดที่ 2 $$D_2 = \sqrt{(X_2 - x)^2 + (Y_2 - y)^2 + (Z_2 - z)^2} + \text{C dt} \quad (2.4)$$

ระยะจากจุดที่ 3 $$D_3 = \sqrt{(X_3 - x)^2 + (Y_3 - y)^2 + (Z_3 - z)^2} + \text{C dt} \quad (2.5)$$

เมื่อ C เป็นความเร็วแสง ในกรณีที่มีจำนวนดาวเทียมมากกว่านี้ ก็จะมีจำนวนสมการมากขึ้นเท่ากับจำนวนดาวเทียมสังเกตการณ์

2.1.2. การทำงานของ Microcontroller Arduino Mega1280

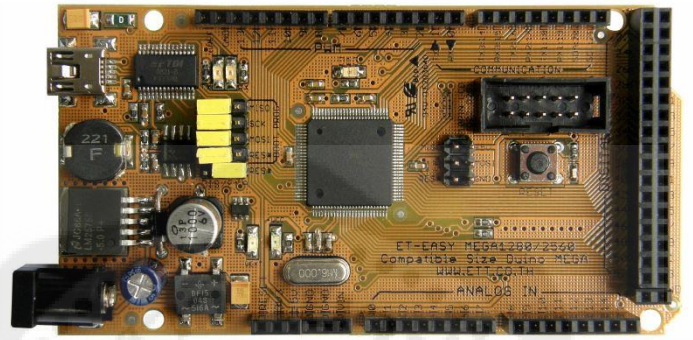
Arduino คือเครื่องมือที่จะทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าใช้เครื่องคอมพิวเตอร์แบบตั้งโต๊ะ ตัวบอร์ดออกแบบจากไมโครคอมพิวเตอร์ชิพเดี่ยวและมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้บอร์ดทำงาน

Arduino สามารถประยุกต์ทำเครื่องใช้อัจฉริยะ รับสัญญาณจากสวิทซ์ หรือเซนเซอร์และควบคุม หลอดไฟ มอเตอร์ หรืออุปกรณ์อื่น ๆ ได้ โดยการใช้บอร์ด Arduino เป็นได้ทั้งแบบทำงานอิสระ หรือทำงานติดต่อกับโปรแกรมที่ทำงานบนเครื่องพีซี ตัวบอร์ดสามารถประกอบขึ้นใช้เองหรือจะซื้อแบบสำเร็จ ซึ่งองค์ประกอบหลักของ Arduino ประกอบไปด้วย 2 ส่วนหลักคือ ฮาร์ดแวร์และซอฟต์แวร์

2.1.2.1. คุณสมบัติฮาร์ดแวร์ Arduino Mega1280

- ใช้ ATMEGA1280 เป็น MCU ประจำบอร์ด Run ความถี่ 16 MHz จาก Crystal Oscillator
- 128 Kbyte Flash (สงวนไว้ 4 Kbyte สำหรับ Bootloader) / 8 Kbyte SRAM /
- 4 Kbyte EEPROM
- รองรับการพัฒนาโปรแกรมด้วยภาษา C++ ตามแบบ Arduino Mega ได้ 100 %
- ใช้ USB Bridge ของ FTDI เบอร์ FT232RL พร้อม Over Current Protection สำหรับติดต่อสื่อสารและ Download Code จากคอมพิวเตอร์ให้บอร์ด พร้อม Jumper สำหรับปรับใช้งานบอร์ดเป็นการ Program Bootloader ให้กับ MCU จากพอร์ต USB ในบอร์ดได้เอง โดยไม่ต้องใช้เครื่องโปรแกรม AVRISP จากภายนอก
- 54 Pin Digital I/O โดยมี 14 Pin สามารถโปรแกรมหน้าที่เป็น PWM ได้
- 16 Pin Analog Input (ADC ขนาด 10 บิต 16 ช่อง)
- 4 UART(Hardware Serial Port) แบบ TTL Logic
- ขนาดของ PCB บอร์ด และตำแหน่ง Pin Connector ต่าง ๆ ตรงกันกับ Arduino Mega ทั้งหมด ทำให้สามารถนำไปติดตั้งใช้งานร่วมกับบอร์ด Shield แบบต่าง ๆ ที่มีการผลิตขึ้นมาใช้งานร่วมกันกับบอร์ด Arduino Mega ได้ทั้งหมด โดยบอร์ดมีขนาด PCB Size 5.3 cm. x 10.2 cm.
- มีขั้ว Header 10 Pin IDE ของ 8 บิต Digital I/O (D22...D29) สำหรับเชื่อมต่อกับ LCD หรือบอร์ด I/O แบบต่าง ๆ ของอีทีที เพื่อเพิ่มความสะดวกในการใช้งาน
- รองรับการใช้งานกับ External Supply ทั้งแบบกระแสสลับและกระแสตรงขนาด 7-20 V โดยเลือกใช้ Regulate แบบ Switching ขนาด 1A (LM2575-5V) ลดปัญหาเรื่องความร้อนเมื่อมีการใช้กระแสสูง สามารถใช้แหล่งจ่ายจากพอร์ต USB ได้ในกรณีใช้กระแสไม่เกิน 500 mA โดยมีวงจรเลือกแหล่งจ่ายอัตโนมัติโดยจะตัดการใช้ไฟเลี้ยงจาก USB โดยอัตโนมัติ เมื่อมีการต่อแหล่งจ่ายจากภายนอกให้บอร์ด

- ภายในบอร์ดมาพร้อมอุปกรณ์ Over Current Protection สำหรับใช้ติดต่อสื่อสารและ Download Code ที่เขียนจากเครื่องคอมพิวเตอร์มายังตัวบอร์ด พร้อม Jumper สำหรับปรับใช้งานบอร์ดเป็นการ Program Bootloader ให้กับ MCU จาก Port USB ในบอร์ดได้เองดังนั้นจึงไม่ต้องใช้เครื่องโปรแกรม AVR ISP จากภายนอก

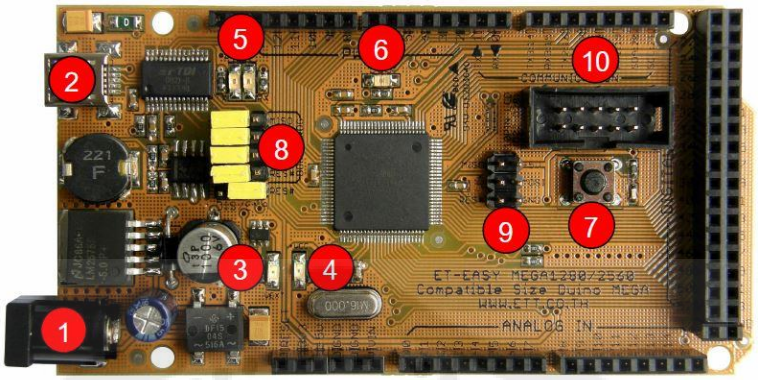


รูปที่ 2.3 ลักษณะของบอร์ด ET-Easy Mega1280
ที่มา : www.etteam.com/product2009/ET-AVR/ET-EASY_MEGA1280.html

2.1.2.2. T-Easy Mega1280 (Duino Mega)

จากที่ Arduino เป็นโครงการพัฒนาระบบ MCU ของ AVR แบบ Open Source ได้รับการแนะนำเผยแพร่ออกมาสู่สาธารณะ ซึ่งได้รับความนิยมกันอย่างแพร่หลายจากผู้คนทั่วโลก ภายในระยะเวลาอันรวดเร็ว ทางด้านของซอฟต์แวร์จึงมีการพัฒนากันอย่างต่อเนื่อง โดยทางด้านฮาร์ดแวร์เองก็ได้มีการพัฒนาปรับปรุงอย่างต่อเนื่องควบคู่กันไปด้วยเช่นเดียวกัน ขนาดของทรัพยากรต่าง ๆ รวมทั้งขนาดของหน่วยความจำ สำหรับเขียนโปรแกรม ที่มีอยู่ในชิพ AVR รุ่นเล็กที่มีอยู่เริ่มไม่เพียงพอกับการประยุกต์ใช้งานในงานบางประเภทแล้ว ทาง Arduino จึงได้ทำการพัฒนาให้ Arduino สามารถรองรับการใช้งานขนาดใหญ่ขึ้นอีก โดยปรับปรุงโปรแกรมให้ใช้กับชิพ AVR รุ่นใหญ่ขึ้น เพื่อให้มีจำนวน I/O ทั้ง Digital, Analog, PWM, UART และขนาดหน่วยความจำเพิ่มมากขึ้นกว่าเดิม แต่ก็ยังคงใช้แนวทางในการพัฒนาโปรแกรม แบบเดียวกันกับรุ่นเล็กทุกประการ โดยได้เลือกใช้ชิพ AVR เบอร์ ATMEGA1280 และออกแบบพัฒนาฮาร์ดแวร์บอร์ดขึ้นมารองรับ โดยใช้รหัสชื่อรุ่นว่า “Arduino Mega” โดยให้มีโครงสร้างการทำงานเช่นเดียวกันกับ Arduino Mega ขึ้นมา โดยใช้ชื่อว่า “ET-Easy Mega1280” และได้ออกแบบให้มีการจัดสรร Pin I/O ต่าง ๆ รวมทั้ง

ขนาดให้ตรงตามมาตรฐานของบอร์ด “Arduino Mega” เพียงแต่ได้มีการปรับปรุงข้อจำกัดบางอย่างให้ดีขึ้นกว่า Arduino Mega รุ่นมาตรฐาน เพื่อเพิ่มความสะดวกกับผู้ใช้งานมากยิ่งขึ้น



รูปที่ 2.4 โครงสร้างของบอร์ด ET-Easy Mega1280 (Duino Mega)
ที่มา : www.etteam.com/product2009/ET-AVR/man-ET-EASY-MEGA1280.pdf

2.1.2.3. อธิบายโครงสร้างของบอร์ด ET-Easy Mega1280 (Duino MeGA)

จากรูปที่ 2.7 โครงสร้างของบอร์ด ET-Easy Mega1280 (Duino Mega)

หมายเลข 1 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงจากภายนอก สามารถใช้ได้กับแหล่งจ่ายทั้งแบบกระแสสลับ และกระแสตรง พร้อมวงจร Bridge Rectifier และ Regulate แบบ Switching ช่วยลดความร้อนของ IC Regulate เมื่อมีการดึงกระแสมาก ๆ ได้เป็นอย่างดี สามารถใช้กับแรงดัน Input 7-20 V

หมายเลข 2 เป็นขั้วต่อ USB สำหรับติดต่อสื่อสารกับคอมพิวเตอร์ PC โดยใช้ FT232RL เป็น USB Bridge ในการเชื่อมต่อระหว่างคอมพิวเตอร์ PC และ MCU ในบอร์ด และยังสามารถใช้ไฟจาก พอร์ต USB เป็นแหล่งจ่ายให้กับบอร์ดได้ด้วย โดยจะมี Poly Fuse ขนาด 500 mA สำหรับป้องกันการดึงกระแสเกินจากพอร์ต USB ด้วย และที่พิเศษคือมีวงจรสำหรับตรวจสอบแหล่งจ่ายเพื่อสลับการใช้งานแหล่งจ่ายจาก USB ไปเป็น External Supply ได้เอง โดยอัตโนมัติ โดยเมื่อไม่ได้ต่อ External Supply บอร์ดจะใช้ไฟจากพอร์ต USB เป็นแหล่งจ่ายในการทำงาน แต่เมื่อมีการต่อ External Supply จะสลับไปใช้แหล่งจ่ายภายนอกอัตโนมัติ

- LED +VCC ใช้แสดงสถานะเมื่อมีการจ่ายไฟให้กับบอร์ด
- LED VEXT ใช้แสดงสถานะเมื่อมีการจ่ายไฟจาก External Supply

หมายเลข 3 เป็น LED VEXT ใช้แสดงสถานะเมื่อมีการจ่ายไฟเลี้ยงจาก

หมายเลข 4 เป็น LED +VCC ใช้แสดงสถานะของแหล่งจ่ายไฟเลี้ยง (+VCC) ของบอร์ด โดยเมื่อบอร์ดใช้แหล่งจ่ายจาก External Supply จะแสดงสถานะโดยการให้ LED VEXT และ LED +VCC ติดสว่างพร้อมกันทั้งคู่ แต่ถ้าบอร์ดใช้แหล่งจ่ายจากพอร์ต USB จะแสดงสถานะโดยการให้ LED +VCC ติดสว่างเพียงดวงเดียว

หมายเลข 5 เป็น LED แสดงสถานะของ RX และ TX ใช้สำหรับแสดงการรับส่งข้อมูลระหว่างบอร์ด ET-EASY MEGA1280 กับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB หมายเลข 6 เป็น LED D13 ใช้สำหรับทดแสดงการทำงานของ Bootloader และ ใช้ทดสอบการทำงานของบอร์ดจากการควบคุมของ Pin Digital-13 ทำงานด้วย Logic “1” และ หยุดทำงานด้วย Logic “0”

หมายเลข 7 เป็นสวิตช์ Reset ใช้สำหรับสั่ง Reset การทำงานของบอร์ด

หมายเลข 8 เป็นชุด Jumper สำหรับเลือก การ Program Bootloader โดยผ่านทางUSB Port และการใช้งานตามปกติ

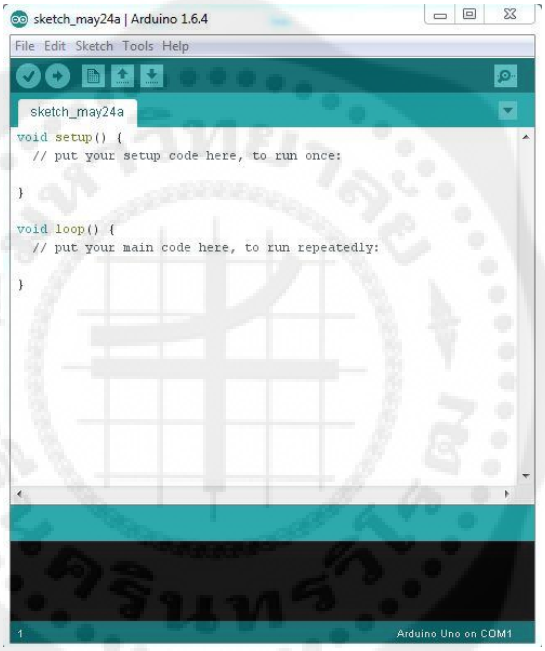
หมายเลข 9 เป็นขั้วต่อ AVRISP ใช้สำหรับ Download Code ให้กับ MCU โดยขั้วต่อ AVRISP นี้จะสามารถใช้งานได้กับเครื่องโปรแกรมทุกรุ่นที่รองรับการใช้งานกับ ATMEGA1280 และใช้ขั้วต่อตรงตามมาตรฐาน AVRISP หมายเลข 10 เป็นขั้วต่อสัญญาณจาก D[22..29] สำหรับเชื่อมต่อกับบอร์ด I/O ของอีทีที รวมทั้งจอแสดงผล LCD โดยใช้ร่วมกับ 10 PIN LCD หรือ ET-CONV SPI TO LCD

หมายเลข 10 เป็นขั้วต่อสัญญาณจาก D[22..29] สำหรับเชื่อมต่อกับบอร์ด I/O ของ อีทีที รวมทั้งจอแสดงผล LCD โดยใช้ร่วมกับ 10 PIN LCD หรือ ET-CONV SPI TO LCD

2.1.2.4. รูปแบบการทำงานของ Arduino

ซอฟต์แวร์สำหรับเขียนโปรแกรม Arduino เป็นแบบ Open Source คือ เป็นแบบเปิดเผยต้นฉบับสามารถดาวน์โหลดได้ที่ www.arduino.cc/en/Main/Software

โปรแกรม Arduino ไม่ใช่ C-Compiler จะทำหน้าที่เป็น Text Editer ของภาษา C++ ตัวหนึ่ง โดยจะทำงานร่วมกับ Utility บางส่วนที่ Arduino สร้างขึ้นมารองรับ โดย Arduino จะใช้รูปแบบการทำงานเป็น Text Editor เพื่อสื่อสารกับผู้ใช้เท่านั้น แล้ว Arduino จะไปเรียกใช้ตัวแปลภาษา C และ Utility อื่น ๆ ที่เป็นเครื่องมือพัฒนาโปรแกรมของไม โครคอนโทรลเลอร์ตระกูล AVR อีกครั้งหนึ่ง

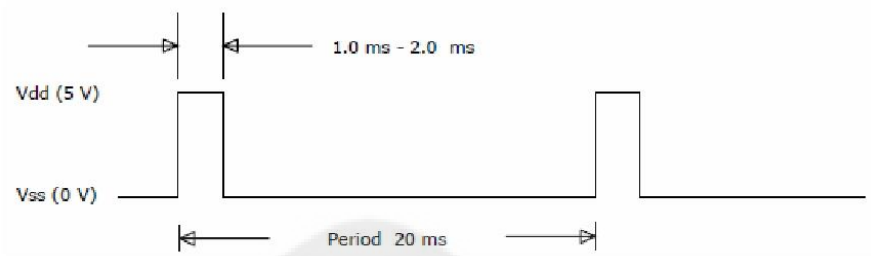


รูปที่ 2.5 แสดงหน้าต่างโปรแกรม Aduino Version 1.6.4

2.1.3. หลักการทำงานของเซอร์โวมอเตอร์ (Servo Motor)

Servo Motor เป็นมอเตอร์ไฟฟ้ากระแสตรงที่ถูกประกอบรวมกับชุดเกียร์และส่วนควบคุมไว้ในโมดูลเดียวกันหรือภายในกล่องพลาสติกเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งาน 3 เส้นเท่านั้น คือ VCC, GND และสายสัญญาณควบคุม ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้ายหรือขวาได้จากสายสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณพัลส์วิดมอด (PWM) แบบ TTL level ระดับแรงดันที่จ่ายให้มอเตอร์นี้จะอยู่ในช่วงประมาณ 4 ถึง 6 โวลต์ ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว

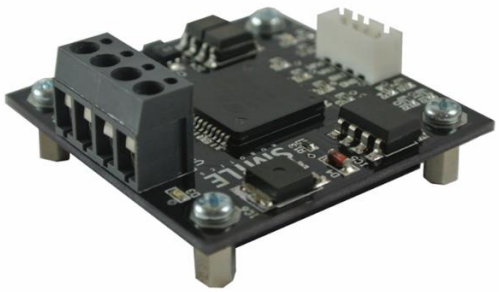
การควบคุมการทำงานของ Servo Motor ทำได้โดย การป้อนสัญญาณความกว้างของพัลส์ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้น ๆ โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด



รูปที่ 2.6 การป้อนสัญญาณความกว้างของพัลส์
ที่มา : www.semi-shop.com/knowledge/knowledge_detail.php?sk_id=35

2.1.4. ชุดขับมอเตอร์ (Motor Driver)

บอร์ดขับ DC Motor รุ่น EVO24V9.1 เป็นบอร์ดขับ DC Motor ที่ใช้ไอซี Full bridge motor drive เบอร์ VNH3SP30 ของบริษัท ST Microelectronic ซึ่งสามารถขับกระแสได้สูงสุด 30 A บอร์ดถูกออกแบบให้สามารถใช้แรงดันไฟฟ้าอินพุตสูงสุดที่ 28 V มีวงจรป้องกันสัญญาณรบกวนจากมอเตอร์ด้วย Opto Isolator มีวงจรป้องกันการต่อไฟเลี้ยงกลับขั้ว มีความทนทานสูงบอร์ดมีขนาดเล็กใช้งานง่าย สามารถต่อสัญญาณควบคุมจากไมโครคอนโทรลเลอร์หรือวงจรควบคุมอื่น ๆได้เหมาะสำหรับใช้ขับ DC Motor สำหรับหุ่นยนต์หรือประยุกต์ใช้ควบคุมอุปกรณ์ไฟฟ้าอื่น ๆ



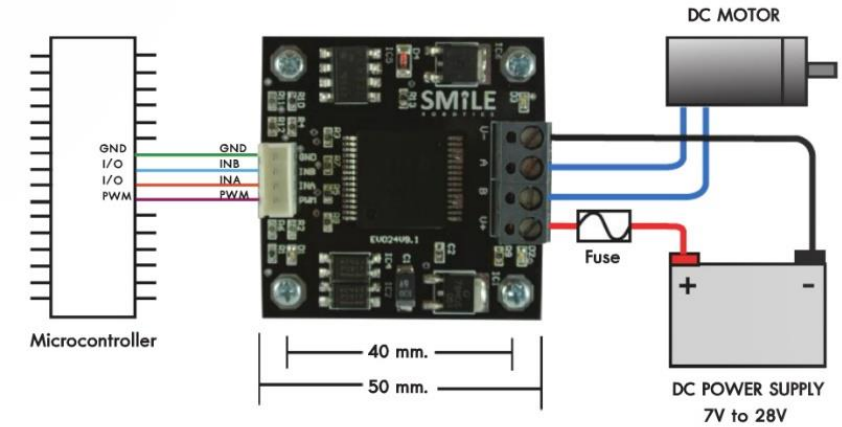
รูปที่ 2.7 วงจรขับมอเตอร์รุ่น EVO24V9.1
ที่มา : www.arduinothai.com/product/96/บอร์ดไดร์มอเตอร์-evo24v9-1/

2.1.4.1. คุณสมบัติ EVO24V9.1

- ขับกระแสสูงสุดชั่วขณะได้ถึง 30A
- ขับกระแสต่อเนื่อง 9A ที่แรงดัน 24V
- รองรับแรงดันได้ในช่วง 7-28V
- ปรับความเร็วมอเตอร์แบบ PWM ได้ที่ความถี่ไม่เกิน 5kHz
- มีวงจรป้องกันการจ่ายไฟกลับขั้ว
- มีแผ่นระบายความร้อน
- ตัดการทำงานเมื่ออุณหภูมิสูงกว่า 100 องศา

ตารางที่ 2.1 การเชื่อมต่อ Motor Driver (EVO24V9.1)

