รายงานการวิจัยฉบับสมบูรณ์

การพัฒนาการตรวจจับการเคลื่อนไหว 3 มิติ เพื่อควบคุมหุ่นยนต์กู้ภัย เพื่อใช้ใน 3 จังหวัดชายแดนภาคใต้

Development of 3D motion detection for controlling rescue robot for use in the three southernmost provinces in Thailand

ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ

ได้รับทุนสนับสนุนงานวิจัยจากเงินงบประมาณ ประจำปีงบประมาณ 2558

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รายงานการวิจัยฉบับสมบูรณ์

การพัฒนาการตรวจจับการเคลื่อนไหว 3 มิติ เพื่อควบคุมหุ่นยนต์กู้ภัย เพื่อใช้ใน 3 จังหวัดชายแดนภาคใต้

Development of 3D motion detection for controlling rescue robot for use in the three southernmost provinces in Thailand

ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ

ได้รับทุนสนับสนุนงานวิจัยจากเงินงบประมาณ ประจำปีงบประมาณ 2558

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

RCH
ก 366ก
2558

เลขหมู่.....
เลขทะเบียน 143712
วัน.เดือน.ปี 29 ก.ย. 2559

b 12802177
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่องานวิจัย: การพัฒนาการตรวจจับการเคลื่อนไหว 3 มิติ เพื่อควบคุมหุ่นยนต์กู้ภัย เพื่อใช้ใน 3 จังหวัดชายแดนภาคใต้

นักวิจัย: ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ

คณะ: วิศวกรรมศาสตร์ **สาขาวิชา:** วิศวกรรมอิเล็กทรอนิกส์

บทคัดย่อ

โครงงานนี้นำเสนอการควบคุมการเคลื่อนไหวของหุ่นยนต์ด้วยกล้องตรวจจับการเคลื่อนไหว(Kinect) โดยการทำงานแบ่งออกเป็น 3 ส่วน ส่วนแรก จับลักษณะการเคลื่อนไหวของกระดูก Skeleton Tracking โดยใช้กล้องตรวจจับการเคลื่อนไหว(Kinect) ส่วนสอง เขียน Software สั่งการทำงานผ่านโปรแกรม Visual Studio 2010 เพื่อไปติดต่อกับ Kinect โดยที่เราจะออกแบบให้หุ่นยนต์จะเคลื่อนไหวตามลักษณะท่าทางของผู้สั่งการ ทั้งนี้ผู้จัดทำมีจุดประสงค์เพื่อเป็นหุ่นยนต์ต้นแบบในการปฏิบัติงานที่มีความเสี่ยงในด้านความปลอดภัย อาทิเช่น การเก็บกู้วัตถุระเบิดใน 3 จังหวัดชายแดนภาคใต้

คำสำคัญ: หุ่นยนต์, ระบบตรวจจับการเคลื่อนไหวแบบ 3 มิติ, การเก็บกู้วัตถุระเบิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Research Title: **Development of 3D motion detection for controlling rescue robot for use in the three southernmost provinces in Thailand**

Researcher: Asst.Prof.Dr.Pattarapong Phasukkit

Faculty: Engineering **Department:** Electronic Engineering

ABSTRACT

This project presents a motion control Robot by Kinect .The work is divided into three parts. The first caught the movement of Skeleton Tracking using camera motion detection (Kinect). Second software written to work through the program Visual Studio 2010 to interface with the Kinect. We will design the robot can move by a man by catch motion of human . The preparation of project is intended to be a prototype robot in security performance . Such as for Explosive Ordnance Disposal (EOD) use for recover bomb and survey risk Area

Keywords : ROBOT , KINECT, Explosive Ordnance Disposal (EOD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การวิจัยครั้งนี้ได้รับทุนสนับสนุนการวิจัยจากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จากแหล่งทุนวิจัยเงินงบประมาณ ประจำปีงบประมาณ พ.ศ 2558

ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 1

1.3 ขอบเขตการวิจัย 2

1.4 แผนการดำเนินงาน 2

บทที่ 2 ทฤษฎีและหลักการ 3

2.1 Kinect 4

2.1.1 ความสามารถของตัว Kinect 8

2.2 ความรู้เบื้องต้นเกี่ยวกับไมโครคอนโทรเลอร์ 9

2.2.1 Arduino Mega 2560 R3 11

2.3 Servo Motor 15

2.4 โปรแกรมที่ใช้ในการทำงาน คือ Microsoft Visual Studio 2010 โดยภาษาที่ใช้เขียนจะเป็น C# 17

2.5 Skelton Tracking 18

2.6 เครื่องพิมพ์ 3 มิติ (3D Printers) 20

บทที่ 3 การออกแบบระบบ 40

3.1 การออกแบบโปรแกรม 44

3.2 สร้างหุ่นยนต์ 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลอง 60

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 117

อ้างอิง 118

ภาคผนวก 119

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 Kinect Sensor 3

2.2 ระบบภายในของ Kinect Sensor 4

2.3 รายละเอียดของ Kinect 5

2.4 การทำงานของเซนเซอร์ 6

2.5 ตัวอย่างการทำงานของ Kinect 7

2.6 แสดงขอบเขตของการรับเสียงของตัว Kinect 8

2.7 แสดงองศาในการหมุนของ Kinect 8

2.8 แสดงระยะในการจับผู้ใช้งาน 9

2.9 Arduino Mega 2560 R3 Front 11

2.10 Arduino Mega 2560 R3 Back 11

2.11 แสดง Port ต่าง ๆ บน Arduino Mega 2560 R3 14

2.12 เซอร์โวมอเตอร์ 15

2.13 การส่งสัญญาณควบคุมเซอร์โวมอเตอร์ 16

2.14 รูปตัวอย่าง PWM 17

2.15 ตัวอย่างโปรแกรม Microsoft Visual Studio 2010 17

2.16 Skelton 18

2.17 รูปแสดงข้อต่อของ Skelton 19

2.18 เปรียบเทียบข้อต่อระหว่างท่ายืนและท่านั่ง 19

2.19 โปรแกรมสร้างแบบจำลอง 3 มิติ 21

2.20 พิมพ์ชิ้นงานด้วยเครื่องพิมพ์ 3 มิติ 22

3.1 ผังการทำงานของระบบ 24

3.2 ผังการทำงานของโปรแกรม 25

3.3 ผังการทำงานของโปรแกรมส่วนของมือซ้าย 26

3.4 ผังการทำงานของโปรแกรมส่วนของมือขวา 27

3.5 ผังการทำงานของโปรมแกรมการเคลื่อนไหวข้อศอกของแขนขวา 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.6 ผังการทำงานของโปรมแกรมการเคลื่อนไหวข้อศอกของแขนซ้าย ..29
3.7 เครื่องพิมพ์ 3 มิติ ..30
3.8 หน้าต่างโปรแกรมตั้งค่าก่อนการพิมพ์ชิ้นงาน ..31
3.9 ส่วนประกอบของหุ่นยนต์ที่ยังไม่ได้ประกอบหลังพิมพ์เสร็จ ..31
3.10 ส่วนลำตัวของหุ่นยนต์ที่ได้ประกอบแล้ว ..32
3.11 ส่วนขาของหุ่นยนต์ที่ได้ประกอบแล้ว ..32
3.12 หุ่นยนต์ที่ประกอบเสร็จสมบูรณ์แล้ว ..33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันเทคโนโลยีมีความก้าวหน้าไปมากแล้วได้เข้ามามีส่วนในการดำเนินชีวิตของเรามากขึ้น ทางด้านผู้จัดทำจึงเห็นว่าหุ่นยนต์ก็เป็นเทคโนโลยีที่เราน่าจะนำใช้ให้เกิดประโยชน์สูงสุดได้และน่าจะเข้ามามีส่วนช่วยในการทำงานของได้ อย่างงานที่มีความอันตรายต่อชีวิตของผู้ปฏิบัติหน้าที่ ซึ่งในตอนนี้ทางด้านสามจังหวัดชายแดนภาคใต้มีเหตุการณ์ที่อันตรายต่อชีวิตของประชาชนที่ได้อาศัยอยู่และเจ้าหน้าที่ที่เข้าไปปฏิบัติหน้าที่ในบริเวณที่เกิดเหตุ พวกเราจึงได้คิดว่าจะนำหุ่นยนต์เข้ามาปฏิบัติหน้าที่แทนคนในงานที่เสี่ยงอันตราย เช่น การเก็บกู้วัตถุระเบิดในสามจังหวัดชายแดนภาคใต้ โดยเราจะบังคับหุนยนต์ให้เคลื่อนที่ตามคนเราโดยใช้อุปกรณ์ที่มีความสามารถในการตรวจจับที่มีความแม่นยำที่มีชื่อว่า Kinect เป็นอุปกรณ์เสริมของเคลื่อนเล่น XBOX ทางด้านผู้จัดทำจึงคิดว่าจะนำเอา Kinect มาตรวจจับการเคลื่อนไหวของมนุษย์แล้วส่งค่าไปยังหุ่นยนต์ให้เคลื่อนไหวตามที่เราต้องการ

1.2 วัตถุประสงค์

1. เพื่อให้นักศึกษาได้มีความรู้ความเข้าใจในการเขียนโปรแกรม Microsoft Visual Studio 2010
2. ศึกษาการทำงานของ Kinect เพื่อเป็นแนวทางในการนำไปประยุกต์ทางด้านอื่นที่เหมาะสม
3. เพื่อเป็นหุ่นยนต์ต้นแบบในการปฏิบัติงานที่มีความเสี่ยงในด้านความปลอดภัย อาทิเช่น การเก็บกู้วัตถุระเบิดใน 3 จังหวัดชายแดนภาคใต้

1.3 ขอบเขตของโครงงาน

โครงงานนี้เป็นการควบคุมหุ่นยนต์ช่วงบนด้วย Kinect โดยที่เราจะเขียนโปรแกรมบนคอมพิวเตอร์โดยใช้โปรแกรม Microsoft Visual Studio 2010 เพื่อเรียกใช้ค่าฟังก์ชันต่าง ๆ ที่ได้จาก SDK (Software Development Kit) ของ Kinect แล้วส่งต่อให้บอร์ด Arduino เพื่อเปลี่ยนค่าจากฟังก์ชันที่เป็นตัวเลขให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น Pulse ไปควบคุม Servo ในหุ่นยนต์ โดยช่วงบนของหุ่นยนต์นั้นจะเคลื่อนไหวตามที่เราต้องการในขณะที่ผู้ควบคุมเคลื่อนไหวร่างกาย

1.4 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ม.ค.2557	ก.พ.2557	มี.ค.2557	เม.ย.2557
ศึกษาเกี่ยวกับส่วนประกอบของหุ่นยนต์	⟷	⟷		
เขียน code แก้ไขข้อบกพร่อง/ทดลอง		⟷	⟷	
รวบรวมข้อมูลทำรายงาน/สรุปผล			⟷	⟷

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 Kinect

Kinect คือ อุปกรณ์เสริมของเครื่องเล่นเกมส์ Xbox360 จาก Microsoft โดยสามารถจดจำผู้เล่น (Facial Recognition) และให้ผู้เล่นควบคุมเกมส์ผ่านทางการเคลื่อนไหวร่างกายของผู้เล่นโดยตรง (3D Motion Recognition) โดยไม่จำเป็นต้องมี joystick อีกต่อไป และสามารถจดจำเสียงของผู้เล่นได้ (Voice Recognition) และนอกจากใช้ฟังค์ชั่นพวกนี้เล่นเกมส์แล้วเรายังใช้Kinectควบคุมการดูหนัง ฟังเพลง เล่น Window live messenger ได้ด้วย

รูปที่ 2.1 Kinect Sensor

โครงสร้างของ Kinect (Kinect Sensor) จะมีลักษณะเป็นแท่งแบนๆ เพื่อใช้ในการวางในแนวราบที่เชื่อมต่อกับฐานขนาดเล็กที่มีมอเตอร์เพื่อใช้ในการเคลื่อนไหวในการปรับมุมกล้อง โดยภายในจะประกอบด้วย

1.กล้อง CMOS RGB 1ตัว

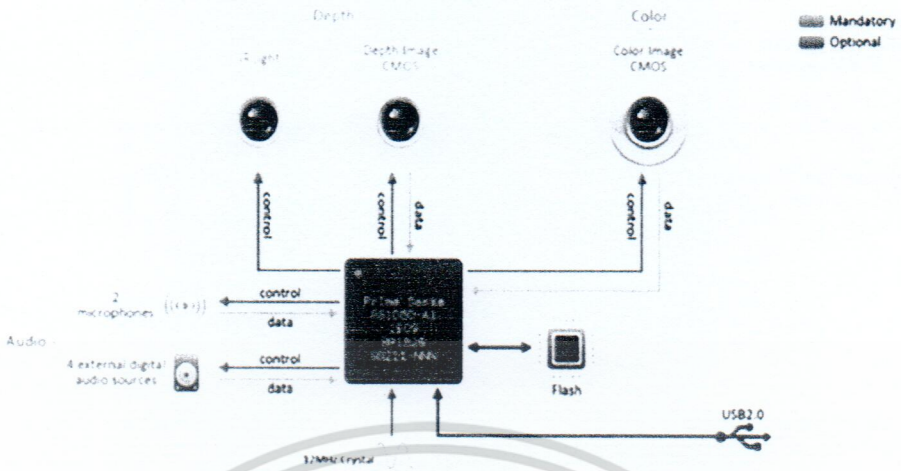
2.กล้อง IR (Infrared) 1 ตัว

3.ตัวส่งแสง IR (Infrared)

4.ไมโครโฟน 4 ตัว

5.มอเตอร์ไว้ปรับมุมกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ระบบภายในของ Kinect Sensor

Kinect มีการหน้าที่ใหญ่ๆอยู่ 3 หน้าที่ ดังนั้นเราจะเชื่อมระหว่าง Hardware กับหน้าที่การทำงานของ Kinect ดังนี้

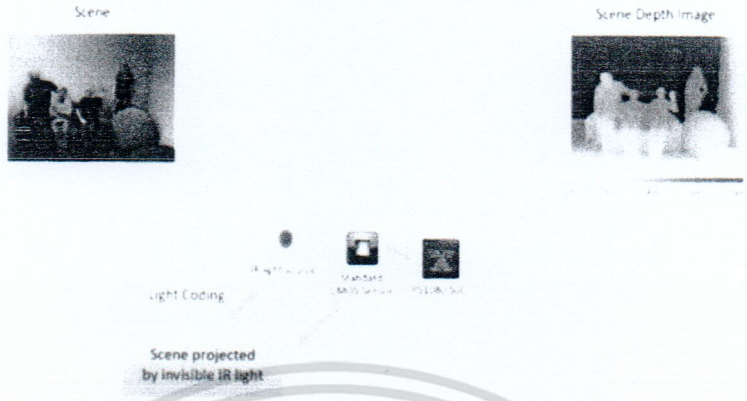
1.จดจำผู้เล่นได้โดยใช้ ข้อมูลจากกล้อง CMOS RGB (ข้อมูล แดง เขียว น้ำเงิน) และประมวลผล

2.จดจำการเคลื่อนไหวร่างกายของผู้เล่นเป็นแบบ 3 มิติแบ่งเป็น 3 อย่างย่อย

2.1 จดจำวัตถุแบบ 3 มิติ : โดยใช้ตัวส่งแสง IR และกล้อง IR , ตัวส่งแสง IR จะส่งแสงไปกระทบกับวัตถุ ไม่ว่าจะเป็นคนหรือสิ่งของ และแสง IR จะสะท้อนจากวัตถุนั้นๆกลับไปที่กล้อง IR และ Kinect จะใช้ข้อมูลพวก “เวลาในการสะท้อนกลับ”และ“ความยาวคลื่นแสง” ในการประมวลผลวัตถุต่างๆเป็น 3 มิติ (ยกตัวอย่างเช่น ใช้เวลานานในการสะท้อนกลับแสดงว่าวัตถุอยู่ไกล,ของที่มีสีต่างกันจะดูดกลืนแสงที่พบและสะท้อนแสงกลับไปในความยาวคลื่นที่ต่างกัน

2.2 จดจำและแบ่งแยกประเภทของคน ว่าเป็นเพศไหน อายุประมาณเท่าไหร่ ขนาดสัดส่วนตัวประมาณไหนเท่าไร ใส่เสื้อผ้าหรือยัง โดยใช้เปรียบเทียบกับฐานข้อมูลที่มีอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ตัวอย่างการทำงานของ Kinect

2.3 จดจำการเคลื่อนไหวของคน โดยหลังจากแบ่งแยกประเภทของคนแล้ว จะใช้วิธี skeletal movement (การเคลื่อนไหวตามลักษณะกระดูก) เข้าไปวิเคราะห์คนประเภทนั้นว่า ส่วนไหน คือหัว ตัวคอหัวไหล่ แขนขวาแขนซ้าย ขาขวาขาซ้าย ข้อศอก, โดยใช้สัดส่วน ลักษณะ หรือข้อจำกัดต่างๆ 3. จดจำเสียงผู้เล่น โดยใช้ไมโครโฟน 4 ตัว โดยไมโครโฟนนั้นจะเป็นแบบ Wide-Field, conic audio capture รับเสียงในพื้นที่กว้าง และให้ความสำคัญกับเสียงเป็นรูปกรวย(สนใจเสียงไกลๆ เสียงใกล้ๆอย่าง เสียงที่มาจาก Xbox เอง, Kinect จะไม่สนใจ) หลังจากนั้นจะใช้วิธี Voice Recognition เพื่อจดจำและ แปลความหมายของสิ่งที่ผู้เล่นแต่ละคนพูด โดย Kinect สามารถแยกแยะเสียงของแต่ละผู้เล่นได้\

2.1.1 หลักการทำงานของ Kinect Sensor

Kinect Sensor จะตรวจจับการเคลื่อนไหวของผู้ใช้โดยเริ่มต้นจากการให้ Infrared projector ฉายแสงอินฟราเฟรดในลักษณะเป็นแพทเทิร์นจุดดังภาพ

รูปที่ 2.4 อินฟราเฟรดของ Kinect

จากนั้น Infrared camera จะรับระดับความสว่างของแสงที่ตกกระทบบนวัตถุ โดยวัตถุที่อยู่ใกล้แสงจะมีความสว่างมากและวัตถุที่อยู่ไกลจะมีแสงสว่างน้อยกว่า จากนั้นจะนำข้อมูลที่ได้ไปประมวลผลด้วยซอฟต์แวร์จะได้ระดับความลึกตื้นออกมาดังภาพด้านล่าง โดยวัตถุที่อยู่ใกล้กับตัวเซนเซอร์จะมีสีอ่อนและวัตถุที่อยู่ไกลจะมีสีเข้มไล่ระดับกันไป

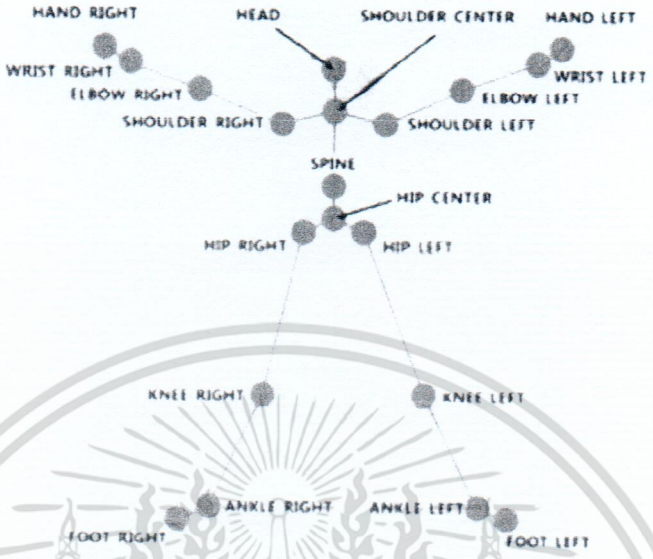
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 ระดับความลึกตื้นของภาพที่ถ่ายด้วย Kinect

เมื่อ Kinect รู้ระดับความตื้นลึกแล้วก็จะสามารถแยกผู้เล่นออกจากสภาพแวดล้อมภายในห้องได้ นอกจากนั้น Kinect Sensor ยังมีระบบ Skeletal Tracking ที่ใช้ติดตามโครงกระดูกของผู้ใช้งาน ซึ่งสามารถติดตามได้มากที่สุด 2 คน แต่จะมองเห็นทั้งหมด 6 คน ซึ่งภาพที่แสดงจะเป็นภาพของโครงกระดูกมนุษย์ที่แทนด้วยข้อต่อ 20 จุดสำคัญตามร่างกาย

รูปที่ 2.6 Skeletal Tracking จับที่คน

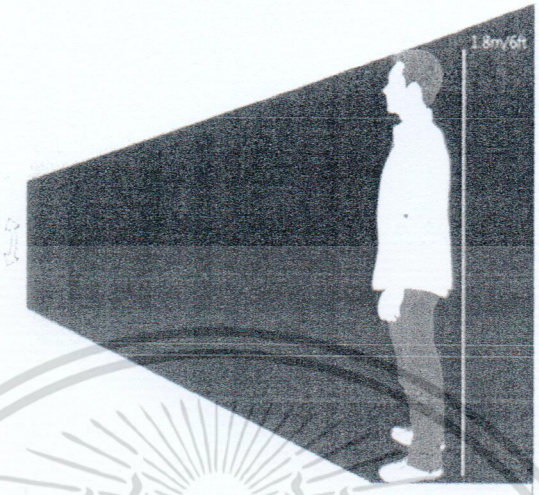
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ระบบ Skeletal Tracking

ระยะการตรวจจับการเคลื่อนไหวของ Kinect นั้นจะตรวจจับได้ใกล้ที่สุดในระยะที่ห่างออกไป 0.8 เมตร จนไปถึง 4 เมตร แต่ระยะที่ทำการตรวจจับได้ดีที่สุดคือห่างจาก Kinect ตั้งแต่ 1.2 เมตร จนไปถึง 3.5 เมตร

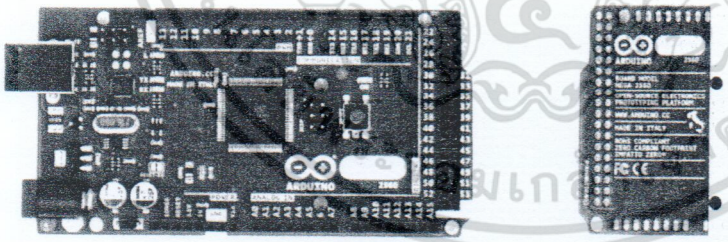
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



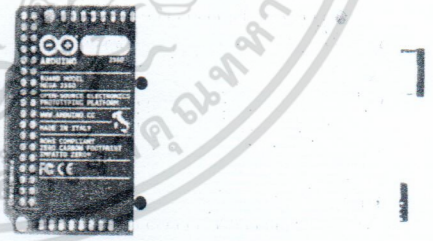
รูปที่ 2.8 ระยะการตรวจจับการเคลื่อนไหวของ Kinect

2.2 Arduino

Arduino Mega 2560



รูปที่ 2.9 Arduino Mega 2560 R3 Front



รูปที่ 2.10 Arduino Mega 2560 R3 Back

Arduino อ่านว่า (อา-ดู-อิ-โน่ หรือ อาดุยโน่) เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัว บอร์ดถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้น ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Shield) ประเภทต่างๆ เช่น XBee Shield, Music Shield, Relay Shield, Wireless Shield, GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

Arduino Mega 2560 บอร์ดรุ่นใหญ่ในของตระกูล Arduino มีคุณสมบัติต่างๆ เพิ่มขึ้นจาก Arduino Uno R3 ใช้ชิพ ATmega2560 ที่มีหน่วยความจำแฟลช 256 KB แรม 8 KB ใช้ไฟเลี้ยง 7 ถึง 12 V แรงดันของระบบอยู่ที่ 5 V มี Digital Input / Output มากถึง 54 ขา (เป็น PWM ได้ 14 ขา) มี Analog Input 16 ขา Serial UART 4 ชุด I2C 1 ชุด SPI 1 ชุด เขียนโปรแกรมบน Arduino IDE และโปรแกรมผ่าน USB เหมาะสำหรับผู้ที่สนใจเริ่มต้นเรียนรู้การพัฒนาไมโครคอนโทรลเลอร์ที่ต้องการบอร์ด Arduino ที่มีหน่วยความจำและขาสัญญาณต่างๆ ให้ต่อใช้งานมากขึ้น

Technical Specifications

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins 54 (of which 14 provide PWM output , 4 UART TTL)	
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory 256 KB of which 8 KB used by bootloader	
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Arduino Mega สามารถเชื่อมรับพลังงานโดยการเชื่อมต่อ micro USB connector หรือ จาก power supply จากภายนอกได้ โดยแหล่งพลังงานจะถูกเลือกโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แหล่งจ่ายจากภายนอกสามารถมาได้จาก AC-to-DC adapter หรือจากแบตเตอรี่ โดยต่อเข้ากับ 2.1mm center-positive plug ไปยังช่องเสียบแหล่งจ่าย และการต่อเข้ากับแบตเตอรี่สามารถทำได้โดยการต่อเข้ากับ GND และ Vin pin header ของ power connecter

บอร์ดสามารถทำงานได้ในช่วงแรงดัน 6 ถึง 20 volts ถ้า แหล่งจ่ายมีค่าต่ำกว่า 7 V อาจส่งผลให้ 5 V pin มีแรงดันที่ต่ำกว่า 5V และ บอร์ดอาจจะไม่เสถียร แต่ถ้าหากแรงดันมีค่าสูงกว่า 12 V อาจส่งผลให้บอร์ด Overheat และอาจทำให้บอร์ดเสียหายได้ ดังนั้นช่วงแรงดันที่เหมาะสมกับบอร์ดคือ 7 V ถึง 12 V

- VIN เป็น input voltage ของบอร์ด Arduino โดยใช้แหล่งจ่ายจากภายนอก
- 5V เป็น output pin ที่ควบคุม 5 V จากบอร์ด
- 3V3 เป็น 3.3 volt supply ที่สร้างขึ้นจาก regulator บนบอร์ด และให้กระแสได้สูงสุด 50 mA
- GND เป็น ground pin
- IOREF เป็น pin ที่ให้ voltage reference กับไมโครคอนโทรลเลอร์ เพื่อเลือกค่าแรงดันให้กับ shield ที่มาเชื่อมต่อกับบอร์ด

Memory

ATmega2560 มีหน่วยความจำ 256 KB (8 KB ใช้สำหรับ bootloader นอกจากนี้ยังมีอีก 8 KB สำหรับ SRAM และ 4 KB สำหรับ EEPROM

Input and Output

ในแต่ละ digital pins ทั้ง 54 pins บนบอร์ด Arduino Uno สามารถเป็นได้ทั้ง input และ output โดยจะทำงานที่แรงดัน 5 V และให้กระแสสูงสุด 40 mA

ฟังก์ชันอื่นๆเพิ่มเติม

Serial: 0 (Rx) และ 1(Tx); Serial 1: 19(Rx) และ 18 (Tx); Serial 2: 17 (Rx) และ 16(Tx); Serial 3:15 (Rx) และ 14 (Tx) ใช้สำหรับรับ (Rx) และส่ง(Tx) TTL serial data โดย pin 0 และ 1 จะถูกเชื่อมต่อไปยัง corresponding pins ของ ATmega16U2 USB-to-TTL serial chip

External Interrupts: 2 (interrupt 0) , 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), 21 (interrupt 2). pins เหล่านี้สามารถที่จะกำหนดค่าที่เรียก interrupt ในค่าต่ำๆ, ขอบขาขึ้นและลง หรือเปลี่ยนแปลงค่า

PWM: 2 ถึง 13 และ 44 ถึง 46 ให้ output PWM output 8-bits

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) ใช้สำหรับรองรับการสื่อสารแบบ SPI โดยที่ไม่เกี่ยวข้องกันกับ ICSP header ซึ่งจะมีลักษณะคล้ายกับ Uno, Duemilanove และ Diecimila

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LED 13 : เป็น build-in LED ที่เชื่อมต่อกับ digital pin 13 เมื่อ pin มีค่าเป็น HIGH LED จะติด , แต่เมื่อ pin เป็น LOW LED จะดับ

TWI : 20 (SDA) and 21 (SCL). รองรับการเชื่อมต่อแบบ TWI(I2C)

บอร์ด Mega2560 มี 16 analog inputs แต่ละ pins ให้ความละเอียด 10 bits

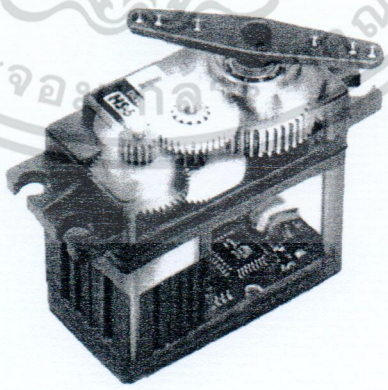
AREF. แรงดันอ้างอิง สำหรับ analog input

Reset ใช้ในการ reset ไมโครคอนโทรลเลอร์ โดยทั่วไปจะใช้โดยการเพิ่มปุ่ม reset ไว้บน shield เพื่อป้องกันปุ่มที่อยู่บนบอร์ด

2.3 Servo Motor

Servo Motor เป็นมอเตอร์ที่ทำงานโดยใช้สัญญาณพัลส์ โดยภายในเซอร์โวมอเตอร์จะประกอบไปด้วย มอเตอร์ไฟฟ้ากระแสตรง ชุดเกียร์และส่วนควบคุม โดยจะประกอบกันอยู่ภายในชุดเดียวกัน ตัวเซอร์โวมอเตอร์จะมีสายสัญญาณ 3 เส้น คือ สานใช้งาน 1 เส้น อีก 2 เส้นจะเป็นสายสำหรับจ่ายไฟให้เซอร์โวมอเตอร์และสายสำหรับต่อลงกราวด์ ในการควบคุมเซอร์โวมอเอตร์นั้นจะทำให้หมุนไปทางซ้ายได้ 90 องศา ไปทางขวาได้ 90 องศา (180 องศา) และสามารถสั่งให้หมุนไปตามองศาที่กำหนดได้ (ในการที่จะทำให้หมุน 360 องศาจะต้องดัดแปลงแก้ไขส่วนประกอบภายใน ซึ่งจะไม่ขอกล่าวถึงในที่นี้)

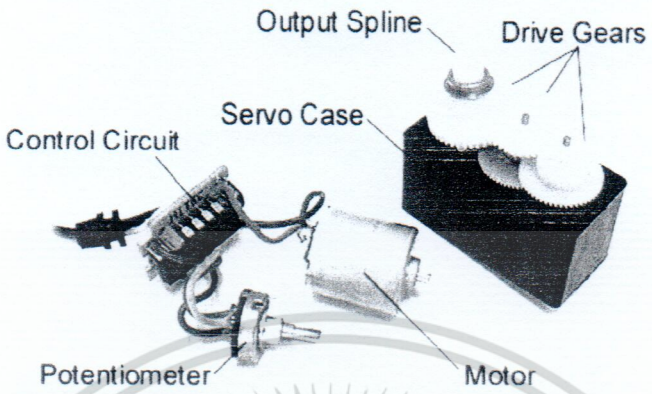
การใช้งานเซอร์โวมอเตอร์นั้นจะนำ ไปใช้ในที่ที่ต้องการความแม่นยำในเรื่ององศา หรือ การหมุนไปตามองศาที่ต้องการ เช่น ใช้เป็นมอเตอร์บังคับการเลี้ยวของหางเสือ เรือ หรือ การเลี้ยวของเครื่องบินบังคับวิทยุ แม้แต่สร้างเป็นหุ่นยนต์เดินขนาดเล็ก เพราะตัวเซอร์โวมอเอตร์เองจะมีแรงบิดค่อนข้างสูง (เพราะภายในจะมีชุดเกียร์อยู่แล้ว)



รูปที่ 2.11 Servo Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 องค์ประกอบพื้นฐานของเซอร์โว



รูปที่ 2.12 Servo-parts

องค์ประกอบหลักของเซอร์โวโดยทั่วไปแล้วจะมีส่วนประกอบหลักดังนี้คือ

1.Servo Case ซึ่งส่วนใหญ่จะทำมาจากพลาสติก

2.Motor ซึ่งเป็นส่วนให้กำลังในการหมุนของเซอร์โว

3.Control Circuit มีหน้าที่ในการถอดรหัสสัญญาณควบคุมจากรีซีฟซึ่งส่งมาเป็นแบบ PWM และส่งการควบคุมไปสั่งการทำงานของมอเตอร์ให้หมุนแขนของเซอร์โวให้อยู่ในตำแหน่งที่ได้ถอดรหัสมา

4.Potentiometer คือส่วนที่ตรวจวัดตำแหน่งของเซอร์โวและส่งสัญญาณกลับไปยัง Control Circuit เพื่อแก้ไขตำแหน่งให้ถูกต้องตามสัญญาณที่ได้เซ็ตไว้

5.Drive Gear คือชุดทดรอบจากการหมุนของมอเตอร์เพื่อให้ได้แรงบิดที่สูง

6.Output Spline คือส่วนที่ป้องกันการเสียดสีระหว่าง Servo Case และ Output shaft ซึ่งอาจใช้อุปกรณ์ประเภท Baring เพื่อช่วยลดแรงเสียดทานที่ดี

7.Servo wire คือสายไฟของเซอร์โวซึ่งมีรายละเอียดดังนี้

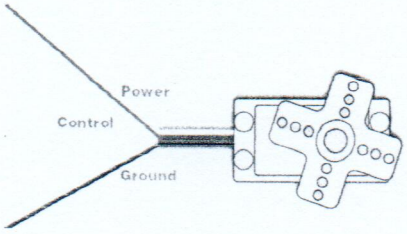
สายไฟของเซอร์โวจะมีอยู่สามเส้นซึ่งจะติดเป็นชุดเดียวกัน ซึ่งจะมีหน้าที่คือ

เส้นที่ 1 จ่ายไฟกระแส + DC ซึ่งแรงดันปรกติจะอยู่ที่ 5-6 โวลท์

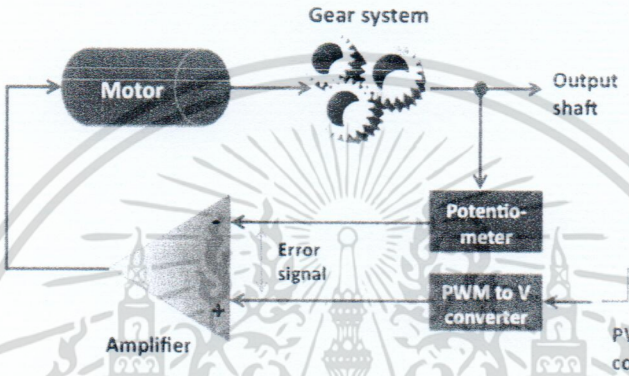
เส้นที่ 2 เป็นสาย Ground หรือเป็นขั้ว – DC

เส้นที่ 3 เป็นสายสัญญาณ โดยที่รีซีฟจะส่งสัญญาณลักษณะ on/off pulsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 Servo wire

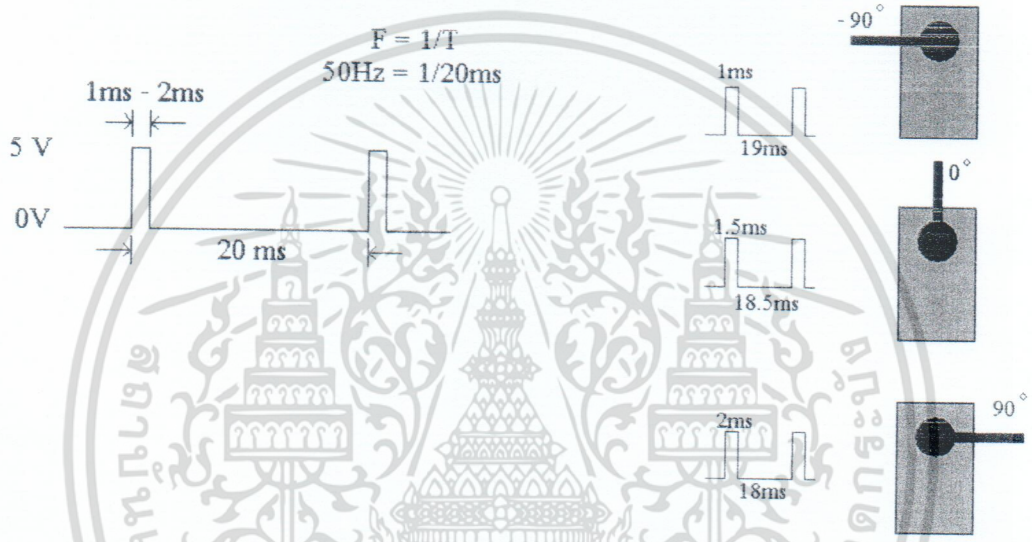


รูปที่ 2.14 Diagram การทำงานของ Servo Motor

ตามภาพด้านบนซึ่งแสดงระบบการทำงานของเซอร์โวโดยที่รีซีฟจะส่งสัญญาณการควบคุมตำแหน่งของเซอร์โวไปยังส่วน Control Circuit ของเซอร์โวโดยสัญญาณที่ส่งมาจะเป็นสัญญาณแบบ PWM(Pule Width Modulation) จากนั้น Control Circuit จะถอดรหัสสัญญาณ PWM ที่ได้ให้เป็นตำแหน่งของเซอร์โวที่ถูกต้องโดยเปรียบเทียบค่าตำแหน่งปัจจุบันกับสัญญาณกลับจาก Potentiometer แล้วจึงส่งแรงดันไฟฟ้าไปยังมอเตอร์ให้ไปหมุนไปในทิศทางที่จะทำให้ตำแหน่งของ Potentiometer มีค่าที่ถูกต้องเท่ากับค่าที่ได้ถอดรหัสมา ซึ่งขณะที่มอเตอร์หมุนก็จะมีเฟืองที่ไปต่อกับแกนของ Potentiometer(ปรกติจะอยู่ในแกนเดียวกับ output shaft) ด้วยดังนั้นกระบวนการนี้จะเกิดขึ้นซ้ำๆ จนกว่าค่าของ Potentiometer จะมีค่าเท่ากับการถอดรหัสสัญญาณที่ได้รับมาจากรีซีฟการทำงานของมอเตอร์จึงจะหยุด แต่กระบวนการทำงานของ Control Circuit จะยังทำงานอยู่ตลอดเวลาเพียงแต่หากค่าของ Potentiometer มีค่าเท่ากับสัญญาณที่ถอดรหัสมาจากรีซีฟแล้วก็จะไม่มีการส่งแรงดันไฟฟ้าไปยังมอเตอร์(ไม่มีการเปลี่ยนแปลงของสติ๊กที่รีโมท) ซึ่งกระบวนการนี้เรียกว่าการอัพเดทสัญญาณโดยมีความเร็วที่ 50 ครั้งต่อหนึ่งวินาที เราจึงเห็นเป็นการเคลื่อนที่ของเซอร์โว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณพัลส์ที่ใช้ควบคุมตำแหน่งของเซอร์โวมอเตอร์จะเป็นสัญญาณความถี่ 50 Hz มีคาบเวลา คือ 20 ms โดยสัญญาณช่วงบวกจะมีค่าอยู่ระหว่าง 1ms - 2 ms และ สัญญาณช่วงลบจะมีค่าอยู่ประมาณ 18 ms ซึ่งเมื่อรวมคาบสัญญาณระหว่างช่วงบวกและช่วงลบจะมีค่าเท่ากับ 20 ms นั้นเอง รูปด้านล่างเป็นลักษณะของสัญญาณพัลส์ ที่จ่ายไปยังขา Signal ของเซอร์โวมอเตอร์ โดยความกว้างของสัญญาณพัลส์ช่วงบวกจะมีผลต่อตำแหน่งของมอเตอร์



รูปที่ 2.15 สัญญาณพัลส์ การทำงานของ Servo Motor

2.4 ทฤษฎีเกี่ยวกับหุ่นยนต์

หุ่นยนต์ฮิวแมนนอยด์ (Humanoid robot) คือหุ่นยนต์ที่ออกแบบขึ้นมาโดยมีพื้นฐานมาจากร่างกายมนุษย์ โดยทั่วไปแล้วหุ่นยนต์ฮิวแมนนอยด์มีลำตัวพร้อมหัว สองแขน และสองขา แม้หุ่นยนต์ฮิวแมนนอยด์บางรูปแบบจะจำลองเฉพาะบางส่วนของร่างกายเท่านั้น ตัวอย่างเช่น ตั้งแต่เอวขึ้นไป หุ่นยนต์ฮิวแมนนอยด์บางตัวยังอาจมี 'ใบหน้า' พร้อม 'ตา' และ 'ปาก' อีกด้วย

แอนดรอยด์ คือหุ่นยนต์ฮิวแมนนอยด์ที่สร้างเลียนแบบมนุษย์เพศชาย และ -gynoid คือหุ่นยนต์ฮิวแมนนอยด์ที่สร้างเลียนแบบมนุษย์เพศหญิง

หุ่นยนต์ฮิวแมนนอยด์เป็นหุ่นยนต์อัตโนมัติ เนื่องจากมันสามารถปรับตัวเข้ากับการเปลี่ยนแปลงของสภาพแวดล้อมหรือตัวมันเอง และยังคงทำงานต่อเพื่อบรรลุเป้าหมาย สิ่งนี้เป็นข้อแตกต่างหลักระหว่าง ฮิวแมนนอยด์และหุ่นยนต์ชนิดอื่น เช่นหุ่นยนต์อุตสาหกรรม ที่ใช้ปฏิบัติภารกิจในสภาพแวดล้อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มีโครงสร้างชัดเจนมาก ในบริบทนี้ ความสามารถของหุ่นยนต์ฮิวแมนนอยด์อาจรวมถึง แต่ไม่จำกัดแค่สิ่งเหล่านี้ ดูแลรักษาตัวเอง (เติมพลังงานให้ตัวมันเอง) เรียนรู้อัตโนมัติ (เรียนรู้หรือได้มาซึ่งความสามารถใหม่โดยไม่ต้องได้รับความช่วยเหลือจากภายนอก, ปรับเปลี่ยนยุทธศาสตร์ตามสิ่งแวดล้อม และปรับตัวเข้ากับสถานการณ์ใหม่ ๆ) หลีกเลี่ยงสถานการณ์ที่จะเป็นอันตรายต่อมนุษย์ ทรัพย์สิน และตัวมันเอง โต้ตอบกับมนุษย์และสภาพแวดล้อมอย่างปลอดภัย

2.4.1 การพัฒนาหุ่นยนต์ฮิวแมนนอยด์

ระบบปัญญาประดิษฐ์ที่ใช้ในการควบคุมการเคลื่อนที่หรือการเดินของหุ่นยนต์ในขั้นต้น คือ การเดินแบบสถิตย์ หรือการเคลื่อนที่โดยอาศัยจุดศูนย์ถ่วงที่อยู่ภายในพื้นที่ครอบคลุมบริเวณขาทั้ง 2 ข้างของหุ่นยนต์ จากนั้นจึงเป็นการพัฒนาเป็นรูปแบบ การเดินแบบจลน์ หรือการเคลื่อนที่โดยอาศัยจุดศูนย์ถ่วงที่อยู่นอกพื้นที่ครอบคลุมของขาทั้ง 2 ข้าง ซึ่งเป็นรูปแบบการเดินของมนุษย์ตามลำดับ ซึ่งทีมวิศวกรได้ทำการศึกษาวิจัยและพัฒนาตามข้อมูลที่ทำการทดลองและจดบันทึกเป็นฐานข้อมูล จากการทดลองรูปแบบการเคลื่อนที่ของมนุษย์ ทีมวิศวกรได้คำนึงถึงองค์ประกอบสำคัญ 3 อย่างในการพัฒนาหุ่นยนต์ฮิวแมนนอยด์ให้สามารถเดินได้เช่นเดียวกับมนุษย์ คือ

1. การพัฒนาความเร็วในการเคลื่อนที่ไปด้านหน้าของหุ่นยนต์
2. การเพิ่มเติมในระดับถัดไปของร่างกาย เช่น แขน มือและศีรษะ
3. การพัฒนาความสามารถในการเคลื่อนที่ของหุ่นยนต์ เช่นการก้าวเดินขึ้นลงบันได หรือการวิ่ง

2.4.2 การก้าวเดินแบบอิสระ

พื้นฐานในการเคลื่อนไหวของหุ่นยนต์ฮิวแมนนอยด์ 2 ขานั้น สามารถศึกษาได้โดยตรงจากการเคลื่อนไหวของหุ่นยนต์บนพื้นผิวที่เรียบ ไม่ขรุขระ ซึ่งหุ่นยนต์สามารถก้าวเดินได้ตามปกติ ลำดับต่อไปของการก้าวเดินคือการพัฒนาศักยภาพไปสู่การเคลื่อนที่ไปด้านหน้าของหุ่นยนต์อย่างอิสระ สามารถก้าวเดินบนพื้นผิวขรุขระ พื้นเอียงหรือพื้นที่ลาดชันรวมทั้งระดับของขั้นบันไดอย่างมีเสถียรภาพ และไม่เสี่ยงต่อการเสียหลักหกล้มขณะก้าวเดิน เทคนิคการก้าวเดินแบบอิสระของหุ่นยนต์ฮิวแมนนอยด์ สามารถสรุปได้ดังนี้

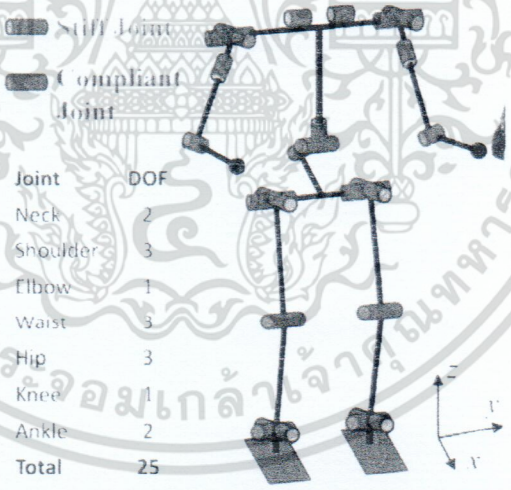
1. เทคนิคขั้นพื้นฐานของการวางเท้าของหุ่นยนต์ โดยจะต้องคงสภาพแน่นอนไม่แปรเปลี่ยนไปตามลักษณะของพื้นผิว แม้หุ่นยนต์จะก้าวเดินในพื้นที่ขรุขระ
2. เทคนิคขั้นพื้นฐานของการปรับสภาพของหุ่นยนต์ หากในกรณีที่หุ่นยนต์ก้าวเดินแล้วมีการหกล้มหรือเสียการทรงตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เทคนิคขั้นพื้นฐานของการปรับสภาพของหุ่นยนต์ โดยให้ผลของการเคลื่อนที่ถูกต้องแม่นยำโดยการจดจำเป้าหมายที่ต้องการ ซึ่งเทคนิคเหล่านี้ทีมวิศวกรได้นำมาพัฒนาเป็นส่วนต่าง ๆ ก่อนจะนำมารวมกันในรูปแบบการเคลื่อนไหวของหุ่นยนต์ฮิวแมนนอยด์

2.4.3 เทคนิคการก้าวเดินอย่างมีเสถียรภาพ

ปกติแล้วเมื่อมีการเคลื่อนที่หรือยืนอยู่นิ่ง ๆ ร่างกายของมนุษย์จะถ่ายโอนน้ำหนักตัวตามธรรมชาติ เพื่อรักษาสมดุลของร่างกายในท่านั้นไว้ แต่ถ้าการถ่ายโอนน้ำหนักของร่างกายไม่สมดุลกัน ร่างกายมนุษย์จะสามารถปรับสภาพให้สมดุลและไม่ล้ม โดยเคลื่อนตำแหน่งของเท้าซ้ายหรือขวาออกจากจุดที่ยืนอยู่ ซึ่งลักษณะดังกล่าวสามารถเกิดขึ้นกับหุ่นยนต์ฮิวแมนนอยด์ได้เช่นกัน เมื่อหุ่นยนต์ก้าวเดินไปข้างหน้า ผลจากแรงเฉื่อยและแรงดึงดูดของโลกจะมีผลโดยตรง ต่อการเพิ่มและลดความเร่งในท่าทางการเดินของหุ่นยนต์ ซึ่งแรงเหล่านี้เรียกว่าแรงเฉื่อยรวม และเมื่อเท้าของหุ่นยนต์กระแทกกับพื้น จะได้รับผลกระทบนี้โดยตรง เรียกว่าแรงปฏิกิริยาจากพื้น



รูปที่ 2.16 ข้อต่อส่วนต่างๆของหุ่นยนต์

การตัดกันในระหว่างแนวแรงเฉื่อยพื้น และรวมทั้งตำแหน่งดังกล่าวจะมีค่าโมเมนต์ เท่ากับศูนย์ เรียกตำแหน่งในจุดนี้ว่า จุดโมเมนต์ศูนย์ (ZMP) ซึ่งเป็นจุดที่แรงปฏิกิริยาลง เรียกว่าจุดปฏิกิริยาที่พื้นฐาน ลักษณะท่าทางการเดินของหุ่นยนต์จะถูกกำหนดจากคอมพิวเตอร์โดยระบบปัญญาประดิษฐ์ ทำการคำนวณ ประมวลผลและส่งผลไปยังข้อหมุนต่าง ๆ ของหุ่นยนต์ โดยให้มีการสอดคล้องกันกับความเฉื่อยที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เกิดขึ้นจากการคำนวณ เรียกว่า ZMP เป้าหมาย เมื่อหุ่นยนต์เกิดความสมดุลของร่างกายในขณะที่ก้าวเดินได้อย่างสมบูรณ์แบบ แกนของแรงเฉื่อยรวมเป้าหมายและแรงปฏิกิริยาที่พื้น จะเป็นตำแหน่งเดียวกัน และเมื่อหุ่นยนต์ก้าวเท้าเดินผ่านพื้นผิวที่ขรุขระ ตำแหน่ง 2 ตำแหน่งดังกล่าวจะหนีออกจากกัน ส่งผลให้เกิดความสมดุลลงแรงที่จะทำให้หุ่นยนต์หกล้มเกิดขึ้นมาทันที

แรงที่ทำให้หุ่นยนต์เกิดการหกล้มเมื่อก้าวเดิน เกิดจากความเหลื่อมล้ำในระหว่าง ZMP เป้าหมายและแรงปฏิกิริยารวมที่พื้น ซึ่งเมื่อพิจารณาและวิเคราะห์แล้วพบว่า นั่นคือสาเหตุหลักที่ทำให้ความไม่สมดุลเกิดขึ้น และเมื่อหุ่นยนต์เกิดการเสียความสมดุล ระบบป้องกัน 3 ระบบที่จะป้องกันการหกล้มหรือเสียหลักการทรงตัวของหุ่นยนต์ ที่สามารถทำให้หุ่นยนต์ก้าวเดินต่อไปได้อย่างต่อเนื่องคือ

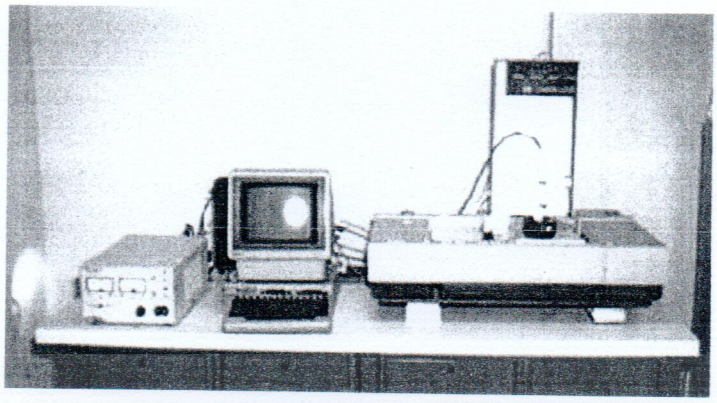
- ระบบควบคุมแรงปฏิกิริยา
- ระบบควบคุม ZMP
- ระบบควบคุมการวางเท้าของหุ่นยนต์

2.5 เครื่องพิมพ์ 3 มิติ (3D Printers)

ความก้าวหน้าของเทคโนโลยีที่รุดหน้าอย่างรวดเร็ว ทำให้เกิดเทคโนโลยีใหม่ ๆ ขึ้นมามากมาย และเทคโนโลยีที่กำลังเป็นที่นิยมและกล่าวถึงในขณะนี้ก็คือ 3D printing หรือการพิมพ์แบบสามมิติ คำว่า 3D printing อาจทำให้คิดว่าน่าจะเป็นการพิมพ์ภาพสามมิติทั่ว ๆ ไป แต่จริง ๆ แล้ว 3D printing นี้เป็นการสร้างโมเดลเสมือนจริงหรือการขึ้นรูปชิ้นงานนั่นเอง เครื่องพิมพ์แบบสามมิติไม่ใช่เรื่องใหม่ เพราะมีประวัติการพัฒนายาวนานกว่า 30 ปี แต่สิ่งที่ทำให้นวัตกรรมการพิมพ์ 3 มิติ กลับมาฮือฮาอีกครั้ง ก็เพราะมีความพยายามพัฒนาเครื่องพิมพ์ 3 มิติ ให้เข้าสู่ผู้ใช้ในระดับครัวเรือนมากขึ้น นอกเหนือจากนั้นนักวิทยาศาสตร์จากหลายสาขายังพยายามประยุกต์ใช้หลักการของการพิมพ์แบบสามมิติไปสู่งานวิจัยในระดับที่เล็กลงแต่ซับซ้อนขึ้นเรื่อย ๆ จนวันหนึ่งข้างหน้า เราอาจเห็น “ชีวิตสังเคราะห์” จากเครื่องพิมพ์ 3 มิติถือกำเนิดได้จริงในห้องปฏิบัติการ เครื่องพิมพ์แบบสามมิติถูกสร้างขึ้นตั้งแต่ในคริสตวรรษที่ 19 ช่วงปลาย (ประมาณปีค.ศ. 1984) โดย Charles W. (Chuck) Hull เป็นผู้ออกแบบเครื่องพิมพ์แบบสามมิติให้กับบริษัท 3D Systems Corporation (Charles Hull เป็นนักประดิษฐ์เครื่องพิมพ์ 3D ที่ทันสมัยและเป็นผู้ริเริ่มเทคโนโลยีมาตรฐาน de facto) โดยเครื่องพิมพ์สามมิตินี้ถูกตั้งชื่อว่า Stereolithographic 3-D printer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

143712



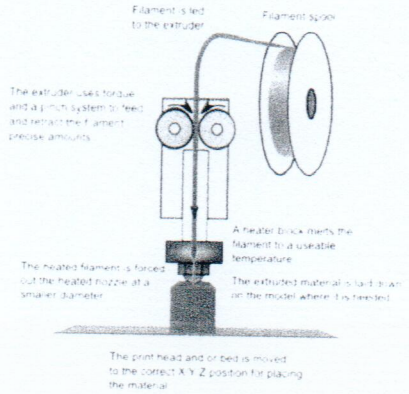
รูปที่ 2.17 Stereolithographic 3-D printer

หลังจากนั้นเทคโนโลยีการพิมพ์แบบสามมิติก็พัฒนาขึ้นเรื่อย ๆ มีเทคนิควิธีการที่ซับซ้อนขึ้น พร้อมกับมีบทบาทมากขึ้นในหลาย ๆ สายงาน เช่น การออกแบบ วิศวกรรม การแพทย์ ไปจนถึงวิทยาการอวกาศและการบิน อีกทั้งยังมีเทคนิคและวิธีการพิมพ์แบบสามมิติเพิ่มมากขึ้นเพื่อให้เหมาะกับชิ้นงานและชนิดของวัสดุที่ต้องการขึ้นรูป ตัวอย่างของเทคนิคที่ใช้ในการขึ้นรูปวัสดุ เช่น

2.5.1 Stereolithography (SLA) ถือว่าเป็นเทคโนโลยี rapid prototype แรกที่คิดค้นขึ้นมา เป็นการขึ้นรูปชิ้นงานโดยการยิงแสงเลเซอร์ไปบนของเหลว เพื่อให้ของเหลวในแต่ละชั้นแข็งตัว โดยวัสดุของเหลวเป็นแบบ liquid photopolymer (resin) ชิ้นงานที่ได้จะมีความละเอียดและเที่ยงตรงสูง แต่ราคาเครื่องจักรและค่าใช้จ่ายในการดูแลรักษาก็สูงเช่นกัน ตัวอย่างการใช้เทคโนโลยี SLA ในการทำหมากรุกใช้เวลาทำทั้งสิ้น 2 ชั่วโมง

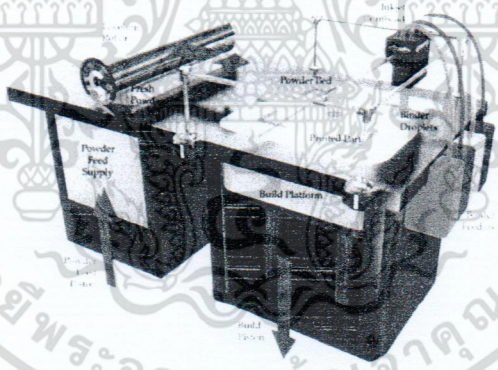
2.5.2 Fused Deposition Modeling (FDM) เป็นวิธีที่ใช้แพร่หลายที่สุด ใช้วัสดุหลักเป็นแท่งพลาสติกหรือโลหะที่พันเป็นม้วน ส่งผ่านไปยังหัวจ่ายซึ่งจะหลอมให้พลาสติกละลายก่อนนำมาพ่นลงบนแท่นวาง ซึ่งเมื่อพ่นออกมาแล้วจะแข็งตัวอย่างรวดเร็ว เครื่องจะพ่นทีละชั้นกระทั่งกลายเป็นโมเดล 3 มิติตามรูปทรงที่ต้องการ วัสดุที่ใช้ในการขึ้นรูปด้วยวิธีนี้ได้แก่ เทอร์โมพลาสติก เช่น ABS, โพลีคาร์บอเนต และโพลีฟีนิลซัลโฟนจำพวกอีลาสโตเมอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 วิธีการแบบ Fused Deposition Modeling (FDM)

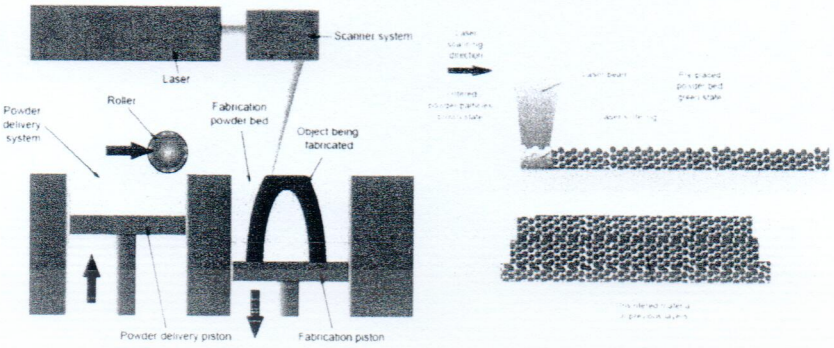
2.5.3 3D jet printer systems ใช้หลักการของ Ink Jet Printer แต่เป็นการ print ใน 3 มิติ โดยการพ่นวัสดุที่เป็นเทอร์โมพลาสติกเช่น โพลีเอสเทอร์ ผ่านหัวพ่นให้เป็นรูปแบบตามที่ต้องการ ใช้งานง่ายและสร้างแบบจำลองได้อย่างรวดเร็ว วิธีนี้มีจุดเด่นอยู่ที่สามารถเลือกสีต่างๆ ได้



รูปที่ 2.19 เครื่อง 3D jet printer systems

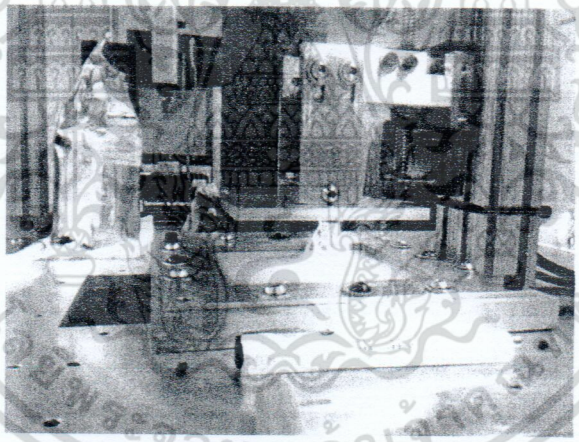
2.5.4 Selective Laser Sintering (SLS) คือ การเผาผนึกวัตถุดิบที่ใช้ด้วยแสงเลเซอร์ ใช้หลักการเดียวกันกับ SLA แต่ทำให้สามารถใช้วัสดุที่มีความแข็งแรงมากกว่า เช่น Thermoplastic,Nylon, Polyamide และ Polystyrene โดยวัตถุดิบที่ใช้จะอยู่ในรูปผง เช่น โลหะ พลาสติก เซรามิก และแก้ว เครื่องจะใช้เลเซอร์วาดรูปแบบขึ้นทีละชั้นและพ่นผงเหล่านี้ลงไปหลอมละลายกระทั่งกลายเป็นชิ้นงาน วิธีนี้ใช้ในอุตสาหกรรมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 Selective Laser Sintering (SLS)

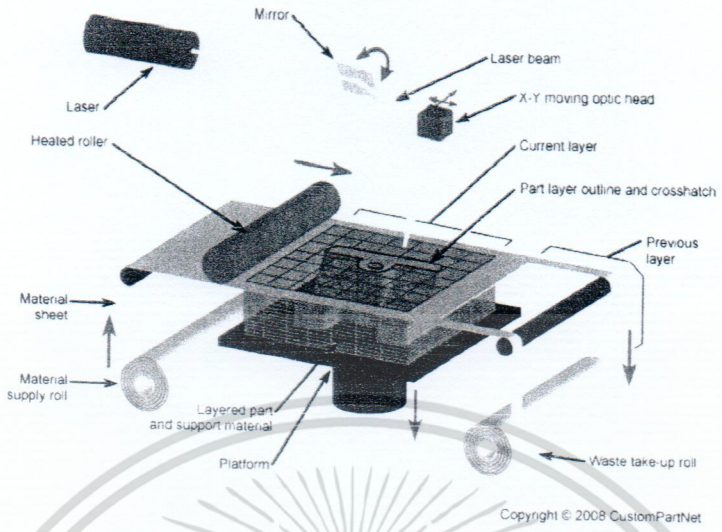
2.5.5 Digital Light Processing (DLP) ใช้หลักการเดียวกันกับ SLS แต่ใช้แหล่งกำเนิดแสงเป็นแสงสีขาวเป็นแหล่งกำเนิดแสงจากไฟแอลอีดีเพื่อให้ความร้อนแทนเลเซอร์ วัสดุเป็นยาง, เรซิ่น



รูปที่ 2.21 วิธีแบบ Digital Light Processing (DLP)

2.5.6 Laminated Object Manufacturing (LOM) เป็นการใช้วัสดุที่เป็นแผ่นบาง ๆ คล้ายกระดาษ เช่น และมีสารยึดติดที่หน้าหนึ่งของแผ่น แล้ว feed เข้าสู่เครื่องตัดด้วยเลเซอร์ เป็นชั้นต่อชั้นขึ้นไป วัสดุที่จะขึ้นรูปจะอยู่ในกลุ่มของเทอร์โมพลาสติก(พีวีซี) โลหะที่เป็นเหล็ก และไม่ใช่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 วิธีการแบบ Laminated Object Manufacturing (LOM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 Kinect

Kinect คือ อุปกรณ์เสริมของเครื่องเล่นเกมส์ Xbox360 จาก Microsoft โดยสามารถจดจำผู้เล่น (Facial Recognition)และให้ผู้เล่นควบคุมเกมส์ผ่านทางการเคลื่อนไหวร่างกายของผู้เล่นโดยตรง(3D Motion Recognition)โดยไม่จำเป็นต้องมี joystickอีกต่อไป และสามารถจดจำเสียงของผู้เล่นได้ (Voice Recognition) และนอกจากใช้ฟังค์ชั่นพวกนี้เล่นเกมส์แล้วเรายังใช้Kinectควบคุมการดูหนัง ฟังเพลง เล่น Window live messenger ได้ด้วย

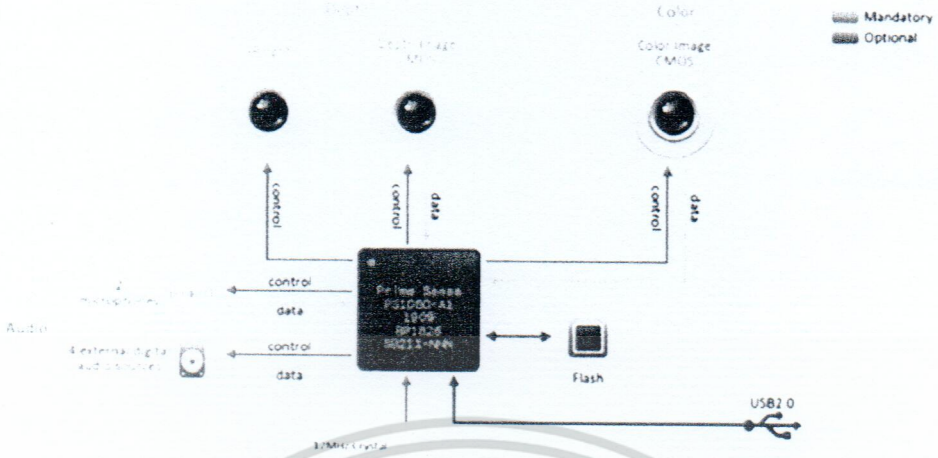
รูปที่ 2.1 Kinect Sensor

(ที่มา : http://news.blogsdna.com/11228/kinect-tech-specs-revealed-limited-to-2-active-players.htm)

โครงสร้างของ Kinect (Kinect Sensor) จะมีลักษณะเป็นแท่งแบนๆ เพื่อใช้ในการวางในแนวราบที่เชื่อมต่อกับฐานขนาดเล็กที่มีมอเตอร์เพื่อใช้ในการเคลื่อนไหวในการปรับมุมกล้อง โดยภายในจะประกอบด้วย

1. กล้อง CMOS RGB 1 ตัว
2. กล้อง IR (Infra Red) 1 ตัว
3. ตัวส่งแสง IR (Infra Red)
4. ไมโครโฟน 4 ตัว
5. มอเตอร์ไว้ปรับมุมกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ระบบภายในของ Kinect Sensor

(ที่มา : http://www.manager.co.th/Game/ViewNews.aspx?NewsID=9530000156439)

PRODUCT SPECIFICATION

Field of View (Horizontal, Vertical, Diagonal)	58° H, 45° V, 70° D	Color image size	UXGA (1600×1200)
Depth image size	VGA (640×480)	Audio, built-in microphones	2 mics
Spatial x/y resolution (@2m distance from sensor)	3mm	Audio, digital inputs	4 inputs
Depth z resolution (@2m distance from sensor)	1cm	Data interface	USB 2.0
Maximal image throughput (frame rate)	60fps	Power supply	USB 2.0
Average image latency in full VGA resolution	40msec	Power consumption	2.25W
Operation range	0.8m - 3.5m	Dimensions (Width x Height x Depth)	14cm x 3.5cm x 5cm
		Operation environment (every lighting condition)	indoor
		Operating temperature	0°C - 40°C

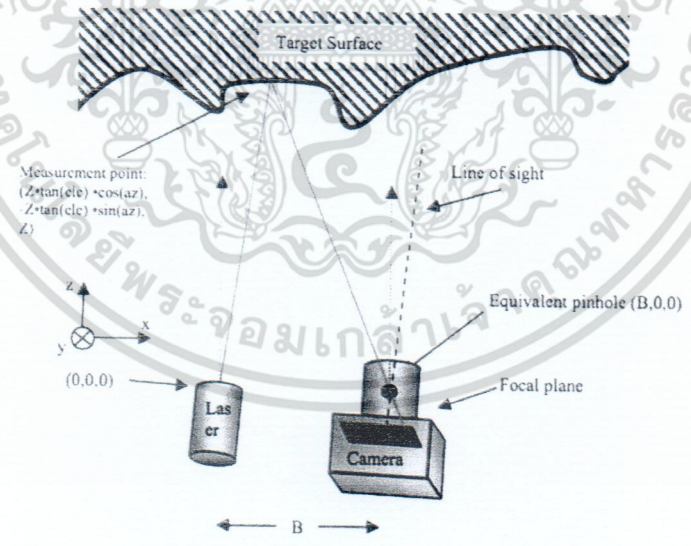
รูปที่ 2.3 รายละเอียดของ Kinect

(ที่มา : http://www.anandtech.com/show/4057/microsoft-kinect-the-anandtech-review/2)

Kinect มีระบบการรับรู้การเคลื่อนไหวของผู้เล่นอยู่ภายในเซนเซอร์เองซึ่งเป็นการนำเทคโนโลยีปัญญาประดิษฐ์เข้ามาช่วยโดยจะมีการส่งข้อมูลการเคลื่อนไหวของผู้เล่นในลักษณะต่างๆเข้าไปเป็นจำนวนมาก ไม่ว่าจะเป็นท่าทางการนั่ง ยืน การเอียงตัว การขว้างลูกบอลการหยิบสิ่งของหรือแม้แต่กระทั่งการตวัดมือข้อมูลการเคลื่อนไหวเหล่านี้จะถูกประมวลเหลือเพียงการเคลื่อนไหวของโครงกระดูกซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

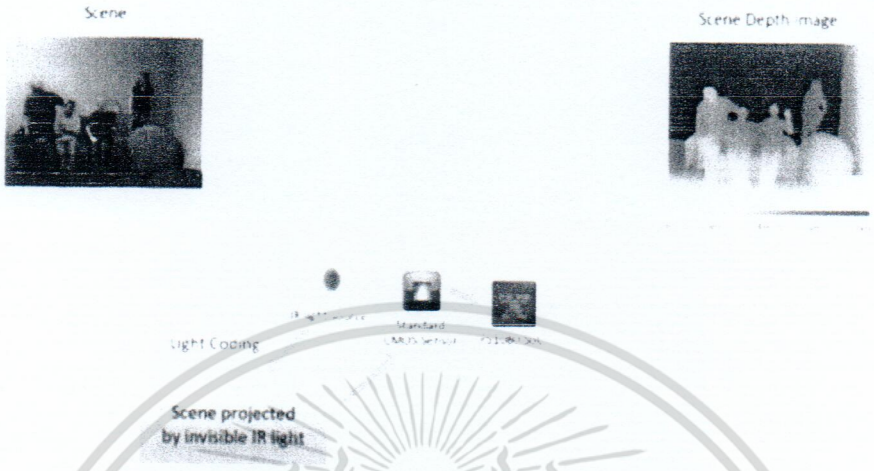
Kinect จะวิเคราะห์ลักษณะการเคลื่อนไหวของข้อต่อแต่ละข้อรวมทั้งสิ้น 20ข้อต่อเพื่อนำไปวิเคราะห์อีกครั้งว่าขณะนี้ผู้เล่นกำลังแสดงท่าทางอะไรอยู่ซึ่งเรียกว่า Skeletal Tracking การทำงานของ Kinect นั้นมีหลักการทำงานซึ่งเชื่อมโยงกับการทำงานของ hardware คือ การทำงานจะเริ่มจากการฉายแสงอินฟราเรดออกจากตัว Kinect ซึ่งไม่สามารถมองเห็นได้ด้วยตาเปล่าแสงที่ถูกฉายออกมาจะมีลักษณะเป็นจุดๆ ตามแนวตั้ง 480 จุด แนวนอน 640 จุดแต่ละจุดห่างกัน 3 มิลลิเมตร(ที่ระยะสองเมตรจากแหล่งกำเนิดแสง)ไปกระทบกับวัตถุโดยรอบไม่ว่าจะเป็นคนหรือสิ่งของและแสงอินฟราเรดที่สะท้อนกลับมานั้นจะรับสัญญาณโดยกล้องอินฟราเรด หลังจากนั้นกล้องวัดความลึกจะรับภาพระดับความสว่างของแสงอินฟราเรดที่ตกกระทบลงบนวัตถุส่งไปให้เซนเซอร์เพื่อทำการวัดความลึกตามแนวแกน Z (Z-Axis) โดย เซนเซอร์สามารถปรับตัวให้เข้ากับสภาพแวดล้อมต่างๆได้อัตโนมัติ เช่นใช้เวลานานในการสะท้อนแสงกลับมาแสดงว่าวัตถุอยู่ไกลหรือ ของที่มีสีต่างกันจะดูดกลืนและสะท้อนแสงความในความยาวคลื่นที่ไม่เท่ากันนอกจากนี้ Kinect สามารถจดจำเสียงผู้เล่นโดยใช้ไมโครโฟน 4 ตัว โดยไมโครโฟนจะเป็นแบบ Wide-Field, conic audio capture รับเสียงในพื้นที่กว้างและให้ความสำคัญกับเสียงเป็นรูปกรวย(สนใจเสียงไกลๆ เสียงใกล้ๆเซนเซอร์จะไม่สนใจ)หลังจากนั้นจะใช้วิธีVoice Recognition เพื่อจดจำและแปลความหายของที่สิ่งที่ผู้เล่นแต่ละคนพูด ซึ่ง Kinect สามารถแยกเสียงของผู้เล่นได้



รูปที่ 2.4 การทำงานของเซนเซอร์

(ที่มา : http://www.anandtech.com/show/4057/microsoft-kinect-the-anandtech-review/2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ตัวอย่างการทำงานของ Kinect
(ที่มา : http://www.ayarafun.com/2010/11/what-is-kinect/)

นอกจากนี้ Kinect ยังมีความสามารถในการจดจำผู้เล่นเกมหรือผู้ใช้งานได้คือ

1. จดจำผู้เล่นได้โดยใช้ ข้อมูลจากกล้อง CMOS RGB(ข้อมูล แดง เขียว น้ำเงิน) และประมวลผล
2. จดจำการเคลื่อนไหวร่างกายของผู้เล่นเป็นแบบ 3 มิติแบ่งเป็น 3 อย่างย่อย

2.1 จดจำวัตถุแบบ 3 มิติ : โดยใช้ตัวส่งแสง IR และกล้อง IR , ตัวส่งแสง IR จะส่งแสงไปกระทบกับวัตถุไม่ว่าจะเป็นคนหรือสิ่งของ และแสง IR จะสะท้อนจากวัตถุนั้นๆกลับไปที่กล้อง IR และ Kinect จะใช้ข้อมูลพวก "เวลาในการสะท้อนกลับ"และ"ความยาวคลื่นแสง" ในการประมวลผลวัตถุต่างๆเป็น 3 มิติ (ยกตัวอย่างเช่น ใช้เวลานานในการสะท้อนกลับแสดงว่าวัตถุอยู่ไกล,ของที่มีสีต่างกันจะดูดกลืนแสงที่พบและสะท้อนแสงกลับไปในความยาวคลื่นที่ต่างกัน

2.2 จดจำและแบ่งแยกประเภทของคน ว่าเป็นเพศไหน อายุประมาณเท่าไหร่ ขนาดสัดส่วนตัวประมาณไหนเท่าไร ใส่เสื้อผ้าหรือยัง โดยใช้เปรียบเทียบกับฐานข้อมูลที่มีอยู่

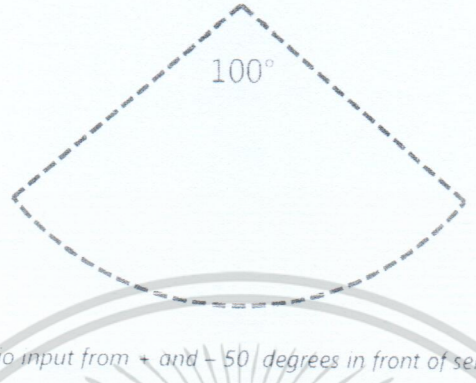
2.3 จดจำการเคลื่อนไหวของคน โดยหลังจากแบ่งแยกประเภทของคนแล้ว จะใช้วิธี skeletal movements(การเคลื่อนไหวตามลักษณะกระดูก)เข้าไปวิเคราะห์คนประเภทนั้นว่า ส่วนไหนคือหัว ตัวคอ หัวไหล่ แขนขวาแขนซ้าย ขาขวาขาซ้าย ข้อศอก, โดยใช้สัดส่วน ลักษณะ หรือข้อจำกัดต่างๆ

3. จดจำเสียงผู้เล่น โดยใช้ไมโครโฟน 4 ตัว โดยไมโครโฟนนั้นจะเป็นแบบ Wide-Field, conic audio capture รับเสียงในพื้นที่กว้าง และให้ความสำคัญกับเสียงเป็นรูปกรวย(สนใจเสียงไกลๆ เสียงใกล้ๆอย่างเสียงที่มาจาก Xbox เอง, Kinect จะไม่สนใจ) หลังจากนั้นจะใช้วิธี Voice Recognition เพื่อจดจำและแปลความหมายของสิ่งที่ผู้เล่นแต่ละคนพูด โดย Kinect สามารถแยกแยะเสียงของแต่ละผู้เล่นได้

2.1.1 ความสามารถของตัว Kinect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Kinect สามารถจับเสียงของผู้เล่นได้ในรัศมี 100 องศาดังรูป 2.6



รูปที่ 2.6 แสดงขอบเขตของการรับเสียงของตัว Kinect
(ที่มา : http://mportiz08.github.io/cpe486-research-project/speech.html)

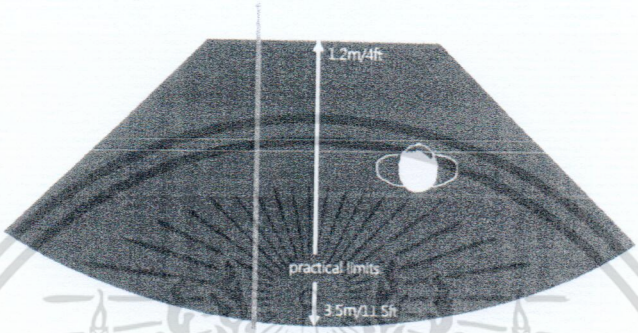
2. Kinect สามารถหมุนหาผู้เล่นได้ โดยสามารถหมุนในแนวราบได้ 57.5 องศา และในแนวตั้งได้ 43.5 องศา ดังรูปที่ 2.7

รูปที่ 2.7 แสดงองศาในการหมุนของ Kinect
(ที่มา : http://files.channel9.msdn.com/wlwimages/f1dda9cc6de74512b7c19f0101402403/image%5B8%5D-83.png)

3. Kinect สามารถจับผู้ใช้ได้ในระยะ 0.8-4 เมตร แต่มีระยะที่ทำงานได้ดีคือ 1.2-3.5 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Distance ranges for Depth (default mode)
Physical limits: 0.8 to 4m
Practical limits: 1.2 to 3.5m



รูปที่ 2.8 แสดงระยะในการจับผู้ใช้งาน
(ที่มา : https://msdn.microsoft.com/en-us/library/hh973074.aspx)

2.2 ความรู้เบื้องต้นเกี่ยวกับไมโครคอนโทรเลอร์

ไมโครคอนโทรลเลอร์เป็นอุปกรณ์ไอซี (IC: Integrated Circuit) ที่สามารถโปรแกรมการทำงานได้ซับซ้อน สามารถรับข้อมูลในรูปสัญญาณดิจิตอลเข้าไปทำการประมวลผลแล้วส่งผลลัพธ์ข้อมูลดิจิตอลออกมาเพื่อนำไปใช้งานตามที่ต้องการได้ โครงสร้างโดยทั่วไป ของไมโครคอนโทรลเลอร์นั้นเราสามารถที่จะแบ่งออกมาได้เป็น5ส่วนใหญ่ๆพื้นฐาน ดังต่อไปนี้

1. หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)เป็นส่วนที่ทำการประมวลผล
2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกกระดาษทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วๆไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปทันทีเมื่อเราปิดหรือไม่จ่ายไฟเลี้ยงมาเลี้ยงตัวคอนโทรลเลอร์และเป็นอีอีพรอม (EEPROM) ซึ่งเป็นหน่วยความจำที่เก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

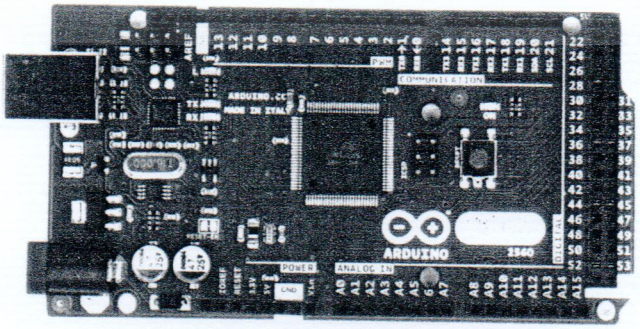
3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะด้วยการกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) , บัสแอดเดรส (Address Bus) และบัส

5. วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับการกำหนดจังหวะ หากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์นั้น มีความเร็วประมวลผลสูง ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานอย่างกว้างขวาง โดยมักจะเป็นการนำไปใช้ฝังในระบบของ อุปกรณ์อื่น ๆ เพื่อใช้ควบคุมการทำงานบางอย่าง เช่น ใช้ในรถยนต์, เตาอบไมโครเวฟ, เครื่องปรับอากาศ, เครื่องซักผ้าอัตโนมัติ เป็นต้น เพราะว่าไมโครคอนโทรลเลอร์มีข้อดีเหมาะสมต่อการใช้ในงานควบคุมหลายประการ เช่น

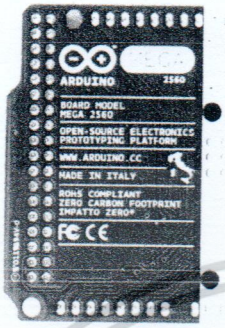
- ชิพไอซีและระบบที่ได้มีขนาดเล็ก
- ระบบที่ได้มีราคาถูกกว่าการใช้ชิพไมโครโพรเซสเซอร์
- วงจรที่ได้จะมีความซับซ้อนน้อย ช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นได้ในการต่อวงจร
- มีคุณสมบัติเพิ่มเติมสำหรับงานควบคุมโดยเฉพาะซึ่งใช้งานได้ง่าย

2.2.1 Arduino Mega 2560 R3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.9 Arduino Mega 2560 R3 Front
(ที่มา : http://www.arduino.cc/en/Main/ArduinoBoardMega2560)



รูปที่ 2.10 Arduino Mega 2560 R3 Back
(ที่มา : http://www.arduino.cc/en/Main/ArduinoBoardMega2560)

2.2.1.1 Overview

Arduino Mega 2560 เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ทำงานบนพื้นฐานของ ATmega2560 ซึ่งประกอบด้วย

- 54 digital input/output pins (15 pin สามารถใช้เป็น PWM output ได้)
- 16 analog inputs
- 4 UARTs (hardware serial ports)
- 16 MHz crystal oscillator (ใช้สำหรับกรองความถี่ให้กับบอร์ดไมโครคอนโทรลเลอร์)
- USB connection
- ช่องเสียบแหล่งจ่าย
- ICSP header :In-Circuit Serial Programming (ส่วนที่เป็น AVR ขนาดเล็กสำหรับการโปรแกรม Arduino ซึ่งประกอบด้วย MOSI, MISO, SCK, RESET, VCC,GND)
- ปุ่มกด reset

โดยบอร์ด Arduino Leonardo นี้มีทุกสิ่งที่ไมโครคอนโทรลเลอร์จำเป็นต้องใช้อย่างการต่อไฟเลี้ยงสามารถทำได้ทั้งการเชื่อมต่อเข้ากับ USB cable หรือ จ่ายไฟด้วย AC-DC adapter หรือ การใช้แบตเตอรี่ซึ่ง Mega เป็นบอร์ดที่เข้ากันได้กับ shield ที่ออกแบบมาเพื่อ Arduino Duemilanove หรือ Diecimila

Mega 2560 นี้มีความแตกต่างจากบอร์ดก่อนหน้าตรงที่ไม่ใช้ FTDI USB-to-serial driver chip แต่จะมี ATmega16U2 เข้ามาเป็นโปรแกรมแปลง USB-to-serial

Arduino Mega2560 Revision 2 มี ATmega8U2 ทำให้อัพเดท firmware ผ่าน USB protocol ที่เรียกว่า DFU(Device Firmware Update) ได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Mega Revision 3 มี featureใหม่ๆเพิ่มขึ้นมาดังนี้

- 1.0 pinout: เพิ่ม SDA และ SCL (อยู่ใกล้กับ AREF pin) และอีกสอง pins ใหม่คือ IOREF เป็น pin ที่ใช้ในการเชื่อมต่อกับ shields เพื่อแปลงเป็นแรงดันที่ได้จากบอร์ด ส่วนอีก 1 pin ที่เหลือมีไว้สำหรับใช้ร่วมกับ AVR ในอนาคต
- วงจร Reset ที่ดีขึ้น
- ใช้ ATmega 16U2 แทน 8U2

2.2.1.2 Power

Arduino Mega สามารถเชื่อมรับพลังงานโดยการเชื่อมต่อ micro USB connector หรือ จาก power supply จากภายนอกได้ โดยแหล่งพลังงานจะถูกเลือกโดยอัตโนมัติ

แหล่งจ่ายจากภายนอกสามารถมาได้จาก AC-to-DC adapter หรือจากแบตเตอรี่ โดยต่อเข้ากับ 2.1mm center-positive plug ไปยังช่องเสียบแหล่งจ่าย และการต่อเข้ากับแบตเตอรี่สามารถทำได้โดยการต่อเข้ากับGND และ Vin pin header ของ power connector

บอร์ดสามารถทำงานได้ในช่วงแรงดัน 6 ถึง 20 volts ถ้า แหล่งจ่ายมีค่าต่ำกว่า 7 V อาจส่งผลให้ 5 V pinมีแรงดันที่ต่ำกว่า 5V และ บอร์ดอาจจะไม่เสถียร แต่ถ้าหากแรงดันมีค่าสูงกว่า 12 V อาจส่งผลให้บอร์ดOverheat และอาจทำให้บอร์ดเสียหายได้ ดังนั้นช่วงแรงดันที่เหมาะสมกับบอร์ดคือ 7 V ถึง 12 V

- VIN เป็น input voltage ของบอร์ด Arduino โดยใช้แหล่งจ่ายจากภายนอก
- 5V เป็น output pin ที่ควบคุม 5 V จากบอร์ด
- 3V3 เป็น 3.3 volt supply ที่สร้างขึ้นจาก regulator บนบอร์ดและให้กระแสได้สูงสุด 50 mA
- GND เป็น ground pin
- IOREF เป็น pin ที่ให้ voltage reference กับไมโครคอนโทรลเลอร์ เพื่อเลือกค่าแรงดันให้กับ shield ที่มาเชื่อมต่อกับบอร์ด

2.2.1.3 Memory

ATmega2560 มีหน่วยความจำ 256 KB (8 KB ใช้สำหรับ boot loader) นอกจากนี้ยังมีอีก 8 KB สำหรับ SRAM และ 4 KB สำหรับ EEPROM

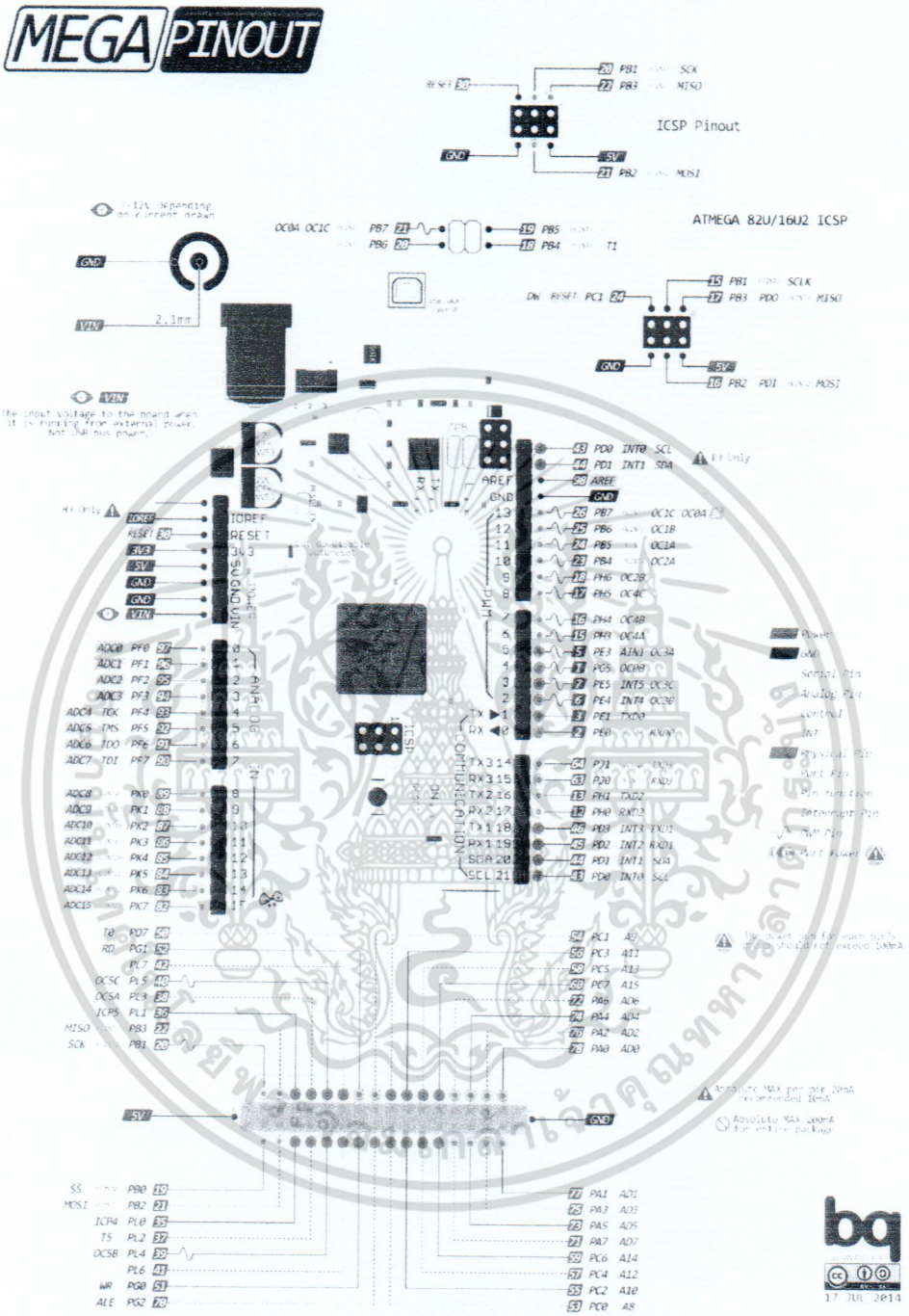
2.2.1.4 Input and Output

ในแต่ละ digital pins ทั้ง 54 pins บนบอร์ด Arduino Uno สามารถเป็นได้ทั้ง input และ output โดยจะทำงานที่แรงดัน 5 V และให้กระแสสูงสุด 40 mA

ฟังก์ชันอื่นๆเพิ่มเติม

Serial: 0 (Rx) และ 1(Tx); Serial 1: 19(Rx) และ 18 (Tx); Serial 2: 17 (Rx) และ16(Tx); Serial 3:15 (Rx) และ 14 (Tx) ใช้สำหรับรับ (Rx) และส่ง(Tx) TTL serial dataโดย pin 0 และ 1 จะถูกเชื่อมต่อไปยัง corresponding pins ของ ATmega16U2 USB-to-TTL serial chip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดง Port ต่าง ๆ บน Arduino Mega 2560 R3

(ที่มา : http://forum.arduino.cc/index.php/topic,146511.0.html)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Servo Motor

Servo Motor คือมอเตอร์ขนาดเล็กมีโครงสร้างภายในประกอบด้วยDC-Motor ขนาดเล็กพร้อมชุดเกียร์ทดรอบทำให้มีแรงบิดสูงเมื่อเทียบกับขนาดของตัวมอเตอร์เองนอกจากนี้ตัว Servo Motor ยังมีวงจรขับมอเตอร์และวงจรป้อนกลับ (Feedback control) รวมอยู่ภายในชุดด้วยซึ่งสามารถควบคุมตำแหน่งการหมุนได้อย่างแม่นยำดังนั้นจึงได้มีการนำไปประยุกต์ใช้กันอย่างมากมายในปัจจุบัน

คุณสมบัติ

1. ให้แรงบิดในการหมุนที่สูง
2. มีคุณสมบัติในการควบคุมตำแหน่งการทำงานได้อย่างแม่นยำควบคุมจากความกว้างของสัญญาณ PULSE
3. มีความสามารถในการหมุนเคลื่อนที่+/-90 องศาหรือ0-180 องศา
4. สามารถต่อสัญญาณการควบคุมร่วมกับไมโครคอนโทรลเลอร์ได้โดยตรง[TTL Signal]
5. สามารถควบคุมความเร็วในการหมุนได้ง่าย

รูปที่ 2.12 เซอร์โวมอเตอร์

(ที่มา : http://mco-s1-p.mlstatic.com/servo-motor-estandar-futaba-s3003-arduino-microcontrolador-15782-MCO20108633256_062014-F.jpg)

การควบคุมและหลักการทำงานของ Servo motor ตัวเซอร์โวมอเตอร์โดยตัวเซอรโวมอเตอร์ควบคุมโดยเราต้องทำการส่งสัญญาณที่เป็น Pulse คือสัญญาณที่เป็นรูปสี่เหลี่ยมเพื่อทำการควบคุม

1. current control loop เป็นส่วนของการควบคุมกระแสไฟที่จ่ายให้กับมอเตอร์ซึ่งจะแปรผันทางแรงบิดโดยรับสัญญาณ analog ที่มาจาก output ที่ได้ของ speed control loop (KV)
2. Speed control loop เป็นส่วนของการควบคุมความเร็วของมอเตอร์ โดยรับสัญญาณ Analog มาจาก output ของ Position control loop มาเปรียบเทียบกับ speed feedback จาก Encoder

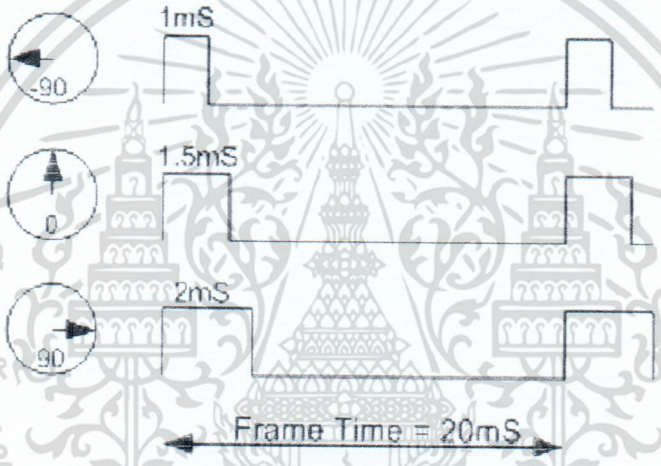
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Position control loop เป็นส่วนของการควบคุมตำแหน่งโดยรับสัญญาณมาจาก signal command อาจจะเป็นสัญญาณ Analog หรือสัญญาณ Pulse มาเปรียบเทียบกับ Position feed back จาก encoder

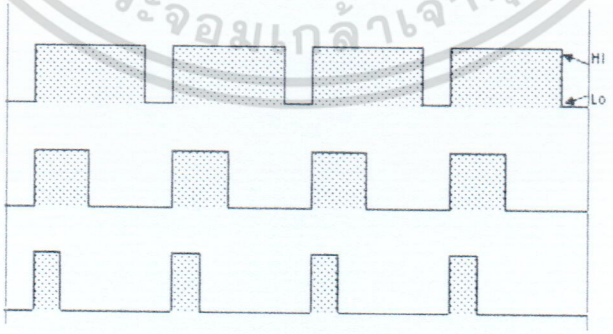
PWM (Pulse Width Modulation) คือ คือสัญญาณที่สามารถปรับความกว้างของ Pulse ได้ ที่เรียกว่า Duty cycle ซึ่งจะเป็นตัวกำหนดปริมาณของพลังงานไฟฟ้า ที่จ่ายให้กับอุปกรณ์ เช่น มอเตอร์และหลอดไฟ ใช้ควบคุมความเร็วของมอเตอร์ และความสว่างของหลอดไฟ เป็นต้น

ในการควบคุมเซอร์โวมอเตอร์เราใช้ความกว้างของ Pulse ในการควบคุมให้หมุนในองศาที่ต้องการ

1. สัญญาณPulse ควบคุมตำแหน่งมีความกว้างของสัญญาณPulse อยู่ระหว่าง 1-2 mS
2. สัญญาณPulse ควบคุมตำแหน่งจะต้องถูกส่งต่อเนื่องทุกระยะเวลาประมาณ 20 mS



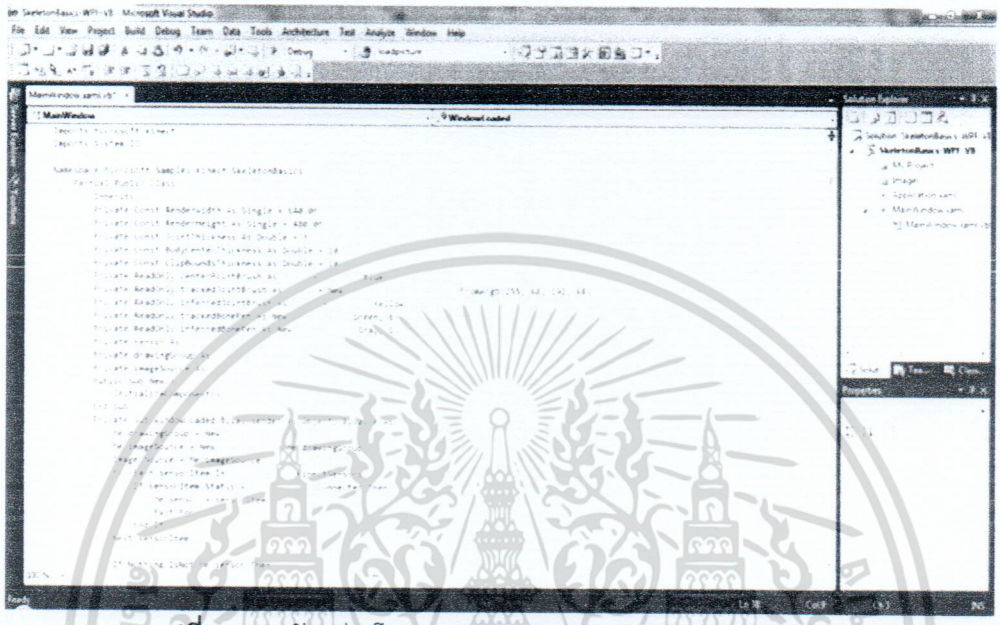
รูปที่ 2.13 การส่งสัญญาณควบคุมเซอร์โวมอเตอร์



รูปที่ 2.14 รูปตัวอย่าง PWM

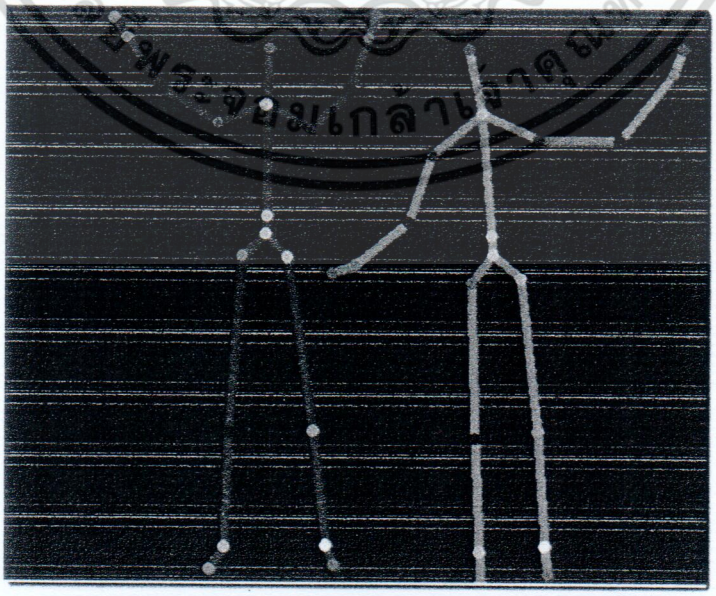
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โปรแกรมที่ใช้ในการทำงาน คือ Microsoft Visual Studio 2010 โดยภาษาที่ใช้เขียนจะเป็น C#



รูปที่ 2.15 ตัวอย่างโปรแกรม Microsoft Visual Studio 2010
(ที่มา : ผู้จัดทำ)

2.5 Skelton Tracking

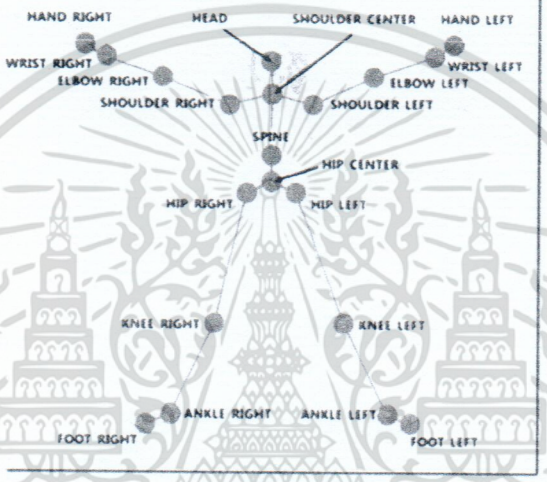


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.16 Skelton

(ที่มา : http://midimotion.blogspot.com/2013/05/midi-library-to-play-music.html)

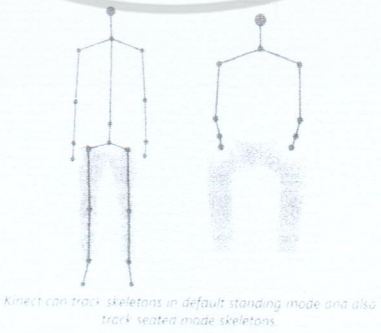
การจับร่างกายผู้ใช้ของ Kinect แปลงออกมาเป็น Skelton โดยที่เมื่อผู้ใช้ยืน Skelton ที่ได้จะมีข้อต่อ 20 จุด ดังรูป 2.13 จะประกอบด้วย ศีรษะ หัวไหล่ขวา หัวไหล่ซ้าย บริเวณแกนกลางไหล่ มือขวา มือซ้าย ศอกขวา ศอกซ้าย ข้อมือขวา ข้อมือซ้าย กระดูกสันหลัง สะโพก สะโพกขวา สะโพกซ้าย หัวเข่าซ้าย หัวเข่าขา ข้อเท้าขวา ข้อเท้าซ้าย เท้าขวา เท้าซ้าย



รูปที่ 2.17 รูปแสดงข้อต่อของ Skelton

(ที่มา : https://msdn.microsoft.com/en-us/library/jj131025.aspx)

แต่ในขณะที่จับผู้ใช้ในท่านั่งจะจับได้เพียง 10 จุด คือ มือซ้าย มือขวา ข้อมือซ้าย ข้อมือขวา ศอกซ้าย ศอกขวา หัวไหล่ซ้าย หัวไหล่ขวา แกนกลางไหล่ ศีรษะ



รูปที่ 2.18 เปรียบเทียบข้อต่อระหว่างท่ายืนและท่านั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(**ที่มา** : https://msdn.microsoft.com/en-us/library/hh973077.aspx)

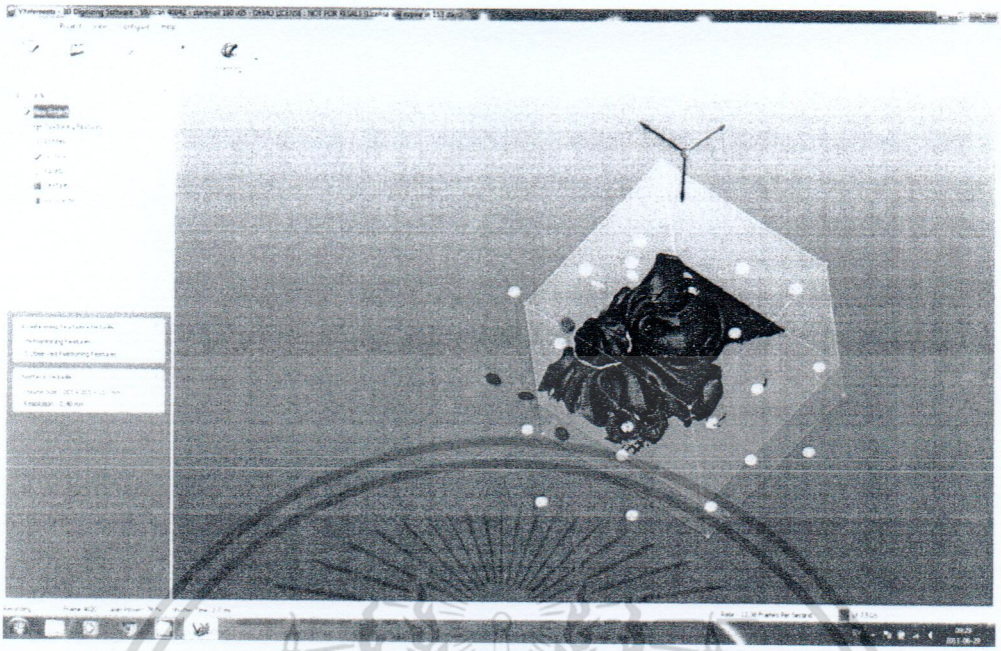
2.6 เครื่องพิมพ์ 3 มิติ (3D Printers)

เทคโนโลยีการพิมพ์แบบสามมิติ (3D printing) คือนวัตกรรมเปลี่ยนโลกที่ได้รับการกล่าวถึงมากที่สุดในช่วงหลายปีที่ผ่านมา เพราะเทคโนโลยีที่ถูกคิดค้นมาตั้งแต่ปี 1984 กำลังขยายการใช้งานเข้าสู่ผู้ใช้ระดับครัวเรือนมากขึ้นในราคาที่ต่ำลงเรื่อยๆ จนมีผู้นำไปพัฒนาต่อยอดและประยุกต์ใช้ในแวดวงต่างๆ อย่างกว้างขวาง ตั้งแต่ของเล่น ตุ๊กตาคนจริงย่อส่วน เครื่องประดับ รองเท้า เฟอร์นิเจอร์ ไปจนถึงชิ้นส่วนรถยนต์หรือแม้กระทั่งอาหารและอวัยวะเทียมซึ่งผลิตขึ้นเฉพาะบุคคล

สำหรับเทคนิคพื้นฐานในการสร้างชิ้นงานของเครื่องพิมพ์สามมิติ (3D printing) จะเริ่มจากการนำวัสดุมาขึ้นรูปทีละชั้นตามแบบที่กำหนดในไฟล์คอมพิวเตอร์ เพื่อให้ปรับเปลี่ยนรายละเอียดและสั่งพิมพ์ในปริมาณที่ต้องการได้ทันที ทำให้มีค่าใช้จ่ายที่ต่ำกว่าการสั่งผลิตในโรงงานอุตสาหกรรมแบบเดิมซึ่งต้องสร้างเบ้าหล่อ (mold) ก่อนแล้วจึงฉีดวัสดุลงไป นอกจากนี้ เทคนิคการสร้างชิ้นงานด้วยการเติมวัสดุ (additive manufacturing) ของเครื่องพิมพ์สามมิติยังทำให้สูญเสียวัตถุดิบน้อยกว่าการผลิตแบบทั่วไปซึ่งมักเริ่มด้วยวัสดุที่เป็นบล็อกใหญ่และตัดส่วนที่ไม่ต้องการออกอีกด้วย

2 ขั้นตอนง่ายๆ ในการพิมพ์สามมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 โปรแกรมสร้างแบบจำลอง 3 มิติ
(ที่มา : http://www.tcdc.or.th/creativethailand/article/Insight/20010/?ref=fb-siam3dprinter)

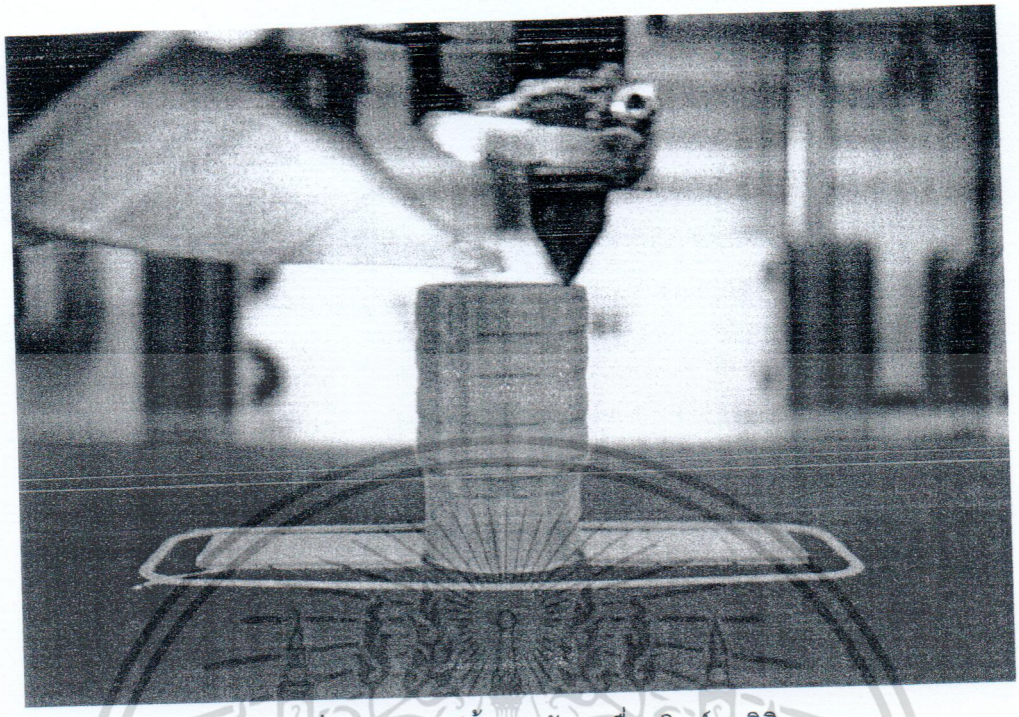
1. เตรียมไฟล์รูปจำลองสามมิติ (.stl)
   - สร้างแบบจำลองสามมิติด้วยโปรแกรมออกแบบสามมิติ*
   - ดาวน์โหลดไฟล์รูปจำลองสามมิติจากเว็บไซต์ที่ให้บริการ เช่น Thingiverse**
   - สแกนวัตถุต้นแบบเพื่อสร้างไฟล์รูปจำลองด้วยเครื่องสแกนสามมิติ

   *ซอฟต์แวร์สร้างแบบจำลองที่ได้รับความนิยมและเปิดให้ใช้งานฟรี เช่น Blender, SketchUp, 123D Design, 3D Canvas, Seamless3d ฯลฯ

   **Thingiverse คือเว็บไซต์ยอดนิยมที่เปิดให้สามารถอัพโหลด-ดาวน์โหลดไฟล์สามมิติ พร้อมฟังก์ชันแสดงผลไฟล์ในตัว โดยแบบจำลองสามมิติเหล่านี้บางแบบสามารถนำมาปรับแต่งได้ตามความต้องการ

2. ตั้งค่าและสั่งพิมพ์ด้วยเครื่องพิมพ์ที่ใช้เทคนิคและวัสดุที่เหมาะสมกับชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 พิมพ์ชิ้นงานด้วยเครื่องพิมพ์ 3 มิติ

(ที่มา : http://www.tcdc.or.th/creativethailand/article/Insight/20010/?ref=fb-siam3dprinter)

2.6.1 การพิมพ์แบบต่าง ๆ

2.6.1.1 การพิมพ์แบบหัวฉีด (FDM : Fused Deposition Modeling)

เทคนิค : ทำงานด้วยกลไกหัวฉีด (nozzle) ซึ่งจะทำความร้อนเพื่อให้วัสดุที่มีลักษณะเป็นเส้น (filament) อ่อนตัวลง แล้วจึงสร้างชิ้นงานขึ้นทีละชั้นโดยเริ่มจากฐาน

วัสดุ : พลาสติกชนิดต่างๆ โดยเฉพาะ PLA และ ABS ปัจจุบันเริ่มมีการนำเทคนิคมาต่อยอดเพื่อใช้กับวัสดุดิบอาหารอย่างช็อกโกแลต ไอซิ่ง ชีส ฯลฯ รวมถึงคอนกรีตสำหรับสร้างอาคาร

เหมาะสำหรับสร้างตัวต้นแบบ (rapid prototype) ด้วยพลาสติก ชิ้นงานไม่ละเอียดเท่าการพิมพ์แบบอื่นๆ จึงมีพื้นผิวไม่เรียบและจำเป็นต้องมีการขัดเก็บงานก่อนในกรณีที่จะนำไปใช้งานจริง FDM เป็นเทคนิคการพิมพ์ที่ได้รับความนิยมมากที่สุด เพราะดัดแปลงใช้ได้กับวัสดุหลากหลายประเภท และมีต้นทุนต่ำกว่าการพิมพ์แบบอื่นๆ ทั้งในแง่ของวัสดุและเครื่องพิมพ์

2.6.1.2 การพิมพ์ด้วยรังสีอัลตราไวโอเลต (SLA : Stereolithography)

เทคนิค : สร้างชิ้นงานโดยยิงลำแสงอัลตราไวโอเล็ตให้ผิวน้ำเรซินแข็งตัวทีละชั้นและเชื่อมต่อกับชั้นก่อนหน้า

วัสดุ : ใช้ได้กับเรซินอย่างเดียวเท่านั้น

เหมาะสำหรับการสร้างชิ้นส่วนกลไกต่างๆ เพื่อนำไปใช้ทดสอบการทำงานของเครื่องต้นแบบและสามารถสร้างวัตถุเพื่อเป็นชิ้นส่วนจริงในเครื่องมือต่างๆ ได้ เพราะมีความละเอียดมากกว่าและผลิตชิ้นงานได้เร็วกว่าการพิมพ์แบบ FDM มาก งานจึงมีผิวเรียบแต่ก็มีต้นทุนสูงกว่าทั้งในแง่เครื่องพิมพ์และวัสดุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1.3 การพิมพ์ด้วยแสงเลเซอร์ (SLS: Selective Laser Sintering)

เทคนิค : เครื่องพิมพ์จะยิงแสงเลเซอร์ลงบนผงวัสดุให้เกิดการหลอมละลายเฉพาะจุดและเกิดการเกาะติดกันทีละชั้น

วัสดุ : ผงโลหะ แก้ว เซรามิก พลาสติก อีลาสโตเมอร์ (โพลิเมอร์ที่มีความยืดหยุ่นคล้ายยาง)

ชิ้นงานที่ได้มีความคงทนกว่าการพิมพ์แบบ SLA เหมาะสำหรับทำสร้างชิ้นงานเพื่อใช้จริง เช่น เครื่องประดับเงินและทองคำ ตุ๊กตาย่อส่วนคนจริงจากเซรามิก เครื่องดนตรีอย่างกีต้าร์ ไวโอลิน ฟรุต ฯลฯ

2.6.1.4 การพิมพ์ด้วยการซ้อนแผ่นวัสดุ (LOM : Laminated Object Manufacturing)

เทคนิค : ใช้เลเซอร์หรือมีดตัดวัสดุที่มีลักษณะเป็นแผ่นบางทีละชั้น และเชื่อมแต่ละชั้นด้วยกาว

วัสดุ : แผ่นกระดาษ ไม้ โลหะ

เหมาะสำหรับสร้างชิ้นงานเพื่อเป็นวัตถุต้นแบบ เพราะจุดเด่นของการพิมพ์แบบ LOM คือ ความเร็ว แต่ความละเอียดของงานยังต้องอาศัยการเก็บงานที่ดีด้วย ต้นทุนของวัสดุค่อนข้างต่ำเมื่อเทียบกับการพิมพ์แบบอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

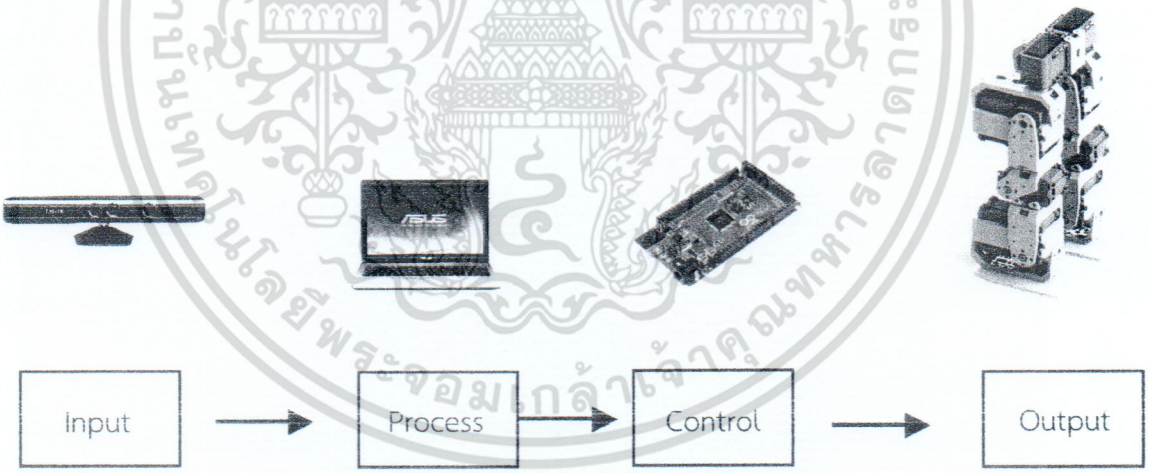
บทที่3

การออกแบบและการสร้าง

ในส่วนของการออกแบบของหุ่นยนต์ควบคุมด้วย Kinect และ Arduino ได้แบ่งการออกแบบเป็น 3 ส่วนสำคัญดังนี้คือ การออกแบบและการสร้าง ส่วนการออกแบบโปรแกรมควบคุมการเคลื่อนไหวของหุ่นยนต์ และส่วนของชิ้นงานหุ่นยนต์

3.1 การออกแบบและการสร้าง

จากที่ได้ศึกษาหลักการทำงานของตัวอุปกรณ์ Kinect เซนเซอร์ตรวจจับท่าทางของผู้ใช้แล้วรับค่ามายังคอมพิวเตอร์ เขียนโค้ดโปรแกรมให้ Kinect จับ skeleton tracking 20 จุดของผู้ใช้ แล้วส่งค่าต่างๆ ไปยังบอร์ดไมโครคอนโทรลเลอร์ Arduino เพื่อควบคุมหุ่นยนต์ เพื่อให้หุ่นยนต์สามารถเคลื่อนไหวตามผู้ใช้

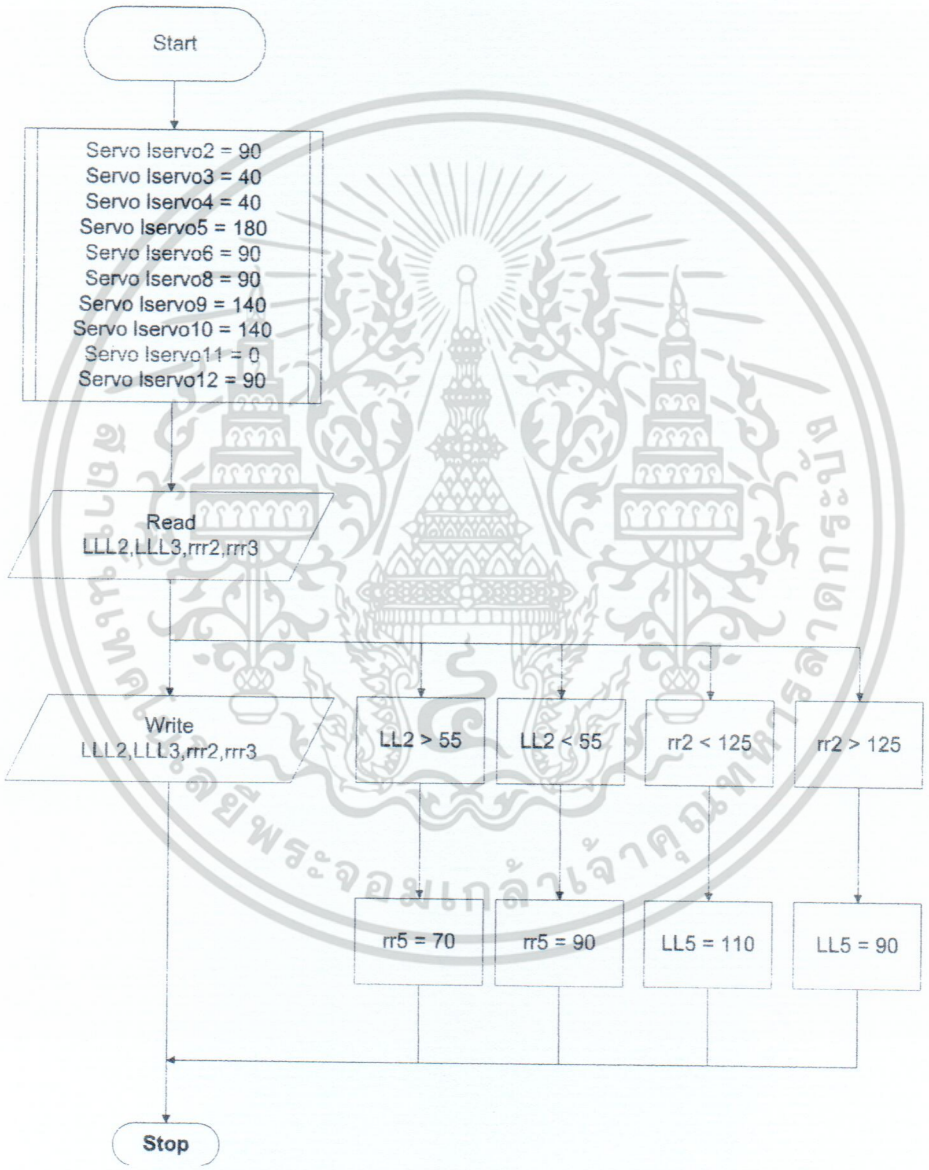


รูป 3.1 ผังการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบโปรแกรม

3.2.1 ส่วนของ Arduino



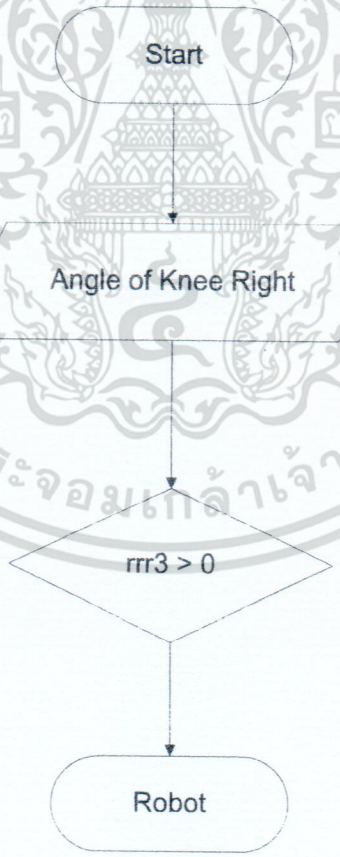
รูปที่ 3.2 ผังการทำงานของโปรแกรม Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นการทำงานด้วยการกำหนดค่าเริ่มต้นของเซอร์โวแต่ละตัว ต่อมารับค่าจาก Kinect ซึ่งมีหน่วยเป็นองศาและแสดงผลออกมาตามที่ค่าได้ประมวลผลไว้ โดยแยกเป็นเงื่อนไขเพื่อรับน้ำหนักของหุ่นเมื่อยกขาขึ้น โดยเมื่อยกขาข้างซ้ายขึ้น (LL2 > 55) จะทำให้เซอร์โวอีกตัว(rr5)ทำงานหมุนไป 20 องศา (rr5 = 70) และเมื่อยกขาข้างขวาขึ้น (rr2 < 125) จะทำให้เซอร์โวอีกตัว(LL5) ทำงานหมุนไป 20 องศา (LL 5 = 110)

3.2.2 ส่วนของ Kinect

ผังการทำงานของโปรแกรม เมื่อเริ่มการทำงาน Kinect จะทำการจับผู้ใช้งาน โดยที่จะจับ เป็น Skelton Tacking 20 จุดแต่ในส่วนของโปรแกรมนี้เราจะ ใช้เพียง 6 จุดคือ หนึ่งส่วนของเอวซ้ายผู้ใช้ จุดที่สองเข่าซ้ายของผู้ใช้ จุดที่สามเท้าซ้ายของผู้ใช จุดที่สี่เอวขวาของผู้ใช้ จุดที่ห้าหัวเข่าขวาของผู้ใช้ จุดที่หกเท้าขวาของผู้ใช้ โดย Kinect จะจับจุดทั้ง 6 ตายตัวแล้ว จะส่งเข้าในส่วนของการประมวลผลโดยเราจะใช้ทางคอมพิวเตอร์ในการประมวลผลโดยเมื่อทางคอมพิวเตอร์รับค่าจากตัว Kinect ก็จะส่งค่าไปเชื่อมต่อกับตัวหุ่นยนต์ให้หุ่นยนต์ขยับขึ้นลงตามการเคลื่อนไหว ของตำแหน่งของเอวซ้าย เอวขวา หัวเข่าซ้าย หัวเข่าขวา เท้าซ้าย และเท้าขวา



รูปที่ 3.3 ผังการทำงานของโปรแกรม Kinect ส่วนหัวเข่าขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นด้วยรับค่ามาจาก Kinect เป็นค่าองศาไปคำนวณในสมการของหัวเข่าที่ได้กำหนดไว้ เมื่อผ่านเงื่อนไขมากกว่า 0 เซอร์โวจะแสดงค่าตามที่ Kinect ได้รับมาและจะหยุดนิ่งเมื่อไม่ผ่านเงื่อนไข(เซอร์โวมีค่าเริ่มต้นที่กำหนดไว้ = 140 องศา)

รูปที่ 3.4 ผังการทำงานของโปรแกรม Kinect ส่วนหัวเข่าซ้าย

เริ่มต้นด้วยรับค่ามาจาก Kinect เป็นค่าองศาไปคำนวณในสมการของหัวซ้ายที่ได้กำหนดไว้ เมื่อผ่านเงื่อนไขมากกว่า 0 เซอร์โวจะแสดงค่าตามที่ Kinect ได้รับมาและจะหยุดนิ่งเมื่อไม่ผ่านเงื่อนไข(เซอร์โวมีค่าเริ่มต้นที่กำหนดไว้ = 40 องศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 ผังการทำงานของโปรแกรม Kinect ส่วนเอวด้านซ้าย

เริ่มต้นด้วยรับค่ามาจาก Kinect เป็นค่าองศาไปคำนวณในสมการของเอวด้านซ้ายที่ได้กำหนดไว้ เมื่อผ่านเงื่อนไขมากกว่า 0 เซอร์โวจะแสดงค่าตามที่ Kinect ได้รับมาและจะหยุดนิ่งเมื่อไม่ผ่านเงื่อนไข (เซอร์โวมีค่าเริ่มต้นที่กำหนดไว้ = 40 องศา)

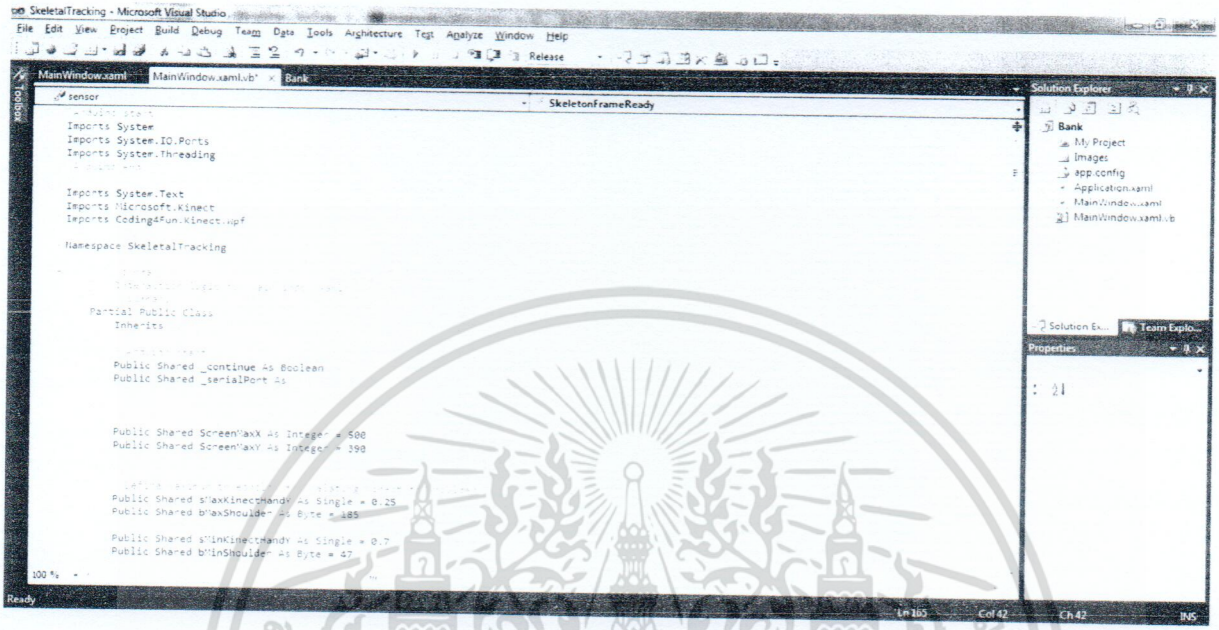
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6 ผังการทำงานของโปรแกรม Kinect ส่วนเอวด้านขวา

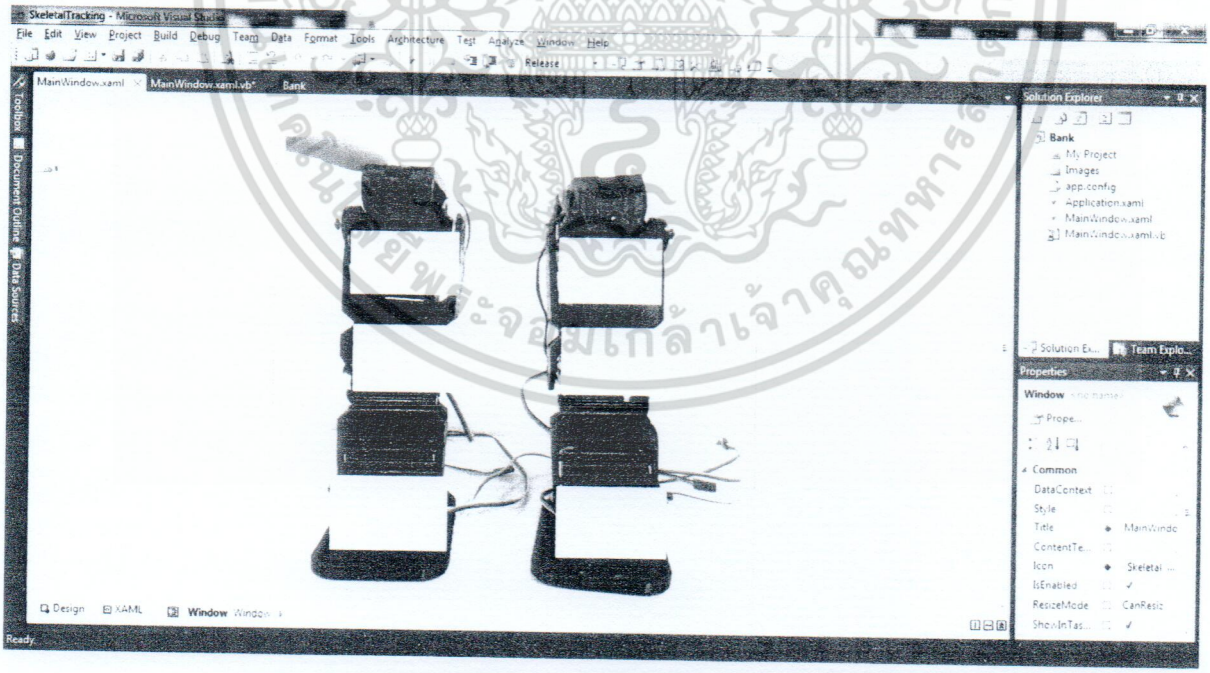
เริ่มต้นด้วยรับค่ามาจาก Kinect เป็นค่าองศาไปคำนวณในสมการของเอวด้านซ้ายที่ได้กำหนดไว้ เมื่อผ่านเงื่อนไขมากกว่า 0 เซอร์โวจะแสดงค่าตามที่ Kinect ได้รับมาและจะหยุดนิ่งเมื่อไม่ผ่านเงื่อนไข (เซอร์โวมีค่าเริ่มต้นที่กำหนดไว้ = 140 องศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของหน้าต่างโปรแกรม Microsoft Visual Basic



รูปที่ 3.7 ภาพแสดงหน้าต่างโปรแกรม Microsoft Visual Basic ส่วนโค้ดโปรแกรม



รูปที่ 3.8 ภาพแสดงหน้าต่างโปรแกรม Microsoft Visual Basic ส่วนหน้าต่างแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของหน้าต่างโปรแกรม Arduino

รูปที่ 3.9 ภาพแสดงหน้าต่างโปรแกรม Arduino ส่วนโค้ดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

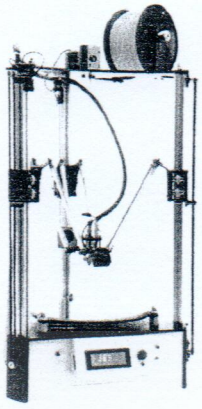
3.3 การออกแบบชิ้นงาน

ในส่วนของการออกแบบชิ้นงาน ในส่วนนี้ได้นำโครงสร้างของหุ่นยนต์ Bioloid robot มาเป็นต้นแบบในการออกแบบ ซึ่งชิ้นงานส่วนใหญ่จะใช้พลาสติก PLA ดังแสดงรูปที่ 3.10 ในการทำ โดยใช้เครื่องพิมพ์ 3D ดังแสดงรูปที่ 3.11 ทำให้ได้ชิ้นงานที่ใช้ในการประกอบกันเป็นตัวหุ่นยนต์ ดังแสดงในรูปที่ 3.12 – 3.14

รูปที่ 3.10 พลาสติก PLA

พลาสติก PLA เป็นพลาสติกชีวภาพ สามารถย่อยสลายได้และสามารถนำมาละลายหล่อขึ้นรูปต่างๆได้ด้วยหัวฉีดจากเครื่องปริ๊น 3D ที่มีอุณหภูมิสูงถึง 200 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

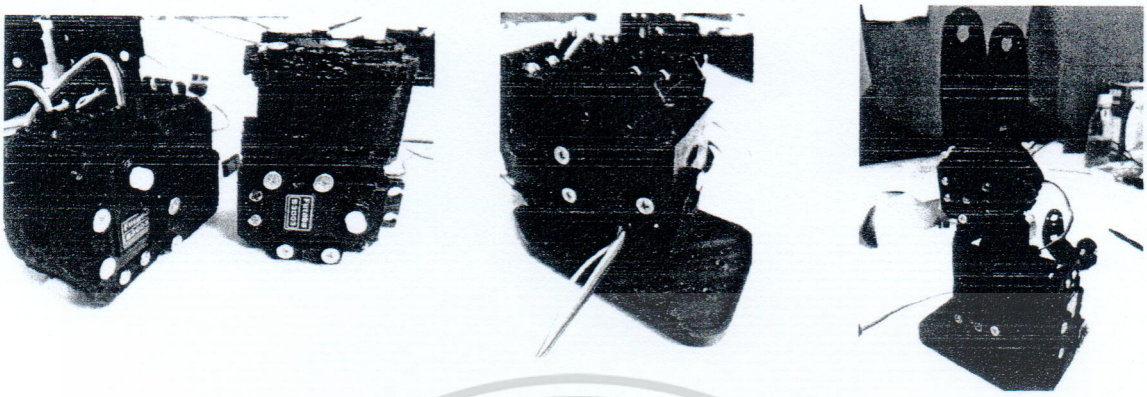


รูปที่ 3.11 เครื่องปริ๊น 3D

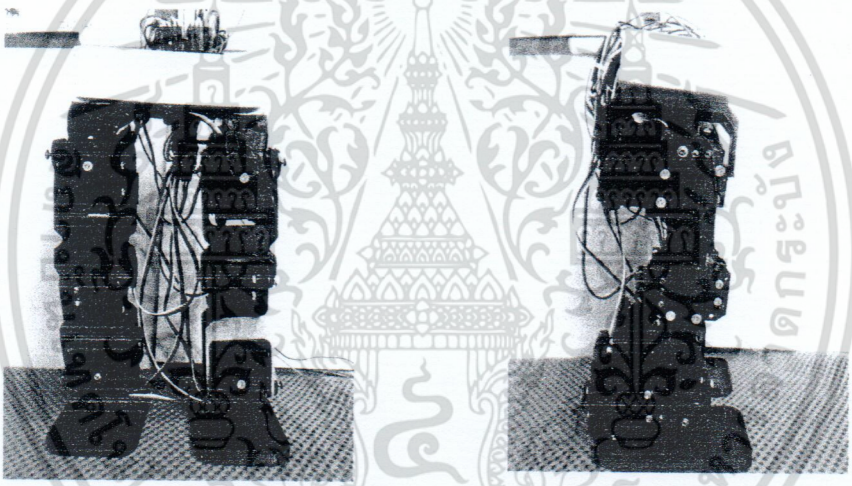
เครื่องปริ๊น 3D แบบ 3 แกน (X,Y,Z) และมีหัวฉีดขนาดเล็กที่มีอุณหภูมิสูงถึง 200 องศาเซลเซียส สามารถสั่งงานออกแบบได้ด้วยโปรแกรมออกแบบ 3D ต่างๆ

รูปที่ 3.12 ชิ้นส่วนของหุ่นยนต์ก่อนประกอบที่ทำมาจากเครื่องปริ๊น 3D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงชิ้นส่วนของหุ่นยนต์ที่ทำมาจากเครื่องปริ้น 3D



รูปที่ 3.14 หุ่นยนต์ที่ประกอบเสร็จสมบูรณ์ที่ทำมาจากเครื่องปริ้น 3D

จากที่เราได้เรียนรู้หลักการทำงานของตัวอุปกรณ์ต่างๆทั้งทฤษฎีของ Kinect เซนเซอร์ตรวจจับท่าทางของผู้ใช้แล้วรับค่ามายังคอมพิวเตอร์ เขียนโค้ดให้ Kinect จับskeleton 20 จุดของผู้ใช้ แล้วส่งค่าต่าง ๆ ไปยังบอร์ด Arduino เพื่อเปลี่ยนเป็นสัญญาณ Pulse แล้วส่งต่อไปที่ Servo หุ่นยนต์เพื่อให้หุ่นยนต์เคลื่อนไหวตาม การเคลื่อนไหวของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 3.1 ผังการทำงานของระบบ

(ที่มา : ผู้จัดทำ)

3.1 การออกแบบโปรแกรม

ในส่วนของการออกแบบโปรแกรมเราจะเขียนโปรแกรมผ่านโปรแกรม visual studio 2010 โดยเลือกใช้ภาษา VB ในการเขียน รับค่าจากเซนเซอร์ Kinect ในการจับผู้ใช้ แล้วเขียนโค้ดเพื่อทำการจับ Skeleton ตามจุดต่างๆของมนุษย์ 20 จุด แล้วประมวลผลส่งค่าออกไป ควบคุมการเคลื่อนไหวของหุ่นยนต์ตามจุดข้อต่อต่างๆ 16 จุดแต่ในส่วนของโปรแกรมของเรา จะใช้เพียง 8 จุดของผู้ใช้งานคือ ส่วนมือซ้าย มือขวา ข้อมือซ้าย ข้อมือขวา ข้อศอกซ้าย ข้อชอกขวา หัวไหล่ซ้าย หัวไหล่ขวา และเมื่อเราจับจุดทั้ง 8 จุดแล้วเราก็ทำการประมวลผลทางคอมพิวเตอร์โดยภาษาที่ใช้ในการเขียนโปรแกรมเป็นภาษา Visual Basic แล้วทำการเชื่อมต่อไปยังหุ่นยนต์ โดยค่าที่คอมพิวเตอร์ประมวลผลได้จะถูกส่งไปยัง บอร์ด Arduino ที่จะเป็นตัวสั่งให้ Servoแต่ละตัวทำงานตามที่เราได้เขียนโปรแกรมไว้ โดยเราจะทำการทดลองด้วยการกำหนดท่าทางต่างๆในส่วนบนของร่างกาย

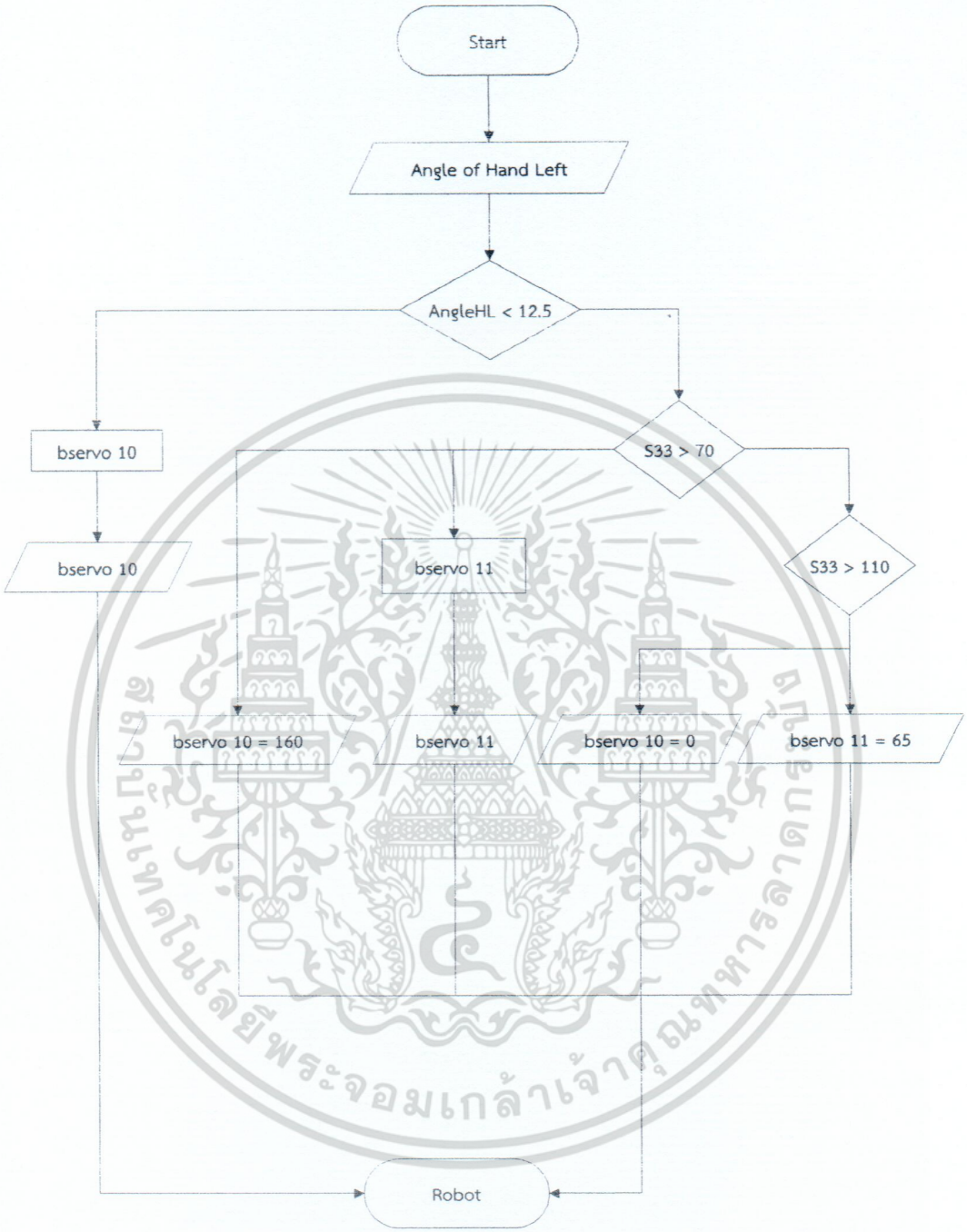
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 ผังการทำงานของโปรแกรม

(ที่มา : ผู้จัดทำ)

ผังการทำงานของโปรแกรม เมื่อเริ่มการทำงาน Kinect จะทำการจับผู้ใช้งาน โดยที่จะจับ เป็น Skelton Tacking 20 จุดแต่ในส่วนของโปรแกรมนี้เราจะ ใช้เพียง 8 จุด โดย Kinect จะจับจุดทั้ง 8 ตายตัวแล้ว จะส่งค่าในส่วนของการประมวลผลโดยเราจะใช้ทางคอมพิวเตอร์ในการประมวลผลโดยเมื่อทางคอมพิวเตอร์รับค่าจากตัว Kinect ก็จะส่งค่าไปเชื่อมต่อกับตัวหุ่นยนต์ให้หุ่นยนต์ขยับขึ้นลงตามการเคลื่อนไหว ของตำแหน่งของ มือซ้าย มือขวา ข้อมือซ้าย ข้อมือขวา ข้อศอกซ้าย ข้อซอกขวา หัวไหล่ซ้าย หัวไหล่ขวา

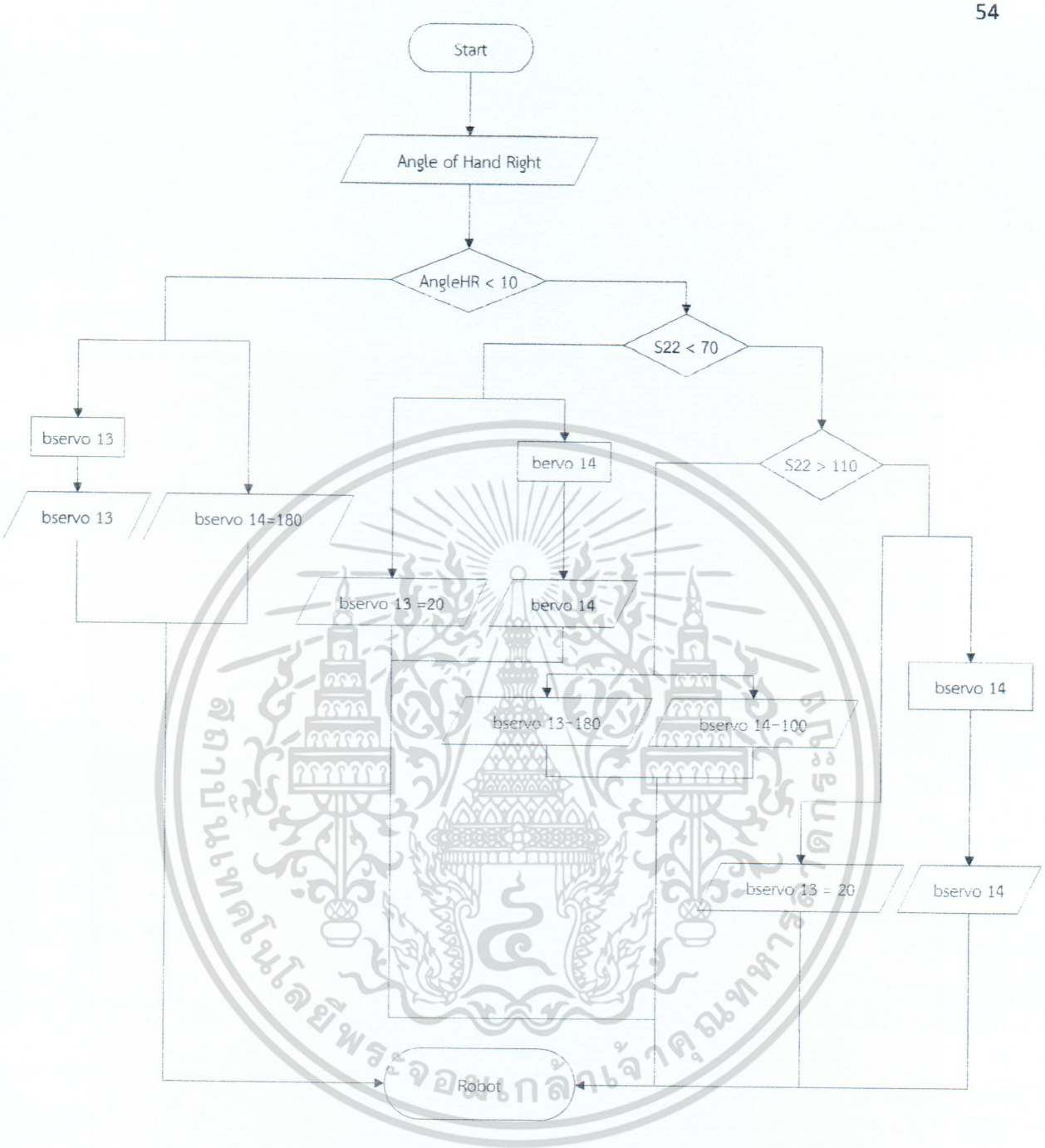
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ผังการทำงานของโปรแกรมส่วนของมือซ้าย

(ที่มา : ผู้จัดทำ)

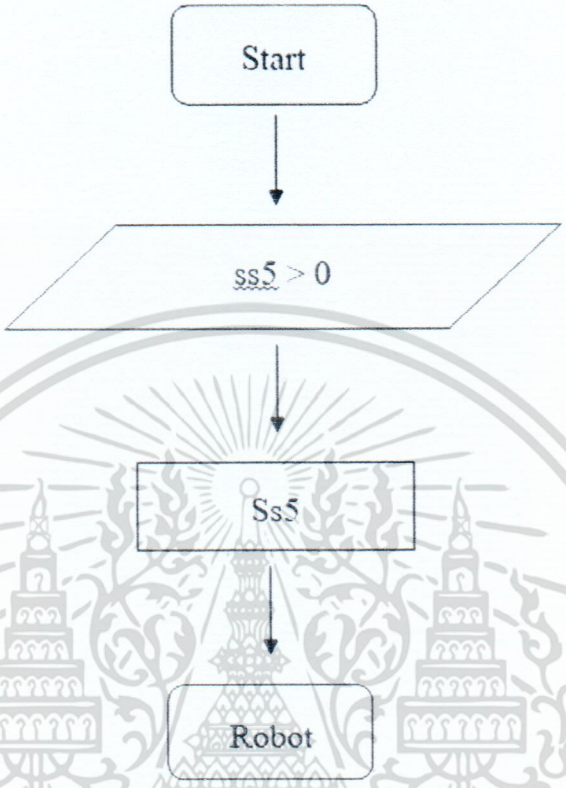
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ผังการทำงานของโปรแกรมส่วนของมือขวา

(ที่มา : ผู้จัดทำ)

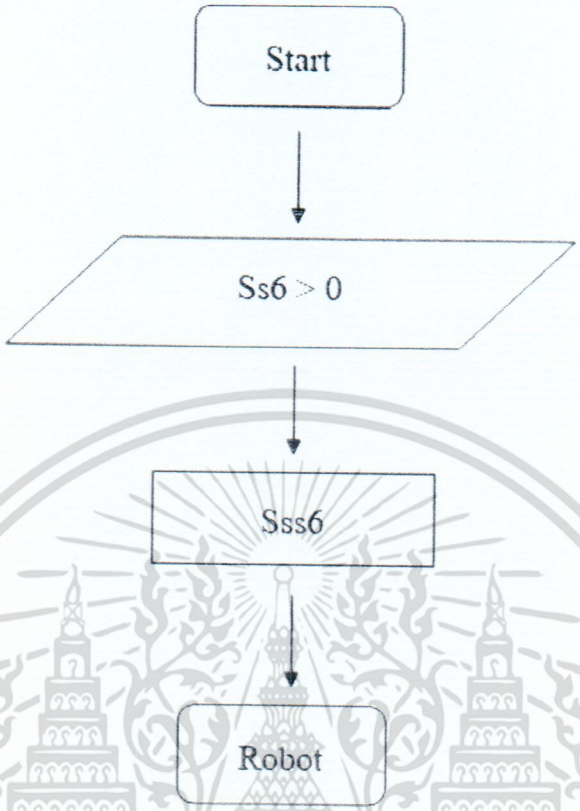
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ผังการทำงานของโปรมแกรมการเคลื่อนไหวข้อศอกของแขนขวา

(ที่มา : ผู้จัดทำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



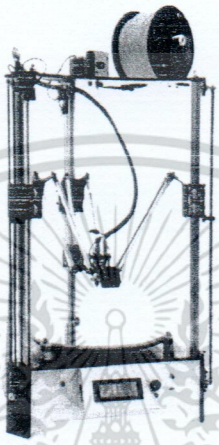
รูปที่ 3.6 ผังการทำงานของโปรมแกรมการเคลื่อนไหวข้อศอกของแขนซ้าย

(ที่มา : ผู้จัดทำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 สร้างหุ่นยนต์

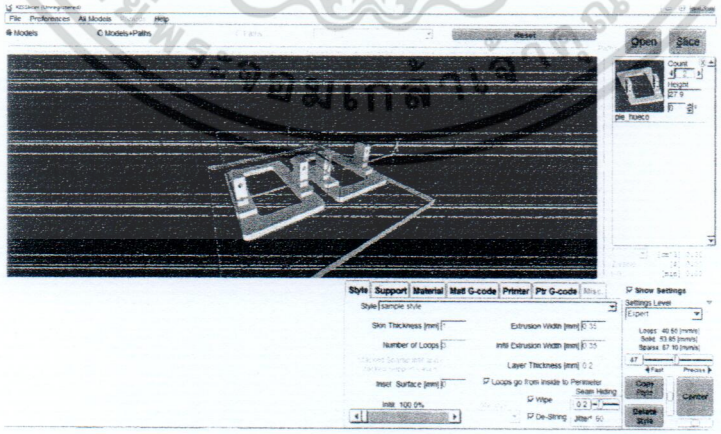
สร้างชิ้นส่วนต่าง ๆ จากเครื่องพิมพ์ 3 มิติ ซึ่งเครื่องพิมพ์ 3 มิติ เป็นเครื่องที่เอาไว้สำหรับขึ้นรูปโมเดลจากโปรแกรม 3 มิติ โดยใช้การฉีดพลาสติกเพื่อขึ้นรูปชิ้นงาน เป็นชิ้นส่วนประกอบ แล้วนำมาประกอบเป็นตัวหุ่นยนต์



รูปที่ 3.7 เครื่องพิมพ์ 3 มิติ

(ที่มา : http://www.siamreprap.com/newsite/wordpress/shop/3d-printer/delta-x-pro/)

นำไฟล์ของชิ้นส่วนของหุ่นยนต์ซึ่งเป็นไฟล์ 3 มิติ มาเปิดที่โปรแกรมตั้งค่าก่อนการพิมพ์เพื่อแปลงไฟล์และตั้งค่าการพิมพ์ ดังรูปที่ 3.8

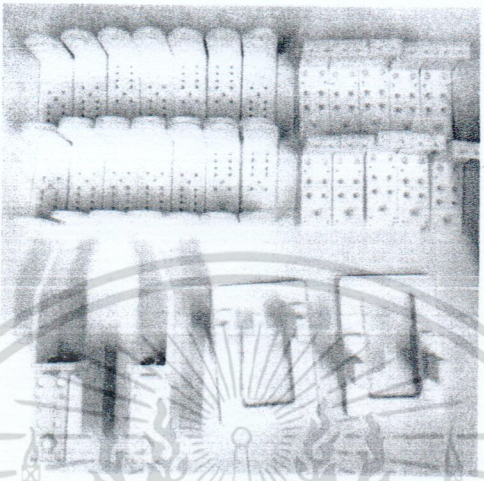


รูปที่ 3.8 หน้าต่างโปรแกรมตั้งค่าก่อนการพิมพ์ชิ้นงาน

(ที่มา : ผู้จัดทำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

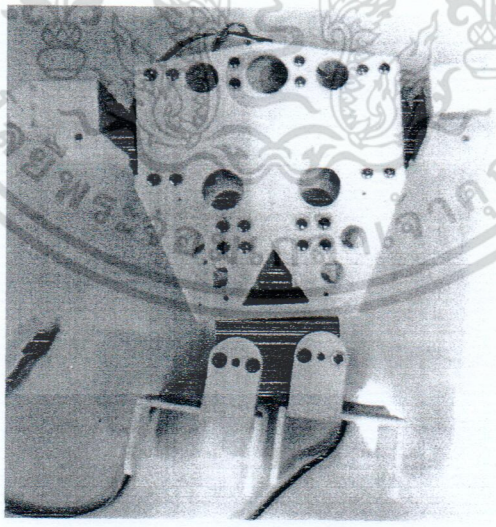
หลังจากตั้งค่าในการพิมพ์เสร็จเรียบร้อยแล้วก็นำไฟล์ที่ได้ไปพิมพ์ผ่านเครื่องพิมพ์ 3 มิติ ซึ่งเมื่อพิมพ์เสร็จแล้วจะได้ชิ้นงานออกมา ดังรูปที่ 3.9



รูปที่ 3.9 ส่วนประกอบของหุ่นยนต์ที่ยังไม่ได้ประกอบหลังพิมพ์เสร็จ

(ที่มา : ผู้จัดทำ)

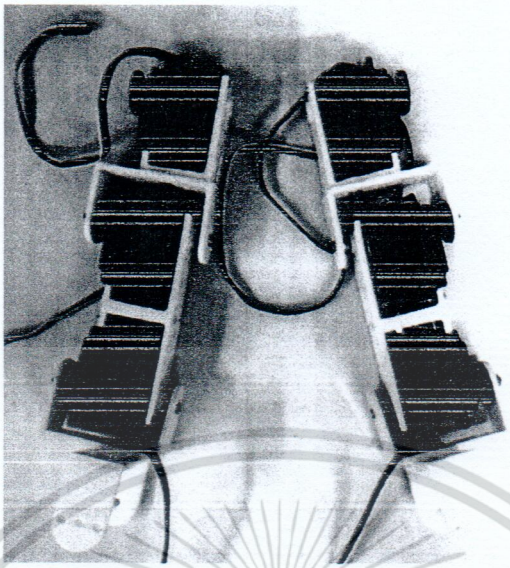
เมื่อได้ชิ้นส่วนครบตามที่ต้องการแล้วก็นำชิ้นส่วนเหล่านั้นมาประกอบเป็นตัวหุ่นยนต์ดังรูปต่อไปนี้



รูปที่ 3.10 ส่วนลำตัวของหุ่นยนต์ที่ได้ประกอบแล้ว

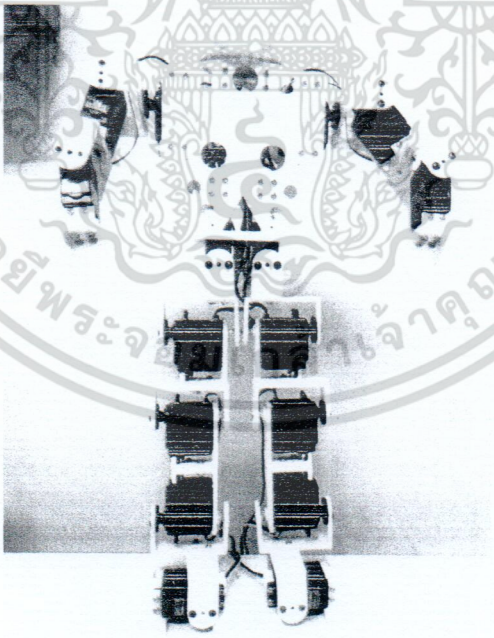
(ที่มา : ผู้จัดทำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ส่วนขาของหุ่นยนต์ที่ได้ประกอบแล้ว

(ที่มา : ผู้จัดทำ)



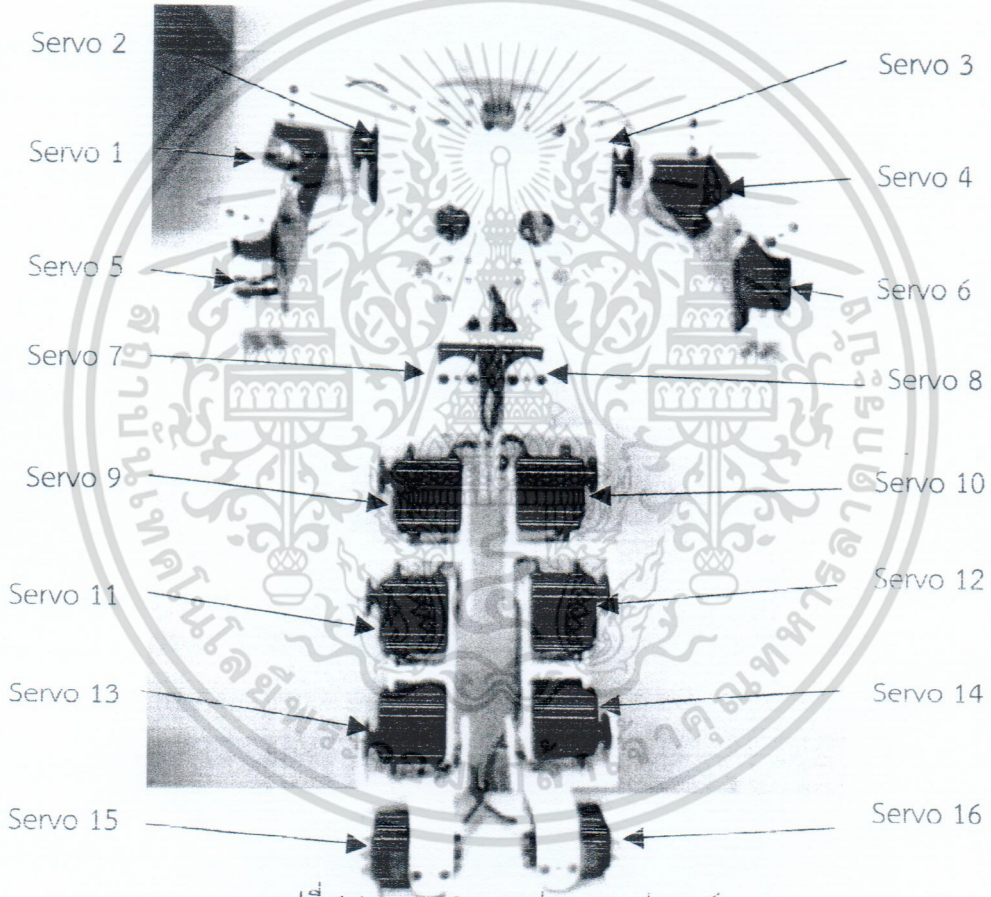
รูปที่ 3.12 หุ่นยนต์ที่ประกอบเสร็จสมบูรณ์แล้ว

(ที่มา : ผู้จัดทำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

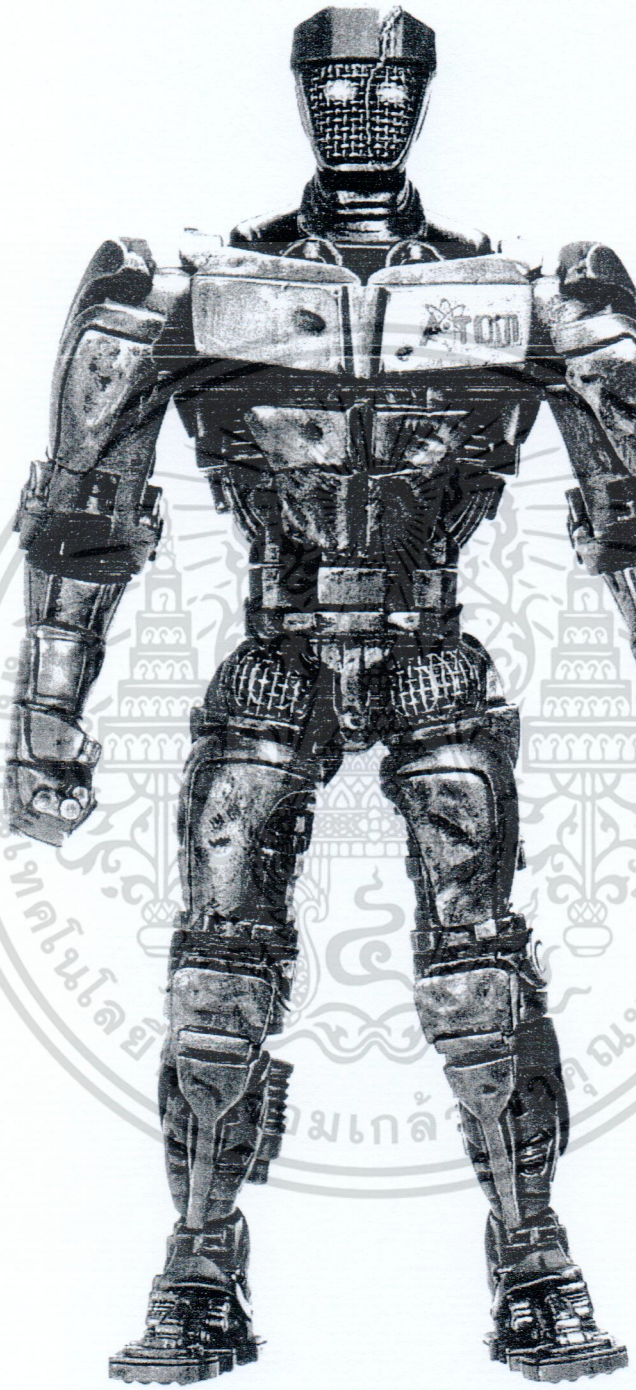
ในส่วนของการทดลองเราจะทำการทดลองการเคลื่อนที่ของหุ่นยนต์โดยการทำท่าทางต่างๆที่ได้กำหนดขึ้นมา แล้ววัดความเปลี่ยนแลงความกว้างของ Pulse ในแต่ละ Servo แล้วบันทึกผลลงในตาราง การทดลองพร้อมแสดงกราฟของ Servo แต่ละตัว



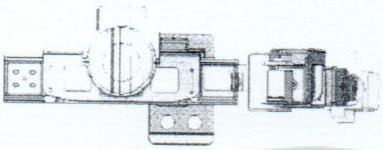
รูปที่ 4.1 แสดง Servo ต่างๆของหุ่นยนต์

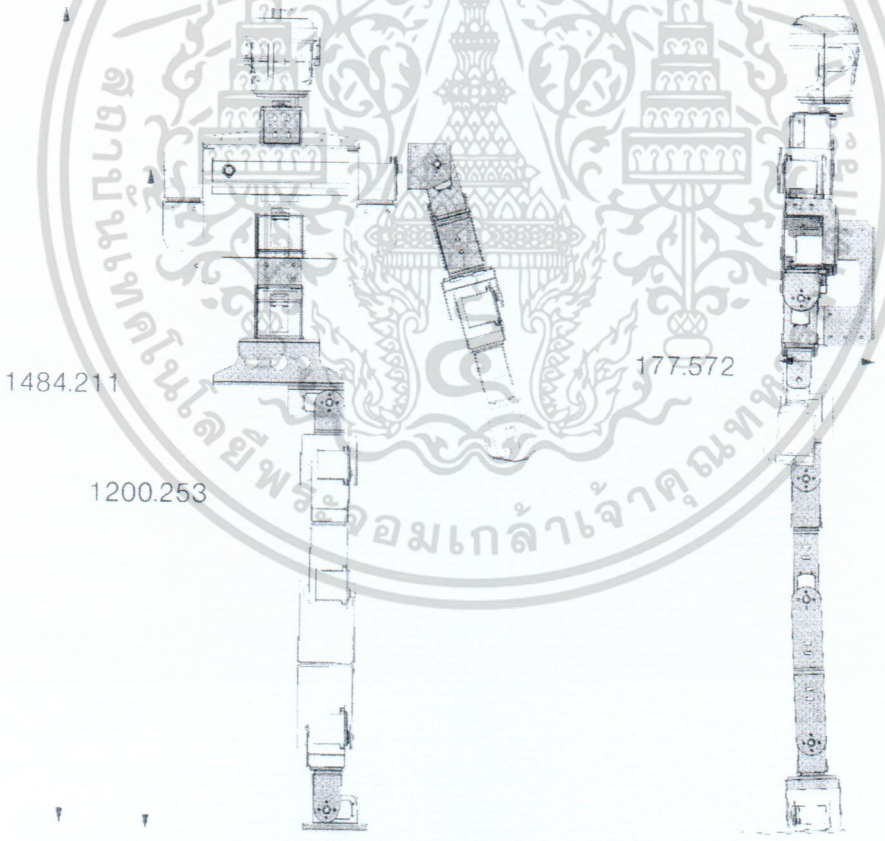
(ที่มา : ผู้จัดทำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

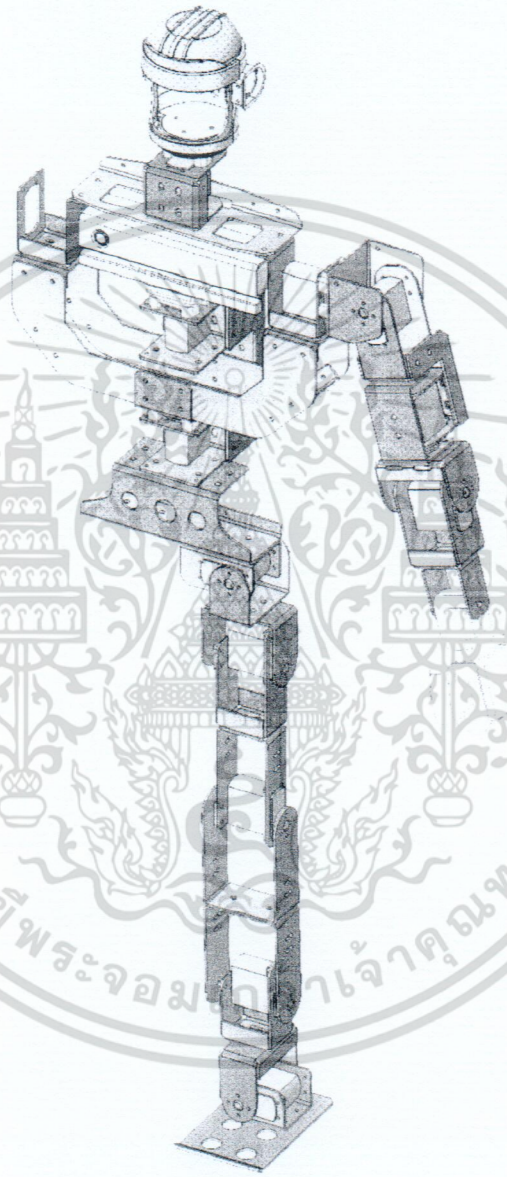


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

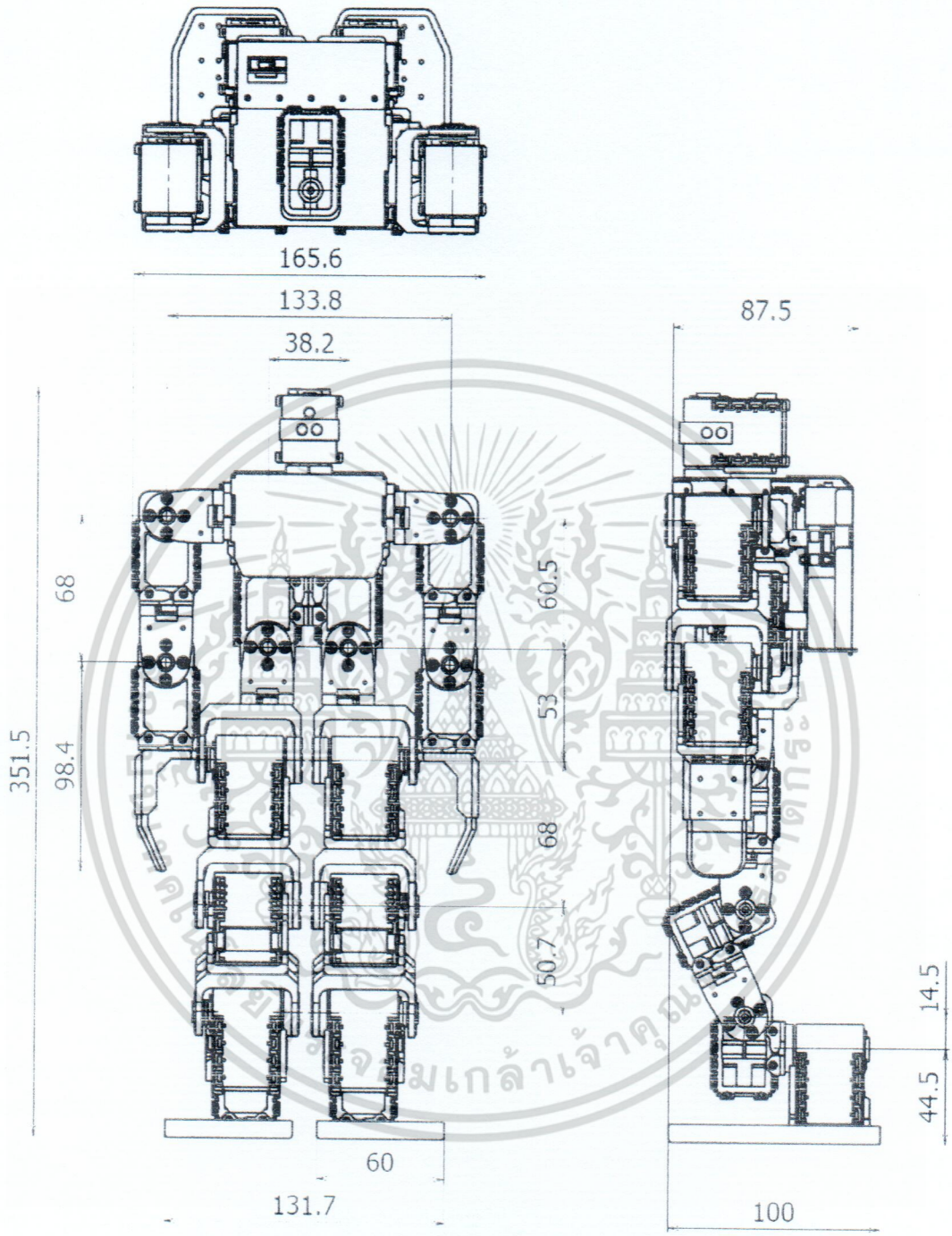




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

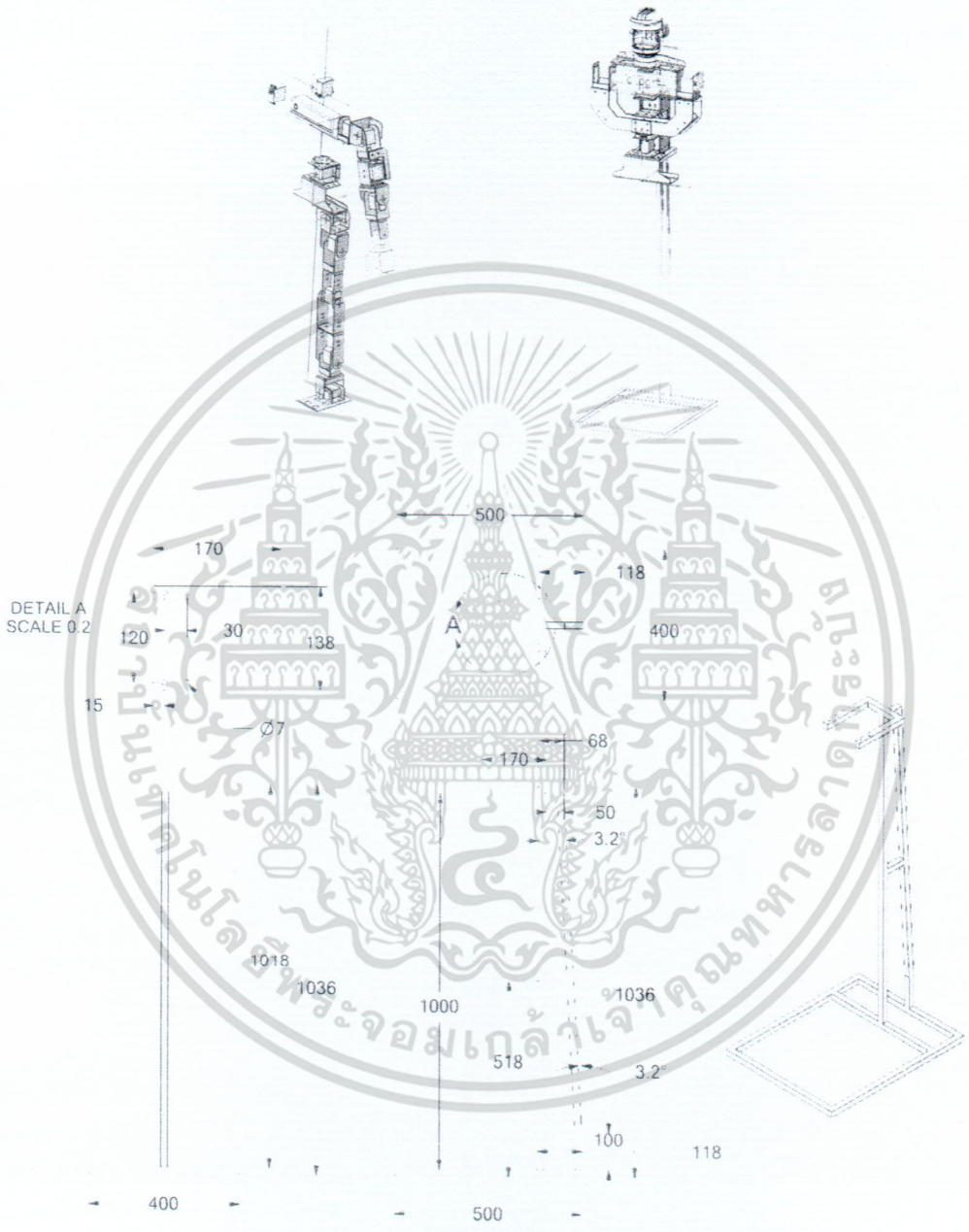


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

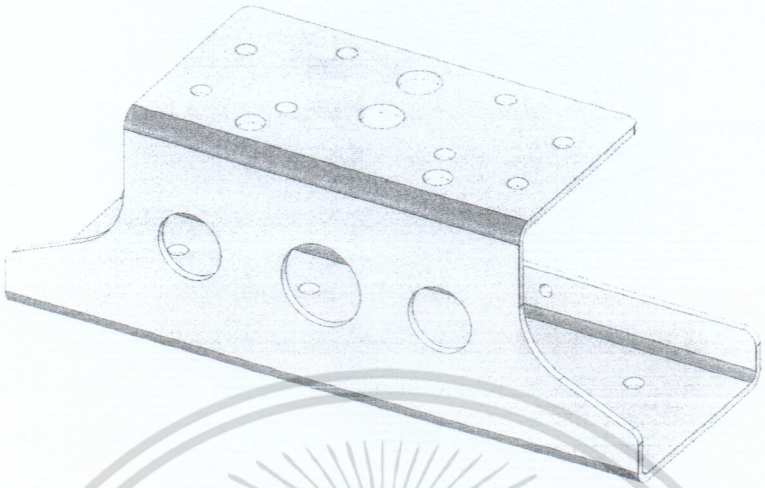


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดในส่วนต่างๆ

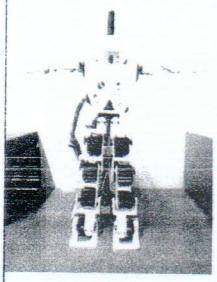


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ยกแขนสองข้างโดยแขนแต่ละข้างท่าเดียวกัน

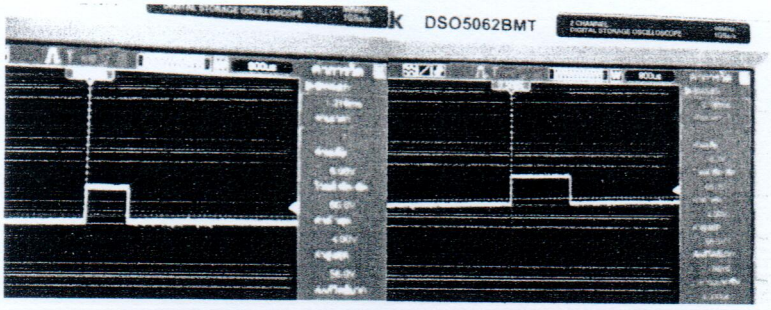


หมายเลข Servo	ค่าที่กำหนด	การทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน (%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	1.47ms	1.45ms	1.49ms	1.44ms	1.46ms	0.68
2	752us	752us	752us	752us	752us	0
3	1.47ms	1.15ms	1.64ms	1.57ms	1.45ms	1.36
4	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
5	1.21ms	1.24ms	1.24ms	1.22ms	1.23ms	1.65
6	1.47ms	1.44ms	1.46ms	1.47ms	1.45ms	1.36

ตารางที่ 4.1 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1

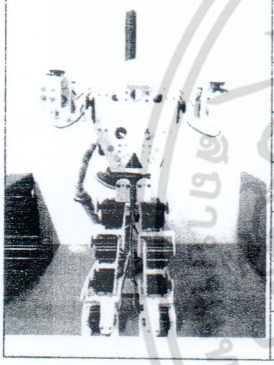
รูปที่ 4.2 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

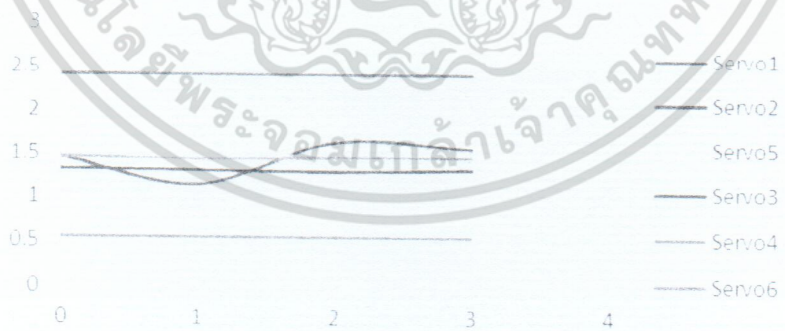


รูปที่ 4.3 กราฟแสดง pulse ของ Servo5

ตารางที่ 4.2 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่2

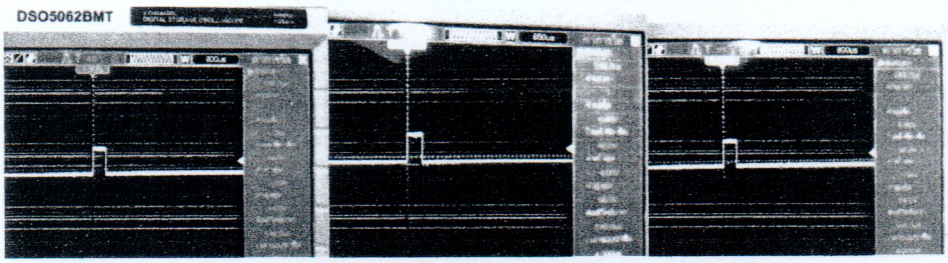


หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน (%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
2	1.47ms	1.15ms	1.64ms	1.57ms	1.45ms	1.36
5	1.47ms	1.45ms	1.49ms	1.44ms	1.46ms	0.68
3	1.32ms	1.31ms	1.30ms	1.32ms	1.23ms	0.75
4	548us	548us	548us	548us	548us	0
6	1.47ms	1.44ms	1.46ms	1.47ms	1.45ms	1.36



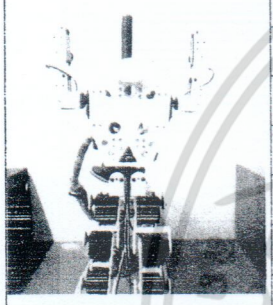
รูปที่ 4.4 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 กราฟแสดง pulse ของ Servo4

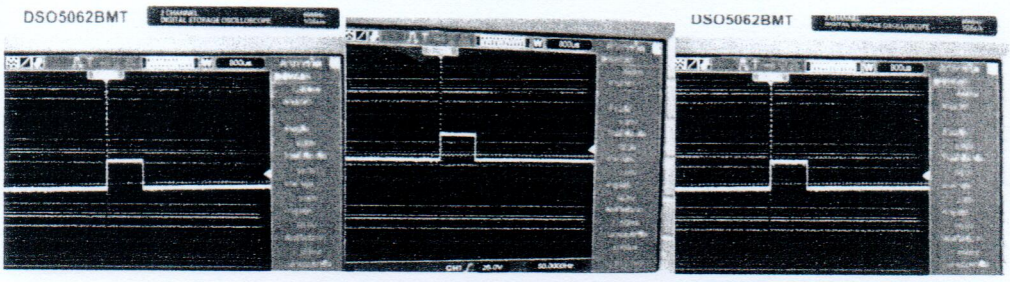
ตารางที่ 4.3 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่3



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
2	752us	752us	752us	752us	752us	0
5	1.47ms	1.44ms	1.46ms	1.47ms	1.45ms	1.36
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	548us	548us	548us	548us	548ms	0
6	1.47ms	1.45ms	1.49ms	1.44ms	1.46ms	0.68

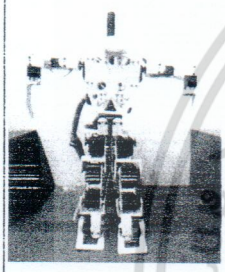
รูปที่ 4.6 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

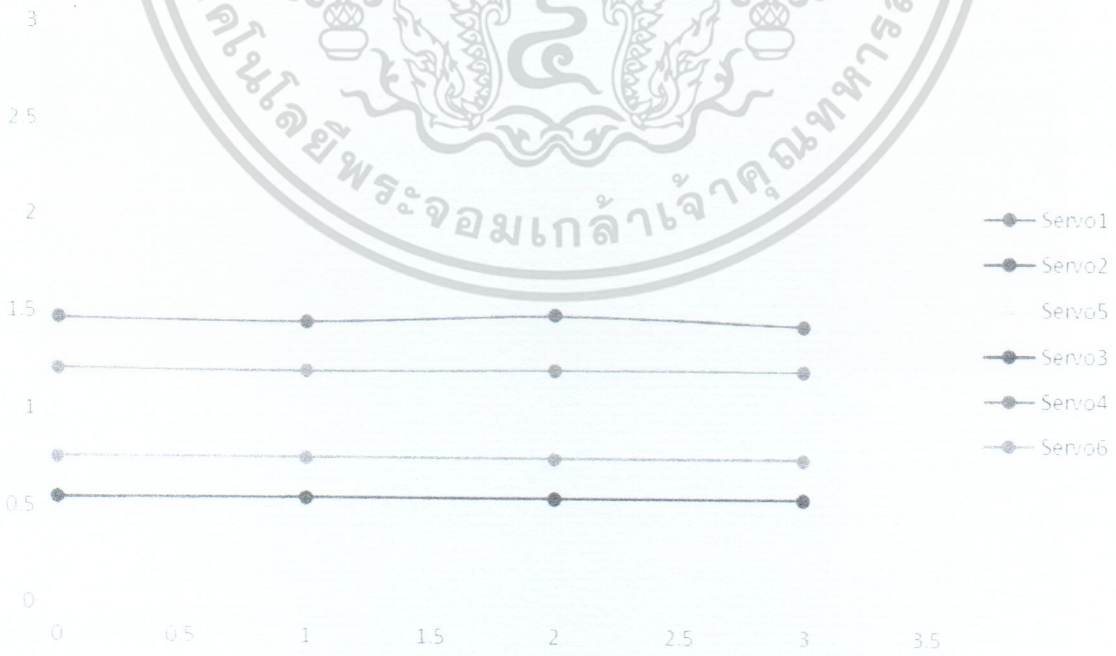


รูปที่ 4.7 กราฟแสดง pulse ของ Servo6

ตารางที่ 4.4 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4

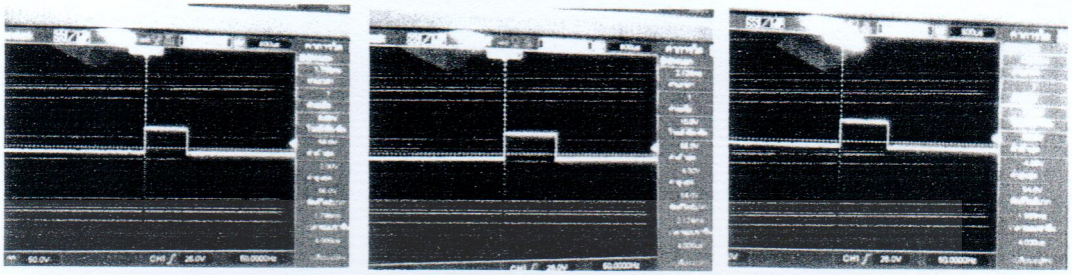


หมายเลข Servo	ค่าที่ กำหนด	ทดลอง ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
1	1.47ms	1.45ms	1.49ms	1.44ms	1.46ms	0.68
2	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
5	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
3	544us	548us	548us	548us	548us	0.73
4	1.21ms	1.20ms	1.21ms	1.21ms	1.20ms	0.82
6	752us	752us	752us	752us	752us	0



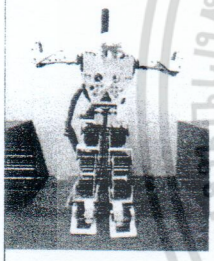
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.8 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4



รูปที่ 4.9 กราฟแสดง pulse ของ Servo5

ตารางที่ 4.5 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่5



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	1.47ms	1.48ms	1.45ms	1.46ms	1.46ms	0.68
2	752us	752us	752us	752us	752us	0
5	2.4ms	2.41ms	2.43ms	2.4ms	2.41ms	0.41
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	1.21ms	1.21ms	1.19ms	1.2ms	1.2ms	0.82
6	752us	750us	755ms	753us	752ms	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

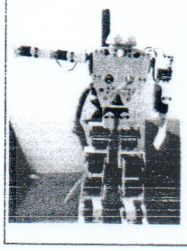
รูปที่ 4.10 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 5

รูปที่ 4.11 กราฟแสดง pulse ของ Servo3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ยกแขนขวาข้างเดียว

ตารางที่ 4.6 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1

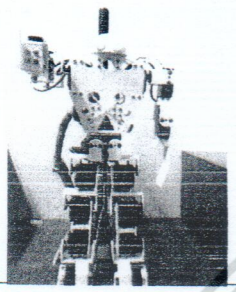
	หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
			ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
	1	1.47ms	1.45ms	1.46ms	1.45ms	1.45ms	1.36
	2	752us	752us	752us	752us	752us	0
	5	1.47ms	1.15ms	1.64ms	1.57ms	1.45ms	1.36

รูปที่ 4.12 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1

รูปที่ 4.13 กราฟแสดง pulse ของ Servo5

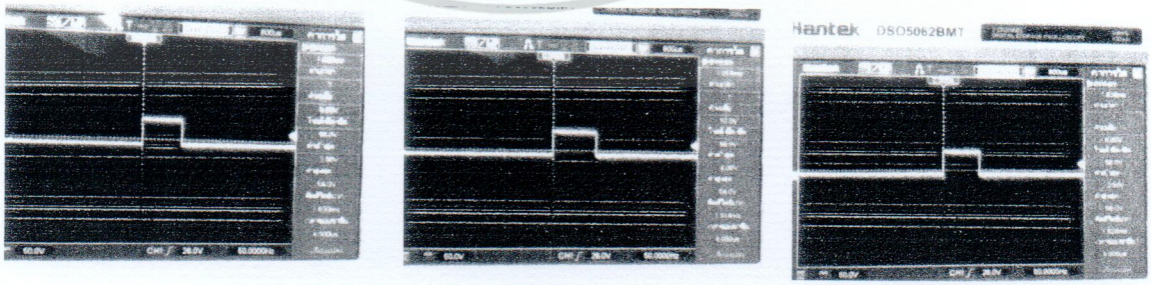
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่2



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
2	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
5	1.47ms	1.45ms	1.48ms	1.47ms	1.46ms	0.68

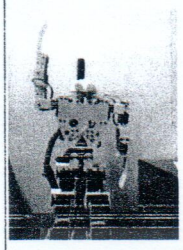
รูปที่ 4.14 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 2



รูปที่ 4.15 กราฟแสดง pulse ของ Servo2

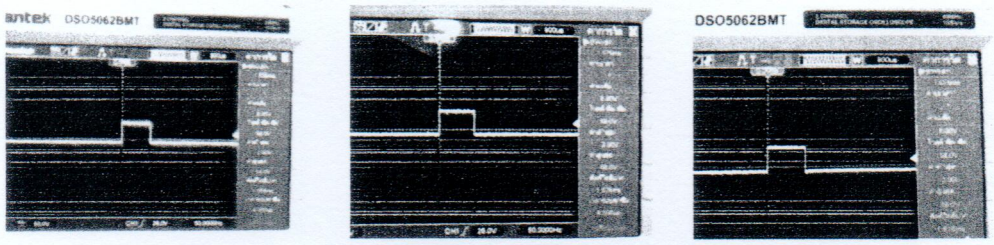
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่3

	หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
			ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
	1	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
	2	752us	752us	752us	752us	752us	0
	5	1.47ms	1.45ms	1.48ms	1.47ms	1.46ms	0.68

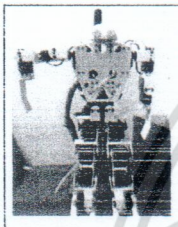
รูปที่ 4.16 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



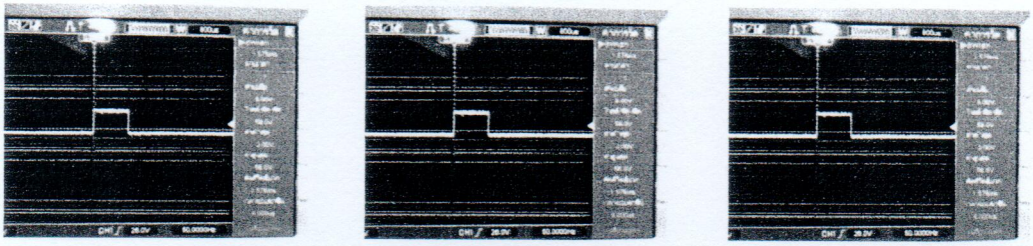
รูปที่ 4.17 กราฟแสดง pulse ของ Servo3

ตารางที่ 4.9 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่4

	หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
			ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
	1	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
	2	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
	5	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0

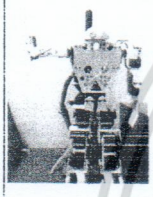
รูปที่ 4.18 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



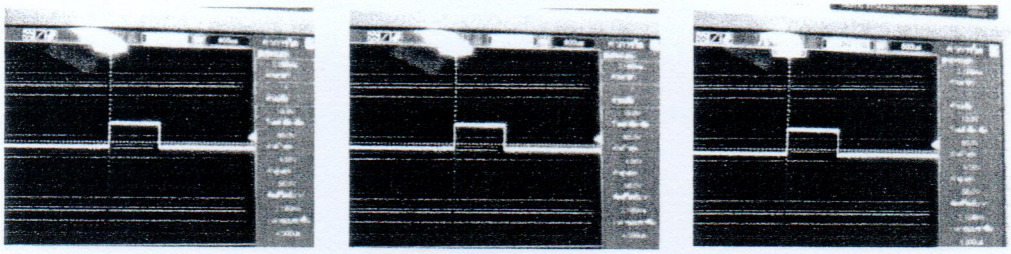
รูปที่ 4.19 กราฟแสดง pulse ของ Servo1

ตารางที่ 4.10 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่5

	หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
			ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
	1	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
	2	752us	752us	752us	752us	752us	0
	5	2.4ms	2.41ms	2.43ms	2.4ms	2.41ms	0.41

รูปที่ 4.20 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

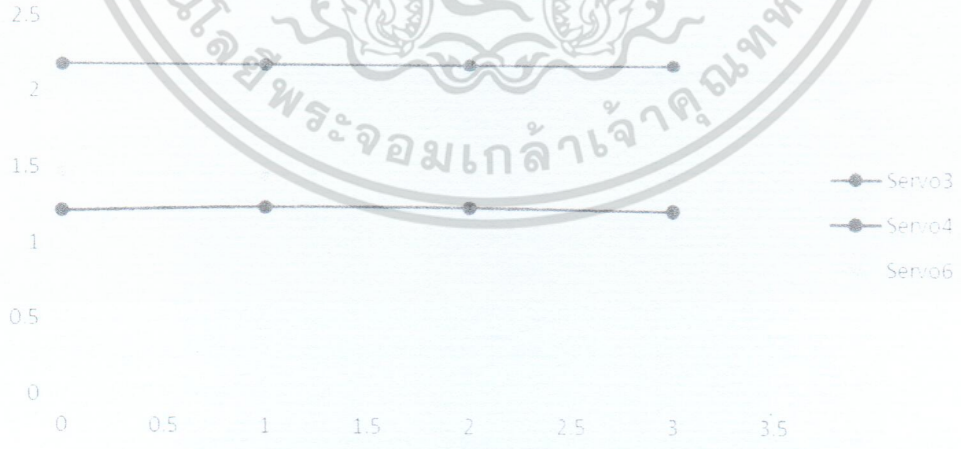


รูปที่ 4.21 กราฟแสดง pulse ของ Servo5

4.3 ยกแขนซ้ายข้างเดียว

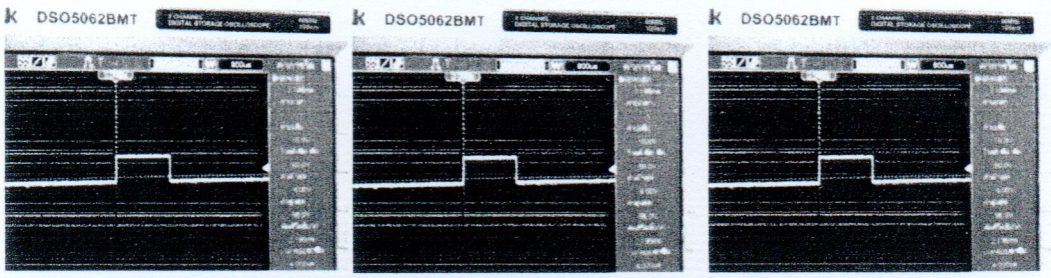
ตารางที่ 4.11 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่1

หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	1.21ms	1.24ms	1.24ms	1.22ms	1.23ms	1.65
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36



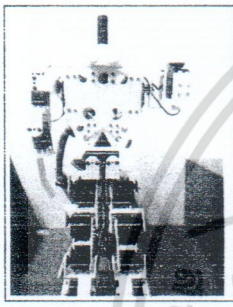
รูปที่ 4.22 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



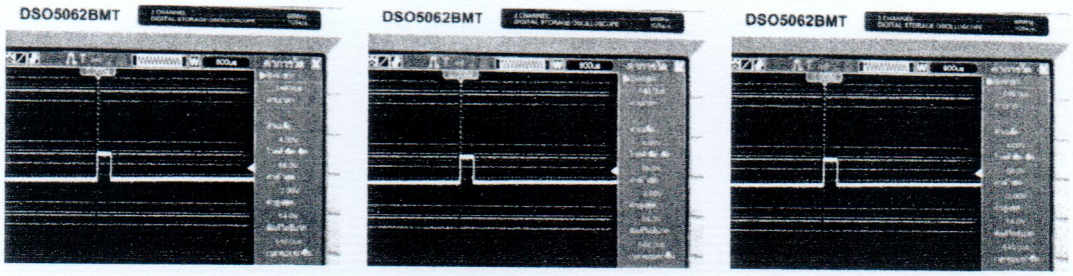
รูปที่ 4.23 กราฟแสดง pulse ของ Servo3

ารางที่ 4.12 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 2

	หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
			ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
	3	1.32ms	1.31ms	1.32ms	1.30ms	1.23ms	0.75
	4	548us	548us	548us	548us	548us	0
	6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36

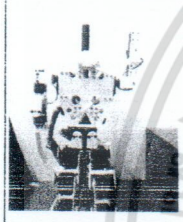
รูปที่ 4.24 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 กราฟแสดง pulse ของ Servo4

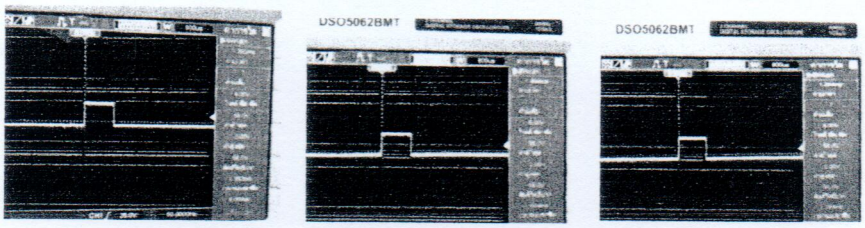
ตารางที่ 4.13 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 3



หมายเลข Servo	ค่าที่กำหนด	ทดสอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	548us	548us	548us	548us	548ms	0
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36

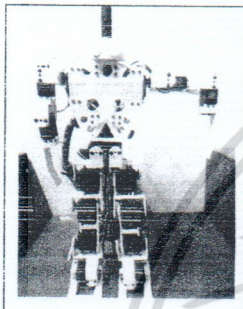
รูปที่ 4.26 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 กราฟแสดง pulse ของ Servo6

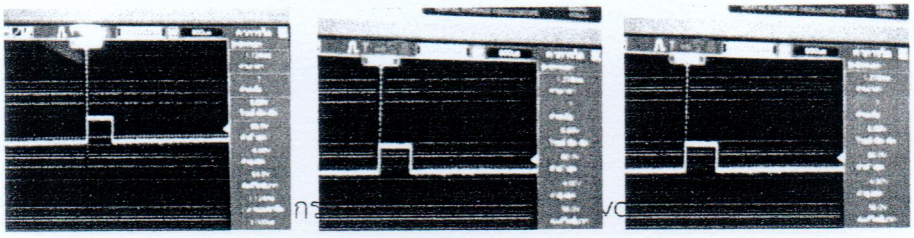
ตารางที่ 4.14 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4



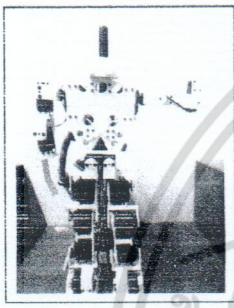
หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
3	544us	548ms	548ms	548ms	548us	0.73
4	1.21ms	1.22ms	1.21ms	1.21ms	1.21ms	0
6	752us	752us	752us	752us	752us	0

รูปที่ 4.28 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



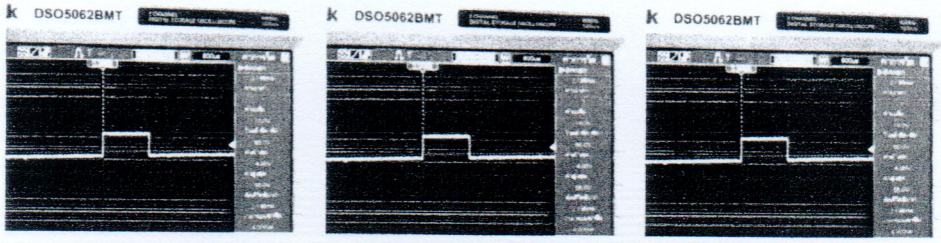
ตารางที่ 4.15 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 5



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	1.21ms	1.21ms	1.22ms	1.20ms	1.21ms	0
6	752us	752us	752us	752us	752us	0

รูปที่ 4.30 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

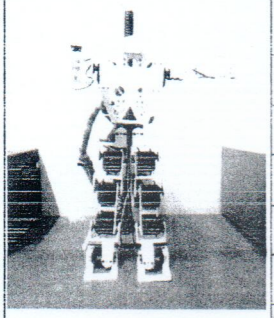


รูปที่ 4.31 กราฟแสดง pulse ของ Servo3

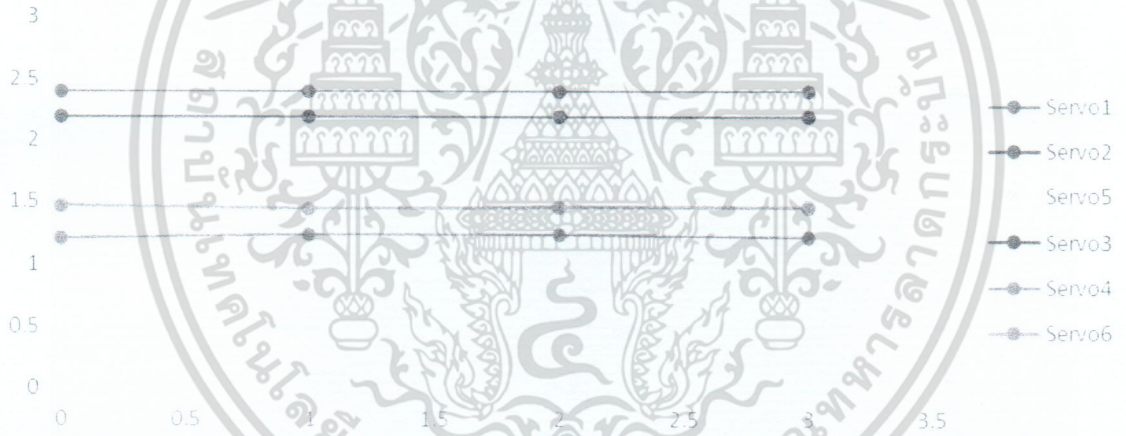
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ยกแขนสองข้างโดยแขนทั้งสองข้างทำคนละท่ากัน

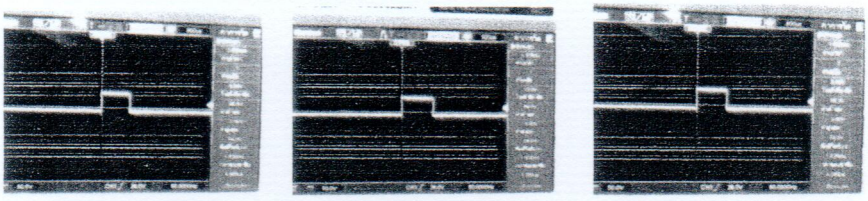
ตารางที่ 4.16 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
2	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
5	1.47ms	1.45ms	1.48ms	1.47ms	1.46ms	0.68
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	1.21ms	1.24ms	1.24ms	1.22ms	1.23ms	1.65
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36



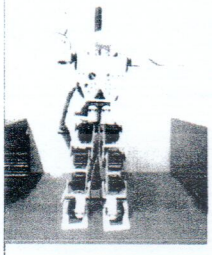
รูปที่ 4.32 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1



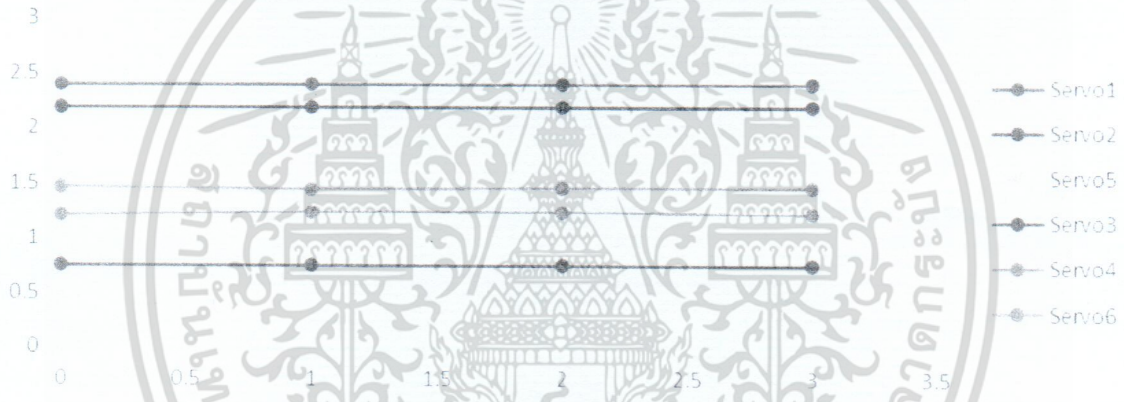
รูปที่ 4.33 กราฟแสดง pulse ของ Servo2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

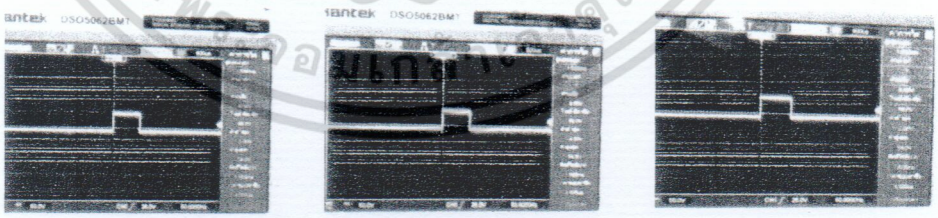
ตารางที่ 4.17 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 2



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
2	752us	752us	752us	752us	752us	0
5	1.47ms	1.45ms	1.48ms	1.47ms	1.46ms	0.68
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	1.21ms	1.24ms	1.24ms	1.22ms	1.23ms	1.65
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36



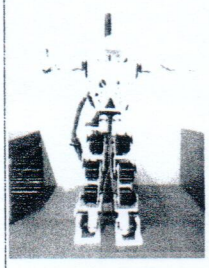
รูปที่ 4.34 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 2



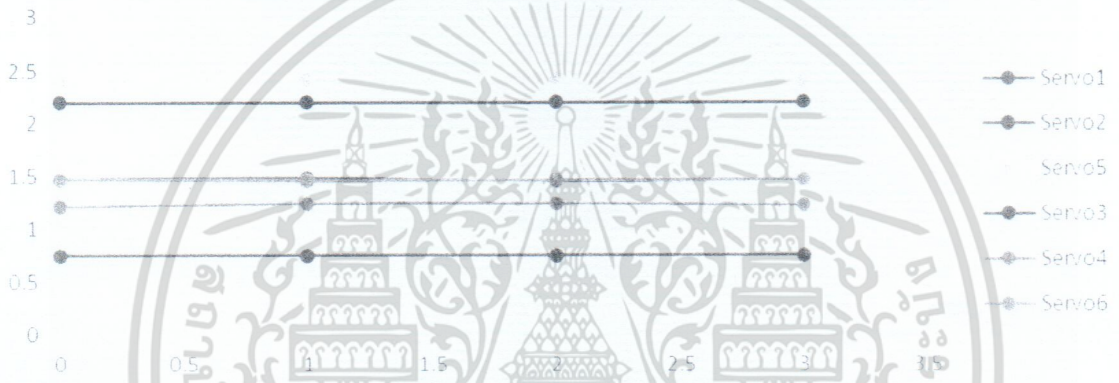
รูปที่ 4.35 กราฟแสดง pulse ของ Servo5

ตารางที่ 4.18 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 3

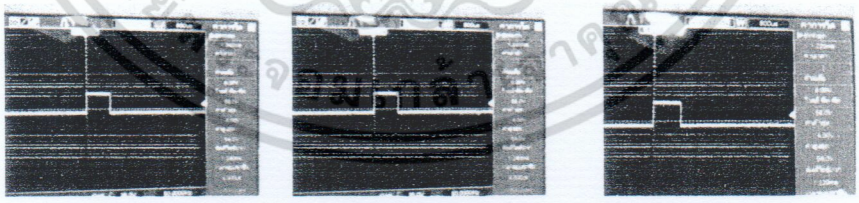
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	1.47ms	1.48ms	1.45ms	1.46ms	1.46ms	0.68
2	752us	752us	752us	752us	752us	0
5	2.4ms	2.41ms	2.43ms	2.4ms	2.41ms	0.41
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	1.21ms	1.24ms	1.24ms	1.22ms	1.23ms	1.65
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36



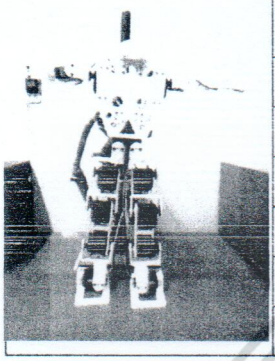
รูปที่ 4.36 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 3



รูปที่ 4.37 กราฟแสดง pulse ของ Servo4

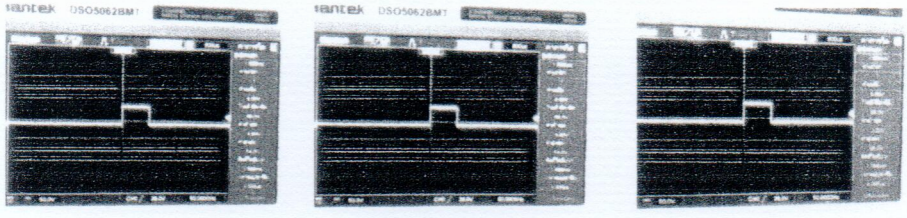
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.19 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
2	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
5	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	1.21ms	1.24ms	1.24ms	1.22ms	1.23ms	1.65
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36

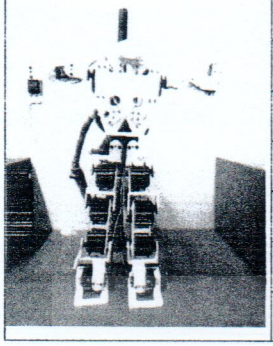
รูปที่ 4.38 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4



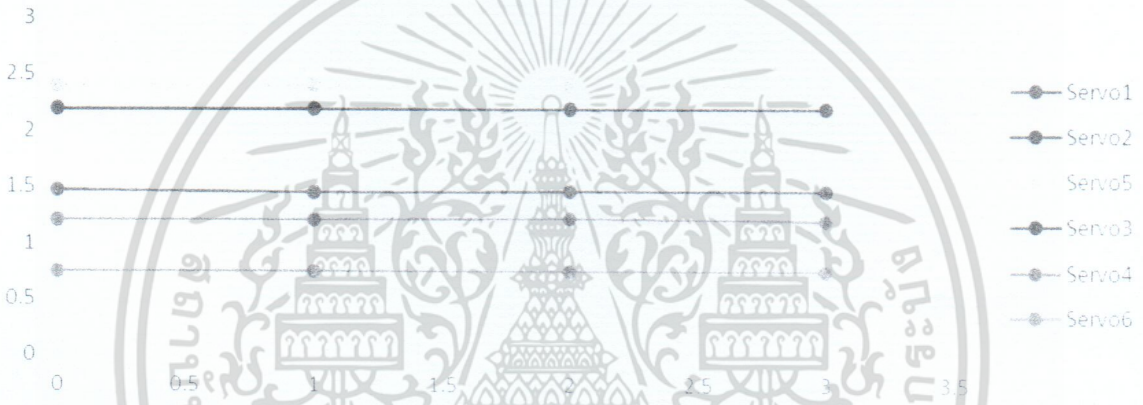
รูปที่ 4.39 กราฟแสดง pulse ของ Servo1

ตารางที่ 4.20 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 5

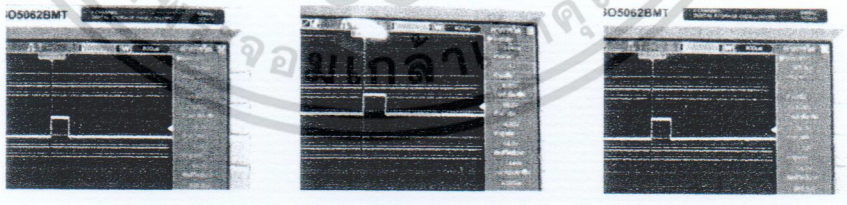
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
2	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
5	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	1.21ms	1.21ms	1.22ms	1.20ms	1.21ms	0
6	752us	752us	752us	752us	752us	0



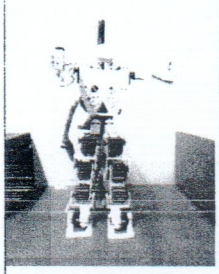
รูปที่ 4.40 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 5



รูปที่ 4.41 กราฟแสดง pulse ของ Servo6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

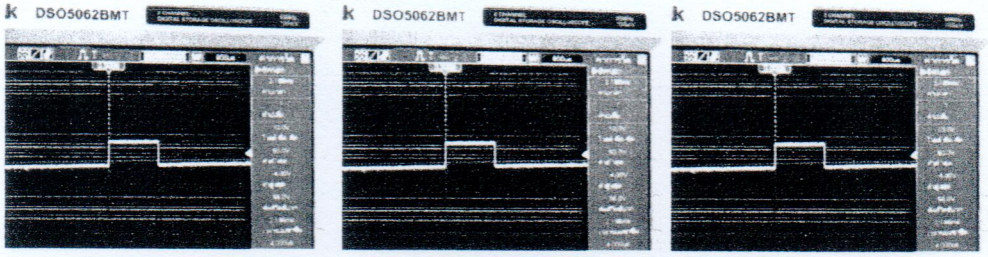
ตารางที่ 4.21 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 6



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
2	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
5	1.47ms	1.45ms	1.48ms	1.47ms	1.46ms	0.68
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	1.21ms	1.21ms	1.22ms	1.20ms	1.21ms	0
6	752us	752us	752us	752us	752us	0

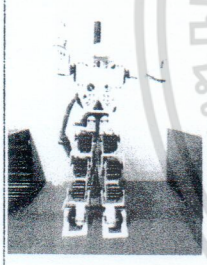
รูปที่ 4.42 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



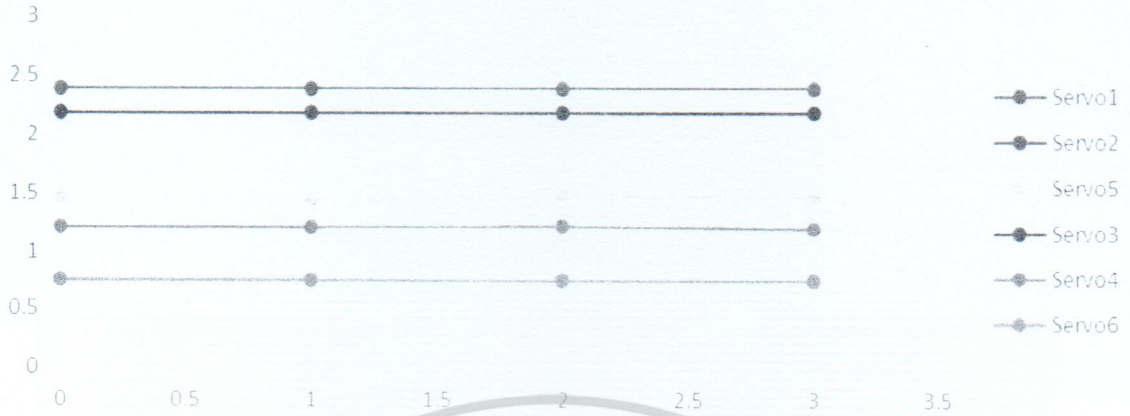
รูปที่ 4.43 การแสดง pulse ของ Servo3

ตารางที่ 4.22 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 7

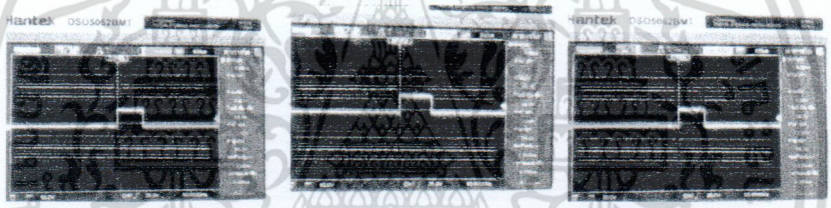


หมายเลข Servo	ค่าที่กำหนด	ทดลอง ครั้งที่ 1	ทดลอง ครั้งที่ 2	ทดลอง ครั้งที่ 3	ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
1	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
2	752us	752us	752us	752us	752us	0
3	1.47ms	1.45ms	1.48ms	1.47ms	1.46ms	0.68
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	1.21ms	1.21ms	1.22ms	1.20ms	1.21ms	0
6	752us	752us	752us	752us	752us	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

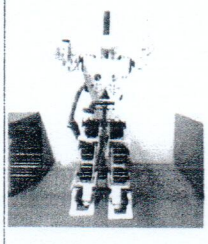


รูปที่ 4.44 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 7



รูปที่ 4.45 กราฟแสดง pulse ของ Servo2

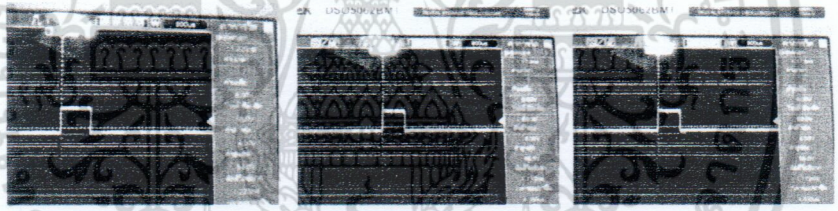
ตารางที่ 4.23 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 8



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
2	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
5	1.47ms	1.45ms	1.48ms	1.47ms	1.46ms	0.68
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	548us	548us	548us	548us	548ms	0
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

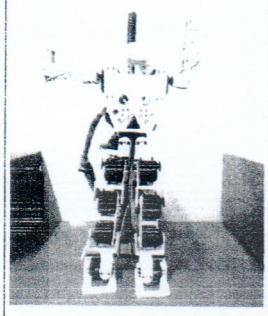
รูปที่ 4.46 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 8



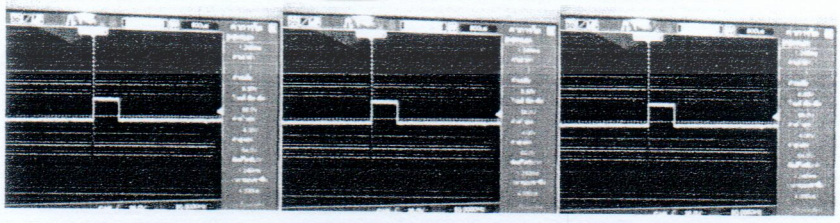
รูปที่ 4.47 กราฟแสดง pulse ของ Servo4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.24 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 9

	หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
			ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
	1	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
	2	752us	752us	752us	752us	752us	0
	5	2.4ms	2.41ms	2.43ms	2.4ms	2.41ms	0.41
	3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
	4	548us	548us	548us	548us	548ms	0
	6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36

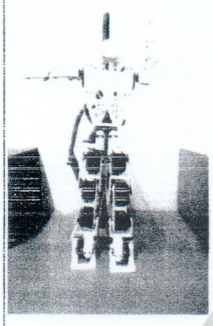
รูปที่ 4.48 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 9



รูปที่ 4.49 กราฟแสดง pulse ของ Servo1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

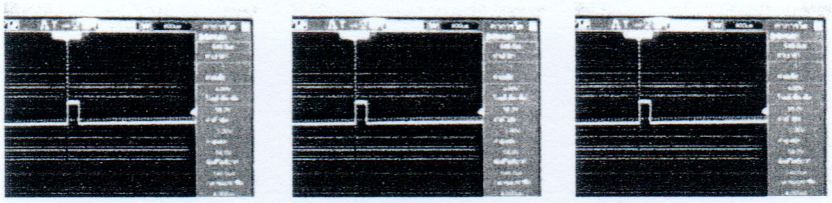
ตารางที่ 4.25 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 10



หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	1.47ms	1.45ms	1.46ms	1.45ms	1.45ms	1.36
2	752us	752us	752us	752us	752us	0
5	1.47ms	1.15ms	1.64ms	1.57ms	1.45ms	1.36
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	548us	548us	548us	548us	548ms	0
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36

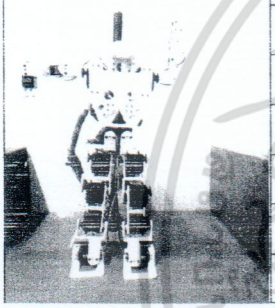
รูปที่ 4.50 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

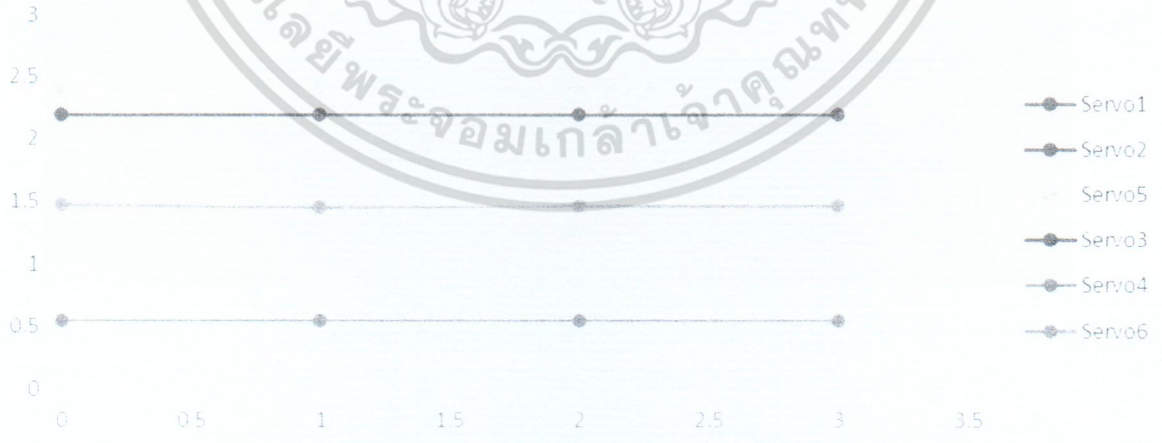


รูปที่ 4.51 กราฟแสดง pulse ของ Servo4

ตารางที่ 4.26 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 11

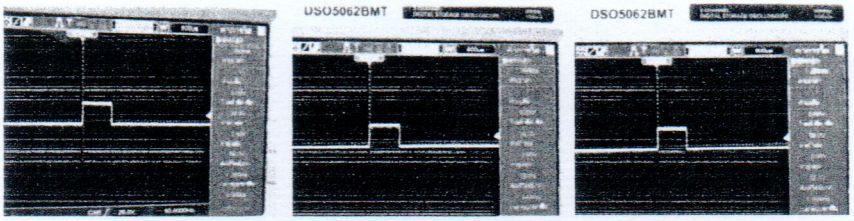


หมายเลข Servo	ค่าที่กำหนด	ทดสอง ครั้งที่ 1	ทดสอง ครั้งที่ 2	ทดสอง ครั้งที่ 3	ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
1	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
2	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
5	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
3	2.19ms	2.19ms	2.19ms	2.19ms	2.19ms	0
4	548us	548us	548us	548us	548ms	0
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36



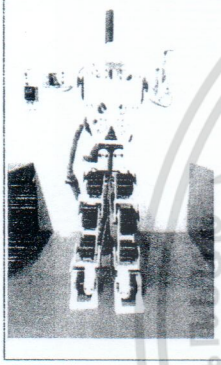
รูปที่ 4.52 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.53 กราฟแสดง pulse ของ Servo6

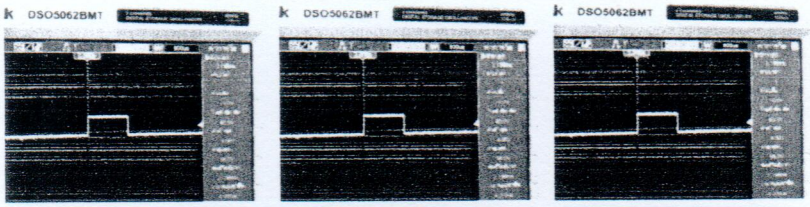
ตารางที่ 4.27 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 12



หมายเลข Servo	ค่าที่กำหนด	ทดลอง ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
1	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
2	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
5	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
3	1.32ms	1.31ms	1.32ms	1.30ms	1.23ms	0.75
4	548us	548us	548us	548us	548us	0
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36

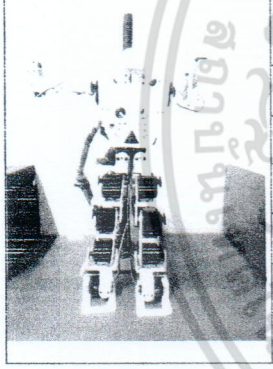
รูปที่ 4.54 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.55 กราฟแสดง pulse ของ Servo3

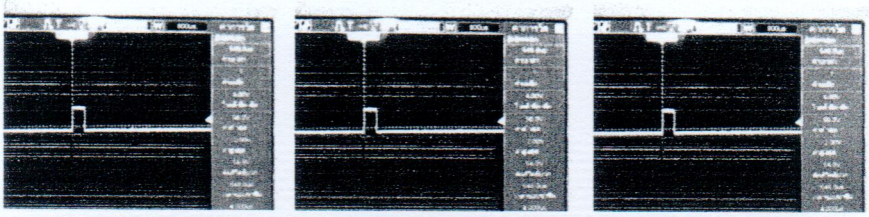
ตารางที่ 4.28 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 13



หมายเลข Servo	ค่าที่กำหนด	ทดลอง ครั้งที่ 1	ทดลอง ครั้งที่ 2	ทดลอง ครั้งที่ 3	ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
1	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36
2	752us	752us	752us	752us	752us	0
5	2.4ms	2.41ms	2.43ms	2.4ms	2.41ms	0.41
3	1.32ms	1.31ms	1.32ms	1.30ms	1.23ms	0.75
4	548us	548us	548us	548us	548us	0
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36

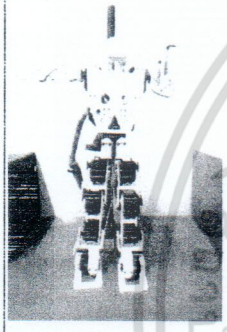
รูปที่ 4.56 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

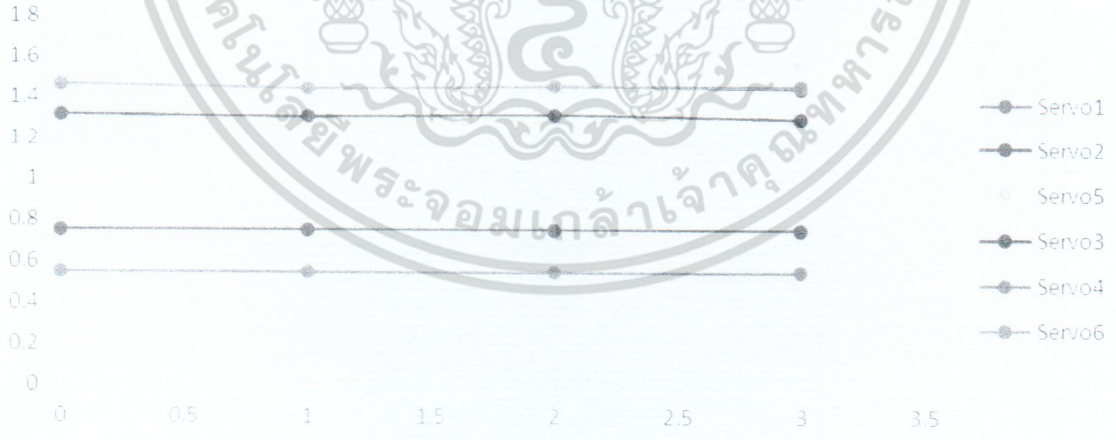


รูปที่ 4.57 กราฟแสดง pulse ของ Servo4

ตารางที่ 4.29 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 14

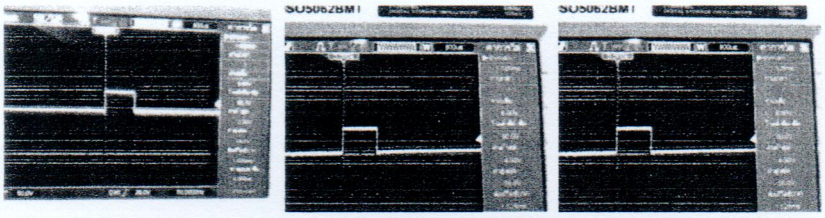


หมายเลข Servo	ค่าที่กำหนด	ทดลอง			ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
		ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3		
1	1.47ms	1.45ms	1.46ms	1.45ms	1.45ms	1.36
2	752us	752us	752us	752us	752us	0
5	1.47ms	1.15ms	1.64ms	1.57ms	1.45ms	1.36
3	1.32ms	1.31ms	1.32ms	1.30ms	1.23ms	0.75
4	548us	548us	548us	548us	548us	0
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36



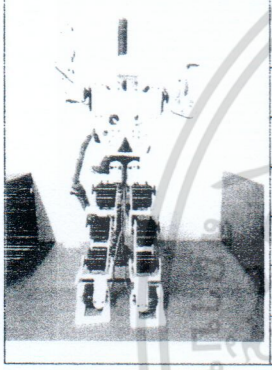
รูปที่ 4.58 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

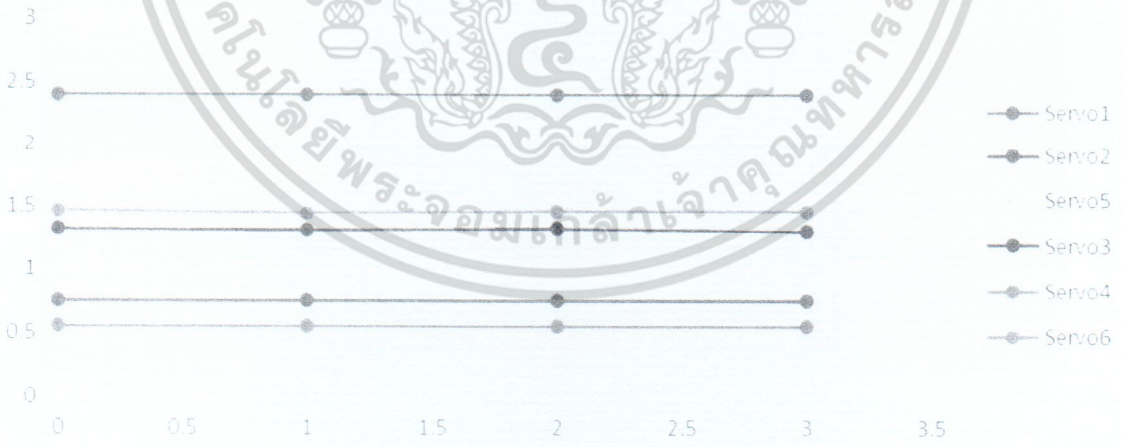


รูปที่ 4.59 กราฟแสดง pulse ของ Servo1

ตารางที่ 4.30 แสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 15

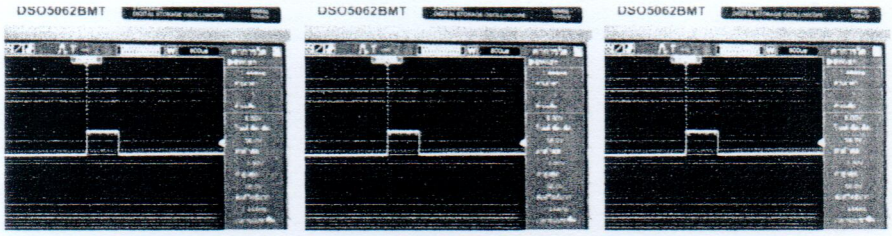


หมายเลข Servo	ค่าที่ กำหนด	ทดลอง ครั้งที่ 1	ทดลอง ครั้งที่ 2	ทดลอง ครั้งที่ 3	ค่าเฉลี่ย	ความคลาดเคลื่อน(%)
1	2.4ms	2.4ms	2.4ms	2.4ms	2.4ms	0
2	752us	752us	752us	752us	752us	0
3	1.47ms	1.45ms	1.48ms	1.47ms	1.46ms	0.68
3	1.32ms	1.31ms	1.32ms	1.30ms	1.23ms	0.75
4	548us	548us	548us	548us	548us	0
6	1.47ms	1.45ms	1.46ms	1.46ms	1.45ms	1.36



รูปที่ 4.60 กราฟแสดงความกว้าง pulse ของ servo ที่ใช้ในการเคลื่อนไหวท่าที่ 15

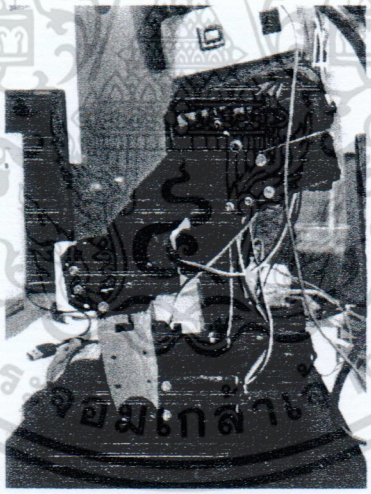
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.61 กราฟแสดง pulse ของ Servo1

ในส่วนของผลการทดสอบโปรแกรมและการเคลื่อนไหวของหุ่นยนต์ ด้วยการวัดค่าความกว้างของพัลซ์(PWM = Pulse Width Modulation) ของ Servo ที่เราเลือกใช้งานในแต่ละตัว โดยแบ่งออกเป็น 8 ท่า ดังต่อไปนี้

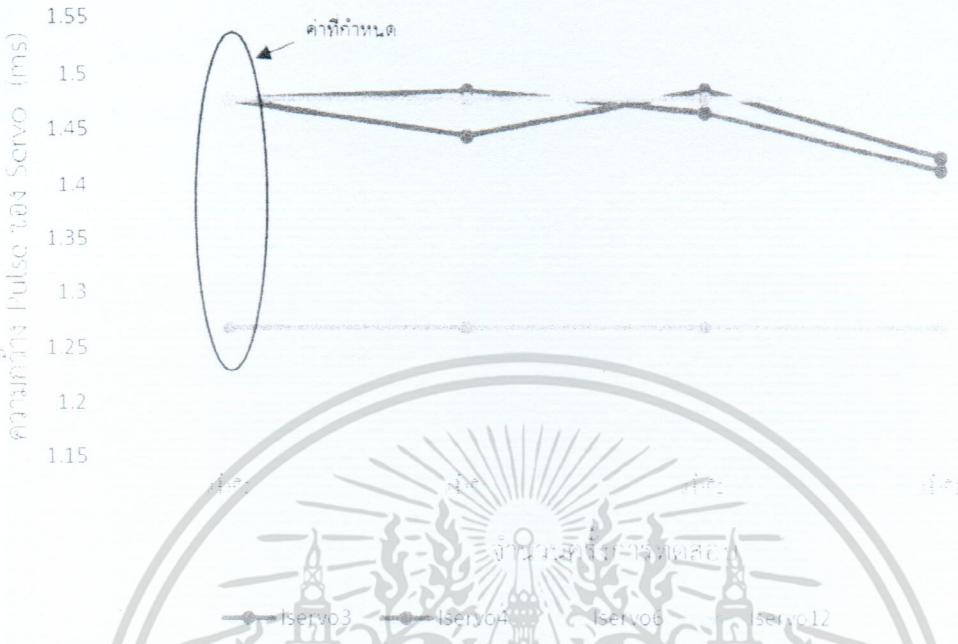
4.1 ผลการทดสอบท่าที่ 1 ท่ายกขาซ้ายขึ้น



รูปที่ 4.1 รูปแสดงการเคลื่อนไหวท่าที่ 1

ตารางที่ 4.1 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



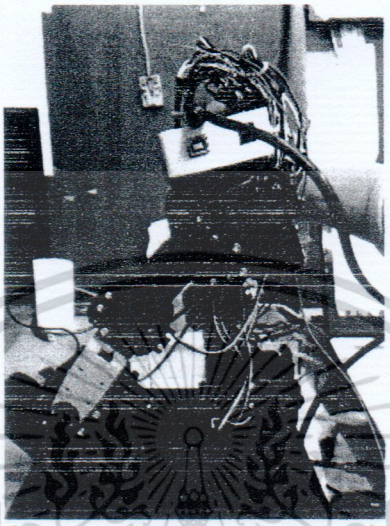
รูปที่ 4.2 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1 จากตารางที่ 4.1

จากรูปที่ 4.2 กราฟของ lservo3 และ lservo4 มีค่าไม่เป็นเส้นตรง เนื่องจากค่าที่ได้มีความคาดเคลื่อน 1.558% และ 1.694% ตามลำดับ แต่จะพบว่า lservo6 และ lservo12 มีค่าเป็นเส้นตรง เนื่องจากเป็นค่าที่กำหนดตายตัว จึงมีค่าเท่ากัน

Servoที่ใช้	ค่าPWMที่กำหนด (ms)	การทดสอบ (ms)			ค่าเฉลี่ย (ms)	ความคาดเคลื่อน (%)
		ครั้งที่1	ครั้งที่2	ครั้งที่3		
lservo3	1.476	1.484	1.464	1.412	1.453	1.558
lservo4	1.476	1.444	1.484	1.424	1.451	1.694
lservo6	1.476	1.476	1.476	1.476	1.476	0
lservo12	1.268	1.268	1.268	1.268	1.268	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดสอบท่าที่ 2 ท่าเตะขาซ้ายไปข้างหน้า

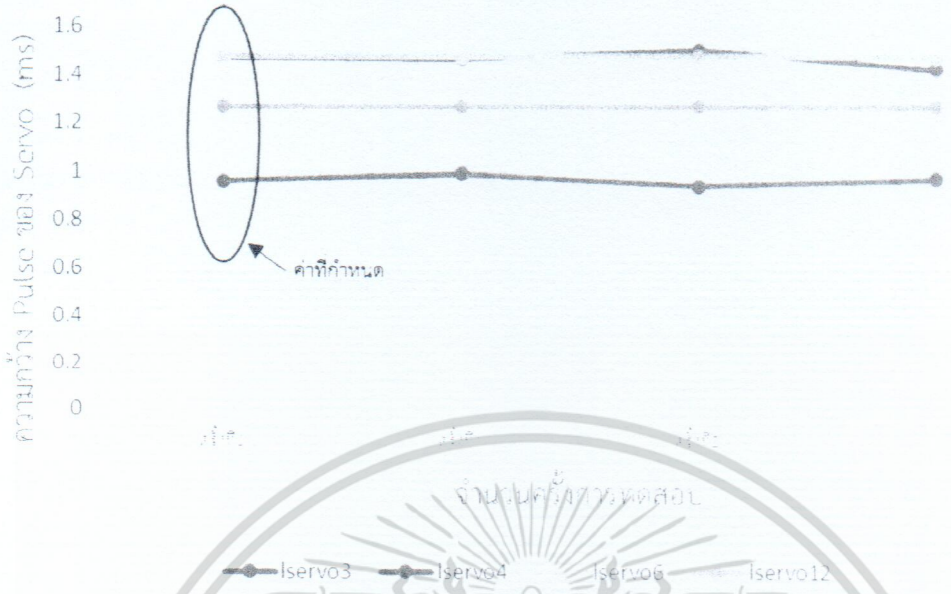


รูปที่ 4.3 รูปแสดงการเคลื่อนไหวท่าที่ 2

ตารางที่ 4.2 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 2

Servoที่ใช้	ค่าPWMที่กำหนด (ms)	การทดสอบ (ms)			ค่าเฉลี่ย (ms)	ความคาดเคลื่อน (%)
		ครั้งที่1	ครั้งที่2	ครั้งที่3		
lservo3	1.476	1.464	1.504	1.424	1.464	0.813
lservo4	0.960	0.990	0.940	0.968	0.966	0.625
lservo6	1.476	1.476	1.476	1.476	1.476	0
lservo12	1.268	1.268	1.268	1.268	1.268	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

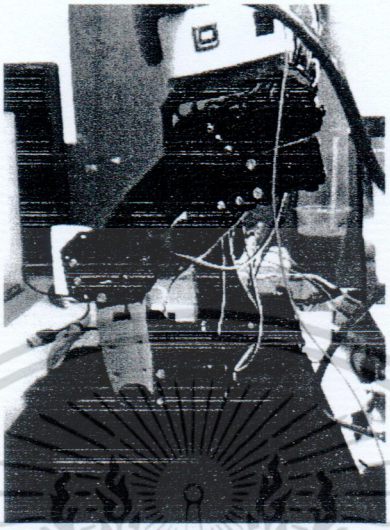


รูปที่ 4.4 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 2 จากตารางที่ 4.2

จากรูปที่ 4.4 กราฟของ lservo3 และ lservo4 มีค่าไม่เป็นเส้นตรง เนื่องจากค่าที่ได้มีความคาดเคลื่อน 0.813% และ 0.625% ตามลำดับ แต่จะพบว่า lservo6 และ lservo12 มีค่าเป็นเส้นตรง เนื่องจากเป็นค่าที่กำหนดตายตัว จึงมีค่าเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบท่าที่ 3 ท่าขาซ้ายลง

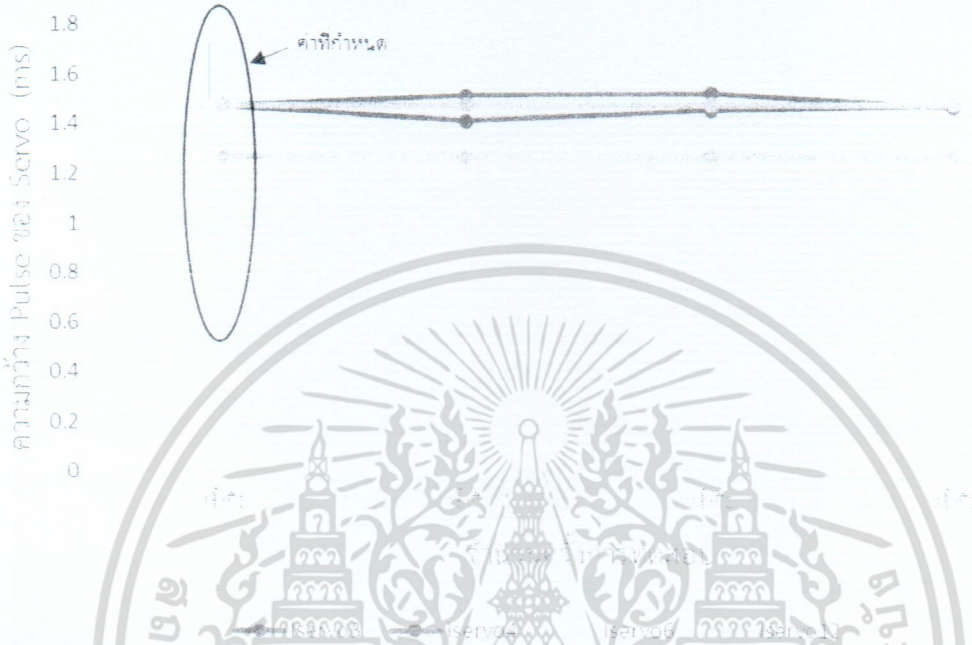


รูปที่ 4.5 รูปแสดงการเคลื่อนไหวท่าที่ 3

ตารางที่ 4.3 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 3

Servoที่ใช้	ค่าPWMที่กำหนด (ms)	การทดสอบ (ms)			ค่าเฉลี่ย (ms)	ความคาดเคลื่อน (%)
		ครั้งที่1	ครั้งที่2	ครั้งที่3		
lservo3	1.476	1.516	1.524	1.464	1.501	1.694
lservo4	1.476	1.412	1.452	1.475	1.446	2.033
lservo6	1.476	1.476	1.476	1.476	1.476	0
lservo12	1.268	1.268	1.268	1.268	1.268	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

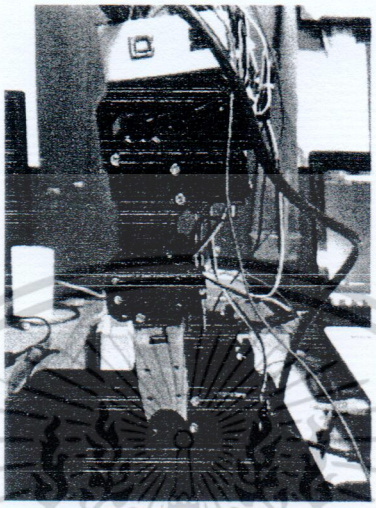


รูปที่ 4.6 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 3 จากตารางที่ 4.3

จากรูปที่ 4.6 กราฟของ lservo3 และ lservo4 มีค่าไม่เป็นเส้นตรง เนื่องจากค่าที่ได้มีความคาดเคลื่อน 1.694% และ 2.033% ตามลำดับ แต่จะพบว่า lservo6 และ lservo12 มีค่าเป็นเส้นตรง เนื่องจากเป็นค่าที่กำหนดตายตัว จึงมีค่าเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดสอบท่าที่ 4 ท่าขาซ้ายอยู่ที่พื้น

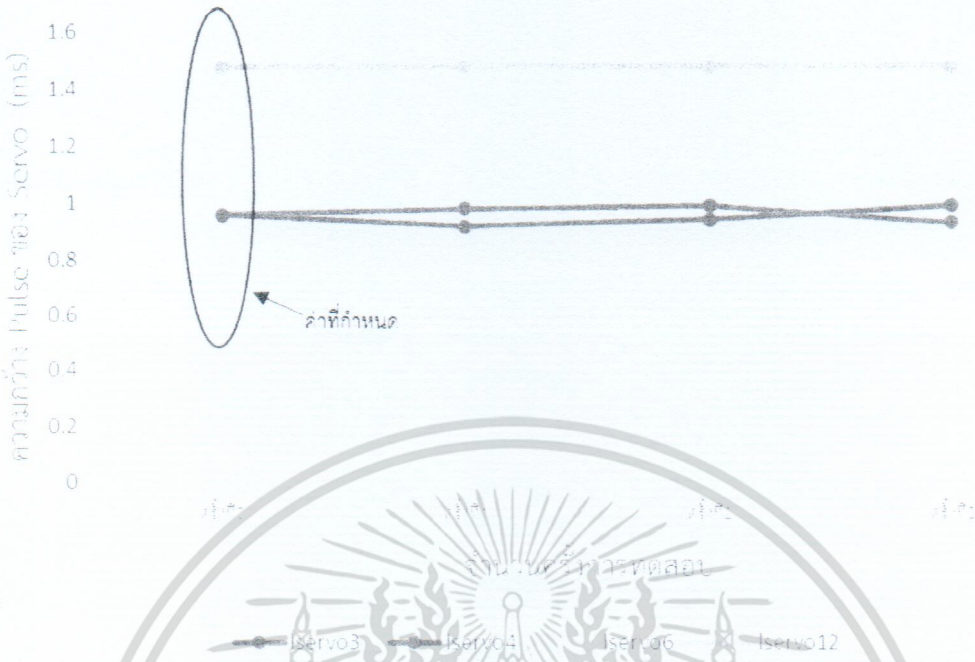


รูปที่ 4.7 รูปแสดงการเคลื่อนไหวท่าที่ 4

ตารางที่ 4.4 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4

Servoที่ใช้	ค่าPWMที่กำหนด (ms)	การทดสอบ (ms)			ค่าเฉลี่ย (ms)	ความคาดเคลื่อน (%)
		ครั้งที่1	ครั้งที่2	ครั้งที่3		
lservo3	0.960	0.980	0.990	0.928	0.966	0.625
lservo4	0.960	0.918	0.940	0.990	0.949	1.146
lservo6	1.476	1.476	1.476	1.476	1.476	0
lservo12	1.476	1.476	1.476	1.476	1.476	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

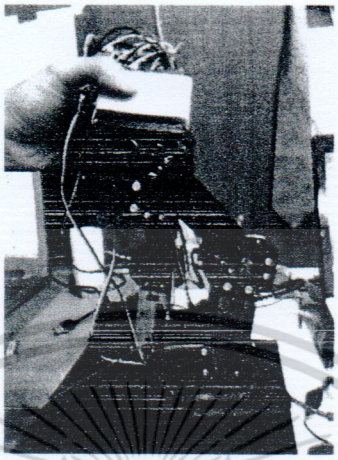


รูปที่ 4.8 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4 จากตารางที่ 4.4

จากรูปที่ 4.8 กราฟของ lservo3 และ lservo4 มีค่าไม่เป็นเส้นตรง เนื่องจากค่าที่ได้มีความคาดเคลื่อน 0.625% และ 1.146% ตามลำดับ แต่จะพบว่า lservo6 และ lservo12 มีค่าเป็นเส้นตรง เนื่องจากเป็นค่าที่กำหนดตายตัว จึงมีค่าเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ผลการทดสอบท่าที่ 5 ท่ายกขาขวาขึ้น

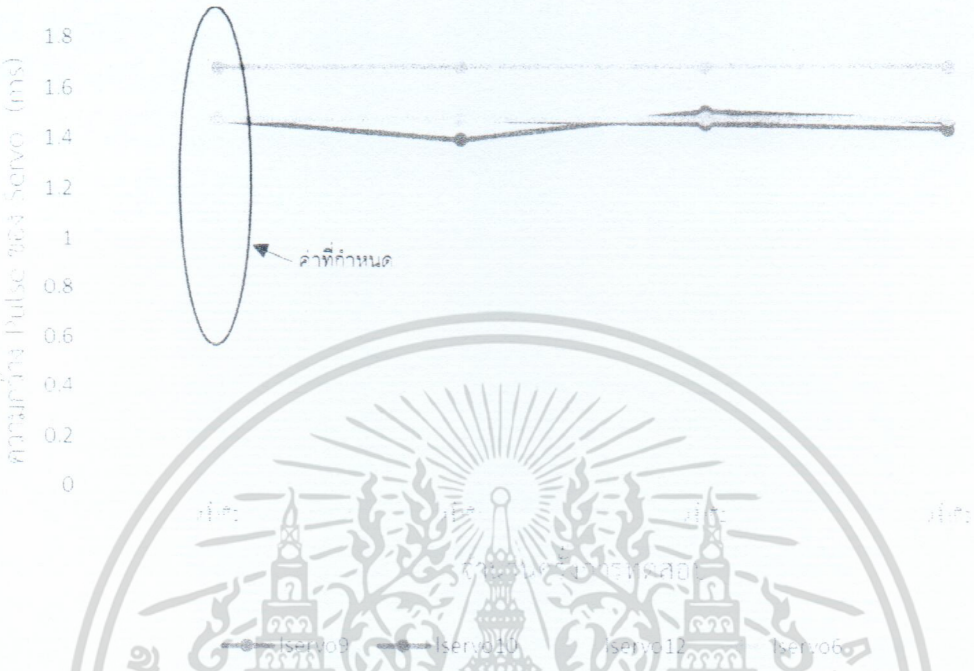


รูปที่ 4.9 รูปแสดงการเคลื่อนไหวท่าที่ 5

ตารางที่ 4.5 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 5

Servoที่ใช้	ค่าPWMที่กำหนด (ms)	การทดสอบ (ms)			ค่าเฉลี่ย (ms)	ความคาดเคลื่อน (%)
		ครั้งที่1	ครั้งที่2	ครั้งที่3		
lservo9	1.476	1.476	1.452	1.432	1.453	1.558
lservo10	1.476	1.392	1.504	1.464	1.453	1.558
lservo12	1.476	1.476	1.476	1.476	1.476	0
lservo6	1.682	1.682	1.682	1.682	1.682	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

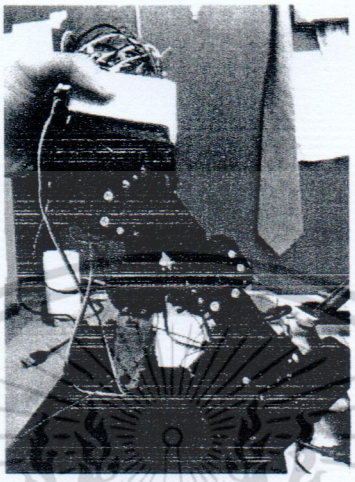


รูปที่ 4.10 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 5 จากตารางที่ 4.5

จากรูปที่ 4.10 กราฟของ lservo9 และ lservo10 มีค่าไม่เป็นเส้นตรง เนื่องจากค่าที่ได้มีความคาดเคลื่อน 1.558% และ 1.558% ตามลำดับ แต่จะพบว่า lservo6 และ lservo12 มีค่าเป็นเส้นตรง เนื่องจากเป็นค่าที่กำหนดตายตัว จึงมีค่าเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลการทดสอบท่าที่ 6 ท่าเตะขาขวาไปข้างหน้า

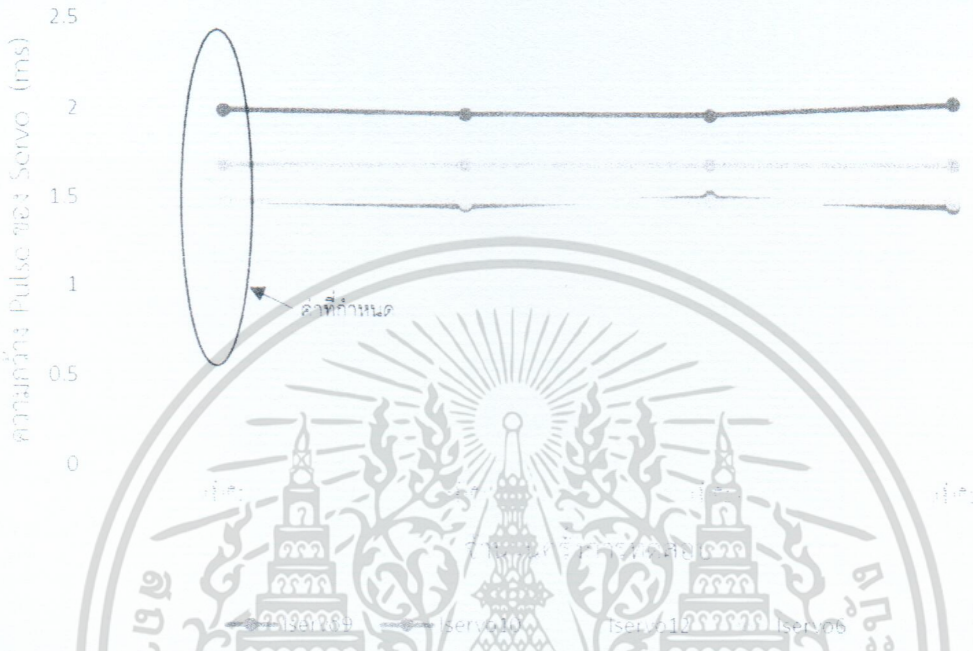


รูปที่ 4.11 รูปแสดงการเคลื่อนไหวท่าที่ 6

ตารางที่ 4.6 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 6

Servoที่ใช้	ค่าPWMที่กำหนด (ms)	การทดสอบ (ms)			ค่าเฉลี่ย (ms)	ความคาดเคลื่อน (%)
		ครั้งที่1	ครั้งที่2	ครั้งที่3		
lservo9	1.476	1.452	1.502	1.444	1.466	0.677
lservo10	1.990	1.968	1.960	2.020	1.982	0.402
lservo12	1.476	1.476	1.476	1.476	1.476	0
lservo6	1.682	1.682	1.682	1.682	1.682	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

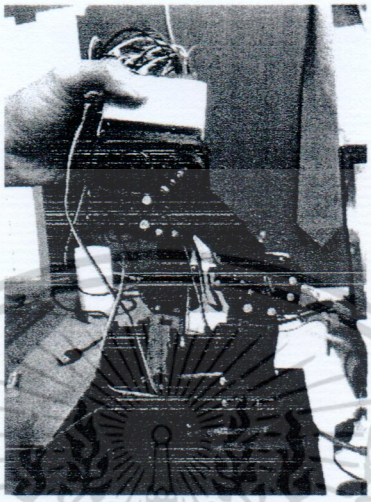


รูปที่ 4.12 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 6 จากตารางที่ 4.6

จากรูปที่ 4.10 กราฟของ lservo9 และ lservo10 มีค่าไม่เป็นเส้นตรง เนื่องจากค่าที่ได้มีความคาดเคลื่อน 0.677% และ 0.402% ตามลำดับ แต่จะพบว่า lservo6 และ lservo12 มีค่าเป็นเส้นตรง เนื่องจากเป็นค่าที่กำหนดตายตัว จึงมีค่าเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ผลการทดสอบท่าที่ 7 ท่าขาขวาลง

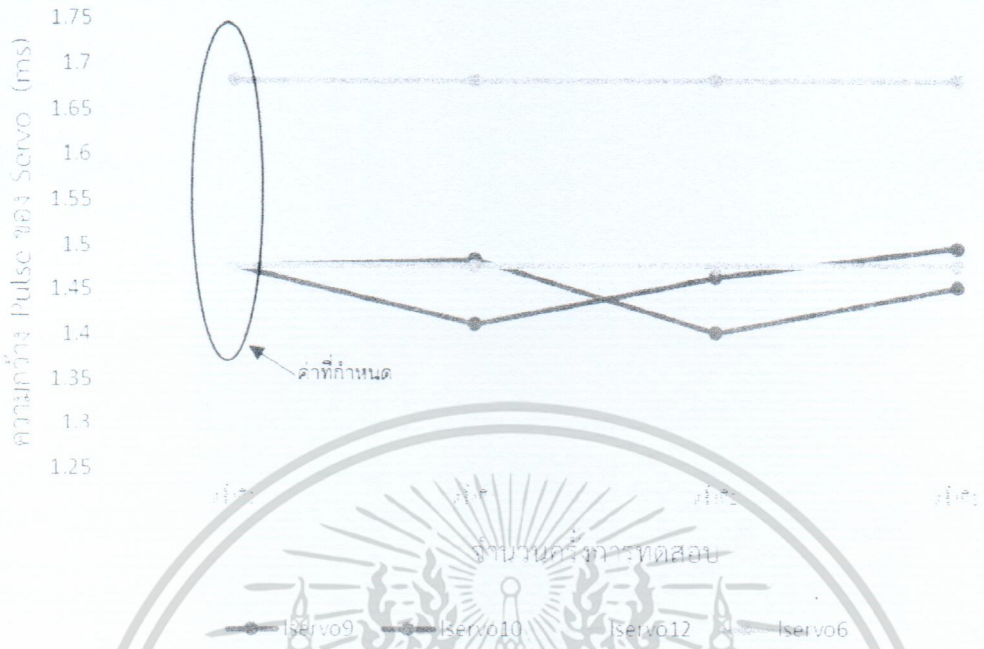


รูปที่ 4.13 รูปแสดงการเคลื่อนไหวท่าที่ 7

ตารางที่ 4.7 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 7

Servoที่ใช้	ค่าPWMที่กำหนด (ms)	การทดสอบ (ms)			ค่าเฉลี่ย (ms)	ความคาดเคลื่อน (%)
		ครั้งที่1	ครั้งที่2	ครั้งที่3		
lservo9	1.476	1.484	1.402	1.452	1.446	2.032
lservo10	1.476	1.412	1.464	1.496	1.457	1.287
lservo12	1.476	1.476	1.476	1.476	1.476	0
lservo6	1.682	1.682	1.682	1.682	1.682	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

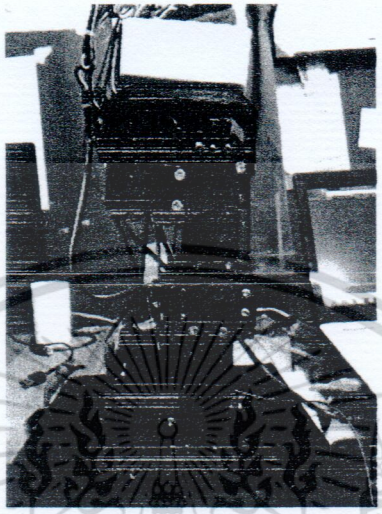


รูปที่ 4.14 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 7 จากตารางที่ 4.7

จากรูปที่ 4.10 กราฟของ lservo9 และ lservo10 มีค่าไม่เป็นเส้นตรง เนื่องจากค่าที่ได้มีความคาดเคลื่อน 2.032% และ 1.287% ตามลำดับ แต่จะพบว่า lservo6 และ lservo12 มีค่าเป็นเส้นตรง เนื่องจากเป็นค่าที่กำหนดตายตัว จึงมีค่าเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 ผลการทดสอบท่าที่ 8 ท่าขาขวาอยู่ที่พื้น

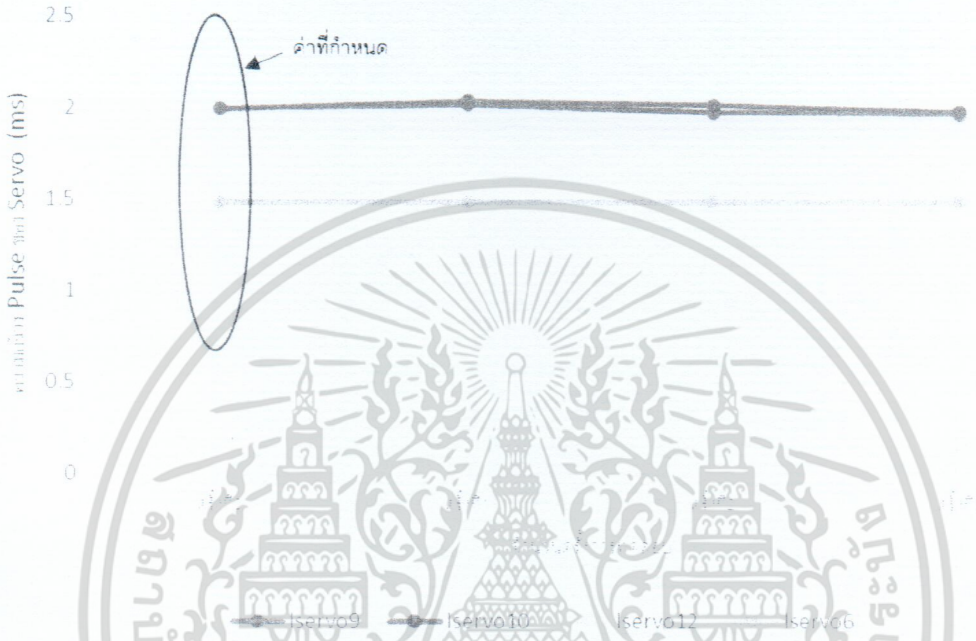


รูปที่ 4.15 รูปแสดงการเคลื่อนไหวท่าที่ 8

ตารางที่ 4.8 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 8

Servoที่ใช้	ค่าPWMที่กำหนด (ms)	การทดสอบ (ms)			ค่าเฉลี่ย (ms)	ความคาดเคลื่อน (%)
		ครั้งที่1	ครั้งที่2	ครั้งที่3		
lservo9	1.990	2.012	1.968	1.960	1.980	0.502
lservo10	1.990	2.032	2.010	1.970	2.004	0.703
lservo12	1.476	1.476	1.476	1.476	1.476	0
lservo6	1.476	1.476	1.476	1.476	1.476	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 8 จากตารางที่ 4.8

จากรูปที่ 4.10 กราฟของ lservo9 และ lservo10 มีค่าไม่เป็นเส้นตรง เนื่องจากค่าที่ได้มีความคาดเคลื่อน 0.502% และ 0.703% ตามลำดับ แต่จะพบว่า lservo6 และ lservo12 มีค่าเป็นเส้นตรง เนื่องจากเป็นค่าที่กำหนดตายตัว จึงมีค่าเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์

5.1 สรุปผลการดำเนินงาน

โครงงานนี้ออกแบบและจำลองหุ่นยนต์ขนาดใหญ่ เพื่อใช้ในการเก็บกู้วัตถุระเบิด โดยอาศัยการควบคุมท่าทางของหุ่นยนต์ด้วยการควบคุมด้วยกล้องตรวจจับแบบ 3 มิติ โดย ขนาดของหุ่นยนต์ในการสร้างจริงจะมีขนาดใกล้เคียงกับมนุษย์ 150 เซนติเมตร สามารถเลียนแบบท่าทางของผู้ควบคุมได้ แต่ยังมีความเชื่องช้าอยู่อันเนื่องมาจาก กำลังของมอเตอร์ไม่เพียงพอ

5.2 ปัญหาที่พบ

จากการทำโครงการนี้พบปัญหาที่เกิดจากเครื่อง Printer 3D เนื่องจากเป็นเครื่องมือใหม่สำหรับผู้จัดทำจึงทำให้ความไม่มีชำนาญในการใช้เครื่องมือชนิดนี้ ปัญหาที่เกิดขึ้นอาทิเช่น การปริ้นชิ้นงานที่มีความซับซ้อนจะต้องใช้เวลาในการปริ้นเป็นเวลานานแล้วบางชิ้นจะต้องนำกลับไปปริ้นใหม่เนื่องจากชิ้นงานมีการผิดเพี้ยนไปจากเดิม มีรอยแตกทำให้ความแข็งแรงของชิ้นงานลดลงไป ส่วนของแบบที่ทำขึ้นมาบางชิ้นอาจมีความไม่พอดีกับ servo ที่ได้เตรียมไว้จึงทำให้ต้องนำแบบไปแก้ให้เข้ากันได้กับ Servo ในส่วนของโปรแกรมปัญหาที่เราได้พบเกิดจากการติดต่อกันระหว่าง Visual Basic 2010 กับ Arduino เนื่องจากตัวผู้จัดทำไม่เคยเขียน Arduino จึงยังไม่ค่อยเข้าใจในโค้ดต่างๆ ส่วนการบังคับ Servo ปัญหาจะอยู่ที่เราจะใช้ส่วนใดของร่างกายมาควบคุมการหมุนของ Servo ให้มีความคลาดเคลื่อนน้อยที่สุด และระยะห่างระหว่างผู้ทดลองกับ Kinect ที่จะทำให้ได้ค่าที่ดีที่สุด ปัญหาต่างๆเหล่านี้จึงเป็นอุปสรรคการทำโครงการให้เกิดความล้าช้า ไม่ได้ผลตรงตามที่ได้ตั้งเป้าหมายไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Kinect SDK. Retrieved. 6 January,2015 from http://www.microsoft.com/en-us/kinectforwindows/

[2] How to installKinect SDK. Retrieved 6 January,2015 from http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx

[3] Kinect Guides. Retrieved 8 January ,2015 from http://www.kinecthacks.com/

[4] Jarrett Webb , James Ashley (2015) : Kinect Programming with the Microsoft Kinect SDK.

[5] Skeleton Tracking with Kinect. Retrieved 8 January ,2015 from http://msdn.microsoft.com/en-us/library/jj131025.aspx

[6] Skeleton Tracking Fundamentals. Retrieved 8 January,2015 from http://channel9.msdn.com/Series/KinectSDKQuickstarts/Skeletal-Tracking-Fundamentals

[7] การควบคุม Servo motor. Retrieved 2 February 2015 from http://www.premierac.co.th/index.php?lay=boardshow&ac=webboard_show&No=119584

[8] Kinect Controls Arduino wired Servo using VB. Retrieved 8 February,2015 from http://www.instructables.com/id/Kinect-controls-Arduino-wired-Servos-using-Visual-/

[9] ข้อมูลเกี่ยวกับ Arduino Mega 2560 Retrieved 13 April,2015 from http://www.arduino.cc/en/Main/ArduinoBoardMega2560

[10] Siriwimon Sunthon (2557) : ศึกษาข้อมูลของบอร์ด Arduino Mega2560 Retrieved 13 April,2015 from http://mbeddedweekly.blogspot.com/2014/08/arduino-mega2560.html

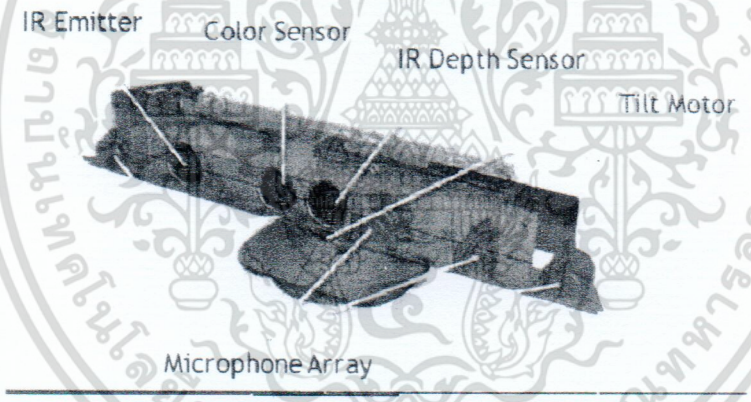
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

Datasheet Kinect

PRODUCT SPECIFICATION

Field of View (Horizontal, Vertical, Diagonal)	58° H, 40° V, 70° D	Color image size	UXGA (1600×1200)
Depth image size	VGA (640×480)	Audio: built-in microphones	2 mics
Spatial x/y resolution (@2m distance from sensor)	3mm	Audio: digital inputs	4 inputs
Depth z resolution (@2m distance from sensor)	1cm	Data interface	USB 2.0
Maximal image throughput (frame rate)	60fps	Power supply	USB 2.0
Average image latency in full VGA resolution	40msec	Power consumption	2.25W
Operation range	0.8m - 3.5m	Dimensions (Width x Height x Depth)	14cm x 3.5cm x 5cm
		Operation environment (every lighting condition)	indoor
		Operating temperature	0°C - 40°C



Product Feature:Horizontal field of view : 57 degrees

Vertical field of view : 43 degrees

Physical tilt range : ± 27 degrees

Depth sensor range : 1.2 m – 3.5m

Data Streams : 320x240 16-bit depth @ 30 frames/sec

640x480 32-bit colour @ 30 frames/sec

16-bit audio @ 16 kHz

Skeletal Tracking System : Tracks up to 6 People , includeing 2 active player

Tracks 20 joints per active player

Audio System : Echo cancellation system enhances voice input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code Project

```
'Arduino start
Imports System
Imports System.IO.Ports
Imports System.Threading
'Arduino end

Imports System.Text
Imports Microsoft.Kinect
Imports Coding4Fun.Kinect.Wpf

Namespace SkeletalTracking

    ''' <summary>
    ''' Interaction logic for MainWindow.xaml
    ''' </summary>
    Partial Public Class MainWindow
        Inherits Window

    ' Arduino start
    Public Shared _continue As Boolean
    Public Shared _serialPort As SerialPort



    Public Shared ScreenMaxX As Integer = 500
    Public Shared ScreenMaxY As Integer = 390


    ' Define maximum thresholds for relating Kinect to Shoulder
    Public Shared sMaxKinectHandY As Single = 0.25
    Public Shared bMaxShoulder As Byte = 185
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Shared sMinKinectHandY As Single = 0.7
Public Shared bMinShoulder As Byte = 47
Public Shared rbPCR As RobobuilderLib.PCremote
Public Shared rbWCK As RobobuilderLib.wckMotion
Public bRBConnected As Boolean = False

Private WithEvents sensor As KinectSensor
        Public Sub New()
                InitializeComponent()
        End Sub
Private Property components As ComponentModel.Container
Private Property control As UserControl
Private Property Timer As Windows.FontStyle
        Private Sub Window_Loaded(ByVal sender As Object, ByVal e As
RoutedEventArgs)
    ' will discover and initialize Kinect
    DiscoverKinectSensor()
    ' DisconnectFromRB()
End Sub
Private Sub sensor_SkeletonFrameReady(sender As Object, e As
Microsoft.Kinect.SkeletonFrameReadyEventArgs) Handles sensor.SkeletonFrameReady
    Dim skFrame As SkeletonFrame
    Dim skeletonSlot As Integer = 0
    Dim playerSkeleton As Skeleton
    skFrame = e.OpenSkeletonFrame
    If Not skFrame Is Nothing Then
        Dim skeletonData(skFrame.SkeletonArrayLength - 1) As Skeleton
        skFrame.CopySkeletonDataTo(skeletonData)
        playerSkeleton = (From s In skeletonData Where s.TrackingState =
SkeletonTrackingState.Tracked Select s).FirstOrDefault()
        If Not playerSkeleton Is Nothing Then

            '''''''''''''''''''''''''''''''''''''''''''''''''''Servo5''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            Dim shoulderRx =
CStr(playerSkeleton.Joints(JointType.ShoulderRight).Position.X)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Dim shoulderRy =
CStr(playerSkeleton.Joints(JointType.ShoulderRight).Position.Y)
            Dim shoulderRz =
CStr(playerSkeleton.Joints(JointType.ShoulderRight).Position.Z)
            Dim elbowRx =
CStr(playerSkeleton.Joints(JointType.ElbowRight).Position.X)
            Dim elbowRy = CStr(playerSkeleton.Joints(JointType.ElbowRight).Position.Y)
            Dim elbowRz =
CStr(playerSkeleton.Joints(JointType.ElbowRight).Position.Z)
            Dim wristRx = CStr(playerSkeleton.Joints(JointType.WristRight).Position.X)
            Dim wristRy = CStr(playerSkeleton.Joints(JointType.WristRight).Position.Y)
            Dim wristRz = CStr(playerSkeleton.Joints(JointType.WristRight).Position.Z)

            Dim ABVecterX = CStr((shoulderRx - elbowRx) * (wristRx - elbowRx)) +
((shoulderRy - elbowRy) * (wristRy - elbowRy)) + ((shoulderRz - elbowRz) * (wristRz -
elbowRz))
            Dim ABVecter = CStr((((shoulderRx - elbowRx) ^ 2) + ((shoulderRy -
elbowRy) ^ 2) + ((shoulderRz - elbowRz) ^ 2)) ^ 0.5) * ((((wristRx - elbowRx) ^ 2) +
((wristRy - elbowRy) ^ 2) + ((wristRz - elbowRz) ^ 2)) ^ 0.5))
            Dim ss5 = (260 - CInt((180 * Math.Acos((ABVecterX) / (ABVecter))) /
3.14159))
            Dim sss5 As Byte
            If ss5 > 0 Then
                sss5 = (260 - CInt((180 * Math.Acos((ABVecterX) / (ABVecter))) / 3.14159))
            End If
            check.Text = sss5
    'Servo6'
            Dim shoulderLx =
CStr(playerSkeleton.Joints(JointType.ShoulderLeft).Position.X)
            Dim shoulderLy =
CStr(playerSkeleton.Joints(JointType.ShoulderLeft).Position.Y)
            Dim shoulderLz =
CStr(playerSkeleton.Joints(JointType.ShoulderLeft).Position.Z)
            Dim elbowLx = CStr(playerSkeleton.Joints(JointType.ElbowLeft).Position.X)
            Dim elbowLy = CStr(playerSkeleton.Joints(JointType.ElbowLeft).Position.Y)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Dim elbowLz = CStr(playerSkeleton.Joints(JointType.ElbowLeft).Position.Z)
            Dim wristLx = CStr(playerSkeleton.Joints(JointType.WristLeft).Position.X)
            Dim wristLy = CStr(playerSkeleton.Joints(JointType.WristLeft).Position.Y)
            Dim wristLz = CStr(playerSkeleton.Joints(JointType.WristLeft).Position.Z)

            Dim LABVecterX = CStr((shoulderLx - elbowLx) * (wristLx - elbowLx)) +
((shoulderLy - elbowLy) * (wristLy - elbowLy)) + ((shoulderLz - elbowLz) * (wristLz -
elbowLz))
            Dim LABVecter = CStr(((((shoulderLx - elbowLx) ^ 2) + ((shoulderLy -
elbowLy) ^ 2) + ((shoulderLz - elbowLz) ^ 2)) ^ 0.5) * ((((wristLx - elbowLx) ^ 2) +
((wristLy - elbowLy) ^ 2) + ((wristLz - elbowLz) ^ 2)) ^ 0.5))
            Dim ss6 = 190 - (260 - CInt((180 * Math.Acos((LABVecterX) / (LABVecter))) /
3.14159))
            Dim sss6 As Byte
            If ss6 > 0 Then
                sss6 = 190 - (260 - CInt((180 * Math.Acos((LABVecterX) / (LABVecter))) /
3.14159))
            End If
            check2.Text = sss6
            Dim bServo11 As Byte
            Dim bServo14 As Byte
            Dim bServo13 As Byte
            Dim bServo10 As Byte
            ' Right Arm
            Dim HR = CStr(((180 *
Math.Atan(playerSkeleton.Joints(JointType.WristRight).Position.X /
playerSkeleton.Joints(JointType.WristRight).Position.Z)) / 3.14159) + 2)
            ww.Text = HR
            Dim R0 = playerSkeleton.Joints(JointType.HandRight).ScaleTo(1, 180, 0.5F,
0.5F)
            Dim s22 = R0.Position.Y
            Dim scaledRightArm1 =
playerSkeleton.Joints(JointType.HandRight).ScaleTo(1, 180, 0.5F, 1.5F)
            If HR < 10 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Dim scaledRightArm3 =
playerSkeleton.Joints(JointType.HandRight).ScaleTo(1, 180, 0.5F, 0.5F)
                bServo13 = CByte(240 - (180 - scaledRightArm3.Position.Y +
bMinShoulder))
                ' If HR < 10 Then
                bServo14 = 180
            Else
                If s22 < 70 Then
                    bServo13 = 20
                    bServo14 = 178 - CByte(227 - (180 - scaledRightArm1.Position.Y +
bMinShoulder))
                Else
                    If s22 > 110 Then
                        bServo13 = 180
                        bServo14 = 100
                    Else
                        If 70 < s22 < 110 Then
                            bServo13 = 20
                            bServo14 = 178 - CByte(227 - (180 - scaledRightArm1.Position.Y
+ bMinShoulder))

                        End If
                    End If
                End If
            End If

            ' Left Arm
            Dim HL = CStr(((-1 * 180 *
Math.Atan(playerSkeleton.Joints(JointType.WristLeft).Position.X /
playerSkeleton.Joints(JointType.WristLeft).Position.Z)) / 3.14159))
            ee.Text = HL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Dim L0 = playerSkeleton.Joints(JointType.HandLeft).ScaleTo(1, 180, 0.5F,
0.5F)
            Dim s33 = L0.Position.Y
            Dim scaledLeftArm1 =
playerSkeleton.Joints(JointType.HandLeft).ScaleTo(1, 180, 0.5F, -0.5F)
            Dim b1 = CInt(scaledLeftArm1.Position.Y) - 20
            Dim scaledLeftArm = playerSkeleton.Joints(JointType.HandLeft).ScaleTo(1,
180, 0.5F, 1.5F)
            If HL < 12.5 Then
                If b1 > 0 Then
                    bServo10 = CInt(scaledLeftArm1.Position.Y) - 20
                End If
            Else
                If s33 < 70 Then
                    bServo10 = 160
                    bServo11 = CByte(scaledLeftArm.Position.Y) - 15
                Else
                    If s33 > 110 Then
                        bServo10 = 0
                        bServo11 = 65
                    End If
                End If
            End If
            Dim bServo12 As Byte
            Dim dServo11 As Double = CDbl(bServo11)
            Dim bServo15 As Byte
            Dim dServo14 As Double = CDbl(bServo14)
            Dim bservo6 As Byte
            Dim bservo1 As Byte
            Dim scaledLeftArm2 =
playerSkeleton.Joints(JointType.FootRight).ScaleTo(1, 300, 0.5F, -1.0F)
            bservo6 = CByte(272 - (180 - scaledLeftArm2.Position.Y + 0))
            Dim scaledLeftFoot2 =
playerSkeleton.Joints(JointType.FootLeft).ScaleTo(1, 300, 0.5F, -1.0F)
            bservo1 = CByte(51 - (190 - (180 - scaledLeftFoot2.Position.Y + 50)))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            s1.Text = bServo14        '180
            s2.Text = bServo13
            s3.Text = bServo10
            s4.Text = bServo11        '0
            s5.Text = sss5
            s6.Text = sss6
            s9.Text = bservo6
            s10.Text = bservo1
            Me.components = New System.ComponentModel.Container()
            MainWindow._serialPort = New
System.IO.Ports.SerialPort(Me.components())
            MainWindow._serialPort.PortName = "COM3"
            _serialPort.DataBits = 8
            _serialPort.Handshake = 0
            _serialPort.ReadTimeout = 500
            _serialPort.WriteTimeout = 500
            Me.control = New System.Windows.Controls.UserControl()
            Me.Timer = New System.Windows.FontStyle()
            MainWindow._serialPort.BaudRate = 9600
            Dim test() As Byte = {bServo10, bServo11, sss6, bServo14, bServo13, sss5,
bservo6, bservo1}
            _serialPort.Open()
            _serialPort.Write(test, 0, test.Length)
            _serialPort.Close()
            Me.InvalidateVisual()
        End If
    End If
End Sub
Private Sub SetEllipsePosition(ByVal ellipse As FrameworkElement, ByVal kjoint As
Joint, ByVal JID As JointType)
    Dim scaledJoint = kjoint.ScaleTo(ScreenMaxX, ScreenMaxY, 0.5F, 0.2F)
    Canvas.SetLeft(ellipse, scaledJoint.Position.X)
    Canvas.SetTop(ellipse, scaledJoint.Position.Y)
End Sub
Private Sub Window_Closed(ByVal sender As Object, ByVal e As EventArgs)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        'Cleanup
        sensor.Stop()
        sensor.Dispose()
    End Sub
    Private Sub DiscoverKinectSensor()
        For Each iterSensor As KinectSensor In KinectSensor.KinectSensors
            If iterSensor.Status = KinectStatus.Connected Then
                sensor = iterSensor
                Exit For
            End If
        Next
        If sensor Is Nothing Then
            MsgBox("Could not find any valid Kinect Connected. Please restart the
program", vbCritical)
            Exit Sub
        End If
        Select Case sensor.Status
            Case KinectStatus.Connected
            Case KinectStatus.Disconnected
                MsgBox("Kinect is Disconnected", vbExclamation)
            Case KinectStatus.NotPowered
                MsgBox("Kinect is Not Powered. Connect to Power", vbExclamation)
            Case Else
                MsgBox("Unkown Kinect Status", vbExclamation)
        End Select
        If (sensor.Status = KinectStatus.Connected) Then
            InitializeKinect()
        End If
    End Sub
    Private Sub InitializeKinect()
        Dim parameters = New TransformSmoothParameters With {.Smoothing = 0.8F,
.Correction = 0.3F, .Prediction = 0.4F, .JitterRadius = 1.0F, .MaxDeviationRadius = 0.5F}
        sensor.SkeletonStream.Enable(parameters)

        sensor.Start()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End Sub
End Class
End Namespace
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้