# หุ่นยนต์ปีนเสา

**Pole Climbing Robot**


นายพรชัย ประทีปพรศักดิ์ รหัส 52362786



ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 20 ก.ค. 2558
เลขทะเบียน ๆ68๐๐๙๓๙
เลขเรียกหนังสือ ปร
มหาวิทยาลัยนเรศวร ๑๗๒๓ ๑ ห ๒๕๕๕


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555

# ใบรับรองปริญญานิพนธ์

| | | | |
|---|---|---|---|
| **หัวข้อโครงงาน** | **หุ่นยนต์ปีนเสา** | | |
| **ผู้ดำเนินโครงงาน** | นายพรชัย | ประทีปพรศักดิ์ | รหัส 52362786 |
| **อาจารย์ที่ปรึกษา** | ดร. พรพิศุทธิ์ | วรจิรันตน์ | |
| **อาจารย์ที่ปรึกษาร่วม** | ดร. ศลิษา | วีรพันธุ์ | |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ | | |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ | | |
| **ปีการศึกษา** | 2555 | | |

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์**ฉบับนี้เป็นส่วนหนึ่ง**ของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....**ที่ปรึกษาโครงงาน**

(ดร. พรพิศุทธิ์ วรจิรันตน์)

.....**ที่ปรึกษาร่วมโครงงาน**

(ดร. ศลิษา วีรพันธุ์)

.....**กรรมการ**

(ดร. พนัส นัถฤทธิ์)

.....**กรรมการ**

(ดร. สุรเดช จิตประไพกุลศาล)

.....**กรรมการ**

(นายเศรษฐา ตั้งค้าวานิช)

| | | | |
|---|---|---|---|
| **หัวข้อโครงงาน** | หุ่นยนต์ปีนเสา | | |
| **ผู้ดำเนินโครงงาน** | นายพรชัย | ประทีปพรศักดิ์ | รหัส 52362786 |
| **อาจารย์ที่ปรึกษา** | ดร. พรพิศุทธิ์ | วรจิรันตน์ | |
| **อาจารย์ที่ปรึกษาร่วม** | ดร. ศลิษา | วีรพันธุ์ | |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ | | |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ | | |
| **ปีการศึกษา** | 2555 | | |


.....

## บทคัดย่อ


การสร้างหุ่นยนต์ปีนเสาแบบกึ่งอัตโนมัติควบคุมผ่านสัญญาณไร้สาย เลียนแบบการเคลื่อนที่ของหนอน โดยมีน้ำหนักรวม 1.4 กิโลกรัม ความกว้าง 18 เซนติเมตร ความยาว 75 เซนติเมตร ความสูงขณะจับยึด 33 เซนติเมตร มีจำนวนข้อหมุนอิสระ (Degree of Freedom) ทั้งหมด 5 ข้อหมุน ทำการทดลองโดยใช้ซิมูลิงค์ (Simulink) จำลองการขับเคลื่อนของหุ่นยนต์และวิเคราะห์การทำงานของระบบร่วมกับบอร์ดไฟโอ (FiO Std) ผ่านโมดูลบล็อกเซ็ตราปิดเอสทีเอ็ม32 (RapidSTM32 Blockset) ในการโปรแกรมเพื่อควบคุมการปล่อยสัญญาณพัลส์วิดมอดูเลชัน (Pulse Width Modulation Signal) ให้เซอร์โวมอเตอร์ (Servo Motor) 5 ตัว เพื่อบังคับการเคลื่อนที่ของหุ่นยนต์โดยใช้เซอร์โวมอเตอร์ 1 ตัวทำหน้าที่ยึดหดลำตัว 2 ตัว ปรับมุมระหว่างแขน และ 2 ตัว สำหรับการจับยึด จากระบบควบคุมระยะไกลผ่านสัญญาณบลูทูธ (Bluetooth) ด้วยโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์ (Android) ซึ่งหุ่นยนต์ที่ได้มีความสามารถในการปีนขึ้นและลงเสาโดยรักษาสมดุลไม่ให้ตกจากเสาได้

| | | | |
|---|---|---|---|
| **Project Title** | Pole Climbing Robot | | |
| **Name** | Mr. Pornchai | Prateeppornsak | ID. 52362786 |
| **Project Advisor** | Dr. Ponpisut | Worajiran | |
| **Co- Project Advisor** | Dr. Salisa | Veerapan | |
| **Major** | Computer Engineering | | |
| **Department** | Electrical and Computer Engineering | | |
| **Academic Year** | 2012 | | |

## Abstract

Creating of Pole Climbing Robot semi-automatic control via a wireless signal imitate the movement of a worm. The total weight 1.4 kg, length 75 cm, width 18 cm, height 33 cm, while clamping. There are 5 degrees of freedom. Use the Simulink to drive model of the robot for experiments. Analyze the system with FiO Board by RapidSTM32 Blockset modules. Use 5 servo motors for all movements of robot and need to input pulse width modulation signal to them. One servo motor use for elastic body, two servo motors use for angles between 2 hands and two servo motors use for gripping. Remote control system via Bluetooth with application on the Android operating system. The robot is capable of climbing up and down a pole and it isn't fall.

# กิตติกรรมประกาศ

โครงงานนี้เกี่ยวกับการสร้างหุ่นยนต์ปีนเสา ผู้จัดทำโครงงานนี้ขอขอบพระคุณ ดร.พรพิศุทธิ์ วรจิรันตน์ อาจารย์ที่ปรึกษา และ ดร. ศลิษา วีรพันธุ์ อาจารย์ที่ปรึกษาร่วม ซึ่งเป็นผู้ให้ความรู้ และแนวทางในการแก้ปัญหาต่าง ๆ ขอขอบพระคุณบิดา มารดา และพี่สาว ที่ให้คำปรึกษา กำลังใจ และงบประมาณในการจัดทำโครงงานนี้

พร้อมกันนี้ใคร่ขอขอบคุณภาควิชาไฟฟ้าและคอมพิวเตอร์ที่ได้อำนวยความสะดวกในการยืมเครื่องมือต่าง ๆ และสถานที่ในการทำงาน ขอขอบคุณคณะอาจารย์ที่ให้ข้อคิดเห็นและคำแนะนำที่เป็นประโยชน์ และขอขอบคุณเพื่อนทุกคนที่คอยช่วยเหลือการทำโครงงานมาโดยตลอด ซึ่งส่งผลทำให้โครงงานนี้สำเร็จลุล่วงไปด้วยดี

ผู้ดำเนินโครงงาน
พรชัย ประทีปพรศักดิ์

# สารบัญ

หน้า

# สารบัญ (ต่อ)

# สารบัญตาราง

# สารบัญรูป

รูปที่ หน้า

# สารบัญรูป (ต่อ)

รูปที่ หน้า

# สารบัญรูป (ต่อ)

รูปที่ หน้า

# สารบัญสัญลักษณ์และอักษรย่อ

| | | |
|---|---|---|
| PWM | = | Pulse-Width Modulation |
| kg | = | กิโลกรัม |
| oz | = | ออนซ์ |
| in | = | นิ้ว |
| cm | = | เซนติเมตร |
| sec | = | วินาที |
| ° | = | องศา |
| V | = | โวลต์ |
| A | = | แอมแปร์ |
| ms | = | มิลลิวินาที |
| mm | = | มิลลิเมตร |

# บทที่ 1
# บทนำ

## 1.1 ที่มาและความสำคัญของโครงงาน

หุ่นยนต์เริ่มเข้ามามีบทบาทกับชีวิตประจำวันของมนุษย์เรื่อยมา เทคโนโลยีได้รับการพัฒนาอย่างต่อเนื่อง การวิจัยและออกแบบหุ่นยนต์มีรูปแบบที่หลากหลาย ทั้งในด้านการแพทย์ งานวิจัย ความมั่นคง งานอุตสาหกรรม บันเทิง และงานครัวเรือน ในงานด้านอุตสาหกรรมมีการพัฒนาระบบหุ่นยนต์ปฏิบัติการที่มีความแม่นยำและสามารถทำงานแทนมนุษย์ได้อย่างมีประสิทธิภาพ เพื่อสามารถตอบสนองต่อความต้องการของอุตสาหกรรมในงานที่อันตรายและมีความเสี่ยงสูง อันอาจก่อให้เกิดการบาดเจ็บ พิการ หรือเสียชีวิตขึ้น เช่น งานติดตั้งและซ่อมอุปกรณ์ต่าง ๆ บนเสา งานตรวจสอบติดตามดูแลสภาพแวดล้อมจากมุมสูง เป็นต้น

หุ่นยนต์ปีนเสาถือเป็นอีกหนึ่งนวัตกรรมที่มีบทบาทอย่างมากในการลดภาวะความเสี่ยงจากการปฏิบัติงานภาคสนาม รูปแบบการเคลื่อนที่ของหุ่นยนต์จึงมีความหลากหลาย เช่น การเคลื่อนที่แบบหมุนเป็นเกลียวของหุ่นยนต์หลายข้อต่อ การเคลื่อนที่แบบล้อ และการเคลื่อนที่แบบจับยึด เป็นต้น และหนึ่งในการเคลื่อนที่ที่น่าสนใจ คือ การเคลื่อนที่แบบหนอน (Inchworm Motion) ซึ่งเป็นการเลียนแบบการเคลื่อนที่ของหนอน มีความยืดหยุ่น สามารถเคลื่อนที่บนเสาได้หลายแบบ เช่น การเคลื่อนที่ในแนวนอนและแนวตั้ง ซึ่งมีข้อดีคือ ลอดหรือข้ามสิ่งกีดขวางได้หลายแบบ ตลอดจนการเคลื่อนที่ขึ้นเสากลมและเสาสี่เหลี่ยมโดยใช้แขนกลจับยึด มีขั้นตอนการทำงานของหุ่นยนต์ไม่ซับซ้อน สะดวกแก่การศึกษา

ดังนั้น โครงงานนี้จึงได้จัดทำหุ่นยนต์ต้นแบบปีนเสาขึ้น เพื่อนำไปพัฒนาใช้งานลดความเสี่ยงของอันตรายอันอาจเกิดขึ้นจากการปฏิบัติงานบนเสาของเจ้าหน้าที่ และศึกษาพฤติกรรมและปัจจัยที่มีผลต่อการเคลื่อนที่ของหุ่นยนต์แบบหลายข้อต่อ ทั้งประยุกต์ใช้ความรู้เชิงวิศวกรรมอิเล็กทรอนิกส์และคอมพิวเตอร์ ในการสร้างหุ่นยนต์ปีนเสาแบบกึ่งอัตโนมัติ

## 1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อสร้างต้นแบบหุ่นยนต์ปีนเสาแบบกึ่งอัตโนมัติ โดยควบคุมผ่านสัญญาณไร้สาย

## 1.3 ขอบเขตของโครงงาน

1. สร้างแบบจำลองโดยใช้ซิมูลิงค์ (Simulink) ซึ่งเป็นโปรแกรมที่มาพร้อมกับแมทแลบ (MATLAB) เพื่อวิเคราะห์และออกแบบระบบการทำงานภายในตัวหุ่นยนต์ปีนเสา
2. สร้างหุ่นยนต์ปีนเสาแบบกึ่งอัตโนมัติโดยใช้หลักการของเซอร์โวมอเตอร์ (Servo Motor)

ในการขับเคลื่อนหุ่นยนต์ด้วยรูปแบบการเคลื่อนที่แบบหนอน และใช้มือจับ (Gripper) สำหรับเกาะและปีนเสา โดยเสามีขนาดเส้นผ่านศูนย์กลางไม่เกิน 8 เซนติเมตร ความสูงไม่เกิน 3 เมตร เคลื่อนที่บนแนวแกนตั้ง

3. สร้างระบบควบคุมการทำงานของหุ่นยนต์ผ่านสัญญาณไร้สาย โดยใช้โปรแกรมประยุกต์การใช้งาน (Application) บนระบบปฏิบัติการแอนดรอยด์ (Android) ลงบนบอร์ด STM32TM ARM 32-bits CortexTM - M3 Processors

## 1.4 แผนการดำเนินงาน

ตารางที่ 1.1 แสดงแผนการดำเนินงาน

| รายการ | พ.ศ. 2555 | | | | | | | พ.ศ. 2556 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | มิ.ย. | ก.ค. | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. | พ.ค. |
| 1. วางแผนการดำเนินโครงงาน | ← | → | | | | | | | | | | |
| 2. ออกแบบโครงสร้างและระบบการทำงานทางกายภาพของหุ่นยนต์ | ← | → | | | | | | | | | | |
| 3. ออกแบบระบบควบคุมและการโปรแกรมของหุ่นยนต์ | | ← | → | | | | | | | | | |
| 4. ส่งแบบฟอร์มและสอบหัวข้อโครงงาน | | ← | → | | | | | | | | | |
| 5. จัดเตรียมอุปกรณ์ เครื่องมือ และสถานที่ | | ← | → | | | | | | | | | |
| 6. ติดต่อหน่วยงานที่เกี่ยวข้องเพื่อของบประมาณสนับสนุนโครงงาน | | ← | → | | | | | | | | | |

| รายการ | พ.ศ. 2555 | | | | | | | พ.ศ. 2556 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | มิ.ย. | ก.ค. | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. | พ.ค. |
| 7. สร้าง ปรับปรุง และทดสอบหุ่นยนต์ตามแบบโครงสร้างทางกายภาพของหุ่นยนต์ | | | | ← | — | — | → | | | | | |
| 8. สร้าง ปรับปรุง และทดสอบโปรแกรมควบคุมระบบการทำงานของหุ่นยนต์ | | | | | ← | — | — | — | — | — | | |
| 9. เก็บข้อมูลการทำงานเชิงสถิติและวิเคราะห์อัลกอริทึมการโปรแกรมของระบบควบคุมหุ่นยนต์ | | | | | | | | ← | — | — | → | |
| 10. สรุปผลการดำเนินโครงการว่าบรรลุตามวัตถุประสงค์หรือไม่ | | | | | | | | | | | | |
| 11. สรุปหาสาเหตุของผลดีและผลเสียของโครงงานเพื่อเป็นแนวทางการพัฒนาต่อไป | | | | | | | | | | | | ← → |
| 12. นำผลสรุปการดำเนินโครงงานไปปรับปรุงแก้ไข | | | | | | | | | | | | ← → |

## 1.5 ผลที่คาดว่าจะได้รับ

1. ต้นแบบจำลองของหุ่นยนต์ปีนเสา

## 1.6 งบประมาณของโครงงาน

1. ชุดไมโครคอนโทรลเลอร์ เป็นเงิน 2,700 บาท

| | | |
|---|---|---|
| 2. ชุดเซอร์โวมอเตอร์ | เป็นเงิน | 4,500 บาท |
| 3. ค่าอะคริลิก | เป็นเงิน | 1,100 บาท |
| 4. ชุดหม้อแปลงไฟฟ้า | เป็นเงิน | 600 บาท |
| 5. ค่าวัสดุสำนักงาน | เป็นเงิน | 500 บาท |
| 6. ค่าถ่ายเอกสาร | เป็นเงิน | 500 บาท |
| 7. ค่าวัสดุอื่น ๆ | เป็นเงิน | 1,000 บาท |

**รวมเป็นเงิน 10,900 บาท (หนึ่งหมื่นเก้าร้อยบาทถ้วน)**

หมายเหตุ ถัวเฉลี่ยทุกรายการ

มหาวิทยาลัยนเรศวร

# บทที่ 2
# หลักการและทฤษฎีที่เกี่ยวข้อง

หุ่นยนต์ในปัจจุบันมีหลากหลายประเภทตามวัตถุประสงค์การใช้งาน ซึ่งมีต้นกำลังที่ทำให้หุ่นยนต์สามารถเคลื่อนไหวได้ เช่น เซอร์โวมอเตอร์และไฮดรอลิก เป็นต้น มีทั้งการเคลื่อนที่ตามแกนเชิงเส้น และแกนเชิงมุม โดยทั่วไปหุ่นยนต์จะต้องประกอบด้วยส่วนต่าง ๆ ดังนี้ [1]

**1. แอคชูเอเตอร์ (Actuator)**

เป็นอุปกรณ์หรือชุดขับเคลื่อนที่เปลี่ยนพลังงานที่ป้อนเข้าให้กลายเป็นแรง เพื่อให้แขนกลหรือหุ่นยนต์เคลื่อนไหวได้ เช่น มอเตอร์ไฟฟ้า กระบอกสูบไฮดรอลิก เป็นต้น

**2. แขนกล (Manipulator)**

เป็นส่วนประกอบที่อาจจะเป็นข้อต่อที่เกี่ยวกับการเคลื่อนที่และทำให้หุ่นยนต์ทำงาน แขนกลจะช่วยในเรื่องของระยะการเคลื่อนที่ซึ่งประกอบด้วยแขน (Arm) และข้อต่อ (Joints)

**3. มือของหุ่นยนต์ (End Effector)**

เป็นส่วนท้ายสุดของแขนกลที่ใช้เพื่อหยิบจับสิ่งของต่าง ๆ แบ่งเป็นสองลักษณะใหญ่ ๆ คือ มือจับ (Gripper) และอุปกรณ์ (Tool as End Effector) เป็นต้น

**4. เซ็นเซอร์ (Sensor)**

เป็นอุปกรณ์ที่ใช้ตรวจจับการเปลี่ยนแปลงต่าง ๆ เช่น แสง แรง ความดัน ระยะทาง สิ่งกีดขวาง ความสูง เป็นต้น แล้วเปลี่ยนแปลงสิ่งที่ตรวจจับได้ให้เป็นข้อมูลหรือสัญญาณที่ต้องการ

**5. แหล่งจ่ายไฟ (Power Supply)**

เป็นแหล่งจ่ายพลังงานให้กับอุปกรณ์ต่าง ๆ ของแขนกล เช่น กระแสไฟฟ้าหรือลมที่ใช้ควบคุมกระบอกสูบ

**6. ตัวควบคุม (Controller)**

เป็นส่วนหนึ่งของหุ่นยนต์ที่ใช้ในการเคลื่อนไหวของระบบเคลื่อนจักรกล ซึ่งรับค่าเข้ามาจากสภาพแวดล้อมในขณะนั้น ๆ ผ่านเซนเซอร์ต่าง ๆ

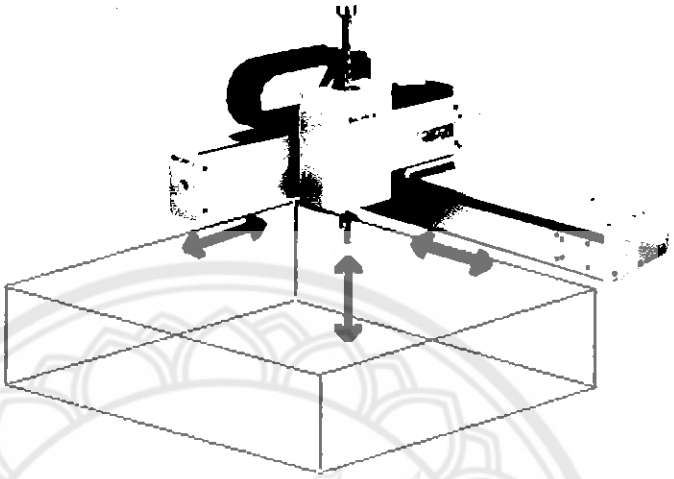
**7. โปรแกรม (Program)**

เป็นการกำหนดขั้นตอนต่าง ๆ ของการทำงานเพื่อให้หุ่นยนต์สามารถปีนเสาได้ตามงานที่ต้องการ

## 2.1 ชนิดของหุ่นยนต์

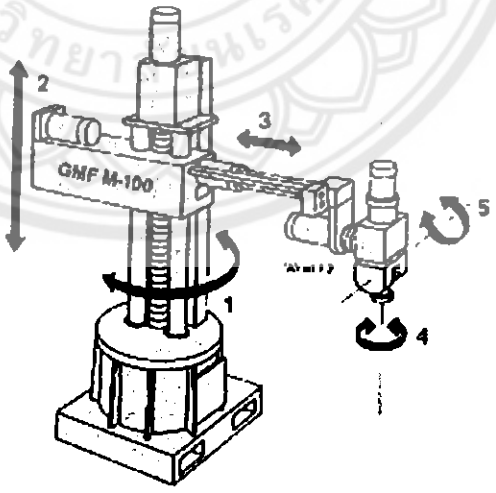
หุ่นยนต์ที่ใช้ในอุตสาหกรรมสามารถแบ่งชนิดของหุ่นยนต์ตามลักษณะรูปทรงของพื้นที่ทำงาน (Envelope Geometric) [2] ดังนี้

**1. The Cartesian Robot** เป็นหุ่นยนต์ที่มีการเคลื่อนที่เป็นเชิงเส้นทั้ง 3 แกน มีลักษณะการทำงานที่ไม่ซับซ้อน แต่มีขีดจำกัดด้านพื้นที่ที่สามารถทำงานของหุ่นยนต์ เพราะอาณาเขตการเคลื่อนที่เป็นแบบกล่องสี่เหลี่ยม เหมาะสำหรับงานลักษณะเคลื่อนย้ายของหนัก ดังรูปที่ 2.1



**รูปที่ 2.1** แสดงลักษณะการเคลื่อนไหวของ The Cartesian Robot [2]

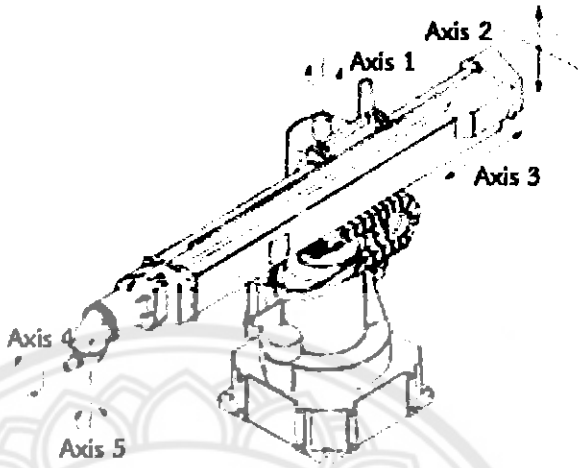
**2. The cylindrical Robot** เป็นหุ่นยนต์ที่มีลักษณะการทำงานแบบทรงกระบอก มีการเคลื่อนที่เป็นแบบเชิงเส้น 2 แกน คือ 2 และ 3 และเคลื่อนที่เชิงมุม 3 แกน คือ 1 4 และ 5 เหมาะสำหรับงานลักษณะจับวาง ดังที่แสดงในรูปที่ 2.2




**รูปที่ 2.2** แสดงลักษณะการเคลื่อนไหวของ The Cylindrical Robot [2]

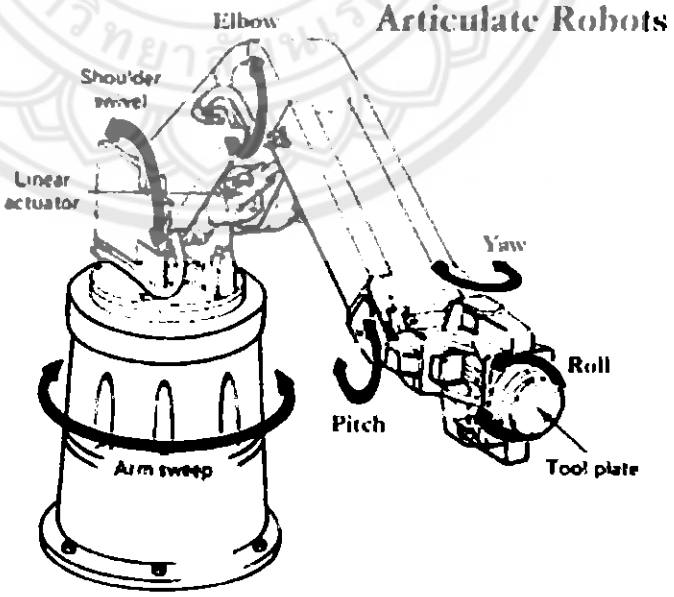
**3. The Spherical Robot** เป็นหุ่นยนต์ที่มีลักษณะการเคลื่อนที่แบบเชิงเส้น 2 แกน และเคลื่อนที่เชิงมุม 3 แกน เหมาะสำหรับงานลักษณะเข้าถึงพื้นที่ที่มีความซับซ้อนในระดับหนึ่ง แต่ไม่

สามารถรับวัตถุที่มีน้ำหนักมากได้ ดังรูปที่ 2.3 โดยมีแกนเชิงเส้น คือ Axis 2 และ Axis 3 และแกนเชิงมุม คือ Axis 1 Axis 4 และ Axis 5



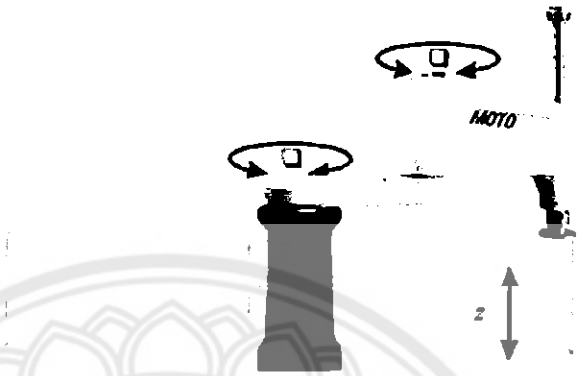

รูปที่ 2.3 แสดงลักษณะการเคลื่อนไหวของ The Spherical Robot [2]

4. **The Articulated Robot** หรือ The Revolute Robot เป็นหุ่นยนต์ที่มีลักษณะการเคลื่อนที่แบบเชิงมุมทั้ง 6 แกน มีพื้นที่การทำงานรอบตัว สามารถเข้าถึงพื้นที่ที่ยากต่อการเข้าถึงได้ สามารถประยุกต์ใช้งานได้มากมาย เช่น เพิ่มแขนและข้อต่อเข้าไป หุ่นยนต์จะสามารถทำงานในจุดที่มนุษย์ไม่สามารถเข้าถึงได้ ดังแสดงในรูปที่ 2.4



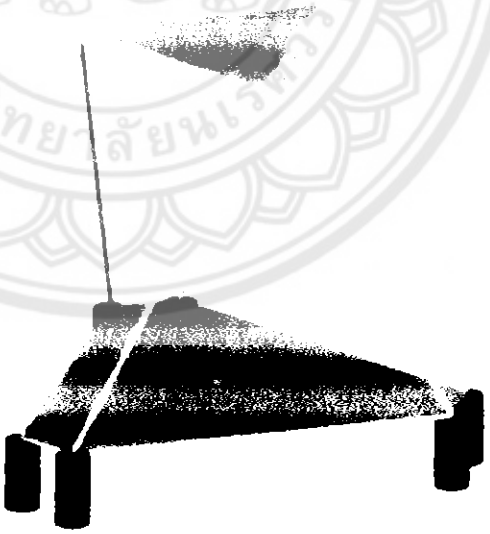

รูปที่ 2.4 แสดงลักษณะการเคลื่อนไหวของ The Articulated Robot [2]

**5. The SCARA Robot** เป็นหุ่นยนต์ที่มีลักษณะการเคลื่อนที่แบบเชิงเส้น 1 แกน และเคลื่อนที่แบบเชิงมุม 2 แกน เหมาะกับการป้อนชิ้นงานตามรางขนส่ง หรือประกอบชิ้นงานที่มีขนาดไม่ใหญ่นัก ดังรูปที่ 2.5 โดยมีแกนเชิงเส้น คือ z และแกนเชิงมุมอีก 2 มุม คือ □




**รูปที่ 2.5** แสดงลักษณะการเคลื่อนไหวของ The SCARA Robot [2]

**6. The parallel Robot** เป็นหุ่นยนต์ที่มีลักษณะเป็นระนาบ 2 ระนาบ ควบคุมโดยขาแต่ละขา ซึ่งหุ่นยนต์ประเภทนี้จะมีน้ำหนักเบา ใช้มอเตอร์ขนาดเล็ก มีความแม่นยำสูง ซึ่งสามารถนำไปประยุกต์เพื่อใช้งานได้หลากหลาย เช่น ใช้ประกอบเครื่องประดับ เป็นต้น ดังแสดงในรูปที่ 2.6



**รูปที่ 2.6** แสดงโครงสร้างของ The Parallel Robot [3]

การเลือกหุ่นยนต์ชนิดต่าง ๆ มาใช้งาน ควรพิจารณาให้เหมาะสมกับงานที่ต้องการให้หุ่นยนต์ทำ ในการออกแบบหุ่นยนต์ปีนเสา จะต้องเลือกประเภทของหุ่นยนต์ที่มีการเคลื่อนที่ที่เอื้อต่อการปีนเสา และต้องสามารถนาพัฒนาเพิ่มเติมให้ได้แบบที่เหมาะสมกับการทำงานได้

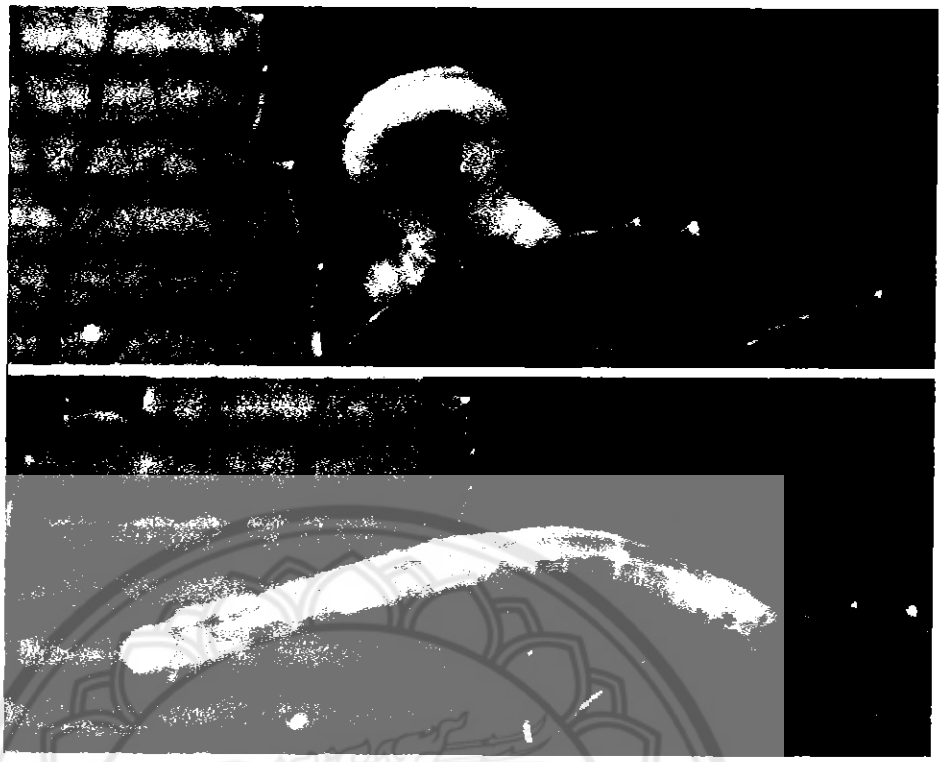
หุ่นยนต์ปีนเสานั้นจัดเป็นหุ่นยนต์ประเภท The Articulated Robot เนื่องจากแกนทั้ง 5 ของเซอร์โวมอเตอร์เป็นแกนที่มีการเคลื่อนที่ตามแกนเชิงมุม ซึ่งลักษณะการเคลื่อนไหวของหุ่นยนต์จะต่อไปอธิบายหัวข้อ 2.2

## 2.2 ลักษณะการเคลื่อนไหวของหุ่นยนต์

การเคลื่อนที่ของหุ่นยนต์โดยหลักแล้วจะพิจารณาออกแบบตามวัตถุประสงค์การใช้ และสภาพการทำงานของหุ่นยนต์เป็นสำคัญ หากหุ่นยนต์นั้นถูกใช้ในโรงงานอุตสาหกรรม ซึ่งงานส่วนใหญ่จะเป็นงานที่ทำในขอบเขตจำกัด การเคลื่อนที่ของหุ่นยนต์จึงไม่มีความจำเป็น ดังนั้นหุ่นยนต์จึงถูกออกแบบให้มีลักษณะเป็นแขนกลชนิดติดตั้งอยู่กับที่ แต่หากการทำงานเป็นไปในเชิงสำรวจตรวจการณ์ หรืองานที่มีขอบเขตการทำงานที่กว้าง จำเป็นที่หุ่นยนต์ต้องสามารถเคลื่อนที่ไปอยู่ในจุดต่าง ๆ ได้ หุ่นยนต์จะถูกออกแบบให้สามารถเคลื่อนที่ได้ ซึ่งการเคลื่อนที่ของหุ่นยนต์มีด้วยกันหลายลักษณะ เช่น การเคลื่อนที่โดยใช้ล้อ (Wheel-drive locomotion) การเคลื่อนที่โดยใช้ล้อสายพาน (Track-drive locomotion) การเคลื่อนที่โดยใช้ขา (Legged locomotion) การเคลื่อนที่โดยการบิน (Flight locomotion) การเคลื่อนที่ในน้ำ (Swimming locomotion) และการเคลื่อนที่ในรูปแบบอื่น (Other locomotion)