PIN	Signal	Description
V+	+VCC	ใช้ไฟเลี้ยงระบบที่แรงดันไฟ +7...+28 VDC
V-	Power GND	กราวด์ของไฟเลี้ยง
A	Motor winding (+M)	แรงดันไฟฟ้าเอาต์พุตมอเตอร์
B	Motor winding (-M)	แรงดันไฟฟ้าเอาต์พุตมอเตอร์
IN A	IN A	สัญญาณลอจิกอินพุต หมุนตามเข็มนาฬิกา
IN B	IN B	สัญญาณลอจิกอินพุต หมุนทวนเข็มนาฬิกา
PWM	PWM	สัญญาณพัลซ์วิดธ์มอดูเลต (PWM) อินพุต
GND	GND	กราวด์ของสัญญาณควบคุม



รูปที่ 2.8 การต่อใช้งาน EVO24V9.1

2.1.5. กล้องบันทึกภาพวีดีโอพร้อมเสียงระบบ High Definition

D-Link DCS-942L Wireless N Network Camera สามารถอัดได้ในตัวกล้องเอง โดยผ่าน MicroSDcard (8 GB อัดได้ 3 วัน) และกล้องตัวนี้ยังจะช่วยให้คุณสามารถเข้าภาพในสถานที่ที่คุณติดตั้งตัวกล้องได้ทุกเวลา และทุกสถานที่ผ่านทางระบบเครือข่ายอินเทอร์เน็ต จึงสามารถตรวจสอบความเคลื่อนไหวภายในบ้าน ออฟฟิศได้ผ่านทางเว็บไซต์ mydlink.com ตัวกล้องวีดีโอมีการติดตั้งไมโครโฟนไว้ที่ตัวเครื่องด้วย จึงทำให้สามารถเห็นและได้ยินสิ่งที่เกิดขึ้นภายในบ้านหรือออฟฟิศด้วยเพียงแค่เชื่อมต่อสายเคเบิล และนอกจากนั้นตัวเครื่องยังมาพร้อมด้วยซอฟต์แวร์ D-ViewCam สำหรับการใช้งานบนเครื่องพีซีในระบบวินโดวส์หรือแม็คโอเอสได้ โดยซอฟต์แวร์นี้มาพร้อมด้วยความสามารถในการบันทึกข้อมูลจากกล้องวีดีโอไปยังฮาร์ดไดร์ฟได้โดยตรง สามารถตรวจจับความเคลื่อนไหว และตั้งตารางการบันทึก

รูปที่ 2.9 ด้านหลังของ D-Link IP Camera wireless 300 Mbps (DCS-942L)

ที่มา : www.jib.co.th/web/index.php/product/readProduct/10542/16/D-LINK-IP-CAMERA-WIRELESS-300MBPS--DCS-942L-

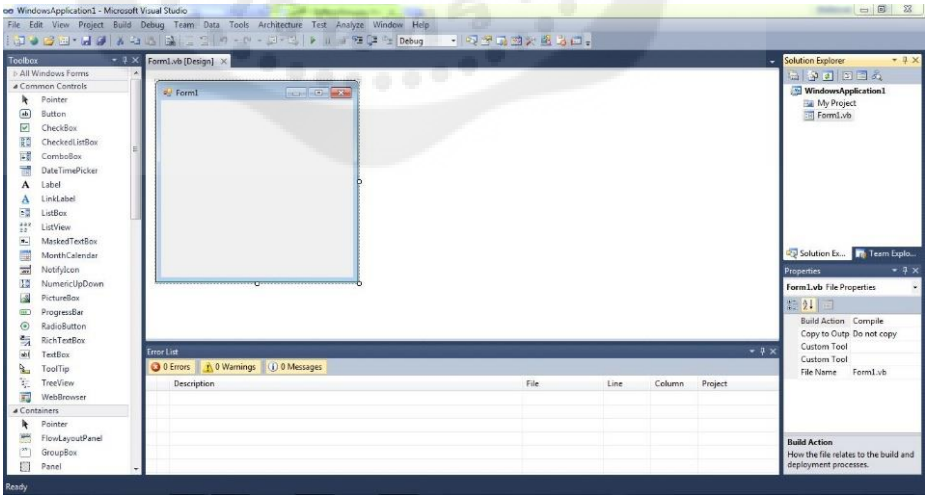
2.2 ซอฟต์แวร์ของรถบังคับวิทยุ

ในส่วนซอฟต์แวร์ของรถบังคับวิทยุนี้จะกล่าวถึงตัวโปรแกรมที่ใช้เขียนเพื่อควบคุมการทำงานของตัวรถบังคับวิทยุ นั่นคือ Microsoft Visual Basic 2010

2.2.1. Microsoft Visual Basic

Visual Basic 2010 แทบจะถูกใช้เป็นเครื่องมือหลักในการพัฒนาแอพพลิเคชันเพราะเทคโนโลยีสมัยใหม่ ล้วนแล้วแต่มาจากไมโครซอฟต์แทบทั้งสิ้น เช่น สามารถสร้างแอพพลิเคชันชนิด DHTML ซึ่งใช้ Run บนเว็บได้ รวมถึงการผนวกเทคโนโลยี ActiveX เข้ากับตัวคอนโทรลของ Visual Basic ทำให้สามารถเชื่อมโยงเข้ากับเครื่องมือที่สนับสนุนเทคโนโลยีนี้ได้แต่ Visual Basic ยังคงรักษาเอกลักษณ์อย่างหนึ่งไว้ได้เป็นอย่างดีนั่นคือ สามารถที่จะพัฒนาแอพพลิเคชันได้ ในระยะเวลาอันสั้น รวมถึงความง่ายต่อการเรียนรู้ในตัวภาษา อีกทั้งทำความเข้าใจได้อย่างรวดเร็ว เมื่อเทียบกับการเริ่มต้นเรียนรู้ด้วยภาษาอื่น ๆ และสามารถสร้างแอพพลิเคชันออกมาได้อย่างรวดเร็ว

Visual Basic 2010 ได้จัดเตรียมเครื่องมือต่าง ๆ ที่เรียกว่าคอนโทรล (Controls) ไว้คอยอำนวยความสะดวกให้แก่โปรแกรมเมอร์มากมาย ซึ่งตัวคอนโทรลเหล่านี้ที่อยู่เบื้องหลังนั่นเองเป็นเหตุให้ Visual Basic นั้นประสบความสำเร็จเป็นอย่างยิ่ง เพราะเนื่องจากมันได้ลดขั้นตอนต่าง ๆ ในการพัฒนาไปได้มากทีเดียว



รูปที่ 2.10 แสดงหน้าต่างของโปรแกรม Microsoft Visual Basic 2010

แนวทางการเขียนโปรแกรมด้วยภาษา Visual Basic จะเป็นไปในลักษณะการนำคอนโทรลชนิดต่าง ๆ เช่น TextBox, Label, ComboBox เป็นต้น นำมาวาด เพื่อออกแบบหน้าตาแอพพลิเคชันที่เรียกว่า กราฟฟิกยูสเซอร์ อินเตอร์เฟส (Graphic User Interface-GUI) คุณสามารถที่จะออกแบบอินเตอร์เฟสได้อย่างอิสระ ให้ตรงกับจุดประสงค์และ การนำไปใช้งานของคุณก่อน แล้วจึงเริ่มเขียนโค้ด เพื่อตอบสนองการกระทำของผู้ใช้ซึ่งถือเป็นหลักการเขียนโปรแกรมที่เรียกว่า การเขียนโปรแกรมเพื่อตอบสนองเหตุการณ์ที่เกิดขึ้น (Event-Driven Programming)

2.3 ส่วนของการเชื่อมต่อ

ในส่วนนี้จะเป็นการกล่าวถึงการเชื่อมต่อระหว่างรถบังคับวิทยุและคอมพิวเตอร์ ประกอบด้วย ระบบสัญญาณไร้สายแบบ Wi-Fi การกระจายข้อมูลแบบ UDP และแบบ TCP

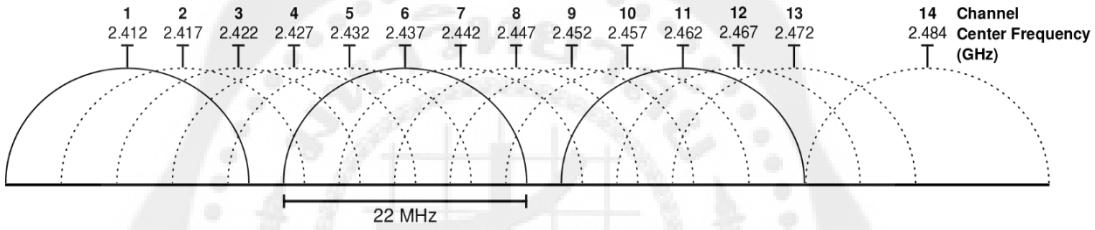
2.3.1. ระบบสัญญาณไร้สายแบบ Wi-Fi

Wi-Fi เป็นเทคโนโลยีที่ได้รับความนิยมที่ช่วยให้อุปกรณ์อิเล็กทรอนิกส์ในการแลกเปลี่ยนข้อมูลหรือการเชื่อมต่ออินเทอร์เน็ตแบบไร้สายโดยใช้คลื่นวิทยุ ซึ่งเป็นเครื่องหมายการค้าของ Wi-Fi Alliance ที่ได้ให้คำนิยามของวายฟายนี้ว่าเป็นชุดผลิตภัณฑ์ใด ๆ ที่สามารถทำงานได้ตามมาตรฐานเครือข่ายคอมพิวเตอร์แบบแลนไร้สาย (Wireless Local Area Network: WLAN) โดยใช้การส่งคลื่นวิทยุภายใต้ความถี่ 2.4 GHz ที่สามารผ่านอากาศ ทะลุกำแพง เพดาน หรือสิ่งก่อสร้างโดยอยู่บนมาตรฐาน IEEE 802.11

รูปที่ 2.11 Access Point มาตรฐาน IEEE 802.11g
ที่มา : www.dms.moph.go.th/dmsit/it03.html

2.3.1.1. การกำหนดคลื่นความถี่และข้อจำกัดในการดำเนินงานของระบบไร้สาย

สัญญาณ Wi-Fi จำนวนหนึ่งสัญญาณใช้พื้นที่ถึงห้าแชนแนล ในแถบความถี่ 2.4 GHz ตามภาพประกอบ ตัวเลขของแชนแนลใด ๆ สองแชนแนลที่แตกต่างกันห้าตัวเลขหรือมากกว่า เช่น แชนแนล 2 และแชนแนล 7 จะใช้คลื่นความถี่ที่ไม่ทับซ้อนกัน เพราะฉะนั้นแชนแนลที่ 1, 6 และ 11 เป็นกลุ่มของสามแชนแนลที่ไม่ทับซ้อนกัน ซึ่งในทวีปอเมริกาเหนือและสหราชอาณาจักรจะใช้แชนแนลดังกล่าวข้างต้นในการติดตั้งระบบไร้สายภายในอาคารหรือชั้นต่าง ๆ แต่สำหรับในทวีปยุโรปและในประเทศญี่ปุ่นจะแนะนำให้ใช้ ช่อง 1, 5, 9, และ 13 สำหรับมาตรฐาน IEEE 802.11g และมาตรฐาน IEEE 802.11n



รูปที่ 2.3 ภาพแสดงช่องความถี่นของ Wi-Fi ในแถบความถี่ 2.4 GHz
ที่มา : th.wikipedia.org/wiki/วายฟาย

การเชื่อมต่อ Wi-Fi สามารถจะหยุดชะงักหรืออินเทอร์เน็ตนั้นจะมีความเร็วลดลงอันเนื่องมาจากอุปกรณ์อื่น ๆ ในพื้นที่เดียวกันหลาย ๆ แอคเซสพอยต์ที่ใช้มาตรฐาน 802.11b และ 802.11g ที่ 2.4 GHz มีค่า Default ในการเริ่มต้นที่เป็นแชนแนลเดียวกัน ทำให้นำไปสู่ ความแออัดในบางแชนแนลโดยเฉพาะอย่างยิ่งในแชนแนลข้างเคียงที่สามารถกีดขวางการเข้าถึงและแทรกแซงการใช้แอคเซสพอยต์ของอุปกรณ์อื่น ๆ สาเหตุจากการซ้อนทับกันของแชนแนล ในแถบความถี่ของ 802.11g/b รวมทั้งมีการลดลงของอัตราส่วนสัญญาณต่อคลื่นรบกวน (Signal To Noise Ratio) หรือ SNR ในระหว่างแอคเซสพอยต์ด้วยกัน ดังนั้นสิ่งนี้จึงกลายเป็นปัญหาในพื้นที่ที่มีความหนาแน่นสูง เช่น อพาร์ทเมนต์คอมเพล็กซ์ หรืออาคารสำนักงานขนาดใหญ่ที่มีหลาย Wi-Fi Access Point

นอกจากนี้อุปกรณ์อื่น ๆ ที่ใช้แถบความถี่ 2.4 GHz เช่น กล้องที่ใช้รักษาความปลอดภัย อุปกรณ์บลูทูธ โทรศัพท์ไร้สาย เครื่องมอนิเตอร์ทารก และวิทยุ ซึ่งทั้งหมดนี้สามารถก่อให้เกิดการรบกวนเพิ่มเติมอย่างมีนัยสำคัญ นอกจากนี้ยังเป็นปัญหาเมื่อหลาย ๆ องค์กรขนาดใหญ่อื่น ๆ (เช่น มหาวิทยาลัย) พยายามที่จะให้ครอบคลุมพื้นที่ขนาดใหญ่และเกิดการทับซ้อนกัน

2.3.1.2. การลดทอนของคลื่น (Attenuation)

การแพร่กระจายของคลื่นวิทยุนั้นจะมีลักษณะที่คล้ายกับแสง เพราะคลื่นวิทยุก็เป็นคลื่นแม่เหล็กไฟฟ้าชนิดหนึ่ง ความแรงของสัญญาณวิทยุก็จะลดลงตามระยะห่างจากต้นกำเนิดเป็นอัตราส่วนผกผันกำลังสอง ยิ่งอยู่ห่างจากต้นกำเนิดแสงมากเท่าไหร่ ความแรงของสัญญาณก็จะยิ่งลดลงมากเป็นทวีคูณ ตามสมการความเข้มข้นของสัญญาณของ Friis

$$P_d = \frac{P_t}{4\pi d^2} \tag{2.6}$$

โดย Pt คือ ความแรงของสัญญาณจากเครื่องส่ง

Pd คือ ความเข้มของสัญญาณที่เครื่องรับซึ่งห่างจากเครื่องส่งเป็นระยะทาง

d คือ ระยะห่างระหว่างเครื่องส่งและเครื่องรับ

จากสมการที่ 2.1 จะเห็นว่าความเข้มของสัญญาณจะแปรผกผันในอัตราส่วนของระยะทางกำลังสอง ตัวอย่างเช่น เมื่อสถานีวิทยุส่งสัญญาณความแรง 1 วัตต์ เมื่ออยู่ห่าง 1 กิโลเมตร ความแรงของสัญญาณที่ หน้าจอเครื่องรับก็จะเต็มสเกล แต่เมื่อคุณอยู่ห่าง 10 กิโลเมตร ความแรงของสัญญาณก็เกือบจะเป็นศูนย์

2.3.1.3. สิ่งกีดขวางมีผลต่อการรับสัญญาณคลื่นวิทยุ

ในสภาวะการใช้งานทั่วไป คงหลีกเลี่ยงไม่ได้ที่จะมีสิ่งกีดขวางต่าง ๆ ไม่ว่าจะเป็นอาคาร ต้นไม้ เสาไฟฟ้าหรือกระจก สิ่งเหล่านี้อาจจะมีคุณสมบัติในการลดทอนและการดูดซับคลื่น ซึ่งวัสดุที่มีผลต่อคลื่นวิทยุที่สำคัญก็คือโลหะ ซึ่งเราจะสามารถสังเกตได้ง่าย ๆ ว่าเมื่อที่อยู่ในลิฟต์โทรศัพท์มือถือมักจะรับสัญญาณไม่ได้ หรือสายมักจะหลุดบ่อย ประการที่สองก็คือ ผนังคอนกรีตจะมีอัตราการลดทอนสัญญาณสูง เมื่อคลื่นวิทยุปะทะกับผนังคอนกรีตก็จะผ่านไปไม่ได้

ดังจะสังเกตได้จากการเดินเข้าไปในอาคารจอดรถชั้นใต้ดิน ซึ่งมักจะมีผนังคอนกรีตหนา และวัสดุเหล่านี้จะป้องกันไม่ให้คลื่นผ่านได้ทำให้รับสัญญาณไม่ได้ ลักษณะของสิ่งกีดขวางอีกประการหนึ่งก็คือ การบังคลื่นของสิ่งกีดขวางขนาดใหญ่ทำให้เกิดเขตเงา (Shadow) อาการนี้มักจะเกิดขึ้นเมื่ออยู่หลังอาคารสูง ๆ หรือภูเขา หากยิ่งอยู่ในอาคารสูงมาก ๆ หรืออาศัยอยู่ในอาคารพาณิชย์ หรือ

ตึกแถวอาจจะพบกับปัญหา ตัวอย่างเช่น เมื่ออยู่ในออฟฟิตจะใช้สัญญาณ Wireless ได้ อย่างไม่มีปัญหาแต่พอออกไปข้างนอก จะใช้งานไม่ค่อยได้ เนื่องจากเกิดการลดทอนของคลื่น เนื่องจากความหนาของผนังคอนกรีตของสำนักงาน

2.3.1.4. รูปแบบการเชื่อมต่อแบบ Client/Server (Infrastructure Mode)

ระบบเครือข่ายไร้สายแบบ Client/ Server หรือ Infrastructure Mode เป็นลักษณะการรับส่งข้อมูลโดยอาศัยแอคเซสพอยต์ที่ทำหน้าที่เป็นสะพานเชื่อมต่อระหว่างระบบเครือข่ายแบบใช้สายกับเครื่องคอมพิวเตอร์ลูกข่าย (Client) โดยจะกระจายสัญญาณคลื่นวิทยุเพื่อ รับส่งข้อมูลเป็นรัศมีโดยรอบ เครื่องคอมพิวเตอร์ที่อยู่ในรัศมีของแอคเซสพอยต์จะกลายเป็นเครือข่ายกลุ่มเดียวกันในทันที โดยเครื่องคอมพิวเตอร์จะสามารถติดต่อกันหรือติดต่อกับแม่ข่าย (Server) เพื่อแลกเปลี่ยนและค้นหาข้อมูลได้ โดยต้องติดต่อผ่านแอคเซสพอยต์เท่านั้น ซึ่งแอคเซสพอยต์ 1 จุด สามารถให้บริการเครื่องลูกข่ายได้ถึง 15-50 อุปกรณ์ ของเครื่องลูกข่าย จึงเหมาะสำหรับการนำไปขยายเครือข่ายหรือใช้ร่วมกับระบบเครือข่ายแบบใช้สายเดิมในออฟฟิต ห้องสมุด หรือในห้องประชุมเพื่อเพิ่มประสิทธิภาพในการทำงานให้มากขึ้น

รูปที่ 2.12 การเชื่อมต่อแบบ Client/ Server (Infrastructure Mode)
ที่มา : www.dms.moph.go.th/dmsict/it03.html

2.3.1.5. วิธีคำนวณระยะทางใช้งานสูงสุดของระบบแลนไร้สาย

ในการออกแบบระบบสื่อสารไร้สายทั่วไป ผู้ออกแบบส่วนใหญ่มักจะต้องการให้สัญญาณเดินทางไปได้ไกล ๆ และต้องการให้สัญญาณมีความคมชัด นั่นหมายความว่าจะต้อง ส่งสัญญาณให้มีความแรงมากพอที่จะเดินทางไปถึงปลายทางได้ โดยจะต้องคำนวณเผื่อถึงอัตราสูญเสียประเภทต่าง ๆ ด้วย ยิ่งเป็นความถี่ย่านไมโครเวฟ (2.4 GHz) ที่อุปกรณ์ Wireless LAN ใช้งานอยู่จะมีอัตราการสูญเสียค่อนข้างสูงและอ่อนไหวจากผลกระทบรอบข้างได้มาก เมื่อออกแบบนะบบสื่อสารไร้สายจึงต้องเผื่อค่าความแรงของสัญญาณให้มากพอที่เครื่องรับวิทยุจะทำงานได้ เพื่อให้เข้าใจมากขึ้น เราจะทำความเข้าใจกับอุปสรรคที่ทำให้คลื่นมีความแรงลดลงกันต่อไป

2.3.1.6. การลดทอนสัญญาณของคลื่นตามระยะทาง

การลดทอนนี้เกิดจากความแรงของสัญญาณที่ลดลงซึ่งแปรผันกับระยะทางที่เกิดขึ้นในสภาวะสุญญากาศ โดยไม่มีตัวแปรอื่น ๆ มาเกี่ยวข้องในสภาวะนี้จะไม่มีสิ่งกีดขวางมาเกี่ยวข้องเราเรียกค่านี้ว่า อัตราลดทอนในสภาวะสุญญากาศ (Free Space Loss) ดังสมการที่ 2.2

$$\text{Free Space Loss} = -27.5+20 \log_{10} (d)+20 \log_{10} (f) \qquad (2.7)$$

2.3.1.7. การลดทอนของคลื่นที่เดินทางผ่านตัวกลาง

ตัวกลางที่มีความสามารถดูดซับสัญญาณไว้ เช่นต้นไม้ ผนัง หน้าต่างกระจกหรือพื้นอาคาร โดยอัตราการลดทอนนั้นจะขึ้นอยู่กับโครงสร้างของวัตถุ ยิ่งวัตถุมีความหนาก็จะมีอัตราลดทอนที่สูง ซึ่งโดยทั่วไปจะมีค่าดังต่อไปนี้

