รายงานการวิจัย พัฒนานวัตกรรมและสิ่งประดิษฐ์

หุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

(Tennis Training Robot)

หัวหน้าโครงการ

ผู้ช่วยศาสตราจารย์ ดร. กองพล อารีรักษ์

สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ได้รับทุนอุดหนุนการวิจัย พัฒนานวัตกรรมและสิ่งประดิษฐ์จาก

กองทุนนวัตกรรมและสิ่งประดิษฐ์ สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี

ประจำปี พ.ศ. 2555

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว

กันยายน 2555

ศูนย์บรรณสารและสื่อการศึกษา

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทคัดย่อ

หุ่นยนต์ฝึกซ้อมกีฬาเทนนิสที่สร้างขึ้นนี้ มีวัตถุประสงค์ที่สำคัญ คือ ใช้สำหรับการฝึกซ้อมกีฬาเทนนิสด้วยตนเอง โดยเฉพาะอย่างยิ่งการซ้อมในตำแหน่งการตีลูกเทนนิสที่เป็นข้อบกพร่องของนักกีฬาแต่ละคน ดังนั้นการออกแบบหุ่นยนต์ดังกล่าวจึงมีลักษณะการทำงานที่ตอบสนองความต้องการของนักกีฬาอย่างแท้จริง โดยสมรรถนะของหุ่นยนต์สามารถปรับระดับการยิงลูกเทนนิสได้ทั้งแนวราบและแนวดิ่ง รวมถึงการปรับความแรงในการยิงลูกได้ตามต้องการ นอกจากนี้ยังสามารถยิงลูกไซส์โค้งและลูกพุ่งตรงได้อีกด้วย การควบคุมการทำงานของหุ่นยนต์ใช้บอร์ดไมโครคอนโทรลเลอร์ โดยรายละเอียดการใช้งานของหุ่นยนต์ไม่ซับซ้อน เหมาะกับนักกีฬาที่ไม่มีพื้นฐานทางด้านวิศวกรรม หุ่นยนต์ตัวนี้สามารถนับลูกเทนนิสที่ตีได้ในแต่ละวันผ่านทางหน้าจอที่ติดตั้งไว้บนตัวหุ่นยนต์ แหล่งพลังงานสำหรับหุ่นยนต์ คือ แบตเตอรี ที่สามารถชาร์จพลังงานได้ตลอดเวลา ผลงานสิ่งประดิษฐ์ที่สร้างขึ้นนี้ได้ดำเนินการยื่นจดสิทธิบัตรการออกแบบผลิตภัณฑ์จำนวน 5 ผลงาน

มหาวิทยาลัยเทคโนโลยีสุรนารี

กิตติกรรมประกาศ

โครงการวิจัย พัฒนานวัตกรรมและสิ่งประดิษฐ์ เรื่อง หุ่นยนต์ฝึกซ้อมกีฬาเทนนิส สามารถสำเร็จลุล่วงไปด้วยดี ทั้งนี้ต้องขอขอบคุณ กองทุนนวัตกรรมและสิ่งประดิษฐ์ สมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี และมหาวิทยาลัยเทคโนโลยีสุรนารี ที่ให้ทุนสนับสนุนการทำสิ่งประดิษฐ์นี้ นอกจากนี้ผู้วิจัยต้องขอขอบคุณนายกมล ไชยศรี และนายสุทธิศักดิ์ เหลาสิงห์ ที่เป็นผู้ช่วยในการดำเนินการสร้างหุ่นยนต์ดังกล่าว ด้วยความทุ่มเท และการเอาใจใส่อย่างยิ่ง

กองพล อารีรักษ์

กันยายน 2555

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
บทที่ 1 บทนำ	
1.1 ชื่อโครงการ	1
1.2 คณะผู้วิจัย	1
1.3 สาขาวิชาและกลุ่มวิชาที่ทำวิจัย	1
1.4 คำสำคัญของเรื่องที่ทำการวิจัย	1
1.5 ความสำคัญและที่มาของปัญหาที่ทำการวิจัย	1
1.6 วัตถุประสงค์ของโครงการ	2
1.7 ประโยชน์ที่คาดว่าจะได้รับและหน่วยงานที่นำผลการวิจัยไปใช้ประโยชน์	2
1.8 ทฤษฎีหรือกรอบแนวความคิดของโครงการวิจัย	2
1.9 ระเบียบวิธีวิจัย	2
1.10 ขอบเขตของโครงการ	3
1.11 ระยะเวลาที่ทำการวิจัย	3
1.12 แผนการดำเนินงานตลอดโครงการ	3
1.13 อุปกรณ์ที่จำเป็นสำหรับโครงการ	4
บทที่ 2 แบบทางวิศวกรรมของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส	
2.1 ชุดเก็บและป้อนลูกเทนนิสของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส	5
2.2 ชุดปรับระดับในแนวดิ่งของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส	6
2.3 ชุดปรับระดับในแนวราบของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส	7
2.4 ชุดยิงลูกเทนนิสของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส	8
2.5 หุ่นยนต์ฝึกซ้อมกีฬาเทนนิส	8
2.6 สรุป	10
บทที่ 3 บอร์ด ET-EASY MEGA1280 (Duino Mega) สำหรับหุ่นยนต์อัตโนมัติ	
3.1 กล่าวนำ	11
3.2 โครงสร้างบอร์ด ET-EASY MEGA1280 (Duino Mega)	11

สารบัญ (ต่อ)

	หน้า
3.3 คุณสมบัติของบอร์ด	13
3.4 การติดตั้ง Driver ของ USB Bridge ของบอร์ด ET-EASY MEGA1280 (Duino Mega)	15
3.5 การติดตั้งโปรแกรม Arduino	21
3.6 สรุป	29
บทที่ 4 วงจรขับเคลื่อนและการควบคุม	
4.1 กล่าวนำ	30
4.2 วงจรควบคุมมอเตอร์	30
4.3 ชุดควบคุมมอเตอร์ไฟฟ้ากระแสตรงสำหรับหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส	33
4.4 การใช้สัญญาณ PWM จากบอร์ดไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 สำหรับควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรง	34
4.5 การโปรแกรมเพื่อควบคุมมอเตอร์ไฟฟ้ากระแสตรง	38
4.6 สรุป	45
บทที่ 5 คู่มือการใช้งานหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส	47
เอกสารอ้างอิง	59
ประวัติผู้วิจัย	60

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 1

บทนำ

1. **ชื่อโครงการ (ภาษาไทย)** หุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

   (ภาษาอังกฤษ) Tennis Training Robot

2. **คณะผู้วิจัย**

   หัวหน้าโครงการวิจัย ผู้ช่วยศาสตราจารย์ ดร.กองพล อารีรักษ์

3. **สาขาวิชาและกลุ่มวิชาที่ทำวิจัย**

   สาขาวิชาวิศวกรรมไฟฟ้า กลุ่มวิชาวิศวกรรมศาสตร์

4. **คำสำคัญของเรื่องที่ทำการวิจัย**

   หุ่นยนต์, กีฬาเทนนิส

5. **ความสำคัญและที่มาของปัญหาที่ทำการวิจัย**

   ปัจจุบันกีฬาเทนนิสได้รับความนิยมอย่างมากสำหรับประชาชนในประเทศ การนำเทคโนโลยีไปสนับสนุนการฝึกซ้อมกีฬาดังกล่าวยังไม่ปรากฏแพร่หลาย และเนื่องจากกีฬาชนิดนี้อย่างน้อยต้องมีผู้เล่นสองคน ด้วยเหตุนี้การฝึกซ้อมทักษะการตีเทนนิสด้วยตัวคนเดียวจึงเป็นเรื่องยาก โดยเฉพาะอย่างยิ่งในกรณีที่นักกีฬาเทนนิสต้องการฝึกซ้อมเพียงคนเดียวไม่ว่าจะฝึกการตีหน้ามือ หรือหลังมือ รวมถึงลูกเล่นต่าง ๆ ที่ต้องอาศัยเวลาในการฝึกฝนจนชำนาญ การประดิษฐ์และพัฒนาหุ่นยนต์เพื่อฝึกซ้อมกีฬาดังกล่าวจึงเป็นสิ่งที่จำเป็นในการนำเทคโนโลยีไปพัฒนาทางด้านกีฬา นอกจากนี้การควบคุมหุ่นยนต์สามารถทำได้โดยการโปรแกรมบนบอร์ดไมโครคอนโทรลเลอร์ เพื่อควบคุมทิศทางของลูกเทนนิสให้เป็นไปตามที่ต้องการ และถ้านักกีฬาต้องการฝึกตีในลักษณะเดิมซ้ำ ๆ ก็สามารถโปรแกรมตามที่ต้องการได้ ลักษณะของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิสดังกล่าวจึงเป็นสิ่งที่น่าสนใจในการพัฒนานักกีฬาเทนนิสในทุกระดับ ไม่ว่าจะเป็นระดับโรงเรียน จังหวัด เขต มหาวิทยาลัย หรือทีมชาติ เป็นต้น

6. วัตถุประสงค์ของโครงการ

6.1 เพื่อสร้างหุ่นยนต์สำหรับฝึกซ้อมกีฬาเทนนิส

6.2 เพื่อสร้างองค์ความรู้ รวมถึงเทคโนโลยีในการสร้างหุ่นยนต์

7. ประโยชน์ที่คาดว่าจะได้รับ และหน่วยงานที่นำผลการวิจัยไปใช้ประโยชน์

7.1 ได้หุ่นยนต์ต้นแบบสำหรับฝึกซ้อมกีฬาเทนนิส

7.2 ได้สิทธิบัตรการออกแบบผลิตภัณฑ์

7.3 ได้องค์ความรู้ และเทคโนโลยีในการสร้างหุ่นยนต์ เพื่อการผลิตเชิงพาณิชย์ในอนาคต

7.4 หน่วนงานที่สามารถนำผลการวิจัยไปใช้ประโยชน์

- ชมรมเทนนิสตามมหาวิทยาลัยต่าง ๆ
- สมาคมเทนนิสประจำจังหวัดต่าง ๆ
- ศูนย์ฝึกเทนนิสต่าง ๆ
- สมาคมเทนนิสแห่งประเทศไทย

8. ทฤษฎีหรือกรอบแนวความคิดของโครงการวิจัย

การออกแบบหุ่นยนต์ฝึกซ้อมกีฬาเทนนิสดังกล่าว ต้องอาศัยองค์ประกอบที่สำคัญสองส่วนด้วยกัน คือ โครงสร้างของหุ่นยนต์ และตัวประมวลผลเพื่อควบคุมหุ่นยนต์ ในการออกแบบโครงสร้างของหุ่นยนต์ต้องสามารถรองรับการใช้งานได้อย่างหลากหลาย เช่น ความจุของลูกเทนนิสของหุ่น วิถีการยิงลูกเทนนิส ความแรงการยิงลูกเทนนิส เป็นต้น ส่วนตัวประมวลผลเพื่อควบคุมหุ่นยนต์จะใช้บอร์ดไมโครคอนโทรลเลอร์ในการประมวลผลเพื่อควบคุมมอเตอร์ไฟฟ้าตามจุดต่าง ๆ ให้ได้ตามที่ต้องการ เช่น ทิศทางการยิงลูกเทนนิส ตำแหน่งในการยิง เป็นต้น นอกจากนี้ปัจจัยที่ต้องนำมาพิจารณาในการสร้างหุ่นดังกล่าวที่สำคัญ ได้แก่ ความสูงของหุ่น วัสดุที่ใช้ กำลังของมอเตอร์ไฟฟ้า ต้นทุนที่ใช้ และความสามารถเสริมอื่น ๆ เป็นต้น

9. ระเบียบวิธีวิจัย

การสร้างหุ่นยนต์ดังกล่าวจะเริ่มต้นจากการออกแบบโครงสร้างหุ่นยต์ เพื่อตอบโจทย์ต่าง ๆ ในการใช้งาน ซึ่งการออกแบบโครงสร้างหุ่นยนต์นี้จะอาศัยคอมพิวเตอร์ในการ

ออกแบบ หลังจากออกแบบแล้วจะดำเนินการสร้างโครงสร้างของหุ่นยนต์ตามที่ได้ออกแบบ และเมื่อโครงสร้างหุ่นยนต์เสร็จเรียบร้อยจะดำเนินการสร้างชุดวงจร และชุดควบคุมทางไฟฟ้า เพื่อควบคุมการยิงลูกเทนนิสของหุ่นยนต์ หลังจากนั้นจะมีการทดสอบจริงที่สนามเทนนิส รวมถึงปรับปรุงแก้ไขข้อผิดพลาดให้ได้ตามวัตถุประสงค์

0. ขอบเขตของโครงการ

10.1 หุ่นยนต์ที่สร้างขึ้นสำหรับฝึกซ้อมกีฬาเทนนิสเท่านั้น

10.2 หุ่นยนต์ที่สร้างขึ้นสามารถเลือกวิถีการยิงลูกเทนนิสได้อย่างอิสระ โดยการโปรแกรมด้วยบอร์ดไมโครคอนโทรลเลอร์

1. ระยะเวลาที่ทำการวิจัย 1 ปี

2. แผนการดำเนินงานตลอดโครงการ (ระยะเวลาที่ทำการวิจัย: 1 ปีนับตั้งแต่วันที่อนุมัติโครงการวิจัย)

กิจกรรม	เดือนที่											
	1	2	3	4	5	6	7	8	9	10	11	12
1. ออกแบบโครงสร้างหุ่นยนต์ด้วยคอมพิวเตอร์	▬	▬										
2. เลือกวัสดุอุปกรณ์ที่ใช้ในการสร้างหุ่นยนต์ รวมถึงอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ในการควบคุม			▬									
3. สร้างโครงสร้างหุ่นยนต์ตามรูปแบบที่ได้ออกแบบไว้				▬	▬	▬						
4. เขียนโปรแกรมควบคุมการทำงานของหุ่นยนต์ด้วยบอร์ดไมโครคอนโทรลเลอร์							▬	▬				

5. ติดตั้งมอเตอร์ไฟฟ้าตามจุดต่าง ๆ พร้อมวงจรที่เกี่ยวข้อง								▬	▬			
6. ทดสอบหุ่นยนต์ทั้งระบบในห้องปฏิบัติการ										▬	▬	
7. ทดสอบหุ่นยนต์ที่สนามเทนนิสจริง											▬	▬
8. จัดทำรายงานวิจัยฉบับสมบูรณ์									▬	▬	▬	▬

13. อุปกรณ์ที่จำเป็นสำหรับโครงการ

13.1 บอร์ดไมโครคอนโทรลเลอร์สำหรับประมวลผลควบคุมการทำงานของหุ่นยนต์

13.2 มอเตอร์ไฟฟ้าสำหรับควบคุมวิถีการยิงลูกเทนนิส

13.3 อุปกรณ์อิเล็กทรอนิกส์ทางไฟฟ้า

13.4 วัสดุสำหรับทำโครงสร้างหุ่นยนต์

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 2

แบบทางวิศวกรรมของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

การออกแบบหุ่นยนต์ฝึกซ้อมกีฬาเทนนิสสำหรับโครงงานนี้ได้ออกแบบโดยการเขียนแบบด้วยคอมพิวเตอร์โดยสามารถแบ่งออกได้เป็น 5 ส่วนประกอบด้วยกันดังนี้

2.1 ชุดเก็บและป้อนลูกเทนนิสของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

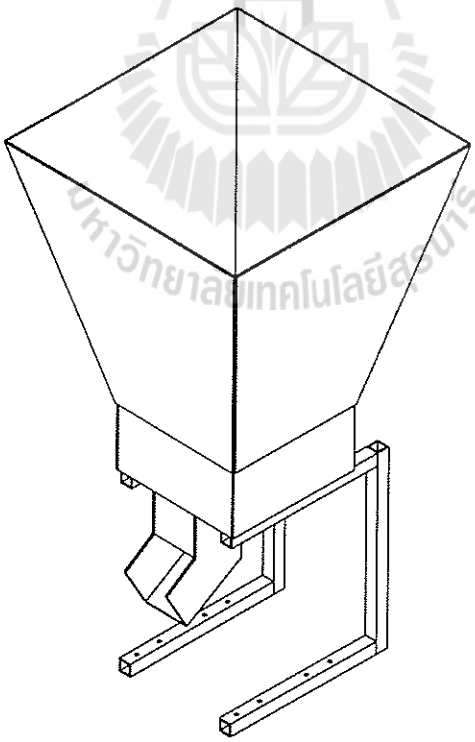
การออกแบบในส่วนนี้ได้มีการยื่นจดสิทธิบัตรการออกแบบผลิตภัณฑ์ โดยมีรายละเอียดดังต่อไปนี้

ชื่อเรื่อง ชุดเก็บและป้อนลูกเทนนิสของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

เลขที่คำขอ 1202001473

วันที่ยื่นคำขอ 31 พฤษภาคม 2555

ลักษณะของชุดเก็บและป้อนลูกเทนนิสดังกล่าวแสดงได้ดังรูปที่ 2.1 ดังนี้



รูปที่ 2.1 ภาพสามมิติของชุดเก็บและป้อนลูกเทนนิส

2.2 ชุดปรับระดับในแนวดิ่งของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

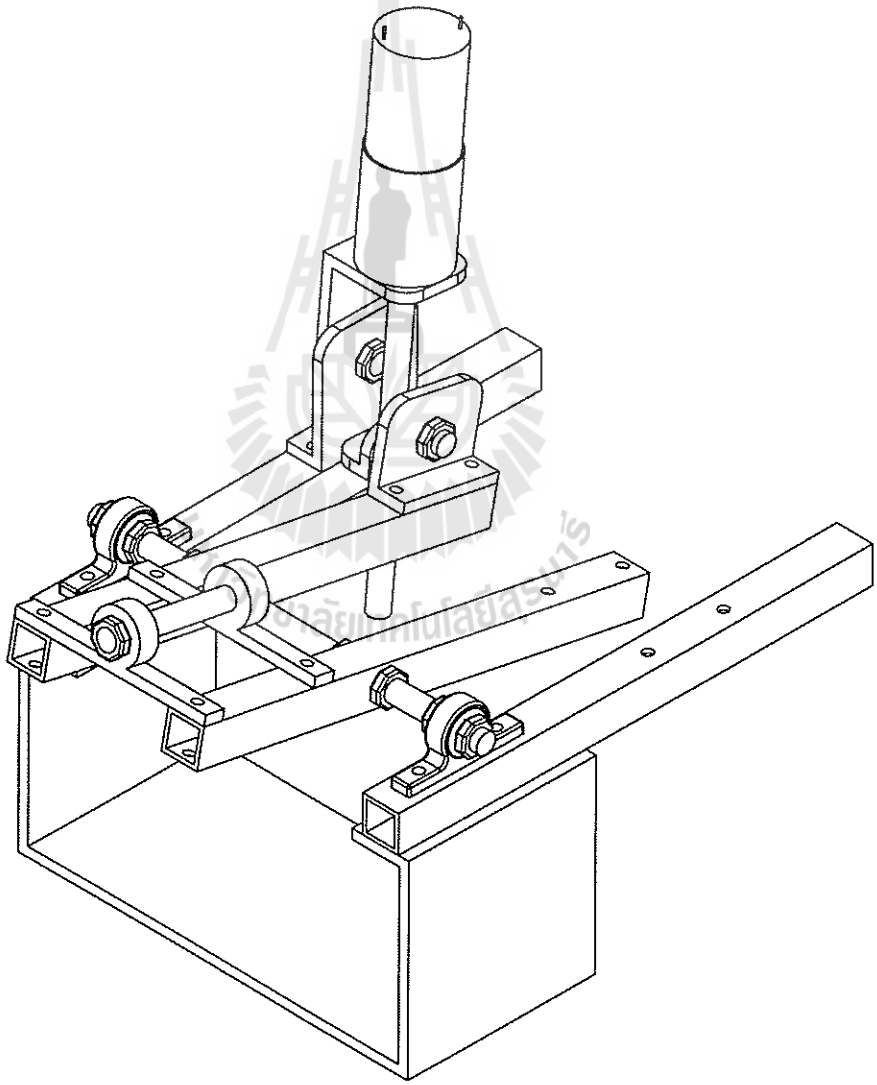
การออกแบบในส่วนนี้ได้มีการยื่นจดสิทธิบัตรการออกแบบผลิตภัณฑ์ โดยมีรายละเอียดดังต่อไปนี้

ชื่อเรื่อง ชุดปรับระดับในแนวดิ่งของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

เลขที่คำขอ 1202001474

วันที่ยื่นคำขอ 31 พฤษภาคม 2555

ลักษณะของชุดปรับระดับในแนวดิ่งดังกล่าวแสดงได้ดังรูปที่ 2.2 ดังนี้



รูปที่ 2.2 ภาพสามมิติของชุดปรับระดับในแนวดิ่งของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

2.3 ชุดปรับระดับในแนวราบของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

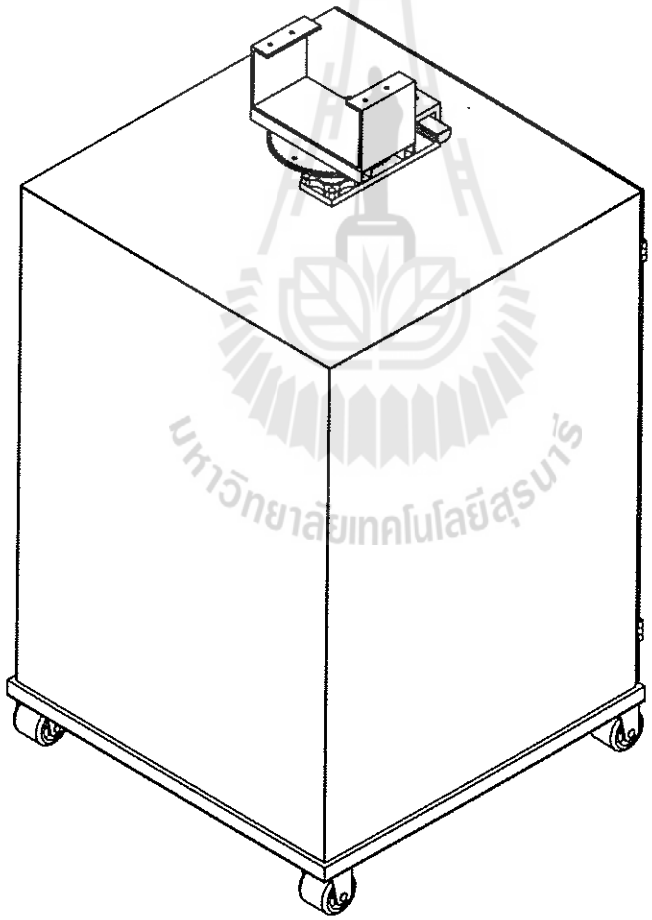
การออกแบบในส่วนนี้ได้มีการยื่นจดสิทธิบัตรการออกแบบผลิตภัณฑ์ โดยมีรายละเอียดดังต่อไปนี้

ชื่อเรื่อง ชุดปรับระดับในราบของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