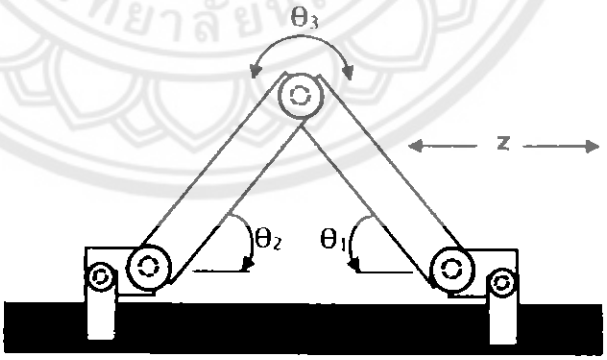
การเคลื่อนที่ของหุ่นยนต์ ต้องคำนึงถึงวิธีการหรือรูปแบบของการเคลื่อนที่ด้วยเหตุผลหลายประการ เช่น เพื่อให้เกิดการใช้พลังงานในการเคลื่อนที่ต่ำที่สุด เพื่อให้หุ่นยนต์มีความสามารถในการหลบหลีกสิ่งกีดขวางได้ หรือเพื่อให้เกิดเสถียรภาพในขณะเคลื่อนที่ เป็นต้น การพัฒนาการเคลื่อนที่ของหุ่นยนต์เลียนแบบธรรมชาติเริ่มมีบทบาทสูงขึ้น เช่น การออกแบบหุ่นยนต์เคลื่อนที่โดยใช้สองขาเหมือนมนุษย์ การออกแบบหุ่นยนต์เคลื่อนที่ใต้น้ำโดยอาศัยหางและครีบที่โบกไปมาเหมือนปลา หรือหุ่นยนต์ที่บินได้โดยอาศัยปีกที่กระพือเหมือนนก จะเห็นได้ว่ากลไกการเคลื่อนที่ของธรรมชาติล้วนอาศัยกลไกการเคลื่อนที่แบบกลับไปกลับมา เนื่องจากกล้ามเนื้อของสิ่งมีชีวิตมีระยะยืดหดที่จำกัด ต่างจากต้นกำลังในหุ่นยนต์ซึ่งส่วนมากจะใช้มอเตอร์เพื่อขับเคลื่อนเป็นหลัก

การเคลื่อนที่ของหนอนเป็นตัวอย่างที่น่าสนใจ เนื่องจากขณะที่หนอนเคลื่อนที่จะใช้ขาหลังยึดเกาะกับวัสดุพื้นผิวแล้วยังยืดตัวออกจนสุดความยาวลำตัว จากนั้นจะใช้ขาหน้าเพื่อยึดเกาะวัสดุพื้นผิวไว้ แล้วจึงหดตัวหรืองอส่วนกลางของร่างกายตามเพื่อดึงส่วนท้ายของลำตัวเคลื่อนมาข้างหน้า ซึ่งการเคลื่อนที่ในลักษณะนี้เกิดจากการทำงานร่วมกันของกล้ามเนื้อ โดยใช้คุณสมบัติการหดตัวและคลายตัวเป็นระลอกคลื่นจากด้านหน้ามาด้านหลัง เพื่อให้เกิดการเคลื่อนที่ไปข้างหน้า ดังรูปที่ 2.7



**รูปที่ 2.7** แสดงการเคลื่อนที่ของหนอน [4]

เมื่อพิจารณาลักษณะการเคลื่อนที่ขึ้นและลงของหุ่นยนต์ที่มีการเคลื่อนที่แบบหนอน จะเห็นว่าการเคลื่อนที่แบบเชิงเส้น 1 แกน และเคลื่อนที่แบบเชิงมุม 3 แกน เมื่อศึกษาขณะที่หุ่นยนต์อยู่ในลักษณะที่เสถียรภาพ (Static Stability) สามารถสรุปลักษณะการเคลื่อนที่ได้ ดังรูปที่ 2.8

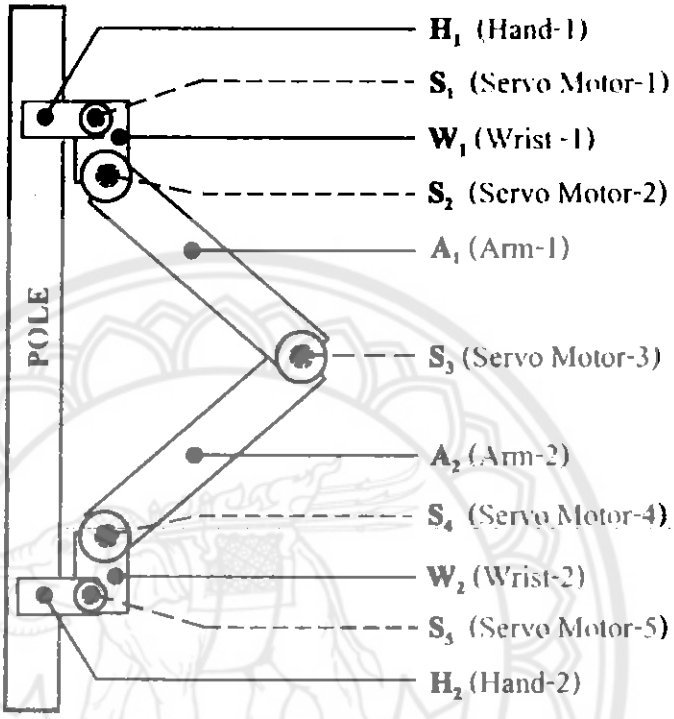



**รูปที่ 2.8** แสดงลักษณะการเคลื่อนที่เชิงเส้นและเชิงมุมของหุ่นยนต์ที่มีการเคลื่อนที่แบบหนอน [5]

### 2.2.1 รูปแบบการเคลื่อนที่ของหุ่นยนต์ปีนเสา

การเคลื่อนที่ขึ้นเสาเป็นการก้าวเดินขึ้น โดยอาศัยมือจับกลทั้งสองข้างเป็นตัวจับยึด และใช้แขนหุ่นทั้งสองเป็นตัวก้าวเดิน ทำให้การก้าวเดินอาศัยหลักก้าวที่ละขา เมื่อเคลื่อนที่จะต้องมี

มือจับมือใดมือหนึ่งปล่อยออกจากเสา ทำให้มือจับที่จับยึดเสาอยู่ต้องรับน้ำหนักมากที่สุด เนื่องจากมีเพียงมือจับล่างหรือมือจับบนเท่านั้นที่สัมผัสกับเสา โดยการเดินขึ้น - ลงลักษณะนี้ สามารถแจกแจงเป็นขั้นตอนได้ดังนี้ ซึ่งได้กำหนดสัญลักษณ์ตำแหน่งของขาไว้ ดังรูปที่ 2.9

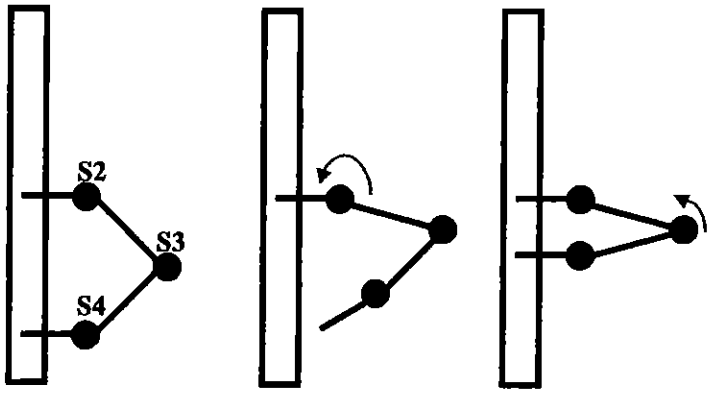



**รูปที่ 2.9** แสดงการกำหนดสัญลักษณ์ของแต่ละตำแหน่งของหุ่นยนต์

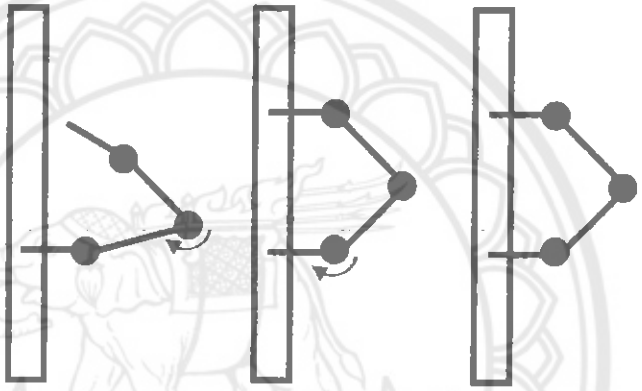
รูปแบบของการเคลื่อนที่ขึ้นเสาในโครงงานนี้จะแยกเป็น 3 รูปแบบ คือ การเคลื่อนที่ทีละข้อต่อ การเคลื่อนที่หลายข้อต่อ และการเคลื่อนที่แบบขนานกับเสา เพื่อทำการทดลองและหารูปแบบการเคลื่อนที่ขึ้นเสาที่ดีที่สุด

2.2.1.1 การเคลื่อนที่ขึ้นเสาทีละข้อต่อ

หุ่นยนต์จะทำการเคลื่อนไหวทีละข้อต่อ ต้องรอข้อต่อที่ทำงานอยู่ ให้ทำงานเสร็จก่อนข้อต่อถัดไปจึงจะทำงาน แรงส่วนมากจะเกิดขึ้นกับเซอร์โวมอเตอร์ที่ใช้ยกตัวหุ่นยนต์ขึ้น เพราะใช้แค่เซอร์โวตัวเดียวในการยก อาจทำให้เซอร์โวมอเตอร์มีความร้อนสูงอย่างรวดเร็ว การเคลื่อนไหวของหุ่นยนต์ขณะทำการปีนขึ้นเสา มีขั้นตอนการทำงาน ดังรูปที่ 2.10




(ก) (ข) (ค)



(ง) (จ) (ฉ)

**รูปที่ 2.10** แสดงลักษณะการเคลื่อนที่ขึ้นเสาของหุ่นยนต์แบบทีละข้อต่อ

ก. หุ่นยนต์อยู่ในสถานะจับยึดเสา ซึ่งเป็นสถานะเริ่มต้น

ข. มือจับ H2 คลายออก จากนั้นปรับข้อหมุน S2 ให้หุ่นยนต์ห่างออกจากเสา เพื่อเพิ่มพื้นที่ไม่ให้มือจับ H2 ติดเสา

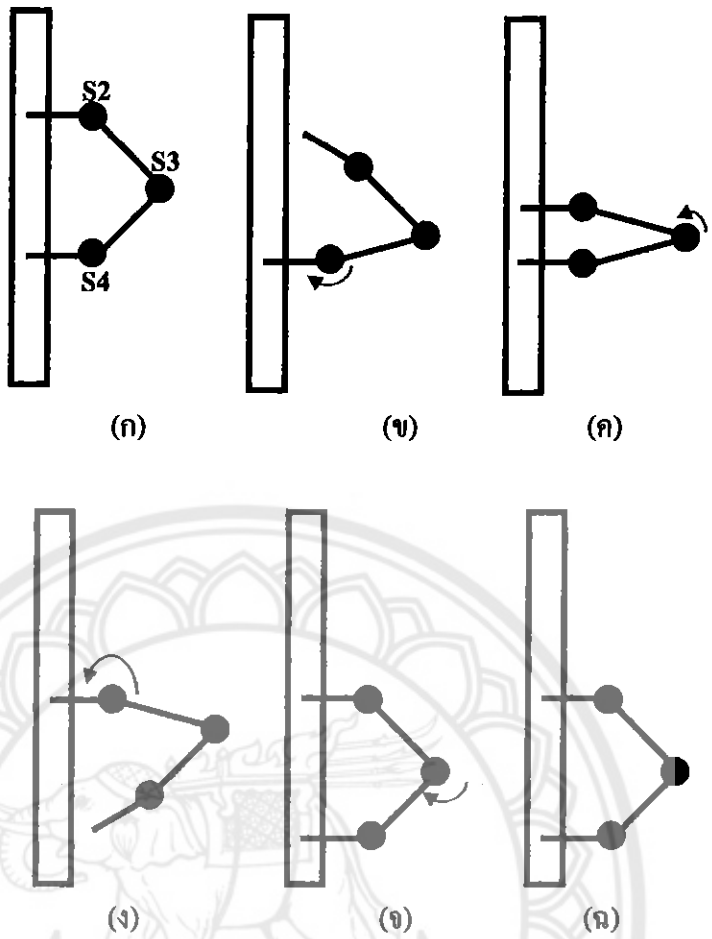
ค. ปรับข้อหมุน S3 ให้หุ่นยนต์หดตัว และให้มือจับ H2 จับเสา

ง. มือจับ H1 คลายออก จากนั้นปรับข้อหมุน S4 ให้หุ่นยนต์ห่างออกจากเสา เพื่อเพิ่มพื้นที่ไม่ให้มือจับ H1 ติดเสา

จ. ปรับข้อหมุน S3 ให้หุ่นยนต์ยืดตัว และให้มือจับ H1 จับเสา

ฉ. หุ่นยนต์อยู่ในสถานะจับยึดเสา ซึ่งเป็นสถานะพร้อมปีนต่อไป โดยทำซ้ำตั้งแต่ขั้นตอน ข-ฉ

และการเคลื่อนที่ลงเสาแบบทีละข้อต่อจะมีขั้นตอนการเคลื่อนไหว ดังรูปที่ 2.11




**รูปที่ 2.11** แสดงลักษณะการเคลื่อนที่ลงเสาของหุ่นยนต์แบบทีละข้อต่อ

ก. หุ่นยนต์อยู่ในสถานะจับยึดเสา ซึ่งเป็นสถานะเริ่มต้น

ข. มือจับ H1 คลายออก จากนั้นปรับข้อหมุน S4 ให้หุ่นยนต์ห่างออกจากเสา เพื่อเพิ่มพื้นที่ไม่ให้มือจับ H2 ติดเสา

ค. ปรับข้อหมุน S3 ให้หุ่นยนต์หดตัว และให้มือจับ H1 จับเสา

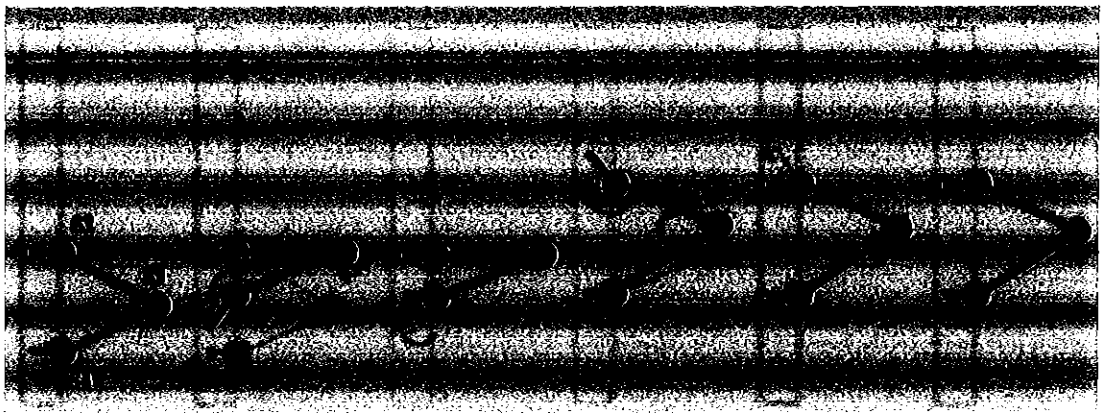
ง. มือจับ H2 คลายออก จากนั้นปรับข้อหมุน S2 ให้หุ่นยนต์ห่างออกจากเสา เพื่อเพิ่มพื้นที่ไม่ให้มือจับ H2 ติดเสา

จ. ปรับข้อหมุน S3 ให้หุ่นยนต์ยืดตัว และให้มือจับ H2 จับเสา

ฉ. หุ่นยนต์อยู่ในสถานะจับยึดเสา ซึ่งเป็นสถานะพร้อมปีนต่อไป โดยทำซ้ำตั้งแต่ขั้นตอน ข-ฉ

**2.2.1.2 การเคลื่อนที่ขึ้นเสาทีละหลายข้อต่อ**

การเคลื่อนที่แบบหลายข้อต่อจะมีความเหมือนกับการเคลื่อนที่ขึ้นเสาแบบทีละข้อต่อ แต่การเคลื่อนไหวของหุ่นยนต์นั้น เซอร์โวมอเตอร์จะทำงานพร้อมกัน ไม่ต้องรอข้อต่อใดข้อต่อหนึ่งทำงานจนเสร็จ ซึ่งการเคลื่อนที่ขึ้นเสาทีละหลายข้อต่อมีลำดับการทำงาน ดังรูปที่ 2.12



(ก) (ข) (ค) (ง) (จ) (ฉ)

**รูปที่ 2.12** แสดงลักษณะการเคลื่อนที่ขึ้นเสาของหุ่นยนต์แบบทีละหลายข้อต่อ

ก. หุ่นยนต์อยู่ในสถานะจับยึดเสา ซึ่งเป็นสถานะเริ่มต้น

ข. มือจับ H2 คลายออก จากนั้นปรับข้อหมุน S2 ให้หุ่นยนต์ห่างออกจากเสา เพื่อเพิ่มพื้นที่ไม่ให้มือจับ H2 ติดเสา ปรับข้อหมุน S3 เพื่อให้หุ่นยนต์หดตัว และปรับข้อหมุน S2 หมุนกลับ ให้มือจับ H2 กลับมาในตำแหน่งที่มือจับ H2 สามารถจับเสาได้

ค. ปรับข้อหมุน S4 จนมือจับ H2 ตั้งฉากกับเสาเพื่อให้มือจับจับยึดเสา

ง. มือจับ H1 คลายออก จากนั้นปรับข้อหมุน S4 ให้หุ่นยนต์ห่างออกจากเสา เพื่อเพิ่มพื้นที่ไม่ให้มือจับ H1 ติดเสา ปรับข้อหมุน S3 เพื่อให้หุ่นยนต์ยืดตัว และปรับข้อหมุน S4 หมุนกลับ ให้มือจับ H1 กลับมาในตำแหน่งที่มือจับ H1 สามารถจับเสาได้

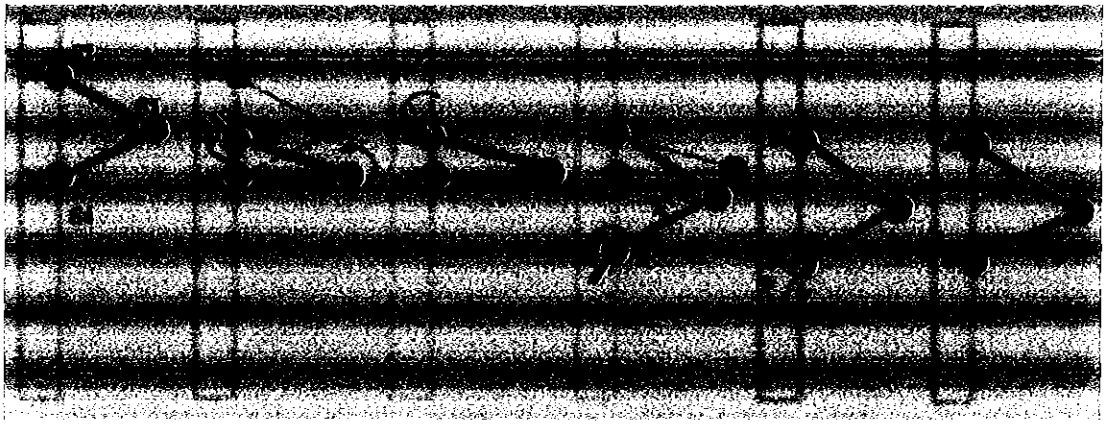
จ. ปรับข้อหมุน S2 จนมือจับ H1 ตั้งฉากกับเสาเพื่อให้มือจับจับยึดเสา

ฉ. หุ่นยนต์อยู่ในสถานะจับยึดเสา ซึ่งเป็นสถานะพร้อมปีนต่อไป โดยทำซ้ำตั้งแต่ขั้นตอน ข. - ฉ.

เมื่อ ------- คือ ตำแหน่งมุมเริ่มต้น (0 องศา) ของหุ่นยนต์ และ

• คือ ตำแหน่งข้อหมุนอิสระของแขนกล

**การเคลื่อนที่ลงเสาของหุ่นยนต์แบบทีละหลายข้อต่อมีลำดับ ดังรูปที่ 2.13**



(ก) (ข) (ค) (ง) (จ) (ฉ)

**รูปที่ 2.13** แสดงลักษณะการเคลื่อนที่ลงเสาของหุ่นยนต์แบบทีละหลายข้อต่อ

ก. หุ่นยนต์อยู่ในสถานะจับยึดเสา ซึ่งเป็นสถานะเริ่มต้น

ข. มือจับ H1 คลายออก จากนั้นปรับข้อหมุน S4 ให้หุ่นยนต์ห่างออกจากเสา เพื่อเพิ่มพื้นที่ไม่ให้มือจับ H1 ติดเสา ปรับข้อหมุน S3 เพื่อให้หุ่นยนต์หดตัว และปรับข้อหมุน S4 หมุนกลับ ให้มือจับ H1 กลับมาในตำแหน่งที่มือจับ H1 สามารถจับเสาได้

ค. ปรับข้อหมุน S2 จนมือจับ H1 ตั้งฉากกับเสาเพื่อให้มือจับจับยึดเสา

ง. มือจับ H2 คลายออก จากนั้นปรับข้อหมุน S2 ให้หุ่นยนต์ห่างออกจากเสา เพื่อเพิ่มพื้นที่ไม่ให้มือจับ H2 ติดเสา ปรับข้อหมุน S3 เพื่อให้หุ่นยนต์ยืดตัว และปรับข้อหมุน S2 หมุนกลับ ให้มือจับ H2 กลับมาในตำแหน่งที่มือจับ H2 สามารถจับเสาได้

จ. ปรับข้อหมุน S4 จนมือจับ H2 ตั้งฉากกับเสาเพื่อให้มือจับจับยึดเสา

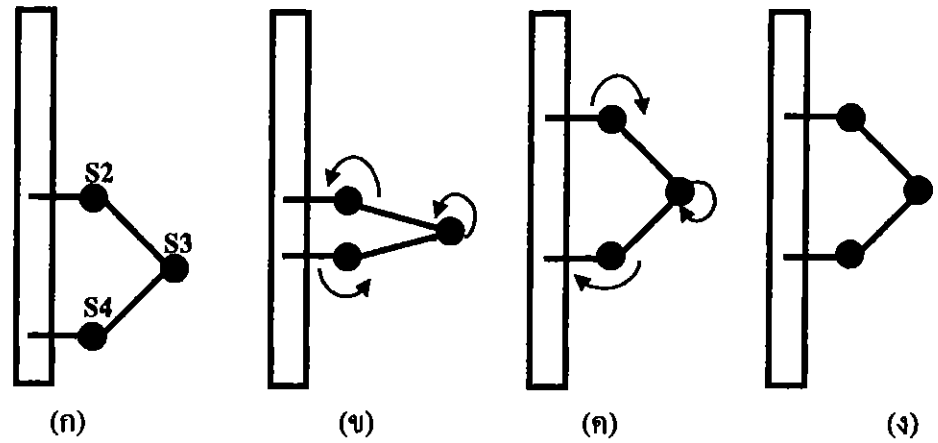
ฉ. หุ่นยนต์อยู่ในสถานะจับยึดเสา ซึ่งเป็นสถานะพร้อมปีนต่อไปโดยทำซ้ำตั้งแต่ขั้นตอนที่ ข. – ฉ.

เมื่อ ------- คือ ตำแหน่งมุมเริ่มต้น (0 องศา) ของหุ่นยนต์ และ

• คือ ตำแหน่งข้อหมุนอิสระของแขนกล

### 2.2.1.3 การเคลื่อนที่ขึ้นเสาแบบขนานกับแนวเสา

รูปแบบการเคลื่อนที่นี้จะแตกต่างออกไปจากสองรูปแบบแรก จะเน้นการทำงานพร้อม ๆ กันข้อแต่ละข้อหมุน และแนวของมือจับจะขนานไปกับเสาตลอด โดยลำดับการเคลื่อนไหวจะเป็นดังรูปที่ 2.14




**รูปที่ 2.14** แสดงลำดับการเคลื่อนที่ขึ้นเสาแบบขนานกับเสา

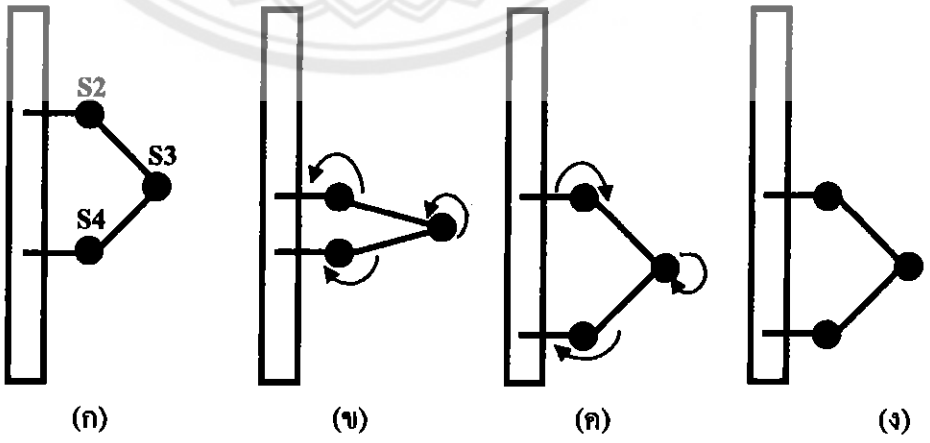
ก. หุ่นยนต์อยู่ในสถานะจับยึดเสา ซึ่งเป็นสถานะเริ่มต้น

ข. มือจับ H2 คลายออก ปรับข้อหมุน S2 ให้ยกตัวหุ่นยนต์ขึ้นมา ปรับ S3 ให้หุ่นยนต์หดตัว และ S4 ให้มือจับ H2 ตั้งฉากกับเสา โดยการปรับ S2 S3 และ S4 จะทำงานไปพร้อม ๆ กัน เพื่อให้ขนานกับเสาตลอดการเคลื่อนที่ และใช้มือจับ H2 จับเสา

ค. มือจับ H1 คลายออก ปรับข้อหมุน S2 ให้มือจับ H1 ตั้งฉากกับเสา ปรับ S3 ให้หุ่นยนต์ยืดตัวออก และปรับ S4 ให้ยกตัวหุ่นยนต์ขึ้น โดยการปรับ S2 S3 และ S4 จะทำงานไปพร้อม ๆ กัน เพื่อให้ขนานกับเสาตลอดการเคลื่อนที่ และใช้มือจับ H1 จับเสา

ง. หุ่นยนต์จะอยู่ในสถานะจับยึดเสา พร้อมทำงานต่อ โดยจะเริ่มตั้งแต่ขั้นตอน ข. – ง.

การเคลื่อนที่ลงเสาแบบขนานกับเสา มีลำดับดังรูปที่ 2.15




**รูปที่ 2.15** แสดงลำดับการเคลื่อนที่ขึ้นเสาแบบขนานกับเสา

ก. หุ่นยนต์อยู่ในสถานะจับยึดเสา ซึ่งเป็นสถานะเริ่มต้น

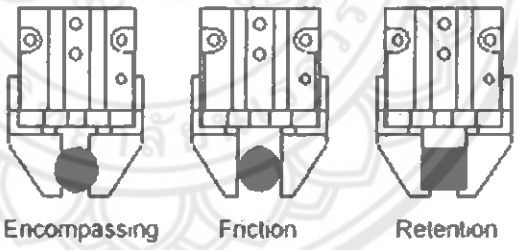
ข. มือจับ H1 คลายออก ปรับข้อหมุน S2 ให้มือจับ H1 ตั้งฉากกับเสา ปรับ S3 ให้หุ่นยนต์ยืดตัวออก และปรับ S4 ให้ยกตัวหุ่นยนต์ลงมา โดยการปรับ S2 S3 และ S4 จะทำงานไปพร้อม ๆ กัน เพื่อให้ขนานกับเสาตลอดการเคลื่อนที่ และใช้มือจับ H1 จับเสา

ค. มือจับ H2 คลายออก ปรับข้อหมุน S2 ให้ยกตัวหุ่นยนต์ลง ปรับ S3 ให้หุ่นยนต์หดตัว และ S4 ให้มือจับ H2 ตั้งฉากกับเสา โดยการปรับ S2 S3 และ S4 จะทำงานไปพร้อม ๆ กัน เพื่อให้ขนานกับเสาตลอดการเคลื่อนที่ และใช้มือจับ H2 จับเสา

ง. หุ่นยนต์จะอยู่ในสถานะจับยึดเสา พร้อมทำงานต่อ โดยจะเริ่มตั้งแต่ขั้นตอน ข. – ง.

