หุ่นยนต์ควบคุมด้วยการเคลื่อนไหวของมือ
ROBOT CONTROL BY HAND MOVEMENT

จิรัฏฐ์ เมธาสถิตย์สุข
JIRHAT MEATHASTIDSOOK
ศักดาวุธ มณีกัญญ์
SAKDAWUT MANEEKAN

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2556

หุ่นยนต์ควบคุมด้วยการเคลื่อนไหวของมือ

ROBOT CONTROL BY HAND MOVEMENT

โดย

จิรัฏฐ์ เมธาสถิตย์สุข

ศักดาวุธ มณีกัญญ์

อาจารย์ที่ปรึกษา

ดร. แสงระวี บัวแก้ว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2556

ROBOT CONTROL BY HAND MOVEMENT

JIRHAT MEATHASTIDSOOK

SAKDAWUT MANEEKAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRONICS ENGINEERING
SCHOOL OF GRAUATE STUDIES
KING MONGKUT'S INSTITUT OF TECHNOLOGY LADKRABANG
2013

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ควบคุมด้วยการเคลื่อนไหวของมือ

ROBOT CONTROL BY HAND MOVEMENT

ผู้จัดทำ นายจิรัฏฐ์ เมธาสถิตย์สุข รหัสประจำตัว 53010326

นายศักดาวุธ มณีกัญญ์ รหัสประจำตัว 53011552

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

( )

อาจารย์ที่ปรึกษา

หัวข้อปริญญานิพนธ์	หุ่นยนต์ควบคุมด้วยการเคลื่อนไหวของมือ	
นักศึกษา	นาย จิรัฏฐ์ เมธาสถิตย์สุข	รหัสประจำตัว 53010326
	นาย ศักดาวุธ มณีกัญญ์	รหัสประจำตัว 53011552
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์	
ปีการศึกษา	2556	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ดร. แสงระวี บัวแก้ว	

บทคัดย่อ

รายงานฉบับนี้อธิบายการออกแบบและการสร้างหุ่นยนต์สี่ขา(Quadro Pod) และชุดควบคุมหุ่นยนต์โดยตรวจจับการเคลื่อนไหวของมือและนิ้วมือ โดยมีหลักการทำงานคือแบ่งเป็น 2 ส่วน ส่วนที่ 1 ส่วนควบคุมเป็นถุงมือที่มีเซ็นเซอร์วัดความงอ(Flex sensor) ติดตามนิ้วทั้ง 5 เพื่อตรวจจับการเคลื่อนไหวเวลาที่งอและคลายนิ้วซึ่งได้เป็นค่าความต้านทานและเซ็นเซอร์วัดความเร่ง(Acclerometers) ที่จะตรวจจับการเปลี่ยนแปลงตำแหน่งของฝ่ามือในแนวแกน 3 มิติ ซึ่งได้ค่าเอาท์พุทเป็นแรงดัน ค่าทั้ง 2 ส่วนปฏิบัติการ คือตัวหุ่นยนต์ที่จะทำการประมวลผลข้อมูลจากส่วนควบคุมที่ส่งผ่านมาทาง Wireless และนำมาปฏิบัติตามโปรแกรมที่ตั้งไว้โดยตัวหุ่นนั้นจะมี Wireless Module รับข้อมูลที่ส่วนควมคุมส่งมาแล้วส่งต่อเข้าบอร์ด Arduino เพื่อทำการประมวลผล เมื่อประมวลผลเสร็จบอร์ด Arduino จะทำการสั่งงาน Servo motor เพื่อให้ตัวหุ่นยนต์เคลื่อนไหวตามที่ออกแบบไว้

Thesis Title	Robot Control by Hand Movement	
Student	Mr.Jirhat Maethastidsook	Student ID 53010326
	Mr.Sakdawut Maneekan	Student ID 53011552
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2556	
Thesis Advisor	Dr.Seangrawee Buakeaw	

Abstract

This project presents the design and construction four legged robot and control the robot detect movement of the hand and fingers, There are two principle parts. Frist Control part glove with flex sensor track all five finger to detect the movement. This will be the resistance and accelerometers to detect changes in the position of hand in axial 3D will be has output are voltage value the two values was sent to the input of the arduino board then convert data and sent by wireless module to the robot. Second Operation part robot is operate procressing data from control part then transmitted via wireless and compliance program setting. Robot has wireless module for get information sent by control part and then sent to arduino board for operate procressing. When processing is complete arduino board will command servo motor for make the robot move as designed.

กิตติกรรมประกาศ

การทำโครงงานหุ่นยนต์ควบคุมด้วยการเคลื่อนไหวของมือนี้ ได้รับการสนับสนุนและช่วยเหลือให้คำแนะนำจาก อาจารย์ภาควิชาอิเล็กทรอนิกส์ และได้รับความอนุเคราะห์ในด้านอุปกรณ์และเครื่องมือๆจากภาควิชาอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่ง ดร.แสงระวี บัวแก้ว ที่ให้แนวคิดและคำปรึกษา การแก้ไขปัญหาและการสนับสนุนในการทำวิจัย รวมไปถึงบิดาแลมารดาของผู้ทำโครงงาน ที่ให้การสนับสนุนในด้านทุนทรัพย์เพื่อทำการจัดหาอุปกรณ์บางส่วน ซึ่งทางคณะผู้จัดทำรู้สึกซาบซึ้งในความกรุณาของทุกท่านอย่างที่สุดและขอกราบขอบพระคุณเป็นอย่างสูง

จิรัฏฐ์ เมธาสถิตย์สุข
ศักดาวุธ มณีกัญญ์

สารบัญ

หน้า

สารบัญ(ต่อ)

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

หุ่นยนต์สี่ขา(quadropod robot) เป็นหุ่นยนต์ที่เคลื่อนที่โดยใช้ขา 4 ข้าง ลักษณะคล้ายแมงมุม ขาแต่ล่ะข้างเคลื่อนไหวโดยใช้ servo motor เป็นตัวขยับข้อต่อ ตามอัลกอริทึมเพื่อให้ตัวหุ่นสามารถเคลื่อนที่ได้ โดยส่วนควบคุมการเคลื่อนที่คือบอร์ด arduino ที่รับข้อมูลการสั่งการจากคอนโทรลเลอร์และประมวลผล ติดต่อกับคอนโทรลเลอร์โดยใช้ wireless module หุ่นยนต์สี่ขาสามารถประยุกต์ใช้งานได้หลายแบบ จุดเด่นคือสามารถเข้าไปในพื้นที่ที่ไม่ต้องการเข้าไป หรือสำรวจพื้นที่ก่อนให้มนุษย์เข้า ส่วนของคอนโทรลเลอร์ตรวจจับการเคลื่อนไหวของมือเซ็นเซอร์ที่ได้ใช้สามารถตรวจจับได้ทั้งการพลิกของฝ่ามือ และการงอของนิ้ว ทำให้คนควบคุมสามารถสั่งงานได้หลากหลาย

การประยุกต์ใช้หุ่นยนต์สี่ขา เช่นในการสำรวจพื้นที่ที่เป็นอันตรายต่อมนุษย์ ติดกล้องเพื่อสั่งหุ่นยนต์ไปบันทึกภาพในตำแหน่งที่ต้องการ นำตัวคอนโทรลเลอร์ไปควบคุมอุปกรณ์อย่างอื่นนอกจากหุ่นยนต์ได้ เช่น เครื่องบินบังคับ โทรทัศน์ หรือเปิดปิดไฟในอาคาร และตัวหุ่นยังสามารถพัฒนาเรื่องขนาด และอุปกรณ์บนตัวหุ่นได้อีกมากมายเพื่อให้สามารถทำงานที่ต้องการได้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

วิทยานิพนธ์นี้มีวัตถุประสงค์ดังต่อไปนี้

1.2.1 ศึกษาเกี่ยวกับบอร์ด Arduino, Senser Flexและ Accelerometer เพื่อให้สามารถนำไปใช้ประยุกต์ใช้ในงานต่างๆได้

1.2.2 ศึกษาคำสั่งที่ใช้ในการควบคุม arduino เขียนคำสั่งรวบรวมข้อมูลจากเซ็นเซอร์ต่างๆ และรับส่งข้อมูลผ่านอุปกรณ์ไร้สาย(wireless module)ได้

1.2.3 เพื่อศึกษาการสร้างและควบคุมหุ่นยนต์ระยะไกล

1.3 ขอบเขตในปริญญานิพนธ์

ศึกษาการทำงานของ Arduino ประยุกต์ให้สามารถทำงานร่วมกับเซ็นเซอร์และอุปกรณ์แบบต่างๆ ได้ นำมาพัฒนาเพื่อให้สามารถควบคุมหุ่นยนต์ระยะไกลได้ โดยใช้การขยับของมือ

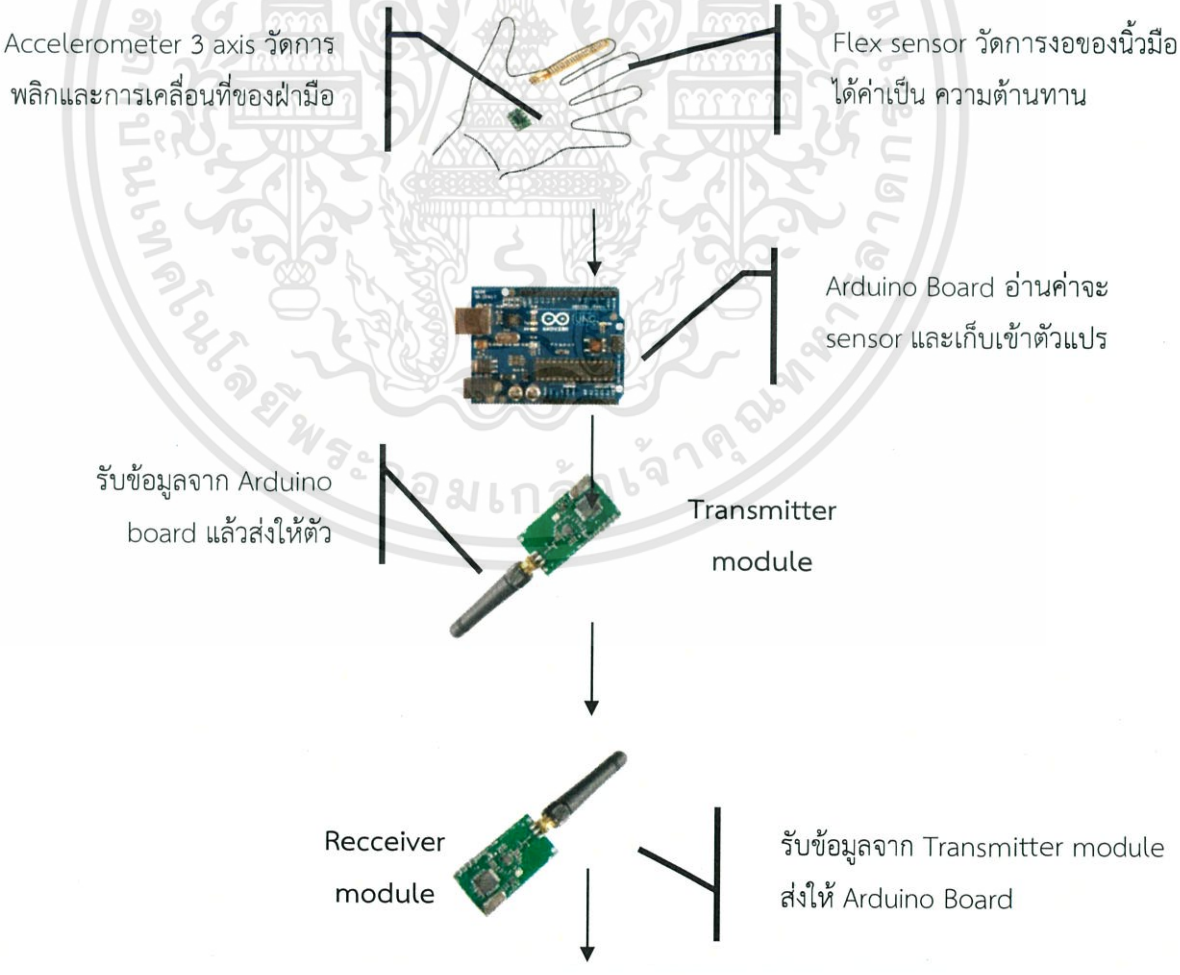
1.4 ประโยชน์ที่จะได้รับ

เป็นแนวทางการพัฒนาและออกแบบอุปกรณ์ควบคุมหุ่นยนต์ระยะไกลที่สามารถใช้งานได้จริง เป็นอีกรูปแบบของการควบหุ่นยนต์หรืออุปกรณ์เอาท์พุทแบบต่างๆ ตัวหุ่นสามารถที่ไปยังจุดต่างๆได้จึงทำให้สามารถเพิ่มเติมอุปกรณ์อย่างอื่นใส่ตัวหุ่นเพื่อให้สามารถทำงานที่ต้องการได้ โดยไม่จำเป็นต้องให้มนุษย์เข้าไปจุดที่ต้องการได้