เลขที่คำขอ 1202001475

วันที่ยื่นคำขอ 31 พฤษภาคม 2555

ลักษณะของชุดปรับระดับในแนวราบดังกล่าวแสดงได้ดังรูปที่ 2.3 ดังนี้



รูปที่ 2.3 ภาพสามมิติของชุดปรับระดับในแนวราบของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

2.4 ชุดยิงลูกเทนนิสของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

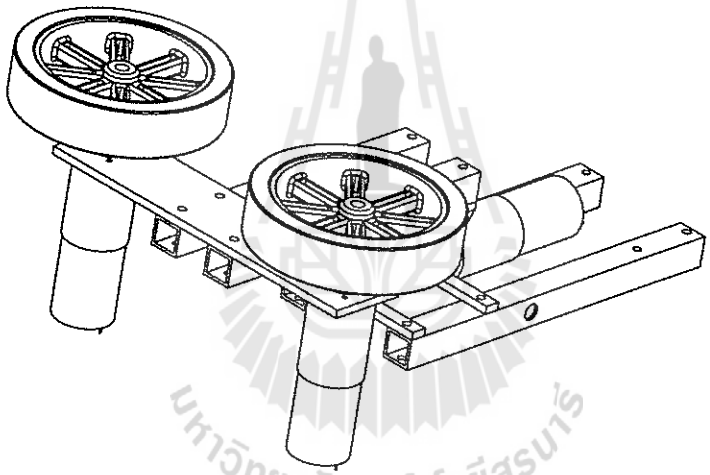
การออกแบบในส่วนนี้ได้มีการยื่นจดสิทธิบัตรการออกแบบผลิตภัณฑ์ โดยมีรายละเอียดดังต่อไปนี้

ชื่อเรื่อง ชุดยิงลูกเทนนิสของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

เลขที่คำขอ 1202001476

วันที่ยื่นคำขอ 31 พฤษภาคม 2555

ลักษณะของชุดยิงลูกเทนนิสดังกล่าวแสดงได้ดังรูปที่ 2.4 ดังนี้



รูปที่ 2.4 ภาพสามมิติของชุดยิงลูกเทนนิสของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

2.5 หุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

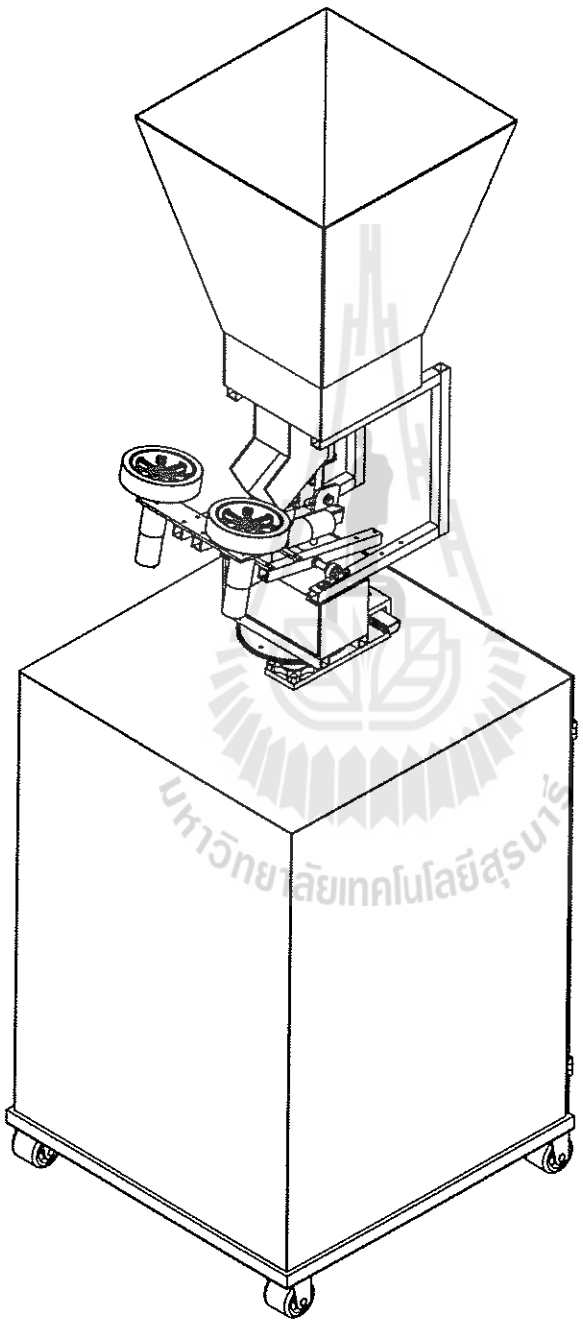
การออกแบบในส่วนนี้ได้มีการยื่นจดสิทธิบัตรการออกแบบผลิตภัณฑ์ โดยมีรายละเอียดดังต่อไปนี้

ชื่อเรื่อง หุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

เลขที่คำขอ 1202001477

วันที่ยื่นคำขอ 31 พฤษภาคม 2555

ลักษณะของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิสดังกล่าวแสดงได้ดังรูปที่ 2.5 ดังนี้



รูปที่ 2.5 ภาพสามมิติของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

2.6 สรุป

การออกแบบหุ่นยนต์ฝึกซ้อมกีฬาเทนนิสที่นำเสนอในบทนี้ได้ดำเนินการยื่นจดสิทธิบัตรการออกแบบผลิตภัณฑ์ทั้งหมด 5 ผลงานด้วยกัน โดยการออกแบบดังกล่าวเป็นสิ่งที่สำคัญในการสร้างหุ่นยนต์เพื่อใช้งานจริงที่จะดำเนินการสร้างในลำดับถัดไป

มหาวิทยาลัยเทคโนโลยีสุรนารี

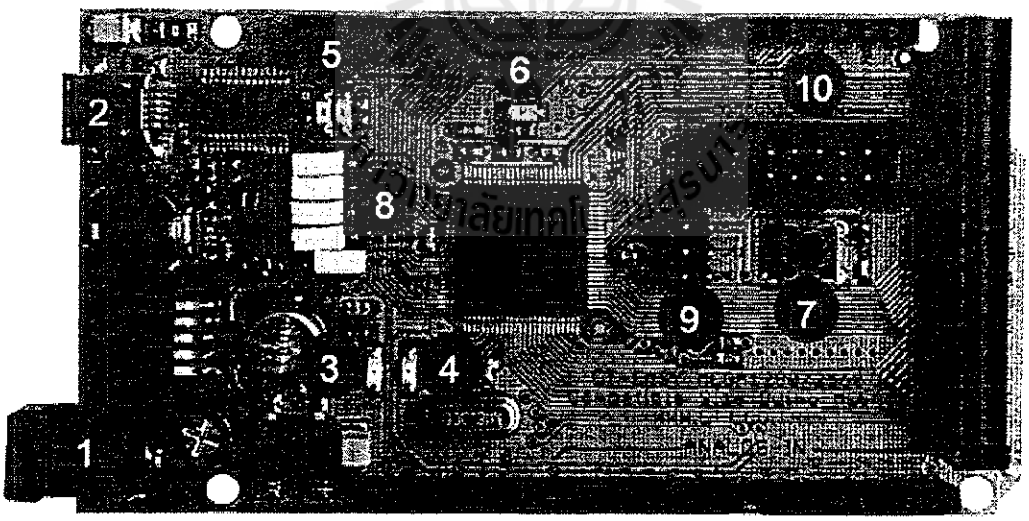
บทที่ 3

บอร์ด ET-EASY MEGA1280 (Duino Mega)

3.1 กล่าวนำ

บอร์ดไมโครคอนโทรลเลอร์รุ่น ET-EASY MEGA1280 ที่จะกล่าวถึงในบทนี้ เป็นบอร์ดที่เหมาะสำหรับหุ่นยนต์อัตโนมัติที่พัฒนามาจาก Arduino Mega โดยได้ออกแบบให้มีการจัดสรรพิน I/O ต่าง ๆ รวมทั้งขนาดให้ตรงตามมาตรฐานของบอร์ด "Arduino Mega" เพียงแต่ได้มีการปรับปรุงข้อจำกัดบางอย่างให้ดีขึ้นกว่า Arduino Mega รุ่นมาตรฐาน เพื่อเพิ่มความสะดวกให้กับผู้ใช้งานมากยิ่งขึ้นโดยปรับปรุงโปรแกรมให้ใช้กับ AVR รุ่นใหญ่ขึ้น เพื่อให้มีจำนวน I/O ทั้ง Digital, Analog, PWM, UART และ ขนาดหน่วยความจำเพิ่มมากขึ้นกว่าเดิม ซึ่งบอร์ดรุ่นนี้มีราคาไม่แพงมากนักและหาซื้อได้ภายในประเทศไทย อีกทั้งในบทนี้ยังกล่าวถึงโครงสร้างของบอร์ด คุณสมบัติต่าง ๆ ของบอร์ด รวมถึงการติดตั้ง Driver ของ USB Bridge ของบอร์ด ET-EASY MEGA1280 การติดตั้งโปรแกรม Arduino และตัวอย่างโค้ดโปรแกรม Arduino ที่สั่งงานบอร์ดอีกด้วย

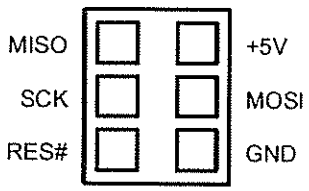
3.2 โครงสร้างบอร์ด ET-EASY MEGA1280 (Duino Mega)



รูปที่ 3.1 โครงสร้างของบอร์ด ET-EASY MEGA1280 (Duino MeGA)

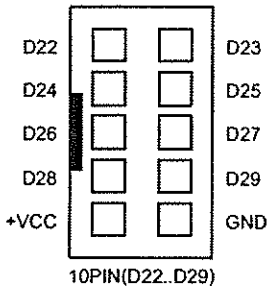
- *หมายเลข 1* คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงจากภายนอก สามารถใช้ได้กับแหล่งจ่ายทั้งแบบ AC และ DC พร้อมวงจรเรียงกระแสแบบบริดจ์ และเรกูเลเตอร์แบบสวิตชิง ช่วยลดความร้อนของไอซี

- *หมายเลข 2* เป็นขั้วต่อ USB สำหรับติดต่อสื่อสารกับคอมพิวเตอร์ PC โดยใช้ FT232RL เป็น USB Bridge ในการเชื่อมต่อระหว่างคอมพิวเตอร์ PC และ MCU ในบอร์ด และยังสามารถใช้ไฟจาก พอร์ต USB เป็นแหล่งจ่ายให้กับบอร์ดได้ด้วย โดยจะมีฟิวส์ขนาด 500 mA สำหรับป้องกันการดึงกระแสเกินจากพอร์ต USB ด้วยและที่พิเศษคือมีวงจรสำหรับตรวจสอบแหล่งจ่ายเพื่อสลับการใช้งานแหล่งจ่ายจาก USB ไปเป็นแหล่งจ่ายจากภายนอก (External Supply) ได้เอง อย่างอัตโนมัติ โดยเมื่อไม่ได้ต่อแหล่งจ่ายภายนอก บอร์ดจะใช้ไฟจากพอร์ต USB เป็นแหล่งจ่ายในการทำงาน แต่เมื่อ มีการต่อแหล่งจ่ายภายนอกวงจรจะสลับไปใช้แหล่งจ่ายจากภายนอก เองโดยอัตโนมัติ
    - o LED +VCC ใช้แสดงสถานะเมื่อมีการจ่ายไฟให้กับบอร์ด
    - o LED VEXT ใช้แสดงสถานะเมื่อมีการจ่ายไฟจากแหล่งจ่ายภายนอก
- *หมายเลข 3* เป็น LED VEXT ใช้แสดงสถานะเมื่อมีการจ่ายไฟเลี้ยงจากแหล่งจ่ายภายนอก
- *หมายเลข 4* เป็น LED +VCC ใช้แสดงสถานะของแหล่งจ่ายไฟเลี้ยง (+VCC) ของบอร์ด โดยเมื่อบอร์ดใช้แหล่งจ่ายจากภายนอกจะแสดงสถานะโดยการให้ LED VEXT และ LED +VCC ติดสว่างพร้อมกันทั้งคู่ แต่ถ้าบอร์ดใช้แหล่งจ่ายจากพอร์ต USB จะแสดงสถานะโดยการให้ LED +VCC ติดสว่างเพียงดวงเดียว
- *หมายเลข 5* เป็น LED แสดงสถานะของ RX และ TX ใช้สำหรับแสดงการรับส่งข้อมูลระหว่างบอร์ด ET-EASY MEGA1280 กับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB
- *หมายเลข 6* เป็น LED D13 ใช้สำหรับทดสอบแสดงการทำงานของ Bootloader และ ใช้ทดสอบการทำงานของบอร์ดจากการควบคุมของ Pin Digital-13 ทำงานด้วย Logic "1" และ หยุดทำงานด้วย Logic "0"
- *หมายเลข 7* เป็นสวิตช์ Reset ใช้สำหรับสั่ง Reset การทำงานของบอร์ด
- *หมายเลข 8* เป็นชุด Jumper สำหรับเลือก การ Program Bootloader ผ่าน USB Port และการใช้งานตามปกติ
- *หมายเลข 9* เป็นขั้วต่อ AVRISP ใช้สำหรับดาวน์โหลดโค้ดให้กับ MCU โดยขั้วต่อ AVRISP นี้จะสามารถใช้งานได้กับเครื่องโปรแกรมทุกรุ่นที่รองรับการใช้งานกับ ATMEGA1280 และใช้ขั้วต่อตรงตามมาตรฐาน AVRISP ดังรูปที่ 3.2



รูปที่ 3.2 ขั้วต่อ AVRISP

หมายเลข 10 เป็นขั้วต่อสัญญาณจาก D[22-29] สำหรับเชื่อมต่อกับบอร์ด I/O ของ อีทีที รวมทั้งจอแสดงผล LCD โดยใช้ร่วมกับ 10PIN LCD หรือ ET-CONV SPI TO LCD ดังรูปที่ 3.3



รูปที่ 3.3 ขั้วต่อสัญญาณจาก D[22-29]

3.3 คุณสมบัติของบอร์ด

- ใช้ ATMEGA1280 เป็น MCU ประจำบอร์ดที่ความถี่ 16MHz จาก Crystal Oscillator
- 128KByte Flash(สงวนไว้ 4KByte สำหรับ Bootloader) / 8KByte SRAM / 4KByte EEPROM
- รองรับการพัฒนาโปรแกรมด้วยภาษา C++ ของ Arduino ตามแบบ Arduino Mega ได้ 100%
- ใช้ USB Bridge ของ FTDI เบอร์ FT232RL พร้อม Over Current Protection สำหรับติดต่อสื่อสารและ Download Code จากคอมพิวเตอร์ให้บอร์ด พร้อม Jumper สำหรับปรับใช้งานบอร์ดเป็นการ Program Bootloader ให้กับ MCU จากพอร์ต USB ในบอร์ดได้เอง โดยไม่ต้องใช้เครื่องโปรแกรม AVRISP จากภายนอก
- 54 พิน Digital I/O โดยมี 14 พิน สามารถโปรแกรมหน้าที่เป็น PWM ได้
- 16 พิน Analog Input (ADC ขนาด 10 บิต 16 ช่อง)
- 4 UART(Hardware Serial Port) แบบ TTL Logic
- ขนาดของ PCB บอร์ด และ ตำแหน่งพิน Connector ต่างๆ ตรงกันกับ Arduino Mega ทั้งหมด ทำให้สามารถนำไปติดตั้งใช้งานร่วมกับบอร์ด Shield แบบต่างๆที่มีการผลิตขึ้นมาใช้งานร่วมกันกับบอร์ด Arduino Mega ได้ทั้งหมดโดยบอร์ดมีขนาด PCB Size 5.3 cm x 10.2 cm
- ที่ขั้ว Header 10 พิน IDE ของ 8 บิต Digital I/O(D22-D29) สำหรับเชื่อมต่อกับ LCD หรือบอร์ด I/O แบบต่างๆ ของ อีทีที เพิ่มความสะดวกในการใช้งาน

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี

- รองรับการใช้งานกับแหล่งจ่ายจากภายนอกทั้งแบบ AC และ DC ขนาด 7-20V โดยเลือกใช้เรกูเลเตอร์แบบสวิตชิงขนาด 1 A (LM2575-5V) ลดปัญหาเรื่องความร้อนเมื่อมีการใช้กระแสสูงๆ สามารถใช้แหล่งจ่ายจากพอร์ต USB ได้ในกรณีใช้กระแสไม่เกิน 500 mA โดยมีวงจรเลือกแหล่งจ่ายอัตโนมัติโดยจะตัดการใช้ไฟเลี้ยงจาก USB โดยอัตโนมัติ เมื่อมีการต่อแหล่งจ่ายจากภายนอกให้บอร์ด

3.3.1 คุณสมบัติของสัญญาณต่างๆของบอร์ด ET-EASY MEGA1280

- RESET# เป็นสัญญาณ Input Logic Reset ของ MCU เมื่อเป็น Logic Low จะทำให้ MCU อยู่ในสภาวะรีเซ็ต เมื่อเป็น Logic High จะทำให้ MCU อยู่ในสภาวะทำงานตามปกติ โดยสัญญาณRESET# นี้จะถูกควบคุมจาก 2 แหล่ง คือ จาก สวิตช์ RESET ภายในบอร์ด และ จากสัญญาณ DTR ของ FT232RL ถ้ามีการเลือก Enable Jumper ของ Auto Reset จาก DTR ไว้
- +3.3 V เป็นแหล่งจ่ายไฟขนาด +3.3 V ที่ได้จากวงจรเรกูเลเตอร์ ภายในของ FT232RL สามารถจ่ายกระแสได้สูงสุด 50 mA ซึ่งเมื่อต้องการนำแหล่งจ่าย +3.3V นี้ไปใช้งานเป็นแหล่งจ่ายให้กับอุปกรณ์ต่าง ๆ ต้องระมัดระวังเรื่องการดึงกระแสของโหลดด้วย ถ้าโหลดมีการดึงกระแสมากกว่า 50 mAอาจทำให้ FT232RL เกิดความเสียหายได้
- +VIN เป็นไฟ DC ที่รับมาจาก Jack VIN(External Supply) แต่ผ่านการวงจรเรียงกระแส และวงจรกรอง (Filter) เป็น DC แล้ว มีขนาดแรงดันเฉลี่ยตามขนาดแรงดันที่ป้อนให้กับบอร์ดทาง Jack VIN
- +5 V เป็นจุดต่อแหล่งจ่ายไฟของบอร์ดออกไปใช้งาน ซึ่งมาจากแหล่งกำเนิด 2 แหล่ง คือ จากพอร์ต USB และจากแหล่งจ่ายภายนอกซึ่งถ้าต่อแหล่งจ่ายให้บอร์ดจากแหล่งจ่ายภายนอก ผ่านทาง Jack VIN แหล่งจ่าย +5 V นี้จะมาจาก Switching Regulate (LM2575-5V) สามารถจ่ายกระแสได้สูงสุดถึง 1 A แต่ถ้าใช้แหล่งจ่ายจากพอร์ต USB แหล่งจ่าย +5 V นี้จะมาจากพอร์ต USB โดยตรงโดยจะมีฟิวส์ขนาด 500 mA ต่อป้องกันการดึงกระแสเกินเพื่อป้องกันความเสียหายของพอร์ต USB โดยจะจ่ายกระแสได้สูงสุดไม่เกิน 500 mA ขึ้นอยู่กับคุณสมบัติในการจ่ายกระแสของพอร์ต USB และการกำหนดค่าให้กับ FT232RL ด้วย
- A0-A15 เป็นขาสัญญาณ Analog Input แบบ ADC มีขนาดความละเอียด 10 บิต มี 16 พิน สามารถรับแรงดัน Analog Input ได้ 0-5 VDC

- D0-D53 เป็นขาสัญญาณ Digital Input/Output แบบ TTL มีทั้งหมด 54 พิน สามารถใช้ทำหน้าที่เป็น Input หรือ Output ตามการกำหนดจากโปรแกรม โดยมีบางพิน สามารถกำหนดหน้าที่ใช้งานเป็นฟังก์ชั่นพิเศษต่าง ๆ เพิ่มเติมได้อีก
    - o D0-D1 ใช้ทำหน้าที่เป็นพอร์ตสื่อสารอนุกรม RS232 (UART0) โดยได้ทำการเชื่อมต่อกับ USB Bridge ของ FT232RL เพื่อใช้ Upload Code ให้กับบอร์ด และยังสามารถใช้ทดลองติดต่อสื่อสารรับส่งข้อมูลระหว่างบอร์ดกับคอมพิวเตอร์ PC
    - o D2-D13 ทำหน้าที่เป็น PWM ขนาด 8 บิต มี 14 พิน
    - o D14 ทำหน้าที่เป็น TX3 สำหรับ ส่งข้อมูลของ UART3
    - o D15 ทำหน้าที่เป็น RX3 สำหรับ รับข้อมูลให้กับ UART3
    - o D16 ทำหน้าที่เป็น TX2 สำหรับ ส่งข้อมูลของ UART2
    - o D17 ทำหน้าที่เป็น RX2 สำหรับ รับข้อมูลให้กับ UART2
    - o D18 ทำหน้าที่เป็น TX1 สำหรับ ส่งข้อมูลของ UART1
    - o D19 ทำหน้าที่เป็น RX1 สำหรับ รับข้อมูลให้กับ UART1
    - o D20 ทำหน้าที่เป็น SDA ของ I2C Bus สำหรับใช้สื่อสารกับ I2C
    - o D21 ทำหน้าที่เป็น SCL ของ I2C Bus สำหรับใช้สื่อสารกับ I2C

AREF เป็นสัญญาณ Analog Reference จากภายนอกที่ต้องการป้อนให้กับ MCU ซึ่งตามปกติแล้ว ATMEGA1280 สามารถโปรแกรมให้เลือกใช้แรงดันอ้างอิงจากภายในได้อยู่แล้วโดยสามารถเลือกเป็น 1.1 V หรือ 2.56 V หรือ AVCC (+5 V) โดยไม่จำเป็นต้องป้อนแรงดันอ้างอิงจากภายนอกให้กับบอร์ดอีก แต่ถ้าต้องการแรงดันอ้างอิงที่มีความแตกต่างจากที่กล่าวมาแล้วก็สามารถป้อนเป็นแรงดันอ้างอิงจากภายนอกผ่านทางขา AREF นี้เข้าไปเองได้ระหว่าง 0-5 V