### 2.2.2 การจับยึดเสาของมือจับ

มือจับ (Gripper) เป็นอุปกรณ์แขนกลประเภทจับยึดชนิดหนึ่ง ที่ใช้การเสียดสี (Friction) และการโอบล้อมหรือการคงที่ไว้ (Encompassing or Retention) ในการจับยึด โดยอาศัยการโปรแกรมและระบบเฟืองในการควบคุมการทำงาน ซึ่งการเสียดสีเพื่อจับยึดนั้นอาศัยแรงจากการบีบของมือจับเพื่อให้ตัวจับยึดไม่ถูกเคลื่อนย้าย ส่วนการโอบล้อมจะเป็นการโอบอุ้มของตัวจับยึดเพื่อลดผลของการไถล และเพิ่มความมั่นคงด้วยกำลังของการโอบล้อม [6] ดังรูปที่ 2.16

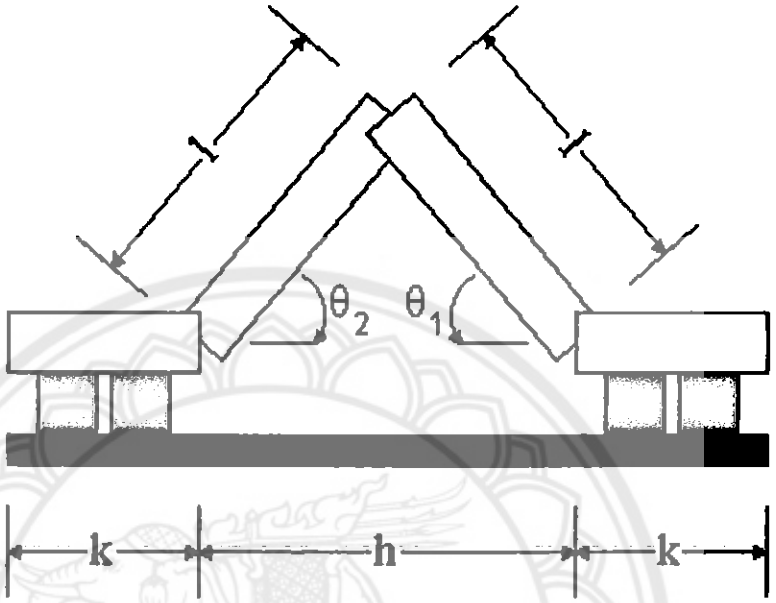



รูปที่ 2.16 แสดงลักษณะการจับยึดของมือจับ [6]

มือจับถือเป็นส่วนสำคัญอย่างมากสำหรับหุ่นยนต์ปีนเสา เพราะเป็นส่วนที่ต้องรับน้ำหนักของหุ่นยนต์ทั้งหมด จากรูปที่ 2.9 จะเห็นว่าเมื่อหุ่นยนต์อยู่ในสถานะหยุดนิ่งมือจับ H1 และ H2 จะจับยึดเสา แรงที่เกิดขึ้นจากน้ำหนักและค่าแรงโน้มถ่วงโลกจะมีผลต่อการจับยึดของมือจับ ในกรณีนี้แรงทั้งหมดถูกจับมือจับกลทั้งสองรับไว้ แต่เมื่อเกิดการเคลื่อนที่ของหุ่นยนต์แรงทั้งหมดจะกระทำต่อมือจับข้างใดข้างหนึ่งเพียงข้างเดียว ทำให้ภาระของมือจับที่ต้องใช้แรงจับยึดกับเสาต้องมีแรงมากพอที่จะตอบรับกับแรงศูนย์ถ่วงและการพยุงตัวของหุ่นยนต์ให้ยังคงอยู่ในสถานะเสถียรภาพตลอดเวลา เพื่อไม่ให้หุ่นยนต์ตกลงจากเสาได้

### 2.2.3 ระยะทางการเคลื่อนที่ของหุ่นยนต์

ระยะทางการเคลื่อนที่ของหุ่นยนต์สามารถคำนวณได้ [7] โดยการกำหนดตัวแปรควบคุม ดังนี้ เมื่อ $k$ คือ ความยาวของมือจับ และ $l$ คือ ความยาวของแขน ดังรูปที่ 2.17




รูปที่ 2.17 แสดงไดอะแกรมหุ่นยนต์ปีนเสาสำหรับกำหนดตัวแปรควบคุม [7]

เมื่อความยาวแขนยืดออกเต็มที่ กำหนดให้ความยาวของหุ่นยนต์มีค่า $2k + h = 50$ เซนติเมตร และเมื่อหดตัวเต็มที่ กำหนดให้ความยาวของหุ่นยนต์มีค่า $2k + h = 30$ เซนติเมตร

เนื่องจากแรงขับเคลื่อนของหุ่นยนต์นี้จะถูกบังคับโดยสมการ

$$kx + (1 - k - 2l)(x + -12h) = 0 \qquad (2.1)$$

เมื่อ $x$ คือ ระยะทางการเคลื่อนที่

$k$ คือ ความยาวของมือจับแขนหุ่นยนต์

และ $l$ คือ ความยาวของแต่ละแขนหุ่นยนต์

$$h = l\cos\Theta_1 + l\cos\Theta_2 \qquad (2.2)$$

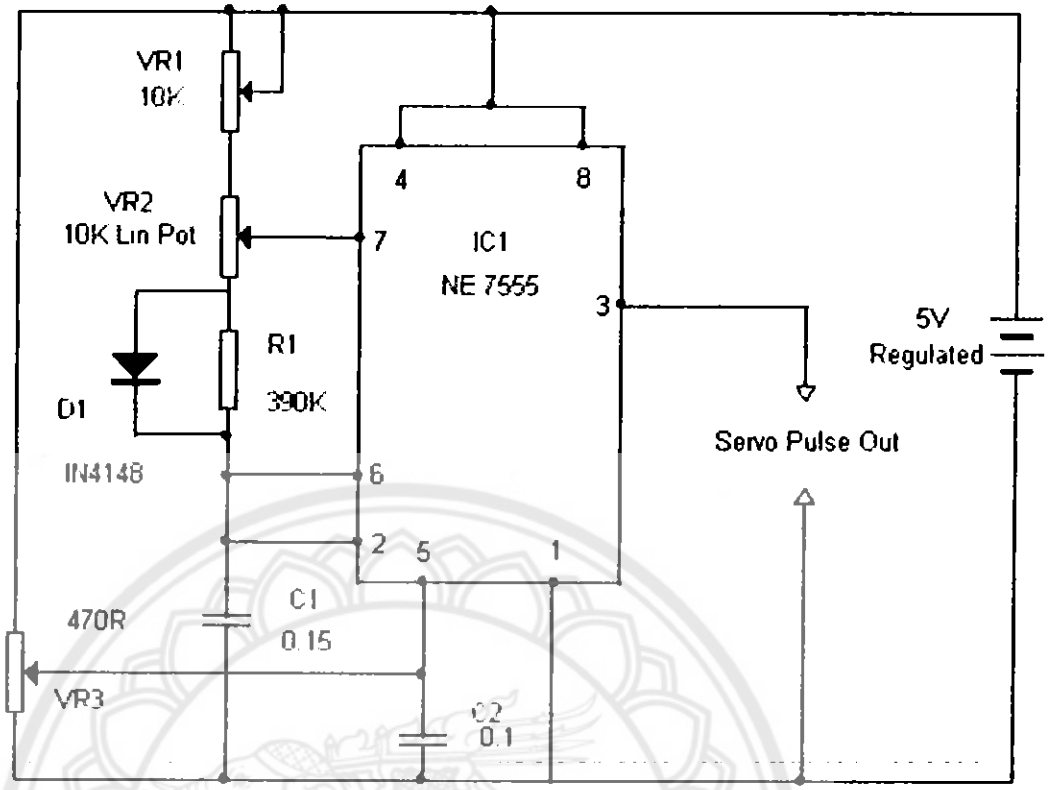
โดย $\Theta_1$ และ $\Theta_2$ หมายถึงมุมระหว่างด้านหน้าจนถึงด้านหลังที่เชื่อมต่อกันของความสัมพันธ์ในการเคลื่อนที่ ดังนั้นแรงขับเคลื่อนของหุ่นยนต์ คือ

$$x + (1 - k - 2l)(x - l\Theta_1 \sin\Theta_1 - l\Theta_2 \sin\Theta_2) = 0 \quad (2.3)$$

## 2.3 แหล่งให้กำลังงานทางกล

การสร้างหุ่นยนต์ส่วนใหญ่จะใช้มอเตอร์ที่มีขนาดเล็ก แรงบิดสูง น้ำหนักเบา และมีวงจรป้อนกลับ เพื่อให้สะดวกต่อการควบคุมและสามารถตอบสนองต่อการเคลื่อนที่ต่าง ๆ ได้อย่างมีประสิทธิภาพ ซึ่งเซอร์โวมอเตอร์ (Servo Motor) นั้นมีความเหมาะสมในการขับเคลื่อนหุ่นยนต์ เนื่องจากมีวงจรป้อนกลับภายใน และมีแรงบิดขนาดต่าง ๆ ให้เลือกใช้ได้ตามความต้องการของผู้ออกแบบหุ่นยนต์ในแต่ละรูปแบบ

เซอร์โวมอเตอร์ คือ มอเตอร์ไฟฟ้ากระแสตรง (DC Motor) ที่ถูกประกอบรวมกับชุดเกียร์และส่วนควบคุมต่าง ๆ ไว้ในโมดูลเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายต่อสำหรับใช้งานเพียง 3 เส้น คือ สายไฟ (Vcc) สายดิน (GND) และสายสัญญาณควบคุม (Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้ายหรือขวาได้ จากสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้เป็นสัญญาณพัลส์วิดโมดูเลชัน (PWM) แบบ TTL Level ระดับแรงดันที่จ่ายให้มอเตอร์นี้จะอยู่ในช่วง 4 – 6 โวลต์ ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว ข้อดีของมอเตอร์ชนิดนี้คือ มีขนาดเล็ก น้ำหนักเบา ให้แรงบิดสูง กินพลังงานน้อย และสามารถควบคุมด้วยแรงดันตรรกะ (Logic) ที่เป็น TTL ได้โดยตรง ไม่จำเป็นต้องต่อกับวงจรขับ (Driver) อื่น ๆ เพราะมอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายใน ดังรูปวงจรที่แสดงในรูปที่ 2.18 ซึ่งสามารถควบคุมให้หมุนในตำแหน่งหรือทิศทางองศาด้วยความกว้างของสัญญาณพัลส์ที่ป้อนให้กับเซอร์โวมอเตอร์ แต่จะหมุนได้เพียง 180 องศา หรือครึ่งรอบเท่านั้น หรือในบางรุ่นอาจหมุนได้ถึง 210 องศา แต่ไม่สามารถหมุนเป็นวงรอบได้ เนื่องจากโครงสร้างภายในประกอบด้วยตัวต้านทานชนิดปรับค่าได้ (VR) ที่ทำหน้าที่ตรวจสอบตำแหน่งการหมุนของเซอร์โวมอเตอร์ และตัวต้านทานนี้ถูกยึดติดกับแกนหมุนของมอเตอร์ เซอร์โวมอเตอร์จึงถูกออกแบบให้หมุนได้เพียง 180 องศาเท่านั้น เพื่อป้องกันความเสียหายที่จะเกิดขึ้นกับตัวต้านทานปรับค่าได้ แต่หากต้องการให้มอเตอร์หมุนเป็นวงรอบหรือหมุนได้ 360 องศา สามารถทำได้โดยการปรับแต่ง (Modify) ดัดแปลงชิ้นส่วนของมอเตอร์เอง




**รูปที่ 2.18** แสดงตัวอย่างวงจรควบคุมภายในเซอร์โวมอเตอร์ [8]

เซอร์โวมอเตอร์ที่เลือกใช้ในโครงงานนี้ พิจารณาจากน้ำหนักของหุ่นยนต์ที่กำหนดไว้ คือ 3 กิโลกรัม เนื่องจากต้องการให้หุ่นยนต์สามารถยกตัวเองขึ้นได้จึงต้องใช้หลักสมดุลของวัตถุแข็งเกร็ง (Equilibrium of a Rigid Body) เข้ามาช่วย ซึ่งสมดุลของโมเมนต์จะเกิดขึ้นเมื่อผลรวมของโมเมนต์จากแรงภายนอกที่กระทำกับวัตถุแข็งเกร็งเท่ากับศูนย์ $\sum M = 0$ [9]

โมเมนต์ คือ ความพยายามที่จะทำให้วัตถุแข็งเกร็งหมุน มีความสัมพันธ์ดังสมการที่ 2.4

$$M = F \times L \tag{2.4}$$

เมื่อ $M$ คือ โมเมนตัม

$F$ คือ แรงภายนอกที่กระทำกับวัตถุแข็งเกร็ง

และ $L$ คือ ระยะห่างจากจุดหมุนไปยังจุดที่แรงกระทำกับวัตถุ

ซึ่งมีความสัมพันธ์กับสมการของแรงบิด ดังสมการที่ 2.5

$$T = F \times L \tag{2.5}$$

เมื่อ $T$ คือ แรงบิด

$F$ คือ แรงภายนอกที่กระทำกับวัตถุแข็งเกร็ง

และ $L$ คือ ระยะห่างจากจุดหมุนไปยังจุดที่แรงกระทำกับวัตถุ

เพราะฉะนั้น แรงบิด $= (3\ kg \times 9.8\ m/sec^2) \times (0.2\ m)$

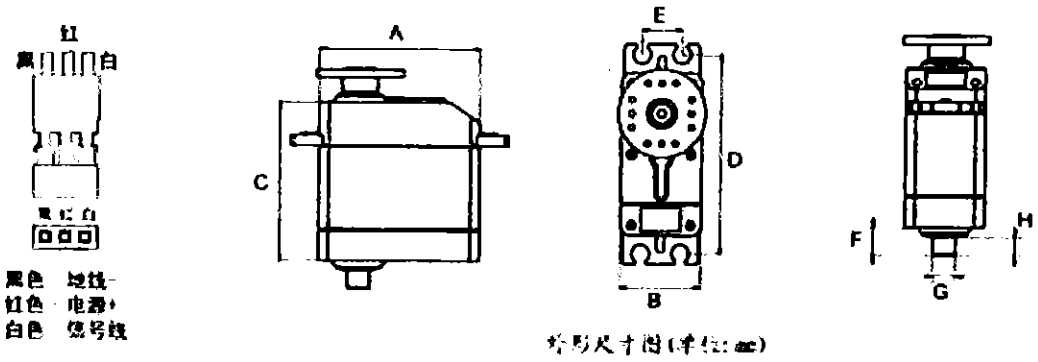
$= 5.88\ kg\text{-}cm$

ดังนั้น แรงบิดของมอเตอร์ควรจะใช้อย่างน้อย ควรจะเป็น 5.88 kg-cm ซึ่งเป็นขนาดของเซอร์โวมอเตอร์ที่เหมาะสมโดยรวมค่าขอบเขตความปลอดภัย (Safety Margin) ในกรณีที่มอเตอร์ต้องรับโหลด (น้ำหนัก) มากกว่าค่าที่กำหนดไว้ โดยเผื่อไว้ที่ 20 เปอร์เซ็นต์ ได้ 7.01 kg-cm จึงเลือกเซอร์โวมอเตอร์เป็น SR431 [10] ซึ่งมีรายละเอียดดังนี้

| | | |
|---|---|---|
| Name | : | SRC SR431 |
| Size | : | 42.0 x 20.5 x 39.5mm / 1.65 x 0.81 x 1.56 in |
| Dimension (W x L x H) | : | 20.5 x 520 x 39.5 mm |
| Speed (sec/60deg) | : | 0.20 (At 6.0 V), 0.18 (At 7.4 V) |
| Torque | : | 12.2 kg-cm (At 6.0 V), 14.5 kg-cm (At 7.4 V) |
| Weight (g/oz) | : | 62/2.18 |
| Control | : | 1.5 ms pulse wide for 0 Degree Position |
| Feature | : | 180 degree robot servo, all metal gears, double ball bearings |
| Purpose | : | robot |

ตารางที่ 2.1 ตารางแสดงขนาดและรายละเอียดของเซอร์โวมอเตอร์ SR431 [10]

| Product Size | | | | | | | | Specification | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Size (mm) | | | | | Weight | | Line | 4.8V | | | 6V | | | Rotational Angle |
| | | | | | | | | Speed | Torque Force | | Speed | Torque Force | | |
| A | B | C | D | E | g | oz | cm | sec / 60° | kg · cm | oz · in | sec / 60° | kg · cm | oz · in | |
| 42.0 | 20.5 | 39.5 | 49.0 | 10.0 | 62 | 2.18 | 30.0 | 0.2 | 12.2 | 169.72 | 0.18 | 14.5 | 201.7 | 180° |




**รูปที่ 2.19** แสดงโครงสร้างของเซอร์โวมอเตอร์ SR431 [10]

**รูปที่ 2.20** แสดงภาพเซอร์โวมอเตอร์ SR431 [10]

### 2.3.1 หลักการทำงานของเซอร์โวมอเตอร์ SR431

การควบคุมการทำงานของเซอร์โวมอเตอร์ SR431 ทั้งทิศทางและตำแหน่งการหมุนทำได้โดยการป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ ซึ่งอ้างอิงสัญญาณพัลส์ที่จ่ายให้เซอร์โวมอเตอร์ทุก ๆ 20 มิลลิวินาที (50 Hz) โดยมีจุดอ้างอิง 3 จุด คือ

1. สัญญาณความกว้างพัลส์ขนาด 1.4 ms ควบคุมเซอร์โวมอเตอร์ให้หมุนไปในตำแหน่งมุม 0 องศาหรือจุดกึ่งกลางของมอเตอร์

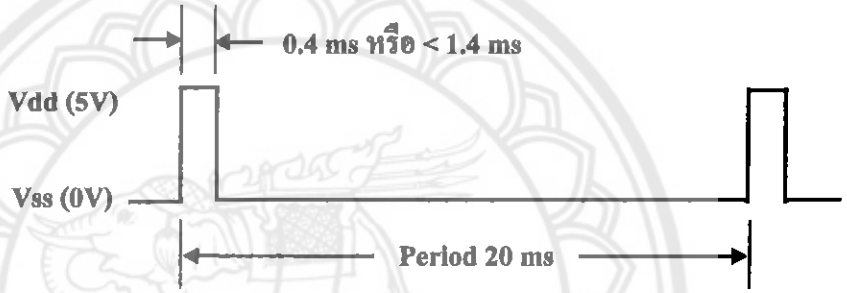
2. สัญญาณความกว้างพัลส์ขนาด 0.4 ms ควบคุมเซอร์โวมอเตอร์ให้หมุนไปในตำแหน่งมุม -90 องศาหรือในทิศทางตามเข็มนาฬิกา

3. สัญญาณความกว้างพัลส์ขนาด 2.4 ms ควบคุมเซอร์โวมอเตอร์ให้หมุนไปในตำแหน่งมุม +90 องศาหรือในทิศทางทวนเข็มนาฬิกา

หากต้องการให้เซอร์โวมอเตอร์มุมไปมุมอื่น สามารถทำได้โดยป้อนสัญญาณพัลส์วิดมอดูเลชันต่าง ๆ จากการอ้างอิงจุดทั้งสามจุด เช่น ต้องการให้เซอร์โวมอเตอร์หมุนไปที่ -45 องศา จะต้องป้อนสัญญาณพัลส์วิดมอดูเลชัน 0.9 ms เป็นต้น

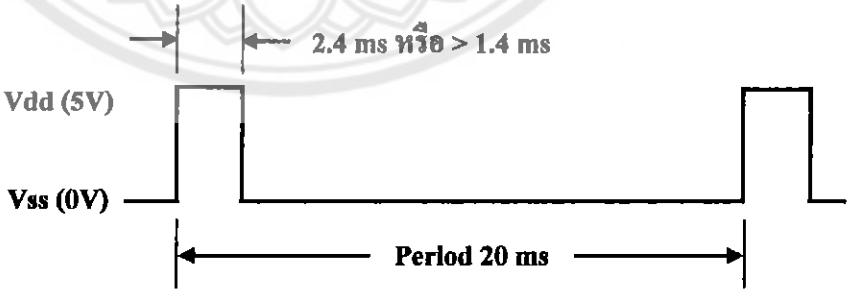
### 2.3.2 การควบคุมเซอร์โวมอเตอร์ SR431

การควบคุมให้เซอร์โวมอเตอร์หมุนทางด้านซ้ายจะต้องป้อนสัญญาณพัลส์ที่มีขนาดความกว้าง 0.4 ms หรือน้อยกว่า 1.4 ms โดยต้องป้อนสัญญาณพัลส์นี้ทุก ๆ 20 ms ดังรูปที่ 2.21



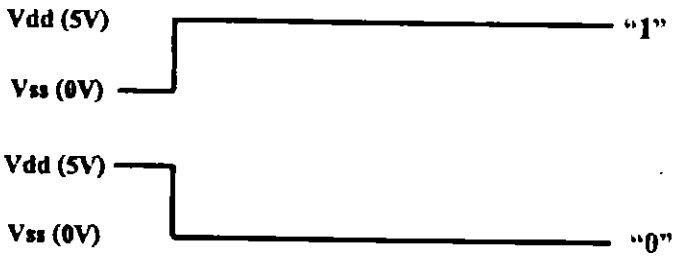

**รูปที่ 2.21** แสดงสัญญาณพัลส์เพื่อควบคุมให้เซอร์โวมอเตอร์หมุนทางด้านซ้าย

การควบคุมให้เซอร์โวมอเตอร์หมุนทางด้านขวาจะต้องป้อนสัญญาณพัลส์ที่มีขนาดความกว้างพัลส์ 2 ms หรือไม่น้อยกว่า 1.5 ms โดยต้องป้อนสัญญาณพัลส์นี้ทุก ๆ 20 ms เช่นกัน ดังรูปที่ 2.22




**รูปที่ 2.22** แสดงสัญญาณพัลส์เพื่อควบคุมให้เซอร์โวมอเตอร์หมุนทางด้านขวา

การควบคุมให้เซอร์โวมอเตอร์หยุดหมุน ทำได้โดยการส่งลอจิก "0" หรือ "1" ให้กับมอเตอร์ซึ่งเป็นการไม่จ่ายสัญญาณพัลส์ให้กับมอเตอร์นั่นเอง ดังรูปที่ 2.23




**รูปที่ 2.23** แสดงสัญญาณสั่งหยุดการหมุนของเซอร์โวมอเตอร์

## 2.4 โครงสร้างของหุ่นยนต์

โครงสร้างของหุ่นยนต์ปีนเสา สามารถแบ่งออกเป็น 2 ส่วนสำคัญ คือ ส่วนลำตัวและแขน (Body and Arm) และส่วนข้อมือ (Wrist) โดยมีระดับขั้นความเสรี (Degree of freedom) เป็นลักษณะการเคลื่อนที่ของหุ่นยนต์ จากงานวิจัย The Optimal Design for Grip Force of Material Handling [11] แสดงให้เห็นว่า โดยมากส่วนลำตัวและแขนจะมีระดับขั้นความเสรี 3 ระดับ และในส่วนข้อมือจะมีระดับขั้นความเสรีอยู่ 2 - 3 ระดับ ที่ปลายของข้อมือจะต่อเข้ากับมือจับ (Gripper) ที่ควบคุมการทำงานด้วยเซอร์โวมอเตอร์ ซึ่งมีความสัมพันธ์กับงานที่หุ่นยนต์ต้องทำ เพื่อจับยึดเสา ส่วนลำตัวและแขนของหุ่นยนต์จะใช้ในการจัดตำแหน่งที่ถูกต้องเพื่อการเคลื่อนที่ตามรูปแบบหนอน และส่วนข้อมือของหุ่นยนต์จะใช้สำหรับจัดทิศทางการวางตัวที่เหมาะสมสำหรับการจับยึดเสา ซึ่งในโครงงานนี้จะใช้เซอร์โวมอเตอร์ (Servo Motor) เพื่อเป็นต้นกำลังไปขับตัวรับกำลังเพื่อขับเคลื่อนหุ่นยนต์ และใช้มือจับสำหรับจับยึดและปีนเสา

## 2.5 มือจับยึด (Gripper)

การยึดเกาะพื้นผิวมีด้วยกันหลายลักษณะ ทั้งการใช้ขนขนาดเล็กจำนวนมากเพื่อยึดเกาะพื้นผิวในที่อยู่บนนิ้วของตุ๊กแก การใช้ข้อต่อของโครงกระดูกเพื่อจับยึดต้นไม้ของลิง เป็นต้น จะเห็นว่าการยึดเกาะพื้นผิวหรือวัสดุที่น่าสนใจสำหรับการศึกษาโครงงานนี้คือการบีบจับและคลายออกของหนอน โดยมีขาจับฝั่งซ้ายและฝั่งขวาสำหรับการบีบเข้าและคลายออก เพื่อเป็นการออกแบบการจับยึดของหุ่นยนต์

รูปที่ 2.24 แสดงการเปรียบเทียบการจับยึดพื้นผิวของหนอนและมือจับยึดที่มีลักษณะคล้ายกัน [12]

### 2.5.1 ความต้องการแรงของมือจับ (Robotic Gripper Force Requirements)

เมื่อพิจารณาการใช้แรงของมือจับต้องคำนึงถึงน้ำหนักของส่วนต่าง ๆ ที่มือจับรับน้ำหนักไว้ตลอดจนค่าแรงโน้มถ่วงและค่าความเร่ง โดยเฉพาะค่าความเร่งนี้ถือว่ามีผลอย่างมากต่อการใช้แรงของมือจับเนื่องจากน้ำหนักของชิ้นส่วนต่าง ๆ ที่มือจับรับน้ำหนักไว้จะมีค่าเพิ่มขึ้นโดยค่าของแรงโน้มถ่วงโลก (Gravity) ดังสมการที่ 2.4 [13]

$$1\ G = 9.8\ Meters/sec^2 \tag{2.6}$$

ดังนั้นแรงยึดเกาะที่มือจับต้องใช้คือ

$$Grip\ Force\ Required = Part\ Weight\ x\ (1 + Part\ G\ side\ acceleration)\ x\ Jaw\ Style\ factor \tag{2.7}$$

แต่ในขณะเดียวกันค่าความเร็วนั้นถือว่าไม่มีผลต่อการพิจารณาในการใช้แรงของมือจับเนื่องจากความเร็วแทบไม่มีผลต่อน้ำหนักที่มือจับต้องรับไว้

### 2.5.2 การควบคุมมือจับ

มือจับสำหรับยึดจับนั้นมีด้วยกันหลายลักษณะขึ้นอยู่กับประเภทการใช้งานของหุ่นยนต์นั้น ๆ โดยพิจารณาถึงลักษณะการเคลื่อนไหลของการจับยึดตามระดับขั้นความเสรีที่ใช้ในการออกแบบหุ่นยนต์ ซึ่ง 1 แอคชูเอเตอร์ (Actuator) จำเป็นต้องมี 1 ระดับขั้นความเสรีเพื่อตอบสนองและปรับเปลี่ยนรูปร่าง ตำแหน่ง หรือลักษณะเฉพาะทางกลอื่น ๆ ตามการโปรแกรมเพื่อการควบคุม ซึ่งโดยทั่วไปมอเตอร์ส่วนใหญ่จะใช้แอคชูเอเตอร์เพื่อทำให้เกิดการเคลื่อนที่แบบมุมหรือวงกลม

ดังนั้นในการทำมือจับที่มีข้อต่อจำนวนมากจึงมีความซับซ้อนในการควบคุมมากตามจำนวนมอเตอร์ที่ต้องใช้

สำหรับตัวจับยึดที่ใช้ในการศึกษาโครงงานนี้เป็นประเภทแขนกลจับยึดสองนิ้ว ซึ่งมีกลไกของตัวยึดด้วยเซอร์โวมอเตอร์ 1 ตัว โดยมีโหลดหรือน้ำหนักของวัตถุกระทำลงมาในแนวตั้ง และถูกต้านด้วยแรงรวมในแนวตั้งฉาก ในกรณีนี้ออกแบบให้ค่าสัมประสิทธิ์ของแรงเสียดสีระหว่างนิ้วและสสารของวัตถุที่จับยึด จากสมดุลแรงทั้งสองชนิดจะพิจารณาได้เป็น

$$W = 2(\mu N) \tag{2.8}$$

หรือ

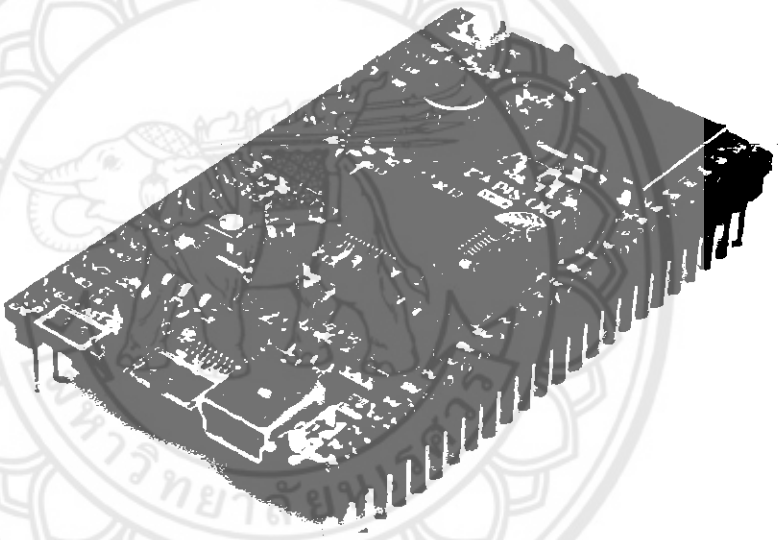
$$N = W/(2\mu) \tag{2.9}$$

หนึ่งในวิธีที่ง่ายที่สุดเพื่อสร้างมือจับ 2 นิ้ว คือ ใช้การประกอบเซอร์โวมอเตอร์ควบคุมการหมุนของเฟืองตรงอันใดอันหนึ่งจากเฟืองตรงทั้งสองอันที่หันเข้าหากัน ซึ่งเป็นตัวควบคุมการบีบเข้าและคลายออกของการจับยึด โดยมีข้อต่อนิ้วเป็นส่วยขยายของการจับยึด ดังรูป 2.25

**รูปที่ 2.25** แสดงการประกอบชุดเฟืองเพื่อควบคุมการจับยึดของมือจับ [14]

## 2.6 บอร์ดไฟโอ (FiO Board)

การควบคุมหุ่นยนต์ปีนเสาอาศัยการควบคุมโดยไมโครคอนโทรลเลอร์ (Microcontroller) ในโครงงานนี้เลือกใช้บอร์ดไฟโอ (FiO Board) ซึ่งเป็นชุดทดลองระบบสมองกลฝังตัวที่ถูกออกแบบมาเพื่อใช้ร่วมกับชุดกล่องคำสั่งราปิดเอสทีเอ็ม32 (RapidSTM32 Blockset) สามารถใช้ร่วมกับซิมูลิงค์ มีจุดเด่นคือการเขียนโปรแกรมแบบกราฟิก (Graphic Programming) โดยอาศัยโค้ดเจเนอเรชัน (Code Generation) ในการแปลงโปรแกรมแบบกราฟิกลงบนบอร์ดไฟโอ และยังสามารถทำ Hardware in the Loop ได้ นั่นคือ สามารถทดสอบอัลกอริทึมและโปรแกรมก่อนที่จะต้องซื้ออุปกรณ์เซ็นเซอร์ (Sensor) และแอคชูเอเตอร์ (Actuator) โดยใช้การจำลองระบบบนโปรแกรม Simulink ได้

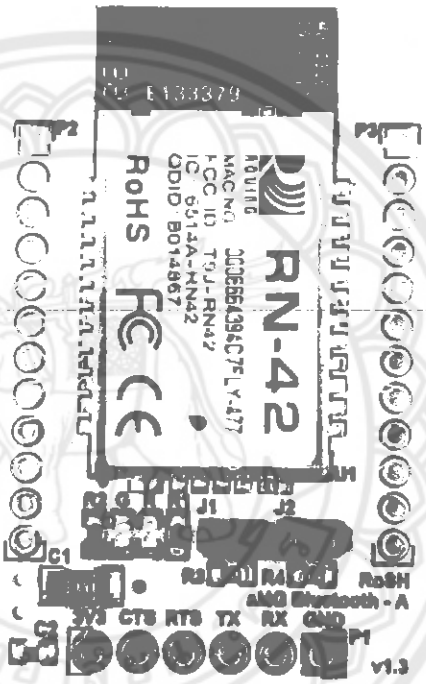


รูปที่ 2.26 แสดงบอร์ด FiO Std [15]

## 2.7 บลูทูธ (Bluetooth) [16]

บลูทูธเป็นเทคโนโลยีของอินเตอร์เฟสทางคลื่นวิทยุ ใช้ในการเชื่อมโยงการสื่อสารไร้สายในแถบความถี่ 2.45 GHz ทำให้อุปกรณ์อิเล็กทรอนิกส์ที่เคลื่อนย้ายได้สามารถติดต่อสื่อสารกันแบบไร้สายระหว่างกันในระยะห่างสั้น ๆ ได้ อุปกรณ์แต่ละตัวสามารถติดต่อสื่อสารกับอุปกรณ์อื่น ๆ ได้สูงสุดถึง 7 ตัวพร้อมกัน ซึ่งเรียกเครือข่ายขนาดย่อมนี้ว่า พิโคเน็ต (Piconet) ยิ่งไปกว่านั้นอุปกรณ์แต่ละตัวยังสามารถสังกัดอยู่กับเครือข่ายพิโคเน็ตได้หลายเครือข่ายพร้อมกันอีกด้วย เทคโนโลยีการส่งคลื่นวิทยุของบลูทูธจะใช้การกระโดดเปลี่ยนความถี่ (Frequency Hop) เพราะว่าเทคโนโลยีนี้เหมาะที่จะใช้กับการส่งคลื่นวิทยุที่มีกำลังส่งต่ำและราคาถูก ช่องสัญญาณ

และแบนด์วิดธ์จะถูกแบ่งระหว่างอุปกรณ์ในพิโดเน็ต โดยจะแบ่งออกเป็นหลายช่องความถี่ขนาดเล็ก ในระหว่างที่มีการเปลี่ยนช่องความถี่ที่ไม่แน่นอนทำให้สามารถหลีกหนีสัญญาณรบกวนที่เข้ามาแทรกแซงได้ บลูทูธจะใช้คลื่นความถี่ในการติดต่อในย่าน ISM (Industrial, Scientific, Medical) ที่มีความถี่ 2.4 GHz และใช้พลังงานต่ำ โดยทางปฏิบัติแล้วอุปกรณ์ของบลูทูธ นั้นจะมีพื้นที่การใช้งานไม่เกิน 10 เมตร การติดต่อผ่านทางช่องสัญญาณที่สนับสนุนทั้งข้อมูลและเสียงที่ความเร็ว 741 Kbps ซึ่งในโครงงานนี้เลือกใช้ RN-42 Bluetooth เป็นตัวเชื่อมโยงการสื่อสารระหว่างโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์ (Android) ของโทรศัพท์กับหุ่นยนต์




รูปที่ 2.27 แสดง aMG Bluetooth – AC2 (with RN-42 Module) [15]

### 2.7.1 โมดูลการทำงานของบลูทูธ (Bluetooth Module)

บลูทูธถูกกำหนดให้มีโครงสร้างการทำงานจำนวน 8 ชั้น ซึ่งมากกว่าโมดูล OSI อยู่ 1 ชั้น ทำให้ขอบเขตการทำงานในแต่ละชั้น แตกต่างจากโมดูล OSI แต่ลำดับการทำงาน มีลักษณะเหมือนกัน โดยแต่ละชั้นของโมดูลบลูทูธมีชื่อและหน้าที่การทำงานดังนี้

**2.7.1.1 ชั้นที่ 1 Radio** เป็นส่วนที่เกิดการรับและส่งคลื่นวิทยุจริง ๆ ในวงจรฮาร์ดแวร์ ภาคส่งรับคลื่นวิทยุที่ถูกควบคุมจากชั้น Base Band ไม่ว่าจะเป็นความถี่และระดับความแรงของสัญญาณที่ใช้ รวมไปถึงเฟรมข้อมูลที่จะส่ง

