

หุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ

AUTONOMOUS HUMAN TRACKING MOBILE ROBOT



นายประกายเพชร ฟองงาม รหัส 53363614
นายโรจน์อนันต์ สังข์นวล รหัส 53363874

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2557

ห้องสมุดคณะ วิศวกรรมศาสตร์
วันที่รับ 20 ก.ค. 2558
เลขที่รับ 16876201
เลขเรียกหนังสือ
มหาวิทยาลัยนเรศวร 21/10/11



ใบรับรองปริญญาโท

ชื่อหัวข้อโครงการ หุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ
ผู้ดำเนินงาน นายประกายเพชร ฟองงาม รหัส 53363614
 นายโรจน์อานันท์ สังข์นวล รหัส 53363874
ที่ปรึกษาโครงการ ผศ.ดร.พรพิศุทธิ์ วรจิรันตน์
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2557

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....ที่ปรึกษาโครงการ
(ผศ.ดร.พรพิศุทธิ์ วรจิรันตน์)

Sunit Kiravittayaกรรมการ
(ผศ.ดร.สุวิทย์ กิระวิทยา)

.....กรรมการ
(ผศ.ดร.พนัส นัถฤทธิ)

ชื่อหัวข้อโครงการงาน	หุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ
ผู้ดำเนินโครงการงาน	นายประกายเพชร ฟองงาม รหัส 53363614 นายโรจน์อานันท์ สังข์นวล รหัส 53363874
ที่ปรึกษาโครงการงาน	ศศ.ดร.พรพิศุทธิ์ วรจิรันคนี่
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2557

บทคัดย่อ

หุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัตินี้ออกแบบเพื่อให้สามารถติดตามมนุษย์ในระยะใกล้ เพื่อนำไปประยุกต์ใช้ในการสร้างหุ่นยนต์ซึ่งมีหน้าที่อำนวยความสะดวกโดยเคลื่อนที่ไปในจุดต่างๆ ตามผู้ถูกติดตามได้ ในอนาคตอาจพัฒนาแบบจำลองนี้ให้เป็นรถบรรทุกสิ่งของขนาดเล็กสำหรับผู้พิการ คนชรา หรือผู้ที่ก้ำกึ่งเนื้ออ่อนแรง

การติดตามทำได้โดยใช้อุปกรณ์รับส่งสัญญาณอัลตราโซนิก (Ultrasonic Sensor) วัดระยะห่างระหว่างหุ่นยนต์กับมนุษย์ และควบคุมความเร็วของมอเตอร์โดยใช้ระบบควบคุมแบบ PID ให้เคลื่อนที่ด้วยความเร็วเท่ากับผู้ถูกติดตาม ส่วนของการควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ใช้อุปกรณ์ไมโครเข็มทิศ (Compass Module) เปรียบเทียบค่าของมุมกับแอปพลิเคชันตรวจจับทิศทางบนโทรศัพท์สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ เพื่อควบคุมให้หุ่นยนต์มีทิศทางเดียวกับทิศทางเคลื่อนที่ของผู้ถูกติดตาม และใช้บอร์ดทดลอง Fio STD ออกแบบด้วยโปรแกรมแมทแลบซิมูลิงค์ (Matlab Simulink) เป็นอุปกรณ์ควบคุมหลัก

Project title Autonomous human tracking mobile robot
Name Mr. Prakayphet Fongngam ID. 53363614
Mr. Roj-arnun Sungnual ID.53363874
Project advisor Asst.Prof.Dr. Pongpisut Worrajiran
Major Computer Engineering
Department Electrical and Computer Engineering
Academic year 2014

Abstract

An autonomous mobile human tracking robot was designed with the ability to closely follow a person's movements. This robotic device can be developed to control, perhaps, a small transport device or mobile cart to assist the disabled, the elderly or those with little muscular weakness, where the cart can closely follow the person without direct hands-on controlling.

An ultrasonic sensor is used to measure the distance between the robot and the person. A PID control is used to control the speed of movement of the device to match the speed of the person being followed. A compass module is used to control the direction of movement by comparing the angle of change of direction value calculated from the direction of an Android smartphone carried by the target person, which has a directional detection application, communicating by a Bluetooth connection. A Fio STD testing board is used as the main controller, which was designed using the Matlab Simulink program.

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาเป็นอย่างยิ่งจาก ศศ.ดร.พรพิศุทธิ์ วรจิรันตน์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการ และให้ความกรุณาในการช่วยเหลือ ให้คำแนะนำปรึกษา คณะผู้ดำเนินโครงการขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้กับคณะผู้ดำเนินโครงการ

นอกจากนี้ยังต้องขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ ที่ให้สัมปทานและเครื่องมือวัดมาใช้ในการดำเนินโครงการจนทำให้สำเร็จลุล่วงไปด้วยดี ขอขอบคุณเพื่อนและผู้ที่มีอุปการคุณทุกๆ คนของคณะผู้ดำเนินโครงการที่ไม่ได้กล่าวมาไว้ ณ ที่นี้ด้วย

ผู้ดำเนินโครงการ

นายประกายเพชร ฟองงาม

นายโรจน์อนันต์ สัจจันวล

สารบัญ

	หน้า
ใบรับรองปริญญาบัตร	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญรูป	ฉ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของ โครงการงาน	1
1.2 วัตถุประสงค์ของ โครงการงาน	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	2
1.4 ขอบเขตการทำโครงการงาน	2
1.5 ขั้นตอนการทำโครงการงาน	2
1.6 แผนการดำเนินงาน	3
1.7 รายละเอียดงบประมาณตลอดโครงการงาน	3
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	4
2.1 ระบบควบคุมส่วนผู้ใช้	4
2.1.1 ระบบปฏิบัติการแอนดรอยด์	4
2.1.2 แอปพลิเคชันตรวจจับทิศทาง	4
2.2 ระบบควบคุมส่วนหุ่นยนต์	6
2.2.1 การสื่อสารแบบ I ² C สำหรับการติดต่อกับโมดูลเข็มทิศดิจิทัล	7
2.2.2 การสื่อสารแบบ UART สำหรับการติดต่อกับบลูทูท	7
2.2.3 เซอร์โวมอเตอร์	7
2.2.4 Ultrasonic Sensor	9
2.2.5 บอร์ด ไฟโอ (FiO Board)	10

สารบัญ (ต่อ)

	หน้า
2.2.6 หลักการสร้างสัญญาณ PWM	10
2.2.7 ทฤษฎีคอนโทรลเลอร์แบบ PID	11
2.2.8 Rotary Encoder	19
2.2.9 aMG IMU-9A	20
2.2.10 aMG Bluetooth	22
2.2.11 วงจรขับเคลื่อนมอเตอร์ L298N	24
2.2.12 Module ET-MINI 3T05 TTL	26
2.2.13 DC Motor	27
บทที่ 3 ขั้นตอนการดำเนินโครงการ	29
3.1 โครงสร้างของหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ	29
3.2 การติดตั้งและเชื่อมต่ออุปกรณ์	30
3.2.1 Compass Module	31
3.2.2 Ultrasonic Sensor	32
3.2.3 Module Bluetooth	33
3.2.4 Servo Motor	33
3.2.5 Rotary Encoder	34
3.2.6 Opto Isolator	35
3.2.7 DC Motor	35
3.3 โปรแกรมการทำงานหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ	36
3.3.1 โปรแกรมสำหรับ Module Compass	37
3.3.2 โปรแกรมสำหรับ Ultrasonic Sensor	39
3.3.3 โปรแกรมสำหรับ Module Bluetooth	40
3.3.4 โปรแกรมสำหรับ DC Motor และ Servo Motor	44
3.3.5 โปรแกรมสำหรับ Encoder	45
3.3.6 การควบคุมแบบ PID	47
3.3.7 โปรแกรมสำหรับหุ่นยนต์เดินตามมนุษย์อัตโนมัติ	49

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการทดสอบ	53
4.1 การทดลองการเคลื่อนไหวของหุ่นยนต์	53
4.1.1 การเลี้ยวของล้อหน้า	53
4.1.2 การควบคุมความเร็วของมอเตอร์	55
4.1.3 การวิเคราะห์ของ Ultrasonic Sensor	57
4.2 การทดลองการเคลื่อนที่ติดตามมนุษย์ของหุ่นยนต์	58
บทที่ 5 สรุปผลการดำเนินโครงการและข้อเสนอแนะ	60
5.1 สรุปผลการทดลอง	60
5.2 ปัญหาที่เกิดจากการทดลอง	61
5.3 ข้อเสนอแนะ	61
เอกสารอ้างอิง	62
ภาคผนวก ก รายละเอียดของ Ultrasonic Sensor	65
ภาคผนวก ข รายละเอียดของ บอร์ดไฟโอ (FiO Board)	68
ภาคผนวก ค รายละเอียดของ aMG IMU-9A	76
ภาคผนวก ง รายละเอียดของ aMG Bluetooth	80
ประวัติผู้จัดทำ	84

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการศึกษาโครงการ	3
2.1 การหาค่า K_p , T_i และ T_d ด้วยวิธี ZN (วิธีการที่หนึ่ง)	17
2.2 การหาค่า K_p , T_i และ T_d ด้วยวิธี ZN (วิธีการที่สอง)	18
2.3 ตารางแสดงการควบคุมสถานะมอเตอร์ A	25
2.4 ตารางแสดงการควบคุมสถานะมอเตอร์ B	25
4.1 ตารางบันทึกการเลี้ยวของล้อ	53
4.2 ตารางบันทึกผลการควบคุมความเร็วมอเตอร์	55
4.3 ตารางบันทึกผลการทดลองการวัดระยะของ Ultrasonic Sensor	57
4.4 ตารางการบันทึกสถานะการติดตามของหุ่นยนต์	58



สารบัญรูป

รูปที่	หน้า
2.1 ระบบพิกัด	5
2.2 ระบบพิกัด orientation based	6
2.3 ลักษณะการการเชื่อมต่ออุปกรณ์แบบ PC BUS	7
2.4 การสื่อสารแบบ UART	7
2.5 เซอร์โวมอเตอร์(SG91R (9g))	8
2.6 ความกว้างพัลส์	8
2.7 เซนเซอร์ชนิดอัตราไซคลิก	9
2.8 การสะท้อนกลับของคลื่นเสียงจากวัสดุที่เป็นของแข็งและของเหลว	9
2.9 แสดงบอร์ด FIO Std	10
2.10 หลักการสร้างสัญญาณ PWM	11
2.11 แสดงระบบควบคุม	12
2.12 Process Variable (PV) vs time โดยเปลี่ยนค่า K_p (ค่า K_i และ K_d คงที่)	12
2.13 Process Variable (PV) vs time โดยเปลี่ยนค่า K_i (ค่า K_p และ K_d คงที่)	13
2.14 Process Variable (PV) vs time โดยเปลี่ยนค่า K_d (ค่า K_p และ K_i คงที่)	14
2.15 ผลตอบสนองรูปตัว S	16
2.16 แผนภาพกล่องของระบบวงปิดที่มีตัวควบคุม	17
2.17 T ที่มีระยะเวลาการสั้นอย่างต่อเนื่อง	17
2.18 Rotary Encoder	19
2.19 หลักการทำงานของ Rotary Encoder	19
2.20 aMG IMU-9A	20
2.21 SCHEMATIC DIAGRAM aMG IMU-9A	21
2.22 aMG Bluetooth – A	22
2.23 SCHEMATIC DIAGRAM aMG Bluetooth – A	22
2.24 วงจรขับมอเตอร์ L298N	24
2.25 SCHEMATIC DIAGRAM L298N	25
2.26 รูปวงจร ET-MINI 3 T0 5 TTL	26
2.27 SCHEMATIC DIAGRAM ET-MINI 3T05 TTL	27
2.28 DC Motor	27

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.1 แผนผังระบบของการทดลอง	29
3.2 แผนภาพการเชื่อมต่อทั้งหมด	30
3.3 การเชื่อมต่อ Compass Module	31
3.4 การเชื่อมต่อวงจร LSM303DLM	32
3.5 การเชื่อมต่อ Ultrasonic Sensor	32
3.6 การเชื่อมต่อของ โมดูลกลตุลฑูร	33
3.7 การเชื่อมต่อเซอร์โวมอเตอร์	34
3.8 การเชื่อมต่อของ Rotary Encoder	34
3.9 รูปวงจร ET-MINI 3T05 TT	35
3.10 การเชื่อมต่อวงจรจับมอเตอร์	36
3.11 แผนภาพการทำงานหุ่นยนต์	36
3.12 กล้องคำสั่ง PC Master	37
3.13 การส่งสัญญาณแบบ PC	38
3.14 การอ่านข้อมูลจาก โมดูลเข็มทิศ	38
3.15 โปรแกรมการอ่านค่ามุมหน่วยเป็นองศา (degree)	39
3.16 โปรแกรมการอ่านค่าระยะห่าง	40
3.17 โปรแกรมการรับค่ามุมจากเซ็นเซอร์บนสมาร์ตโฟน	41
3.18 กำหนดการส่งข้อมูล	41
3.19 แสดงโปรแกรม myCompass	43
3.20 โปรแกรมการรับสัญญาณกลตุลฑูร	44
3.21 แสดงภาพบล็อก PWM ที่ใช้สร้างสัญญาณพัลส์วึมอดูลเกซัน	44
3.22 แสดงภาพการส่งค่าให้กับบล็อก PWM	45
3.23 กราฟการทำ FFT หาความถี่ตัดผ่าน	45
3.24 คำสั่ง c2d	46
3.25 กราฟแสดงความเร็วก่อนและหลังใส่วงจรฟิลเตอร์	46
3.26 ระบบการควบคุมแบบ PID	47
3.27 ฟังก์ชัน ident	48
3.28 Simulate PID Controller	48

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.29 ค่า K_p , K_i , K_d และ n จากฟังก์ชัน PID Tuner	49
3.30 กราฟความเร็วที่ควบคุมแบบ PID	49
3.31 การเปลี่ยนแปลงของมุม	50
3.32 การเปลี่ยนแปลงของมุม	51
3.33 โปรแกรมของหุ่นยนต์เดินตามมนุษย์อัตโนมัติ	52
4.1 แสดงภาพขณะเดินซ้าย – ขวา และขณะเริ่มต้น	54
4.2 กราฟแสดง Step Repose เฉลี่ยของการทดลอง 10 ครั้ง และ กราฟ Step Repose ที่ได้จากการ Simulate บน โปรแกรม MATLAB	56
4.3 แสดงภาพการติดตามของหุ่นยนต์	59



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ในปัจจุบันหุ่นยนต์เป็นนวัตกรรมใหม่ที่น่าสนใจต่อมนุษย์ในยุคเทคโนโลยีเป็นอย่างมาก ซึ่งหุ่นยนต์อำนวยความสะดวกในด้านต่าง ๆ มีความสำคัญต่อการดำเนินชีวิตของมนุษย์อย่างมาก เช่นนำไปใช้ในการทดแทนแรงงานมนุษย์ ใช้ลดความเสี่ยงอันตรายในการทำงาน ใช้ลดเวลาในการทำงาน ด้วยเหตุนี้ผู้ศึกษาจึงมีความสนใจในการศึกษาและพัฒนาหุ่นยนต์เคลื่อนที่ตามคนอัตโนมัติ ซึ่งจะเป็นอุปกรณ์หนึ่งที่ช่วยอำนวยความสะดวกให้เราได้

หุ่นยนต์เคลื่อนที่ตามคนอัตโนมัตินี้ ถูกพัฒนาโดยใช้ไมโครคอนโทรลเลอร์ Fio STD ออกแบบบนโปรแกรมแมตแลบ (MATLAB) ใช้อุปกรณ์รับส่งสัญญาณที่มีความไวต่อเสียง (Ultrasonic Sensor) และแอปพลิเคชันตรวจจับทิศทาง (Android Compass Application) ซึ่งอยู่บนโทรศัพท์สมาร์ทโฟนที่มีระบบปฏิบัติการแอนดรอยด์ มาควบคุมการทำงานของเซอร์โวมอเตอร์ และมอเตอร์ ซึ่งในการออกแบบนี้ จะต้องมีระบบควบคุมการทำงานของอุปกรณ์ที่เหมาะสม และจะต้องมีระบบป้อนกลับที่จะช่วยรักษาความเร็วในการเคลื่อนที่ของหุ่นยนต์ เพราะในการควบคุมการเคลื่อนที่ของหุ่นยนต์นั้น จะเกิดสิ่งต่าง ๆ ที่ไม่คงที่และรบกวนการทำงานของอุปกรณ์อยู่ตลอดเวลา เช่น ความเร็วและทิศทางของมนุษย์ หรือวัตถุอื่น ๆ ที่ผ่านเข้ามาซึ่งไม่ใช่สิ่งที่ต้องการตรวจจับ

โครงการหุ่นยนต์เคลื่อนที่ตามคนอัตโนมัตินี้ ช่วยอำนวยความสะดวกให้ผู้ใช้ ทั้งยังลดอุบัติเหตุในกรณีต่าง ๆ. และยังสามารถนำไปประยุกต์ใช้เป็นหุ่นยนต์อำนวยความสะดวกในรูปแบบอื่น ๆ ได้

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อสร้างหุ่นยนต์เคลื่อนที่ตามคนอัตโนมัติ

1.2.2 เพื่อออกแบบระบบควบคุมอัตโนมัติ ควบคุมการทำงานของหุ่นยนต์เคลื่อนที่ตามคนอัตโนมัติ

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 ได้หุ่นยนต์ต้นแบบเคลื่อนที่ตามคนอัตโนมัติ
- 1.3.2 ได้ระบบควบคุมอัตโนมัติที่ใช้ในการควบคุมหุ่นยนต์เคลื่อนที่ตามคนอัตโนมัติ

1.4 ขอบเขตการทำโครงการ

- 1.4.1 ควบคุมหุ่นยนต์ให้สามารถเคลื่อนที่ตามอุปกรณ์สมาร์ทโฟนที่มีระบบปฏิบัติการแอนดรอยด์เวอร์ชัน 2.3.6 หรือสูงกว่าได้อัตโนมัติ
- 1.4.2 พื้นที่หุ่นยนต์เดินเป็นพื้นราบ เรียบ โลง ไม่มีคนเข้ามารบกวน

1.5 ขั้นตอนการทำโครงการ

- 1.5.1 ศึกษาค้นคว้างานวิจัยที่เกี่ยวข้องและหลักการทำงานของหุ่นยนต์เดินตามอัตโนมัติ
- 1.5.2 ศึกษาและทดลองใช้โปรแกรม MATLAB Block Set ของ rapidSTM32 และโปรแกรม Eclipse Android Development tool
- 1.5.3 จัดซื้ออุปกรณ์ที่ใช้ในการทำโครงการ
- 1.5.4 ศึกษาค้นคว้าและทำการเชื่อมต่ออุปกรณ์กับ MATLAB
- 1.5.5 เขียนโปรแกรมควบคุม
- 1.5.6 สร้างชุดขับเคลื่อนอิเล็กทรอนิกส์และโครงสร้างหุ่น
- 1.5.7 ทดสอบการใช้งานจริง
- 1.5.8 ตรวจสอบและแก้ไขข้อบกพร่อง

1.6 แผนการดำเนินงาน

กิจกรรม	พ.ศ. 2557					พ.ศ. 2558			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1.ศึกษาค้นคว้าข้อมูลในการทำโครงการ									
2.ศึกษาโปรแกรมที่ใช้ในการทำโครงการ									
3.ศึกษาการเชื่อมต่ออุปกรณ์									
4.เขียนโปรแกรมควบคุม									
5. ทำการสร้างชุดขับเคลื่อนอิเล็กทรอนิกส์ และ โครงสร้างหุ่น									
6. ทดสอบการใช้งานจริง									
7. ตรวจสอบพัฒนาและแก้ไขข้อบกพร่อง									
8.สรุปผลการทำโครงการและจัดทำรายงาน									

1.7 รายละเอียดงบประมาณตลอดโครงการ

ค่าอุปกรณ์ในการดำเนินโครงการ	1,400	บาท
ค่าเอกสารที่ใช้ในการดำเนินโครงการ	300	บาท
ค่าเช่าเล่มโครงการ	300	บาท
รวมเป็นเงินทั้งสิ้น	2,000	บาท

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้กล่าวถึงการควบคุมหุ่นยนต์ ประกอบด้วยการควบคุม 2 ระบบคือ ระบบควบคุมใน ส่วนของผู้ใช้งาน และระบบควบคุมในส่วนของหุ่นยนต์

ระบบควบคุมในส่วนของผู้ใช้งาน มีหน้าที่รับคำสั่งตำแหน่งของผู้ใช้งาน ส่งไปให้ระบบควบคุม ในส่วนของหุ่นยนต์ เพื่อเปรียบเทียบกับตำแหน่งหุ่นยนต์กับผู้ใช้งาน โดยมีหลักการและทฤษฎีที่เกี่ยวข้อง ดังนี้

2.1 ระบบควบคุมส่วนผู้ใช้ (User Control System)

ในระบบควบคุมส่วนผู้ใช้งานจะมีสมาร์ทโฟนที่มีระบบปฏิบัติการแอนดรอยด์เป็นตัวหลักใน การทำงาน โดยใช้โปรแกรม Eclipse for Android ในการพัฒนาโปรแกรม โดยจะทำการรับค่ามุมจาก แอปพลิเคชันตรวจจับทิศทาง (Android Compass Application) แล้วส่งค่าไปให้ระบบควบคุมหุ่นยนต์ อีกครั้งเพื่อส่งให้ไมโครคอนโทรลเลอร์เปรียบเทียบกับค่ากับ โมดูลเข็มทิศดิจิทัลแล้วส่งให้หุ่นยนต์ เคลื่อนที่ไปยังเป้าหมายที่ต้องการ

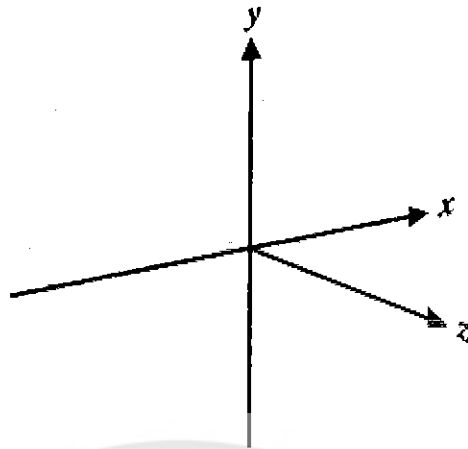
2.1.1 ระบบปฏิบัติการแอนดรอยด์

ระบบปฏิบัติการแอนดรอยด์ (Android Operating System) [2] เป็นชื่อเรียกชุดซอฟต์แวร์ หรือแพลตฟอร์ม (Platform) สำหรับอุปกรณ์อิเล็กทรอนิกส์ ที่มีหน่วยประมวลผลเป็นส่วนประกอบ แอนดรอยด์นั้น นำมาลดทอนขนาด เพื่อให้เหมาะสมแก่การนำไปติดตั้งบนอุปกรณ์พกพา ที่มีขนาด พื้นที่จัดเก็บข้อมูลที่จำกัด

2.1.2 แอปพลิเคชันตรวจจับทิศทาง (Android Compass Application)

แอปพลิเคชันตรวจจับทิศทาง[1] เป็นแอปพลิเคชันเข็มทิศแม่เหล็กที่ไว้ใช้ชี้บอกทิศทาง โดยตัว แอปพลิเคชันมีรูปแบบการใช้งานเรียบง่าย โดยการทำงานหลัก ๆ ตัวแอปพลิเคชันนั้นจะใช้เซนเซอร์ แม่เหล็กภายในทำงาน

โดยคำนวณจากความเอียงของเมตริกซ์ (inclination matrix I) และเมตริกซ์หมุน (rotation matrix R) แปลงเวกเตอร์จากระบบพิกัด[16]



รูปที่ 2.1 ระบบพิกัด [16]

โดยที่ X คือ ผลคูณเวกเตอร์ Y, Z (แนวสัมผัสพื้นดินชี้ไปทางทิศตะวันออก)
 Y คือ แนวสัมผัสพื้นดินชี้ไปทางทิศเหนือ
 Z คือ แนวสัมผัสพื้นดินชี้ขึ้นฟ้าตั้งฉากกับพื้นดิน

$$[0 \ 0 \ g] = R * \text{gravity} (g) \quad (2.1)$$

$$[0 \ m \ 0] = I * R * \text{geomagnetic} (m) \quad (2.2)$$

โดยที่ R คือ อาร์เรย์ ขนาด 9 rotation matrix R และ R สามารถเป็นค่า NULL ได้

I คือ อาร์เรย์ ขนาด 9 rotation matrix I และ I สามารถเป็นค่า NULL ได้

Gravity คือ อาร์เรย์ 3 ค่าที่ได้จาก sensor accelerometer

Geomagnetic คือ อาร์เรย์ 3 ค่าที่ได้จาก sensor magnetic field

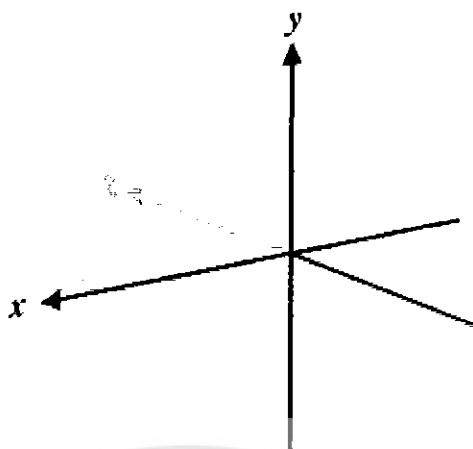
นำเมทริกซ์ R (Rotation Matrix) มาคำนวณจะได้ 3 ค่าและค่าทั้งหมดอยู่ในมุมมองศา

ค่า (0) : มุมทิศ (azimuth), มุมระหว่างทิศเหนือแม่เหล็ก และแกน y รอบแกน z (0-359)°

0° คือทิศเหนือ 90° คือทิศตะวันออก 180° คือทิศใต้ และ 270° คือทิศตะวันตก

ค่า (1) : การเอนเอียง (pitch), การหมุนไปยังแกน x (-180° to 180°), ด้วยค่าบวก เมื่อ แกน z
 เคลื่อนไปยังแกน y

ค่า (2) : การหมุน (roll), การหมุนไปยังแกน x (-90° to 90°) ที่เพิ่มขึ้นเป็นอุปกรณ์เคลื่อนที่ตาม
 เช็มนาฬิกา



รูปที่ 2.2 ระบบพิกัด orientation based [16]

2.2 ระบบควบคุมส่วนหุ่นยนต์ (Mobile Robot Control System)

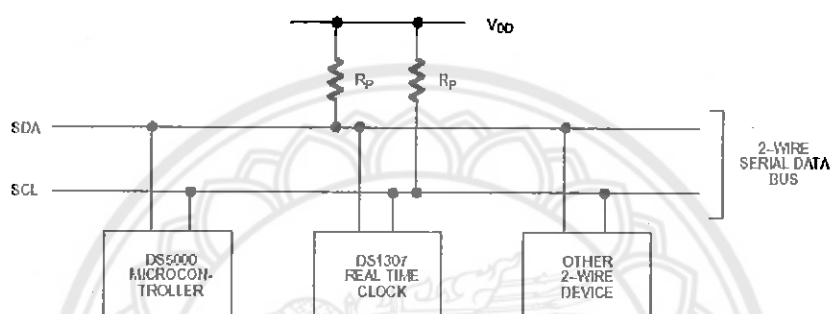
ในระบบควบคุมส่วนหุ่นยนต์จะมีไมโครคอนโทรลเลอร์ Fio STD [7] เป็นตัวหลักในการประมวลผล โดยใช้โปรแกรมเมทแลบ เวอร์ชัน R21012a [22] ซึ่งจะมีกล่องคำสั่งของ rapidstm32 [7] ทำหน้าที่สร้างระบบควบคุมหุ่นยนต์ และใช้ Keil Arm [11] เป็นคอมไพเลอร์

เมื่อระบบควบคุมส่วนหุ่นยนต์ได้รับค่าระยะทางระหว่างตัวหุ่นยนต์กับผู้ใช้งานจากอุปกรณ์รับส่งสัญญาณอัลตราโซนิก (Ultrasonic Sensor) แล้วจะนำค่าไปควบคุมให้มอเตอร์หมุนด้วยความเร็วที่เหมาะสม และนำไปกำหนดการทำงานของเซอร์โวมอเตอร์ ให้ทำงานตามการคำนวณของค่ามุมที่ได้รับจากแอปพลิเคชันตรวจจับทิศทาง

ซึ่งค่ามุมที่ได้นั้นมาจากการเชื่อมต่อไมโครคอนโทรลเลอร์เข้ากับ โมดูลเข็มทิศดิจิทัล (Digital Compass) ซึ่งใช้การติดต่อบนระบบ I2C นำมาเปรียบเทียบกับค่ามุมที่ได้มาจากแอปพลิเคชันเข็มทิศ (Android Compass Application) โดยไมโครคอนโทรลเลอร์ติดต่อกับสมาร์ทโฟนผ่านบลูทูธ (Bluetooth) บนระบบ UART

2.2.1 การสื่อสารแบบ I²C สำหรับการติดต่อกับโมดูลเข็มทิศดิจิทัล

I²C (Inter-IC Communication) [20] หมายถึง การติดต่อสื่อสารระหว่างไอซี บัส I²C เชื่อมต่อกับไมโครคอนโทรลเลอร์ โดยใช้สายสัญญาณ 2 เส้น ได้แก่ ขา SDA (รับและส่งข้อมูล) และ SCL (ขาสัญญาณนาฬิกา) โดยขาสัญญาณทั้งสองจะต้องต่อตัวต้านทานพูลอัพไว้เพื่อกำหนดสถานะลอจิก "1" ให้กับระบบบัส



รูปที่ 2.3 ลักษณะการการเชื่อมต่ออุปกรณ์แบบ I²C BUS [20]

2.2.2 การสื่อสารแบบ UART สำหรับการติดต่อกับบลูทูธ

UART [24] (Universal Asynchronous Receiver Transmitter) หมายถึง รูปแบบการส่งข้อมูลที่ ถูกกำหนดขึ้นมาเพื่อใช้รับส่งข้อมูลแบบ อะซิงโครนัส เริ่มต้นจาก Start Bit เป็น Logic 0 จากนั้นจะตามด้วย Data ที่เราส่ง แล้วจะถูกปิดด้วย STOP Bit เป็น Logic 1 UART สามารถรับส่งข้อมูลได้ทั้งแบบ Half-Duplex (การส่งแบบทิศทางเดียว) และ Full-Duplex (สามารถรับและส่งข้อมูลได้ในคราวเดียวกัน)

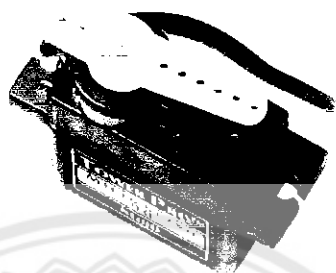


รูปที่ 2.4 การสื่อสารแบบ UART [24]

2.2.3 เซอร์โวมอเตอร์

เซอร์โวมอเตอร์ [6] คือ มอเตอร์ไฟฟ้ากระแสตรง (DC Motor) ที่ถูกประกอบรวมกับชุดเกียร์ และส่วนควบคุมต่าง ๆ ไว้ในโมดูลเดียวกันหรือภายในกล่องพลาสติกเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายสัญญาณ 3 สายคือ VCC, GND และสายสัญญาณควบคุม (Control line) มอเตอร์ชนิดนี้สามารถ

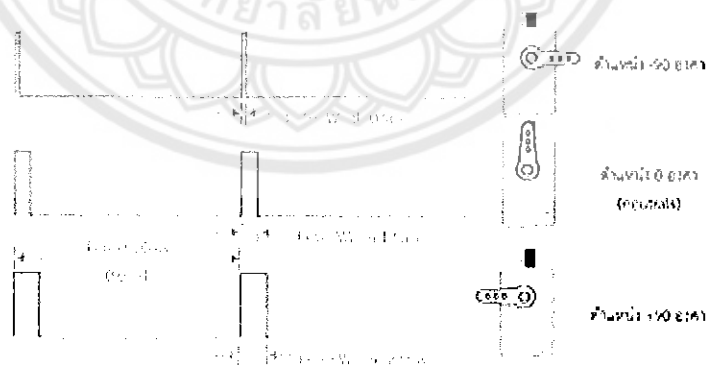
ควบคุมให้หมุนไปตำแหน่งที่ต้องการ ได้โดยอาศัยสัญญาณความกว้างพัลส์ที่ป้อนให้กับมอเตอร์แต่ใน เซอร์โวตัวนี้จะหมุนได้เพียง 180°



รูปที่ 2.5 เซอร์โวมอเตอร์ (SG91R (9g)) [6]

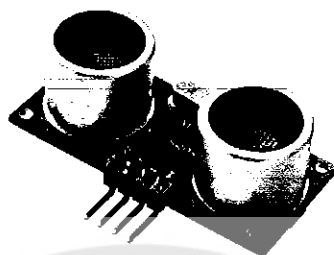
หลักการทำงานของเซอร์โวมอเตอร์

การควบคุมการทำงานของเซอร์โวมอเตอร์ทำได้โดยการป้อนสัญญาณความกว้างพัลส์ให้กับ มอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนจะขึ้นอยู่กับความกว้างพัลส์ โดยทั่วไปแล้วความกว้างพัลส์จะมี ลักษณะดังรูปที่ 2.6 ความกว้างพัลส์



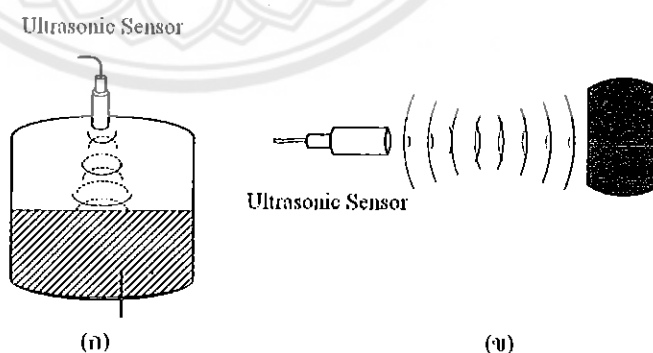
รูปที่ 2.6 ความกว้างพัลส์ [6]

2.2.4 Ultrasonic Sensor



รูปที่ 2.7 เซนเซอร์ชนิดอัลตราโซนิก [4]

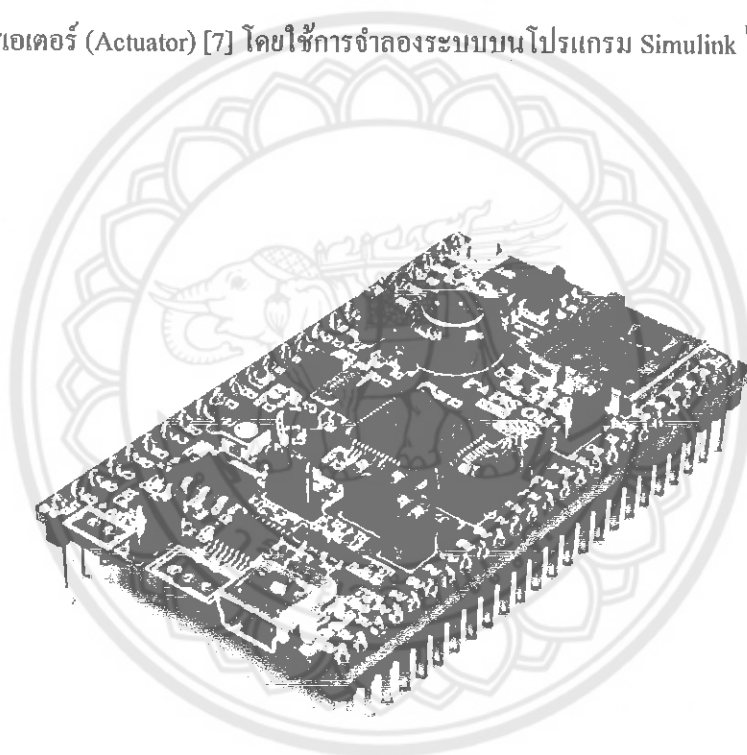
เซนเซอร์ชนิดใช้เสียง หรือเซนเซอร์ชนิดอัลตราโซนิก (ultrasonic sensor) [4] เป็นเซนเซอร์ (sensor) ที่ทำงานโดยอาศัยคลื่นเสียงที่มีความถี่สูงกว่า 20 กิโลเฮิรต์ (kHz) ซึ่งเป็นคลื่นในย่านที่มนุษย์ไม่สามารถได้ยินเสียง เซนเซอร์ชนิดอัลตราโซนิกทำงานโดยอาศัยการกระจาย หรือการเคลื่อนที่ของคลื่นเสียงไปกระทบกับพื้นผิวของตัวกลาง ซึ่งอาจเป็นของแข็งหรือของเหลว บางส่วนของคลื่นเสียงจะแทรกผ่านเข้าไปในตัวกลางนั้น และส่วนใหญ่ของคลื่นความถี่สูงนี้จะสะท้อนกลับเรียกว่า "Echo" โดยช่วงเวลาของการสะท้อนกลับของคลื่นเสียงเป็นสัดส่วนโดยตรงกับระยะห่างระหว่างวัตถุกับเซนเซอร์ (ดังรูปที่ 2.8)



รูปที่ 2.8 การสะท้อนกลับของคลื่นเสียงจากวัตถุที่เป็นของแข็งและของเหลว
(ก) การตรวจจับระดับความสูงของของเหลว (ข) การตรวจจับระยะห่างของวัตถุ [4]

2.2.5 บอร์ดไฟโอ (FIO Board)

การควบคุมหุ่นยนต์อาศัยการควบคุมโดยไมโครคอนโทรลเลอร์ (Microcontroller) ในโครงการนี้เลือกใช้บอร์ดไฟโอ (FIOBoard) [7] ซึ่งเป็นชุดทดลองระบบสมองกลฝังตัวที่ถูกออกแบบมาเพื่อใช้ร่วมกับชุดคำสั่งราปิดเอสทีเอ็ม 32 (RapidSTM32 Blockset) สามารถใช้ร่วมกับซิมูลิงค์ มีจุดเด่นคือ การเขียนโปรแกรมแบบกราฟิก (Graphic Programming) โดยอาศัยโค้ด เจเนอเรชัน (Code Generation) ในการแปลงโปรแกรมแบบกราฟิกลงบนบอร์ดไฟโอ และยังสามารถทำ Hardware in the Loop ได้ นั่นคือ สามารถทดสอบอัลกอริทึมและโปรแกรมก่อนที่จะต้องซื้ออุปกรณ์เซ็นเซอร์ (Sensor) และแอกชูเอเตอร์ (Actuator) [7] โดยใช้การจำลองระบบบนโปรแกรม Simulink ได้

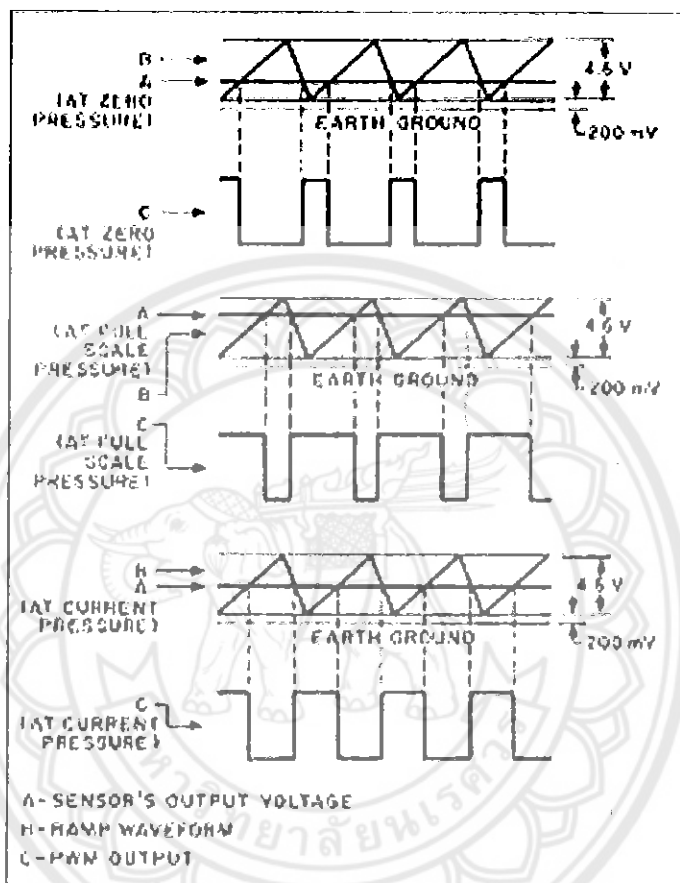


รูปที่ 2.9 แสดงบอร์ด FIO Std [7]

2.2.6 หลักการสร้างสัญญาณ PWM

PWM (Pulse Width Modulation) [10] เป็นสัญญาณที่เกิดจากการผสมกันระหว่างสัญญาณรูปสามเหลี่ยม (Triangle wave) กับระดับแรงเคลื่อนสัญญาณ ไฟดีซีที่มีการตั้งค่าความถี่คงที่ แต่การเปิด (สัญญาณมีสถานะ High) และการปิด (สัญญาณมีสถานะเป็น Low) เมื่อความกว้างของรูปคลื่นที่เป็น High และ Low มีความกว้างต่างกัน โดยคิดเป็นเปอร์เซ็นต์ 0-100 % ค่าเฉลี่ยของแรงดัน เอาท์พุทเมื่อผ่านการกรองก็จะมีค่ามากขึ้นเป็นเปอร์เซ็นต์เหมือนกัน ซึ่งลักษณะของการทำงานจะเป็นการทำงานแบบสวิทช์ ซึ่งทำให้กระแสที่ไหลผ่านอุปกรณ์อิเล็กทรอนิกส์กำลังมีการสูญเสียพลังงานในรูปแบบของ

พลังงานความร้อนน้อยลง แต่การสูญเสียซึ่งเกิดการสวิตช์มีเพิ่มมากขึ้น, การทำให้ค่าเอาต์พุตของแรงดันมีค่าการกระเพื่อมมากหรือน้อยขึ้นอยู่กับเปลี่ยนแปลงค่าความถี่ โดยส่วนมาก PWM จะทำงานที่ความถี่สวิตช์ 10kHz ขึ้นไป (หรือที่ความถี่ที่มนุษย์ไม่ได้ยิน 20 KHz)



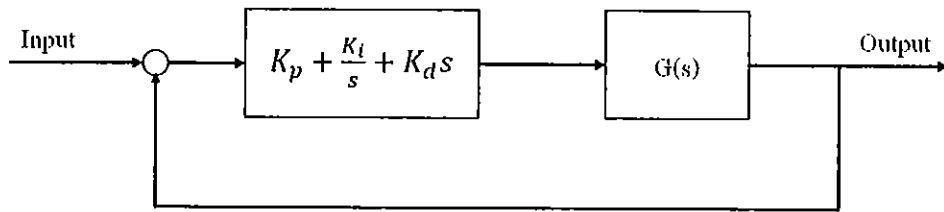
รูปที่ 2.10 หลักการสร้างสัญญาณ PWM [10]

2.2.7 ทฤษฎีคอนโทรลเลอร์แบบ PID

การควบคุมแบบ PID นั้นประกอบด้วยเทอมสำหรับการปรับแต่ง 3 เทอม รวมกันเป็น Manipulated Variable (MV)

$$MV(t) = P(t) + I(t) + D(t) \quad (2.3)$$

โดยที่ $P(t)$, $I(t)$ และ $D(t)$ เป็นเอาต์พุตจากแต่ละเทอมตามลำดับ



รูปที่ 2.11 แสดงระบบควบคุม [19]

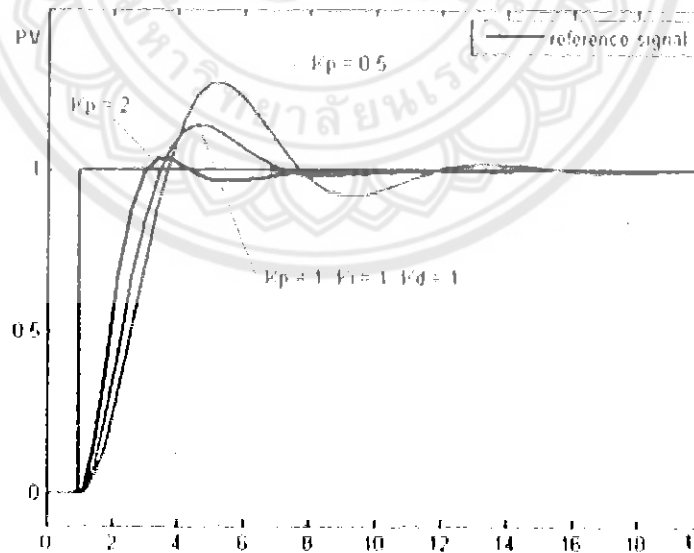
เทอม Proportional

เทอม Proportional เปลี่ยนแปลงตามอัตราส่วนของค่า Error ปัจจุบัน ซึ่งค่า Pout สามารถ

คำนวณได้โดยการนำค่า Error มาคูณกับค่าคงที่ K_p จากรูป 2.11 ถ้า $G(s) = \frac{K_p}{s + \frac{1}{T_{P1}}}$ (2.4)

$$P(t) = K_p e(t) \quad (2.5)$$

ค่า K_p ที่สูง จะเป็นผลให้ค่าเอาต์พุตมากขึ้นตาม หากค่า K_p มากเกินไประบบจะไม่คงที่และเกิดการกวัดแกว่ง (Oscillation) และในทางตรงข้ามหากค่า K_p น้อยเกินไปอาจทำให้ระบบตอบสนองช้าเกินไป

รูปที่ 2.12 Process Variable (PV) vs time โดยเปลี่ยนค่า K_p (ค่า K_i และ K_d คงที่) [3]

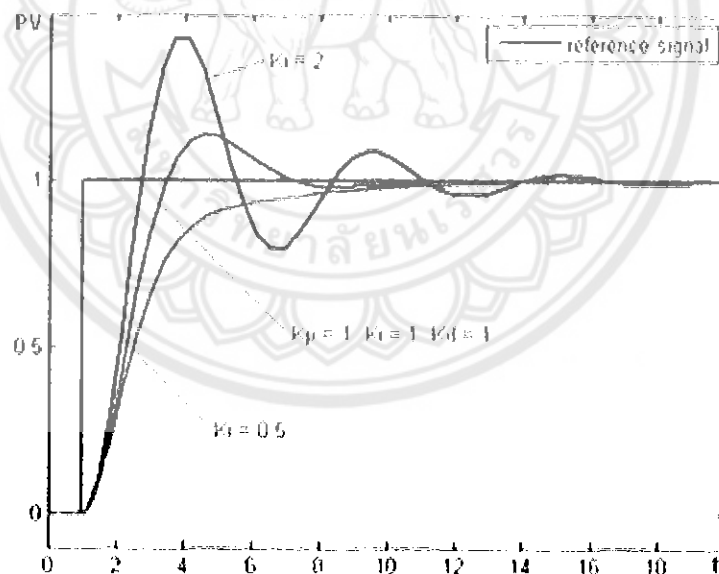
ในกรณีระบบที่ไม่มี Disturbance การใช้เทอม Proportional อย่างเดียวจะทำให้ระบบเกิดการแกว่ง (Oscillate) รอบ ๆ Set point (SP) จะไม่อยู่ที่ SP พอดี

เทอม Integral

เทอม Integral เป็นอัตราส่วนของค่า Error สะสมในหนึ่งช่วงเวลา (ปัจจุบัน ซ้อนไปในอดีต) ค่า $I(t)$ เกิดจากผลคูณของค่าคงที่ K_i กับผลรวมของ $e(t)$ ซึ่งเป็นค่าสะสมของ Error ที่ควรจะต้องถูกแก้ไขมาก่อนหน้านี้

$$I(t) = K_i \int_0^t e(t) dt \quad (2.6)$$

เทอม Integral (เมื่อใช้ร่วมกับเทอม Proportional) จะช่วยเร่งให้ระบบวิ่งเข้าหา SP เร็วขึ้น และช่วยลด Error ที่เกิดจากการใช้เทอม Proportional อย่างเดียว อย่างไรก็ตาม เนื่องจากว่าเทอม Integral นั้นเกิดจากการคำนวณ โดยรวม Error ที่เกิดขึ้นในอดีตด้วย อาจจะทำให้เกิด Overshoot เกิน SP ในค่าปัจจุบันด้วย



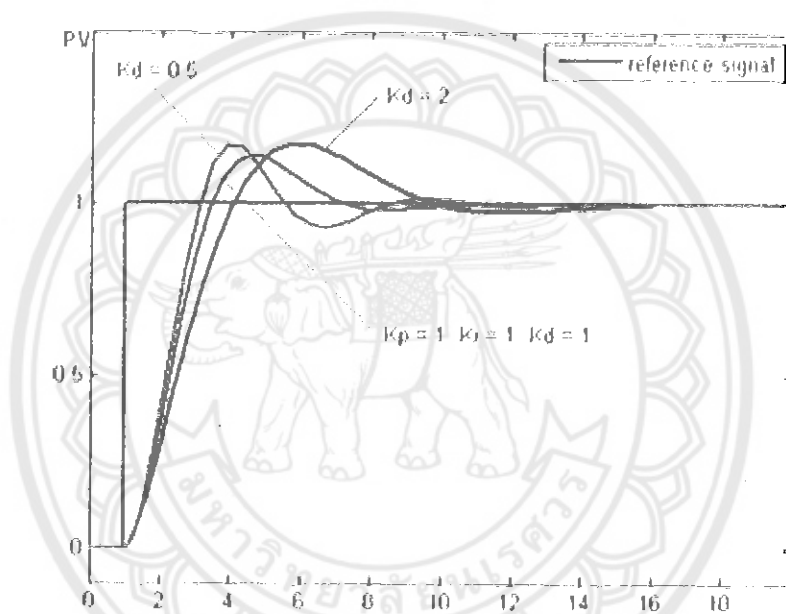
รูปที่ 2.13 Process Variable (PV) vs time โดยเปลี่ยนค่า K_i (ค่า K_p และ K_d คงที่) [3]

เทอม Derivative

อัตราการเปลี่ยนแปลงของความผิดพลาดจากกระบวนการนั้นคำนวณหาจากความชันของความผิดพลาดทุก ๆ เวลา (นั่นคือ เป็นอนุพันธ์อันดับหนึ่งสัมพันธ์กับเวลา) และคูณกับค่าคงที่ K_d ก็จะได้

เทอม Derivative

$$D(t) = K_d \frac{de(t)}{dt} \quad (2.7)$$



รูปที่ 2.14 Process Variable (PV) vs time โดยเปลี่ยนค่า K_d (ค่า K_p และ K_i คงที่) [3]

เทอม Derivative จะช่วยดึงเอาที่พุ่งจากคอนโทรลเลอร์ให้ช้าลง ซึ่งจะเห็นผลได้ชัดเมื่อ PV เข้าใกล้ Set Point ฉะนั้นเทอม Derivative จึงช่วยลดอาการ Overshoot ซึ่งเกิดจากเทอม Integral และช่วยปรับปรุงเสถียรภาพของระบบ

รวมสามเทอมเข้าด้วยกัน

เมื่อรวมเทอม Proportional, Integral, และ Derivative เข้าด้วยกันก็จะได้ เอาท์พุทจากคอนโทรลเลอร์ PID ดังนี้

$$MV(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.8)$$

โดยค่าพารามิเตอร์ต่าง ๆ คือ

ค่า Proportional gain, K_p : ค่าที่มากขึ้นหมายถึงการตอบสนองที่เร็วขึ้น เพราะค่า Error ยิ่งมาก ค่าชดเชยจากเทอมนี้ก็จะมากขึ้นตาม ค่า Gain ที่มากเกินไปจะนำไปสู่ระบบที่ไม่คงที่ และการแกว่งกวัด (Oscillation)

ค่า Integral gain, K_i : ค่าที่มากขึ้นหมายถึง ค่า error แบบ Steady-state จะถูกกำจัดได้เร็วขึ้น ข้อเสียก็คือ Overshoot ค่า Error ที่เป็นลบจะต้องถูกแก้ด้วย Error ที่เป็นบวก ก่อนที่ระบบจะเข้าสู่ Steady-state

ค่า Derivative gain, K_d : ค่าที่มากขึ้นหมายถึงขนาด Overshoot ที่ลดลง แต่ก็อาจทำให้การตอบสนองช้าลงบ้าง และอาจนำไปสู่ความไม่คงตัวของระบบเนื่องจาก Noise ได้

การปรับแต่งค่า (Loop Tuning)

ในการปรับแต่งค่ามี 3 วิธีหลักดังนี้

การปรับแต่งด้วยมือ (Manual tuning)

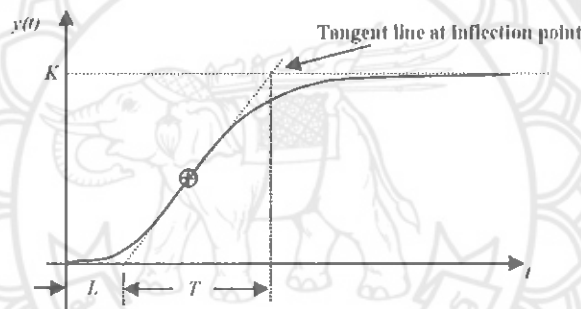
จะเริ่มโดยการเซตค่า K_i และ K_d เป็นศูนย์ และค่อยๆ เพิ่มค่า K_p เรื่อย ๆ จนกระทั่งระบบเริ่มเกิดการแกว่ง ให้เซตค่า K_p เป็นครึ่งหนึ่งของค่านั้น เริ่มเพิ่มค่า K_i จนได้เวลาการตอบสนองของระบบที่ต้องการเพิ่มค่า K_d จนกระทั่งการควบคุมเร็วพอที่ยอมรับได้ โดยเทียบกับเมื่อระบบมี Disturbance อ้างอิง รายละเอียดการปรับแต่งค่า Gain ต่าง ๆ ด้วย

การปรับแต่งด้วยวิธี Ziegler–Nichols

วิธีนี้ถูกนำเสนอโดย John G. Ziegler และ Nathaniel B. Nichols ในปี 1940 เป็นวิธีการที่ใช้สำหรับปรับปรุง และพัฒนาตัวควบคุมเพื่อให้ระบบควบคุมมีผลตอบสนองที่ดีไม่เกิดค่าพุ่งสูงสุดมากนัก

วิธีการที่หนึ่ง

ในวิธีการนี้ใช้ได้กับระบบที่ไม่มีตัวปริพันธ์ และผลตอบสนองของระบบต้องอยู่ในรูปของตัว S เท่านั้นดังรูปที่ 2.15 โดยทำการป้อนสัญญาณเข้าแบบสัญญาณขั้นบันไดหนึ่งหน่วย แล้ววัดผลตอบสนองที่ได้



รูปที่ 2.15 ผลตอบสนองรูปตัว S [19]

จากผลตอบสนองในรูปที่ 2.15 มีพารามิเตอร์ที่สำคัญอยู่ 3 ตัวคือ K, L และ T ถูกกำหนดโดยลากเส้นสัมผัสกับผลตอบสนองที่จุดเปลี่ยนเว้าดังรูปที่ 2.15 ดังนั้นระบบ $G(s)$ สามารถประมาณได้ด้วยระบบรวมกับ Transport lag ดังนี้

$$G(s) = \frac{Y(s)}{U(s)} \cong \frac{Ke^{-Ls}}{Ts+1} \quad (2.9)$$

จากพารามิเตอร์ L และ T ที่ได้จากผลตอบสนอง ซิกเลอร์และนิโคลส์ได้แนะนำวิธีการหาค่าพารามิเตอร์ K_P , T_i และ T_d ตามตารางที่ 2.1

ตารางที่ 2.1 การหาค่า K_p , T_i และ T_d ด้วยวิธี ZN (วิธีการที่หนึ่ง) [19]

Controller	K_p	T_i	T_d
P-Control	$\frac{T}{L}$	∞	0
PI-Control	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID-Control	$1.2 \frac{T}{L}$	$2L$	$0.5L$

โดยฟังก์ชันถ่ายโอนของตัวควบคุม PID คือ

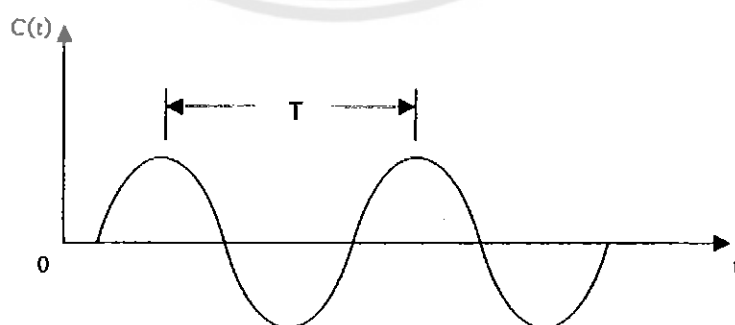
$$G(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = K_p + \frac{K_i}{s} + K_d s \quad (2.10)$$

วิธีการที่สอง

วิธีนี้ใช้กับระบบวงปิดดังรูปที่ 2.16 โดยทำการปรับค่า K จนถึงค่าวิกฤต K^* ซึ่งเป็นค่าที่ทำให้ผลตอบสนองต่อสัญญาณเข้าแบบฟังก์ชันขั้นบันไดหนึ่งหน่วยเกิดการแกว่งกวัด (Oscillation) จากนั้นนำผลตอบสนองที่ได้มาวัดคาบ T ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 แผนภาพกล่องของระบบวงปิดที่มีตัวควบคุม [19]



รูปที่ 2.17 T ที่มีระยะเวลาการสั่นอย่างต่อเนื่อง

นำค่า K^* และ T มาคำนวณหาพารามิเตอร์ K_p , T_i และ T_d ตามตารางที่ 2.2

ตารางที่ 2.2 การหาค่า K_p , T_i และ T_d ด้วยวิธี ZN (วิธีการที่สอง) [19]

Controller	K_p	T_i	T_d
P-Control	$0.5K^*$	∞	0
PI-Control	$0.45K^*$	$0.833T$	0
PID-Control	$0.6K^*$	$0.5T$	$0.125T$

การปรับแต่งด้วยซอฟต์แวร์ MATLAB [23]

กระบวนการปรับแต่งการควบคุมแบบ PID ตามหลักการนี้ ทำงานโดยปรับแต่ง PID เพื่อให้ได้รับความสมดุลระหว่างประสิทธิภาพการทำงานและความทนทาน โดยเริ่มต้นจาก ขั้นตอนการเลือกค่าความถี่ตัดผ่าน (วงความกว้างของแถบคลื่นความถี่) ซึ่งอยู่บนพื้นฐานของแผนพลวัต และออกแบบมาสำหรับเป้าหมายที่มีขอบเขตมุม 60° เมื่อปรับเปลี่ยน เวลาการตอบสนอง, ความกว้างของแถบคลื่นความถี่, การตอบสนองแบบชั่วคราว หรือการใช้ขอบเขตมุมในการปรับแต่ง PID กระบวนการนี้ จะทำให้ได้ ค่า PID ใหม่

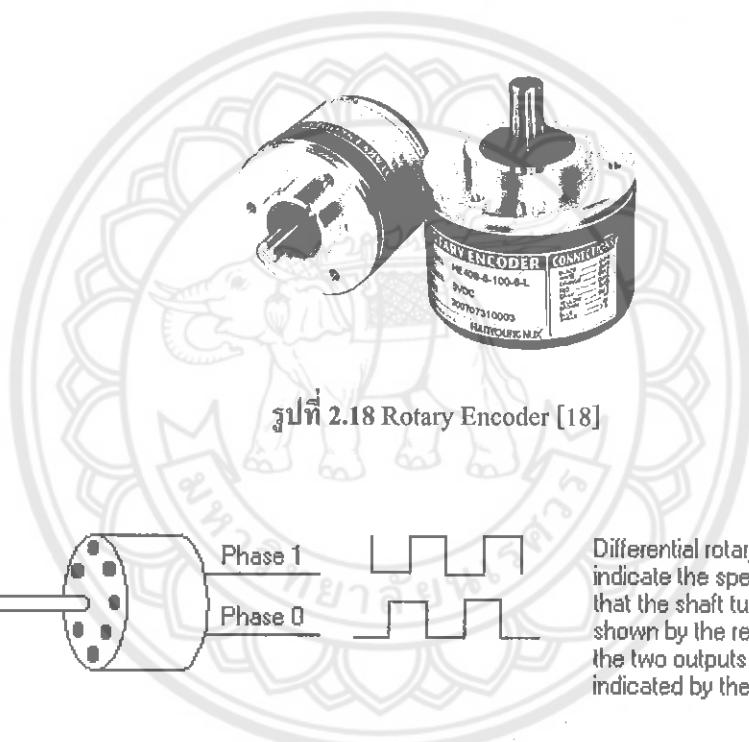
เพื่อกำหนดความทนทาน (ขอบเขตมุมขั้นต่ำ) ขั้นตอนปรับแต่งจะเลือกการออกแบบการควบคุมที่ปรับสมดุลระหว่างสองหลักการของประสิทธิภาพการทำงาน ได้แก่ การติดตามการอ้างอิง และการปฏิเสธการรบกวน และสามารถเปลี่ยนการมุ่งเน้นการออกแบบเพื่อสนับสนุนหนึ่งในหลักการการทำงานนี้ โดยใช้ตัวเลือก DesignFocus ของ pidtune ที่ในบรรทัดคำสั่ง หรือ ตัวเลือกกล่องโต้ตอบใน PID tuner

เมื่อปรับเปลี่ยนรูปแบบการออกแบบแล้ว กระบวนการนี้จะปรับอัตราการสนับสนุน การติดตามการอ้างอิง หรือการปฏิเสธการรบกวน ขณะเดียวกันก็ทำให้ได้ขอบเขตมุมขั้นต่ำ พารามิเตอร์ที่สามารถปรับได้ส่วนมากจะอยู่ในระบบ จะมีโอกาสมากขึ้นเมื่อขั้นตอนของ PID สามารถรับความต้องการการปรับแต่งโดยปราศจากการลดความทนทานได้ตัวอย่างเช่น การตั้งค่าการออกแบบ จุดมุ่งเน้น มีแนวโน้มที่จะมีผลต่อการควบคุม PID มากกว่าการควบคุม P หรือ PI จากกรณีทั้งหมดนี้

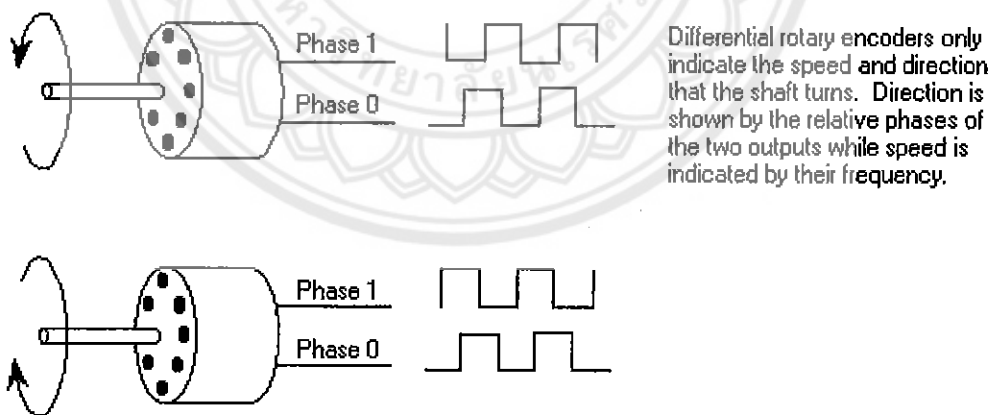
การปรับแต่งประสิทธิภาพการทำงานของระบบขึ้นอยู่กับคุณสมบัติของแผนงานเป็นอย่างมาก
 ในบางแผนงานการปรับเปลี่ยนการออกแบบการมุงเน้น มีผลค่อนข้างน้อย หรือ ไม่มีเลย

2.2.8 Rotary Encoder

เป็นอุปกรณ์ที่ใช้ในการนับรอบการหมุนของมอเตอร์โดย Encoder ที่ใช้จะเป็น Rotary Encoder
 [18] การหมุนทุก ๆ หนึ่งรอบจะทำให้เกิดสัญญาณพัลส์ทั้งหมด 400 พัลส์ และมีสายสัญญาณ 2 สาย
 แต่ละสายสัญญาณมีเฟสต่างกัน 90° จึงทำให้สามารถบอกทิศทางการหมุนได้



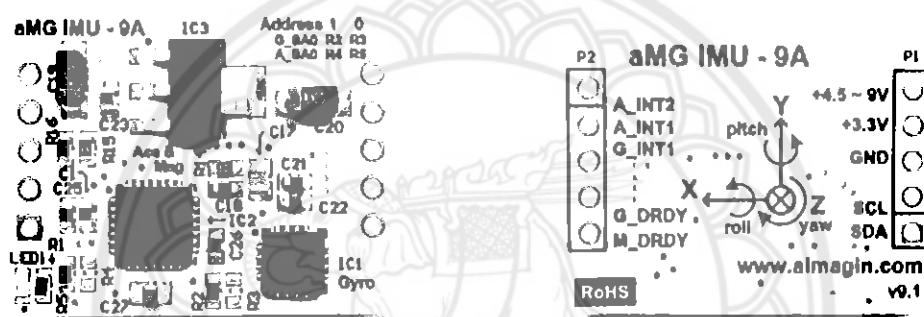
รูปที่ 2.18 Rotary Encoder [18]



รูปที่ 2.19 หลักการทำงานของ Rotary Encoder [18]

2.2.9 aMG IMU-9A

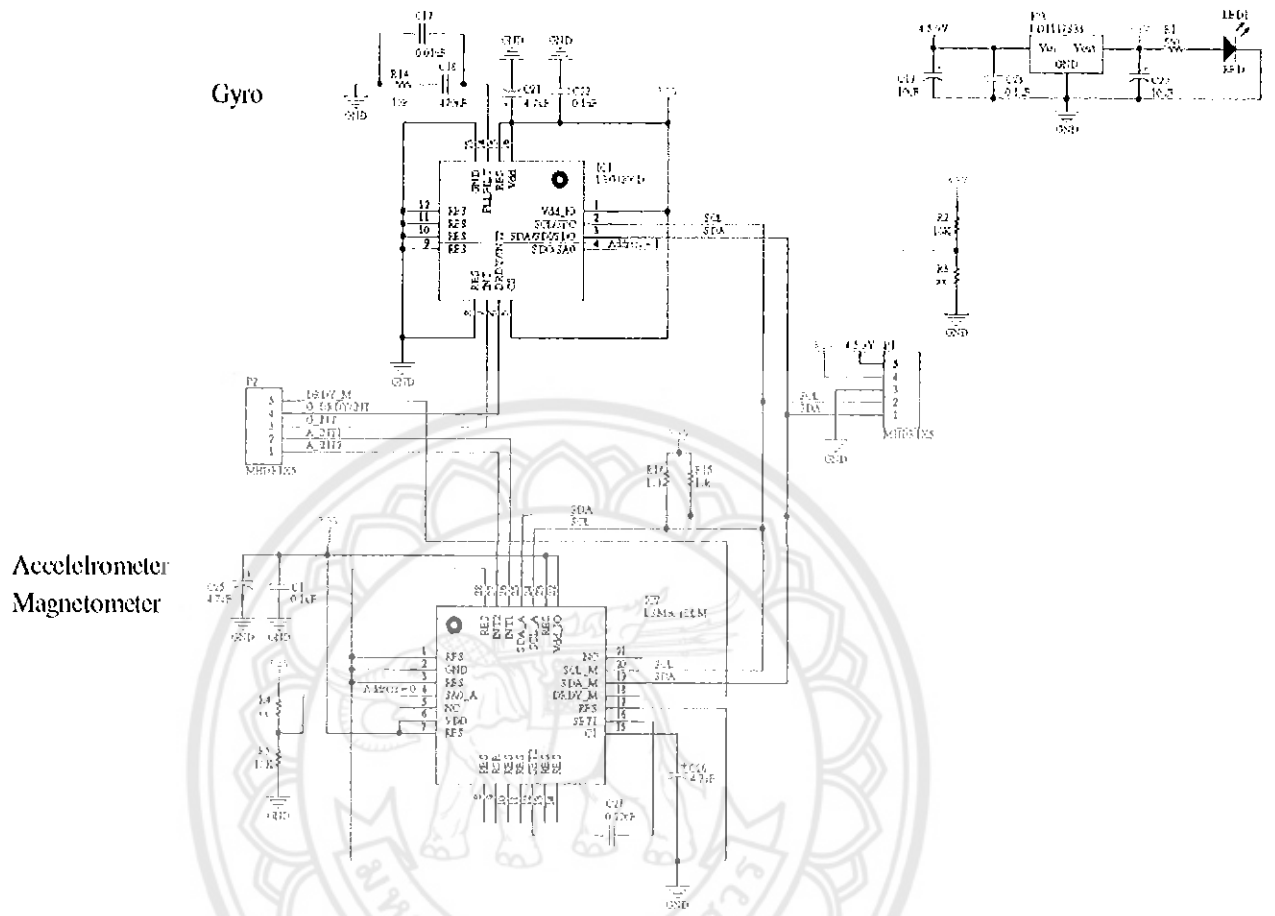
เนื่องจากการออกแบบหุ่นยนต์ติดตามมนุษย์จำเป็นต้องทำให้หุ่นยนต์วิ่งไปในทิศทางที่มนุษย์เดินไปได้ ซึ่งมีการเลือกใช้ compass sensor (aMG IMU-9A) [9] เพื่อสามารถระบุทิศทางได้ Compass sensor (aMG IMU-9A) [9] จะประกอบไปด้วย accelerometer, gyro meter และ magnetometer เพื่อหามุมเป็นองศาจำเป็นต้องใช้ accelerometer และ magnetometer เพื่อคำนวณหาค่ามุมองศา ซึ่ง Compass sensor จะส่งสัญญาณแบบ I²C Accelerometer ทำหน้าที่ วัดความเร่งในแนวแต่ละแกน ส่วน Magnetometer ทำหน้าที่วัดความเข้มของสนามแม่เหล็ก



(ก) ด้านหน้า

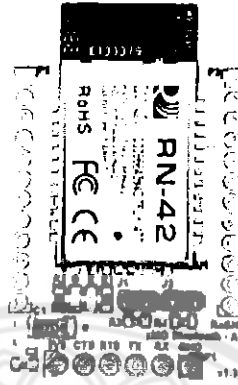
(ข) ด้านหลัง

รูปที่ 2.20 aMG IMU-9A [9]

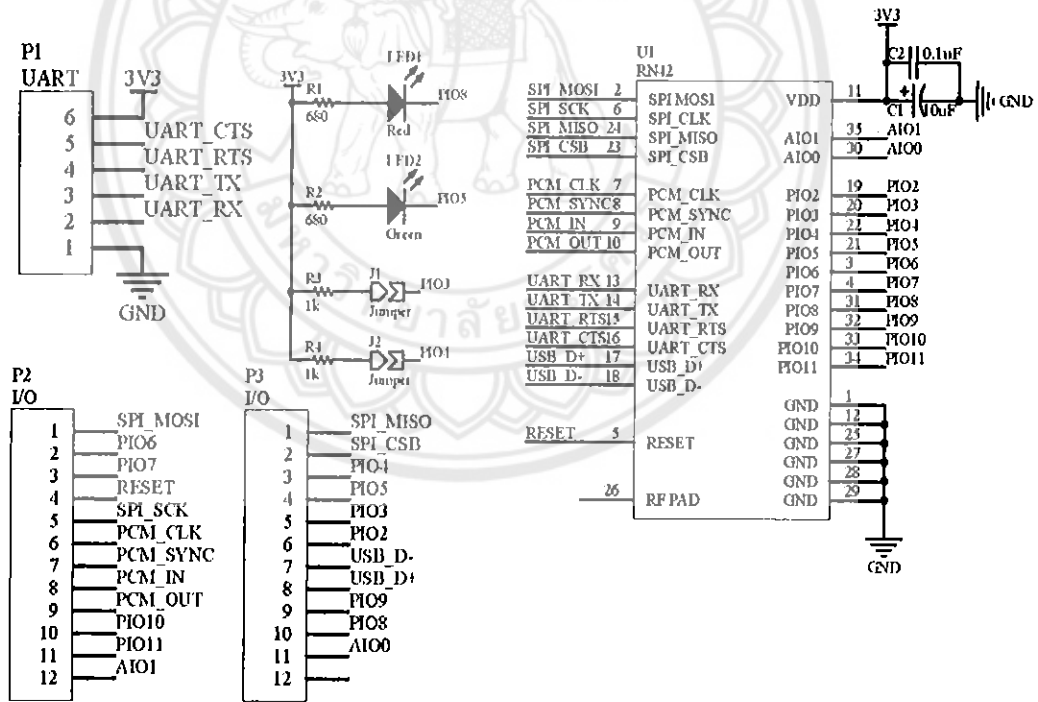


รูปที่ 2.21 SCHEMATIC DIAGRAM aMG IMU-9A [9]

2.2.10 aMG Bluetooth



รูปที่ 2.22 aMG Bluetooth – A [8]



รูปที่ 2.23 SCHEMATIC DIAGRAM aMG Bluetooth – A [8]

บลูทูธเป็นเทคโนโลยีของอินเทอร์เน็ตเฟสทางคลื่นวิทยุ ใช้ในการเชื่อมโยงการสื่อสารไร้สายในแถบความถี่ 2.45 GHz ทำให้อุปกรณ์อิเล็กทรอนิกส์ที่เคลื่อนย้ายได้สามารถติดต่อสื่อสารกันแบบไร้สายระหว่างกันในระยะห่างสั้น ๆ ได้ อุปกรณ์แต่ละตัวสามารถติดต่อสื่อสารกับอุปกรณ์อื่น ๆ ได้สูงสุดถึง 7 ตัวพร้อมกัน ซึ่งเรียกเครือข่ายขนาดย่อมนี้ว่า พิคอนเน็ต (Piconet) ซึ่งไปกว่านั้น อุปกรณ์แต่ละตัวยังสามารถสังกัดอยู่กับเครือข่ายพิคอนเน็ตได้หลายเครือข่ายพร้อมกัน เทคโนโลยีการส่งคลื่นวิทยุของบลูทูธจะใช้การกระโดดเปลี่ยนความถี่ (Frequency Hop) เพราะเทคโนโลยีนี้เหมาะที่จะใช้กับการส่งคลื่นวิทยุที่มีกำลังส่งต่ำและราคาถูก โดยจะแบ่งออกเป็นหลายช่องความถี่ขนาดเล็ก ในระหว่างที่มีการเปลี่ยนช่องความถี่ที่ไม่แน่นอนทำให้สามารถหลีกเลี่ยงสัญญาณรบกวนที่เข้ามาแทรกแซงได้ บลูทูธจะใช้คลื่นความถี่ในการติดต่อในย่าน ISM (Industrial, Scientific, Medical) ที่มีความถี่ 2.4 GHz และใช้พลังงานต่ำ โดยทางปฏิบัติแล้วอุปกรณ์ของบลูทูธนั้นจะมีพื้นที่การใช้งานไม่เกิน 10 เมตร โดยการติดต่อผ่านทางช่องสัญญาณที่สนับสนุนทั้งข้อมูลและเสียงที่มีความเร็ว 741Kbps

โมดูลการทำงานของบลูทูธ (Bluetooth Module) [5]

บลูทูธถูกกำหนดให้มีโครงสร้างการทำงานจำนวน 8 ชั้น ซึ่งมากกว่าโมดูล OSI อยู่ 1 ชั้น ทำให้ขอบเขตการทำงานในแต่ละชั้น แตกต่างจาก โมดูล OSI แต่ลำดับการทำงานมีลักษณะเหมือนกัน โดยแต่ละชั้นของโมดูลบลูทูธมีชื่อและหน้าที่การทำงานดังนี้

ชั้นที่ 1 Radio เป็นส่วนที่เกิดการรับและส่งคลื่นวิทยุจริง ๆ ในวงจรรีเลย์ภาคส่งรับคลื่นวิทยุที่ถูกควบคุมจากชั้น Base Band ไม่ว่าจะมีความถี่และระดับความแรงของสัญญาณที่ใช้รวมไปถึงเฟรมข้อมูลที่จะส่ง

ชั้นที่ 2 Base Band หน้าที่หลักของชั้นนี้ คือ การควบคุมวงจรรีเลย์ภาคส่ง - รับคลื่นวิทยุที่อยู่ชั้นล่างสุด ซึ่งจุดสำคัญที่สุดของการควบคุม ก็คือการเลือกช่องความถี่ในการรับส่งข้อมูลให้ตรงกันระหว่างมาสเตอร์และสเลฟที่ต้องมีการกระโดดไปในรูปแบบเดียวกัน

ชั้นที่ 3 Link Controller ควบคุมการเชื่อมต่อพื้นฐานของบลูทูธทั้งหมด ไม่ว่าจะเป็ในสถานะของอุปกรณ์ โหมดการทำงานของอุปกรณ์ การค้นหาอุปกรณ์บลูทูธใกล้เคียง รวมไปถึงจนถึงการเลือกว่าจะเป็นมาสเตอร์หรือสเลฟในสภาพแวดล้อมต่าง ๆ

ชั้นที่ 4 Link Manager ทำหน้าที่แปลงคำสั่งที่ได้รับจากชั้นบนเป็นลำดับหน้าที่การทำงานที่ชั้นล่างรู้จัก และคอยส่งคำสั่งลงไปควบคุมการทำงานของชั้นล่างทั้งหมด

ชั้นที่ 5 HCI (Host Control Interface) เป็นโปรโตคอลเชื่อมต่อระหว่างโปรแกรมชั้นบนที่ทำงานอยู่บนระบบหนึ่ง (เช่น การ์ด PCMCIA Bluetooth ที่ต่ออยู่ในเครื่องคอมพิวเตอร์โน้ตบุ๊ก) ทำให้โปรแกรมรู้จักคำสั่งควบคุมอุปกรณ์บลูทูธ

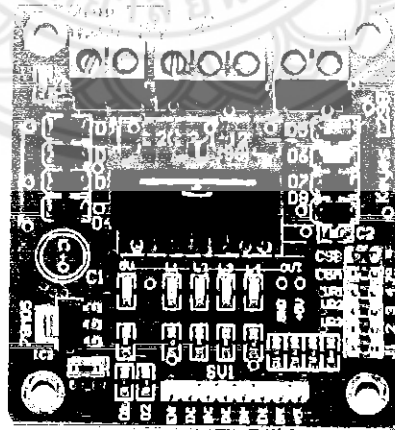
ชั้นที่ 6 L2CAP (Logical Link Control and Adaptation Protocol) ทำหน้าที่มีกลไกเพื่อกำหนดข้อมูลจากชั้นบนซึ่งอาจจะมีการทำงานของโปรแกรมหลายโปรแกรมพร้อมกัน และจัดแบ่งข้อมูลออกเป็นแพ็คเกจ (Package)

ชั้นที่ 7 RFCOMM/SDP สำหรับ RFCOMM เป็นโปรโตคอลเสมือนที่ทำให้แอปพลิเคชันด้านบนมองบลูทูธ เป็นเหมือนพอร์ตอนุกรม (Serial Port) ทัวไป ส่วน SDP (Service Discovery Protocol) เป็นโปรโตคอลที่ช่วยค้นหาบริการจากอุปกรณ์ บลูทูธตัวอื่นที่อยู่ในขอบเขต พิกอนเนตเดียวกัน

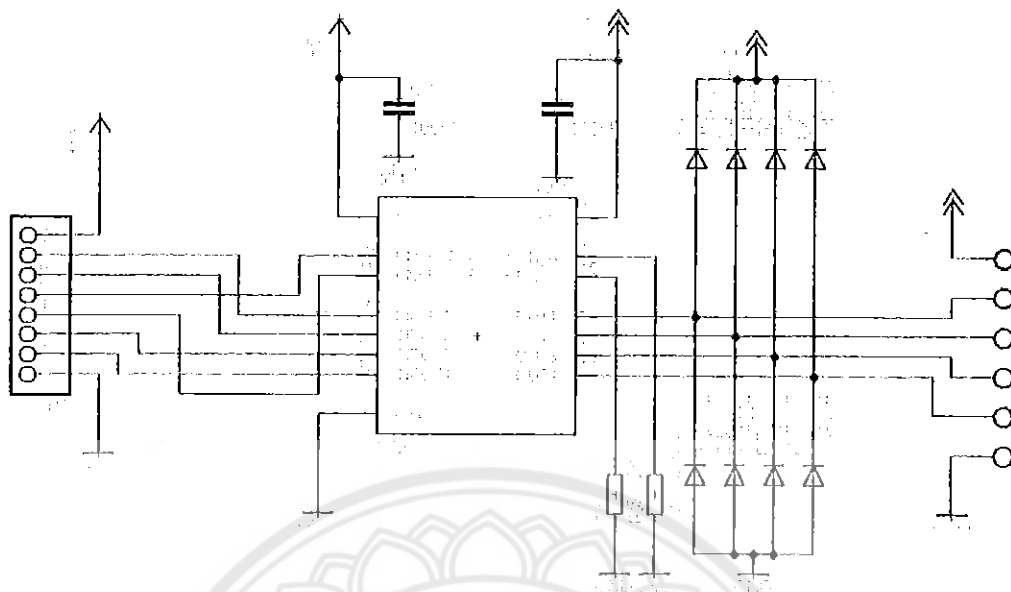
ชั้นที่ 8 Application เป็นส่วนของโปรแกรมที่ติดต่อบริการหรือส่งข้อมูลกับผู้ใช้

2.2.11 วงจรขั้วมอเตอร์ L298N

วงจรขั้วมอเตอร์สำเร็จ โดยใช้ IC L298 ซึ่งมีประสิทธิภาพสูง สามารถขั้วมอเตอร์ได้ 2 ชุด และสามารถขั้วกระแสได้สูงสุด 4A (เมื่อบริดจ์ 2CH เข้าด้วยกัน) ภายในบอร์ดได้ต่อ diode เพื่อกันไฟที่จะเข้ามาบรบกวนระบบ



รูปที่ 2.24 วงจรขั้วมอเตอร์ L298N [21]



รูปที่ 2.25 SCHEMATIC DIAGRAM L298N [17]

ตารางที่ 2.3 ตารางแสดงการควบคุมสถานะมอเตอร์ A [21]

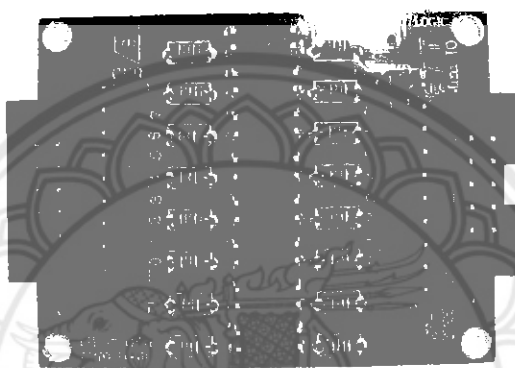
IN1	IN2	สถานะมอเตอร์ A
LOW	LOW	หยุด
HIGH	LOW	หมุนขวา
LOW	HIGH	หมุนซ้าย
HIGH	HIGH	หยุดเบรก

ตารางที่ 2.4 ตารางแสดงการควบคุมสถานะมอเตอร์ B [21]

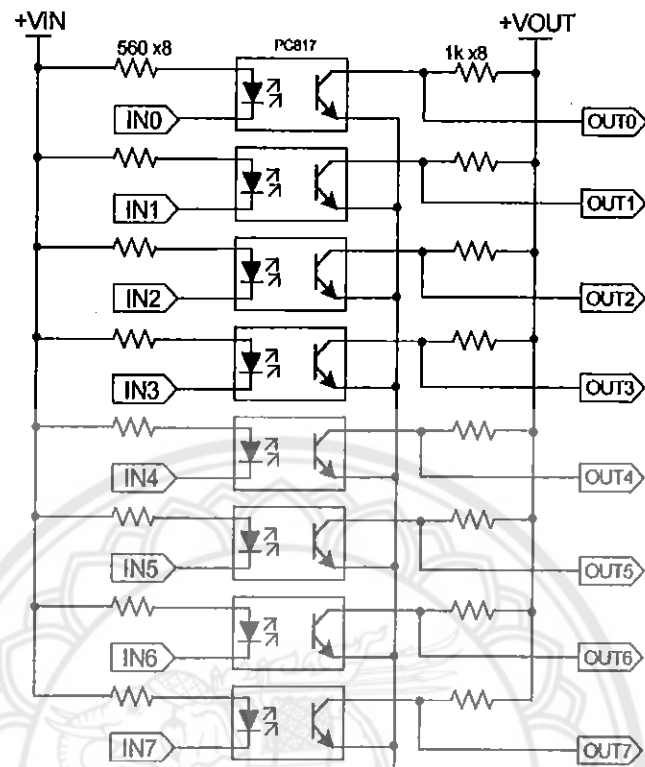
IN3	IN4	สถานะมอเตอร์ B
LOW	LOW	หยุด
HIGH	LOW	หมุนขวา
LOW	HIGH	หมุนซ้าย
HIGH	HIGH	หยุดเบรก

2.2.12 Module ET-MINI 3T05 TTL

Module ET-MINI 3T05 TTL เป็นชุด OPTO สำหรับแปลงระดับลอจิกจาก 3V เป็น 5V หรือจาก 5V เป็น 3V ซึ่งมีอยู่ด้วยกัน 8 ช่องสัญญาณ โดยวัตถุประสงค์หลักจะใช้กับ MCU ที่ใช้ไฟ 3.3 V ให้สามารถขับหรือรับโหลดที่ทำงานที่ 5V ได้ โดยทั้ง 8 ช่องสัญญาณของโมดูลนี้ จะใช้งานได้ 2 แบบ คือ ใช้งานเป็น Input อย่างเดียวหรือใช้งานเป็น Output อย่างเดียว ไม่สามารถใช้งานเป็น Input หรือ Output ปนกันได้ต้องเลือกใช้อย่างใดอย่างหนึ่ง

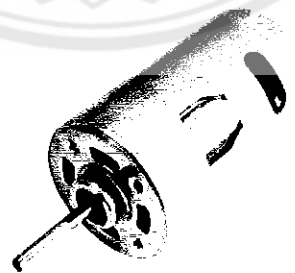


รูปที่ 2.26 รูปวงจร ET-MINI 3 T0 5 TTL [15]



รูปที่ 2.27 SCHEMATIC DIAGRAM ET-MINI 3T05 TTL [14]

2.2.13 DC Motor



รูปที่ 2.28 DC Motor [25]

การทำงานของมอเตอร์ไฟฟ้าส่วนใหญ่เกิดจากการทำงานร่วมกันระหว่างสนามแม่เหล็กของแม่เหล็กในตัวมอเตอร์ และสนามแม่เหล็กที่เกิดจากกระแสในขดลวดทำให้เกิดแรงดูดและแรงผลักของสนามแม่เหล็กทั้งสอง มอเตอร์ที่ใช้มีขนาด 12 V มีความเร็ว 8 รอบต่อวินาที

ในบทนี้นำเสนอหลักการและทฤษฎีต่าง ๆ ที่นำมาใช้กับอุปกรณ์ และการทำการทดลองในการทำโครงการนี้ ในส่วนของวิธีการทดลองนั้นจะนำเสนอในบทต่อไป



บทที่ 3

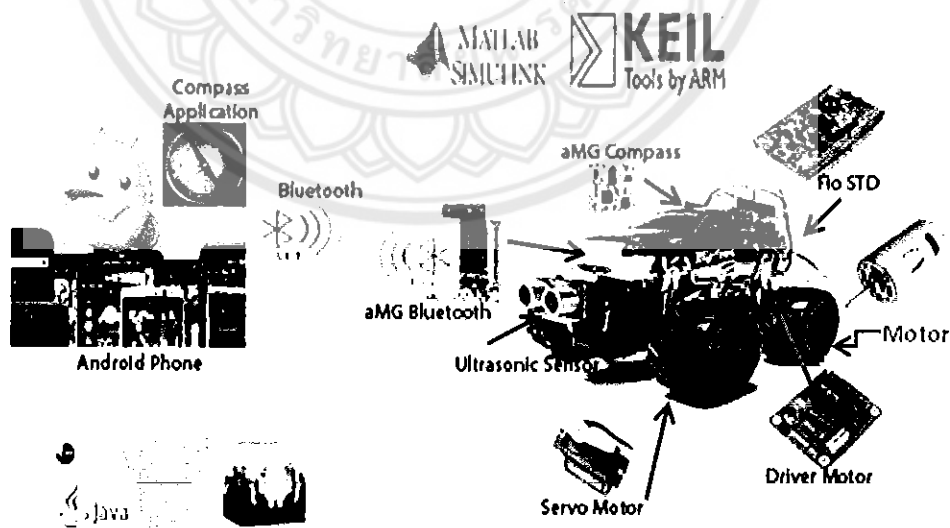
ขั้นตอนการดำเนินงาน

การติดตามทำได้โดยใช้อุปกรณ์รับส่งสัญญาณอัลตราโซนิก (Ultrasonic Sensor) วัดระยะห่างระหว่างหุ่นยนต์กับมนุษย์ และควบคุมความเร็วของมอเตอร์โดยใช้ระบบควบคุมแบบ PID ให้เคลื่อนที่ด้วยความเร็วเท่ากับที่ถูกติดตาม ส่วนของการควบคุมทิศทางเคลื่อนที่ของหุ่นยนต์ใช้อุปกรณ์โมดูลเข็มทิศ (Compass Module) เปรียบเทียบค่าของมุมกับแอปพลิเคชันตรวจจับทิศทางบนโทรศัพท์สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ เพื่อควบคุมให้หุ่นยนต์มีทิศทางเดียวกับทิศทางเคลื่อนที่ของผู้ถูกติดตาม และใช้บอร์ดทดลอง Fio STD ออกแบบด้วยโปรแกรมเมตแลบซิมูลิงค์ (MATLAB Simulink) เป็นอุปกรณ์ควบคุมหลัก

การออกแบบหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติโดยแบ่งออกเป็น 3 ส่วน คือ

- 3.1 โครงสร้างภายนอกของหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ
- 3.2 อุปกรณ์การทำงานภายในของหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ (Hardware)
- 3.3 โปรแกรมการทำงานหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ (Software)

3.1 โครงสร้างของหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ

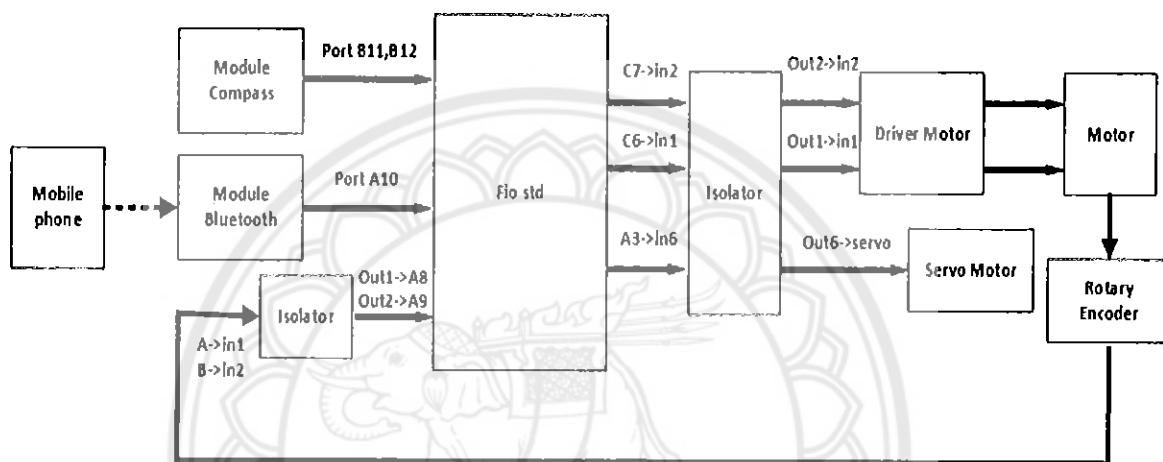


รูปที่ 3.1 แผนผังระบบของการทดลอง

3.2 การติดตั้งและเชื่อมต่ออุปกรณ์

ระบบการทำงานภายในของหุ่นยนต์ติดตามมนุษย์อัตโนมัติจะประกอบด้วยอุปกรณ์และวงจรต่าง ๆ ที่ทำงานร่วมกันโดยจะกล่าวถึงหน้าที่การทำงานการต่อวงจรต่าง ๆ เข้าด้วยกันเพื่อให้ระบบสามารถประสานการทำงานร่วมกันได้โดยจะกล่าวดังแผนภาพต่อไปนี้

โดยอุปกรณ์แต่ละชนิดมีหน้าที่การทำงานดังต่อไปนี้

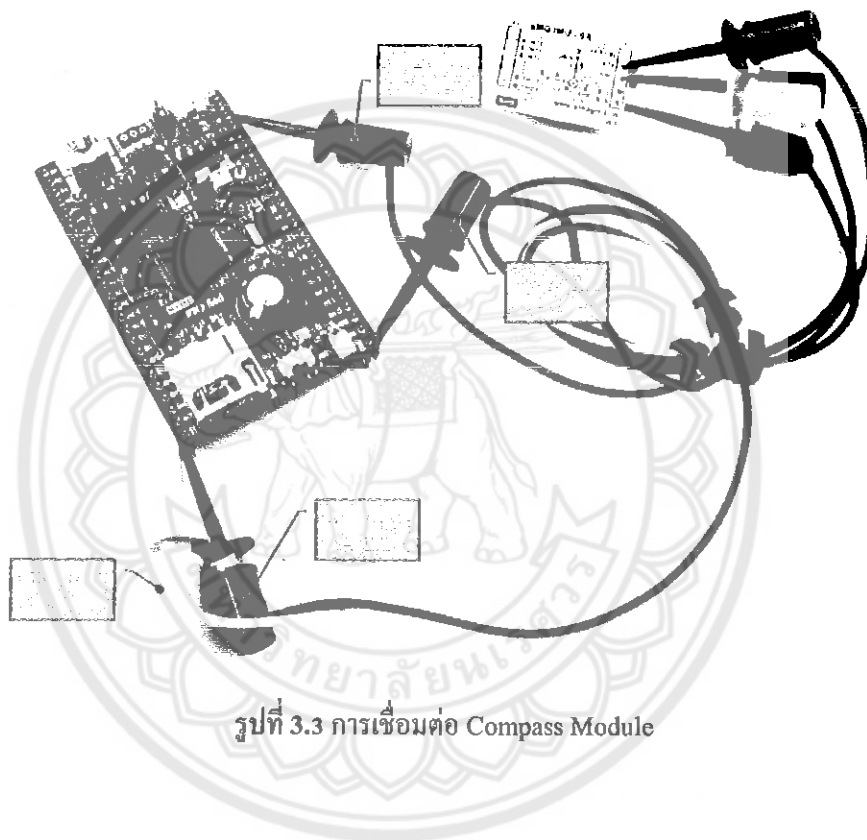


รูปที่ 3.2 แผนภาพการเชื่อมต่อทั้งหมด

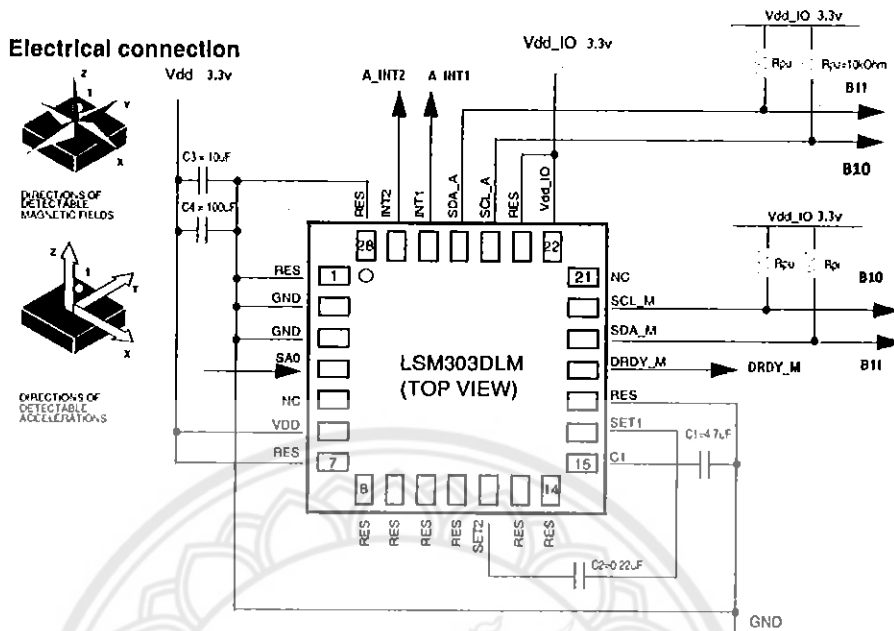
Compass Module	ทำหน้าที่ หาค่ามุมของหุ่นยนต์
Module Bluetooth	ทำหน้าที่ รับค่ามุมของจากสมาร์ตโฟน
Servo motor	ทำหน้าที่ บังคับเลี้ยว
Motor	ทำหน้าที่ ควบคุมความเร็วของหุ่นยนต์
Ultrasonic sensor	ทำหน้าที่ หาระยะห่างระหว่างมนุษย์กับหุ่นยนต์
Isolator	ทำหน้าที่ แปลงสัญญาณระหว่าง 3-5V
Rotary Encoder	ทำหน้าที่ วัดความเร็วของหุ่นยนต์

3.2.1 Compass Module

เนื่องจากการควบคุมทิศทางของหุ่นยนต์จำเป็นต้องใช้โมดูลเข็มทิศเพื่อเปรียบเทียบทิศทางกับแอปพลิเคชันเข็มทิศที่ส่งมาจากสมาร์ทโฟนผ่านระบบบลูทูธ โดยโมดูลเข็มทิศจะมีการเชื่อมต่อแบบ I²C ซึ่ง SDA จะต่อเข้ากับพอร์ต B11 และ SCL จะต่อเข้ากับพอร์ต B10 ดังรูป 3.3 และจำเป็นต้องมีไฟเลี้ยง 3.3 V ด้วย



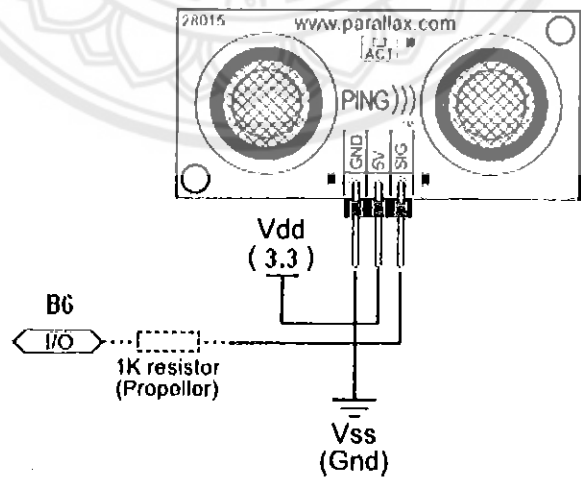
รูปที่ 3.3 การเชื่อมต่อ Compass Module



รูปที่ 3.4 การเชื่อมต่อวงจร LSM303DLM [12]

3.2.2 Ultrasonic Sensor

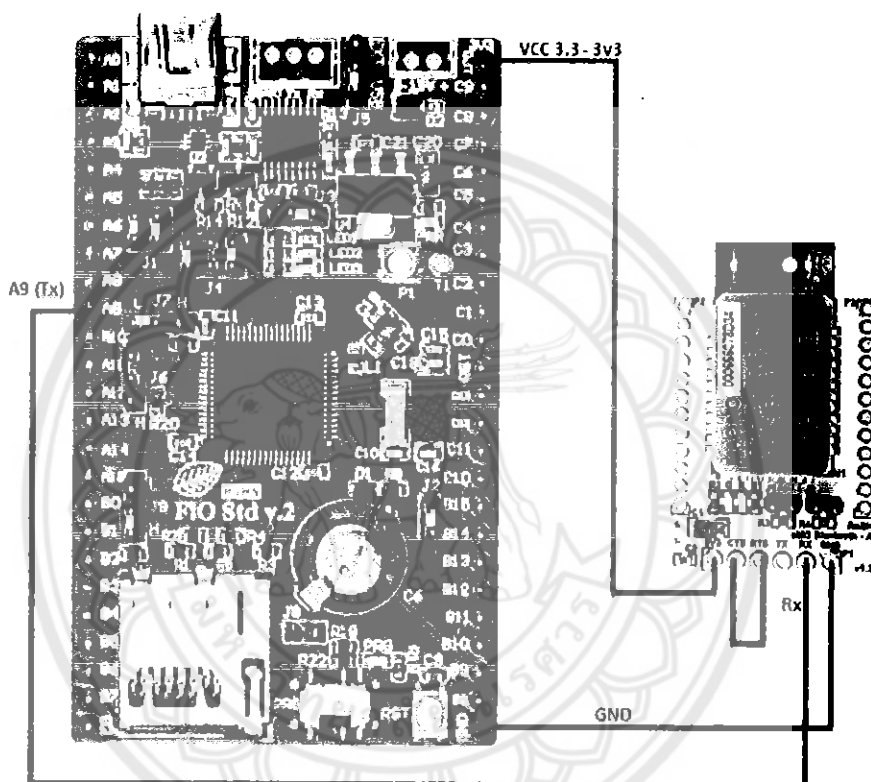
โดยการติดตั้ง Ultrasonic Sensor จำเป็นต้องติดตั้งไว้ที่ด้านหน้าของหุ่นยนต์เพื่อที่จะวัดระยะห่างระหว่างหุ่นยนต์กับมนุษย์ได้ง่ายขึ้น การเชื่อมต่อเซ็นเซอร์กับไมโครคอนโทรลเลอร์นั้นจะเป็นไปตามรูปที่ 3.5 การเชื่อมต่อ Ultrasonic Sensor



รูปที่ 3.5 การเชื่อมต่อ Ultrasonic Sensor

3.2.3 Module Bluetooth [16]

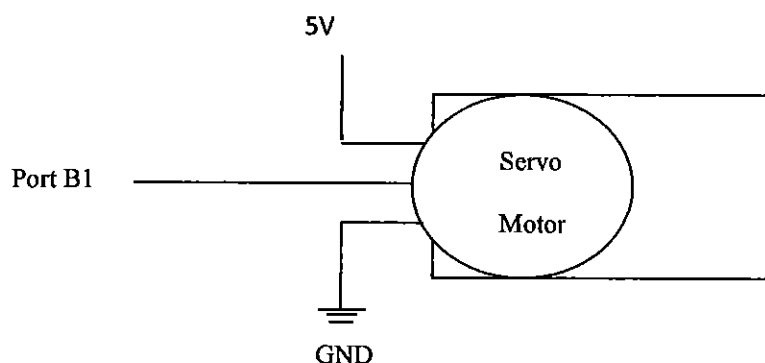
โมดูลบลูทูธเป็นโมดูลที่ใช้รับส่งสัญญาณบลูทูธ โดยการส่งข้อมูลแบบ UART เพื่อเชื่อมต่อกับสมาร์ทโฟนที่ติดตั้งแอปพลิเคชันในการส่งสัญญาณบลูทูธส่งค่ามุมมองศาเพื่อใช้ในการคำนวณในไมโครคอนโทรลเลอร์ ซึ่ง Rx จะต่อเข้ากับพอร์ต A9 ส่วนของ RTB จะต่อเข้ากับ CTB ดังรูป 3.7



รูปที่ 3.6 การเชื่อมต่อของโมดูลบลูทูธ

3.2.4 Servo Motor

การควบคุมทิศทางการเคลื่อนที่ถือเป็นเรื่องสำคัญอันดับต้น ๆ ของหุ่นยนต์เดินตามมนุษย์อัตโนมัติการควบคุมทิศทางนั้นทำได้โดยต่อเซอร์โวมอเตอร์เข้ากับแกนบังคับเลี้ยวของหุ่นยนต์แล้วให้ไมโครคอนโทรลเลอร์เป็นตัวส่งสัญญาณควบคุมมาที่เซอร์โวมอเตอร์ กำหนดให้หมุนในทิศทางใดและมุมกี่องศา

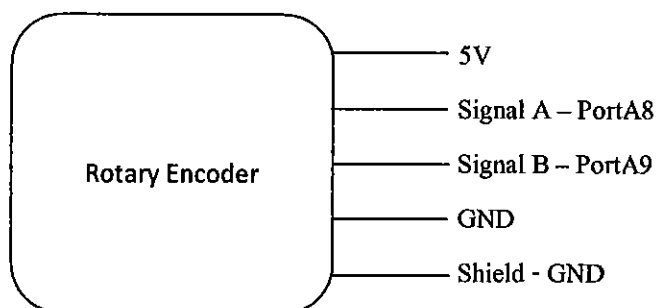


รูปที่ 3.7 การเชื่อมต่อเซอร์โวมอเตอร์

เซอร์โวมอเตอร์เป็นมอเตอร์ที่ประกอบไปด้วยชุดเกียร์ มอเตอร์กระแสตรง และส่วนควบคุมอิเล็กทรอนิกส์ที่รวมอยู่ในตัวมอเตอร์แสดงดังรูปที่ 3.5 การเชื่อมต่อเซอร์โวมอเตอร์จะต่อสัญญาณควบคุมที่มาจากไมโครคอนโทรลเลอร์ (พอร์ต B1) ซึ่งเป็นสัญญาณพัลส์ที่มีความกว้างอยู่ระหว่าง 2 ถึง 13 p percent duty cycle มีค่า period เท่ากับ 20 millisecond (ms)

3.2.5.1 Rotary Encoder

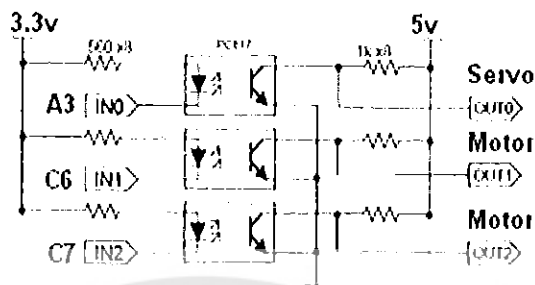
วงจรร Encoder เมื่อมีการหมุนแกน Rotary Encoder จะเกิดสัญญาณ 2 สัญญาณที่มีเฟสต่างกัน 90° ขึ้นอยู่กับทิศทางการหมุน การต่อวงจรร Encoder เข้ากับไมโครคอนโทรลเลอร์โดยที่ Encoder ต้องการไฟเลี้ยงวงจรตั้งแต่ 5-12V เนื่องจากสัญญาณพัลส์ที่ออกมาจาก Rotary Encoder มีขนาดของสัญญาณไม่คงที่จำเป็นต้องทำการแปลงสัญญาณให้เป็นพัลส์ที่ไมโครคอนโทรลเลอร์สามารถอ่านค่าได้ โดยใช้วงจรร opto-isolator ในการแปลงสัญญาณพัลส์



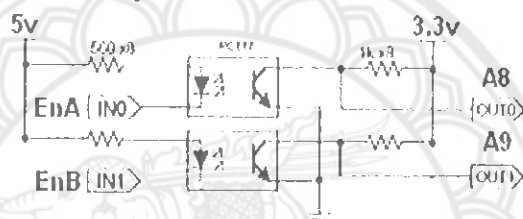
รูปที่ 3.8 การเชื่อมต่อของ Rotary Encode

3.2.6 Opto Isolator

ในการเชื่อมต่อมอเตอร์กับบอร์ดไฟโอะและจาก Rotary Encoder กับบอร์ดไฟโอะจะมีวงจร Isolator ช่วยแปลงสัญญาณจาก 3.3v เป็น 5v และจาก 5v เป็น 3.3v



(ก) รูปวงจรแปลงไฟจาก 3.3v เป็น 5v

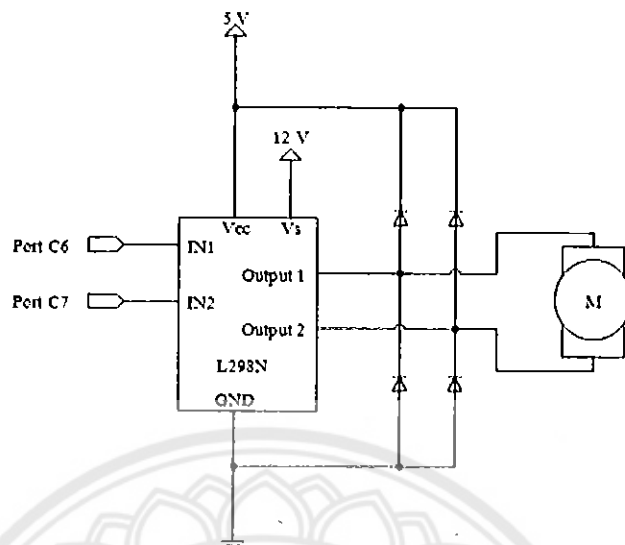


(ข) รูปวงจรแปลงไฟจาก 5v เป็น 3.3v

รูปที่ 3.9 รูปวงจร ET-MINI 3T05 TTL [14]

3.2.7 Motor

สัญญาณที่ได้จากไมโครคอนโทรลเลอร์เป็นสัญญาณขนาดเล็กไม่สามารถควบคุมมอเตอร์ได้เต็มประสิทธิภาพ จึงมีการต่อสัญญาณจากไมโครคอนโทรลเลอร์เข้ากับวงจรขับมอเตอร์เพื่อแปลงสัญญาณให้มีแรงดันเพิ่มขึ้น



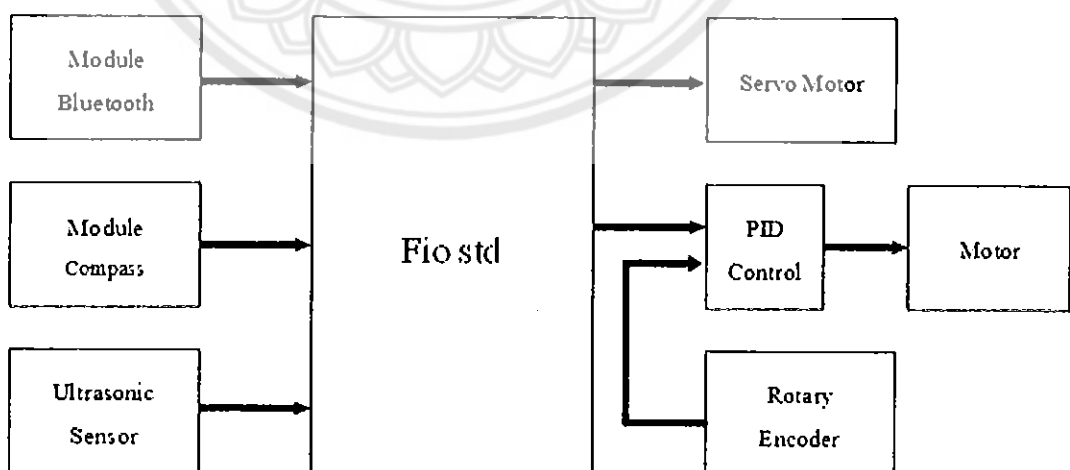
รูปที่ 3.10 การเชื่อมต่อวงจรขับมอเตอร์

3.3 โปรแกรมการทำงานหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ

หุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติมีส่วนที่ใช้ควบคุมการทำงานอุปกรณ์ต่าง ๆ โครงการนี้ใช้ Simulink ในโปรแกรม Matlab ในการเขียนโปรแกรมโดยการทำงานของโปรแกรมแบ่งออกเป็น 2 ส่วนคือ

ส่วนที่ใช้ในการควบคุมความเร็วของมอเตอร์

ส่วนที่ใช้ในการควบคุมทิศทางของหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ

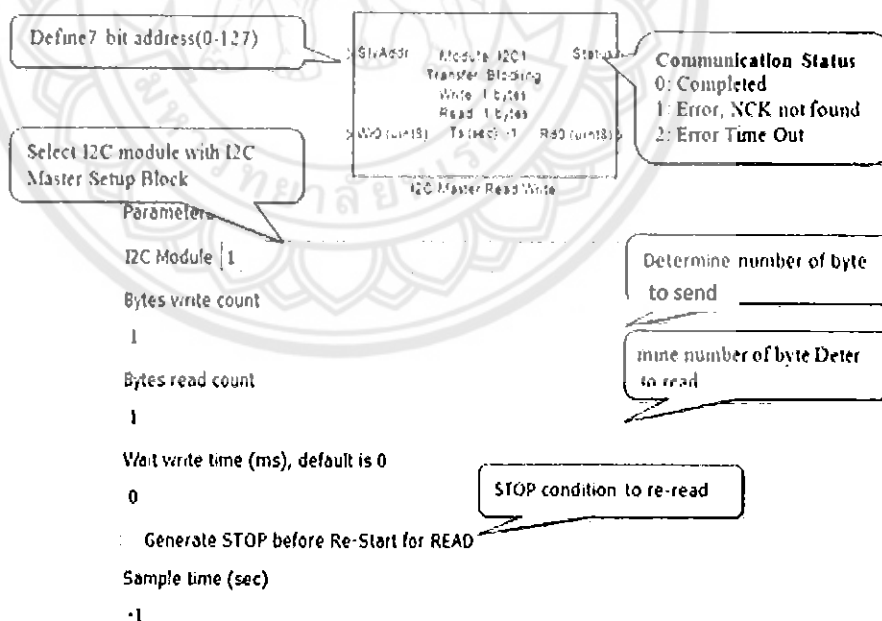


รูปที่ 3.11 แผนภาพการการทำงานหุ่นยนต์

เมื่อเปิดสวิตช์หุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ โปรแกรมจะทำงานตรงส่วนของการเชื่อมต่อโมดูลบลูทูธกับสมาร์ตโฟนที่ติดตั้งแอปพลิเคชันในการส่งค่ามุมด้วยสัญญาณบลูทูธ และหาค่ามุมจากโมดูลเข็มทิศและคำนวณหาค่ามุมที่เปลี่ยนไป เพื่อแปลงค่ามุมเป็น PWM ให้เซอร์โวมอเตอร์หมุนไปตามทิศการเคลื่อนที่ของผู้ถูกติดตาม และในด้านการควบคุมความเร็วจะรับค่าระยะห่างระหว่างหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติกับผู้ถูกติดตาม และนำมาเปรียบเทียบว่ามากกว่าหรือน้อยกว่า 60 รอบต่อ 10 วินาที และแปลงค่าที่ได้จากการเปรียบเทียบเป็นความเร็วที่ต้องการให้หุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติ (Set point) เพื่อควบคุมความเร็วแบบ PID โดยหาข้อผิดพลาดจากความเร็วที่ต้องการให้หุ่นยนต์เคลื่อนที่ (Set point) กับ ความเร็วของหุ่นยนต์ (rpm)

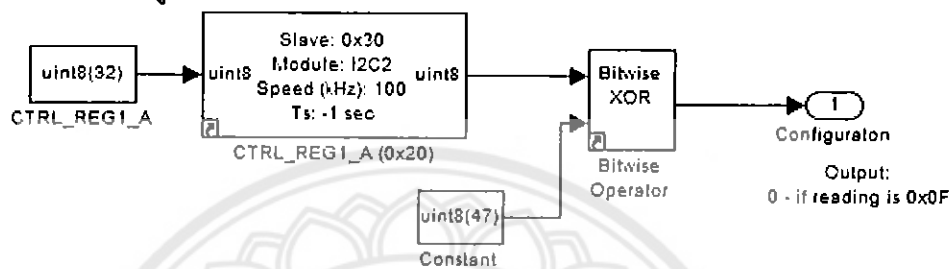
3.3.1 โปรแกรมสำหรับ Module Compass

การรับค่าจากโมดูลเข็มทิศ (Module Compass) จำเป็นต้องรับการเชื่อมต่อแบบ I²C ซึ่งในการเขียนโปรแกรมจำเป็นต้องใช้กล่องคำสั่ง I²C Master ในไลบรารีชุดกล่องคำสั่งrapidเอสทีเอ็ม 32 (RapidSTM32 Blockset)



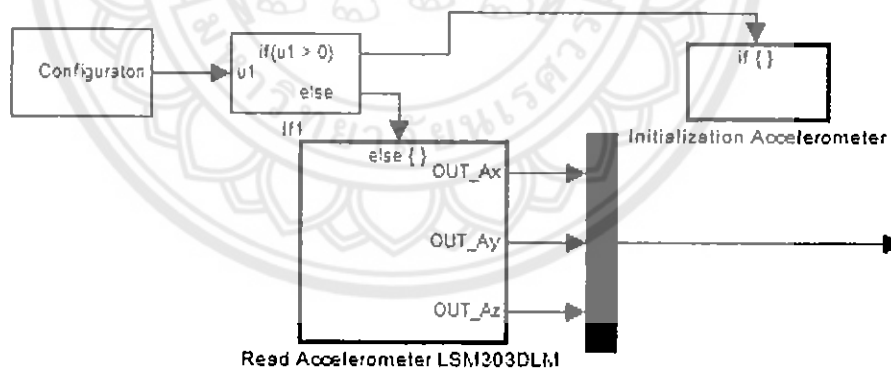
รูปที่ 3.12 กล่องคำสั่ง I2C Master

การส่งสัญญาณแบบ I²C จำเป็นต้องเซตค่า Slave address เพื่อระบุว่าต้องการรับค่าไหนจากตัว Compass sensor ดังนั้น เมื่อต้องการอ่านค่าของ accelerometer ต้องกำหนดค่า Slave address เท่ากับ 30f และกำหนดค่า control register เท่ากับ 32 เพื่อระบุว่าต้องการที่จะอ่านค่าของ accelerometer และต้องรับค่าจาก sensor ว่าเท่ากับ 47 หรือไม่ โดยการนำค่าที่อ่านได้มา XOR กับ 47 ถ้าเหมือนกันทั้งหมดค่าจะออกมาเป็น 0 ดังรูปที่ 3.13



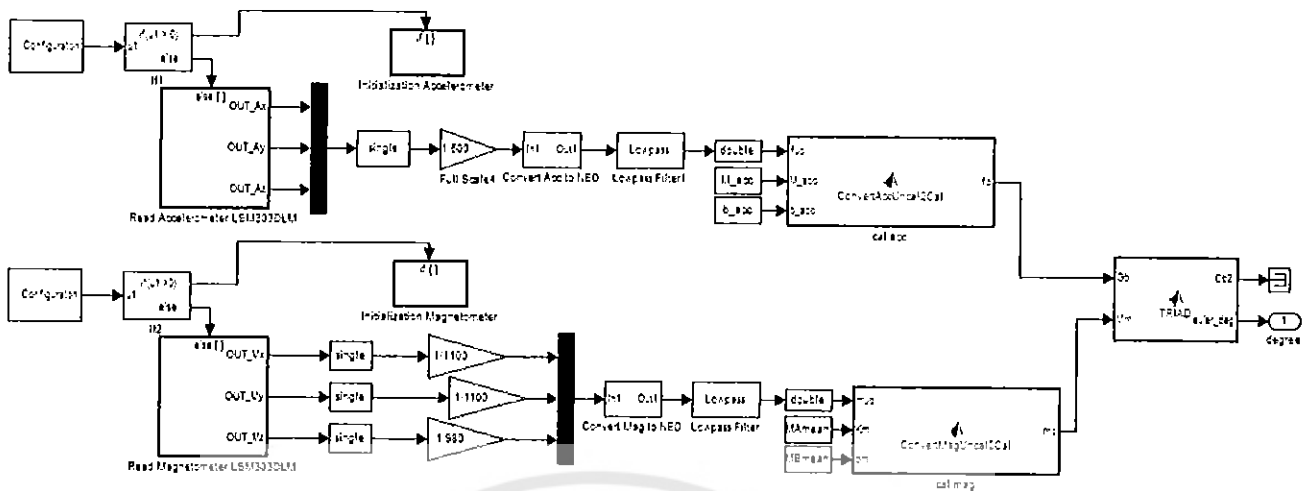
รูปที่ 3.13 การส่งสัญญาณแบบ I2C

เมื่อต้องการอ่านค่าจากตัว sensor จำเป็นต้องเขียน โปรแกรมใน simulink ดังรูปที่ 3.14 เพื่ออ่านค่าของแต่ละแกน (แกน x, y, z)



รูปที่ 3.14 การอ่านข้อมูลจาก โมดูลเข็มทิศ

ตัวโปรแกรมในการรับส่งสัญญาณแบบ I²C โดยนำค่าที่ได้ไปคำนวณหามุมหน่วยเป็นองศา (degree) ดังรูปที่ 3.15



รูปที่ 3.15 โปรแกรมการอ่านค่ามุมหน่วยเป็นองศา (degree)

จากรูปที่ 3.15 เป็นโปรแกรมที่ใช้รับค่าจากเซนเซอร์ 2 ชนิด คือ accelerometer และ Magnetometer เพื่อหาค่ามุมการหมุน ของโมดูลเข็มทิศในหน่วยองศา (degree)

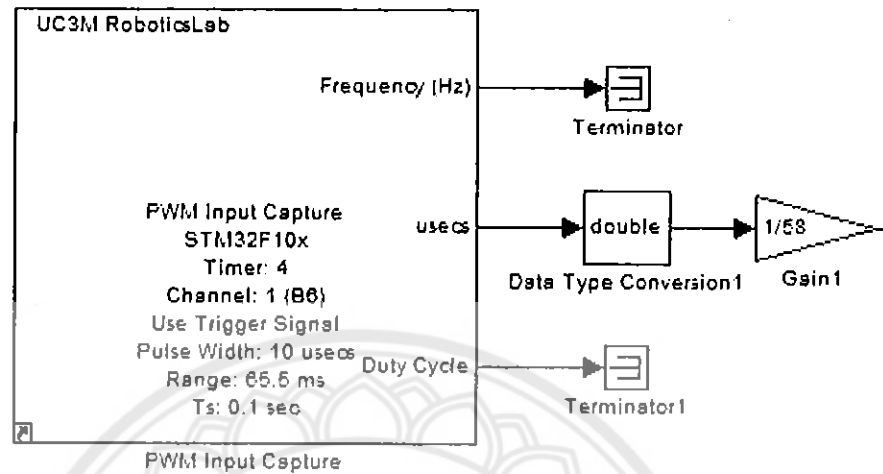
3.3.2 โปรแกรมสำหรับ Ultrasonic Sensor

เพื่อที่จะอ่านค่า ค่าระยะห่างระหว่าง Ultrasonic Sensor กับวัตถุจำเป็นต้องใช้กล่องคำสั่งในการส่งสัญญาณ Trigger ให้กับ Ultrasonic Sensor และรับสัญญาณพัลส์เป็นระยะเวลา 50 ms จำนวนพัลส์ที่นับได้ในช่วงระยะเวลานั้นจะนำมาคำนวณ โดยสมการที่ 3.1 ได้ระยะห่างเป็นหน่วยเซนติเมตร (cm.) ตามรูปที่ 3.18

$$distance(cm) = \frac{pulse\ width(\mu s)}{58} \tag{3.1}$$

โดยที่ pulse width (μs) คือ จำนวนพัลส์ที่นับได้

58 คือ ตัวหารที่เปลี่ยนหน่วยเป็นเซนติเมตร



รูปที่ 3.16 โปรแกรมการอ่านค่าระยะห่าง

จากรูปที่ 3.16 เป็นการตั้งค่าชุดกล่องคำสั่งรปเปิดเอสทีเอ็ม 32 ให้รับสัญญาณทางพอร์ต B6 และทำการนับจำนวนพัลส์ในช่วงเวลา 50 ms นำจำนวนพัลส์ที่นับได้มาหารกับ 58 เพื่อเปลี่ยนหน่วยให้เป็นหน่วยเซนติเมตร (cm.)

3.3.3 โปรแกรมสำหรับ Module Bluetooth

โมดูลบลูทูธเป็น โมดูลที่ใช้การส่งข้อมูลแบบ UART เพื่อเชื่อมต่อกับสมาร์ตโฟนที่ติดตั้งแอปพลิเคชันในการส่งสัญญาณบลูทูธส่งค่ามุมมองสา เพื่อใช้ในการคำนวณในไมโครคอนโทรลเลอร์ โดยจะแบ่งเป็นสองส่วนคือส่วนของแอปพลิเคชันบนสมาร์ตโฟนและส่วนของโมดูลบลูทูธแอปพลิเคชันบนสมาร์ตโฟน

สมาร์ตโฟนใช้ระบบปฏิบัติการแอนดรอยด์ 2.3.6 โดยพัฒนาแอปพลิเคชันบนโปรแกรม Eclipse หากค่าของมุมจากเซนเซอร์ในสมาร์ตโฟนโดยใช้คำสั่ง ตามรูปที่ 3.17

```

textViewDegree2 = (TextView) findViewById(R.id.textViewDegree2);
mySensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
List<Sensor> mySensors = mySensorManager
    .getSensorList(Sensor.TYPE_ORIENTATION);

if (mySensors.size() > 0) {
    mySensorManager.registerListener(mySensorEventListener,
        mySensorManager.get(0), SensorManager.SENSOR_DELAY_NORMAL);
    sensorrunning = true;
    textViewDegree2.makeText(this, "Start ORIENTATION Sensor", textViewDegree2.LENGTH_LONG)
        .show();
} else {
    textViewDegree2.makeText(this, "No ORIENTATION Sensor", textViewDegree2.LENGTH_LONG)
        .show();
    sensorrunning = false;
    finish();
}

```

รูปที่ 3.17 โปรแกรมการรับค่ามุมจากเซ็นเซอร์บนสมาร์ทโฟน

การส่งข้อมูลแบบ UART เป็นการส่งข้อมูลแบบอะซิงโครนัสโดยชุดการส่งข้อมูลจะมี header กับ closing ตามรูปที่ 3.18

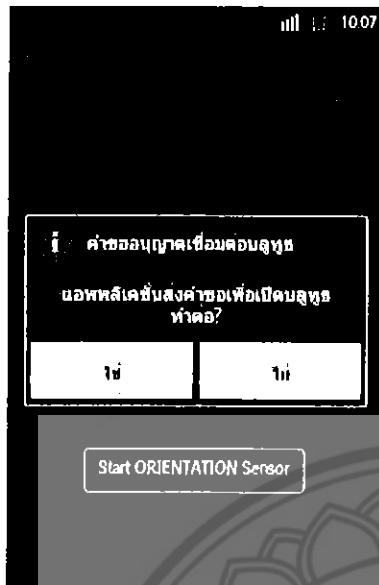
```

if (SEND) {
    byte[] sendingbytes = new byte[sendingDouble.length * 8 + 4]; // byte array for packet
    ByteBuffer buff2 = ByteBuffer.wrap(sendingbytes).order(
        ByteBuffer.LITTLE_ENDIAN); // wrapping byte array for modification
    buff2.put((byte) 0x7e); // Header 7E
    buff2.put((byte) 0x7e);
    for (double d : sendingDouble)
        buff2.putDouble(d); // three double values going in byte array
    buff2.put((byte) 0x03); // Closing 03
    buff2.put((byte) 0x03);

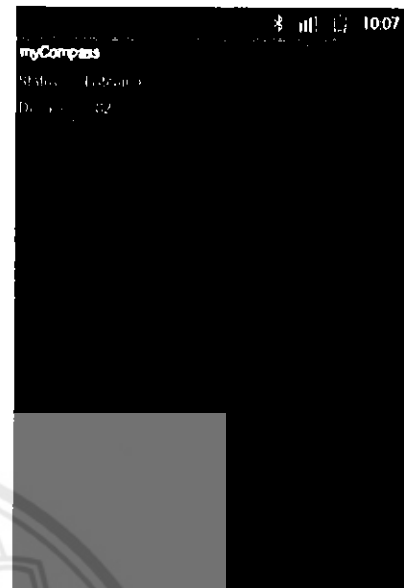
    if (mState == STATE_CONNECTED) {
        mConnectedThread.write(sendingbytes);
    }
} else {
    if (ReceiveUpdate) {
        ReceiveUpdate = false;
    }
}
}

```

รูปที่ 3.18 กำหนดการส่งข้อมูล



(ก) การขออนุญาตเปิดบลูทซ์



(ข) หน้าจอโปรแกรม



(ค) เมนู



(ง) ค้นหาอุปกรณ์



(จ) กำลังเชื่อมต่อ



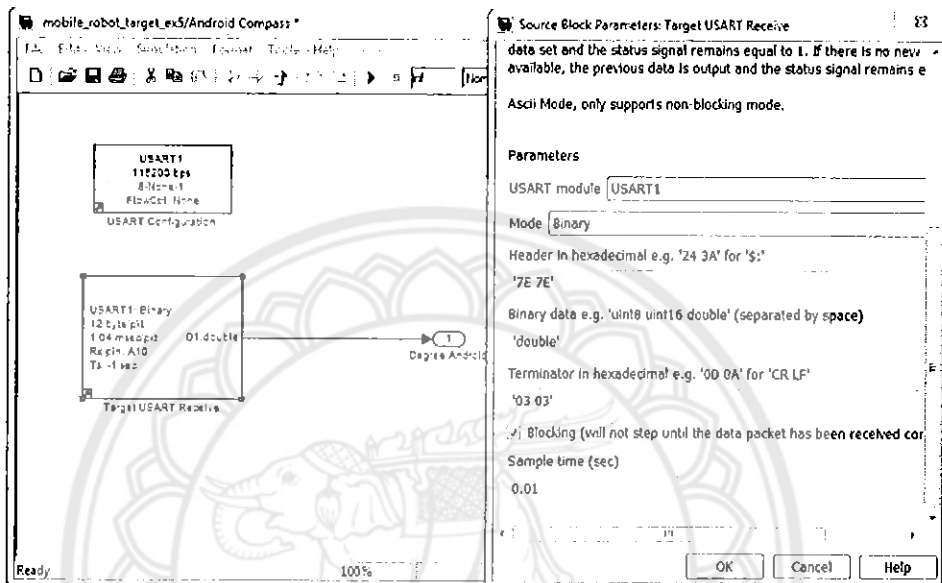
(ข) เชื่อมต่อกับอุปกรณ์

รูปที่ 3.19 แสดงโปรแกรม myCompass



โมดูลบลูทูธ

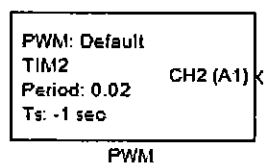
ในการเขียนโปรแกรมในส่วนของโมดูลบลูทูธจะต้องตั้งค่าการใช้งานในส่วนของ UART และ ตั้งการรับข้อมูลให้ตรงกับส่วนของแอปพลิเคชันสมาร์ตโฟน โดยการตั้งค่าจะเป็นดังรูปที่ 3.21



รูปที่ 3.20 โปรแกรมการรับสัญญาณบลูทูธ

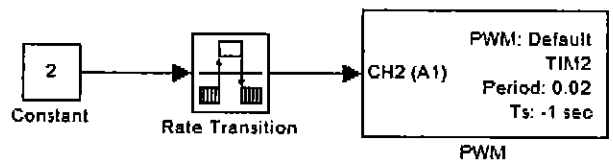
3.3.4 โปรแกรมสำหรับ Motor และ Servo Motor

การควบคุมเซอร์โวมอเตอร์ (Servo Motor) และมอเตอร์ (Motor) ต้องอาศัยการป้อนสัญญาณพัลส์วิดมอดูเลชันจากบอร์ดไฟโอ (FiO Board) โดยใช้บล็อก PWM ในไลบรารีชุดกล่องคำสั่งรปิดเอสทีเอ็ม 32 (RapidSTM 32 Blockset) ดังรูปที่ 3.21



รูปที่ 3.21 แสดงภาพบล็อก PWM ที่ใช้สร้างสัญญาณพัลส์วิดมอดูเลชัน

ข้อมูลที่บล็อก PWM รับเข้ามานั้นต้องผ่านบล็อก Rate Transition เพื่อปรับอัตราการถ่ายโอนข้อมูลระหว่างบล็อกให้มีอัตราการถ่ายโอนเท่ากันดังรูปที่ 3.22

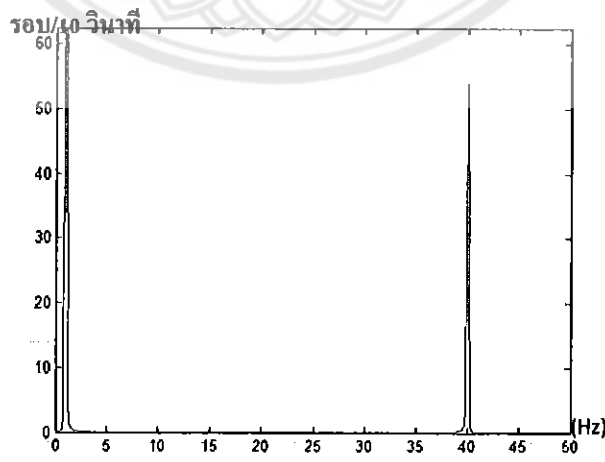


รูปที่ 3.22 แสดงภาพการส่งค่าให้กับบล็อก PWM

3.3.5 โปรแกรมสำหรับ Encoder

การวัดความเร็วของหุ่นยนต์ (Encoder) จะใช้กล่องคำสั่ง Incremental Quadrature Encoder Interface จะนับสัญญาณที่ส่งเข้าไปทั้งขอบขาขึ้นและขอบขาลง ซึ่งเมื่อ Rotary Encoder หมุนครบหนึ่งรอบจะเกิดสัญญาณพัลส์ 400 พัลส์ คูณกับ 4 จากขอบขาขึ้นและขอบขาลงของสายสัญญาณ A และ B กล่อง Incremental Quadrature Encoder Interface จะนับได้ทั้งหมด $400 \times 4 = 1600$ เพื่อที่จะหาความเร็วจึงนำข้อมูลที่ได้ มาหาอนุพันธ์อันดับหนึ่งโดยใช้กล่องคำสั่ง Discrete Derivative เมื่อหาอนุพันธ์อันดับหนึ่งจะทำให้ข้อมูลที่ได้มีสัญญาณรบกวนปะปนอยู่ เพื่อที่จะแก้สัญญาณรบกวน จำเป็นต้องใช้ฟิลเตอร์ช่วยกรองสัญญาณรบกวนออก

การหาสมการฟิลเตอร์ จะใช้ ฟังก์ชัน Fast Fourier transform (FFT) ที่มีอยู่ในโปรแกรม MATLAB โดยจะเก็บข้อมูลความเร็วที่ต้องการทำ FFT โดยจะเก็บข้อมูลแค่ช่วงระยะเวลาหนึ่งที่แสดงในกราฟเส้นสีเหลืองในรูปที่ 3.25 ลงใน Workspace เพื่อจะเอาข้อมูลที่เก็บไว้มาคำนวณโดยใช้คำสั่ง FFT จะสามารถหาค่า ความถี่ที่ตัดออฟได้ โดยที่จากกราฟข้อมูล Harmonic หลัก จะเป็นข้อมูลความเร็วของ motor จริง และข้อมูลช่วงหลังจะเป็นสัญญาณรบกวน ทำให้สามารถกำหนดให้ความถี่ตัดออฟเท่ากับ 5 Hz ตามรูปที่ 3.23



รูปที่ 3.23 กราฟการทำ FFT หาความถี่ตัดออฟ

หาสมการฟิลเตอร์ได้จาก

$$\text{lowpassfilter} = \frac{1}{1 + \frac{s}{2\pi f_{\text{cutoff}}}} \quad (3.2)$$

จะได้สมการฟิลเตอร์ดังนี้

$$\text{lowpassfilter} = \frac{1}{0.3979s + 1} \quad (3.3)$$

แต่วงจรฟิลเตอร์ที่ได้จะเป็นแบบ Continuous จำเป็นต้องแปลงให้เป็นแบบ Discrete ก่อนจึงจะ
ใช้กับระบบได้ แปลงโดยใช้คำสั่งดังนี้

```
>> sys = tf([1],[0.39788 1])
```

```
Transfer function:
```

```
1
-----
0.3979 s + 1
```

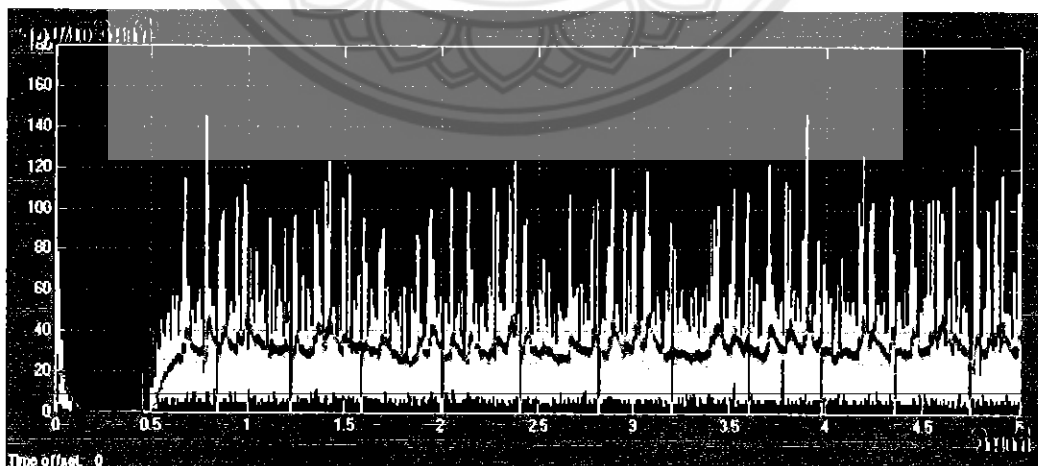
```
>> c2d(sys,0.005)
```

```
Transfer function:
```

```
0.01249
-----
z - 0.9875
```

รูปที่ 3.24 คำสั่ง c2d

เมื่อใส่วงจรฟิลเตอร์เข้าไป ความเร็วที่ได้จะเป็นไปตามรูปที่ 3.25 กราฟเส้นสีเหลืองแสดง
ความเร็วก่อน และกราฟเส้นสีฟ้าแสดงความเร็วหลังใส่วงจรฟิลเตอร์



รูปที่ 3.25 กราฟแสดงความเร็วก่อนและหลังใส่วงจรฟิลเตอร์

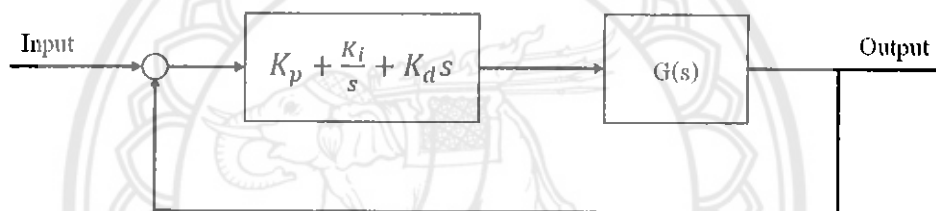
3.3.6 การควบคุมแบบ PID

การควบคุมแบบ PID นั้นประกอบด้วยเทอมสำหรับการปรับแต่ง 3 เทอม รวมกันเป็น Manipulated Variable (MV)

$$MV = K_p + \frac{K_i}{s} + K_d s \quad (3.4)$$

โดยที่การจะหาค่าของ K_p , K_i , K_d ได้นั้นจำเป็นต้องหา Transfer

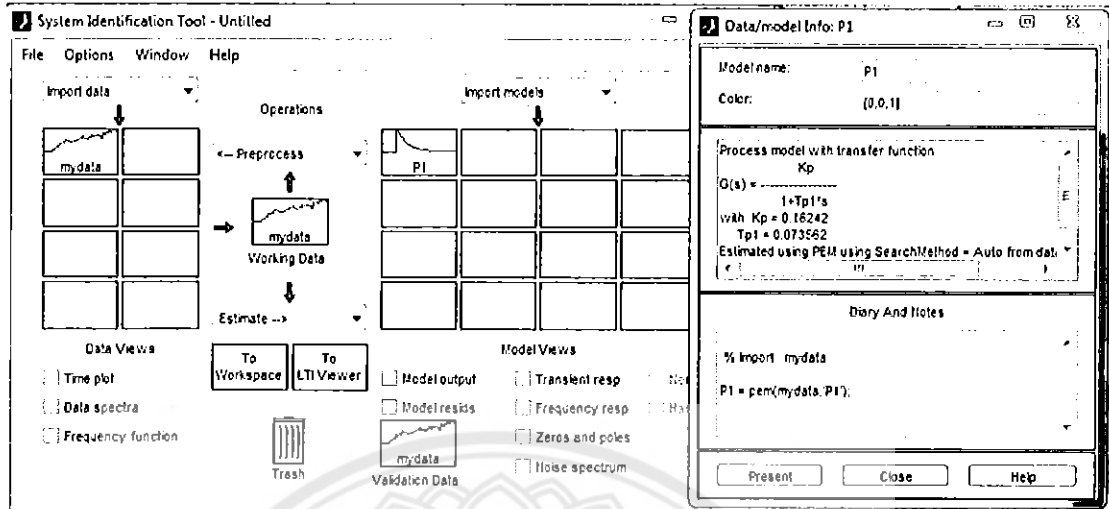
Function ของระบบมาเสียก่อนเพื่อทำการหาค่า K_p , K_i , K_d



รูปที่ 3.26 ระบบการควบคุมแบบ PID

การจะหา Transfer Function ของระบบได้นั้นจำเป็นต้องเก็บข้อมูลเพื่อมาวิเคราะห์ โดยการเก็บข้อมูลนั้น จะต้องมีการกระตุ้นระบบโดยการจ่าย Unit Step ที่มี amplitude เท่ากับ 40 รอบต่อ 10 วินาที ให้กับระบบ และเก็บข้อมูลไว้ใน Workspace นำข้อมูลที่ได้มาตัดเฉพาะส่วนที่ต้องการวิเคราะห์ ตามรูปที่ 3.25 กราฟเส้นสีฟ้า

เมื่อได้ข้อมูลแล้วจะทำการหา Transfer Function ของระบบโดยใช้ฟังก์ชัน `ident` ในโปรแกรม MATLAB เพื่อหา Transfer Function เมื่อทำการแอดข้อมูลเข้าไปในฟังก์ชัน `ident` เลือกเมนู `Process Models` และกด `Estimate` จะมีหน้าต่างขึ้นมาดังรูปที่ 3.27

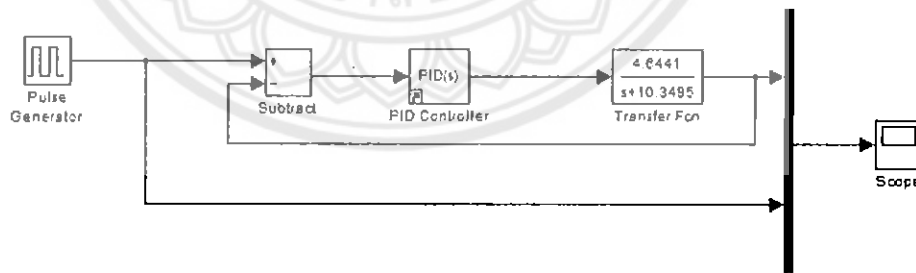


รูปที่ 3.27 ฟังก์ชัน ident

นำค่า K_p , T_{p1} มาคำนวณหา Transfer function ของระบบดังสมการที่ 3.5

$$G(s) = \frac{K_p}{s + \frac{1}{T_{p1}}} \tag{3.5}$$

การเขียน โปรแกรมจำลองระบบจะมีรูปแบบ โปรแกรมดังนี้



รูปที่ 3.28 Simulate PID Controller

เมื่อได้ Transfer Function ของระบบมาแล้วทำการหาค่าของ K_p , K_i , K_d โดยใช้ pidtune ตามรูปที่ 3.30

```
>> pidtune(sys, 'pidf')
```

Continuous-time PIDF controller in parallel form:

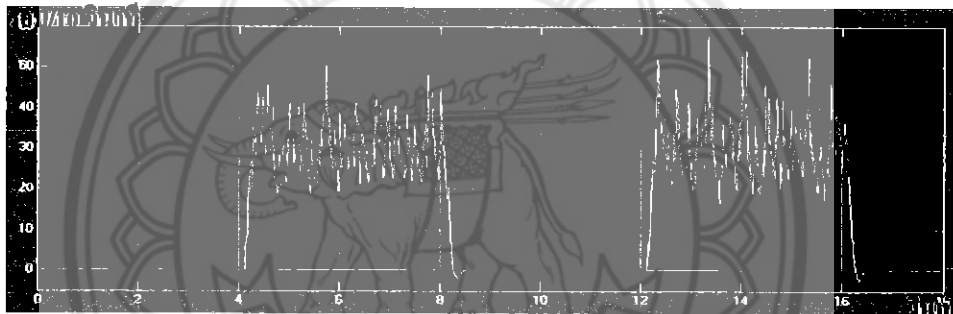
$$K_p + K_i * \frac{1}{s} + K_d * \frac{s}{T_f * s + 1}$$

With $K_p = 2.94$, $K_i = 51.4$, $K_d = 0.0589$, $T_f = 0.02$

รูปที่ 3.29 ค่า K_p , K_i , K_d และ n จากฟังก์ชัน PID Tuner

จะได้ค่า K_p , K_i , K_d , $N = \frac{1}{T_f}$ เมื่อใส่ค่าลงไป PID Control ในระบบจะได้ Output ออกมา

ตามรูปที่ 3.32 โดยที่กราฟสีฟ้าจะแสดงการควบคุมความเร็ว และ กราฟสีม่วงจะแสดงค่าของ Set Point



รูปที่ 3.30 กราฟความเร็วที่ควบคุมแบบ PID

3.3.7 โปรแกรมสำหรับหุ่นยนต์เดินตามมนุษย์อัตโนมัติ

ในการออกแบบโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์จะแบ่งออกเป็น 2 ส่วน คือ ส่วนของการติดตาม (การคำนวณหาความเร็ว) และส่วนของทิศทาง (การคำนวณหาทิศทาง)

การคำนวณหาความเร็ว

ในส่วนนี้จะควบคุมการติดตามโดยการหาความเร็วที่ต้องการให้หุ่นยนต์เคลื่อนที่ (Set point) และควบคุมความเร็วของมอเตอร์ด้วยระบบควบคุมแบบ PID

จะหาความเร็วที่ต้องการให้หุ่นยนต์เคลื่อนที่ (Set point) ได้นั้นจะหาจากความสัมพันธ์ของระยะห่างกับความเร็วของรถจากสมการด้านล่าง

$$\text{Set point (รอบ/10วินาที)} = \frac{(\text{ระยะห่าง}-60)}{\text{เส้นรอบล้อของหุ่นยนต์}} \quad (3.6)$$

โดย ระยะห่าง คือ ระยะที่วัดได้จาก Ultrasonic sensor

เส้นรอบล้อของหุ่นยนต์ คือ ความยาวของเส้นรอบล้อ

60 คือ ระยะห่างที่ต้องการให้หุ่นยนต์ติดตาม (เซนติเมตร)

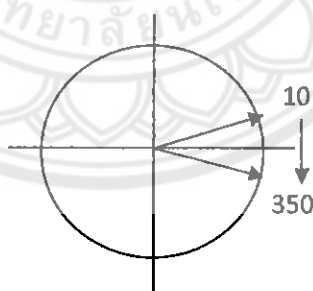
โดยมีเงื่อนไขการติดตามคือเมื่อหุ่นยนต์เคลื่อนที่เป็นแบบเส้นตรงจะใช้สมการด้านบนในการหา Set point แต่ถ้ามีการตัดสินใจให้เลี้ยวเกิดขึ้น หุ่นยนต์จะเคลื่อนที่ด้วยด้วยความเร็วเท่าเดิมตลอดไม่มีการเปลี่ยนแปลง

การคำนวณหาทิศทาง

ในการติดตามของหุ่นยนต์จะใช้การเปรียบเทียบค่าของมุมทั้ง 2 คือ มุมที่ได้จาก โมดูลเข็มทิศ และมุมที่ได้จากแอปพลิเคชันบนสมาร์ตโฟน โดยการเปรียบเทียบจะเป็นไปตามสมการที่ 3.9

เมื่อค่ามุมเริ่มต้นอยู่ที่ 10° และมีการเปลี่ยนแปลงมุมทางขวา 20° ดังรูปที่ 3.31 จำเป็นต้องลบออกด้วย 360° เพื่อให้ค่ามุมเปลี่ยนแปลงที่ได้ตรงกับความจริง

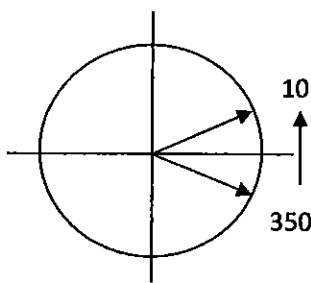
$$\theta = (\theta_{compass} - \theta_{smartphone}) - 360^\circ \quad (3.7)$$



รูปที่ 3.31 การเปลี่ยนแปลงของมุม

เมื่อค่ามุมเริ่มต้นอยู่ที่ 350° และมีการเปลี่ยนแปลงมุมทางซ้าย 20° ดังรูปที่ 3.32 จำเป็นต้องบวกเพิ่มด้วย 360° เพื่อให้ค่ามุมเปลี่ยนแปลงที่ได้ตรงกับความจริง

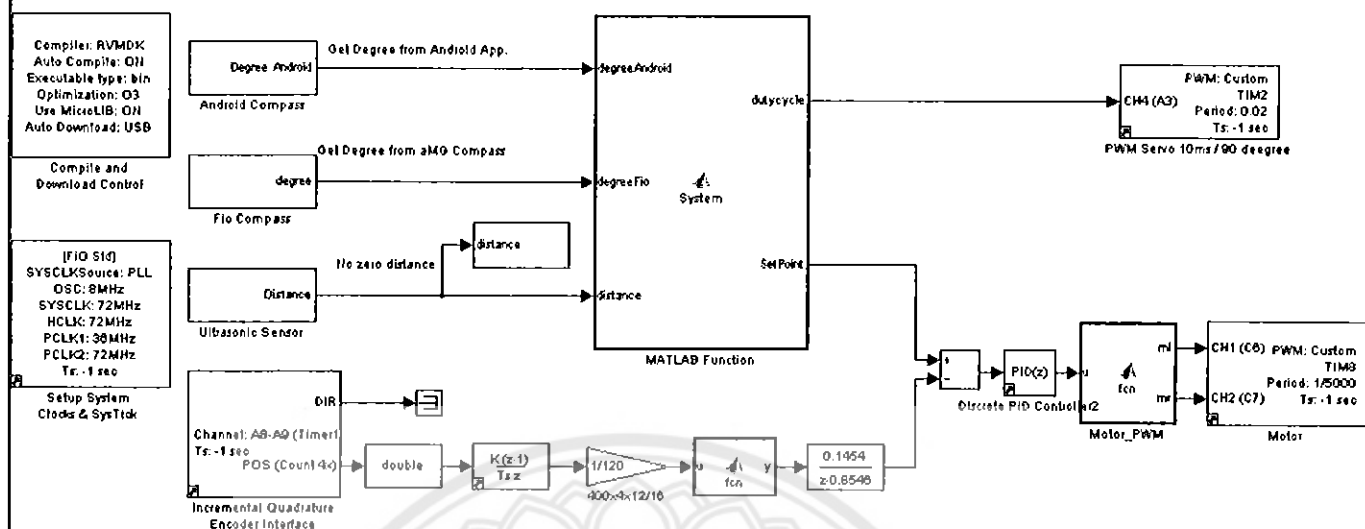
$$\theta = (\theta_{compass} - \theta_{smartphone}) + 360^\circ \quad (3.8)$$



รูปที่ 3.32 การเปลี่ยนแปลงของมุม

$$\text{มุมที่เซอร์โวมอเตอร์ (\%Duty cycle)} = 6.7 + (\theta_{compass} - \theta_{smartphone}) \times \frac{2.5}{45} \quad (3.9)$$

โดย 6.7 คือ % duty cycle ที่ทำให้ servo motor หมุนล้อให้หุ่นยนต์วิ่งไปข้างหน้า 2.5 คือ % duty cycle ที่ทำให้ servo motor หมุนล้อให้หุ่นยนต์เลี้ยวซ้ายหรือเลี้ยวขวามีเงื่อนไข โดยที่มุมของ Servo motor หมุนให้หุ่นยนต์หันล้อตรงนั้นจะต้องไม่เกิน 45° ถ้าเกิน 45° จะบังคับให้ Servo motor หมุนไปแค่ 45° องศาเท่านั้น และกรณีที่ค่ามุมไม่ถูกต้องดังต่อไปนี้



รูปที่ 3.33 โปรแกรมของหุ่นยนต์เคลื่อนตามมนุษย์อัตโนมัติ

จากรูปที่ 3.33 เป็นโปรแกรมที่ใช้ในการควบคุมหุ่นยนต์ โดยการรับค่าจาก Sensor เข็มทิศ โมดูลบลูทูธ เซนเซอร์ชนิดอัลตราโซนิก และ Encoder และนำค่าที่ได้จาก Sensor ต่างมาประมวลผล เพื่อที่จะขับมอเตอร์และเซอร์โวมอเตอร์ให้ตามมนุษย์ได้

ในบทนี้นำเสนอขั้นตอนการดำเนินงานโดยจะแบ่งเป็นสามส่วนหลัก ๆ คือ โครงสร้างของ หุ่นยนต์ การเชื่อมต่อวงจร และส่วนของโปรแกรม ในส่วนของผลการทดลองนั้นจะนำเสนอในบท ถัดไป

บทที่ 4

การทดลองผลการทดลอง

4.1 การทดลองการเคลื่อนไหวของหุ่นยนต์

4.1.1 การเคลื่อนไหวของล้อหน้า

เป็นการทดลองวัดความกว้างของมุมขณะที่หุ่นยนต์ทำการเคลื่อนไหว ซึ่งวัดความกว้างของมุมระหว่างมุมล้อ ณ จุดเริ่มต้นที่ใช้ PWM (%duty) = 6.7 กับมุม ณ จุดที่ล้อที่เคลื่อนไปข้างละ 20° และค่าสัญญาณพัลส์วิดิมของเซ็นเซอร์ที่ป้อนให้กับเซอร์โวมอเตอร์ ดังตารางที่ 4.1

ตารางที่ 4.1 ตารางบันทึกการเคลื่อนไหวของล้อ

ครั้งที่	ซ้าย		ขวา	
	มุม (องศา)	PWM(%duty)	มุม (องศา)	PWM(%duty)
1	20	5.2	20	8.1
2	20	5.3	20	8.2
3	20	5.2	20	8.1
4	20	5.3	20	8.2
5	20	5.2	20	8.2
6	20	5.2	20	8.2
7	20	5.3	20	8.2
8	20	5.2	20	8.2
9	20	5.3	20	8.2
10	20	5.2	20	8.1
12	20	5.2	20	8.2
13	20	5.2	20	8.2
14	20	5.2	20	8.2
15	20	5.3	20	8.2
16	20	5.2	20	8.1
17	20	5.2	20	8.2

ครั้งที่	ซ้าย		ขวา	
	มุม (องศา)	PWM(%duty)	มุม (องศา)	PWM(%duty)
18	20	5.2	20	8.1
19	20	5.2	20	8.2
20	20	5.2	20	8.1
เฉลี่ย	20	5.22	20	8.16



(ก) เลี้ยวขวา 20°



(ข) เลี้ยวซ้าย 20°



(ค) ขณะเริ่มต้น

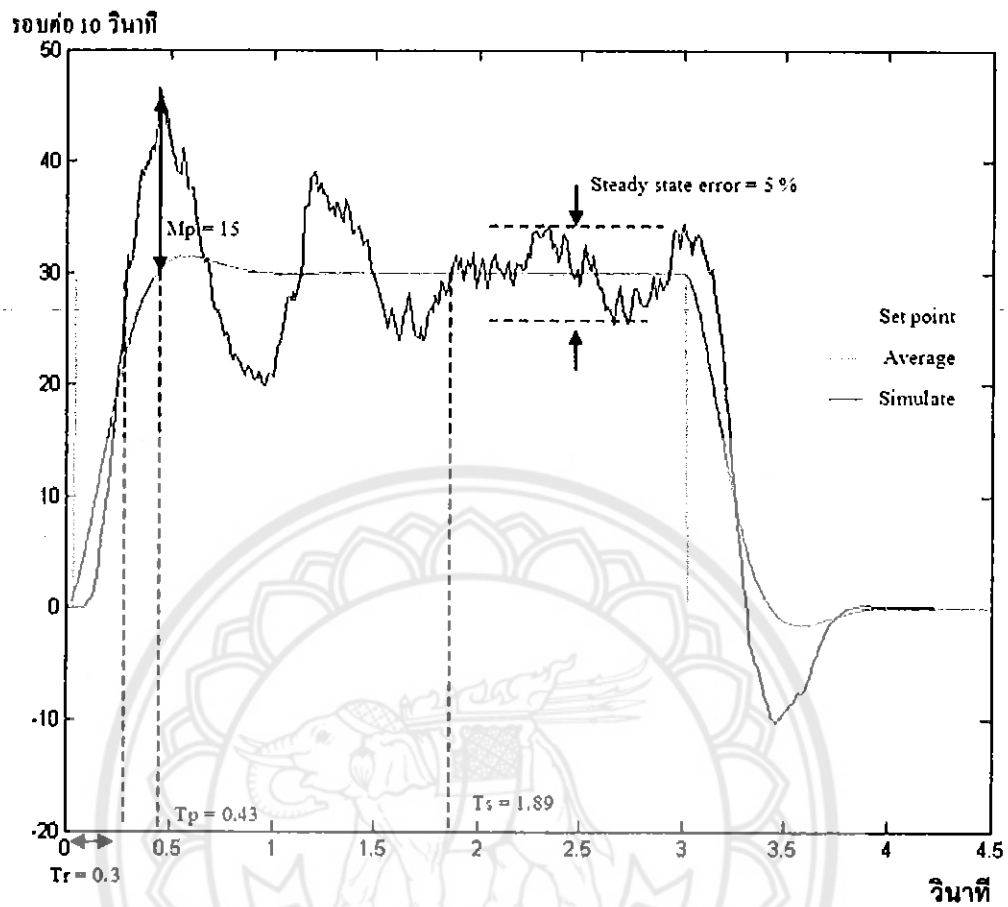
รูปที่ 4.1 แสดงภาพขณะเลี้ยวซ้าย - ขวา และขณะเริ่มต้น

4.1.2 การควบคุมความเร็วของมอเตอร์

การทดลองในส่วนของการควบคุมความเร็วของมอเตอร์เป็นการทดลองเพื่อบันทึกค่าของ Peak time, Steady state error, Rise time, Setting time และ Mp (Maximum Overshoot) ที่ได้จากระบบควบคุมแบบ PID ดังตารางที่ 4.2

ตาราง 4.2 ตารางบันทึกผลการควบคุมความเร็วมอเตอร์

ครั้งที่	Peak time(s) (Tp)	Rise time(s) (Tr)	Setting time(s) (Ts)	Steady state error(%)
1	0.41	0.31	1.85	26
2	0.38	0.28	1.96	28
3	0.45	0.30	1.92	24
4	0.42	0.33	1.95	23
5	0.46	0.29	1.72	30
6	0.46	0.27	1.87	29
7	0.40	0.35	1.90	28
8	0.41	0.30	1.93	25
9	0.48	0.28	1.84	29
10	0.43	0.29	1.96	22
เฉลี่ย	0.43	0.3	1.89	26.4



รูปที่ 4.2 กราฟแสดง Step response เฉลี่ยของการทดลอง 10 ครั้ง และ
กราฟ Step response ที่ได้จากการ Simulate บนโปรแกรม MATLAB

1. เวลาของค่ายอด (Peak Time, T_p) เวลาที่สัญญาณผลการตอบสนองมีค่าสูงสุดค่าแรกของผลการตอบสนองนั้น ๆ
2. ช่วงเวลาขึ้น (Rise Time, T_r) ช่วงเวลาที่ผลตอบสนองผู้ทดลองเลือกใช้ช่วงเวลาเมื่อสัญญาณเอาต์พุตเพิ่มจาก จาก 0% ถึง 100%
3. โอเวอร์ชูดสูงสุด (Maximum overshoot, M_p) ค่าการตอบสนองสูงสุดที่วัดจากสถานะอยู่ตัวสุดท้าย (Final Steady State) การบอกค่าโอเวอร์ชูดสูงสุดมักจะบอกเป็นเปอร์เซ็นต์

4. เวลาเข้าที่ (Settling Time, T_s) เวลาที่ผลการตอบสนองลดลงจนเริ่มเข้าไปอยู่ในช่วงที่กำหนด ซึ่งผู้ทดลองกำหนดอยู่ในช่วง 10 % ซึ่งจะวัดเทียบกับค่าสุดท้ายของผลการตอบสนองในสถานะอยู่ตัวสุดท้าย (Final Steady State)

5. ความผิดพลาดสถานะคงตัว (Steady state error) ความแตกต่างระหว่างอินพุตและเอาต์พุตเมื่อป้อนสัญญาณ รูปแบบต่าง ๆ ที่ใช้ในทดสอบเข้าไปที่อินพุตของระบบ แล้วตรวจสอบความแตกต่างที่เวลาเข้าสู่นันต์หรือ เวลาที่ระบบอยู่ในภาวะคงตัว

4.1.3 การวัดระยะของ Ultrasonic Sensor

เป็นการทดลองในส่วนของ Ultrasonic Sensor ว่าสามารถวัดระยะได้ถูกต้องเพียงใด กำหนดการทดลองวัดระยะ 10 ครั้ง โดยกำหนดการระยะที่ใช้วัดดังนี้ 60 cm, 100 cm, 150 cm, 200 cm และ 250 cm ผลการทดลองดังตารางที่ 4.3

ตาราง 4.3 ตารางบันทึกผลการทดลองการวัดระยะของ Ultrasonic Sensor

ครั้งที่	60 cm	100 cm	150 cm	200 cm	250 cm
1	60	101	153	204	252
2	61	97	151	199	255
3	61	100	150	202	248
4	60	102	150	200	250
5	61	99	148	201	245
6	60	102	149	198	251
7	59	100	151	199	255
8	61	100	147	203	250
9	59	102	150	200	253
10	60	101	153	202	253

4.2 การทดสอบการเคลื่อนที่ติดตามมนุษย์ของหุ่นยนต์

ทดสอบการติดตามของหุ่นยนต์

ทดสอบจากสถานะการติดตามของหุ่นยนต์ว่าติดตามได้หรือไม่ จำนวน 10 ครั้ง โดยมีเส้นทางเป็น

1. ทางตรง ระยะทาง 10 เมตร
2. เลี้ยวซ้าย เป็นมุม 30° ระยะทาง 2 เมตร
3. เลี้ยวขวา เป็นมุม 30° ระยะทาง 2 เมตร
4. เลี้ยวซ้าย เป็นมุม 20° ระยะทาง 3 เมตร
5. เลี้ยวขวา เป็นมุม 20° ระยะทาง 3 เมตร

ดังตารางที่ 4.4

ตารางที่ 4.4 ตารางการบันทึกสถานะการติดตามของหุ่นยนต์

ครั้งที่	ตรง	ซ้าย 30 องศา	ขวา 30 องศา	ซ้าย 20 องศา	ขวา 20 องศา
1	ติดตามได้	ติดตามได้	ติดตามไม่ได้	ติดตามได้	ติดตามได้
2	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามไม่ได้
3	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้
4	ติดตามได้	ติดตามไม่ได้	ติดตามได้	ติดตามไม่ได้	ติดตามได้
5	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้
6	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้
7	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้
8	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้
9	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามได้	ติดตามไม่ได้
10	ติดตามได้	ติดตามไม่ได้	ติดตามได้	ติดตามได้	ติดตามได้



(ก) ขณะเลี้ยว

(ข) เดินตรง

รูปที่ 4.3 แสดงภาพการติดตามของหุ่นยนต์



บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

การสร้างหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติทำการทดลองโดยใช้ Simulink ซึ่งเป็นโปรแกรมที่มาพร้อมกับเมทแลบ (MATLAB) สำหรับการจำลองขั้นตอนการขับเคลื่อนหุ่นยนต์ และวิเคราะห์การทำงานของระบบร่วมกับบอร์ดไฟโอ (FiOSid) ผ่านชุดกล่องคำสั่งราปิเดสที่เอ็ม 32 (RapidSTM32 Blockset)

ในการโปรแกรมเพื่อควบคุมให้เซนเซอร์ชนิดอัลตราโซนิค (ultrasonic sensor) 1 ตัวทำหน้าที่วัดระยะห่างระหว่างหุ่นยนต์กับผู้ถูกติดตามให้มอเตอร์ 1 ตัวทำหน้าที่ในการควบคุมความเร็วของการเคลื่อนที่แบบ PID และเซอร์โวมอเตอร์ 1 ตัว เพื่อบังคับทิศทางการเลี้ยวของหุ่นยนต์ รวมทั้งสร้าง ระบบควบคุมหุ่นยนต์ผ่านสัญญาณบลูทูธ (Bluetooth) ด้วยโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์ (Android) เพื่อเปรียบเทียบค่ามุมในการเลี้ยวของหุ่นยนต์

การพัฒนาหุ่นยนต์เคลื่อนที่ตามมนุษย์อัตโนมัติมีน้ำหนักรวม 2.8 กิโลกรัม ความกว้าง 29 เซนติเมตร ความยาว 38 เซนติเมตร ความสูง 20 เซนติเมตร ใช้มอเตอร์ทำหน้าที่ในการควบคุมความเร็วของการเคลื่อนที่ และใช้เซอร์โวมอเตอร์ที่มีขนาดแรงบิด 1.3 กิโลกรัมเซนติเมตร เป็นตัวบังคับเลี้ยวโดยมุมเลี้ยวแคบสุดคือ 5° และสามารถเคลื่อนที่ตามมนุษย์อัตโนมัติได้ มีความเร็วสูงสุด 8 รอบต่อวินาที โดยมีขนาดเส้นผ่าศูนย์กลางของล้อเท่ากับ 13 เซนติเมตร

จากการทดลองในตารางที่ 4.1 แสดงการเลี้ยวซ้ายและเลี้ยวขวาจำนวน 20 ครั้ง สามารถสรุปได้ว่า เมื่อเลี้ยวซ้ายมีค่ามุมเป็น 20° สัญญาณพัลส์วัดมุมดูเลขชี้ที่ใช้จะมีค่าอยู่ที่ 5.2 – 5.3 โดยส่วนใหญ่จะมีค่าสัญญาณเท่ากับ 5.2 ส่วนการเลี้ยวขวาค่ามุมเป็น 20° สัญญาณพัลส์วัดมุมดูเลขชี้ที่ใช้จะมีค่าอยู่ที่ 8.1 – 8.2 โดยส่วนใหญ่จะมีค่าสัญญาณเท่ากับ 8.2

จากการทดลองในตารางที่ 4.2 การควบคุมความเร็วของมอเตอร์โดยค่าเฉลี่ยของ $T_r = 0.27$ s $T_p = 0.42$ s, $T_s = 1.76$ s, $M_p = 15$ รอบต่อ 10 วินาที และ steady state error = 5% เมื่อนำข้อมูลมาวิเคราะห์เป็นกราฟรูปที่ 4.2 ได้จากค่าเฉลี่ยการทดลอง 10 ครั้ง มาวิเคราะห์กับกราฟที่ได้จากการ

Simulate บนโปรแกรม MATLAB จะพบว่าค่าที่ได้จากค่าเฉลี่ยมีค่าสูงกว่าค่าที่ได้จากการ Simulate สามารถสรุปได้ว่า ค่าที่ได้จากการทดลองจริงจะมีประสิทธิภาพดีต่อการ Simulate

จากการทดลองดังตารางที่ 4.3 การวัดระยะของ Ultrasonic Sensor 10 ครั้งในระยะ 60 cm., 100 cm., 150 cm., 200 cm. และ 250 cm. สามารถสรุปความคลาดเคลื่อนของการวัดระยะได้ดังนี้ ที่ 60 cm. อยู่ในช่วง 59 – 61 cm. = 1%, 100 cm. อยู่ในช่วง 97 – 102 cm. = 2%, 150 cm. อยู่ในช่วง 147 -153 cm. = 2%, 200 cm. อยู่ในช่วง 197 – 204 cm. = 2%, 250 cm. อยู่ในช่วง 245 – 255 cm. = 2%,

จากการทดลองการเคลื่อนที่ติดตามมนุษย์ของหุ่นยนต์จำนวน 10 ครั้ง สามารถสรุปได้ว่า

1. ทางตรง ระยะทาง 10 เมตร ถูกต้อง 100 %
2. เลี้ยวซ้าย เป็นมุม 30° ระยะทาง 2 เมตร ถูกต้อง 80 %
3. เลี้ยวขวา เป็นมุม 30° ระยะทาง 2 เมตร ถูกต้อง 90 %
4. เลี้ยวซ้าย เป็นมุม 20° ระยะทาง 3 เมตร ถูกต้อง 90 %
5. เลี้ยวขวา เป็นมุม 20° ระยะทาง 3 เมตร ถูกต้อง 80 %

ข้อผิดพลาดที่เกิดขึ้นเกิดจากการเลี้ยวของหุ่นยนต์ซึ่งอาจเกิดจากการถือโทรศัพท์ไม่ได้ระดับ กับพื้นจึงทำให้ค่ามุมที่ส่งไปให้กับระบบคลาดเคลื่อนได้

5.2 ปัญหาที่เกิดจากการทดลอง

แบตเตอรี่มีความจุไฟฟ้าที่น้อยจึงให้ระยะเวลาในการใช้งานของหุ่นยนต์สั้นลงด้วยการออกแบบส่วนตัวควบคุมมีประสิทธิภาพที่ไม่ดีมากจึงทำให้ได้ผลตอบสนองไม่เป็นไปตามที่ต้องการ โมดูลบลูทูธมีข้อจำกัดในการใช้งาน คือจะให้ค่าที่แม่นยำในสถานที่ที่โล่งกว้าง ถ้าอยู่ในอาคารจะทำให้ค่าของมุมมีความคลาดเคลื่อนและจะส่งผลต่อการเลี้ยวของหุ่นยนต์

5.3 ข้อเสนอแนะ

การทำให้หุ่นยนต์มีความสามารถในการติดตามที่ดีควรมีโครงสร้างที่แข็งแรงมากขึ้น เลือกใช้มอเตอร์ที่มีกำลังสูงมากขึ้น เลือกใช้เซนเซอร์ชนิดอัลตราโซนิกที่ให้ผลตอบสนองที่เร็วขึ้น หรือเลือกใช้เซนเซอร์อื่นที่สามารถทำงานได้คล้ายกันแต่มีประสิทธิภาพที่ดีกว่า ในส่วนของการทำการทดลอง เซนเซอร์ชนิดอัลตราโซนิกควรทำในพื้นที่ที่กว้าง ไม่มีสิ่งกีดขวาง เนื่องการปล่อยสัญญาณเสียงจะปล่อยสัญญาณไปด้านหน้าและกระจายออก 30° การทดลองโมดูลบลูทูธควรทำในพื้นที่โล่งภายนอกอาคาร เพราะจะทำให้ได้ค่าที่แม่นยำมากขึ้น ในการเลือกใช้แบตเตอรี่ควรเลือกใช้แบตเตอรี่ที่มีความจุของไฟฟ้าที่มาก เพื่อจะทำให้ระยะเวลาการใช้งานของหุ่นยนต์ได้ระยะเวลานานขึ้น

เอกสารอ้างอิง

- [1] กระจุก Android Guide. (30 เมษายน 2557).รวมแอปเข็มทิศ (Compass) สำหรับ Android และ IOS. สืบค้นเมื่อ 15 พฤศจิกายน 2557, จาก <http://android.kapook.com/view87469.html>
- [2] ดวงพร เฟิร์ชแบน. (17 กุมภาพันธ์ 2013). ระบบปฏิบัติการแอนดรอยด์. สืบค้นเมื่อ 12 พฤศจิกายน 2557, จาก http://potinimi.blogspot.com/2013/02/1_16.html
- [3] น.ต.กัปตัน เตียวตระกูล. (2554).คอนโทรลเลอร์แบบ PID.สืบค้นเมื่อ 13 ตุลาคม 2557, จาก <http://www.navy.mi.th/elecwww/magaz/magazine/no16/18.pdf>
- [4] นวกัทธา และ ทวีพล. (2555). Ultrasonic sensor / เซนเซอร์ชนิดใช้เสียง หรือเซนเซอร์ชนิดอัลตราโซนิก. สืบค้นเมื่อ 15 ธันวาคม 2557, จาก <http://www.foodnetworksolution.com/wiki/word/4348/ultrasonic-sensor-เซนเซอร์ชนิดใช้เสียง-หรือเซนเซอร์ชนิดอัลตราโซนิก>
- [5] พีรช.(5 กุมภาพันธ์ 2551).Bluetooth สืบค้นเมื่อ 11 พฤศจิกายน 2557, จาก <http://w123456.exteen.com/>
- [6] Aimagin Co.Ltd. (26 เมษายน 2014). การขับ Step Motor และ RC Servo Motor. สืบค้นเมื่อ 25 พฤศจิกายน 2557, จาก <http://aimagin.com/blog/driving-step-motor-and-rc-servo-motor/?lang=th>
- [7] Aimagin Co.Ltd. (2555). fio-std.สืบค้นเมื่อ 26 พฤศจิกายน 2557 , จาก <https://www.aimagin.com/fio-std.html>
- [8] Aimagin Co.Ltd. (2555). amg-bluetooth.สืบค้นเมื่อ 26 พฤศจิกายน 2557 , จาก https://www.aimagin.com/downloads/dl/file/id/48/amg_bluetooth_a_datasheet.pdf
- [9] Aimagin Co.Ltd. (2555). amg_imu_9a.สืบค้นเมื่อ 26 พฤศจิกายน 2557 , จาก https://www.aimagin.com/downloads/dl/file/id/81/amg_imu_9a_datasheet.pdf
- [10] Atom (23 กันยายน 2550).หลักการสร้างสัญญาณ PWM สืบค้นเมื่อ 23 พฤศจิกายน 2557, จาก <http://introduction-pwm.blogspot.com/2007/09/pwm.html>

- [11] ARM Ltd and ARM Germany GmbH. (2557). **MDK-ARM**. สืบค้นเมื่อ 15 ตุลาคม 2557, จาก <https://www.keil.com/download/product/>
- [12] digchip.(2548). **LSM303DLM**. สืบค้นเมื่อ 16 พฤศจิกายน 2557 , จาก <http://www.digchip.com/datasheets/parts/datasheet/456/LSM303DLM-pdf.php>
- [13] Electronic-Circuits-Diagrams.(2555). **Ultrasonic Switch**. สืบค้นเมื่อ 15 ธันวาคม 2557, จาก <http://www.electronic-circuits-diagrams.com/ultrasonic-switch-circuit/>
- [14] ETT CO.,LTD .(27 มิถุนายน 2549). **Schematic ET-MINI 3T05 TTL** . สืบค้นเมื่อ 12 ธันวาคม 2557, จาก <http://www.es.co.th/Schematic/PDF/ET-MINI3TO5TTL.PDF>
- [15] ETT CO.,LTD .(27 มิถุนายน 2549). **ET-MINI 3T05 TTL** . สืบค้นเมื่อ 12 ธันวาคม 2557, จาก <http://www.ett.co.th/product/InterfaceBoard/P-ET-A-00237.html>
- [16] Googleplay. (2557) . **SensorManager**. สืบค้นเมื่อ 15 ตุลาคม 2557, จาก <http://developer.android.com/reference/android/hardware/SensorManager.html#getOrientation>
- [17] instructables.(2549). **Dual H-Bridge - L298**. สืบค้นเมื่อ 21 พฤศจิกายน 2557 , จาก <http://www.instructables.com/id/Dual-H-Bridge-L298-Breakout-Board-Homemade/>
- [18] John M. Powell. (2548). **Switches & Indicators**. สืบค้นเมื่อ 23 พฤศจิกายน 2557 , จาก http://www.mikesflightdeck.com/pedestal/switches_indicators.html
- [19] Katsuhiko Ogata. (2545). **Modern Control Engineer**. New Jersey. Prentice Hall.
- [20] Thaimicrotron. (2554). **PC BUS**. สืบค้นเมื่อ 15 พฤศจิกายน 2557, จาก <http://www.thaimicrotron.com/CCS-628/Reference/I2CBUS.htm>
- [21] Thaiprojectshop. (2550). **Motor Driver 2A L298 H-Bridge**. สืบค้นเมื่อ 30 พฤศจิกายน 2557 , จาก http://www.thaiprojectshop.com/index.php/component/virtuemart/view/productdetails/virtuemart_product_id/28/virtuemart_category_id/9
- [22] The MathWorks. (2553). **Support**. สืบค้นเมื่อ 5 ตุลาคม 2557, จาก http://www.mathworks.com/support/?s_tid=gn_supp

[23] The MathWorks. (2553). **PID Tuning Algorithm**. สืบค้นเมื่อ 6 ตุลาคม 2557, จาก

<http://www.mathworks.com/help/control/getstart/pid-tuning-algorithm.html>

[24] thaieasyelec.(2555). **Uart**. สืบค้นเมื่อ 6 พฤศจิกายน 2557, จาก

<http://www.thaieasyelec.com/article->

[wiki/basic-electronics/uart-ttl-rs232-max232-max3232.html](http://www.thaieasyelec.com/article-wiki/basic-electronics/uart-ttl-rs232-max232-max3232.html)

[25]wikipedia. (2556). **DC Motor**. สืบค้นเมื่อ 12 พฤศจิกายน 2557, จาก

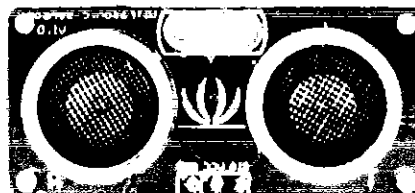
<http://th.wikipedia.org/wiki/%E0%B8%A1%E0%B8%AD%E0%B9%80%E0%B8%95%E0%B8%AD%E0%B8%A3%E0%B9%8C>





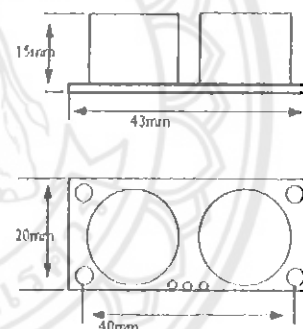
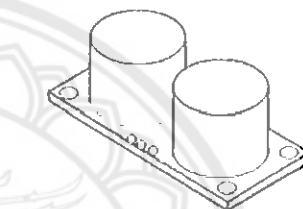
Seeed Ultrasonic Sensor

Seeed ultrasonic sensor is non-contact distance measurement module, which is also compatible with electronic brick. It's designed for easy modular project usage with industrial performance.



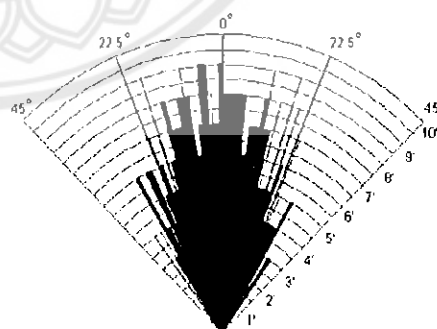
Features

- Detecting range: 3cm-4m
- Best in 30 degree angle
- Electronic brick compatible Interface
- 5VDC power supply
- Breadboard friendly
- Dual transducer
- Arduino library ready



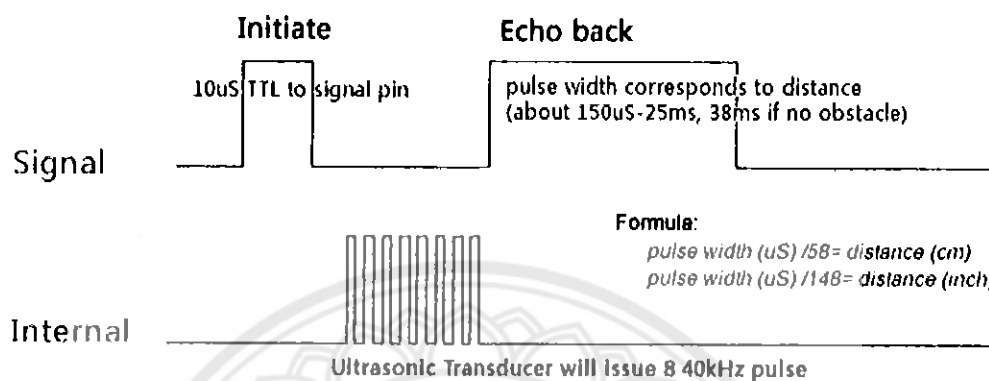
Specifications

Supply voltage	5 v
Global Current Consumption	15 mA
Ultrasonic Frequency	40k Hz
Maximal Range	400 cm
Minimal Range	3 cm
Resolution	1 cm
Trigger Pulse Width	10 μ s
Outline Dimension	43x20x15 mm



*Practical test of performance,
Best in 30 degree angle*

Sequence chart



A short ultrasonic pulse is transmitted at the time 0, reflected by an object. The sensor receives this signal and converts it to an electric signal. The next pulse can be transmitted when the echo is faded away. This time period is called cycle period. The recommended cycle period should be no less than 50ms.

If a 10µs width trigger pulse is sent to the signal pin, the Ultrasonic module will output eight 40kHz ultrasonic signal and detect the echo back. The measured distance is proportional to the echo pulse width and can be calculated by the formula above. If no obstacle is detected, the output pin will give a 38ms high level signal.

Revision History

Rev.	Descriptions	Release date
1.0	Seeed Ultrasonic Sensor	14.05.2010



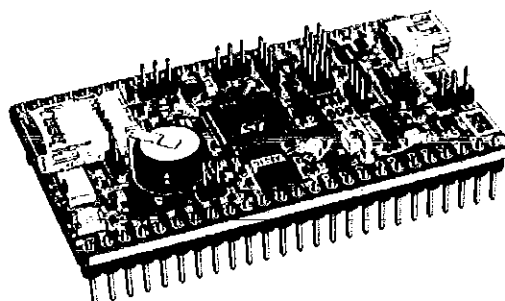
FIO Std Datasheet

FEATURES

- Built-in RapidSTM32 native-support bootloader.
- ARM 32-bits Cortex™ – M3 Processor (STM32F103RET6)
 - 90MIPS maximum speed
 - 64 KBytes SRAM
 - 51 GPIO
 - 16 channels 12-bit, 1µs ADC
 - 5 USART, 3 SPI, 2 I²C, 1 CAN
 - On-chip temperature sensor
 - 4 General purpose 16-bits timers with 4 IC/OC/PWM per timer
 - 2 16-bits Advanced control timers
 - 2 Watchdog timer
 - 2 channels 12-bit DAC
 - USB 2.0 full speed interface
- Two onboard crystals
 - 32 kHz (for real-time clock)
 - 8 MHz (for MCU clock)
- Power supply management
 - Reverse supply voltage protection
 - USB or External (up to 15VDC) supply input selection via jumper settings
 - 3.3V (up to 800mA) LDO regulator for internal and external circuitries
- Flash programming
 - 496Kbytes available flash memory
 - In-Application Programming via either USB HID or serial (RS232) interfaces directly from Matlab™ or MS Windows™.
 - Automatic compile and download directly from Matlab™.
- Other onboard peripherals
 - 3 user LED (red, yellow, green)
 - 2 user logic (H/L) input jumpers
 - 10K potentiometer
- High capacitance (0.33F) capacitor as RTC backup battery
- RS232 (3232-series) level converter
- Micro SD card socket
- Four built-in operating modes selectable via jumper settings
 - USB Mass Storage Device (micro SD card reader)
 - In-application programming via USB HID interface
 - In-application programming via serial (RS232/USART) interface
 - Stand-alone custom user application
- RoSH compliant

SAMPLE APPLICATIONS

- Affordable Rapid Prototyping training tools (especially for Automatic Control and Digital Signal Processing studies) when used together with RapidSTM32 blockset, example features include:
 - Support C code generation of a custom user program for STM32 from a Simulink™ model (see minimum requirements).
 - Support various Hardware-in-the-Loop (HIL) simulation configurations, e.g. data acquisition in to, signal generation from Matlab™/Simulink™, and closed-loop control HIL simulation design and analysis.
- Create a data logger to micro SD card (FAT32) program directly from a Simulink model.
- Use in a laboratory as embedded system training kits or in-the-field for any student projects.
- Generic microcontroller evaluation board.



FIO Std

INTRODUCTION

FiO Std is an evaluation board (EVB) from FiO family of evaluation boards that are based on STMicroelectronics STM32™ ARM 32-bits Cortex™ – M3 processors.

FiO EVBs have been primarily designed as affordable embedded system training boards for used by instructors and students to supplement such courses and curriculums as design and analysis of automatic or embedded control systems and digital signal processing (DSP) systems. It is hoped that the introduction of FiO and RapidSTM32 Blockset will help lower the barrier and open up opportunities for more users to play around with Rapid Prototyping technology.

FiO family of evaluation boards have been specifically designed to be applicable both as LAB kits for uses in laboratories as well as a stand-alone system for used in-the-field in any student's science and engineering projects.

When used together with RapidSTM32 Blockset, FiO Std offers several capabilities, for examples:

- Real-Time Hardware in the Loop Simulations via USB HID.

Real-Time Hardware in the Loop (HIL) Simulation via USB HID interface		Maximum Update Rate ¹ (Hz)
Open-Loop	Target → Host (Data Acquisition)	450
	Host → Target (Signal Generation)	200
Closed-Loop	Target ↔ Host (Control System)	200

- Higher HIL update rate available via other specially designed interfaces.
- Create working stand-alone embedded systems from Simulink™ models with just one click.

¹ The actual update rate depends on several factors such as the host PC speed and other USB hardware in the communication loop; for example, communication speed is faster when the target is connected directly to host USB controller than via a USB hub.

- Generated source code compatible with Realview MDK™ for ARM with options for automatic compile and download option.
- Data logger to micro SD card (FAT32).
- Support simulation and code generation for typical peripherals such as push/toggle button or character LCD.
- Supported on-chip peripherals include Digital IO, ADC, DAC, RTC, PWM, USART, USB (Virtual COM) and more.
- Support integration with Finite State Machine simulation and code generation via Stateflow™ and Stateflow Coder™.

MINIMUM REQUIREMENTS

Minimum requirements for FiO Std to fully function with RapidSTM32 Blockset.

- Matlab™ 2009a (v7.8)
- Simulink™ (v7.3)
- Real-Time Workshop™ (v7.3)
- Real-Time Workshop Embedded Coder™ (v7.3)
- Realview MDK™ for ARM (v4.0)
- Microsoft .NET Framework (v3.5)

CAUTIONS!!

All FiO EVBs are pre-installed with proprietary bootloader. FiO EVBs are solely intended to be programmed using FiO own Flash programmer which can program any FiO EVBs directly from Matlab and MS Windows via USB or RS232 interfaces. **DO NOT** use STMicroelectronics Flash Loader Program to program FiO EVBs because this will erase the pre-installed bootloader and void your warranty totally. Therefore, jumper J8 should not be tampered with in any circumstances.

FURTHER INFORMATION

For further information and tutorials please visit: <http://www.aimagin.com/learningresources>

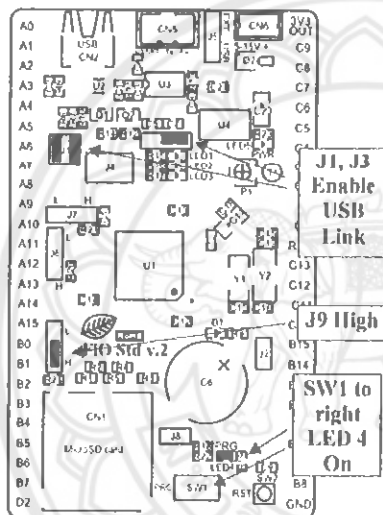
OPERATING MODES

FiO Std supports 4 operating modes

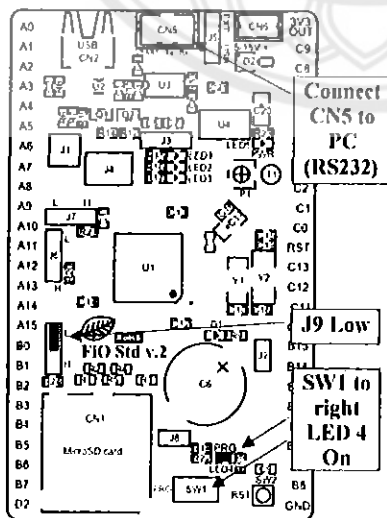
1. In-Application Programming via USB HID
2. In-Application Programming via RS232
3. Mass Storage Device (microSD card reader)
4. Run Custom User Program

The following figures show required jumper(s) and program switch (SW1) settings for each operating mode. Each operating mode can be activated by pressing the reset switch (SW2).

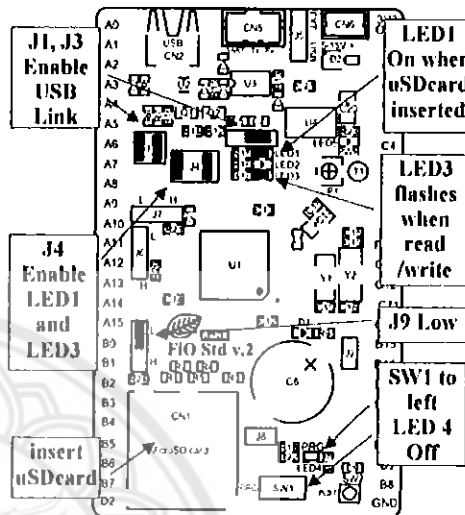
1.1 USB IAP



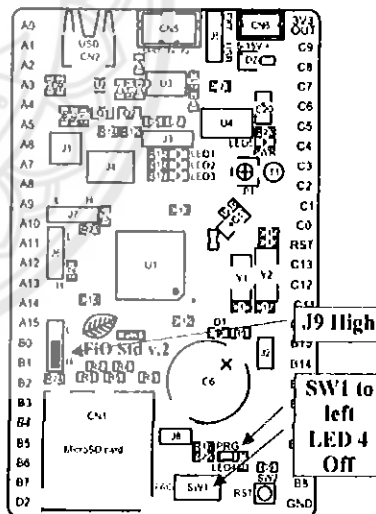
1.2 RS232 IAP



1.3 Mass Storage Device

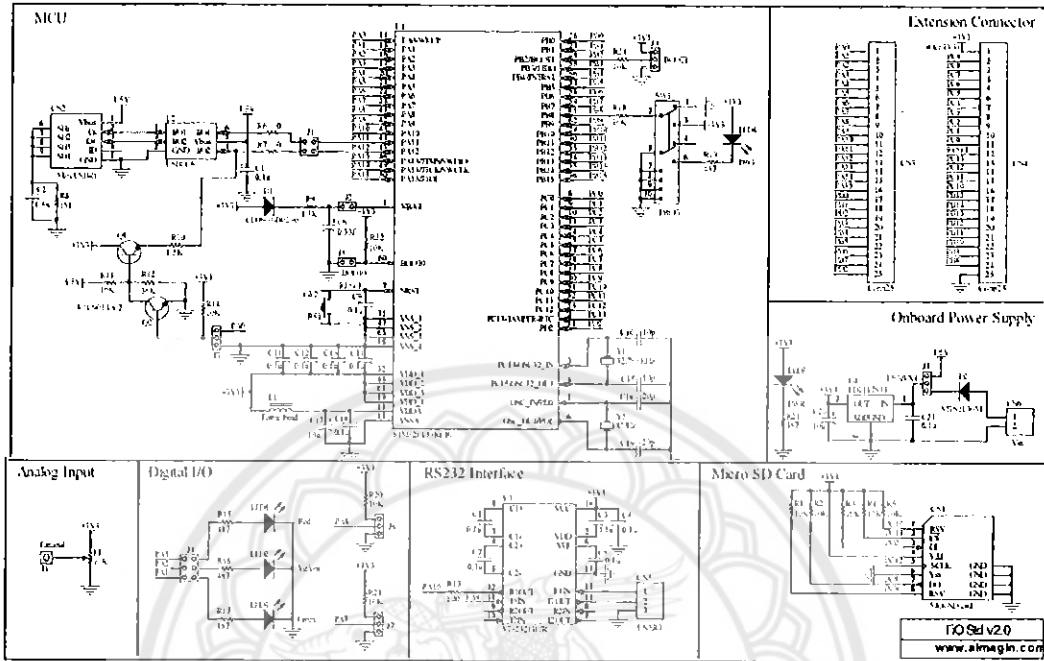


1.4 Run Custom User Program

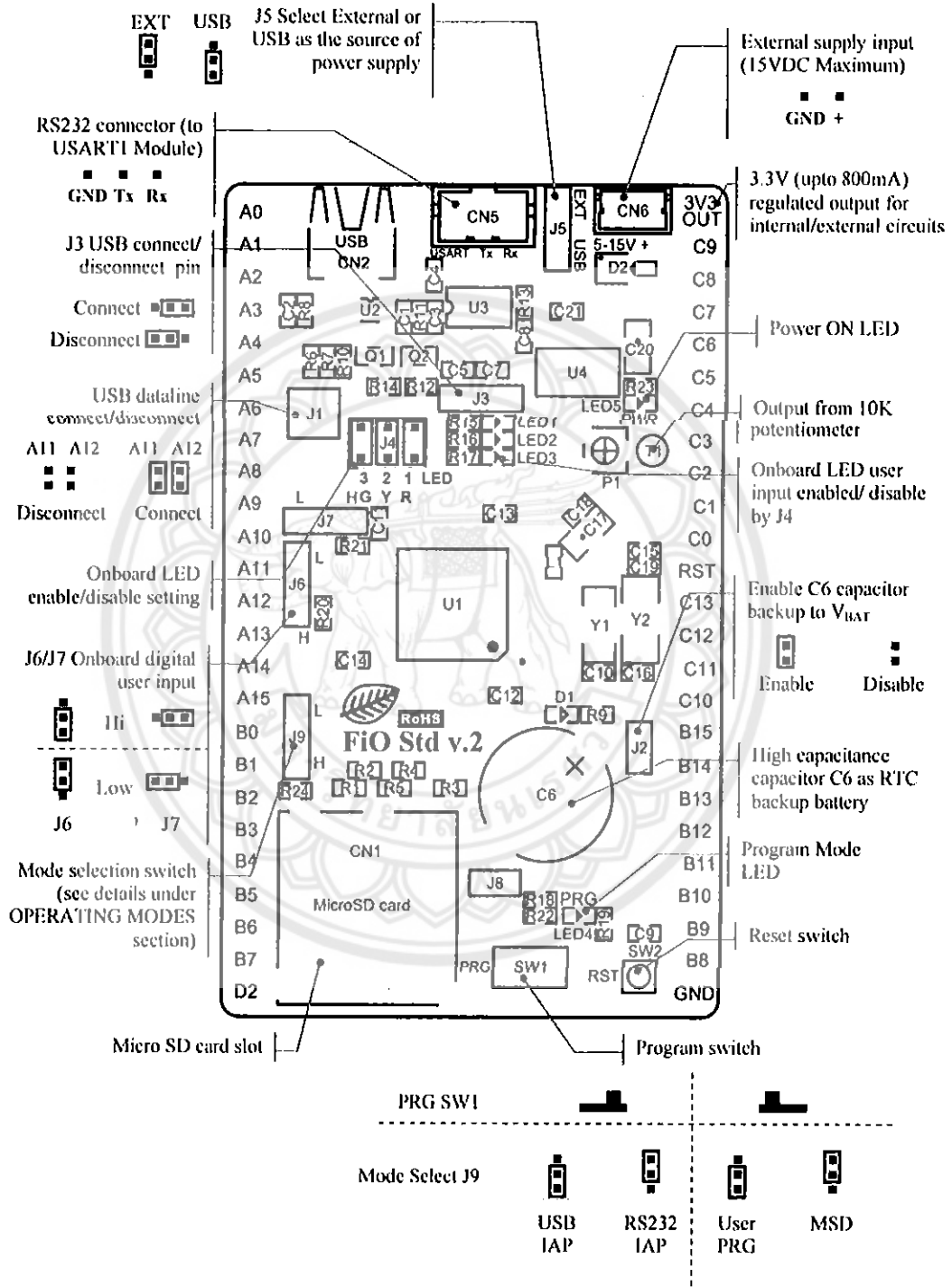


Warning: J8 should not be tampered with at all. Warranty is void if J8 is tampered.
Note: Other jumpers, e.g. J2, J6, J7 not mentioned in the above figures may be set or not, depending upon required functionalities.

SCHEMATIC DIAGRAM



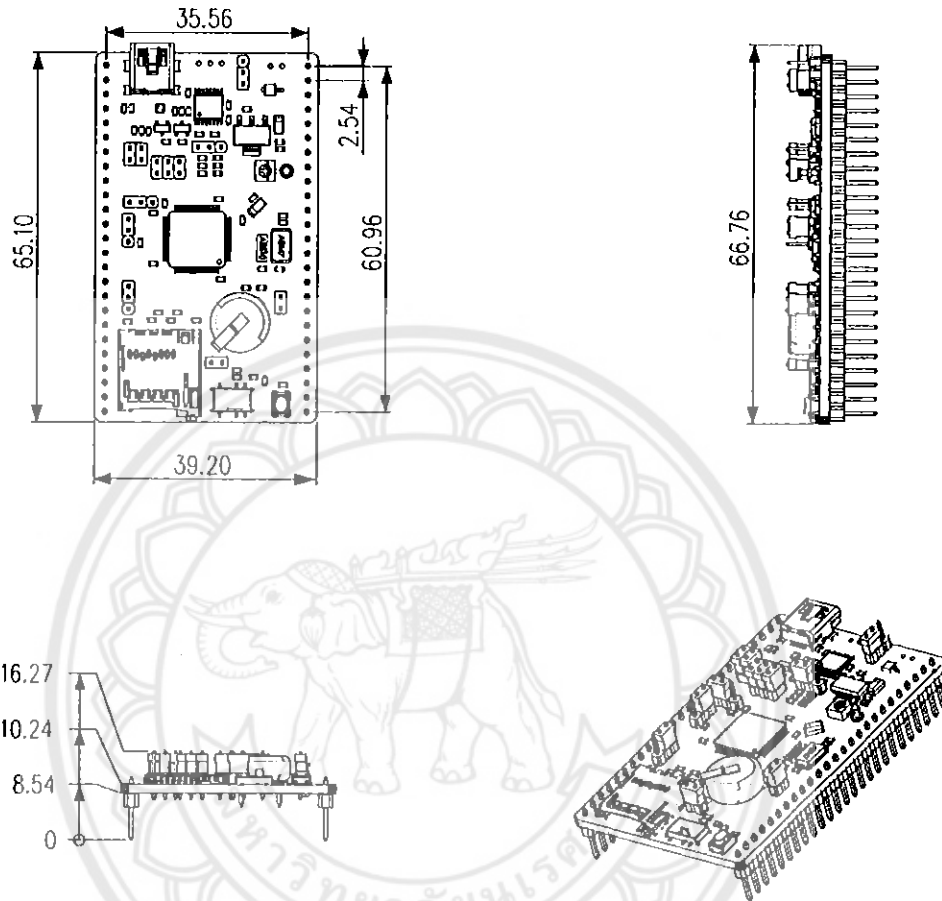
FiO Std PCB LAYOUT



BILL OF MATERIALS

[Manufacture Part No.] Name	Designator	Source
[0603B682K500CT] 6.8nF	C2	www.es.co.th
[0603ZC104KAT2A] 0.1uF	C1, C3, C4, C5, C7, C8, C9, C11, C12, C13, C14, C18, C21	www.mouser.com
[C1608C0G1H100C] 10pF	C10, C15	www.mouser.com
[C1608C0G1H200J] 20pF	C16, C19	www.mouser.com
[293D106X0010A2TE3] 10uF	C17, C20	www.mouser.com
[EECS0HD334H] 0.33F	C6	www.mouser.com
[2908-05WB-MG] MicroSD socket	CN1	www.mouser.com
[67503-1020] USB Mini-B female	CN2	www.mouser.com
[CD0603-B0230] CD0603-B0230	D1	www.mouser.com
[STPS1L30M] STPS1L30M	D2	www.mouser.com
[MMZ1608S601A] Ferrite Beads	L1	www.mouser.com
[APT1608EC] LED Red	LED1	www.mouser.com
[APT1608YC] LED Yellow	LED2	www.mouser.com
[APT1608SGC] LED Green	LED3, LED4, LED5	www.mouser.com
[PVG3A103C01R00] 10K	P1	www.mouser.com
[KTC9014S-C-RTK/P] KTC9014	Q1,Q2	www.es.co.th
[0603WAJ0103T5E] 10K	R1, R2, R3, R4, R5, R11, R14, R18, R20, R21, R22, R24	www.es.co.th
[RC0603FR-0736KL] 36k	R12	www.mouser.com
[0603WAJ0101T5E] 100	R13	www.mouser.com
[CR0603-FX-4870ELF] 487	R15, R16, R17, R19, R23	www.mouser.com
[WR06X000PTL] 0	R6, R7	www.es.co.th
[RC0603JR-071ML] 1M	R8	www.mouser.com
[CRCW06031K50JNEB] 1.5k	R9, R10	www.mouser.com
[SSSS820201] Switch	SW1	www.mouser.com
[SKRSPACE010] Switch	SW2	www.mouser.com
[534-1593-2] Terminal	T1	www.mouser.com
[STM32F103RET6TR] STM32F103RET6	U1	www.mouser.com
[USBLC6-2P6] USBLC6	U2	www.mouser.com
[ST3232EBTR] ST3232EBTR	U3	www.mouser.com
[LD1117S33TR] LD1117S33	U4	www.mouser.com
[ABS10-32.768KHZ-9-T] Xtal 32.768kHz	Y1	www.mouser.com
[ABM7-8.000MHZ-D2Y-T] Xtal 8MHz	Y2	www.mouser.com

ENGINEERING DRAWINGS (Units: mm)



Patent Pending
 PCT/TH2010/000010, PCT/TH2010/000020, and PCT/TH2010/000037

Information in this document is provided solely in connection with Aimagin products. Aimagin reserves the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

www.almagin.com

Copyright 2010 Aimagin - All rights reserved



ภาคผนวก ค

รายละเอียดของ aMG IMU-9A

มหาวิทยาลัยพระนคร

aMG IMU – 9A Datasheet

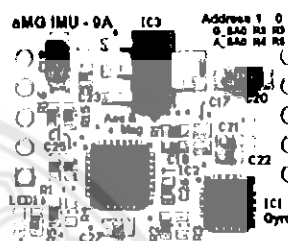
SCH

FEATURES

- Based on STMicroelectronics MEMS Sensors
 - LSM303DLM 3-axis accelerometer and 3-axis magnetometer
 - L3G4200D 3-axis rate gyro
- I2C Interface with adjustable address using resistors
- 3.3V System
- On-board 3.3V regulator capable of accepting up to 12VDC external supply
- Import real-time data for analysis directly into Matlab with RapidSTM32 Blockset I2C Block and FiO Boards
- Develop and test your own algorithm in Matlab / Simulink
- ROHS Compliant

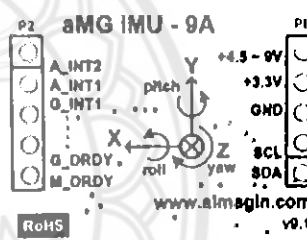
APPLICATIONS

- Compensated compass
- Motion sensing
- Orientation sensing (6 Degree of Freedom)
- Vibration monitoring and compensation



LSM303DLM

- 3-axis accelerometer and 3-axis magnetometer
- Wide supply voltage: 2.16 V to 3.6 V
- Low voltage-compatible IOs (1.8 V)
- Power-down mode
- ± 1.3 to ± 8.1 Gauss magnetic field full-scale
- $\pm 2g / \pm 4g / \pm 8g$ dynamically selectable full-scale
- 2 independent programmable interrupt generators for free-fall and motion detection
- Accelerometer sleep-to-wakeup function
- I²C interface

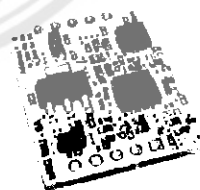


aMG IMU – 9A

L3G4200D Rate Gyro

- 3-axis rate gyro with embedded temperature sensor
- Wide supply voltage: 2.4 V to 3.6 V
- Low voltage-compatible IOs (1.8 V)
- Three selectable full scales (250/500/2000 degree per second)
- I²C/SPI digital output interface (aMG IMU only provide I²C Interface)
- 6 bit-rate value data output
- 8-bit temperature data output
- Two digital output lines (interrupt and data ready)
- Integrated low- and high-pass filters with user selectable bandwidth
- Ultra-stable over temperature and time
- Embedded power-down and sleep mode
- High shock survivability
- Extended operating temperature range (-40 °C to +85 °C)

aMG IMU – 9A



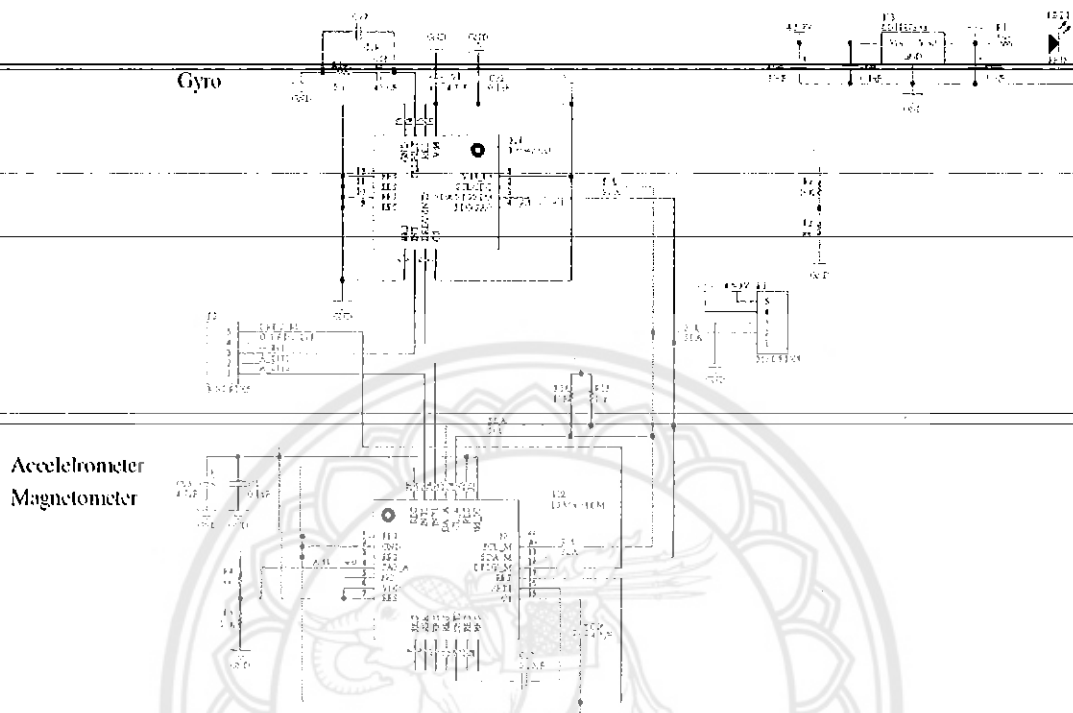
2.54mm pin header



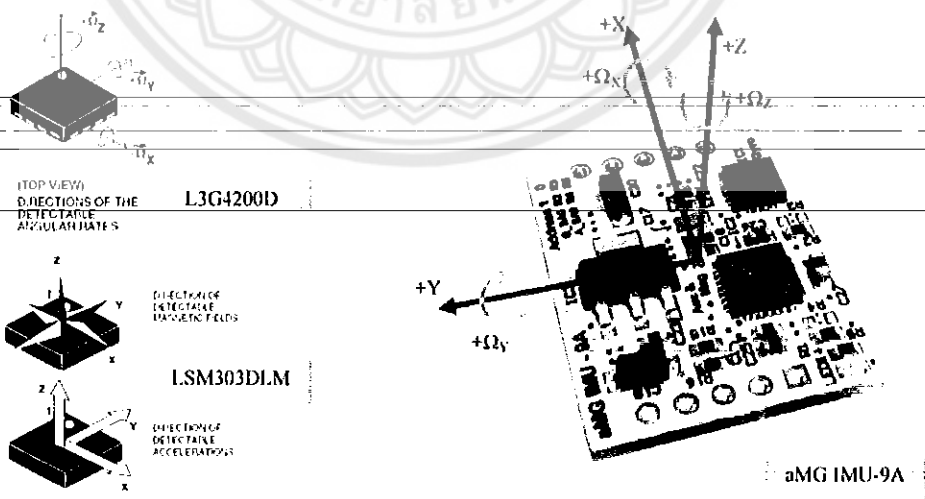
Available in a package

Accel
Magn

SCHEMATIC DIAGRAM



DIRECTION OF DETECTABLE FIELDS



SAMPLE APPLICATIONS 1

To-Be-Added

SAMPLE APPLICATIONS 2

To-Be-Added

DIMENSIONS (mm)

To-Be-Added

FURTHER INFORMATION

For further information and tutorials please visit: <http://www.almagin.com/ang-inm-9a.html>



Information in this document is provided solely in connection with Almagin products. Almagin reserves the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

www.almagin.com

Copyright 2012 Almagin - All rights reserved



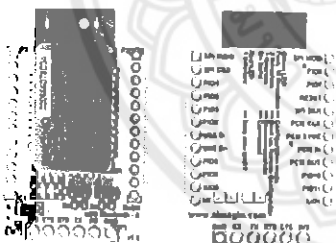
aMG Bluetooth – AC1 / AC2 Datasheet

FEATURES

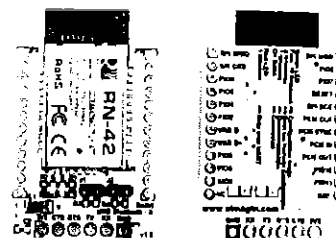
- Based on Roving Networks RN-41 (Class 1 ~100meters) or RN-42 (Class 2 ~20meters) Modules
 - Fully qualified Class 1/2 Bluetooth Versions: 2.1 + EDR, 2.0, 1.2, 1.1
 - Bluetooth SIG qualified
 - UART (SPP or HCI) and USB (HCI only) data connection hardware interfaces
 - Onboard embedded Bluetooth stack (no host processor required)
 - Supports Bluetooth data link to iPhone/iPad/iPod Touch
 - Supports HID profile for making accessories such as keyboards, mouse, pointing devices
 - Programmable low power modes
 - Secure communications, 128 bit encryption
 - Error correction for guaranteed packet delivery
 - UART local and over-the-air RF configuration
 - Auto-discovery/pairing requires no software configuration (instant cable replacement)
 - Castellated SMT pads for easy and reliable PCB mounting

Technical Specifications (RN-41/RN-42 Module)

- Data rate: With onboard stack: 300Kbps: HCI mode: 1.5Mbps sustained, 3Mbps burst
- Frequency Band: 2.412 - 2.484 GHz
- Modulation Techniques: FHSS/GFSK modulation, 79 channels at 1MHz intervals
- Profiles: SPP, DUN, HID, iAP, HCI, RFCOM, L2CAP, SDP
- Supply voltage: 3.3V ± 10%
- Output power: +15dBm (Class 1) +4dBm (Class 2)
- Power consumption: Standby/Idle 25 mA : Connected (normal mode) 30 mA : Connected (low power Sniff) 8 mA : Standby/Idle (Deep sleep enabled) 250uA
- Operating temperature range: -40C to +85C
- Interface: UART, USB, Bluetooth
- Antenna: Chip (RN-41) or PCB Trace (RN-42)
- Size (Bluetooth Module): 0.52" x 1" x 0.07" (13.4mm x 25.8mm x 2mm)
- Certifications: FCC, IC, CE, Bluetooth SIG



aMG Bluetooth – AC1 (with RN-41 Module)



aMG Bluetooth – AC2 (with RN-42 Module)

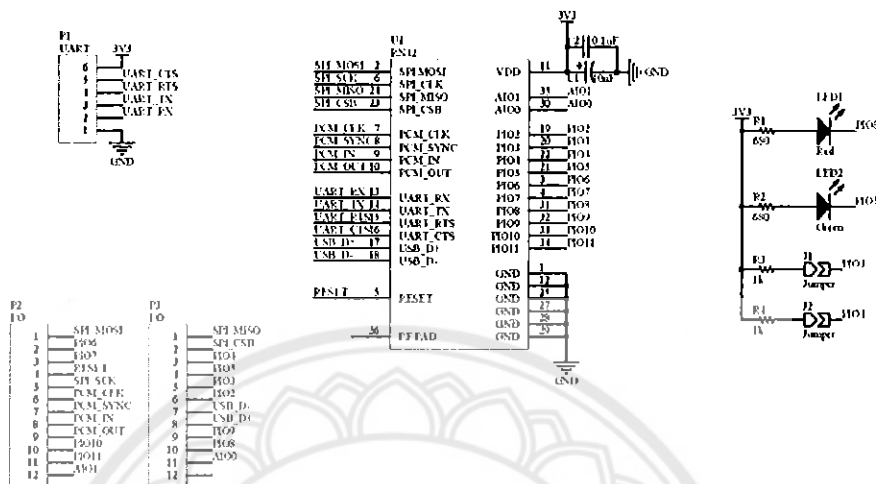
aMG Bluetooth – AC1 / AC2 Module

- 3.3V Supply and Logic
- 2 onboard LED
 - Status LED (Green)
 - Data LED (Red)
- 2 user jumpers
 - Auto Discovery ON/OFF (J1)
 - Factory Default Reset (J2)
- All RN-41 / RN-42 pins available as 2.54mm pitch (3V3 Supply, GND, UART_CTS, UART_TRS, UART_TX, UART_RX) and 2mm pitch (the rest)
- ROHS Compliance

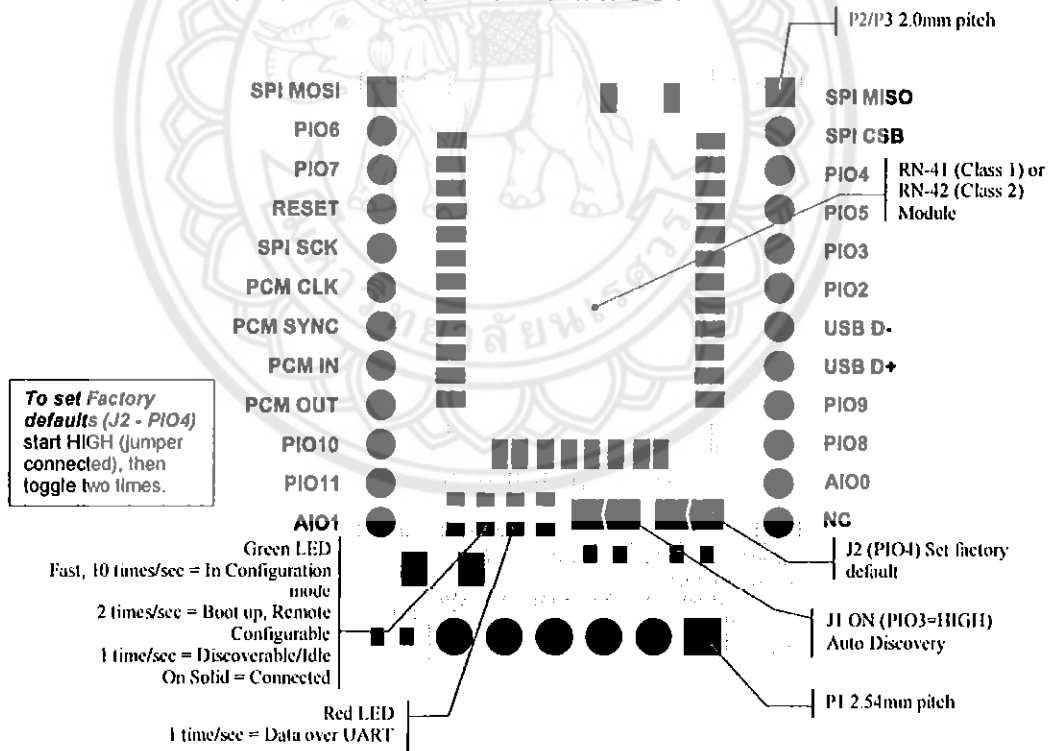
SAMPLE APPLICATIONS

- iPhone, Android phone, TABLET devices wireless data link with embedded devices
- Wireless - RS232 Cable Replacement
- Short-range wireless data link between two embedded devices.

SCHEMATIC DIAGRAM



aMG Bluetooth – AC1 / AC2 PCB LAYOUT

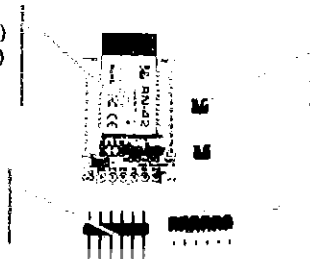


To set Factory defaults (J2 - PIO4) start HIGH (Jumper connected), then toggle two times.

WHAT ARE IN A PACKAGE?

1x aMG Bluetooth-AC1 (RN-41)
or aMG Bluetooth-AC2 (RN-42)
Module

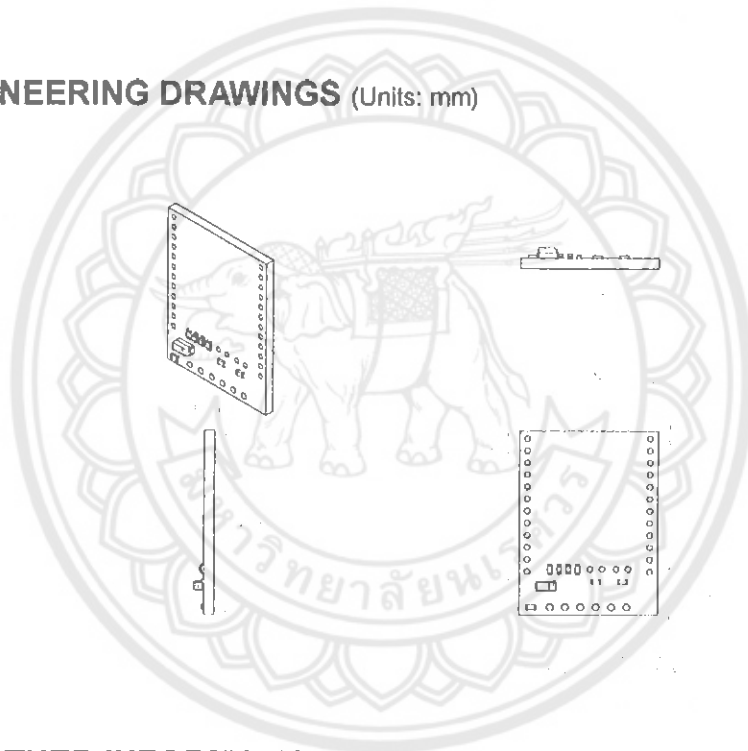
1x 2.54mm 6-pins
straight header



2x 2mm jumpers

1x 2.54mm 6-pins
right-angle header

ENGINEERING DRAWINGS (Units: mm)



FURTHER INFORMATION

For further information about Roving Networks RN-41 and RN-42 module please visit:
http://www.rovingnetworks.com/products/RN_41 or http://www.rovingnetworks.com/products/RN_42

Information in this document is provided solely in connection with Almagin products. Almagin reserves the right to make changes, corrections, modifications or Improvements, to this document, and the products and services described herein at any time, without notice.

www.almagin.com

Copyright 2011 Almagin - All rights reserved

ประวัติผู้ดำเนินโครงการ



ชื่อ นายประกายเพชร ฟองงาม
 ภูมิลำเนา 217/1 หมู่ 12 ต.สุขสำราญ อ.ตากฟ้า จ.นครสวรรค์
 ประวัติการศึกษา

- จบระดับมัธยมศึกษาจาก โรงเรียนตากฟ้าวิชาประสิทธิ์
 จ.นครสวรรค์
- ปัจจุบันกำลังศึกษาระดับปริญญาตรี ชั้นปีที่ 5
 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยนเรศวร

E-mail: phetkhung@hotmail.com



ชื่อ นายโรจน์อานันท์ สังข์นวล
 ภูมิลำเนา 35/1 หมู่ 1 ต.นครป่าหมาก อ.บางกระทุ่ม จ.พิษณุโลก
 ประวัติการศึกษา

- จบระดับมัธยมศึกษาจาก โรงเรียนบางกระทุ่มพิทยาคม
 จ.พิษณุโลก
- ปัจจุบันกำลังศึกษาระดับปริญญาตรี ชั้นปีที่ 5
 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยนเรศวร

Email: pookuny@gmail.com