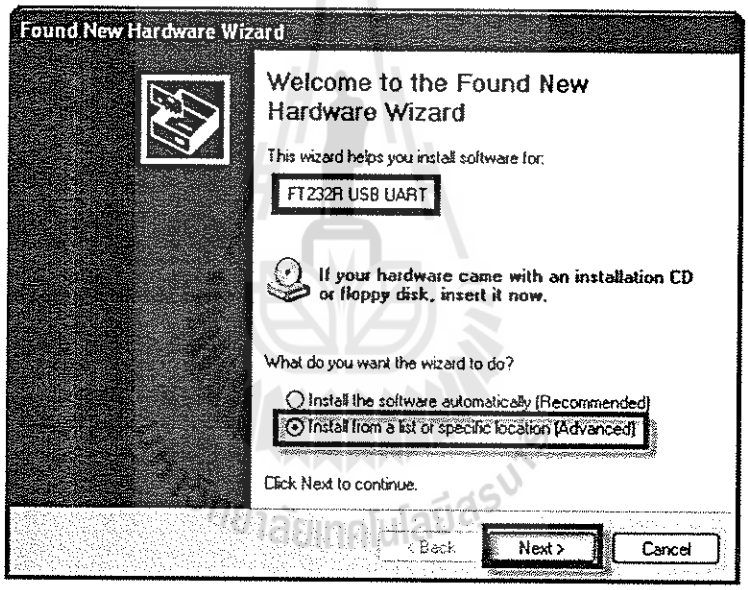
3.4 การติดตั้ง Driver ของ USB Bridge ของบอร์ด ET-EASY MEGA1280 (Duino Mega)

บอร์ด ET-EASY MEGA1280 ใช้ USB Bridge ของ FTDI เป็นตัวกลางในการเชื่อมต่อกับคอมพิวเตอร์ PC โดย USB Bridge ของ FTDI ทำหน้าที่เป็นตัวกลางในการเชื่อมต่อและติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ PC กับ MCU ATmega1280 ของบอร์ด ET-EASY MEGA1280 ในรูปแบบของพอร์ตอนุกรม (Visual Com Port) โดยโปรแกรมประยุกต์ ต่าง ๆ ที่ทำงานอยู่บนคอมพิวเตอร์ PC รวมทั้งโปรแกรม Arduino จะมองเห็น พอร์ต USB ที่เชื่อมต่อกับบอร์ด ET-EASY MEGA1280 เป็นพอร์ตสื่อสารอนุกรม (Com Port) ช่องหนึ่งเท่านั้น ซึ่งถ้า

เครื่องคอมพิวเตอร์ของผู้ใช้เคยทำการติดตั้ง Driver สำหรับ USB Bridge ของ FTDI ไว้ก่อนแล้ว เมื่อทำการเชื่อมต่อสาย USB ของบอร์ด ET-EASY MEGA1280 เข้ากับUSB HUB ของเครื่องคอมพิวเตอร์ PC แล้ว Windows จะทำการติดตั้ง Driver ให้เองโดยอัตโนมัติ แต่ถ้าเครื่องคอมพิวเตอร์ PC ยังไม่เคยติดตั้ง Driver ของ FTDI ไว้ก่อนก็จะต้องทำการติดตั้ง Driver ให้กับบอร์ดให้เรียบร้อยเสียก่อนซึ่งมีลำดับขั้นตอนดังนี้

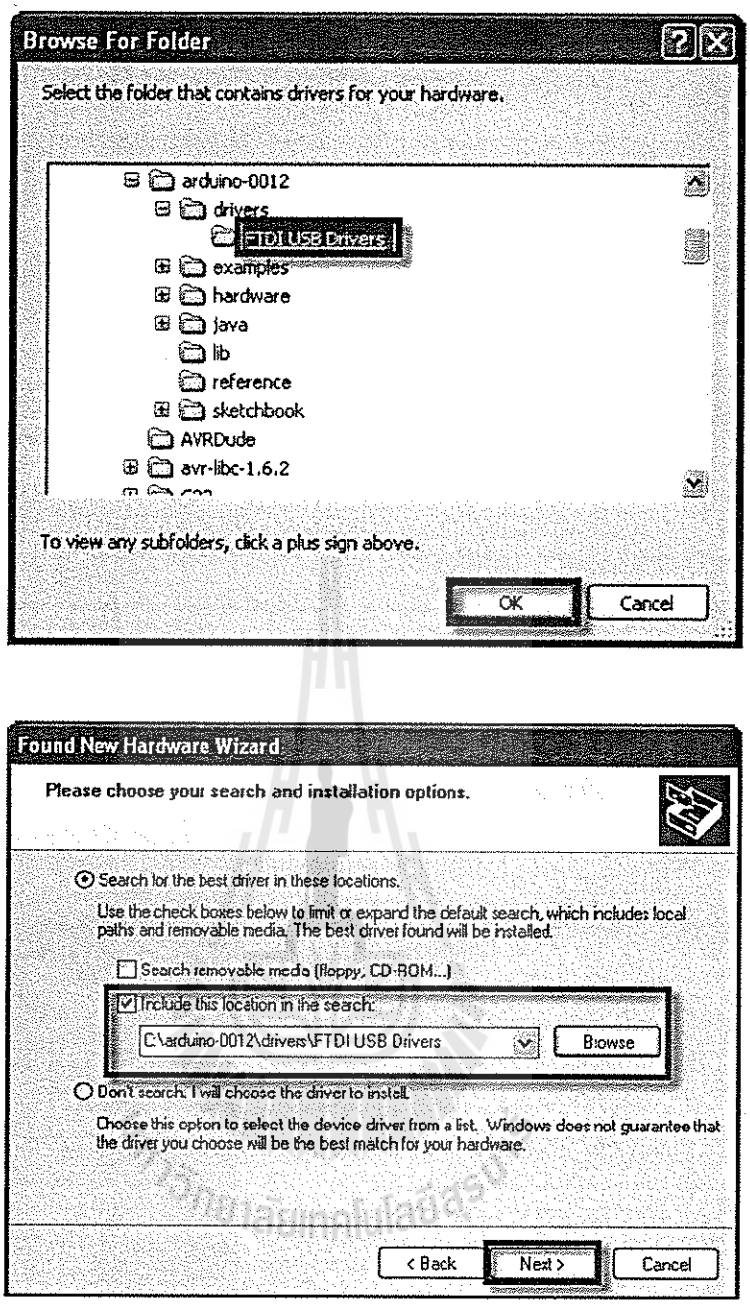
ขั้นที่ 1 เตรียมแผ่น CD ROM ที่บรรจุ Driver ของ FTDI ไว้ให้พร้อม หรือ ในกรณีที่ผู้ใช้ได้ทำการติดตั้งโปรแกรมของ Arduino ไว้เรียบร้อยแล้ว ภายในโฟล์เดอร์ของโปรแกรม Arduino ก็จะมี Driver ของ FTDI จัดเตรียมไว้ให้เรียบร้อบแล้ว โดยจะอยู่ที่ "C:\arduino-0012\drivers\FTDI USB Drivers\"

ขั้นที่ 2 ทำการเสียบสาย USB ของบอร์ด ET-EASY MEGA1280 เข้ากับพอร์ต USB HUB ของเครื่องคอมพิวเตอร์ PC ซึ่ง Windows จะตรวจพบอุปกรณ์ใหม่ โดยเป็น "FT232R USB UART" และแจ้งให้ผู้ใช้ทำการติดตั้ง Driver ให้กับอุปกรณ์ดังรูปที่ 3.4



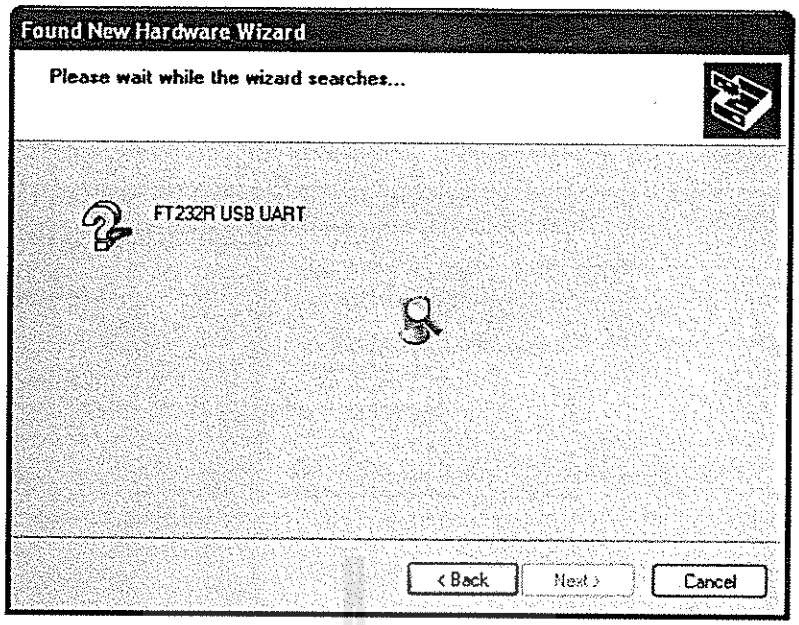
รูปที่ 3.4 ขั้นตอนการติดตั้ง Driver ของ USB Bridge ขั้นที่ 2

ขั้นที่ 3 ให้เลือก Install from list or specific location(Advanced) แล้วเลือก Nextซึ่ง Windows ก็จะแจ้งให้ผู้ใช้ระบุตำแหน่งโฟลเดอร์ที่บรรจุไฟล์ Driver ของ FTDI ไว้ ก็ให้เลือกที่ Browse และเลือกไปยังDrive และ โฟลเดอร์ที่เก็บไฟล์ Driver ไว้ ซึ่งถ้าผู้ใช้ได้ทำการติดตั้งโปรแกรมของ Arduino ไว้แล้ว ก็ให้เลือกไปที่ "C:\arduino-0012\drivers\FTDI USB Drivers" แล้วเลือก Next ดังรูปที่ 3.5



รูปที่ 3.5 ขั้นตอนการติดตั้ง Driver ของ USB Bridge ขั้นที่ 3

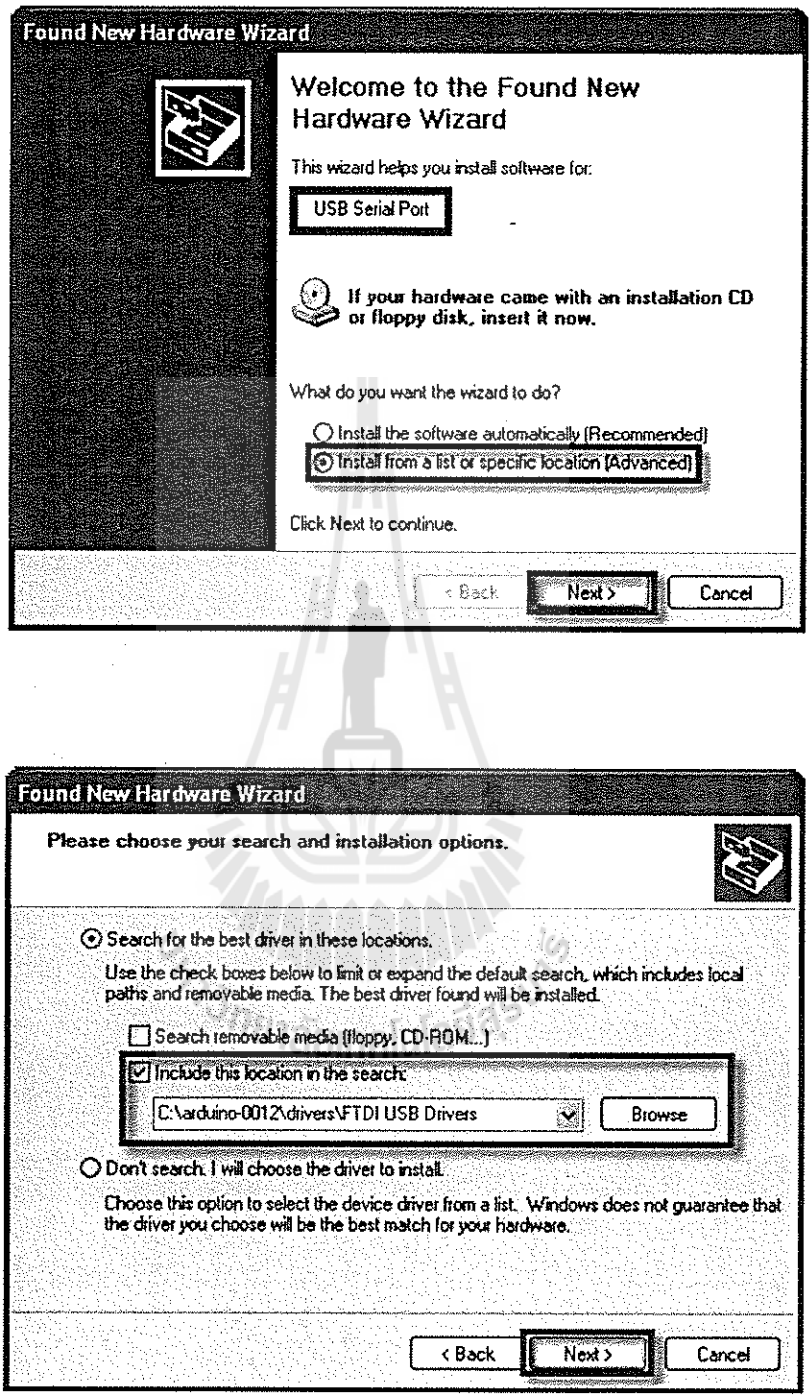
ขั้นที่ 4 ในขั้นตอนนี้โปรแกรม Windows จะทำการค้นหาและติดตั้ง Driver ให้กับอุปกรณ์ ให้รอสักครู่จนการทำงานเสร็จเรียบร้อย แล้วเลือก Finish ดังรูปที่ 3.6



รูปที่ 3.6 ขั้นตอนการติดตั้ง Driver ของ USB Bridge ขั้นที่ 4

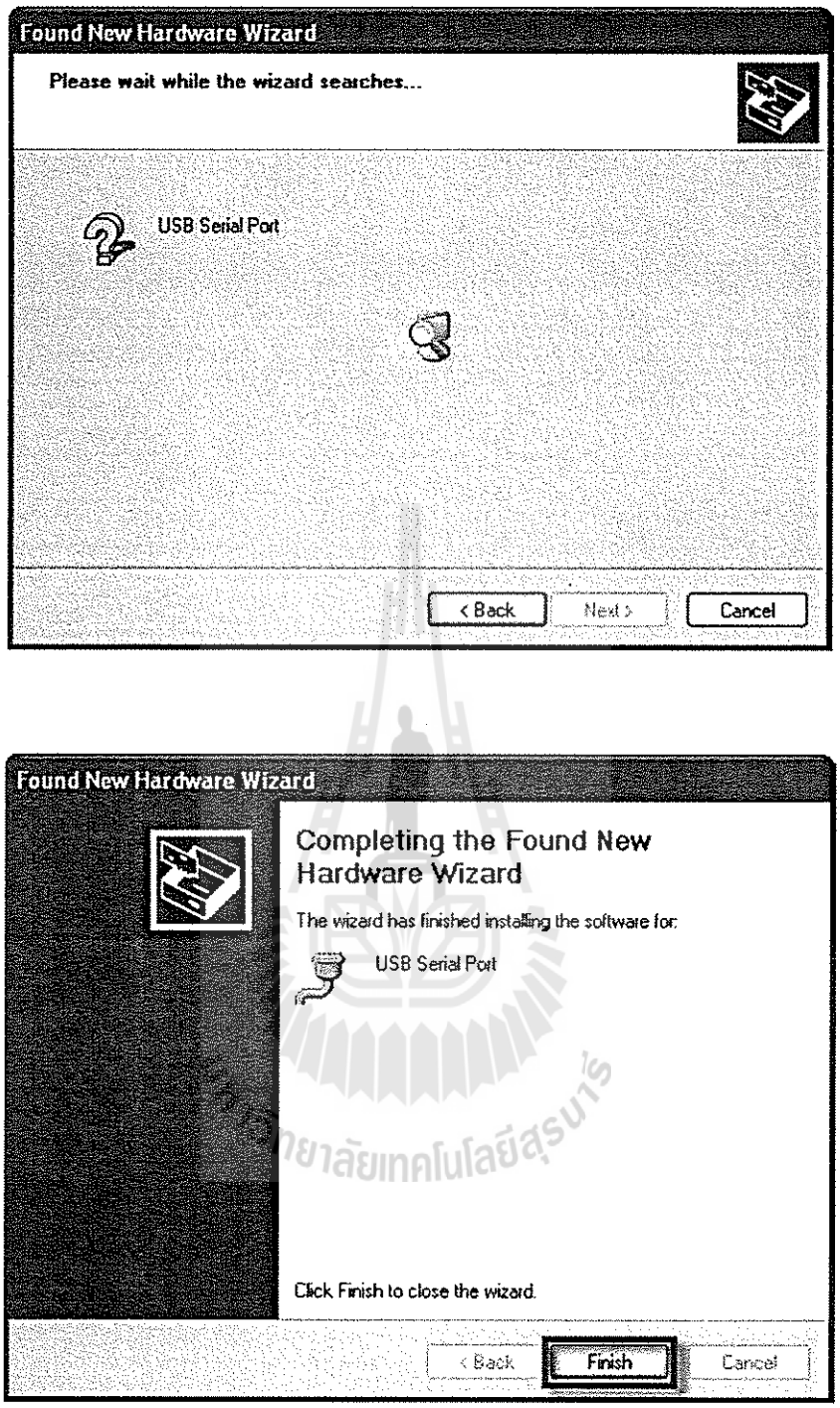
5 หลังจากทำการติดตั้ง Driver ของฮาร์ดแวร์เรียบร้อยแล้ว Windows ก็จะตรวจพบว่ามีอุปกรณ์ใหม่ถูกเชื่อมต่ออยู่ โดยเป็นอุปกรณ์ประเภท "USB Serial Port" และแจ้งให้ผู้ใช้ทำการติดตั้ง Driver ให้กับอุปกรณ์ใหม่ที่ระบุเป็น "USB Serial Port" อีกครั้งหนึ่ง

ซึ่งก็ให้เลือกระบุตำแหน่งโฟลเดอร์ที่เก็บไฟล์ Driver ไว้ โดยให้เลือกเหมือนขั้นตอนในขั้นที่ 3 ดังรูปที่ 3.7



รูปที่ 3.7 ขั้นตอนการติดตั้ง Driver ของ USB Bridge ขั้นที่ 5

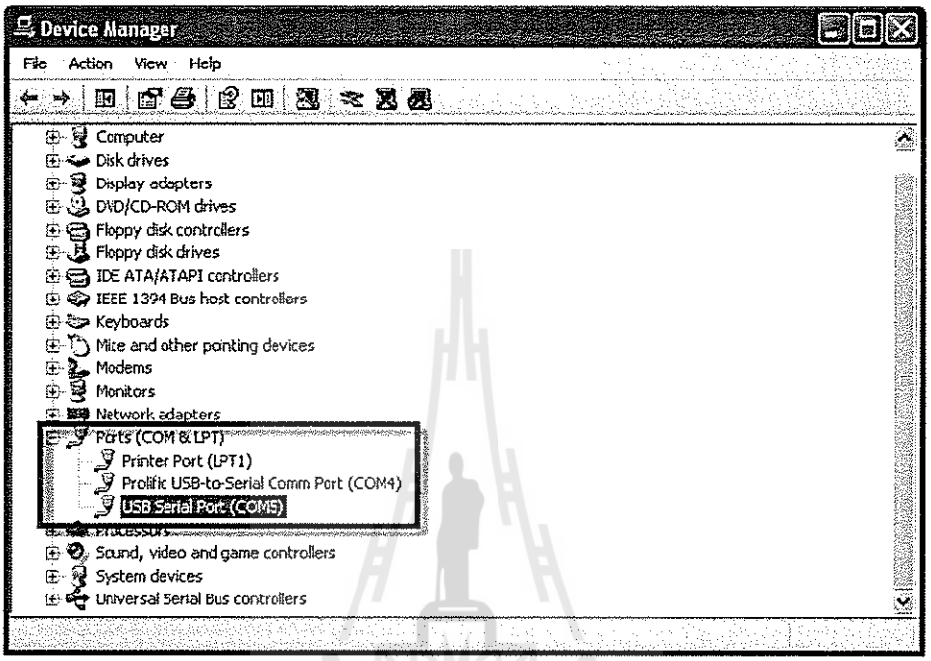
ขั้นที่ 6 ในขั้นตอนนี้โปรแกรม Windows จะทำการค้นหาและติดตั้ง Driver ให้กับอุปกรณ์ ให้รอสักครู่จนการทำงานเสร็จเรียบร้อย แล้วเลือก Finish ดังรูปที่ 3.8



รูปที่ 3.8 ขั้นตอนการติดตั้ง Driver ของ USB Bridge ขั้นที่ 6

ขั้นที่ 7 หลังจากทำการติดตั้ง Driver เรียบร้อยแล้ว ก็สามารถใช้งานอุปกรณ์ดังกล่าวได้แล้ว แต่เพื่อความถูกต้องในครั้งแรกนี้ควรต้องเข้าไปทำการตรวจสอบและปรับแต่งค่าให้กับ

อุปกรณ์ก่อน โดยในขั้นตอนนี้ให้ไปที่ "My Computer → Control Panel → System → Hardware → Device Manager" แล้วทำการตรวจสอบที่ Ports (COM&LPT) แล้วดูที่ชื่อของ "USB Serial Port" ซึ่งให้ผู้ใช้จดจำหมายเลขของ Com Port ของอุปกรณ์ดังกล่าวไว้เพื่อใช้อ้างอิงถึงในการเรียกใช้งาน ดังรูปที่ 3.9