- ต้นไม้ มีค่าอัตราการลดทอนอยู่ระหว่าง 10-20 dB โดยจะขึ้นอยู่กับขนาดของต้นไม้ ซึ่งกรณีต้นไม้ที่มีใบมากจะมีอัตราการลดทอนที่สูงกว่าต้นไม้ที่มีใบน้อย
- - ผนัง มีอัตราการลดทอนอยู่ระหว่าง 10-15 dB โดยจะขึ้นอยู่กับความหนาและวัสดุที่ใช้ ถ้าเป็นผนังยิบซัมเบา จะมีอัตราการลดทอนน้อยกว่าผนังปูนและอิฐ
- พื้นอาคาร มีอัตราการลดทอนระหว่าง 12-27 dB โดยจะขึ้นอยู่กับความหนาและวัสดุที่ใช้ หากเป็นคอนกรีตเสริมเหล็กจำนวนมาก จะมีอัตราการลดทอนที่สูงกว่าปกติ และถ้าเป็นพื้นไม้ จะมีอัตราการลดทอนที่ต่ำกว่ามาก
- กระจก มีอัตตราการลดทอนไม่มาก แต่ถ้าเป็นกระจกเคลือบปรอท ป้องกันความร้อน จะมีอัตราการลดทอนที่สูงกว่า

2.3.1.8. การคำนวณความแรงของสัญญาณระหว่างภาครับและภาคส่ง (Link Margin)

การคำนวณค่าความแรงของสัญญาณที่เชื่อมต่อกันระหว่างภาครับและภาคส่งนั้น ให้ประเมินว่าในระหว่างภาครับและภาคส่งมีความแรงของสัญญาณดีเพียงใด หากค่านี้มีค่าน้อย เมื่อมีสัญญาณรบกวนเข้ามาในระบบสื่อสารก็จะทำให้การสื่อสารล้มเหลวลงได้ ราคานี้มีค่ามากก็แสดงถึงโอกาสที่จะรับสัญญาณได้ชัดเจนมีมากขึ้นและมีโอกาสที่คลื่นเดินทางได้ไกลขึ้น ค่านี้มีตัวแปรที่มีผลกระทบดังต่อไปนี้

- กำลังส่งของคลื่นวิทยุ (Transmit Power)
- อัตราขยายของเสาอากาศภาคส่ง (Transmit Antenna Gain)
- อัตราการลดทอนของสายนำสัญญาณภาคส่ง (Transmit Cable Loss)
- อัตราขยายของเสาอากาศภาครับ (Receive Antenna Gain)
- ความแรงของสัญญาณต่ำสุดที่ภาครับจะทำงานได้ (Minimum Received Signal Level)
- อัตราการลดทอนของสายนำสัญญาณภาครับ (Receive Cable Loss)

รูปแบบการคำนวณค่าความแรงของสัญญาณวิทยุส่วนใหญ่จะใช้หน่วยเป็นเดซิเบลเพราะว่าสะดวกกว่าวิธีอื่น อ่านต้องทำงานด้วยตัวเลขธรรมดาผมต้องเป็นค่าจุดทศนิยม และมีสูตรต่าง ๆ ตามมามากมาย ดังนั้นก่อนทำการคำนวณจึงต้องแปลงค่าต่าง ๆ ให้อยู่ในรูปแบบเดซิเบลก่อน

ตัวอย่างต่อไปนี้ในตารางที่ 2.2 เป็นการคำนวณค่าความแรงสัญญาณที่เชื่อมต่อระหว่างภาครับและภาคส่งของ Access Point กับ Wireless LAN Card ที่ภาครับมีความไวที่-83 dBm ทั้งสองใช้เสาอากาศที่มีอัตราขยายเท่ากับ 2.2 dBi และต่อสายนำสัญญาณย่อยก่อน จึงต้องสายนำสัญญาณหลัก (Main Cable) ไปสู่เสาอากาศ

ตารางที่ 2.2 ตารางค่าความแรงของสัญญาณที่เชื่อมต่อระหว่างภาครับและภาคส่ง (Link Margin)

อัตราภาครับและภาคส่ง	**ค่าความแรงสัญญาณ**
กำลังส่ง (Tx Power)	15 dBm
อัตราลดทอนสายนำสัญญาณย่อย (Pigtail Loss)	-3 dBm
อัตราลดทอนสายนำสัญญาณหลักก (Main Cable Loss)	-0.5 dBm
อัตราขยายเสาอากาศภาคส่ง (Tx Antenna Gain)	2.2 dBi
ค่าอัตราส่งประสิทธิผล (EIRP) Sub Total	13.7 dBm
อัตราขยายเสาอากาศภาครับ (Rx Antenna Gain)	2.2 dBi
อัตราลดทอนสายนำสัญญาณหลัก (Main Cable Loss)	-0.5 dB
อัตราลดทอนสายนำสัญญาณย่อย (Pigtail Loss)	-3 dB
ความแรงของสัญญาณต่สุดที่ภาครับจะทำงานได้ (Rx Sensitivity)	-83 dBm
Link Margin Total	95.4 dBm

2.3.1.9. อัตราลดทอนสัญญาณวิทยุของสิ่งกีดขวางประเภทต่าง ๆ

การใช้แอคเซสพอยต์ในที่ใกล้สิ่งกีดขวางต่าง ๆ ต้องคำนึงถึงอัตราการลดทอนของสัญญาณของสิ่งกีดขวางประเภทต่าง ๆ ไปด้วย เช่น ใช้หน้าประตูเหล็กกันไฟ ตำแหน่งนี้คลื่นวิทยุมีอัตราการลดทอนสูง ไม่สามารถทะลุผ่านประตูไปได้ หรือการติดตั้งแอคเซสพอยต์ในอาคารผ่านสำนักงานที่ใช้ผนังยิปซัมกั้นห้อง คลื่นความถี่วิทยุจะสามารถเดินทางผ่านไปได้ไกลกว่าอาคารที่ใช้อิฐกั้นห้อง ซึ่งอัตราการลดทอนทั้งหมดสามารถสรุปออกมา ดังตารางที่ 2.3

ตารางที่ 2.3 ตารางแสดงอัตราการลดทอนของสิ่งกีดขวางชนิดต่าง ๆ

ประเภทสิ่งกีดขวาง	**อัตราการลดทอน**	**ตัวอย่าง**
พื้นที่โล่งแจ้ง	ไม่มี/ต่ำมาก	ห้องโถง
ไม้	ต่ำ	ผนังบ้านไม้ ประตู พื้นไม้
พลาสติก	ต่ำ	พาร์ติชันพลาสติก
อิฐเบา ผนังยิปซัม	ต่ำ	อิฐเบา ผนังยิปซัม
แก้ว กระจก	ต่ำ	ประตูแก้ว
กระจกนิรภัย	ปานกลาง	กระจกนิรภัยเสริมเส้นลวด
ร่างกายมนุษย์	ปานกลาง	กลุ่มคนจำนวนมาก
น้ำ	ปานกลาง	ตู้ปลา
อิฐแดง	ปานกลาง	ผนังห้อง
หินอ่อน	ปานกลาง	พื้น ผนัง ป้าย
กระดาษ	สูง	กระดาษ กองหนังสือ
คอนกรีต	สูง	ผนังห้อง พื้น เสา
กระจกกันกระสุน	สูง	กระจกกันกระสุนในห้องนิรภัย
โลหะ	สูงมาก	โต๊ะเหล็ก ประตูกันไฟ
กระจกเงา	สูงมาก	กระจกเงาในห้องน้ำ

2.3.2. User Datagram Protocol: UDP

โปรโตคอลแบบ UDP นั้นไม่ซับซ้อนและเข้าใจได้ง่าย โดยคุณสมบัติของโปรโตคอลแบบ UDP มีดังต่อไปนี้

- เป็นการสื่อสารแบบ End-to-End โปรโตคอล UDP สามารถระบุถึงโปรเซสที่ทำงานบนเครื่องปลายทางได้ โปรเซสหมายถึงโปรแกรมที่กำลังทำงานอยู่ เช่น การเปิด IE สามตัว จะมีทั้งหมด 3โปรเซส แต่จริงแล้ว เปิดแค่หนึ่งโปรแกรมอาจจะมีหลายโปรเซสเรียกว่าเทรด (Thread) ส่วนใหญ่มักจะเป็นโปรแกรมประเภท Server ที่รับการร้องขอจาก Client หลาย ๆ ตัวพร้อมกัน
- เป็นการเชื่อมต่อแบบ Connectionless
- ใช้การส่งข้อมูลแบบ Message Oriented โปรเซสที่ใช้โปรโตคอล UDP โดยทั่วไปจะรับและส่งบนเมสเซจแบบเดี่ยวโดยศัพท์ทางเทคนิคเรียกว่า เซกเมนต์ (Segment) คำว่าเมจเซจแบบเดี่ยวคือไม่มีการต่อหรือแบ่งเมจเซจโดยโปรโตคอล UDP
- โปรโตคอล UDP ใช้การเชื่อมต่อแบบ Connectionless โปรเซสที่ใช้โปรโตคอล UDP ไม่จำเป็นต้องสร้างการเชื่อมต่อกับปลายทางก่อนที่จะส่งข้อมูล ยกตัวอย่างง่าย ๆ คือไม่ต้องต่อสายโทรศัพท์ก่อน และเมื่อโปรเซสต้องการหยุดการสื่อสารก็สามารถจะหยุดได้ทันทีโดยไม่จำเป็นต้องทำการแฮนด์แชคกิ้ง (Handshaking) ใด ๆ การสื่อสารจะประกอบด้วยเซกเมนต์ส่วนของข้อมูลเท่านั้น ไม่มีส่วนควบคุมการสื่อสาร
- โปรโตคอล UDP ใช้อินเตอร์เฟสแบบเมสเซจ แต่ละเมสเซจจะถูกส่งโดยใช้หนึ่งแพ็กเก็ตหรือหนึ่งดาต้าแกรม ของ UDP เท่านั้น นั้นหมายความว่าผู้ออกแบบโปรแกรมประยุกต์ต้องรู้ขอบเขตของข้อมูลที่จะส่ง
- แต่อย่างไรก็ตามขนาดสูงสุดของดาต้าแกรมของ UDP ที่ดีนั้นขึ้นอยู่กับขนาดสูงสุดของดาต้าแกรมของ IP การทำให้ดาต้าแกรมของ UDP มีขนาดใหญ่นั้นอาจจะทำให้เกิดปัญหาในการสื่อสาร อันเนื่องจากดาต้าแกรมขนาดใหญ่สามารถทำให้เกิดการแตกเป็นดาต้าแกรมย่อย ๆ ในชั้นเลเยอร์ของ IP (Fragmentation) โดยปกติมักจะเกิดกับเครื่องคอมพิวเตอร์ส่วนระบบงานอุตสาหกรรม เช่น PLC มักไม่เกิดปัญหาดังกล่าวเนื่องจากมักส่งข้อมูลในจำนวนที่ไม่มาก เพราะเน้นแค่ความเร็ว
- โปรโตคอล UDP ให้ระดับประสิทธิภาพเทียบเท่ากับโปรโตคอล IP นั้นหมายว่าดาต้าแกรมสามารถหายได้ ซ้ำได้ และเสียหายได้ในการติดต่อสื่อสาร ฉะนั้นโปรโตคอล UDP จึง

เหมาะสมมากกับระบบงานประยุกต์ทางด้าน เสียงหรือวีดีโอ ที่ทนต่อความผิดพลาดในการส่งข้อมูลได้สูง

- โปรแกรมประยุกต์ที่ส่งดาต้าแกรมไปยังเซิร์ฟเวอร์จำเป็นต้องระบุจุดหมายปลายทางที่มากกว่าหมายเลข IP ดำเนินการ ก็เพราะว่าดาต้าแกรมนั้นปกติจะส่งตรงไปยังโปรเซสปลายทางที่แน่นอน ดังนั้นโปรโตคอล UDP ต้องจัดการงานเหล่านี้โดยใช้หมายเลขพอร์ต

2.3.2.1. ปัญหาของ UDP

เนื่องจากโปรโตคอล UDP สนับสนุนเพียงการจัดส่งข้อมูลแบบพื้นฐาน ปัญหาเกี่ยวกับโปรโตคอล UDP มักจะเกี่ยวข้องกับการจัดส่งข้อมูล โปรแกรมประยุกต์ที่ใช้โปรโตคอล UDP อาจจะต้องทำงานหนักในระบบเครือข่ายที่มีความคับคั่งของข้อมูลสูง การหายไปของดาต้าแกรมของ UDP จำเป็นต้องถูกจัดการที่ระดับโปรแกรมประยุกต์ ดังนั้นผู้พัฒนาโปรแกรมประยุกต์ต้องทราบว่าส่วนงานใดที่โปรโตคอล UDP ไม่ดำเนินการ โดยมีรายละเอียดที่ต้องทราบดังต่อไปนี้

- โปรโตคอล UDP ไม่สร้างการเชื่อมต่อก่อนส่งข้อมูล ส่งทันทีถ้าเตรียมข้อมูลเสร็จเรียบร้อย เปรียบเทียบได้กับการส่งจดหมาย ซึ่งไม่ทราบว่าผู้รับอยู่ที่ปลายทางหรือไม่
- โปรโตคอล UDP ไม่ต้องการการยืนยันว่าได้รับข้อมูลที่ส่งไปหรือไม่เปรียบเสมือนการส่งจดหมายไม่ลงทะเบียน
- โปรโตคอล UDP ไม่รับประกันว่าข้อมูลไปถึงปลายทางทุกครั้ง
- โปรโตคอล UDP ไม่ตรวจสอบการสูญหายของดาต้าแกรม และไม่มีการส่งซ้ำ
- โปรโตคอล UDP ไม่รับประกันว่าดาต้าแกรมที่ส่งออกไปจะไปถึงปลายทางตามลำดับก่อนหลัง
- โปรโตคอล UDP ไม่มีกลไกเกี่ยวกับการควบคุมการไหลของข้อมูลในกรณีที่มีความคับคั่งของข้อมูลสูง

อย่างไรก็ตามก็ยังมีโอกาสที่ข้อมูลที่ส่งไม่ไปถึงปลายทางตามลำดับ ถ้าต้องการความสามารถส่วนนี้ก็ต้องใช้โปรโตคอล TCP หรือต้องไปจัดการที่ระดับโปรแกรมประยุกต์ การใช้โปรโตคอล UDP ค่อนข้างจะพัฒนาง่าย เพราะโปรโตคอล UDP ไม่ต้องติดตามผลลัพธ์ในการส่งและรับทุกแพ็กเก็ต รวมทั้งไม่ต้องสร้างการเชื่อมต่อและจบการเชื่อมต่อ เสมือนการรอการรับสายโทรศัพท์ และการขอจบการสนทนา

2.3.3. Transmission Control Protocol: TCP

TCP เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหาโปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้ นอกจากนี้ทีซีพี เป็นโปรโตคอลที่ได้รับความนิยมที่สุดในโลกของอินเทอร์เน็ต มีแอพพลิเคชันจำนวนมากที่ใช้โปรโตคอลทีซีพีเป็นสื่อกลางในการเชื่อมต่อ เช่น เวิลด์ไวด์เว็บ เป็นต้น

TCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐาน 3 ประการ คือ

1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน

2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่น ในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ

3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ Real Time และทั้งการสื่อสารแบบเสียง และข้อมูล

TCP/IP เป็นโพรโตคอลที่มีความน่าเชื่อถือ โดยการเคลื่อนย้ายข้อมูลจะเป็นไปในลักษณะการเรียงลำดับจากฝั่งผู้ส่งข้อมูลออกไปยังผู้รับข้อมูลและมีการตอบรับจากทางฝั่งของผู้รับอย่างมีความน่าเชื่อถือ รวมถึงพฤติกรรมในการป้องกันการใช้เครือข่ายอย่างไม่มีประสิทธิภาพ และการควบคุมการไหลของข้อมูลทั้งทางฝั่งผู้ส่งและผู้รับข้อมูล

2.3.3.1. บริการถ่ายโอนข้อมูลแบบน่าเชื่อถือ (Reliable Data Transfer Service)

การให้บริการแบบนี้จะรับประกันความถูกต้องของการถ่ายโอนข้อมูล กล่าวคือ ข้อมูลที่ถูกส่งออกจากโฮสต์ต้นทาง จะสามารถส่งถึงปลายทางเสมอ โดยมีโปรโตคอลทีซีพีเป็นตัวควบคุมกระบวนการทำงาน โดยมีคุณสมบัติการทำงานเป็นดังนี้

- ก่อนจะมีการถ่ายโอนข้อมูล ต้องมีการสร้างการเชื่อมต่อก่อน โดยวิธีการทำแฮนด์เชคกิ้ง
- ความน่าเชื่อถือและเรียงลำดับข้อมูลที่จะส่งให้กับแอพพลิเคชั่น ซึ่งมีกลไกการตอบรับและการส่งข้อมูลซ้ำ เพื่อช่วยให้ข้อมูลสามารถส่งส่งถึงปลายทางได้อย่างสมบูรณ์และเรียงลำดับ

- การควบคุมการไหลของข้อมูล (Flow Control) เพื่อมิให้ข้อมูลที่กำลังถ่ายโอนเกิดการท่วมล้น (Overflow) ที่โฮสต์ด้านรับ
- การควบคุมความแออัดของจราจรบนระบบเครือข่าย (Congestion Control) เพื่อควบคุมไม่ให้โฮสต์ทำการปล่อยข้อมูลลงบนระบบเครือข่ายมากเกินไป ในขณะที่ระบบเครือข่ายเกิดสภาพความแออัด

2.3.3.2. บริการแบบไม่น่าเชื่อถือ (Unreliable Data Transfer Service, Best Effort)

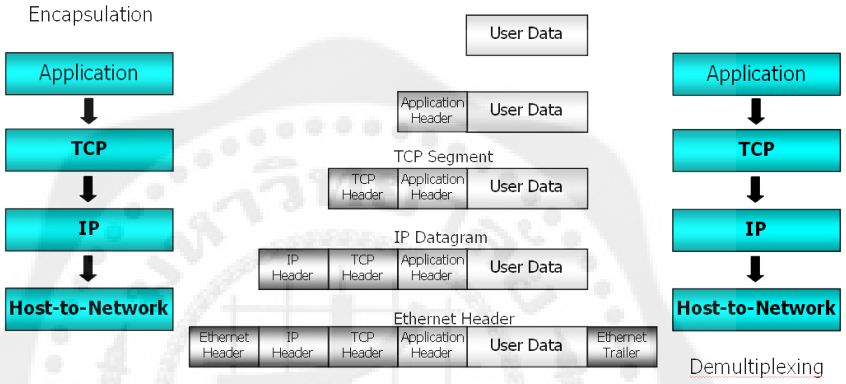
การให้บริการแบบนี้จะไม่รับประกันความถูกต้องของการถ่ายโอนข้อมูล กล่าวคือ ข้อมูลที่ถูกส่งออกจากโฮสต์ต้นทาง อาจจะจะสามารถส่งถึงปลายทาง หรือไม่ถึงปลายทางก็ได้ โดยมีโปรโตคอล UDP เป็นตัวควบคุมกระบวนการทำงาน โดยมีคุณสมบัติการทำงานเป็นดังนี้

- ไม่มีการสร้างการเชื่อมต่อก่อน (No Connection Setup)
- ไม่รับประกันความถูกต้อง และไม่เรียงลำดับข้อมูลที่จะส่งให้กับแอพพลิเคชั่น
- ไม่มีการควบคุมการไหลของข้อมูล (No Flow Control)
- ไม่มีการควบคุมความแออัดของจราจรบนระบบเครือข่าย (No Congestion Control)

แอพพลิเคชั่นที่ให้บริการโดยใช้การบริการในกลุ่มนี้ได้แก่ การทำมีเดียสตรีมมิ่งหรือการโทรศัพท์ผ่านระบบอินเทอร์เน็ต (Internet Telephony)

2.3.3.3. Encapsulation/Demultiplexing

เลเยอร์แต่ละเลเยอร์จะทำการประกอบข้อมูลที่ได้รับมากับข้อมูลส่วนควบคุม ซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลเรียกว่า Header ภายใน Header จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูล ก็จะเกิดกระบวนการทำงานย้อนกลับ คือ โปรโตคอลเดียวกันทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็น Header ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมาว่ามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing



รูปที่ 2.13 ขั้นตอนการ Encapsulation และ Demultiplexing

ที่มา : www.tnetsecurity.com/content_basic/tcp_ip_knowledge.php

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละเลเยอร์มีชื่อเรียกแตกต่างกัน ดังนี้

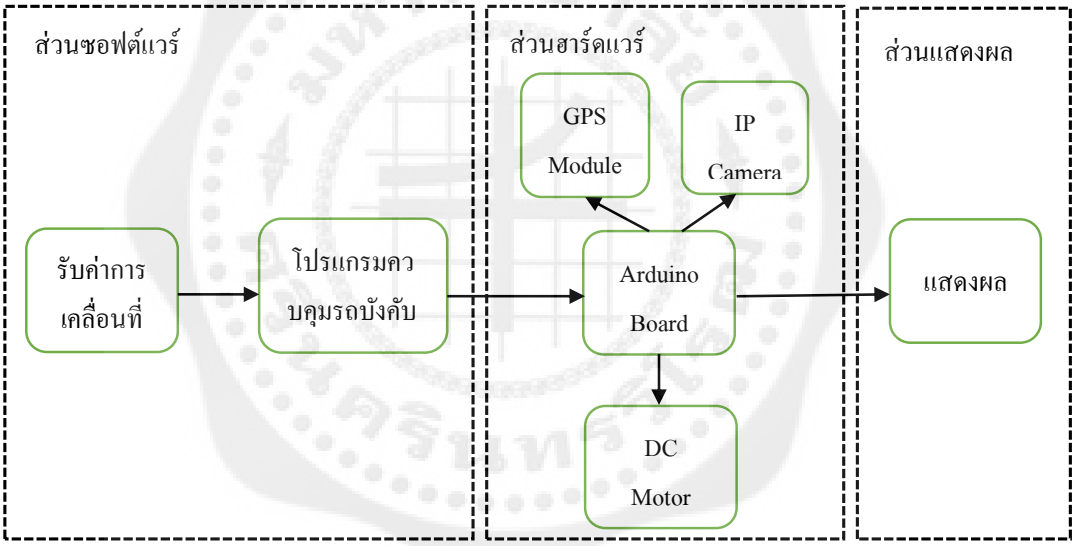
- ข้อมูลที่มาจากผู้ใช้หรือข้อมูลที่ผู้ใช้เป็นผู้ป้อนให้กับ Application เรียกว่า User Data
- เมื่อแอพพลิเคชั่นได้รับข้อมูลจากผู้ใช้ก็จะนำมาประกอบกับส่วนหัวของแอพพลิเคชั่นเรียกว่า Application Data และส่งต่อไปยังโปรโตคอล TCP
- เมื่อโปรโตคอล TCP ได้รับ Application Data ก็จะนำมารวมกับ Header ของโปรโตคอล TCP เรียกว่า TCP Segment และส่งต่อไปยังโปรโตคอล IP
- เมื่อโปรโตคอล IP ได้รับ TCP Segment ก็จะนำมารวมกับ Header ของ โปรโตคอล IP เรียกว่า IP Datagram และส่งต่อไปยังเลเยอร์ Host-to-Network Layer
- ในระดับ Host-to-Network จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า และทำการส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป

ทุกสิ่งทุกอย่างในแอพพลิเคชัน Visual Basic จะมองเป็น Object เช่น Menu bar, dialog box, toolbars, TextBox ที่สามารถควบคุมพฤติกรรม แก้ไข และกระทำโดยตรงนั้นได้ด้วยการเขียนโค้ด หรือสามารถเปลี่ยนแปลงคุณสมบัติประจำตัวได้โดยตรง ตัวคอนโทรลก็ถูกมองเป็น Object เช่นกัน ในทุก ๆ Object จะมีคุณสมบัติและวิธีการประจำตัว ซึ่งในแต่ละ Object อาจจะมีคุณสมบัติและวิธีการที่เหมือนหรือต่างกันก็ได้ ขึ้นอยู่กับชนิดของ Object

ในการพัฒนาแอพพลิเคชันด้วย Visual Basic การเขียนโค้ดจะถูกแบ่งออกเป็นส่วน ๆ ที่เรียกว่า โพรซีเดอร์ แต่ละโพรซีเดอร์จะประกอบไปด้วยโค้ดที่พิมพ์เข้าไปแล้วทำให้คอนโทรลหรือ Object นั้น ๆ ตอบสนองการกระทำ ของผู้ใช้ได้โดยสมบูรณ์ในตัวมันเอง ซึ่งเรียกว่า การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) หรือเรียกว่า OOP แต่ตัวภาษา Visual Basic ยังไม่ถือว่าเป็น OOP อย่างแท้จริง เนื่องจากข้อจำกัดหลาย ๆ อย่างที่ Visual Basic ไม่สามารถทำได้เหมือนกับภาษา C++ การเขียนโปรแกรมเชิงวัตถุ มีข้อดีก็คือ ตัวโค้ดจะถูกแบ่งออกเป็นส่วน ๆ ทำให้ง่ายต่อการตรวจสอบและดักจับข้อผิดพลาด (Debug) ซึ่งการแก้ไขดังกล่าวนี้จะไม่ไปกระทบกับโค้ดส่วนอื่น ๆ ในตัวแอพพลิเคชันนั้น ทำให้โปรแกรมเมอร์สามารถพัฒนาแอพพลิเคชันออกมา ได้อย่างสมบูรณ์แบบมากที่สุดและใช้เวลาน้อยกว่าในอดีต

บทที่ 3
การออกแบบหลักการทำงาน

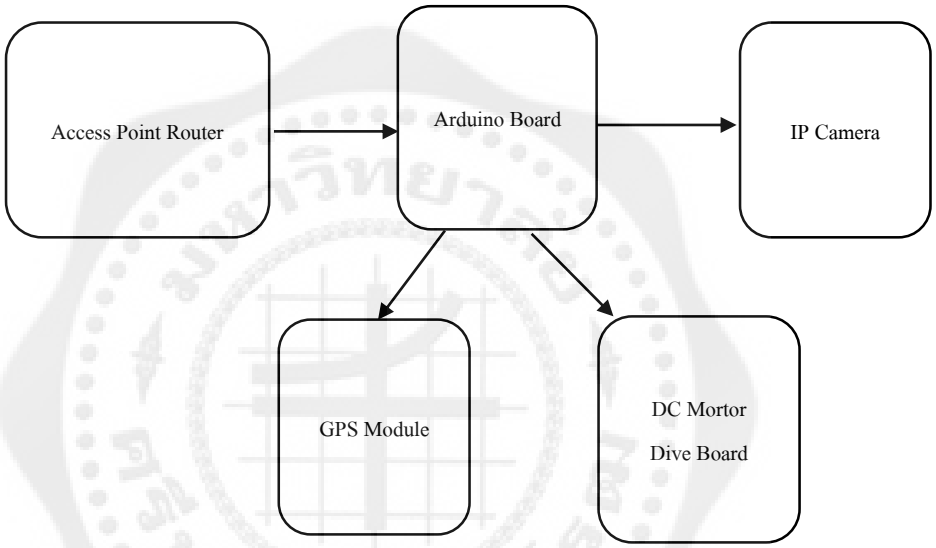
การออกแบบรถบังคับ นั้นเริ่มต้นจากการศึกษา ลักษณะพื้นที่แคบเข้าถึงได้ยาก ในหลากหลายพื้นผิว ว่ามีลักษณะเป็นแบบใด จึงได้ทำการออกแบบรถสำรวจที่สามารถเข้าถึงในพื้นที่นั้นให้ทำงานได้ตามขอบเขตที่วางเอาไว้ ซึ่งจะแบ่งการทำงานออกเป็น 3 ส่วน คือ ส่วนฮาร์ดแวร์ ส่วนซอฟต์แวร์ และ ส่วนของการแสดงผล ซึ่งจะมีบล็อกไดอะแกรมตามรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมทั้ง 3 ส่วน

3.1 ส่วนฮาร์ดแวร์

ในส่วนของฮาร์ดแวร์นั้นได้ออกแบบให้รถบังคับสามรถใช้งานได้ในสภาพพื้นผิวที่หลากหลายและสามารถเคลื่อนที่ได้ในพื้นที่แคบและยาก การทำงานจะเริ่มจากการเปิดสวิตส์หลักเพื่อให้อุปกรณ์ส่วนต่าง ๆ ทำงาน โดย Access Point Router นั้นจะส่งข้อมูลให้ Arduino Board และส่งสัญญาณการเชื่อมต่อไปที่ Computer Client เพื่อเริ่มใช้งาน ซึ่งจะมีบล็อกไดอะแกรมดังรูปที่ 3.2

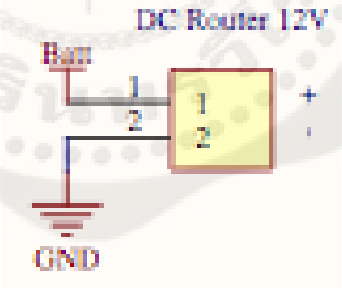


รูปที่ 3.2 บล็อกไดอะแกรม

3.1.1. Access Point Router

Access Point Router ทำหน้าที่เป็นตัวกลางในการเชื่อมต่อข้อมูล Computer Client กับ Arduino Board โดยในโครงงานนี้จะใช้ อุปกรณ์ Access Point Router ของ Linksys wrt54g ซึ่งจะมีคุณสมบัติในการทำหน้าที่เป็น Client Bridge เชื่อมต่อ computer กับ อุปกรณ์ ฮาร์ดแวร์ ซึ่งจะมีรูปแบบการต่อวงจรดังรูปที่ 3.4

รูปที่ 3.3 Access Point Rounter

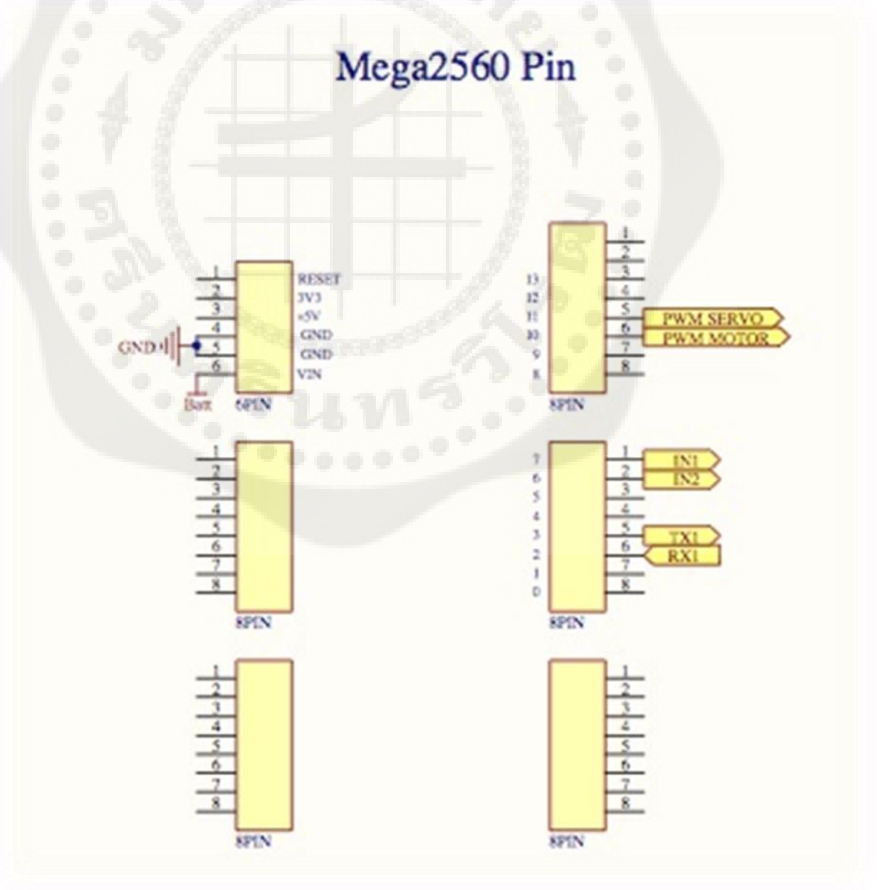


รูปที่ 3.4 การต่อวงจร Rounter

3.1.2. Arduino Board

Arduino Board ทำหน้าที่ในการประมวลผลข้อมูลการเคลื่อนที่และควบคุมชุดขับเคลื่อนรถบังคับให้ทำงานได้ตามคำสั่งอีกทั้งยังทำหน้าที่ในการควบคุมการทำงานของอุปกรณ์ระบุตำแหน่ง GPS อีกด้วย โดยจะมีวงจรการเชื่อมต่อดังรูปที่ 3.6

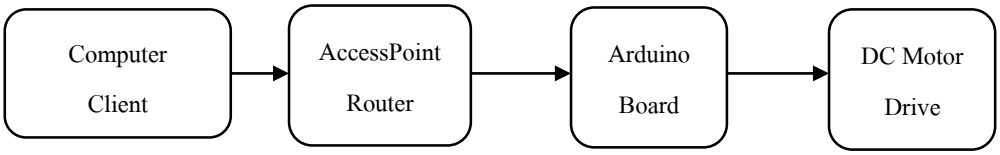
รูปที่ 3.5 ตัวอย่าง Arduino Board



รูปที่ 3.6 รูปแบบการเชื่อมต่อของ Arduino Board

3.1.3. DC Motor Drive Board

DC Motor Drive Board ทำหน้าที่ในการจ่ายกระแสไฟฟ้าให้ DC Motor เพื่อให้ DC Motor ทำงาน โดยจะรับ Input เป็นไฟ DC ขนาด 12V แบตเตอรี่ และทำการจ่ายกระแสให้ DC Motor และจะรูปแบบการทำงานตามรูปที่ 3.7

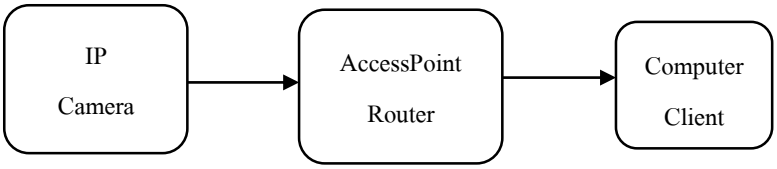


รูปที่ 3.7 บล็อกไดอะแกรมของ DC Motor Drive Board

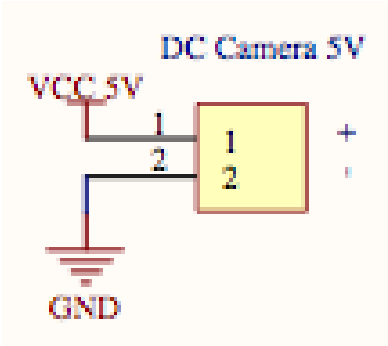
รูปที่ 3.8 DC Motor Driver Board

3.1.4. IP Camera *DC Motor Driver Board*

ทำหน้าที่ในการส่งข้อมูลภาพและเสียงจากรถบังคับแล้วนำไปแสดงบน User Interface โดย IP Camera จะเชื่อมต่อโดยตรงกับ Access Point Router ที่อยู่บนรถบังคับ ในโครงงานนี้จะเลือกใช้ IP Camera ของ D-link รุ่น DCS-942L ซึ่งจะมีคุณสมบัติในการส่งภาพและเสียงเมื่อในสภาวะที่มีแสงน้อยจะทำการปรับภาพเป็นโหลดกลางคืนโดยอัตโนมัติ จะมีรูปแบบการต่อวงจรดังรูปที่ 3.10



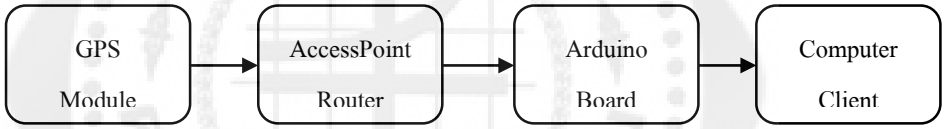
รูปที่ 3.9 บล็อกไดอะแกรมของ IP Camera



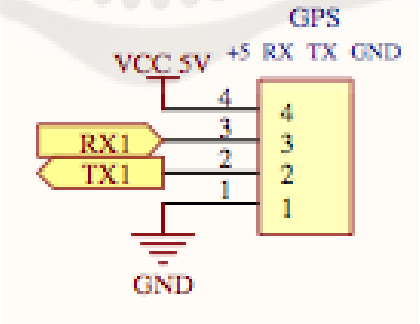
รูปที่ 3.10 วงจรIP Camera

3.1.5. อุปกรณ์ระบุตำแหน่ง GPS

GPS ทำหน้าที่ในการส่งข้อมูลพิกัดตำแหน่งของรถบังคับแล้วส่งข้อมูลไปที่ Arduino Board เพื่อนำไปแสดงผลใน User Interface จะมีรูปแบบการต่ออุปกรณ์ดังรูปที่ 3.12



รูปที่ 3.11 บล็อกไดอะแกรมของ GPS Module



รูปที่ 3.12 การต่อวงจร GPS

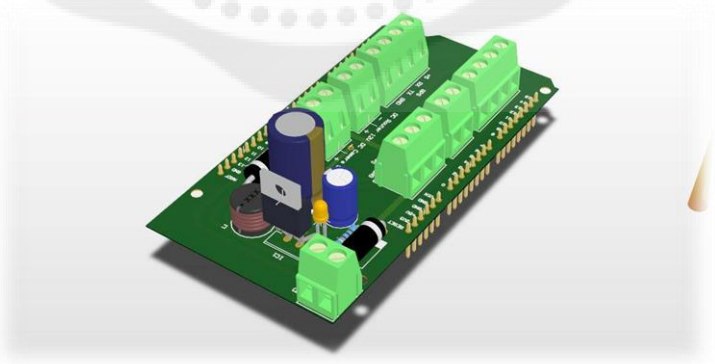
3.1.6. LIPO 3s. Battery

แบตเตอรี่ทำหน้าที่ในการจ่ายพลังงานให้กับอุปกรณ์ฮาร์ดแวร์ทั้งหมดใช้งานจำนวน 1 ก้อน โดยจะต่อเข้าโดยตรงกับวงจรแปลงกระแส มีขนาด 11.1 V 2200 mAh

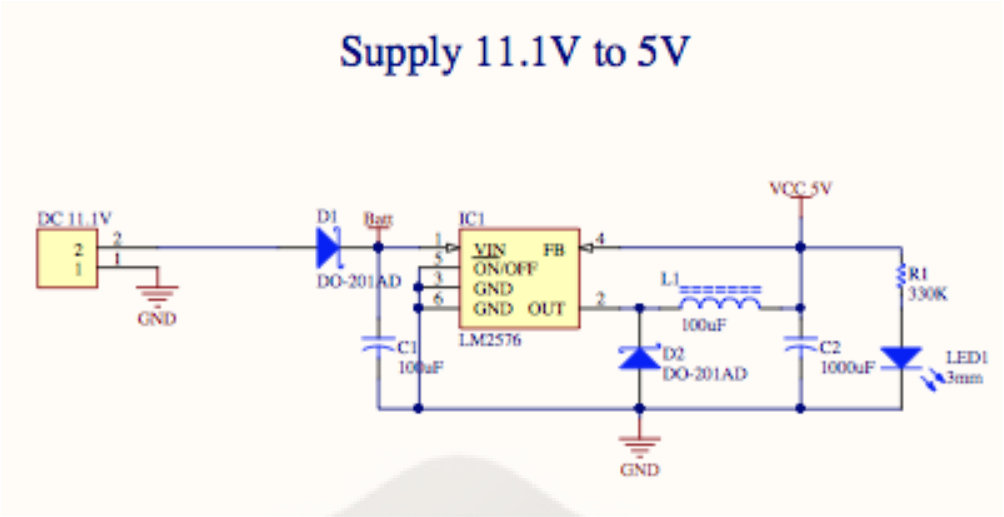
รูปที่ 3.13 รูป LIPO 3s. Battery

3.1.7. วงจรแปลงกระแส*รูป LIPO 3s. Battery*

วงจรแปลงกระแสทำหน้าที่ในการแปลงกระแส Battery จาก 11.1 เป็น 5V. เพื่อจ่ายให้กับฮาร์ดแวร์ส่วนต่าง ๆ โดยจะมีรูปตัวอย่างอุปกรณ์และแบบการต่อวงจรดังรูปที่ 3.5 และ รูปที่ 3.6 ตามลำดับ



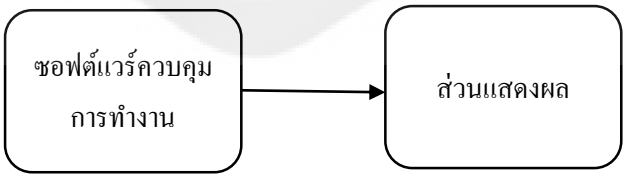
รูปที่ 3.14 วงจรแปลงกระแส



รูปที่ 3.15 รูปแบบการต่อวงจรแปลงกระแส

3.2 หลักการทำงานด้านซอฟต์แวร์

โดยในส่วนของ ซอฟต์แวร์นั้นจะถูกออกแบบให้แบ่งออกเป็น 2 ส่วน คือ ส่วนซอฟต์แวร์ควบคุมการทำงานของ Arduino Board และ ส่วนแสดงผล ทั้งสองส่วนจะทำงานสอดคล้องกัน เมื่อได้รับค่า Input ซอฟต์แวร์คุบคุมการทำงานของ Arduino Board จะสั่งการทำงานของรถบังคับให้เคลื่อนที่ และจะ ส่งค่า output เป็น ภาพและเสียงจากกล้อง และ ตำแหน่งพิกัด GPS ของรถลงบน User Interface ซึ่งจะแสดงตามบล็อกไดอะแกรมที่ 3.16



รูปที่ 3.16 บล็อกไดอะแกรมของซอฟต์แวร์

3.2.1. ซอฟต์แวร์ควบคุมการทำงานของ Arduino Board

ซอฟต์แวร์ควบคุมการทำงานของ Arduino Board จะถูกออกแบบขึ้นเพื่อควบคุมการทำงานของ DC Motor และ Servo Motor ให้สอดคล้องกับ การกดปุ่มทิศทางบน Jokystick ยกตัวอย่างเช่น กดปุ่มทิศทางด้านซ้ายให้ Servo Motor หมุนล้อรถบังคับไปทางด้านซ้าย กดปุ่มทิศทางด้านขวาให้ Servo Motor หมุนล้อรถบังคับไปทางด้านขวา เป็นต้น

ในส่วนของการสร้างซอฟต์แวร์ควบคุมจะใช้โปรแกรม Arduino Software โดยจะใช้ภาษา c++ ในการเขียนชุดคำสั่งควบคุมการทำงาน Arduino Board

3.2.2. ซอฟต์แวร์ในส่วนของการแสดงผล

ซอฟต์แวร์ในส่วนของการแสดงผลจะถูกออกแบบขึ้นเพื่อแสดงข้อมูลภาพและเสียงพร้อมทั้งตำแหน่ง ของ รถบังคับตามพิกัด GPS ซึ่งจะถูกแสดงผ่าน Google map อีกทั้งยังแสดงในส่วนของตัวแปรที่เกิดจากการรับและส่งข้อมูล ระหว่าง Computer Client กับ Arduino Board

ในส่วนของการ สร้างซอฟต์แวร์ในส่วนของการแสดงผลจะใช้โปรแกรม Visual Basic 2010 ในรูปแบบของ Window Form และใช้รูปแบบของการส่งข้อมูลแบบ TCP/IP โดยผ่านการเชื่อมต่อแบบการใช้สัญญาณไร้สาย