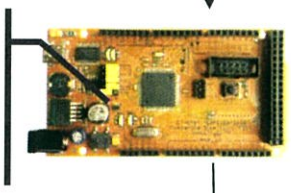
บทที่ 2
ทฤษฎีและหลักการ

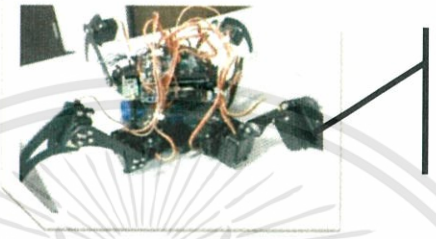
ปัจจุบันเป็นยุคดิจิตอลที่เทคโนโลยีมีเข้ามามีความจำเป็นอย่างมากในชีวิตเรา ผู้ผลิตต่างพากันพัฒนาเทคโนโลยีรูปแบบใหม่ๆออกมาเพื่อตอบสนองความต้องการของลูกค้า ทำให้อุปกรณ์ที่ใช้ในชีวิตประจำวันเราดูแตกต่างไปจากอดีตอย่างมาก จากยุคที่อุปกรณ์อิเล็กทรอนิกส์รับคำสั่งจากมนุษย์ด้วยสวิทช์ เปลี่ยนเป็นปุ่มกด จนในปัจจุบันนี้ระบบทัชสกรีนได้รับความนิยมเป็นอย่างมาก จนเทคโนโลยีใหม่ที่ผลิตออกมาต้องรองรับระบบนี้แทบทั้งสิ้น การสั่งงานอุปกรณ์อิเล็กทรอนิกส์ได้รับการพัฒนาให้เปลี่ยนแปลงไปอย่างมาก ผมมีแนวคิดที่จะพัฒนาการสั่งงานอุปกรณ์อิเล็กนิกส์ในรูปแบบใหม่ โดยใช้หลักการและทฤษฎีต่างดังต่อไปนี้ เพื่อพัฒนาการสั่งงานหุ่นยนต์ด้วยการเคลื่อนไหวของมือมนุษย์

2.1 โครงสร้างของปริญญานิพนธ์



Arduino Board ตัวนี้จะทำหน้าที่ควบคุมตัวหุ่นยนต์ โดยรับคำสั่งที่ส่งผ่านWireless





ต้นแบบหุ่นยนต์สี่ขา ใช้ servo motor เพื่อทำให้เคลื่อนไหวได้

รูปที่ 2.1 โครงสร้างของโครงงาน

ระบบการทำงานเริ่มจากการเก็บข้อมูลของมือว่าขยับไปแบบใด โดยเราใช้เซ็นเซอร์วัดความงอ(flex sensor)และเว็นเซอร์วัดความเร่ง(accelerometer)ในการวัดความงอของนิ้วและการพลิกของฝ่ามือ การที่จะเก็บค่าดังกว่านั้นจำเป็นต้องเขียนโปรแกรมเพื่อให้อ่านค่าอนาล็อคจากเซ็นเซอร์เข้ามาเก็บเป็นข้อมูลดิจิตอลในตัว arduino ด้วยrange จาก 512 -614 (เป็นค่าที่ใช้ในการทดสอบ สามารถเปลี่ยนแปลงขนาดของrangeได้) เพื่อใช้ค่าที่ได้ ไป ตั้งค่าให้ความกว้างพัลส์(pulse width)ที่ output pin

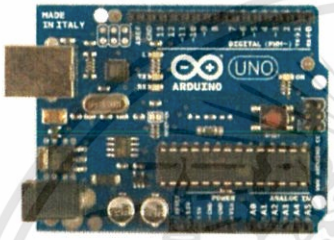
2.2 Arduino

บอร์ด Arduino ประกอบด้วยไมโครคอนโทรลเลอร์ตระกูล AVR ของ Atmel 8-bit กับองค์ประกอบเสริม เพื่ออำนวยความสะดวกการเขียนโปรแกรมและการรวมตัวกันเป็นวงจรอื่นๆ สิ่งสำคัญของ Arduino เป็นวิธี มาตรฐานที่ใช้เชื่อมต่อช่วยให้ CPU board ที่จะเชื่อมต่อกับ add-on module ต่างๆ เปรียบเสมือนshields สามารถสื่อสารกับบอร์ด Arduino โดยตรงผ่าน pin ต่างๆ แต่บาง shields เชื่อมต่อด้วยแอดเดรสผ่านทาง I^2C serial bus ช่วยให้ shields มากมายที่ซ้อนทับกัน สามารถทำงานแบบขนานพร้อมกันได้ official Arduinos ได้ใช้ ชุด mega AVRชิป เฉพาะatmega8 , ATmega168 , ATMEGA328 , ATMEGA1280 และ ATMEGA2560 เป็นต้น บอร์ดส่วนมากทำงานร่วมกับไฟ DC 5 โวลต์ และคริสตัล 16 MHz(หรือ ceramic resonator ในบางประเภท)แม้ว่าการออกแบบบางอย่าง เช่น การทำงาน Lilypad ที่ 8 MHz เนื่องจากข้อจำกัดรูปแบบปัจจัยที่เฉพาะเจาะจง Arduino ยังเป็นไมโครคอนโทรลเลอร์ที่ถูกโปรแกรมไว้ล่วงหน้าเพื่อช่วยลดความยุ่งยากในการอัพโหลดโปรแกรมลงหน่วยความจำแบบ flash บนชิป

เมื่อใช้ Arduino software stack, Board ทั้งหมดถูกโปรแกรมผ่านการเชื่อมต่อ แบบ RS-232 serial conection แต่วิธีการนี้มีการใช้งานแตกต่างกันไปตามรุ่นของฮาร์ดแวร์บนบอร์ด Arduino

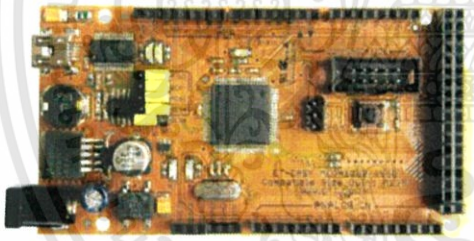
ปัจจุบันสามารถโปรแกรมผ่านทาง USB , โดยใช้ USB-to-serial adapter chips อย่าง FTDI FT232 บางตระกูล เช่น Arduino mini ใช้ USB adapter Board, เคเบิล, Bluetooth หรือ วิธีการอื่น ๆ(เมื่อใช้กับเครื่องมือไมโครคอนโทรลเลอร์แบบดั้งเดิมแทนArduino IDE, standard AVR ISP ในการเขียนโปรแกรม)เป็นเรื่องธรรมดาที่ Arduino มักจะใช้ในการศึกษาระดับโรงเรียนเพื่อให้ง่ายต่อการสร้างรถบังคับ, หุ่นยนต์ขนาดเล็ก และอื่น ๆ Arduino ที่เลือกมาให้ใช้ คือ

1. Arduino Uno R3 เป็นบอร์ด Arduino ที่ใช้ ATmega328 controller ใช้ input voltage 7-12V ที่ Clock speed 16MHz มี Analog inputs 6 pin และ 14 Digital I/O pins



รูปที่ 2.2 Arduino Uno R3

2. Arduino MEGA1280 เป็นบอร์ด Arduino ที่ใช้ ATmega1280 controller ใช้ input voltage 7-12V ที่ Clock speed 16MHz มี Analog inputs 16 pin และ 54 Digital I/O pins



รูปที่ 2.3 Arduino DUINO MEGA 1280

2.2.1 ข้อได้เปรียบของ Arduino

2.2.1.1 ราคาไม่แพง - ราคา Arduino บอร์ดไม่แพงเมื่อเทียบกับ บอร์ดอื่น

2.2.1.2 ทำงานได้หลายแพลตฟอร์ม - โปรแกรมพัฒนา Arduino ทำงานได้ทั้งบนวินโดวส์, Macintosh OSX, และ บนลีนุกซ์ ในขณะที่บอร์ดอื่นทำงานได้เฉพาะบนวินโดวส์

2.2.1.3 มีPort I/O เพียงพอต่อความต้องการ สำหรับใช้งาน

2.2.1.4 โปรแกรมพัฒนา Arduino มีความสามารถครอบคลุม สามารถเขียนโปรแกรมการทำงานที่ต้องการได้

2.2.1.5 เปิดเผย source code และ นำไปพัฒนาต่อยอดได้ - โปรแกรม Arduino ตีพิมพ์แบบเปิดเผยsource code และสามารถเพิ่มเติมความสามารถผ่าน C++ library

2.2.2 โปรแกรม Arduino

2.2.2.1 เปิดโปรแกรม Arduino

รูปที่ 2.4 ไอคอนโปรแกรม arduino

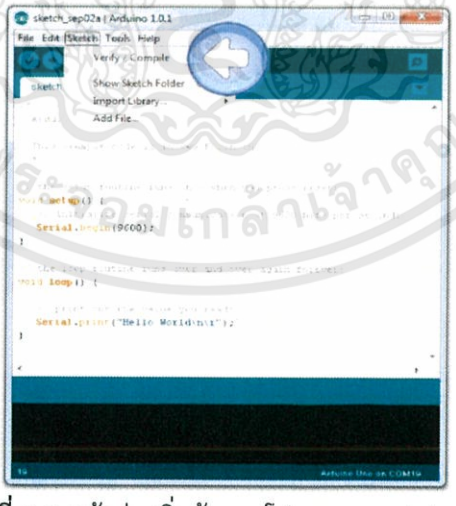
2.2.2.2 เมื่อเปิดโปรแกรมแล้วจะพบกับหน้าต่างของ IDE ดังรูป