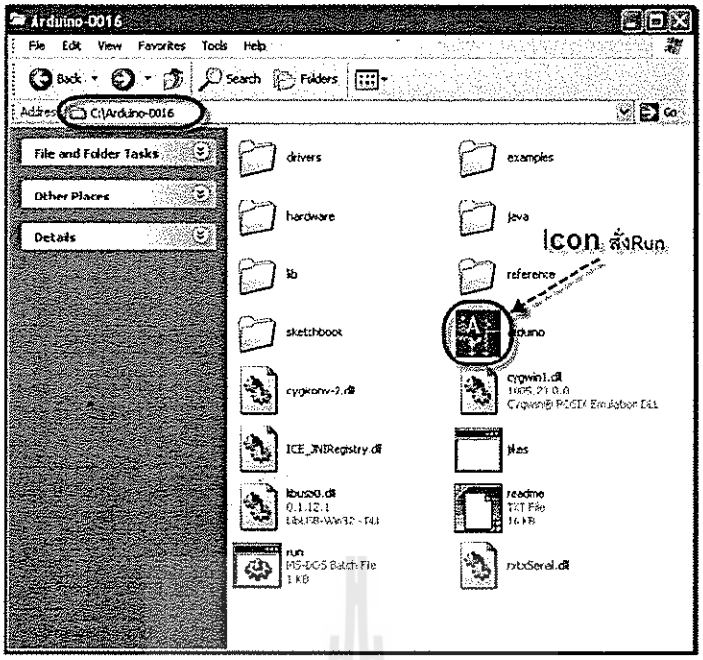


รูปที่ 3.9 ขั้นตอนการติดตั้ง Driver ของ USB Bridge ขั้นที่ 7

การติดตั้งโปรแกรม Arduino

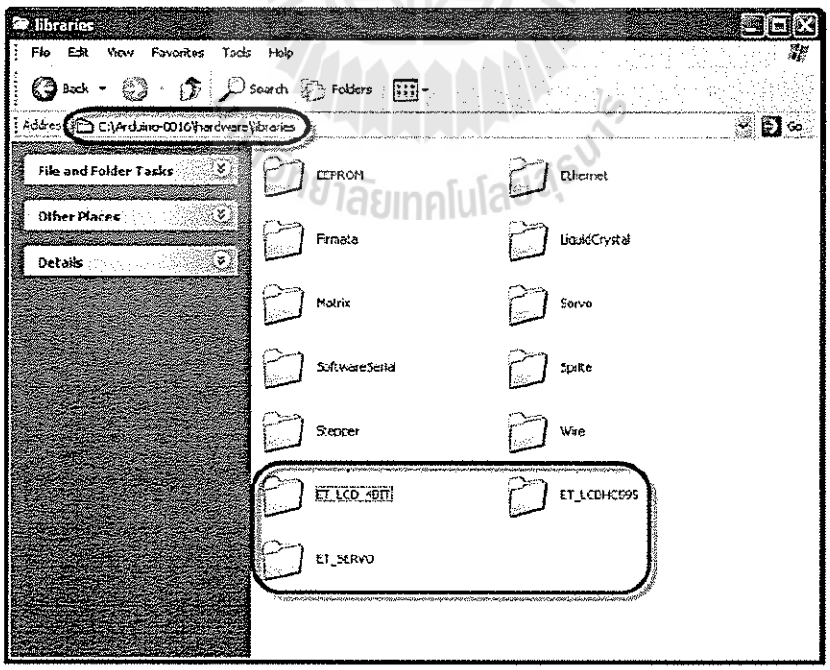
โปรแกรมของ Arduino ได้รับการปรับปรุงเป็นรุ่น เวอร์ชั่น "Arduino-0016" แล้ว โดยมีโปรแกรมให้เลือกใช้งาน 4 Platform ทั้ง Windows,Mac OSx และ Linux โดยผู้อ่านสามารถเข้าไปตรวจสอบหรือ Download โปรแกรมรุ่นใหม่ของ Arduino มาใช้งานได้ฟรีโดยไม่เสียค่าใช้จ่ายใดๆจาก "http://arduino.cc/" หรือ "http://arduino.cc/en/Main/Software" ซึ่งเป็นเวปไซต์ที่ได้รวบรวมรายละเอียดและข่าวคราวความเคลื่อนไหวต่างๆ เกี่ยวกับ Arduino มากมาย ซึ่งข้อมูลต่าง ๆ จะได้รับการปรับปรุงอย่างต่อเนื่องเป็นประจำ ซึ่งการติดตั้งโปรแกรม Arduino มีลำดับขั้นตอนการติดตั้งดังต่อไปนี้

ขั้นที่ 1 โดยในการติดตั้งโปรแกรมของ Arduino นั้นให้ทำการ Unzip แล้ว Copy ไปติดตั้งไว้ในตำแหน่งโฟลเดอร์ "c:\arduino-0016" ดังรูปที่ 3.10



รูปที่ 3.10 ขั้นตอนการติดตั้งโปรแกรม Arduino ขั้นที่ 1

ขั้นที่ 2 ให้ทำการ Copy โฟลเดอร์ Library ต่างๆ ของ อีทีที ในแผ่น CD-ROM เพิ่มเข้าไปในโฟลเดอร์ที่ได้ทำการติดตั้งโปรแกรมArduinoไว้คือ "c:\Arduino-0016\hard ware\Library" ดังรูปที่ 3.11

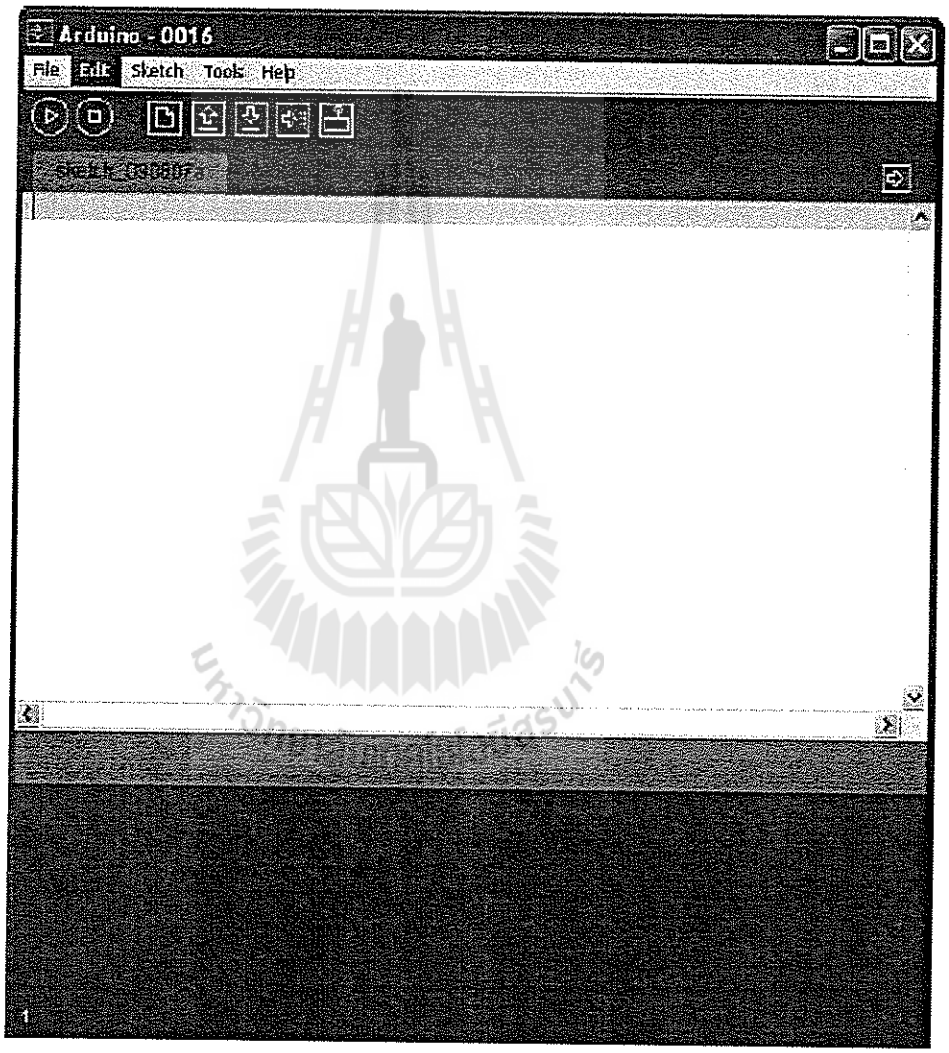


รูปที่ 3.11 ขั้นตอนการติดตั้งโปรแกรม Arduino ขั้นที่ 2

3.5.1 ทดสอบเขียนโปรแกรมใช้งานด้วย Arduino

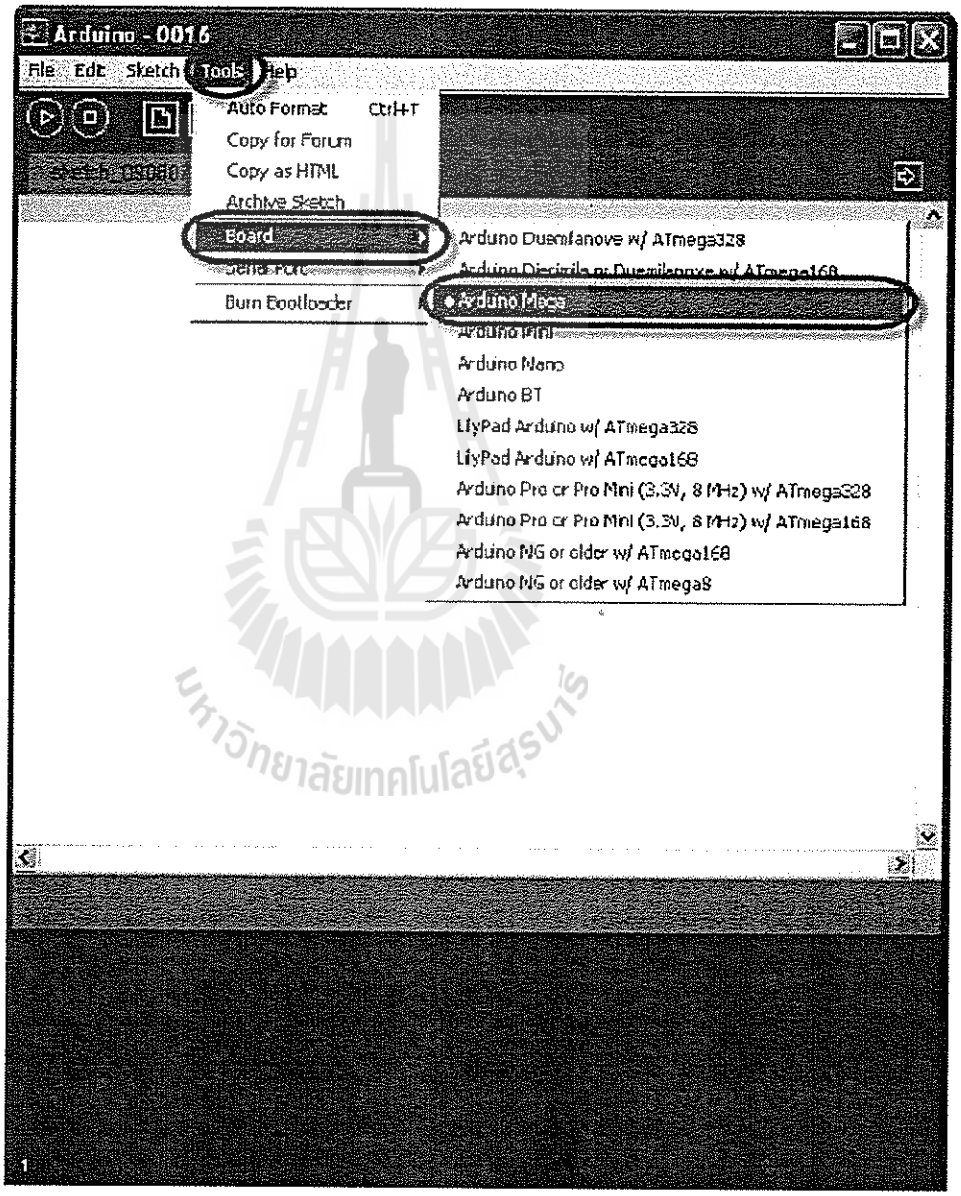
หลังจากที่ได้ทำการติดตั้งโปรแกรม Arduino เป็นที่เรียบร้อยแล้ว ลำดับขั้นตอนต่อจากนี้เป็นต้นไป เป็นเรื่องของการใช้งาน การเขียนโปรแกรม และการศึกษาเรียนรู้ในเรื่องต่าง ๆ แต่ก่อนอื่นจะต้องทำการติดตั้งโปรแกรมของ Arduino เพื่อใช้เป็นโปรแกรมสำหรับศึกษาเรียนรู้ ซึ่งมีลำดับขั้นตอนการติดตั้งดังต่อไปนี้

ขั้นที่ 1 ทำการสั่ง Run โปแกรม "arduino.exe" จะได้ผลดังรูปที่ 3.12



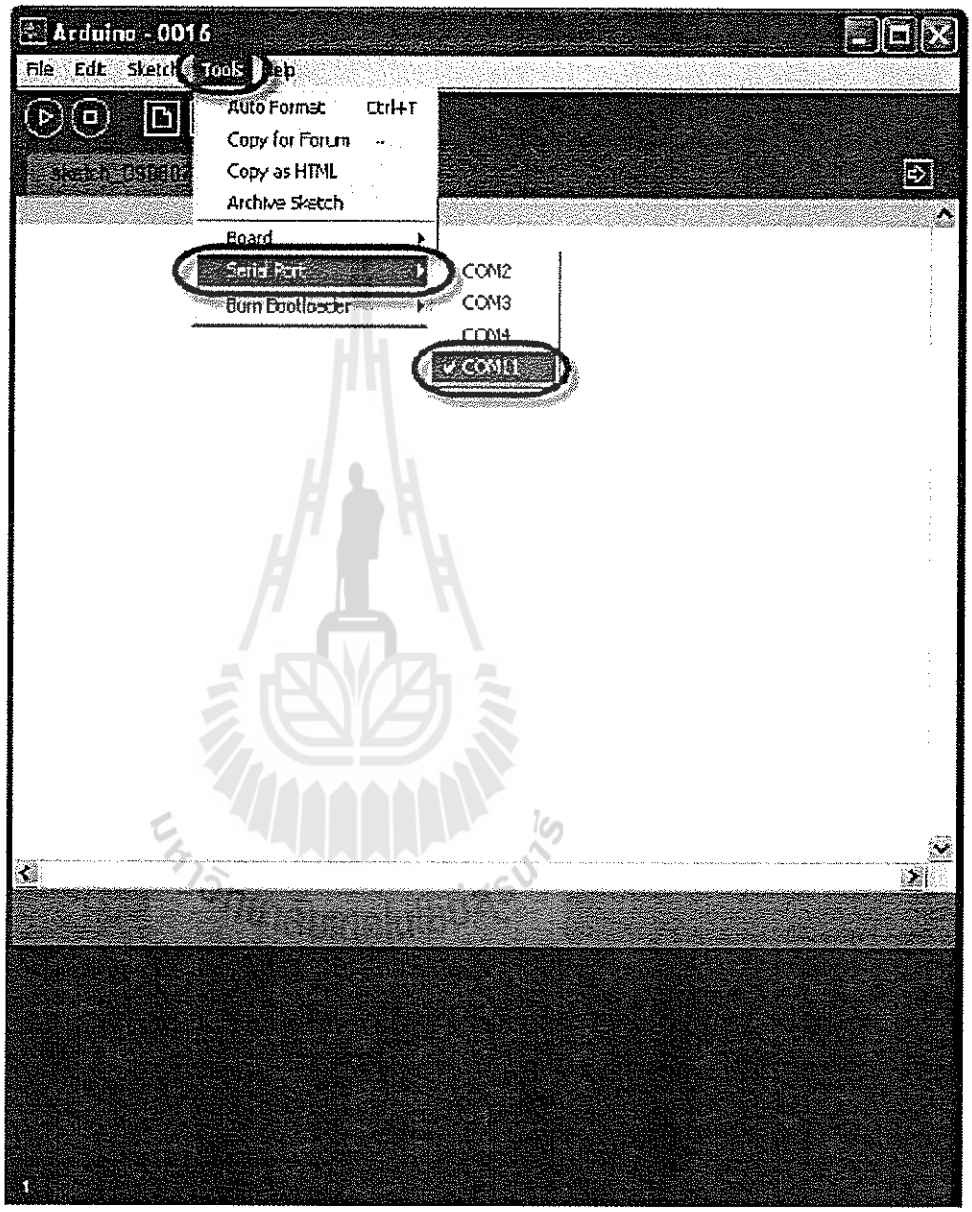
รูปที่ 3.12 การทดสอบเขียนโปรแกรมใช้งานด้วย Arduino ขั้นที่ 1

ขั้นที่ 2 ในครั้งแรกของการเรียกใช้งานโปรแกรม ให้ทำการกำหนดระบบฮาร์ดแวร์ที่จะใช้งานกับโปรแกรมของ Arduino ให้เรียบร้อยเสียก่อน เนื่องจากในปัจจุบันนี้ มีการออกแบบวงจรและสร้างฮาร์ดแวร์บอร์ดแบบต่างๆสำหรับนำมาใช้งานร่วมกับโปรแกรมพัฒนาของ Arduino ไว้มากมายหลายรุ่น โดยในกรณีของบอร์ด ET-EASY MEGA1280 ให้ทำการเลือกกำหนดชื่อบอร์ดเป็น "Arduino Mega" โดยคลิกเมาส์ที่ "Tools → Board → "Arduino Mega" ดังรูปที่ 3.13



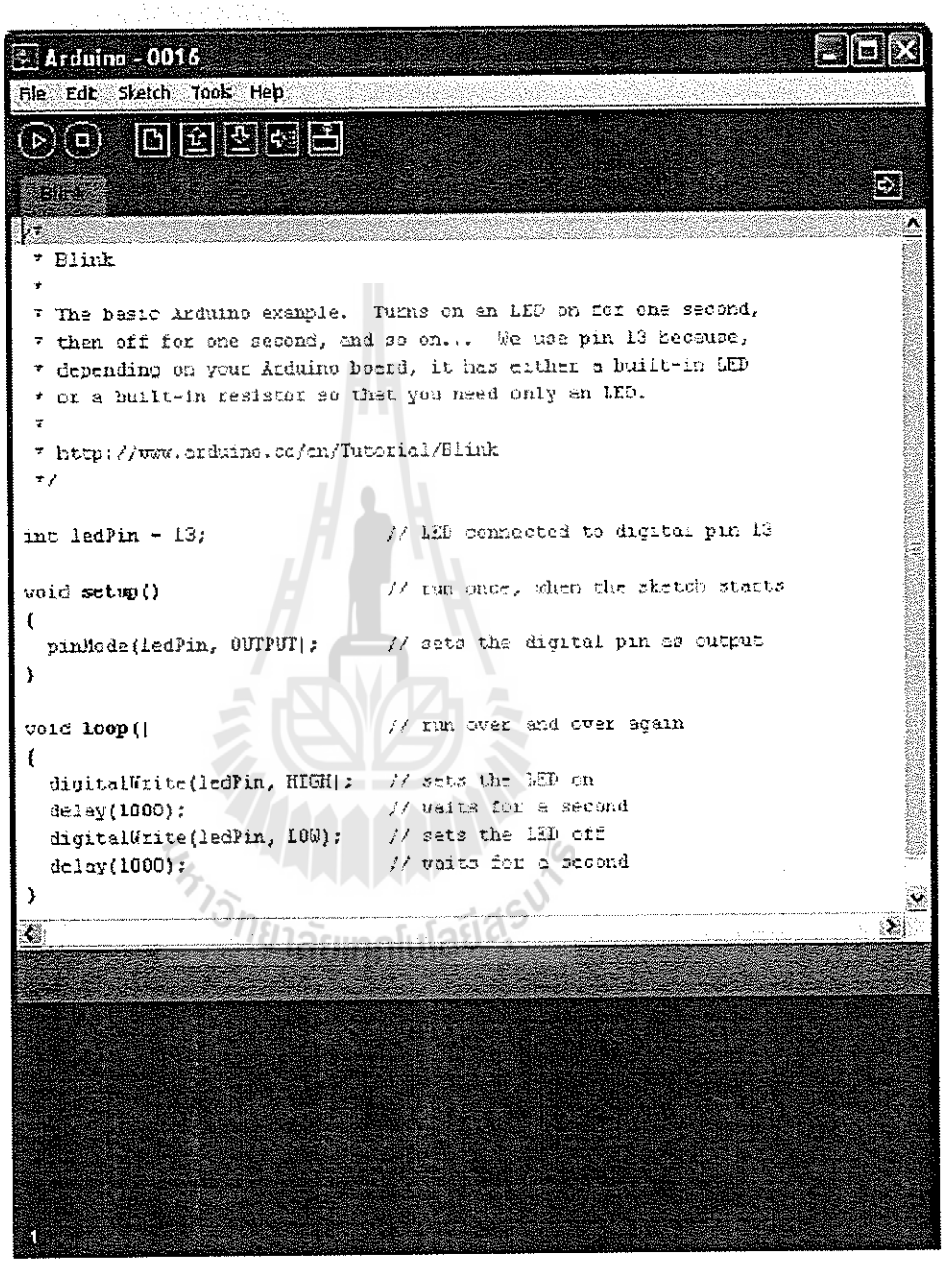
รูปที่ 3.13 การทดสอบเขียนโปรแกรมใช้งานด้วย Arduino ขั้นที่ 2

ขั้นที่ 3 เลือกกำหนดหมายเลขพอร์ต สำหรับติดต่อสื่อสารกับบอร์ด ให้ตรงกับหมายเลข Comport ที่ต่อใช้งานไว้จริงในเครื่องคอมพิวเตอร์ PC เช่น ถ้าหมายเลข Comport ของเครื่องคอมพิวเตอร์ PC เป็น COM11 ให้คลิกเมาส์ที่ Tools → Serial Port → COM11 ดังรูปที่ 3.14



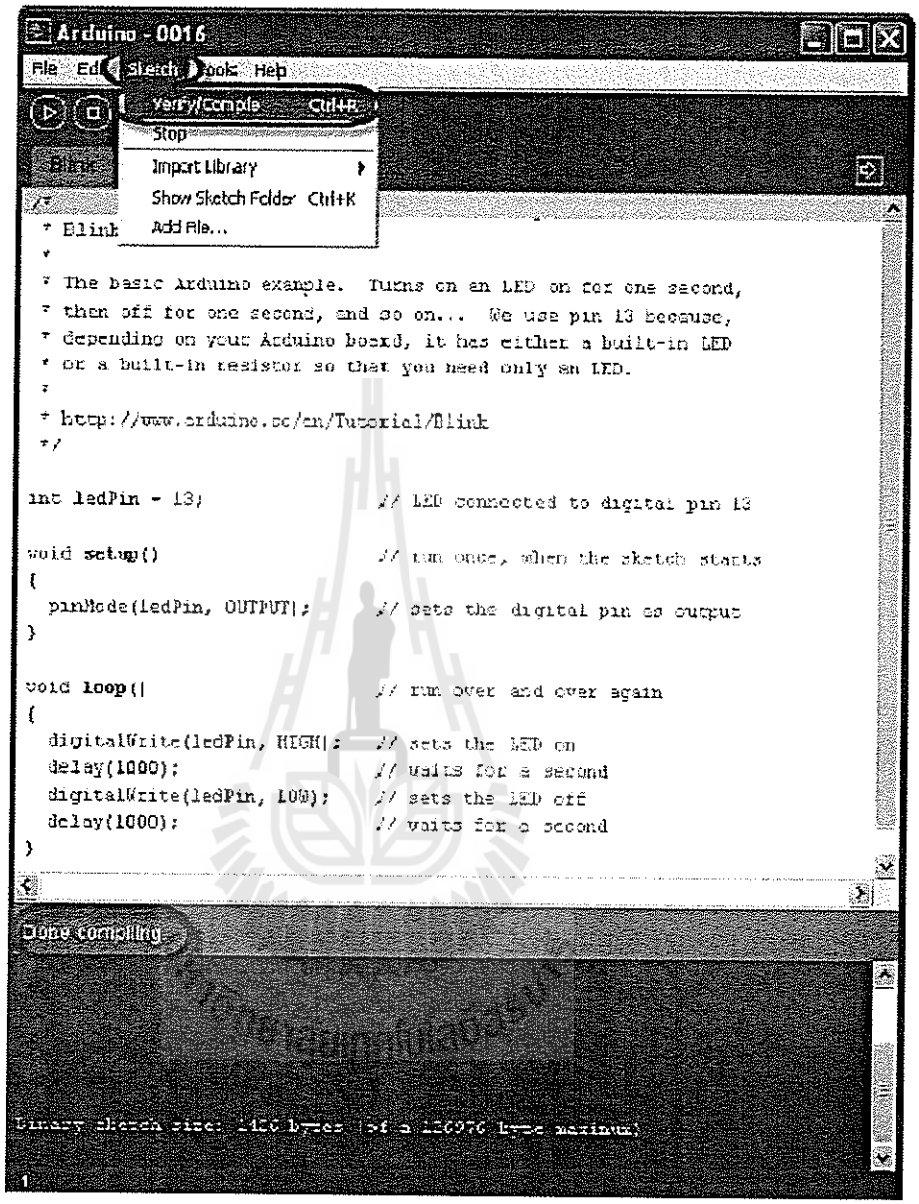
รูปที่ 3.14 การทดสอบเขียนโปรแกรมใช้งานด้วย Arduino ขั้นที่ 3