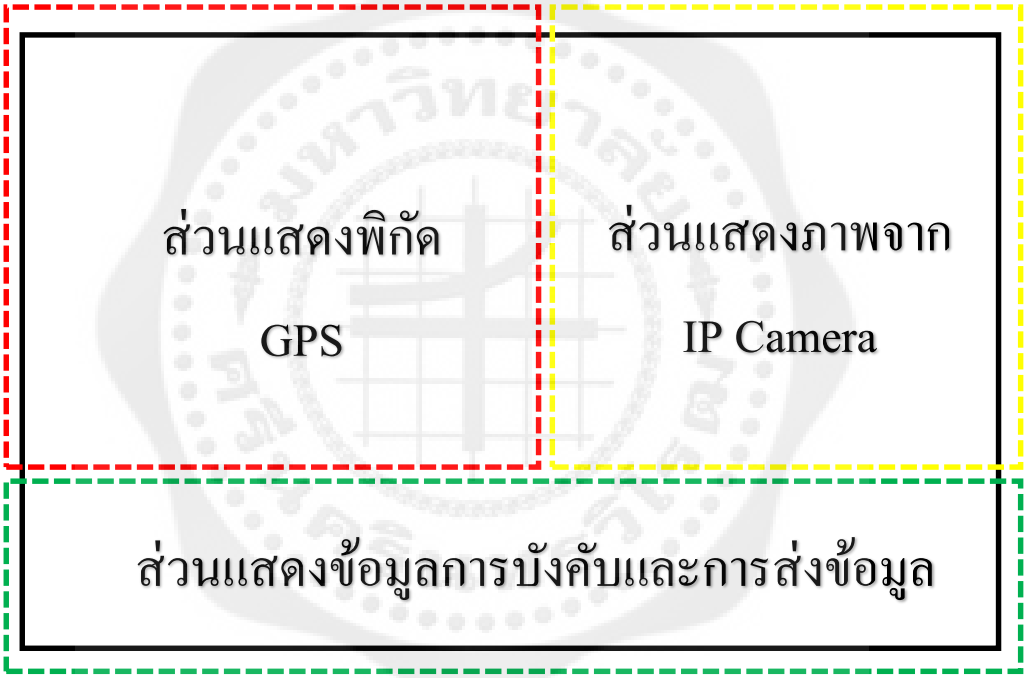
3.3 การออกแบบในส่วนแสดงผล

โดยในส่วนของการแสดงผลนั้นจะถูกออกแบบให้แบ่งออกเป็น 3 ส่วน คือ ส่วนของการแสดงตำแหน่งพิกัดของรถบน Googlemap ส่วนที่สองเป็นส่วนแสดงภาพจากกล้อง IP Camera ที่ติดอยู่หน้ารถบังคับ ส่วนที่สามเป็นส่วนของการแสดงข้อมูลจากการกด Joystick ควบคุม และ ข้อมูลจากการส่งค่าตัวแปรจาก Computer Client ไปที่ Board Arduino การทำงานจะแสดงผลเมื่อเปิดโปรแกรมควบคุมรถบังคับและเปิดสวิตซ์การทำงานของรถบังคับ การวางรูปแบบในส่วนต่างๆ เน้นเพื่อความสะดวกต่อการใช้งาน โดยจะมีตำแหน่งการออกแบบดังนี้

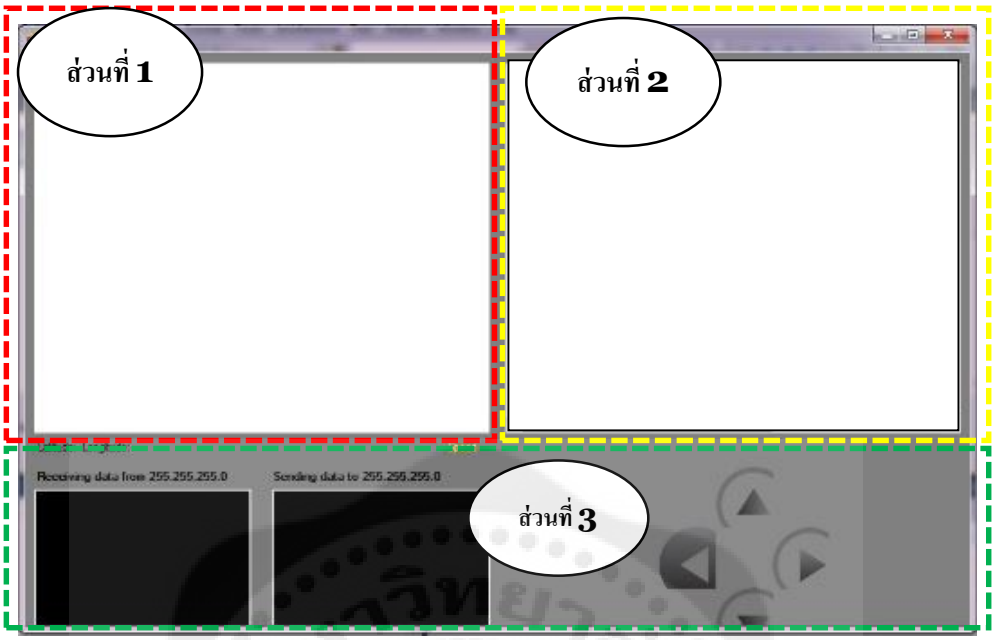
1. ส่วนตำแหน่งแสดงพิกัด GPS จะแสดงอยู่ส่วนด้านซ้ายของหน้าต่างโปรแกรมควบคุมรถบังคับ ซึ่งการทำงานจะแสดงเป็นจุดกระพริบตำแหน่งปัจจุบันของรถบังคับบน Googlemap และจุดกระพริบจะเคลื่อนที่เมื่อรถทำการวิ่งตามตำแหน่งที่เคลื่อนที่ไปด้วยเช่นกัน

2. ส่วนแสดงภาพจาก IP Camera จะแสดงอยู่ส่วนด้านขวามือ ของหน้าต่างโปรแกรมควบคุมรถบังคับ ซึ่งการทำงานจะแสดงภาพเคลื่อนไหวจากกล้องที่ติดอยู่บริเวณหน้ารถบังคับ และจะทำการปรับเป็นภาพโหมดกลางคืนโดยอัตโนมัติเมื่ออยู่ในบริเวณที่มีแสงน้อย

3. ส่วนแสดงการควบคุมและการส่งข้อมูล จะแสดงอยู่ส่วนด้านล่าง ของหน้าต่างโปรแกรมควบคุมรถบังคับ ซึ่งการทำงานจะแสดงเป็นปุ่มกระพริบ เมื่อทำการกดปุ่มทิศทางจาก Joystick และอีกส่วนจะเป็นการแสดงการส่งค่าตัวแปรที่ถูกกำหนดค่าในโปรแกรม ซึ่งรูปตัวอย่างหน้าต่างการออกแบบของส่วนแสดงผลจะมีรูปแบบตามรูปที่ 3.17



รูปที่ 3.17 การออกแบบหน้า User Interface



รูปที่ 3.18 หน้า User Interface

จากรูปที่ 3.18 จะแบ่งออกเป็น 3 ส่วนมีรายละเอียดดังนี้

- ส่วนที่ 1 แสดงข้อมูล ตำแหน่งของรถบน Google Map และ พิกัด Latitue-Longtitue การออกแบบจะใช้ Google Mmapap API ในการสร้างรูปแบบแผนที่และใช้การ ใช้การดึง ภาพจาก Google อุปกรณ์ระบุตำแหน่ง GPS ที่จะติดอยู่บน รถบังคับ การทำงานจะแสดงผลเมื่อเปิดโปแกรมควบคุมรถบังคับ
- ส่วนที่ 2 ภาพจากกล้อง IP Camera การออกแบบจะใช้การดึงภาพจาก IP Camera และจะแสดงภาพกล้องที่ติดอยู่บริเวณหน้ารถบังคับ การทำงานจะแสดงผลเมื่อเปิโปรแกรมควบคุมรถบังคับ
- ส่วนที่ 3 แสดงข้อมูลจากการ กดปุ่มทิศทางบน Joystick รวมถึงการส่งและรับตัวแปรระหว่าง Computer Client กับ Board Arduino โดยจะกระพริบเมื่อทำการกดปุ่มทิศทาง

บทที่ 4
การทดลองและผลการทดลองโครงงาน

การทดลองครั้งนี้มีวัตถุประสงค์เพื่อทราบถึงระยะที่รถเคลื่อนที่ไปได้ และเปรียบเทียบระยะทางจริงของรถที่เคลื่อนที่ได้กับระยะทางของรถที่เคลื่อนที่ได้บน Google Map เพื่อหาความคลาดเคลื่อนของระยะทางบน GPS และการบังคับรถบนสภาพพื้นผิวต่าง ๆ

4.1 การทดลองหาระยะการรับส่งสัญญาณ

การทดลองนี้ถูกออกแบบมาเพื่อทดสอบหาระยะทางที่สามารถรับส่งสัญญาณได้โดยการบังคับรถให้เคลื่อนที่เป็นเส้นตรงโดยปราศจากสิ่งกีดขวางในพื้นที่ที่มีสัญญาณรบกวนต่ำ เช่น สถานที่โล่งแจ้งและพื้นที่ที่มีสัญญาณรบกวนสูง เช่น ภายในอาคารที่มีคลื่นรบกวนจากอุปกรณ์ Access Point เป็นต้น ไปจนรถไม่สามารถรับสัญญาณต่อได้จำนวน 5 ครั้ง ดังตารางที่ 4.1

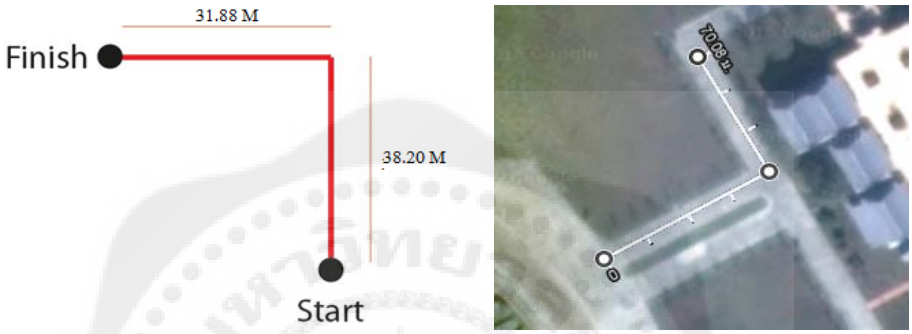
ตารางที่ 4.1 แสดงระยะทางที่ไกลที่สุดที่รถสามารถรับสัญญาณได้

ครั้งที่	สถานที่ที่มีคลื่นรบกวนต่ำ (เมตร)	สถานที่ที่มีคลื่นรบกวนสูง (เมตร)
1	70.9	67.8
2	71.2	66.3
3	74.3	71.1
4	71.8	69.4
5	71.9	68.4
เฉลี่ย	**72.0**	**68.6**

จากตารางการทดลองที่ 4.1 พบว่าระยะทางมากสุด ที่รถบังคับวิทยุสามารถรับและส่งค่าได้ในพื้นที่ที่มีคลื่นรบกวนต่ำ คือ 72 เมตร ส่วนระยะทางมากสุด ที่รถบังคับวิทยุสามารถรับและส่งค่าได้ในพื้นที่ที่มีคลื่นรบกวนสูง คือ 68.6 เมตร

4.2 การทดลองในส่วนของ GPS

การทดลองในส่วนนี้ถูกออกแบบมาเพื่อทดสอบโดยจะเริ่มจากการหาพิกัดและระยะทางของแต่ละจุดอ้างอิงเพื่อใช้เปรียบเทียบกับตำแหน่งบน GPS สำหรับทดลองวัดระยะทางหาความคลาดเคลื่อนของ GPS โดยใช้อุปกรณ์ที่ระบุพิกัดได้อย่างแม่นยำ



รูปที่ 4.1 แสดงตำแหน่งและระยะอ้างอิง

เมื่อได้จุดอ้างอิงดังรูปที่ 4.1 จะเริ่มการทดลองโดยไปเริ่มที่จุดอ้างอิงที่ 1 และบังคับรถไปจนพิกัดเท่ากับจุดอ้างอิงที่ 2 วัดระยะห่างระหว่างรถและจุดอ้างอิงที่ 2 บันทึกค่า แล้วนำรถไปวางไว้ที่จุดอ้างอิงที่ 2 บังคับรถต่อไปยังจุดพิกัดที่เท่ากับจุดอ้างอิงที่ 3 และวัดระยะห่างระหว่างรถและพิกัดจุดอ้างอิงที่ 3 และบันทึกผล ได้ผลดังตารางที่ 4.2 ซึ่งระยะทางที่ใช้อ้างอิงระหว่างจุดที่ 1 และ 2 คือ 38.20 เมตร และระยะทางที่ใช้อ้างอิงระหว่างจุดที่ 2 และ 3 คือ 31.88 เมตร โดยจะใช้วิธีการคำนวณหาค่าร้อยละของความคลาดเคลื่อนจากตำแหน่งอ้างอิง ใน GPS ดังสมการที่ 2.7

$$\frac{\text{ระทางที่วัดได้ - ระยะทางที่ใช้อ้างอิง}}{\text{ระยะทางที่ใช้อ้างอิง}} \times 100 \qquad (2.7)$$

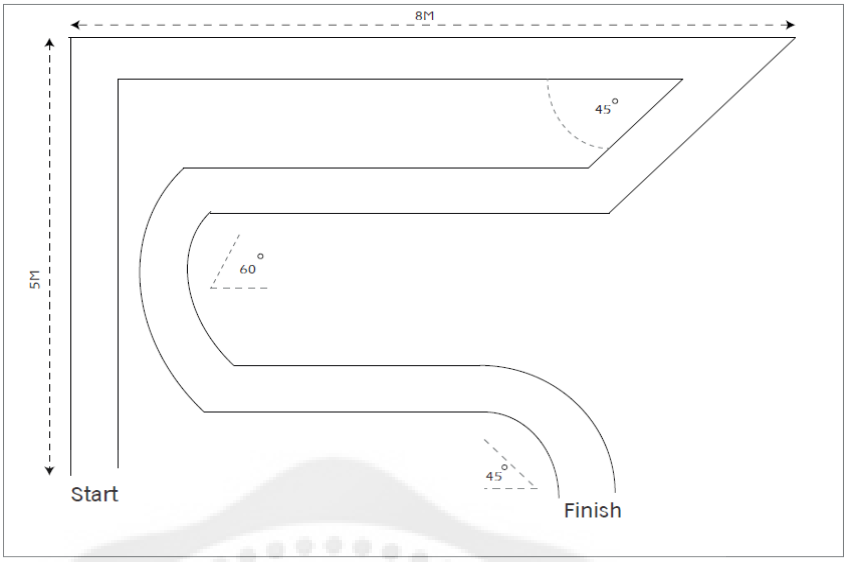
ตารางที่ 4.2 แสดงระยะทางจากจุดเริ่มต้นไปยังจุดอ้างอิง

รอบที่	ระยะทางระหว่างจุดอ้างอิง 1 และ 2 (เมตร)	ระยะทางระหว่างจุดอ้างอิง 2 และ 3 (เมตร)
1	38.85	33.44
2	37.90	33.23
3	39.32	32.20
4	38.92	34.12
5	39.14	33.96
เฉลี่ย	**38.83**	**33.39**

จากการทดลองจะเห็นได้ว่าเมื่อเปรียบเทียบค่าความคลาดเคลื่อนของ GPS กับระยะที่วัดได้จริงมีค่าเท่ากับ 3.05 %

4.3 การทดลองในส่วนของการควบคุมรถบนสภาพพื้นผิวต่าง ๆ

การทดลองในส่วนนี้มีวัตถุประสงค์เพื่อทดสอบการเลี้ยวรถ โดยรถจะเคลื่อนที่ไปตามลู่ที่วางไว้ตามแปลนทดสอบการเลี้ยวรถดังรูปที่ 4.2 บนสภาพพื้นผิวต่าง ๆ เพื่อทดสอบในการเลี้ยวรถว่าสามารถเลี้ยวผ่านโค้งตามมุมที่กำหนดไว้ได้หรือไม่ โดยถ้าสามารถเลี้ยวผ่านทางโค้งได้โดยไม่หลุดออกจากลู่จึงจะนับว่าสามารถบังคับรถผ่านการเลี้ยวที่มุมนั้น ๆ ได้



รูปที่ 4.2 แผนผังทดสอบการเลี้ยวของรถบังคับวิทยุ

การทดลองจะเริ่มโดยการบังคับรถให้เคลื่อนที่ไปบนลู่แบบพื้นผิวตัวอย่างต่าง ๆ ขนาดความกว้าง 40 ซม. เพื่อทดสอบการเข้าถึงพื้นที่ที่จำกัดความกว้างบนพื้นผิวตัวอย่าง ว่าที่พื้นผิวตัวอย่างสามารถเลี้ยวในมุมตัวอย่างได้หรือไม่ หากไม่สามารถเลี้ยวโค้งโดยไม่หลุดออกจากจากลู่ได้จะวัดค่าคลาดเคลื่อนจากการเปรียบเทียบกับความกว้างของลู่ เมื่อรถมีทิศทางขนานกับลู่โดยคำนวณจากสมการที่ 2.8

$$\frac{\text{ระทางที่วัดได้ - ระยะทางที่ใช้อ้างอิง}}{\text{ระยะทางที่ใช้อ้างอิง}} \times 100 \qquad (2.8)$$

รูปที่ 4.3 แสดงระยะห่างระหว่างขอบเส้นถึงขอบล้อ

รูปที่ 4.4 การทดลองขับเคลื่อนรถบนสภาพพื้นผิวที่ต่างกัน

ตารางที่ 4.3 แสดงการทดลองหาระยะความคลาดเคลื่อนของการเข้าโค้งแต่ละมุมบนพื้นกระเบื้อง

รอบที่	**ระยะความคลาดเคลื่อน (เซนติเมตร)**			
	45˚	**60˚**	**90˚**	**135˚**
1	0	0	7.2	15.6
2	0	0	8.3	17.8
3	0	0	7.5	14.9
4	0	0	7.9	15.8
5	0	0	8.5	16.2
เฉลี่ย	**0**	**0**	**7.9**	**16.1**

จากตารางที่ 4.3 จะเห็นว่าในการเลี้ยวโค้งที่มุม 45˚ และ 60˚ บนพื้นกระเบื้องนั้น สามารถเลี้ยวได้โดยที่ไม่มีความคลาดเคลื่อนเลย และที่มุม 90˚ มีความคลาดเคลื่อน 19.75% และที่มุม 135˚ มีความคลาดเคลื่อน 40.25%

ตารางที่ 4.4 แสดงการทดลองหาระยะความคลาดเคลื่อนของการเข้าโค้งแต่ละมุมบนพื้นหินกรวด

รอบที่	ระยะความคลาดเคลื่อน (เซนติเมตร)			
	45˚	**60˚**	**90˚**	**135˚**
1	0	0	7.0	16.2
2	0	0	7.4	15.4
3	0	0	8.3	17.1
4	0	0	7.9	15.8
5	0	0	7.4	16.5
เฉลี่ย	**0**	**0**	**7.6**	**16.2**

จากตารางที่ 4.4 จะเห็นว่าในการเลี้ยวโค้งที่มุม 45˚ และ 60˚ บนพื้นหินกรวดนั้น สามารถเลี้ยวได้โดยที่ไม่มีความคลาดเคลื่อนเลย และ ที่มุม 90˚ มีความคลาดเคลื่อน 19.00% และที่มุม 135˚ มีความคลาดเคลื่อน 40.50%

ตารางที่ 4.5 แสดงการทดลองหาระยะความคลาดเคลื่อนของการเข้าโค้งแต่ละมุมบนพื้นหญ้า

รอบที่	ระยะความคลาดเคลื่อน (เซนติเมตร)			
	45˚	**60˚**	**90˚**	**135˚**
1	0	0	8.0	18.2
2	0	0	8.9	17.9
3	0	0	8.3	18.9
4	0	0	9.1	17.1
5	0	0	8.5	19.1
เฉลี่ย	**0**	**0**	**8.6**	**18.2**

จากตารางที่ 4.5 จะเห็นว่าในการเลี้ยวโค้งที่มุม 45° และ 60° บนพื้นหญ้านั้น สามารถเลี้ยวได้โดยที่ไม่มีความคลาดเคลื่อนเลย และ ที่มุม 90° มีความคลาดเคลื่อน 21.50% และที่มุม 135° มีความคลาดเคลื่อน 45.50%

ดังนั้นในการทดสอบการเคลื่อนที่เข้าโค้งของรถบังคับวิทยุ จากทั้ง 3 ตารางจะเห็นได้ว่าการเข้าโค้งในช่วง 45° และ 60° สามารถผ่านไปได้โดยไม่หลุดโค้งเหมือนกันทั้ง 3 สภาพพื้นผิว และจะเริ่มมีค่าความคลาดเคลื่อนที่โค้ง มุม 90° และ 135° ซึ่งจากการทดลองข้างต้นจะเห็นได้ว่ายิ่งที่ทางโค้งของทุกสภาพพื้นผิวมีมุมเยอะเท่าใด จะทำให้ค่าความผิดพลาดเพิ่มมากขึ้นตามไปด้วย และในแต่ละสภาพพื้นผิวมีค่าความผิดพลาดที่ต่างกันอยู่เพียงเล็กน้อยเท่านั้น

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

โครงงานการพัฒนาโปรแกรมควบคุมรถบังคับวิทยุผ่านการสื่อสารไร้สายแบบวายฟายโดยใช้การกระจายข้อมูลแบบเกณฑ์วิธีควบคุมการส่งผ่านมีวัตถุประสงค์เพื่อใช้รถบังคับวิทยุ ในการสำรวจพื้นที่อันตรายและเข้าถึงได้ยาก เพื่อสามารถควบคุมรถบังคับวิทยุมีความถูกต้อง แม่นยำ ผ่านทางระบบ Wi-Fi และ GPS สามารถควบคุมรถบังคับวิทยุได้ในสภาพผิวที่หลากหลาย โดยแบ่งการทดลองออกเป็น 3 ส่วน คือ การทดลองหาระยะการรับส่งสัญญาณ การทดลองในส่วนของ GPS และ การทดลองในส่วนของการควบคุมรถบนสภาพพื้นผิวต่าง ๆ จากการทดลองหาระยะการรับส่งสัญญาณผลปรากฏว่าระยะทางมากสุด ที่รถบังคับวิทยุสามารถรับและส่งค่าได้ในพื้นที่ที่มีคลื่นรบกวนต่ำ คือ 72 เมตร ส่วนระยะทางมากสุด ที่รถบังคับวิทยุสามารถรับและส่งค่าได้ในพื้นที่ที่มีคลื่นรบกวนสูง คือ 68.6 เมตร จากการทดลองในส่วนของ GPS ผลปรากฏว่าเมื่อเปรียบเทียบค่าความคลาดเคลื่อนของ GPS กับระยะที่วัดได้จริง มีค่าความคลาดเคลื่อนเท่ากับ 3.05 % จากการทดลองในส่วนของการควบคุมรถบนสภาพพื้นผิวต่าง ๆ ผลปรากฏว่าการเข้าโค้งในช่วง 45° และ 60° สามารถผ่านไปได้โดยไม่หลุดโค้งเหมือนกันทั้ง 3 สภาพพื้นผิว และจะเริ่มมีค่าความคลาดเคลื่อนที่โค้ง มุม 90° และ 135° ซึ่งจากการทดลองจะได้ว่ายิ่งที่ทางโค้งของทุกสภาพพื้นผิวมีมุมเยอะเท่าใด จะทำให้ค่าความผิดพลาดเพิ่มมากขึ้นตามไปด้วย และในแต่ละสภาพพื้นผิวมีค่าความผิดพลาดที่ต่างกันอยู่เพียงเล็กน้อยเท่านั้น

