สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาโปรแกรมควบคุมหุ่นยนต์ขนาดเล็กผ่านโทรศัพท์มือถือ

THE DEVELOPMENT OF MICRO ROBOT CONTROL VIA MOBILE PHONE

T117193

นางสาว ศศิธร ขำศิริ

นางสาว สิทธิรศน์ หงษ์สุวรรณ

ประทานจาก
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2553

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
วิทยบริการ
เลขทะเบียน 117193
วัน,เดือน,ปี 19 ก.ค. 2554

b 12211782
i

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE DEVELOPMENT OF MICRO ROBOT CONTROL VIA MOBILE PHONE

MISS SASITHORN KHAMSIRI

MISS SITTHIROS HONGSUWAN

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR DEGREE OF BACHELOR OF SCIENCE

IN COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงานพิเศษ	การพัฒนาโปรแกรมควบคุมหุ่นยนต์ขนาดเล็กผ่านโทรศัพท์มือถือ		
	The development of micro robot control via mobile phone		
ชื่อนักศึกษา	นางสาวศศิธร	ขำศิริ	50050205
	นางสาวสิทธิรศน์	หงษ์สุวรรณ	50050218
ปริญญา	วิทยาศาสตรบัณฑิต		
สาขาวิชา	วิทยาการคอมพิวเตอร์		
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ธีรวัฒน์ ประกอบผล		

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชา วิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2553

คณะกรรมการสอบ	ลายมือชื่อ
ผู้ช่วยศาสตราจารย์ ดร.จีรพร วีระพันธุ์ ประธานกรรมการ	
ผู้ช่วยศาสตราจารย์ ดร.ศรัณย์ อินทโกสุม กรรมการ	
รองศาสตราจารย์ธีรวัฒน์ ประกอบผล กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงานพิเศษ การพัฒนาโปรแกรมควบคุมหุ่นยนต์ขนาดเล็กผ่านโทรศัพท์มือถือ

ชื่อนักศึกษา นางสาวศศิธร ขำศิริ
นางสาวสิทธิรศน์ หงษ์สุวรรณ

ปริญญา วิทยาศาสตรบัณฑิต

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2553

อาจารย์ที่ปรึกษา รองศาสตราจารย์ธีรวัฒน์ ประกอบผล

บทคัดย่อ

ปัจจุบันเทคโนโลยีต่างๆได้ถูกพัฒนาขึ้นมาอย่างมากมาย เพื่อตอบสนองความต้องการของมนุษย์ เช่น เครื่องมือสื่อสาร หรือ เทคโนโลยีการส่งข้อมูล ล้วนแต่มีความทันสมัย สะดวกสบาย ใช้งานง่าย และทำงานได้อย่างรวดเร็ว ดังเห็นได้จากโทรศัพท์มือถือในปัจจุบันที่มีฟังก์ชั่น และแอพพลิเคชั่นมากมาย เช่นเทคโนโลยีบลูทูธสำหรับรองรับการรับส่งข้อมูล หรือ เทคโนโลยีจาวาที่รองรับแอพพลิเคชั่นที่เขียนโดยภาษาจาวา ผู้จัดทำจึงคิดนำเทคโนโลยีเหล่านี้มาพัฒนาเข้าด้วยกัน เพื่อเป็นแอพพลิเคชั่นบนโทรศัพท์มือถือที่ถูกออกแบบโดยใช้เทคโนโลยี J2ME ซึ่งเป็นเทคโนโลยีหนึ่งของจาวา มาทำการควบคุมทิศทางการเคลื่อนที่เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา ของหุ่นยนต์ขนาดเล็กผ่านเทคโนโลยีบลูทูธ โดยการกดปุ่มบนโทรศัพท์มือถือ จะมีการส่งสัญญาณบลูทูธไปยังบอร์ดไมโครคอนโทรลเลอร์ เพื่อให้โปรแกรมบนบอร์ดไมโครคอนโทรเลอร์ที่ถูกเขียนด้วยภาษาพีเบสิกทำการประมวลผล แล้วควบคุมการทำงานของรีเลย์ที่ต่อเข้ากับมอเตอร์ และเมื่อมอเตอร์ทำงาน หุ่นยนต์จะเคลื่อนที่ไปตามทิศทางที่กำหนด

คำสำคัญ : โปรแกรม, หุ่นยนต์, โทรศัพท์, บลูทูธ, J2ME, ไมโครคอนโทรลเลอร์, พีเบสิก, รีเลย์, มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	The development of micro robot control via mobile phone
Student	Miss Sasithorn Khamsiri
	Miss Sitthiros Hongsuwan
Degree	Bachelor of Science
Major Program	Computer Science
Academic Year	2010
Advisor	Associate Professor Teerawat Prakobphon

ABSTRACT

In this day, a number of technologies have been developed tremendously to respond to people's need; For instance, communication or data transmission technology. These technologies are up-to-date, easy to use and quick at processing. However, as mobile phones these days have many functions and applications; for instance, Bluetooth for data transmission or Java applications technology written. All the technologies mentioned above have been gathered together so as to be able to success fully design mobile telephone applications using J2ME technology, which is one of Java technology, to control the directions, forward, backward, left and right, of a micro robot via Bluetooth technology by pressing a button on mobile phone. Signals are then sent to the Bluetooth microcontroller board; in order for the program in board to be, written with PBasic language. Such process will accordingly control the relay which is connected to the motor. Once the motor works properly, as well as the robot will move in the direction specified.

Keywords : Program, Robot, Phone, Bluetooth, J2ME, Microcontroller, PBasic, relay, motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการจัดทำโครงงานพิเศษเรื่อง การพัฒนาโปรแกรมควบคุมหุ่นยนต์ขนาดเล็กผ่านโทรศัพท์มือถือ นี้สำเร็จลุล่วงได้ด้วยดี คณะผู้จัดทำขอขอบพระคุณ รองศาสตราจารย์ธีรวัฒน์ ประกอบผล อาจารย์ที่ปรึกษาโครงงานพิเศษ ที่ได้เสียสละเวลาให้คำปรึกษา คำแนะนำ ตรวจแก้ไขปัญหาต่างๆ และเอาใจใส่ตลอดระยะเวลาทั้งหมดที่ทำโครงงานพิเศษนี้

ขอขอบคุณ สาขาวิชาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับสิ่งอำนวยความสะดวกมากมายที่มีให้บริการ ทั้งห้องทำงาน และอินเตอร์เน็ตเพื่อการค้นคว้าข้อมูลต่างๆที่จำเป็นสำหรับการทำโครงงานพิเศษนี้

ขอขอบพระคุณคณาจารย์ สาขาวิทยาการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาให้แก่คณะผู้จัดทำ

สุดท้ายนี้ต้องขอกราบขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้มีวันนี้คือ บิดา มารดา อันเป็นที่เคารพรักของคณะผู้จัดทำ ซึ่งได้เลี้ยงดูมาเป็นอย่างดี พร้อมทั้งให้โอกาสการศึกษาอย่างเต็มที่และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆด้านอันหาที่เปรียบมิได้ รวมถึงขอบคุณพี่ๆ เพื่อนๆ และน้องๆ ที่ให้กำลังใจ คำปรึกษา และความช่วยเหลือไว้ ณ ที่นี้

คุณค่าและประโยชน์อันพึงมาจากปัญหาพิเศษฉบับนี้ คณะผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

นางสาวศศิธร ขำศิริ

นางสาวสิทธิรศน์ หงษ์สุวรรณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VIII

สารบัญภาพ IX

บทที่ 1 บทนำ

1.1 ความสำคัญและความเป็นมา 1

1.2 วัตถุประสงค์ของปัญหาพิเศษ 2

1.3 ขอบเขตของปัญหาพิเศษ 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

1.5 ขั้นตอนในการดำเนินการ 2

1.6 อุปกรณ์ที่ใช้ทำในปัญหาพิเศษ 2

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 เทคโนโลยี J2ME 3

2.1.1 สถาปัตยกรรมของ J2ME 4

2.1.1.1 คอนฟิกกูเรชัน 4

2.1.1.2 โพรไฟล์ 6

2.1.2 MIDlet 6

2.1.3 ความสำคัญของไฟล์ .jar และ .jad 7

2.1.4 MIDlet Suit 9

2.1.5 การทำงานของ MIDlet 9

2.1.6 วงจรการทำงานของ MIDlet 10

2.1.7 ซอฟท์แวร์จัดการแอพพลิเคชั่น 11

2.2 เทคโนโลยีบลูทูธ 11

2.2.1 ความเป็นมาของเทคโนโลยีบลูทูธ 12

2.2.2 จุดมุ่งหมายของบลูทูธ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.2.3 วัตถุประสงค์ของบลูทูธ....14
2.2.4 Protocol Stack.... 14
2.2.5 เทคนิคการส่งข้อมูล.... 16
2.2.6 เทคนิคการกระโดดข้ามทางความถี่....17
2.2.7 รูปแบบแพคเกตของบลูทูธ....18
2.2.8 ระบบรักษาความปลอดภัยของบลูทูธ....20
2.2.9 ลักษณะเฉพาะของแอพพลิเคชั่นซอฟท์แวร์ของบลูทูธ....20
2.2.9.1 โพรไฟล์อุปกรณ์ทั่วไป....21
2.2.9.2 โพรไฟล์บริการเชื่อมต่อระหว่างอุปกรณ์.... 21
2.2.9.3 โพรไฟล์พอร์ตอนุกรม.... 22
2.2.9.4 โพรไฟล์แลกเปลี่ยนข้อมูล....22
2.2.9.5 โพรไฟล์กำหนดสัญญาณ.... 23
2.2.10 ส่วนควบคุมการเชื่อมต่อ.... 23
2.3 ไมโครคอนโทรลเลอร์.... 24
2.3.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์.... 25
2.3.1.1 หน่วยประมวลผลกลางซีพียู....25
2.3.1.2 หน่วยความจำ....26
2.3.1.3 ส่วนติดต่ออุปกรณ์ภายนอก....26
2.3.1.4 บัสข้อมูลและแอดเดรส.... 27
2.3.1.5 วงจรกำเนิดสัญญาณนาฬิกา....27
2.3.2 บอร์ดเบสิกแสตมป์.... 27
2.3.2.1 การจัดขาของบอร์ดเบสิกแสตมป์....28
2.3.2.2 วงจรภายในบอร์ดเบสิกแสตมป์.... 29
2.3.3 ภาษาพีเบสิก.... 31
2.3.3.1 รูปแบบโปรแกรม....31
2.3.3.2 รูปแบบการเขียนภาษาพีเบสิก.... 32
2.3.3.3 รายละเอียดชุดคำสั่งของเบสิกแสตมป์.... 33
2.3.4 รีเลย์อิเล็กทรอนิกส์....36
2.3.5 มอเตอร์กระแสตรง.... 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.5.1 หลักการทำงานมอเตอร์.....38
2.3.5.2 การนำรีเลย์ควบคุมทิศทางหมุนของมอเตอร์กระแสตรง.....38

บทที่ 3 การออกแบบและการพัฒนา

3.1 ส่วนโปรแกรมบนโทรศัพท์มือถือที่ใช้ควบคุมทิศทางของหุ่นยนต์ขนาดเล็ก.....40
3.1.1 ส่วน Interface.....40
3.1.2 ส่วนบลูทูธ.....41
3.1.2.1 ส่วนการค้นหาอุปกรณ์.....41
3.1.2.2 ส่วนการส่งรหัสคำสั่ง.....41
3.1.2.3 ส่วนการรับรหัสคำสั่ง.....42
3.1.3 สรุปการทำงานของโปรแกรมบนโทรศัพท์มือถือ.....42
3.2 ส่วนโปรแกรมบนไมโครคอนโทรลเลอร์.....45
3.2.1 ส่วนของการตรวจสอบรหัสผ่าน.....45
3.2.2 ส่วนของการควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก.....45
3.2.3 สรุปการทำงานของโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์.....45
3.3 การประกอบหุ่นยนต์ขนาดเล็ก.....48
3.3.1 ขั้นตอนการประกอบตัวหุ่นยนต์ขนาดเล็ก.....49
3.3.2 การต่อวงจรมอเตอร์ของหุ่นยนต์ขนาดเล็ก.....56

บทที่ 4 ผลการทดลอง

4.1 การทดลองบนโทรศัพท์มือถือ.....60
4.2 การทดลองบนบอร์ดไมโครคอนโทรลเลอร์.....65

บทที่ 5 สรุปผลการดำเนินงาน และข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน.....70
5.1.1 สรุปผลการดำเนินงาน.....70
5.1.2 การพัฒนาโปรแกรม.....70
5.1.3 ประสิทธิภาพและคุณสมบัติของโปรแกรม.....70
5.1.4 ข้อจำกัดของโปรแกรม.....71
5.2 ข้อเสนอแนะ.....71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

รายการอ้างอิง 72

ภาคผนวก ก. โค้ดโปรแกรมบนโทรศัพท์มือถือ..... 74

ภาคผนวก ข. โค้ดโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์..... 87

ภาคผนวก ค. คุณสมบัติของอุปกรณ์ที่ใช้ในโครงงานปัญหาพิเศษ..... 90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แพ็คเกจของไพรไฟล์ MIDP-1.0 6
2.2 แพ็คเกจของไพรไฟล์ MIDP-2.0 7
2.3 Method พื้นฐานของ MIDlet 9
2.4 ตารางแสดงการเปรียบเทียบระหว่าง OSI Model และ Bluetooth Module 15
2.5 รายละเอียดของการจัดขาเชื่อมต่อและหน้าที่ของเบสิกแสตมป์ 28
3.1 แสดงรูปแบบการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก 46
4.1 หมายเลขควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

2.1 โครงสร้างของ J2ME 4

2.2 ความสัมพันธ์ของ CLDC กับ CDC 5

2.3 วงจรการทำงานของ MIDlet..... 10

2.4 สัญลักษณ์บลูทูธ..... 13

2.5 Frequency Hopping และการเกิด Collision 17

2.6 Slot แพคเกต..... 18

2.7 การสร้างแพคเกตข้อมูล..... 18

2.8 รูปแบบของบลูทูธแพคเกต..... 20

2.9 Bluetooth Profile 21

2.10 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 25

2.11 โครงสร้างและส่วนประกอบหลักของซีพียู..... 26

2.12 การจัดขาเชื่อมต่อของเบสิกแสตมป์ 28

2.13 วงจรสมบูรณ์ของเบสิกแสตมป์ 30

2.14 ส่วนประกอบภายในรีเลย์ 36

2.15 รีเลย์และสัญลักษณ์ของรีเลย์ 36

2.16 สภาวะการทำงานของรีเลย์..... 37

2.17 ตำแหน่งขา และ รายละเอียดการใช้งาน..... 37

2.18 มอเตอร์กระแสตรงขนาดเล็กและสัญลักษณ์มอเตอร์กระแสตรง..... 37

2.19 ลักษณะการทำงานของมอเตอร์ 38

2.20 แสดงการกลับทิศทางของมอเตอร์กระแสตรงโดยใช้รีเลย์..... 38

3.1 ระบบควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก 39

3.2 แป้นควบคุมบนโทรศัพท์มือถือ..... 42

3.3 Flow chart การทำงานของโปรแกรมบนโทรศัพท์เคลื่อนที่..... 44

3.4 Flow chart การทำงานของโปรแกรมบนไมโครคอนโทรลเลอร์..... 47

3.5 วงจรหุ่นยนต์ขนาดเล็ก 48

3.6 ติดตั้งบอร์ดเบสิกแสตมป์ เข้ากับแผงวงจรสำหรับพัฒนาโครงงาน..... 49

3.7 ติดตั้งรีเลย์เข้ากับแผงวงจรสำหรับพัฒนาโครงงาน 49

3.8 ติดตั้งบลูทูธเข้ากับแผงวงจรสำหรับพัฒนาโครงงาน 50

3.9 ติดตั้งสายจ่ายไฟจากแผงวงจรสำหรับพัฒนาโครงงานเข้ากับรีเลย์..... 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

3.10 ติดตั้งสายไฟเข้ากับขา NC ของรีเลย์.....51
3.11 ติดตั้งสายไฟเข้ากับขา NO ของรีเลย์.....51
3.12 ติดตั้งสายไฟเข้ากับขา C ของรีเลย์.....52
3.13 ติดตั้งสายไฟเข้ากับกระบะถ่าน.....52
3.14 ติดตั้งชุดเฟืองขับมอเตอร์เข้ากับถาดรองกระบะถ่าน.....53
3. .15 ติดตั้งเสารองโลหะเข้ากับถาดรองกระบะถ่าน.....53
3.16 นำล้อพลาสติกพร้อมยางเสียบเข้ากับแกนมอเตอร์.....53
3.17 ติดตั้งชุดมอเตอร์เข้ากับฐานพลาสติกของหุ่นยนต์.....54
3.18 ติดตั้งสายไฟที่ต่อมาจากขา C เข้ากับมอเตอร์.....54
3.19 ติดตั้งกระบะถ่านสำหรับถ่าน 12V.....55
3.20 ประกอบแผงวงจรสำหรับพัฒนาโครงงานเข้ากับชุดล้อ.....55
3.21 การต่อวงจรมอเตอร์ภายในหุ่นยนต์ขนาดเล็ก.....56
3.22 การทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 1 ทำงาน.....56
3.23 การทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 2 ทำงาน.....57
3.24 การทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 3 ทำงาน.....57
3.25 การทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 4 ทำงาน.....58
3.26 การทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 1 และ 2 ทำงานพร้อมกัน.....58
3.27 การทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 3 และ 4 ทำงานพร้อมกัน.....59
4.1 ข้อความแสดงเมื่อค้นหาและการเชื่อมต่ออุปกรณ์รับสัญญาณบลูทูธสำเร็จ.....60
4.2 ข้อความแสดงเมื่อการค้นหาและการเชื่อมต่ออุปกรณ์รับสัญญาณบลูทูธไม่สำเร็จ.....61
4.3 หน้าจอแสดงการใส่รหัสผ่าน.....61
4.4 ข้อความแสดงเมื่อใส่รหัสผ่านผิดยังไม่ครบ 3 ครั้ง.....62
4.5 หน้าจอแสดงเมื่อใส่รหัสผ่านผิดครบ 3 ครั้ง.....62
4.6 หน้าจอแสดงเมื่อใส่รหัสผ่านถูกต้อง.....63
4.8 ข้อความแสดงเมื่อมีข้อผิดพลาดขณะส่งคำสั่ง.....64
4.9 หน้าจอแสดงการออกจากโปรแกรม.....64
4.10 หน้าจอโปรแกรม BASIC Stamp Editor.....65
4.11 หน้าต่างแสดงสถานะของบอร์ดไมโครคอนโทรเลอร์ในการรอรับรหัสผ่านจากผู้ใช้.....65
4.12 หน้าต่างโปรแกรมแสดงเมื่อใส่รหัสผ่านผิด.....66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

4.13 หน้าต่างโปรแกรมแสดงเมื่อใส่รหัสผ่านถูกต้อง..... 67
4.14 หน้าต่างโปรแกรมแสดงสถานะของการควบคุมการปิดเปิดรีเลย์..... 68
4.15 หน้าต่างโปรแกรมแสดงการออกจากโปรแกรม..... 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและความเป็นมา

มนุษย์มีความต้องการพัฒนาเทคโนโลยีต่างๆ ขึ้นมากมาย เพื่ออำนวยความสะดวกในชีวิตประจำวัน และมีการพัฒนามาอย่างต่อเนื่อง ตัวอย่างเช่น การเดินทางในอดีตใช้เรือเป็นยานพาหนะ ต่อมามีการประดิษฐ์รถขึ้นเพื่ออำนวยความสะดวกสบายในการเดินทางและเป็นการประหยัดเวลา หรือการติดต่อสื่อสารผ่านทางโทรเลข ต่อมามีการผลิตโทรศัพท์เพื่อความรวดเร็วในการสื่อสาร และพัฒนาอย่างต่อเนื่องจนสามารถใช้งานได้เสมือนเป็นคอมพิวเตอร์ขนาดพกพา ในด้านเทคโนโลยีการเชื่อมต่อระหว่างอุปกรณ์อิเล็กทรอนิกส์ แรกเริ่มใช้สายเคเบิลเป็นตัวกลาง แต่เนื่องจากการใช้สายเคเบิลไม่ได้รับความสะดวกสบาย จึงมีการพัฒนาจนกลายเป็นเทคโนโลยีไร้สายที่หลากหลายรูปแบบ เทคโนโลยีไร้สายรูปแบบหนึ่งที่มีใช้อย่างแพร่หลายนั้น คือ เทคโนโลยีบลูทูธ ซึ่งใช้ในการเชื่อมโยงการสื่อสารไร้สายในแถบความถี่ 2.45 GHz ในการส่งข้อมูลใช้วิธีการกระโดดเปลี่ยนความถี่ช่องสัญญาณ มีรัศมีการทำงานประมาณ 10 เมตร ข้อดีของเทคโนโลยีบลูทูธ คือ ทำให้อุปกรณ์อิเล็กทรอนิกส์สามารถติดต่อกันแบบไร้สายในระยะห่างสั้นๆ มีความปลอดภัยในการสื่อสาร ขนาดเล็ก ใช้พลังงานน้อยและมีราคาถูก ปัจจุบันเทคโนโลยีบลูทูธถูกบรรจุเป็นหนึ่งในฟังก์ชั่นบนโทรศัพท์มือถือและอุปกรณ์อิเล็กทรอนิกส์อื่นๆ จึงคิดนำเอาเทคโนโลยีบลูทูธมาประยุกต์ใช้ โดยเชื่อมต่อสัญญาณบลูทูธระหว่างโทรศัพท์มือถือกับบอร์ดไมโครคอนโทรลเลอร์ประกอบด้วยบอร์ดรับสัญญาณบลูทูธ (ZX-Bluetooth), บอร์ดพัฒนาโครงงานสำหรับ i-Stamp (JX-2SX plus II), บอร์ดเบสิกแสตมป์ (i-Stamp 2P24) ซึ่งภายในบอร์ดเบสิกแสตมป์มีโปรแกรมที่ถูกเขียนด้วยพีภาษาเบสิก แล้วนำมาประดิษฐ์เป็นหุ่นยนต์ขนาดเล็กที่สามารถควบคุมทิศทางการเคลื่อนที่ผ่านโทรศัพท์มือถือ โดยเขียนแอพพลิเคชั่นการควบคุมลงบนโทรศัพท์มือถือ ซึ่งเป็นแอพพลิเคชั่นที่ถูกเขียนด้วยเทคโนโลยี J2ME หรือ Java 2 Micro Edition ที่เป็นเทคโนโลยีหนึ่งของจาวา ข้อดีของ J2ME คือ สามารถเข้าไปรันและทำงานอยู่ในอุปกรณ์อิเล็กทรอนิกส์ที่มีความจำกัดหลายอย่าง ทั้งขนาด หน่วยความจำ ความเร็วของการประมวลผล และระยะเวลาการใช้งานแบตเตอรี่ เช่น โทรศัพท์มือถือ คอมพิวเตอร์ส่วนตัวขนาดเล็ก (PDA) หรือเพจเจอร์ และในเทคโนโลยี J2ME มีแพ็คเกจและคำสั่งต่างๆในการทำงานและส่วนติดต่อกับผู้ใช้ที่ง่ายต่อการเรียกใช้ จึงนำคำสั่งต่างๆ และเทคโนโลยีบลูทูธมาประยุกต์เข้าด้วยกัน เพื่อเป็นแอพพลิเคชั่นที่สามารถใช้ควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็กได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของปัญหาพิเศษ

1.) ศึกษาเกี่ยวกับการทำงานภาษาจาวา J2ME ของโทรศัพท์มือถือทั่วไป เพื่อเชื่อมโยงข้อมูลในการรับ-ส่งข้อมูลและควบคุมการทำงานของไมโครคอนโทรเลอร์ได้

2.) เพื่อวิเคราะห์และพัฒนาโปรแกรมควบคุมการทำงานไมโครโทรลเลอร์ โดยใช้เทคโนโลยีบลูทูธ

1.3 ขอบเขตของปัญหาพิเศษ

1.) ออกแบบโปรแกรมควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ขนาดเล็กผ่านบลูทูธบนโทรศัพท์มือถือ

2.) สามารถเขียนภาษาจาวา J2ME ของโทรศัพท์มือถือทั่วไป เพื่อเชื่อมโยงข้อมูลในการส่งข้อมูลและควบคุมไมโครคอนโทรเลอร์ได้

3.) มีความเข้าใจในวงจรไมโครคอนโทรลเลอร์ เพื่อให้สามารถติดต่อและควบคุมได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.) มีความรู้ในการเขียนภาษาจาวา J2ME เพื่อใช้ในโทรศัพท์มือถือทั่วไป

2.) ทำให้อุปกรณ์ที่ใช้เทคโนโลยีบลูทูธเป็นที่รู้จักอย่างกว้างขวาง

3.) เป็นต้นแบบในการพัฒนา เพื่อใช้ร่วมกับงานประเภทอื่นๆ ในอนาคต

4.) มีความรู้ในการเขียนภาษาพีเบสิก เพื่อใช้ในไมโครคอนโทรเลอร์

1.5 ขั้นตอนในการดำเนินการ

1.) ศึกษาค้นคว้าข้อมูลเกี่ยวกับการทำงานของบลูทูธ และการติดต่อกับไมโครคอนโทรลเลอร์

2.) วิเคราะห์และออกแบบการทำงาน

3.) เขียนโค้ดโปรแกรมที่สามารถนำไปรันบนโทรศัพท์มือถือได้

4.) เขียนโค้ดเพื่อทดสอบการรับค่าการส่งสัญญาณบลูทูธจากโค้ดบนโทรศัพท์มือถือ

5.) เขียนโค้ดโปรแกรมควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ขนาดเล็กที่บอร์ดไมโครคอนโทรลเลอร์

6.) ทดสอบ ตรวจสอบและแก้ไขข้อผิดพลาด

7.) สรุปและเขียนผลการทำงาน

1.6 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1.) โทรศัพท์มือถือที่มีเทคโนโลยีบลูทูธ และรองรับการทำงานของจาวาแอพพลิเคชั่น

2.) อุปกรณ์สำหรับสร้างและพัฒนาโปรแกรมเพื่อควบคุมหุ่นยนต์ขนาดเล็ก

3.) วงจรไมโครคอนโทรเลอร์ และตัวรับ-ส่งสัญญาณบลูทูธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 เทคโนโลยี J2ME

ภาษาจาวาพัฒนาโดยบริษัท Sun Microsystems เป็นภาษาเชิงวัตถุที่มีจุดเด่นที่สามารถทำงานได้ทุกระบบปฏิบัติการ (Write Once Run Anywhere) โดยไม่ต้องคอมไพล์โปรแกรมใหม่ หลักการทำงานของภาษาคือ โปรแกรมที่เขียนด้วยภาษาจาวา จะถูกนำไปคอมไพล์โดยจาวาคอมไพเลอร์ ให้เป็นกลุ่มของคำสั่งไบต์โค้ด (byte code) ที่เรียกว่าคลาส โดยไบต์โค้ดเหล่านี้จะถูกนำไปใช้โดย JVM (Java Virtual Machine) เพื่อแปลความหมายเป็นภาษาเครื่องของแต่ละระบบปฏิบัติการ ด้วยการทำงานในลักษณะนี้ การพัฒนาโปรแกรมโดยจาวาจึงไม่ต้องคำนึงถึงระบบปฏิบัติการที่จะมารองรับการทำงานของโปรแกรม

บริษัท Sun เป็นผู้ริเริ่มในการพัฒนาเทคโนโลยี J2ME แต่ในปัจจุบันได้รับการสนับสนุนจากผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ชั้นนำของโลกหลายบริษัท โดยอยู่ภายใต้การดูแลของ JCP (Java Community Process) เพื่อให้การพัฒนา J2ME เป็นไปในทิศทางเดียวกันและอยู่ใต้มาตรฐานจาวาของทางบริษัท Sun ซึ่งในจาวาเวอร์ชัน 2 นี้ได้แยกออกเป็น 3 เอดิชัน (Edition) คือ

1.) J2SE (Java 2 Standard Edition) เป็นตัวพื้นฐานของจาวาทุกเอดิชัน โดยในเอดิชันนี้มีคลาส แพ็คเกจ และ API หลักของจาวาไว้ เพื่อใช้พัฒนาแอพพลิเคชั่นจาวามาตรฐาน แอพพลิเคชั่นที่รันในเซิร์ฟเวอร์ รวมทั้งแอพพลิเคชั่นที่รันในบราวเซอร์หรือที่เรียกว่าแอพเพลต (Applet) ทำงานบน JVM (Java Virtual Machine)

2.) J2EE (Java 2 Enterprise Edition) เป็นเอดิชันที่รวมเอา API พื้นฐานของจาวา และเทคโนโลยีอื่นที่ไม่ใช่ของจาวา ตัวอย่างเช่น JDBC, CORBA หรือ XML ไว้ด้วยกันสำหรับใช้พัฒนาแอพพลิเคชั่นแบบมัลติเทียร์ (Multitiered) หรือแอพพลิเคชั่นแบบกระจาย (Distributed Application) ซึ่งส่วนมากเป็นแอพพลิเคชั่นที่ทำงานบนฝั่งเซิร์ฟเวอร์ ใช้งานในระบบงานใหญ่ๆ โดยเพิ่มมาจากตัว J2SE เพื่อสามารถรองรับการทำงานแบบ Server Side และรองรับการใช้งานจาก Client จำนวนมากๆ ได้ ในเวอร์ชันเก่าทำงานบน HotSpot VM (HotSpot Virtual Machine) ส่วนเวอร์ชัน 5.0 ทำงานบน JVM[TM] (Java [TM] Virtual Machine)

3.) J2ME (Java 2 Micro Edition) เป็นเอดิชันย่อยของ J2SE โดยมีคลาส แพ็คเกจ และ API พิเศษ (ที่ไม่มีใน J2SE) ซึ่งเป็นทั้งสภาวะแวดล้อมของการพัฒนาและตัวรันไทม์ (Run Time) ไปพร้อมๆ กัน เป้าหมายหลักของ J2ME คือ การนำซอร์ฟแวร์ของจาวา เข้าไปรันและทำงานอยู่ในอุปกรณ์อิเล็กทรอนิกส์ เช่น โทรศัพท์มือถือ หรือ PDAs ซึ่งอุปกรณ์เหล่านี้มีขนาดเล็ก

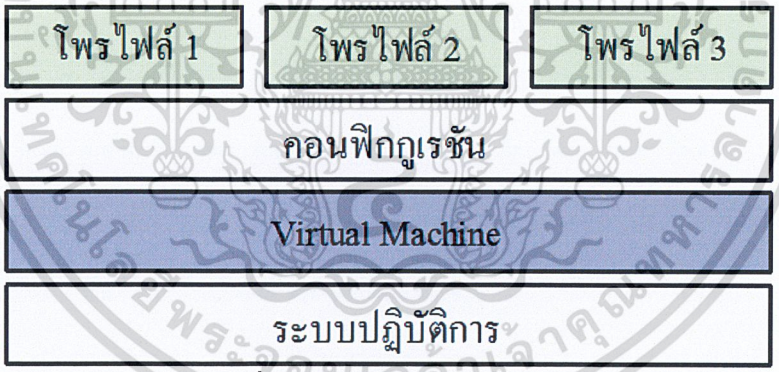
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำน้อย และมีขีดความสามารถในการประมวลผลต่ำกว่าคอมพิวเตอร์ทั่วไป โดยตัดฟังก์ชั่นที่ไม่จำเป็นออกไป ทำงานบน KVM (Kilobyte Virtual Machine)

จุดประสงค์ของการแบ่งจาวาออกเป็นเอดิชันย่อยก็เพื่อความเหมาะสมกับการเลือกไปพัฒนา แอพพลิเคชั่นตามขนาดและลักษณะของงาน โดยไม่จำเป็นต้องใช้ความสามารถทั้งหมด เพราะภาษาจาวาได้จัดเตรียมแพ็คเกจ คลาส และ API สำหรับเขียนโปรแกรมเรียกใช้งานไว้เป็นจำนวนมาก ทั้งการรับข้อมูล การแสดงผล และการตรวจจับข้อผิดพลาดต่างๆ ดังนั้นการแบ่งจาวาออกเป็นเอดิชันจึงเสมือนการจัดกลุ่มของคลาสและแพ็คเกจเท่าที่จำเป็นจะต้องใช้ในเอดิชันนั้นๆ การพัฒนาคลาสหรือแพ็คเกจของแต่ละเอดิชันก็จะเป็นอิสระจากกันด้วย

2.1.1 สถาปัตยกรรมของ J2ME

แนวคิดในเรื่องโครงสร้างหรือสถาปัตยกรรมของ J2ME ได้แบ่งออกเป็น 2 ส่วน คือ คอนฟิกกูเรชัน (Configuration) และโพรไฟล์ (Profile) เพื่อให้มีความยืดหยุ่นในการนำไปใช้งานกับอุปกรณ์ต่างๆ กัน โดยคอนฟิกกูเรชันเป็นกลุ่มของ API ระดับต่ำ (Low-level API) ทำหน้าที่เป็นตัวกำหนดคุณลักษณะในระหว่างการรันแอพพลิเคชั่น ส่วนโพรไฟล์จะดูเกี่ยวกับคุณลักษณะของอุปกรณ์ที่จะรัน J2ME เช่น การแสดงส่วนติดต่อกับผู้ใช้ของอุปกรณ์ วิธีการจัดเก็บข้อมูล เป็นต้น โครงสร้างของ J2ME เป็นดังรูปที่ 2.1



รูปที่ 2.1 โครงสร้างของ J2ME

สภาวะแวดล้อมของ J2ME ประกอบด้วยโพรไฟล์อย่างน้อย 1 ตัว คอนฟิกกูเรชัน และ Virtual Machine (VM) โดย Virtual Machine จะทำหน้าที่เป็นตัวกลางระหว่างคอนฟิกกูเรชันและระบบปฏิบัติการ ส่วนคอนฟิกกูเรชันจะเป็นตัวกลางระหว่างโพรไฟล์กับ Virtual Machine และโพรไฟล์จะเป็นตัวกลางระหว่างแอพพลิเคชั่นกับสภาวะแวดล้อมของ J2ME

2.1.1.1 คอนฟิกกูเรชัน (Configuration)

คอนฟิกกูเรชันของ J2ME ถูกกำหนดโดยกลุ่ม JCP และบริษัทต่างๆ ที่พัฒนาอุปกรณ์ที่สนับสนุน J2ME ซึ่งเป็นกลุ่มของ API ระดับต่ำ (Low-level API) ทำหน้าที่เป็นตัวกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณลักษณะในระหว่างการรันแอพพลิเคชั่น ปัจจุบันมีคอนฟิกกูเรชัน 2 ตัว โดยจัดกลุ่มโดยแบ่งกลุ่มตามอุปกรณ์ที่มีความคล้ายกันในเรื่องของขนาดความจำ และ กำลังของหน่วยประมวลผล คือ CLDC (Connected Limited Device Configuration) และ CDC (Connected Device Configuration) แต่ละตัวใช้ VM (Virtual Machine) ต่างกัน

1.) CLDC (Connected Limit Device Configuration) เป็นคอนฟิกกูเรชันสำหรับอุปกรณ์ที่มีหน่วยความจำน้อย มีกำลังของหน่วยประมวลผลต่ำ กินไฟน้อยและมีข้อจำกัดในด้านการเชื่อมต่อเครือข่าย โดยคอนฟิกกูเรชัน นี้จะใช้ Kilobyte Virtual Machine (KVM) เป็น Virtual Machine ซึ่งอุปกรณ์ที่จะใช้คอนฟิกกูเรชันนี้ได้แก่ โทรศัพท์มือถือ PDAs เป็นต้น

ตัวอย่างคุณลักษณะของอุปกรณ์ที่จะใช้คอนฟิกกูเรชัน CLDC คือ

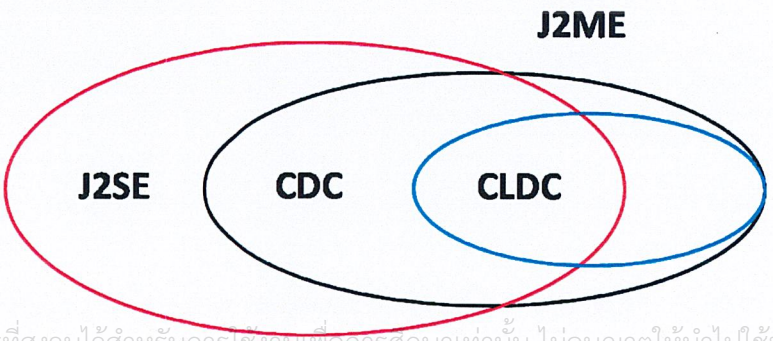
- ใช้หน่วยความจำ 128 – 512 กิโลไบต์ สำหรับใช้งานจาวา
- ใช้หน่วยความจำ 32 กิโลไบต์ ขณะรันไทม์
- ใช้พลังงานหรือแหล่งจ่ายไฟน้อย ซึ่งส่วนมากจะเป็นแบตเตอรี่
- เชื่อมต่อสัญญาณด้วยแบนด์วิธต่ำ

2.) CDC (Connect Device Configuration) เป็นคอนฟิกกูเรชันสำหรับอุปกรณ์ที่มีความสามารถสูงกว่า CLDC โดยสนับสนุนอุปกรณ์ที่มีหน่วยความจำตั้งแต่ 512 กิโลไบต์จนถึง 2 เมกกะไบต์ โดยใช้ C-Virtual Machine (CVM) เป็น Virtual Machine ซึ่งอุปกรณ์ที่ใช้คอนฟิกกูเรชันนี้ได้แก่ Pocket PC และอุปกรณ์จำพวก set-top box ของโทรทัศน์ เป็นต้น

ตัวอย่างคุณลักษณะของอุปกรณ์ที่จะใช้คอนฟิกกูเรชัน CLDC คือ

- ใช้หน่วยความจำอย่างต่ำ 512 กิโลไบต์ สำหรับใช้งานจาวา
- ใช้หน่วยความจำอย่างต่ำ 256 กิโลไบต์ ขณะรันไทม์
- ใช้พลังงานมาก
- เชื่อมต่อสัญญาณด้วยแบนด์วิธสูง

ความสัมพันธ์ของ CLDC กับ CDC เป็นดังรูปที่ 2.2 เห็นได้ว่ามีบางส่วนของ J2ME ไม่ได้อยู่ใน J2SE บางส่วนที่ว่านี้คือ คลาส แพ็คเกจ หรือ API ที่เกี่ยวข้องกับการทำงาน และการติดต่อกับอุปกรณ์ที่สนับสนุน J2ME นั้นเอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 ความสัมพันธ์ของ CLDC กับ CDC

2.1.1.2 โพรไฟล์ (Profiles)

เป็นข้อกำหนดทางด้านคุณลักษณะของอุปกรณ์ที่จะรัน J2ME ซึ่งอาจจะเป็นโพรไฟล์ที่เกี่ยวข้องกับคุณลักษณะทางด้านฮาร์ดแวร์ของอุปกรณ์แต่ละตัวหรืออุปกรณ์หนึ่งชิ้นโดยเฉพาะ โพรไฟล์ คือส่วนเพิ่มเติมจากคอนฟิกกูเรชัน เป็นกลุ่มคลาสไลบรารีที่รองรับข้อแตกต่างของอุปกรณ์ เช่น โพรไฟล์ของอุปกรณ์ประเภทมือถือคือ MIDP (Mobile Information Device Profile) เป็นกลุ่มคลาสไลบรารีต่างๆ ที่รองรับการพัฒนาแอพพลิเคชั่นบนอุปกรณ์มือถือ ประกอบด้วย UI (User Interface), Input/Output, Event Handling, Persistent Storage, Networking และ Timers ตัวอย่างข้อกำหนดของโพรไฟล์ เช่น อุปกรณ์มีอินเทอร์เฟซติดต่อกับผู้ใช้อย่างไร อุปกรณ์จะติดต่อกับเครือข่ายอย่างไรหรือเก็บข้อมูลไว้อย่างไร เป็นต้น ดังนั้นในการพัฒนาแอพพลิเคชั่น J2ME จำเป็นจะต้องเลือกโพรไฟล์อย่างน้อย 1 ตัว เพื่อเป็นข้อกำหนดแอพพลิเคชั่นที่พัฒนาขึ้นว่าสามารถจะนำไปรันกับอุปกรณ์ใดได้บ้าง โพรไฟล์ที่ได้รับความนิยมมากที่สุด คือ MIDP การพัฒนาแอพพลิเคชั่นด้วยจาวานั้น สำหรับ J2ME แล้วแอพพลิเคชั่นที่ได้ออกมาจะเรียกว่า MIDlet ซึ่งจะนำไปรันบนอุปกรณ์ต่างๆที่สนับสนุนเทคโนโลยี J2ME

2.1.2 MIDlet

MIDlet หมายถึง โปรแกรมหรือแอพพลิเคชั่นที่พัฒนาบนเทคโนโลยี J2ME สำหรับใช้กับอุปกรณ์ MIDP Device ซึ่ง MIDlet แต่ละตัวประกอบด้วยไฟล์ 2 ไฟล์ คือ ไฟล์นามสกุล .jad และ .jar โดย MIDlet นี้เทียบเคียงได้กับแอพเพลตที่เป็นแอพพลิเคชั่นที่ได้จาก J2SE ที่รันบนบราวเซอร์ เนื่องจาก MIDlet เป็นแอพพลิเคชั่นขั้นสุดท้าย ที่นำไปรันในอุปกรณ์ที่สนับสนุนเทคโนโลยี J2ME การทำความเข้าใจการทำงานของไฟล์ .jad และ .jar จึงเป็นสิ่งจำเป็นสำหรับผู้พัฒนา MIDlet

MIDP-1.0 คือ โพรไฟล์สำหรับพัฒนาแอพพลิเคชั่นบนอุปกรณ์ประเภทมือถือ ประกอบด้วยแพ็คเกจ 7 แพ็คเกจดังตารางที่ 2.1

ตารางที่ 2.1 แพ็คเกจของโพรไฟล์ MIDP-1.0

แพ็คเกจ	คำอธิบาย
java.io	กลุ่มคลาสบางส่วนที่มาจาก J2SE สำหรับรับส่งข้อมูล Input/Output
java.lang	กลุ่มคลาสบางส่วนที่มาจาก J2SE ที่เกี่ยวกับภาษาจาวา
java.util	กลุ่มคลาสบางส่วนที่มาจาก J2SE ประกอบด้วยคลาสยูทิลิตี้ต่างๆ
javax.microedition.io	กลุ่มคลาสสำหรับรับส่งข้อมูลผ่านระบบเนตเวิร์ก
javax.microedition.lcdui	กลุ่มคลาสของคอมโพเนนต์ต่างๆ
javax.microedition.MIDlet	กลุ่มคลาสสำหรับแอพพลิเคชั่นของ J2ME
javax.microedition.rms	กลุ่มคลาสสำหรับจัดการข้อมูลบนหน่วยความจำถาวรของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIDP-2.0 คือโพรไฟล์ที่ปรับปรุงมาจาก MIDP-1.0 สามารถช่วยให้การพัฒนาแอพพลิเคชั่นทำได้ง่ายขึ้น โดยมีแพ็คเกจเพิ่มเติมจาก MIDP-1.0 ดังนี้

ตารางที่ 2.2 แพ็คเกจของไพรไฟล์ MIDP-2.0

แพ็คเกจ	คำอธิบาย
javax.microedition.lcdui.game	กลุ่มคลาสสำหรับการพัฒนาเกม ที่เรียกว่า Game API
javax.microedition.midia	กลุ่มคลาสสำหรับงานด้านมัลติมีเดีย
javax.microedition.media.control	กลุ่มคลาสสำหรับงานด้านมัลติมีเดียด้านเสียง
javax.microedition.pki	กลุ่มคลาสที่สนับสนุนการสร้างความน่าเชื่อถือของข้อมูล

2.1.3 ความสำคัญของไฟล์ .jar และ .jad

ไฟล์ .jar เป็นที่เก็บคลาสไฟล์ต่างๆ (นามสกุล .class) ของ MIDlet สามารถใช้โปรแกรมวินแรร์ (WinRAR) เปิดดูคลาสไฟล์ที่อยู่ข้างในได้ ส่วนไฟล์ .jad เป็นไฟล์ข้อความที่ใช้อธิบายไฟล์ .jar ว่า ชื่ออะไร เวอร์ชันอะไร หรือมีขนาดไฟล์เท่าไหร่ เป็นต้น สามารถใช้โปรแกรมโน้ตแพด (Notepad) เปิดดูไฟล์ข้อมูลในไฟล์นี้ได้ ในการพัฒนา MIDlet จะต้องสร้างไฟล์ .jar ขึ้นมาก่อน จากนั้นเอาข้อมูลของไฟล์ .jar ไปสร้างไฟล์ .jad และเมื่อจะเผยแพร่ MIDlet ต้องให้ไฟล์ทั้งสองไฟล์คู่กันไป เพราะในการรัน MIDlet นั้นตัวซอฟต์แวร์จัดการแอพพลิเคชั่นจะอ่านข้อมูลจากไฟล์ .jad ก่อน แล้วจึงไปเรียกไฟล์ .jar มารัน หากขาดไฟล์ใดไฟล์หนึ่งจะไม่สามารถรัน MIDlet นั้นได้ MIDlet ที่ถูกสร้างขึ้นจะต้องนำไฟล์ .jad และ .jar มารวมกันทำเป็นแพ็คเกจ (packaging) เพื่อสามารถนำไปแจกจ่ายใช้งาน ซึ่งผลลัพธ์ของการทำ packaging จะเรียกว่า MIDlet suite ประกอบด้วย file 2 file ต่อไปนี้

1.) JAR file (Java Archive) ประกอบด้วยคลาสไฟล์ทั้งหมดของ MIDlet application ที่ต้องถูกทำการตรวจสอบก่อน (preverify) รวมไปถึง resource file และ manifest file ซึ่ง resource file ประกอบด้วยตัวอักษร ไฟล์รูปภาพ ไฟล์เสียง ที่ถูกใช้โดย MIDlet ตอนรัน แต่ละ MIDlet ที่อยู่ใน MIDlet suite เดียวกันสามารถใช้คลาสไฟล์ร่วมกันได้ ถ้าสร้าง MIDlet หลายตัวแล้วทำเป็นแพ็คเกจเดียวกัน จะประหยัดขนาดแพ็คเกจที่ต้องดาวน์โหลดด้วย และถ้าต้องใช้ Third party class library ภายใน MIDlet เช่น library สำหรับการคำนวณเลขทศนิยม (floating point) เป็นต้น ต้อง include ตัว Third party class file ลงใน JAR file หากอุปกรณ์ที่ใช้งานไม่มีคลาสไฟล์นี้ไว้ในอุปกรณ์อยู่แล้วจากผู้ผลิต สามารถแพ็คเกจ MIDlet ลงใน JAR file โดยใช้คำสั่งดังนี้

" jar cvmf MANIFEST.MF HelloMIDlet.jar HelloMIDlet.class icon.pn "

manifest file เป็น file หนึ่งที่ต้องถูกแพ็คเกจอยู่ใน JAR file ของ MIDlet Suite มีไว้เพื่อเป็นตัวบอกว่าใน JAR file นั้นๆ มี content อะไรบ้างและประกอบด้วยข้อมูลอื่นของ MIDlet Suite ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIDlet-Name	--> ชื่อของ MIDlet
MIDlet-Version	--> เวอร์ชันของ MIDlet Suite
MIDlet-Vendor	--> ชื่อของผู้พัฒนาหรือผู้จัดจำหน่าย MIDlet
MicroEdition-Profile	--> เวอร์ชันของ MIDP ที่ MIDlet ต้องการหรือทำงานได้
MicroEdition-Configuration	--> เวอร์ชันของ CLDC ที่ MIDlet ต้องการหรือทำงานได้
MIDlet-<n>	--> อธิบาย MIDlet แต่ละตัวใน MIDlet Suite โดย n เป็นหมายเลข ค่าที่ระบุมี 3 ส่วน ได้แก่ ชื่อของ MIDlet , ชื่อไฟล์ไอคอนของ MIDlet และชื่อคลาสไฟล์ที่จะโหลดมาทำงาน (สาเหตุที่มีแอตทริบิวต์นี้เพราะว่าใน MIDlet 1 ตัวสามารถมี MIDlet ย่อยได้หลายตัว)

2.) JAD file (Java Application Descriptor) เป็น file ที่ทำหน้าที่เป็น Application Descriptor ลักษณะของไฟล์จะคล้าย manifest file ที่อยู่ใน .jar file แต่จุดประสงค์ต่างกัน โดย manifest file จะใช้สำหรับ packaging MIDlet suite แต่ .jad file จะใช้ตอน deploy MIDlet ลงอุปกรณ์ โดยเฉพาะทำ OTA deployment process ซึ่ง JAD file จะประกอบด้วยแอตทริบิวต์หลักดังนี้

MIDlet-Name	--> ชื่อของ MIDlet
MIDlet-Version	--> เวอร์ชันของ MIDlet Suite
MIDlet-Vendor	--> ชื่อของผู้พัฒนาหรือผู้จัดจำหน่าย MIDlet
MIDlet-Jar-URL	--> URL สำหรับดาวน์โหลดไฟล์ .jar
MIDlet-Jar-Size	--> ขนาดของไฟล์ .jar (หน่วยเป็นไบต์)
MicroEdition-Profile	--> เวอร์ชันของ MIDP ที่ MIDlet ต้องการหรือทำงานได้
MicroEdition-Configuration	--> เวอร์ชันของ CLDC ที่ MIDlet ต้องการหรือทำงานได้
MIDlet-<n>	--> อธิบาย MIDlet แต่ละตัวใน MIDlet Suite ค่าที่ระบุมี 3 ส่วน ได้แก่ ชื่อ MIDlet, ชื่อไฟล์ไอคอน MIDlet, ชื่อคลาสไฟล์ที่จะโหลดมาทำงาน

นอกจากนี้ใน manifest.mf และ JAD ยังมีแอตทริบิวต์เสริมอื่นๆ ที่สามารถเขียนเพิ่มลงไปได้อีก ข้อควรทราบสำหรับการเขียนแอตทริบิวต์ มีดังนี้

- ชื่อแอตทริบิวต์เป็นแบบตัวอักษรพิมพ์ใหญ่ - พิมพ์เล็กไม่เหมือนกัน (case sensitive)
- ค่าของแอตทริบิวต์ MIDlet-Name , MIDlet-Version และ MIDlet-Vendor ใน manifest.mf และ JAD ของ MIDlet Suite เดียวกันจะต้องเขียนให้เหมือนกัน
- แอตทริบิวต์ที่ผู้ใช้กำหนดขึ้นเอง จะตั้งชื่ออะไรก็ได้ แต่ห้ามขึ้นต้นด้วยคำว่า MIDlet เพราะคำนี้เป็นของแอตทริบิวต์บังคับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่เป็นแอตทริบิวต์เสริมในไฟล์ manifest.mf และ JAD ได้แก่

MIDlet-Description --> คำอธิบายหรือรายละเอียดของ MIDlet Suite

MIDlet-Icon --> ชื่อไฟล์รูป PGN ในไฟล์ .jar ที่จะใช้เป็นไอคอนของ MIDlet Suit

MIDlet-Info-URL --> URL ที่มีข้อมูลอธิบายเพิ่มเติมของ MIDlet Suit

MIDlet-Data-Size --> ขนาดพื้นที่หรือหน่วยความจำต่ำสุด

<User-Defined Attribute> --> แอตทริบิวต์ที่ผู้ใช้กำหนดขึ้นเอง

ก่อนที่ MIDlet suite ถูกดาวน์โหลดลงอุปกรณ์จะมีซอฟท์แวร์จัดการแอพพลิเคชั่น (Application Management Software) คอยตรวจสอบค่าแอตทริบิวต์ใน .jad file 5 ค่าแรก เพื่อดูว่าเหมาะสมกับอุปกรณ์นั้นหรือไม่ ตัวอย่างเช่น ถ้าอุปกรณ์ไม่รองรับเวอร์ชันของโพรไฟล์ซึ่งอยู่ในค่าของ MicroEdition-Profile ตัว jar file จะไม่ถูกดาวน์โหลดลงมา หรือ ถ้าขนาดของ jar file ที่อยู่ใน MIDlet-Jar-Size ใหญ่เกินกว่าที่อุปกรณ์รองรับได้ก็จะไม่ถูกดาวน์โหลดมาเช่นกัน

2.1.4 MIDlet Suit

MIDlet Application เมื่อถูกสร้างขึ้นจะต้องนำมาแพ็คเกจรวมกันเป็น JAR file จำนวน 1 file โดยสามารถมี MIDlet Application หลายตัวใน JAR file เดียวกันได้ ซึ่งเรียกกลุ่มของ MIDlet Application นี้ว่า MIDlet Suite หลังจากทำ Packaging แล้วก็จะสามารถดาวน์โหลดและติดตั้งลงในอุปกรณ์ได้โดยอาจจะผ่านทางสายอนุกรม (Serial cable) ต่อโดยตรงจากคอมพิวเตอร์หรือจาก wireless network (OTA : Over-The-Air) เมื่อติดตั้ง jar file บนอุปกรณ์แล้วจะปรากฏเมนูสำหรับแต่ละ MIDlet ที่อยู่ใน MIDlet Suite นั้นๆ ทำให้สามารถเลือกที่จะใช้งาน MIDlet แต่ละตัวแยกกันได้

2.1.5 การทำงานของ MIDlet

MIDlet เป็น class ที่อยู่ในแพ็คเกจของ java.microedition.MIDlet และเป็นโปรแกรมที่พัฒนาขึ้นจาก MIDP และสืบทอดจาก MIDlet คลาส และเรียกใช้ (Implements) 3 เมธอด คือ startApp() , pauseApp() , destroyApp() นอกจากนี้ MIDlet ยังกำหนด constructor แบบ public ที่ไม่มีอาร์กิวเมนท์ใดๆ อีกด้วย Method พื้นฐานที่ต้องมีในทุก MIDlet มีดังนี้

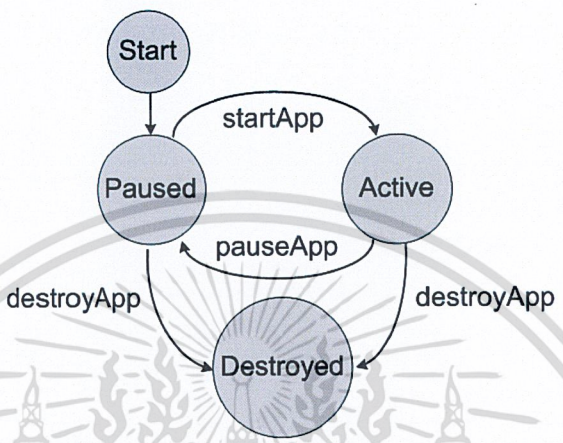
ตารางที่ 2.3 Method พื้นฐานของ MIDlet

Method Name	จุดประสงค์การใช้งาน
StartApp()	ใช้สำหรับ allocate system resources และ initialize application.
PauseApp()	ใช้ suspend MIDlet ชั่วคราว
DestroyApp()	ใช้ปล่อยการใช้งาน resources ซึ่งถูกใช้โดย MIDlet และเพื่อกำจัด MIDlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 วงจรการทำงานของ MIDlet

กระบวนการทำงานของ MIDlet ประกอบไปด้วย 3 สถานะ คือ กำลังทำงาน หยุดชั่วคราว และถูกทำลาย อาศัยซอฟต์แวร์จัดการแอพพลิเคชั่น เป็นตัวควบคุมการเปลี่ยนสถานะหนึ่งไปยังอีกสถานะหนึ่งควบคุมด้วยเมธอด startApp() pauseApp() และ destroyApp() ที่มาพร้อมกับ MIDlet จากรูปที่ 2.3 แสดงให้เห็นถึงจุดเปลี่ยนระหว่างสถานะทั้ง 3 โดยการเรียกใช้เมธอดข้างต้น



รูปที่ 2.3 วงจรการทำงานของ MIDlet

เมื่อ MIDlet พร้อมทำงาน ซอฟต์แวร์จัดการแอพพลิเคชั่นจะสร้างตัวอย่าง MIDlet ขึ้นมาก่อน โดยใช้ constructor แบบ public ที่ไม่มีอาร์กิวเมนต์ใดๆ โดย MIDlet จะอยู่ในสถานะหยุดชั่วคราว จากนั้นซอฟต์แวร์จัดการแอพพลิเคชั่นจะเรียกเมธอด startApp() ขึ้นมา และ MIDlet จะเข้าสู่สถานะกำลังทำงานเปิดรับทรัพยากรที่ต้องการและเริ่มต้นการทำงาน ในสถานะนี้ MIDlet จะทำงานและดึงทรัพยากรที่ต้องการไว้ใช้งาน ถ้าซอฟต์แวร์จัดการแอพพลิเคชั่นไม่ต้องการให้ MIDlet ทำงานต่อไป จะเรียกเมธอด pauseApp() จากนั้น MIDlet จะหยุดทำงานและเข้าสู่สถานะหยุดชั่วคราว คืนทรัพยากรที่ดึงมาใช้งานและเข้าสู่สถานะไม่ทำงาน MIDlet และสามารถกลับไปอยู่ที่สถานะกำลังทำงานได้ เมื่อซอฟต์แวร์จัดการแอพพลิเคชั่นเรียกเมธอด startApp() ขึ้นมา ท้ายที่สุดเมื่อซอฟต์แวร์จัดการแอพพลิเคชั่นไม่ต้องการเรียกใช้งาน MIDlet หรือต้องการเคลียร์หน่วยความจำเพื่อให้โปรแกรมอื่นได้ใช้งาน ก็จะส่งสัญญาณเตรียมทำลาย MIDlet ทิ้งโดยการเรียกเมธอด destroyApp() และเข้าสู่สถานะถูกทำลาย MIDlet จะปล่อยทรัพยากรทั้งหมด ทำการจัดเก็บข้อมูลถาวรต่างๆ และหยุดการทำงานทั้งหมด MIDlet จะเข้าสู่สถานะหยุดทำงานก็ต่อเมื่อผ่านสถานะกำลังทำงานมาก่อนแล้ว ในทางตรงกันข้ามหากต้องการเข้าสู่สถานะถูกทำลาย สามารถเข้าได้โดยตรงจากทั้งสถานะหยุดชั่วคราวและสถานะกำลังทำงาน นอกจากนี้ยังสามารถเข้าสู่สถานะหยุดทำงานได้ในขณะที่กำลังทำงาน หรือเมื่อได้รับคำสั่งจากซอฟท์แวร์จัดการแอพพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.7 ซอฟท์แวร์จัดการแอพพลิเคชั่น

ซอฟท์แวร์จัดการแอพพลิเคชั่น เป็นซอฟท์แวร์ที่มาพร้อมกับ MIDP ทำหน้าที่ควบคุมการติดตั้ง สั่งกระทำการและลบ MIDlet บางครั้งเรียกซอฟท์แวร์จัดการแอพพลิเคชั่นว่า ซอฟท์แวร์จัดการ MIDlet (MIDlet Management Software) หรือ โปรแกรมบริหารแอพพลิเคชั่นจาวา (JAM : Java Application Manager)

การติดตั้งซอฟท์แวร์จัดการแอพพลิเคชั่นแตกต่างกันออกไปตามแต่ประเภทของอุปกรณ์ แต่ว่ามีการให้บริการขั้นพื้นฐานเหมือนกัน ได้แก่

1.) ยอมให้ผู้ใช้งานติดตั้งและถอดถอน MIDlet จากอุปกรณ์ไร้สาย ทั้งผ่านสายเคเบิลที่เชื่อมต่อเข้ากับเครื่องคอมพิวเตอร์ส่วนบุคคล หรือจากระยะไกลผ่านอินเตอร์เน็ต

2.) เตรียม environment ที่สั่งกระทำการให้แก่ MIDlet หลังจาก MIDlet เริ่มทำงาน ซอฟท์แวร์จัดการแอพพลิเคชั่นจะจัดเตรียมทรัพยากรของระบบ เช่น คลาสของ CLDC และ MIDP และ KVM ไว้ให้ MIDlet ได้ และใช้เป็นรันไทม์ในการทำงานอีกด้วย

3.) ซอฟท์แวร์จัดการแอพพลิเคชั่นยังจัดการแก้ไขข้อผิดพลาดทั้งหมดที่เกิดขึ้นระหว่างการติดตั้ง และสั่งกระทำการแอพพลิเคชั่นโดยไม่ทำให้ระบบล่มอีกด้วย

2.2 เทคโนโลยีบลูทูธ

ปัจจุบันการรับ-ส่งข้อมูลชนิดต่างๆ จากที่หนึ่งไปยังอีกที่หนึ่งสามารถทำได้ในระยะเวลาอันสั้น และยังคงความถูกต้องของข้อมูลไว้ได้ ในการส่งข้อมูลดิจิตอลก็เช่นกัน สามารถส่งผ่านตัวกลางได้หลายชนิด ตัวกลางที่ใช้งานง่ายและถูกที่สุดคือ สายส่งสัญญาณที่มีตัวนำไฟฟ้าเป็นสื่อกลางหรือที่เรียกว่า สายเคเบิล การส่งข้อมูลผ่านสายเคเบิลนั้นถูกนำมาใช้งานกันอย่างแพร่หลาย เพราะไม่ต้องเปลี่ยนสัญญาณข้อมูลไปอยู่ในรูปแบบอื่นก่อน อีกทั้งยังสามารถรองรับความเร็วในการส่งสัญญาณได้สูง แต่การเชื่อมต่อสายเคเบิลระหว่างอุปกรณ์มากกว่าสองชิ้นเข้าด้วยกัน มีความยุ่งยาก ไม่สะดวกในการใช้งานอุปกรณ์ที่ไม่ได้ติดตั้งอยู่กับที่ จึงมีแนวคิดหาสื่อกลางชนิดอื่นเข้ามาแทนสายเคเบิล และปัจจุบันมีการส่งข้อมูลมากขึ้น จึงทำให้การส่งข้อมูลนั้นจำเป็นต้องใช้ความเร็วที่เพิ่มมากขึ้น ดังนั้นสื่อกลางที่จะนำมาทำหน้าที่แทนนั้นต้องสามารถรองรับความเร็วในการส่งข้อมูลที่สูงนี้ได้ด้วย ตัวกลางที่ถูกนำมาใช้แทนสายเคเบิลตัวแรกคือ แสงอินฟราเรด แต่การใช้แสงอินฟราเรดเป็นตัวกลางนั้นสามารถถูกรบกวนจากแสงภายนอกได้ง่าย อีกทั้งตำแหน่งของอุปกรณ์รับส่งทั้งสองจะต้องอยู่ในแนวเส้นตรงเดียวกัน และไม่มีอะไรมาตัดขวางลำแสงจึงจะสามารถใช้งานได้ จากข้อจำกัดดังกล่าวทำให้เกิดความคิดที่จะนำคลื่นวิทยุมาใช้แทน เพราะคลื่นวิทยุสามารถรับส่งข้อมูลที่ความเร็วสูง สามารถทะลุวัสดุต่างๆได้และไม่จำเป็นต้องให้อุปกรณ์รับส่งอยู่ในแนวเส้นตรงเดียวกันในขณะใช้งาน ในปัจจุบันนี้การนำคลื่นวิทยุมาใช้รับส่งข้อมูลสำหรับอุปกรณ์ระยะใกล้ ได้กำหนดเป็นมาตรฐานอยู่ 3 ระบบด้วยกันคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.) IEEE 802.11 หรือ Wireless LAN (WLAN) ออกแบบมาเพื่อสร้างระบบเนตเวิร์กไร้สายทดแทนการใช้สาย UTP หรือโคแอกเชียลที่เชื่อมต่ออุปกรณ์ต่างๆเข้ากับระบบ LAN แบบ Ethernet

2.) HomeRF เป็นมาตรฐานที่กำหนดขึ้นเพื่อทดแทนการเชื่อมต่อเครื่องใช้ไฟฟ้าที่อยู่ภายในบ้าน ซึ่งรูปแบบของข้อมูลที่รับส่งนั้นมีทั้งภาพและเสียงรวมกันอยู่ จึงต้องการความเร็วในการส่งข้อมูลที่สูงมาก

3.) บลูทูธ เป็นมาตรฐานที่กำหนดขึ้น เพื่อทดแทนการใช้สายเชื่อมต่อระหว่างอุปกรณ์ในระยะใกล้ เช่น การเชื่อมต่อระหว่างโทรศัพท์มือถือกับหูฟังสมอลทอร์ก เป็นต้น

2.2.1 ความเป็นมาของเทคโนโลยีบลูทูธ

ในปี 1994 บริษัท Ericsson Mobile Communication ได้เริ่มศึกษาสิ่งที่มาทดแทนสายเคเบิลสำหรับการรับส่งข้อมูลระหว่างโทรศัพท์มือถือโดยในการศึกษาจะมุ่งไปที่การใช้งานสัญญาณวิทยุซึ่งดีกว่าการเชื่อมต่อแบบอินฟาเรด และในการศึกษาไม่ได้มุ่งศึกษาเฉพาะการส่งข้อมูลเพียงอย่างเดียว แต่ยังศึกษาถึงการส่งข้อมูลที่เป็นเสียงด้วยเพื่อใช้สำหรับ Headset ของโทรศัพท์มือถือ

Bluetooth Special Interest Group (SIG) คือ กลุ่มของบริษัทที่ร่วมกันสนับสนุนเทคโนโลยีบลูทูธ และเป็นกลุ่มที่วางแนวทางมาตรฐานของบลูทูธทั้งหมด ก่อตั้งในปี 1998 ซึ่งมี 5 บริษัทเป็นสมาชิก ได้แก่

1.) Ericsson Mobile Communication AB.

2.) Intel Corp.

3.) IBM Corp.

4.) Toshiba Corp.

5.) Nokia Mobile Phones.

ในเดือนพฤษภาคมปี 1998 บริษัทเหล่านี้ได้ประกาศการรวมตัวกันเพื่อก่อตั้ง SIG ต่อมาในเดือน กรกฎาคมของปี 1999 กลุ่มบริษัทเหล่านี้ได้ทำการเผยแพร่ Bluetooth Specification Version 1.0 ที่ website http://www.bluetooth.com นอกจากนี้ก็ยังมีการเปิดรับสมาชิกของกลุ่มเพิ่มขึ้นเรื่อยๆ จนปัจจุบันนี้มีสมาชิกอยู่มากกว่า 1800 บริษัท Bluetooth SIG ได้ทำการแบ่งกลุ่มการพัฒนาออกเป็น 3 กลุ่มโดย

- กลุ่มที่ 1 ทำการแก้ไข Bluetooth version 1.0
- กลุ่มที่ 2 ทำการพัฒนาเพิ่มเติม Bluetooth version 1.0
- กลุ่มที่ 3 พัฒนา Bluetooth version 2.0

ในส่วนชื่อของบลูทูธนั้นได้มาจากชื่อของ Harald Blatant กษัตริย์ของประเทศเดนมาร์ก ซึ่งในช่วงที่ครองราชย์อยู่นั้นสามารถรวมเดนมาร์กและนอร์เวย์เข้าด้วยกัน เปรียบเสมือนเทคโนโลยีบลูทูธที่จะรวมอุตสาหกรรมด้านการสื่อสารและคอมพิวเตอร์เข้าด้วยกัน จากชื่อ Blatant

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาการออกเสียงจากภาษาดั้งเดิมโดยผิดเพี้ยนน้อยที่สุด จึงเรียกชื่อว่า Bluetooth และสาเหตุที่เลือกใช้ชื่อนี้เพราะว่า Ericsson ซึ่งเป็นผู้ริเริ่มเทคโนโลยีนี้มีบริษัทแม่อยู่ที่เดนมาร์กนั่นเอง

มาตรฐานอุปกรณ์ เข้าร่วมในการกำหนดคุณสมบัติของอุปกรณ์ โดยเวอร์ชันของอุปกรณ์เวอร์ชันแรกๆ มีข้อจำกัดคือไม่สามารถใช้งานแอพพลิเคชั่นพร้อมกันบนอุปกรณ์บลูทูธเดียวกันได้ และได้รับการแก้ไขต่อมา โดยเน้นที่การตัดการรบกวนสัญญาณและใช้พลังงานอย่างประหยัด เพื่อให้การรับส่งมีความเร็ว และรัศมีในการรับส่งข้อมูลเพิ่มขึ้น ปัจจุบันได้การรับรองมาตรฐานโดย IEEE 8021.15 และมีข้อกำหนดบลูทูธออกมาแล้ว 7 รุ่น คือ Bluetooth 1.0, Bluetooth 1.0B, Bluetooth 1.1, Bluetooth 1.2, Bluetooth 2.0, Bluetooth 2.0 EDR, Bluetooth 2.1 EDR, Bluetooth 3.0 + HS และ Bluetooth 4.0

2.2.2 จุดมุ่งหมายของบลูทูธ

บลูทูธเป็นมาตรฐานเปิด (Open Specification) ที่ออกแบบมาเพื่อใช้ส่งข้อมูลไร้สายระยะใกล้ (Short Range Wireless) ซึ่งถูกกำหนดขึ้นโดย Bluetooth SIG ซึ่งมีจุดมุ่งหมายหลักในการทำงานมาตรฐานหลักดังนี้

1.) เป็นมาตรฐานเปิดที่ทุกคนสามารถใช้ข้อมูลที่ส่งผ่านบลูทูธนี้ได้ โดยไม่ต้องเสียค่าธรรมเนียมใดๆ ทั้งสิ้น เพื่อให้เกิดความแพร่หลายในการใช้งานและพัฒนาระบบได้อย่างรวดเร็ว

2.) รับส่งข้อมูลไร้สายในระยะใกล้ คือ อุปกรณ์ต่างๆ สามารถส่งข้อมูลได้โดยไม่ต้องใช้สาย

3.) สามารถรองรับการรับส่งเสียงและข้อมูลได้ในเวลาเดียวกันนั่นคือ ระบบจะต้องมีความเร็วในการรับส่งข้อมูลเพียงพอสำหรับการส่งเสียงและข้อมูลไปพร้อมๆกัน

4.) สามารถใช้งานได้ในทุกที่ทั่วโลก หมายความว่า อุปกรณ์ที่ผลิตตามมาตรฐานของบลูทูธไม่ว่าจะผลิตจากผู้ผลิตใดหรืออยู่ ณ ตำแหน่งใดบนโลกสามารถใช้งานร่วมกันได้ จากจุดมุ่งหมายในข้อนี้ทำให้ต้องใช้ความถี่คลื่นวิทยุที่สามารถใช้งานได้ในทุกประเทศและมีการกำหนดสัญลักษณ์บลูทูธขึ้นได้ดังรูป

รูปที่ 2.4 สัญลักษณ์บลูทูธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 วัตถุประสงค์ของบลูทูธ

สาเหตุที่เทคโนโลยีบลูทูธเป็นที่สนใจสำหรับกลุ่มบริษัทต่างๆ คือ ช่วยทำให้การเชื่อมต่อระหว่างโทรศัพท์มือถือกับอุปกรณ์ต่างๆ สามารถทำได้สะดวกมากยิ่งขึ้น และบริษัทต่างๆ สามารถทำกำไรจากเทคโนโลยีนี้ได้ โดยการขายผลิตภัณฑ์ที่ใช้เทคโนโลยีบลูทูธ รวมไปถึงการขายซอฟต์แวร์สำหรับการใช้งานอุปกรณ์ที่บริษัทผลิตขึ้นมา วัตถุประสงค์แรกของเทคโนโลยีบลูทูธคือขายให้แก่บริษัทที่ผลิตโทรศัพท์มือถือ เนื่องจากเทคโนโลยีบลูทูธสามารถทำให้ผลิตภัณฑ์ของบริษัทเหล่านี้ใช้งานได้ดีขึ้น โดยเพิ่มขีดความสามารถของการติดต่อสื่อสารระหว่างโทรศัพท์มือถือกับอุปกรณ์ต่างๆ เนื่องจากในอดีตการสื่อสารนี้ทำได้โดยใช้สายเคเบิล ซึ่งมีความเสี่ยงที่ข้อมูลจะเกิดความเสียหายได้ ดังนั้นวัตถุประสงค์ต่อมาของเทคโนโลยีบลูทูธ คือ การแทนที่การสื่อสารระหว่างอุปกรณ์เคลื่อนที่ต่างๆ จากการใช้สายเคเบิลมาเป็นการใช้เทคโนโลยีบลูทูธ ซึ่งสามารถทำงานโดยใช้แบตเตอรี่ ใช้พลังงานน้อยและสามารถทำงานได้แม้ขณะที่มีแรงดันไฟฟ้าต่ำ ดังนั้นเทคโนโลยีบลูทูธจึงถูกนำไปใช้กับอุปกรณ์ขนาดเล็กอื่นๆ เช่น Headset และ PDAs

2.2.4 Protocol Stack

สำหรับอุปกรณ์ที่มีการเชื่อมต่อเข้าด้วยกัน การส่งข้อมูลจากอุปกรณ์ต้นทางไปยังอุปกรณ์ปลายทางจำเป็นต้องอาศัยการส่งข้อมูลอื่นๆ ประกอบเข้าไปกับข้อมูลที่ต้องการส่งนั้นด้วย เพื่อควบคุมเส้นทางของข้อมูลให้สามารถส่งไปยังอุปกรณ์ปลายทางได้อย่างถูกต้อง ทำให้การส่งข้อมูลแต่ละครั้งเกิดการทำงานในระดับต่างๆ ขึ้นมากมาย จึงเกิดการสร้างโมเดลแทนการทำงานนี้ขึ้นเพื่อให้มองเห็นภาพรวมของการทำงานทั้งหมดได้ง่ายขึ้น

การทำงานของอุปกรณ์บลูทูธจะใช้ข้อตกลงในการสื่อสารข้อมูลที่เรียกว่า โปรโตคอลสแต็ก (Protocol Stack) เป็นส่วนของโครงสร้างการทำงานของโปรแกรมที่มาควบคุม การทำงานของอุปกรณ์บลูทูธหรือ ที่เรียกกันว่า "ไดรเวอร์" จะเป็นตัวที่อนุญาตให้ application software ส่งและรับข้อมูลกับอุปกรณ์บลูทูธได้

สำหรับโมเดลการทำงานของบลูทูธ (Bluetooth Model) ถูกกำหนดให้มีโครงสร้างการทำงานดังตารางที่ 2.4 และจะเห็นได้ว่ามีจำนวน 8 ชั้นซึ่งมากกว่าโมเดล OSI (OSI model) อยู่ 1 ชั้น ทำให้ขอบเขตการทำงานในแต่ละชั้นแตกต่างจากโมเดล OSI แต่มีลำดับการทำงานเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 ตารางแสดงการเปรียบเทียบระหว่าง OSI Model และ Bluetooth Module

Application Layer	Applications
Presentation Layer	RFCOMM/SDP
Session Layer	L2CAP
Transport Layer	HCI
Network Layer	Link Manager
Data Link Layer	Link Controller
Physical Layer	Baseband
	Radio
OSI Model	**Bluetooth Module**

การทำงานของโมเดลบลูทูธในแต่ละชั้นมีหน้าที่ ดังนี้

ชั้นที่ 1 Radio เป็นส่วนที่เกิดการรับและการส่งคลื่นวิทยุจริง ซึ่งเป็นส่วนของวงจรฮาร์ดแวร์ การรับ - ส่งคลื่นวิทยุจะถูกควบคุมจากชั้น Baseband ไม่ว่าจะเป็นความถี่และระดับความแรงของสัญญาณที่ใช้ รวมไปถึงเฟรมข้อมูลที่จะส่ง

ชั้นที่ 2 Baseband การทำงานของชั้นนี้เป็นหัวใจของ Bluetooth ในด้านฮาร์ดแวร์ หน้าที่หลักของชั้นนี้คือ ควบคุมวงจรรับ - ส่งคลื่นวิทยุที่อยู่ชั้นล่างสุด โดยจุดสำคัญที่สุดของการควบคุมคือ เลือกช่องความถี่ในการรับ - ส่งข้อมูลให้ตรงกันระหว่าง Master และ Slave ที่ต้องกระโดดไปในลักษณะเดียวกัน

ชั้นที่ 3 Link Controller เป็นชั้นที่มีหน้าที่ควบคุมการเชื่อมต่อพื้นฐานของบลูทูธทั้งหมด ไม่ว่าจะเป็นสถานะของอุปกรณ์ โหมดการทำงานของอุปกรณ์ การค้นหาอุปกรณ์บลูทูธที่ใกล้เคียง รวมถึงการเลือกว่าเป็น Master หรือ Slave ในสภาพแวดล้อมต่างๆ

ชั้นที่ 4 Link Manager มีหน้าที่แปลงคำสั่งที่ได้รับจากชั้นบนมาเป็นลำดับการทำงานที่ชั้นล่างรู้จัก และคอยส่งคำสั่งไปควบคุมการทำงานทั้งหมดของชั้นล่าง โดย Link Manager เป็นส่วนที่ควบคุมจัดการลิงค์ต่างๆ ได้แก่ Link setup, Link configuration, Link packet control, Transfer ตลอดจนจัดการ Link security ในระหว่างช่วงกำหนดค่าเริ่มต้นของการเชื่อมต่อรวมถึงขณะที่กำลังเชื่อมต่อด้วย และยังเป็นตัวที่ควบคุมการทำงานแบบประหยัดกำลังไฟฟ้า (power saving mode)

ชั้นที่ 5 HCI (Host Control Interface) เป็นส่วนที่จัดเตรียม interface มาตรฐานระหว่างอุปกรณ์บลูทูธ (Bluetooth module) และ Link Manager เป็นโปรโตคอลที่ใช้เชื่อมต่อระหว่างโปรแกรมชั้นบนที่ทำงานอยู่บนระบบหนึ่งกับส่วนควบคุมการทำงานของบลูทูธ (เช่น card PCMCIA Bluetooth ที่ต่ออยู่ที่โน้ตบุ๊ก) ทำให้โปรแกรมรู้จักคำสั่งควบคุมอุปกรณ์บลูทูธ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นที่ 6 L2CAP (Logical Link Control and Adaptation Protocol) ประกอบไปด้วยโปรโตคอล multiplexing, Segmentation, Quality of service โครงสร้างของ L2CAP จะจัดการเกี่ยวกับช่องสัญญาณเมื่อมีการเชื่อมต่อกัน โดยจะกำหนดให้ทุกช่องมีการติดต่อแบบ Full-duplex คือสามารถรับ - ส่งข้อมูลได้ทุกช่อง และยังทำหน้าที่ multiplex ข้อมูลจากชั้นบนซึ่งอาจจะมีการทำงานของโปรแกรมหลายๆ โปรแกรมพร้อมกัน จัดแบ่งข้อมูลออกเป็นแพ็คเกต และยังทำให้ application ใช้งานโปรโตคอลระดับสูงบางตัวได้ (Higher-layer protocol) เช่น TCP/IP, RFCOMM

ชั้นที่ 7 RFCOMM / SDP สำหรับ RFCOMM (Radio Frequency Communication) เป็นโปรโตคอลเสมือนที่ทำให้ application ข้างบนมองบลูทูธเป็นเหมือนพอร์ตอนุกรม (Serial port) ทั่วไป ส่วน SDP (Service Discovery Protocol) เป็นโปรโตคอลที่ช่วยค้นหาบริการจากอุปกรณ์บลูทูธตัวอื่นๆ ที่อยู่ในขอบเขต piconet เดียวกัน SDP เป็น layer ที่เปิดทางให้กับบริการระดับสูง (High-Level Services) เช่น LAN access or printer ให้ผู้ใช้และ application อื่นๆ เข้าถึงได้

ชั้นที่ 8 Applications เป็นส่วนของโปรแกรมที่ใช้รับและส่งข้อมูลกับผู้ใช้

2.2.5 เทคนิคการส่งข้อมูล

เมื่อได้ข้อตกลงในความถี่ที่จะใช้งานจำนวนช่องสัญญาณและ Bandwidth ของแต่ละช่องสัญญาณแล้ว ต่อไปคือการเลือกวิธีการ Modulate เข้าไปกับคลื่นพาหะ เนื่องจากการ Modulate ในแต่ละแบบจะมีผลต่อความเร็วในการส่งข้อมูล และข้อจำกัดของ Bandwidth ที่กว้างเพียง 1 MHz ต่อช่องสัญญาณ บวกกับความต้องการในการส่งข้อมูลด้วยความเร็วสูงสุด Bluetooth จึงได้เลือกวิธีการ Modulate แบบ Gaussian Frequency-Shift Keying (GFSK)

การ Modulate ด้วยวิธี GFSK สามารถส่งข้อมูลได้ 1 บิตต่อความถี่คลื่นพาหะ 1 Hz ซึ่งหมายความว่าแต่ ละช่องสัญญาณสามารถส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาที โดยถ้าบิตข้อมูลเป็น 1 จะเกิดการเปลี่ยนแปลงความถี่ในด้านลบจากความถี่พาหะ

การรับส่งข้อมูลจะทำโดยการแบ่งข้อมูลออกเป็นแพคเกตย่อยๆ แล้วใช้วิธีการส่งแบบ Half - duplex เพื่อประหยัด จังหวะการรับและส่งข้อมูลทั้งหมดถูกกำหนดโดยอุปกรณ์ที่เป็น Master ในลักษณะของการ Pole ซึ่งอุปกรณ์ที่เป็น Slave จะต้องตอบกลับมายังอุปกรณ์ Master ในทุกๆแพคเกตเพื่อให้ Master รู้ว่ายังคงติดต่อกับ Slave ได้

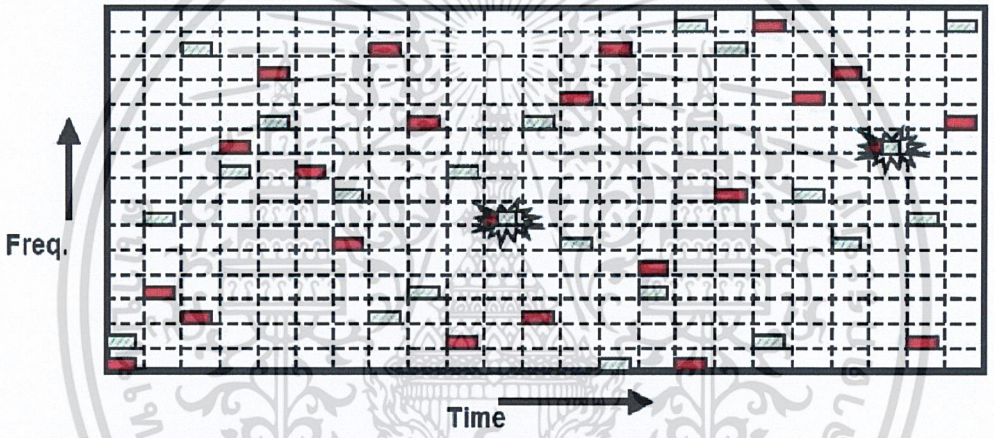
เมื่อมีการแบ่งข้อมูลออกเป็นแพคเกต ทำให้แต่ละแพคเกตต้องมีข้อมูลส่วนหัว (Header) เพิ่มเข้ามาเพื่อให้ฝั่งรับสามารถประกอบรวมข้อมูลทั้งหมดเข้าด้วยกันได้อย่างถูกต้อง นอกจากนั้นก่อนการส่งแต่ละครั้งจะต้องมีการส่งข้อมูลเพื่อทำการซิงโครไนซ์สัญญาณนาฬิกาของฝั่งรับและฝั่งส่งให้เท่ากัน เพื่อให้สามารถรับและส่งข้อมูลกันได้อย่างถูกต้อง เมื่อรวมปริมาณข้อมูลทั้งหมดที่จำเป็นในการส่งแต่ละครั้งจะทำให้ความเร็วในการส่งข้อมูลจริงลดลงจาก 1 Mb/s เหลือ 723.2 kb/s ในทิศทางหนึ่ง และ 57.6 kb/s ในอีกทิศทางหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากความเร็วและความสะดวกสบายแล้ว สิ่งที่จำเป็นอย่างยิ่งสำหรับการสื่อสารข้อมูลในปัจจุบันคือ ความปลอดภัยของข้อมูล โดยเฉพาะอุปกรณ์บลูทูธที่สามารถทำงานได้ในทุกที่ยิ่งมีความจำเป็นที่จะต้องมีการรักษาความปลอดภัยของข้อมูลเป็นอย่างดี เทคนิคการส่งข้อมูลที่บลูทูธใช้คือ เทคนิคการกระโดดข้ามทางความถี่ (Frequency Hopping Spread Spectrum : FHSS)

2.2.6 เทคนิคการกระโดดข้ามทางความถี่

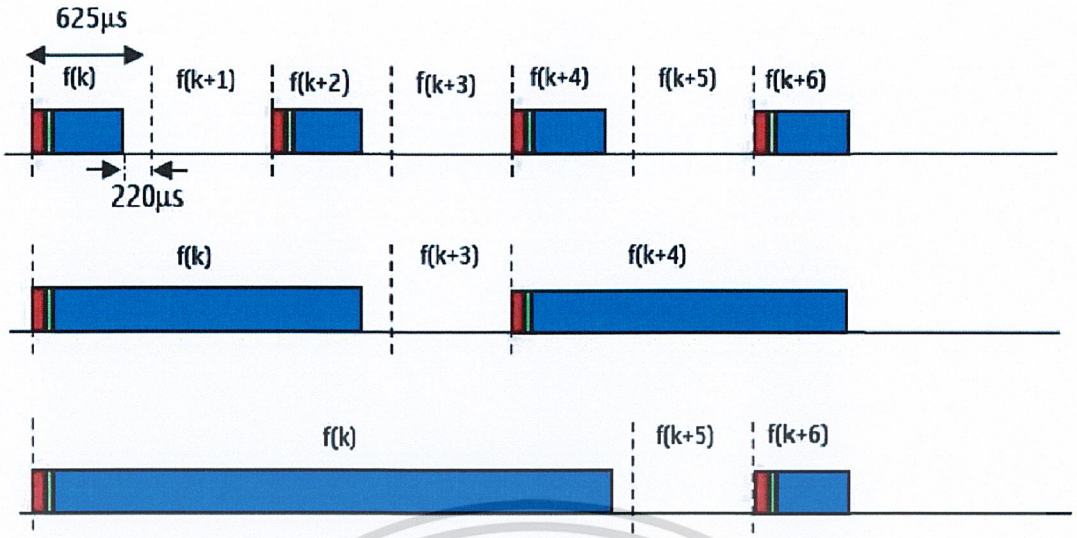
เทคโนโลยีบลูทูธใช้เทคนิคการกระโดดข้ามทางความถี่ (Frequency Hopping Spread Spectrum : FHSS) ซึ่งทุกๆ แพคเกตจะถูกส่งไปยังช่องความถี่ที่แตกต่างกัน การต่อต้านสัญญาณรบกวนสามารถทำได้ดีเนื่องจากมี hop rate สูงถึง 1600 hops/sec เมื่อมีอุปกรณ์อื่นๆ มารบกวนการส่ง แพคเกตนั้นๆ จะถูกส่งใหม่ในอีกช่องสัญญาณ เช่น ในรูปที่ 2.5 แพคเกตของอุปกรณ์ 2 ตัว ต้องการที่จะใช้ความถี่เดียวกัน



รูปที่ 2.5 Frequency Hopping และการเกิด Collision

ในการรับและการส่งสัญญาณจะมีการตัดแบ่งข้อมูลออกเป็นช่วงๆ ความยาวปกติคือ 625 ms โดยปกติแล้วแพคเกตหนึ่งๆสามารถรวมอยู่ในช่อง (slot) เดียวกันได้ แต่แพคเกตบางอันสามารถขยายได้ถึง 3 หรือ 5 slot ดังที่แสดงในรูปที่ 2.6 Multi-slot แพคเกตทุกๆ slot จะถูกส่งในช่วงความถี่เดียวกันจนจบแพคเกตนั้นและในการส่ง Multi-slot แพคเกตจะมี data rate ที่สูงเพราะ header จะใช้งานเพียงครั้งเดียวใน 1 แพคเกต อย่างไรก็ตาม ในกรณีที่การส่งข้อมูลมีความหนาแน่นและแพคเกตที่มีความยาวมากๆก็จะมี โอกาสสูญเสียมากเช่นเดียวกัน

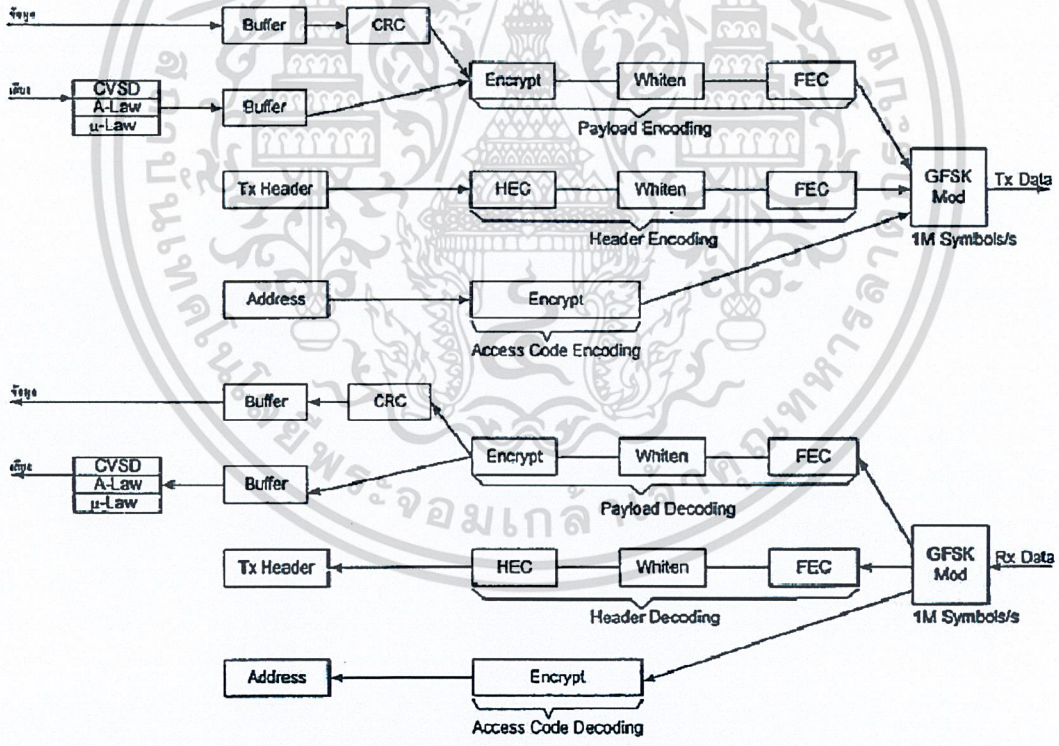
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
117193



รูปที่ 2.6 Slot แพคเกต

2.2.7 รูปแบบแพคเกตของบลูทูธ

ในการส่งข้อมูลนั้นจะเป็นการส่งทีละแพคเกต ซึ่งมีวิธีการสร้างแพคเกตดังนี้



รูปที่2.7 การสร้างแพคเกตข้อมูล

จากรูปที่ 2.7 แสดงให้เห็นถึงกระบวนการสร้างแพคเกตข้อมูล โดยในขั้นตอนการสร้างแพคเกตข้อมูลจะมีบล็อกการทำงานย่อยๆ ได้แก่

- บล็อก CRC แสดงถึงการคำนวณส่วนป้องกันความผิดพลาดแบบ CRC-16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บล็อก HEC แสดงถึงการคำนวณส่วนป้องกันความผิดพลาดแบบ CRC-8 ในส่วนของ Header
- บล็อก Whitening ข้อมูลที่หมายถึงการเติมบิตข้อมูลที่สุ่มอย่างมีรูปแบบเข้าไปในตัวข้อมูลเพื่อป้องกันการเกิดปัญหาจากการส่งบิตข้อมูล '1' หรือ '0' ต่อเนื่องกันเป็นจำนวนมากๆ
- บล็อก FEC (Forward Error Correction) เป็นการเข้ารหัสข้อมูลด้วยการเพิ่มพาริตีบิตแบบพิเศษ เพื่อตรวจสอบความผิดพลาดในการส่ง และเมื่อพบว่าผิดพลาดก็สามารถแก้ไขข้อมูลให้กลับมาอย่างถูกต้องได้ (ถ้าไม่ผิดมากเกินไป) การเข้ารหัส FEC มีอยู่ 2 แบบคือ แบบ 1/3 และ 2/3 โดยการ เข้ารหัสแบบ 1/3 จะมีความแข็งแรงต่อความผิดพลาดมากกว่าแบบ 2/3 แต่ต้องเพิ่มพาริตีบิตเป็นจำนวนมากกว่า ใช้กับข้อมูลที่มีความสำคัญมากเช่น Header เท่านั้น ส่วนข้อมูลทั่วไปจะใช้การ เข้ารหัสแบบ 2/3 เท่านั้น

โดยแต่ละแพคเกตจะประกอบไปด้วย 3 ส่วนย่อยๆ ได้แก่ ส่วน Access Code, ส่วน Header และ ส่วนข้อมูลจริง (Payload) โดยขนาดของ Access Code และ Header จะมีขนาดคงที่ (Fixed) คือ 72 และ 54 บิตตามลำดับ ส่วน Payload นั้นมีขนาดขึ้นอยู่กับลักษณะการใช้งานซึ่งอยู่ในช่วงระหว่าง 0 - 2745 บิต

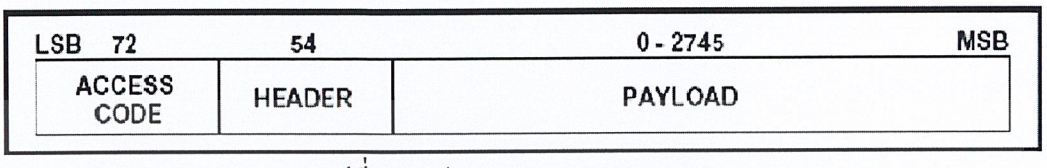
ลักษณะการใช้งานแพคเกตจะแบ่งเป็น 2 ประเภท คือแพคเกตควบคุม (Control Packet) สามารถมีได้เพียงแค่ Access Code หรือมี Access Code กับ Header โดยไม่ต้องมี Payload ส่วนแพคเกตข้อมูล (Data Packet) นั้น จำเป็นจะต้องมีครบสมบูรณ์ทั้ง 3 ส่วน ดังนี้

- Access Code เกิดจากการนำเอา Address ของอุปกรณ์มาผ่านการเข้ารหัสตามที่ระบุไว้ในมาตรฐานบลูทูธ Master หรือ Slave ที่ทำการรับข้อมูลบนเครือข่าย piconet จะทำการเปรียบเทียบสัญญาณที่ เข้ามากับค่า Access Code ถ้า Access Code ไม่ตรงกันก็จะไม่พิจารณาแพคเกตที่รับเข้ามา ประโยชน์ของ Access Code นั้นนอกจากใช้แยกแยะแพคเกตแล้ว ยังใช้สำหรับการซิงโครไนซ์และการชดเชย สำหรับ offset อีกด้วย รหัส Access Code ต้องทนทานต่อสัญญาณแปลกปลอมที่เข้ามาแทรกแซงรบกวน
- Header เป็นส่วนที่เก็บข้อมูลการควบคุมที่สำคัญๆเช่น Media Access Control (MAC), Address, Packet type, Flow control bits, บิตที่ใช้สำหรับ ARQ (Automatic Retransmission Query) และ ฟิลด์ของ Header Error Check (HEC) โดย Header มีความยาว 54 บิตเกิดจากการนำข้อมูลบ่งบอกลักษณะ ของแพคเกตและข้อมูลอื่นๆที่จำเป็นมาคำนวณส่วนป้องกันความผิดพลาดทั้งแบบ CRC และแบบ FEC
- ส่วนของข้อมูล (Payload) อาจจะมีหรือไม่มีก็ได้ และสามารถแบ่งข้อมูลออกได้เป็น 2 ชนิดคือ ข้อมูลที่เป็นเสียงกับข้อมูลทั่วไป ซึ่งข้อมูลทั้ง 2 ชนิดจะมีกระบวนการสร้างแพคเกตที่แตกต่างกัน 1 ขั้นตอนคือ ถ้าเป็นข้อมูลทั่วไปจะมีการนำตัวข้อมูลไปคำนวณส่วนตรวจสอบความผิดพลาด CRC-16 แต่ถ้าเป็นข้อมูลเสียงจะไม่มีการคำนวณส่วนตรวจสอบความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากการส่งข้อมูลเสียงจะเป็นการส่งแบบเรียลไทม์ซึ่งจะไม่สามารถส่งข้อมูลซ้ำได้ถึงแม้จะเกิดการผิดพลาดจึงไม่จำเป็นต้องมีส่วนตรวจสอบ

ข้อมูลทั้ง 3 ส่วน เมื่อผ่านกระบวนการทั้งหมดแล้วก็จะถูกนำมารวมกันเพื่อสร้างเป็นแพค-เกต โดยส่วนแรกของแพคเกตคือ Access Code ตามมาด้วย Header แล้วก็ปิดท้ายด้วยตัวข้อมูลที่จะส่ง ตามที่ แสดงไว้ในรูปที่ 2.8



รูปที่ 2.8 รูปแบบของบลูทูธแพคเกต

2.2.8 ระบบรักษาความปลอดภัยของบลูทูธ

ในการส่งสัญญาณระหว่างอุปกรณ์บลูทูธนั้นมีระบบรักษาความปลอดภัย (Security) เพื่อป้องกันการบุกรุกมาขโมยข้อมูล (Hack) โดยแบ่งการทำงานออกเป็น 3 ระดับดังนี้

1.) ไม่ใช้ระบบรักษาความปลอดภัย (Unsecured) คือ ไม่มีการป้องกันการล้วงเอาข้อมูลโดยผู้ใช้ที่มีอุปกรณ์บลูทูธสามารถเชื่อมต่อกันในเครือข่ายได้ทันทีเมื่อเข้ามาในขอบเขตรัศมีการติดต่อ

2.) ระบบรักษาความปลอดภัยแบบเซอร์วิส (Service Secure) คือ มีการแสดงตัวตนของผู้ใช้ (User) ก่อนการเข้าถึงหรือเชื่อมต่อ เพื่อเป็นการยืนยันว่าผู้ที่จะเข้ามาเชื่อมต่อเป็นผู้ที่ได้รับอนุญาตให้เข้ามาเชื่อมต่อได้

3.) ระบบรักษาความปลอดภัยแบบลิงค์ (Link Secure) คือ มีการแสดงตัวตนของผู้ใช้เช่นเดียวกับระบบรักษาความปลอดภัยแบบ Service แต่เพิ่มความปลอดภัยมากขึ้นด้วยการเข้ารหัส (Encryption) ข้อมูลที่จะส่งถึงกันก่อนทำการส่งสัญญาณ ซึ่งหากว่ามีการล้วง (Hack) เอาข้อมูลระหว่างที่ทำการส่งสัญญาณนั้น ผู้ล้วงเอาข้อมูล (Hacker) จะไม่สามารถเข้าใจข้อมูลที่ล้วงเอาไปได้เนื่องจากได้ทำการเข้ารหัสเอาไว้

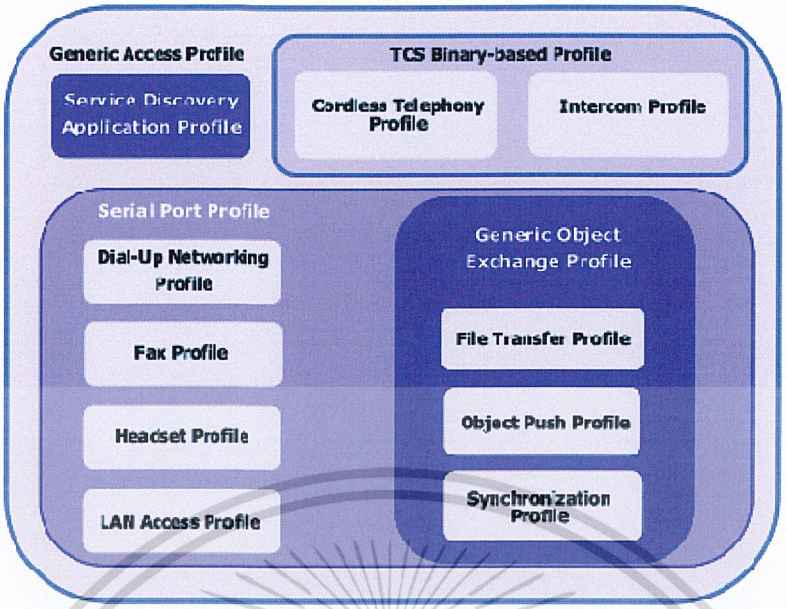
ความปลอดภัยในการเชื่อมต่อ และ ส่งข้อมูลผ่านทาง Bluetooth มีเป้าหมายดังนี้

- การรักษาความลับ(Confidentiality)
- การพิสูจน์ตัวตน(device authentication)
- การกำหนดสิทธิ์(device authorization)
- ความถูกต้อง สมบูรณ์ของข้อมูล (Integrity)

2.2.9 ลักษณะเฉพาะของแอพพลิเคชั่นซอฟท์แวร์ของบลูทูธ

แอพพลิเคชั่นซอฟท์แวร์ (Application Software) เป็นส่วนประกอบระดับสูง (High-Level) ของบลูทูธประกอบด้วย User interface และ Profile ต่างๆ มากมาย ซึ่งจะทำให้แอพพลิเคชั่นเข้าถึงการทำงานของฮาร์ดแวร์ได้ รูปที่ 2.9 แสดง Profile ที่ใช้ในบลูทูธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 Bluetooth Profile

Bluetooth Profile ทำให้อุปกรณ์บลูทูธแต่ละชนิดสามารถติดต่อเชื่อมถึงกันได้ โดยมีมาตรฐานในการติดต่อและเข้าถึงตัวอุปกรณ์ ทำให้อุปกรณ์บลูทูธชนิดหนึ่งจากผู้ผลิตหนึ่งสามารถเชื่อมต่อกับอุปกรณ์บลูทูธอีกชนิดหนึ่งจากอีกผู้ผลิตได้

2.2.9.1 โพรไฟล์อุปกรณ์ทั่วไป (Generic Access Profile - GAP)

GAP เป็นโพรไฟล์พื้นฐาน กล่าวคือโพรไฟล์อื่นๆจะอาศัยหลักการ เบื้องต้นของ GAP และเนื่องจากโพรไฟล์นี้เป็นพื้นฐานของชุดอุปกรณ์ที่ใช้ติดต่อสื่อสารทั้งหมด ดังนั้นทุกๆอุปกรณ์จึงต้องสนับสนุนการทำงานของโพรไฟล์นี้ด้วยความต้องการพื้นฐานที่เหมือนกันคือ

1.) ต้องสามารถใช้ได้กับอุปกรณ์บลูทูธทุกๆ ชิ้น

2.) ลักษณะการทำงานโดยทั่วไป คือค้นหาอุปกรณ์บลูทูธที่จะเชื่อมต่อ แล้วทำการเชื่อมต่อ

3.) สามารถจัดการลิงค์ (Link) ที่ใช้ในการเชื่อมต่อระหว่างอุปกรณ์บลูทูธได้ง่ายและสะดวก

4.) โพรซีเยอร์ (Procedure) ที่เกี่ยวข้องกับโพรไฟล์นี้จะต้องสามารถใช้ในระดับการรักษาความปลอดภัย (Security Level) ขั้นต่างๆได้ ซึ่งมี 3 ระดับดังที่กล่าวมาแล้ว

ด้วยเหตุนี้วัตถุประสงค์ของ GAP คือ การกำหนดหมวด (Mode) การทำงานที่ใช้สำหรับ Transport Profile และ Application Profile และใช้เป็นตัวบอกสถานการณ์ติดต่อ รวมถึงรับผิดชอบด้านลิงค์ (Link) ที่มาเชื่อมต่อและช่องสัญญาณ (Channel) ที่ใช้ติดต่อระหว่างอุปกรณ์บลูทูธอีกด้วย

2.2.9.2 โพรไฟล์บริการเชื่อมต่อระหว่างอุปกรณ์ (Service Discovery Profile - SDP)

โดยทั่วไปในอุตสาหกรรมอุปกรณ์บลูทูธจะคาดหวังถึงปริมาณและชนิดของบริการที่สามารถรองรับการเชื่อมต่อระหว่างอุปกรณ์บลูทูธซึ่งเพิ่มขึ้นอย่างรวดเร็ว หน้าที่ของ SDP คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดเตรียมโพรซีเยอร์ (Procedure) มาตรฐานสำหรับบริการชนิดต่างๆให้สามารถใช้งานร่วมกับอุปกรณ์บลูทูธทุกแบบได้

2.2.9.3 โพรไฟล์พอร์ตอนุกรม (Serial Port Profile - SPP)

โพรไฟล์นี้จะจัดเตรียมโปรโตคอล (Protocol) และโพรซีเยอร์ (Procedure) สำหรับให้อุปกรณ์บลูทูธ ทำงานในลักษณะเดียวกันกับพอร์ทอนุกรม (Serial Port) ที่เชื่อมต่อด้วยสายอนุกรม (Serial cable) ทั่วไป โดยจะทำการแปลงสัญญาณแบบ RS232 ก่อนที่จะทำการส่งสัญญาณไปยังปลายทาง เมื่อถึงปลายทางแล้ว จะแปลงสัญญาณนั้นกลับไปเป็น RS232 เพื่อให้สามารถใช้งานได้เช่นเดียวกับพอร์ทอนุกรม (Serial Port) ทั่วไป

SPP นี้ใช้ในการพัฒนาโพรไฟล์ย่อย ซึ่งเป็นโพรไฟล์การใช้งานอีก 5 โพรไฟล์ได้แก่

1.) โพรไฟล์อุปกรณ์สวมศรีษะ (Headset Profile) เป็นโพรไฟล์ที่ใช้ในอุปกรณ์บลูทูธประเภทสวมศรีษะ (Headset) ที่ใช้กับโทรศัพท์มือถือ ซึ่งไม่จำเป็นต้องสนับสนุนระบบรักษาความปลอดภัย (Security) หรือการเข้ารหัส (Encryption) ยังไม่จำเป็นต้องมีการกำหนดรูปแบบในลักษณะของมาสเตอร์ (Master) และสเลฟ (Slave) เหมือนกับการเชื่อมต่อปกติ

2.) โพรไฟล์อุปกรณ์โทรสาร (Fax Profile) เป็นโพรไฟล์ที่ให้บริการในการส่งข้อมูลในรูปแบบของโทรสาร (Fax) ซึ่งไม่จำเป็นต้องมีการกำหนดรูปแบบในลักษณะของมาสเตอร์ (Master) และ สเลฟ (Slave) แต่สนับสนุนการเข้ารหัส (Encryption) ข้อมูลที่จะทำการส่ง

3.) โพรไฟล์เชื่อมต่อโครงข่าย (Dial-up Networking Profile) เป็นโพรไฟล์ที่สนับสนุนการเชื่อมต่อกับอินเตอร์เน็ต โดยอาศัยอุปกรณ์บลูทูธเชื่อมต่อแทนโมเด็ม (modem) จะต้องใช้ระบบรักษาความปลอดภัยอย่างสมบูรณ์แบบ เพื่อป้องกันข้อมูลที่ส่งผ่านระหว่างอุปกรณ์ขณะเชื่อมต่อ

4.) โพรไฟล์เชื่อมต่อระบบ (LAN Access Profile) เป็นโพรไฟล์ที่สนับสนุนการเชื่อมต่อกันระหว่างอุปกรณ์บลูทูธในลักษณะเครือข่ายเนตเวิร์ครูปแบบแลน (LAN) ซึ่งสามารถทำให้เข้าถึงและใช้ข้อมูลร่วมกันระหว่างอุปกรณ์ได้

2.2.9.4 โพรไฟล์แลกเปลี่ยนข้อมูล (Generic Object Exchange Profile - GOEP)

GOEP เป็นโพรไฟล์ที่เกี่ยวกับการแลกเปลี่ยนวัตถุ (Object) ระหว่างอุปกรณ์บลูทูธ เช่น แฟ้มข้อมูล (file) เสียง (voice) เป็นต้น ภายในมีโพรไฟล์การใช้งาน 3 โพรไฟล์ คือ

1.) โพรไฟล์ส่งข้อมูลเพื่อการแลกเปลี่ยนข้อมูล (Object Push Profile)

2.) โพรไฟล์อุปกรณ์กำหนดจังหวะการแลกเปลี่ยนข้อมูล (File Transfer Profile)

3.) โพรไฟล์อุปกรณ์ถ่ายทอดแฟ้มข้อมูล(Synchronization Profile)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.9.5 โพรไฟล์กำหนดสัญญาณ (TCS Binary Based Profile – TCS BIN)

TCS BIN เป็นโพรไฟล์ที่ทำหน้าที่กำหนดสัญญาณการควบคุม สำหรับสร้างข้อมูลเสียงภายในมีโพรไฟล์การใช้งาน 2 โพรไฟล์ คือ

1.) โพรไฟล์สำหรับโทรศัพท์ (Cordless Telephony Profile - CTP) เป็นโพรไฟล์สนับสนุนการเชื่อมโยงไร้สายบลูทูธของโทรศัพท์มือถือ

2.) โพรไฟล์เชื่อมโยงบลูทูธ (Intercom Profile - ICP) เป็นโพรไฟล์ที่กำหนดวิธีการเปิดใช้งานบลูทูธของโทรศัพท์มือถือในเครือข่ายเดียวกัน สามารถสื่อสารกันได้โดยตรงโดยไม่ต้องใช้เครือข่ายโทรศัพท์สาธารณะ

2.2.10 ส่วนควบคุมการเชื่อมต่อ

ส่วนควบคุมการเชื่อมต่ออุปกรณ์บลูทูธมีสถานะต่างๆ ดังนี้

1.) Standby เป็นสถานะปกติที่จะมีการใช้พลังงานต่ำ ไม่มีการส่งข้อมูลระหว่างอุปกรณ์ใช้งานเพียงแต่ clock ภายในและไม่มีการติดต่อใดๆกับอุปกรณ์อื่นๆเลย

2.) Inquiry เป็นสถานะค้นหาอุปกรณ์ตัวอื่นและอุปกรณ์ที่จะติดต่อด้วย อุปกรณ์ที่ถูกค้นหาจะส่งแพคเกต FHS จะมีข้อมูลสำคัญที่จะทำให้อุปกรณ์ที่ค้นหานั้นสร้างการเชื่อมต่อขึ้นมา

3.) Inquiry scan อุปกรณ์บลูทูธที่อยู่ในบริเวณการค้นหาของ master คอยรับแพคเกต inquiry ที่ใช้ในการเข้าถึง เมื่อได้รับแล้วจะตอบสนองต่อ inquiry โดยใช้ข้อมูลจากแพคเกต FHS ที่ได้รับมา

4.) Page Master ที่พบอุปกรณ์ในระยะค้นหา จะทำการเรียกอุปกรณ์ชิ้นนั้น เพื่อทำการเชื่อมต่อและเมื่อ Slave ได้รับการเรียกก็จะทำการตอบรับด้วย Device Access Code เมื่อ Master ได้รับการตอบสนองก็จะทำการตอบรับด้วยการส่งค่า Clock-realtime, ค่าแอดเดรสและ ค่า FHS แพคเกตที่จะใช้ในการ hopping จากนั้นทั้งคู่ก็จะอยู่ในสถานะ Connection

5.) Page scan Slave เริ่มรอคอย Page Message ที่จะส่งมาหา

6.) Connection จะมีการแลกเปลี่ยนแพคเกตต่างๆผ่านทาง Channel, Access Code และใช้งาน Master Clock และทำ Frequency hopping

7.) Connection - Hold อุปกรณ์บลูทูธจะไม่รับส่งแพคเกต ACL และจะอยู่ใน Sleep mode ที่ประหยัดพลังงานแต่พร้อมที่จะรับการ Page และการทำ Scanning

8.) Connection - sniff slave จะเริ่มพักผ่อน แทนที่จะต้องมีการเชื่อมต่อกันตลอดในทุกๆ timeslot จึงมีการกำหนด timeslot ที่จะใช้ติดต่อกัน และ timeslot อื่นๆจะพักเพื่อประหยัดพลังงาน

9.) Connection -Park Slave ไม่ต้องการติดต่อใดๆแต่จะยังคง Synch อยู่ ที่สถานะนี้จะมีการให้แอดเดรสใหม่ที่ชื่อว่า Park Mode Address เพื่อใช้แทนแอดเดรสเก่า เมื่อมีการติดต่อกันก็จะติดต่อผ่านแอดเดรสใหม่ซึ่งเป็นการประหยัดพลังงานได้อย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของส่วนควบคุมการเชื่อมต่อ

1.) host ร้องขอการ inquiry

2.) inquiry ถูกส่งไปโดยใช้ลำดับความถี่กระโดดที่ใช้ในการ inquiry

3.) อุปกรณ์ที่ทำการ inquiry ตอบกลับด้วยแพคเกต FHS ประกอบด้วยข้อมูลในการสร้างการเชื่อมต่อ

4.) ข้อมูลในแพคเกต FHS ถูกส่งกลับไปยัง host

5.) host ร้องขอการเชื่อมต่อไปยังอุปกรณ์ที่ตอบรับการ inquiry

6.) กระบวนการ Paging ใช้ในการเริ่มต้นการเชื่อมต่อไปยังอุปกรณ์ที่ต้องการติดต่อด้วย

7.) อุปกรณ์ที่ต้องการติดต่อด้วยซึ่งทำการ Page scan อยู่จะตอบกลับมา

8.) ถ้าอุปกรณ์ที่ทำการ Page scan อยู่ยอมรับการเชื่อมต่อก็จะกระโดดไปใช้ความถี่ที่ Master กำหนดให้และติดต่อสื่อสารผ่านลำดับความถี่ที่ใช้ในการกระโดดนั้น

ส่วนจัดการการเชื่อมต่อ จัดการการทำงานต่างๆดังนี้

1.) นำ Slave ไปรวมกับ piconet และกำหนดเลข AM_ADDRESS ให้กับอุปกรณ์นั้น

2.) หยุดการเชื่อมต่อชั่วคราวเพื่อนำ Slave เพื่อรวมเข้าไปใน piconet

3.) ปรับแต่งค่าการเชื่อมต่อแบบ SCO และ ACL

4.) นำการเชื่อมต่อไปยังโหมดประหยัดพลังงาน เช่น พัก, sniff, หยุดรอ

5.) ควบคุมโหมดการทดลอง

ส่วนการจัดการการเชื่อมต่อบลูทูธนี้จะติดต่อกับส่วนที่อยู่ใน Bluetooth device ตัวอื่นโดยใช้โปรโตคอลจัดการการเชื่อมต่อ ซึ่งส่วนการจัดการการเชื่อมต่อมีหน้าที่เริ่มต้นการเชื่อมต่อ และคอยดูแลการเชื่อมต่อให้ Bluetooth device สามารถติดต่อสื่อสารกันได้ ซึ่งจะเริ่มสร้างการเชื่อมต่อแบบ ALC โดยข้อความ LMP จะถูกใช้ในการเชื่อมต่อแบบ ALC จากนั้นข้อความ LMP ก็จะถูกใช้ในการเชื่อมต่อ SCO บนการเชื่อมต่อ ACL เดิม ส่วนการจัดการการเชื่อมต่อจะดูแลข้อมูลบนอุปกรณ์ที่เป็น Slave ซึ่งได้กำหนด AM_ADDRESS ให้

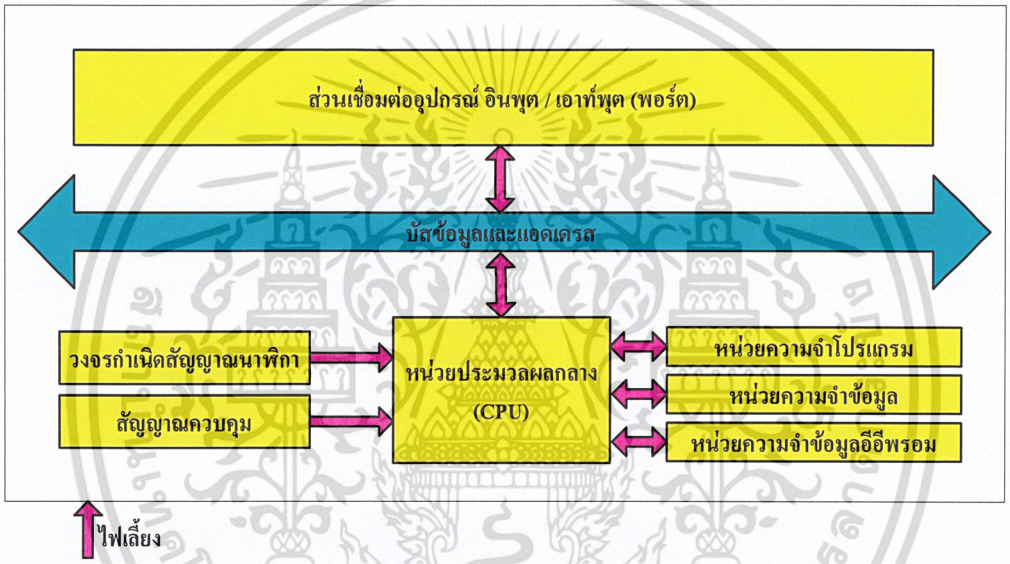
2.3 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) มาจากคำ 2 คำ คือ ไมโคร (Micro) หมายถึง ขนาดเล็ก และคำว่า คอนโทรเลอร์ (controller) หมายถึง ตัวควบคุมหรืออุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์จึงหมายถึง อุปกรณ์ควบคุมขนาดเล็ก ซึ่งเป็นอุปกรณ์ประเภทสารกึ่งตัวนำ มีหน้าที่เป็นหน่วยประมวลผลกลาง โดยรวบรวมคำสั่งการทำงานต่างๆไว้ในตัวเอง ทำให้สามารถโปรแกรมให้ทำงานตามเงื่อนไขที่กำหนด และสามารถสร้างสัญญาณส่งออกหรือรับสัญญาณเข้าเพื่อติดต่อกับอุปกรณ์ต่างๆภายนอกได้ มักถูกนำไปเป็นหน่วยประมวลผลภายในอุปกรณ์อิเล็กทรอนิกส์สำเร็จรูปต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์เป็นอุปกรณ์พิเศษที่รวมซีพียู, หน่วยความจำ, และพอร์ต อินพุต/เอาท์พุต บรรจุอยู่ภายใต้ตัวถังเดียวกันดังแสดงในรูปที่ 2.10 สามารถทำงานได้เมื่อจ่ายไฟเลี้ยงและต่อวงจรกำเนิดสัญญาณนาฬิกา จากนั้นซีพียูจะติดต่อกับหน่วยความจำโปรแกรมเพื่ออ่านคำสั่งที่ระบุไว้ โดยทำการอ้างตำแหน่งของหน่วยความจำผ่านสายสัญญาณที่เรียกว่า บัสแอดเดรส (Address Bus) แล้วทำการอ่านข้อมูลคำสั่งออกมาจากหน่วยความจำโปรแกรมในแอดเดรสนั้นๆ จากนั้นทำการประมวลผล โดยมีหน่วยความจำข้อมูลแรมเป็นที่พักของข้อมูลที่อยู่ระหว่างการประมวลผล ข้อมูลในการประมวลผลจะส่งผ่านสายสัญญาณที่เรียกว่า บัสข้อมูล (Data Bus) แล้วส่งต่อไปยังอุปกรณ์ภายนอกผ่านทางขาพอร์ตอินพุต เอาท์พุต

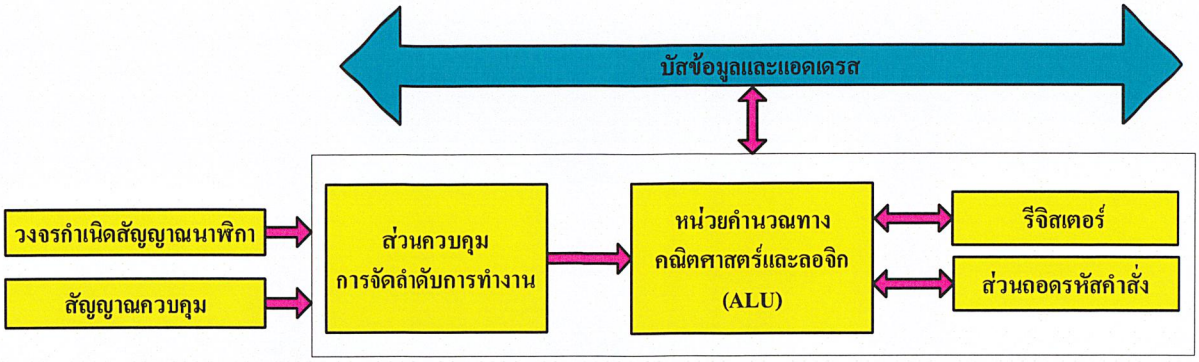


รูปที่ 2.10 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

2.3.1.1 หน่วยประมวลผลกลางซีพียู (CPU : Central Processing Unit)

ซีพียูทำหน้าที่ประมวลผลข้อมูลที่เข้ามาในระบบ แล้วทำการส่งต่อไปยังส่วนต่างๆเพื่อควบคุมการทำงานต่อไป จากรูปที่ 2.11 จะเห็นว่าหัวใจหลักของซีพียูคือหน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU : Arithmetic and logic unit) ซึ่งได้รับการกำหนดจังหวะการทำงานจากส่วนควบคุมลำดับการทำงาน โดยจังหวะการทำงานจะสัมพันธ์กับสัญญาณนาฬิกา เมื่อซีพียูทำการติดต่อหน่วยความจำ สิ่งที่ปรากฏขึ้นบนบัสข้อมูลภายในซีพียูคือ รหัสคำสั่ง (instruction code) ซึ่งต้องผ่านการทำงานของส่วนถอดรหัสคำสั่ง (instruction decode) เพื่อแปลงเป็นข้อมูลคำสั่งที่ซีพียูเข้าใจและสามารถดำเนินการต่อได้ หลังจากที่หน่วยคำนวณทางคณิตศาสตร์และลอจิกประมวลผลแล้ว จะส่งข้อมูลมายังส่วนเชื่อมต่อรีจิสเตอร์ภายในซีพียู เพื่อติดต่อกับส่วนอื่นๆ ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 โครงสร้างและส่วนประกอบหลักของซีพียู

การทำงานของซีพียูมี 2 จังหวะคือ เฟตช์ (fetch) และ เอ็กซีคิวต์ (execute) โดยการทำงานจะเริ่มจากการเฟตช์ คือการเรียกหรือการเข้าถึงคำสั่ง และทำการถอดรหัสเป็นภาษาเครื่องเพื่อเตรียมประมวลผล จากนั้นจะเป็นจังหวะเอ็กซีคิวต์ คือการกระทำตามคำสั่งที่กำหนดให้เสร็จสิ้น

2.3.1.2 หน่วยความจำ (Memory)

ไมโครคอนโทรเลอร์มีหน่วยความจำ 2 ประเภท คือ

1.) หน่วยความจำอีอีพรอม (EEPROM memory) หรือ หน่วยความจำโปรแกรม (Program memory) เป็นที่เก็บโปรแกรมและข้อมูลสำหรับการประมวลผลที่ไม่ต้องการให้สูญหายในกรณีที่ไม่มีไฟเลี้ยงวงจร มีการแบ่งออกเป็นส่วนๆเรียกว่า เพจ เพื่อเก็บโปรแกรมสำหรับทำงานที่แตกต่างและไม่เกี่ยวข้องกัน

2.) หน่วยความจำข้อมูลแรม (RAM data memory) แบ่งเป็น 2 กลุ่ม คือ

- หน่วยความจำข้อมูลสำหรับเก็บค่าตัวแปร (Variable RAM) หรือ รีจิสเตอร์ (register) ใช้เก็บค่าตัวแปรเพื่อทำการประมวลผล และข้อมูลทั่วไปชั่วคราว
- หน่วยความจำแรมสแครตช์แพด (Scratch pad RAM) เป็นที่เก็บข้อมูลชั่วคราวกรณีที่มีการกระโดดข้ามไปทำงานในโปรแกรมที่อยู่ต่างเพจกัน ซึ่งการเก็บข้อมูลจะเก็บในลักษณะของสแต็ก (stack) และในการดึงข้อมูลจะเป็นแบบ FIFO (เฉพาะรุ่น โดยทั่วไปเป็นแบบ LIFO)

ข้อมูลในหน่วยความจำจะหายไป เมื่อไม่มีการจ่ายไฟให้แก่ไมโครคอนโทรเลอร์ และเมื่อมีการจ่ายไฟให้ ค่าต่างๆในหน่วยความจำจะกลายเป็น 0

2.3.1.3 ส่วนติดต่ออุปกรณ์ภายนอก (Port)

ส่วนติดต่ออุปกรณ์ภายนอกหรือเรียกว่าพอร์ต (port) มี 2 ลักษณะคือ พอร์ตรับสัญญาณหรือพอร์ตอินพุต (input port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (output port) ส่วนนี้มีความสำคัญมาก เนื่องจากใช้เชื่อมต่อกับอุปกรณ์ภายนอก เช่น พอร์ตอินพุตใช้ต่อกับสวิตช์เพื่อรับข้อมูลที่ผู้ใช้งานกดป้อนเข้ามา หรือต่อกับหน่วยความจำเพื่อเพิ่มพื้นที่ในการเก็บข้อมูล ส่วนพอร์ตเอาต์พุตก็คือส่วนที่ต่อกับจอแสดงผลแอลซีดี เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.4 บัสข้อมูลและแอดเดรส (Bus)

การติดต่อแลกเปลี่ยนสัญญาณข้อมูลระหว่างซีพียู หน่วยความจำและพอร์ต จะกระทำบนสายสัญญาณจำนวนมาก เรียกว่า เส้นทางสัญญาณหรือบัส โดยแบ่งเป็นบัสข้อมูล (data bus), บัสแอดเดรส (address bus) และบัสควบคุม (control bus)

1.) บัสข้อมูลเป็นสายสัญญาณที่บรรจุข้อมูลสำหรับการประมวลผลทั้งหมด ขนาดของบัสจะขึ้นอยู่กับความสามารถในการประมวลผลของซีพียูและเทคโนโลยีของไมโครคอนโทรลเลอร์นั้นๆ

2.) บัสแอดเดรสเป็นสายสัญญาณที่บรรจุค่าตำแหน่งของหน่วยความจำ โดยการติดต่อกับหน่วยความจำนั้น ซีพียูต้องกำหนดตำแหน่งที่ต้องการอ่านหรือเขียนก่อน ซึ่งก็คือการกำหนดค่าแอดเดรส จำนวนสายสัญญาณของบัสแอดเดรส จึงต้องมีจำนวนมาก และถ้ายิ่งมีมากเท่าใดจะเป็นการแสดงถึงความจุของหน่วยความจำที่ไมโครคอนโทรลเลอร์ตัวนั้นสามารถติดต่อได้ สามารถคำนวณได้จาก

จำนวนแอดเดรสของหน่วยความจำ = 2^n โดยที่ n คือจำนวนสายสัญญาณ

3.) บัสควบคุมเป็นกลุ่มของสายสัญญาณควบคุมการติดต่อทั้งหมดของซีพียูกับหน่วยความจำและพอร์ต สำหรับสายสัญญาณควบคุมหลักได้แก่ สายสัญญาณเลือกอ่าน-เขียนหน่วยความจำ สายสัญญาณเลือกอ่าน-เขียนข้อมูลกับพอร์ต

2.3.1.5 วงจรกำเนิดสัญญาณนาฬิกา

การทำงานทั้งหมดในไมโครคอนโทรลเลอร์จะขึ้นอยู่กับการกำหนดจังหวะโดยใช้สัญญาณนาฬิกา หากสัญญาณนาฬิกามีความถี่สูงจังหวะในการทำงานของไมโครคอนโทรลเลอร์ก็จะถี่และมีมากตามส่งผลให้ไมโครคอนโทรลเลอร์นั้นมีความเร็วในการประมวลผลสูงตามไปด้วย

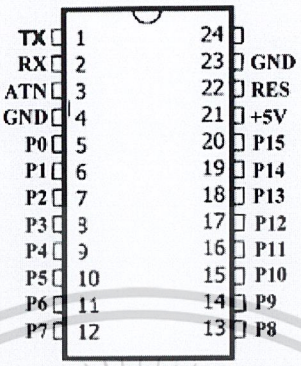
2.3.2 บอร์ดเบสิกแสตมป์

ปัจจุบันมีไมโครคอนโทรลเลอร์อยู่หลายตระกูลเช่น MCS-51, AVR, PIC, Z80, Basic Stamp ซึ่งแต่ละตระกูลจะมีจุดเด่นของตัวเอง ในปัญหาพิเศษนี้เลือกใช้ไมโครคอนโทรลเลอร์เบสิกแสตมป์ (Basic Stamp) ที่ง่ายต่อการศึกษา และสามารถศึกษาการใช้งานได้ด้วยตนเอง ซึ่งเบสิกแสตมป์เป็นไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัท Parallax Inc. มีตัวแปลภาษาเบสิก (Basic Interpreter) อยู่ภายในตัวเอง ในการพัฒนาเบสิกแสตมป์จะใช้ชุดคำสั่งภาษาเบสิกที่เรียกว่า พีเบสิก (PBASIC) ซึ่งมี 71 คำสั่ง (ขึ้นอยู่กับรุ่น) แต่ละคำสั่งสามารถนำไปใช้ได้ทันทีไม่ต้องเขียนโปรแกรมย่อย ทำให้มีความรวดเร็วในการประมวลผลที่สูงเพียงพอต่อการนำไปศึกษาการทำงานที่ไม่ซับซ้อนมากนัก เช่น สร้างหุ่นยนต์ หรือการประยุกต์ใช้งานเล็กๆทั่วไป ประกอบกับมีแหล่งข้อมูลสอนการนำไปศึกษาใช้งาน และเครื่องมือพัฒนาระบบที่สามารถดาวน์โหลดได้ฟรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.1 การจัดขาของบอร์ดเบสิกแสตมป์

ไมโครคอนโทรลเลอร์แต่ละตระกูล แต่ละรุ่นจะมีการจัดขาและวงจรที่แตกต่างกัน ในปัญหาพิเศษนี้เลือกใช้ไมโครคอนโทรลเลอร์เบสิกแสตมป์ (Basic Stamp) รุ่น i-stamp 2P24



รูปที่ 2.12 การจัดขาเชื่อมต่อของเบสิกแสตมป์

ตารางที่ 2.5 รายละเอียดของการจัดขาเชื่อมต่อและหน้าที่ของเบสิกแสตมป์

ขาที่	ชื่อขา	หน้าที่
1	Tx หรือ Sout	ส่งข้อมูลออกแบบอนุกรม เชื่อมต่อกับขา RxD (ขา 2) ของพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านคอนเน็กเตอร์ DB-9
2	Rx หรือ Sin	รับข้อมูลเข้าแบบอนุกรม เชื่อมต่อขา TxD (ขา 3) ของพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านคอนเน็กเตอร์ DB-9
3	ATN	ขารีเซตลอจิกสูง เชื่อมต่อกับขา DTR (ขา 4) ของพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านคอนเน็กเตอร์ DB-9
4	Vss/GND	ขากราวด์เชื่อมต่อกับขา GND (ขา 5) ของพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านคอนเน็กเตอร์ DB-9
5-20	P0-P15	ขาพอร์ตอินพุต เอาต์พุต 16 ขา
21	+5V	ขาไฟเลี้ยง +5V จากภายนอก
22	RES	ขารีเซตลอจิกต่ำ สำหรับรีเซต i-Stamp
23	Vss/GND	ขากราวด์ของ i-Stamp
24	N/A	ขานี้ไม่มีการใช้งาน

ส่วนการโปรแกรมข้อมูลใช้การโปรแกรมในลักษณะอนุกรมด้วยการติดต่อกับพอร์ตอนุกรมของเครื่อง คอมพิวเตอร์ โดยต่อสายจากขา TxD, RxD, DTR และ กราวด์ของพอร์ตอนุกรมเข้าสู่ขา S_{IN}, S_{0UT}, ATN และ กราวด์ของ i-Stamp นอกจากนั้นที่ขา 6 และ 7 ของพอร์ตอนุกรมต้องต่อถึงกันด้วย เพื่อให้สามารถโปรแกรมข้อมูลบน i-Stamp ได้ตลอดเวลาอย่างอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณจากพอร์ตอนุกรมของคอมพิวเตอร์โดยทั่วไป มีระดับแรงดันสูงกว่าระดับทีทีแอล กล่าวคือมี แรงดันประมาณ ±12V สำหรับคอมพิวเตอร์ตั้งโต๊ะ และ ±3V ถึง ±8V สำหรับคอมพิวเตอร์โน้ตบุ๊ก ดังนั้น จึงต้องมีวงจรแปลงระดับสัญญาณจากคอมพิวเตอร์ให้เป็นระดับทีทีแอล (0-5V) เพื่อให้สามารถติดต่อกับเบสิกแสตมป์อินเตอร์พรีตเตอร์ได้

ขา S_{IN} เป็นขาสำหรับรับข้อมูลจากคอมพิวเตอร์ เมื่อขาคอมพิวเตอร์ส่งข้อมูล "1" มา ทำให้ S_{IN} เป็นลบ ทรานซิสเตอร์ Q1 จะไม่ทำงาน ทำให้ IC1 มีแรงดัน +5V เกิดเป็นลอจิก "1" ในทางตรงข้ามเมื่อคอมพิวเตอร์ส่งข้อมูล "0" ขา S_{IN} จะเป็นบวกทรานซิสเตอร์ Q1 จะทำงาน ส่งผลให้ขา IC1 เสมือนต่อกราวนด์ มีลอจิกเป็น "0"

ขา S_{OUT} ใช้สำหรับส่งข้อมูลไปยังคอมพิวเตอร์ เมื่อเบสิกแสตมป์ส่งข้อมูล “1” ทรานซิสเตอร์ Q3 ไม่ทำงาน ที่ S_{OUT} จะได้รับแรงดันจากลบจากขา S_{IN} ผ่านทางตัวต้านทาน ทำให้ที่ขา S_{OUT} มีระดับแรงดันเป็นลบ ซึ่งคอมพิวเตอร์จะอ่านเป็นลอจิก “1” คือระดับแรงดันตั้งแต่ -3 ถึง –12V ในขณะที่ลอจิก “0” คือระดับแรงดันตั้งแต่ +3 ถึง +12V เมื่อเบสิกแสตมป์ส่งข้อมูล “0” ทรานซิสเตอร์ Q3 ทำงานที่ขา S_{OUT} จึงเกิดแรงดัน +5V ทำให้คอมพิวเตอร์อ่านข้อมูลได้เป็น “0” เมื่อเป็นเช่นนี้ในการติดต่อระหว่างเบสิกแสตมป์กับคอมพิวเตอร์จะต้องผลัดกันรับส่งข้อมูล เมื่อคอมพิวเตอร์ส่งข้อมูลมาเบสิกแสตมป์ต้องรับข้อมูลอย่างเดียว ไม่สามารถส่งข้อมูลกลับไปในเวลาเดียวกันได้

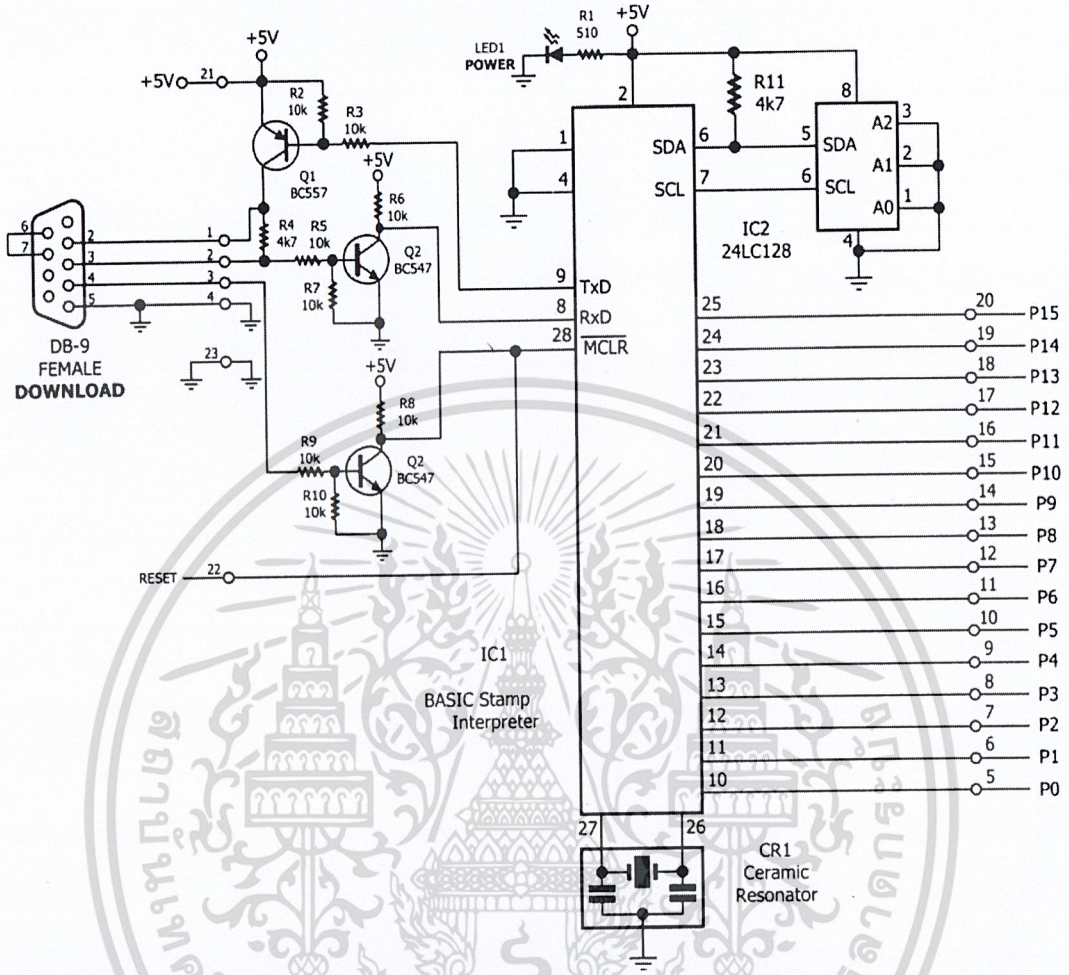
ขา ATN ซึ่งต่อเข้ากับขา DTR (Data Terminal Ready) ใช้ในตรวจสอบความพร้อมในการรับส่งข้อมูลของพอร์ตอนุกรม จะมีลักษณะทำงานคล้ายกับขา S_{IN} ถ้าหากคอมพิวเตอร์ส่งข้อมูล “0” หรือทำให้ขา DTR มีแรงดันเป็น +12V ทรานซิสเตอร์ Q2 ทำงาน ทำให้ขาคอลเล็กเตอร์ของ Q2 เสมือนต่อกราวด์ อันเป็นการสร้างสัญญานรีเซตให้แก่เบสิกแสตมป์ ดังนั้นในขณะที่ทำการโปรแกรมข้อมูลลงบนเบสิกแสตมป์ ซอฟต์แวร์ที่ใช้ในการโปรแกรมจะส่งพัลส์มายังขา ATN เพื่อรีเซต IC1 แล้วตามด้วยการส่งข้อมูลมายังขา S_{IN} เพื่อแจ้งให้ทราบว่า ต้องเขียนโปรแกรมใหม่ลงบนเบสิกแสตมป์ แต่ถ้าหากซอฟต์แวร์ทำให้ขา ATN เป็น -12V หรือเป็นลอจิก “1” ก็จะหมายความว่าขณะนี้เบสิกแสตมป์อยู่ในโหมดรัน นั้นคืออยู่ในโหมดทำงานปกติ

2.3.2.2 วงจรภายในบอร์ดเบสิกแสตมป์

สำหรับวงจรสมบูรณ์ของเบสิกแสตมป์แสดงในรูปที่ 2.13 หัวใจของวงจรคือ IC1 ซึ่งภายในบรรจุโปรแกรมแปลภาษาพีเบสิก เรียกว่า เบสิกแสตมป์อินเตอร์พรีตเตอร์ เมื่อผู้ใช้งานทำการเขียนโปรแกรมพีเบสิกเรียบร้อยแล้ว จะทำการดาวน์โหลดลงเบสิกแสตมป์ผ่านพอร์ตอนุกรม สัญญาณไฟฟ้าจะถูกแปลงระดับสัญญาณให้เป็นระดับทีทีแอลด้วยวงจรระดับแปลงสัญญาณซึ่งประกอบด้วย R2 - R10 และ Q1 - Q3 โดย Q1 ทำหน้าที่ส่งข้อมูลออกจาก IC2 ไปยังพอร์ตอนุกรมของคอมพิวเตอร์ผ่านทางขา 2 (RxD) ในขณะที่ Q2 ทำหน้าที่จัดระดับสัญญาณที่รับมาจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์เพื่อส่งข้อมูลให้แก่ IC2 ส่วน Q3 ใช้สำหรับรับสัญญาณรีเซตมาจากคอมพิวเตอร์เพื่อส่งต่อไปยัง IC1 อย่างไรก็ตามผู้ใช้งานสามารถทำการรีเซตระบบได้ด้วยการกดสวิตช์ S2



รูปที่ 2.13 วงจรสมบูรณ์ของเบสิกแสตมป์

IC1 จะทำการแปลภาษาที่เขียนขึ้นแล้วเก็บข้อมูลของโปรแกรมลงใน IC2 ซึ่งเป็นไอซีหน่วยความจำอีอีพรอมอนุกรมขนาด 16 กิโลไบต์ นั้นหมายความว่า IC2 จะทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ ไม่ว่าโปรแกรมที่เขียนขึ้นมีความผิดพลาดหรือไม่ IC1 ก็จะส่งสัญญาณแจ้งกลับไปยังคอมพิวเตอร์เพื่อให้ผู้ใช้งานทราบถึงผลของการเขียนและดาวน์โหลดโปรแกรม IC1 ได้รับสัญญาณนาฬิกาจาก CR1 เซรามิกเรโซเนเตอร์ 20MHz ภายใน CR1 จะมีตัวเก็บประจุชดเชยอยู่แล้ว ทำให้เบสิกแสตมป์ใช้อุปกรณ์ในการกำเนิดสัญญาณนาฬิกาน้อยลง เบสิกแสตมป์ต้องการไฟเลี้ยง +5V กระแสอย่างน้อย 65mA ทั้งนี้เพื่อให้ขนาดของบอร์ดเล็กที่สุดเท่าที่จะเป็นไปได้ จึงไม่บรรจุวงจรภาคจ่ายไฟรวมไว้ด้วย เนื่องจากต้องต่อร่วมกับวงจรหรืออุปกรณ์ภายนอกอื่นๆ ที่ต้องใช้ไฟเลี้ยง +5V อยู่แล้ว ผู้ใช้งานเพียงต่อสาย +5V และกราวด์เข้ามายังเบสิกแสตมป์เท่านั้นก็จะสามารถใช้งานได้ทันที ไม่จำเป็นต้องใช้อุปกรณ์สำหรับสร้างวงจรภาคจ่ายไฟเพิ่มเติมอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 ภาษาพีเบสิก

ในการเขียนโปรแกรมเพื่อควบคุมการทำงานของเบสิกแสตมป์นั้นจะใช้ภาษาเบสิกที่เรียกว่า พีเบสิก (PBASIC) โดยผ่านโปรแกรมที่ชื่อว่า เบสิกแสตมป์เอดิเตอร์ (BASIC Stamp Editor) โปรแกรมนี้จะถูกใช้ในการติดต่อคอมพิวเตอร์กับเบสิกแสตมป์ผ่านทางพอร์ตอนุกรม ใช้ในการเขียนและแก้ไขโปรแกรมภาษาเบสิก หรือทำหน้าที่เป็นเอดิเตอร์นั่นเอง โดยโปรแกรมจะทำการตรวจสอบการเขียนโปรแกรมในทุกบรรทัดว่าผิดพลาดหรือไม่ และใช้ในการโหลดหรือเขียนโปรแกรมลงบนตัวเบสิกแสตมป์ เพื่อทำการรัน อาจกล่าวได้ว่า โปรแกรมเบสิกแสตมป์เอดิเตอร์เพียงตัวเดียวทำงานได้ครบวงจร ทำให้การเรียนรู้เบสิกแสตมป์ง่ายและมีขั้นตอนน้อยที่สุด

2.3.3.1 รูปแบบโปรแกรม

โปรแกรมที่เขียนขึ้นเพื่อใช้งานเบสิกแสตมป์เป็นโปรแกรมภาษาเบสิกที่เรียกว่า พีเบสิก (PBASIC) และ ต้องบันทึกเป็นแฟ้มข้อมูลนามสกุล .bsp เท่านั้น มีกฎเกณฑ์และรูปแบบที่ต้องทราบและปฏิบัติตามดังนี้

1.) ค่าคงที่ ค่าของตัวเลขที่ปรากฏในโปรแกรม สามารถแบ่งได้เป็น 4 รูปแบบ คือ

- ตัวเลขฐานสิบ (decimal หรือ DEC) สามารถกำหนดได้โดยตรง โดยที่ไม่ต้องมีสัญลักษณ์หรือการระบุอื่นใดเป็นพิเศษ เช่น 100 หมายถึง ข้อมูล 100 ในเลขฐานสิบ
- ตัวเลขฐานสอง (binary หรือ BIN) ประกอบด้วยข้อมูล 0 และ 1 ในการเขียนหรืออ้างถึงต้องใส่ เครื่องหมาย % นำหน้าตัวเลขเสมอ เช่น %00110001 หมายถึง ข้อมูล 00110001 ในเลขฐานสอง เท่ากับ 31 ในเลขฐานสิบหก หรือ 49 ในเลขฐานสิบ
- ตัวเลขฐานสิบหก (hexadecimal หรือ HEX) ประกอบด้วยข้อมูล 0 ถึง F เวลาเขียนต้องนำหน้าด้วย เครื่องหมาย $ เสมอ เช่น $11 หมายถึง ข้อมูล 11 ฐานสิบหก เท่ากับ 17 ในเลขฐานสิบ และ 0001 0001 ในเลขฐานสอง
- รหัสแอสกี้ (ASCII) เป็นรหัสของตัวอักษรและตัวเลขต่างๆ รวมถึงการเว้นวรรค การขึ้นต้นบรรทัดใหม่ ย่อหน้า ซึ่งเกิดขึ้นในทุกครั้งที่มีการพิมพ์ข้อความบนโปรแกรมคอมพิวเตอร์ โดยตัวอักษรและตัวเลขแต่ละตัวจะมีค่าหรือข้อมูลประจำตัวอยู่ เช่น ช่องว่างหรือเว้นวรรค รหัสแอสกี้คือ $20 ($ ที่อยู่หน้าตัวเลขเป็นการแจ้งให้ทราบว่า ตัวเลขที่ต่อจากนี้เป็นตัวเลขฐานสิบหก)ในการระบุว่าข้อความหรือตัวเลขต่อไปนี้เป็นรหัสแอสกี้ สามารถทำได้โดยเขียนข้อมูลให้อยู่ภายในเครื่องหมายคำพูดหรือ "..." เช่น "A" ข้อมูลที่ระบุนี้คือรหัสแอสกี้ของตัวอักษร A ซึ่งมีค่าเท่ากับ $65

2.) ลาเบล (label) คือการให้ข้อความตัวอักษรหรือสัญลักษณ์ แทนแอดเดรสของหน่วยความจำโปรแกรมที่ i-Stamp ต้องกระโดดไปทำงานยังตำแหน่งเหล่านั้น เพื่อหลีกเลี่ยงการระบุแอดเดรสด้วยการเขียนตัวเลขแอดเดรสของหน่วยความจำโปรแกรมลงในโปรแกรมพีเบสิก ซึ่งทำได้ยากและเกิดความผิดพลาดได้ง่าย ชื่อของลาเบลสามารถให้ขึ้นต้นด้วยตัวอักษรหรือเครื่องหมาย _ แต่ไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถขึ้นต้นด้วยตัวเลข และชื่อของลาเบลจะต้องไม่เป็นคำสงวน เช่น ชื่อของคำสั่ง (SERIN, toggle, GOTO) และ ชื่อของตัวแปร (port, w2, b13) เป็นต้น และจะต้องตามด้วยเครื่องหมาย : แต่เมื่อต้องการกำหนดให้กระโดดไปยังลาเบลที่ต้องการ การเขียนคำสั่งเพื่อกระโดดไปนั้นชื่อลาเบลไม่จำเป็นต้องมีเครื่องหมาย : อีก

3.) คอมเมนต์ (comment) เมื่อต้องการเพิ่มข้อความเพื่ออธิบายการทำงานของโปรแกรม ข้อความที่เพิ่มขึ้นมานี้เรียกว่า คอมเมนต์ จะต้องขึ้นต้นด้วยเครื่องหมาย ' ทุกครั้ง ข้อความที่อยู่หลังเครื่องหมาย ' จนจบบรรทัดจะเป็นคำอธิบาย หรือคอมเมนต์ทั้งหมด หรืออาจใช้คำสั่ง REM นำหน้าข้อความที่ไม่ต้องการให้คอมไพล์หรือแปลงค่า

4.) สัญลักษณ์ของการประมวลผลทางคณิตศาสตร์ ข้อความ ตัวย่อและสัญลักษณ์ต่างๆต่อไปนี้เป็นเครื่องหมายที่ใช้แทนการกระทำทางคณิตศาสตร์ภายในโปรแกรมภาษาพีเบสิก

- + แทนการบวก
- - แทนการลบ
- * แทนการคูณ (ส่งค่าผลลัพธ์ไบต์ด้านต่ำกลับมา)
- ** แทนการคูณ (ส่งค่าผลลัพธ์ไบต์ด้านสูงกลับมา)
- / แทนการหาร (ส่งค่าผลลัพธ์ที่ได้จากการหารกลับมา)
- // แทนการหาร (ส่งค่าเศษที่เหลือจากการหารกลับมา)
- min เก็บค่าตัวแปรที่มีค่ามากกว่าหรือเท่ากับค่าข้อมูลที่กำหนดหลังคำสั่งนี้
- max เก็บค่าตัวแปรที่มีค่าน้อยกว่าหรือเท่ากับค่าข้อมูลที่กำหนดหลังคำสั่งนี้
- & แทนการแอนด์ (AND) ทางลอจิก
- | (เป็นคีย์ระดับสองของคีย์ \ เลือกโดยการกด Shift แล้วตามด้วยคีย์ \) แทนการออร์ (OR) ทางลอจิก
- ^ แทนการเอ็กคลูซีฟออร์ (XOR) ทางลอจิก
- &/ แทนการแอนด์-นอต (AND NOT) หรือแนนด์ (NAND) ทางลอจิก
- |/ แทนการออร์-นอต (OR NOT) หรือนอร์ (NOR) ทางลอจิก
- ^/ แทนการเอ็กคลูซีฟออร์-นอต (XOR NOT) หรือเอ็กคลูซีฟนอร์ (XNOR) ทางลอจิก

2.3.3.2 รูปแบบการเขียนภาษาพีเบสิก

มีหลักเกณฑ์ที่สำคัญและไม่ซับซ้อนอยู่ 3 ประการคือ

1.) บรรทัดแรกของโปรแกรมต้องระบุรุ่นของเบสิกแสตมป์ที่ใช้งานเรียกว่า กำหนดไดเร็กตีฟ (directive) เช่น '{$STAMP BS2p}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) ในการเขียนโปรแกรมคำสั่งหรือตัวแปรต่าง ๆ จะเป็นตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็กก็ได้

3.) สามารถเขียนคำสั่งหลายๆ คำสั่งบนบรรทัดเดียวกันได้โดยแยกคำสั่งด้วยการใช้เครื่องหมาย : คั่น

2.3.3.3 รายละเอียดชุดคำสั่งของเบสิกแสตมป์

คำสั่งหลักในโปรแกรมพีเบสิกของ i-Stamp มีทั้งสิ้น 71 คำสั่ง โดยจะรวมหมดทั้งคำสั่งใช้งานทางฮาร์ดแวร์ 47 คำสั่ง, คำสั่งคำนวณทางคณิตศาสตร์และลอจิก 22 คำสั่ง, คำสั่งเทียมใช้ประกอบในการเขียนโปรแกรม 2 คำสั่ง แบ่งออกเป็นกลุ่มตามลักษณะการทำงานได้ 15 กลุ่ม ดังนี้

1.) กลุ่มคำสั่งกระโดดและควบคุมการทำงาน (Branching/Program control) มี 11 คำสั่ง คือ

- GOTO กระโดดไปยังแอดเดรสใดๆ
- IF...THEN เปรียบเทียบเงื่อนไขก่อนกระโดด
- IF...THEN...ELSE เปรียบเทียบเงื่อนไข 2 ทางเลือก
- BRANCH กระโดดไปยังตำแหน่งที่กำหนดตามค่าของตัวแปร
- ON...GOTO เลือกตำแหน่งของโปรแกรมที่จะกระโดดไปทำงาน
- GOSUB กระโดดไปยังโปรแกรมย่อย
- ON…GOSUB เลือกโปรแกรมย่อยที่จะกระโดดไปทำงาน
- RETURN กระโดดออกจากโปรแกรมย่อย
- RUN รันโปรแกรมที่ต้องการ เลือกได้ 8 โปรแกรม
- SELECT…CASE เลือกการตรวจสอบเงื่อนไขหลายๆเงื่อนไขในคราวเดียว
- STOP หยุดการทำงาน

2.) กลุ่มคำสั่งทำงานวนรอบ (Looping) มี 3 คำสั่ง คือ

- FOR...NEXT กำหนดจำนวนรอบที่ต้องการวนลูปหรือวนทำงานซ้ำ
- DO…LOOP กำหนดให้ทำงานวนลูปภายใต้เงื่อนไขในคราวเดียว
- EXIT ออกจากลูปแบบทันทีทันใด

3.) กลุ่มคำสั่งติดต่อหน่วยความจำอีอีพรอม (EEPROM access) มี 3 คำสั่ง คือ

- DATA เก็บข้อมูลลงหน่วยความจำอีอีพรอมก่อนดาวน์โหลดโปรแกรมพีเบสิก
- READ อ่านข้อมูลระดับไบต์จากหน่วยความจำอีอีพรอมมาเก็บไว้ในตัวแปร
- WRITE เขียนข้อมูลระดับไบต์ไปเก็บไว้ในหน่วยความจำอีอีพรอม

4.) กลุ่มคำสั่งติดต่อหน่วยความจำแรม (RAM access) มี 2 คำสั่ง คือ

- GET อ่านข้อมูลจากหน่วยความจำสแครตช์แพดแรมในตัวอินเตอร์พรีตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PUT เก็บข้อมูลลงหน่วยความจำสแครตช์แพดแรมในตัวอินเตอร์พรีตเตอร์

5.) กลุ่มคำสั่งจัดการตัวเลข (Numerics) มี 3 คำสั่ง คือ

- RANDOM สุ่มตัวเลข
- LOOKUP เปิดตารางข้อมูล
- LOOKDOWN ค้นหาตัวเลขที่เหมือนกัน แล้วเก็บค่าไว้ในตัวแปร RANDOM

6.) กลุ่มคำสั่งจัดการอินพุตเอาต์พุตดิจิตอล (Digital I/O) มี 11 คำสั่ง คือ

- INPUT กำหนดให้ทำงานเป็นอินพุต
- OUTPUT กำหนดให้ทำงานเป็นเอาต์พุต
- HIGH ทำให้ขาเอาต์พุตเป็นลอจิก "1"
- LOW ทำให้ขาเอาต์พุตเป็นลอจิก "0"
- TOGGLE ทำให้ขาเอาต์พุตกลับสถานะลอจิก
- REVERSE เปลี่ยนจากขาอินพุตเป็นเอาต์พุตหรือจากเอาต์พุตเป็นอินพุต
- PULSIN วัดสัญญาณพัลส์อินพุต (ความละเอียด 0.8 ไมโครวินาที)
- PULSOUT ส่งสัญญาณพัลส์ออก (ความละเอียด 0.8 ไมโครวินาที)
- BUTTON ตรวจสอบการรับค่าสวิตช์
- COUNT นับจำนวนรอบของสัญญาณอินพุต (มีค่า 0 ถึง 312.5kHz)
- XOUT กำเนิดรหัสควบคุมสำหรับอุปกรณ์ต่อพ่วงอนุกรม X-10

7.) กลุ่มคำสั่งจัดการอินพุตเอาต์พุตอนุกรมแบบอะซิงโครนัส (Asynchronous I/O) มี 2 คำสั่ง คือ

- SERIN รับข้อมูลอนุกรมในรูปแบบ N81 หรือ E71 เข้ามาทางขา S_{IN}
- SEROUT ส่งข้อมูลอนุกรมในรูปแบบ N81 หรือ E71 ออกไปทางขา S_{OUT}

8.) กลุ่มคำสั่งจัดการอินพุตเอาต์พุตอนุกรมแบบซิงโครนัส (Synchronous I/O) มี 2 คำสั่ง คือ

- SHIFTIN เลื่อนข้อมูลเข้าจากแบบขนานเป็นแบบอนุกรม
- SHIFTOUT ส่งข้อมูลออกจากแบบอนุกรม I^2C_{IN}

9.) กลุ่มคำสั่งพอร์ตสัญญาณอะนาลอก (Analog I/O) มี 2 คำสั่ง คือ

- PWM สร้างสัญญาณ PWM ขนาด 8 บิต ออกไปทางขาพอร์ต
- RCTIME วัดค่าเวลาการประจุและคายประจุของวงจร RC นำไปใช้วัดค่าของตัวต้านทานได้

10.) กลุ่มคำสั่งจัดการด้านเวลาและความถี่ (Timing & Frequency) มี 3 คำสั่ง คือ

- PAUSE หน่วงเวลา 0 ถึง 65,536 มิลลิวินาที
- FREQOUT กำเนิดสัญญาณหนึ่งไซน์หรือสองความถี่ ตั้งแต่ 0 - 82.917kHz
- DTMFOUT กำเนิดสัญญาณ DTMF ของระบบโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.) กลุ่มคำสั่งตรวจสอบการทำงานของโปรแกรม (Program debugging) มี 2 คำสั่ง คือ

- DEBUG แสดงค่าของตัวแปรผ่านทางคอมพิวเตอร์
- DEBUGIN รับค่าของตัวแปรกลับจากคอมพิวเตอร์ผ่านพอร์ตอนุกรม

12.) กลุ่มคำสั่งควบคุมพลังงาน (Power control) มี 3 คำสั่ง คือ

- NAP หยุดทำงานในช่วงเวลาสั้น ๆ การกินพลังงานจะลดลง
- END ทำงานในโหมดสลีป จนกว่าจะมีการจ่ายไฟหรือต่อเบสิกแสตมป์เข้ากับคอมพิวเตอร์ เมื่อทำคำสั่งนี้จะกินกระแสไฟฟ้าเพียง 50μA
- SLEEP ทำงานในโหมดประหยัดพลังงาน

13.) กลุ่มคำสั่งเทียมสำหรับใช้ประกอบการเขียนโปรแกรม มี 2 คำสั่ง คือ

- CON ใช้กำหนดค่าคงที่
- VAR ใช้กำหนดตัวแปร

14.) กลุ่มคำสั่งประมวลผลทางคณิตศาสตร์ มี 16 คำสั่ง คือ

- + คำสั่งการบวก
- - คำสั่งการลบ
- * คำสั่งการคูณ โดยส่งค่าผลลัพธ์ด้านต่ำกลับมา
- ** คำสั่งการคูณ โดยส่งค่าผลลัพธ์ด้านสูงกลับมา
- / คำสั่งการหาร โดยส่งค่าผลลัพธ์ที่ได้จากการหารกลับมา
- // คำสั่งการหาร โดยส่งค่าเศษที่เหลือจากการหารกลับมา
- */ คำสั่งคูณเลขทศนิยม
- >> คำสั่งเลื่อนข้อมูลไปทางขวา มีค่าเท่ากับหารด้วย 2
- << คำสั่งเลื่อนข้อมูลไปทางซ้าย มีค่าเท่ากับคูณด้วย 2
- MIN เก็บค่าตัวแปรที่มีค่ามากกว่าหรือเท่ากับค่าข้อมูลที่กำหนด
- MAX เก็บค่าตัวแปรที่มีค่าน้อยกว่าหรือเท่ากับค่าข้อมูลที่กำหนด
- ABS ประมวลผลเฉพาะข้อมูลตัวเลขแบบไม่คิดเครื่องหมาย
- SQR ถอดรากที่สอง
- SIN คำนวณค่า SINE ทางตรีโกณมิติ
- COS คำนวณค่า COSINE ทางตรีโกณมิติ
- DIG เรียกค่าของข้อมูลในหลักที่กำหนดในรูปของเลขฐานสิบ

15.) กลุ่มคำสั่งประมวลผลทางลอจิก มีทั้งสิ้น 6 คำสั่ง คือ

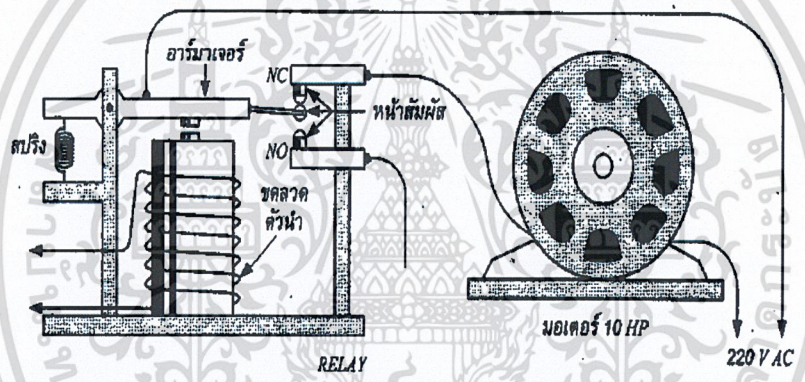
- & แอนด์ (AND) ทางลอจิก
- | ออร์ (OR) ทางลอจิก
- ^ เอ็กคลูซีฟออร์ (XOR) ทางลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- REV สลับบิตข้อมูล
- DCD เลือกเซตบิตที่ต้องการของข้อมูลขนาด 16 บิต
- NCD แสดงบิตนัยสำคัญของข้อมูลขนาด 16 บิต

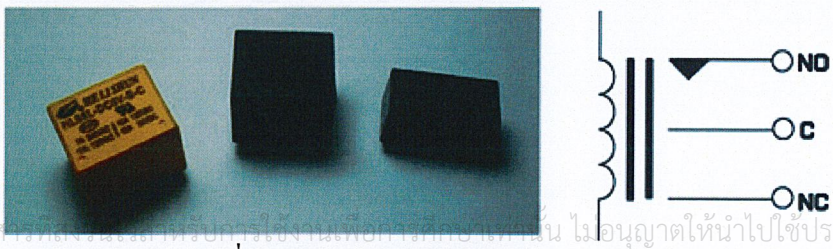
2.3.4 รีเลย์อิเล็กทรอนิกส์

รีเลย์ คือ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่ตัด-ต่อวงจร คล้ายกับสวิตซ์ เป็นอุปกรณ์ควบคุมวงจรไฟฟ้าที่มีการทำงานในลักษณะเป็นเครื่องกลไฟฟ้า นิยมใช้ในวงจรควบคุมแบบต่างๆ กันอย่างแพร่หลาย โดยโครงสร้างพื้นฐานและการทำงานของรีเลย์จะประกอบไปด้วยขดลวดตัวนำและแกนโลหะที่สามารถเคลื่อนที่ขึ้นลงได้ เรียกว่า อาร์มาเจอร์ ซึ่งมีหน้าที่เปิดปิดหน้าสัมผัสของรีเลย์ และรีเลย์จะเริ่มทำงานได้เมื่อมีกระแสไฟฟ้าไหลผ่านไปที่ขดลวดตัวนำ ทำให้เกิดสนามแม่เหล็กไปดึงดูดแกนของอาร์มาเจอร์ ถ้าแรงดึงดูดที่เกิดจากสนามแม่เหล็กสามารถชนะแรงดึงของสปริงได้ ก็จะดึงแกนของอาร์มาเจอร์ให้หน้าสัมผัสของรีเลย์มาอยู่ในตำแหน่งอีกทางหนึ่ง



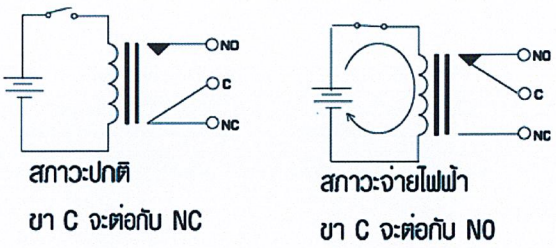
รูปที่ 2.14 ส่วนประกอบภายในรีเลย์

รีเลย์จะมีหน้าสัมผัสอยู่สองแบบ คือ แบบปกติเปิดและแบบปกติปิด รีเลย์แบบปกติเปิดหน้าสัมผัสของรีเลย์จะเปิดเมื่อไม่มีกระแสไฟฟ้าไหลผ่านไปยังขดลวดของรีเลย์และหน้าสัมผัสจะปิดเมื่อมีกระแสไฟฟ้าไหลผ่านไปที่ขดลวดของรีเลย์ ซึ่งการทำงานก็จะตรงกันข้ามกันในรีเลย์แบบปกติปิด รีเลย์มีหลายชนิดด้วยกัน โดยมากรีเลย์จะถูกนำมาใช้ในวงจรการควบคุมมอเตอร์ไฟฟ้าในลักษณะของการหน่วงเวลาเพื่อทำให้เกิดการทำงานของวงจรควบคุมเป็นไปตามลำดับหรือใช้เพื่อป้องกันอันตรายที่จะเกิดขึ้นภายในวงจรควบคุม



รูปที่ 2.15 รีเลย์ และ สัญลักษณ์ของรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

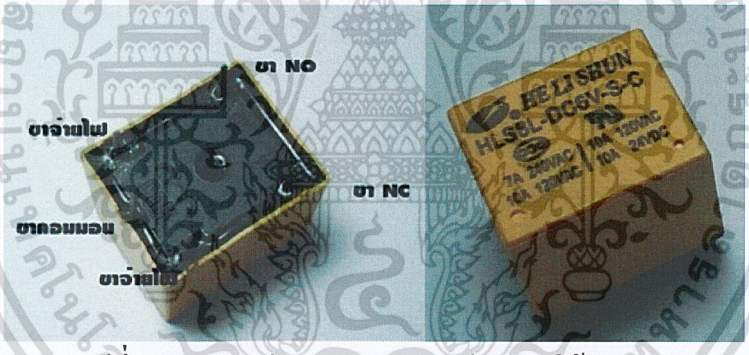


รูปที่ 2.16 สภาวะการทำงานของรีเลย์

ข้อคำนึงถึงในการใช้งานรีเลย์ทั่วไป

1.) แรงดันใช้งาน หรือแรงดันที่ทำให้รีเลย์ทำงานได้ หากดูที่ตัวรีเลย์จะระบุค่าแรงดันใช้งานไว้ เช่น 12V DC คือต้องใช้แรงดันที่ 12V DC เท่านั้น ถ้าใช้มากกว่านี้ขดลวดภายในตัวรีเลย์อาจขาดได้ หรือใช้แรงดันต่ำกว่ามาก รีเลย์จะไม่ทำงาน ส่วนในการต่อวงจรนั้นสามารถต่อขั้วใดก็ได้ เพราะตัวรีเลย์จะไม่ระบุขั้วต่อไว้

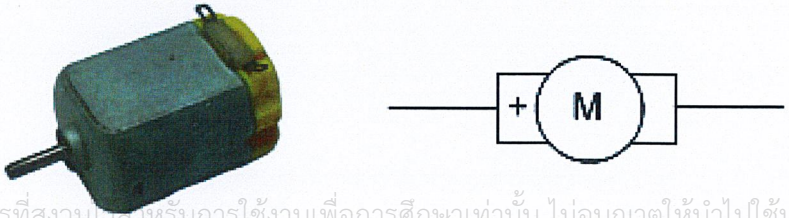
2.) การใช้งานกระแสผ่านหน้าสัมผัส ซึ่งที่ตัวรีเลย์จะระบุไว้ เช่น 10A 220VAC คือ หน้าสัมผัสของรีเลย์นั้นสามารถทนกระแสได้ 10 แอมแปร์ที่ 220V AC แต่ควรใช้งานที่ระดับกระแสต่ำกว่านี้ เพราะถ้ากระแสมากหน้าสัมผัส ของรีเลย์จะละลายเสียหายได้



รูปที่ 2.17 ตำแหน่งขา และ รายละเอียดการใช้งาน

2.3.5 มอเตอร์กระแสตรง

มอเตอร์ เป็นอุปกรณ์ที่ทำหน้าที่เปลี่ยนแปลงพลังงานไฟฟ้าให้เป็นพลังงานกล (หมุน) มอเตอร์โดยทั่วไปแบ่งเป็น 2 ชนิดคือ มอเตอร์กระแสตรง และ มอเตอร์ไฟฟ้ากระแสสลับ แต่ในที่นี้จะขอกล่าวถึงมอเตอร์ไฟฟ้ากระแสตรงขนาดเล็กที่ใช้ในปัญหาพิเศษ

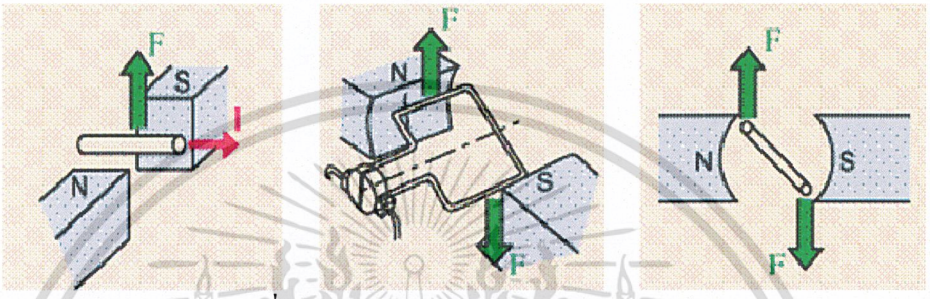


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.18 มอเตอร์กระแสตรงขนาดเล็กและสัญลักษณ์มอเตอร์กระแสตรง

2.3.5.1 หลักการทำงานมอเตอร์

มอเตอร์จะทำงานตามหลักการแม่เหล็กไฟฟ้าเมื่อเรานำเอาขดลวดพันแกนเหล็กหลายรอบแล้วป้อนกระแสไฟฟ้าเข้าไปในขดลวดนั้นแกนเหล็กจะกลายเป็นแม่เหล็กชั่วคราว และเมื่อเรานำไฟฟ้าออก แกนเหล็กจะกลายเป็นแกนเหล็กธรรมดา กล่าวคือ เมื่อแม่เหล็ก 2 อันมีขั้วต่างกันจะผลักกัน ดังนั้นในตัวมอเตอร์เองจึงประกอบด้วยเหล็กมีขั้วต่างกัน เมื่อเราป้อนพลังงานให้กับมอเตอร์จะเกิดเป็นแม่เหล็กไฟฟ้า และจะมีขั้วต่างกันกับแม่เหล็กที่ขั้ว จึงเกิดการพลักกันทำให้ขดลวดตัวนำหมุนเคลื่อนที่ได้

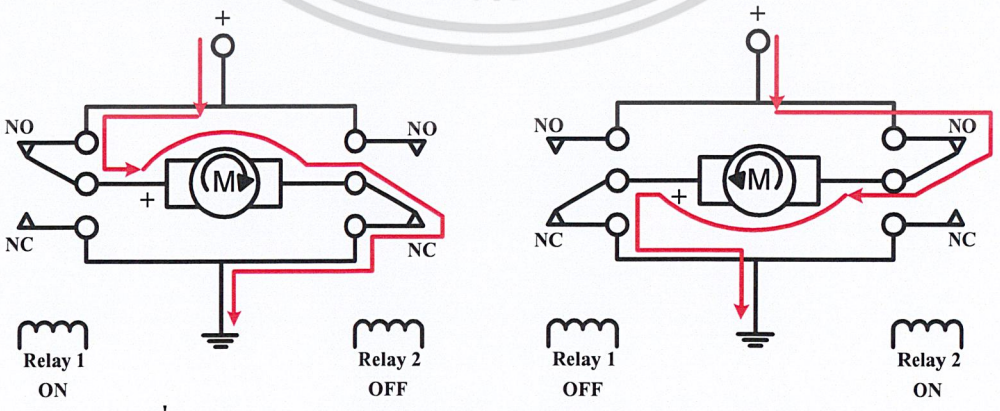


รูปที่ 2.19 ลักษณะการทำงานของมอเตอร์

2.3.5.2 การนำรีเลย์ควบคุมทิศทางหมุนของมอเตอร์กระแสตรง

ในการใช้ไอซีไมโครคอนโทรเลอร์เป็นตัวควบคุมการหมุน และทิศทางของมอเตอร์กระแสตรงนั้น เราจะต้องมีส่วนของวงจร ที่เรียกว่าวงจรขับมอเตอร์ (Driver) ในส่วนของวงจรกลับทิศทางของมอเตอร์นั้น สามารถที่จะใช้รีเลย์ต่อวงจรสวิตช์เพื่อกลับทิศทางของขั้วไฟกระแสตรง

จากรูปเป็นการใช้รีเลย์ควบคุมการเปลี่ยนทิศทางการหมุนของมอเตอร์ โดยการควบคุมการปิด - เปิดที่รีเลย์ 2 ตัว ซึ่งจะทำหน้าที่กลับทิศทางของขั้วไฟที่ป้อนให้กับมอเตอร์ โดยการสลับการทำงานของรีเลย์ เช่นให้รีเลย์ตัวที่ 1 ทำงาน (ON) และรีเลย์ตัวที่ 2 หยุดทำงาน (OFF) จะทำให้มอเตอร์หมุนไปทางซ้าย และในทำนองเดียวกันถ้าหากรีเลย์ตัวที่ 1 หยุดทำงาน (OFF) และรีเลย์ตัวที่ 2 ทำงาน (ON) ก็จะทำให้มอเตอร์หมุนไปทางขวา



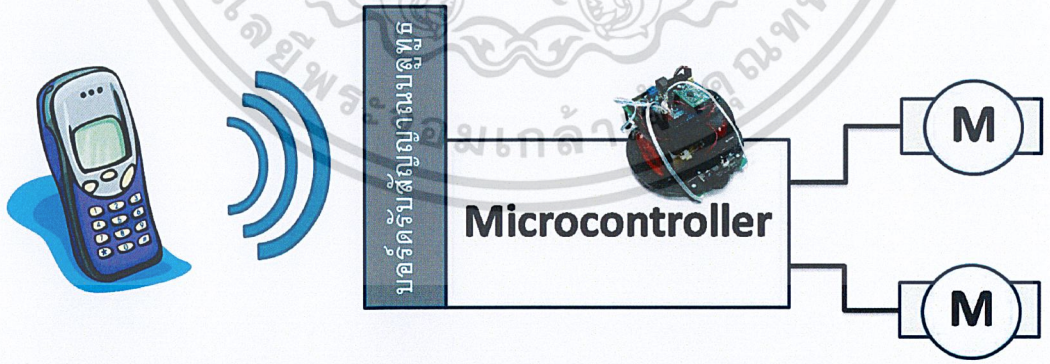
รูปที่ 2.20 แสดงการกลับทิศทางของมอเตอร์กระแสตรงโดยใช้รีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการพัฒนา

โครงงานปัญหาพิเศษนี้ได้นำเทคโนโลยี J2ME และเทคโนโลยีบลูทูธมาพัฒนาเข้าด้วยกัน โดยทำเป็นแอพพลิเคชั่นบนโทรศัพท์มือถือ มาทำการควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ขนาดเล็กผ่านเทคโนโลยีบลูทูธ เมื่อกดปุ่มบนโทรศัพท์มือถือ จะมีการส่งสัญญาณบลูทูธไปยังบอร์ดไมโครคอนโทรลเลอร์ เพื่อให้โปรแกรมบนบอร์ดไมโครคอนโทรเลอร์ทำการประมวลผลแล้วควบคุมการทำงานของรีเลย์ที่ต่อเข้ากับมอเตอร์ และเมื่อมอเตอร์ทำงาน หุ่นยนต์จะเคลื่อนที่ไปตามทิศทางที่กำหนด

การพัฒนาได้แบ่งออกเป็น 2 ส่วนคือ ส่วนโปรแกรมบนโทรศัพท์มือถือ และส่วนของโปรแกรมบนไมโครคอนโทรลเลอร์ โดยการพัฒนาในส่วนของโปรแกรมบนโทรศัพท์มือถือจะใช้ J2ME ซึ่งเป็นหนึ่งในรูปแบบของภาษาจาวา และใช้เทคโนโลยี jsr-82 Java Bluetooth API เป็นเครื่องมือในการพัฒนาโปรแกรม และส่วนของไมโครคอนโทรลเลอร์ซึ่งเป็นส่วนที่ใช้ควบคุมการหมุนของมอเตอร์ทำให้หุ่นยนต์ขนาดเล็กสามารถเคลื่อนที่ได้ จะใช้ภาษาพีเบสิกในการพัฒนา ในการทำงานนั้นเริ่มจากผู้ใช้ทำการเชื่อมต่อโทรศัพท์มือถือเข้ากับบอร์ดรับสัญญาณบลูทูธ เมื่อยืนยันรหัสผ่านเรียบร้อยแล้วจะแสดงหน้าจอเมนูหลัก ผู้ใช้เลือกคำสั่งบนโทรศัพท์มือถือเพื่อใช้ในการควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก โดยส่งคำสั่งผ่านสัญญาณบลูทูธไปยังบอร์ดไมโครคอนโทรลเลอร์ หลังจากนั้นบอร์ดไมโครคอนโทรลเลอร์จะทำการประมวลผลเพื่อควบคุมรีเลย์ เพื่อไปควบคุมให้มอเตอร์หมุนตามคำสั่งที่ได้รับมาจากโทรศัพท์มือถือ แสดงได้ดังรูป



รูปที่ 3.1 ระบบควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก

อุปกรณ์ที่ใช้ทำโครงงานพิเศษนี้ ประกอบด้วยอุปกรณ์หลักทั้งหมดจำนวน 7 ชิ้น ดังนี้

1.) แผงวงจรสำหรับพัฒนาโครงงาน JX-2SX Plus II

2.) บอร์ดไมโครคอนโทรลเลอร์ i-Stamp 2P24

3.) แผงวงจรขับรีเลย์ 4 ช่อง (Relay 4i)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.) แผงวงจรสื่อสารข้อมูลไร้สายผ่านโมดูลบลูทูธแบบสเลฟ (ZX-Bluetooth)

5.) มอเตอร์ไฟตรงพร้อมชุดเฟืองขับ

6.) ถ่านขนาด 1.5 V 12 ก้อน

7.) โทรศัพท์มือถือ

3.1 ส่วนโปรแกรมบนโทรศัพท์มือถือที่ใช้ควบคุมทิศทางของหุ่นยนต์ขนาดเล็ก

โปรแกรมนี้สามารถแบ่งออกได้เป็น 2 ส่วนหลักได้แก่ ส่วน interface บนโทรศัพท์มือถือที่ติดต่อกับผู้ใช้ เพื่อเลือกคำสั่งในการควบคุมหุ่นยนต์ขนาดเล็ก และส่วนบลูทูธที่ใช้ส่งสัญญาณไปยังบอร์ดรับสัญญาณบลูทูธ

3.1.1 ส่วน Interface

ส่วน interface เป็นส่วนติดต่อกับผู้ใช้ ซึ่งจะใช้ J2ME โดยจะแสดงรายการปุ่มที่ใช้ในการควบคุมทิศทางของหุ่นยนต์ขนาดเล็ก การสร้าง interface จะใช้ MIDlet และ canvas ในการสร้างเพื่อติดต่อกับผู้ใช้ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก

- J2ME หรือ Java 2 Micro Edition มีจุดมุ่งหมายสำหรับอุปกรณ์ที่มีขนาดเล็กซึ่งมีทรัพยากรจำกัดทางด้านพลังงาน ความสามารถในการประมวลผล และหน่วยความจำ เช่น เพจเจอร์ โทรศัพท์มือถือ ปาล์ม พีดีเอ เป็นต้น นอกจากนี้ J2ME ยังสามารถใช้พัฒนาแอพพลิเคชั่นให้ทำงานบนอุปกรณ์ที่ไม่ได้เป็นอุปกรณ์ไร้สายได้อีกด้วย ถ้าไม่มีข้อจำกัดเพื่อดาวน์โหลดและติดตั้งซอฟต์แวร์เช่น กล่องรับสัญญาณดาวเทียมสำหรับโทรทัศน์ อินเตอร์เน็ตทีวี เป็นต้น ในการออกแบบโปรแกรม J2ME มีการแบ่งออกเป็นเอดิชันย่อยก็เพื่อความเหมาะสมกับการเลือกไปพัฒนาแอพพลิเคชั่นตามขนาดและลักษณะของงาน ซึ่งอาจจะแบ่งแยกออกเป็นกลุ่มๆ เหมือนกับเป็นการจัดแบ่งกลุ่มของคลาสและแพ็คเกจของแต่ละเอดิชัน
- canvas เป็น class หนึ่งใน java วัตถุประสงค์คือเอาไว้สำหรับการวาด display ต่างๆ ลงบนหน้าจอและจัดการ low-level event ต่างๆ เช่น การกด key ต่างๆ บนโทรศัพท์มือถือหรือการตรวจรับ command
- MIDlet เป็น class หนึ่งใน java แบบ high-level ซึ่งจะมีการกำหนดรูปแบบการแสดงผลไว้แล้ว ในการเขียนโปรแกรมจะเป็นการเรียกใช้รูปแบบต่างๆ ที่ถูกกำหนดไว้แล้วมาแสดงโดยไม่ต้องกำหนดค่าต่างๆ ในการแสดงผลทั้งหมด ซึ่งโปรแกรมที่ได้เมื่อนำไปใช้งานในโทรศัพท์มือถือจะมีการแสดงผลที่ใกล้เคียงกัน

เริ่มต้นการทำงานคือการตรวจสอบรหัสผ่านของผู้ใช้ เมื่อเข้าสู่แอพพลิเคชั่นในหน้าแรกผู้ใช้จะต้องใส่รหัสผ่าน ซึ่งโปรแกรมบนโทรศัพท์มือถือจะทำการตรวจสอบรหัสผ่านที่ผู้ใช้ป้อนเข้ามาว่าถูกต้องหรือไม่ ถ้ารหัสผ่านที่ป้อนเข้ามาถูกต้อง ผู้ใช้ก็จะสามารถเข้าใช้งานแอพพลิเคชั่นนั้นได้ ในกรณีที่ผู้ใช้ใส่รหัสผ่านไม่ถูกต้องครบตามที่กำหนด แอพพลิเคชั่นจะถูกปิดลงโดยอัตโนมัติ

เอกสาร... ไม่ว่า... ใช้...

ซึ่งรหัสผ่านจะให้มาพร้อมกับชุดตัวบอร์ดรับสัญญาณบลูทูธ ดังนั้นในกรณีที่ผู้ใช้ลืมรหัสผ่าน ผู้ใช้สามารถตรวจสอบรหัสได้ที่บอร์ดรับสัญญาณบลูทูธ และเป็นการป้องกันการนำโทรศัพท์มือถือไปใช้งานโดยบุคคลอื่น ซึ่งการตรวจสอบรหัสผ่านของแอพพลิเคชั่นนั้น โทรศัพท์มือถือจะรับรหัสผ่านที่ผู้ใช้ป้อนเข้ามา และส่งค่าไปยังบอร์ดรับสัญญาณบลูทูธ เพื่อนำรหัสผ่านไปตรวจสอบกับบอร์ดไมโครคอนโทรเลอร์ว่ารหัสผ่านที่ผู้ใช้ป้อนเข้ามานั้นถูกต้อง และตรงกันหรือไม่

3.1.2 ส่วนบลูทูธ

ส่วนนี้จะแบ่งการทำงานออกเป็น 3 ส่วนคือ

3.1.2.1 ส่วนการค้นหาอุปกรณ์ (Device Discovery)

ส่วนนี้จะเริ่มทำงานโดยอัตโนมัติเมื่อโปรแกรมถูกเรียกขึ้นมา ระบบจะทำการตรวจสอบสถานะของอุปกรณ์บลูทูธในเครื่อง จากนั้นจะทำการค้นหาอุปกรณ์บลูทูธ (บอร์ดรับสัญญาณบลูทูธ) ตามที่กำหนดไว้ที่อยู่ในรัศมีทำการ 10 เมตรและทำการเชื่อมต่อ ในกรณีที่ไม่พบอุปกรณ์บลูทูธที่กำหนดไว้โปรแกรมจะไม่ทำงานในขั้นตอนต่อไป โดยจะใช้คำสั่ง

```
con = (StreamConnection)Connector.open(conStr);
```

เพื่อเชื่อมต่อกับอุปกรณ์บลูทูธ ซึ่ง conStr คือ Address ของอุปกรณ์บลูทูธที่จะทำการเชื่อมต่อ

3.1.2.2 ส่วนการส่งรหัสคำสั่ง

ส่วนนี้จะทำงานเมื่อผู้ใช้กดตัวเลือกที่ต้องการบนโทรศัพท์มือถือ ที่เป็นคำสั่งที่ใช้ควบคุมทิศทางของหุ่นยนต์ขนาดเล็ก โทรศัพท์มือถือจะทำการส่งคำสั่งนั้นไปยังอุปกรณ์บลูทูธที่ได้ทำการเชื่อมต่อไว้แล้ว ซึ่งในที่นี้คือบอร์ดรับสัญญาณบลูทูธที่เชื่อมต่อไว้กับบอร์ดไมโครคอนโทรลเลอร์ โดยโปรแกรมจะทำการแปลงหมายเลขที่ผู้ใช้เลือก เป็นรหัส ASCII เพื่อส่งไปยังบอร์ดรับสัญญาณบลูทูธ โดยจะใช้คำสั่ง

```
out = con.openOutputStream();
outwrite(Command);
out.close();
```

เพื่อส่งค่าสถานะให้บอร์ดไมโครคอนโทรลเลอร์ โดย

- คำสั่ง con.openOutputStream(); เป็นการเตรียมช่องทางการติดต่อก่อนจะทำการส่ง
- คำสั่ง outwrite(Command); เป็นการส่งคำสั่งไปยังปลายทาง ในที่นี้คือบอร์ดไมโครคอนโทรลเลอร์ โดย Command คือรหัส ASCII ของค่าที่จะส่ง
- คำสั่ง out.close(); เป็นการปิดช่องทางการเชื่อมต่อที่ใช้ส่งข้อมูลเพื่อลดการใช้ทรัพยากรของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.3 ส่วนการรับรหัสคำสั่ง

ส่วนนี้จะทำงานหลังจากโทรศัพท์มือถือทำการส่งคำสั่งไปยังบอร์ดไมโครคอนโทรลเลอร์แล้ว โดยจะรอรับผลการทำงานตอบกลับจากไมโครคอนโทรลเลอร์ ซึ่งรับคำสั่งเป็นรหัสแบบ ASCII จากนั้นจะถูกแปลงมาเป็นข้อความเพื่อแสดงที่หน้าจอ ซึ่งจะใช้คำสั่ง

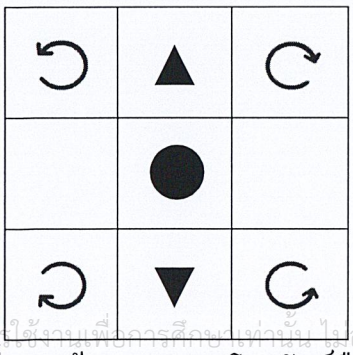
```
in = con.openInputStream();
byte[] buffer = new byte[80];
int bufferread = in.read(buffer);
in.close();
status = new String(buffer, 0, buffer read);
```

เพื่อรับค่าสถานะจากบอร์ดไมโครคอนโทรลเลอร์โดย

- คำสั่ง con.openInputStream(); เป็นการเปิดช่องทางการติดต่อเพื่อรับข้อมูล
- คำสั่ง in.read(buffer); เป็นการรอรับข้อมูลประเภทไบต์ที่มีขนาดข้อมูลเท่ากับ buffer
- คำสั่ง in.close(); เป็นการปิดช่องทางการเชื่อมต่อที่ใช้รอรับข้อมูลเพื่อลดการใช้ทรัพยากรของเครื่อง

3.1.3 สรุปการทำงานของโปรแกรมบนโทรศัพท์มือถือ

โปรแกรมทำการค้นหาอุปกรณ์บลูทูธที่จะทำการเชื่อมต่อ ในกรณีที่ไม่พบอุปกรณ์ที่จะทำการเชื่อมต่อจะแสดงหน้าจอว่า “ไม่พบอุปกรณ์บลูทูธที่จะทำการเชื่อมต่อ” และจะมีคำสั่งให้ผู้ใช้เลือกทำการค้นหาต่อหรือออกจากโปรแกรม ในกรณีที่พบก็จะทำการเชื่อมต่อกับอุปกรณ์บลูทูธและแสดงหน้าจอให้ผู้ใช้ใส่รหัสผ่าน 4 หลัก เมื่อผู้ใช้ใส่รหัสผ่าน 4 หลักและกดตกลง โปรแกรมจะทำการส่งรหัสผ่านไปตรวจสอบที่บอร์ดไมโครคอนโทรลเลอร์ ถ้ารหัสผ่านไม่ถูกต้องบอร์ดไมโครคอนโทรลเลอร์จะส่งค่าจำนวนครั้งที่สามารถใส่รหัสผ่านได้กลับไปที่โทรศัพท์มือถือ ซึ่งถ้าจำนวนครั้งที่สามารถใส่รหัสผ่านได้มีค่าเป็น 0 โปรแกรมจะแสดงหน้าจอให้ออกจากโปรแกรมโดยอัตโนมัติ แต่ถ้าใส่รหัสผ่านถูกต้องโปรแกรมบนโทรศัพท์มือถือจะแสดงหน้าจอตัวเลือกที่ใช้ในการควบคุมทิศทางของหุ่นยนต์ขนาดเล็ก ซึ่งจะมีรูปแบบการกดดังแสดงในรูปที่ 3.2



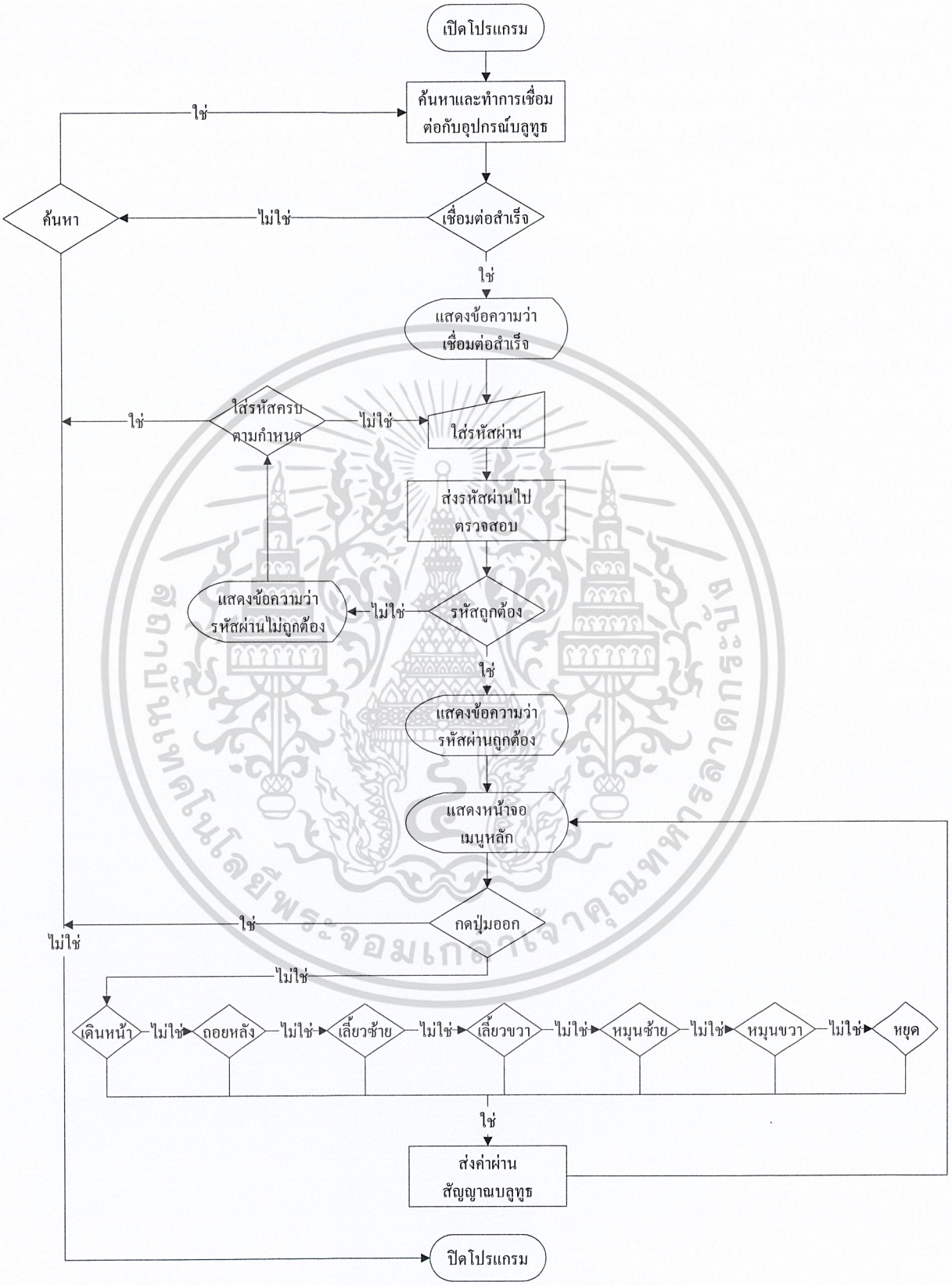
รูปที่ 3.2 แป้นควบคุมบนโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ผู้ใช้เลือกเมนูออกจากโปรแกรม โปรแกรมจะถามยืนยันว่าต้องการจะออกจากโปรแกรมใช่หรือไม่ เมื่อผู้ใช้กดใช่เพื่อออกจากโปรแกรม โปรแกรมจะส่งค่า 0 ไปยังบอร์ดไมโครคอนโทรลเลอร์เพื่อให้บอร์ดไมโครคอนโทรลเลอร์อยู่ในสถานะรอรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมบนโทรศัพท์มือถือ สามารถแสดง Flow chart การทำงานได้ดังรูป



รูปที่ 3.3 Flow chart การทำงานของโปรแกรมบนโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ส่วนโปรแกรมบนไมโครคอนโทรลเลอร์

ส่วนนี้ได้ทำการติดต่อกับบอร์ดรับสัญญาณบลูทูธ โดยทำการรับข้อมูลคำสั่งที่ส่งมาจากโทรศัพท์มือถือ ซึ่งเป็นรหัส ASCII โดยไมโครคอนโทรลเลอร์นั้นจะนำข้อมูลที่รับมาประมวลผลและส่งเป็นคำสั่งไปควบคุมการหมุนของมอเตอร์เพื่อให้หุ่นยนต์ขนาดเล็กเคลื่อนที่ ซึ่งใช้ภาษาพี-เบสิกในการพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์ โดยการเขียนโค้ดนั้นจะประกอบไปด้วย 2 ส่วนหลัก คือ

3.2.1 ส่วนของการตรวจสอบรหัสผ่าน

ในส่วนการตรวจสอบรหัสผ่าน ไมโครคอนโทรลเลอร์จะทำการรอรับรหัสผ่านจากโทรศัพท์มือถือ บอร์ดจะไม่ทำคำสั่งอื่นจนกว่าจะได้รับรหัสผ่านและตรวจสอบว่าถูกต้องแล้ว โดยจะส่งค่าผลลัพธ์การตรวจสอบไปให้โทรศัพท์มือถือ ในกรณีที่รหัสผ่านไม่ถูกต้อง บอร์ดจะส่งจำนวนครั้งที่สามารถใส่รหัสผ่านได้ไปยังโทรศัพท์มือถือ โทรศัพท์มือถือจะทำการเช็คค่าว่าใส่รหัสผ่านไม่ถูกต้องครบตามจำนวนที่กำหนดหรือยัง ถ้าครบแล้วก็จะแสดงหน้าจอให้ออกจากโปรแกรม แต่ในกรณีที่ใส่รหัสผ่านถูกต้อง บอร์ดจะส่งค่า p กลับไปยังโทรศัพท์มือถือ เพื่อให้โทรศัพท์มือถือทราบว่าขณะนี้หุ่นยนต์ขนาดเล็กพร้อมใช้งานแล้ว

3.2.2 ส่วนของการควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก

กรณีที่ 1 เมื่อผู้ใช้ต้องการออกจากโปรแกรม ผู้ใช้สามารถกดปุ่มออกบนโทรศัพท์มือถือ ซึ่งจะส่งค่า 0 ไปยังบอร์ดไมโครคอนโทรลเลอร์เพื่อออกจากโปรแกรมได้เลย

กรณีที่ 2 เมื่อผู้ใช้ต้องการควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก ผู้ใช้ทำการกดหมายเลขบนโทรศัพท์มือถือ โทรศัพท์มือถือจะส่งคำสั่งไปยังบอร์ดไมโครคอนโทรลเลอร์ผ่านสัญญาณบลูทูธ เมื่อบอร์ดไมโครคอนโทรลเลอร์ได้รับคำสั่งแล้ว จะสั่งการรีเลย์เพื่อควบคุมให้มอเตอร์หมุนไปตามทิศทางที่กำหนด

3.2.3 สรุปการทำงานของโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์

หลังจากที่เขียนโปรแกรมลงบนบอร์ดไมโครคอนโทรลเลอร์และสั่งให้ทำงานแล้ว บอร์ดไมโครคอนโทรลเลอร์จะอยู่ในสถานะรอรับรหัสผ่านจากโทรศัพท์มือถือ เมื่อได้รับการเชื่อมต่อและได้รับรหัสผ่านจากโทรศัพท์มือถือแล้ว บอร์ดไมโครคอนโทรลเลอร์จะทำการตรวจสอบรหัสผ่านนั้น ถ้ารหัสผ่านไม่ถูกต้อง จะส่งค่าจำนวนครั้งที่สามารถใส่รหัสผ่านได้กลับไปยังโทรศัพท์มือถือ ถ้ารหัสผ่านถูกต้องจะส่งค่า p กลับไปยังโทรศัพท์มือถือเพื่อแจ้งให้ทราบว่าบอร์ดไมโครคอนโทรลเลอร์อยู่ในสถานะพร้อมใช้งาน และรอการกดหมายเลขจากโทรศัพท์มือถือ เมื่อผู้ใช้ทำการกดหมายเลขบนโทรศัพท์มือถือ จะเป็นการส่งคำสั่งเป็นรหัส ASCII ไปยังบอร์ดไมโครคอนโทรลเลอร์เพื่อควบคุมการหมุนของมอเตอร์ทั้งสองตัว เมื่อกดตัวเลือกออกจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมบนโทรศัพท์มือถือ บอร์ดไมโครคอนโทรลเลอร์จะได้รับค่า 0 จากโทรศัพท์มือถือแล้วจะออกจากการทำงาน และอยู่ในสถานะรอรับรหัสผ่านเพื่อรอการเชื่อมต่อจากโทรศัพท์มือถืออีกครั้ง

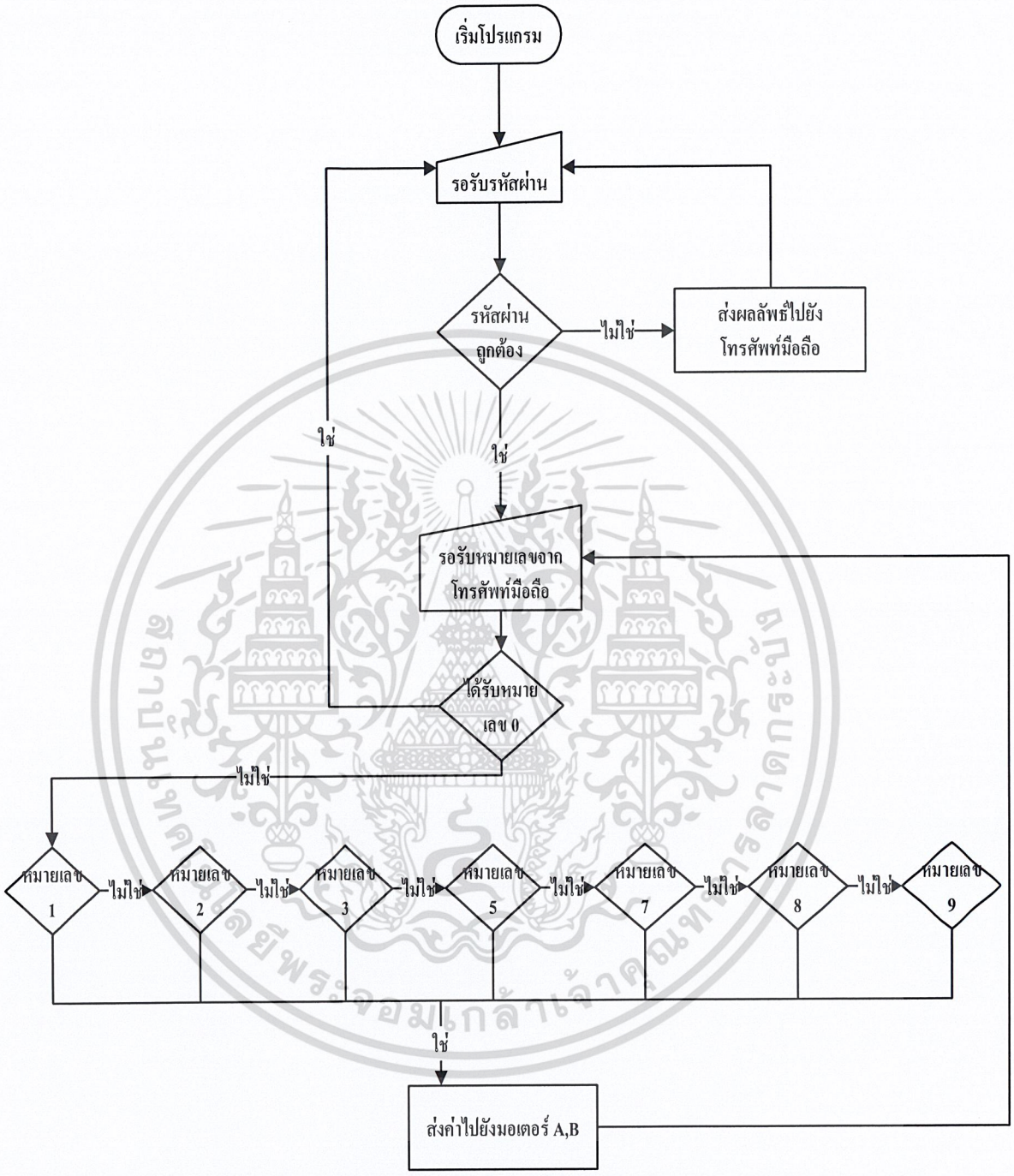
การทำงานของมอเตอร์ทั้งสองตัวที่ถูกควบคุมโดยบอร์ดไมโครคอนโทรลเลอร์สามารถแสดงได้ดังตารางที่ 3.1

ตารางที่ 3.1 รูปแบบการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก

รูปแบบการเคลื่อนที่	มอเตอร์ A		มอเตอร์ B	
	ทิศทางการไหลของกระแสไฟฟ้า	ทิศทางการหมุน	ทิศทางการไหลของกระแสไฟฟ้า	ทิศทางการหมุน
เดินหน้า (หมายเลข 2)	⊕ → ⊖	▲	⊕ → ⊖	▲
ถอยหลัง (หมายเลข 8)	⊖ ← ⊕	▼	⊖ ← ⊕	▼
เลี้ยวซ้าย (หมายเลข 1)	⊕ → ⊖	▲	✖	✖
เลี้ยวขวา (หมายเลข 3)	✖	✖	⊕ → ⊖	▲
หมุนซ้าย (หมายเลข 7)	⊖ ← ⊕	▼	✖	✖
หมุนขวา (หมายเลข 9)	✖	✖	⊖ ← ⊕	▼
หยุด (หมายเลข 5)	✖	✖	✖	✖

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมบนไมโครคอนโทรลเลอร์สามารถแสดง flow chart การทำงานได้ดังรูป

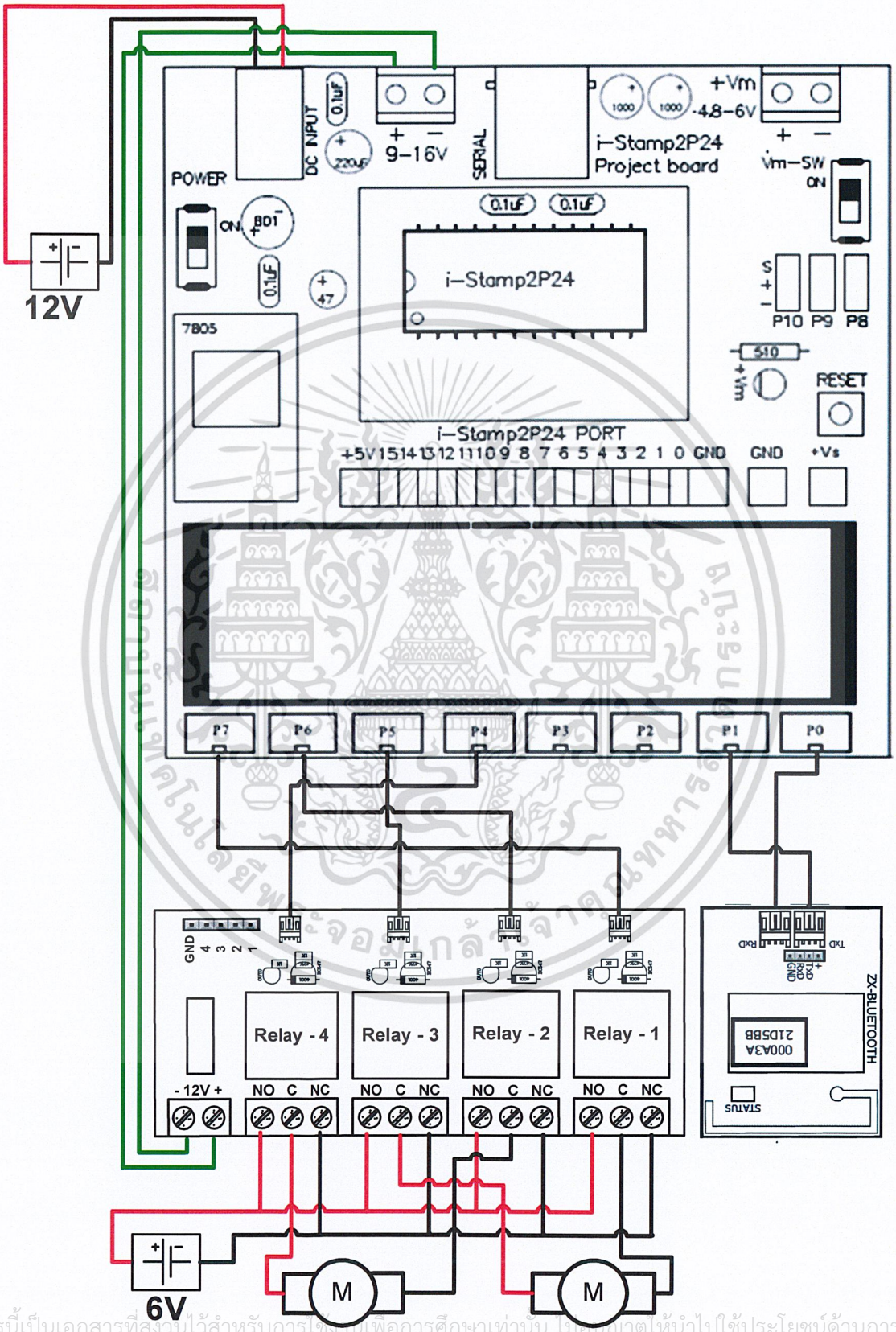


รูปที่ 3.4 Flow chart การทำงานของโปรแกรมบนไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การประกอบหุ่นยนต์ขนาดเล็ก

วงจรหุ่นยนต์ขนาดเล็กนั้นมีวิธีการต่อใช้งานดังรูป

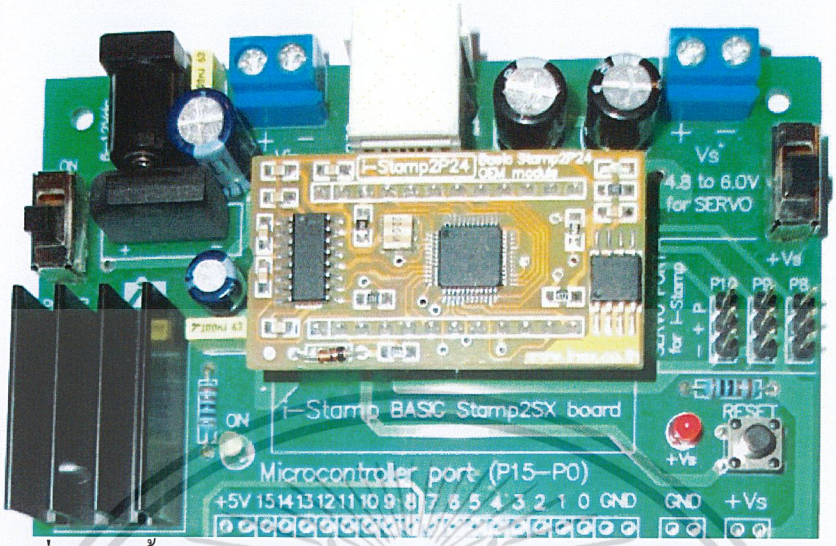


รูปที่ 3.5 วงจรหุ่นยนต์ขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 ขั้นตอนการประกอบตัวหุ่นยนต์ขนาดเล็ก

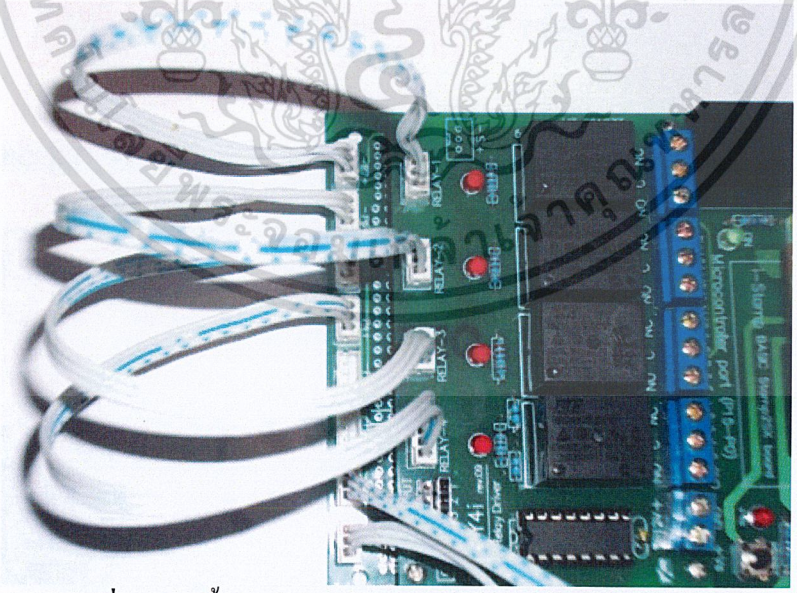
1.) นำบอร์ดเบสิกแสตมป์ ติดตั้งเข้ากับแผงวงจรสำหรับพัฒนาโครงงาน ดังรูปที่ 3.6



รูปที่ 3.6 ติดตั้งบอร์ดเบสิกแสตมป์ เข้ากับแผงวงจรสำหรับพัฒนาโครงงาน

2.) ติดตั้งรีเลย์เข้ากับแผงวงจรสำหรับพัฒนาโครงงาน ดังรูปที่ 3.7 โดย

- ขาพอร์ต P7 ของแผงวงจรต่อกับขาพอร์ต Relay-1 ของบอร์ด Relay 4i
- ขาพอร์ต P6 ของแผงวงจรต่อกับขาพอร์ต Relay-2 ของบอร์ด Relay 4i
- ขาพอร์ต P5 ของแผงวงจรต่อกับขาพอร์ต Relay-3 ของบอร์ด Relay 4i
- ขาพอร์ต P4 ของแผงวงจรต่อกับขาพอร์ต Relay-4 ของบอร์ด Relay 4i

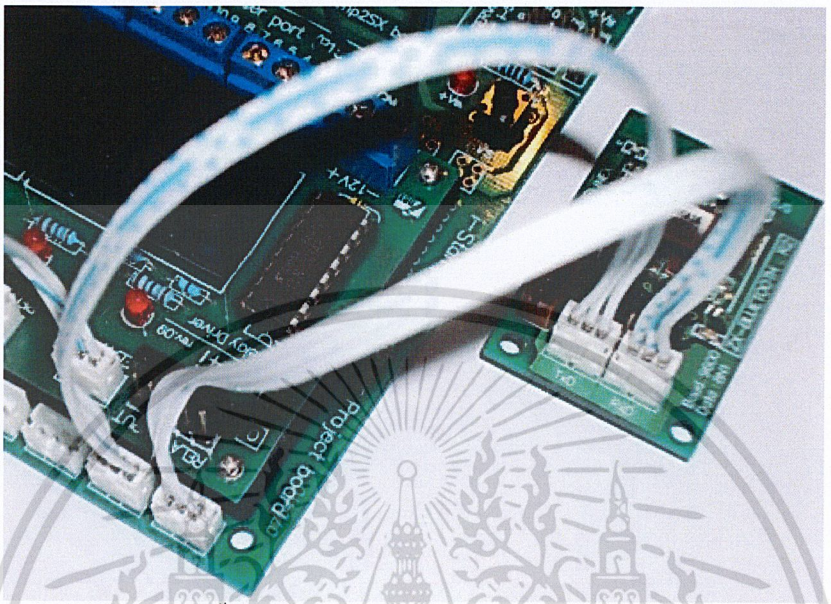


รูปที่ 3.7 ติดตั้งรีเลย์เข้ากับแผงวงจรสำหรับพัฒนาโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

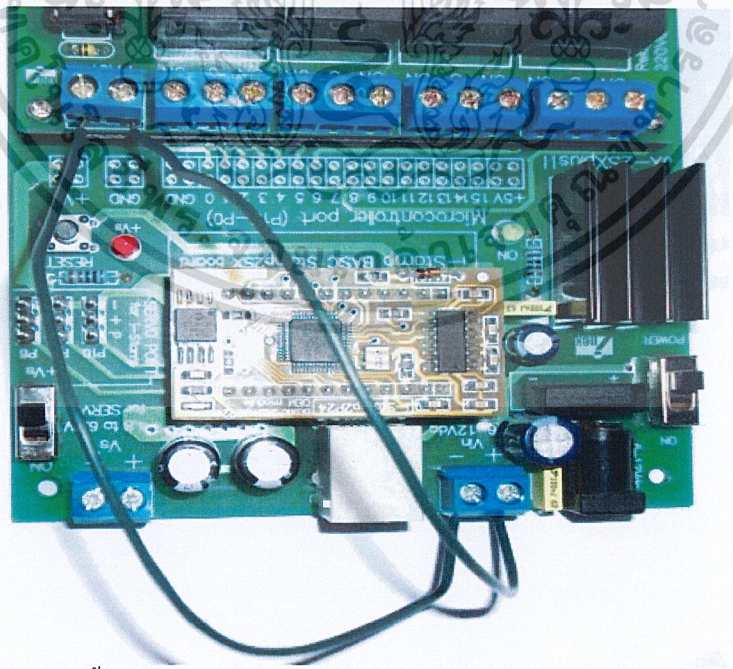
3.) ติดตั้งบลูทูธเข้ากับแผงวงจรสำหรับพัฒนาโครงงาน โดย

- ขาพอร์ต P0 ของแผงวงจรต่อกับขาพอร์ต RxD ของบอร์ด ZX-Bluetooth
- ขาพอร์ต P1 ของแผงวงจรต่อกับขาพอร์ต TxD ของบอร์ด ZX-Bluetooth



รูปที่ 3.8 ติดตั้งบลูทูธเข้ากับแผงวงจรสำหรับพัฒนาโครงงาน

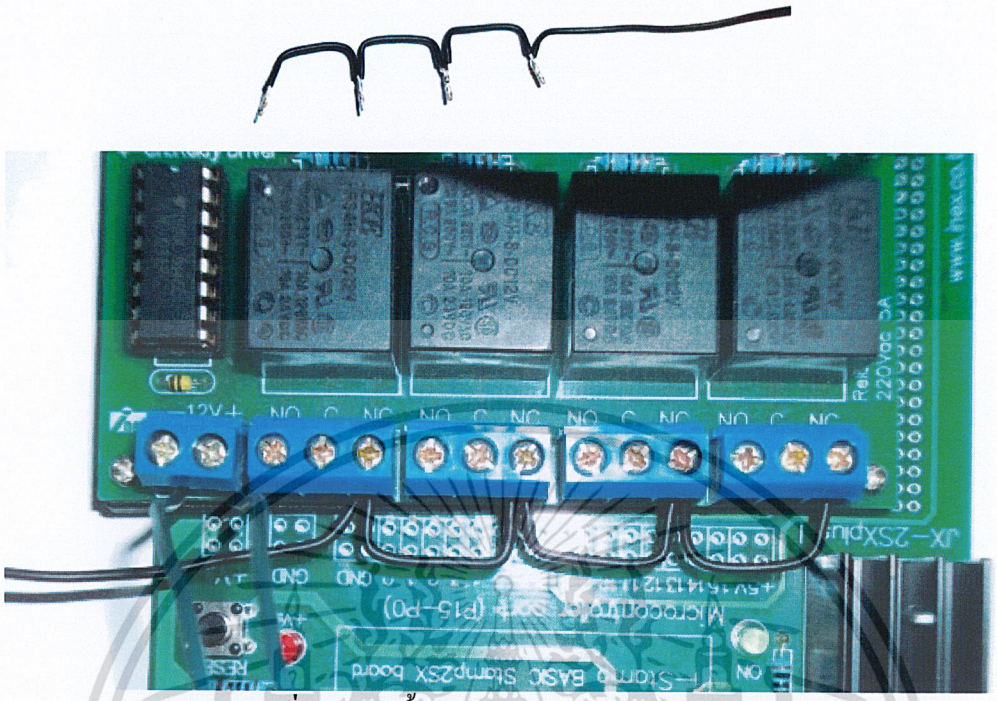
4.) ต่อสายจ่ายไฟจากแผงวงจรสำหรับพัฒนาโครงงานเข้ารีเลย์ โดยขั้วบวกต่อกับขั้วบวก และขั้วลบต่อกับขั้วลบ



รูปที่ 3.9 ติดตั้งสายจ่ายไฟจากแผงวงจรสำหรับพัฒนาโครงงานเข้ากับรีเลย์

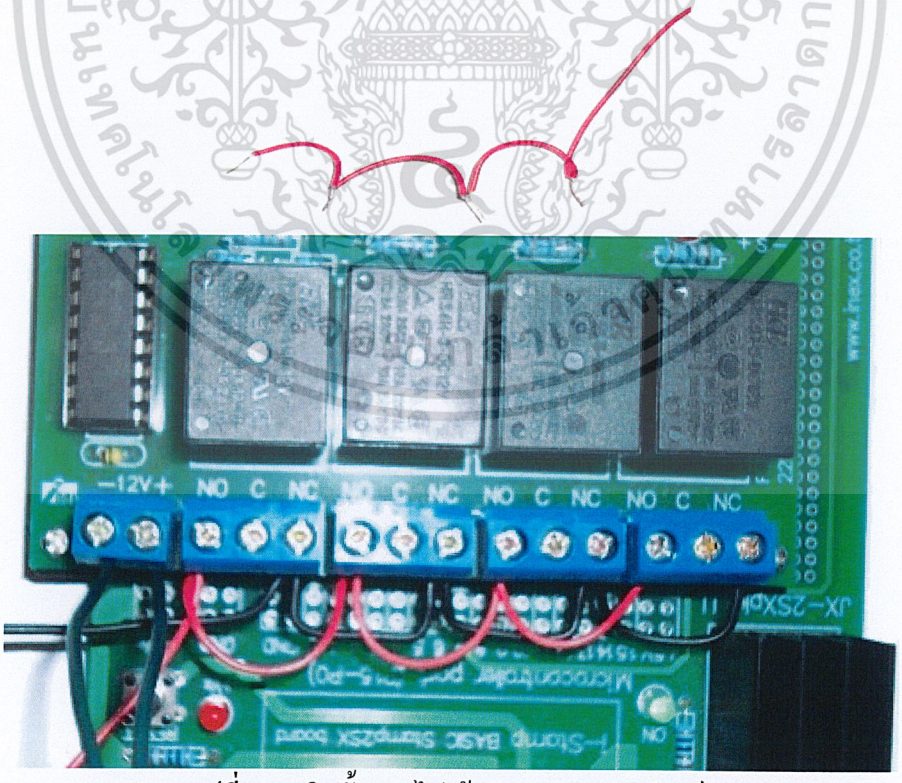
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.) ตัดสายไฟดังรูป แล้วต่อเข้ากับขา NC ของรีเลย์แต่ละตัว



รูปที่ 3.10 ติดตั้งสายไฟเข้ากับขา NC ของรีเลย์

6.) ตัดสายไฟดังรูป แล้วต่อเข้ากับขา NO ของรีเลย์แต่ละตัว



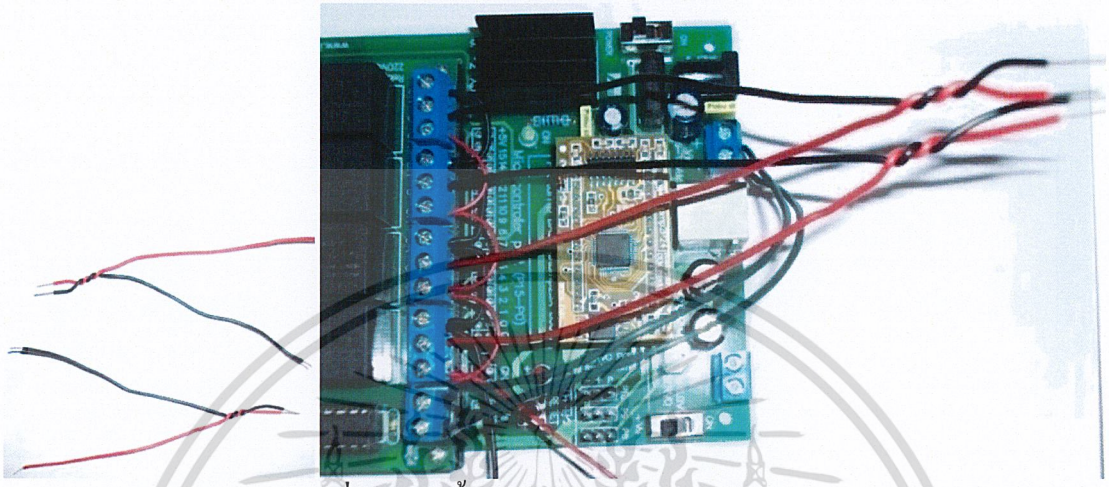
รูปที่ 3.11 ติดตั้งสายไฟเข้ากับขา NO ของรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.) ตัดสายไฟ 2 สี สีละ 2 เส้นจับคู่กันดังรูปที่ 3.12 แล้วต่อเข้ากับขา C ของรีเลย์แต่ละตัว โดย

- สายไฟคู่แรกต่อเข้าขา C รีเลย์ตัวที่ 1 และ รีเลย์ตัวที่ 3
- อีกคู่หนึ่ง ต่อเข้าขา C รีเลย์ตัวที่ 2 และ รีเลย์ตัวที่ 4

ซึ่งรีเลย์ตัวที่ 1 และ รีเลย์ตัวที่ 2 ต่อสายสีเดียวกัน, รีเลย์ตัวที่ 3 และ รีเลย์ตัวที่ 4 ต่อสายสีเดียวกัน



รูปที่ 3.12 ติดตั้งสายไฟเข้ากับขา C ของรีเลย์

8.) นำปลายสายไฟของเส้นที่ต่อออกมาจากขา NC ต่อเข้ากับขั้วลบของกระบะถ่าน และ นำปลายสายไฟของเส้นที่ต่อออกมาจากขา NO ต่อเข้ากับขั้วบวกของกระบะถ่าน ดังรูปที่ 3.13

รูปที่ 3.13 ติดตั้งสายไฟเข้ากับกระบะถ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

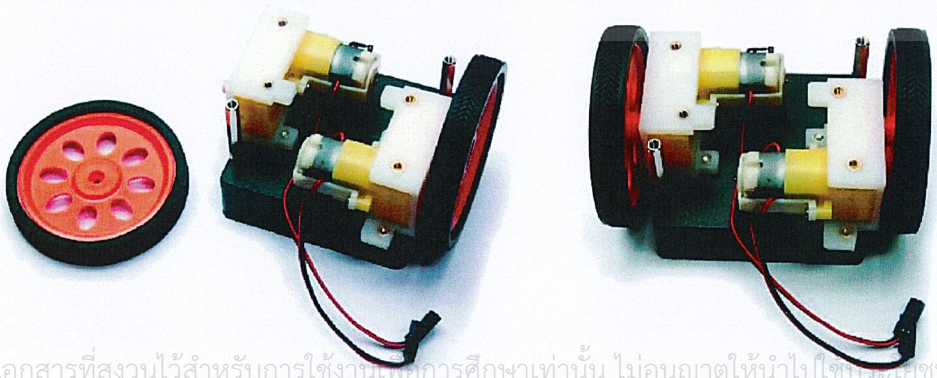
9.) นำชุดเฟืองขับมอเตอร์ยึดเข้ากับถาดรองกระบะถ่าน โดยใช้สกรูหัวตัดขนาด 3 X 8 มม. ขันยึดในตำแหน่งดังรูปที่ 3.14

รูปที่ 3.14 ติดตั้งชุดเฟืองขับมอเตอร์เข้ากับถาดรองกระบะถ่าน

10.) นำเสารองโลหะขนาด 33 มม. ยึดเข้ากับถาดรองกระบะถ่าน แล้วใช้สกรูหัวตัดขนาด 3 X 8 มม. ขันยึดในตำแหน่ง ดังรูปที่ 3.15

รูปที่ 3.15 ติดตั้งเสารองโลหะเข้ากับถาดรองกระบะถ่าน

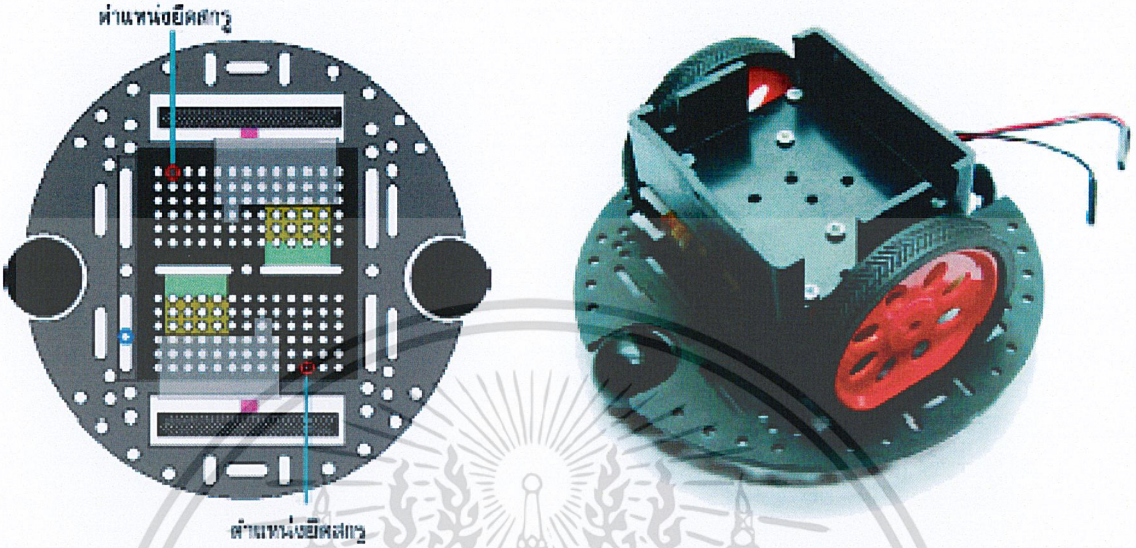
11.) นำล้อพลาสติกพร้อมยางเสียบเข้ากับแกนมอเตอร์สีชมพู แล้วใช้สกรูเกลียวปล่อย 2 มม. ขันยึด



รูปที่ 3.16 ติดตั้งล้อพลาสติกพร้อมยางเสียบเข้ากับแกนมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

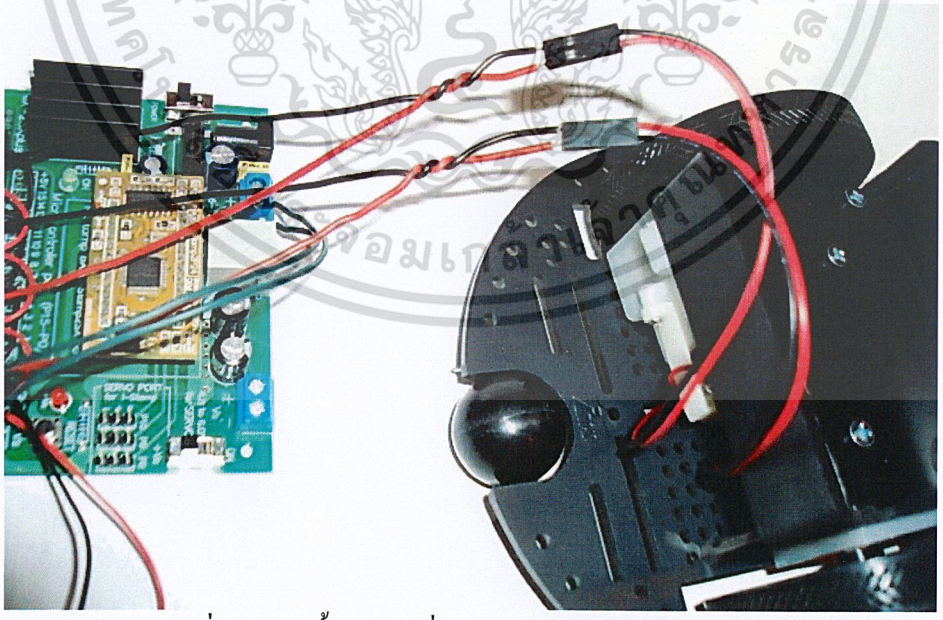
12.) นำชุดมอเตอร์ยึดเข้ากับฐานพลาสติกของหุ่นยนต์แล้วใช้สกรูขนาด 3x6 มม.ขันยึดในตำแหน่งดังรูป จะสังเกตได้ว่าการวางมอเตอร์ทั้งสองด้านไม่สมมาตรกัน แต่ล้อพลาสติกจะอยู่ตรงกลางพอดี



รูปที่ 3.17 ติดตั้งชุดมอเตอร์เข้ากับฐานพลาสติกของหุ่นยนต์

13.) นำสายไฟที่ต่อมาจากขา C ของรีเลย์แต่ละตัวเข้ากับมอเตอร์ ดังรูปที่ 3.18 โดย

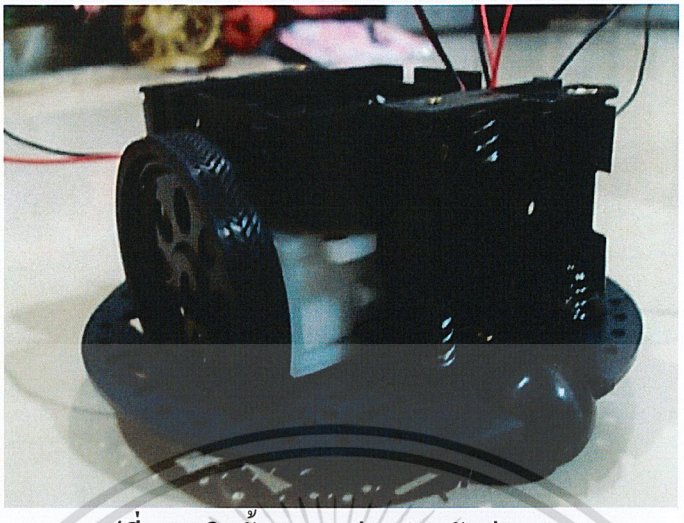
- เส้นที่ออกมาจากรีเลย์ตัวที่ 1 และ รีเลย์ตัวที่ 3 ต่อแบบตรง
- เส้นที่ออกมาจากรีเลย์ตัวที่ 2 และ รีเลย์ตัวที่ 4 ต่อแบบสลับ



รูปที่ 3.18 ติดตั้งสายไฟที่ต่อมาจากขา C เข้ากับมอเตอร์

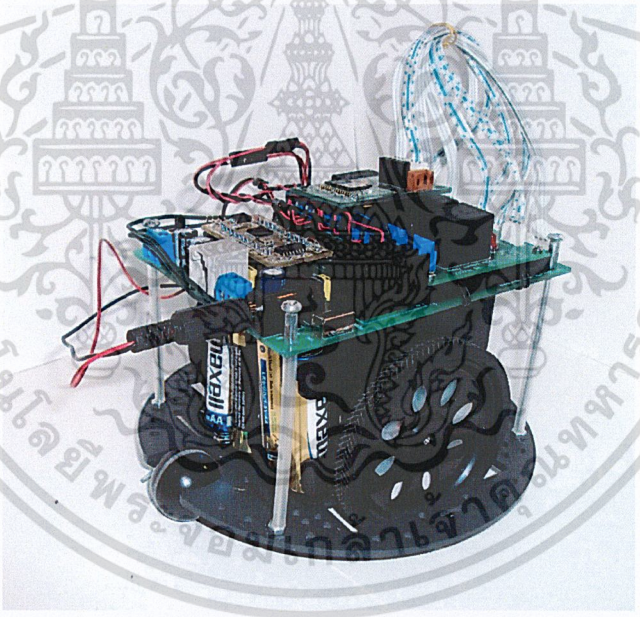
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14.) ต่อกระบะถ่านสำหรับถ่าน 12V สำหรับจ่ายไฟเข้าแผงวงจรสำหรับพัฒนาโครงงาน



รูปที่ 3.19 ติดตั้งกระบะถ่านสำหรับถ่าน 12V

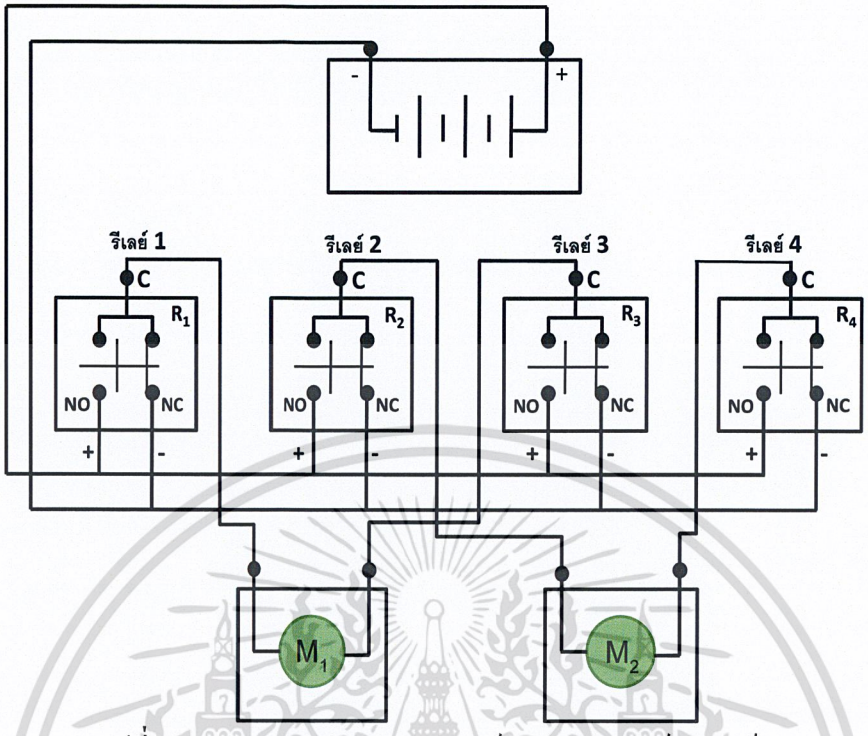
15.) ประกอบแผงวงจรสำหรับพัฒนาโครงงานเข้ากับชุดล้อ



รูปที่ 3.20 การประกอบแผงวงจรสำหรับพัฒนาโครงงานเข้ากับชุดล้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

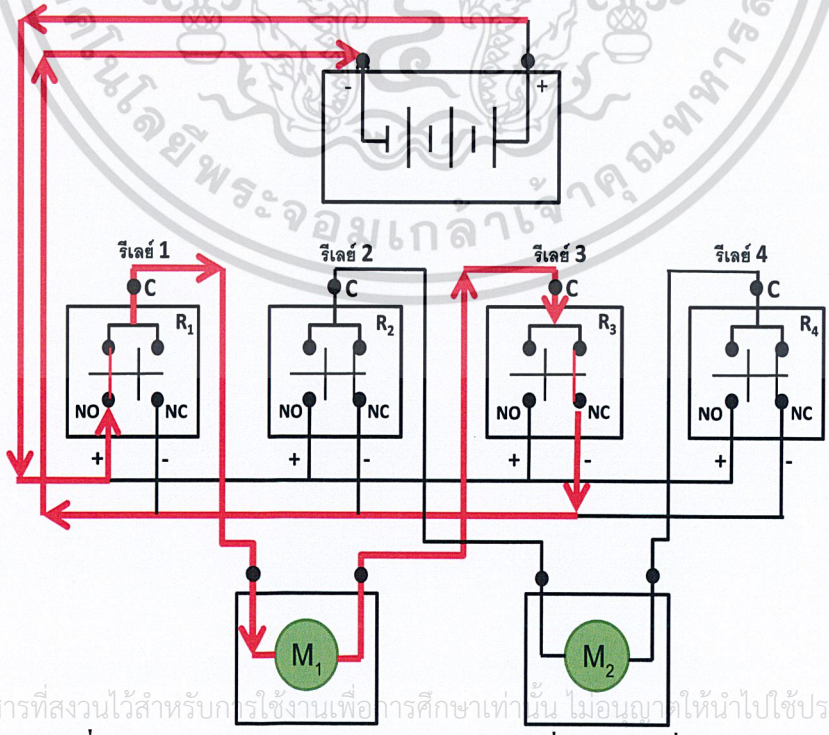
3.3.2 การต่อวงจรมอเตอร์ของหุ่นยนต์ขนาดเล็ก



รูปที่ 3.21 แสดงการต่อวงจรมอเตอร์ภายในหุ่นยนต์ขนาดเล็ก

การทำงานของรีเลย์มีดังนี้

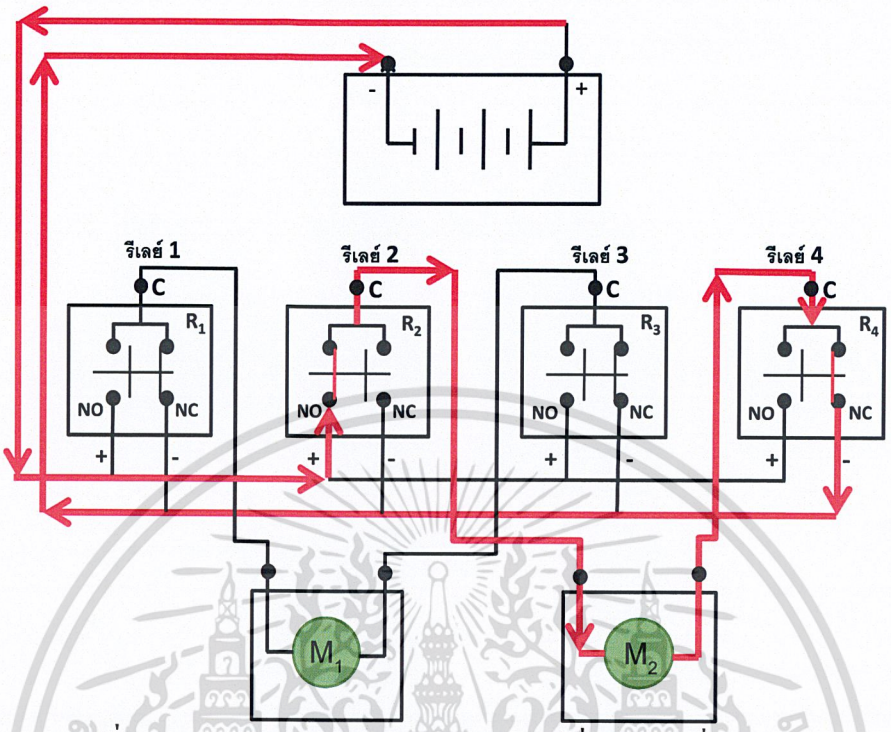
1.) เมื่อสั่งให้รีเลย์ตัวที่ 1 ทำงาน ทำให้หน้าสัมผัสรีเลย์จากเดิมแตะอยู่ที่ NC จะมาแตะด้าน NO กระแสไฟจะไหลดังรูปที่ 3.22



รูปที่ 3.22 แสดงการทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 1 ทำงาน

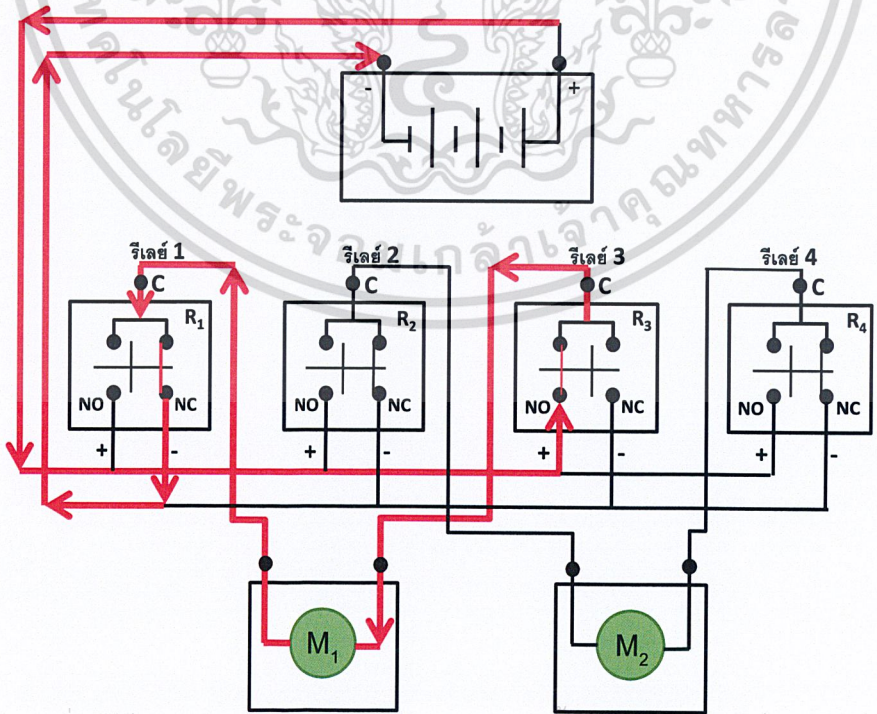
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) เมื่อสั่งให้รีเลย์ตัวที่ 2 ทำงาน ทำให้หน้าสัมผัสรีเลย์จากเดิมแตะอยู่ที่ NC จะมาแตะด้าน NO กระแสไฟจะไหลดังรูปที่ 3.23



รูปที่ 3.23 แสดงการทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 2 ทำงาน

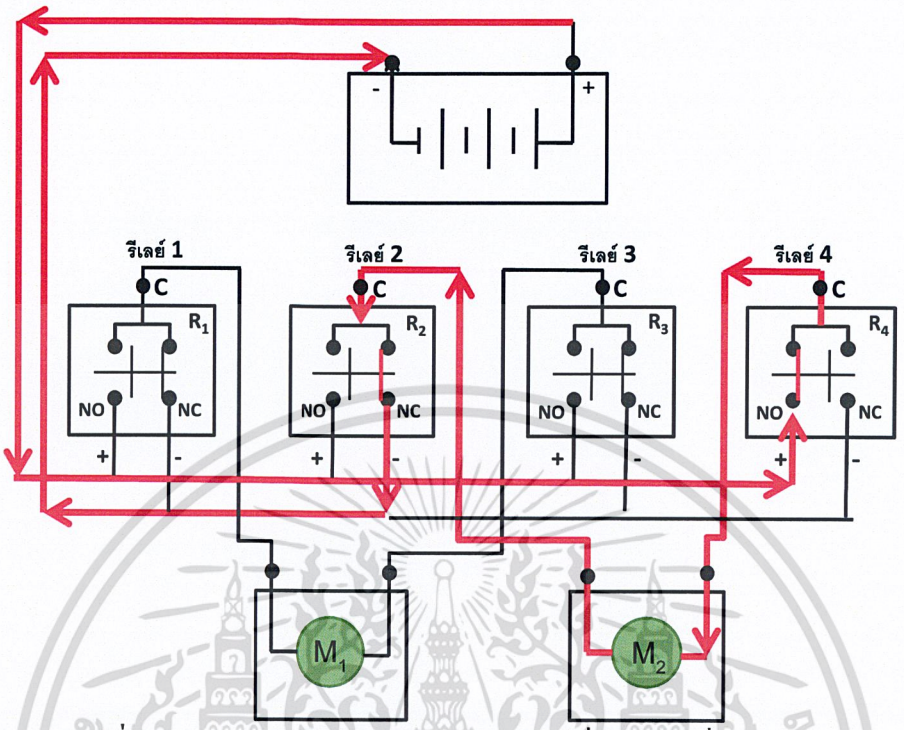
3.) เมื่อสั่งให้รีเลย์ตัวที่ 3 ทำงาน ทำให้หน้าสัมผัสรีเลย์จากเดิมแตะอยู่ที่ NC จะมาแตะด้าน NO กระแสไฟจะไหลดังรูปที่ 3.24



รูปที่ 3.24 แสดงการทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 3 ทำงาน

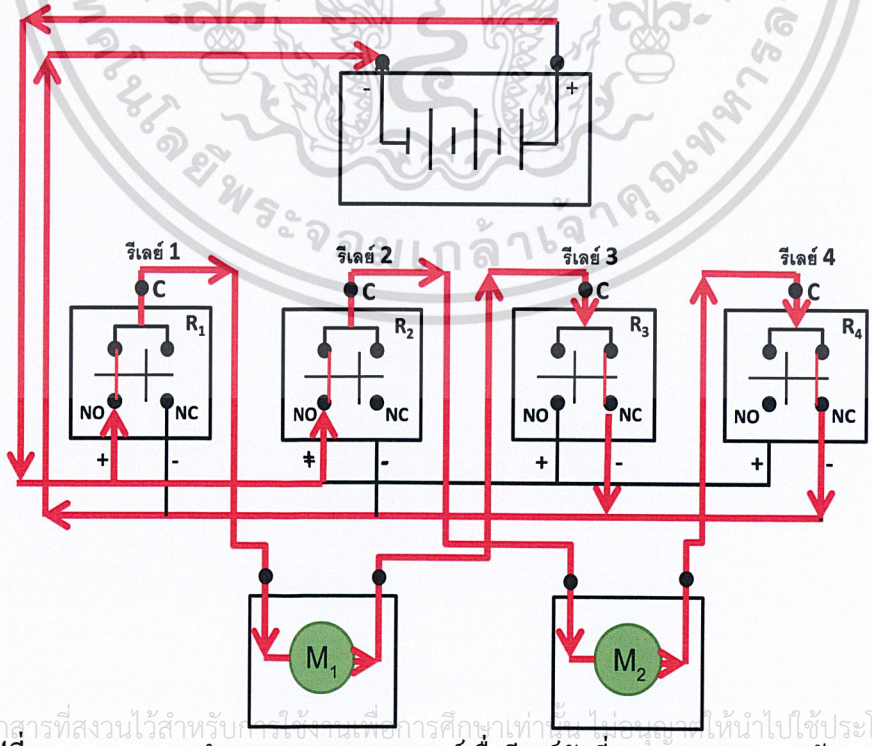
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.) เมื่อสั่งให้รีเลย์ตัวที่ 4 ทำงาน ทำให้หน้าสัมผัสรีเลย์จากเดิมแตะอยู่ที่ NC จะมาแตะด้าน NO กระแสไฟจะไหลดังรูปที่ 3.25



รูปที่ 3.25 แสดงการทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 4 ทำงาน

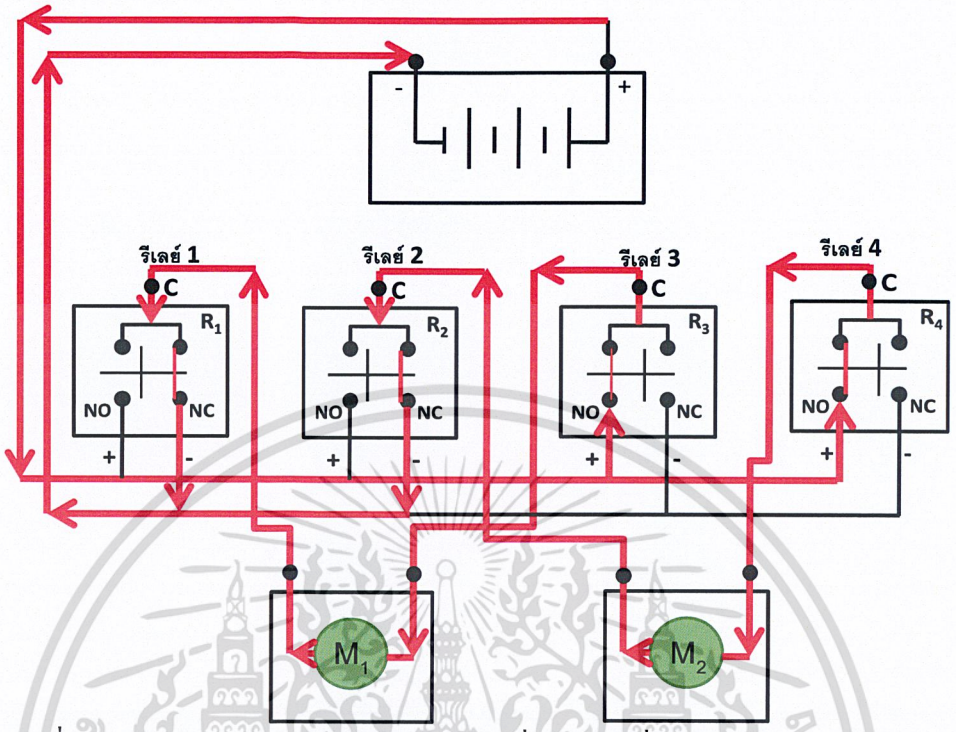
5.) เมื่อสั่งให้รีเลย์ตัวที่ 1 และ 2 ทำงาน เพื่อให้มอเตอร์หมุนเดินหน้าพร้อมกัน กระแสไฟจะไหลดังรูปที่ 3.26



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.26 แสดงการทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 1 และ 2 ทำงานพร้อมกัน

6.) เมื่อสั่งให้รีเลย์ตัวที่ 3 และ 4 ทำงาน เพื่อให้มอเตอร์หมุนกลับด้านพร้อมกัน กระแสไฟจะไหลดังรูปที่ 3.27



รูปที่ 3.27 แสดงการทำงานของวงจรมอเตอร์เมื่อรีเลย์ตัวที่ 3 และ 4 ทำงานพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

จากการพัฒนาโปรแกรมควบคุมหุ่นยนต์ขนาดเล็กผ่านโทรศัพท์มือถือ ได้แบ่งการทดลองออกเป็น 2 ส่วน คือ ส่วนการทดลองบนโทรศัพท์มือถือ และ ส่วนการทดลองบนบอร์ดไมโครคอนโทรลเลอร์ที่เชื่อมต่อกับบอร์ดรับสัญญาณบลูทูธ เพื่อดูผลลัพธ์การทำงานที่เกิดขึ้น

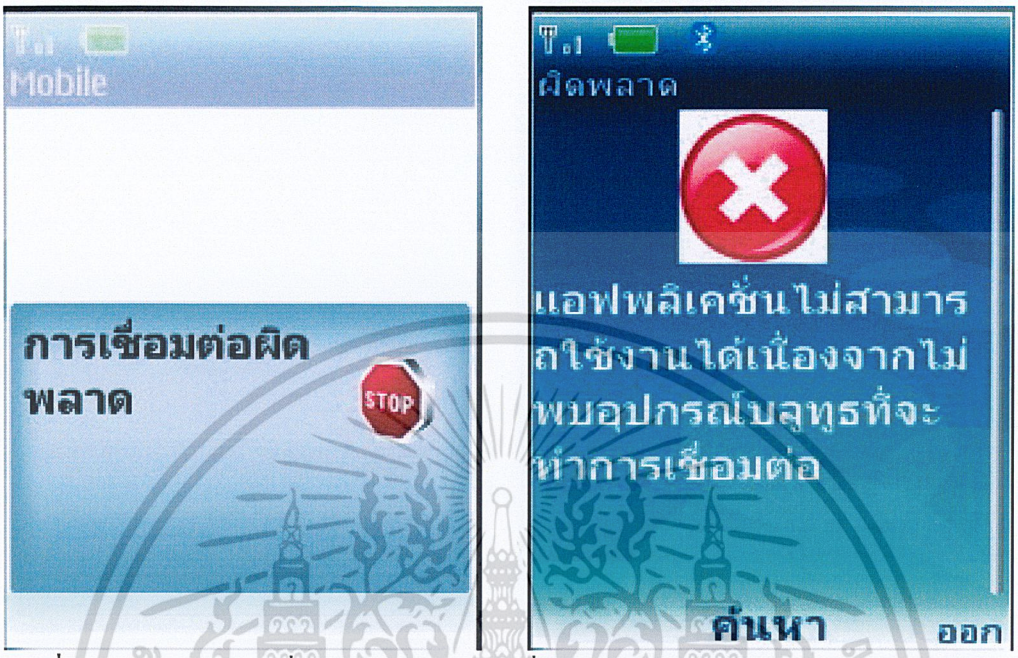
4.1 การทดลองบนโทรศัพท์มือถือ

1.) ทำการติดตั้งโปรแกรม RoboStamp ลงบนโทรศัพท์มือถือ เปิดโปรแกรมขึ้นมา ตัวโปรแกรมจะทำการค้นหาและเชื่อมต่อกับอุปกรณ์รับสัญญาณบลูทูธโดยอัตโนมัติ ถ้าทำการเชื่อมต่อสำเร็จ จะปรากฏข้อความ “เชื่อมต่อเรียบร้อยแล้ว” ดังรูป

รูปที่ 4.1 ข้อความแสดงเมื่อค้นหาและการเชื่อมต่ออุปกรณ์รับสัญญาณบลูทูธสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) ในกรณีที่ค้นหาอุปกรณ์รับสัญญาณบลูทูธไม่พบ และไม่สามารถทำการเชื่อมต่อกับอุปกรณ์รับสัญญาณบลูทูธได้ ที่หน้าจอโทรศัพท์มือถือจะปรากฏข้อความเตือน “การเชื่อมต่อผิดพลาด” และจะแสดงหน้าจอเพื่อให้ทำการค้นหาใหม่ หรือปิดโปรแกรม ดังรูป



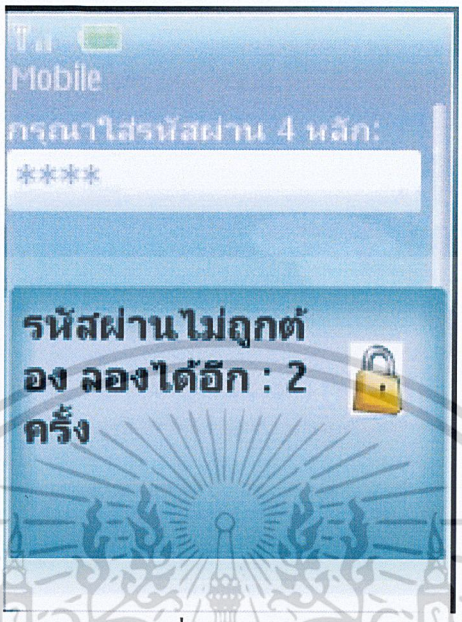
รูปที่ 4.2 ข้อความแสดงเมื่อการค้นหาและการเชื่อมต่ออุปกรณ์รับสัญญาณบลูทูธไม่สำเร็จ

3.) เมื่อทำการเชื่อมต่อโทรศัพท์มือถือกับอุปกรณ์รับสัญญาณบลูทูธสำเร็จ โทรศัพท์มือถือจะแสดงหน้าจอให้ผู้ใช้ใส่รหัสผ่านจำนวน 4 หลัก เพื่อเข้าใช้งานฟังก์ชันต่างๆ ที่ใช้ในการควบคุมหุ่นยนต์ขนาดเล็ก

รูปที่ 4.3 หน้าจอแสดงการใส่รหัสผ่าน

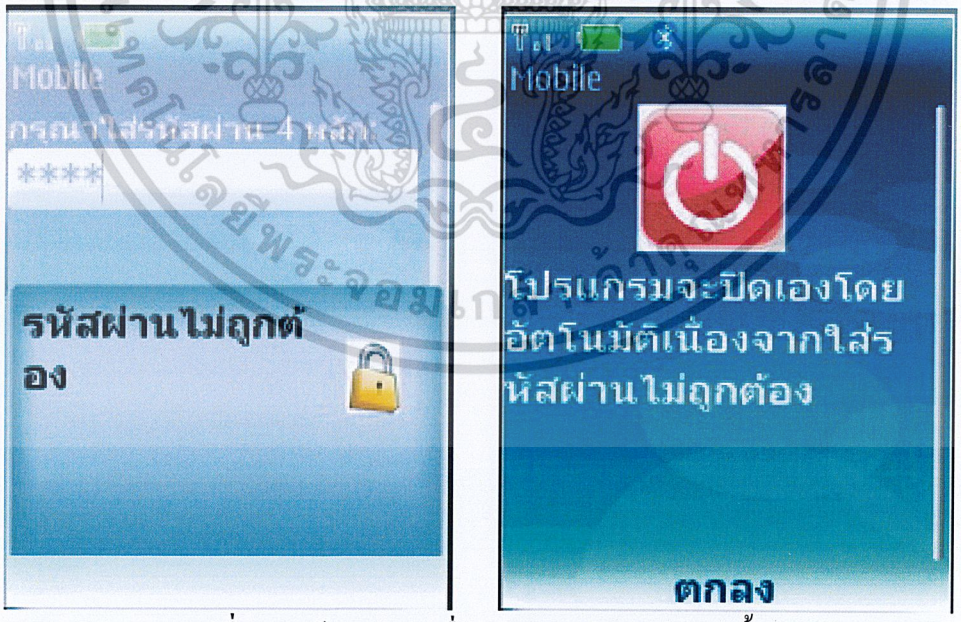
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.) ในกรณีที่ใส่รหัสผ่านไม่ถูกต้องจะแสดงข้อความเตือน “รหัสผ่านไม่ถูกต้อง” พร้อมทั้งบอกจำนวนครั้งที่สามารถใส่รหัสผ่านได้ ซึ่งใส่รหัสผ่านได้ทั้งหมด 3 ครั้ง แล้วจะกลับไปยังหน้าใส่รหัสผ่านอีกครั้ง



รูปที่ 4.4 ข้อความแสดงเมื่อใส่รหัสผ่านผิดยังไม่ครบ 3 ครั้ง

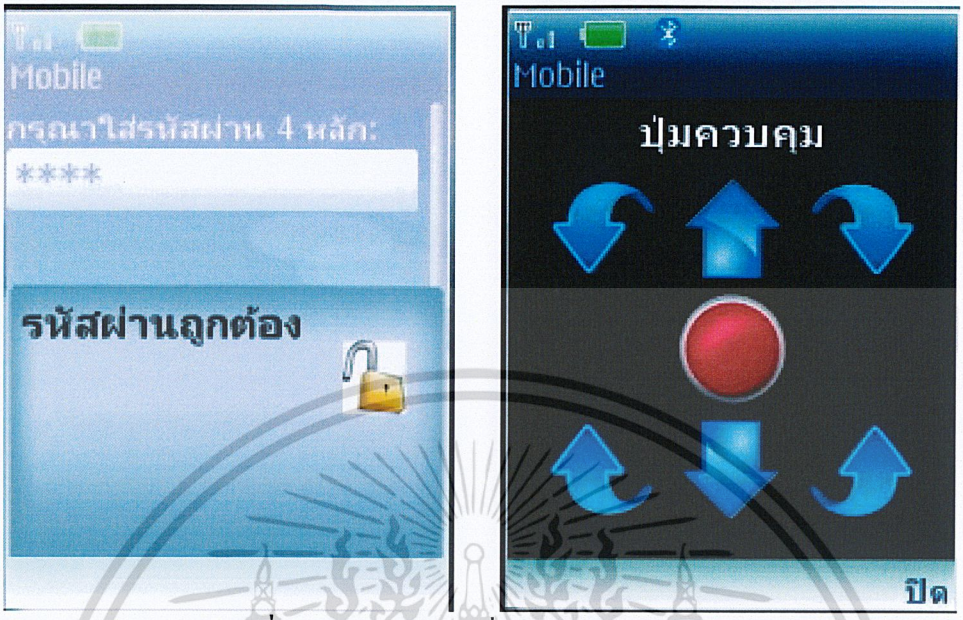
5.) ในกรณีที่ใส่รหัสผ่านผิดครบ 3 ครั้ง จะแสดงข้อความเตือน “รหัสผ่านไม่ถูกต้อง” หลังจากนั้นจะทำการปิดโปรแกรมโดยอัตโนมัติ



รูปที่ 4.5 หน้าจอแสดงเมื่อใส่รหัสผ่านผิดครบ 3 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.) ในกรณีที่ใส่รหัสผ่านถูกต้อง จะแสดงข้อความ “รหัสผ่านถูกต้อง” แล้วจะเข้าสู่หน้าจอเมนูหลัก เพื่อเลือกคำสั่งในการควบคุมหุ่นยนต์ขนาดเล็กดังรูป



รูปที่ 4.6 หน้าจอแสดงเมื่อใส่รหัสผ่านถูกต้อง

7.) ในหน้าจอเมนูหลัก ผู้ใช้สามารถกดปุ่มหมายเลขเพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็กได้ โดยหมายเลขที่ใช้ในการควบคุมจะแสดงดังตาราง

ตารางที่ 4.1 หมายเลขควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก

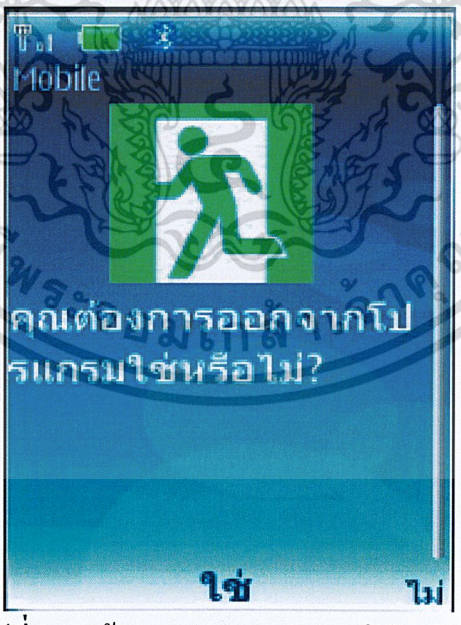
หมายเลข	ทิศทางการเคลื่อนที่
1	เลี้ยวซ้าย
2	เดินหน้า
3	เลี้ยวขวา
5	หยุด
6	หมุนตามเข็มนาฬิกา
7	ถอยหลัง
8	หมุนทวนเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.) หากเกิดข้อผิดพลาดขณะส่งคำสั่งไปควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก จะแสดงข้อความเตือน “ไม่สามารถส่งค่าได้” เพื่อให้ผู้ใช้ทราบถึงข้อผิดพลาดดังรูป

รูปที่ 4.8 ข้อความแสดงเมื่อมีข้อผิดพลาดขณะส่งคำสั่ง

9.) เมื่อผู้ใช้ต้องการออกจากโปรแกรม สามารถกดปุ่มออก หลังจากนั้น โปรแกรมจะทำการถามเพื่อยืนยันอีกครั้งว่าต้องการออกจากโปรแกรมใช่หรือไม่ ดังรูป

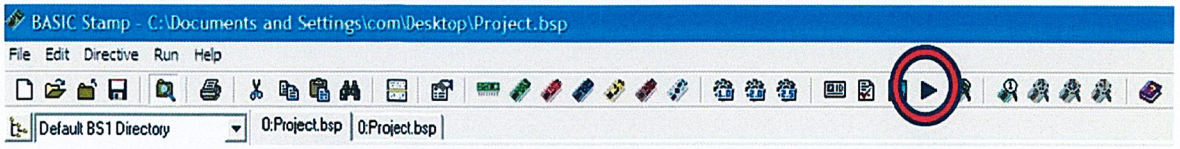


รูปที่ 4.9 หน้าจอแสดงการออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

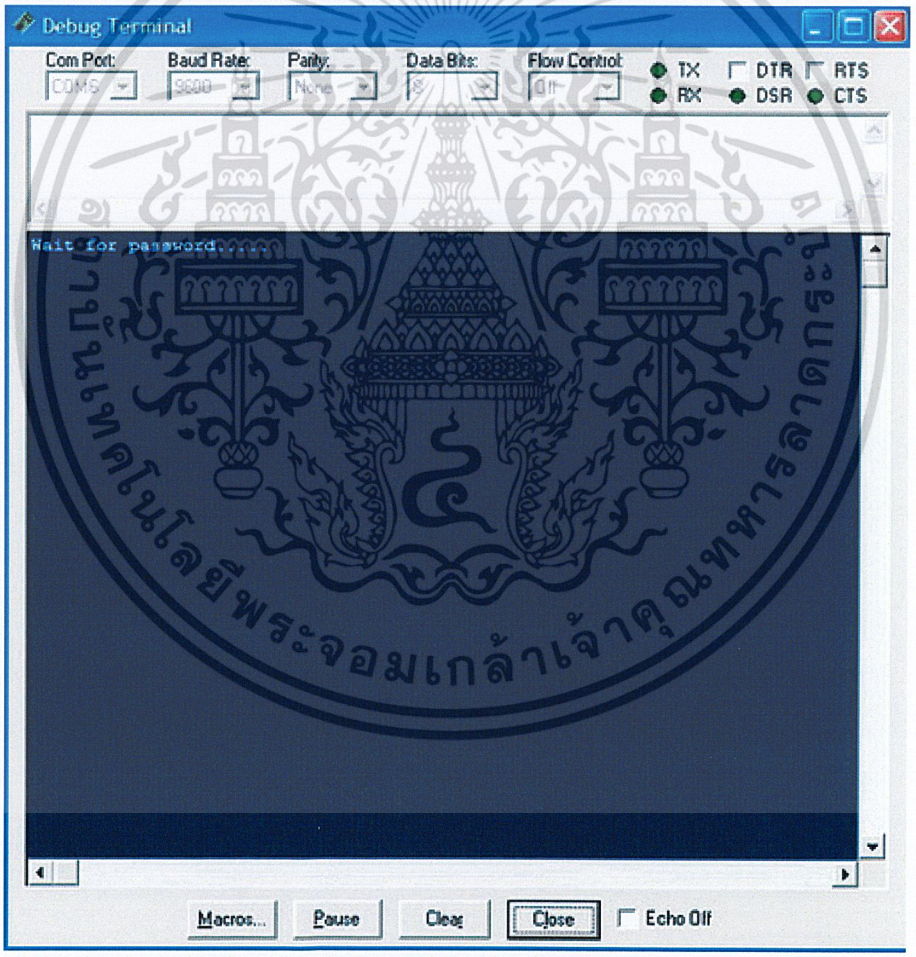
4.2 การทดลองบนบอร์ดไมโครคอนโทรลเลอร์

ในการทดลองบนบอร์ดไมโครคอนโทรเลอร์ใช้โปรแกรม BASIC Stamp Editor ในการทดลอง โดยเลือกที่ปุ่ม รันโปรแกรม ดังรูป เพื่อเปิดหน้าต่าง Debug Terminal อันเป็นหน้าต่างการติดต่อระหว่างเบสิกแสตมป์กับคอมพิวเตอร์ผ่านพอร์ตอนุกรม



รูปที่ 4.10 หน้าจอโปรแกรม BASIC Stamp Editor

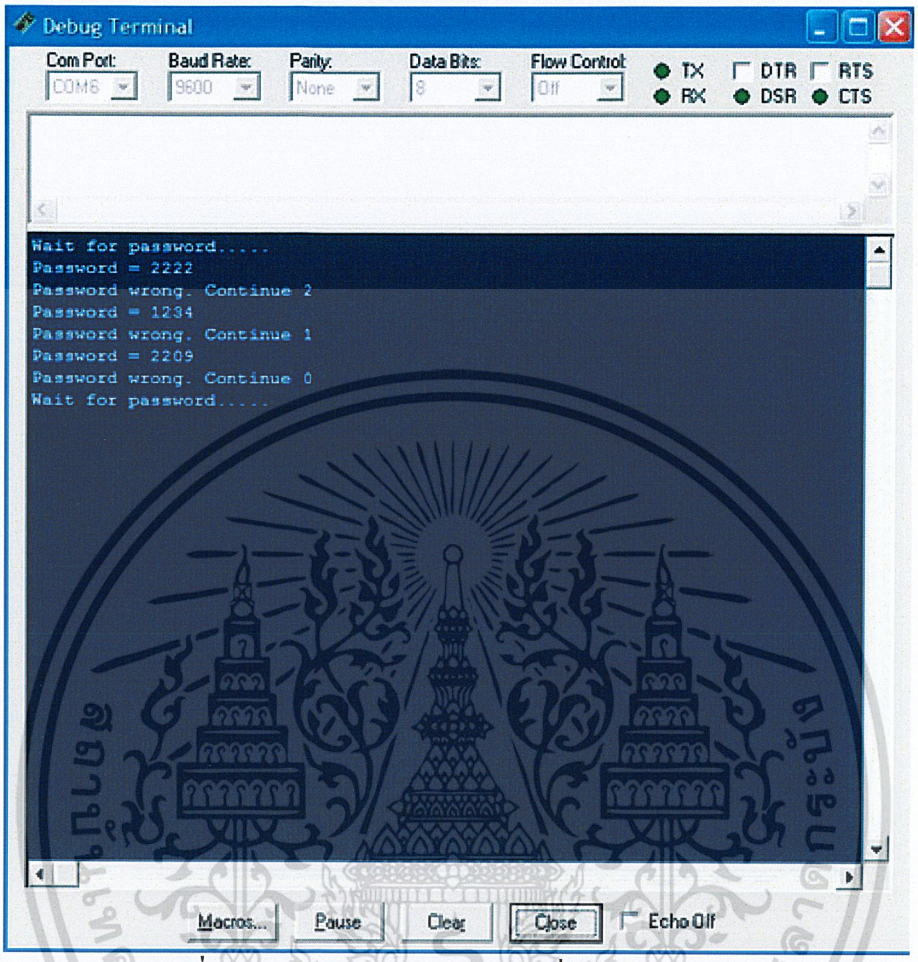
1.) ขั้นตอนแรกจะเป็นการตรวจสอบรหัสผ่านที่ผู้ใช้ป้อนเข้ามาจากโทรศัพท์มือถือ โดยบอร์ดไมโครคอนโทรเลอร์จะอยู่ในสถานะรอรับรหัสเพื่อนำรหัสผ่านดังกล่าวไปตรวจสอบ ดังรูป



รูปที่ 4.11 หน้าต่างแสดงสถานะของบอร์ดไมโครคอนโทรเลอร์ในการรอรับรหัสผ่านจากผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

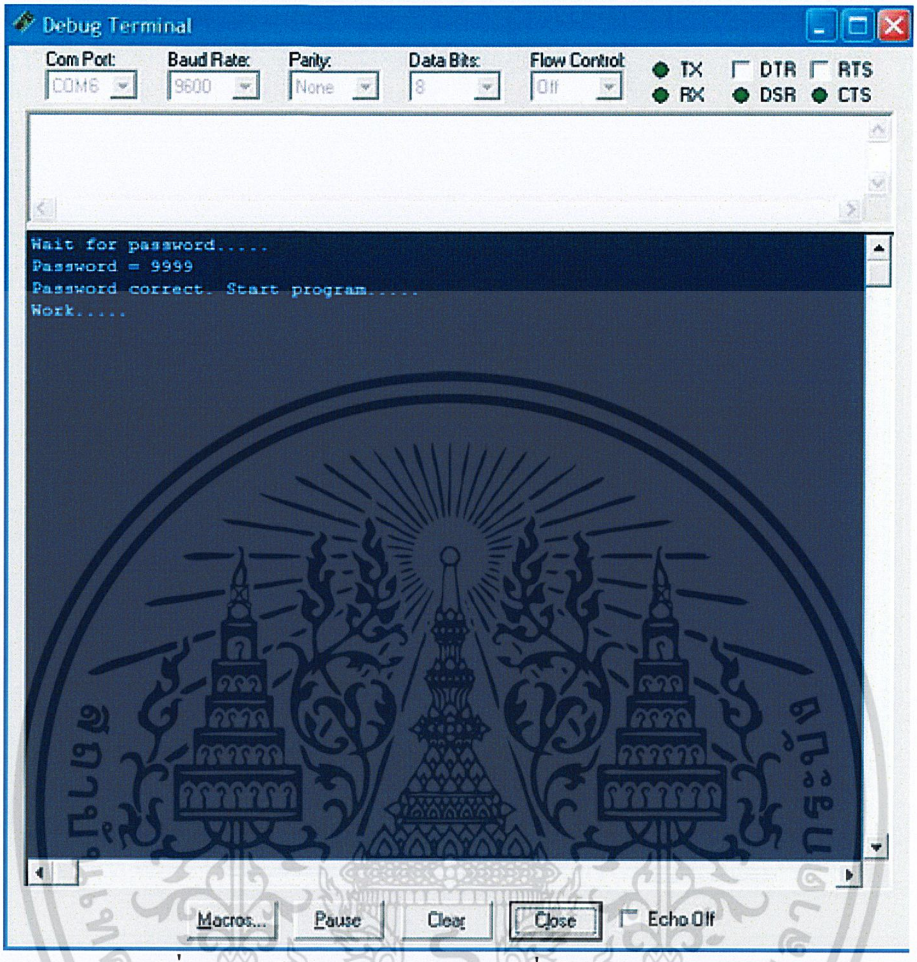
2.) ในกรณีที่ใส่รหัสผ่านผิดครบ 3 ครั้ง บอร์ดไมโครคอนโทรเลอร์จะกลับไปอยู่ในสถานะ รอรับรหัสผ่าน เพื่อรอให้ผู้ใช้ทำการเชื่อมต่อใหม่อีกครั้ง



รูปที่ 4.12 หน้าต่างโปรแกรมแสดงเมื่อใส่รหัสผ่านผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

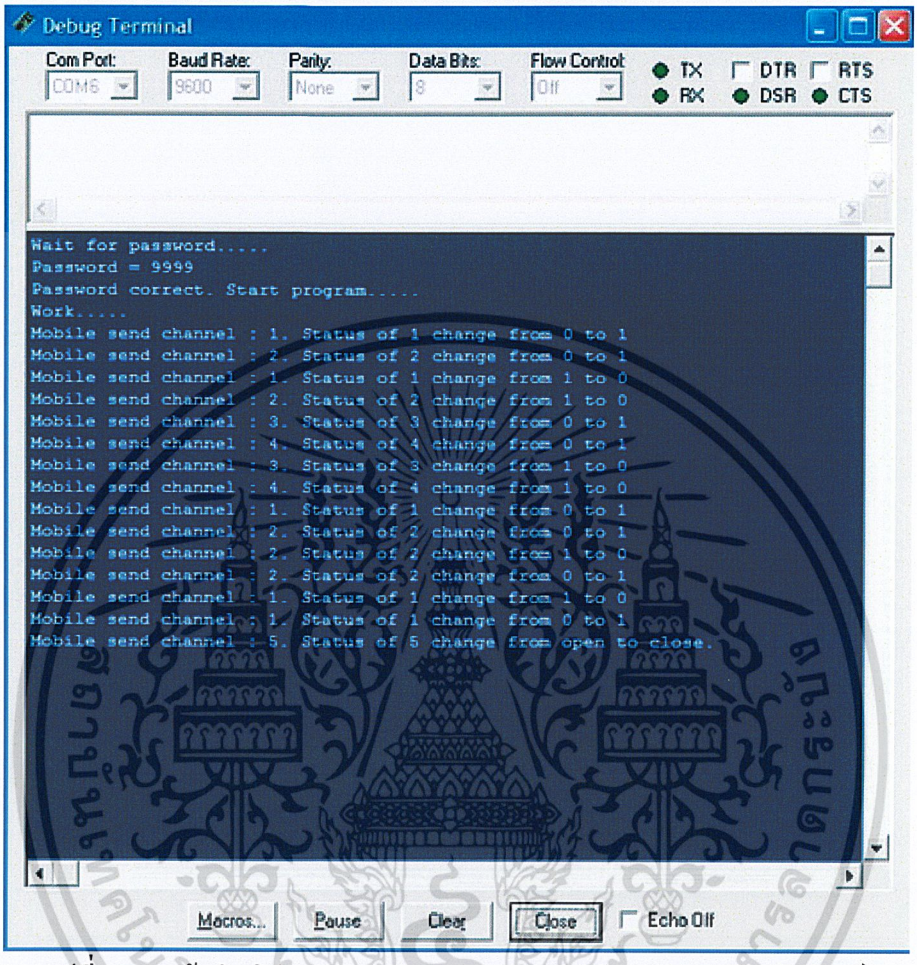
3.) ในกรณีที่ใส่รหัสผ่านถูกต้อง บอร์ดไมโครคอนโทรลเลอร์จะอยู่ในสถานะรอรับคำสั่งในการควบการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก ดังรูป



รูปที่ 4.13 หน้าต่างโปรแกรมแสดงเมื่อใส่รหัสผ่านถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

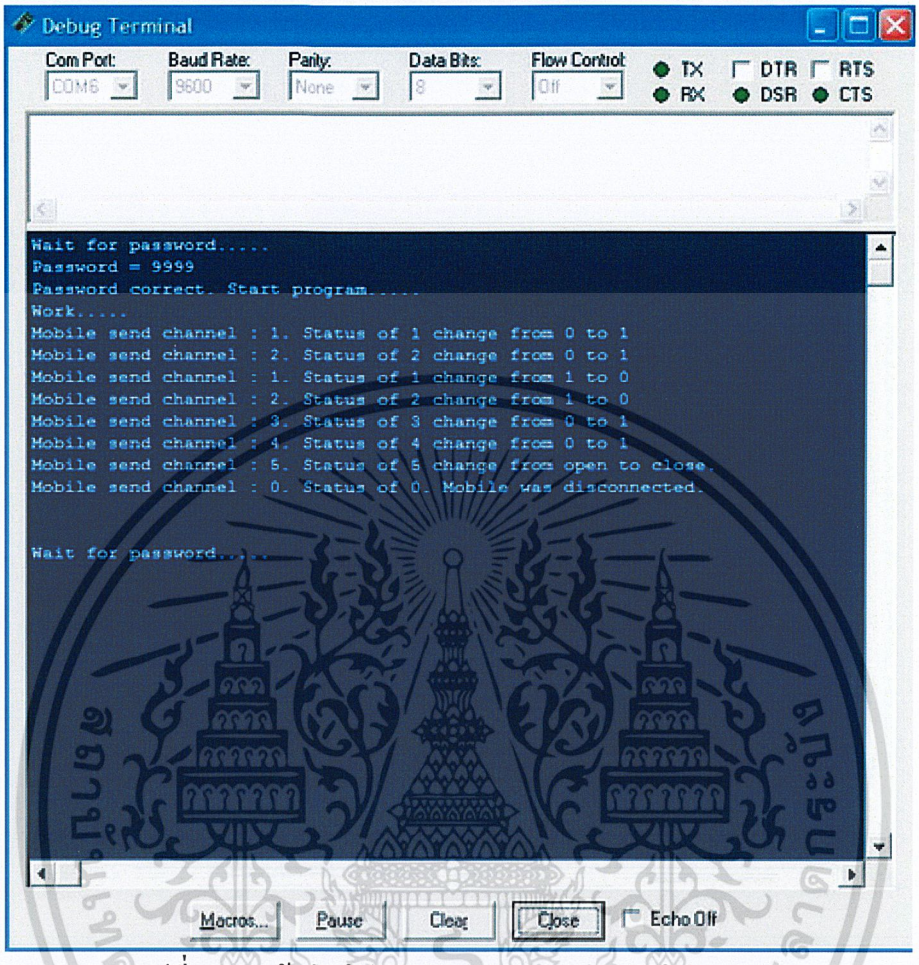
4.) เมื่อกดหมายเลขบนโทรศัพท์มือถือเพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก บอร์ดไมโครคอนโทรลเลอร์จะส่งคำสั่งไปขับมอเตอร์ให้หมุนตามทิศทางที่กำหนดไว้ โดยหากกดที่ปุ่มหมายเลข 5 จะเป็นการเซตค่าให้ Relay ทั้ง 4 ตัวมีค่าเป็น 0 ทั้งหมด ซึ่งคือการสั่งให้รถหยุด



รูปที่ 4.14 หน้าต่างโปรแกรมแสดงสถานะของการควบคุมการปิดเปิดรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.) เมื่อผู้ใช้กดปุ่มปิด เพื่อออกจากโปรแกรม บอร์ดไมโครคอนโทรลเลอร์จะเข้าสู่สถานะรอรับรหัสผ่านใหม่อีกครั้ง



รูปที่ 4.15 หน้าต่างโปรแกรมแสดงการออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงาน และข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการพัฒนา โปรแกรมควบคุมหุ่นยนต์ขนาดเล็กผ่านโทรศัพท์มือถือ สามารถสรุปผลการพัฒนาระบบได้ดังนี้

5.1.1 สรุปผลการดำเนินงาน

จากการพัฒนาโปรแกรมควบคุมหุ่นยนต์ขนาดเล็กผ่านโทรศัพท์มือถือ เป็นการพัฒนาโดยใช้ภาษา Java J2ME ในการเขียนโปรแกรมบนโทรศัพท์มือถือ เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก และใช้ภาษาเบสิกในการเขียนโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์ เพื่อควบคุมการหมุนของมอเตอร์ทั้งสองตัวที่เชื่อมต่ออยู่กับบอร์ดไมโครคอนโทรลเลอร์ โดยการส่งรหัสคำสั่งระหว่างโทรศัพท์มือถือกับบอร์ดไมโครคอนโทรลเลอร์ จะส่งผ่านสัญญาณบลูทูธ การใช้งานโปรแกรมเริ่มจาก เมื่อผู้ใช้เปิดแอพพลิเคชั่นขึ้นมา ระบบสามารถทำการค้นหาอุปกรณ์รับสัญญาณบลูทูธ และทำการเชื่อมต่อโดยอัตโนมัติ หลังจากผู้ใช้ใส่รหัสผ่านเรียบร้อย ระบบจะทำตรวจสอบรหัสผ่านเพื่อเป็นการระบุตัวตนผู้ใช้ (Identification) ซึ่งเป็นการป้องกันไม่ให้บุคคลอื่นลักลอบเข้ามาใช้งานแอพพลิเคชั่น เพื่อป้องกันความเสียหายที่อาจจะเกิดขึ้นกับโปรแกรม ผู้ใช้ควบคุมหุ่นยนต์ขนาดเล็กให้เคลื่อนที่เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา หมุนซ้าย หรือหมุนขวา โดยการกดหมายเลขบนโทรศัพท์มือถือ

5.1.2 การพัฒนาโปรแกรม

การพัฒนาโปรแกรมควบคุมหุ่นยนต์ขนาดเล็กผ่านโทรศัพท์มือถือ ใช้ภาษา Java J2ME ในการเขียนโปรแกรมควบคุมการทำงานบนโทรศัพท์มือถือ ซึ่งโทรศัพท์มือถือที่ใช้ในการพัฒนานั้นจะต้องเป็นโทรศัพท์มือถือที่มีบลูทูธและรองรับการทำงานจาวาแอพพลิเคชั่น และใช้ภาษาเบสิกในการเขียนโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์ โดยในหัวข้อปัญหาพิเศษนี้ ใช้มอเตอร์สองตัวต่อเข้ากับบอร์ดรีเลย์ เพื่อใช้รีเลย์ในการควบคุมทิศทางการหมุนของมอเตอร์

5.1.3 ประสิทธิภาพและคุณสมบัติของโปรแกรม

จากการพัฒนาโปรแกรมควบคุมหุ่นยนต์ขนาดเล็กผ่านโทรศัพท์มือถือ สามารถสรุปคุณสมบัติของโปรแกรมได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.) โปรแกรมมีระบบการตรวจสอบรหัสผ่านจากผู้ใช้ก่อนเข้าสู่หน้าเมนูหลัก โดยอนุญาตให้ผู้ใช้สามารถใช้รหัสผ่านผิดได้ไม่เกิน 3 ครั้ง เท่านั้น ซึ่งเป็นการระบุตัวตนผู้ใช้งาน เพื่อป้องกันไม่ให้บุคคลอื่นลักลอบเข้าใช้งานโปรแกรม

2.) โปรแกรมสามารถควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ขนาดเล็กได้ โดยการกดหมายเลขบนโทรศัพท์มือถือ

5.1.4 ข้อจำกัดของโปรแกรม

1.) การแสดงผลบนหน้าจอโทรศัพท์มือถืออาจจะมีความแตกต่างกันบ้างเล็กน้อย ขึ้นอยู่กับรุ่นและยี่ห้อของโทรศัพท์มือถือของผู้ใช้

2.) โปรแกรมสามารถใช้งานได้ดีกับ โนเกีย series 40 3rd edition ซึ่งในการพัฒนาโปรแกรมนั้น ทางกลุ่มได้ทำการทดลอง บนโทรศัพท์โนเกีย 6300 ซึ่งได้ผลเป็นที่น่าพอใจ

3.) เหมาะกับผู้ใช้ที่มีความเข้าใจในด้านฮาร์ดแวร์ เพื่อให้ง่ายต่อการทำความเข้าใจการทำงานบนบอร์ดไมโครคอนโทรลเลอร์ ซึ่งผู้ใช้สามารถปรับเปลี่ยน แก้ไข หรือบำรุงรักษา ในส่วนของหุ่นยนต์ขนาดเล็กได้

4.) เนื่องจากการส่งค่าคำสั่งจากโทรศัพท์มือถือไปยังบอร์ดไมโครคอนโทรลเลอร์ เป็นการส่งแบบทีละคำสั่ง ทำให้เมื่อกดปุ่มเดินหน้าหรือถอยหลัง ที่ต้องมีการควบคุมมอเตอร์ให้หมุนพร้อมกันทั้งสองตัว จึงไม่สามารถสั่งการให้มอเตอร์ทั้งสองเริ่มหมุนพร้อมกันได้ ดังนั้นเมื่อเลือกคำสั่งดังกล่าว ในขณะที่มอเตอร์เริ่มหมุนจึงทำให้หุ่นยนต์ขนาดเล็กกระตุกไปทางซ้ายหรือขวาเล็กน้อย

5.) อีมูเลเตอร์ (Emulator) ที่ใช้ในการจำลองการทำงานบนโทรศัพท์มือ โนเกีย series 40 3rd edition รองรับแค่เพียงระบบปฏิบัติการ Microsoft Windows XP Professional SP2 เท่านั้น

5.2 ข้อเสนอแนะ

1.) สามารถนำแอพพลิเคชั่นไปพัฒนาเพื่อไม่ให้จำกัดอยู่แค่บนโทรศัพท์มือถือ หรือนำไปประยุกต์ใช้กับอุปกรณ์ต่างๆ ที่รองรับเทคโนโลยีบลูทูธ เช่น คอมพิวเตอร์โน้ตบุ๊ก พีดีเอ เป็นต้น

2.) สามารถพัฒนาการควบคุมการเคลื่อนที่ของหุ่นยนต์ขนาดเล็กให้มีลักษณะการเคลื่อนที่ได้หลากหลาย เช่น การตีลังกา การเคลื่อนที่บนพื้นที่ต่างระดับ เป็นต้น

3.) พัฒนาให้สามารถใช้สัญญาณไวเลสแทนสัญญาณบลูทูธ เพื่อเพิ่มขอบเขตในการเคลื่อนที่ของหุ่นยนต์ขนาดเล็ก

4.) พัฒนาให้สามารถติดตั้งอุปกรณ์เสริมต่างๆ เช่น กล้องถ่ายวีดีโอ กลไกในการหยิบจับสิ่งของ หรืออุปกรณ์ตรวจจับ เพื่อเพิ่มขีดความสามารถของหุ่นยนต์ขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอ้างอิง

กาญจนา ตันวิสุทธิ์. 2547. เขียนเกมและโปรแกรมบนมือถือ J2ME. กรุงเทพฯ: ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์.

ชัยวัฒน์ ลิ้มพรจิตรวิไล และ ณัฏฐพล วงศ์สุนทรชัย. ม.ป.ป. เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ PIC16F628. กรุงเทพฯ: อินโนเวตีฟ เอ็กเพอริเมนต์.

ทรงเกียรติ ภาวดี. 2546. เก่ง J2ME ให้ครบสูตร. กรุงเทพฯ: วิตตี้ กรุ๊ป.

ธนกฤต เหมวรางค์กูล, กฤษดา ใจเย็น และ ชัยวัฒน์ ลิ้มพรจิตรวิไล. ม.ป.ป. เรียนรู้และปฏิบัติการระบบควบคุมไร้สายบลูทูธเบื้องต้น. กรุงเทพฯ: อินโนเวตีฟ เอ็กเพอริเมนต์.

ประพนธ์ อัศวภาณุวัฒน์. 2549. เรียน เล่น ใช้ บนสายงาน Software Technical ตอนการโปรแกรมไมโครคอนโทรลเลอร์เบสิกแสตมป์. ใน Microcomputer, ปีที่ 24 (ฉบับที่ 254), (167-176). กรุงเทพ: ซีเอ็ด ยูเคชั่น.

ประพนธ์ อัศวภาณุวัฒน์. 2549. เรียน เล่น ใช้ บนสายงาน Software Technical ตอนการโปรแกรมไมโครคอนโทรลเลอร์เบสิกแสตมป์. ใน Microcomputer, ปีที่ 24 (ฉบับที่ 255), (161-173). กรุงเทพ: ซีเอ็ด ยูเคชั่น.

ประพนธ์ อัศวภาณุวัฒน์. 2549. เรียน เล่น ใช้ บนสายงาน Software Technical ตอนการโปรแกรมไมโครคอนโทรลเลอร์เบสิกแสตมป์. ใน Microcomputer, ปีที่ 24 (ฉบับที่ 256), (171-180). กรุงเทพ: ซีเอ็ด ยูเคชั่น.

ประพนธ์ อัศวภาณุวัฒน์. 2549. เรียน เล่น ใช้ บนสายงาน Software Technical ตอนการโปรแกรมไมโครคอนโทรลเลอร์เบสิกแสตมป์. ใน Microcomputer, ปีที่ 24 (ฉบับที่ 257), (171-175). กรุงเทพ: ซีเอ็ด ยูเคชั่น.

ไพฑูรย์ ไพวิโรจน์, ลลิตา เล็กน้อย และ วัตถาภรณ์ เอี่ยมสินธร. 2552. ระบบควบคุมสวิตช์เปิด-ปิดเครื่องใช้ไฟฟ้าภายในบ้านด้วยเทคโนโลยีบลูทูธผ่านโทรศัพท์เคลื่อนที่. วิทยานิพนธ์ปริญญาบัณฑิต. สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล. ม.ป.ป. คุณสมบัติทางฮาร์ดแวร์และชุดคำสั่งของเบสิกแสตมป์ 2SX. กรุงเทพฯ: อินโนเวตีฟ เอ็กเพอริเมนต์.

สุธี พงศาสกุลชัย และ หทัยชนก งามอินทร์. 2549. คัมภีร์ JAVA เล่ม 3. กรุงเทพฯ: เคทีพี คอมพ์ แอนด์ คอนซัลท์.

อรรณพ ขันธิกุล และ อำนาจ มีมงคล. 2553. ออกแบบและติดตั้งระบบ Wireless LAN 2nd Edition. นนทบุรี: ไอดีซี พรีเมียร์.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอ้างอิง (ต่อ)

โอภาส ศิริครรชิตถาวร, วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล. ม.ป.ป. เรียนรู้ระบบควบคุมอย่างง่ายด้วยโปรแกรมภาษา C กับ Arduino และบอร์ดไมโครคอนโทรเลอร์ POP-168. กรุงเทพฯ: อินโนเวตีฟ เอ็กเพอริเมนต์.

Jode, Martin de and others. 2004. Programming Java 2 Micro Edition on Symbian OS. New Jersey : John Wiley & Sons.

[Online].Available: [30/03/2011]--http://www.basiclite.com
[Online].Available: [30/03/2011]--http://www.bp-smakom.org/BP_School/ComCenter/StuRobot/data/structure.htm
[Online].Available: [30/03/2011]--http://www.chontech.ac.th/~electric/e-learn/unit5/unit5.htm
[Online].Available: [30/03/2011]--http://elec.chandra.ac.th/learn/elearning/microprocessor/register.html
[Online].Available: [30/03/2011]--http://en.wikipedia.org/wiki/Bluetooth
[Online].Available: [30/03/2011]--http://www.forum.nokia.com/
[Online].Available: [30/03/2011]--http://www.nokia.co.th/
[Online].Available: [30/03/2011]--http://www.saneengineer.com
[Online].Available: [30/03/2011]--http://www.technicnan.ac.th/nan_ntc/adisak51/page21.html
[Online].Available: [30/03/2011]--http://th.wikipedia.org/wiki/รีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

โค้ดโปรแกรมบนโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mobile.java

```
package program;

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Mobile extends MIDlet implements CommandListener
{
    private Display display;
    private Command seekC, loginC, offC, yesC, noC, exitC, backC,
closeC;
    private Image not_found, exit, lockOpen, lockClose, autoClose;
    private ImageItem error, off, auto;
    private MainPage mainPage;
    private BluetoothConnect bluetoothConnect;
    private Form form, exitForm, loginForm, autoCloseForm;
    private TextField passItem;
    private String addressURL = "btspp://000A3A2EF47F:1";
    private int timeOut = 2000;
    String used = "0";

    public Mobile()
    {
        seekC = new Command("ค้นหา", Command.OK, 0);
        loginC = new Command("ตกลง", Command.OK, 0);
        offC = new Command("ปิด", Command.EXIT, 0);
        yesC = new Command("ใช่", Command.OK, 0);
        noC = new Command("ไม่", Command.EXIT, 0);
        exitC = new Command("ออก", Command.EXIT, 0);
        backC = new Command("กลับ", Command.BACK, 0);
        closeC = new Command("ตกลง", Command.OK, 0);

        try
        {
            lockOpen = Image.createImage("/image/lock_open.png");
            lockClose =
Image.createImage("/image/lock_close.png");
            exit = Image.createImage("/image/exit.png");
            off = new ImageItem("", exit, ImageItem.LAYOUT_CENTER,
"");
            not_found = Image.createImage("/image/not_found.png");
            error = new ImageItem("", not_found,
ImageItem.LAYOUT_CENTER, "");
            autoClose =
Image.createImage("/image/auto_close.png");
            auto = new ImageItem("", autoClose,
ImageItem.LAYOUT_CENTER, "");
        }
        catch(Exception ex)
        {}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        //กำหนดหน้าแรก
        mainPage = new MainPage(this);
        mainPage.addCommand(offC);
        mainPage.setCommandListener(this);

        //กำหนดหน้าการเชื่อมต่อ
        form = new Form("ผิดพลาด");
        form.append(error);
        form.append("\nแอฟพลิเคชั่นไม่สามารถใช้งานได้เนื่องจากไม่พบอุปกรณ์บลูทูธที่จะทำการเชื่อมต่อ");
        form.addCommand(seekC);
        form.addCommand(exitC);
        form.setCommandListener(this);

        //กำหนดหน้าออกจากโปรแกรม
        exitForm = new Form("");
        exitForm.append(off);
        exitForm.append("\nคุณต้องการออกจากโปรแกรมใช่หรือไม่?");
        exitForm.addCommand(yesC);
        exitForm.addCommand(noC);
        exitForm.setCommandListener(this);

        //กำหนดหน้าล็อกอิน
        loginForm = new Form("");
        passItem = new TextField("กรุณาใส่รหัสผ่าน 4 หลัก", "", 4,
TextField.PASSWORD|TextField.NUMERIC);
        loginForm.append(passItem);
        loginForm.addCommand(loginC);
        loginForm.addCommand(exitC);
        loginForm.setCommandListener(this);

        //กำหนดหน้าปิดโปรแกรมอัตโนมัติ
        autoCloseForm = new Form("");
        autoCloseForm.append(auto);
        autoCloseForm.append("\nโปรแกรมจะปิดเองโดยอัตโนมัติเนื่องจากใส่รหัสผ่านไม่ถูกต้อง");
        autoCloseForm.addCommand(closeC);
        autoCloseForm.setCommandListener(this);
    }

    public void startApp()
    {
        display = Display.getDisplay(this);
        bluetoothConnect = new BluetoothConnect(addressURL);
        seekBluetooth();
    }

    public void pauseApp()
    {}
    public void destroyApp(boolean unconditional)
    {}

    public void commandAction(Command c, Displayable d)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(c == exitC || c == yesC)
        {
            bluetoothConnect.sendCommand("0");
            bluetoothConnect.closeConnect();
            destroyApp(true);
            notifyDestroyed();
        }
        else if(c == closeC)
        {
            bluetoothConnect.closeConnect();
            destroyApp(true);
            notifyDestroyed();
        }
        else if(c == loginC)
            checkPass(passItem.getString());
        else if(c == offC)
            display.setCurrent(exitForm);
        else if(c == noC)
            display.setCurrent(mainPage);
        else if(c == seekC)
            seekBluetooth();
        else if(c == backC)
            display.setCurrent(mainPage);
    }

    public void seekBluetooth()
    {
        if(bluetoothConnect.connection())
        {
            Alert a = new Alert("", "เชื่อมต่อเรียบร้อยแล้ว", null,
AlertType.INFO);
            a.setTimeout(timeOut);
            display.setCurrent(a, loginForm);
        }
        else
        {
            Alert a = new Alert("", "การเชื่อมต่อผิดพลาด", null,
AlertType.ERROR);
            a.setTimeout(timeOut);
            display.setCurrent(a, form);
        }
    }

    public void checkPass(String password)
    {
        bluetoothConnect.sendCommand(password);
        String count = bluetoothConnect.receiveStatus();
        if(count.equals("p"))
        {
            Alert a = new Alert("", "รหัสผ่านถูกต้อง", lockOpen,
AlertType.INFO);
            a.setTimeout(timeOut);
            display.setCurrent(a, mainPage);
        }
        else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            if(count.equals("0"))
            {
                Alert a = new Alert("", "รหัสผ่านไม่ถูกต้อง", lockClose,
AlertType.WARNING);
                a.setTimeout(timeOut);
                display.setCurrent(a, autoCloseForm);
            }
            else
            {
                Alert a = new Alert("", "รหัสผ่านไม่ถูกต้อง ลองได้อีก :"+count+"
ครั้ง", lockClose, AlertType.WARNING);
                a.setTimeout(timeOut);
                display.setCurrent(a, loginForm);
            }
        }
    }

     public void sendHotKey(String channelID)
    {
        if(bluetoothConnect.sendCommand(channelID))
        {
            display.setCurrent(mainPage);
        }
        else
        {
            Alert a = new Alert("", "ไม่สามารถส่งค่าได้", null,
AlertType.INFO);
            a.setTimeout(timeOut);
            display.setCurrent(a, mainPage);
        }
    }

    public void sendHotKey(String channelID1, String channelID2)
    {
        if(bluetoothConnect.sendCommand(channelID1))
        {
            try {
                Thread.sleep(100);
                bluetoothConnect.sendCommand(channelID2);
                Thread.sleep(100);
                }
            catch (InterruptedException ex)
            {
                ex.printStackTrace();
            }
            display.setCurrent(mainPage);
        }
        else
        {
            Alert a = new Alert("", "ไม่สามารถส่งค่าได้", null,
AlertType.INFO);
            a.setTimeout(timeOut);
            display.setCurrent(a, mainPage);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    }

    public void checkWay(String way)
    {
        used = way;
    }
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MainPage.java

```
package program;

import javax.microedition.lcdui.*;

public class MainPage extends Canvas
{
    private Image bt1, bt2, bt3, center, bt7, bt8, bt9;
    private int widthImage, heightImage, width = getWidth(),
height = getHeight();
    private Font font;
    private Mobile mobile ;

    public MainPage(Mobile m)
    {
        this.mobile = m;
        widthImage = width/2;
        heightImage = (height/2)-54;
        font = Font.getFont(Font.FACE_SYSTEM,
Font.STYLE_PLAIN,Font.SIZE_LARGE); //กำหนดรูปแบบตัวอักษร
        try
        {
            bt1 = Image.createImage("/image/1.png");
            bt2 = Image.createImage("/image/2.png");
            bt3 = Image.createImage("/image/3.png");
            center = Image.createImage("/image/center.png");
            bt7 = Image.createImage("/image/7.png");
            bt8 = Image.createImage("/image/8.png");
            bt9 = Image.createImage("/image/9.png");
        }
        catch(Exception ex)
        {}
    }

    public void paint(Graphics g)
    {
        g.setColor(0, 0, 0);
        g.fillRect(0, 0, width, height); //เป็นเมธอดวาดสี่เหลี่ยมทึบ
        g.setColor(255, 255, 255);
        drawIcon(g);
        g.setFont(font);
        g.drawString("ปุ่มควบคุม", width/2, (height/2)-120, Graphics.TOP
| Graphics.HCENTER);
    }

    public void keyPressed(int key)
    {
        if (key >= 0)
        {
            switch(key)
            {   //เดินหน้า
                case KEY_NUM2:
                    if (mobile.used.equals("0")) //รถหยุดนิ่ง
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                    String channel1 = "1";
                    String channel2 = "2";
                        mobile.sendHotKey(channel1,channel2); //สั่งให้
เดินหน้า
                        mobile.checkWay("1");
                }
                 else if (mobile.used.equals("2")) //รถถอยหลังอยู่
                {
                    String channelB3 = "3";
                    String channelB4 = "4";
                        mobile.sendHotKey(channelB3,channelB4); //
ยกเลิกการถอยหลัง
                    String channel1 = "1";
                    String channel2 = "2";
                        mobile.sendHotKey(channel1,channel2); //สั่งให้
เดินหน้า
                        mobile.checkWay("1");
                }
                 else if(mobile.used.equals("1")) //รถเดินหน้าอยู่
                {
                    mobile.checkWay("0");
                    String channel1 = "1";
                    String channel2 = "2";
                        mobile.sendHotKey(channel1,channel2); //สั่งให้
หยุด
                }
                break;

                //ถอยหลัง
                case KEY_NUM8:
                if (mobile.used.equals("0")) //รถหยุดนิ่ง
                {
                    String channel3 = "3";
                    String channel4 = "4";
                        mobile.sendHotKey(channel3,channel4); //สั่งให้
ถอยหลัง
                        mobile.checkWay("2");
                }
                else if (mobile.used.equals("1")) //รถเดินหน้าอยู่
                {
                    String channelF1 = "1";
                    String channelF2 = "2";
                        mobile.sendHotKey(channelF1,channelF2); //
ยกเลิกการเดินหน้า
                    String channel3 = "3";
                    String channel4 = "4";
                        mobile.sendHotKey(channel3,channel4); //สั่งให้
ถอยหลัง
                        mobile.checkWay("2");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
else if(mobile.used.equals("2")) //รถถอยหลังอยู่
{
    mobile.checkWay("0");
    String channel3 = "3";
    String channel4 = "4";
        mobile.sendHotKey(channel3,channel4); //สั่งให้
หยุด
}
break;

//วนหน้าซ้าย
case KEY_NUM1:
if(mobile.used.equals("1")) //เลี้ยวซ้ายขณะรถเดินหน้า
{
    String channel5 = "2";
        mobile.sendHotKey(channel5);
}
else
{
    String channel5 = "1";
        mobile.sendHotKey(channel5);
}
break;

//วนหน้าขวา
case KEY_NUM3:
if(mobile.used.equals("1")) //เลี้ยวขวาขณะรถเดินหน้า
{
    String channel6 = "1";
        mobile.sendHotKey(channel6);
}
else
{
    String channel6 = "2";
        mobile.sendHotKey(channel6);
}
break;

//วนหลังตามเข็ม
case KEY_NUM7:
if(mobile.used.equals("2")) //หมุนตามเข็มขณะรถถอยหลัง
{
    String channel7 = "4";
        mobile.sendHotKey(channel7);
}
else
{   String channel7 = "3";
        mobile.sendHotKey(channel7);
}
break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                //วนหลังทวนเข็ม
                case KEY_NUM9:
                if(mobile.used.equals("2")) //หมุนทวนเข็มขณะรถถอยหลัง
                {
                    String channel8 = "3";
                        mobile.sendHotKey(channel8);
                }
                else
                {
                    String channel8 = "4";
                        mobile.sendHotKey(channel8);
                }
                break;

                case KEY_NUM5:
                    String channel5 = "5";
                        mobile.sendHotKey(channel5);
                        mobile.checkWay("0");
                break;
            }
        }
}

public void keyReleased(int key)
{
    if (key >= 0)
    {
        switch(key)
        {
            //วนหน้าซ้าย
            case KEY_NUM1:
            if(mobile.used.equals("1")) //เลี้ยวซ้ายขณะรถเดินหน้า
            {
                String channel5 = "2";
                    mobile.sendHotKey(channel5);
            }
            else
            {
                String channel5 = "1";
                    mobile.sendHotKey(channel5);
            }
            break;

            //วนหน้าขวา
            case KEY_NUM3:
            if(mobile.used.equals("1")) //เลี้ยวขวาขณะรถเดินหน้า
            {
                String channel6 = "1";
                    mobile.sendHotKey(channel6);
            }
            else
            {
                String channel6 = "2";
                    mobile.sendHotKey(channel6);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                break;

                //วนหลังตามเข็ม
                case KEY_NUM7:
                if(mobile.used.equals("2")) //หมุนตามเข็มขณะรถถอยหลัง
                {
                    String channel7 = "4";
                        mobile.sendHotKey(channel7);
                }
                else
                {   String channel7 = "3";
                        mobile.sendHotKey(channel7);
                }
                break;

                //วนหลังทวนเข็ม
                case KEY_NUM9:
                if(mobile.used.equals("2")) //หมุนทวนเข็มขณะรถถอยหลัง
                {
                    String channel8 = "3";
                        mobile.sendHotKey(channel8);
                }

                else
                {
                    String channel8 = "4";
                        mobile.sendHotKey(channel8);
                }
                break;
            }
        }
    }

    public void drawIcon(Graphics g)
    {
        g.drawImage(bt1, widthImage-70, heightImage,
Graphics.HCENTER | Graphics.VCENTER);
        g.drawImage(bt2, widthImage, heightImage,
Graphics.HCENTER| Graphics.VCENTER);
        g.drawImage(bt3, widthImage+70, heightImage,
Graphics.HCENTER | Graphics.VCENTER);
        g.drawImage(center, widthImage, heightImage+65,
Graphics.HCENTER | Graphics.VCENTER);
        g.drawImage(bt7, widthImage-70, heightImage+130,
Graphics.HCENTER | Graphics.VCENTER);
        g.drawImage(bt8, widthImage, heightImage+130,
Graphics.HCENTER | Graphics.VCENTER);
        g.drawImage(bt9, widthImage+70, heightImage+130,
Graphics.HCENTER | Graphics.VCENTER);
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BluetoothConnect.java

```
package program;

import javax.bluetooth.*;
import javax.microedition.io.*;
import java.io.*;

public class BluetoothConnect
{
    private LocalDevice local;
    private DiscoveryAgent agent;
    private StreamConnection con;
    private OutputStream out;
    private InputStream in;
    private boolean connected = false;
    private String conStr;

    public BluetoothConnect(String address)
    {
        try
        {
            local = LocalDevice.getLocalDevice();
            agent = local.getDiscoveryAgent();
            local.setDiscoverable(DiscoveryAgent.GIAC);
            this.conStr = address;
        }
        catch(Exception ex)
        {}
    }


    public boolean connection()
    {
        try
        {
            if(conStr != null)
            {
                con = (StreamConnection)Connector.open(conStr);
                connected = true;
            }
        }
        catch(Exception ex)
        {   connected = false;  }
        return connected;
    }

    public boolean sendCommand(String channel)
    {
        boolean send = false;
        channel += " ";
        try
        {
            out = con.openOutputStream();
            out.write(channel.getBytes());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งนี้ต้องได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            out.close();
            send = true;
        }
        catch(Exception ex)
        {}
        return send;
    }


    public String receiveStatus()
    {
        String status = null;
        try
        {
            in = con.openInputStream();
            byte[] buffer = new byte[80];
            int buffer_read = in.read(buffer);
            in.close();
            status = new String(buffer, 0, buffer_read);
        }
        catch(Exception ex)
        {   status = "error";   }
        return status;
    }

    public void closeConnect()
    {
        try
        {
            con.close();
            connected = false;
        }
        catch(Exception ex)
        {}
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

โค้ดโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

roboStamp.bsp

```
' {$STAMP BS2p}
' {$PBASIC 2.5}
' {$PORT COM6}

s1 PIN 7
s2 PIN 6
s3 PIN 5
s4 PIN 4
channel VAR Nib
number VAR Nib
password VAR Word
continue VAR Nib
limit CON 3

Main:
  DEBUG "Wait for password.....",CR
  GOSUB checkPassword
  DEBUG "Work.....",CR
  GOSUB work
GOTO Main

checkPassword:
  FOR number = 1 TO limit
    SERIN 1,240,[DEC password]
    DEBUG "Password = ",DEC password,CR
    SELECT password
      CASE 9999
        SEROUT 0,240,["p"]
        DEBUG "Password correct. Start program.....",CR
        'SERIN 1,240,[DEC port]
        'DEBUG DEC port
        RETURN
      CASE ELSE
        continue = limit-number
        SEROUT 0,240,[DEC continue]
        DEBUG "Password wrong. Continue ",DEC continue,CR
        SELECT continue
          CASE 0
               GOTO Main
        ENDSELECT
    ENDSELECT
  NEXT
GOTO checkPassword

work:
  SERIN 1,240,[DEC channel]
  DEBUG "Mobile send channel : ",DEC channel,". Status of ",DEC
channel
  SELECT channel
    CASE 0
      DEBUG ". Mobile was disconnected.",CR,CR,CR
      RETURN
    CASE 1
      DEBUG " change from ",DEC s1," to "
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

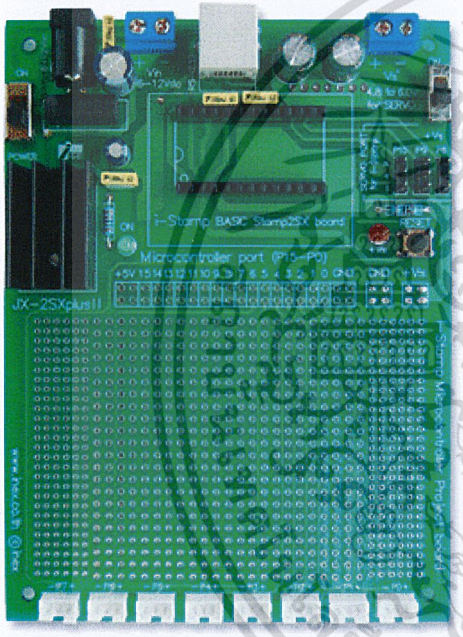
```
      TOGGLE s1
      DEBUG DEC s1,CR
    CASE 2
      DEBUG " change from ",DEC s2," to "
      TOGGLE s2
      DEBUG DEC s2,CR
    CASE 3
      DEBUG " change from ",DEC s3," to "
      TOGGLE s3
      DEBUG DEC s3,CR
    CASE 4
      DEBUG " change from ",DEC s4," to "
      TOGGLE s4
      DEBUG DEC s4,CR
    CASE 5
      DEBUG " change from open to close.",CR
      LOW s1
      LOW s2
      LOW s3
      LOW s4
  ENDSELECT
GOTO work
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

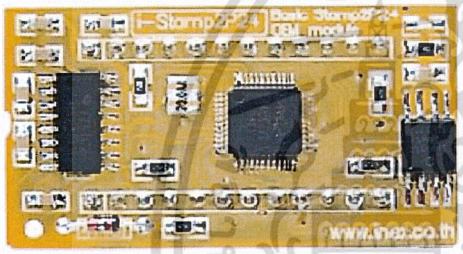
ภาคผนวก ค.

คุณสมบัติของอุปกรณ์ที่ใช้ในโครงงานปัญหาพิเศษ

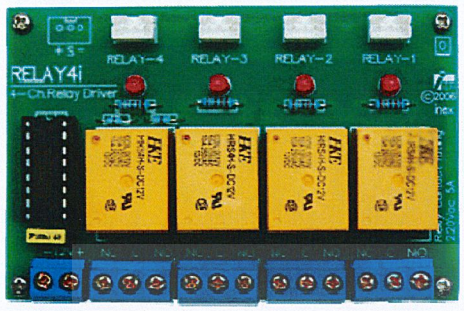
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ	คุณสมบัติ
	**แผงวงจรสำหรับพัฒนาโครงงาน JX-2SX Plus II** • มีคอนเน็กเตอร์ตัวเมีย 24 ขา สำหรับติดตั้ง i-Stamp / i-Stamp 2P24 • เชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม RS-232 • มีจุดต่อขาพอร์ตของ i-Stamp ครบทั้ง 16 ขา จัดสรรขาพอร์ต P0 ถึง P7 เป็นคอนเน็กเตอร์ JST สำหรับต่อกับแผงวงจรสนับสนุนและตัวตรวจจับของ inex • จัดสรรขาพอร์ต P8 ถึง P10 สำหรับเชื่อมต่อกับ RC เซอร์โวมอเตอร์ขนาดเล็ก โดยมีจุดต่อไฟเลี้ยงสำหรับ เซอร์โวมอเตอร์แยกต่างหาก • มีสวิตช์ RESET • มีจุดต่อไฟเลี้ยง 2 จุด • จุดจ่ายไฟเลี้ยงสำหรับ i-Stamp 2P24 เป็นแจ๊กอะแดปเตอร์ และเทอร์มินอลบล็อก 2 ขา รองรับไฟเลี้ยงในย่าน +6 ถึง 12V พร้อมสวิตช์เปิด-ปิด มีวงจรควบคุมไฟเลี้ยงคงที่ที่ +5V มีไฟแสดงสภาวะไฟเลี้ยง • จุดจ่ายไฟเลี้ยงสำหรับเซอร์โวมอเตอร์ เป็นเทอร์มินอลบล็อก 2 ขา รองรับไฟเลี้ยงในย่าน +4.8 ถึง 6V พร้อมสวิตช์เปิด-ปิด และมีไฟแสดงสภาวะไฟเลี้ยง • พื้นที่สำหรับพัฒนาวงจร (Proto area) ขนาด 8.5 x 5.5 เซนติเมตร • สามารถติดตั้งแผงต่อวงจรหรือเบรดบอร์ดขนาดเล็กได้ • มีจุดบัดกรีมีทั้งแบบระยะห่าง 100 มิล หรือ 25.4 มิลลิเมตร และ 80 มิล หรือ 20 มิลลิเมตร • ขนาด 9.5 x 12.5 เซนติเมตร • ราคา 470 บาท

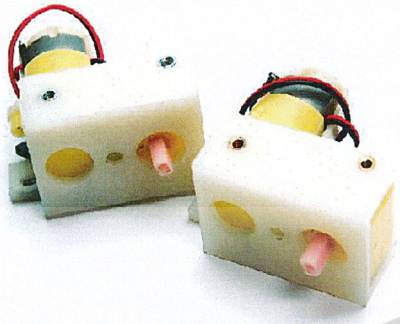
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ	คุณสมบัติ
	**บอร์ดไมโครคอนโทรลเลอร์ i-Stamp 2P24** • ความถี่สัญญาณนาฬิกา 20MHz ใช้เซรามิกเรโซเนเตอร์ 20MHz ในการกำเนิดสัญญาณนาฬิกา • ความเร็วในการประมวลผล 12,000 คำสั่งภาษาพีเบสิกต่อวินาที • ขนาดหน่วยความจำ 16 กิโลไบต์แบ่งเป็น 8 ส่วน ส่วนละ 2 กิโลไบต์ทำให้สามารถเก็บโปรแกรมสำหรับทำงานที่แตกต่างและไม่เกี่ยวข้องกันได้มากถึง 8 โปรแกรม รวมทั้งยังสามารถเรียกให้ทำงานต่อเนื่องกันทั้ง 8 โปรแกรมได้ • หน่วยความจำแรม 32 ไบต์ • หน่วยความจำแรมสแครตช์แพด 128 ไบต์ • ใช้ไฟเลี้ยง +5V ต้องการกระแส 65mA ในขณะทำงานและ 60mA ในโหมดสลีป • เชื่อมกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม RS-232 สำหรับดาวน์โหลดโปรแกรมและสื่อสารข้อมูล • จำนวนพอร์ตอินพุตเอาท์พุต 16 ช่อง ทำงานอิสระจากกัน หรือจัดเป็นกลุ่มได้ตั้งแต่ 4, 8 และ 16 ขา • ความสามารถในการจ่ายกระแสซอร์ส/ซิงก์ต่อขา 30mA/30mA • พัฒนาด้วยภาษาพีเบสิก • ขาเชื่อมต่อด้วยคอนเน็กเตอร์ IDC ตัวผู้ชุบทองอย่างดีสามารถใช้กับคอนเน็กเตอร์ IDC ตัวเมียได้ • มีไฟแสดงสภาวะการทำงาน • ขนาดของบอร์ด 1 x 1.7 นิ้ว • ราคา 790 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ	คุณสมบัติ
	**แผงวงจรขับรีเลย์ 4 ช่อง (Relay 4i)** • ใช้ไอซีขับโหลดกระแสสูงบนบอร์ดจัดวงจรเพื่อขับรีเลย์ 12V 4 ช่อง • ใช้ไฟเลี้ยง +12V แยกต่างหาก • รับสัญญาณลอจิก “1” ในการกระตุ้นให้รีเลย์ทำงาน • มีไฟแสดงการทำงานของรีเลย์ • อัตราทนได้ของหน้าสัมผัสรีเลย์ 220V AC 5A สามารถรองรับโหลดได้ไม่เกิน 300 วัตต์
	แผงวงจรสื่อสารข้อมูลไร้สายผ่านโมดูลบลูทูธแบบสเลฟ (ZX-Bluetooth) • ใช้โมดูล MB-CB04 บลูทูธแบบสเลฟ (เป็นตัวลูก) คลาส 2 มีรหัสประจำตัวเฉพาะ • มีสายอากาศในตัว • รัศมีทำการสูงสุด 10 เมตร • อัตราเร็วในการถ่ายทอดข้อมูลหรืออัตราบอด 9,600 บิตต่อวินาที รูปแบบข้อมูล 8N1 (8 บิตข้อมูล ไม่มีการตรวจสอบพาริตี้ และ 1 บิตหยุด) • มีจุดต่อ TxD สำหรับส่งข้อมูลออก และ RxD สำหรับรับข้อมูลอนุกรม • ใช้ไฟเลี้ยง +5V บนแผงวงจรมีวงจรควบคุมไฟเลี้ยงคงที่ที่ +3.3V สำหรับเลี้ยงโมดูลบลูทูธ • ขนาด 3x4 เซนติเมตร • ราคา 1,190 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ	คุณสมบัติ
	**มอเตอร์ ไฟตรงพร้อมชุดเฟืองขับ** • ชุดมอเตอร์พร้อมเฟืองขับ อัตราทด 48:1 มีสายต่อ 2 เส้น • ต้องการไฟเลี้ยงในย่าน +3 ถึง +9V DC • กินกระแสไฟฟ้า 130mA (ที่ไฟเลี้ยง +6V และไม่โหลด) • ความเร็วเฉลี่ย 170 ถึง 250 รอบต่อนาที (ที่ไฟเลี้ยง +6V และไม่มีโหลด) • น้ำหนัก 30 กรัม • แรงบิดต่ำสุด 0.5 กิโลกรัม-เซนติเมตร • ติดตั้งกับตัวยึดพลาสติกแบบมีพุกสำหรับยึดในตัว • ขนาด (กว้าง x ยาว x สูง) 42 x 45 x 22.7 มิลลิเมตร
	โทรศัพท์มือถือ • Bluetooth เวอร์ชัน 2.0 • รองรับ Java Application (MIDP 2.0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้