รูปที่ 2.5 หน้าต่างเริ่มต้นของโปรแกรม arduino 1

2.2.2.3 ไปที่ Tools -> Board แล้วเลือกให้ตรงกับบอร์ดที่ใช้งาน

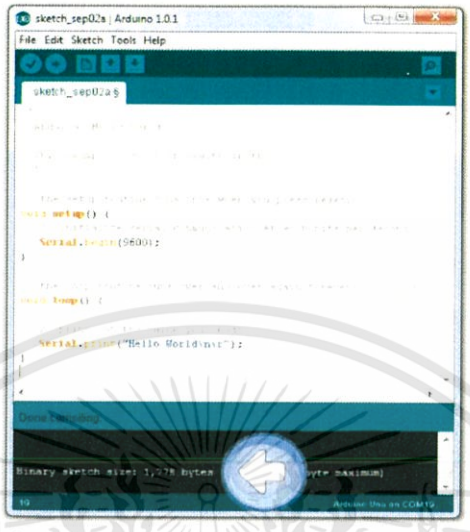
2.2.2.4 เขียนโปรแกรม

2.2.2.5 จากนั้นคอมไฟล์โปรแกรมโดยไปที่ Sketch->Verify / Compile



รูปที่ 2.6 หน้าต่างเริ่มต้นของโปรแกรม arduino 2

2.2.2.6 เมื่อคอมไฟล์เรียบร้อยจะมีข้อความปรากฏดังรูป



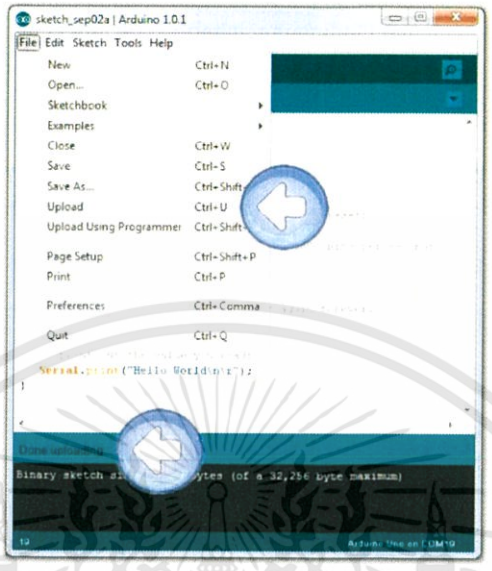
รูปที่ 2.7 หน้าต่างเริ่มต้นของโปรแกรม arduino 3

2.2.2.7 ต่อบอร์ด Arduino เข้ากับคอมพิวเตอร์ผ่านทางพอร์ต USB

2.2.2.8 จากนั้นให้ไปที่ Tools->Serial Port และเลือกให้ตรงกับบอร์ด Arduino ที่ใช้งาน

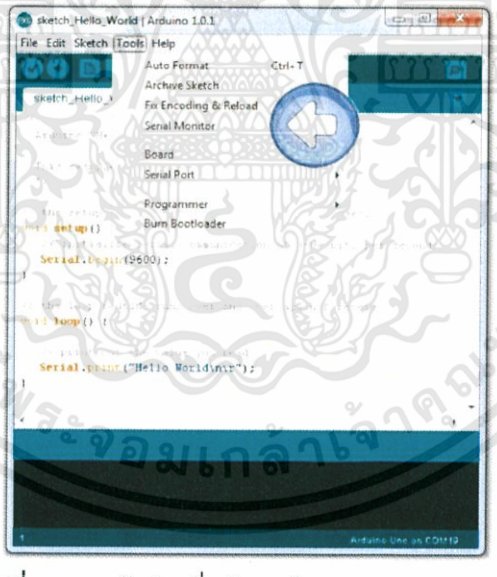
รูปที่ 2.8 หน้าต่างเริ่มต้นของโปรแกรม arduino 4

2.2.2.9 โหลดโปรแกรมเข้าบอร์ด Arduino โดยไปที่ File->Upload



รูปที่ 2.9 หน้าต่างเริ่มต้นของโปรแกรม arduino 5

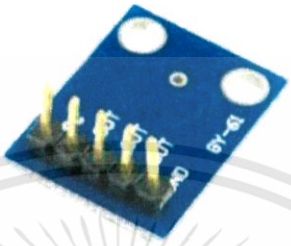
2.2.2.10 จากนั้นเปิด Serial Monitor ของ Arduino โดยไปที่ Tools->Serial Monitor



รูปที่ 2.10 หน้าต่างเริ่มต้นของโปรแกรม arduino 6

2.3 Accelerometer

Accelerometer คือ เซนเซอร์ประเภทหนึ่งที่วัดความเร่ง ไม่ว่าจะเป็นความเร่งในการเคลื่อนที่ หรือความเร่งของโลก ค่า G ประโยชน์ของเซนเซอร์ที่นำมาใช้กันมากที่สุด คือการหามุมเอียง ของวัตถุ โดยหลักการคิดมุมจะคิดแบบเวกเตอร์

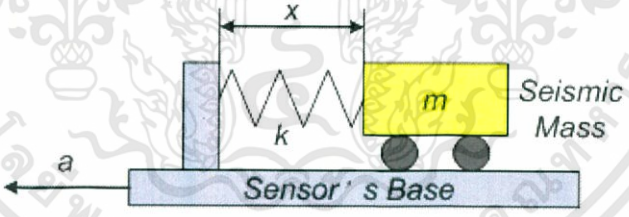


รูปที่ 2.11 เซ็นเซอร์วัดความเร่ง ADXL335 บน GY-61 Breakout Board

มิเตอร์วัดความเร่งนี้โดยหลักๆแล้วจะแบ่งเป็น 2 ชนิด

2.3.1 มิเตอร์วัดอัตราเร่งแบบไซซมิกแมส (seismic mass accelerometer)

มิเตอร์ชนิดนี้อาศัยหลักการตรวจวัดระยะขจัดเชิงเส้นแล้วนำไปคำนวณหาอัตราเร่งที่เกิดขึ้น โดยเทคนิคดังกล่าวสามารถอธิบายง่ายๆ ได้ก็คือ วัตถุชิ้นหนึ่งจะมีความเร่งได้ ก็จะต้องมีแรงมากระทำ ยิ่งมีแรงมากระทำมาก ก็จะยิ่งมีความเร่งมาก ในขณะเดียวกันแรงต้านการเคลื่อนที่ก็จะมากด้วย นอกจากนี้เมื่อมีแรงมาทำให้วัตถุ เกิดการเคลื่อนที่ ก็จะมีระยะขจัด ซึ่งก็จะแปรผันตรงกับแรงที่มากระทำที่วัตถุ ยิ่งแรงมากระยะขจัดยิ่งมาก จากความสัมพันธ์ดังกล่าวได้นำไป ใช้เป็นหลักการพื้นฐานของมิเตอร์วัดอัตราเร่งแบบไซซมิกแมสในการตรวจวัดอัตรา เร่งของวัตถุในเทอมของระยะขจัดที่เกิดขึ้น



รูปที่ 2.12 โครงสร้างพื้นฐานของมิเตอร์วัดอัตราเร่งแบบไซซมิกแมส

โครงสร้างนี้มีมวล m ที่เรียกว่ามวลตรวจการสั่นไหว (seismic mass) ยึดติดอยู่กับสปริงที่มีค่า spring constant เท่ากับ k และมวลนี้สามารถเคลื่อนที่ในแนวระดับได้ เมื่อตัวเซนเซอร์ตัวนี้ถูกทำให้มีอัตราเร่งเกิดขึ้นจะส่งผลให้มวล m เคลื่อนที่ ซึ่งระยะที่เคลื่อนที่ออกไปจะเป็นระยะขจัดเท่ากับ x และมีทิศทางตรงกันข้ามกับการเคลื่อนที่ของตัวมิเตอร์ ดังนั้นอัตราเร่ง a ของวัตถุสามารถคำนวณหาค่าได้จากความสัมพันธ์ต่อไปนี้

$$a = xk/m$$

โดยที่

a คือ อัตราเร่งของวัตถุ หน่วย เมตร/วินาที

x คือ ระยะขจัดของมวล m หน่วย เมตร

k คือ ค่าคงที่ของสปริง หน่วย นิวตัน/เมตร

m คือ น้ำหนักของมวล m หน่วย กิโลกรัม

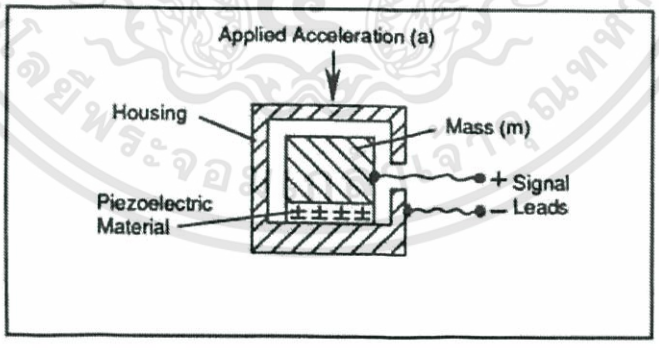
จากสมการดังกล่าวจะแสดงให้เห็นว่า

- เมื่ออัตราเร่งของวัตถุมีค่าเพิ่มขึ้น ทำให้ระยะขจัดของมวล m มีค่าเพิ่มขึ้นตามไปด้วย
- เมื่ออัตราเร่งของวัตถุม่ค่าลดลง ทำให้มวล m เคลื่อนที่ไปดันสปริง
- เมื่ออัตราเร่งของวัตถุหยุดลง ก็จะทำให้มวล m เคลื่อนที่กลับมาอยู่ตำแหน่งเดิม (ตำแหน่งอ้างอิง)

แต่ในทางปฏิบัติเราสามารถวัดระยะขจัดของมวล m ได้โดยอาศัยมิเตอร์อีกชนิดหนึ่ง คือมิเตอร์วัดระยะขจัดเชิงเส้น (LVDT, potentiometer) ส่วนการวิเคราะห์หาค่าอัตราเร่งที่เกิดขึ้นเราสามารถคำนวณหาได้โดยใช้คอมพิวเตอร์ มิเตอร์วัดอัตราเร่งแบบไซซมิกแมส นี้จะนิยมใช้ในการตรวจวัดลักษณะการช็อกและลักษณะการสั่นสะเทือนที่มีความถี่ ต่ำมากๆ เช่น ในเครื่องมือตรวจวัดแผ่นดินไหว หรือในเครื่องมือตรวจวัดการปะทุใต้ดินของภูเขาไฟ ฯลฯ

2.3.2 มิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริก (piezoelectric accelerometer)

คุณสมบัติพื้นฐานทางไฟฟ้าของผลึก เพียโซอิเล็กทริก (piezoelectric crystal) ถูกค้นพบโดย Pierre และ Jacques Curie ในราวปี ค.ศ.1880 ซึ่งเจ้า piezoelectric crystal นี้มันมีคุณสมบัติพิเศษ คือ เมื่อมันถูกแรงทางกลมากระทำ มันจะสร้างประจุไฟฟ้าขึ้นมา โดยเป็นสัดส่วนกับแรงกระทำนั้น ซึ่งจากคุณสมบัติพิเศษนี้ได้ถูกดัดแปลงนำไปใช้สร้างอุปกรณ์ต่างๆมากมาย เช่น ใช้เป็นแบตเตอรี่จ่ายพลังงานไฟฟ้าให้กับนาฬิกาข้อมือดิจิตอลที่เราใช้ทั่วไป และยังใช้สร้างมิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริกอีกด้วย โครงสร้างของมอเตอร์วัดอัตราเร่ง แบบเพียโซอิเล็กทริกจะประกอบด้วย seismic mass ยึดติดกับ piezoelectric crystal และบรรจุอยู่ในตัวถังป้องกัน โดย piezoelectric crystal ที่นิยมนำมาใช้งานได้แก่ ผลึกควอตซ์ และผลึกโซเดียมโปตัสเซียมตาเตรต (sodium potassium tartrate) เพราะมีความทนทานต่อแรงกระทำ และราคาไม่แพงมากนัก



รูปที่ 2.13 โครงสร้างพื้นฐานของมิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริก

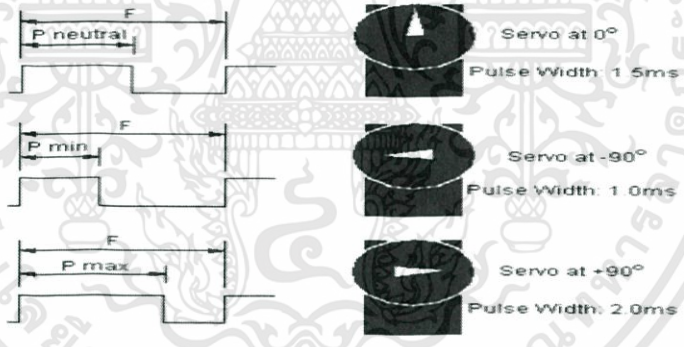
เมื่อ seismic mass (m) ถูกทำให้เกิดอัตราเร่งขึ้น (ถูกกด) มันจะส่งผ่านแรงกดไปกระทำกับ piezoelectric crystal ที่ถูกยึดติดอยู่ด้วยกัน ด้วยคุณสมบัติพิเศษของมันจะทำให้ประจุไฟฟ้าถูกสร้างขึ้นและถูกสายนำสัญญาณออกไปยังเอาต์พุตของวงจร โดยที่ด้านเอาต์พุตจะต้องมีวงจร ขยายประจุไฟฟ้า (charge amplifier) เพื่อขยายค่าประจุไฟฟ้าที่ได้ให้เป็นแรงดันเอาต์พุตตามสัดส่วนของอัตราเร่ง ที่เกิด

จะได้สามารถแสดงผลได้ด้วยโวลต์มิเตอร์ มิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริกตอบสนองต่อทางด้านความถี่สูงได้ดี แต่ในทางกลับกันก็จะมีผลตอบ สนองทางด้านความถี่ต่ำที่ไม่ดีนัก มีขนาดค่อนข้างเล็กน้ำหนักเบา และสามารถใช้งานที่มีอัตราเร่งได้สูงถึง 250,000 m.s-2 ส่วนการประยุกต์ใช้งานที่น่าสนใจและกำลังเป็นที่นิยมอยู่ในขณะนี้เห็นจะเป็นเทคโนโลยี ทัชสกรีน (Touch Screen) ที่ใช้ accelerometer ตรวจจับการเคลื่อนไหวนิ้วมือของผู้ใช้ เวลาเราใช้นิ้วลากเร็วๆ มิเตอร์วัดความเร่งจะจับความเร่งที่นิ้วเราเคลื่อนไหวแล้วสั่งให้หน้าจอเลื่อนไปตามความเร่งนั้น ถ้าเราเลื่อนนิ้วเร็วหน้าจอก็เลื่อนเร็ว แต่ถ้าเลื่อนนิ้วช้าหน้าจอก็จะค่อยๆเลื่อนไป

Accelerometer ที่ใช้คือ MMA7341L 3-Axis Accelerometer ±3/11g เป็น เซ็นเซอร์วัดสัญญาณความเร่งแบบ 3 แกม (X , Y , Z) สามารถเลือกระดับค่า Gravity ได้ ±3g และ ±11g ให้สัญญาณ Output เป็น Analog โดย PCB บอร์ดได้เพิ่มชุด Voltage Regulator 3.3 Volt ทำให้สามารถ Supply Voltage ได้ 2.2-16 Volt

2.4 เซอร์โวมอเตอร์ (Servomotor)

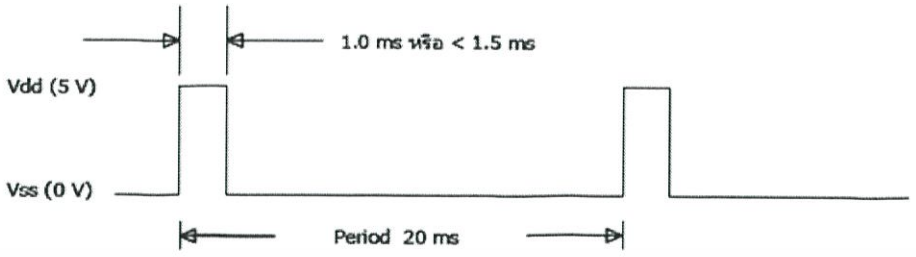
การควบคุมการทำงานของ เซอร์โวมอเตอร์ ทำได้โดย การป้อนสัญญาณความกว้างพัลส์ ให้กับมอเตอร์ซึ่งตำแหน่ง และทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์ จะมีจุดให้อ้างอิง 3 จุด



รูปที่ 2.14 จุดอ้างอิงของ Servo motor

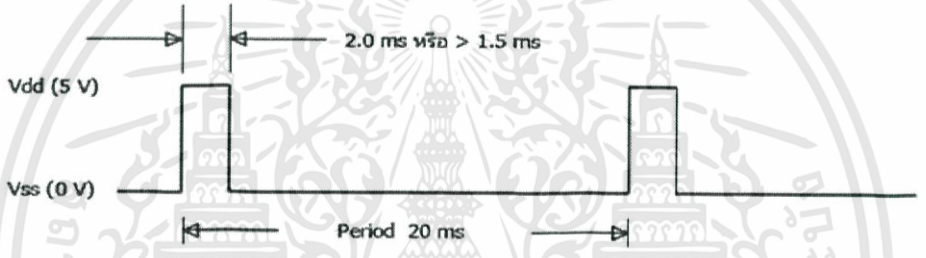
- สัญญาณความกว้างพัลส์ขนาด 1.5 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือ จุดกึ่งกลางของมอเตอร์
- สัญญาณความกว้างพัลส์ขนาด 1 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม - 90 องศา หรือในทิศทางทวนเข็มนาฬิกา
- สัญญาณความกว้างพัลส์ขนาด 2 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม + 90 องศา หรือในทิศทางตามเข็มนาฬิกา

การควบคุมให้มอเตอร์หมุนทางด้านซ้ายจะต้องป้อนสัญญาณพัลส์ที่มีขนาดความกว้างพัลส์ 1 ms หรือ ให้น้อยกว่า 1.5 msโดยจะต้องป้อนสัญญาณพัลส์นี้ทุกๆ 20 ms (หรือในช่วงประมาณ 20ms – 30ms)



รูปที่ 2.15 การควบคุมให้มอเตอร์หมุนทางด้านซ้าย

การควบคุมให้มอเตอร์หมุนทางด้านขวาจะต้องป้อนสัญญาณพัลส์ที่มีขนาดความกว้างพัลส์ 2 ms หรือ ไม่ต่ำกว่า 1.5 ms และจะต้องป้อนสัญญาณพัลส์นี้ทุกๆ 20 ms (หรือในช่วงประมาณ 20ms – 30ms) เช่นกัน

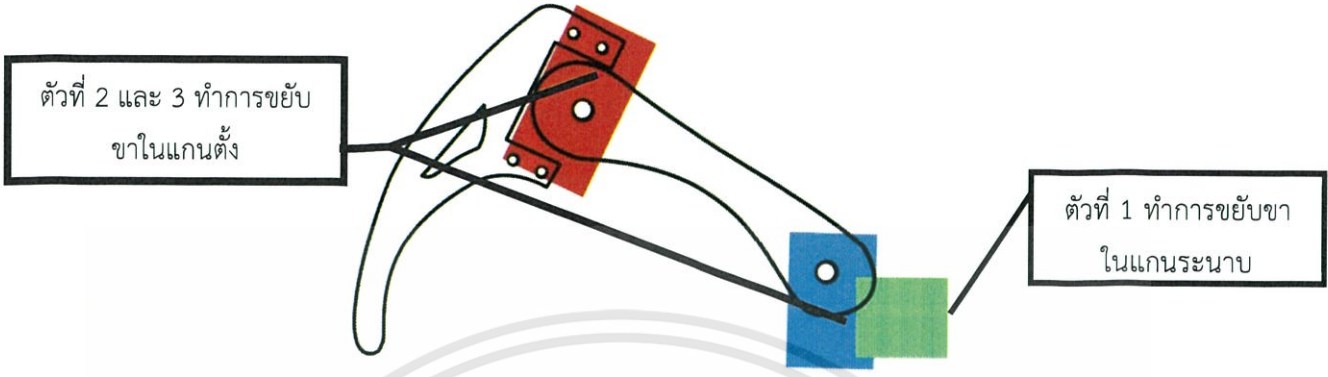


รูปที่ 2.16 การควบคุมให้มอเตอร์หมุนทางด้านขวา

การควบคุมให้มอเตอร์หยุดหมุน ทำได้โดยการส่งลอจิก “0” หรือ “1” ให้กับมอเตอร์ หรือ ก็คือการไม่จ่ายสัญญาณพัลส์ให้กับมอเตอร์

รูปที่ 2.17 การควบคุมให้มอเตอร์หยุดหมุน

การออกแบบการเคลื่นไหวของขาหุ่น ใช้Servo motor ทั้งหมด 3 ตัวต่อ 1 ขา โดยใช้มอเตอร์ตัวที่ 1 ทำการขยับขาในแนวระนาบ และตัวที่ 2 และ 3 ขยับขาหุ่นในแนวตั้งตามรูปที่ 2.18



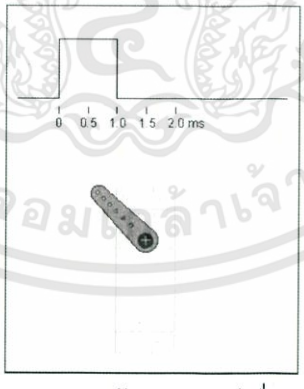
รูปที่ 2.18 ตำแหน่งของ Servomotor ของขาหุ่นยนต์แต่ละด้าน

เมื่อใช้ทั้ง 4 ขา หุ่นยนต์สามารถที่ได้ และขยับในรูปแบบอื่นๆได้อีก การเคลื่อนที่แบบนี้ทำให้สามารถเคลื่อนได้แม้จะไม่ใช่พื้นเรียบ

2.4.1 Servo Control

การมอดูเลชั่นทางความกว้างของพัลส์เพื่อควบคุมการหมุนของ servo ขนาดของพัลซ์ที่ใช้ควบคุมเซอร์โวมีขนาด 1ms-2ms โดยที่ ความกว้างของพัลซ์ 1.5 ms จะทำให้เซอร์โวอยู่ที่ตำแหน่งกึ่งกลาง ดังนั้นเราจึงต้องสร้างสัญญาณที่มีความกว้างเท่ากับ 1.5ms x 2 = 3ms หรือมีความถี่เท่ากับ 333 Hz

จากการคำนวนหาคาบเวลาPWM (PWM PERIOD) หรือใช้โปรแกรมคำนวน เพื่อกำหนดค่าให้กับ PR2 เมื่อความถี่เท่ากับ 4 MHz, PreScaler = 1:16 Duty cycle 50 % จะได้ ค่า PR2= 186 และค่า (CCPR1L:CCP1X:CCP1Y) =375 (0101110111)



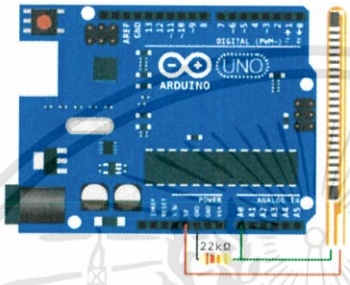
รูปที่ 2.19 การมอดูเลชั่นทางความกว้างของพัลส์เพื่อควบคุมการหมุนของ servo

2.5 Flex Sensor

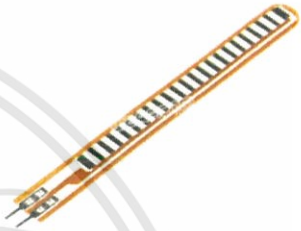
Flex Sensor เป็น Sensor ที่ใช้ตรวจจับการโค้งงอ จะติดไว้ด้านบนของนิ้ว เพื่อตรวจจับการโค้งงอของนิ้วมือแต่ละนิ้ว ที่งอแตกต่างกัน เมื่อ ผู้ใช้ที่สวมถุงมือ และมีการงอนิ้วมือ Flex Sensor จะส่งค่าที่ตรวจจับได้ ซึ่งจะเป็นค่าความต้านทาน ที่น้อยลงเรื่อยๆ เมื่อมีการโค้งงอของนิ้วมือมากขึ้น ส่งค่านี้ไปยัง

Microcontroller เพื่อทำการประมวลผลค่าที่รับมานี้ ตามคำสั่งการทำงานที่เขียนโปรแกรมโดยใช้ภาษา C ฝังไว้ใน Arduino แล้วก่อนหน้านี้ เปลี่ยนค่าจากเซ็นเซอร์ไปเป็นคำสั่ง ค่าใหม่ เพื่อสั่งการออกไปยังหุ่นยนต์

- Flat Resistance : 25K Ohms
- Bend Resistance Range : 45K to 125K Ohms
- Rower Rating : 0.50 Watts continuous. 1 watt Peak



รูปที่ 2.20 วงจรการต่อflex sensor กับ arduino



รูปที่ 2.21 Flex Sensor 2.2"

2.6 Wire module

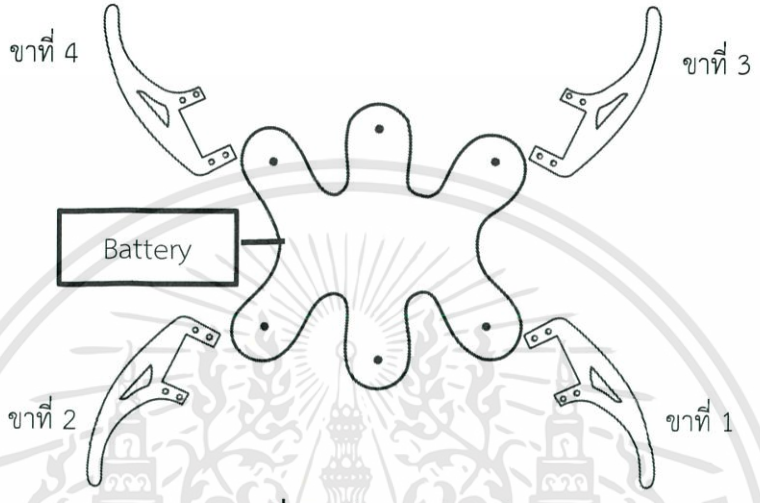
อุปกรณ์สื่อสารแบบไร้สาย เพื่อให้สามารถควบคุมยนต์ได้ แม้มีสิ่งกีดขว้างระหว่างผู้ใช้กับตัวหุ่นยนต์ ตัวที่ใช้คือ RF1100-232 RF 433MHz Transceiver Module

- Module : RF1100-232
- Color : Green + black
- Material : PCB
- Low power module. Max. transmission rate :10 mW
- Working voltage : 2.7~5.5V
- Work frequency : 433 MHz

รูปที่ 2.22 RF1100-232 RF 433MHz Transceiver Module

บทที่ 3
การออกแบบหุ่น

3.1 ส่วนประมวลผล

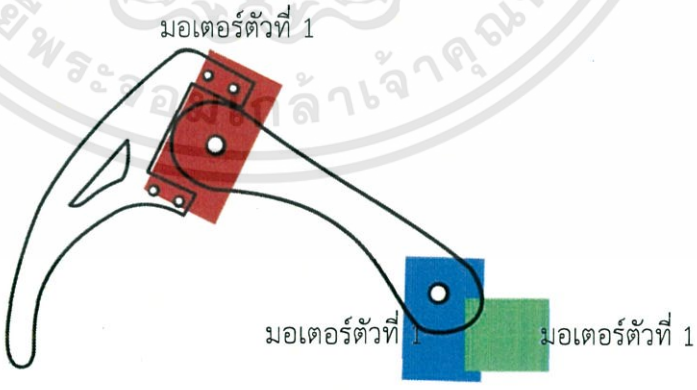


รูปที่ 3.1 ต้นแบบหุ่นยนต์

การออกแบบตัวหุ่นนั่นจะใช้มอเตอร์ทั้ง 12 ตัวในการเคลื่อนไหวข้อต่อ ซึ่งใช้ arduino ให้การควบคุมมอเตอร์ และตัวหุ่นจะรับคำสั่งจากผู้ใช้โดยผ่าน wireless module คำสั่งจากส่วนควบจะถูกประมวลผลโดยโปรแกรมบนตัว arduino ที่อยู่บนหุ่นเพื่อให้หุ่นเคลื่อนไปตามที่ผู้ใช้ต้องการ

3.1.1 การเคลื่อนไหว

ในขา 1 ข้างของหุ่นจะมีมอเตอร์ทั้งหมด 3 ตัว โดย 1 ตัวทำการควบคุมขาในแนวราบ และอีก 2 ตัว จะทำดารควบคุมให้ขาสามารถกางและหุบได้



รูปที่ 3.2 โครงสร้างขาของหุ่นยนต์

ซึ่งการที่หุ่นจะเคลื่อนที่ได้นั่นต้องให้ขาทั้ง 4 ข้าง ทำงานประสานกัน โดยออกแบบให้ทำงานร่วมกันเป็นรอบ ใน 1 รอบ ขาทั้ง 4 จะขยับทีล่ะข้างเป็นลำดับจนครบ 1 รอบแล้วทำซ้ำจ่อไหเรื่อย ๆ จะทำให้หุ่นยนต์เดินไปข้างหน้าได้

3.1.1.1 การเดินหน้า

จะเริ่มก้าวที่ขา 1 ตามลำดับต่อไปนี้

ขา1> ขา2> ขา3> ขา4

เมื่อครบรอบจะทำตามคำสั่งเดิมซ้ำไปเรื่อยๆ จนกว่าจะมีคำสั่งให้เปลี่ยนการกระทำ

3.1.1.2 การถอยหลัง

จะเริ่มก้าวที่ขา 4 ตามลำดับต่อไปนี้

ขา4> ขา3> ขา2> ขา1

เมื่อครบรอบจะทำตามคำสั่งเดิมซ้ำไปเรื่อยๆ จนกว่าจะมีคำสั่งให้เปลี่ยนการกระทำ

3.1.1.3 การหมุนซ้าย

จะเริ่มกว่าที่ ขา 2 ตามลำดับต่อไปนี้

ขา2> ขา1> ขา4> ขา3

เมื่อครบรอบจะทำตามคำสั่งเดิมซ้ำไปเรื่อยๆ จนกว่าจะมีคำสั่งให้เปลี่ยนการกระทำ

3.1.1.4 การหมุนขวา

จะเริ่มกว่าที่ ขา 4 ตามลำดับต่อไปนี้

ขา4> ขา3> ขา1> ขา2

เมื่อครบรอบจะทำตามคำสั่งเดิมซ้ำไปเรื่อยๆ จนกว่าจะมีคำสั่งให้เปลี่ยนการกระทำ

3.1.2 โค้ดโปรแกรมควบคุมหุ่น

```
#include <Servo.h>
#define rightFront 1                    // การระบุตำแหน่งของ ขาว่าให้ขาไหนเป็นซ้ายหรือขวา

#define rightBack  2
#define leftFront  3
#define leftBack   4

double coxa = 1.619;
double femur = 2;
double tibia = 4.664;

double pi = 3.14;
```

```
                                                    //default postion for Y
double centerY = 4;
                                                    //default postion for X
double centerX = 3;
double xVal=-pi/4;
double RFx2,RFy2,RFz2;
double RFx,RFy,RFz;                                 // ตัวแปรที่เอามาใช้ ในการระบุส่วน
                                                    ของ ตำแหนง
double RBx,RBy,RBz;
double LFx,LFy,LFz;
double LBx,LBy,LBz;
                                                    //Right Front Servos (1)
Servo RF0;                                          // ชื่อ ของ servo แต่ละตัว
Servo RF1;
Servo RF2;
                                                    //Right Back Servos (2)
Servo RB0;
Servo RB1;
Servo RB2;
                                                    //Left Front Servos (3)
Servo LF0;
Servo LF1;
Servo LF2;
                                                    //Left Back Servos (4)
Servo LB0;
Servo LB1;
Servo LB2;
unsigned long tNow = 0;
unsigned long tOld = 0;
unsigned long tNowSM = 0;
unsigned long tOldSM = 0;
unsigned long tNow1 = 0;
unsigned long tOld1 = 0;
int ck_1 = 0 ;
int ck_2 = 0 ;
int ck_3 = 0 ;
int ck_4 = 0 ;
int ck_5 = 0 ;
int ck_6 = 0 ;
int ck_7 = 0 ;
```

```
                                                    //counters
int j = 0;
int counter = 0;
char inByte;
void setup(){
                                                    //attach all the servos
  RB0.attach(13);                                   //1 การระบุตำแหน่ง ขาที่ต่อกับ servo
                                                    แต่ละตัว
  RB1.attach(12);
  RB2.attach(11);
  RF0.attach(4);                                    //4
  RF1.attach(3);
  RF2.attach(2);
  LB0.attach(10);                                   //2
  LB1.attach(9);
  LB2.attach(8);
  LF0.attach(7);                                    //3  การระบุตำแหน่ง ขาที่ต่อกับ
                                                    servo แต่ละตัว
  LF1.attach(6);
  LF2.attach(5);
  Serial.begin(9600);
  Serial2.begin(9600);
                                                    //Starting position at power
                                                    up   ค่าเริ่มต้นในการ setup ตัวหุ่น
  fullMove(0,0,2);
  delay(2000) ;
                                                    // Serial2.println("s");
                                                    //  Serial.println("s");
  delay(200) ;
}
void loop(){
                                                    //  delay(1000) ;
  if (Serial2.available() > 0) {                    // ตรวจสอบค่าที่เข้ามาจาก serial
                                                    port
    int inByte = Serial2.read();
    switch (inByte) {
    case '0':
      delay(100) ;
      break;
    case '1':
```

```
      shiftWalk5(12,15);                    // ถอยหลัง
     break;
   case '4':
      shiftWalk5(12,10);                    // ถอยหลัง
     break;
   case '5':
    shiftWalk3(10,10);
     break;
   case '6':
      shiftWalk2(10,10);
     break;
   case 'u':                                // เล่นท่า
      up(3,5);                              // ชูตัวขึ้น
      up(3,5);
      up(3,5);
      up(3,5);
      delay(1000) ;
      rotate(25);                           // หมุนตัว
      rotate(-50);
      rotate(25);
      rotate(25);
      rotate(-50);
      rotate(25);
      rotate(25);
      rotate(-50);
      rotate(25);
      delay(1000) ;
    break;
  case '8':
     break;
  }
}
if (Serial.available() > 0) {
  int inByte = Serial.read();
  switch (inByte) {
  case '0':
    delay(100) ;
    break;
  case '1':
     shiftWalk(12,15);
```

```
      break;
    case '4':
       shiftWalk5(12,10);
      break;
    case '5':
      shiftWalk3(10,10);
      break;
    case '6':
       shiftWalk2(10,10);
                                                    // calibrate() ;
      break;
       case '7':
                                                   // shiftWalk2(10,10);
   calibrate() ;
      break;
    case 'u':
       up(3,5);
       up(3,5);
       up(3,5);
       up(3,5);
       delay(1000) ;
       rotate(25);
       rotate(-50);
       rotate(25);
       rotate(25);
       rotate(-50);
       rotate(25);
       rotate(25);
       rotate(-50);
       rotate(25);
       delay(1000) ;
      break;
    case '8':
                                                   //   delay(1000) ;
      break;
                                                   //   default:
    }
  }
}
void rotate(int degree){                           //  การหมุนตำแหน่งของ servo แต่ละ
ตัว
```

```
  Serial.end() ;
Serial2.end() ;
  int RFcur=RF0.read();
  int RBcur=RB0.read();
  int LFcur=LF0.read();
  int LBcur=LB0.read();



                                                    //  Serial.println(RFcur);
  if(degree>0){
    for (int count1=0;count1<degree;count1++){      //  การ for loop เพื่อปรับ servo ไปที่
                                                    ละ 1 องศาโดยการ บวก หรือลบ
      RFcur+=1;
      RBcur+=1;
      LFcur+=1;
      LBcur+=1;
      RF0.write(RFcur);                             //  คำสั่งเขียน servo ให้หมุนไป
                                                    ตำแหน่งที่เราต้องการ
      RB0.write(RBcur);
      LF0.write(LFcur);
      LB0.write(LBcur);
      delay(10);
    }
  }
  else{
    for (int count1=0;count1>degree;count1--){      // การ for loop เพื่อปรับ servo ไปที่
                                                    ละ 1 องศาโดยการ บวก หรือลบ
      RFcur-=1;
      RBcur-=1;
      LFcur-=1;
      LBcur-=1;
      RF0.write(RFcur);
      RB0.write(RBcur);
      LF0.write(LFcur);
      LB0.write(LBcur);
      delay(10);
    }
  }
  Serial.begin(9600);
```

```
        Serial2.begin(9600);
}
void smoothMove(int servoNumber, double x, double y, double z, int t1, int steps){
                                                    // เป็นการ ระบุกการก้าวขาของ
                                                    robot ว่าให้ขาไหนทำงาน
                                                    ก่อนหลัง เช่น ยกก่อนแล้วก้าว แล้ว
                                                    เลื่อน และววาง
                                                    //RF
 if(servoNumber == 1){

                                                    //how big each step is
   double dx = (x-RFx)/steps;
   double dy = (y-RFy)/steps;
   double dz = (z-RFz)/steps;
   for(int i=0;i<=steps;i++){                       // การระบุ step
     RFx+=dx;
     RFy+=dy;
     RFz+=dz;
     moveServos2RF(RFx,RFy,RFz);
     delay(t1);
   }
 }
                                                    //RB
 else if(servoNumber == 2)
 {
   double dx = (x-RBx)/steps;
   double dy = (y-RBy)/steps;
   double dz = (z-RBz)/steps;
   for(int i=0;i<=steps;i++){
     RBx+=dx;
     RBy+=dy;
     RBz+=dz;
     moveServos2RB(RBx,RBy,RBz);
     delay(t1);
   }
 }

                                                    //LF
 if(servoNumber == 3)                               // เป็นการแยก โดยการเช็ความคำสั่ง
                                                    ที่มาจะระบุให้ขาไหนทำงาน โดยเลือก
                                                    จากตัวเลขที่เข้ามา
 {
```

```
    double dx = (x-LFx)/steps;
    double dy = (y-LFy)/steps;
    double dz = (z-LFz)/steps;
    for(int i=0;i<=steps;i++){
      LFx+=dx;
      LFy+=dy;
      LFz+=dz;
      moveServos2LF(LFx,LFy,LFz);
      delay(t1);
    }
  }
                                                    //LB
  else if(servoNumber == 4)                         //  เป็นการแยก โดยการเช็ความคำสั่ง
                                                    ที่มาจะระบุให้ขาไหนทำงาน โดยเลือก
                                                    จากตัวเลขที่เข้ามา
  {
    double dx = (x-LBx)/steps;
    double dy = (y-LBy)/steps;
    double dz = (z-LBz)/steps;
    for(int i=0;i<=steps;i++){
      LBx+=dx;
      LBy+=dy;
      LBz+=dz;
      moveServos2LB(LBx,LBy,LBz);
      delay(t1);
    }
  }
  else if(servoNumber == 0){
  }
}
void shiftWalk(int steps,int time){               //  การสั่งเดิน โดยต้องระบุทั้ง จำนวน
                                                    ที่ทำและความเร็ว

    Serial.end() ;
    Serial2.end() ;

                                                    //  delay(5000) ;
                                                    //  Serial.begin(9600);
                                                    //เริ่มต้นตำแหน่งแนวตั้ง
  double z = 3;
                                                    //ยกขาสูงเท่าไหร่ในการเคลื่อนไหว
```

```
  double dz = -2;
  smoothFullMove(2,-1,z,2,1,z,steps*2,time);          // การเอียงตัวเพื่อรักษาน้ำหนัก
                                                      //  print3(LBx,LBy,LBz);
                                                      //Moves left back leg up then
                                                      forward then down
  smoothMove(4,LBx,LBy,z+dz,steps,time);              //moveServos2LB(3,5,1);
                                                      //  การก้าวของขาแต่ละ servo ว่าทำ
                                                      ขาไหนก่อนหลัง    เช่นยกก่อน
  smoothMove(4,1,5,z+dz,steps,time);                  //moveServos2LB(1,5,1);
                                                      // ต่อ มาก็หมุนขาไปข้างหน้า
  smoothMove(4,1,5,z,steps,time);                     //moveServos2LB(0,5,3);
                                                      // แล้ววางขาลง
                                                      //Moves left front leg up then
                                                      forward then down
  smoothMove(3,-1,5,z+dz,steps,time);                 //moveServos2LF(-1,5,0);
  smoothMove(3,-3,5,z+dz,steps,time);                 //moveServos2LF(-3,5,1);
  smoothMove(3,-3,5,z,steps,time);                    //moveServos2LF(-3,5,3);
                                                      //delay(200000) ;

  smoothFullMove(2,1,z,2,-1,z,steps*2,time);          // การเอียงตัวเพื่อรักษาน้ำหนัก
                                                      //Moves right back leg up then
                                                      forward then down
  smoothMove(2,3,5,z+dz,steps,time);                  //moveServos2LB(3,5,1);
  smoothMove(2,1,5,z+dz,steps,time);                  //moveServos2LB(1,5,1);
  smoothMove(2,1,5,z,steps,time);                     //moveServos2LB(0,5,3);
                                                      //Moves right front leg up then
                                                      forward then down
                                                      //  Serial.end() ;
                                                      //  delay(5000) ;

     Serial.begin(9600);
       Serial2.begin(9600);
  smoothMove(1,-1,5,z+dz,steps,time);                 //moveServos2LF(-1,5,0);
  smoothMove(1,-3,5,z+dz,steps,time);                 //moveServos2LF(-3,5,1);
  smoothMove(1,-3,5,z,steps,time);                    //moveServos2LF(-3,5,3);
}
void shiftWalk2(int steps,int time){                  // เหมือนกัน กับ shiftWalk ต่างกัน
                                                      ตรง ขาที่ทำและลำดับการทำงาน
                                                      //starting vertical position

  double z = 3;
  double dz = -2;
```

```
Serial.end() ;
    Serial2.end() ;
                                                    //  delay(5000) ;
                                                    //  Serial.begin(9600);
  smoothFullMove(2,-1,z,2,1,z,steps*2,time);
  smoothMove(4,1,5,3,steps,time);                   //moveServos2LB(0,5,3);
                                                    //Moves left front leg up then
                                                    forward then down
  smoothMove(3,-1,5,1,steps,time);                  //moveServos2LF(-1,5,0);
                                                    // servo 1 ยกขาขึ้น
                                                    //  3 คือ servo ตัว 3
  smoothMove(3,1,5,1,steps,time);                   //moveServos2LF(-3,5,1);
  smoothMove(3,1,5,3,steps,time);                   //moveServos2LF(-3,5,3);
                                                    //Moves left back leg up then
                                                    forward then down
  smoothMove(4,3,5,1,steps,time);                   //moveServos2LB(3,5,1);
                                                    // servo 1 ยกขาขึ้น
                                                    //  4 คือ servo ตัว 4
  smoothMove(4,9,5,1,steps,time);                   //moveServos2LB(1,5,1);
  smoothMove(4,9,5,3,steps,time);                   //moveServos2LB(0,5,3);

  smoothFullMove(2,1,z,2,-1,z,steps*2,time);        //Moves right back leg up then
                                                    forward then down
  smoothMove(2,3,5,z+dz,steps,time);                //moveServos2LB(3,5,1);
  smoothMove(2,1,5,z+dz,steps,time);                //moveServos2LB(1,5,1);
  smoothMove(2,1,5,z,steps,time);                   //moveServos2LB(0,5,3);
                                                    //delay(2000) ;
                                                    //Moves right front leg up then
                                                    forward then down
                                                    //  Serial.end() ;
                                                    //  delay(5000) ;
    Serial.begin(9600);
Serial2.begin(9600);
  smoothMove(1,-1,5,1,steps,time);                  //moveServos2LF(-1,5,0);
  smoothMove(1,-10,5,1,steps,time);                 //moveServos2LF(-3,5,1);
                                                    //delay(20000) ;
  smoothMove(1,-10,5,3,steps,time);                 //moveServos2LF(-3,5,3);
                                                    //delay(2000) ;
}
void shiftWalk3(int steps,int time){
```

```
                                                    //starting vertical position
  double z = 3;
                                                    //How far to raise the leg for
                                                    each movement
double dz = -2;
  Serial.end() ;
    Serial2.end() ;
                                                    //  delay(5000) ;
                                                    //Serial.begin(9600);
                                                    //  print3(LBx,LBy,LBz);

  smoothFullMove(2,1,z,2,-1,z,steps*2,time);
  smoothMove(2,-1,5,3,steps,time);                  //moveServos2LB(0,5,3);
                                                    //Moves right front leg up then
                                                    forward then down
  smoothMove(1,-1,5,1,steps,time);                  //moveServos2LF(-1,5,0);
  smoothMove(1,3,5,1,steps,time);                   //moveServos2LF(-3,5,1);
  smoothMove(1,3,5,3  ,steps,time);                 //moveServos2LF(-3,5,3);
                                                    //delay(200000) ;
                                                    //Moves right back leg up then
                                                    forward then down
  smoothMove(2,3,5,1,steps,time);                   //moveServos2LB(3,5,1);
  smoothMove(2,7,5,1,steps,time);                   //moveServos2LB(1,5,1);
  smoothMove(2,7,5,3,steps,time);                   //moveServos2LB(0,5,3);
  smoothFullMove(2,-1,z,2,1,z,steps*2,time);
                                                    //  print3(LBx,LBy,LBz);
                                                    //Moves left back leg up then
                                                    forward then down
  smoothMove(4,LBx,LBy,z+dz,steps,time);            //moveServos2LB(3,5,1);
  smoothMove(4,1,5,z+dz,steps,time);                //moveServos2LB(1,5,1);
  smoothMove(4,1,5,z,steps,time);                   //moveServos2LB(0,5,3);
                                                    //   Serial.end() ;
                                                    //  delay(5000) ;

     Serial.begin(9600);
      Serial2.begin(9600);

                                                    //Moves left front leg up then
                                                    forward then down
  smoothMove(3,-1,5,z+dz,steps,time);               //moveServos2LF(-1,5,0);
  smoothMove(3,-3,5,z+dz,steps,time);               //moveServos2LF(-3,5,1);
  smoothMove(3,-3,5,z,steps,time);                  //moveServos2LF(-3,5,3);
```

```
                                                    // Serial2.println(3);
                                                    // Serial.println(3);
}
void shiftWalk4(int steps,int time){
                                                    //starting vertical position
  double z = 3;
                                                    //How far to raise the leg for
                                                    each movement
  double dz = -2;
  Serial.end() ;
    Serial2.end() ;
                                                    //delay(5000) ;
                                                    //Serial.begin(9600);
                                                    //print3(LBx,LBy,LBz);
  smoothFullMove(2,1,3,2,-1,3,steps*2,time);        // 2,1,3,2,-1,3,
                                                    //delay(4000) ;
  smoothMove(2,1,5,3,steps,time);                   //moveServos2LB(3,5,1);
                                                    //delay(5000) ;
                                                    //Moves right front leg up then
                                                    forward then down
  smoothMove(1,-1,5,1,steps,time);                  //moveServos2LF(-1,5,0);
  smoothMove(1,3,5,1,steps,time);                   //moveServos2LF(-3,5,1);
  smoothMove(1,3,5,3  ,steps,time);                 //moveServos2LF(-3,5,3);
                                                    //delay(200000) ;
                                                    //delay(4000) ;
  smoothMove(2,3,5,1,steps,time);                   //moveServos2LB(3,5,1);
  smoothMove(2,7,5,1,steps,time);                   //moveServos2LB(1,5,1);
  smoothMove(2,7,5,3,steps,time);                   //moveServos2LB(0,5,3);
                                                    //delay(4000) ;
  smoothFullMove(2,-1,z,2,1,z,steps*2,time);
                                                    //Moves left back leg up then
                                                    forward then down
  smoothMove(4,3,5,1,steps,time);
                                                    //moveServos2LB(3,5,1);
                                                    // servo 1 ยกขาขึ้น
                                                    //  4 คือ servo ตัว 4
  smoothMove(4,9,5,1,steps,time);                   //moveServos2LB(1,5,1);
  smoothMove(4,9,5,3,steps,time);                   //moveServos2LB(0,5,3);
                                                    //delay(4000) ;
                                                    // Serial.end() ;
```

```
                                                    //  delay(5000) ;
      Serial.begin(9600);
       Serial2.begin(9600);
                                                    //Moves left front leg up then
                                                    forward then down
  smoothMove(3,-1,5,1,steps,time);                  //moveServos2LF(-1,5,0);
                                                    // servo 1 ยกขาขึ้น
                                                    //  3 คือ servo ตัว 3
  smoothMove(3,1,5,1,steps,time);                   //moveServos2LF(-3,5,1);
  smoothMove(3,1,5,3,steps,time);                   //moveServos2LF(-3,5,3);
                                                    //delay(4000) ;
                                                    //   Serial2.println(4);
                                                    // Serial.println(4);
}
void shiftWalk5(int steps,int time)
{
                                                    //starting vertical position
  double z = 3;
                                                    //How far to raise the leg for
                                                    each movement
  double dz = -2;
    Serial.end() ;
       Serial2.end() ;
                                                    //  delay(5000) ;
                                                    //  Serial.begin(9600);
  smoothFullMove(2,1,z,2,-1,z,steps*2,time);
  smoothMove(2,-1,5,3,steps,time);                  //moveServos2LB(0,5,3);
                                                    //Moves right front leg up then
                                                    forward then down
  smoothMove(1,-1,5,1,steps,time);                  //moveServos2LF(-1,5,0);
  smoothMove(1,3,5,1,steps,time);                   //moveServos2LF(-3,5,1);
  smoothMove(1,3,5,3   ,steps,time);                //moveServos2LF(-3,5,3);
                                                    //delay(200000) ;
                                                    //Moves right back leg up then
                                                    forward then down
  smoothMove(2,3,5,1,steps,time);                   //moveServos2LB(3,5,1);
  smoothMove(2,7,5,1,steps,time);                   //moveServos2LB(1,5,1);
  smoothMove(2,7,5,3,steps,time);                   //moveServos2LB(0,5,3);
                                                    //shift entire body forward and to
                                                    the left. This causes the robot to
```

```
                                                    move forward 1 inch
                                                    //and prepares for the left legs
                                                    to move
  smoothFullMove(2,-1,z,2,1,z,steps*2,time);
  smoothMove(4,1,5,3,steps,time);                   //moveServos2LB(0,5,3);
                                                    //Moves left front leg up then
                                                    forward then down
  smoothMove(3,-1,5,1,steps,time);                  //moveServos2LF(-1,5,0);
                                                    // servo 1 ยกขาขึ้น
                                                    //  3 คือ servo ตัว 3
  smoothMove(3,1,5,1,steps,time);                   //moveServos2LF(-3,5,1);
  smoothMove(3,1,5,3,steps,time);                   //moveServos2LF(-3,5,3);
                                                    //Moves left back leg up then
                                                    forward then down
                                                    //   Serial.end() ;
                                                    //  delay(5000) ;
     Serial.begin(9600);
       Serial2.begin(9600);
  smoothMove(4,3,5,1,steps,time);                   //moveServos2LB(3,5,1);
                                                    // servo 1 ยกขาขึ้น
                                                    //  4 คือ servo ตัว 4
  smoothMove(4,9,5,1,steps,time);                   //moveServos2LB(1,5,1);
  smoothMove(4,9,5,3,steps,time);                   //moveServos2LB(0,5,3);
                                                    // Serial2.println(5);
                                                    //Serial.println(5);
}

                                                    /*IK MOVE FUNCTIONS*/
                                                    //moves the robot vertically from
                                                    z1 to z2 and then back to z1
void up(double z1, double z2){                      // ยกตัวหุ่นขึ้น
  Serial.end() ;
  Serial2.end() ;
  double z = z1;
  while(z<=z2){
    fullMove(0,0,z);
    delay(25);
    z+=.1;
  }
  while(z>=z1){
    fullMove(0,0,z);
```

```
    delay(25);
    z-=.1;
  }
        Serial.begin(9600);
        Serial2.begin(9600);
}
                                    //moves the robot in a circle of
                                    radius r and at a height z
void around(double r, double z){
  double x = 0;
  while(x<=2*3.14){
    fullMove(r*cos(x),r*sin(x),z);
    delay(50);
    x+=.1;
  }
}
                                    //Moves the entire robot body to
                                    position x,y,z. Note this results in
                                    very jerky movements
void fullMove(double x, double y, double z){
  double rightY = centerY-y;
  double leftY = centerY+y;
  double frontX=x-centerX;
  double backX=x+centerX;
  RFx=x-centerX;
  RFy=centerY-y;
  RFz=z;
  RBx=x+centerX;
  RBy=centerY-y;
  RBz=z;
  LFx=x-centerX;
  LFy=centerY+y;
  LFz=z;
  LBx=x+centerX;
  LBy=centerY+y;
  LBz=z;
  moveServosRF(gamma(RFx,RFy,RFz),alpha(RFx,RFy,RFz),beta(RFx,RFy,RFz));
  moveServosRB(gamma(RBx,RBy,RBz),alpha(RBx,RBy,RBz),beta(RBx,RBy,RBz));
  moveServosLF(gamma(LFx,LFy,LFz),alpha(LFx,LFy,LFz),beta(LFx,LFy,LFz));
  moveServosLB(gamma(LBx,LBy,LBz),alpha(LBx,LBy,LBz),beta(LBx,LBy,LBz));
```

```
}
                                                    //Moves the entire body from
                                                    x0,y0,z0 to x,y,z smoothly with
                                                    "steps" steps separated by t1
                                                    milliseconds. This is usefully for
                                                    shifting the center of gravity
void smoothFullMove(double x0, double y0, double z0, double x, double y, double z,
int t1, int steps){
  double dx = (x-x0)/steps;
  double dy = (y-y0)/steps;
  double dz = (z-z0)/steps;
  for(int i=0;i<steps;i++){
    x0+=dx;
    y0+=dy;
    z0+=dz;
    fullMove(x0,y0,z0);
    delay(t1);
  }
}
                                                    //used for attaching the legs to
                                                    the servos at the right position
void calibrate(){                                   // ปรับแต่ง servo โดยระบุไปที่ 90
                                                    องศา
  RB0.write(90);
  RB1.write(0);
  RB2.write(90);
  RF0.write(90);
  RF1.write(0);
  RF2.write(90);
  LB0.write(90);
  LB1.write(0);
  LB2.write(90);
  LF0.write(90);
  LF1.write(0);
  LF2.write(90);
}
void calibrate2(){                                  // ปรับแต่ง servo โดยระบุไปที่ 90
                                                    องศา
  moveServosRF(90,90,90);
  moveServosRB(90,90,90);
```

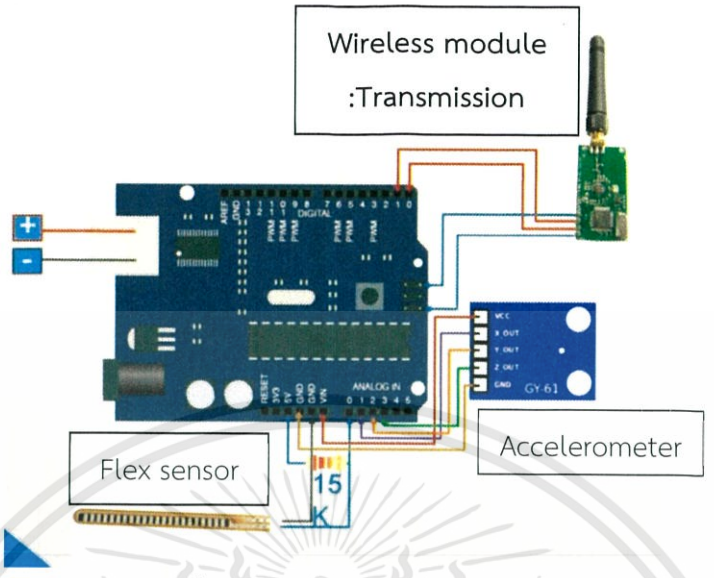
```
  moveServosLF(90,90,90);
  moveServosLB(90,90,90);
}
                                                    //debugging function to print out
                                                    three ints
void print3(int i,int o,int p){
                                                    //  Serial.print(i);
                                                    //  Serial.print("\t");
                                                    //  Serial.print(o);
                                                    //  Serial.print("\t");
                                                    //  Serial.println(p);
}
                                                    /***INVERSE KINEMATICS***/
                                                    //IK for gamma (coxa angle)
int gamma(double x, double y, double a){
  double g = atan(x/y);
  return int(g*57.3)+90;
}
                                                    //IK for beta (femur angle)
double beta(double x, double y, double a){
  double g = atan(x/y);
  x/=cos(g);
                                                    //double bet
                                                    =acos((pow(x,2)+pow(y,2)+pow(a,2
                                                    )-pow(c,2)-pow(b,2))/(-2*c*b));

  double bet =acos((pow(a,2)+pow(y-coxa,2)-pow(tibia,2)-pow(femur,2))/(-
2*femur*tibia));
  return int(bet*57.3);
}
                                                    //IK for alpha (tibia angle)
double alpha(double x, double y, double a){
  double g = atan(x/y);
  x/=cos(g);
  double L2=sqrt(pow(a,2)+pow(y-coxa,2));
  double alp1 = atan((y-coxa)/a);
  double alp2 = acos((pow(tibia,2)-pow(femur,2)-pow(L2,2))/(-2*femur*L2));
  double alp = alp1+alp2;
  return int(alp*57.3);
}
```

```
                                                    /**Move Functions**/
                                                    //The following "moveServos2XX"
                                                    move the tip of the corresponding
                                                    legs to the points x,y,z. This
                                                    probably could have been
                                                    combined with the
                                                    //"moveServosXX" functions but
                                                    were written after so they simply
                                                    call those functions.
void moveServos2RB(double x, double y, double z){
                                                    // พวกนี้จะเป็นการกระทำแค่ขาใดขา
                                                    หนึ่ง ที่จะต้องการสั่ง
  RBx=x;
  RBy=y;
  RBz=z;
  moveServosRB(gamma(RBx,RBy,RBz),alpha(RBx,RBy,RBz),beta(RBx,RBy,RBz));
}
void moveServos2RF(double x, double y, double z){
  RFx=x;
  RFy=y;
  RFz=z;
  moveServosRF(gamma(RFx,RFy,RFz),alpha(RFx,RFy,RFz),beta(RFx,RFy,RFz));
}
void moveServos2LB(double x, double y, double z){
  LBx=x;
  LBy=y;
  LBz=z;
                                                    //  Serial.print("alpha");
                                                    //
                                                    Serial.println(alpha(LBx,LBy,LBz));
  moveServosLB(gamma(LBx,LBy,LBz),alpha(LBx,LBy,LBz),beta(LBx,LBy,LBz));
}
void moveServos2LF(double x, double y, double z){
  LFx=x;
  LFy=y;
  LFz=z;
  moveServosLF(gamma(LFx,LFy,LFz),alpha(LFx,LFy,LFz),beta(LFx,LFy,LFz));
}
                                                    //The following "moveServosXX"
                                                    functions move the servos in one
                                                    leg to the given angles. gamma (g),
```

```
                                                    alpha (a), and beta (b) as defined
                                                    in the IK model
void moveServosRB(int g, int a, int b){            //The following calibrations were
                                                    determined experimentally to get
                                                    the angles to line up with the IK
                                                    equations
  g=(g);
  a=(180-a+10);
  b=(b+10);
  RB0.write(g);
  RB1.write(a);
  RB2.write(b);

                                                    //if out of range
  if(g>180||g<0||a>180||a<0||b>180||b<0)
  {

                                                    // Serial.println("RB error");
    delay(1) ;
  }

                                                    // print3(g,a,b);
}

                                                    //move the joints in the RF servo
                                                    to angles g, a, and b
void moveServosRF(int g, int a, int b){            //The following calibrations were
                                                    determined experimentally to get
                                                    the angles to line up with the IK
                                                    equations
  g=(g);
  a=(180-a+10);
  b=(b+10);
  RF0.write(g);
  RF1.write(a);
  RF2.write(b);

                                                    //if out of range
  if(g>180||g<0||a>180||a<0||b>180||b<0)
  {

                                                    //   Serial.println("RB error");
    delay(1) ;
  }

                                                    //   Serial.println("RF error");
}
```

```
                                        //move the joints in the LB servo
                                        to angles g, a, and b
void moveServosLB(int g, int a, int b){ //The following calibrations were
                                        determined experimentally to get
                                        the angles to line up with the IK
                                        equations
  g=180-g-10;
  a=180-a+5;
  b=b+5;
  LB0.write(g);
  LB1.write(a);
  LB2.write(b);
                                        //if out of range
  if(g>180||g<0||a>180||a<0||b>180||b<0){
                                        //  Serial.println("LB error");
  }
}
                                        //move the joints in the LF servo
                                        to angles g, a, and b
void moveServosLF(int g, int a, int b){
                                        //The following calibrations were
                                        determined experimentally to get
                                        the angles to line up with the IK
                                        equations
  g=180-g+5;
  a=180-a-7;
  b=b+15;
  LF0.write(g);
  LF1.write(a);
  LF2.write(b);
                                        //if out of range
  if(g>180||g<0||a>180||a<0||b>180||b<0)
  {
                                        //  Serial.println("RB error");
    delay(1) ;
  }
                                        //  Serial.println("LF error");
}
```

3.2 ส่วนควบคุม



รูปที่ 3.3 การต่อวงจรของส่วนควบคุม

ส่วนการควบคุมเป็นการรับค่าจาก flex sensor และ accelerometer นำเข้าบอร์ด arduino แล้วส่งค่า ผ่าน wireless เพื่อไปควบคุมตัวหุ่นยนต์

3.2.1 Flex sensor

ต่ออุปกรณ์เข้า pin A0 กับ GND ในส่วนการเขียนโปรแกรมเก็บค่า flex sensor ด้วยคำสั่ง analogRead โดยเก็บจาก pin A0 เข้าเป็นตัวแปรประเภท int ชื่อ flex_1

ตัวอย่าง ::

```
int flex_1 = analogRead(A0);
```

3.2.2 Accelerometer

ต่ออุปกรณ์เข้า pin A1, A2 และA3 ในโปรแกรมตั้ง pin ให้เป็น

```
A1 = xPin
A2 = yPin
A3 = zPin
```

เก็บค่า accelerometer ด้วยคำสั่ง analogRead โดยเก็บจาก pin เข้าเป็นตัวแปรประเภท int ชื่อ xRead, yRead และzRead ตามลำดับ

ตัวอย่าง ::

```
int xRead = analogRead(xPin);
int yRead = analogRead(yPin);
int zRead = analogRead(zPin);
```

ต้องการใช้ค่าที่อ่านเป็นดีกรี แปลงค่าโดยใช้คำสั่ง map กำหนดค่า min กับ max เป็น -90 กับ 90 ตามลำดับ

```
int xAng = map(xRead, minVal, maxVal, -90, 90);
int yAng = map(yRead, minVal, maxVal, -90, 90);
int zAng = map(zRead, minVal, maxVal, -90, 90);
```

3.2.3 Wireless module

ต่อ Wireless module เข้ากับ pin Rx และ Tx ของบอร์ด arduino

3.2.4 Codeโปรแกรม Arduino ของส่วนควบคุม

```
Const int xPin = 1;
Const int yPin = 2;
Const int zPin = 3;
Int minVal = 265;
Int maxVal = 402;
Double x;
Double y;
Double z;
Int flex_1 = 0;
Void setup() {
        Serial.begin(9600);
                }
Void loop(){
        Int flex_1 = analogRead(A0);
        Int xRead = analogRead(xPin);
        Int yRead = analogRead(yPin);
        Int zRead = analogRead(zPin);
        Cheak_system();
        Deley(100):
}
Void check_system()
{
Int flex_1 = analogRead(A0);
Int xRead = analogRead(xPin);
Int yRead = analogRead(yPin);
Int zRead = analogRead(zPin);
Int xAng = map(xRead, minVal, maxVal, -90, 90);
```

```
Int yAng = map(yRead, minVal, maxVal, -90, 90);
Int zAng = map(zRead, minVal, maxVal, -90, 90);
        X = Rad_to_deg * (atan2(-yAng, -zAng)+Pl);
        Y = Rad_to_deg * (atan2(-xAng, -zAng)+Pl);
        Z = Rad_to_deg * (atan2(-yAng, -xAng)+Pl);
If( flex_1 <= 700)
        {
Serial.println("0");
        }
Else if( flex_1 >=900)
        {
Serial.println("u");
        }
Else
        {
If(x>20)&&(x<60))
        {
Serial.println("1");
        }
Else if(x>60)&&(x<100))
        {
Serial.println("2");
        }
Else if(x>320)&&(x<350))
        {
Serial.println("3");
        }
Else if(x>280)&&(x<320))
        {
Serial.println("4");
        }
Else if(y>15)&&(y>80))
        {
Serial.println("5");
```

```
                }
        Else if(y>300)&&(y<340))
                {
        Serial.println("6");
                }
        Else
                }
        Serial.println("0");
                }
        }
Serial.println("  x:");
Serial.println(x);
Serial.println("  y:");
Serial.println(y);
Serial.println("  z:");
Serial.println(z);
Serial.println(flex_1);
delay(50);
}
```

3.1 ตารางการกำหนดค่า Accelerometer

ตารางที่ 3.1 ตารางการกำหนดตัวแปรจากโค้ด

ช่วงค่าของ X และ Y	ค่าอนาล็อคจากFlex sensor		
	700<Flex<900	Flex = 700	Flex>=900
20<X<60	1	0	u
60<X<100	2	0	u
320<X<350	3	0	u
280<X<320	4	0	u
15<Y<80	5	0	u
300<Y<340	6	0	u
Else	0	0	u

บทที่ 4
สรุปผลการทดลอง

จากการทดลองเดินเครื่องหุ่นยนต์บนพื้นราบเรียบ พบว่าหุ่นยนต์เดินหน้า เลี้ยว และถอยหลังได้ การเดินต้องให้ขาทั้ง 4 ข้าง ทำงานครบรอบก่อนจึงจะเปลี่ยนไปยังคำสั่งต่อไปได้ เช่น การสั่งให้เดินถอยหลังในขณะที่เดินหน้าอยู่ จะต้องรอให้ขาหุ่นทำงานครบรอบก่อน(ทำงานจนถึงขาที่ 4 เป็นข้างสุดท้าย)จึงจะเริ่มเดินถอยหลังได้ โครงงานยังไม่สามารถปรับเปลี่ยนความเร็วการเดินของหุ่นได้ ทำได้เพียงปรับเปลี่ยนทิศทางการเคลื่อนที่ได้เท่านั้น สามารถเดินบนระนาบลาดเอียงได้ แต่ไม่สามารถเดินบนพื้นที่เปลี่ยนระดับแบบฉับพลัน เช่น บันได เนื่องจากการก้าวขาของตัวหุ่นมีระดับตายตัวไม่สามารถเปลี่ยนแปลงตามระดับพื้นผิวได้เลย

จากผลการทดลองถือว่าประสบความสำเร็จตามที่วางไว้ และพบว่าตัวหุ่นยนต์ยังพัฒนาความสามารถต่อไปได้ในส่วนต่างๆเพิ่มขึ้นอีกและยังนำไปประยุกต์กับอุปกรณ์ต่างๆ เพื่อใช้งานในลักษณะที่แตกต่างกันได้ เช่น กล่องวีดีโอ เซ็นเซอร์วัดความกดอากาศ เซ็นเซอร์ตรวจจับกัมมันตรังสี เป็นต้น

เอกสารอ้างอิง

[1] รศ.ดร.สมเกียรติ ศุภเดช. **เซมิคอนดัคเตอร์ดีไวซ์.** คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง. กรุงเทพฯ. พ.ศ.2551

[2] ผศ.ดร.เติมพงษ์ เพ็ชรกูล. **อุปกรณ์สารกึ่งตัวนำ.** คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง. กรุงเทพฯ. พ.ศ.2544

[3] บัญชา ปะสีละเตสัง. **Visual Basic 2008.** กรุงเทพฯ. พ.ศ.2551

[4] http://en.wikipedia.org/wiki/Accelerometer

[5] http://th.wikipedia.org

[6] http://www.thaieasyelec.com/Sensors/Accelerometers/3-Axis

[7] http://arduino.cc/en/Tutorial/ADXL3xx#.UySGvfl_si0