5.2 ปัญหาและข้อเสนอแนะในโครงงาน

จากการทดลองในกรณีที่พื้นที่อันตรายหรือพื้นที่เข้าถึงได้ยากมีระยะทางที่ไกลกว่าที่สัญญาณจะสามารถรับส่งได้ทำให้การบังคับเกิดอาการดีเลย์หรือไม่สามารถบังคับรถให้ไปต่อได้ ในโครงงานนี้สามารถที่จะเปลี่ยนตัวรับส่งสัญญาณที่มีคุณภาพสูงกว่าเพื่อสามารถรับส่งสัญญาณไปได้ไกลขึ้นและแม่นยำกว่าเดิม และสำหรับพื้นที่สภาพพื้นผิวอื่น ๆ ซึ่งโครงสร้างของรถในรูปแบบปัจจุบันอาจไม่อำนวยต่อการเคลื่อนที่ในพื้นที่ผิวนั้น ๆ ซึ่งสามารถที่จะปรับเปลี่ยนโครงรถให้เหมาะสมกับสภาพพื้นผิวนั้น ๆได้ และหากต้องการให้ GPS ระบุพิกัดให้แม่นยำมากขึ้นก็สามารถเปลี่ยน GPS เป็นรุ่นที่มีประสิทธิภาพสูงกว่านี้เพื่อให้เร็วต่อการจับสัญญาณและแม่นยำได้มากยิ่งขึ้น

เอกสารอ้างอิง

คอม-ทีเอช ดอทเน็ต. (2552). **ระบบเครือข่ายไร้สาย (Wireless LAN Technology).** วันที่ค้นข้อมูล 23 เมษายน 2558, จาก ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร กรมการแพทย์ กระทรวงสาธารณสุข. เว็บไซต์: http://www.dms.moph.go.th/dmsict/#

ชวลิต ทินกรสูติบุตร และคณะ. (2551). **ความรู้พื้นฐานเกี่ยวกับ โปรโตคอล TCP/IP.** วันที่ค้นข้อมูล 17 มกราคม 2558, จาก บริษัท ที-เน็ต จำกัด เว็บไซต์: http://www.tnetsecurity.com/content_basic/tcp_ip_knowledge.php

ซิสทูยู. (2554). อุปกรณ์ Wireless-G Broadband Router. 12 เมษายน 2558, จาก Basisware Systems เว็บไซต์: http://www.sys2u.com/product.php?ProductID=20070806-174415

ณัฐชนน สัตยานนท์, และกฤษดา สุภัทธาพิศาล. (2556). **การออกแบบระบบวายฟายภายในอาคารสำหรับอาคารสำนักงาน.** ปริญญานิพนธ์ วศ.บ. (ไฟฟ้า) กรุงเทพฯ : บัณฑิตวิทยาลัย. มหาวิทยาลัยศรีนครินทรวิโรฒ

ทศพล ลิ้มทอง, บัณฑิต ท่าดี และออมสิน ศรีสุรินทร์. (2550). **ระบบระบุตำแหน่งอุปกรณ์คอมพิวเตอร์ภายในอาคารโดยใช้เครื่อข่ายไร้สาย.** ปริญญานิพนธ์ วศ.บ. (ไฟฟ้า) กรุงเทพฯ : บัณฑิตวิทยาลัย. มหาวิทยาลัยศรีนครินทรวิโรฒ

ธีรวัฒน์ ประกอบผล. (2552). **ควบคุมอุปกรณ์ฮาร์ดแวร์ Advanced Visual Basic Version 6.0.** พิมพ์ครั้งที่ 6. กรุงเทพฯ: รีไวว่า.

บริษัทโกลเบิ้ล 5 จำกัด. (2549). **ความรู้ทั่วไปเกี่ยวกับ GPS.** วันที่ค้นข้อมูล 28 เมษายน 2558, จาก บริษัท โกลเบิ้ล 5 จำกัด เว็บไซต์: http://www.global5thailand.com

โอภาส ศิริครรชิตถาวร, วรพจน์ กรแก้ววัฒนกุล และชัยวัฒน์ ลิ้มพรจิตร. (2537). **เรียนรู้ระบบควบคุมอย่างง่ายด้วยโปรแกรมภาษาซีกับอาดุยโน่ และบอร์ดไมโครคอนโทรลเลอร์.** กรุงเทพฯ: อินโนเวทีฟ เอ็กเพอริเมนต์.

ภาคผนวก

ภาคผนวก ก ตัวอย่าง Code โปรแกรม

ส่วนของโปรแกรม Arduino

การประกาศ Library และ ตัวแปร

```
1. #include <SPI.h>
2. #include <Ethernet.h>
3. #include <Servo.h>
4. #define FORWARD      7
5. #define BACKWARD     6
6. #define PWM_MOTOR    10
7. #define PWM_SERVO    11
8. #define BUFFSIZ_ALL 300
9. #define BUFFSIZ_RMC 90
10. #define BUFFSIZ_MSG 50
11. Servo myServo;
```

กำหนดค่า IP Address และ Protocol ของอุปกรณ์

```
1. byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
2. IPAddress ip(192,168,1, 188);
3. IPAddress gateway(192,168,1, 1);
4. IPAddress subnet(255, 255, 255, 0);
5. char *parseptrVb;
6. EthernetServer server(8888);
7. boolean alreadyConnected = false;
```

การคัดแยก Latitude , Longitude และ Speed จากค่า NMEA

```
1.  int byteGPS = -1;
2.  char linea[BUFFSIZ_ALL] = "";
3.  const char comandoGPR[7] = "$GPRMC";
4.  int bien = 0;
5.  int conta = 0;
6.  char msgBuffer[BUFFSIZ_MSG] = "";
7.  const char comandoMSG[5] = "$TX:";
8.  int msgCount = 0;
9.  char thisChar;
10. char buffer[BUFFSIZ_RMC];
11. char *parseptr;
12. char buffidx;
13. unsigned long int latitude, longitude;
14. char state;
15. unsigned int groundspeed;
16. String gpsMessage = "$RX,100.982212,14.103344,50,0\n";
17. char gpsBuffer[BUFFSIZ_MSG] = "";
18. ชุดคำสั่งควบคุม PWM  ซึ่งจะรับค่าคำสั่งมาจาก VB
19. void setup() {
20. Ethernet.begin(mac, ip, gateway, subnet);
21. server.begin();
22. myServo.attach(PWM_SERVO);
23. pinMode(FORWARD, OUTPUT);
24. pinMode(BACKWARD, OUTPUT);
25. Serial.begin(9600);
26. for(int i=0; i<BUFFSIZ_ALL; i++){       // Initialize a buffer for received data
27. linea[i] = ' ';
```

```
28. }
29. for(int i=0; i<BUFFSIZ_RMC; i++){       // Initialize a buffer for received data
30. buffer[i] = ' ';
31. }
32. delay(5000);
33. }
34. /**
35. Main loop
36. */
```

กำหนด Client และ Server

```
1.  void loop() {
2.  EthernetClient client = server.available();
3.  if (client.connected()) {
4.  if (!alreadyConnected) {
5.  client.flush();
6.  client.print("Welcome to Robot");
7.  alreadyConnected = true;
8.  }
9.  if (client.available() > 0) {
10. thisChar = client.read();
11. if (thisChar == '$') {
12. for (int i=0; i<7; i++) {
13. msgBuffer[i] = thisChar;
14. if (msgBuffer[i] == 'E') {
15. client.flush();
16. decodeMessage(msgBuffer);
```

```
17. ReadGPS();
18. delay(50);
19. gpsMessage.toCharArray(gpsBuffer, 50);
20. server.write(gpsBuffer);
21. continue;
22. }
23. thisChar = client.read();
24. if (thisChar == -1) {
25. continue;
26. }
27. }
28. }
29. }
30. }
31. }
```

ชุดคำสั่งควบคุม Servo และ PWM

```
1.  void decodeMessage(char* packetBuffer)
2.  {
3.  //Serial.println(packetBuffer);
4.  // Check if the header message is $TX (VB message)
5.  if(strncmp(packetBuffer, "$TX", 3) == 0)
6.  {
7.  // Read data from Arduino
8.  parseptrVb = packetBuffer+4;
9.  if(parseptrVb[0] == 'F') {
10. analogWrite(PWM_MOTOR, 255);  // 0 - 255
```

```
11. digitalWrite(FORWARD, HIGH);
12. digitalWrite(BACKWARD, LOW);
13. }else if(parseptrVb[0] == 'B') {
14. analogWrite(PWM_MOTOR, 255);
15. digitalWrite(FORWARD, LOW);
16. digitalWrite(BACKWARD, HIGH);
17. }else if(parseptrVb[0] == '0') {
18. analogWrite(PWM_MOTOR, 0);
19. digitalWrite(FORWARD, LOW);
20. digitalWrite(BACKWARD, LOW);
21. }
22. if(parseptrVb[1] == '9') {
23. myServo.write(60);
24. }else if(parseptrVb[1] == '8') {
25. myServo.write(70);
26. }else if(parseptrVb[1] == '7') {
27. myServo.write(80);
28. }else if(parseptrVb[1] == '6') {
29. myServo.write(90);
30. }else if(parseptrVb[1] == '5') {
31. myServo.write(96);
32. }else if(parseptrVb[1] == '4') {
33. myServo.write(100);
34. }else if(parseptrVb[1] == '3') {
35. myServo.write(110);
36. }else if(parseptrVb[1] == '2') {
37. myServo.write(120);
38. }else if(parseptrVb[1] == '1') {
```

```
39. myServo.write(130);
40. }
41. }
42. }
43. /**
    a.Read GPS data from uart port
44. */
45. void ReadGPS()
46. {
47. while(1)
48. {
49. byteGPS = Serial.read();    // Read a byte of the serial port
50. if (byteGPS == -1) {         // See if the port is empty yet
51. return;
52. } else {
53. // note: there is a potential buffer overflow here!
54. linea[conta] = byteGPS;       // If there is serial port data, it is put in the buffer
55. conta++;
56. if(byteGPS == 13) {   // If the received byte is = to 13 ( <CR> or \r), end of transmission
57. // note: the actual end of transmission is <CR><LF> (i.e. 0x13 0x10)
58. bien = 0;
59. for(int i=1; i<7; i++)
60. {
61. if(linea[i] == comandoGPR[i-1]) {  // Checking $GPRMC header
62. bien++;
63. }
64. }
65. if(bien == 6) {      // If yes, continue and process the data
```

```
66. for(int i=1; i<BUFFSIZ_RMC; i++)
67. {
68. if(linea[i] == 13)
69. {
70. buffer[i-1] = 0;
71. decodeGPS(buffer);
72. conta=0;                    // Reset the buffer
73. for(int i=0; i<BUFFSIZ_ALL; i++)
74. {
75. linea[i] = ' ';
76. }
77. return;
78. }else{
        a. //Serial.write(linea[i]);
        b. buffer[i-1] = linea[i];
79. }
80. }
81. }
82. conta=0;                    // Reset the buffer
83. for(int i=0; i<BUFFSIZ_ALL; i++)
84. {
85. linea[i] = ' ';
86. }
87. }
88. }
89. }
90. }
91. /**
```

a. Decode GPS message from buffer

```
92. */
93. void decodeGPS(char* gps)
94. {
95. //Serial.write(gps);
96. if(strncmp(gps, "$GPRMC", 6) == 0) {
97. // hhmmss time data
98. parseptr = gps+7;
99. // read status V/A data
100.        parseptr = strchr(parseptr, ',') + 1;
101.        state = parseptr[0];
102.        parseptr += 2;
103.        if(state == 'A') {
104.        // latitude
105.        latitude = parsedecimal(parseptr);
106.        if (latitude != 0) {
107.        latitude *= 10000;
108.        parseptr = strchr(parseptr, '.')+1;
109.        latitude += parsedecimal(parseptr);
110.        }
111.        parseptr = strchr(parseptr, ',')+1;
112.        // longitude
113.        parseptr = strchr(parseptr, ',')+1;
114.        longitude = parsedecimal(parseptr);
115.        if (longitude != 0) {
116.        longitude *= 10000;
117.        parseptr = strchr(parseptr, '.')+1;
118.        longitude += parsedecimal(parseptr);
```

```
119.        }
120.        parseptr = strchr(parseptr, ',')+1;
121.        // groundspeed
122.        parseptr = strchr(parseptr, ',')+1;
123.        groundspeed = parsedecimal(parseptr);
124.        groundspeed = (groundspeed*1.852); // speed in Km/hr (1 Knot/hr = 1.852
    Km/hr)
125.        gpsMessage = "";
126.        // Header
127.        gpsMessage += "$RX,";
128.        // Longitude
129.        gpsMessage += String(longitude/1000000); gpsMessage += ".";
130.        gpsMessage += String((longitude/10000)%100);
131.        gpsMessage += String((longitude%10000)*6/1000);
132.        gpsMessage += String(((longitude%10000)*6/10)%100);
133.        gpsMessage += ",";
134.        // Latitude
135.        gpsMessage += String(latitude/1000000); gpsMessage += ".";
136.        gpsMessage += String((latitude/10000)%100);
137.        gpsMessage += String((latitude%10000)*6/1000);
138.        gpsMessage += String(((latitude%10000)*6/10)%100);
139.        gpsMessage += ",";
140.        // Speed
141.        gpsMessage += String(groundspeed);
142.        gpsMessage += ",";
143.        gpsMessage += "0";
144.        gpsMessage += "\n";
145.        }
```

```
146.        }
147.        }
148.        /**
        a.  Convert char array to decimal
149.        */
150.        unsigned long int parsedecimal(char *str) {
151.        unsigned long int d = 0;
152.        while (str[0] != 0) {
153.        if ((str[0] > '9') || (str[0] < '0')) {
154.        return d;
155.        }
156.        d *= 10;
157.        d += str[0] - '0';
158.        str++;
159.        }
160.        return d;
161.        }
```

ส่วนของโปรแกรม Microsoft Visual Basic

```
1. Imports System
2. Imports System.IO
3. Imports System.Text
4. Imports System.Net
5. Imports System.Net.Sockets
6. Imports System.Threading
7. Imports System.Runtime.InteropServices
8. Public Class formRobot
9. Dim IP As IPAddress
10. Dim Port As Integer = 8888
11. Public TcpClient As TcpClient
12. Public Tcplistener As TcpListener
13. Public Stream As NetworkStream
14. Public LocalName As String = String.Empty
15. Dim sw As StreamWriter
16. Dim longitud As String
17. Dim latitud As String
18. Dim rlStr As String = "0"
19. Dim bfStr As String = "0"
20. Declare Function joyGetPosEx Lib "winmm.dll" (ByVal uJoyID As Integer, ByRef pji
    As JOYINFOEX) As Integer
21. <StructLayout(LayoutKind.Sequential)>
22. Public Structure JOYINFOEX
23. Public dwSize As Integer
24. Public dwFlags As Integer
25. Public dwXpos As Integer
26. Public dwYpos As Integer
```

```
27. Public dwZpos As Integer
28. Public dwTpos As Integer
29. Public dwUpos As Integer
30. Public dwVpos As Integer
31. Public dwButtons As Integer
32. Public dwButtonNumber As Integer
33. Public dwPOV As Integer
34. Public dwReserved1 As Integer
35. Public dwReserved2 As Integer
36. End Structure
37. Dim Joypos1 As JOYINFOEX
38. Private Sub formRobot_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
39. Joypos1.dwSize = 64
40. Joypos1.dwFlags = &HFF
41. Try
42. wbCamera.ScriptErrorsSuppressed = True
43. wbGooglemap.ScriptErrorsSuppressed = True
44. ConnectToRobot()
45. wbCamera.Navigate("http://192.168.1.177")
46. Catch ex As Exception
47. Console.WriteLine(ex.Message)
48. End Try
49. TimerGPS.Interval = 5000
50. TimerGPS.Enabled = True
51. End Sub
52. Public Sub ConnectToRobot()
53. Try
```

54. TcpClient = New TcpClient()
55. IP = IPAddress.Parse("192.168.1.188")
56. TcpClient.Connect(IP, Port)
57. If TcpClient.Connected = True Then
    a. ShowMessageReceive("Connected Successfully to " & IP.ToString & " : " & Port & Environment.NewLine)
    b. LabelReceiveIP.Text = "Receiving from : " & IP.ToString
    c. LabelSendIP.Text = "Sending to : " & IP.ToString
    d. ReadTimer.Enabled = True
58. End If
59. Catch ex As SocketException
60. Console.WriteLine("Error: " & ex.Message)
61. Console.WriteLine("Failure to Connect to " & IP.ToString & " : " & Port)
62. End Try
63. End Sub
64. Public Sub ReadData(ByVal TcpClient As TcpClient)
65. Try
66. Dim bytes(10024) As Byte
67. Dim Data As String
68. Dim FinalString As String
69. Stream = TcpClient.GetStream()
70. If Stream.CanRead = True Then
    a. If Stream.DataAvailable = True Then
    b. Data = Stream.Read(bytes, 0, bytes.Length)
    c. FinalString = Encoding.ASCII.GetString(bytes)
    d. DecodeMsg(FinalString)
    e. ShowMessageReceive("Recieved : " & FinalString)
    f. 'Stream.Flush()

```
        g. End If
71. End If
72. Catch ex As SocketException
73. Console.WriteLine("Error: " & ex.Message)
74. End Try
75. End Sub
76. Public Sub DecodeMsg(ByVal msgRx As String)
77. Dim separators() As String = {","}
78. Dim words() As String = msgRx.Split(separators,
    StringSplitOptions.RemoveEmptyEntries)
79. If String.Compare(words(0), 0, "$RX", 0, 3, True) = 0 Then
80. longitud = words(1)
81. latitud = words(2)
82. LabelPosition.Text = "Longitude: " & longitud & ", " & "Latitude: " & latitud
83. LabelSpeed.Text = "Speed : " & words(3) & " Km/Hr"
84. End If
85. End Sub
86. Public Sub SendData(ByVal TextToSend1 As String)
87. Try
88. Stream = TcpClient.GetStream()
89. Dim Bytes As String = TextToSend1
90. Dim BytesToWrite As Byte() = Encoding.ASCII.GetBytes(Bytes)
91. If Stream.CanWrite = True Then
        a. Stream = TcpClient.GetStream
        b. Stream.Write(BytesToWrite, 0, BytesToWrite.Length)
        c. ShowMessageSend("Sended : " & TextToSend1)
92. Else
        a. ShowMessageSend("***Could Not Send : " & TextToSend1 & " ***")
```

```
93. End If
94. Catch ex As SocketException
95. Console.WriteLine("Error: " & ex.Message)
96. End Try
97. End Sub
98. Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Timer1.Tick
99. joyGetPosEx(0, Joypos1)
100.        Try
101.        If Joypos1.dwXpos >= 65530 Then
        a. rlStr = "9"
        b. PictureBoxRight.Image = My.Resources.button_right_on
        c. PictureBoxLeft.Image = My.Resources.button_left_off
102.        ElseIf Joypos1.dwXpos >= 57279 And Joypos1.dwXpos <= 65530 Then
        a. rlStr = "8"
        b. PictureBoxRight.Image = My.Resources.button_right_on
        c. PictureBoxLeft.Image = My.Resources.button_left_off
103.        ElseIf Joypos1.dwXpos >= 49023 And Joypos1.dwXpos <= 57279 Then
        a. rlStr = "7"
        b. PictureBoxRight.Image = My.Resources.button_right_on
        c. PictureBoxLeft.Image = My.Resources.button_left_off
104.        ElseIf Joypos1.dwXpos >= 40767 And Joypos1.dwXpos <= 49023 Then
        a. rlStr = "6"
        b. PictureBoxRight.Image = My.Resources.button_right_on
        c. PictureBoxLeft.Image = My.Resources.button_left_off
105.        ElseIf Joypos1.dwXpos >= 32000 And Joypos1.dwXpos <= 33000 Then
    'Original center
        a. rlStr = "5"
```

b. PictureBoxRight.Image = My.Resources.button_right_off

c. PictureBoxLeft.Image = My.Resources.button_left_off

106. ElseIf Joypos1.dwXpos <= 24255 And Joypos1.dwXpos >= 15999 Then

a. rlStr = "4"

b. PictureBoxRight.Image = My.Resources.button_right_off

c. PictureBoxLeft.Image = My.Resources.button_left_on

107. ElseIf Joypos1.dwXpos <= 15999 And Joypos1.dwXpos >= 7743 Then

a. rlStr = "3"

b. PictureBoxRight.Image = My.Resources.button_right_off

c. PictureBoxLeft.Image = My.Resources.button_left_on

108. ElseIf Joypos1.dwXpos <= 7743 And Joypos1.dwXpos >= 5 Then

a. rlStr = "2"

b. PictureBoxRight.Image = My.Resources.button_right_off

c. PictureBoxLeft.Image = My.Resources.button_left_on

109. ElseIf Joypos1.dwXpos <= 5 Then

a. rlStr = "1"

b. PictureBoxRight.Image = My.Resources.button_right_off

c. PictureBoxLeft.Image = My.Resources.button_left_on

110. End If

111. If Joypos1.dwButtons = 4 Then

a. bfStr = "B"

b. PictureBoxForward.Image = My.Resources.button_forward_off

c. PictureBoxBackward.Image = My.Resources.button_backward_on

112. ElseIf Joypos1.dwButtons = 1 Then

a. bfStr = "F"

b. PictureBoxForward.Image = My.Resources.button_forward_on

c. PictureBoxBackward.Image = My.Resources.button_backward_off

113. ElseIf Joypos1.dwButtons = 0 Then

a. bfStr = "0"

b. PictureBoxForward.Image = My.Resources.button_forward_off

c. PictureBoxBackward.Image = My.Resources.button_backward_off

114. End If

115. SendData("$TX:" & bfStr & rlStr & "E")

116. Catch ex As Exception

117. Console.WriteLine(ex.Message)

118. End Try

119. End Sub

120. Private Sub ReadTimer_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ReadTimer.Tick

121. ReadData(TcpClient)

122. End Sub

123. Private Sub ShowMessageReceive(ByVal message As String)

124. TextBoxReceive.AppendText(message + vbCrLf)

125. End Sub

126. Private Sub ShowMessageSend(ByVal message As String)

127. TextBoxSend.AppendText(message + vbCrLf)

128. End Sub

129. Private Sub Form1_Closing(ByVal sender As Object, ByVal e As ComponentModel.CancelEventArgs) Handles MyBase.Closing

130. Try

a. Stream.Close()