ขั้นที่ 4 ทดสอบเขียนโปรแกรม โดยคลิกเมาส์ที่ File → New แล้วพิมพ์โปรแกรมทดสอบ หรืออาจใช้การสั่งเปิดไฟล์ตัวอย่างที่สร้างไว้แล้วขึ้นมาแทนก็ได้ โดยในที่นี้ขอแนะนำให้ทดสอบด้วยโปรแกรมไฟกระพริบ โดยให้เลือก "File → sketchbook → Examples → Digital → Blink" ซึ่งจะได้ดังรูปที่ 3.15



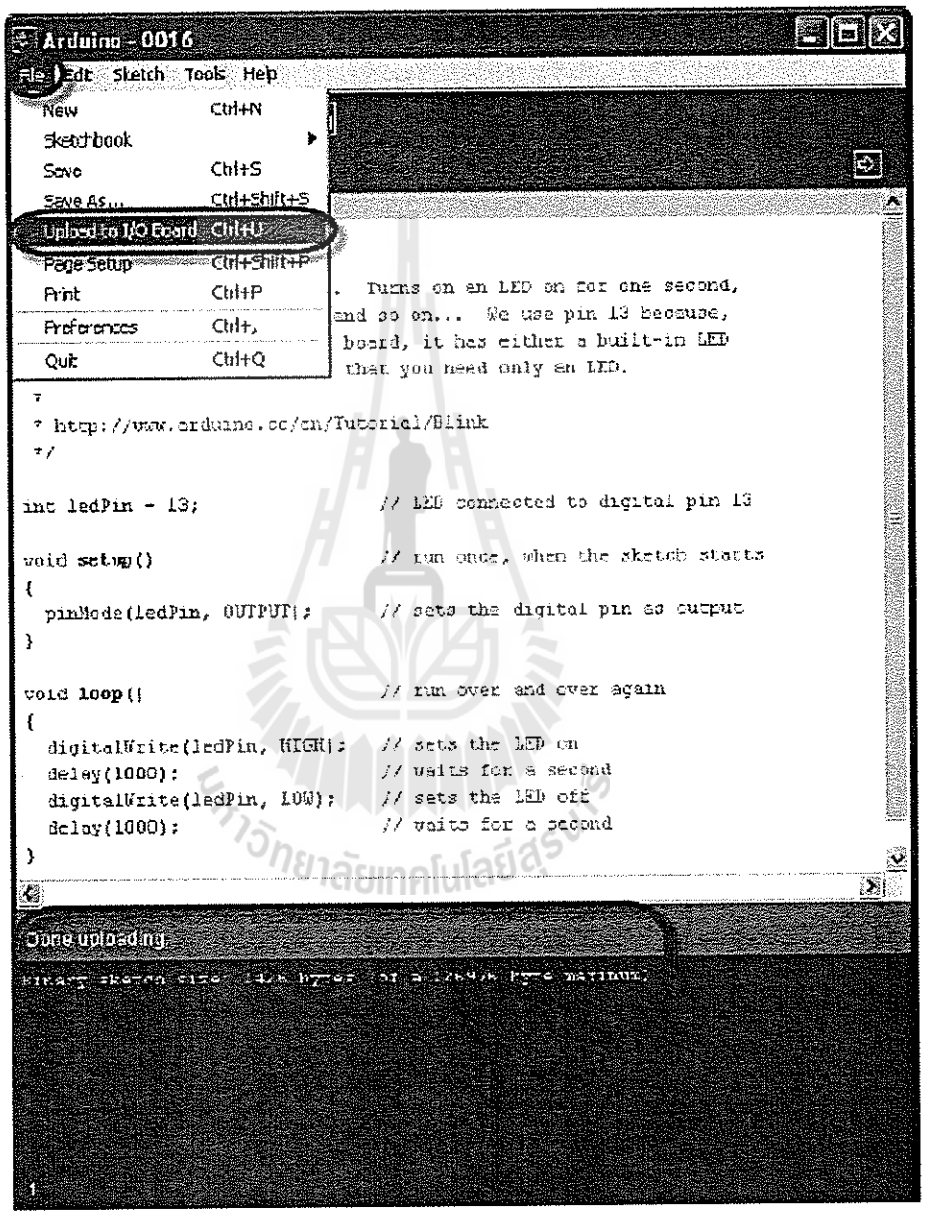
รูปที่ 3.15 การทดสอบเขียนโปรแกรมใช้งานด้วย Arduino ขั้นที่ 4

ขั้นที่ 5 สั่งแปลโปรแกรมโดยคลิกเมาส์ที่ "Sketch → Verify/Compile" เพื่อตรวจสอบคำสั่งต่างๆในโปรแกรมว่าถูกต้องหรือไม่ ดังตัวอย่างรูปที่ 3.16



รูปที่ 3.16 การทดสอบเขียนโปรแกรมใช้งานด้วย Arduino ขั้นที่ 5

ขั้นที่ 6 สั่ง Download Code ให้กับบอร์ด โดยคลิกเมาส์เลือกที่ "File → Upload to I/O Board" แล้วรอสักครู่จนโปรแกรมทำงานเสร็จ หลังจากที่ทำการ Upload Code ให้กับบอร์ดเป็นที่เรียบร้อยแล้วบอร์ดก็จะเริ่มต้นทำงานตามคำสั่งที่เขียนไว้ในโปรแกรมทันที โดยจะสังเกตเห็น LED กระพริบ ติดและ ดับ สลับกันไปมา ด้วยความเร็วประมาณ 1 วินาที ตลอดเวลา ซึ่งควรได้ผลดังรูปที่ 3.17



รูปที่ 3.17 การทดสอบเขียนโปรแกรมใช้งานด้วย Arduino ขั้นที่ 6

3.6 สรุป

บอร์ด ET-EASY MEGA1280 (Duino Mega) เป็นบอร์ดที่เหมาะสำหรับหุ่นยนต์อัตโนมัติอย่างยิ่งเพราะทั้งคุณสมบัติของบอร์ดที่สามารถรองรับกับ AVR และ มีจำนวน I/O ทั้ง Digital, Analog, PWM, UART และ ขนาดหน่วยความจำมากเพียงพอ อีกทั้งยังสามารถรองรับการเขียนโปรแกรมด้วยภาษา C++ ของ Arduino ซึ่งทำให้ง่ายต่อการเขียนโปรแกรมคำสั่งต่างๆ ซึ่งถ้าผู้อ่านได้ศึกษาอย่างละเอียดก็จะสามารถประยุกต์เพื่อใช้ประโยชน์จากบอร์ดนี้ได้มากขึ้นอีกด้วย

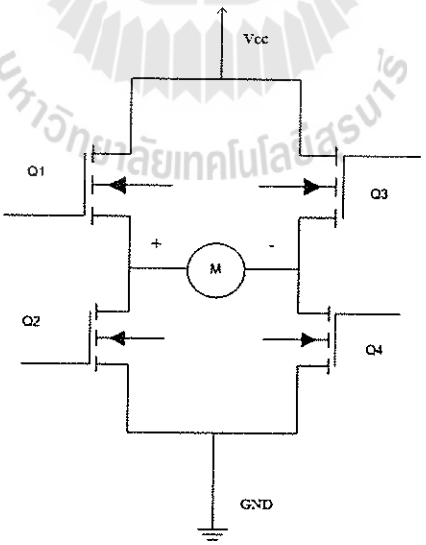
บทที่ 4

วงจรขับเคลื่อนและการควบคุม

4.1 กล่าวนำ

กลไกที่ใช้ขับเคลื่อนหุ่นยนต์ โดยส่วนมากจะเป็นมอเตอร์ และการควบคุมมอเตอร์ดังกล่าวต้องอาศัยวงจรควบคุม เพื่อควบคุมการทำงานของมอเตอร์ เช่น ให้มอเตอร์หมุนไปข้างหน้า ให้มอเตอร์หมุนถอยหลัง หรือการทำให้มอเตอร์หยุดหมุน ซึ่งการควบคุมดังกล่าวจะต้องใช้การสั่งงานจากโปรแกรมเพื่อควบคุมมอเตอร์ให้ทำงานตามที่ต้องการ ซึ่งในบทนี้จะกล่าวถึงวงจรขับเคลื่อนและการควบคุม ชุดขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงสำหรับงานประยุกต์ทางด้านหุ่นยนต์ การใช้สัญญาณ PWM เพื่อควบคุมความเร็วของมอเตอร์ รวมไปถึงการสั่งงานจากโปรแกรมสำหรับหุ่นยนต์อัตโนมัติและหุ่นยนต์บังคับมือ

4.2 วงจรควบคุมมอเตอร์

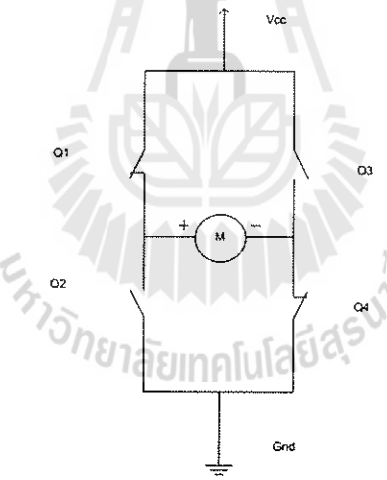


รูปที่ 4.1 วงจรควบคุมมอเตอร์

วงจรควบคุมมอเตอร์เป็นวงจรใช้ควบคุมมอเตอร์ให้หมุนไปข้างหน้า หมุนถอยหลัง และเบรก ซึ่งถูกควบคุมโดยมอสเฟต 4 ตัว ดังรูปที่ 4.1 ซึ่งไอซี IR2110 หนึ่งตัว ใช้ขับเคลื่อนมอสเฟตได้สองตัว ดังนั้นในวงจรควบคุมมอเตอร์นี้จะใช้ไอซี IR2110 สองตัว โดยจากรูปที่ 4.1 ไอซี IR2110 ตัวแรก ใช้ขับเคลื่อนมอสเฟต Q1,Q2 และ ไอซี IR2110 ตัวที่สอง ใช้ขับเคลื่อนมอสเฟต Q3,Q4 โดยจะแบ่งโหมดการทำงาน เป็นดังต่อไปนี้

4.2.1 มอเตอร์หมุนไปข้างหน้า

จากรูปที่ 4.1 มอเตอร์จะหมุนไปข้างหน้าเมื่อมอสเฟตQ1,Q4 ทำงาน และมอสเฟต Q2,Q3 ไม่ทำงาน โดยสั่งการทำงานจากบอร์ดไมโครคอนโทรลเลอร์ ซึ่งเมื่อมอสเฟตทำงาน กระแสไฟฟ้าจากแหล่งจ่าย V_{CC}จะไหลผ่านขั้วบวกของมอเตอร์ลงกราวด์ ทำให้มอเตอร์หมุนไปข้างหน้า ดังรูปที่ 4.2

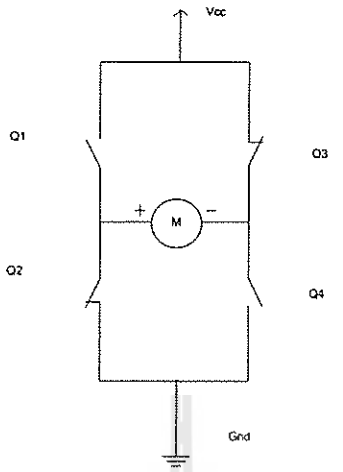


รูปที่ 4.2 วงจรมอเตอร์หมุนไปข้างหน้า

4.1.2 มอเตอร์หมุนถอยหลัง

จากรูปที่ 4.1 มอเตอร์จะหมุนถอยหลังเมื่อมอสเฟต Q2,Q3 ทำงาน และมอสเฟต Q1,Q4 ไม่ทำงาน โดยสั่งการทำงานจากบอร์ดไมโครคอนโทรลเลอร์ ซึ่งเมื่อมอสเฟต

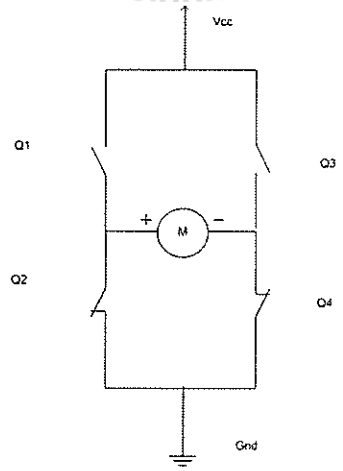
ทำงาน กระแสไฟฟ้าจากแหล่งจ่าย V_{CC} จะไหลผ่านขั้วลบของมอเตอร์ลงกราวด์ ทำให้มอเตอร์หมุนถอยหลัง ดังรูปที่ 4.3



รูปที่ 4.3 วงจรมอเตอร์หมุนถอยหลัง

4.1.3 มอเตอร์เบรกทางไฟฟ้า

จากรูปที่ 4.1 มอเตอร์จะเบรกเมื่อมอสเฟต Q2,Q4 ทำงาน และมอสเฟต Q1,Q3 ไม่ทำงาน โดยสั่งการทำงานจากบอร์ดไมโครคอนโทรลเลอร์ ซึ่งเมื่อมอสเฟตทำงาน ขั้วบวกและขั้วลบของมอเตอร์จะลัดวงจร มอเตอร์จะคายกระแสออกมาเพราะในมอเตอร์มีขดลวด จะทำหน้าที่เป็นแหล่งจ่ายอีกตัวหนึ่ง จึงทำให้มอเตอร์เบรกได้ดังรูปที่ 4.4

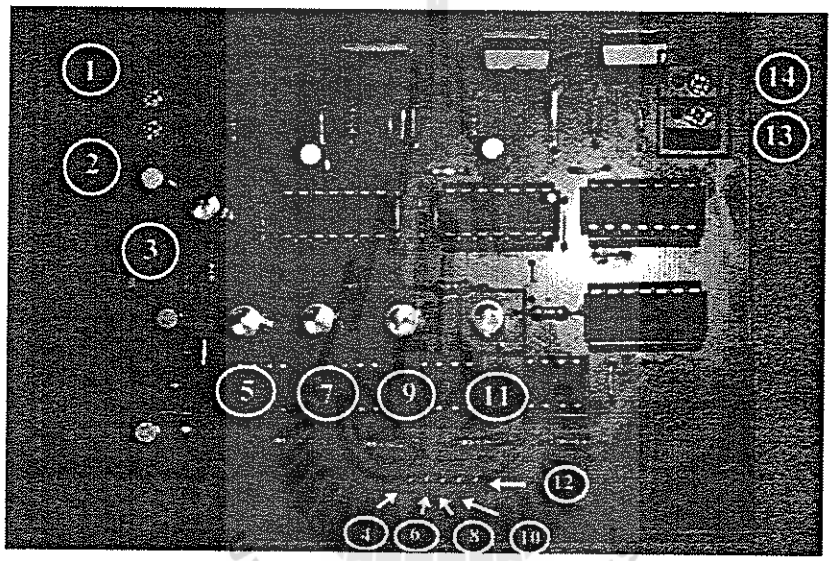


รูปที่ 4.4 วงจรมอเตอร์เบรกทางไฟฟ้า

หมายเหตุ วงจรนี้มีข้อห้าม คือ ห้ามมอสเฟต Q1,Q2 ทำงานพร้อมกัน หรือ มอสเฟต Q3,Q4 ทำงานพร้อมกัน เพราะ V_{CC} จะลัดวงจรลงกราวด์ทำให้มอสเฟตเกิดความเสียหายได้

4.3 ชุดควบคุมมอเตอร์ไฟฟ้ากระแสตรงสำหรับหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

จากวงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรงนำมาสร้างเป็นชุดขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงสำหรับหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส ได้ดังรูปที่ 4.5 ดังนี้



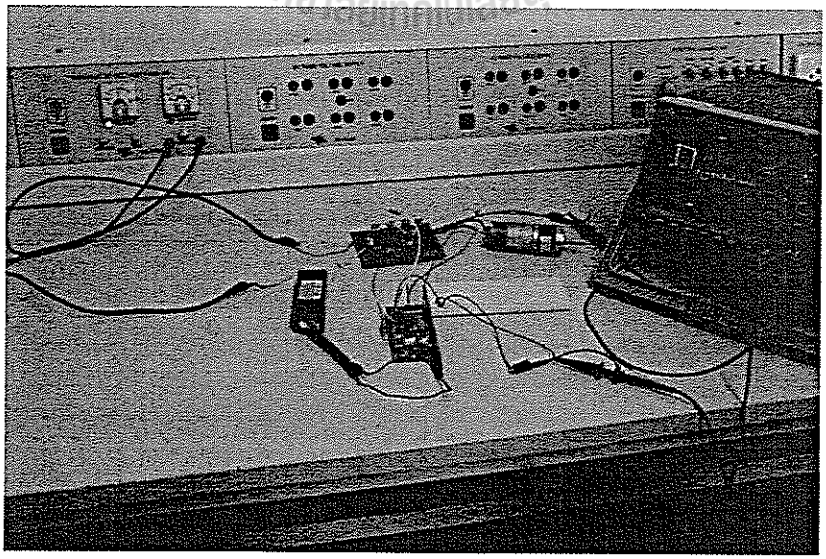
รูปที่ 4.5 ชุดขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงสำหรับหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

- *หมายเลขที่ 1* และ 2 คือ Vcc โดยที่ หมายเลข 1 เป็นขั้วบวก และ หมายเลข 2 เป็นขั้วลบ
- *หมายเลขที่ 3* คือ ไฟแสดงสถานะ
- *หมายเลขที่ 4* คือขา Q1 ซึ่งจะควบคุมมอสเฟต Q1 โดยขานี้จะเป็นตัวควบคุมความเร็วโดยใช้ขา PWM จากบอร์ดไมโครคอนโทรลเลอร์
- *หมายเลขที่ 5* คือ ไฟแสดงสถานะเมื่อ Q1 ทำงาน
- *หมายเลขที่ 6* คือ ขา Q2 ซึ่งจะควบคุมมอสเฟต Q2 โดยใช้ขา Digital จากบอร์ดไมโครคอนโทรลเลอร์
- *หมายเลขที่ 7* คือ ไฟแสดงสถานะเมื่อ Q2 ทำงาน

- *หมายเลขที่ 8* คือ ขา Q3 ซึ่งจะควบคุมมอสเฟต Q3 โดยขานี้จะเป็นตัวควบคุมความเร็วโดยใช้ขา PWM จากบอร์ดไมโครคอนโทรลเลอร์
- *หมายเลขที่ 9* คือ ไฟแสดงสถานะเมื่อ Q3 ทำงาน
- *หมายเลขที่ 10* คือ ขา Q4 ซึ่งจะควบคุมมอสเฟต Q4 โดยใช้ขา Digital จากบอร์ดไมโครคอนโทรลเลอร์
- *หมายเลขที่ 11* คือ ไฟแสดงสถานะเมื่อ Q4 ทำงาน
- *หมายเลขที่ 12* คือ ขากราวด์จากบอร์ดไมโครคอนโทรลเลอร์
- *หมายเลขที่ 13 และ 14* คือ ขั้วของมอเตอร์

4.4 การใช้สัญญาณ PWM จากบอร์ดไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 สำหรับควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรง

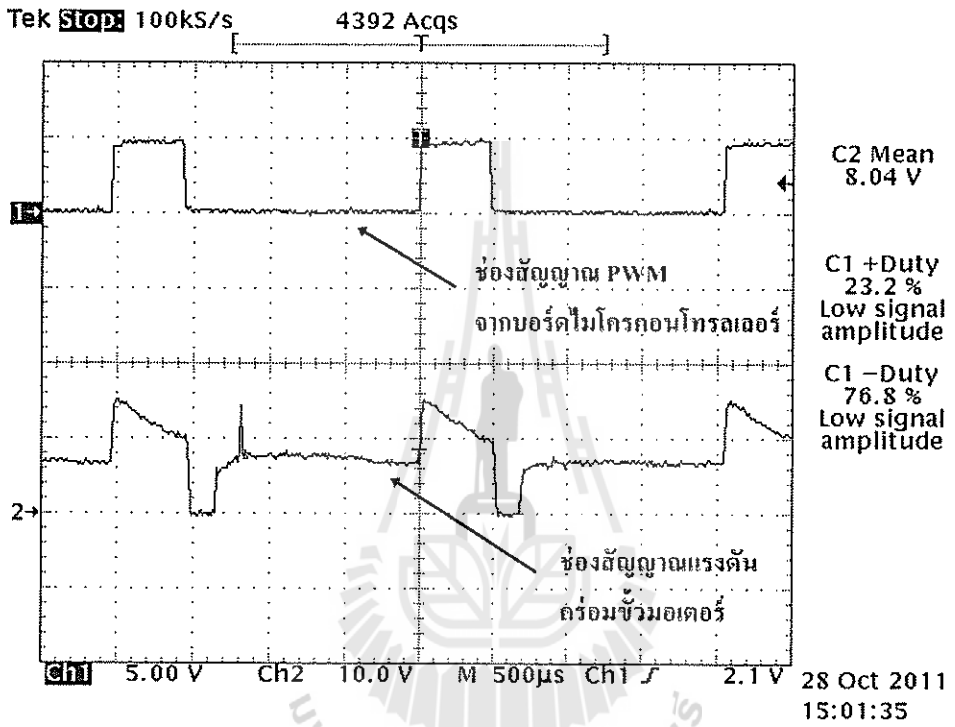
การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงดังกล่าวสามารถทำได้ 2 วิธีซึ่งในงานนี้ใช้การปรับแรงดันอาร์เมเจอร์ (E_a) ผ่านบอร์ดไมโครคอนโทรลเลอร์มาช่วยในการควบคุมความเร็วของมอเตอร์ โดยจะใช้สัญญาณ PWM ขนาด 8 บิต จากบอร์ดไมโครคอนโทรลเลอร์ที่สั่งงานจากโปรแกรม Arduino ซึ่งจะปรับแรงดันอาร์เมเจอร์ (E_a) ที่คร่อมขั้วของมอเตอร์ทำให้ปรับความเร็วของมอเตอร์ได้ โดยจะปรับสัญญาณ PWM ได้ตั้งแต่ 0-255 ออกทางขาสัญญาณ D2-D13 ของบอร์ดไมโครคอนโทรลเลอร์ET-EASY MEGA1280ไปที่วงจรควบคุมมอเตอร์เพื่อควบคุมความเร็วมอเตอร์ โดยการทดสอบแสดงได้ดังรูปที่ 4.6 ดังนี้



รูปที่ 4.6 วงจรทดสอบความเร็วของมอเตอร์กระแสตรงโดยใช้สัญญาณ PWM

จากรูปที่ 4.6 ช่องสัญญาณที่ 1 ของออสซิสโลสโคปทำหน้าที่จับสัญญาณ PWM จากบอร์ดไมโครคอนโทรลเลอร์ และช่องสัญญาณที่ 2 ของออสซิสโลสโคปทำหน้าที่จับสัญญาณแรงดันคร่อมขั้วของมอเตอร์

- ปรับสัญญาณ PWM ไปที่ 60



รูปที่ 4.7 ปรับสัญญาณ PWM จากบอร์ดไมโครคอนโทรลเลอร์ไปที่ 60

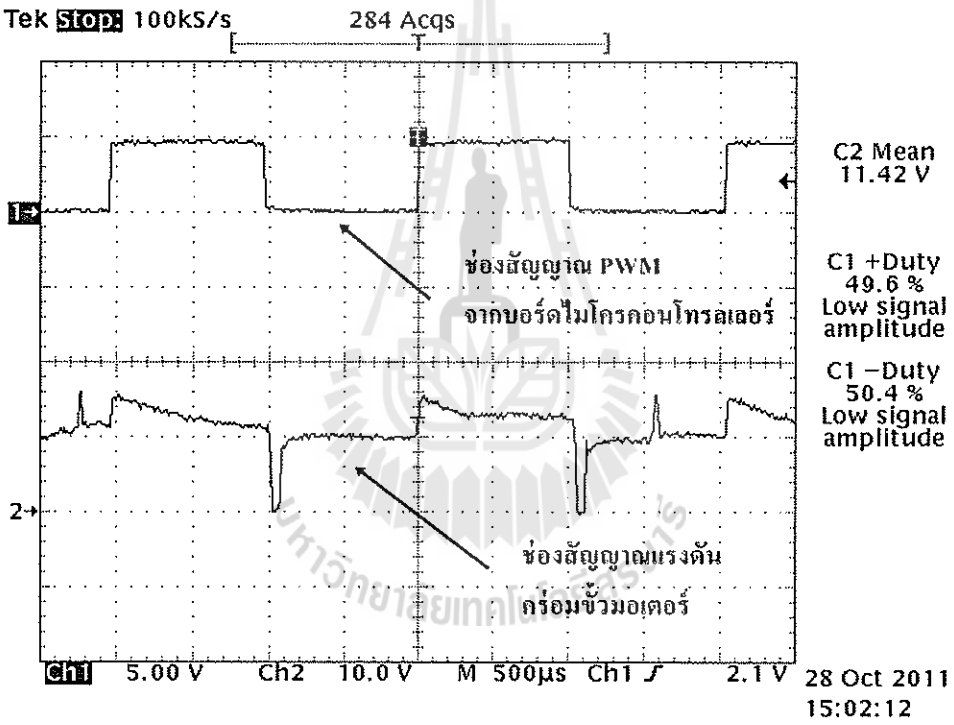
จากทฤษฎี ค่า % Duty cycle นั้นสามารถหาได้จากค่า $\frac{Ton}{Toff}$ x 100% และเมื่อทำการปรับสัญญาณ PWM ไปที่ 60 ค่า % Duty cycle ที่คำนวณได้คือ 23.53% โดยมีค่าใกล้เคียงกับกราฟรูปที่ 4.7 มาก ซึ่งหมายถึง ที่สัญญาณ PWM ที่ 60 มอเตอร์หมุนด้วยความเร็ว 23.2%ของความเร็วทั้งหมด

ตัวอย่างโปรแกรม

```
int Q1=2,Q2=40,Q3=3,Q4=41;
    void setup(){
        pinMode(Q1,OUTPUT);
        pinMode(Q2,OUTPUT);
```

```
        pinMode(Q3,OUTPUT);
        pinMode(Q4,OUTPUT); }
void loop(){
        digitalWrite(Q2,LOW);
        digitalWrite(Q3,LOW);
        analogWrite(Q1,60);                    // Q1 เป็นตัวปรับสัญญาณ PWM
        digitalWrite(Q4,HIGH); }
```

- ปรับ PWM ไปที่ 120



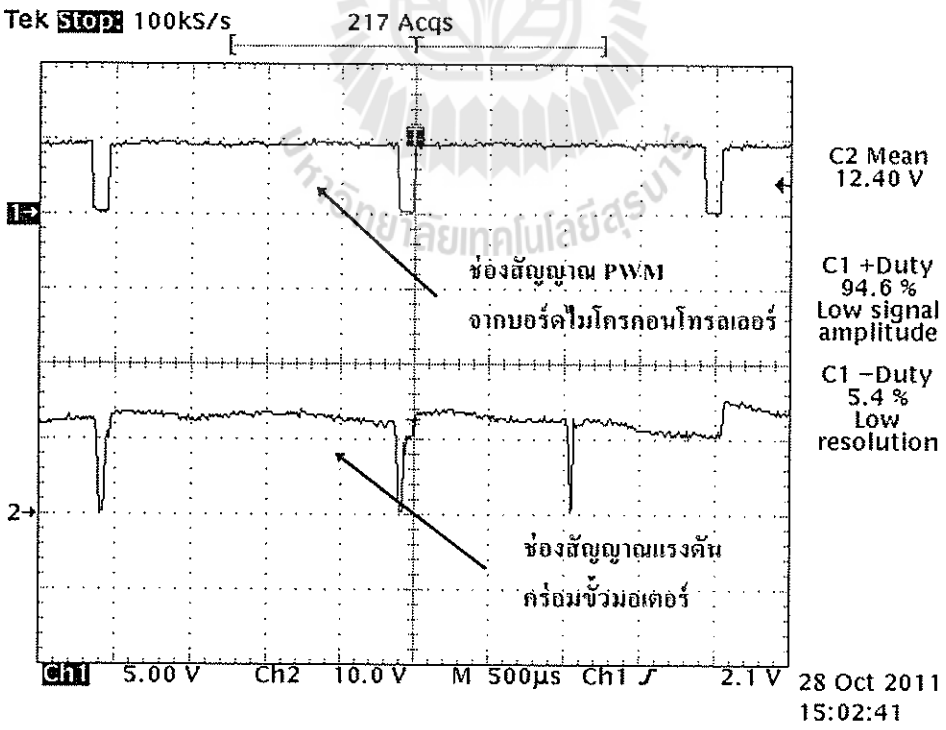
รูปที่ 4.8 ปรับสัญญาณ PWM จากบอร์ดไมโครคอนโทรลเลอร์ไปที่ 120

เมื่อทำการปรับสัญญาณ PWM ไปที่ 120 ค่า % Duty cycle ที่คำนวณได้คือ 47.06 %โดยมีใกล้เคียงกับกราฟรูปที่ 4.8 มาก ซึ่งหมายถึง ที่สัญญาณ PWM ที่ 120 มอเตอร์หมุนด้วยความเร็ว .6 % ของความเร็วทั้งหมด

ตัวอย่างโปรแกรม

```
nt Q1=2,Q2=40,Q3=3,Q4=41;
        void setup(){
                pinMode(Q1,OUTPUT);
                pinMode(Q2,OUTPUT);
                pinMode(Q3,OUTPUT);
                pinMode(Q4,OUTPUT); }
        void loop(){
                digitalWrite(Q2,LOW);
                digitalWrite(Q3,LOW);
                analogWrite(Q1,120);                    // Q1 เป็นตัวปรับสัญญาณ PWM
                digitalWrite(Q4,HIGH);  }
```

- ปรับ PWM ไปที่ 240



รูปที่ 4.9 ปรับสัญญาณ PWM จากบอร์ดไมโครคอนโทรลเลอร์ไปที่ 240

เมื่อทำการปรับสัญญาณ PWM ไปที่ 240 ค่า % Duty cycle ที่คำนวณได้คือ 94.12 %โดยมีค่าใกล้เคียงกับกราฟรูปที่ 4.9 มาก ซึ่งหมายถึง ที่สัญญาณ PWM ที่ 240 มอเตอร์หมุนด้วยความเร็ว 94.6% ของความเร็วทั้งหมด

ตัวอย่างการโปรแกรม

```
int Q1=2,Q2=40,Q3=3,Q4=41;
    void setup(){
        pinMode(Q1,OUTPUT);
        pinMode(Q2,OUTPUT);
        pinMode(Q3,OUTPUT);
        pinMode(Q4,OUTPUT); }
    void loop(){
        digitalWrite(Q2,LOW);
        digitalWrite(Q3,LOW);
        analogWrite(Q1,240);                    // Q1 เป็นตัวปรับสัญญาณ PWM
        digitalWrite(Q4,HIGH);  }
```

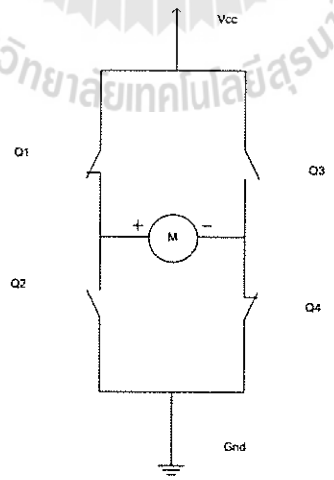
4.5 การโปรแกรมเพื่อควบคุมมอเตอร์ไฟฟ้ากระแสตรงสำหรับหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

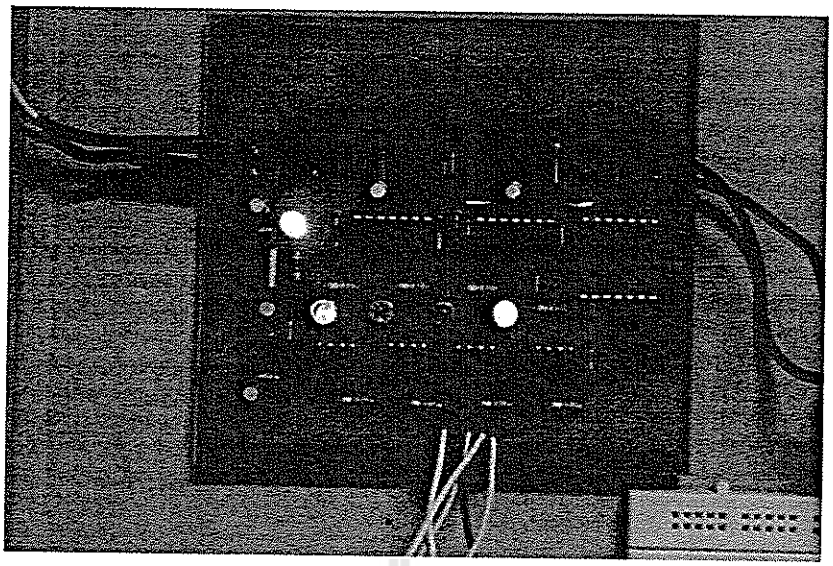
การโปรแกรมเพื่อควบคุมมอเตอร์ไฟฟ้ากระแสตรงสำหรับหุ่นยนต์อัตโนมัติ จะต้องเข้าใจหลักการทำงานในแต่ละโหมดของวงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรงก่อน จึงจะควบคุมการทำงานของมอเตอร์ไฟฟ้ากระแสตรงในแต่ละโหมดได้ถูกต้อง การโปรแกรมเพื่อควบคุมมอเตอร์ไฟฟ้ากระแสตรงจะใช้โปรแกรม Arduino ในการเขียนโปรแกรมอัพโหลดลงบอร์ดไมโครคอนโทรลเลอร์ เพื่อสั่งงานชุดขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง

จากรูปที่ 4.1 เป็นวงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรงสำหรับงานนี้ การทำงานของวงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรงจะแบ่งเป็น 3 โหมดการทำงาน คือ มอเตอร์หมุนไปข้างหน้า มอเตอร์หมุนถอยหลัง และมอเตอร์เบรกทางไฟฟ้า เป็นต้น การทำงานในแต่ละโหมดนั้นจะสั่งงานโดยบอร์ดไมโครคอนโทรลเลอร์ บอร์ดไมโครคอนโทรลเลอร์สำหรับหัวข้อนี้ คือ บอร์ด ET-EASY MEGA1280 จะใช้สัญญาณ PWM จากบอร์ดไมโครคอนโทรลเลอร์ ขาสัญญาณ PWM จากบอร์ด ET-EASY MEGA1280 ได้แก่ ขา D2-D13 เป็นต้น

โดยจะกำหนดให้ Q1 จากชุดขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงต่อกับขา D2 จากบอร์ดไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 เพราะ Q1 จากชุดขับเคลื่อนใช้ควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงโดยใช้สัญญาณ PWM และ Q2 จากชุดขับเคลื่อนจะต่อกับขา D40 จากบอร์ดไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 และ Q3 ต่อกับขา D3 จากบอร์ดไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 เพราะ Q3 จากชุดขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงจะใช้ควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงโดยใช้สัญญาณ PWM และ Q4 จากชุดขับเคลื่อนจะต่อกับขา D41 จากบอร์ดไมโคคอนโทรลเลอร์ ET-EASY MEGA1280

4.5.1 มอเตอร์หมุนไปข้างหน้า





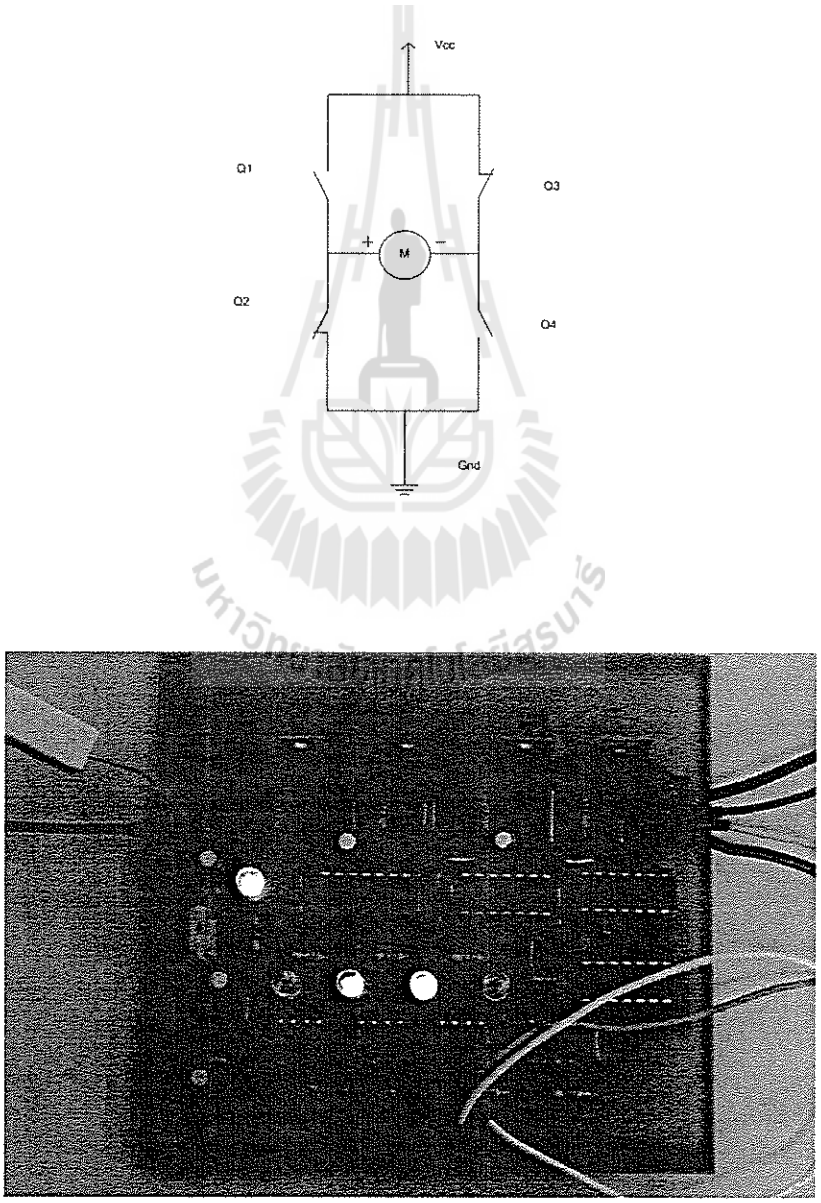
รูปที่ 4.10 การโปรแกรมเพื่อให้มอเตอร์หมุนไปข้างหน้า

การเขียนโปรแกรม

```
int Q1=2,Q2=40,Q3=3,Q4=41;                  // กำหนดตัวแปรให้ตรงกับขาสัญญาณ
                                                ที่ใช้งาน
    void setup(){                           //ฟังก์ชันตั้งค่า
        pinMode(Q1,OUTPUT);                 //กำหนดค่า Q1 เป็นเอาต์พุต
        pinMode(Q2,OUTPUT);                 //กำหนดค่า Q2 เป็นเอาต์พุต
        pinMode(Q3,OUTPUT);                 //กำหนดค่า Q3 เป็นเอาต์พุต
        pinMode(Q4,OUTPUT);                 //กำหนดค่า Q4 เป็นเอาต์พุต
    }
    void loop(){                            //ฟังก์ชันการทำงาน
        digitalWrite(Q2,LOW);               //สั่งงาน Q2 ให้เป็น LOW
        digitalWrite(Q3,LOW);               //สั่งงาน Q3 ให้เป็น LOW
        analogWrite(Q1,120);                // Q1 เป็นตัวปรับความเร็ว
        digitalWrite(Q4,HIGH);              //สั่งงาน Q4 ให้เป็น HIGH
    }
```

คำสั่ง digital Write สามารถสั่งงานขาสัญญาณของบอร์ดไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 ได้สองแบบเท่านั้น คือ HIGH และ LOW และคำสั่ง analogWrite เป็นคำสั่งที่ใช้ในการปรับสัญญาณ PWM หรือ ปรับความเร็วของมอเตอร์ ขาสัญญาณ PWM ของบอร์ดไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 เป็น 8 บิต จะสามารถปรับสัญญาณ PWM ได้ตั้งแต่ 0-255 และจากรูปที่ จะเห็นว่า มอเตอร์หมุนไปข้างหน้า เมื่อ Q1,Q4 ทำงาน และ Q2,Q3 ไม่ทำงาน โดยที่ Q1 เป็นตัวปรับความเร็วของมอเตอร์โดยปรับความเร็วได้ตั้งแต่ 0-255

4.5.2 มอเตอร์หมุนไปถอยหลัง



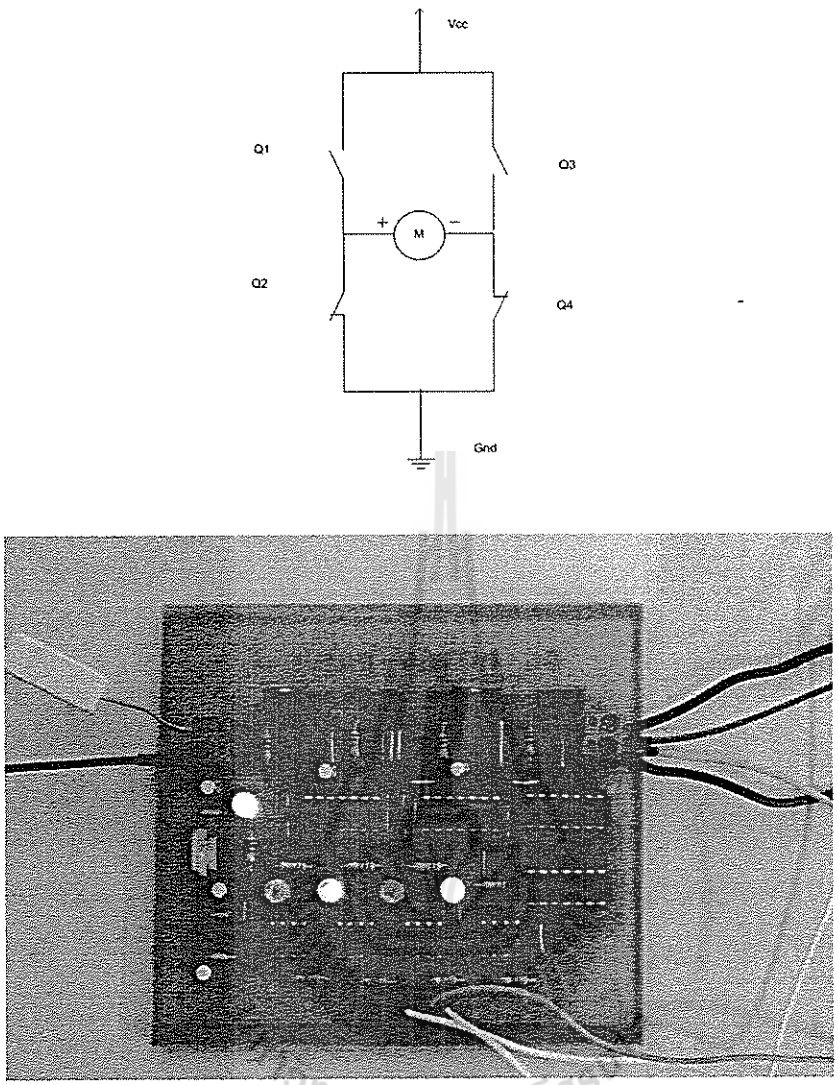
รูปที่ 4.11 การโปรแกรมเพื่อให้มอเตอร์หมุนถอยหลัง

การเขียนโปรแกรม

```
int Q1=2,Q2=40,Q3=3,Q4=41;                    // กำหนดตัวแปรให้ตรงกับขาสัญญาณ
                                                ที่ใช้งาน
    void setup(){                              //ฟังก์ชันตั้งค่า
        pinMode(Q1,OUTPUT);                    //กำหนดค่า Q1 เป็นเอาต์พุต
        pinMode(Q2,OUTPUT);                    //กำหนดค่า Q2 เป็นเอาต์พุต
        pinMode(Q3,OUTPUT);                    //กำหนดค่า Q3 เป็นเอาต์พุต
        pinMode(Q4,OUTPUT);                    //กำหนดค่า Q4 เป็นเอาต์พุต
        }
    void loop(){                               //ฟังก์ชันการทำงาน
        digitalWrite(Q1,LOW);                  //สั่งงาน Q1 ให้เป็น LOW
        digitalWrite(Q4,LOW);                  //สั่งงาน Q4 ให้เป็น LOW
        analogWrite(Q3,120);                   // Q3 เป็นตัวปรับความเร็ว
        digitalWrite(Q2,HIGH);                 //สั่งงาน Q2 ให้เป็น HIGH
        }
```

คำสั่ง digitalWrite สามารถสั่งงานขาสัญญาณของบอร์ดไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 ได้สองแบบเท่านั้น คือ HIGH และ LOW และคำสั่ง analogWrite เป็นคำสั่งที่ใช้ในการปรับสัญญาณ PWM หรือ ปรับความเร็วของมอเตอร์ ขาสัญญาณ PWM ของบอร์ดไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 เป็น 8 บิต จะสามารถปรับสัญญาณ PWM ได้ตั้งแต่ 0-255 และจากรูปที่ จะเห็นว่า มอเตอร์หมุนถอยหลัง เมื่อ Q3,Q2 ทำงาน และ Q1,Q4 ไม่ทำงาน โดยที่ Q3 เป็นตัวปรับความเร็วของมอเตอร์โดยปรับความเร็วได้ตั้งแต่ 0-255

4.5.2 มอเตอร์เบรกทางไฟฟ้า



รูปที่ 4.12 การโปรแกรมเพื่อให้มอเตอร์เบรกทางไฟฟ้า

การเขียนโปรแกรม

```
int Q1=2,Q2=40,Q3=3,Q4=41;                    // กำหนดตัวแปรให้ตรงกับขาสัญญาณ
                                                 ที่ใช้งาน
    void setup(){                              //ฟังก์ชันตั้งค่า
        pinMode(Q1,OUTPUT);                    //กำหนดค่า Q1 เป็นเอาต์พุต
        pinMode(Q2,OUTPUT);                    //กำหนดค่า Q2 เป็นเอาต์พุต
        pinMode(Q3,OUTPUT);                    //กำหนดค่า Q3 เป็นเอาต์พุต
```

```
        pinMode(Q4,OUTPUT);          //กำหนดค่า Q4 เป็นเอาต์พุต
}
void loop(){                         //ฟังก์ชันการทำงาน
        digitalWrite(Q1,LOW);        //สั่งงาน Q1 ให้เป็น LOW
        digitalWrite(Q3,LOW);        //สั่งงาน Q3 ให้เป็น LOW
        digitalWrite(Q2,HIGH);       //สั่งงาน Q2 ให้เป็น HIGH
        digitalWrite(Q4,HIGH);       //สั่งงาน Q4 ให้เป็น HIGH
}
```

คำสั่ง digitalWrite สามารถสั่งงานขาสัญญาณของบอร์ดไมโครคอนโทรลเลอร์ ET-EASY MEGA1280 ได้สองแบบเท่านั้น คือ HIGH และ LOW และจากรูปที่ จะเห็นว่ามอเตอร์เบรกทางไฟฟ้า เมื่อ Q2,Q4 ทำงาน และ Q1,Q3 ไม่ทำงาน

ตัวอย่างการเขียนโปรแกรมควบคุมมอเตอร์ไฟฟ้ากระแสตรงสำหรับหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

```
int Q1=2,Q2=40,Q3=3,Q4=41;           // กำหนดตัวแปรให้ตรงกับขาสัญญาณ
                                     ที่ใช้งาน
        void setup(){                //ฟังก์ชันตั้งค่า
                pinMode(Q1,OUTPUT);  //กำหนดค่า Q1 เป็นเอาต์พุต
                pinMode(Q2,OUTPUT);  //กำหนดค่า Q2 เป็นเอาต์พุต
                pinMode(Q3,OUTPUT);  //กำหนดค่า Q3 เป็นเอาต์พุต
                pinMode(Q4,OUTPUT);  //กำหนดค่า Q4 เป็นเอาต์พุต
        }
        void loop(){                 //ฟังก์ชันการทำงาน
                digitalWrite(Q2,LOW);//สั่งงาน Q2 ให้เป็น LOW
                digitalWrite(Q3,LOW);//สั่งงาน Q3 ให้เป็น LOW
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
analogWrite(Q1,120);          // Q1 เป็นตัวปรับความเร็ว
digitalWrite(Q4,HIGH);        //สั่งงาน Q4 ให้เป็น HIGH
      delay(4000);            //หน่วงเวลา 4 วินาที
digitalWrite(Q1,LOW);         //สั่งงาน Q1 ให้เป็น LOW
digitalWrite(Q3,LOW);         //สั่งงาน Q3 ให้เป็น LOW
digitalWrite(Q2,HIGH);        //สั่งงาน Q2 ให้เป็น HIGH
digitalWrite(Q4,HIGH);        //สั่งงาน Q4 ให้เป็น HIGH
      delay(1000);            //หน่วงเวลา 1 วินาที
digitalWrite(Q1,LOW);         //สั่งงาน Q1 ให้เป็น LOW
digitalWrite(Q4,LOW);         //สั่งงาน Q4 ให้เป็น LOW
analogWrite(Q3,120);          // Q3 เป็นตัวปรับความเร็ว
digitalWrite(Q2,HIGH);        //สั่งงาน Q2 ให้เป็น HIGH
      delay(4000);            //หน่วงเวลา 4 วินาที
}
```

การทำงานของโปรแกรม คือ มอเตอร์จะหมุนไปข้างหน้าเป็นเวลา 4 วินาที จากนั้น มอเตอร์จะเบรกเป็นเวลา 1 วินาที และมอเตอร์หมุนถอยหลังเป็นเวลา 4 วินาที วนลูปการ ทำงานไปเรื่อยๆ

มหาวิทยาลัยเทคโนโลยีสุรนารี

4.6 สรุป

เนื้อหาในบทนี้ได้อธิบาย การใช้สัญญาณ PWM เพื่อควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง การโปรแกรมเพื่อควบคุมมอเตอร์ไฟฟ้ากระแสตรงสำหรับหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส พร้อมทั้งอธิบายขั้นตอนในแต่ละโหมดการทำงานของวงจรขับเคลื่อนมอเตอร์ และส่วนต่างๆของชุดขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง

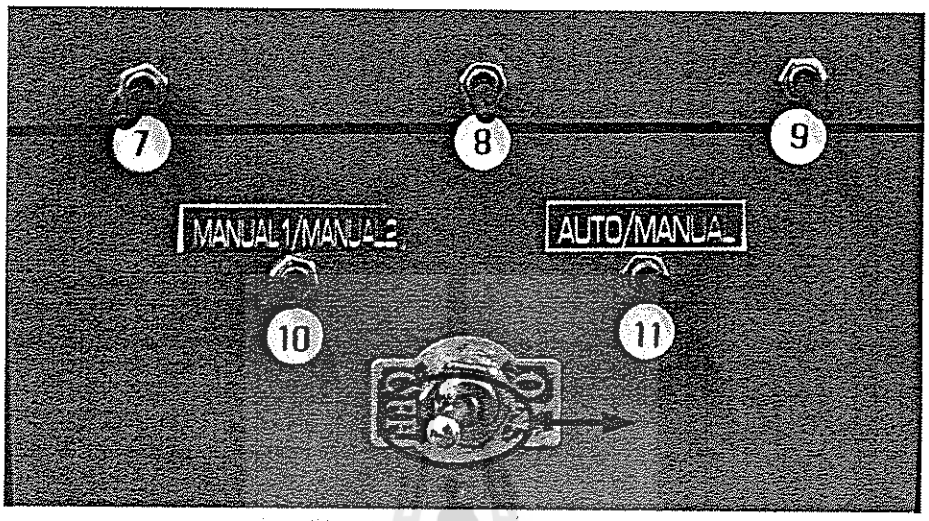
บทที่ 5

คู่มือการใช้งานหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

การใช้งานหุ่นยนต์ฝึกซ้อมกีฬาเทนนิสของนักกีฬาเทนนิสอาจเกิดความยุ่งยากในการใช้งาน เนื่องจากคุณลักษณะที่หลากหลายของหุ่นยนต์โดยมีโหมดการทำงานทั้งหมดด้วยกัน 3 โหมด เพื่อให้ง่ายต่อการใช้งานทางผู้ประดิษฐ์ได้จัดทำคู่มือการใช้งานอย่างละเอียดซึ่งจะนำเสนอในบทนี้

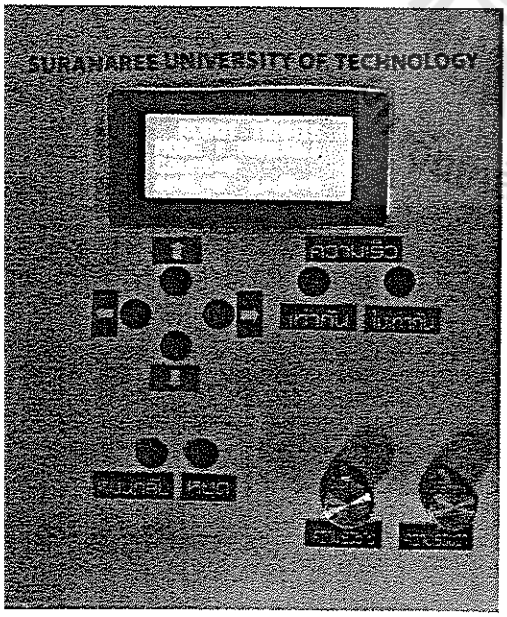
ขั้นตอนการใช้งานหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

เริ่มต้นจะต้องทำการเปิดเครื่อง โดยการสับสวิตช์ไปด้านขวามือ เพื่อให้หุ่นยนต์พร้อมสำหรับรับคำสั่งต่าง ๆ ต่อไป ดังรูปที่ 5.1

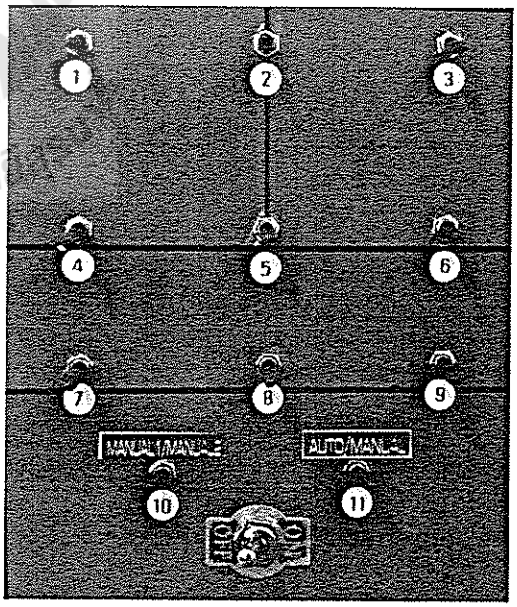


รูปที่ 5.1 การเปิดสวิตช์ของหุ่นยนต์ฝึกซ้อมกีฬาเทนนิส

หุ่นยนต์ตัวนี้ มีปุ่มสวิตช์ Manual 1 และ Manual 2 ดังรูปที่ 5.2 และ รูปที่ 5.3 ตามลำดับ



รูปที่ 5.2 ปุ่ม Manual 1

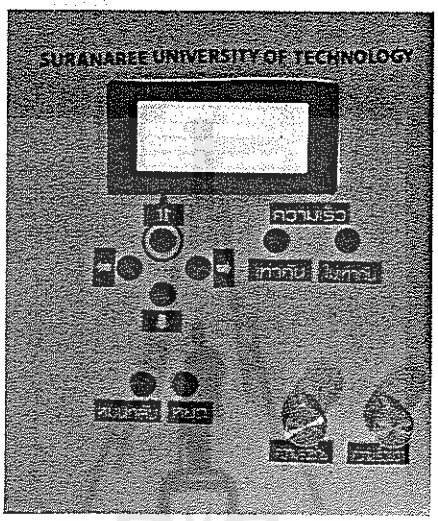


รูปที่ 5.3 ปุ่ม Manual 2

ซึ่งปุ่ม และสวิตช์ทั้งหลายเหล่านี้ จะใช้สำหรับบังคับหุ่นยนต์ฝึกซ้อมกีพาเทนนิส โดยแต่ละปุ่มจะทำงานดังนี้

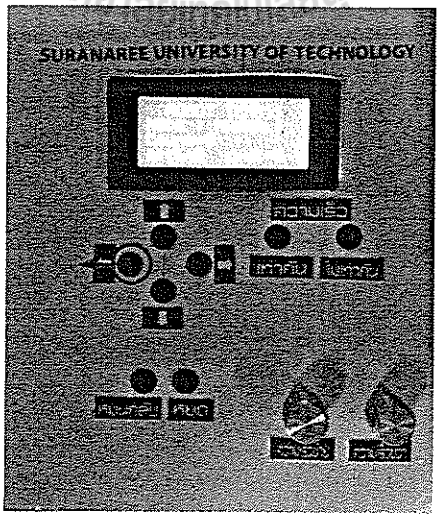
โหมด MANUAL 1 จะมีปุ่มการทำงานดังนี้

- ปุ่มที่ 1 เมื่อทำการกดปุ่มนี้ จะทำให้หุ่นยนต์ปรับมุมในการยิงขึ้นในแนวดิ่งได้ตามที่เราต้องการ ดังรูปที่ 5.4



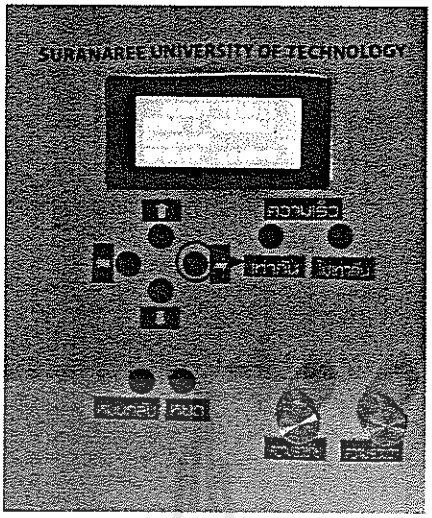
รูปที่ 5.4

- ปุ่มที่ 2 เมื่อทำการกดปุ่มนี้ จะทำให้หุ่นยนต์ปรับมุมในการยิงไปด้านซ้ายมือในแนวระดับ ได้ตามที่เราต้องการ ดังรูปที่ 5.5



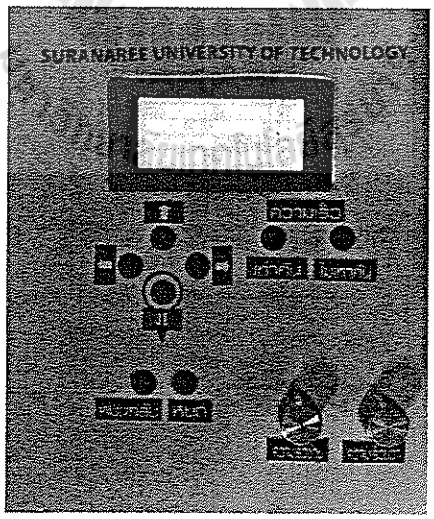
รูปที่ 5.5

- ปุ่มที่ 3 เมื่อทำการกดปุ่มนี้ จะทำให้หุ่นยนต์ปรับมุมในการยิงไปด้านขวามือในแนวระดับ ได้ตามที่เราต้องการ ดังรูปที่ 5.6



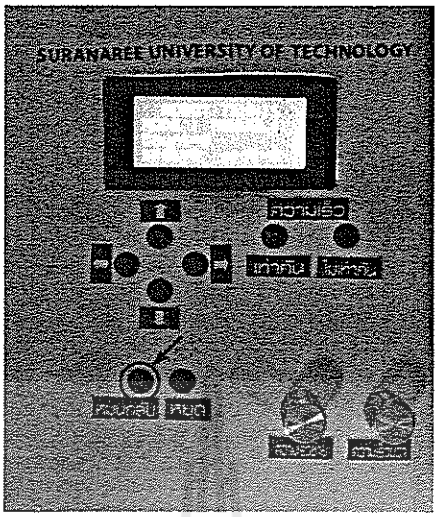
รูปที่ 5.6

- ปุ่มที่ 4 เมื่อทำการกดปุ่มนี้ จะทำให้หุ่นยนต์ปรับมุมในการยิงลงในแนวดิ่งได้ตามที่เราต้องการ ดังรูปที่ 5.7



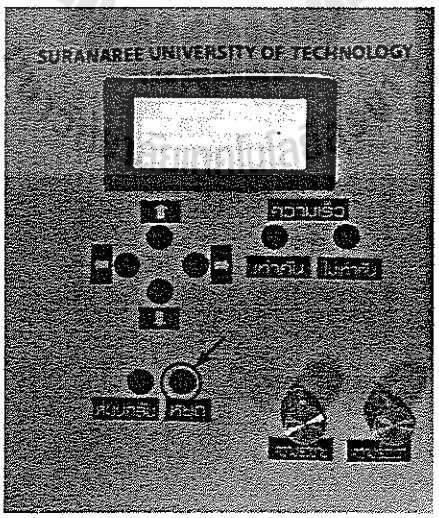
รูปที่ 5.7

- ปุ่มที่ 5 เมื่อทำการกดปุ่มนี้ จะทำให้ชุดเรียงลูกหมุนกลับทาง การกดปุ่มนี้จะกดในกรณีที่ มีลูกติดหรือลูกไม่ลงมายังชุดยิงลูกเทนนิส ดังรูปที่ 5.8



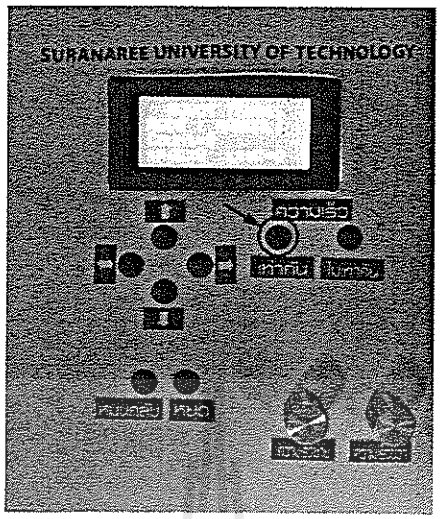
รูปที่ 5.8

- ปุ่มที่ 6 เมื่อทำการกดปุ่มนี้ จะทำให้ชุดเรียงลูกหยุดหมุน ในกรณีที่ ผู้เล่นยังไม่พร้อมที่จะเล่น ดังรูปที่ 5.9



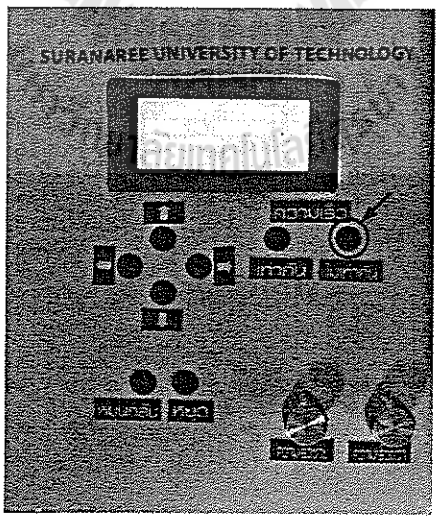
รูปที่ 5.9

- ปุ่มที่ 7 เมื่อทำการกดปุ่มนี้ จะทำให้หุ่นยนต์ปรับชุดยิงลูกเทนนิสทั้งสองล้อให้มีความเร็วรอบเท่ากัน ดังรูปที่ 5.10



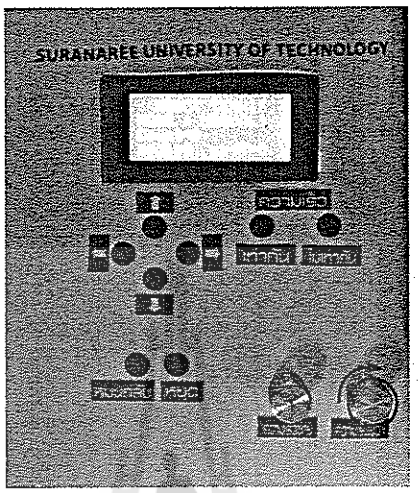
รูปที่ 5.10

- ปุ่มที่ 8 เมื่อทำการกดปุ่มนี้ จะทำให้หุ่นยนต์ปรับชุดยิงลูกเทนนิสทั้งสองล้อให้มีความเร็วรอบไม่เท่ากัน เมื่อต้องการยิงลูกไซส์ ดังรูปที่ 5.11



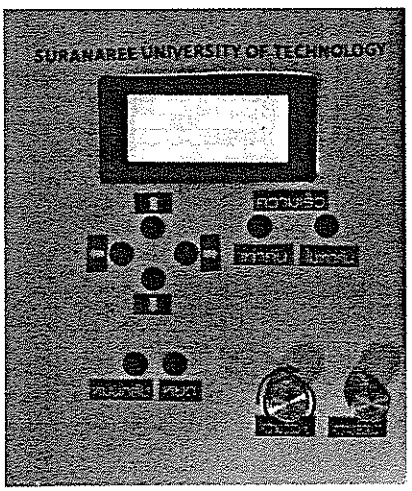
รูปที่ 5.11

- ปุ่มที่ 9 เมื่อทำการหมุนปุ่มนี้ไปทางด้านขวา จะทำให้ล้อชุดยิงลูกเทนนิสทางด้านขวาของหุ่นยนต์หมุนด้วยความเร็วเพิ่มขึ้น แต่ถ้าหมุนมาทางซ้าย ความเร็วจะลดลง ดังรูปที่ 5.12 (การหมุนปรับความเร็วนี้ต้องกดปุ่มที่ 8 เพื่อเลือกโหมดปรับความเร็วไม่เท่ากันเสียก่อน)



รูปที่ 5.12

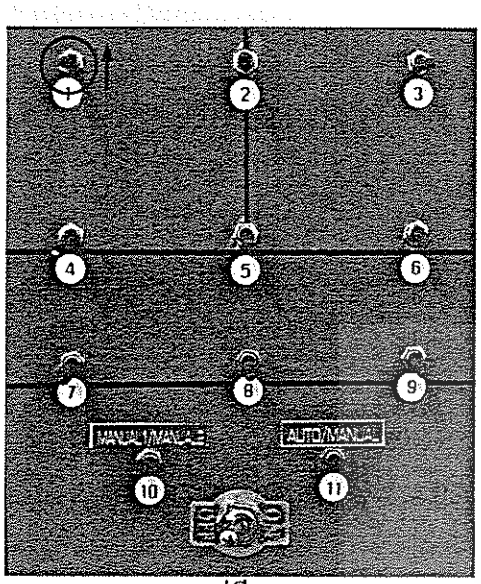
- ปุ่มที่ 13 เมื่อทำการหมุนปุ่มนี้ไปทางด้านขวา จะทำให้ล้อชุดยิงลูกเทนนิสทางด้านซ้ายของหุ่นยนต์หมุนด้วยความเร็วเพิ่มขึ้น แต่ถ้าหมุนมาทางซ้าย ความเร็วจะลดลง ดังรูปที่ 5.13 (การหมุนปรับความเร็วนี้ต้องกดปุ่มที่ 8 เพื่อเลือกโหมดปรับความเร็วไม่เท่ากันเสียก่อน) แต่ถ้าเลือกโหมดในกรณีความเร็วเท่ากัน การหมุนปรับที่ปุ่มนี้จะเป็นการปรับความเร็วของทั้งสองล้อ



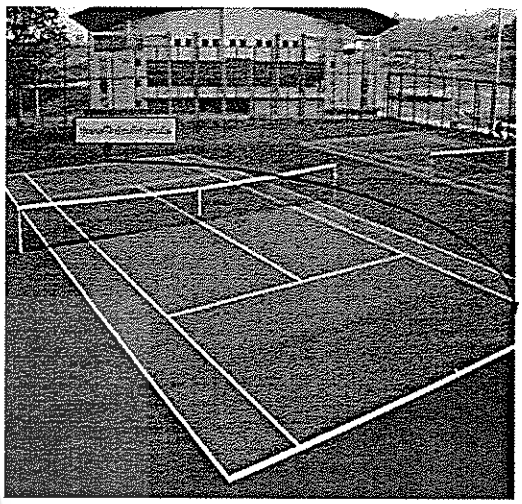
รูปที่ 5.13

โหมด MANUAL 2 จะมีปุ่มการทำงานดังนี้

-เมื่อทำการสับสวิตช์ที่ 1 ขึ้นด้านบน จะทำให้หุ่นยนต์ยิงลูกเทนนิสไปยังท้ายคอร์ดด้านซ้ายของหุ่นยนต์ดังรูปที่ 5.14 และรูปที่ 5.15

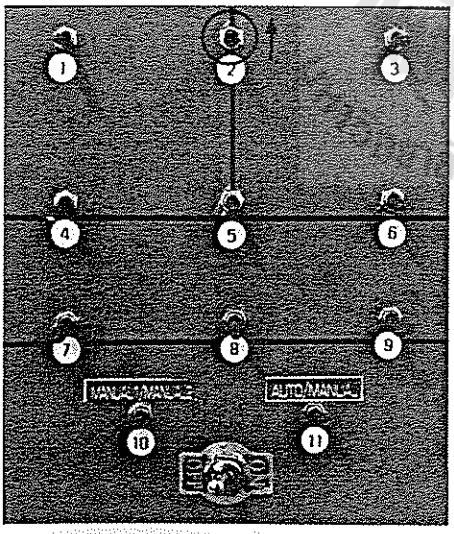


รูปที่ 5.14

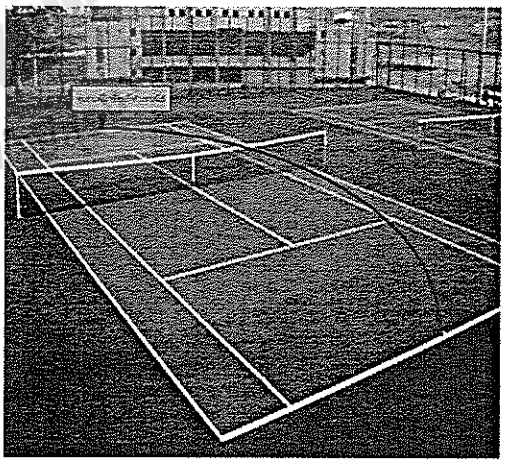


รูปที่ 5.15

-เมื่อทำการสับสวิตช์ที่ 2 ขึ้นด้านบน จะทำให้หุ่นยนต์ยิงลูกเทนนิสไปยังท้ายคอร์ดตรงกลางดังรูปที่ 5.16 และรูปที่ 5.17



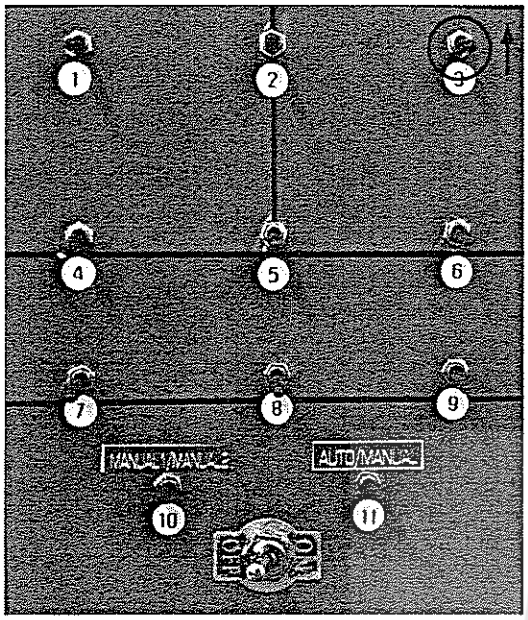
รูปที่ 5.16



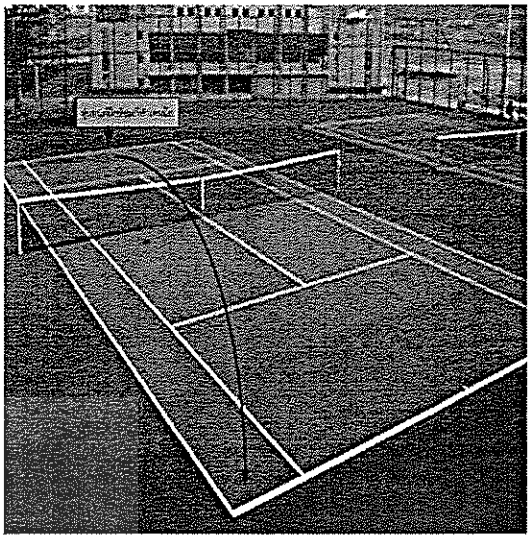
รูปที่ 5.17

-เมื่อทำการสับสวิตช์ที่ 3 ขึ้นด้านบน จะทำให้หุ่นยนต์ยิงลูกเทนนิสไปยังท้ายคอร์ดด้านขวาของหุ่นยนต์

ดังรูปที่ 5.18 และรูปที่ 5.19



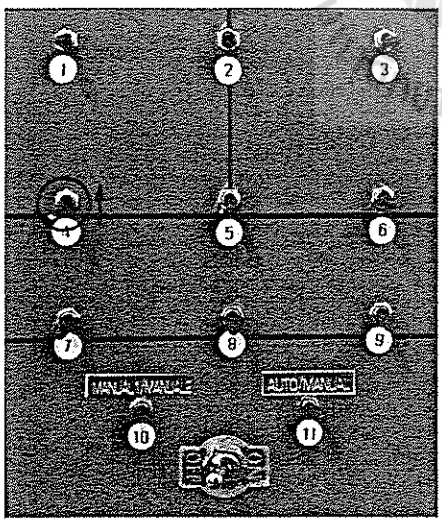
รูปที่ 5.18



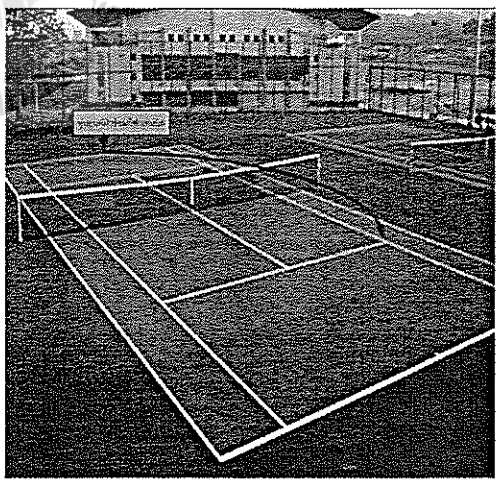
รูปที่ 5.19

-เมื่อทำการสับสวิตช์ที่ 4 ขึ้นด้านบน จะทำให้หุ่นยนต์ยิงลูกเทนนิสไปยังกลางคอร์ดด้านซ้ายของหุ่นยนต์

ดังรูปที่ 5.20 และรูปที่ 5.21



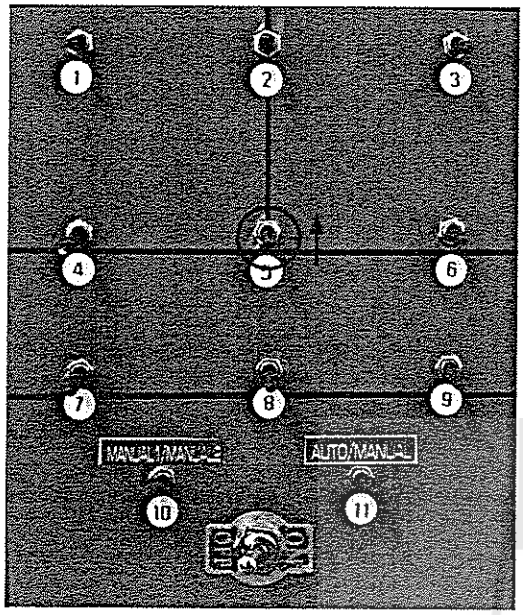
รูปที่ 5.20



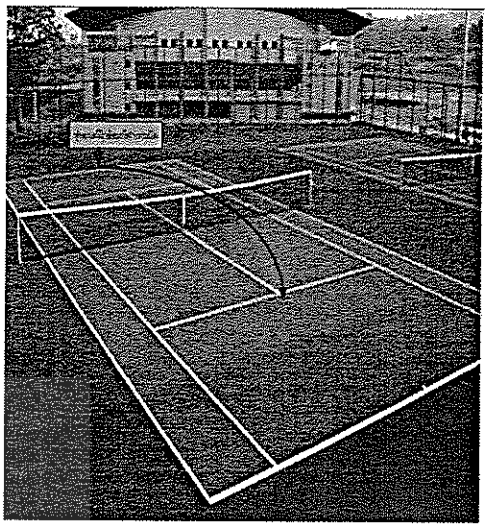
รูปที่ 5.21

-เมื่อทำการสับสวิตช์ที่ 5 ขึ้นด้านบน จะทำให้หุ่นยนต์ยิงลูกเทนนิสไปยังกลางคอร์ดตรงกลาง

ดังรูปที่ 5.22 และ รูปที่ 5.23



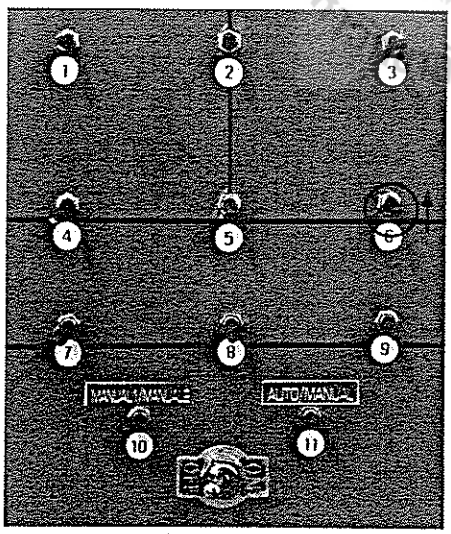
รูปที่ 5.22



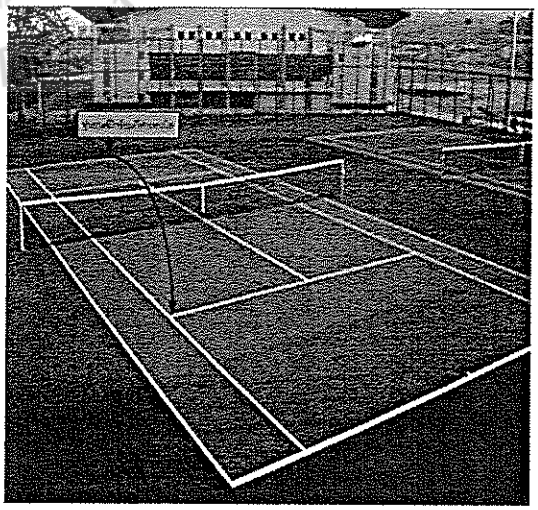
รูปที่ 5.23

-เมื่อทำการสับสวิตช์ที่ 6 ขึ้นด้านบน จะทำให้หุ่นยนต์ยิงลูกเทนนิสไปยังกลางคอร์ดด้านขวาของหุ่นยนต์

ดังรูปที่ 5.24 และ รูปที่ 5.25



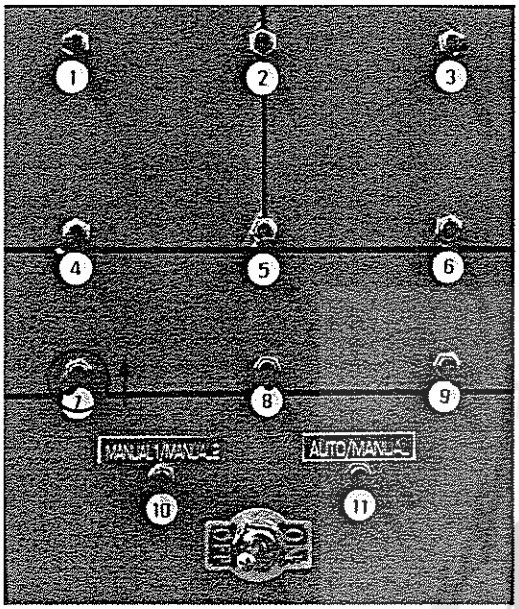
รูปที่ 5.24



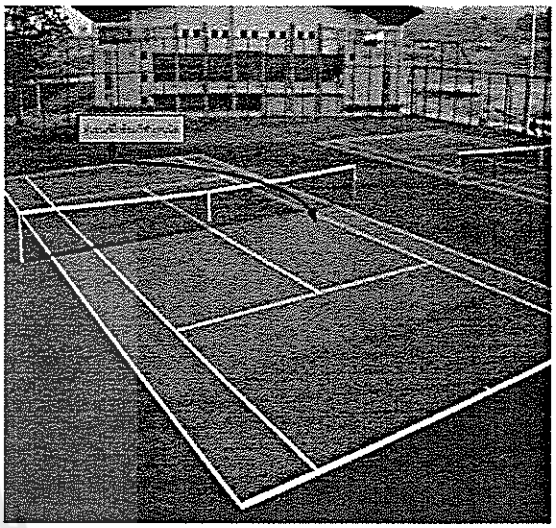
รูปที่ 5.25

-เมื่อทำการสับสวิตช์ที่ 7 ขึ้นด้านบน จะทำให้หุ่นยนต์ยิงลูกเทนนิสไปยังหน้าเน็ตด้านซ้ายของหุ่นยนต์

ดังรูปที่ 5.26 และรูปที่ 5.27



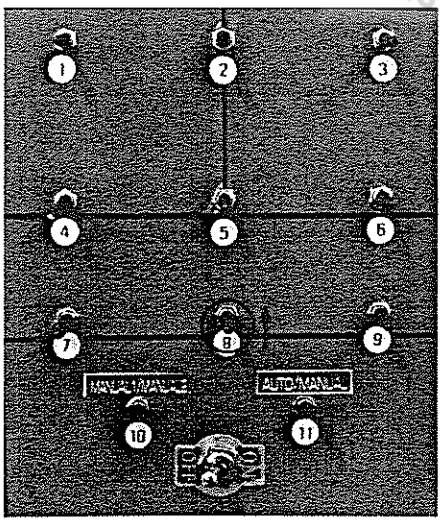
รูปที่ 5.26



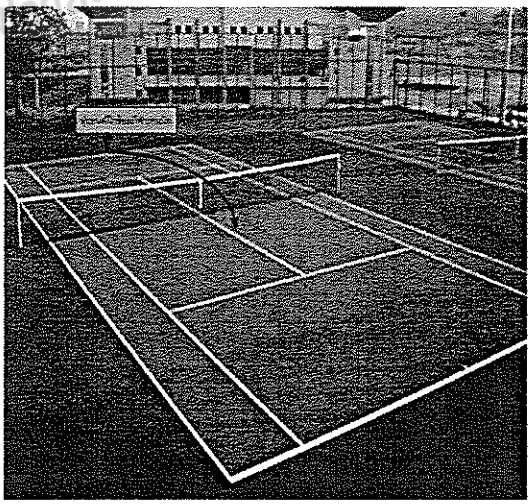
รูปที่ 5.27

-เมื่อทำการสับสวิตช์ที่ 8 ขึ้นด้านบน จะทำให้หุ่นยนต์ยิงลูกเทนนิสไปยังหน้าเน็ตตรงกลางสนาม

ดังรูปที่ 5.28 และรูปที่ 5.29



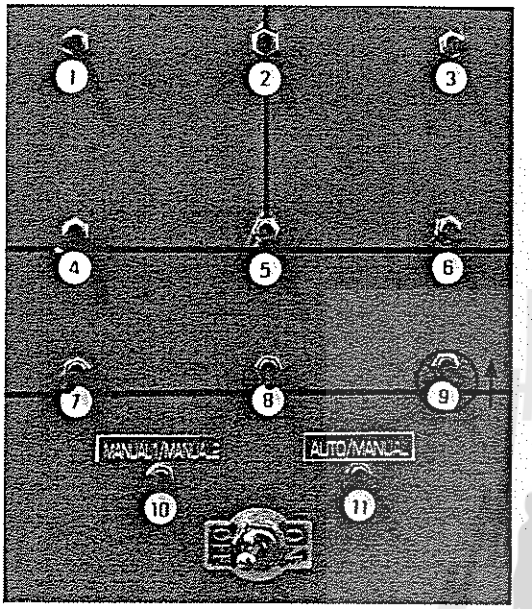
รูปที่ 5.28



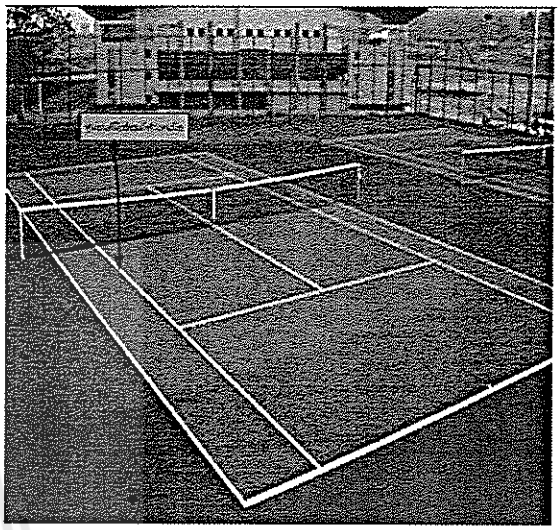
รูปที่ 5.29

-เมื่อทำการสับสวิตช์ที่ 9 ขึ้นด้านบน จะทำให้หุ่นยนต์ยิงลูกเทนนิสไปยังหน้าเน็ตด้านขวาของหุ่นยนต์

ดังรูปที่ 5.30 และรูปที่ 5.31

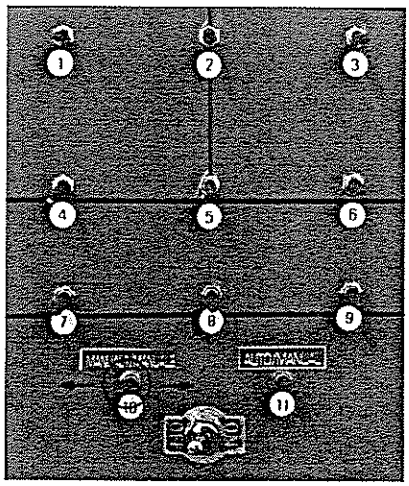


รูปที่ 5.30



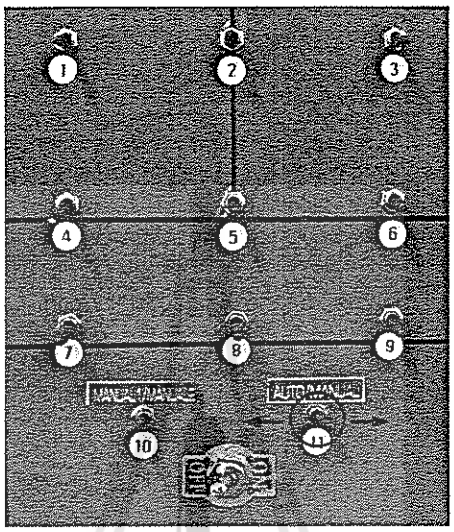
รูปที่ 5.31

-เมื่อทำการสับสวิตช์ที่ 10 ไปทางด้านขวามือ ดังรูปที่ 5.32 จะทำให้หุ่นยนต์เข้าสู่โหมด Manual 2 ที่แสดงดังรูปที่ 5.3 ซึ่งโหมดนี้จะประกอบด้วยสวิตช์ที่ 1- 9 และเมื่อทำการสับสวิตช์ที่ 10 ไปทางด้านซ้ายมือ จะทำให้หุ่นยนต์เข้าสู่โหมด Manual 1 ดังรูปที่ 5.2 ซึ่งโหมดนี้จะทำให้ปรับค่าได้ตามต้องการ ตามที่นำเสนอก่อนหน้านี้



รูปที่ 5.32

- เมื่อทำการสับสวิตช์ที่ 11 ไปทางด้านขวามือ ดังรูปที่ 5.33 จะทำให้หุ่นยนต์เข้าสู่โหมด Manual หรือบังคับด้วยมือ เมื่อทำการเลือกโหมด Manual เสร็จแล้ว จะต้องไปเลือกสวิตช์ที่ 10 ว่าจะใช้ Manual 1/Manual 2 และเมื่อทำการสับสวิตช์ที่ 11 ไปทางด้านซ้ายมือ จะทำให้หุ่นยนต์เข้าสู่โหมด Auto หรืออัตโนมัติ ซึ่งจะทำให้หุ่นยนต์สุ่มตำแหน่งการยิง จากตำแหน่งทั้ง 9 ตำแหน่ง รวมถึงอาจมีลูกไซส์ ตาทที่ได้กล่าวมาแล้ว



รูปที่ 5.33

<u>หมายเหตุ:</u> โหมด Manual หรือโหมดบังคับด้วยมือ จะแบ่งออกเป็น 2 โหมด ดังรูปที่ 5.2 และรูปที่ 5.3 ในกรณีโหมด Manual 2 ตามที่อธิบายมาก่อนหน้านี้ ถ้าทำการ ON สวิตช์ 1- 9 มากกว่า 1 สวิตช์ เช่น ถ้า ON สวิตช์ที่ 1,2,3,5 หุ่นยนต์จะเริ่มยิงตามคำสั่งสวิตช์ที่ 1 เรียง ไปจนถึงคำสั่งสวิตช์ที่ 5 ตามตำแหน่งที่ระบุบนสวิตช์

เอกสารอ้างอิง

[1] มงคล มะการ. ไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino. กรุเทพฯ : บริษัท อีทีที จำกัด, 2552

[2] http://ett.co.th/product2009/ET-AVR/man-ET-EASY-MEGA1280.pdf

[3] http://ett.co.th/product2009/ET-AVR/ET-EASY168_STAMP_man.pdf

มหาวิทยาลัยเทคโนโลยีสุรนารี

ประวัติผู้วิจัย

ดร.กองพล อารีรักษ์ สำเร็จการศึกษาในระดับปริญญาตรี โท และเอก ทางด้านวิศวกรรมไฟฟ้า จากมหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา ในปี พ.ศ. 2543 2546 และ 2550 ตามลำดับ ปัจจุบันดำรงตำแหน่งผู้ช่วยศาสตราจารย์ และหัวหน้าหน่วยวิจัยคุณภาพกำลังไฟฟ้า ประจำสาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี มีความชำนาญทางด้านอิเล็กทรอนิกส์กำลัง วงจรกรองกำลังแอกทีฟ การขับเคลื่อนเครื่องจักรกลไฟฟ้า คุณภาพกำลังไฟฟ้า ระบบควบคุม และการประยุกต์ทางด้านปัญญาประดิษฐ์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี