หุ่นยนต์ควบคุมด้วย KINECTและ ARDUINO

ROBOT CONTROLLED WITH KINECT AND ARDUINO

ศุภฤกษ์ จัตตะมละกุล
SUPAROEK JUTTAMALAKUL

อริยะ อารีชม
ARIYA AREECHOM

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2557

หุ่นยนต์ควบคุมด้วย KINECT และ ARDUINO

ROBOT CONTROLLED WITH KINECT AND ARDUINO

ศุภฤกษ์ จัตตะมละกุล

อริยะ อารีชม

อาจารย์ที่ปรึกษา

ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา	2557
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์
คณะ	วิศวกรรมศาสตร์
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง	หุ่นยนต์ควบคุมด้วย KINECT และ ARDUINO
	(ROBOT CONTROLLED KINECT AND ARDUINO)
ผู้จัดทำ	นายศุภฤกษ์ จัตตะมละกุล รหัสประจำตัว 54011293
	นายอริยะ อารีชม รหัสประจำตัว 54011525

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ
อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์	หุ่นยนต์ควบคุมด้วย Kinect และ Arduino
นักศึกษา	นายศุภฤกษ์ จัตตะมละกุล รหัสนักศึกษา 54011293
	นายอริยะ อารีชม รหัสนักศึกษา 54011525
ปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2557
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ

บทคัดย่อ

โครงงานนี้นำเสนอการควบคุมการเคลื่อนไหวหุ่นยนต์โดยใช้กล้องตรวจจับการเคลื่อนไหว (Kinect) และ Arduino ซึ่งเป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source โดยทางผู้จัดทำโครงงานได้มองเห็นว่าในปัจจุบันหุ่นยนต์ได้เข้ามามีบทบาทในการดำเนินชีวิตของเราเพิ่มมากขึ้น จึงได้คิดโครงงานที่จะนำหุ่นยนต์เข้ามาทำกิจกรรมแทนมนุษย์ที่ส่งผลกระทบทางด้านความปลอดภัยต่อมนุษย์ โดยที่เราจะออกแบบให้หุ่นยนต์สามารถเคลื่อนไหวได้ตามมนุษย์ โดยใช้กล้องตรวจจับการเคลื่อนไหว เป็นตัวจับการเคลื่อนไหวของมนุษย์ในท่าทางการเดินต่างๆ ความมุ่งหวังด้านการกู้ภัยเพื่อประโยชน์ทางด้านความมั่นคง ความปลอดภัยในชีวิตและทรัพย์สิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis	ROBOT CONTROLLED WITH KINECT AND ARDUINO
Student	Mr. Suparoek Juttamalakul ID 54011293
	Mr. Ariya Areechom ID 54011525
Degree Program	Bachelor of Engineering Electronics Engineering
Year	2014
Thesis Advisor	Asst.Prof.Dr.Pattarapong Phasukkit

ABSTRACT

This project presents a motion control Robobuilder with Kinect and this project used board Arduino AVR family of microcontrollers in the development of open source. Can see that the current robot has a role in our lives increase. It was thought the project to bring robots come do the human activities that impact side safe for humans. We will design the robot can move by a man by catch motion of humans in various postures with Kinect. The aim of the rescue to support stability. Safety of life and property.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานเรื่องหุ่นยนต์ควบคุมด้วย Kinect and Arduino (Robot controlled with Kinect and Arduino) สามารถสำเร็จลุล่วงไปได้ด้วยดี ทั้งนี้เพราะได้รับคำแนะนำและคำปรึกษาจาก ผศ.ดร. ภัทรพงษ์ ผาสุขกิจ อาจารย์ที่ปรึกษาในการทำโครงงาน ผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์จากท่าน และขอกราบขอบพระคุณเป็นอย่างสูงนอกจากนี้ขอขอบพระคุณทุกๆท่านที่ให้ความช่วยเหลือ ตลอดจนให้คำแนะนำต่างๆจนทำให้โครงงานฉบับนี้สำเร็จโดยสมบูรณ์ได้คุณค่าและประโยชน์อันพึงมีในรายงานฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

ศุภฤกษ์ จัตตะมละกุล
อริยะ อารีชม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VI
สารบัญรูป.....VII
บทที่ 1 บทนำ.....1
1.1 ที่มาและความสำคัญ.....1
1.2 วัตถุประสงค์.....1
1.3 ขอบเขตของงานที่ทำ.....1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....1
1.5 แผนการดำเนินงาน.....2
1.6 โครงสร้างของโครงงาน.....2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....3
2.1 Kinect.....3
2.1.1 หลักการทำงานของ Kinect Sensor.....5
2.2 Arduino.....8
2.3 Servo Motor.....11
2.3.1 องค์ประกอบพื้นฐานของเซอร์โว.....12
2.4. ทฤษฎีเกี่ยวกับหุ่นยนต์.....14
2.4.1 การพัฒนาหุ่นยนต์ฮิวแมนนอยด์.....15
2.4.2 การก้าวเดินแบบอิสระ.....15
2.4.3 เทคนิคการก้าวเดินอย่างมีเสถียรภาพ.....16
2.5 เครื่องพิมพ์ 3 มิติ (3D Printers).....17
2.5.1 Stereolithography (SLA).....18
2.5.2 Fused Deposition Modeling (FDM).....18
2.5.3 3D jet printer systems.....18
2.5.4 Selective Laser Sintering (SLS).....19
2.5.5 Digital Light Processing (DLP).....19
2.5.6 Laminated Object Manufacturing (LOM).....20
บทที่ 3 การออกแบบ.....21
3.1 การออกแบบและการสร้าง.....21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบโปรแกรม.....22
3.2.1 ส่วนของ Arduino.....22
3.2.2 ส่วนของ Kinect.....23
3.3 การออกแบบชิ้นงาน.....29
บทที่ 4 ผลการทดสอบ.....31
4.1 ผลการทดสอบท่าที่ 1 ท่ายกขาซ้ายขึ้น.....31
4.2 ผลการทดสอบท่าที่ 2 ท่าเตะขาซ้ายไปข้างหน้า.....33
4.3 ผลการทดสอบท่าที่ 3 ท่าขาซ้ายลง.....35
4.4 ผลการทดสอบท่าที่ 4 ท่าขาซ้ายอยู่ที่พื้น.....37
4.5 ผลการทดสอบท่าที่ 5 ท่ายกขาขวาขึ้น.....39
4.6 ผลการทดสอบท่าที่ 6 ท่าเตะขาขวาไปข้างหน้า.....41
4.7 ผลการทดสอบท่าที่ 7 ท่าขาขวาลง.....43
4.8 ผลการทดสอบท่าที่ 8 ท่าขาขวาอยู่ที่พื้น.....45
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....47
5.3.1 สรุปผลการดำเนินงาน.....47
5.3.2 ปัญหาที่พบ.....47
5.3.3 ข้อเสนอแนะ.....47
เอกสารอ้างอิง.....48
ภาคผนวก.....48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า
ตารางที่ 4.1 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1.....31
ตารางที่ 4.2 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 2.....33
ตารางที่ 4.3 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 3.....35
ตารางที่ 4.4 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4.....37
ตารางที่ 4.5 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 5.....39
ตารางที่ 4.6 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 6.....41
ตารางที่ 4.7 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 7.....43
ตารางที่ 4.8 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 8.....45

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า
รูปที่ 2.1 Kinect Sensor....3
รูปที่ 2.2 ระบบภายในของ Kinect Sensor....4
รูปที่ 2.3 ตัวอย่างการทำงานของ Kinect....4
รูปที่ 2.4 อินฟราเฟรดของ Kinect....5
รูปที่ 2.5 ระดับความลึกตื้นของภาพที่ถ่ายด้วย Kinect....5
รูปที่ 2.6 Skeletal Tracking จับที่คน....6
รูปที่ 2.7 ระบบ Skeletal Tracking....6
รูปที่ 2.8 ระยะการตรวจจับการเคลื่อนไหวของ Kinect....7
รูปที่ 2.9 Arduino Mega 2560 R3 Front....8
รูปที่ 2.10 Arduino Mega 2560 R3 Back....8
รูปที่ 2.11 Servo Motor....11
รูปที่ 2.12 Servo-parts....12
รูปที่ 2.13 Servo wire....13
รูปที่ 2.14 Diagram การทำงานของ Servo Motor....13
รูปที่ 2.15 สัญญาณพัลส์ การทำงานของ Servo Motor....14
รูปที่ 2.16 ข้อต่อส่วนต่างๆของหุ่นยนต์....16
รูปที่ 2.17 Stereo lithographic 3-D printer....17
รูปที่ 2.18 วิธีการแบบ Fused Deposition Modeling (FDM)....18
รูปที่ 2.19 เครื่อง 3D jet printer systems....19
รูปที่ 2.20 Selective Laser Sintering (SLS)....19
รูปที่ 2.21 วิธีแบบ Digital Light Processing (DLP)....20
รูปที่ 2.22 วิธีการแบบ Laminated Object Manufacturing (LOM)....20
รูปที่ 3.1 ผังการทำงานของระบบ....21
รูปที่ 3.2 ผังการทำงานของโปรแกรม Arduino....22
รูปที่ 3.3 ผังการทำงานของโปรแกรม Kinect ส่วนหัวเข่าขวา....23
รูปที่ 3.4 ผังการทำงานของโปรแกรม Kinect ส่วนหัวเข่าซ้าย....24
รูปที่ 3.5 ผังการทำงานของโปรแกรมการยกขาไปด้านซ้ายขึ้น
และเอียงตัวไปทางด้านขวา 20 องศา....25
รูปที่ 3.6 ผังการทำงานของโปรมแกรมการยกขาด้านขวาขึ้น
และเอียงตัวไปทางด้านซ้าย 20 องศา....26
รูปที่ 3.7 ภาพแสดงหน้าต่างโปรแกรม Microsoft Visual Basic
ส่วนโค้ดโปรแกรม....27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ 3.8 ภาพแสดงหน้าต่างโปรแกรม Microsoft Visual Basic
ส่วนหน้าต่างแสดงผล....27
รูปที่ 3.9 ภาพแสดงหน้าต่างโปรแกรม Arduino
ส่วนโค้ดโปรแกรม....28
รูปที่ 3.10 พลาสติก PLA....29
รูปที่ 3.11 เครื่องปริ๊น3D....29
รูปที่ 3.12 ชิ้นส่วนของหุ่นยนต์ก่อนประกอบ....30
รูปที่ 3.13 แสดงชิ้นส่วนของหุ่นยนต์....30
รูปที่ 3.14 หุ่นยนต์ที่ประกอบเสร็จสมบูรณ์....30
รูปที่ 4.1 รูปแสดงการเคลื่อนไหวท่าที่ 1....31
รูปที่ 4.2 กราฟแสดงความกว้าง Pulse ของ Servo
ที่ใช้ในการเคลื่อนที่ท่าที่ 1 จากตารางที่ 4.1....32
รูปที่ 4.3 รูปแสดงการเคลื่อนไหวท่าที่ 2....33
รูปที่ 4.4 กราฟแสดงความกว้าง Pulse ของ Servo
ที่ใช้ในการเคลื่อนที่ท่าที่ 1 จากตารางที่ 4.2....34
รูปที่ 4.5 รูปแสดงการเคลื่อนไหวท่าที่ 3....35
รูปที่ 4.6 กราฟแสดงความกว้าง Pulse ของ Servo
ที่ใช้ในการเคลื่อนที่ท่าที่ 1 จากตารางที่ 4.3....36
รูปที่ 4.7 รูปแสดงการเคลื่อนไหวท่าที่ 4....37
รูปที่ 4.8 กราฟแสดงความกว้าง Pulse ของ Servo
ที่ใช้ในการเคลื่อนที่ท่าที่ 1 จากตารางที่ 4.4....38
รูปที่ 4.9 รูปแสดงการเคลื่อนไหวท่าที่ 5....39
รูปที่ 4.10 กราฟแสดงความกว้าง Pulse ของ Servo
ที่ใช้ในการเคลื่อนที่ท่าที่ 1 จากตารางที่ 4.5....40
รูปที่ 4.11 รูปแสดงการเคลื่อนไหวท่าที่ 6....41
รูปที่ 4.12 กราฟแสดงความกว้าง Pulse ของ Servo
ที่ใช้ในการเคลื่อนที่ท่าที่ 1 จากตารางที่ 4.6....42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ 4.13 รูปแสดงการเคลื่อนไหวท่าที่ 7.....43
รูปที่ 4.14 กราฟแสดงความกว้าง Pulse ของ Servo
ที่ใช้ในการเคลื่อนที่ท่าที่ 1 จากตารางที่ 4.7.....44
รูปที่ 4.15 รูปแสดงการเคลื่อนไหวท่าที่ 8.....45
รูปที่ 4.16 กราฟแสดงความกว้าง Pulse ของ Servo
ที่ใช้ในการเคลื่อนที่ท่าที่ 1 จากตารางที่ 4.8.....46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันเทคโนโลยีมีความก้าวหน้าไปมากแล้วได้เข้ามามีส่วนในการดำเนินชีวิตของมนุษย์มากขึ้น ทางด้านผู้จัดทำจึงเห็นว่าหุ่นยนต์ก็เป็นเทคโนโลยีที่เราน่าจะนำใช้ให้เกิดประโยชน์สูงสุดได้และน่าจะเข้ามามีส่วนช่วยในการทำงานของได้ อย่างงานที่มีความอันตรายต่อชีวิตของผู้ปฏิบัติหน้าที่ ซึ่งในตอนนี้ทางด้านสามจังหวัดชายแดนภาคใต้มีเหตุการณ์ที่อันตรายต่อชีวิตของประชาชนที่ได้อาศัยอยู่และเจ้าหน้าที่ที่เข้าไปปฏิบัติหน้าที่ในบริเวณที่เกิดเหตุ ผู้จัดทำจึงได้คิดว่าจะนำหุ่นยนต์เข้ามาปฏิบัติหน้าที่แทนคนในงานที่เสี่ยงอันตราย เช่น การเก็บกู้วัตถุระเบิดในสามจังหวัดชายแดนภาคใต้ โดยมนุษย์จะบังคับหุ่นยนต์ให้เคลื่อนที่ตามคนโดยใช้อุปกรณ์ที่มีความสามารถในการตรวจจับที่มีความแม่นยำที่มีชื่อว่า Kinect เป็นอุปกรณ์เสริมของเคลื่อนเล่น Xbox ทางด้านผู้จัดทำจึงนำเอา Kinect มาตรวจจับการเคลื่อนไหวของมนุษย์แล้วส่งค่าไปยังหุ่นยนต์ให้เคลื่อนไหวตามที่เราต้องการ