131. TcpClient.Close()

132. Catch ex As Exception

133. Console.WriteLine(ex.Message)

134. End Try

135. End Sub

```
136.        Public Sub CreateGoogleMapHtml()
137.        sw.WriteLine("<!DOCTYPE html>")
138.        sw.WriteLine("<html>")
139.        sw.WriteLine("  <head>")
140.        sw.WriteLine("    <title>Localizing the Map</title>")
141.        sw.WriteLine("    <meta name=" + Chr(34) + "viewport" + Chr(34) + "
    content=" + Chr(34) + "initial-scale=1.0, user-scalable=no" + Chr(34) + ">")
142.        sw.WriteLine("    <meta charset=" + Chr(34) + "utf-8" + Chr(34) + ">")
143.        sw.WriteLine("    <style>")
144.        sw.WriteLine("      html, body, #map-canvas {")
145.        sw.WriteLine("        height: 100%;")
146.        sw.WriteLine("        margin: 0;")
147.        sw.WriteLine("        padding: 0;")
148.        sw.WriteLine("      }")
149.        sw.WriteLine("    </style>")
150.        sw.WriteLine("    <script src=" + Chr(34) +
    "https://maps.googleapis.com/maps/api/js?v=3.exp&signed_in=true&language=ja" +
    Chr(34) + "></script>")
151.        sw.WriteLine("    <script>")
152.        sw.WriteLine("function initialize() {")
153.        sw.WriteLine("  var mapOptions = { ")
154.        sw.WriteLine("    zoom: 15,")
155.        sw.WriteLine("    center: new google.maps.LatLng(" + Trim(latitud) + ", " +
    Trim(longitud) + ")")
156.        sw.WriteLine("  };")
157.        sw.WriteLine("  var map = new
    google.maps.Map(document.getElementById('map-canvas'), mapOptions);")
158.        sw.WriteLine("}")
```

```
159.        sw.WriteLine("google.maps.event.addDomListener(window, 'load', initialize);")
160.        sw.WriteLine("    </script>")
161.        sw.WriteLine("  </head>")
162.        sw.WriteLine("  <body>")
163.        sw.WriteLine("    <div id=" + Chr(34) + "map-canvas" + Chr(34) + "></div>")
164.        sw.WriteLine("  </body>")
165.        sw.WriteLine("</html>")
166.        sw.Close()
167.        End Sub
168.        Private Sub TimerGPS_Tick(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles TimerGPS.Tick
169.        Const file As String = "C:\googlemap.html"
170.        sw = New StreamWriter(file, False, Encoding.GetEncoding(437))
171.        Try
172.        CreateGoogleMapHtml()
173.        wbGooglemap.Navigate("file:///C:/googlemap.html")
174.        Catch ex As Exception
175.        Console.WriteLine(ex.Message)
176.        End Try
177.        End Sub
178.        Private Sub wbGooglemap_DocumentCompleted(ByVal sender As
    System.Object, ByVal e As
    System.Windows.Forms.WebBrowserDocumentCompletedEventArgs) Handles
    wbGooglemap.DocumentCompleted
179.        End Sub
180.        End Class
```

ภาคผนวก ข DATASHEET ของฮาร์ดแวร์

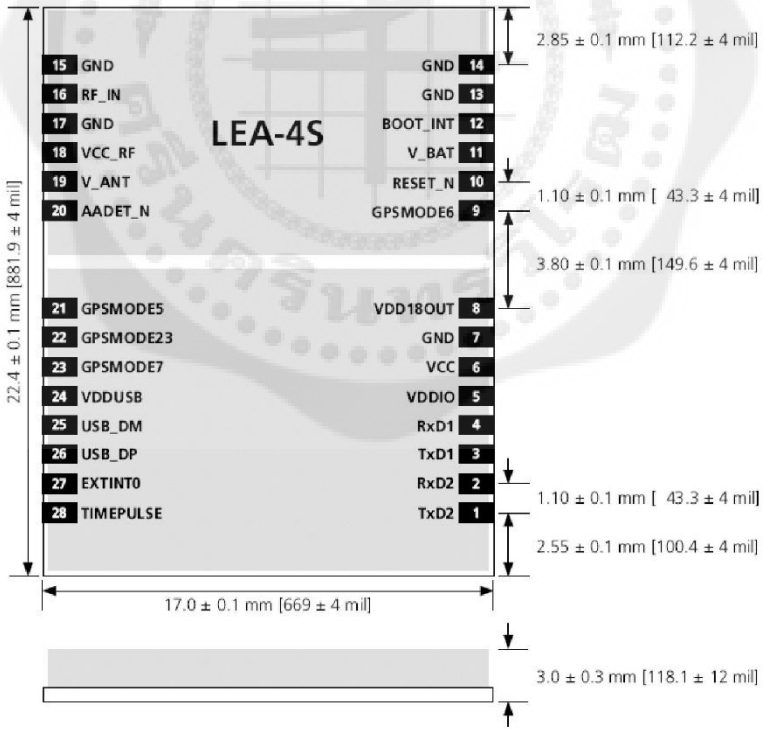
ZEEDLESS Micro GPS001D

U-Blox / LEA-4S GPS Development Kit

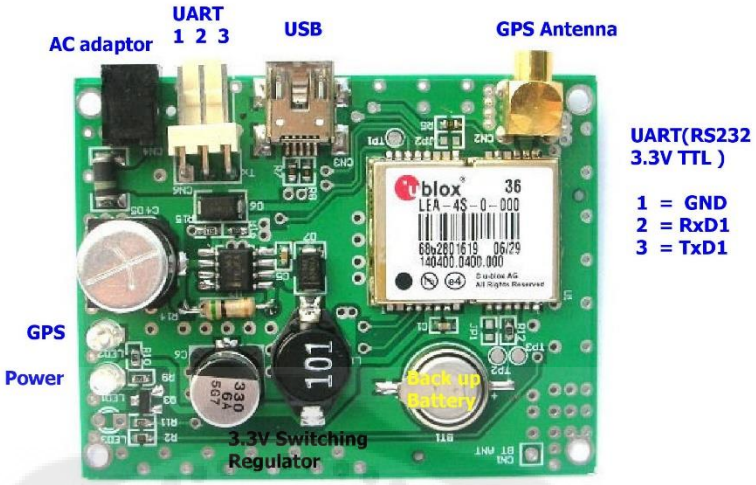
GPS001D เป็นบอร์ดที่ถูกพัฒนาขึ้นให้มีขนาดเล็กโดยใช้อุปกรณ์ที่เป็น SMD ทั้งหมด เหมาะกับผู้ใช้งานที่ต้องการศึกษาการทำงานของโมดูล GPS โดยละเอียด ตัวบอร์ดใช้ แหล่งจ่ายไฟ และ การส่งสัญญาณ ผ่านพอร์ท USB โดยผู้ใช้งานสามารถนำไปประยุกต์ใช้งาน ทั้ง งานด้าน ระบบนำทาง Navigator Systems หรือ งานด้าน Microcontroller ทั้งนี้ตัวบอร์ดยังประกอบไปด้วย จุดเชื่อมต่อช่องสัญญาณ 3.3V TTL UART(RS-232) สำหรับการใช้งาน UART และมี LED แสดงสถานะ ของแหล่งจ่ายไฟและการรับสัญญาณดาวเทียม GPS นอกจากนี้ GPS001D ยังได้เพิ่มในส่วนของแหล่งจ่ายไฟภายนอก AC adaptor และ วงจรจ่ายไฟแบบ Switching Regulator เพื่อช่วยในเรื่องของ ความเสถียรภาพของโมดูล GPS.

คุณสมบัติของบอร์ด

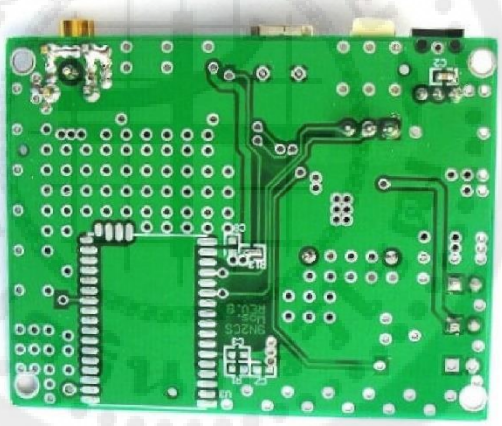
GPS receiver	U-Blox / LEA-4S ANTARIS® 4 ROM-Based GPS Module with SuperSense®
Receiver Type	16 channel , L1 frequency C/A code
Max. Update Rate	4Hz
Start-up / Hot start	<3.5sec
Start-up / Warm start	33 sec
Start-up / Cold start	34 sec
Protocols	NMEA,UBX binary, RTCM
Supply Voltage	+5VDC from USB port and AC adaptor
Interface	USB port V1.1 (V2.0 compatible) , 3.3V TTL compatible UART(RS-232)



รูปที่ 1 รูปแสดง ขนาดและตำแหน่งขาสัญญาณของโมดูล U-Blox / LEA-4S



รูปที่ 2 รูปแสดงด้านบนของ GPS001D



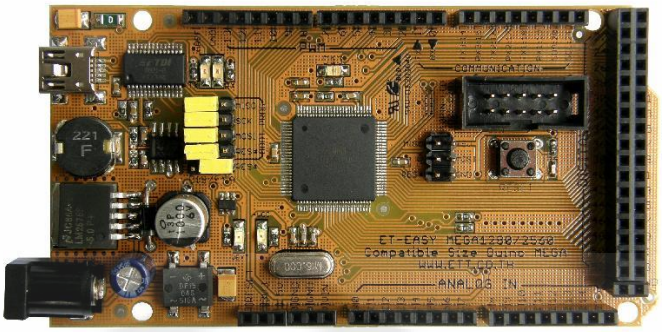
รูปที่ 3 รูปแสดงด้านล่างของ GPS001D

การติดตั้ง

ภายหลังจาก ประกอบเสาอากาศ และ ติดตั้ง GPS001D ที่พอร์ท USB ของคอมพิวเตอร์จะมีการเรียกถาม Device Driver สำหรับ อุปกรณ์ใหม่ (New Device found.) ท่านสามารถเรียกหา Driver file ได้จาก CD-ROM ที่มาพร้อมกันในชุด GPS001D โดยค้นหาได้จากโฟล์เดอร์ ดังนี้

D:\Software_Driver\PersonalComputor\uCenterSetup

ET-EASY MEGA1280 (Duino Mega)

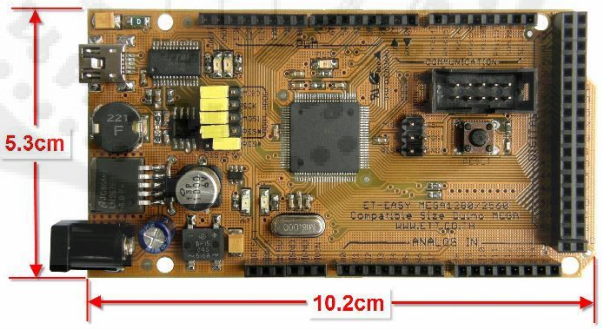


จากการที่ Arduino ที่เป็นโครงการพัฒนาระบบ MCU ของ AVR แบบ Open Source ได้รับการแนะนำเผยแพร่ออกมาสู่สาธารณะ ซึ่งได้รับความนิยมกันอย่างแพร่หลายจากผู้คนทั่วโลก ภายในระยะเวลาอันรวดเร็ว ทางด้านของ Software ก็มีการพัฒนากันอย่างต่อเนื่อง ซึ่งในขณะนี้ (สิงหาคม 2552) โปรแกรมของ Arduino ได้รับการปรับปรุงเป็น Version “arduino-0016” แล้ว โดยทางด้าน Hardware เองก็ได้มีการพัฒนาปรับปรุงอย่างต่อเนื่องควบคู่กันไปด้วยเช่นเดียวกัน ซึ่งจากเดิมที่มีการพัฒนาโปรแกรมให้รองรับกับการใช้งานได้กับชิพ MCU รุ่นเล็ก 28 ขา อย่าง ATMEGA8,ATMEGA88/ATMEGA168/ATMEGA328 สำหรับเป็นจุดเริ่มต้นให้ผู้สนใจได้ใช้เป็นเครื่องมือและอุปกรณ์ในการศึกษาทดลองเรียนรู้ ตลอดจนถึงนำไปประยุกต์ใช้งานกันนั้น มาถึงวันนี้ ขนาดของทรัพยากรต่างๆ รวมทั้งขนาดของหน่วยความจำ สำหรับเขียนโปรแกรม ที่มีอยู่ในชิพ AVR รุ่นเล็กที่มีอยู่เริ่มไม่เพียงพอกับการประยุกต์ใช้งานในงานบางประเภทแล้ว ทาง Arduino เอง จึงได้ทำการพัฒนาให้ Arduino สามารถรองรับการใช้งานขนาดใหญ่ขึ้นอีก โดยปรับปรุงโปรแกรมให้ใช้กับชิพ AVR รุ่นใหญ่ขึ้น เพื่อให้มีจำนวน I/O ทั้ง Digital, Analog, PWM, UART และ ขนาดหน่วยความจำเพิ่มมากขึ้นกว่าเดิม แต่ก็ยังคงใช้แนวทางในการพัฒนาโปรแกรม แบบเดียวกันกับรุ่นเล็กทุกประการ โดยได้เลือกใช้ชิพ AVR เบอร์ ATMEGA1280 และออกแบบพัฒนา Hardware บอร์ดขึ้นมารองรับ โดยใช้รหัสชื่อรุ่นว่า “Arduino Mega” ออกวางจำหน่าย และได้มีการเผยแพร่รายละเอียดทาง Hardware ต่างๆให้ผู้สนใจนำไปพัฒนาต่อยอดกันได้เอง แต่เนื่องจากชิพดังกล่าวมีโครงสร้างตัวถังเป็นแบบ SMD จึงทำให้เป็นอุปสรรคสำหรับผู้ใช้งานจำนวนไม่น้อยในการที่จะผลิตหรือสร้างบอร์ดขึ้นใช้งานเอง

ทาง อีทีที จึงได้นำ ATMEGA1280 มาพัฒนาเป็นบอร์ด โดยให้มีโครงสร้างการทำงานเช่นเดียวกันกับ Arduino Mega ขึ้นมา โดยใช้ชื่อว่า **”ET-EASY MEGA1280”** โดยได้ออกแบบให้มีการจัดสรร Pin I/O ต่างๆ รวมทั้งขนาดให้ตรงตามมาตรฐานของบอร์ด **“Arduino Mega”** เพียงแต่ได้มีการปรับปรุงข้อจำกัดบางอย่างให้ดีขึ้นกว่า Arduino Mega รุ่นมาตรฐาน เพื่อเพิ่มความสะดวกกับผู้ใช้งานมากยิ่งขึ้น

คุณสมบัติของบอร์ด

- ใช้ ATMEGA1280 เป็น MCU ประจำบอร์ด Run ความถี่ 16MHz จาก Crystal Oscillator
- 128KByte Flash(สงวนไว้ 4KByte สำหรับ Bootloader) / 8KByte SRAM / 4KByte EEPROM
- รองรับการพัฒนาโปรแกรมด้วยภาษา C++ ของ Arduino ตามแบบ Arduino Mega ได้ 100%
- ใช้ USB Bridge ของ FTDI เบอร์ FT232RL พร้อม Over Current Protection สำหรับติดต่อสื่อสารและ Download Code จากคอมพิวเตอร์ให้บอร์ด พร้อม Jumper สำหรับปรับใช้งานบอร์ดเป็นการ Program Bootloader ให้กับ MCU จากพอร์ต USB ในบอร์ดได้เอง โดยไม่ต้องใช้เครื่องโปรแกรม AVRISP จากภายนอก
- 54 Pin Digital I/O โดยมี 14Pin สามารถโปรแกรมหน้าที่เป็น PWM ได้
- 16 Pin Analog Input (ADC ขนาด 10 บิต 16 ช่อง)
- 4 UART(Hardware Serial Port) แบบ TTL Logic
- ขนาดของ PCB บอร์ด และ ตำแหน่ง Pin Connector ต่างๆ ตรงกันกับ Arduino Mega ทั้งหมด ทำให้สามารถนำไปติดตั้งใช้งานร่วมกับบอร์ด Shield แบบต่างๆที่มีการผลิตขึ้นมาใช้งานร่วมกันกับบอร์ด Arduino Mega ได้ทั้งหมด โดยบอร์ดมีขนาด PCB Size 5.3cm x 10.2cm
- มีขั้ว Header 10Pin IDE ของ 8 บิต Digital I/O(D22...D29) สำหรับเชื่อมต่อกับ LCD หรือ บอร์ด I/O แบบต่างๆ ของ อีทีที เพิ่มความสะดวกในการใช้งาน
- รองรับการใช้งานกับ External Supply ทั้งแบบ AC และ DC ขนาด 7-20V โดยเลือกใช้ Regulate แบบ Switching ขนาด 1A (LM2575-5V) ลดปัญหาเรื่องความร้อนเมื่อมีการใช้กระแสสูงๆ สามารถใช้แหล่งจ่ายจากพอร์ต USB ได้ในกรณีใช้กระแสไม่เกิน 500mA โดยมีวงจรเลือกแหล่งจ่ายอัตโนมัติ โดยจะตัดการใช้ไฟเลี้ยงจาก USB โดยอัตโนมัติ เมื่อมีการต่อแหล่งจ่ายจากภายนอกให้บอร์ด



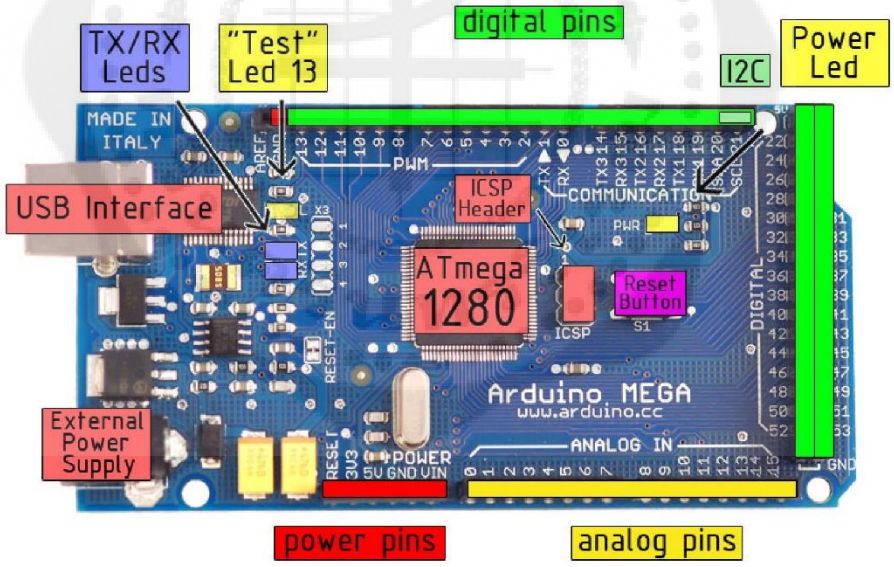
Technical Specification

EAGLE files: arduino-mega-reference-design.zip Schematic: arduino-mega-schematic.pdf

Summary

ATmega1280	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares RADIONICS

Features

- Complies with 802.11g and 802.11b (2.4 GHz) Standards
- Unsurpassed Wireless Security with Wi-Fi Protected Access™ 2 (WPA2)
- Enhanced Internet Security Management Functions including Internet Access Policies with Time Schedules
- All LAN Ports Support Auto-Crossover (MDI/MDI-X) — No Need for Crossover Cables

Cisco Consumer Business Group
121 Theory
Irvine, CA 92617 USA

www.linksysbycisco.com

Linksys, Cisco and the Cisco Logo are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries. Other brands and product names are trademarks or registered trademarks of their respective holders. Copyright © 2009 Cisco Systems, Inc. All rights reserved.

Specifications

Model	WRT54GL
Standards	IEEE 802.3, IEEE 802.3u, IEEE 802.11g, IEEE 802.11b
Ports	Internet: One 10/100 RJ-45 Port Ethernet: Four 10/100 RJ-45 Switched Ports One Power Port
Buttons	One Reset Button
LEDs	Power, DMZ, WLAN, Ethernet (1, 2, 3, 4), Internet
Cabling Type	CAT 5
RF Power (EIRP) in dBm	18
UPnP able/cert	Able
Security Features	Stateful Packet Inspection (SPI) Firewall, Internet Policy
Wireless Security	Wi-Fi Protected Access™ 2 (WPA2), WEP, Wireless MAC Filtering

Environmental

Dimensions	7.32" x 1.89" x 7.87" (186 x 48 x 200 mm)
Weight	17.0 oz (482 g)
Power	12VDC, 1A
Certification	FCC, ICES-003, CE, Wi-Fi (802.11b, 802.11g), WPA2, WMM
Operating Temp.	32 to 104°F (0 to 40°C)
Storage Temp.	-4 to 158°F (-20 to 70°C)
Operating Humidity	10 to 85% Noncondensing
Storage Humidity	5 to 90% Noncondensing

Package Contents

- Wireless-G Linux Broadband Router
- Setup Software and User Guide on CD-ROM
- Power Adapter
- Network Cable

Minimum Requirements

- Internet Explorer 6 or Firefox 2 or Higher for Browser-based configuration
- CD-ROM Drive
- Windows XP, Vista, or Vista 64-bit Edition with Latest Updates
- Wired or Wireless Network Adapter.