**2.7.1.2 ชั้นที่ 2 Base Band** การทำงานของชั้นนี้ถือว่าได้เป็นหัวใจของบลูทูธในด้านฮาร์ดแวร์เลยก็ว่าได้ หน้าที่หลักของชั้นนี้ คือ การควบคุมวงจรภาคส่ง – รับคลื่นวิทยุที่อยู่ชั้นล่างสุด ซึ่งจุดสำคัญที่สุดของการควบคุม ก็คือการเลือกช่องความถี่ในการรับส่งข้อมูลให้ตรงกันระหว่าง

มาสเตอร์และสแลฟที่ต้องมีการกระโดดไปในรูปแบบเดียวกัน

**2.7.1.3 ชั้นที่ 3 Link Controller** ควบคุมการเชื่อมต่อพื้นฐานของบลูทูธทั้งหมด ไม่ว่าจะเป็นสถานะของอุปกรณ์ โหมดการทำงานของอุปกรณ์ การค้นหาอุปกรณ์บลูทูธใกล้เคียง รวมไปจนถึงการเลือกว่าจะเป็นมาสเตอร์หรือสแลฟในสถาพแวดล้อมต่าง ๆ

**2.7.1.4 ชั้นที่ 4 Link Manager** ทำหน้าที่แปลงคำสั่งที่ได้รับจากชั้นบนเป็นลำดับหน้าที่การทำงานที่ชั้นล่างรู้จัก และคอยส่งคำสั่งลงไปควบคุมการทำงานของชั้นล่างทั้งหมด

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

**2.7.1.6 ชั้นที่ 6L2CAP (Logical Link Control and Adaptation Protocol)** ทำหน้าที่มัลติเพล็กซ์ข้อมูลจากชั้นบนซึ่งอาจจะมีการทำงานของโปรแกรมหลายโปรแกรมพร้อมกัน และจัดแบ่งข้อมูลออกเป็นแพคเกจ (Package)

**2.7.1.7 ชั้นที่ 7 RFCOMM/SDP** สำหรับ RFCOMM เป็นโปรโตคอลเสมือน ที่ทำให้แอปพลิเคชั่นด้านบนมองบลูทูธ เป็นเหมือนพอร์ตอนุกรม (Serial Port) ทั่วไป ส่วน SDP (Service Discovery Protocol) เป็นโปรโตคอลที่ช่วยค้นหาบริการจากอุปกรณ์ บลูทูธตัวอื่นที่อยู่ในขอบเขตพิโคเน็ตเดียวกัน

**2.7.1.8 ชั้นที่ 8 Application** เป็นส่วนของโปรแกรมที่ติดต่อรับหรือส่งข้อมูลกับผู้ใช้

## 2.8 โปรแกรมประยุกต์สำหรับการโปรแกรม

### 2.8.1 แมทแลบ (MATLAB) [17]

แมทแลบ (MATLAB) เป็นภาษาคอมพิวเตอร์ขั้นสูง (High-Level Language) สำหรับการคำนวณทางเทคนิคที่ประกอบด้วยการคำนวณเชิงตัวเลข กราฟิกที่ซับซ้อน และการจำลองแบบเพื่อให้หมดเห็นภาพพจน์ได้ง่ายและชัดเจน ชื่อของแมทแลบย่อมาจาก Matrix Laboratory ได้พัฒนาด้วยการแก้ไขปัญหาที่ส่งมาจากหลาย ๆ ผู้ใช้เป็นระยะเวลาหลายปีจึงทำให้โปรแกรมแมทแลบมีฟังก์ชันการต่าง ๆ ให้เลือกใช้มากมาย ในบางมหาวิทยาลัยได้ใช้โปรแกรมแมทแลบเป็นหลักสูตรพื้นฐานในการศึกษาทางด้านคณิตศาสตร์ วิศวกรรม และวิทยาศาสตร์แขนงต่าง ๆ ตลอดจนด้านอุตสาหกรรมได้ใช้โปรแกรมแมทแลบเป็นเครื่องมือสำหรับใช้ในงานวิจัย พัฒนาและวิเคราะห์

โปรแกรมแมทแลบจะมีกล่องที่ใช้ในการหาคำตอบเรียกว่า กล่องเครื่องมือ (Toolbox) ซึ่งจะแยกกล่องเครื่องมือออกตามแต่ละสาขา เช่น การประมวลผลสัญญาณ (Signal Processing Toolbox) การประมวลผลภาพ (Image Processing Toolbox) ระบบควบคุม (Control System Toolbox) โครงข่ายประสาท (Neural Networks Toolbox) ฟัซซี่ลอจิก (Fuzzy Logic Toolbox)

เวฟเลท (Wavelet Toolbox) การติดต่อสื่อสาร (Communication Toolbox) สถิติ (Statistics Toolbox) และสาขาอื่น ๆ มากมายภายในกล่องเครื่องมือแต่ละสาขาก็จะมีฟังก์ชันต่าง ๆ ที่เกี่ยวข้องกับการแก้ปัญหาในสาขานั้น ๆ ให้เลือกประยุกต์ใช้งานเป็นจำนวนมาก

### 2.8.2 ซิมูลิงค์ (Simulink)

ซิมูลิงค์ (Simulink) เป็นโปรแกรมเสริมที่มาพร้อมกับแมทแลบ (MATLAB) ซึ่งเป็นระบบที่มีปฏิสัมพันธ์ (Interactive) สำหรับการจำลองและวิเคราะห์ระบบไดนามิคต่าง ๆ ที่เป็นระบบเชิงเส้น (Linear) ระบบไม่เชิงเส้น (Nonlinear) ซิมูลิงค์เป็นโปรแกรมที่สามารถใช้ระบบโมดูลโดยการวาดบล็อกไดอะแกรมบนจอภาพด้วยการใช้เมาส์ ทำให้โปรแกรมแมทแลบสามารถจำลองระบบได้หลายรูปแบบ ในเชิงเส้น (Linear) ไม่เชิงเส้น (Nonlinear) เวลาต่อเนื่อง (Continuous-times) เวลาไม่ต่อเนื่อง (Discrete-time) และระบบหลายอัตรา (Multirate) ซึ่งในแต่ละรูปแบบที่นำมาสร้างแบบจำลองในการวิเคราะห์จะต้องมีความเข้าใจพื้นฐานในการทำงานของบล็อกแต่ละบล็อกได้เป็นอย่างดี ตลอดจนเข้าใจระบบโดยรวมของงานที่จะกระทำด้วย

### 2.8.3 ชุดกล่องคำสั่ง (Blocksets)

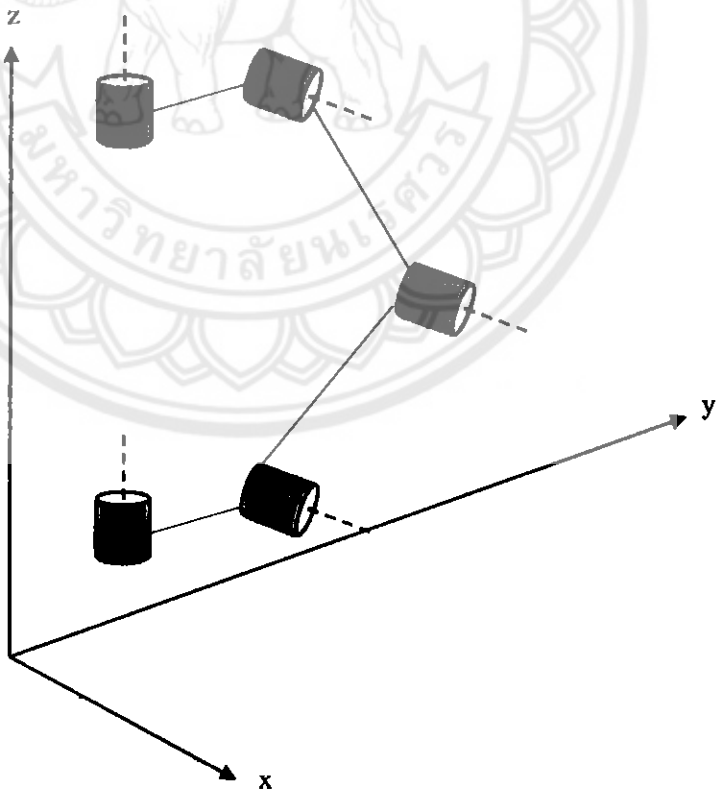
ชุดกล่องคำสั่งเป็นสิ่งที่เพิ่มเติมในซิมูลิงค์ (Simulink) โดยจะเป็นไลบรารีของบล็อกสำหรับการประยุกต์เฉพาะ เช่น การติดต่อสื่อสาร (Communication) การประมวลผลข้อมูล (Signal Processing) และระบบไฟฟ้ากำลัง (Power Systems)

# บทที่ 3
# การออกแบบและสร้างหุ่นยนต์

การออกแบบและสร้างหุ่นยนต์ปีนเสาแบบกึ่งอัตโนมัตินั้น จำเป็นต้องพิจารณาถึงน้ำหนัก ความสูง จำนวนข้อต่อ กำลังขับเคลื่อน และระบบควบคุมมอเตอร์ ซึ่งสิ่งเหล่านี้มีผลต่อความสามารถ ความเสถียร ตลอดจนความคล่องตัวในการเคลื่อนไหวของหุ่นยนต์

## 3.1 การออกแบบหุ่นยนต์เบื้องต้น

การออกแบบหุ่นยนต์ปีนเสา ลำดับแรกคือการออกแบบการวางข้อต่อบนโครงสร้างของหุ่นยนต์ที่มีการติดตั้งเซอร์โวมอเตอร์ (Servo Motor) ในที่ต่าง ๆ เพื่อให้ได้คุณสมบัติใกล้เคียงกับลักษณะการก้าวเดินแบบหนอน ซึ่งมีการจับยึดเสา หกตัว และคลายออก ดังรูปที่ 3.1 โดยกำหนดให้แกนหมุน x มีหน้าที่เป็นแกนที่ทำให้หุ่นยนต์เอียงตัวไปด้านบนหรือด้านล่าง และแกนหมุน z มีหน้าที่เป็นแกนที่ทำให้หุ่นยนต์บิดหรือคลายการจับยึดเสาของหุ่นยนต์



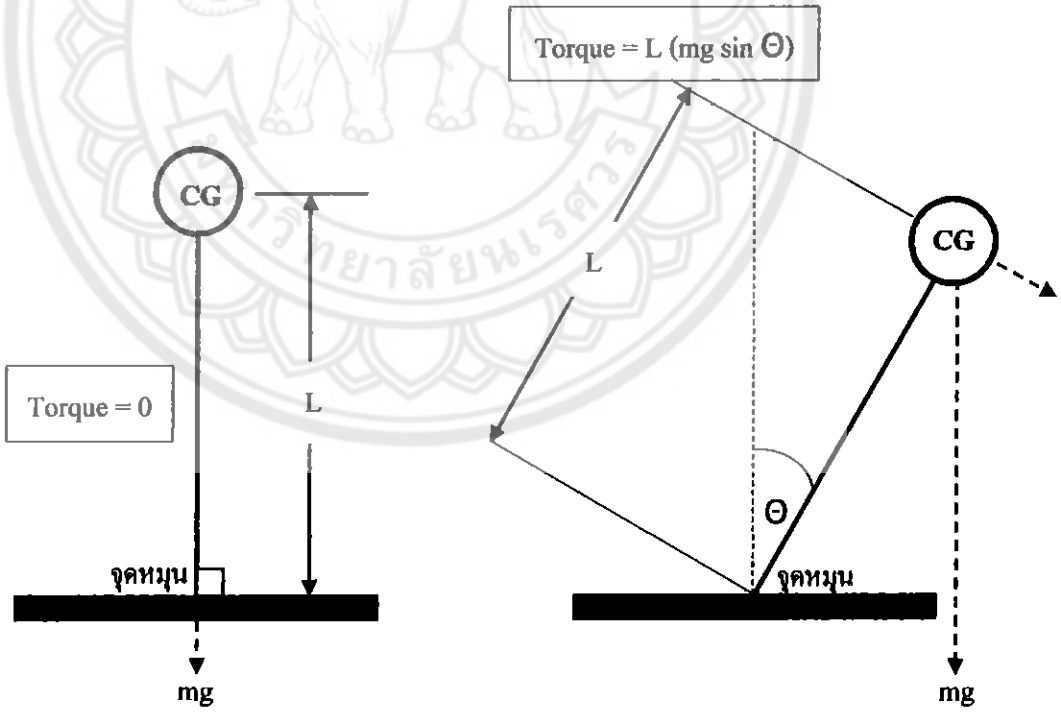

**รูปที่ 3.1** แสดง Wire Diagram ของหุ่นยนต์

โครงสร้างของหุ่นยนต์ที่ออกแบบเป็นโครงสร้างที่มีทิศการหมุนข้อต่อ 5 ข้อต่อ โดยสามารถจัดให้เกิดท่าทางการจับยึด การปีนขึ้น การปีนลง เป็นต้น ในขั้นตอนนี้การออกแบบโครงสร้างหุ่นยนต์ถือว่าบรรลุจุดประสงค์เพียงพอแล้ว เนื่องจากยังมีขั้นตอนที่จะออกแบบอย่างละเอียดในโปรแกรมออกแบบจำลองสามมิติ โซลิดเวิร์ค (SolidWorks)

## 3.2 การเลือกเซอร์โวมอเตอร์สำหรับหุ่นยนต์

เซอร์โวมอเตอร์ คือ มอเตอร์ที่มีระบบควบคุมอยู่ภายใน ทำให้สามารถควบคุมตำแหน่งของมอเตอร์ได้โดยใช้สัญญาความกว้างของพัลส์ (Pulse Width Modulation) ซึ่งสะดวกในการใช้งานเพื่อเป็นส่วนส่งแรงไปขับเคลื่อนข้อต่อของหุ่นยนต์ โดยมีสิ่งที่สำคัญประการหนึ่งที่มีผลต่อความสามารถในการแสดงท่าทางของหุ่นยนต์คือแรงบิด (Torque) ของเซอร์โวมอเตอร์

แรงบิด (Torque) คือ ปริมาณที่บ่งบอกถึงความสามารถในการรองรับแรงที่พยายามหมุนซึ่งแรงนั้นมีทิศทางตั้งฉากกับเวกเตอร์เชื่อมโยงจากจุดหมุนไปยังแรงนั้น ๆ ดังนั้นในการเลือกใช้เซอร์โวมอเตอร์ที่มีแรงบิดต่าง ๆ จึงทำให้หุ่นยนต์มีความสามารถในการแสดงท่าทางที่แตกต่างกัน ดังรูปที่ 3.2




(ก) เมื่อหุ่นยนต์ตั้งตรง (ข) เมื่อหุ่นยนต์เอียงตัว

**รูปที่ 3.2** แสดงลักษณะของจุดหมุนและแนวแรงที่ก่อให้เกิดแรงบิดที่แตกต่างกันในท่าทางของหุ่นยนต์

จากรูปจะได้สูตรของการหาแรงบิดที่เกิดขึ้น [18] คือ

$$Torque = L\ (mg \sin \Theta) \qquad (3.1)$$

โดย $L$ คือ ระยะจากจุด CG (Center of Gravity) ถึงจุดหมุนของบริเวณข้อเท้าของหุ่นยนต์

$m$ คือ มวลของหุ่นยนต์ (kg)

$g$ คือ ความเร่งเนื่องจากแรงโน้มถ่วงของโลก (9.8 m/sec$^2$)

$\Theta$ คือ มุมเอียงจากแกนตั้งฉากกับพื้นโลก

จากรูปที่ 3.2 (ก) จะเห็นว่าเมื่อหุ่นยนต์ตั้งตรง แรงบิดที่เกิดขึ้นจะมีค่าเท่ากับศูนย์เนื่องจากสมการที่ 3.1 sin (0) = 0 และเมื่อหุ่นยนต์เอียงตัวไปจะทำให้เกิดแรงบิดต่าง ๆ กันตามขนาดมุมที่เอียงไปจากแนวตั้งฉากกับพื้นโลก

ความสัมพันธ์ของแรงบิดเป็นไปตามสมการที่ 3.1 คือ Torque = L (mg sin $\Theta$) และเมื่อกำหนดค่าโดยประมาณของน้ำหนักหุ่นยนต์คือ 2 กิโลกรัม และระยะห่างระหว่างจุด CG (Center of Gravity) กับจุดหมุนของบริเวณข้อมือจับของหุ่นยนต์คือ 20 เซนติเมตร จากกราฟจะเห็นว่าขนาดของแรงบิดเพิ่มขึ้นตามขนาดของมุมเอียงจากแนวตั้งฉากกับพื้นโลกของหุ่นยนต์ โดยแรงบิดต่ำสุดคือ 0 กิโลกรัม-เซนติเมตร เมื่อหุ่นยนต์ตั้งตรง และแรงบิดสูงสุดคือ 40 กิโลกรัม-เซนติเมตร เมื่อหุ่นยนต์เอียงตัวตั้งฉากขนานกับพื้นโลก ดังนั้นในการเลือกเซอร์โวมอเตอร์ที่ดีควรต้องให้มีแรงบิดเข้าใกล้ 40 กิโลกรัม-เซนติเมตร แต่จากงบประมาณของโครงงานที่มีจำกัดทำให้สามารถจัดหาเซอร์โวมอเตอร์ได้ขนาดของแรงบิดคือ 12.2 กิโลกรัม-เซนติเมตร ซึ่งเมื่อคำนวณแล้วจะพบว่าหุ่นยนต์มีความสามารถในการเอียงตัวคือ

$$\sin^{-1}\left(\frac{\text{Torque}}{\text{L mg sin}\,\theta}\right) = \sin^{-1}\left(\frac{1.22}{(0.2)(2)(10)(1)}\right) = 17.58 \text{ องศา}$$

## 3.3 การกำหนดรายละเอียดของหุ่นยนต์

การกำหนดรายละเอียดของหุ่นยนต์ที่จะต้องคำนึงถึง เนื่องจากมีผลต่อการออกแบบแบบจำลองสามมิติ การสร้างโครงสร้าง และการควบคุมท่าทางของหุ่นยนต์ โดยรายละเอียดแสดงดังตารางที่ 3.1 ซึ่งได้กำหนดขนาดของหุ่นยนต์ปีนเสาให้มีความสูงไม่เกิน 40 เซนติเมตร ความกว้างไม่เกิน 20 เซนติเมตร และใช้เซอร์โวมอเตอร์ที่มีขนาดแรงบิด 12.2 กิโลกรัม-เซนติเมตร ที่แรงดันไฟฟ้า 6 โวลต์ เนื่องจากเป็นเซอร์โวมอเตอร์ที่มีขนาดรูปร่างที่เหมาะสมและแรงบิดสูงเพียงพอแล้ว โครงสร้างของหุ่นยนต์ที่ออกแบบจึงเลือกใช้อะคริลิก (Acrylic) แทนอลูมิเนียม ซึ่งมี

ความแข็งแรงพอสมควร แต่ก็มีความเปราะเช่นกันจึงเป็นจุดอ่อนในการใช้งานที่มีการสั่นสะเทือนสูง อย่างไรก็ตามหุ่นยนต์ปีนเสานี้เป็นหุ่นยนต์ขนาดเล็ก ผลกระทบจากแรงสะเทือนต่อวัสดุจึงมีน้อยมาก แหล่งจ่ายพลังงานที่เลือกใช้คือกระแสไฟฟ้าสลับความต่างศักย์ 220 โวลต์ ผ่านหม้อแปลงไฟฟ้า (Adapter) ให้เป็นกระแสไฟฟ้าตรง (Direct Current) ขนาด 3 แอมแปร์ ความต่างศักย์ 6 โวลต์ และวงจรควบคุมการทำงานของหุ่นยนต์ที่ใช้เป็นบอร์ดสำเร็จรูปคือ บอร์ดไฟโอ (FiO Board) ซึ่งใช้ STM32TM ARM 32-bits CortexTM – M3 Processors ผ่าน Simulink ซึ่งถูกติดตั้งพร้อมกับแมทแลบ (MATLAB) และการทำ Hardware in the Loop กล่าวคือ สามารถทดสอบอัลกอริทึมและโปรแกรมก่อนที่จะต้องซื้ออุปกรณ์เซนเซอร์ (Sensor) และแอคชูเอเตอร์ (Actuator) โดยผ่านการ Simulate เซนเซอร์และแอคชูเอเตอร์ ในโปรแกรมซิมูลิงค์ได้

ตารางที่ 3.1 แสดงรายละเอียดของหุ่นยนต์ปีนเสา

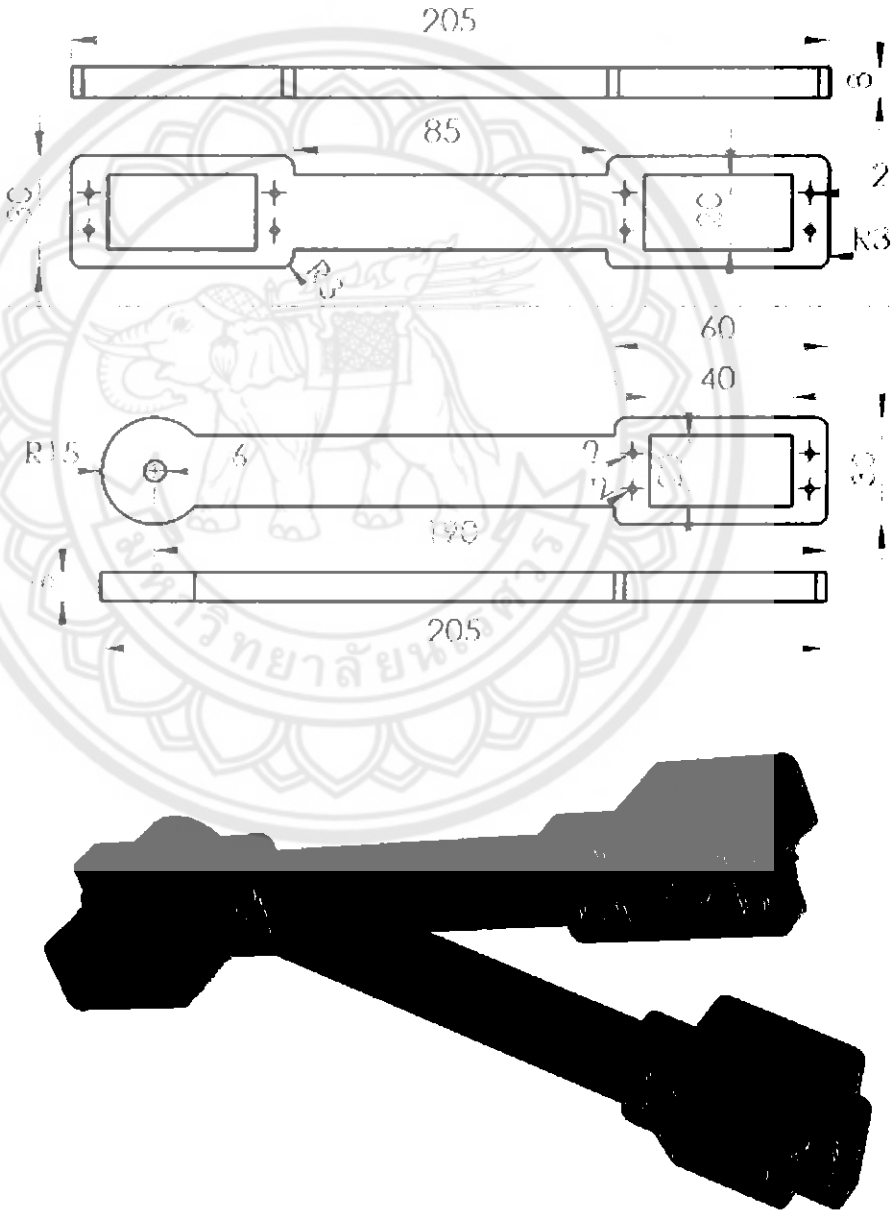
| ข้อมูล | รายละเอียด |
|---|---|
| ชื่อ | Electricity Pole Climbing Robot |
| จำนวนมือจับและข้อหมุนอิสระ | 2 มือจับ 5 ข้อหมุน |
| ขนาด | ความกว้าง 20 เซนติเมตร ความยาว 80 เซนติเมตร ความสูงขณะจับยึด 40 เซนติเมตร |
| น้ำหนักสูงสุด | ประมาณ 2 กิโลกรัม |
| โครงสร้าง | อะคริลิก |
| ตัวขับแต่ละข้อหมุนอิสระ | เซอร์โวมอเตอร์ |
| ความต้องการพลังงาน | ไฟฟ้ากระแสตรง 6 โวลต์ 2 แอมแปร์ |
| แหล่งพลังงาน | ไฟฟ้ากระสลับที่ผ่านหม้อแปลงไฟฟ้า (Adapter) |
| บอร์ดควบคุม | FiO Board (STM32TM – ARM 32-bits CortexTM – M3) |
| โหมดการทำงาน | ใช้โปรแกรมบนระบบปฏิบัติการแอนดรอยด์ควบคุมผ่านสัญญาณบลูทูธให้หุ่นยนต์ทำงานแบบกึ่งอัตโนมัติ |

### 3.4 การออกแบบจำลองสามมิติ

การออกแบบจำลองสามมิติบนโปรแกรมโซลิดเวิร์ค (SolidWorks) มีจุดประสงค์เพื่อออกแบบโครงสร้างหุ่นยนต์ไปใช้สร้างชิ้นงานจริงโดยเริ่มจากการกำหนดวัสดุและอุปกรณ์ที่ใช้ให้แน่นอนคือขนาดเซอร์โวมอเตอร์รุ่น SRC SR431 Metal Gear 180 deg. Rotation Servo และโครงสร้างที่ใช้คือแผ่นอะคริลิกขนาดความหนา 0.8 เซนติเมตร และ 0.5 เซนติเมตร จากนั้นทำการสร้างแบบจำลองสามมิติของเซอร์โวมอเตอร์และชิ้นส่วนของอะคริลิกก่อน เมื่อสร้างแบบจำลอง

### 3.4.2 แบบจำลองแขนของหุ่นยนต์

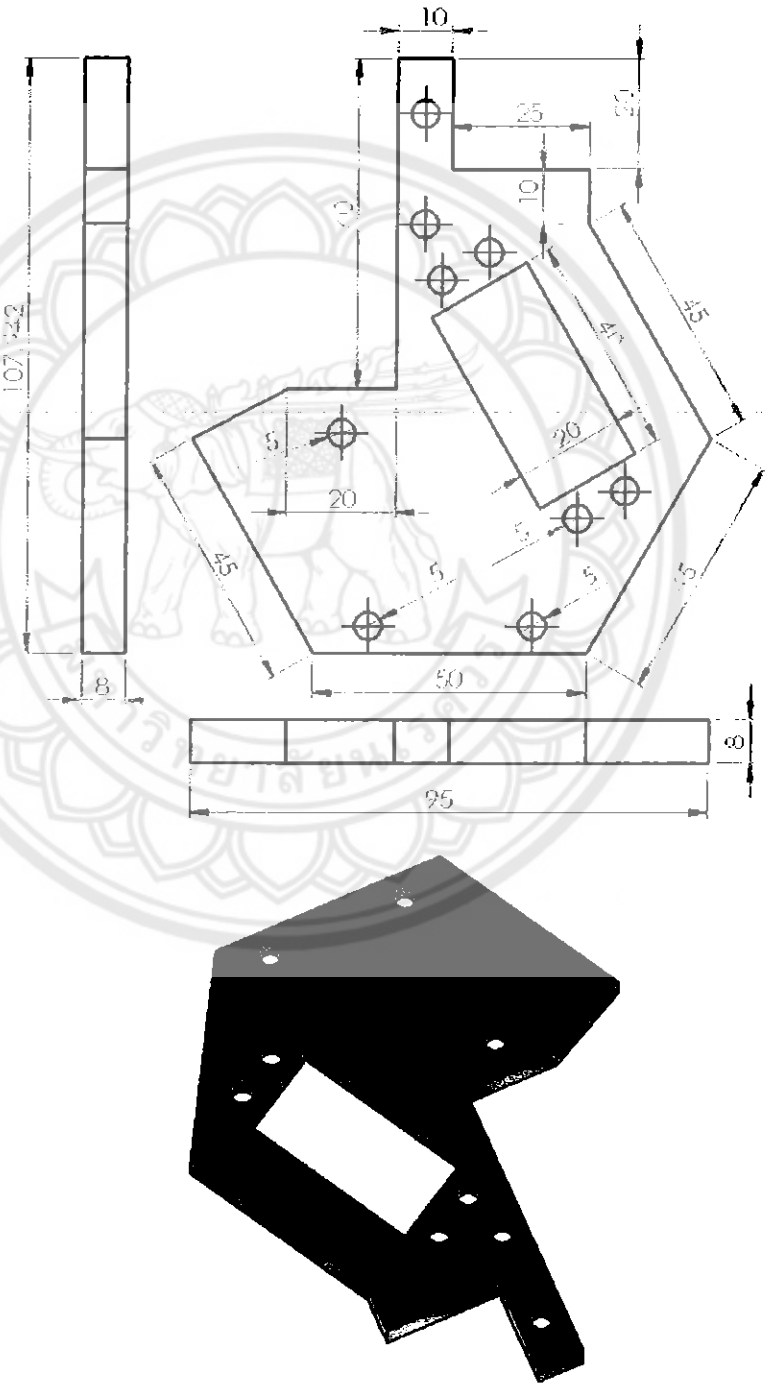
แขนของหุ่นยนต์เป็นส่วนที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ และเพื่อให้มีลักษณะใกล้เคียงกับการเคลื่อนที่แบบหนอน การออกแบบของของหุ่นยนต์จึงออกแบบให้มีแขนสองท่อนต่อเข้ากับข้อต่อซึ่งเป็นเซอร์โวมอเตอร์เพื่อควบคุมการกวาดมุมขึ้นและลง โดยเจาะช่องสำหรับยึดเซอร์โวมอเตอร์ไว้ ส่วนปลายทั้งสองของแขนแต่ละข้างจะใส่เซอร์โวมอเตอร์ไว้ โดยให้ตัวบนใส่เซอร์โวมอเตอร์พ้นจากแขนขึ้นมาเพื่อให้มีแนวการวางเฟืองของมอเตอร์ที่ต่อกับแขนล่าง ซึ่งใช้สำหรับต่อเข้ากับข้อมือของหุ่นยนต์ดังรูปที่ 3.4 (หน่วยทั้งหมดเป็นมิลลิเมตร)



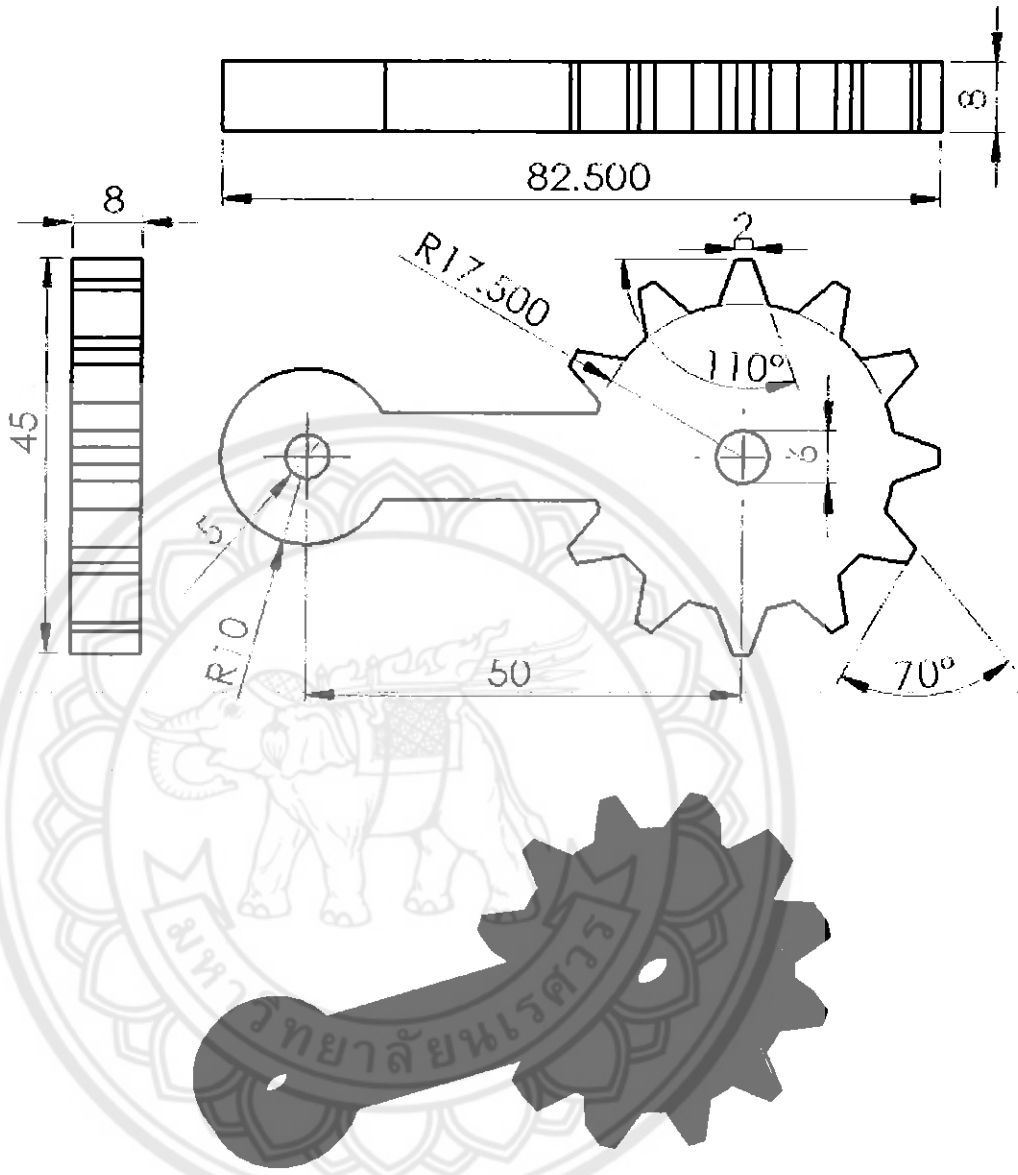

**รูปที่ 3.4** แสดงขนาดและแบบจำลองสามมิติของแขนหุ่นยนต์

### 3.4.3 แบบจำลองมือจับของหุ่นยนต์

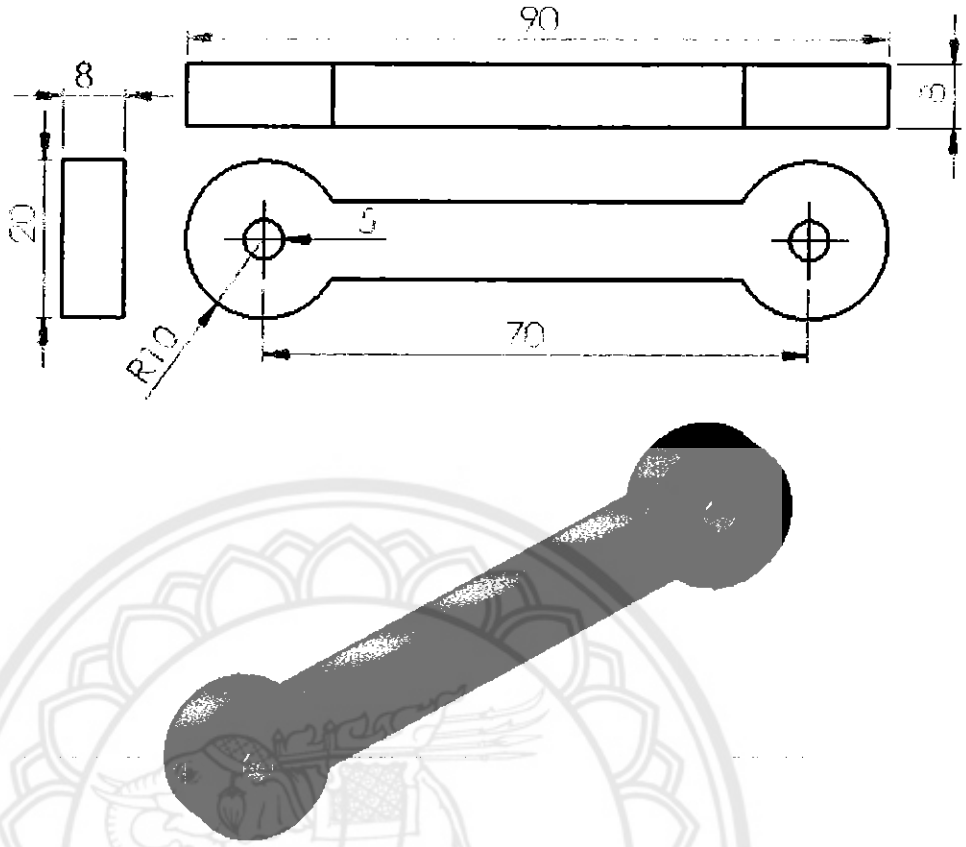
มือจับของหุ่นยนต์ประกอบด้วยชิ้นส่วนย่อยหลายส่วน ซึ่งมีความสำคัญมากเนื่องจากต้องรับแรงจากน้ำหนักของหุ่นยนต์ทั้งตัว และต้องมีแรงบีบที่กระทำต่อเสามากพอจะทำให้หุ่นยนต์ไม่ตกลงจากเสา ขนาดและแบบจำลองสามมิติของชิ้นส่วนต่าง ๆ มีดังนี้ (หน่วยทั้งหมดเป็นมิลลิเมตร)



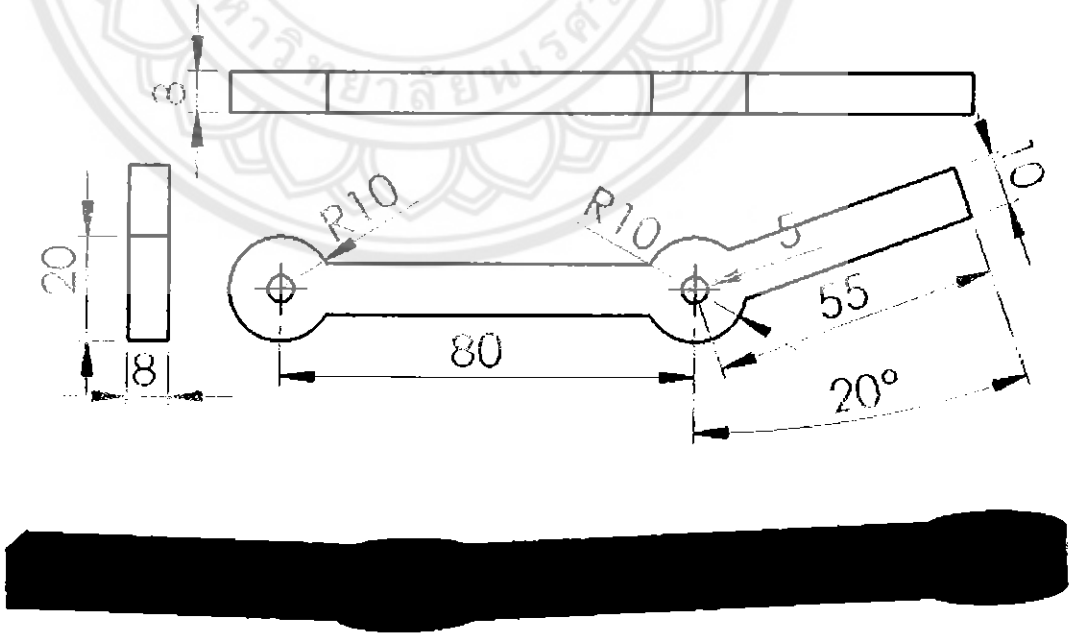

**รูปที่ 3.5** แสดงขนาดและแบบจำลองสามมิติฝ่ามือของหุ่นยนต์



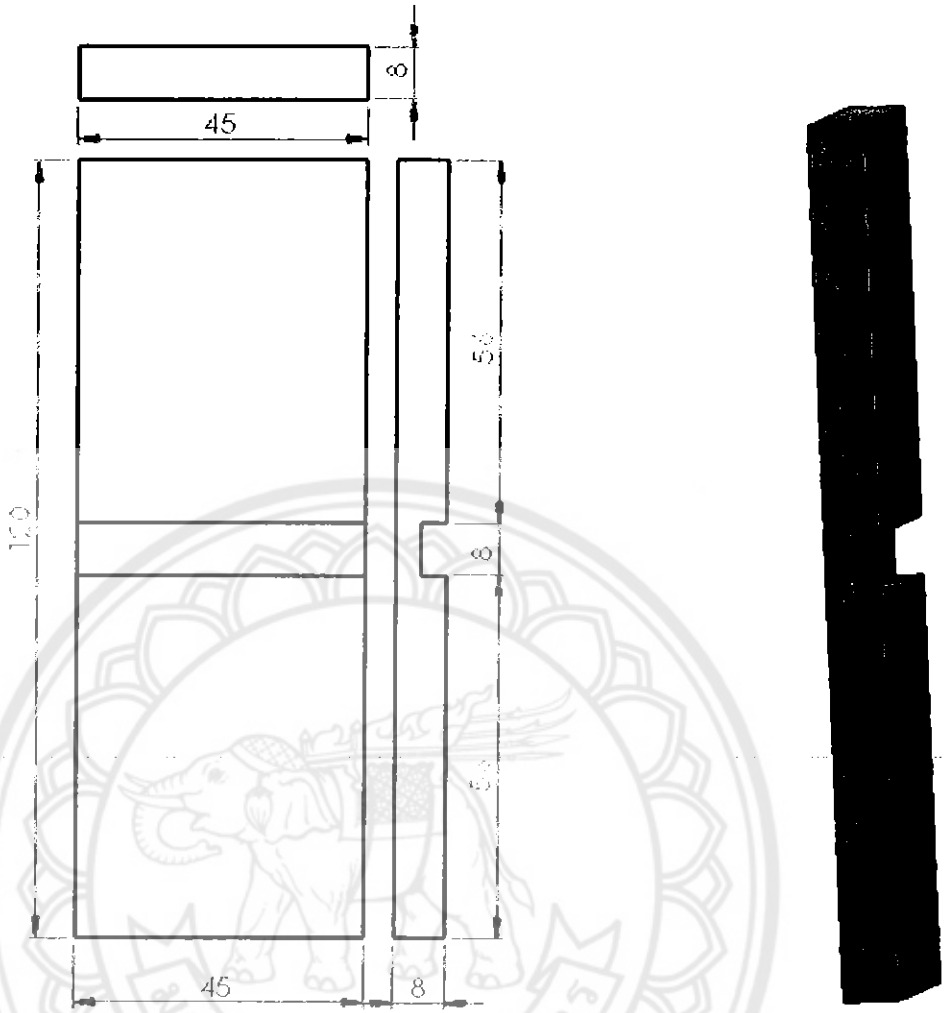

**รูปที่ 3.6** แสดงขนาดและแบบจำลองสามมิติของเฟืองที่ใช้เชื่อมต่อกับเซอร์โวมอเตอร์เพื่อการบีบและคลายของมือจับ




**รูปที่ 3.7** แสดงขนาดและแบบจำลองสามมิติของแท่งอะคริลิกที่ใช้ยึดนิ้วมือ**โดยจะขนานกับเฟือง**ในรูปที่ 3.7

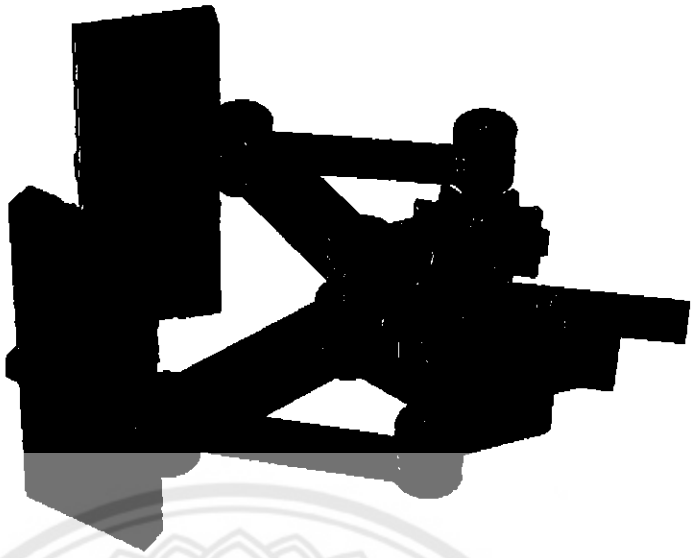



**รูปที่ 3.8** แสดงขนาดและแบบจำลองสามมิติของนิ้วมือที่เชื่อมระหว่างเฟืองในรูปที่ 3.7 และแท่งอะคริลิกในรูปที่ 3.8




**รูปที่ 3.9** แสดงขนาดและแบบจำลองสามมิติของแผ่นอะคริลิกที่ติดกับนิ้ว

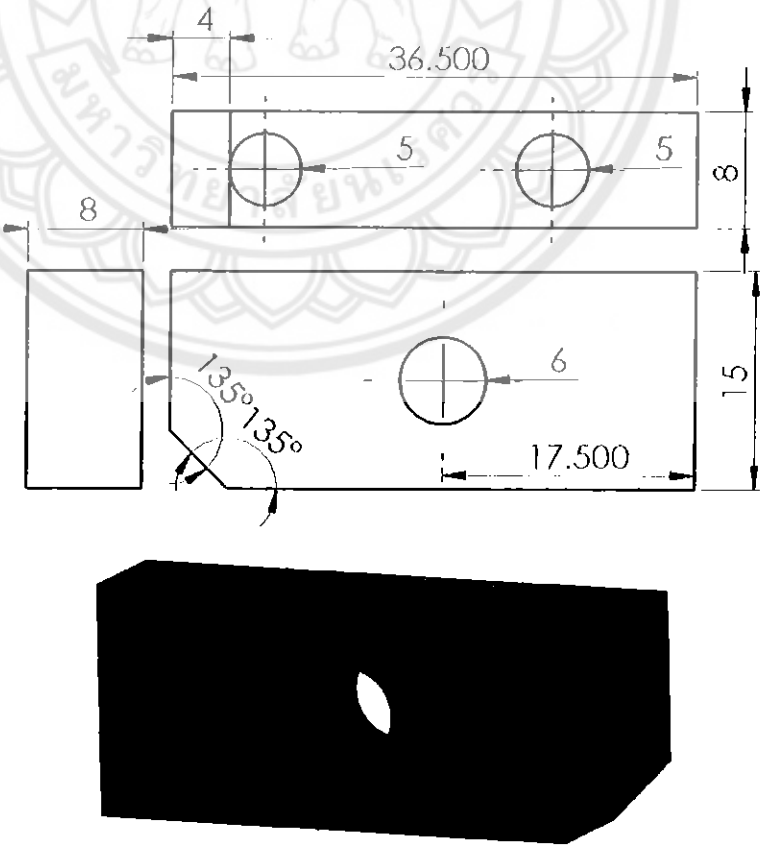
มือจับหนึ่งชิ้นจะมีชิ้นส่วนต่าง ๆ คือ ฝ่ามือ 1 ชิ้น เฟือง 2 ชิ้น แท่งอะคริลิก 4 แท่ง นิ้วมือ 2 นิ้ว และแผ่นอะคริลิก 2 แผ่น ซึ่งนำมาประกอบรวมกันโดยใส่เซอร์โวมอเตอร์เข้าไปในช่องสี่เหลี่ยมของฝ่ามือที่แสดงให้เห็นใน นำเฟืองด้านที่มีฟันเฟืองสวมลงไปในแกนของเซอร์โวมอเตอร์ ส่วนเฟืองอีกตัวนำมาประกบให้เฟืองขับกัน ส่วนแท่งอะคริลิกให้ใส่ในรูด้านหน้าของมือจับทั้งสองรู ประกบด้านบนและด้านล่างฝ่ามือ นำนิ้วมือทั้งสองนิ้วใส่ด้านขวาและด้านซ้ายนิ้วละด้าน หันส่วนที่แหล่มออกจากตัวฝ่ามือ และสุดท้าย นำแผ่นอะคริลิกติดกับปลายนิ้วทั้งสอง เมื่อทำตามขั้นตอนทั้งหมดแล้วจะได้มือจับดังรูปที่ 3.11



**รูปที่ 3.10** แสดงแบบจำลองสามมิติของมือจับที่ประกอบชิ้นส่วนทั้งหมด

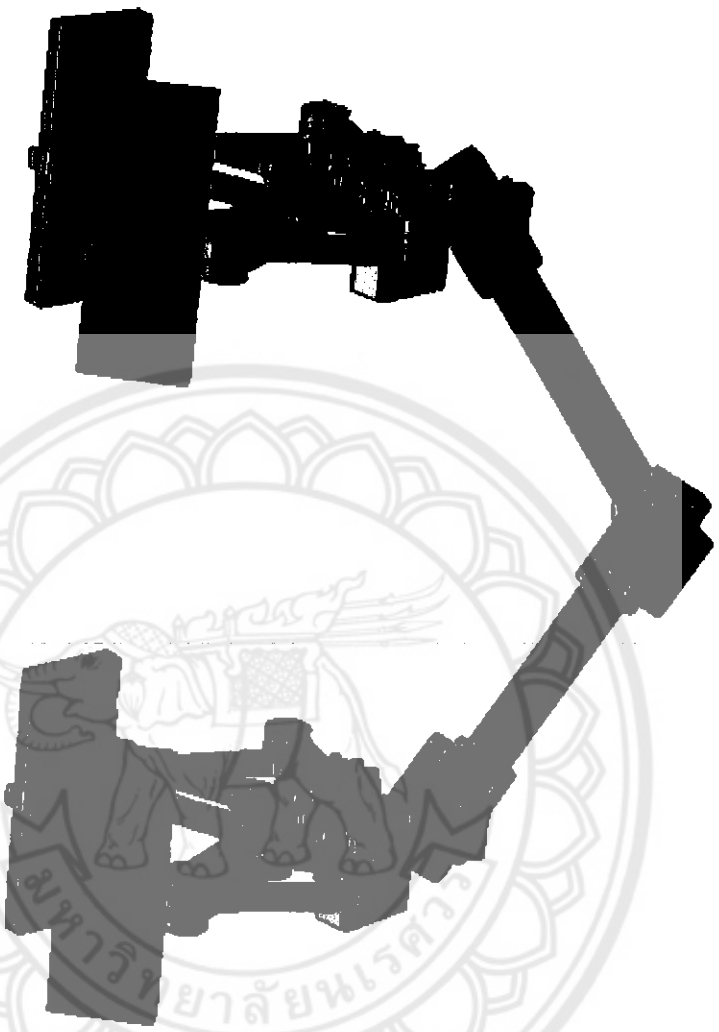
### 3.4.4 แบบจำลองข้อมือของหุ่นยนต์

ข้อมือเป็นชิ้นส่วนที่ใช้เป็นข้อต่อระหว่างมือจับและแขนของหุ่นยนต์ ด้านบนของข้อมือจะมีสองรูใช้ต่อกับฝ่ามือ ส่วนรูตรงกลางเป็นรูที่ใช้ใส่แกนของเซอร์โวมอเตอร์ที่ยึดอยู่กับแขน




**รูปที่ 3.11** แสดงขนาดและแบบจำลองสามมิติข้อมือของหุ่นยนต์

### 3.4.5 แบบจำลองของหุ่นยนต์ปีนเสา

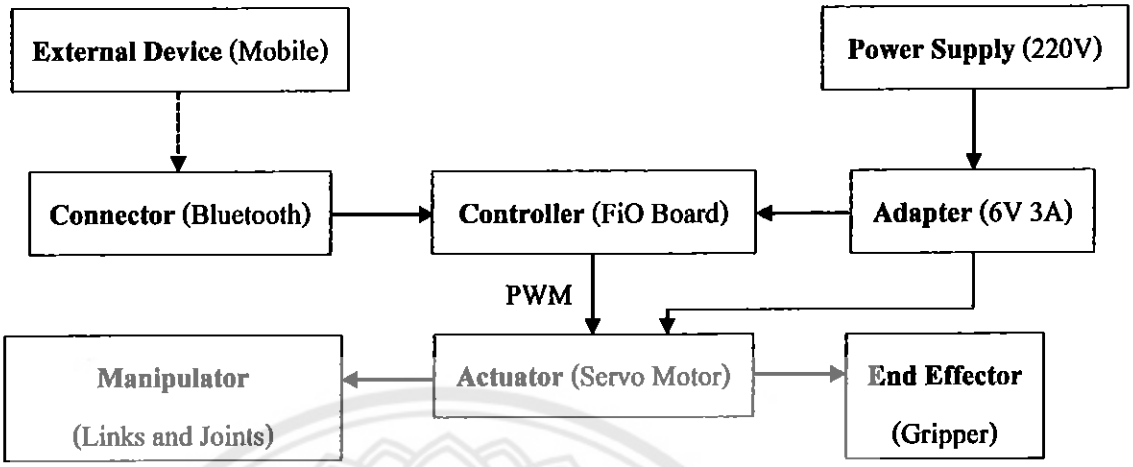


รูปที่ 3.12 แสดงแบบจำลองสามมิติของหุ่นยนต์ปีนเสา

## 3.5 การเชื่อมโยงอุปกรณ์ควบคุมหุ่นยนต์

การเชื่อมโยงอุปกรณ์ควบคุมหุ่นยนต์มีสามส่วนหลัก คือ การเชื่อมโยงสัญญาณควบคุม การเชื่อมโยงพลังงานไฟฟ้า และการเชื่อมโยงการสั่งงานแบบกึ่งอัตโนมัติ ดังรูปที่ 3.13 โดยส่วนของการเชื่อมโยงสัญญาณควบคุมจะเป็นการเชื่อมต่อสัญญาณจาก STM32TM ARM 32-bits CortexTM – M3 Processors บนบอร์ดไฟโอ (FiO Board) โดยส่งสัญญาณ PWM ผ่านตัวเชื่อมต่อ (Connector) ไปยังเซอร์โวมอเตอร์ (Servo Motor) ส่วนการเชื่อมโยงพลังงานไฟฟ้าเป็นการเชื่อมต่อไฟฟ้ากระแสตรงขนาด 6 โวลต์ 3 แอมแปร์ จากหม้อแปลงไฟฟ้า (Adapter) กับบอร์ดไฟโอและเซอร์โวมอเตอร์ ส่วนการเชื่อมโยงการสั่งงานแบบกึ่งอัตโนมัติเป็นการเชื่อมต่อโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์ (Android) กับบอร์ดไฟโอด้วยสัญญาณไร้สายผ่านบลูทูธ

(Bluetooth)

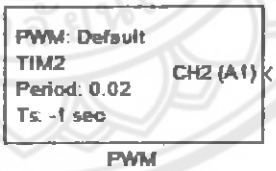



รูปที่ 3.13 แสดงการเชื่อมโยงอุปกรณ์ควบคุมหุ่นยนต์ปีนเสา

----- แสดงการเชื่อมต่อแบบไร้สาย

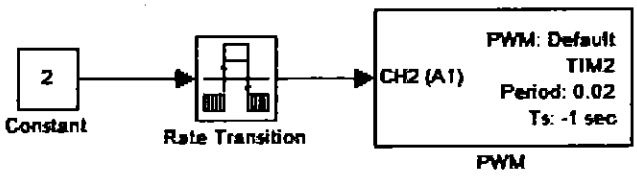
## 3.6 การเขียนโปรแกรมสร้างสัญญาณพัลส์วิดมอดูเลชัน

การควบคุมเซอร์โวมอเตอร์ (Servo Motor) ต้องอาศัยการป้อนสัญญาณพัลส์วิดมอดูเลชัน จากบอร์ดไฟโอ (FiO Board) โดยใช้บล็อก PWM ในไลบรารีชุดกล่องคำสั่งราปิดเอสทีเอ็ม32 (RapidSTM32 Blockset) ดังรูปที่ 3.14



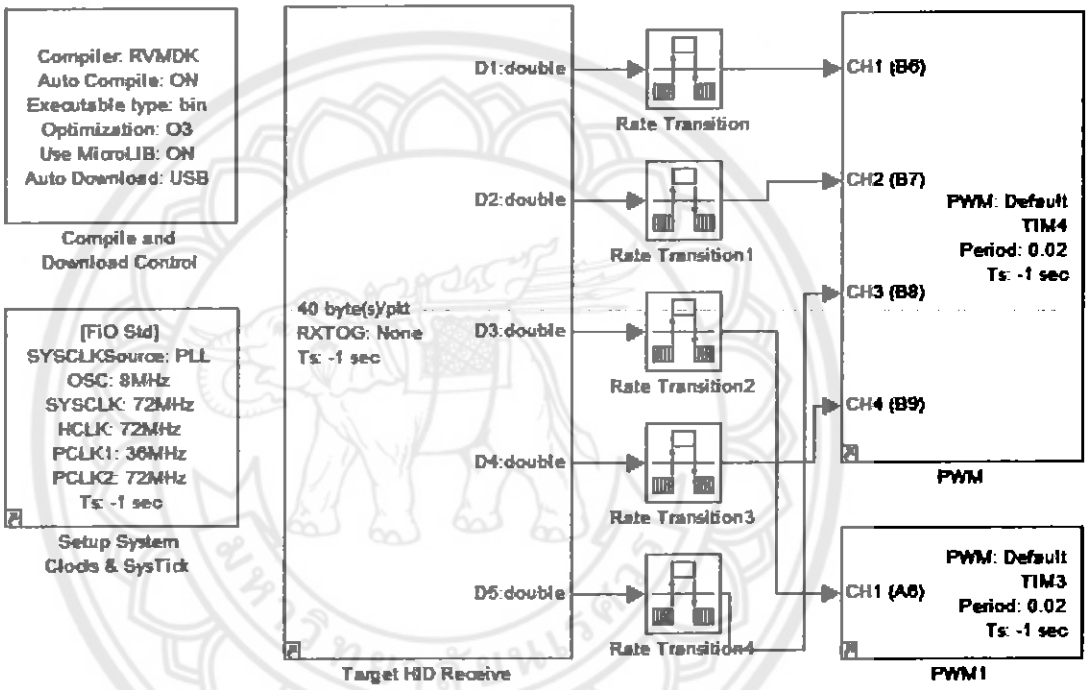

รูปที่ 3.14 แสดงภาพบล็อก PWM ที่ใช้สร้างสัญญาณพัลส์วิดมอดูเลชัน

ข้อมูลที่บล็อก PWM รับเข้ามานั้น ต้องผ่านบล็อก Rate Transition เพื่อปรับอัตราการถ่ายโอนข้อมูลระหว่างบล็อกให้มีอัตราการถ่ายโอนเท่ากัน ดังรูปที่ 3.21



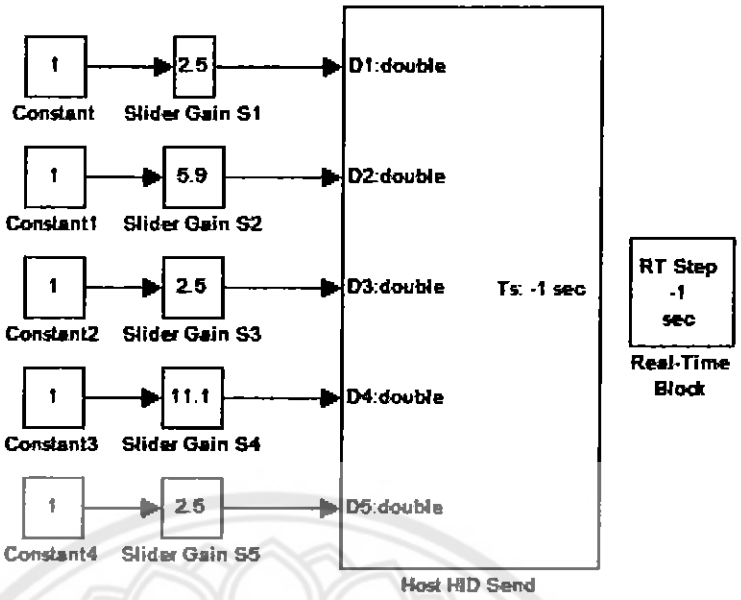

**รูปที่ 3.15** แสดงภาพการส่งค่าให้กับบล็อก PWM

การเขียนฟังก์ชั่นสร้างสัญญาณพัลส์วิดมอดูเลชัน (PWM) เพื่อควบคุมเซอร์โวมอเตอร์ จำเป็นต้องออกแบบให้สามารถทำการส่งสัญญาณพัลส์วิดมอดูเลชันออกทางพอร์ต (Port) ได้หลายช่องทางในเวลาเดียวกัน เพราะต้องควบคุมเซอร์โวมอเตอร์จำนวน 5 ตัว โดยค่าพัลส์วิดมอดูเลชันจะต้องถูกส่งให้แก่เซอร์โวมอเตอร์ตลอดเพื่อคงตำแหน่งของการเคลื่อนไหวของหุ่นยนต์ โปรแกรมที่ใช้เพื่อส่งค่าให้กับเซอร์โวมอเตอร์ทั้งหมดมีชื่อว่า Test_Target.mdl เป็นโปรแกรมที่ลงไว้ในบอร์ดไฟโอ เพื่อใช้ส่งค่าที่ได้รับให้กับเซอร์โวมอเตอร์จากฝั่งของ Host ดังรูปที่ 3.16




รูปที่ 3.16 แสดงโปรแกรม Test_Target.mdl

จากรูปที่ 3.16 กล่อง Target HID Receive จะทำหน้าที่รับข้อมูลจากส่วนของ PC ที่ส่งข้อมูลโดยใช้ Host HID Receive ซึ่งข้อมูลส่งออกจะมี 5 ข้อมูล เป็นชนิด Double เนื่องจากค่าที่ใช้เป็นค่าที่มีทศนิยม ข้อมูลทั้งหมดถูกทำให้ผ่าน Rate Transition เพื่อปรับอัตราการถ่ายโอนข้อมูลให้มีค่าเท่ากัน และส่งข้อมูลไปยังบล็อก PWM นำออกไปยัง ขา A6, B6, B7, B8 และ B9 โดย Test_Target.mdl จะต้องใช้ Test_Host.mdl ในการควบคุมโดยบล็อกที่ใช้จะมีบล็อก Constant อยู่ด้านซ้ายมือเป็นค่าเริ่มต้น ส่วนบล็อกถัดมาจะเป็นบล็อก Slider Gain ซึ่งสามารถเลือกตัวเลขโดยการเลื่อนบาร์ และกล่อง Host HID Send มีหน้าที่ส่งจากฝั่ง Host ไปยังบอร์ดไฟโอ ผ่าน Target HID Receive ดังรูปที่ 3.17




รูปที่ 3.17 แสดงโปรแกรม Test_Host.mdl

## 3.7 การสร้างหุ่นยนต์ปีนเสา

วัสดุที่ใช้ทำหุ่นยนต์ต้นแบบและการสร้างขึ้นส่วนหุ่นยนต์ต้นแบบ ใช้อะคริลิกเป็นวัสดุของตัวหุ่นยนต์และมือจับ ซึ่งมีความแข็งแรงพอสมควร สามารถใช้งานแทนอลูมิเนียมได้ในระดับหนึ่ง แต่อย่างไรก็ตามถึงแม้อะคริลิกจะมีความแข็ง แต่ก็มีความเปราะเช่นกัน และความเปราะนี้ถือเป็นจุดอ่อนในการใช้งานในบริเวณที่มีการสั่นสะเทือนสูง ต่างจากอลูมิเนียมซึ่งมีความยืดหยุ่นกว่า แต่เนื่องจากหุ่นยนต์ปีนเสานี้เป็นหุ่นยนต์ขนาดเล็ก ผลกระทบจากแรงสั่นสะเทือนต่อวัสดุอะคริลิกจึงมีน้อยมาก ชิ้นส่วนทั้งหมดที่ใช้อะคริลิกเป็นวัสดุนี้ถูกทำขึ้นให้มีความใกล้เคียงแบบจำลองที่สร้างจากโปรแกรมโซลิดเวิร์ค (SolidWorks) มากที่สุด ส่วนข้อหมุนต่าง ๆ และการยึดเซอร์โวมอเตอร์นั้นใช้นอตขนาดต่าง ๆ ยึดกับส่วนลำตัวและมือจับ ซึ่งชิ้นส่วนเหล่านี้ทำขึ้นเองเพื่อประหยัดค่าใช้จ่าย เมื่อสร้างชิ้นส่วนของหุ่นยนต์ครบตามต้องการแล้ว จึงทำการประกอบชิ้นส่วนหุ่นยนต์เข้าด้วยกัน ดังรูปที่ 3.18

รูปที่ 3.18 แสดงหุ่นยนต์ปีนเสาที่สร้างสำเร็จแล้ว

ตารางที่ 3.2 รายละเอียดของหุ่นยนต์ปีนเสาที่สร้างจริง

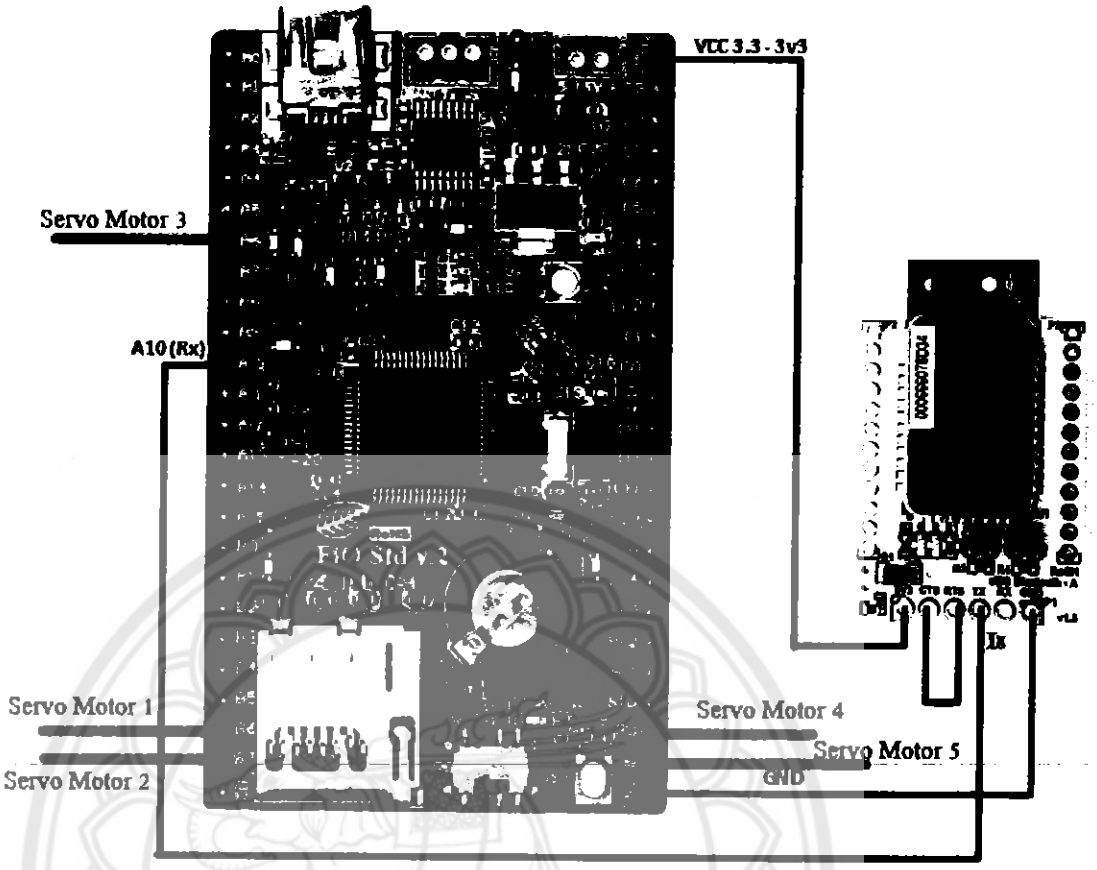
| ข้อมูล | รายละเอียด |
|---|---|
| ขนาด | ความกว้าง 18 เซนติเมตร ความยาว 75 เซนติเมตร ความสูงขณะจับยึด 33 เซนติเมตร |
| จำนวนมือจับและข้อหมุนอิสระ | 2 มือจับ 5 ข้อหมุน |
| น้ำหนักรวม | 1.4 กิโลกรัม |
| โครงสร้าง | อะคริลิก |
| ตัวขับแต่ละข้อหมุนอิสระ | เซอร์โวมอเตอร์ SRC SR431 Metal Gear 180 deg., Metal Gears, Double Ball Bearing จำนวน 5 ตัว Speed (sec/60deg) : 0.20 (At 6.0 V), 0.18 (At 7.4 V) |

| ข้อมูล | รายละเอียด |
|---|---|
| | Torque : 12.2 kg-cm (At 6.0 V), 14.5 kg-cm (At 7.4 V)<br>Dimension (WxLxH) : 20.5 x 42.0 x 39.5 mm<br>Weight (g/oz) : 62/2.18 |
| ความต้องการพลังงาน | ไฟฟ้ากระแสตรงความต่างศักย์ 6 V และกระแส 3 A |
| แหล่งพลังงาน | ไฟฟ้ากระแสสลับผ่านหม้อแปลงไฟฟ้า (Adapter) |
| บอร์ดควบคุม | FiO Board (STM32TM ARM 32-bits CortexTM – M3) |
| อุปกรณ์เสริม | aMG Bluetooth – AC2 (With RN – 42 Module |

## 3.8 การเชื่อมต่อสายไฟระหว่างอุปกรณ์

การเชื่อมต่อระหว่างบอร์ดไฟโอ (FiO Board) และเซอร์โวมอเตอร์ จะต่อสายสัญญาณของเซอร์โวมอเตอร์ (Servo Motor) S1, S2, S3, S4 และ S5 เข้ากับพอร์ต (Port) B6, B7, A6, B9 และ B8 ของบอร์ดไฟโอตามลำดับ สายไฟของเซอร์โวมอเตอร์จะรับจากแหล่งจ่ายไฟภายนอกจากหม้อแปลง (Adapter) เป็นไฟกระแสตรง 6 โวลต์ 3 แอมแปร์ ส่วนสายดิน (Ground) ของเซอร์โวมอเตอร์จะต้องต่อเข้ากับพอร์ต GND เพื่อทำให้เซอร์โวมอเตอร์ทำงาน

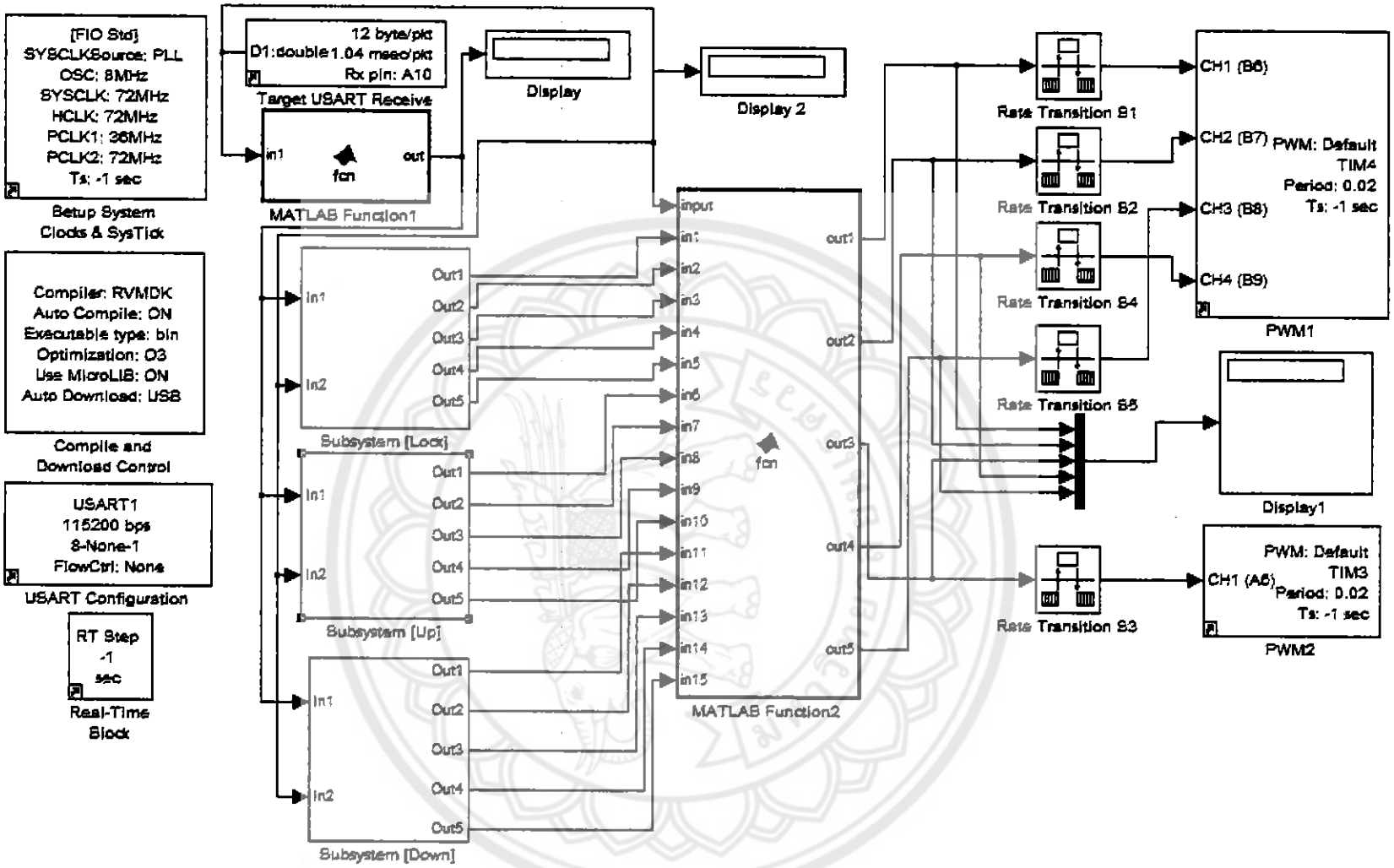
อุปกรณ์บลูทูธ (Bluetooth Device) ที่ใช้เชื่อมต่อกับบอร์ดไฟโอ คือ aMG Bluetooth – AC2 (with RN-42 Module) โดยต่อพอร์ต 3v3 เข้ากับพอร์ต VCC 3.3 – 3v3 ของบอร์ด ต่อพอร์ต GND ของบลูทูธกับพอร์ต GND ของบอร์ด เชื่อมต่อพอร์ต CTB และ RTB เข้าด้วยกัน และพอร์ต TX ของบลูทูธให้ต่อเข้ากับพอร์ต A10 ดังรูปที่ 3.19




รูปที่ 3.19 การเชื่อมต่อสายไฟระหว่างอุปกรณ์

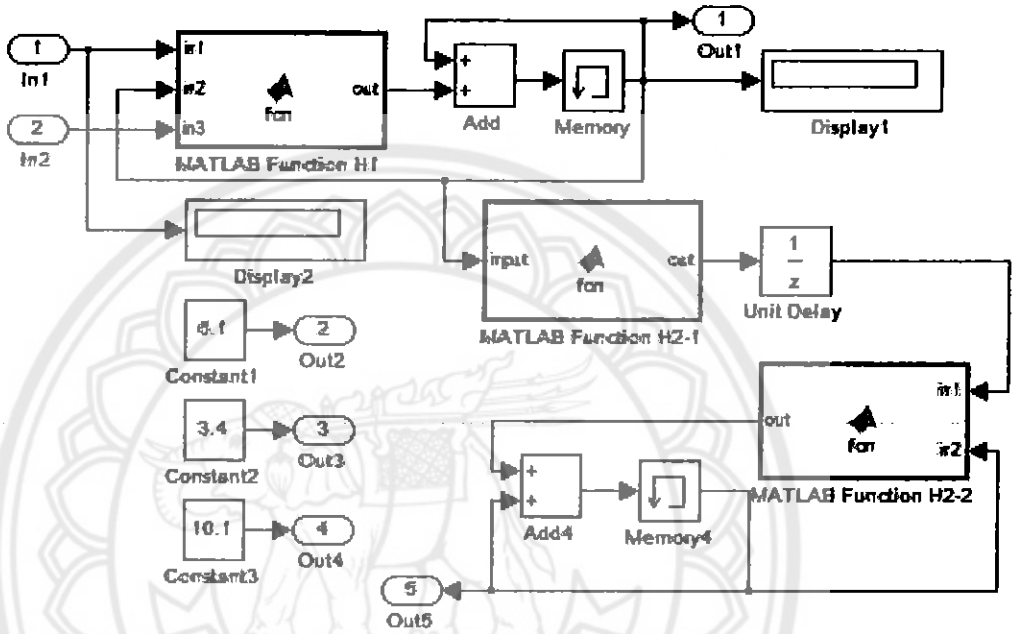
## 3.9 การเขียนโปรแกรมในการควบคุมการปีนเสาของหุ่นยนต์

การควบคุมการเคลื่อนที่ทั้งสามรูปแบบมีหน้าตาโปรแกรมที่เหมือนกัน แต่จะต่างกันด้วยลำดับที่อยู่ภายในระบบย่อย เริ่มจากบล็อกรับข้อมูลยูซาร์ต (Target USART Receive) ใช้รับข้อมูลจากอุปกรณ์บลูทูธ ส่งข้อมูลไปยังบล็อก MATLAB Function 1 ที่มีหน้าที่เลือกงานที่ต้องทำจากข้อมูลที่เข้ามาว่า ปีนขึ้น ปีนลง หรืออยู่ในท่าเตรียมพร้อมจับเสา ซึ่ง Subsystem [Lock] ระบบย่อยที่รวมคำสั่งบังคับให้หุ่นยนต์อยู่ในท่าเตรียมพร้อมจับเสา Subsystem [Up] บังคับให้หุ่นยนต์ปีนขึ้นเสา และ Subsystem [Down] บังคับหุ่นยนต์ให้ปีนลงเสา ส่วน MATLAB Fucntion 2 มีหน้าที่เลือกข้อมูลที่จะส่งให้แก่บอร์ดไฟโอ ซึ่งต้องผ่าน Rate Transition เพื่อปรับอัตราการถ่ายโอนให้เท่า ๆ กันก่อนส่งไปยังบล็อก PWM ที่ทำหน้าที่จ่ายสัญญาณพัลส์วิดมอดูเลชันให้กับเซอร์โวมอเตอร์ ดังรูปที่ 3.20



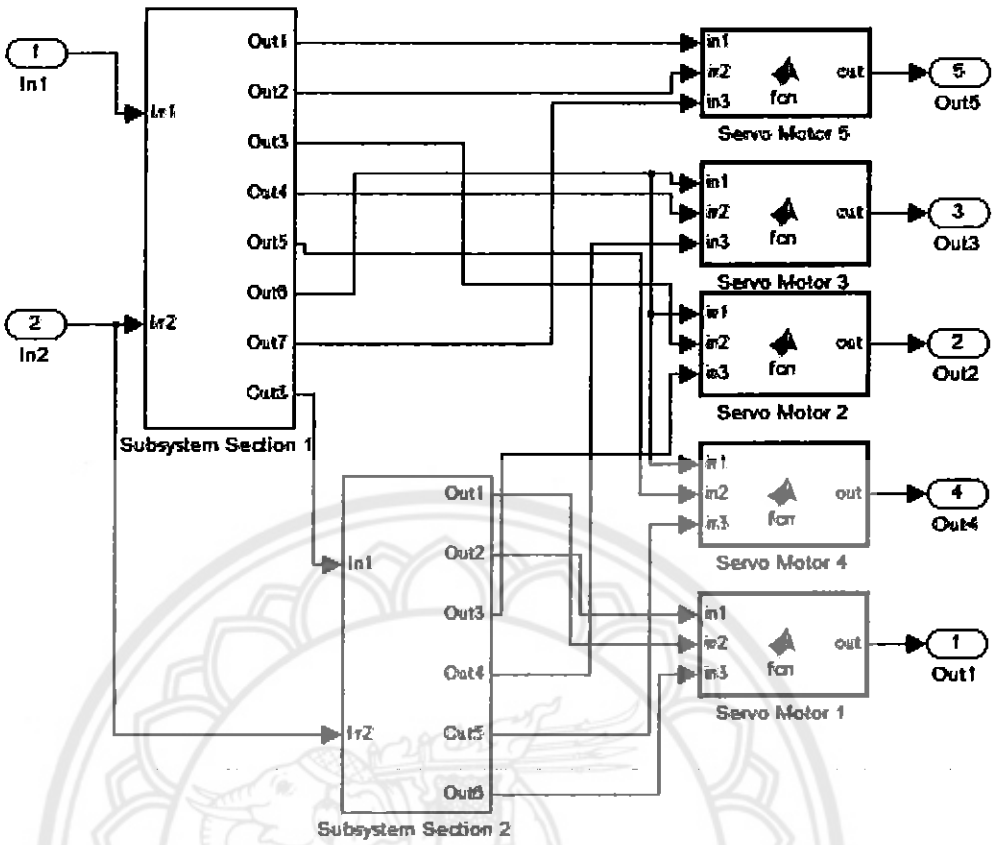

รูปที่ 3.20 แสดงโปรแกรมควบคุมหุ่นยนต์ในโปรแกรมซิมูลิงค์

บล็อก Subsystem [Lock] เป็นบล็อกรวมคำสั่งท่าเตรียมพร้อมของหุ่นยนต์ บล็อก MATLAB Function H1 ทำหน้าที่ปรับให้มือจับบนค่อย ๆ บีบเข้า MATLAB Function H2-1 และ MATLAB Function H2-2 จะสั่งให้มือจับล่างค่อย ๆ บีบเข้า หลังจากมือจับบนบีบเสร็จแล้ว ส่วน Constant1 Constant2 และ Constant3 เป็นค่าที่ส่งให้กับเซอร์โวมอเตอร์ข้อมือบน ลำตัว และข้อมือล่าง ดังรูปที่ 3.21



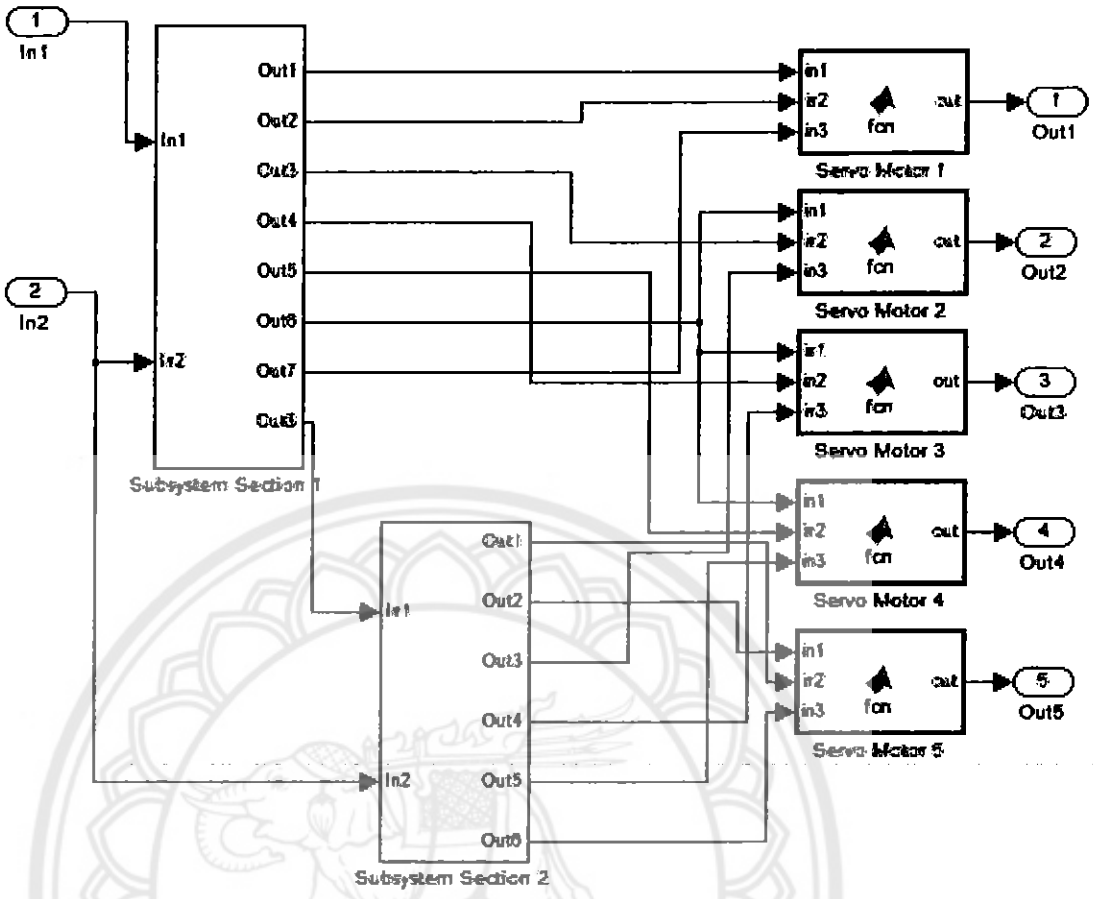

รูปที่ 3.21 แสดงระบบย่อยของคำสั่งท่าเตรียมพร้อม Subsystem [Lock]

บล็อก Subsystem [Up] เป็นบล็อกรวมคำสั่งปีนขึ้นเสา มีความซับซ้อนมาก จึงสร้างระบบย่อยใน Subsystem [Up] อีก 2 บล็อก คือ Subsystem Section 1 และ Subsystem Section 2 ใน Section 1 จะเป็นส่วนของการปีนช่วงแรก ตั้งแต่การปล่อยมือจับล่าง หดตัวเพื่อยกส่วนล่างของหุ่นยนต์ขึ้น จนมือล่างบีบจับเสาอีกครั้ง ส่วน Section 2 จะเป็นการปีนช่วงหลัง คือ ปล่อยมือจับบน ยืดตัวเพื่อยกส่วนส่วนบน จนถึงบีบมือจับบนอีกครั้ง และมี MATLAB Function อีก 5 บล็อก ชื่อ Servo Motor 1 – 5 ทำหน้าที่ลำดับข้อมูลที่ต้องการส่งออก ดังรูปที่ 3.22




**รูปที่ 3.22** แสดงระบบย่อยของคำสั่งปีนขึ้นเสา Subsystem [Up]

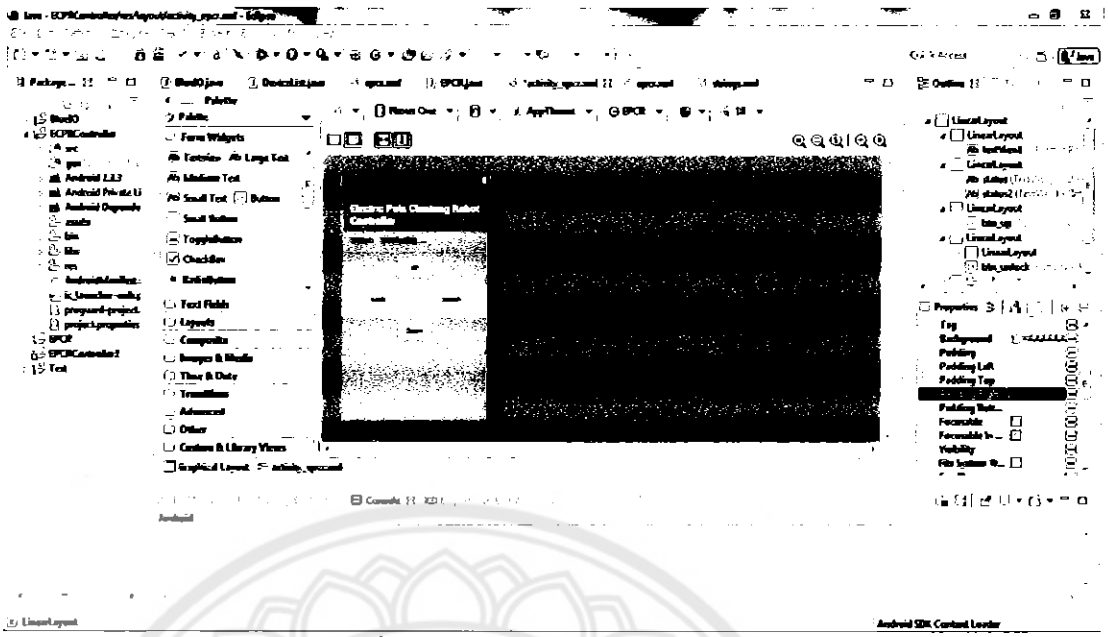
บล็อก Subsystem [Down] รวมคำสั่งการปีนลงเสาเอาไว้ ภายในจะประกอบด้วยระบบย่อยอีก 2 บล็อก และบล็อก MATLAB Function 5 บล็อก ชื่อ Servo Motor 1 – 5 เช่นเดียวกับ Subsystem [Up] แต่ Section 1 จะเริ่มด้วยการปล่อยมือจับบน หดตัวเพื่อดึงส่วนบนลงมา และมือจับบนจับเสา และ Section 2 เริ่มด้วยการปล่อยมือจับล่าง ยืดตัว และมือจับล่างจับเสา ดังรูปที่ 3.23




**รูปที่ 3.23** แสดงระบบย่อยของคำสั่งปีนลงเสา Subsystem [Down]

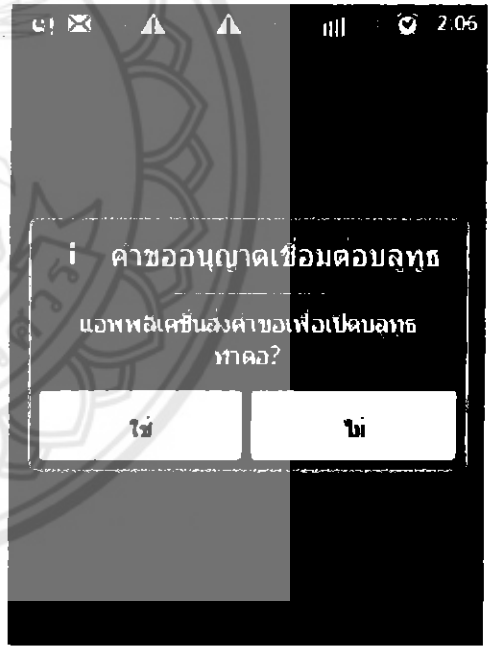
## 3.10 การเขียนโปรแกรมควบคุมการปีนเสาของหุ่นยนต์ผ่านสัญญาณบลูทูธ

การเขียนโปรแกรมควบคุมหุ่นยนต์ปีนเสาในโครงงานนี้ ใช้โปรแกรมอีคลิปส์ (Eclipse) เพื่อเขียนแอพพลิเคชั่น (Application) ชื่อ ECPR Controller ที่เขียนขึ้นมาเองซึ่งทำงานบนระบบปฏิบัติการแอนดรอยด์ (Android Operating System)

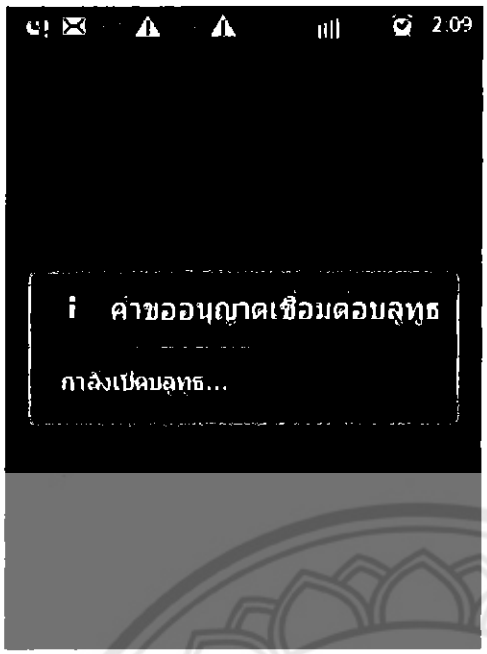


รูปที่ 3.24 แสดงภาพโปรแกรมอีคลิปส์

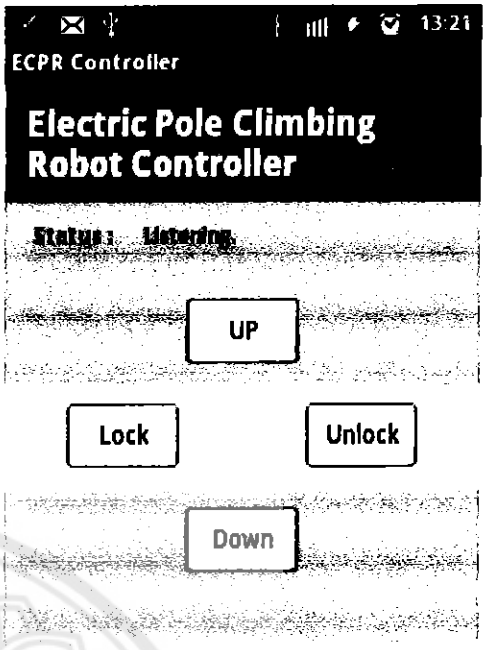

(ก) ไอคอนโปรแกรม ECPR Controller



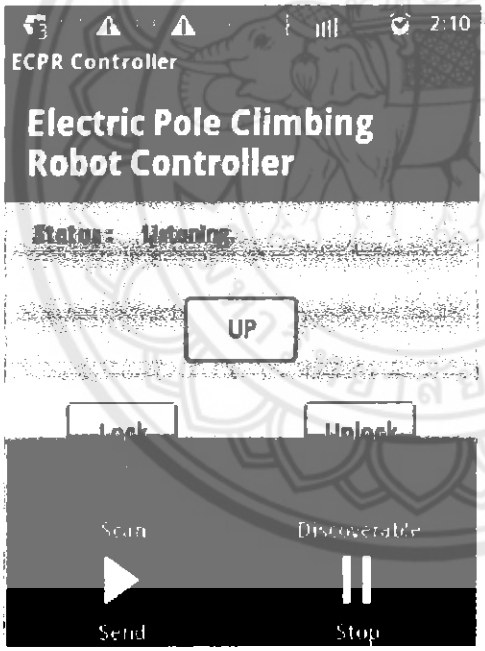

(ข) การขออนุญาติเปิดบลูทูธ



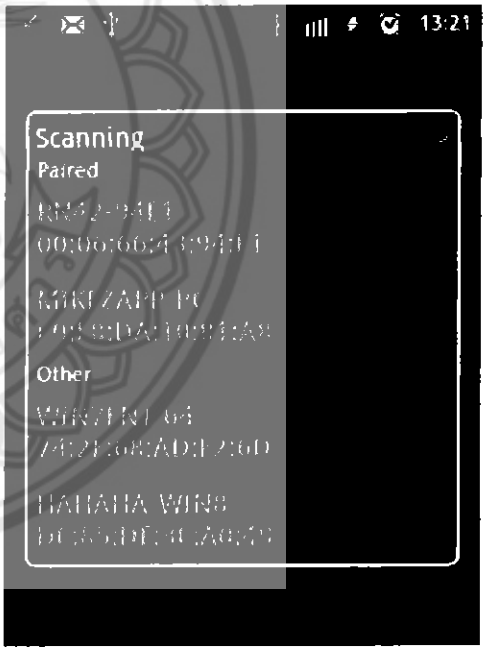

(ค) เปิดบลูทูธ



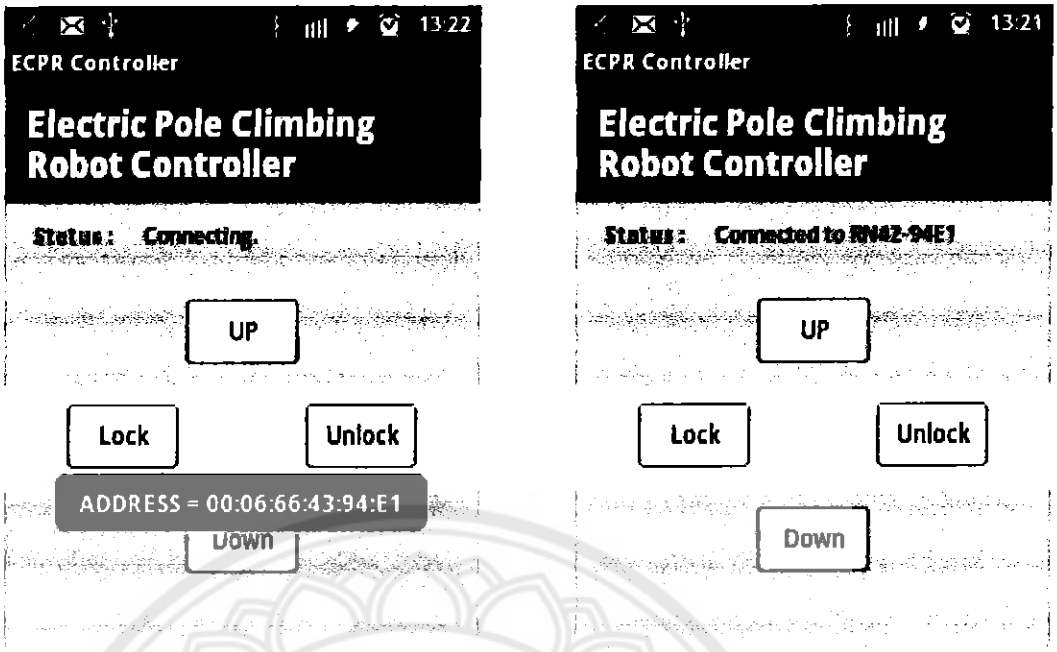

(ง) หน้าจอโปรแกรม




(จ) เมนู




(ฉ) ค้นหาอุปกรณ์




(ช) เมนู (ซ) ค้นหาอุปกรณ์

**รูปที่ 3.25** แสดงโปรแกรม ECPR Controller

(ก) แสดงรูปไอคอนของโปรแกรม ECPR Controller

(ข) โปรแกรมจะแสดงคำขออนุญาตเชื่อมต่อบลูทูธหากอุปกรณ์ไม่ได้เปิดบลูทูธไว้

(ค) เมื่อโปรแกรมเปิดการใช้งานบลูทูธ

(ง) โปรแกรมอยู่ในสถานะ Listening คือ กำลังรอฟังคำสั่ง

(จ) เมื่อกดปุ่ม Menu ของอุปกรณ์จะปรากฏคำสั่ง Scan, Discovery, Send และ Stop โดยแต่ละคำสั่ง จะมีหน้าที่ดังนี้

- Scan ใช้ค้นหาอุปกรณ์อื่น ๆ ที่เปิดบลูทูธอยู่
- Discovery ใช้เพื่อให้อุปกรณ์บลูทูธอื่น ๆ หาบลูทูธของอุปกรณ์นี้เจอ
- Send ใช้เริ่มส่งค่าจากอุปกรณ์นี้ ให้กับบลูทูธของหุ่นยนต์
- Stop ใช้หยุดการทำงานของโปรแกรม

(ฉ) แสดงภาพการค้นหาอุปกรณ์อื่น ๆ จะแสดงอุปกรณ์ที่เปิดบลูทูธอยู่และอุปกรณ์ที่เคยเชื่อมต่อแล้ว

(ช) แสดงภาพเมื่ออุปกรณ์กำลังเชื่อมต่อกับบลูทูธของหุ่นยนต์และแสดงหมายเลข Address ของอุปกรณ์ที่เชื่อมต่อ

(ซ) แสดงภาพเมื่ออุปกรณ์เชื่อมต่อเสร็จแล้ว โดยจะแสดงข้อความสถานะว่า Connected to และตามด้วยชื่อของอุปกรณ์ที่เชื่อมต่อด้วย

# บทที่ 4

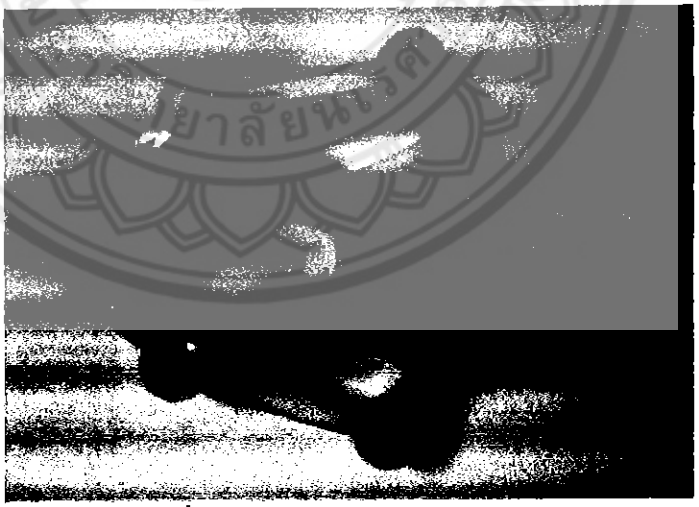
# การทดลองและผลการทดลอง

## 4.1 การทดลองการเคลื่อนไหวของหุ่นยนต์

### 4.1.1 มือจับ

การทดลองการคลายออกและบีบเข้าของมือจับ จะนำค่าที่ได้จากการทดลองบันทึกลงในตาราง เพื่อนำค่าเฉลี่ยมาเขียนโปรแกรมสั่งมือจับ โดยขั้นตอนแรกป้อนสัญญาณพัลส์วิดมอดูเลชันให้กับเซอร์โวมอเตอร์ duty cycle 2% เพื่อให้มือจับคลายออกจนสุด จากนั้นวัดระยะห่างระหว่างนิ้วทั้งสอง ขั้นตอนที่สองคือการทดลองการบีบเข้าของมือจับโดยป้อนสัญญาณพัลส์วิดมอดูเลชัน เพิ่มขึ้นเรื่อย ๆ จนกว่ามือจับจะบีบจนสุด

จากการทดลองจำนวน 20 ครั้ง สามารถสรุปได้ว่า เมื่อบีบจนสุดสัญญาณพัลส์วิดมอดูเลชันที่ใช้จะมีค่าอยู่ที่ duty cycle 5.9 – 6.0% เมื่อนำค่าจากการทดลองมาเฉลี่ยจะได้ค่าพัลส์วิดมอดูเลชัน 6.0% duty cycle ส่วนการคลายออกของมือจับค่าพัลส์วิดมอดูเลชันที่ใช้จะเป็น duty cycle 2% วัดความกว้างของมือจับได้ 15.5 – 15.7 เซนติเมตร ค่าเฉลี่ยคือ 15.5 เซนติเมตร และความคลาดเคลื่อนที่เกิดขึ้น เนื่องจากฟันเฟืองของมือจับที่มีขนาดฟันเฟืองค่อนข้างไม่ละเอียด บันทึกผลการทดลองอยู่ในตารางภาคผนวกที่ ก.1

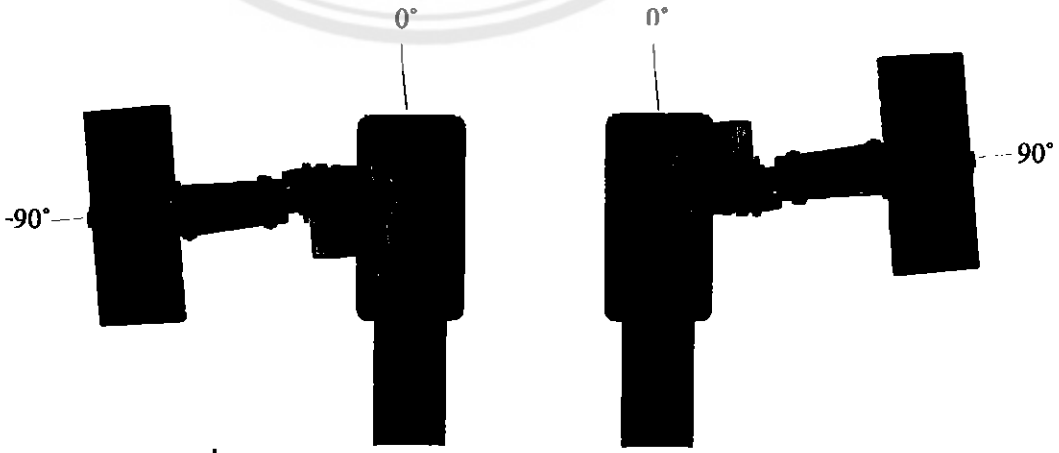


**รูปที่ 4.1** แสดงภาพมือจับขณะบีบเข้า

รูปที่ 4.2 แสดงภาพมือจับขณะคลายออก

4.1.2 ข้อมือ

ทดลองการกวาดมุมของข้อมือและวัดค่ามุม นำค่าที่ได้จากสถิติมาใช้ในการทดลองมาเขียนโปรแกรมสั่งงานด้วยค่าที่เหมาะสม เริ่มด้วยการป้อนค่าพัลส์วิดมอดูเลชันให้กับข้อมือ duty cycle 2% และวัดมุม จากนั้นป้อนค่าพัลส์วิดมอดูเลชัน duty cycle 12% และวัดมุมที่กวาดไป โดยยึดแกนตามรูปที่ 4.2




รูปที่ 4.3 แสดงภาพการหมุนเข้าหาเสาและออกจากเสาของข้อมือ

จากการทดลองพบว่า ข้อมือของหุ่นยนต์สามารถหมุนได้เต็มความสามารถของเซอร์โวมอเตอร์ ตั้งแต่ค่าสัญญาณพัลส์วิดมอดูเลชัน duty cycle 2% มุมที่ได้คือ -90° เมื่อป้อนค่าพัลส์วิดมอดูเลชัน duty cycle 12% จะได้มุม 90° บันทึกผลการทดลองแสดงไว้ในตารางภาคผนวกที่ ก.2

### 4.1.3 ตัวหุ่นยนต์

เป็นการทดลองการยืดตัวและหดตัวของหุ่นยนต์ เพื่อใช้ค่าที่เหมาะสมโดยดูจากสถิติมาเขียนโปรแกรมควบคุมการยืดตัวและหดตัว ขั้นแรกป้อนค่าพัลส์วิดมอดูเลชันให้กับเซอร์โวมอเตอร์ตัวกลาง duty cycle 2% เพื่อให้หุ่นยนต์ยืดจนสุด และวัดมุมที่ได้ ขั้นต่อไป คือ ป้อนค่าพัลส์วิดมอดูเลชันเพิ่มขึ้น จนกว่าหุ่นยนต์จะหดตัวไม่ได้ และวัดมุม

จากการทดลองการยืดและหดตัวของหุ่นยนต์นั้น ป้อนพัลส์วิดมอดูเลชัน duty cycle 2% หุ่นยนต์จะยืดตัวสุด ซึ่งได้มุม -90° และเมื่อป้อน duty cycle 10.2% หุ่นยนต์จะหดตัวจนสุด ทำมุมประมาณ 50° บันทึกผลการทดลองดังแสดงตารางภาคผนวกที่ ก.3

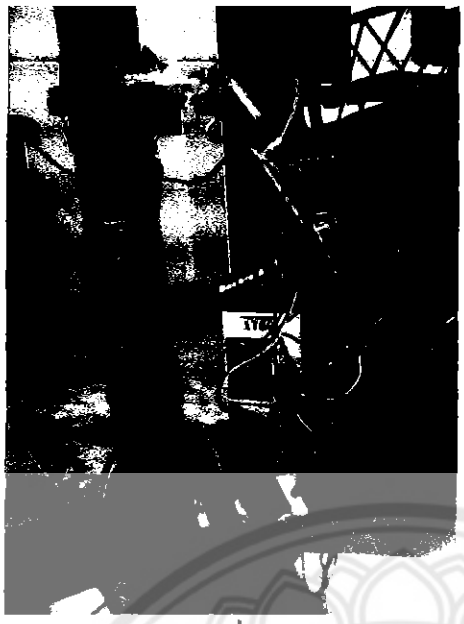

**รูปที่ 4.4** แสดงภาพของการหดและยืดตัวของหุ่นยนต์

## 4.2 การทดลองหาท่าทางในการปีนเสาของหุ่นยนต์

การปีนเสาคือการจับยึดเสาแล้วอาศัยการยืดหด พร้อมเคลื่อนตัวขึ้นลงของหุ่นยนต์ โดยที่การปีนเสาแต่ละท่าทางย่อยในการปีนเสา หุ่นยนต์จะต้องอยู่ในภาวะสมดุลเสมอ

ซึ่งหมายความว่า เมื่อหุ่นยนต์หยุดอยู่ที่ท่าทางใดก็ตาม หุ่นยนต์จะสามารถยืนอยู่ได้โดยไม่ร่วงลงจากเสาการทดลองหาท่าทางนั้น เพื่อนำค่าเฉลี่ยที่ได้ไปเขียนควบคุมการเคลื่อนไหวทั้งหมดของหุ่นยนต์เริ่มจากการติดตั้งโปรแกรม Test_Target.mdl ที่ทำหน้าที่ควบคุมเซอร์โวมอเตอร์แต่ละตัวลงบนบอร์ดไฟโอ จากนั้นรันโปรแกรม Test_Host.mdl ซึ่งเป็นโปรแกรมที่ใช้ส่งค่าพัลส์วิดมอดูเลชันให้กับเซอร์โวมอเตอร์ ป้อนค่าให้หุ่นยนต์ใช้มือจับบนจับยึดเสาไว้ จากนั้นค่อย ๆ ปรับค่าพัลส์วิดมอดูเลชันให้เซอร์โวมอเตอร์แต่ละตัวจนกว่าจะได้ท่าทางที่ต้องการ หากหุ่นยนต์ไม่สามารถปีนขึ้น ยกท่อนบนไม่ขึ้น หรือมือจับไม่ตั้งฉากกับเสา ต้องทำการปรับค่าใหม่จนกว่าจะได้ท่าทางที่เหมาะสมและบันทึกผล ซึ่งท่าทางที่ทดลองจะเป็นดังรูปที่ 4.5

(ก) ท่าเริ่มต้น (ข) ปล่อยมือจับล่าง



(ค) ปรับข้อมือบนเพื่อยกตัวออกจากเสา

(ง) หดตัวเข้าหากัน

(จ) ปรับข้อมือบนเพื่อยกตัวเข้าหาเสา

(ฉ) ปรับข้อมือล่างเตรียมจับยึด

(ช) บีบมือจับล่าง

(ซ) ปล่อยมือจับบน

(ฌ) ปรับข้อมือล่างเพื่อยกตัวออกจากเสา

(ญ) ยืดตัวออก

(ฎ) ปรับข้อมือล่างเพื่อยกตัวเข้าหาเสา

(ฎ) ปรับข้อมือบนเตรียมจับยึด

(ฐ) ท่าเป้าหมาย

**รูปที่ 4.5** แสดงภาพการทดลองการปีนขึ้นเสาของหุ่นยนต์

การทดลองหาท่าทางการปีนขึ้นเสาของหุ่นยนต์ดังที่แสดงในตารางภาคผนวกที่ ก.4 พบว่าบางครั้งเกิดการลื่นไถลของมือจับบนหรือหุ่นยนต์จับเสาโดยที่ไม่ได้ตั้งฉากกับเสา ทำให้ลำดับที่ท่าทางถัดไปผิดพลาดตามไปด้วย ซึ่งการบันทึกผลการทดลองนี้บันทึกเฉพาะค่าพัลส์มอดูเลชันของท่าทางการปีนที่ปีนสำเร็จเท่านั้น และได้เลือกค่าพัลส์มอดูเลชันที่ป้อนให้กับเซอร์โวมอเตอร์แต่ละตัวตามลำดับ ดังตารางที่ 4.1

ตารางที่ 4.1 ค่าพัลส์วิดมอดูเลชันที่เฉลี่ยป้อนให้กับเซอร์โวมอเตอร์

| ลำดับ | ท่าทาง | PWM (% Duty Cycle) | | | | |
|---|---|---|---|---|---|---|
| | | H1 | S2 | S3 | S4 | H5 |
| 1 | จับยึดเสาด้วยสองมือ (ท่าเริ่มต้น) | 8.0 | 6.4 | 2.5 | 10.9 | 7.0 |
| 2 | ปล่อยมือจับล่าง | 8.0 | 6.4 | 2.5 | 10.9 | 2.0 |
| 3 | ปรับข้อมือบนเพื่อยกตัวออกจากเสา | 8.0 | 8.7 | 2.5 | 10.9 | 2.0 |
| 4 | หดตัวเข้าหากัน | 8.0 | 8.7 | 5.7 | 10.9 | 2.0 |
| 5 | ปรับข้อมือบนเพื่อยกตัวเข้าหาเสา | 8.0 | 7.9 | 5.7 | 10.9 | 2.0 |
| 6 | ปรับข้อมือล่างเตรียมจับยึด | 8.0 | 7.9 | 5.7 | 9.0 | 2.0 |
| 7 | บีบมือจับล่าง | 8.0 | 7.9 | 5.7 | 9.0 | 7.0 |
| 8 | ปล่อยมือจับบน | 2.0 | 7.9 | 5.7 | 9.0 | 7.0 |
| 9 | ปรับข้อมือล่างเพื่อยกตัวออกจากเสา | 2.0 | 7.9 | 5.7 | 10.0 | 7.0 |
| 10 | ยืดตัวออก | 2.0 | 7.9 | 2.5 | 10.0 | 7.0 |
| 11 | ปรับข้อมือล่างเพื่อยกตัวเข้าหาเสา | 2.0 | 7.9 | 2.5 | 10.9 | 7.0 |
| 12 | ปรับข้อมือบนเตรียมจับยึด | 2.0 | 6.4 | 2.5 | 10.9 | 7.0 |
| 13 | บีบมือจับบน (ท่าเป้าหมาย) | 8.0 | 6.4 | 2.5 | 10.9 | 7.0 |

## 4.3 การทดลองหารูปแบบการเคลื่อนที่ที่ดีที่สุด

รูปแบบการเคลื่อนที่ทั้ง 3 นั้นใช้ค่าพัลส์วิดมอดูเลชันที่มีค่าเท่ากัน แต่จะแตกต่างกันในเรื่องลำดับการปีน ระยะทางที่ได้จะใกล้เคียงกันมากเนื่องจากค่าพัลส์วิดมอดูเลชันมีค่าเดียวกัน ซึ่งระยะทางของทั้ง 3 รูปแบบ ประมาณ 2.0 เซนติเมตร จึงใช้เวลาที่ใช้เป็นตัวเลือกในการตัดสินใจรูปแบบการเคลื่อนที่ที่ดีที่สุด การเคลื่อนที่รูปแบบที่ 1 หลายครั้งที่หดตัวดึงส่วนล่างขึ้นมาแล้วไม่สามารถยึดตัวยกส่วนบนขึ้นได้ หรือยกตัวแล้วติดเสา ทำให้ระยะทางที่ปีนขึ้นเสาไม่คงที่และยังใช้เวลามากกว่าการเคลื่อนที่อีกสองรูปแบบ เนื่องจากเคลื่อนไหวทีละข้อต่อ จำเป็นต้องรอให้ข้อต่อแรกทำงานเสร็จ ข้อต่อถัดไปจึงจะทำงานต่อได้ ส่วนการเคลื่อนที่ในรูปแบบที่ 2 จะมีลักษณะท่าทางใกล้เคียงกับการเคลื่อนที่รูปแบบที่ 1 แต่จะแตกต่างกันในเรื่องของเวลาที่ใช้ ซึ่งยังคงมีการยกตัวไม่ขึ้นบ้าง ส่วนการเคลื่อนที่ในรูปแบบที่ 3 คือ การเคลื่อนที่แบบขนานกับแนวเสา เป็นการเคลื่อนที่ที่ใช้เวลาน้อยใกล้เคียงกับรูปแบบที่ 2 การทดลองการเคลื่อนที่รูปแบบที่ 3 ไม่พบการติดเสาเมื่อยกตัวขึ้น บันทึกผลการทดลองแสดงดังตารางที่ 4.2

**ตารางที่ 4.2** การทดลองวัดระยะทางและเวลาของการเคลื่อนที่ทั้ง 3 รูปแบบ

| ครั้งที่ | ระยะทาง (เซนติเมตร) | รูปแบบที่ 1 (วินาที) | รูปแบบที่ 2 (วินาที) | รูปแบบที่ 3 (วินาที) | หมายเหตุ |
|---|---|---|---|---|---|
| 1 | 2.0 | 64 | 57 | 52 | |
| 2 | 2.0 | 63 | 56 | 51 | |
| 3 | 2.0 | - | - | 52 | (1) ยกตัวไม่ขึ้น (2) ขาล่างติดเสา |
| 4 | 2.0 | - | 57 | 51 | (1) ยกตัวไม่ขึ้น |
| 5 | 2.0 | 63 | 56 | 51 | |
| 6 | 2.0 | - | 56 | 51 | (1) ยกตัวไม่ขึ้น |
| 7 | 2.0 | 63 | - | 51 | (2) ขาล่างติดเสา |
| 8 | 2.0 | 63 | 56 | 51 | |
| 9 | 2.0 | 63 | 56 | 51 | |
| 10 | 2.0 | 63 | 56 | 52 | |
| 11 | 2.0 | - | 56 | 51 | (1) ยกตัวไม่ขึ้น |
| 12 | 2.0 | - | 56 | 51 | (1) ยกตัวไม่ขึ้น |
| 13 | 2.0 | 63 | 56 | 51 | |
| 14 | 2.0 | 63 | 56 | 51 | |
| 15 | 2.0 | 63 | 57 | 51 | |
| 16 | 2.0 | - | - | 51 | (1, 2) ยกตัวไม่ขึ้น |
| 17 | 2.0 | 63 | 56 | 51 | |
| 18 | 2.0 | 63 | 56 | 51 | |
| 19 | 2.0 | 63 | 56 | 51 | |
| 20 | 2.0 | 63 | - | 51 | (2) ยกตัวไม่ขึ้น |

จากการทดลองการเคลื่อนที่รูปแบบที่ 1 มีความเร็วโดยประมาณ 1.9 เซนติเมตร/นาที รูปแบบที่ 2 มีความเร็ว 2.14 เซนติเมตร/นาที และรูปแบบที่ 3 มีความเร็ว 2.35 เซนติเมตร/นาที รูปแบบที่ 3 จึงมีความเหมาะสมที่สุด ทั้งในเรื่องของระยะเวลาและไม่พบปัญหาขณะปีนขึ้นเสา นอกจากนี้ขนาดของโปรแกรมการเคลื่อนที่รูปแบบที่ 3 ยังมีขนาดที่เล็กกว่าโปรแกรมการเคลื่อนที่รูปแบบที่ 1 และรูปแบบที่ 2 จึงเลือกใช้การปีนเสาเป็นการเคลื่อนที่รูปแบบที่ 3

# บทที่ 5
# สรุปผลและวิจารณ์

หุ่นยนต์ปีนเสาเป็นอีกหนึ่งนวัตกรรมที่มีบทบาทในการลดความเสี่ยงจากการปฏิบัติงานภาคสนาม ซึ่งโครงงานนี้ใช้ซิมูลิงค์ (Simulink) เป็นโปรแกรมขับเคลื่อนหุ่นยนต์และวิเคราะห์การทำงานของระบบร่วมกับบอร์ดไฟโอ (FiO Std) มีเซอร์โวมอเตอร์ 5 ตัว ขับเคลื่อนหุ่นยนต์ รวมทั้งสร้างระบบควบคุมหุ่นยนต์ผ่านสัญญาณบลูทูธ (Bluetooth) ด้วยโปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์ (Android) ซึ่งหุ่นยนต์ปีนเสามีรูปแบบการเคลื่อนที่ขึ้นเสาแบบขนานกับเสา

## 5.1 สรุปและวิจารณ์ผลการทดลอง

การสร้างหุ่นยนต์ปีนเสาแบบกึ่งอัตโนมัติควบคุมผ่านสัญญาณไร้สายนั้นสามารถสร้างขึ้นได้ตรงตามวัตถุประสงค์ที่ตั้งไว้ หุ่นยนต์ปีนเสา มีน้ำหนักรวม 1.4 กิโลกรัม ความกว้าง 18 เซนติเมตร ความยาว 75 เซนติเมตร ความสูงขณะจับยึด 33 เซนติเมตร มีจำนวนข้อหมุนอิสระทั้งหมด 5 ข้อหมุน โดยใช้เซอร์โวมอเตอร์ที่มีขนาดแรงบิด 12.2 kg.cm เป็นตัวขับเคลื่อนหุ่นยนต์ที่มีความสามารถในการปีนเสาโดยใช้สถิติจากการทดลอง และสามารถควบคุมผ่านโปรแกรมบนระบบปฏิบัติการแอนดรอยด์ได้ โดยปัจจัยสำคัญที่ทำให้หุ่นยนต์มีความสามารถในการปีนที่ดี คือ การเลือกใช้เซอร์โวมอเตอร์ที่มีกำลังมากทำให้สามารถรองรับการแสดงท่าทางที่ต้องใช้กำลังมากได้ การเพิ่มจำนวนข้อต่อทำให้สามารถแบ่งหน้าที่ของเซอร์โวมอเตอร์ส่วนหนึ่งเพื่อใช้ในการรักษาสมดุลแยกอิสระได้ ตลอดจนการพัฒนาท่าทางย่อยของการปีนเสาไฟให้สามารถเคลื่อนไหวด้วยความต่อเนื่องและนิ่มนวล

## 5.2 ปัญหาที่เกิดจากการทดลอง

**ตารางที่ 5.1** ปัญหาและวิธีการแก้ไข

| ข้อที่ | ปัญหา | วิธีการแก้ไข |
|---|---|---|
| 1 | การลื่นไถลลงเมื่อหุ่นยนต์ปีนขึ้นเสา | ติดตั้งแผ่นกันลื่น (Anti-Slip) การลื่นไถลเมื่อปีนขึ้นเสาจึงเกิดขึ้นน้อยลง |
| 2 | หุ่นยนต์มีการเคลื่อนไหวของส่วนต่าง ๆ อย่างรวดเร็ว | เปลี่ยนการเขียนโปรแกรมให้ค่อย ๆ เพิ่มหรือลดค่าลงทีละน้อย ทำให้หุ่นยนต์เคลื่อนไหวอย่างนุ่มนวล |

| ข้อที่ | ปัญหา | วิธีการแก้ไข |
|---|---|---|
| 3 | หุ่นยนต์มีการเบนตัวออกจากเสาขณะปีนขึ้นเสา ทำให้จุดศูนย์ถ่วงของหุ่นยนต์เกิดการเปลี่ยนแปลง เซอร์โวมอเตอร์จึงต้องออกแรงเยอะมากขึ้น | เปลี่ยนรูปแบบการเคลื่อนที่ให้มี 3 รูปแบบ คือ แบบเคลื่อนที่ทีละข้อต่อ แบบเคลื่อนที่ทีละหลายข้อต่อ และแบบเคลื่อนที่ขนานกับเสา ซึ่งได้ทำการทดลองหาการเคลื่อนที่ที่ดีสุด ผลการทดลองที่ได้คือ การเคลื่อนที่แบบขนานกับเสา เนื่องจากขณะเคลื่อนที่การเปลี่ยนแปลงของจุดศูนย์ถ่วงของหุ่นยนต์เกิดการเปลี่ยนแปลงไปน้อยมาก ทำให้เซอร์โวมอเตอร์สามารถยกตัวเองขึ้นได้ทุกครั้ง |
| 4 | หุ่นยนต์ปีนเสาใช้วัสดุจากเหล็กมีน้ำหนักมากเกินไป | เปลี่ยนวัสดุที่ใช้โครงสร้างหุ่นยนต์เป็นอะคริลิก (Acrylic) ซึ่งมีน้ำหนักเบากว่าและราคาถูกกว่า |

## 5.3 ข้อเสนอแนะในการแก้ไขปัญหาหากมีการดำเนินการต่อไป

### 5.3.1 ปัญหาและข้อเสนอแนะ

ตารางที่ 5.2 ปัญหาและข้อเสนอแนะ

| ข้อที่ | ปัญหา | ข้อเสนอแนะ |
|---|---|---|
| 1 | หุ่นยนต์ปีนเสาเป็นระบบควบคุมแบบวงเปิด (Open-Loop Control System) ต้องอาศัยการตัดสินใจและการคาดคะเนของมนุษย์ ซึ่งจะไม่มีความแม่นยำมากนัก เมื่อเกิดการลื่นไถลหรือมือจับจับเสาได้ไม่พอดี ค่าของพัลส์วิดมอดูเลชัน (Pulse-Width Modulation) ที่ต้องป้อนให้กับเซอร์โวมอเตอร์จะเปลี่ยนแปลงไป | ควรพัฒนาเป็นระบบควบคุมแบบวงปิด (Close-Loop Control System) เนื่องจากระบบควบคุมแบบวงปิดจะนำสัญญาณเอาต์พุตไปเปรียบเทียบกับสัญญาณอ้างอิงเพื่อวัดค่าความคลาดเคลื่อน และนำไปสร้างสัญญาณกระตุ้น ค่าความคลาดเคลื่อนของเอาต์พุตจึงมีน้อยลง |
| 2 | หุ่นยนต์มีความไม่สมดุลเนื่องจากการทำเองด้วยมือ มีความละเอียดของเฟืองไม่มากนัก และระนาบของมือจับทั้งสองไม่อยู่ใน | ใช้เครื่องมือที่มีประสิทธิภาพและแม่นยำในการทำโครงหุ่นยนต์ |

| ข้อที่ | ปัญหา | วิธีการแก้ไข |
|---|---|---|
| | ระนาบเดียวกัน | |
| 3 | วัสดุที่ใช้ทำโครงสร้างมีน้ำหนักค่อนข้างมาก | ใช้วัสดุที่มีน้ำหนักเบาลง |
| 4 | เซอร์โวมอเตอร์มีแรงบิดไม่เพียงพอที่จะยกหุ่นยนต์ทั้งตัว ทำให้การเคลื่อนที่บางครั้งมีการยกตัวไม่ขึ้น | เลือกใช้เซอร์โวมอเตอร์ที่มีกำลังสูงขึ้น |

5.3.2 ความคิดใหม่ (ขยายชิ้นงาน)

ติดตั้งกล้อง ทำให้หุ่นยนต์เคลื่อนที่ในระนาบ x ได้ขณะอยู่บนเสาและทำให้มือจับบนสามารถหมุนได้ เพื่อทำให้หุ่นยนต์สามารถปฏิบัติการต่าง ๆ บนเสาได้

5.3.3 ความรู้และวิธีที่จำเป็นต่อการทำชิ้นงานต่อ

การเขียนโปรแกรมประยุกต์ (Application) บนระบบปฏิบัติการแอนดรอยด์ (Android) ต้องอาศัยความรู้ภาษาจาวา (Java) โดยศึกษาเกี่ยวกับการใช้งานและการส่งข้อมูลบลูทูธในระบบปฏิบัติการแอนดรอยด์ และศึกษาโปรแกรมซิมูลิงค์ (Simulink) ในแมทแลบ (MATLAB) เกี่ยวกับบล็อกเซตราปิดเอสทีเอ็ม32 (RapidSTM32 Blockset)

# เอกสารอ้างอิง


[1] Ross L., Fardo S., Masterson J. and Towers R. (2011). **Robotics: Theory and Industrial Applications**. Tinley Park, Illinois: The Goodheart-Willcox Company, Inc.

[2] AppliCAD. (November 21, 2012). **Industrial Robot Type**. Retrieved November 27, 2012, from http://www.applicadthai.com/business/articles/industrial-robot-type

[3] Merlet J.P. (2005). **Parallel Robot (Solid Mechanics and Its Applications)**. Dordrecht, the Netherlands: Springer.

[4] Ecoist. (January 28, 2009). **Animal Locomotion: 10 Marvelous Means of Movement**. Retrieved October 2, 2012, from http://webecoist.momtastic.com/2009/01/28/strange-animal-movement-means-locomotion

[5] Rus D. (January, 2000). **Article**. Retrieved October 2, 2012, from http://www.researchgate.net/publication/242917375_The_Inchworm_Robot_A_MultiFunctional_System

[6] OMEGA. (n.d.). **Introduction to Grippers**. Retrieved December 4, 2012, from http://www.omega.com/prodinfo/grippers.html

[7] Kotay, K. and Rus D. (2000). The Inchworm Robot: A Multi-Functional System. **Autonomous Robots**. 8(1), 53-69.

[8] Mohankumar D. (n.d.). **Servo Motor Controller**. Retrieved December 10, 2013, from http://www.electroschematics.com/5335/servo-motor-controller

[9] Luecha S. and Kullawong A. (2012). **Intelligent Hexapod Robot**.

[10] Amax Industrial Group China. (2012). **ser-SR431 Micro digital high speed servo**. Retrieved January 8, 2013, from http://www.alibaba.com/product-gs/445803687/ser_SR431_Micro_digital_high_speed.html

[11] Tawiwat V. and Sarawut S. (November 25, 2008). **The Optimal Design for Grip Force of Materlal Handling**. 216-220.

[12] Tom Acop. (2010). **The Inch Worm**. Retrieved December 4, 2012, from http://www.flickr.com/photos/tomsflick/5356792486/sizes/l/in/photostream

[13] Zajac, T. Jr. (n.d.). **Robotic Gripper Sizing: The Science, Technology and Lore**. Retrieved November 26, 2012, from www.grippers.com/size.htm

[14] Williams K. (May 1, 2012). **3D Printed Robot Gripper**. Retrieved November 26, 2012, from http://www.kwartzlab.ca/2012/05/3d-printed-robot-gripper

[15] Aimagin. (n.d.). **Products**. Retrieved November 26, 2012, from https://www.aimagin.com/index.php/products.html

[16] Palowireless. (n.d.). **Bluetooth Tutorial – Specifications**. Retrieved November 26, 2012, from http://www.palowireless.com/infotooth/tutorial.asp

[17] MathWorks. (n.d.) **MATLAB The Language of Technical Computing**. Retrieved November 21, 2013, from http://www.mathworks.com/products/matlab

[18] Riley K. (November 15, 2000). **Torque and Angular Momentum**. Retrieved November 30, 2012, from http://www.paliden.com/powerpoint/SerFaughn8.ppt

# ภาคผนวก ก

# การทดลอง

**ตารางภาคผนวกที่ ก.1** แสดงค่าการทดลองบีบและคลายของมือจับ

| ครั้งที่ | บีบ | | คลาย | |
|---|---|---|---|---|
| | ความกว้าง (ซ.ม.) | PWM (% Duty Cycle) | ความกว้าง (ซ.ม.) | PWM (% Duty Cycle) |
| 1 | 0.0 | 5.9 | 15.6 | 2.0 |
| 2 | 0.0 | 6.0 | 15.5 | 2.0 |
| 3 | 0.0 | 6.0 | 15.5 | 2.0 |
| 4 | 0.0 | 6.0 | 15.7 | 2.0 |
| 5 | 0.0 | 6.0 | 15.5 | 2.0 |
| 6 | 0.0 | 6.0 | 15.5 | 2.0 |
| 7 | 0.0 | 6.0 | 15.5 | 2.0 |
| 8 | 0.0 | 6.0 | 15.5 | 2.0 |
| 9 | 0.0 | 6.0 | 15.5 | 2.0 |
| 10 | 0.0 | 6.0 | 15.5 | 2.0 |
| 11 | 0.0 | 5.9 | 15.6 | 2.0 |
| 12 | 0.0 | 6.0 | 15.6 | 2.0 |
| 13 | 0.0 | 6.0 | 15.6 | 2.0 |
| 14 | 0.0 | 6.0 | 15.5 | 2.0 |
| 15 | 0.0 | 6.0 | 15.5 | 2.0 |
| 16 | 0.0 | 6.0 | 15.7 | 2.0 |
| 17 | 0.0 | 6.0 | 15.6 | 2.0 |
| 18 | 0.0 | 6.0 | 15.5 | 2.0 |
| 19 | 0.0 | 6.0 | 15.5 | 2.0 |
| 20 | 0.0 | 6.0 | 15.5 | 2.0 |
| ค่าเฉลี่ย | 0.0 | 6.0 | 15.5 | 2.0 |

**ตารางภาคผนวกที่ ก.2** บันทึกค่ามุมและพัลส์วิดมอดูเลชันการกวาดมุมของข้อมือ

| ครั้งที่ | การกวาดมุม | |
|---|---|---|
| | มุม (องศา) | PWM (% Duty Cycle) |
| 1 | -90° ถึง 90° | 2 - 12 |
| 2 | -90° ถึง 90° | 2 - 12 |
| 3 | -90° ถึง 90° | 2 - 12 |
| 4 | -90° ถึง 90° | 2 - 12 |
| 5 | -90° ถึง 90° | 2 - 12 |
| 6 | -90° ถึง 90° | 2 - 12 |
| 7 | -90° ถึง 90° | 2 - 12 |
| 8 | -90° ถึง 90° | 2 - 12 |
| 9 | -90° ถึง 90° | 2 - 12 |
| 10 | -90° ถึง 90° | 2 - 12 |
| 11 | -90° ถึง 90° | 2 - 12 |
| 12 | -90° ถึง 90° | 2 - 12 |
| 13 | -90° ถึง 90° | 2 - 12 |
| 14 | -90° ถึง 90° | 2 - 12 |
| 15 | -90° ถึง 90° | 2 - 12 |
| 16 | -90° ถึง 90° | 2 - 12 |
| 17 | -90° ถึง 90° | 2 - 12 |
| 18 | -90° ถึง 90° | 2 - 12 |
| 19 | -90° ถึง 90° | 2 - 12 |
| 20 | -90° ถึง 90° | 2 - 12 |
| ค่าเฉลี่ย | -90° ถึง 90° | 2 - 12 |

**ตารางภาคผนวกที่ ก.3** แสดงค่าการทดลองการขยายตัวและการหดตัวของหุ่นยนต์

| ครั้งที่ | ยืดตัว | | หดตัว | |
|---|---|---|---|---|
| | มุม (องศา) | PWM (% Duty Cycle) | มุม (องศา) | PWM (% Duty Cycle) |
| 1 | 170° | 2.0 | 30° | 10.1 |
| 2 | 170° | 2.0 | 30° | 10.2 |
| 3 | 170° | 2.0 | 30° | 10.1 |

| ครั้งที่ | ยืดตัว | | หดตัว | |
|---|---|---|---|---|
| | มุม (องศา) | PWM (%) | มุม (องศา) | PWM (%) |
| 4 | 170° | 2.0 | 30° | 10.1 |
| 5 | 170° | 2.0 | 30° | 10.1 |
| 6 | 170° | 2.0 | 30° | 10.1 |
| 7 | 170° | 2.0 | 30° | 10.1 |
| 8 | 170° | 2.0 | 30° | 10.1 |
| 9 | 170° | 2.0 | 30° | 10.1 |
| 10 | 170° | 2.0 | 30° | 10.1 |
| 11 | 170° | 2.0 | 30° | 10.2 |
| 12 | 170° | 2.0 | 30° | 10.1 |
| 13 | 170° | 2.0 | 30° | 10.1 |
| 14 | 170° | 2.0 | 30° | 10.1 |
| 15 | 170° | 2.0 | 30° | 10.1 |
| 16 | 170° | 2.0 | 30° | 10.1 |
| 17 | 170° | 2.0 | 30° | 10.1 |
| 18 | 170° | 2.0 | 30° | 10.1 |
| 19 | 170° | 2.0 | 30° | 10.1 |
| 20 | 170° | 2.0 | 30° | 10.1 |
| ค่าเฉลี่ย | 170° | 2.0 | 30° | 10.1 |

**ตารางภาคผนวกที่ ก.4 การทดลองเพื่อดูสถิติแนวโน้มของค่าพัลส์วิดมอดูเลชันที่ป้อนให้กับ** เซอร์โวมอเตอร์

| ครั้งที่ | PWM (%) | ลำดับที่ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 1 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 8.9 | 8.9 | 8.9 | 10.0 | 10.0 | 10.8 | 10.8 | 10.8 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 2 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 6.4 | 6.4 |

| ครั้งที่ | PWM (%) | ลำดับที่ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.8 | 10.8 | 10.8 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 3 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 4 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.5 | 6.5 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.8 | 10.8 | 10.8 | 10.8 | 10.8 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 5 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 6 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 7 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.8 | 10.8 | 10.8 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 8 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.5 | 6.5 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |

| ครั้งที่ | PWM (%) | ลำดับที่ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 9 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.9 | 10.9 | 10.9 | 10.8 | 10.8 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 10 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 11 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 12 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.8 | 7.8 | 7.8 | 7.8 | 7.8 | 7.8 | 7.8 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 8.9 | 8.9 | 8.9 | 10.0 | 10.0 | 11.0 | 11.0 | 11.0 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 13 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.5 | 6.5 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 14 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.5 | 6.5 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.8 | 10.8 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 15 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |

| ครั้งที่ | PWM (%) | ลำดับที่ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 16 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 17 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.5 | 6.5 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.8 | 10.8 | 10.8 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 18 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.5 | 6.5 | 8.7 | 8.7 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 19 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |
| 20 | H1 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 8.0 |
| | S2 | 6.4 | 6.4 | 8.7 | 8.7 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 | 6.4 | 6.4 |
| | S3 | 2.5 | 2.5 | 2.5 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 5.7 | 2.5 | 2.5 | 2.5 | 2.5 |
| | S4 | 10.9 | 10.9 | 10.9 | 10.9 | 10.9 | 9.0 | 9.0 | 9.0 | 10.0 | 10.0 | 10.9 | 10.9 | 10.9 |
| | H5 | 7.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 |

**ตารางภาคผนวกที่ ก.5** ลำดับการปีนลงเสาและค่าพัลส์วิดมอดูเลชัน

| ลำดับ | ท่าทาง | PWM (%) | | | | |
|---|---|---|---|---|---|---|
| | | H1 | S2 | S3 | S4 | H5 |
| 1 | จับยึดเสาด้วยสองมือ (ท่าเริ่มต้น) | 8.0 | 6.4 | 2.5 | 10.9 | 7.0 |

| ลำดับ | ท่าทาง | PWM (%) | | | | |
|---|---|---|---|---|---|---|
| | | H1 | S2 | S3 | S4 | H5 |
| 2 | ปล่อยมือจับบน | 2.0 | 6.4 | 2.5 | 10.9 | 7.0 |
| 3 | ปรับข้อมือล่างเพื่อยกตัวออกจากเสา | 2.0 | 6.4 | 2.5 | 10.0 | 7.0 |
| 4 | หดตัวเข้าหากัน | 2.0 | 6.4 | 5.7 | 10.0 | 7.0 |
| 5 | ปรับข้อมือล่างเพื่อยกตัวเข้าหาเสา | 2.0 | 6.4 | 5.7 | 9.0 | 7.0 |
| 6 | ปรับข้อมือบนเตรียมจับยึด | 2.0 | 7.8 | 5.7 | 9.0 | 7.0 |
| 7 | บีบมือจับบน | 8.0 | 7.8 | 5.7 | 9.0 | 7.0 |
| 8 | ปล่อยมือจับล่าง | 8.0 | 7.8 | 5.7 | 9.0 | 2.0 |
| 9 | ปรับข้อมือบนเพื่อยกตัวออกจากเสา | 8.0 | 8.7 | 5.7 | 9.0 | 2.0 |
| 10 | ยืดตัวออก | 8.0 | 8.7 | 2.5 | 9.0 | 2.0. |
| 11 | ปรับข้อมือบนเพื่อยกตัวเข้าหาเสา | 8.0 | 6.4 | 2.5 | 10.9 | 2.0 |
| 12 | ปรับข้อมือล่างเตรียมจับยึด | 8.0 | 6.4 | 2.5 | 10.9 | 2.0 |
| 13 | บีบมือจับล่าง (ท่าเป้าหมาย) | 8.0 | 6.4 | 2.5 | 10.9 | 7.0 |

# ภาคผนวก ข

# Source Code โปรแกรม ECPR Controller

**res\layout\activity_ecpr.xml**

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#AAAAAA"
    android:orientation="vertical" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#55CCCC"
        android:paddingTop="10dp" >

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingBottom="10dp"
            android:paddingLeft="15dp"
            android:paddingRight="15dp"
            android:text="@string/control"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:textColor="#FFFFFF"
            android:textSize="22sp"
            android:textStyle="bold" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingLeft="20dp"
        android:paddingTop="10dp" >

        <TextView
            android:id="@+id/status"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/status"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/status2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingLeft="20dp"
            android:text="@string/status2"
            android:textColor="#0000CC" />

    </LinearLayout>

    <LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical"
    android:paddingBottom="10dp"
    android:paddingTop="30dp" >

    <Button
        android:id="@+id/btn_up"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:minWidth="80dp"
        android:text="@string/btn_up" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF"
    android:gravity="center_vertical|center_horizontal"
    android:paddingBottom="10dp"
    android:paddingTop="10dp" >

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:paddingRight="80dp" >

        <Button
            android:id="@+id/btn_lock"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:minWidth="80dp"
            android:text="@string/btn_lock" />

    </LinearLayout>

    <Button
        android:id="@+id/btn_unlock"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:minWidth="80dp"
        android:text="@string/btn_unlock" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="right"
    android:paddingRight="15dp" >

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:paddingRight="50dp" >

        <Button
            android:id="@+id/btn_down"
```

```
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:minWidth="80dp"
                android:text="@string/btn_down" />
        </LinearLayout>

        <Button
            android:id="@+id/btn_clear"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/btn_clear" />

    </LinearLayout>

</LinearLayout>
```

**res\layout\device_list.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <TextView android:id="@+id/title_paired_devices"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/paired"
        android:visibility="gone"
        android:background="#666"
        android:textColor="#fff"
        android:paddingLeft="5dp"
    />
    <ListView android:id="@+id/paired_devices"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:stackFromBottom="true"
        android:layout_weight="1"
    />
    <TextView android:id="@+id/title_new_devices"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/other"
        android:visibility="gone"
        android:background="#666"
        android:textColor="#fff"
        android:paddingLeft="5dp"
    />
    <ListView android:id="@+id/new_devices"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:stackFromBottom="true"
        android:layout_weight="2"
    />
    <Button android:id="@+id/button_scan"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/scan"
    />
</LinearLayout>
```

**res\layout\device_name.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:padding="5dp"
/>
```

**res\menu\ecpr.xml**

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/scan_device"
        android:icon="@android:drawable/ic_menu_search"
        android:title="@string/m_scan"/>

    <item
        android:id="@+id/discoverable"
        android:icon="@android:drawable/ic_menu_mylocation"
        android:title="@string/m_discoverable" />

    <item
        android:id="@+id/send"
        android:icon="@android:drawable/ic_media_play"
        android:title="@string/m_send" />

    <item
        android:id="@+id/stop"
        android:icon="@android:drawable/ic_media_pause"
        android:title="@string/m_stop" />

</menu>
```

**src\ECPR.java**

```
package com.pornchai.ecpr;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
```

```
import java.util.UUID;


import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.SystemClock;
import android.app.Activity;
import android.view.View.OnClickListener;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothServerSocket;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class EPCR extends Activity {
	private BluetoothAdapter mBluetoothAdapter = null;
	private ConnectThread mConnectThread;
	private ConnectedThread mConnectedThread;
	private AcceptThread mAcceptThread;

	private static final UUID MY_UUID = UUID.fromString("00001101-0000-
1000-8000-00805F9B34FB");

	private int mState = 0;

	private static final int REQUEST_ENABLE_BT = 3;
	private static final int REQUEST_CONNECT_DEVICE = 1;

	//name of device
	String mDeviceName = "Device";
	public static final String TOAST = "toast";
	// Constants for state
	public static final int MESSAGE_STATE_CHANGE = 1;
	public static final int MESSAGE_TOAST = 5;
	public static final int STATE_NONE = 0;
	public static final int STATE_LISTEN = 1;
	public static final int STATE_CONNECTING = 2;
	public static final int STATE_CONNECTED = 3;

	double runtime;
	long startTime;
	boolean SEND = true;

	TextView status;
	Button btnUp, btnDown, btnLock, btnUnlock, btnClc;

	double[] receivingDouble = {0.0, 0.0, 0.0};
	double value1 = 0;
	boolean RecieveUpdate = false;
	private Handler myHandler = new Handler();

	@Override
```

```
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_epcr);

		mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
		if(mBluetoothAdapter == null){
			Toast.makeText(this, "Bluetooth is not available",
Toast.LENGTH_LONG).show();
		}
	}

	protected synchronized void onStart(){
		super.onStart();

		if (!mBluetoothAdapter.isEnabled()) {
			Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
			startActivityForResult(enableBtIntent,
REQUEST_ENABLE_BT);
		} else {
			setupInterface();
		}
	}

	protected synchronized void onResume() {
		super.onResume();

	}

	@Override
	public synchronized void onPause() {
		super.onPause();
	}

	@Override
	public synchronized void onStop() {
		super.onStop();

		if (mConnectThread != null) {
			mConnectThread.cancel();
			mConnectThread = null;
		}

		if (mConnectedThread != null) {
			mConnectedThread.cancel();
			mConnectedThread = null;
		}

		if (mAcceptThread != null) {
			mAcceptThread.cancel();
			mAcceptThread = null;
		}

		setState(STATE_NONE);
	}

	private void setupInterface() {
		status = (TextView) findViewById(R.id.status2);
```

```
        btnUp = (Button) findViewById(R.id.btn_up);
        btnDown = (Button) findViewById(R.id.btn_down);
        btnLock = (Button) findViewById(R.id.btn_lock);
        btnUnlock = (Button) findViewById(R.id.btn_unLock);
        btnClc = (Button) findViewById(R.id.btn_clear);

        btnUp.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                if (mState == STATE_CONNECTED) {
                    value1 = 2;
                }
            }
        });

        btnDown.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                if (mState == STATE_CONNECTED) {
                    value1 = 3;

                }
            }
        });

        btnLock.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                if (mState == STATE_CONNECTED) {
                    value1 = 1;

                }
            }
        });

        btnUnlock.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                if (mState == STATE_CONNECTED) {
                    value1 = 4;

                }
            }
        });

        btnClc.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                if (mState == STATE_CONNECTED) {
                    value1 = 5;

                }
            }
        });

        if (mAcceptThread != null) {
            mAcceptThread.cancel();
            mAcceptThread = null;
        }

        mAcceptThread = new AcceptThread();
        mAcceptThread.start();
        setState(STATE_LISTEN);
    }
```

```
        private final Handler mHandler = new Handler() {

                public void handleMessage(Message msg) {
                        switch (msg.what) {
                        case MESSAGE_STATE_CHANGE:
                                switch (msg.arg1) {
                                case EPCR.STATE_CONNECTED:
                                        status.setText("Connected to ");
                                        status.append(mDeviceName);
                                        break;
                                case EPCR.STATE_CONNECTING:
                                        status.setText("Connecting.");
                                        break;
                                case EPCR.STATE_LISTEN:
                                        status.setText("Listening.");
                                        break;
                                case EPCR.STATE_NONE:
                                        status.setText("Not Connect.");
                                        break;
                                }
                                break;
                        case MESSAGE_TOAST:
                                Toast.makeText(getApplicationContext(),
msg.getData().getString(TOAST), Toast.LENGTH_SHORT).show();
                                break;
                        }
                }
        };

        private synchronized void setState(int state) {
                mState = state;
                mHandler.obtainMessage(EPCR.MESSAGE_STATE_CHANGE, state, -
1).sendToTarget();
        }

        @Override
        public synchronized void onDestroy() {
                super.onDestroy();

                if (mBluetoothAdapter != null) {
                        mBluetoothAdapter = null;
                        Toast.makeText(this, "Exiting.",
Toast.LENGTH_SHORT).show();
                }
        }

        private void ensureDiscoverable() {
                if (mBluetoothAdapter.getScanMode() !=
BluetoothAdapter.SCAN_MODE_CONNECTABLE_DISCOVERABLE) {
                        Intent discoverableIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);

        discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURA
TION, 300);
                        startActivity(discoverableIntent);
                }
        }
```

```
	public void onActivityResult(int requestCode, int resultCode, Intent
data) {
		switch (requestCode) {
		case REQUEST_CONNECT_DEVICE:
			if (resultCode == Activity.RESULT_OK) {
				connectDevice(data);
			}
			break;
		case REQUEST_ENABLE_BT:
			if (resultCode == Activity.RESULT_OK) {
				setupInterface();
			} else {
				Toast.makeText(this, "Bluetooth not enabled,
Leaving...", Toast.LENGTH_SHORT).show();
				finish();
			}
		}
	}

	private synchronized void connectDevice(Intent data) {
		String address =
data.getExtras().getString(DeviceList.EXTRA_DEVICE_ADDRESS);
		BluetoothDevice device =
mBluetoothAdapter.getRemoteDevice(address);
		mDeviceName = device.getName();

		if (mState == STATE_CONNECTING) {
			if (mConnectThread != null) {
				mConnectThread.cancel();
				mConnectThread = null;
			}
			if (mConnectedThread != null) {
				mConnectedThread.cancel();
				mConnectedThread = null;
			}
		}

		mConnectThread = new ConnectThread(device);
		mConnectThread.start();

		setState(STATE_CONNECTING);
	}

	@Override
	public boolean onCreateOptionsMenu(Menu menu) {
		// Inflate the menu; this adds items to the action bar if it
is present.
		getMenuInflater().inflate(R.menu.epcr, menu);
		return true;
	}

	@Override
	public boolean onOptionsItemSelected(MenuItem item) {
		Intent serverIntent = null;
		switch (item.getItemId()) {
		case R.id.scan_device:
			serverIntent = new Intent(this, DeviceList.class);
			startActivityForResult(serverIntent,
REQUEST_CONNECT_DEVICE);
```

```
                return true;
            case R.id.discoverable:
                ensureDiscoverable();
                return true;
            case R.id.send: //Start sending and receiving
            if (mState == STATE_CONNECTED) {
                SEND = true;
                startTime = SystemClock.uptimeMillis();
                myHandler.removeCallbacks(mPlottingTask);
                myHandler.postDelayed(mPlottingTask, 10); //wait 10
milisec to start
            }
            return true;
        case R.id.stop: //stop sending and receiving
            myHandler.removeCallbacks(mPlottingTask);
            return true;
                }

            return false;
        }

        private Runnable mPlottingTask = new Runnable() {  //Sending Task
and time updating
                public void run() {
                        long now = SystemClock.uptimeMillis();

                    if (SEND) {
                                byte[] sendingbytes = new byte[12];
//byte array for packet

                                ByteBuffer buff2 =
ByteBuffer.wrap(sendingbytes).order(ByteOrder.LITTLE_ENDIAN); //wrapping
byte array for modification
                                buff2.put((byte) 0x7e); //Header 7E7E
                                buff2.put((byte) 0x7e);
                                buff2.putDouble((byte) value1); //three
double values going in byte array
                                buff2.put((byte) 0x03); //Closing 0303
                                buff2.put((byte) 0x03);

                                if (mState == STATE_CONNECTED)
mConnectedThread.write(sendingbytes);
                    }
                     myHandler.postAtTime(this, now + 10); // call every
10 ms (100 Hz)
                }
        };

        public synchronized void connect(BluetoothDevice device) {
          // Cancel any thread attempting to make a connection
          if (mState == STATE_CONNECTING) {
              if (mConnectThread != null) {mConnectThread.cancel();
mConnectThread = null;}
          }
          // Cancel any thread currently running a connection
          if (mConnectedThread != null) {mConnectedThread.cancel();
mConnectedThread = null;}
          // Start the thread to connect with the given device
          mConnectThread = new ConnectThread(device);
```

```
        mConnectThread.start();
        setState(STATE_CONNECTING);
    }

    public synchronized void connected(BluetoothSocket socket,
BluetoothDevice device) {
        // Cancel the thread that completed the connection
        if (mConnectThread != null) {mConnectThread.cancel();
mConnectThread = null;}
        // Cancel any thread currently running a connection
        if (mConnectedThread != null) {mConnectedThread.cancel();
mConnectedThread = null;}
        // Cancel the accept thread because we only want to connect to one
device
        if (mAcceptThread != null) {
            mAcceptThread.cancel();
            mAcceptThread = null;
        }
        // Start the thread to manage the connection and perform
transmissions
        mConnectedThread = new ConnectedThread(socket);
        mConnectedThread.start();
        mDeviceName = device.getName();
        setState(STATE_CONNECTED);
    }

    public void connectionFailed() {
        Message msg = mHandler.obtainMessage(EPCR.MESSAGE_TOAST);
        Bundle bundle = new Bundle();
        bundle.putString(EPCR.TOAST, "Device connection failed.");
        msg.setData(bundle);
        mHandler.sendMessage(msg);

        if (mAcceptThread != null) {
            mAcceptThread.cancel();
            mAcceptThread = null;
        }
        mAcceptThread = new AcceptThread();
        mAcceptThread.start();

        setState(STATE_LISTEN);
    }

    public void connectionLost() {
        Message msg = mHandler.obtainMessage(EPCR.MESSAGE_TOAST);
        Bundle bundle = new Bundle();
        bundle.putString(EPCR.TOAST, "Device connection was lost.");
        msg.setData(bundle);
        mHandler.sendMessage(msg);

        if (mAcceptThread != null) {
            mAcceptThread.cancel();
            mAcceptThread = null;
        }

        setState(STATE_LISTEN);
    }

    private class AcceptThread extends Thread {
```

```
        private final BluetoothServerSocket mmServerSocket;

    public AcceptThread() {
        BluetoothServerSocket tmp = null;
            // Create a new listening server socket
         try {
              tmp =
mBluetoothAdapter.listenUsingRfcommWithServiceRecord("BlueIO", MY_UUID);
         } catch (IOException e) {

         }
         mmServerSocket = tmp;
    }

    public void run() {
        BluetoothSocket socket = null;
        while (mState != STATE_CONNECTED) {
             try {
                 // This is a blocking call and will only return on
a successful connection or an exception
                 socket = mmServerSocket.accept();
             } catch (IOException e) {
                 break;
             }
             if (socket != null) {
                 synchronized (this) {
                     switch (mState) {
                     case STATE_LISTEN:
                     case STATE_CONNECTING:
                     // Situation normal. Start the connected thread.
                         connected(socket,
socket.getRemoteDevice());
                         break;
                     case STATE_NONE:
                     case STATE_CONNECTED:
                     // Either not ready or already connected. Terminate
new socket.
                         try {
                             socket.close();
                         } catch (IOException e) {

                         }
                         break;
                     }
                 }
             }
        }
    }

    public void cancel() {
        try {
             mmServerSocket.close();
         } catch (IOException e) {

         }
    }
}
```

```
        private class ConnectThread extends Thread {
                private final BluetoothSocket mmSocket;
            private final BluetoothDevice mmDevice;

                public ConnectThread(BluetoothDevice device) {
                        mmDevice = device;
                        BluetoothSocket tmp = null;
                        // Get a BluetoothSocket for a connection with the given
BluetoothDevice
                        try {
                                tmp =
device.createRfcommSocketToServiceRecord(MY_UUID);
                        } catch (IOException e) {

                        }
                        mmSocket = tmp;
                }

                public void run() {
                    // Always cancel discovery because it will slow down a
connection
                    mBluetoothAdapter.cancelDiscovery();
                        // Make a connection to the BluetoothSocket
                        try {
                            // This is a blocking call and will only return on a
successful connection or an exception
                            mmSocket.connect();
                        } catch (IOException e) {
                            // Close the socket
                            try {
                                mmSocket.close();
                            } catch (IOException e2) {

                            }
                            connectionFailed();
                            return;
                        }

                        // Reset the ConnectThread because we're done
                        synchronized (this) {
                            mConnectThread = null;
                        }

                        // Start the connected thread
                        connected(mmSocket, mmDevice);
                    }

                public void cancel() {
                        try {
                            mmSocket.close();
                        } catch (IOException e) {

                        }
                    }
        }

        /**
        * This thread runs during a connection with a remote device.
        * It handles all incoming and outgoing transmissions.
```

```
*/
private class ConnectedThread extends Thread {
      private final BluetoothSocket mmSocket;
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;

    public ConnectedThread(BluetoothSocket socket) {

       mmSocket = socket;
        InputStream tmpIn = null;
        OutputStream tmpOut = null;

        // Get the BluetoothSocket input and output streams
        try {
             tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch (IOException e) {

        }

        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }

    public void run() {

        byte[] buffer = new byte[28]; //default 1024
        int bytes;

        // Keep listening to the InputStream while connected
        while (true) {
            try {
                // Read from the InputStream
                    bytes = mmInStream.read(buffer);
                    // processByte(buffer, bytes);
                    if (bytes==28) {
                                if ((buffer[0]==(byte)
0x7e)&&(buffer[1]==(byte) 0x7e)) {
                                        if ((buffer[27]==(byte)
0x03)&&(buffer[26]==(byte) 0x03)) {
                                                ByteBuffer buff1 =
ByteBuffer.wrap(buffer).order(ByteOrder.LITTLE_ENDIAN);   //wrap byte
packet for processing

                                                for (int i = 0; i <
receivingDouble.length; i++)

      receivingDouble[i] = buff1.getDouble(2 + i*8);  // Skipping first 2
header bytes

                                                RecieveUpdate = true;
                                        }
                                }
                        }

                } catch (IOException e) {
                    connectionLost();
                 break;
                }
```

```
            }
        }

        /**
         * Write to the connected OutStream.
         * @param buffer  The bytes to write
         */
        public void write(byte[] buffer) {
            try {
                mmOutStream.write(buffer);
            } catch (IOException e) {

            }
        }

        public void cancel() {
            try {
                mmInStream.close();
            } catch (IOException e2) {
                e2.printStackTrace();
            }

            try {
                mmOutStream.flush();
            } catch (IOException e1) {
                e1.printStackTrace();
            }

            try {
                mmSocket.close();
            } catch (IOException e) {

            }
        }
    }
}
```

src\DeviceList.java

```
package com.pornchai.ecpr;

import java.util.Set;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.view.View;
import android.view.Window;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
```

```
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

public class DeviceList  extends Activity {

      public static String EXTRA_DEVICE_ADDRESS = "device_address";
      private BluetoothAdapter mBtAdapter;
      private ArrayAdapter<String> mPairedDevicesArrayAdapter;
    private ArrayAdapter<String> mNewDevicesArrayAdapter;

      protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            requestWindowFeature(Window.FEATURE_INDETERMINATE_PROGRESS);
        setContentView(R.layout.device_list);
        // Set result CANCELED incase the user backs out
        setResult(Activity.RESULT_CANCELED);

        Button scanButton = (Button) findViewById(R.id.button_scan);
        scanButton.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                doDiscovery();
                v.setVisibility(View.GONE);
            }
        });

        mPairedDevicesArrayAdapter = new ArrayAdapter<String>(this,
R.layout.device_name);
        mNewDevicesArrayAdapter = new ArrayAdapter<String>(this,
R.layout.device_name);

    // Find and set up the ListView for paired devices
        ListView pairedListView = (ListView)
findViewById(R.id.paired_devices);
        pairedListView.setAdapter(mPairedDevicesArrayAdapter);
        pairedListView.setOnItemClickListener(mDeviceClickListener);

        // Find and set up the ListView for newly discovered devices
        ListView newDevicesListView = (ListView)
findViewById(R.id.new_devices);
        newDevicesListView.setAdapter(mNewDevicesArrayAdapter);
        newDevicesListView.setOnItemClickListener(mDeviceClickListener);

    // Register for broadcasts when a device is discovered
        IntentFilter filter = new
IntentFilter(BluetoothDevice.ACTION_FOUND);
        this.registerReceiver(mReceiver, filter);

        // Register for broadcasts when discovery has finished
        filter = new
IntentFilter(BluetoothAdapter.ACTION_DISCOVERY_FINISHED);
        this.registerReceiver(mReceiver, filter);

        mBtAdapter = BluetoothAdapter.getDefaultAdapter();
        Set<BluetoothDevice> pairedDevices = mBtAdapter.getBondedDevices();
    // If there are paired devices, add each one to the ArrayAdapter
        if (pairedDevices.size() > 0) {
```

```
findViewById(R.id.title_paired_devices).setVisibility(View.VISIBLE);
            for (BluetoothDevice device : pairedDevices) {
                mPairedDevicesArrayAdapter.add(device.getName() + "\n" +
device.getAddress());
            }
        } else {
            String noDevices = "No devices paired";
            mPairedDevicesArrayAdapter.add(noDevices);
        }

    }

    @Override
    protected void onDestroy() {
        super.onDestroy();

        // Make sure we're not doing discovery anymore
        if (mBtAdapter != null) {
            mBtAdapter.cancelDiscovery();
        }

        // Unregister broadcast listeners
        this.unregisterReceiver(mReceiver);
    }

    /**
     * Start device discover with the BluetoothAdapter
     */
    private void doDiscovery() {
        //if (D) Log.d(TAG, "doDiscovery()");

        // Indicate scanning in the title
        setProgressBarIndeterminateVisibility(true);
        setTitle("Scanning");

        // Turn on sub-title for new devices
        findViewById(R.id.title_new_devices).setVisibility(View.VISIBLE);

        // If we're already discovering, stop it
        if (mBtAdapter.isDiscovering()) {
            mBtAdapter.cancelDiscovery();
        }

        // Request discover from BluetoothAdapter
        mBtAdapter.startDiscovery();
    }

    // The on-click listener for all devices in the ListViews
    private OnItemClickListener mDeviceClickListener = new
OnItemClickListener() {
        public void onItemClick(AdapterView<?> av, View v, int arg2, long
arg3) {
            // Cancel discovery because it's costly and we're about to
connect
            mBtAdapter.cancelDiscovery();

            // Get the device MAC address, which is the last 17 chars in
the View
```

```
            String info = ((TextView) v).getText().toString();
            String address = info.substring(info.length() - 17);
            Toast.makeText(getApplicationContext(), "ADDRESS = " + address,
Toast.LENGTH_SHORT).show(); // good

            // Create the result Intent and include the MAC address
            Intent intent = new Intent();
            intent.putExtra(EXTRA_DEVICE_ADDRESS, address);

            // Set result and finish this Activity
            setResult(Activity.RESULT_OK, intent);
            finish();
        }
    };

 // The BroadcastReceiver that listens for discovered devices and
    // changes the title when discovery is finished
    private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String action = intent.getAction();

            // When discovery finds a device
            if (BluetoothDevice.ACTION_FOUND.equals(action)) {
                // Get the BluetoothDevice object from the Intent
                BluetoothDevice device =
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
                // If it's already paired, skip it, because it's been
listed already
                if (device.getBondState() != BluetoothDevice.BOND_BONDED) {
                    mNewDevicesArrayAdapter.add(device.getName() + "\n" +
device.getAddress());
                }
            // When discovery is finished, change the Activity title
            } else if
(BluetoothAdapter.ACTION_DISCOVERY_FINISHED.equals(action)) {
                setProgressBarIndeterminateVisibility(false);
                setTitle("Select device to connect");
                if (mNewDevicesArrayAdapter.getCount() == 0) {
                    String noDevices = "No device found";
                    mNewDevicesArrayAdapter.add(noDevices);
                }
            }
        }
    };

}
```

**res\values\strings.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">ECPR Controller</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
    <string name="status">Status :</string>
```

```
    <string name="status2">Intializating . . .</string>
    <string name="m_discoverable">Discoverable</string>
    <string name="scan">SCAN</string>
    <string name="m_scan">Scan</string>
    <string name="paired">Paired</string>
    <string name="other">Other</string>
    <string name="control">Electric Pole Climbing Robot Controller</string>

    <string name="m_send">Send</string>
    <string name="m_stop">Stop</string>

    <!-- UI -->
    <string name="btn_up">UP</string>
    <string name="btn_down">Down</string>
    <string name="btn_lock">Lock</string>
    <string name="btn_unlock">Unlock</string>
    <string name="btn_clear">Clear</string>

</resources>
```

**AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.pornchai.ecpr"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.pornchai.ecpr.EPCR"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
        <activity
            android:name=".DeviceList"
            android:label="Select Device"
            android:theme="@android:style/Theme.Dialog"
            android:configChanges="orientation|keyboardHidden" />
    </application>

</manifest>
```