1.2 วัตถุประสงค์

1. เพื่อศึกษาการเขียนโปรแกรม Microsoft Visual Studio 2010
2. ศึกษาการทำงานของ Kinect เพื่อเป็นแนวทางในการนำไปประยุกต์ใช้งาน
3. เพื่อในอนาคตอาจจะเป็นตัวแทนมนุษย์ในการใช้ประโยชน์ต่างๆ เช่น กู้ภัย เป็นต้น

1.3 ขอบเขตของงานที่ทำ

งานนี้เป็นการควบคุมหุ่นยนต์ด้วย Kinect ให้หุ่นยนต์สามารถเดินได้ โดยที่ Kinect จะรับค่าจากการที่มนุษย์เคลื่อนไหว ไปยังโปรแกรมที่เขียนไว้บนคอมพิวเตอร์และนำข้อมูลที่ได้ส่งไปยังตัวหุ่นยนต์ให้เคลื่อนไหว

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- ได้รับความรู้ถึงหลักการเขียนโปรแกรม Microsoft Visual Basic และ Arduino
- ได้เรียนรู้ที่ประยุกต์การใช้งานจากหลายโปรแกรมมาทำงานบนชิ้นงานเดียว
- สามารถนำปริญญานิพนธ์นี้มาใช้งานได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ม.ค.2558	ก.พ.2558	มี.ค.2558	เม.ย.2558
1.ศึกษาเกี่ยวกับตัวหุ่นยนต์และประกอบหุ่นยนต์	←	—	→	
2.เขียน code แก้ไขข้อบกพร่อง/ทดลอง		←	—	→
3.รวบรวมข้อมูลทำรายงาน/สรุปผล			←	— →

1.6 โครงสร้างของโครงงาน

โดยเนื้อหาของรายงานได้มีการแบ่งออกเป็นบทต่างๆ รายละเอียดของรายงานแต่ละบทมีดังนี้

บทที่1 บทนำ จะกล่าวถึงที่มาและความสำคัญของโครงงาน วัตถุประสงค์ ขอบเขตของงานที่ทำ ประโยชน์ที่คาดว่าจะได้รับ และแผนการดำเนินงานของโครงงาน

บทที่ 2 ทฤษฎี จะกล่าวถึงทฤษฎีที่เกี่ยวข้องและหลักการพื้นฐานของ Kinect, Arduino , หลักการพื้นฐานของ Servo และทฤษฎีเกี่ยวกับหุ่นยนต์

บทที่ 3 การออกแบบ จะกล่าวถึงขั้นตอนการออกแบบส่วนต่างๆของรายงาน ซึ่งจะแบ่งออกเป็นทางด้านโปรแกรม และทางด้านโครงสร้าง

บทที่ 4 ผลการดำเนินงาน จะกล่าวถึงส่วนของผลที่ได้จากการเขียนโปรแกรมควบคุมการเคลื่อนไหวของหุ่นยนต์

บทที่ 5 สรุปและวิจารณ์ จะกล่าวถึงสรุปผลการดำเนินงาน สิ่งที่ได้รับจากการทำโครงงาน และปัญหาที่พบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 Kinect

Kinect คือ อุปกรณ์เสริมของเครื่องเล่นเกมส์ Xbox360 จาก Microsoft โดยสามารถจดจำผู้เล่น (Facial Recognition) และให้ผู้เล่นควบคุมเกมส์ผ่านทางการเคลื่อนไหวร่างกายของผู้เล่นโดยตรง (3D Motion Recognition) โดยไม่จำเป็นต้องมี joystick 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

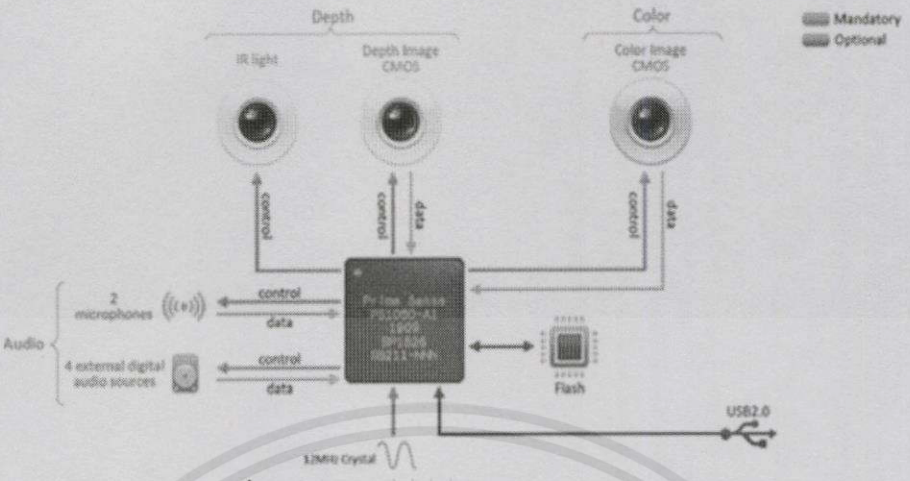
2.กล้อง IR (Infrared) 1 ตัว

3.ตัวส่งแสง IR (Infrared)

4.ไมโครโฟน 4 ตัว

5.มอเตอร์ไว้ปรับมุมกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ระบบภายในของ Kinect Sensor

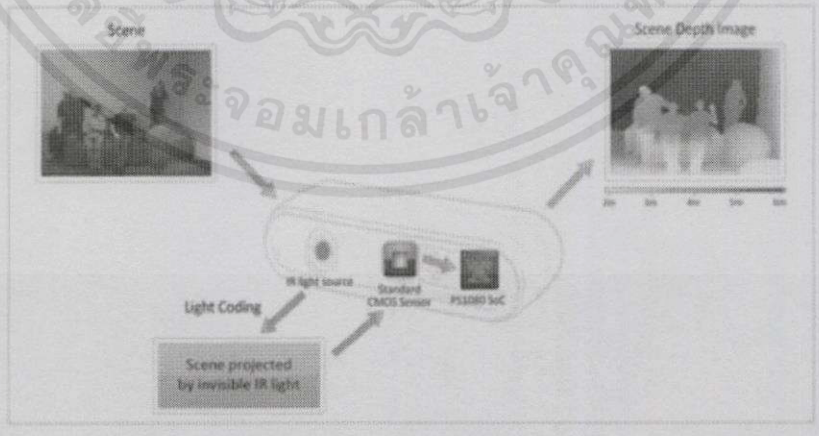
Kinect มีการหน้าที่ใหญ่ๆอยู่ 3 หน้าที่ ดังนั้นเราจะเชื่อมระหว่าง Hardware กับหน้าที่การทำงานของ Kinect ดังนี้

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



รูปที่ 2.3 ตัวอย่างการทำงานของ Kinect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 จดจำการเคลื่อนไหวของคน โดยหลังจากแบ่งแยกประเภทของคนแล้ว จะใช้วิธี skeletal movement (การเคลื่อนไหวตามลักษณะกระดูก) เข้าไปวิเคราะห์คนประเภทนั้นว่า ส่วนไหน คือหัว ตัวคอหัวไหล่ แขนขวาแขนซ้าย ขาขวาขาซ้าย ข้อศอก, โดยใช้สัดส่วน ลักษณะ หรือข้อจำกัดต่างๆ 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

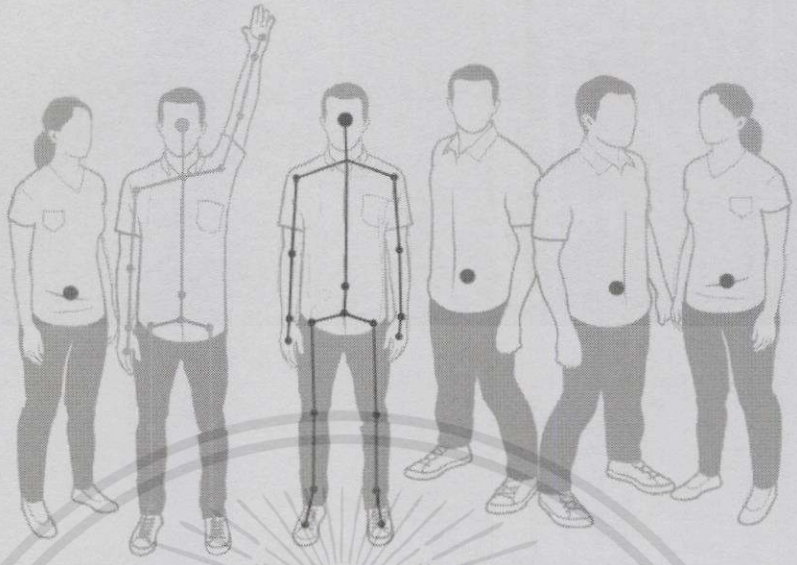
รูปที่ 2.4 อินฟราเฟรดของ Kinect

จากนั้น Infrared camera จะรับระดับความสว่างของแสงที่ตกกระทบบนวัตถุ โดยวัตถุที่อยู่ใกล้ แสงจะมีความสว่างมากและวัตถุที่อยู่ไกลจะมีแสงสว่างน้อยกว่า จากนั้นจะนำข้อมูลที่ได้ไปประมวลผล ด้วยซอฟต์แวร์จะได้ระดับความลึกตื้นออกมาดังภาพด้านล่าง โดยวัตถุที่อยู่ใกล้กับตัวเซนเซอร์จะมีสีอ่อน และวัตถุที่อยู่ไกลจะมีสีเข้มไล่ระดับกันไป

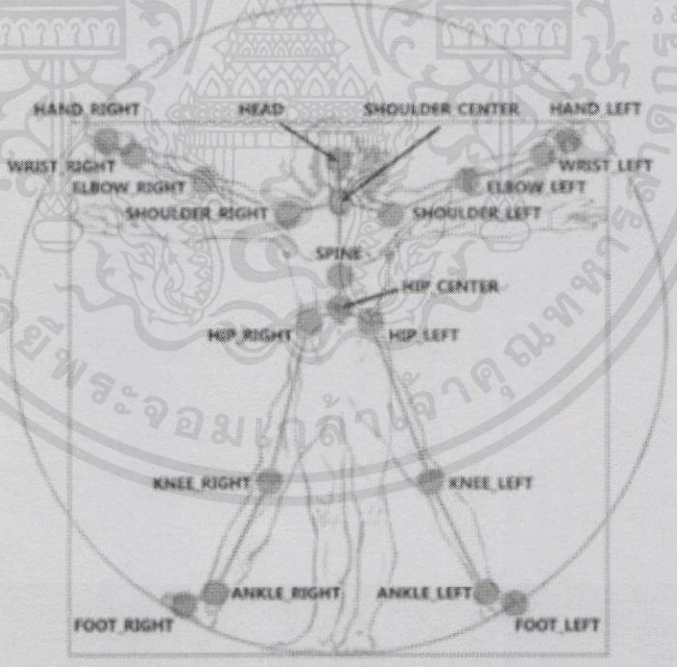
รูปที่ 2.5 ระดับความลึกตื้นของภาพที่ถ่ายด้วย Kinect

เมื่อ Kinect รู้ระดับความตื้นลึกแล้วก็จะสามารถแยกผู้เล่นออกจากสภาพแวดล้อมภายในห้องได้ นอกจากนั้น Kinect Sensor ยังมีระบบ Skeletal Tracking ที่ใช้ติดตามโครงกระดูกของผู้ใช้งาน ซึ่ง สามารถติดตามได้มากที่สุด 2 คน แต่จะมองเห็นทั้งหมด 6 คน ซึ่งภาพที่แสดงจะเป็นภาพของโครง กระดูกมนุษย์ที่แทนด้วยข้อต่อ 20 จุดสำคัญตามร่างกาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



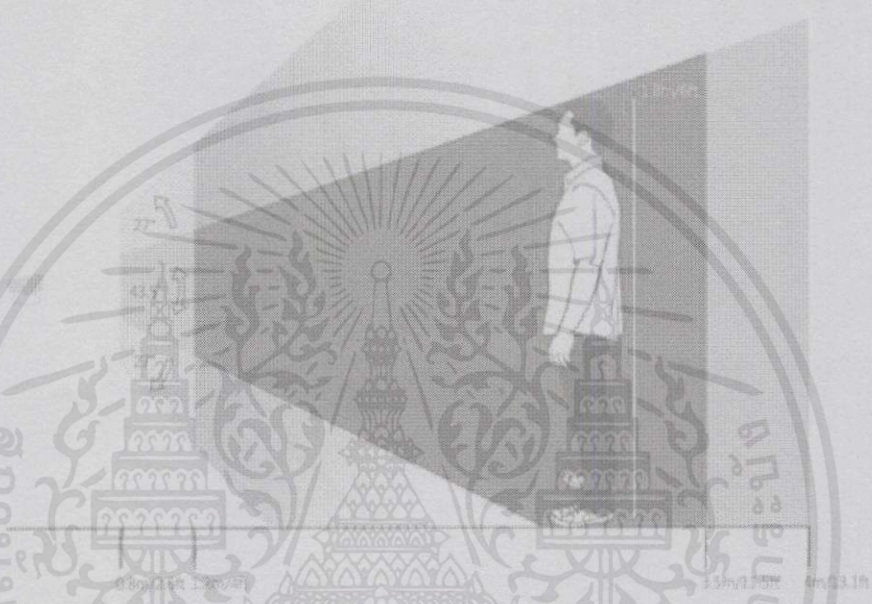
รูปที่ 2.6 Skeletal Tracking จับที่คน



รูปที่ 2.7 ระบบ Skeletal Tracking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

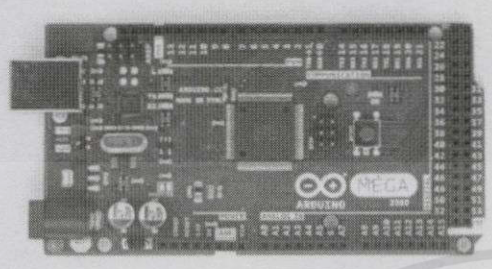


รูปที่ 2.8 ระยะการตรวจจับการเคลื่อนไหวของ Kinect

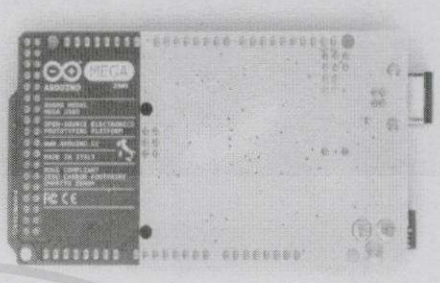
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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 และให้กระแสได้สูงสุด50 mA
- GND เป็น ground pin
- IOREF เป็น pin ที่ให้ voltage reference กับไมโครคอนโทรลเลอร์ เพื่อเลือกค่าแรงดันให้กับ shield ที่มาเชื่อมต่อกับบอร์ด

Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Servo Motor

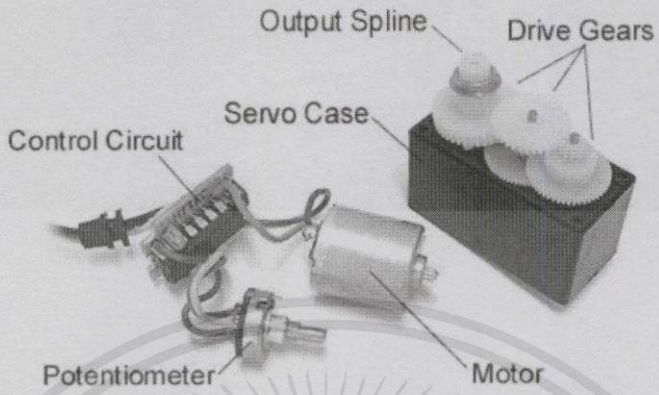
Servo Motor เป็นมอเตอร์ที่ทำงานโดยใช้สัญญาณพัลส์ โดยภายในเซอร์โวมอเตอร์จะประกอบไปด้วย มอเตอร์ไฟฟ้ากระแสตรง ชุดเกียร์และส่วนควบคุม โดยจะประกอบกันอยู่ภายในชุดเดียวกัน ตัวเซอร์โวมอเตอร์จะมีสายสัญญาณ 3 เส้น คือ สายใช้งาน 1 เส้น อีก 2 เส้นจะเป็นสายสำหรับจ่ายไฟให้เซอร์โวมอเตอร์และสายสำหรับต่อลงกราวด์ ในการควบคุมเซอร์โวมอเตอร์นั้นจะทำให้หมุนไปทางซ้ายได้ 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 ไม่ว่ากรณีใดๆทั้งสิ้น 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

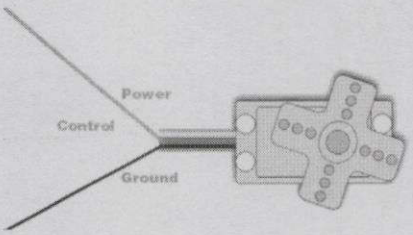
สายไฟของเซอร์โวจะมีอยู่สามเส้นซึ่งจะติดเป็นชุดเดียวกัน ซึ่งจะมีหน้าที่คือ

เส้นที่ 1 จ่ายไฟกระแส + DC ซึ่งแรงดันปรกติจะอยู่ที่ 5-6 โวลท์

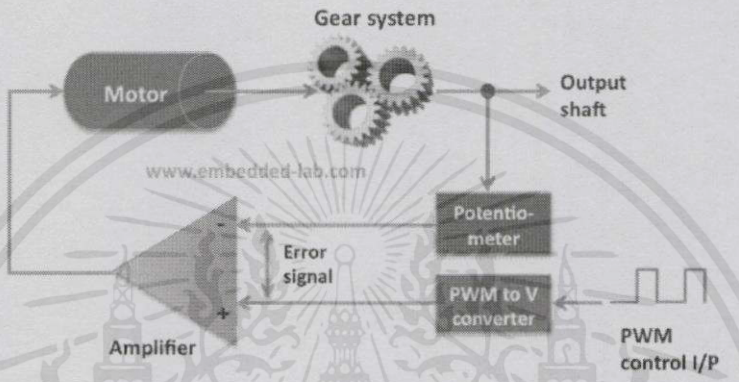
เส้นที่ 2 เป็นสาย Ground หรือเป็นขั้ว – DC

เส้นที่ 3 เป็นสายสัญญาณ โดยที่รีซีฟจะส่งสัญญาณลักษณะ on/off pulsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 Servo wire

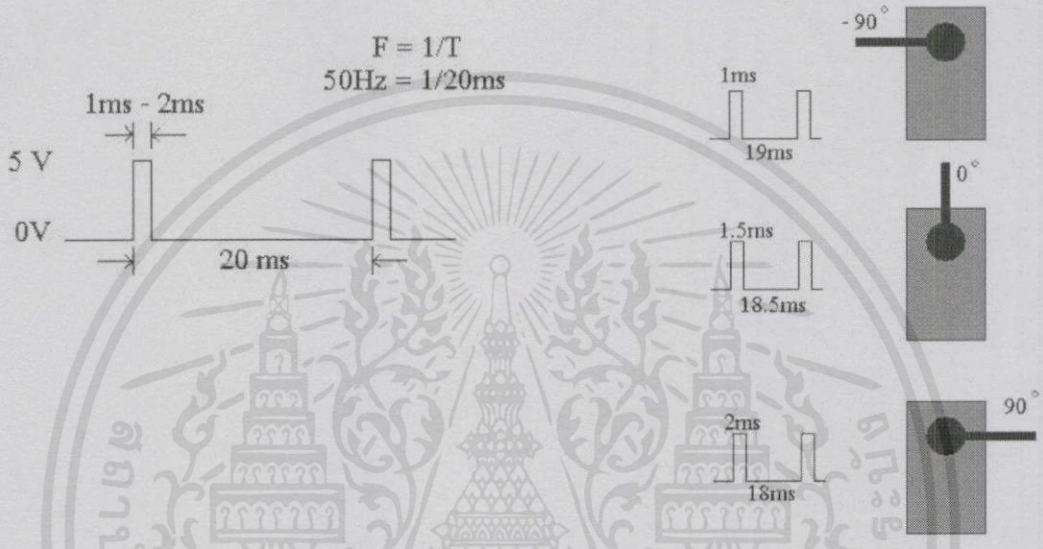


รูปที่ 2.14 Diagram การทำงานของ Servo Motor

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 มีค่าเท่ากับสัญญาณที่ถอดรหัสมาจากรีซีฟแล้วก็จะไม่มีการส่งแรงดันไฟฟ้าไปยังมอเตอร์ซึ่งกระบวนการนี้เรียกว่าการอัพเดทสัญญาณโดยมีความเร็วที่ 50 ครั้งต่อหนึ่งวินาที เราจึงเห็นเป็นการเคลื่อนที่ของเซอร์โว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น 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 ที่ใช้ปฏิบัติภารกิจในสภาพแวดล้อมที่มีโครงสร้างชัดเจนมาก ในบริบทนี้ ความสามารถของหุ่นยนต์ฮิวแมนนอยด์อาจรวมถึง แต่ไม่จำกัดแค่สิ่งเหล่านี้ ดูแลรักษาตัวเอง (เติมพลังงานให้ตัวมันเอง) เรียนรู้อัตโนมัติ (เรียนรู้หรือได้มาซึ่งความสามารถใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

โดยไม่ต้องได้รับความช่วยเหลือจากภายนอก, 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

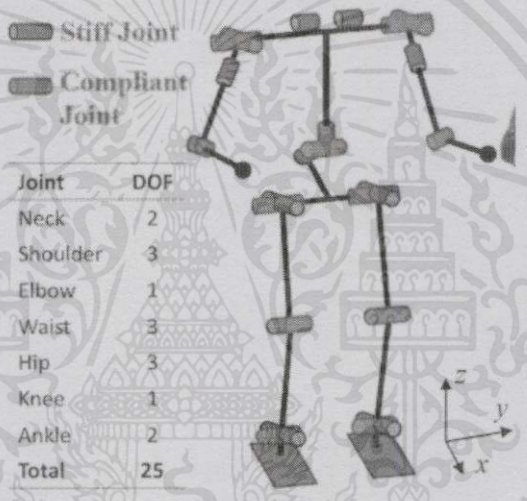
พื้นฐานในการเคลื่อนไหวของหุ่นยนต์ฮิวแมนนอยด์ 2 ขานั้น สามารถศึกษาได้โดยตรงจากการเคลื่อนไหวของหุ่นยนต์บนพื้นผิวที่เรียบ ไม่ขรุขระ ซึ่งหุ่นยนต์สามารถก้าวเดินได้ตามปกติ ลำดับต่อไปของการก้าวเดินคือการพัฒนาศักยภาพไปสู่การเคลื่อนที่ไปด้านหน้าของหุ่นยนต์อย่างอิสระ สามารถก้าวเดินบนพื้นผิวขรุขระ พื้นเอียงหรือพื้นที่ลาดชันรวมทั้งระดับของขั้นบันไดอย่างมีเสถียรภาพ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

1. เทคนิคขั้นพื้นฐานของการวางเท้าของหุ่นยนต์ โดยจะต้องคงสภาพแน่นอนไม่แปรเปลี่ยนไปตามลักษณะของพื้นผิว แม้หุ่นยนต์จะก้าวเดินในพื้นที่ขรุขระ
2. เทคนิคขั้นพื้นฐานของการปรับสภาพของหุ่นยนต์ หากในกรณีที่หุ่นยนต์ก้าวเดินแล้วมีการหกล้มหรือเสียการทรงตัว
3. เทคนิคขั้นพื้นฐานของการปรับสภาพของหุ่นยนต์ โดยให้ผลของการเคลื่อนที่ถูกต้องแม่นยำโดยการจดจำเป้าหมายที่ต้องการ ซึ่งเทคนิคเหล่านี้ทีมวิศวกรได้นำมาพัฒนาเป็นส่วนต่างๆ ก่อนจะนำมารวมกันในรูปแบบการเคลื่อนไหวของหุ่นยนต์ฮิวแมนนอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

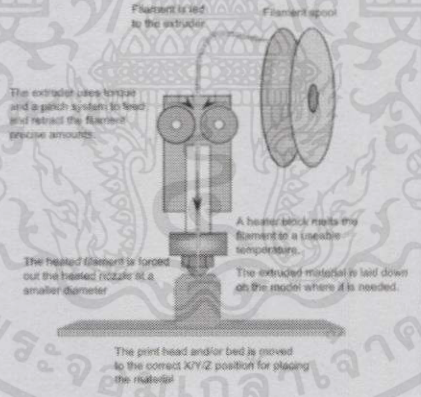
รูปที่ 2.17 Stereolithographic 3-D printer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

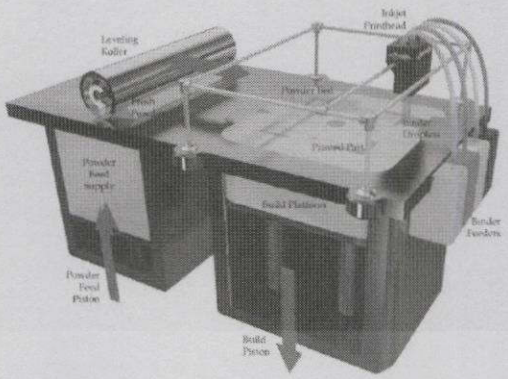
2.5.2 Fused Deposition Modeling (FDM) เป็นวิธีที่ใช้แพร่หลายที่สุด ใช้วัสดุหลักเป็นแท่งพลาสติกหรือโลหะที่พันเป็นม้วน ส่งผ่านไปยังหัวจ่ายซึ่งจะหลอมให้พลาสติกละลายก่อนนำมาพ่นลงบนแท่นวาง ซึ่งเมื่อพ่นออกมาแล้วจะแข็งตัวอย่างรวดเร็ว เครื่องจะพ่นทีละชั้นกระทั่งกลายเป็นโมเดล 3 มิติตามรูปทรงที่ต้องการ วัสดุที่ใช้ในการขึ้นรูปด้วยวิธีนี้ได้แก่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



รูปที่ 2.18 วิธีการแบบ Fused Deposition Modeling (FDM)

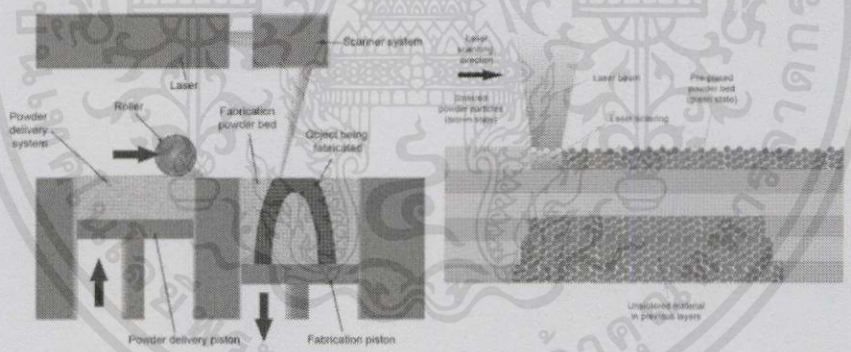
2.5.3 3D jet printer systems ใช้หลักการของ Ink Jet Printer แต่เป็นการ print ใน 3 มิติ โดยการพ่นวัสดุที่เป็นเทอร์โมพลาสติกเช่น โพลีเอสเทอร์ ผ่านหัวพ่นให้เป็นรูปแบบตามที่ต้องการ ใช้งานง่ายและสร้างแบบจำลองได้อย่างรวดเร็ว วิธีนี้มีจุดเด่นอยู่ที่สามารถเลือกสีต่างๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 เครื่อง 3D jet printer systems

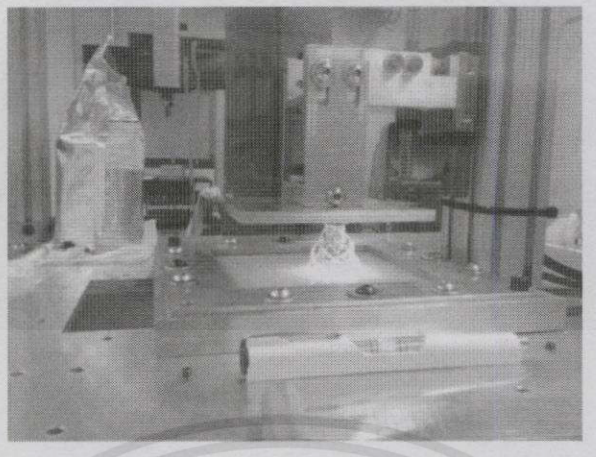
2.5.4 Selective Laser Sintering (SLS) คือ การเผาผนึกวัตถุดิบที่ใช้ด้วยแสงเลเซอร์ ใช้หลักการเดียวกันกับ SLA แต่ทำให้สามารถใช้วัสดุที่มีความแข็งแรงมากกว่า เช่น Thermoplastic,Nylon, Polyamide และ Polystyrene โดยวัตถุดิบที่ใช้จะอยู่ในรูปผง เช่น โลหะ พลาสติก เซรามิก และแก้ว เครื่องจะใช้เลเซอร์วาดรูปแบบขึ้นทีละชั้นและพ่นผงเหล่านี้ลงไปหลอมละลายกระทั่งกลายเป็นชิ้นงาน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



รูปที่ 2.20 Selective Laser Sintering (SLS)

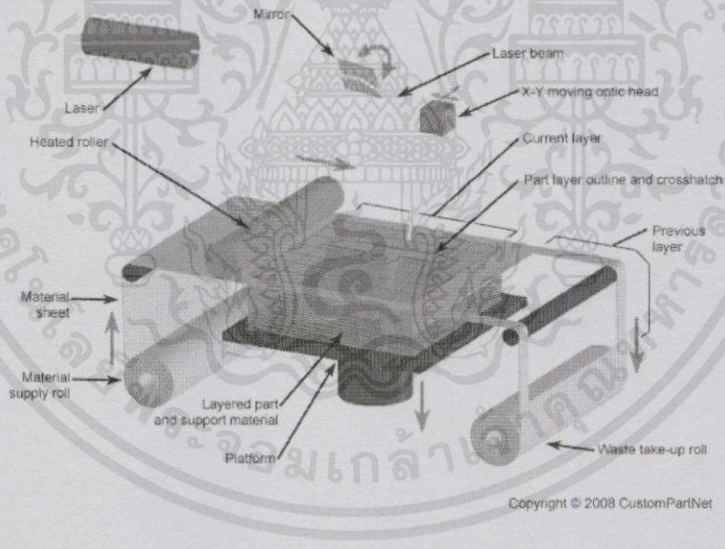
2.5.5 Digital Light Processing (DLP) ใช้หลักการเดียวกันกับ SLS แต่ใช้แหล่งกำเนิดแสงเป็นแสงสีขาวเป็นแหล่งกำเนิดแสงจากไฟแอลอีดีเพื่อให้ความร้อนแทนเลเซอร์ วัสดุเป็นยาง, เรซิ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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



รูปที่ 2.22 วิธีการแบบ Laminated Object Manufacturing (LOM)

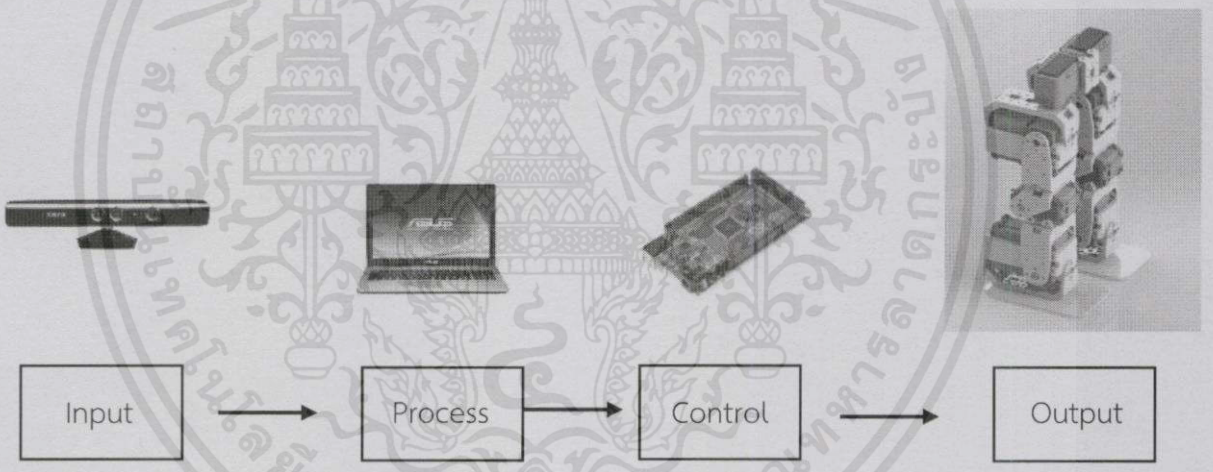
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ

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

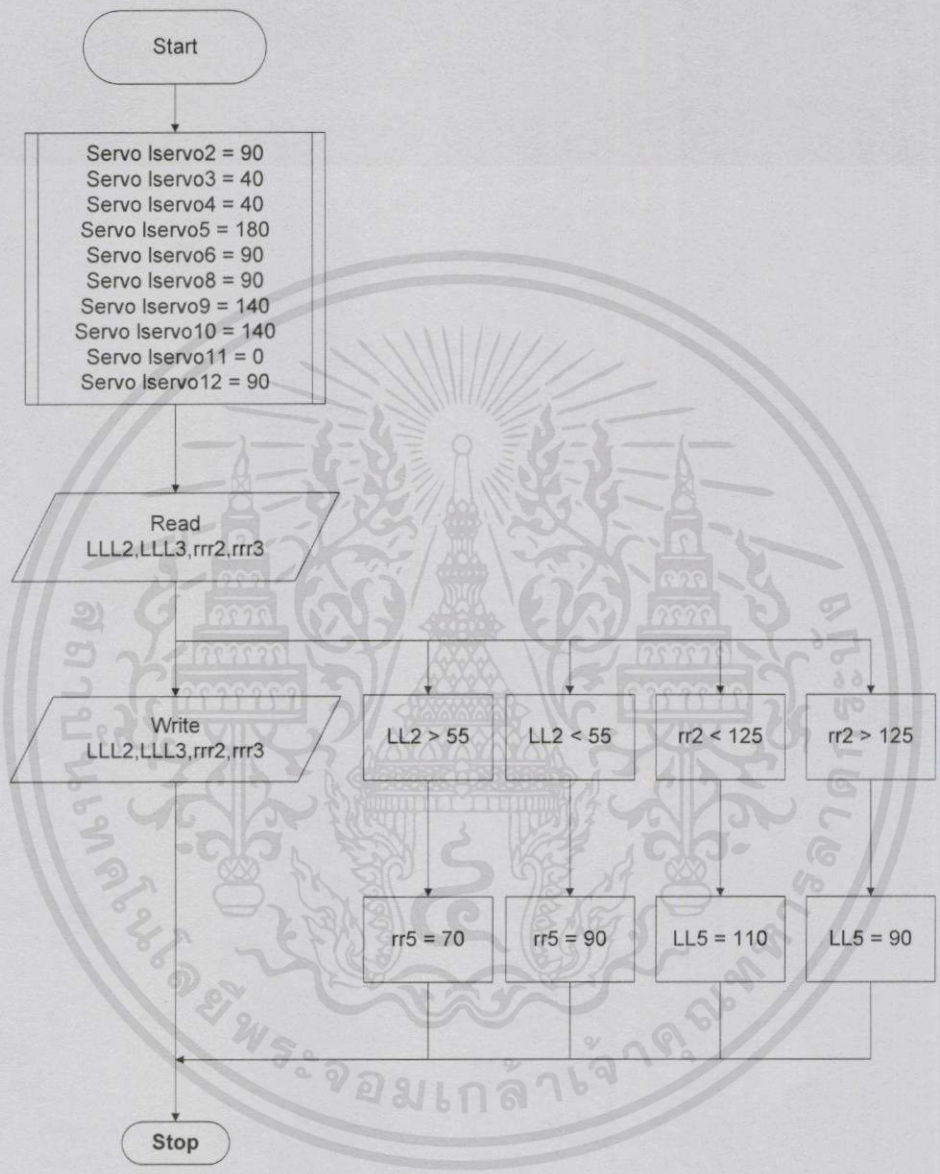


รูปที่ 3.1 ผังการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบโปรแกรม

3.2.1 ส่วนของ Arduino



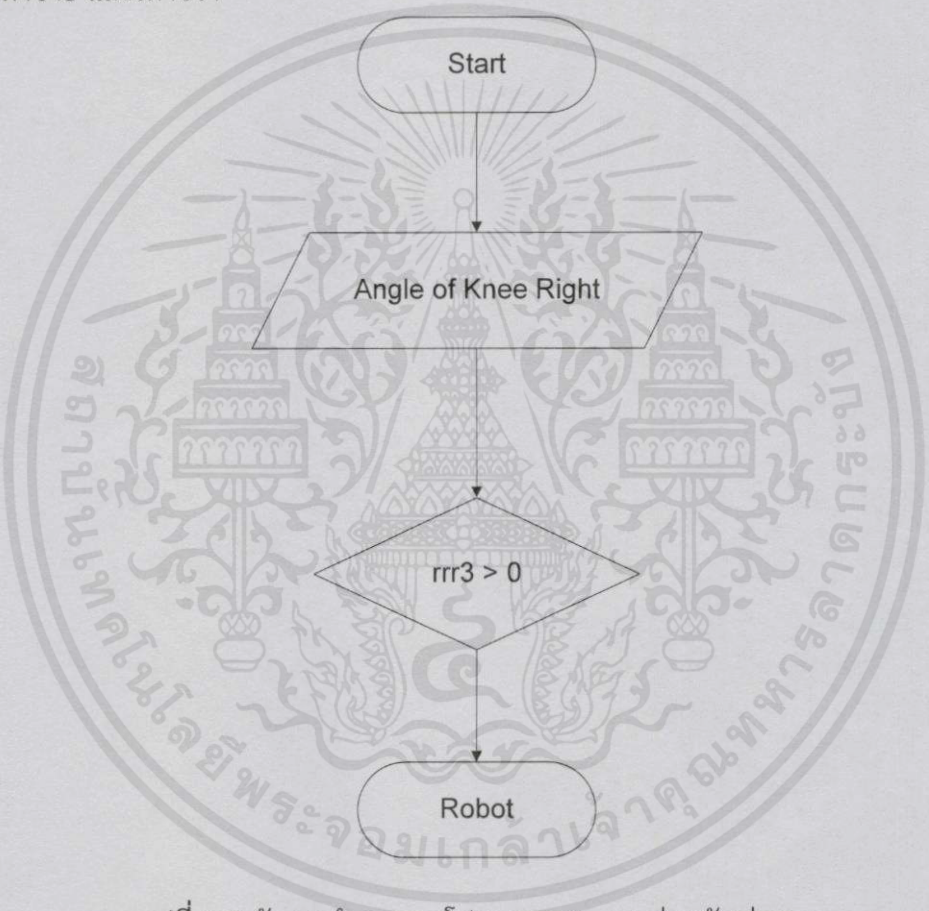
รูปที่ 3.2 ผังการทำงานของโปรแกรม Arduino

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)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ส่วนของ Kinect

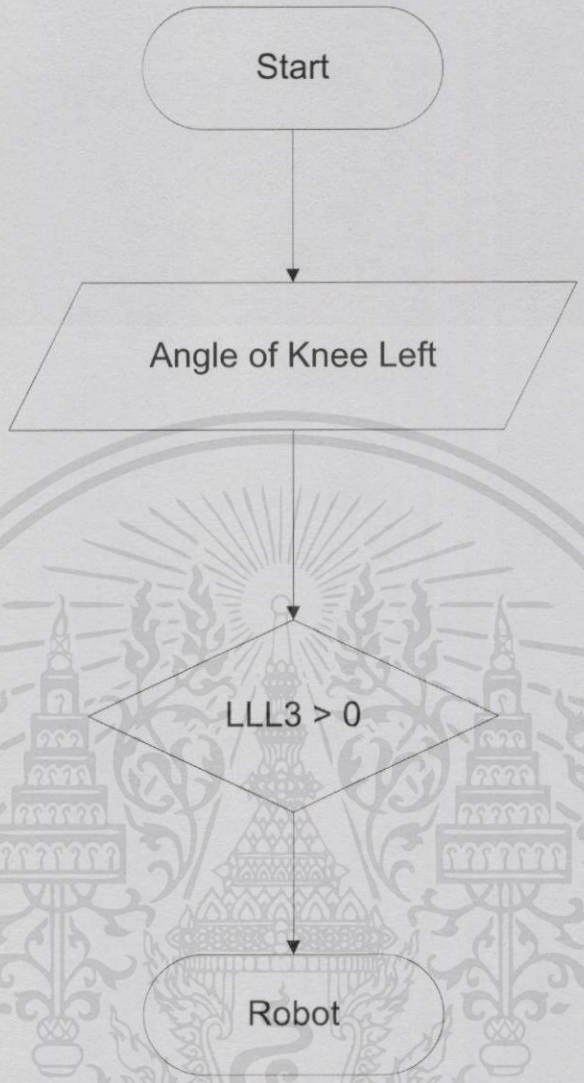
ผังการทำงานของโปรแกรม เมื่อเริ่มการทำงาน Kinect จะทำการจับผู้ใช้งาน โดยที่จะจับ เป็น Skelton Tacking 20 จุดแต่ในส่วนของโปรแกรมนี้เราจะ ใช้เพียง 6 จุดคือ หนึ่งส่วนของเอวซ้ายผู้ใช้ จุดที่สองเข่าซ้ายของผู้ใช้ จุดที่สามเท้าซ้ายของผู้ใช จุดที่สี่เอวขวาของผู้ใช้ จุดที่ห้าหัวเข่าขวาของผู้ใช้ จุดที่หกเท้าขวาของผู้ใช้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



รูปที่ 3.3 ผังการทำงานของโปรแกรม Kinect ส่วนหัวเข่าขวา

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

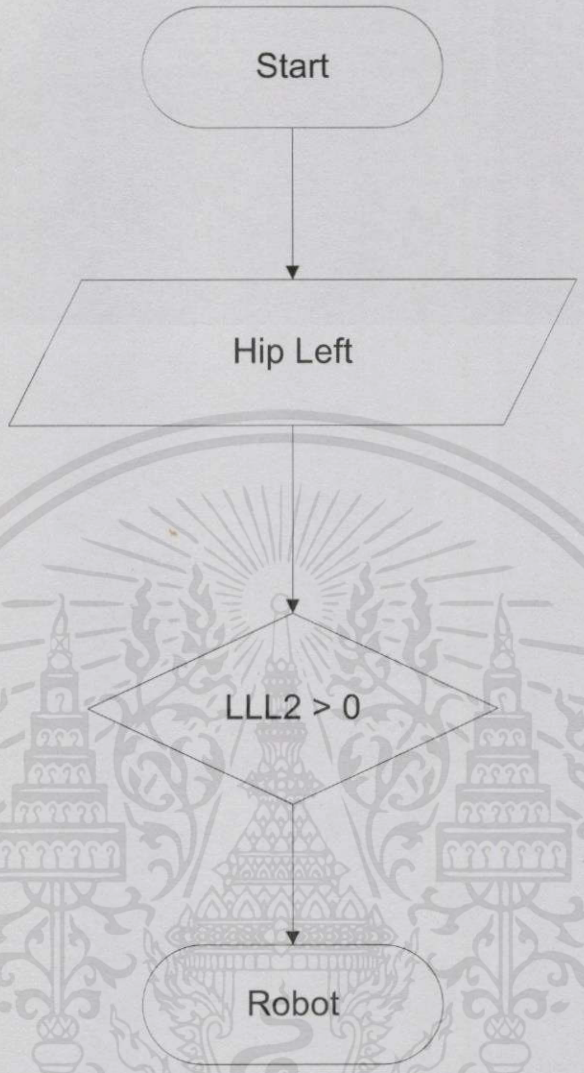
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ผังการทำงานของโปรแกรม Kinect ส่วนหัวเข่าซ้าย

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

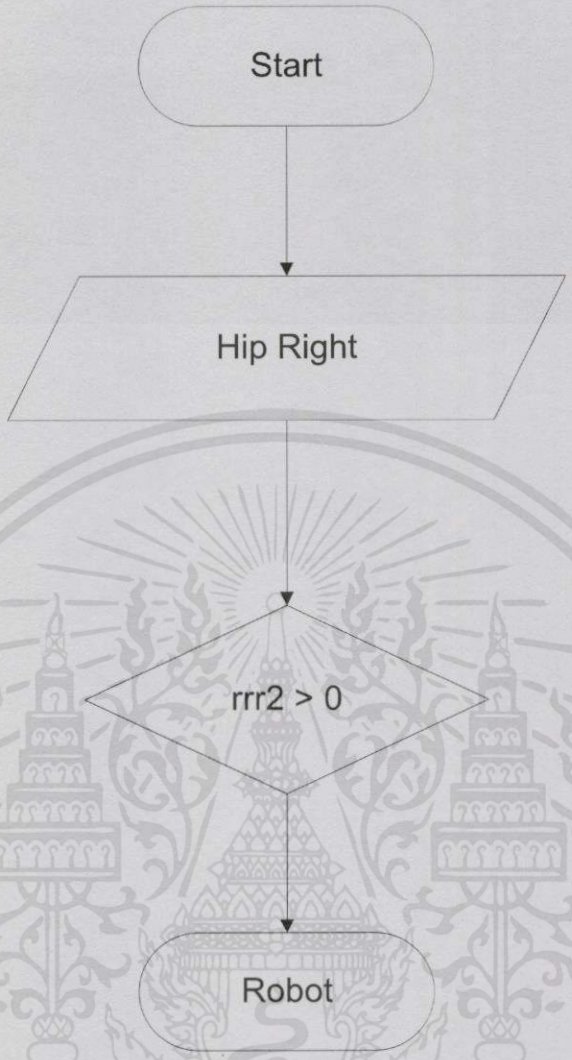
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ผังการทำงานของโปรแกรม Kinect ส่วนเอวด้านซ้าย

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

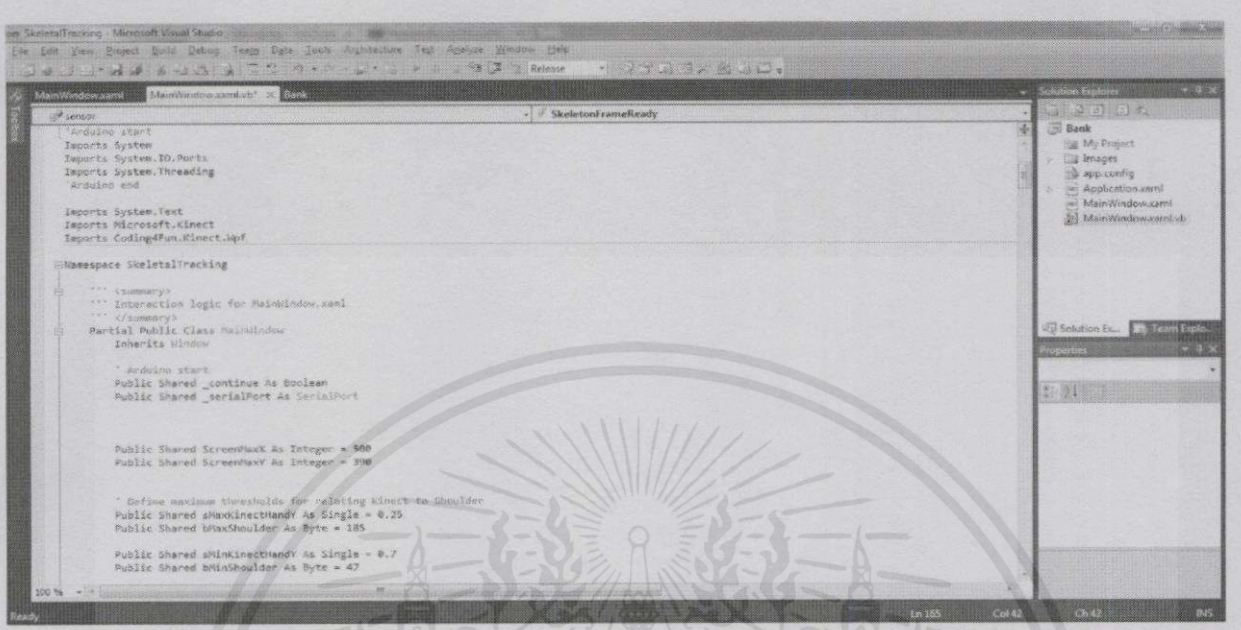


รูปที่ 3.6 ผังการทำงานของโปรแกรม Kinect ส่วนเอวด้านขวา

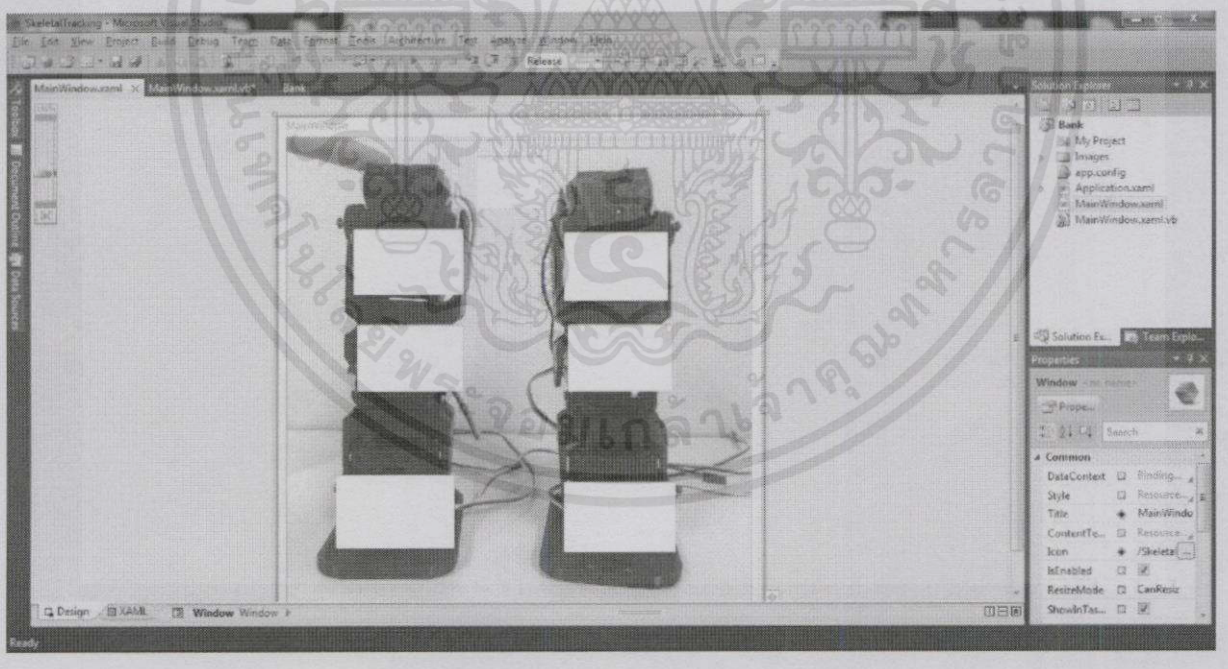
เริ่มต้นด้วยรับค่ามาจาก Kinect เป็นค่าองศาไปคำนวณในสมการของเอวด้านขวาที่ได้กำหนดไว้ เมื่อผ่านเงื่อนไขมากกว่า 0 เซอร์โวจะแสดงค่าตามที่ Kinect ได้รับมาและจะหยุดนิ่งเมื่อไม่ผ่านเงื่อนไข (เซอร์โวมีค่าเริ่มต้นที่กำหนดไว้ = 140 องศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของหน้าต่างโปรแกรม Microsoft Visual Basic



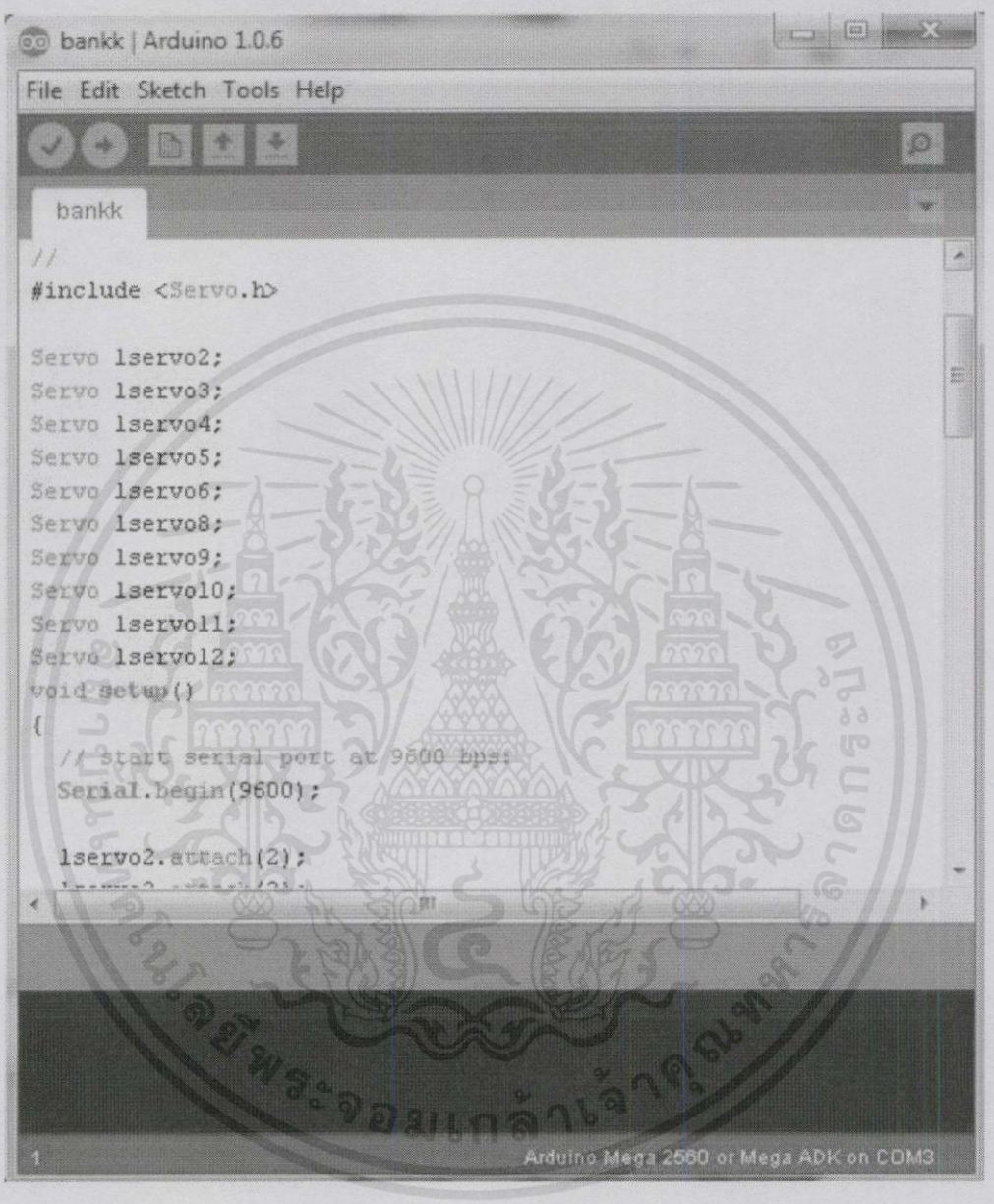
รูปที่ 3.7 ภาพแสดงหน้าต่างโปรแกรม Microsoft Visual Basic ส่วนโค้ดโปรแกรม



รูปที่ 3.8 ภาพแสดงหน้าต่างโปรแกรม Microsoft Visual Basic ส่วนหน้าต่างแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพของหน้าต่างโปรแกรม Arduino

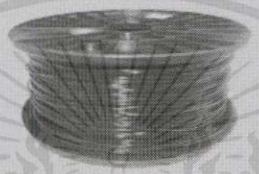


3.9 ภาพแสดงหน้าต่างโปรแกรม Arduino ส่วนโค้ดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

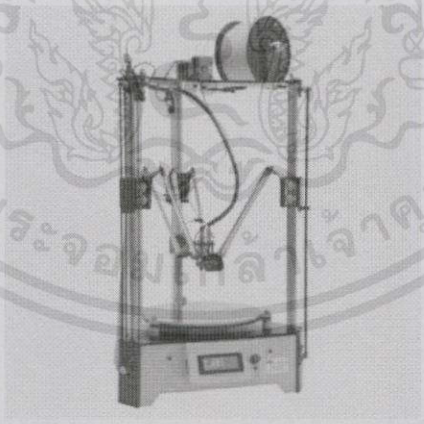
3.3 การออกแบบชิ้นงาน

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

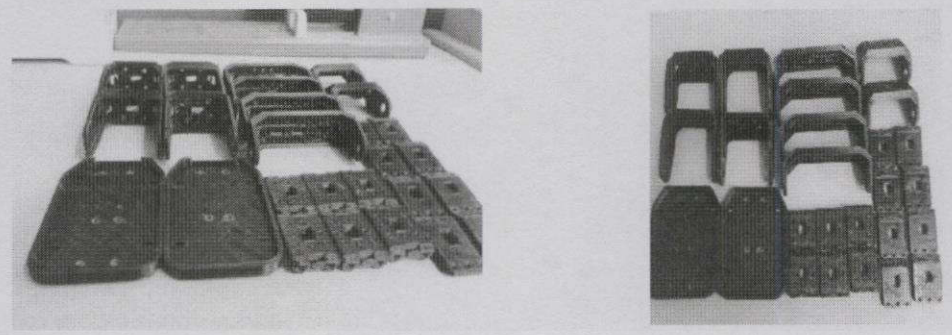
พลาสติก PLA เป็นพลาสติกชีวภาพ สามารถย่อยสลายได้และสามารถนำมาละลายหล่อขึ้นรูปต่างๆได้ด้วยหัวฉีดจากเครื่องปริ๊น 3D ที่มีอุณหภูมิสูงถึง 200 องศาเซลเซียส



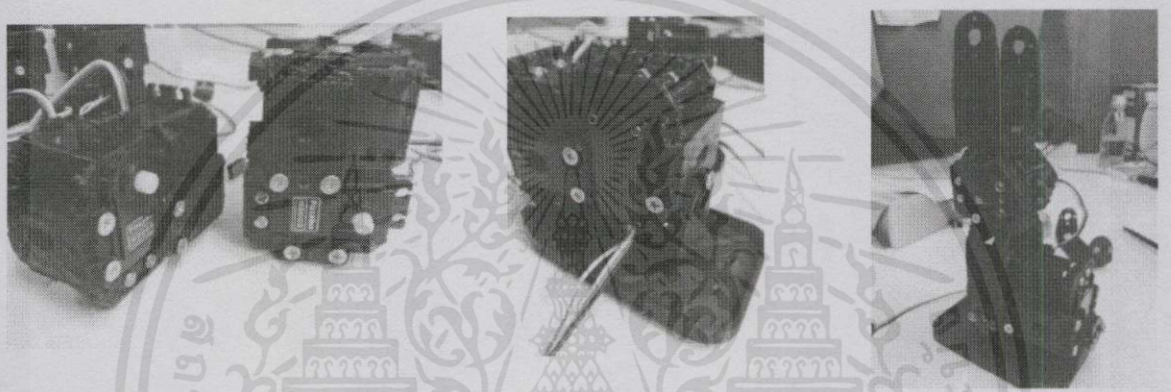
รูปที่ 3.11 เครื่องปริ๊น 3D

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ชิ้นส่วนของหุ่นยนต์ก่อนประกอบที่ทำมาจากเครื่องปริ้น 3D



รูปที่ 3.13 แสดงชิ้นส่วนของหุ่นยนต์ที่ทำมาจากเครื่องปริ้น 3D

รูปที่ 3.14 หุ่นยนต์ที่ประกอบเสร็จสมบูรณ์ที่ทำมาจากเครื่องปริ้น 3D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดสอบ

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

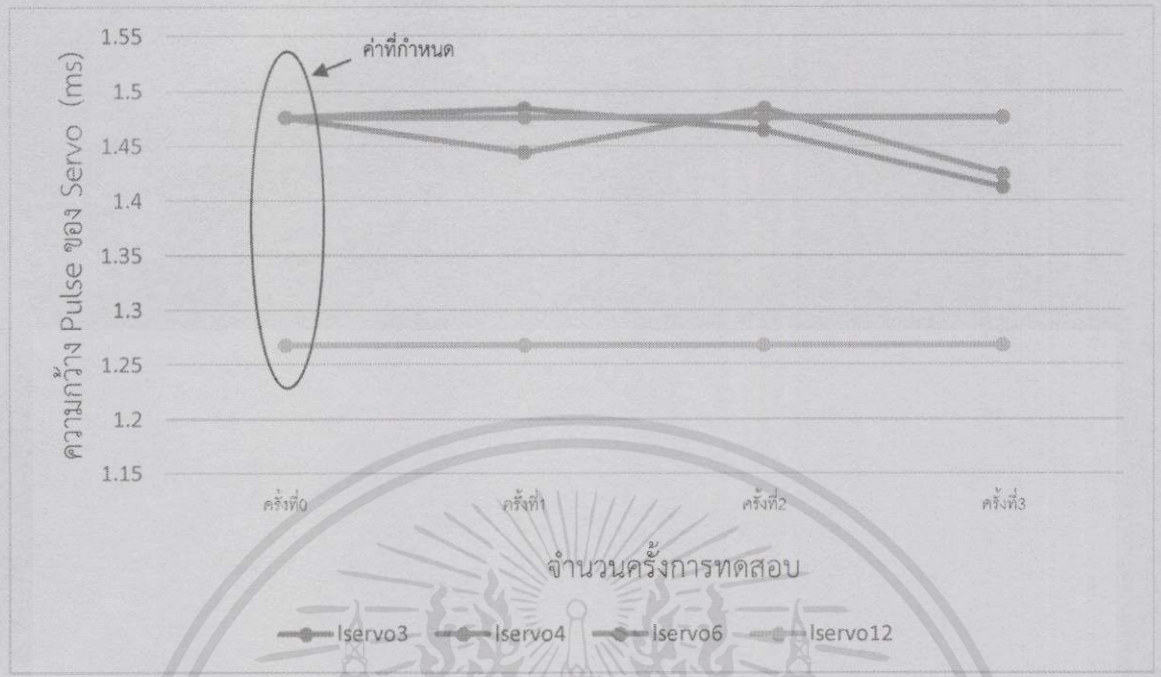
4.1 ผลการทดสอบท่าที่ 1 ท่ายกขาซ้ายขึ้น

รูปที่ 4.1 รูปแสดงการเคลื่อนไหวท่าที่ 1

ตารางที่ 4.1 แสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 1

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 ไม่ว่ากรณีใดๆทั้งสิ้น 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

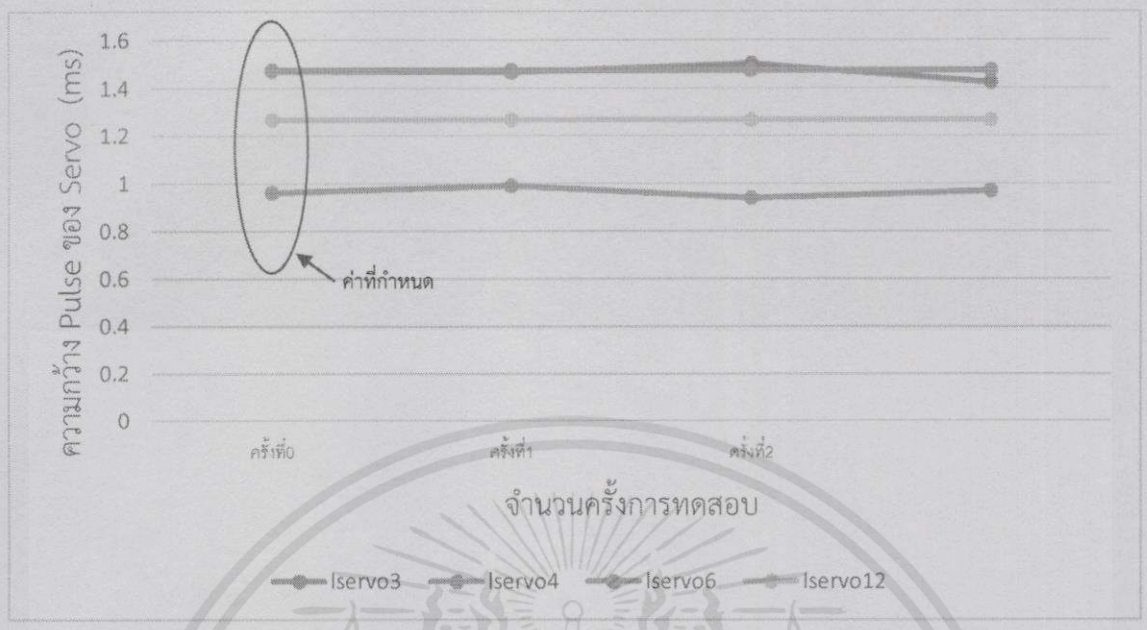
4.2 ผลการทดสอบท่าที่ 2 ท่าเตะขาซ้ายไปข้างหน้า

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 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

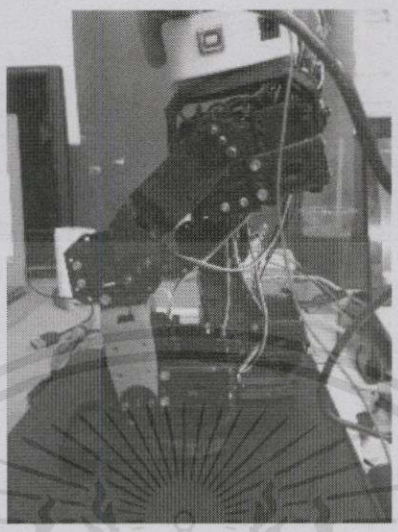


รูปที่ 4.4 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 2 จากตารางที่ 4.2

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบท่าที่ 3 ท่าขาซ้ายลง

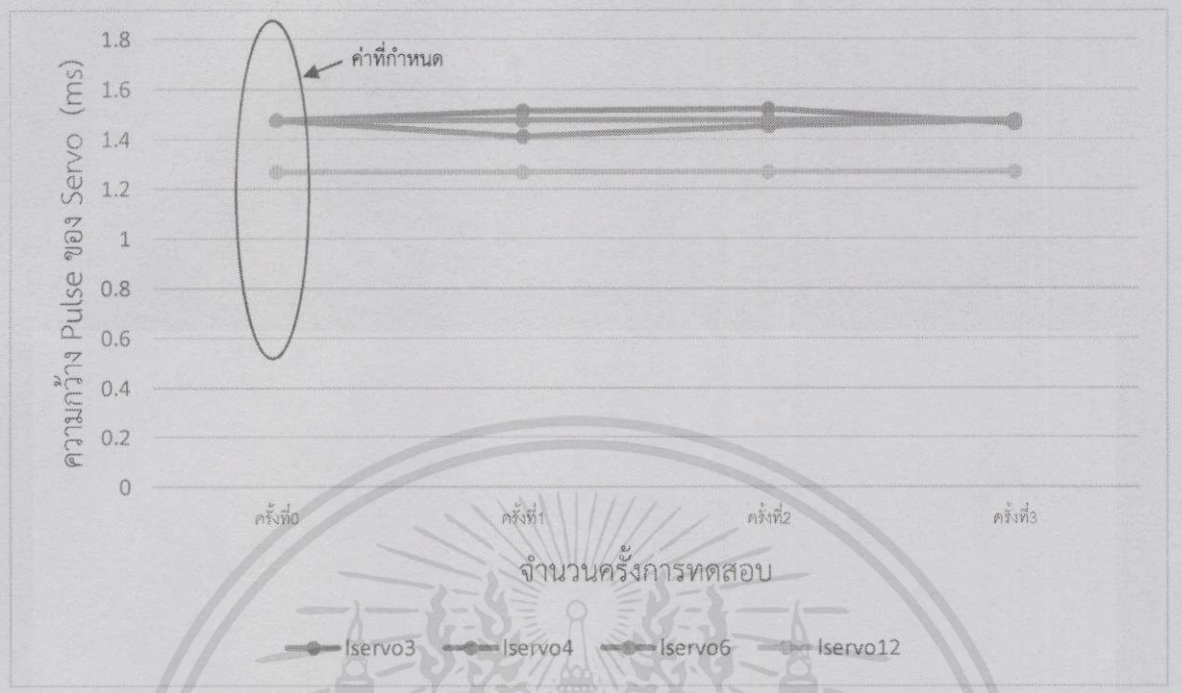


รูปที่ 4.5 รูปแสดงการเคลื่อนไหวท่าที่ 3

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 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

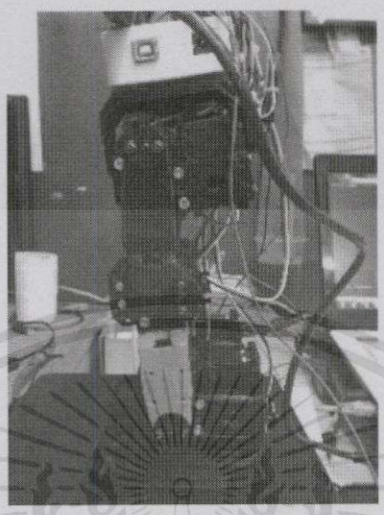


รูปที่ 4.6 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 3 จากตารางที่ 4.3

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดสอบท่าที่ 4 ท่าขาซ้ายอยู่ที่พื้น

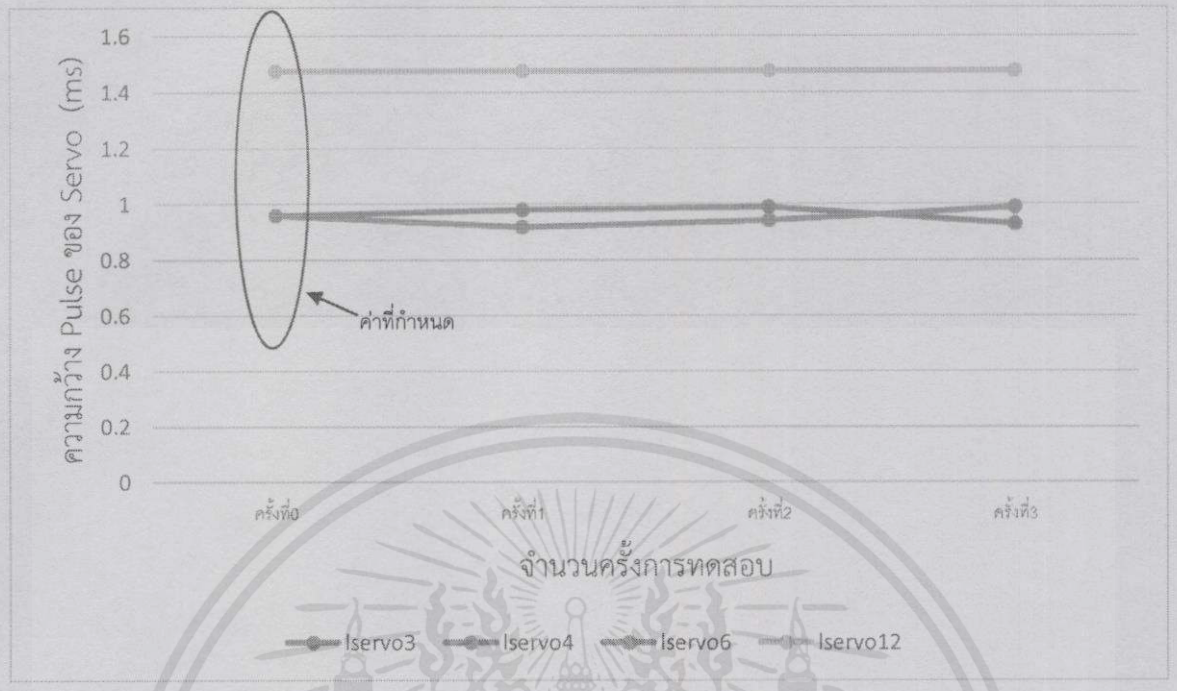


รูปที่ 4.7 รูปแสดงการเคลื่อนไหวท่าที่ 4

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 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

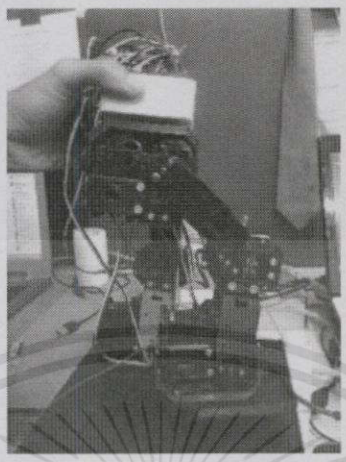


รูปที่ 4.8 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 4 จากตารางที่ 4.4

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ผลการทดสอบท่าที่ 5 ท่ายกขาขวาขึ้น

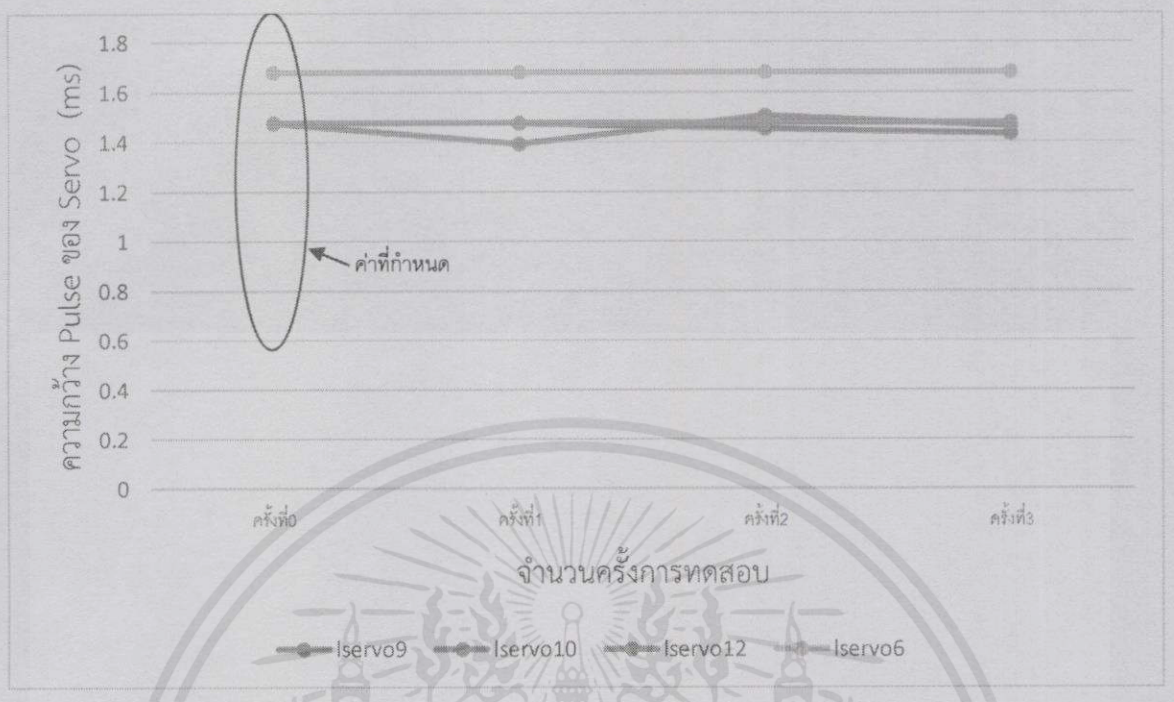


รูปที่ 4.9 รูปแสดงการเคลื่อนไหวท่าที่ 5

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 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

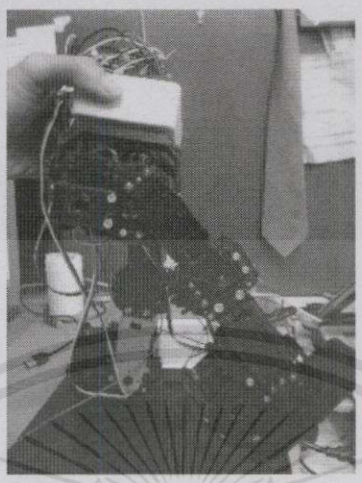


รูปที่ 4.10 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 5 จากตารางที่ 4.5

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลการทดสอบท่าที่ 6 ท่าเตะขาขวาไปข้างหน้า

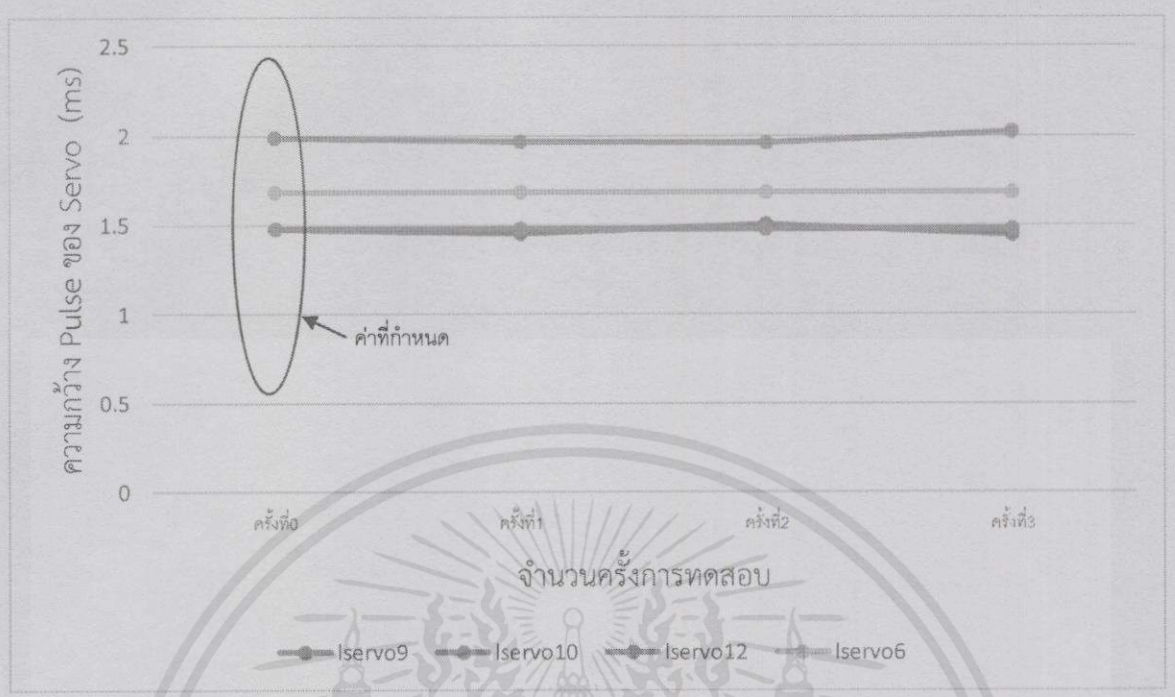


รูปที่ 4.11 รูปแสดงการเคลื่อนไหวท่าที่ 6

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 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

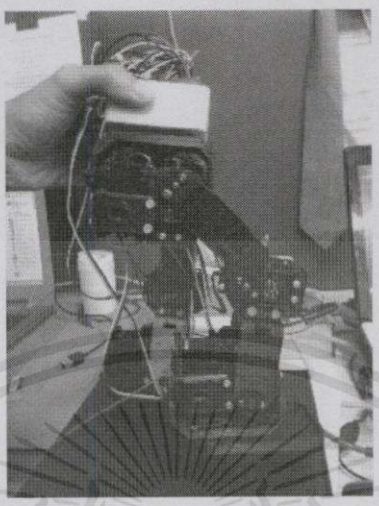


รูปที่ 4.12 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 6 จากตารางที่ 4.6

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ผลการทดสอบท่าที่ 7 ท่าขาขวาลง

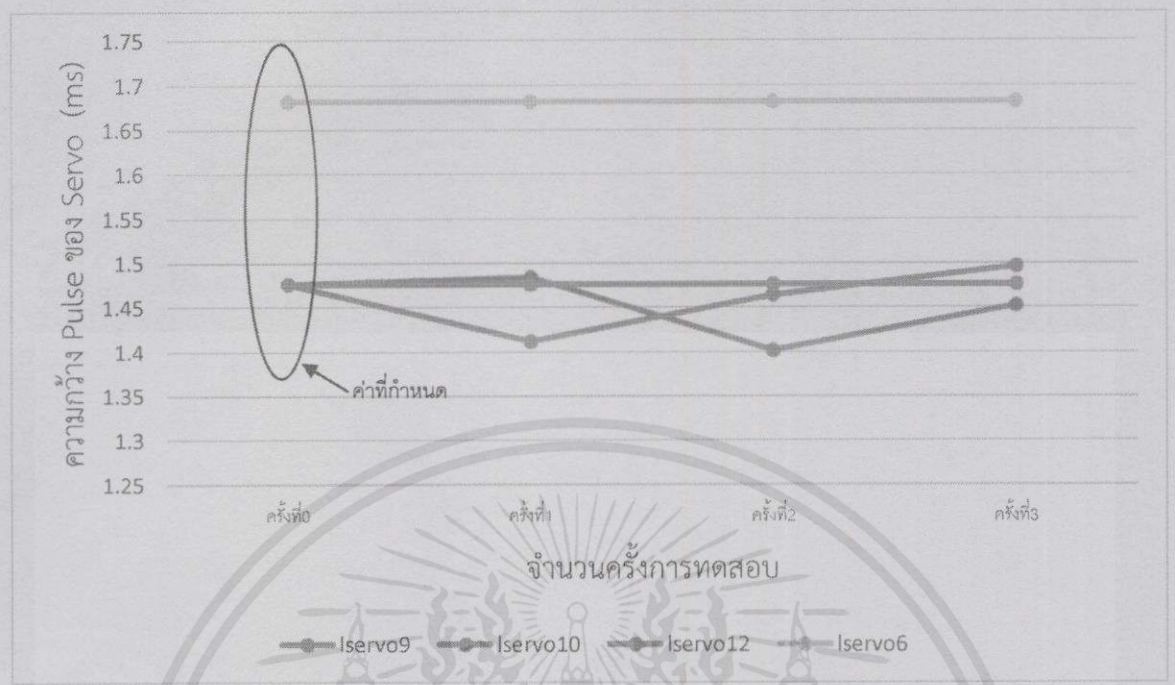


รูปที่ 4.13 รูปแสดงการเคลื่อนไหวท่าที่ 7

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 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

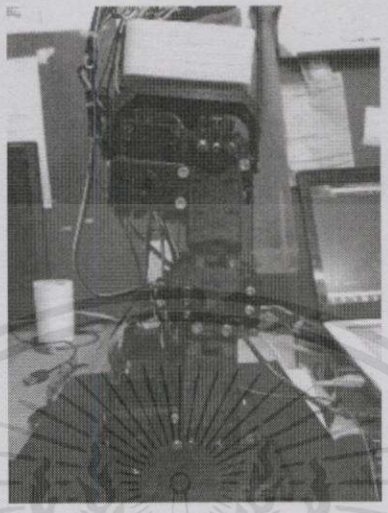


รูปที่ 4.14 กราฟแสดงความกว้าง Pulse ของ Servo ที่ใช้ในการเคลื่อนไหวท่าที่ 7 จากตารางที่ 4.7

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 ผลการทดสอบท่าที่ 8 ท่าขาขวาอยู่ที่พื้น

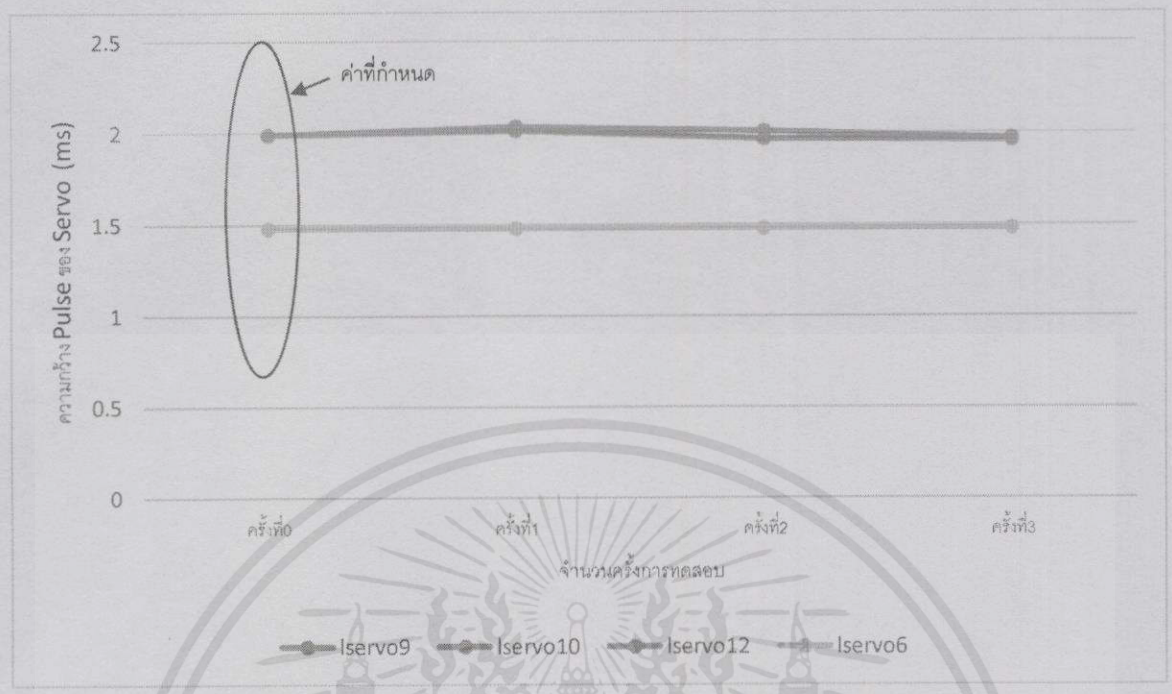


รูปที่ 4.15 รูปแสดงการเคลื่อนไหวท่าที่ 8

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 ไม่ว่ากรณีใดๆทั้งสิ้น 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 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการทดสอบการเคลื่อนที่ของหุ่นยนต์มีการเคลื่อนไหวตามการเคลื่อนไหวของผู้ทดสอบ โดยผู้ที่ทำการทดสอบจะต้องยืนอยู่ตรงตำแหน่งที่กล้องของ Kinect สามารถจับจุดต่างๆของเราได้อย่างชัดเจน เพราะระยะของตัวกล้องมีระยะจำกัดในการตรวจจับ ถ้าผู้ทดสอบไม่ได้อยู่ในตำแหน่งที่กล้องของ Kinect สามารถจับจุดต่างๆของเราได้ หุ่นยนต์ก็จะไม่มีการเคลื่อนไหว ซึ่งในการทดสอบหุ่นยนต์สามารถที่จะยกขาซ้ายและขาขวา ตามผู้ทดสอบได้ แต่ไม่สามารถที่จะก้าวเดินไปข้างหน้าได้ ทำให้หุ่นยนต์ย่ำอยู่กับที่

การทดสอบการเคลื่อนไหวของหุ่นยนต์ทั้ง 8 ท่า ขาข้างขวามีเปอร์เซ็นความคาดเคลื่อน 0.502% และขาข้างซ้ายมีเปอร์เซ็นความคลาดเคลื่อน 0.625% จากผลการทดสอบจะเห็นได้ว่าเป็นเซ็นความคาดเคลื่อนของขาทั้ง 2 ข้าง มีเปอร์เซ็นความคาดเคลื่อนน้อยกว่า 1 % ความคาดเคลื่อนที่เกิดขึ้น อาจเกิดจากการเลือกใช้ servo ที่มีคุณภาพต่ำ และการประกอบที่ไม่ประณีต

5.2 ปัญหาที่พบ

5.2.1 ระยะของตัวกล้อง Kinect ในการจับข้อต่อของแต่ละคนไม่เท่ากัน

5.2.2 ข้อจำกัดของKinect Sensor ไม่สามารถทำงานได้อย่างเต็มประสิทธิภาพในที่ที่มีแสงจ้ามากเกินไป เพราะอินฟาเรดจะกวนจนตรวจจับอะไรไม่ได้

5.2.3 ในการทดลองจะต้องทดลองในที่โล่งและระหว่างผู้ทดลองกับตัว Kinect ต้องไม่มีสิ่งกีดขวางใดๆ เพราะเมื่อมีสิ่งกีดกวางจะทำให้เกิดปัญหาในการส่งข้อมูล

5.2.4 เกิดค่าผิดพลาดขึ้นในส่วนโค้ดโปรแกรม Microsoft Visual Basic ซึ่งเกิดจากความไม่ชำนาญของผู้เขียน

5.3 ข้อเสนอแนะ

5.3.1 ควรหาระยะของกล้อง Kinect ในการจับข้อต่อของคนเทียบกับความสูงของคน

5.3.2 ควรหาพื้นหลังที่เป็นสีเขียวและไม่มีสิ่งกีดขวาง เพื่อให้กล้อง Kinect สามารถทำงานได้มีประสิทธิภาพมากขึ้น

5.3.3 ศึกษารายละเอียดเกี่ยวกับการก้าวเดินแบบอิสระ และการก้าวเดินอย่างมีเสถียรภาพอย่างละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] สัจจะ จรัสรุ่งรวีวร.2554. *คู่มือเรียนและใช้งาน Visual Basic 2010*. พิมพ์ครั้งที่ 1. นนทบุรี : ไอดีซีฯ.

[2] Gwenan Spearing. 2555. *Arduino and Kinect Projects*. New York : Paul Manning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code Part Visual Basic Studio 2010

```
'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 ไม่ว่ากรณีใดๆทั้งสิ้น 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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            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
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''Servo3 Right''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

                    Dim HipRx = CStr(playerSkeleton.Joints(JointType.HipRight).Position.X)

                    Dim HipRy = CStr(playerSkeleton.Joints(JointType.HipRight).Position.Y)

                    Dim HipRz = CStr(playerSkeleton.Joints(JointType.HipRight).Position.Z)

                    Dim KneeRx = CStr(playerSkeleton.Joints(JointType.KneeRight).Position.X)

                    Dim KneeRy = CStr(playerSkeleton.Joints(JointType.KneeRight).Position.Y)

                    Dim KneeRz = CStr(playerSkeleton.Joints(JointType.KneeRight).Position.Z)

                    Dim AnkleRx = CStr(playerSkeleton.Joints(JointType.AnkleRight).Position.X)

                    Dim AnkleRy = CStr(playerSkeleton.Joints(JointType.AnkleRight).Position.Y)

                    Dim AnkleRz = CStr(playerSkeleton.Joints(JointType.AnkleRight).Position.Z)

                    Dim ABVecterX = CStr((HipRx - KneeRx) * (AnkleRx - KneeRx)) + ((HipRy - KneeRy) *
(AnkleRy - KneeRy)) + ((HipRz - KneeRz) * (AnkleRz - KneeRz))

                    Dim ABVecter = CStr(((((HipRx - KneeRx) ^ 2) + ((HipRy - KneeRy) ^ 2) + ((HipRz -
KneeRz) ^ 2)) ^ 0.5) * ((((AnkleRx - KneeRx) ^ 2) + ((AnkleRy - KneeRy) ^ 2) + ((AnkleRz - KneeRz) ^ 2))
^ 0.5))

                    R3.Text = 305 - (339 - CInt((180 * Math.Acos((ABVecterX) / (ABVecter))) / 3.14159))

                    Dim rr3 = 305 - (339 - CInt((180 * Math.Acos((ABVecterX) / (ABVecter))) / 3.14159))

                    Dim rrr3 As Byte
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น รวมทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            If rrr3 > 0 Then

               rrr3 = 305 - (339 - CInt((180 * Math.Acos((ABVecterX) / (ABVecter))) / 3.14159))

            End If

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''Servo8 Left''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Dim HipLx = CStr(playerSkeleton.Joints(JointType.HipLeft).Position.X)

Dim HipLy = CStr(playerSkeleton.Joints(JointType.HipLeft).Position.Y)

Dim HipLz = CStr(playerSkeleton.Joints(JointType.HipLeft).Position.Z)

Dim KneeLx = CStr(playerSkeleton.Joints(JointType.KneeLeft).Position.X)

Dim KneeLy = CStr(playerSkeleton.Joints(JointType.KneeLeft).Position.Y)

Dim KneeLz = CStr(playerSkeleton.Joints(JointType.KneeLeft).Position.Z)

Dim AnkleLx = CStr(playerSkeleton.Joints(JointType.AnkleLeft).Position.X)

Dim AnkleLy = CStr(playerSkeleton.Joints(JointType.AnkleLeft).Position.Y)

Dim AnkleLz = CStr(playerSkeleton.Joints(JointType.AnkleLeft).Position.Z)

 Dim LABVecterX = CStr((HipLx - KneeLx) * (AnkleLx - KneeLx)) + ((HipLy - KneeLy) * (AnkleLy -
KneeLy)) + ((HipLz - KneeLz) * (AnkleLz - KneeLz))

 Dim LABVecter = CStr(((((HipLx - KneeLx) ^ 2) + ((HipLy - KneeLy) ^ 2) + ((HipLz - KneeLz) ^ 2)) ^ 0.5)
* ((((AnkleLx - KneeLx) ^ 2) + ((AnkleLy - KneeLy) ^ 2) + ((AnkleLz - KneeLz) ^ 2)) ^ 0.5))

L3.Text = 210 - CByte((180 * Math.Acos((LABVecterX) / (LABVecter))) / 3.14159)

Dim LL3 = 210 - CByte((180 * Math.Acos((LABVecterX) / (LABVecter))) / 3.14159)

Dim LLL3 As Byte

        If LL3 > 0 Then

               LLL3 = 210 - CByte((180 * Math.Acos((LABVecterX) / (LABVecter))) / 3.14159)

        End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''''''''''''''''''''Servo 2 Right'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Dim scaledLeftArm2 = playerSkeleton.Joints(JointType.KneeRight).ScaleTo(1, 300, 0.5F, -0.7F)

R2.Text = 288 - CByte(200 - (180 - scaledLeftArm2.Position.Y + 0))

Dim rr2 = 288 - CByte(200 - (180 - scaledLeftArm2.Position.Y + 0))

Dim rrr2 As Byte

            If rr2 > 0 Then

               rrr2 = 288 - CByte(200 - (180 - scaledLeftArm2.Position.Y + 0))

            End If

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''''''''''''''''''''Servo 7 Left''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Dim scaledLeftFoot2 = playerSkeleton.Joints(JointType.KneeLeft).ScaleTo(1, 300, 0.5F, -0.6F)

L2.Text = 38 + CByte(33 - (18 - scaledLeftFoot2.Position.Y + 50)) - 90

Dim LL2 = 38 + CByte(33 - (18 - scaledLeftFoot2.Position.Y + 50)) - 90

Dim LLL2 As Byte

            If LL2 > 0 Then

               LLL2 = 38 + CByte(33 - (18 - scaledLeftFoot2.Position.Y + 50)) - 90

            End If

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''''''''''''''''''''R5 Port Servo 6 Right''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Dim rr5 As Byte

            If LL2 > 55 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                rr5 = 70
            Else
                rr5 = 90
            End If
R5.Text = rr5
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''L5 Port Servo 12 Left''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Dim LL5 As Byte
        If rr2 < 125 Then
                LL5 = 110
        Else
                LL5 = 90
        End If
L5.Text = LL5
Me.components = New System.ComponentModel.Container()
Dim nn = CByte(playerSkeleton.Joints(JointType.ShoulderCenter).Position.Z)
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()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Me.Timer = New System.Windows.FontStyle()

MainWindow._serialPort.BaudRate = 9600

Dim test() As Byte = {LL2, LL3, LL5, rr2, rr3, rr5}

_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)

        'Cleanup

        sensor.Stop()

        sensor.Dispose()


        ' DisconnectFromRB()

    End Sub

    Private Sub DiscoverKinectSensor()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

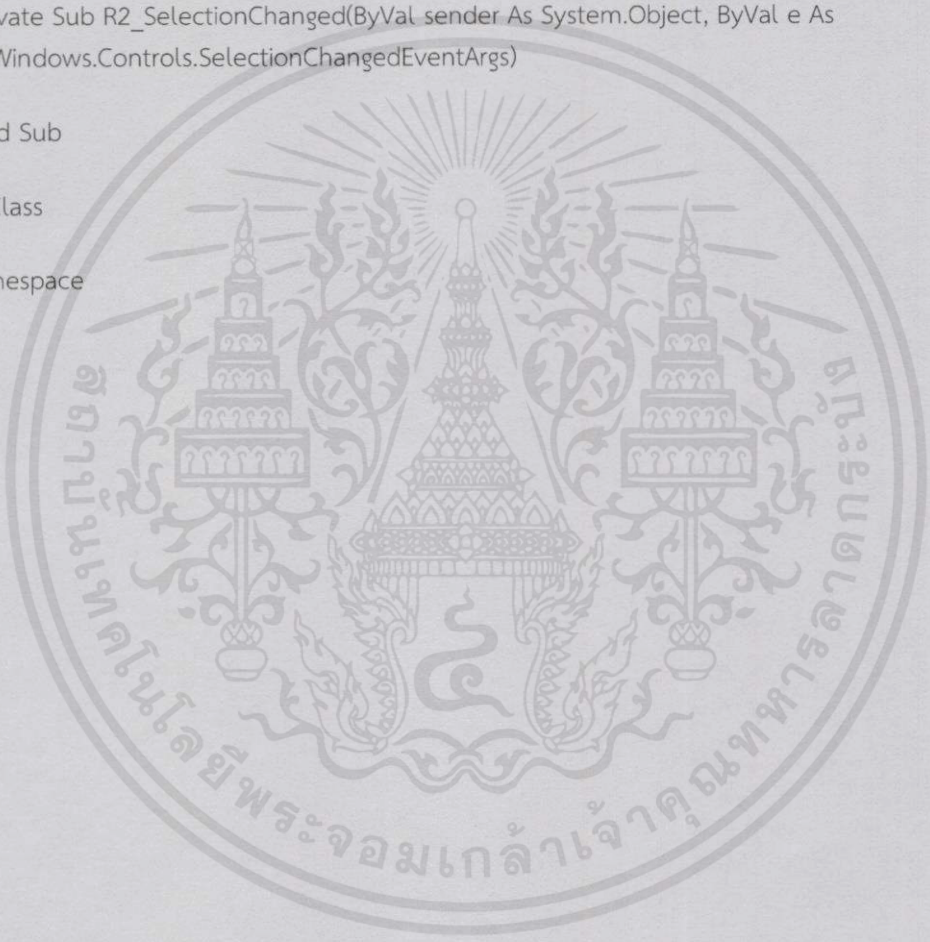
```
'Dim iterSensor As KinectSensor

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

' kinect 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()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        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)

'add event to receive skeleton data

'AddHandler sensor.SkeletonFrameReady, AddressOf nui_SkeletonFrameReady

sensor.Start()

    End Sub

        Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.Windows.Controls.TextChangedEventArgs)

    End Sub

        Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As
System.Windows.Controls.TextChangedEventArgs)

    End Sub

        Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As
System.Windows.Controls.TextChangedEventArgs)

    End Sub

        Private Sub TextBox15_TextChanged(ByVal sender As System.Object, ByVal e As
System.Windows.Controls.TextChangedEventArgs)


    End Sub

        Private Sub TextBox15_TextChanged_1(ByVal sender As System.Object, ByVal e As
System.Windows.Controls.TextChangedEventArgs)

    End Sub
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Private Sub s1_TextChanged(ByVal sender As System.Object, ByVal e As
System.Windows.Controls.TextChangedEventArgs)

        End Sub

        Private Sub Image1_ImageFailed(ByVal sender As System.Object, ByVal e As
System.Windows.ExceptionRoutedEventArgs)

        End Sub

        Private Sub R2_SelectionChanged(ByVal sender As System.Object, ByVal e As
System.Windows.Controls.SelectionChangedEventArgs)

        End Sub

    End Class

End Namespace
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code Part Arduino

```
#include <Servo.h>
Servo lservo2;
Servo lservo3;
Servo lservo4;
Servo lservo5;
Servo lservo6;
Servo lservo8;
Servo lservo9;
Servo lservo10;
Servo lservo11;
Servo lservo12;
void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  lservo2.attach(2);
  lservo3.attach(3);
  lservo4.attach(4);
  lservo5.attach(5);
  lservo6.attach(6);
  lservo8.attach(8);
  lservo9.attach(9);
  lservo10.attach(10);
  lservo11.attach(11);
  lservo12.attach(12);
  lservo2.write(90);
  lservo3.write(40);        ////40+130
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  lservo4.write(40);        ////40+130
  lservo5.write(180);       /////180
  lservo6.write(90);        ////90
  lservo8.write(90);
  lservo9.write(140);       ////140-50
  lservo10.write(140);      ////140-50
  lservo11.write(0);       ////0
  lservo12.write(90);       ////90

}
unsigned char LLL2,LLL3,LL5,rrr2,rrr3,rr5 ;
int val = 0;
void loop()
{
// read the 4-byte byffer sent from VB containing coordinates and joint ID.
  if (Serial.available() >= 6) {
    LLL2 = Serial.read();
    LLL3 = Serial.read();
    //LL4 = Serial.read();
   LL5 = Serial.read();
    rrr2 = Serial.read();
    rrr3 = Serial.read();
  // rr4 = Serial.read();
   rr5 = Serial.read();
   //x = Serial.read();
   // y = Serial.read();
   // lservo2.write();
    lservo3.write(LLL2);
```

เอกสารนี้มีไว้เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    lservo4.write(LLL3);
   // lservo5.write(LL4);
  lservo6.write(LL5);
    //lservo8.write();
    lservo9.write(rrr2);
    lservo10.write(rrr3);
   // lservo11.write(rr4);
  lservo12.write(rr5);
      delay(10);
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้