The maximum performance for wireless is derived from IEEE Standard 802.11 specifications. Actual performance can vary, including lower wireless network capacity, data throughput rate, range and coverage. Performance depends on many factors, conditions and variables, including distance from the access point, volume of network traffic, building materials and construction, operating system used, mix of wireless products used, interference and other adverse conditions.

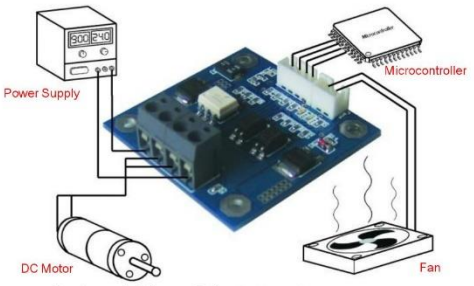
Specifications are subject to change without notice.

09021911NC-AI
3415-01458

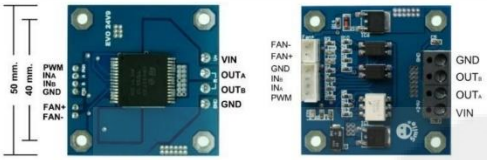
Model: **WRT54GL**

EVO24V9

High Power Motor Driver



ตัวอย่างการต่อใช้งานบอร์ดขับ DC Motor รุ่น **EVO24V9**



ข้อควรระวังในการใช้งาน ในการต่อใช้งาน บอร์ดขับ DC Motor รุ่น EVO24V9 เพื่อเป็นการป้องกันความเสียหายของวงจรควรต่อฟิวส์เพื่อจำกัดกระแสในกรณีที่เกิดการลัดวงจร และควรตรวจสอบความถูกต้องก่อนจ่ายไฟฟ้าให้กับวงจร

การติดตั้งตัวบอร์ดควรติดตั้งในตำแหน่งที่มีอากาศถ่ายเทได้ดี หรือติดตั้งระบบระบายความร้อนเพิ่ม เช่น พัดลม ฮีทซิงค์เป็นต้น เพื่อระบายความร้อนในตัวบอร์ด เมื่อนำบอร์ดไปใช้ในงานขับกระแสสูงๆ ต่อเนื่องเป็นระยะเวลานานเพื่อเพิ่มประสิทธิภาพในการทำงานและช่วยยืดอายุการใช้งานได้อีกด้วย

คุณสมบัติ

- ไอซี Full bridge motor driver เบอร์ VNH3SP30
- กระแสสูงสุดชั่วขณะ 30 A และขับกระแสต่อเนื่อง 9 A ที่แรงดัน 24 VDC
- แรงดันไฟฟ้าอินพุต VCC 7-28 VDC
- แรงดันไฟฟ้าเอาต์พุตสูงสุด 0.98 × VCC
- สัญญาณเลอจิก 1 มีแรงดันไฟฟ้า 3-5.5 V
- แยกสัญญาณไฟฟ้าควบคุมจากไมโครคอนโทรลเลอร์ออกจากสัญญาณขับมอเตอร์ด้วย OPTO-Isolator เพื่อป้องกันสัญญาณรบกวนจากมอเตอร์
- ความถี่ในการควบคุมมอเตอร์ PWM ไม่เกิน 1 kHz
- แสดงสถานะทิศทางหมุน ด้วย LED
- มีวงจรป้องกันการจายไฟเลี้ยงสลับขั้ว
- ตัดการทำงานเมื่ออุณหภูมิภายใน สูงเกินกว่า 150 °C
- สามารถต่อพัดลมระบายความร้อน แรงดันไฟฟ้า 24 V ขนาด 50 mm เพิ่มได้
- ขนาด PCB 50 mm x 50 mm
- น้ำหนัก 8 g

ตาราง การเชื่อมต่อ

PIN	Signal	Description
VIN	+VCC	ไฟเลี้ยงระบบที่แรงดันไฟฟ้า + 7 .....+28 VDC
GND	Power GND	กราวด์ของไฟเลี้ยง
OUT A	Motor winding (+M)	แรงดันไฟฟ้าเอาต์พุตมอเตอร์
OUT B	Motor winding (-M)	แรงดันไฟฟ้าเอาต์พุตมอเตอร์
IN A	IN A	สัญญาณเลอจิกอินพุต หมุนตามเข็มนาฬิกา
IN B	IN B	สัญญาณเลอจิกอินพุต หมุนทวนเข็มนาฬิกา
PWM	PWM	สัญญาณพัลซ์วิดธ์มอดูเลต (PWM) อินพุต
GND	GND	กราวด์ของสัญญาณควบคุม
+FAN	+VFAN	ไฟเลี้ยงพัดลมที่แรงดันไฟฟ้า +VCC
-FAN	-VFAN	กราวด์ของพัดลม

ตาราง ควบคุมการทำงาน

PWM	IN A	IN B	Operating mode
ON	0	0	Free Run Stop
ON	1	1	Free Run Stop
OFF	1	1	Brake to VCC
OFF	0	0	Brake to GND
OFF	1	0	Clockwise (CW)
OFF	0	1	Counterclockwise (CCW)

SMILE ROBOTICS

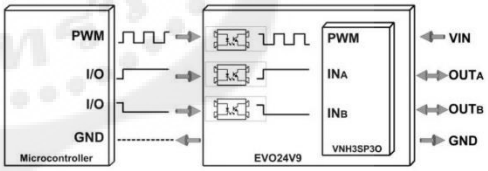
http://www.facebook.com/pages/Smile-Robotics/143619432342623?ref

e-mail : smile–robotics@hotmail.com

บอร์ดขับ DC Motor รุ่น EVO24V9 เป็นวงจรขับ DC Motor ที่ใช้ไอซีแบบ Full bridge motor driver เบอร์ VNH3SP30 ของบริษัท ST Microelectronic ซึ่งสามารถขับกระแสได้สูงสุด 30 A ตัวบอร์ดถูกออกแบบมาให้สามารถ ใช้แรงดันไฟฟ้าอินพุตสูงสุดที่ 28 V มีวงจรแยกสัญญาณไฟฟ้าควบคุมจากไมโครคอนโทรลเลอร์ออกจากสัญญาณเข้มมอเตอร์ ด้วยไอซี OPTO-Isolator เพื่อป้องกันสัญญาณรบกวนจากมอเตอร์ บอร์ดมีขนาดเล็กน้ำหนักเบาใช้งานง่าย สามารถต่อสัญญาณควบคุมจากไมโครคอนโทรลเลอร์หรือวงจรควบคุมอื่นๆได้ เหมาะสำหรับใช้ขับ DC Motor ขนาดไม่เกิน 150 W

การประยุกต์ใช้งานบอร์ดขับ DC Motor รุ่น EVO24V9 ใช้สำหรับควบคุมความเร็ว , ทิศทาง DC Motor สำหรับหุ่นยนต์หรือประยุกต์ใช้ควบคุมอุปกรณ์ไฟฟ้าอื่นๆ เช่น หลอดไฟกำลังสูง โซลินอยล์ เป็นต้น

LED แสดงสภาวะการทำงาน
LED สีแดง จะสว่างเมื่อมีไฟฟ้าจากแหล่งจ่ายเข้าวงจร
LED สีเขียวและสีน้ำเงิน LED จะสว่างเมื่อมีสัญญาณเลอจิก อินพุต IN A และ IN B เพื่อแสดงทิศทางการหมุนของมอเตอร์ตามเข็มนาฬิกา หรือทวนเข็มนาฬิกา



สัญญาณ PWM ที่ใช้ควบคุมมอเตอร์กับสัญญาณ PWM ที่ขับมอเตอร์

หมายเหตุ เนื่องจากใช้วงจร OPTO-Isolator แยกสัญญาณไฟฟ้าควบคุมจากไมโครคอนโทรลเลอร์ออกจากสัญญาณเข้มมอเตอร์ ทำให้สัญญา PWM ที่ขับมอเตอร์มีสถานะตรงข้าม (Invert) กับสัญญาณ PWM ที่ใช้ควบคุมเอเตอร์ โดยสามารถแสดงความสัมพันธ์สัญญาณ PWM ที่ใช้ควบคุมมอเตอร์กับสัญญาณ PWM ที่ขับมอเตอร์ได้ดังรูป

Product Review

กองบรรณาธิการ

เกาะติดทุกความเคลื่อนไหว ง่ายดายเหมือนอยู่ในฝ่ามือ D-Link Home Network Camera รุ่น DCS-942L

DCS-942L กล้องมากคุณสมบัติ ทำงานได้ดี ทั้งกลางวันและกลางคืน พร้อมโซลูชันจาก D-Link ที่สามารถดูภาพจากกล้องวงจรปิดได้ทุกที่ทุกเวลา ช่วยยกระดับความปลอดภัยต่อทรัพย์สินภายในบ้านท่าน

"ความปลอดภัย" ถือเป็นสิ่งสำคัญที่ช่วยรักษา และปกป้องสิ่งสำคัญนั้นๆ ไม่ให้สูญหาย หรือโดนทำลายจากเหตุการณ์ใดๆ ก็ตาม ในมาตรการรักษาความปลอดภัยอาจมีหลายแบบ ซึ่งการใช้กล้องวงจรปิด เพื่อตรวจตราความเคลื่อนไหว ก็ถือเป็นมาตรการรักษาความปลอดภัยแบบหนึ่ง โดยเฉพาะปัจจุบันนี้ที่โจรผู้ร้าย ขุกขุม กล้องวงจรปิดยิ่งเป็นสิ่งสำคัญที่ขาดไม่ได้

บรรดาคนทำงานทั้งหลายที่ไม่มีเวลาอยู่บ้าน หรือในวันหยุดยาวขณะที่ทุกคนกำลังออกไปเที่ยว กล้องวงจรปิดจะเป็นหูเป็นตาแทนเจ้าบ้านที่ไม่อยู่ และที่สำคัญขณะที่เจ้าบ้านอยู่ข้างนอกก็สามารถเข้าถึงตัวกล้องวงจรปิดเพื่อดูความเคลื่อนไหวภายในบ้านได้ ซึ่งกล้องวงจรปิดจาก D-Link รุ่น DCS-942L สามารถทำงานได้ดีไม่ว่าจะในเวลากลางวันหรือกลางคืน ยังสามารถดูภาพ (มอนิเตอร์) ของกล้องผ่านระบบอินเทอร์เน็ตได้อีกด้วย

เสปคและรายละเอียดที่สำคัญของ DCS-942L

Product	D Link Home Network Camera รุ่น DCS 942L
Memory	SDRAM 128 MB, Flash Memory 16 MB
Encoder	H.264, MPEG4 ความละเอียดสูงสุด 640x480 @ 30 fps
Network	10/100 LAN, Wireless 802.11b/g/n
Feature	มีระบบกลางวัน / กลางคืน (Day/Night System), เข้าถึงตัวกล้องผ่านระบบอินเทอร์เน็ตได้ (Remote Access), มีระบบตรวจจับการเคลื่อนไหว (Motion Dectection)
Other	รองรับ microSD card

เนื่องจาก DCS-942L เป็นกล้องวงจรปิดแบบติดตั้งภายในบ้าน (Home Network) ดังนั้นประสิทธิภาพที่มาพร้อมกับความ “ง่าย” จึงเป็นคุณสมบัติของกล้องวงจรปิดตัวนี้ ไม่ว่าจะเป็นความง่ายจากการติดตั้ง หรือจะเป็นความง่ายในการลงโปรแกรม ช่วยให้ผู้ใช้ที่ไม่มีความรู้ด้านกล้องวงจรปิด สามารถใช้งาน บริหารจัดการกล้องวงจรปิดได้ไม่ยาก แต่ก่อนที่จะไปทำความรู้จักกับโปรแกรมที่ใช้บริหารจัดการ เรามาทำความรู้จักกับตัวกล้องกันก่อน

สามารถปรับเปลี่ยนองศาได้ตามใจ

หากมองแค่เฉพาะตัวกล้องวงจรปิด ไม่นับฐานที่ใช้ติดตั้งแล้ว ผู้คนอาจเข้าใจผิดคิดว่าเป็นกล้องดิจิทัลได้ เพราะตัวกล้องมีรูปร่างที่คล้ายกับกล้องดิจิทัลมาก ตัวกล้องมีลักษณะเป็นสี่เหลี่ยมผืนผ้า กล้องวิดีโอที่ใช้ในการจับภาพ ถูกจัดวางไว้ในตำแหน่งกึ่งกลางล้อมรอบด้วย Infrared Light ที่อยู่ในวงแหวน นอกจากนี้ในวงแหวนยังมีไมโครโฟนติดตั้งอยู่ด้วย

เมื่อมาดูหลังกล้องก็จะพบกับพอร์ต Ethernet, ช่องสำหรับเสียบหูฟัง, ช่อง Power, ไฟแสดงสถานะ, reset hole และปุ่ม WPS โดยพอร์ตและช่องต่างๆ เหล่านี้ถูกจัดวางมาอย่างเป็นระเบียบ

พอร์ต ETHERNET และช่องต่อต่างๆ ถูกจัดวางอย่างเป็นระเบียบ

ช่องใส่ microSD card และปุ่ม WPS พร้อมไฟสถานะการทำงาน

จุดเชื่อมต่อระหว่างตัวกล้องกับฐานตั้งกล้องคือบอลพลาสติก และด้วยบอลพลาสติกนี้เอง จึงทำให้ตัวกล้องสามารถปรับองศาและหมุนได้อย่างอิสระ โดยบอลพลาสติกที่เป็นทั้งตัวเชื่อมกับฐานตั้งกล้องและจุดหมุนนั้นจะถูกยึดด้วยตัวล็อคสีเงิน ซึ่งตัวล็อคสีเงินสามารถปรับระดับการบีบล็อคได้ด้วยเกลียวที่อยู่ด้านล่าง

บอลพลาสติกช่วยให้ตัวกล้องหมุนได้อย่างอิสระ และรูน็อตที่ถูกซ่อนไว้อย่างดี

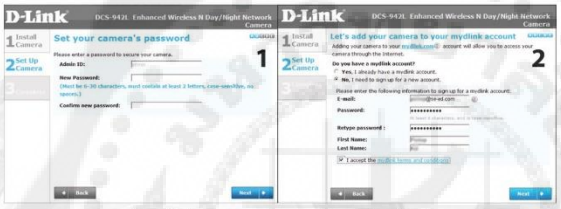
ก่อนใช้งานกล้องวงจรปิด DCS-942L ผู้ใช้ต้องเชื่อมต่อกล้องวงจรปิดเข้ากับระบบเครือข่ายภายในบ้านเสียก่อน และไม่ว่าจะเป็นเครือข่ายแบบใช้สาย หรือไร้สาย กล้องวงจรปิด DCS-942L ก็สามารถเชื่อมต่อได้ทั้งนั้น แต่การที่ผู้ใช้งานจะสามารถตั้งค่า หรือดูภาพจากกล้องวงจรปิดได้จะต้องทำการตั้งค่า User-password และติดตั้งโปรแกรม D-View Cam ก่อน

การตั้งค่า User-password นั้นทำผ่านโปรแกรมที่ชื่อ D-Link Setup Wizard ภายในโปรแกรมได้มีอธิบายวิธีการเชื่อมต่อกล้องวงจรปิดกับเครือข่ายภายในบ้าน รวมไปถึงวิธีการหมุนหรือปรับองศาของตัวกล้อง ทั้งนี้ user-password จะถูกแบ่งเป็นสองส่วนด้วยกัน ส่วนแรกใช้สำหรับลอคอินเข้าโปรแกรม D-View Cam ที่ติดตั้งในเครื่องคอมพิวเตอร์ ส่วน user-password ส่วนที่สองที่มาจากการลงทะเบียน mydlink account นั้น ใช้เพื่อดูภาพ หรือตั้งค่ากล้องวงจรปิดจากระยะไกล ผ่านระบบอินเทอร์เน็ต

ในโปรแกรมยังอธิบายวิธีการปรับองศาของตัวกล้อง

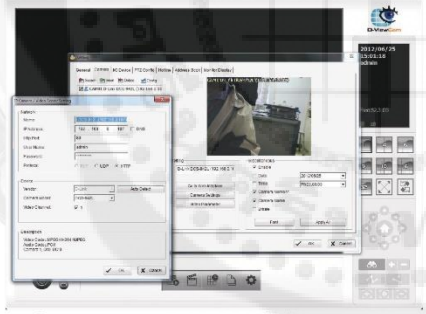
D-View Cam เป็นโปรแกรมที่ใช้บริหารจัดการกล้องวงจรปิด ผู้ใช้งานสามารถใช้โปรแกรม D-View Cam ในการตั้งค่าตัวกล้อง หรือระบบการทำงานต่างๆ ของกล้องได้ เช่น ตั้งเวลาในการบันทึกวิดีโอ หรือตั้งค่าให้ระบบทำการส่งอีเมล์มาให้ผู้ใช้งานเมื่อตัวกล้องตรวจจับความเคลื่อนไหวภายในบ้านได้

D-Link ยังมีเว็บไซต์ www.mydlink.com เว็บไซต์นี้มีประโยชน์เป็นอย่างมาก เพราะช่วยให้ผู้ใช้สามารถดูภาพของกล้องวงจรปิดจากที่ใดก็ได้บนโลก ขอแค่เพียงคอมพิวเตอร์เครื่องนั้นๆ เชื่อมต่ออินเทอร์เน็ตไว้ นอกจากเว็บไซต์แล้ว D-Link ยังมีแอพพลิเคชันที่ชื่อว่า mydlink แอพพลิเคชันนี้มีไว้เพื่อให้ผู้ใช้งาน สามารถดูภาพจากกล้องวงจรปิดบนมือถือแบบสมาร์ทโฟนได้ ซึ่งแอพพลิเคชันดังกล่าวมีให้ดาวน์โหลดทั้งบน Android Market และ Appstore

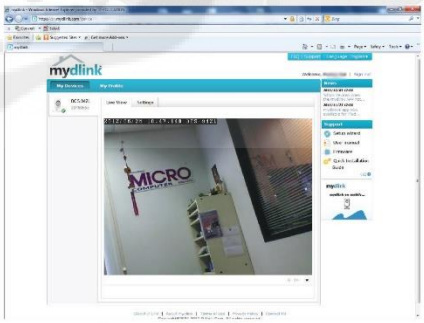


มีการตั้งค่า user-password ถึงสองส่วนด้วยกัน

มอนิเตอร์กล้องวงจรปิดผ่านแอพพลิเคชัน mydlink



โปรแกรม D-View Cam นอกจากจะใช้ตั้งค่าต่างๆ แล้ว ยังใช้ในการมอนิเตอร์ด้วย



มอนิเตอร์กล้องวงจรปิดผ่านเว็บไซต์ www.mydlink.com

ประวัติย่อนิสิตผู้จัดทำโครงงาน

ชื่อ-สกุล	นายชัชชนก จิรวัฒนาพันธ์
วัน เดือน ปีเกิด	2 เมษายน 2536
สถานที่เกิด	กาญจนบุรี
สถานที่อยู่ปัจจุบัน	5 ต.ท่ามะขาม อ.เมือง จ.กาญจนบุรี 71000
โทรศัพท์	081-1788526
E-Mail	thow@hotmail.com

ประวัติการศึกษา

พ.ศ. 2550	มัธยมต้นโรงเรียนโรงเรียนดรุณากาญจนบุรี จังหวัดกาญจนบุรี
พ.ศ. 2553	มัธยมปลายโรงเรียนสารสิทธิ์พิทยาลัย จังหวัดราชบุรี
พ.ศ. 2557	กำลังศึกษาระดับปริญญาตรี คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมไฟฟ้า แขนงวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ประวัติย่อนิสิตผู้ทำโครงงาน

ชื่อ-สกุล นายจักรพันธ์ จันทสิงห์

วัน เดือน ปีเกิด 1 เมษายน 2534

สถานที่เกิด เพชรบูรณ์

สถานที่อยู่ปัจจุบัน 94 หมู่ 12 ต.บุ่งน้ำเต้า อ.หล่มสัก จ.เพชรบูรณ์

โทรศัพท์ 086-4057747

E-Mail joker.chantrasing@gmail.com

ประวัติการศึกษา

พ.ศ. 2549 มัธยมต้น โรงเรียนเซนโจเซฟศรีเพชรบูรณ์ จังหวัดเพชรบูรณ์

พ.ศ. 2552 มัธยมปลาย โรงเรียนเซนโจเซฟศรีเพชรบูรณ์ จังหวัดเพชรบูรณ์

พ.ศ. 2557 กำลังศึกษาระดับปริญญาตรี คณะวิศวกรรมศาสตร์
สาขาวิชาวิศวกรรมไฟฟ้า แขนงวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยศรีนครินทรวิโรฒ

ประวัติย่อนิสิตผู้ทำโครงงาน

ชื่อ-สกุล นายณัฐวุฒิ ภิระบรรณ์

วัน เดือน ปีเกิด 1 เมษายน 2535

สถานที่เกิด เชียงราย

สถานที่อยู่ปัจจุบัน 85 หมู่ 12 ถนนพหลโยธิน ต.โป่งผา
อ.แม่สาย จ.เชียงราย 57130

โทรศัพท์ 081-8829779

E-Mail worldwoody@hotmail.com

ประวัติการศึกษา

พ.ศ. 2550 มัธยมต้น โรงเรียนแม่สายประสิทธิ์ศาสตร์ จังหวัดเชียงราย

พ.ศ. 2553 มัธยมปลาย โรงเรียนสามัคคีวิทยาคม จังหวัดเชียงราย

พ.ศ. 2557 กำลังศึกษาระดับปริญญาตรี คณะวิศวกรรมศาสตร์
สาขาวิชาวิศวกรรมไฟฟ้า แขนงวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยศรีนครินทรวิโรฒ