สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หุ่นยนต์ 6 ขา

HEXRAPOD

โดย

นายด๊าส ณ สงขลา

ส/ว.
ว 454 ๗๖
2551
เลขหมู่.....
เลขทะเบียน 104026
วัน,เดือน,ปี 2 8 ต.ค. 2552

b 12109393
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาอิเล็คทรอนิคส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

หุ่นยนต์ 6 ขา

HEXRAPOD

โดย

นายด๊าส ณ สงขลา 48010291

อาจารย์ที่ปรึกษา

ผศ.ดร.ยุทธนา คิดใจเดียว

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

ปริญญานิพนธ์ ปีการศึกษา 2551

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ 6 ขา

HEXRAPOD

ผู้จัดทำ

1. นายด๊าส ณ สงขลา

.....อาจารย์ที่ปรึกษา

(ผศ.ดร.ยุทธนา คิดใจเดียว)

หุ่นยนต์ 6 ขา

นายด๊าส ณ สงขลา รหัส 48010291
ผศ.ดร.ยุทธนา คิดใจเดียว อาจารย์ที่ปรึกษา

ปีการศึกษา 2551

บทคัดย่อ

รายงานฉบับนี้นำเสนอการควบคุมหุ่นยนต์เดิน 6 ขา ซึ่งใช้การทำงานของ State Transition Machine โดยเริ่มจากโปรแกรมคำสั่งซึ่งถูกโปรแกรมลงในหน่อยความจำ เพื่อเป็นทิศทางให้กับหุ่นยนต์ จากนั้นหุ่นจะตรวจสอบ สถานะของขาจากการเดินครั้งล่าสุดของแต่ละขา เพื่อตัดสินใจในการเดินสถานะต่อไป ซึ่งสถานะ ต่อไปของแต่ละขาจะเปลี่ยนเป็นสัญญาณพัลส์ที่มีพัลส์วิธด์ต่างกันมาใช้ในการขับมอเตอร์ให้ หุ่นยนต์เคลื่อนที่ไปในทางของภาพที่ได้รับมา

การออกแบบการเดินของหุ่นยนต์ 6 ขานี้สามารถนำไปพัฒนารูปแบบการเดินได้โดยไม่ต้อง เปลี่ยนโครงสร้างของโปรแกรม

HEXRAPOD

Mr. Das Na Songkhla ID 48010291

Assistant Prof.Dr. Yuttana

Kidjaideaw (Advisor)

Educational Year 2007

Abstract

This project presents HEXRAPOD controlling via the State Transition Machine. Firstly program from memory will control robot. Next, the robot checks the previous state of each leg. Finally, the robot sends pulse-width signal to drive motors moving to the specified object.

The design of HEXRAPOD has an advantage of easy implement without the need of rewritten the program structure.

กิตติกรรมประกาศ

รายงานฉบับนี้สามารถลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือจากหลายบุคคล โดยเฉพาะอย่างยิ่ง ผศ.ดร.ยุทธนา คิดใจเดียว ที่คอยให้คำปรึกษา และคำแนะนำ อีกทั้งอาจารย์ภาควิศวกรรมการวัดคุมที่เอื้อเฟื้อสถานที่และเครื่องมือ ขอขอบพระคุณ คุณพ่อ คุณแม่ที่เป็นกำลังใจ และให้การสนับสนุนในทุกๆด้าน โดยเฉพาะทางด้านค่าใช้จ่าย ขอบคุณเพื่อนๆ ทุกคนที่คอย ช่วยเหลือในการปฏิบัติงานจนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

จึงขอขอบคุณมา ณ ที่นี้

ผู้จัดทำ

สารบัญ

บทคัดย่อไทย

บทคัดย่ออังกฤษ

กิตติกรรมประกาศ

สารบัญ

สารบัญรูป

สารบัญตาราง

บทที่ 1 บทนำ 1

บทที่ 2 ทฤษฎีการออกแบบโครงสร้าง 3

2.1 เซอร์โวมอเตอร์ 3

2.1.1 หลักการทำงานของเซอร์โวมอเตอร์ 3

2.1.2 ภาคการทำงานของเซอร์โวมอเตอร์ 5

2.2 แบบขา 8

2.2.1 ขาสองข้อต่ออย่างง่าย (Simple Two Link Leg) 9

2.3 การออกแบบและการสร้าง 10

บทที่ 3 ทฤษฎีและหลักการทำงาน 13

3.1 ทฤษฎีและบอร์ดควบคุมเซอร์โวมอเตอร์ 13

3.1.1 บอร์ดควบคุมเซอร์โวมอเตอร์ 13

3.1.2 คุณสมบัติของบอร์ด ET-dsPIC30F2010 TRAINING KIT V1.0 EXP 13

3.1.3 คุณสมบัติของ Peripheral I/O 15

3.1.4 ขั้วต่อ DOWNLOAD (ET-EMPIC) 18

3.2 คุณสมบัติเด่นโดยรวมของ dsPIC 18

3.2.1 คุณสมบัติเด่นของซีพียู 19

3.2.2 คุณสมบัติด้านการประมวลสัญญาณดิจิตอล 19

3.2.3 คุณสมบัติของโมดูลฟังก์ชั่นพิเศษ 20

3.3 การพัฒนาโปรแกรมของบอร์ด 20

3.3.1 การเขียนโปรแกรมใช้งานกับบอร์ดโดยใช้ MPLAB C30 21

3.4 การใช้งานโมดูลเปรียบเทียบข้อมูลใน dsPIC เพื่อสร้างสัญญาณ PWM 23

3.4.1 คุณสมบัติโดยสรุปของโมดูลปรียบเทียบข้อมูล (Output Compare) 23

3.4.2 โหมดการทำงานของโมดูลเปรียบเทียบข้อมูล 23

สารบัญ (ต่อ)

3.4.3 รีจิสเตอร์หลักของโมดูลเปรียบเทียบข้อมูล 24

3.4.3.1 OCxR รีจิสเตอร์หลักสำหรับเก็บข้อมูลของโมดูลเปรียบเทียบข้อมูล 24

3.4.3.2 OCxRS รีจิสเตอร์หลักสำหรับเก็บข้อมูลของโมดูลเปรียบเทียบข้อมูลตัวที่สอง 24

3.4.3.3 OCxCON เป็นรีจิสเตอร์ควบคุมการทำงานของโมดูลเปรียบเทียบข้อมูล 24

3.4.4 การกำเนิดสัญญาณ PWM ของโมดูลเปรียบเทียบข้อมูล 25

3.4.5 การคำนวนเกี่ยวกับคาบเวลาของสัญญาณ PWM (Tpwm) 26

3.5 ทฤษฎี State Transition Machine 27

3.6 ตารางอินพุท 32

3.7 สเตจการเดินของหุ่นยนต์ 33

บทที่ 4 วงจรอิเล็กทรอนิกส์และการทำงาน 34

4.1 ส่วนของวงจรทำงานหลัก 34

4.2 วงจรหลบหลีกสิ่งกีดขวาง 35

บทที่ 5 ขั้นตอนและโปรแกรมการทำงาน 37

5.1 Table การทำงานของหุ่นยนต์ 37

5.2 Flow Chart การทำงานของหุ่นยนต์ 38

5.3 ขั้นตอนการทำงานสำหรับเลือกหา Stage (t) ที่เหมาะสมที่สุด 39

5.4 ตัวอย่างการเลือก Stage (t) ที่เหมาะสม 40

บทที่ 6 การทดลองและผลการทดลอง 42

6.1 การทดลองจ่ายพัลส์ให้กับมอเตอร์ 42

6.2 การทดลองจ่ายสัญญาณพัลส์ให้กับเซนเซอร์ 43

บทที่ 7 วิเคราะห์และสรุปผลการดำเนินโครงงาน 47

สารบัญรูป

รูปที่ 1.1 แสดงภาคการทางานของหุ่นยนต์เดิน 6 ขา 2
รูปที่ 2.1 แสดงการตอบสนองของเซอร์โวมอเตอร์ต่อสัญญาณพัลส์ในความถี่ที่ต่างกัน 5
รูปที่ 2.2 แสดงภาคการทำงานของเซอร์โวมอเตอร์ 6
รูปที่ 2.3 วงจรภาคขยายเซอร์โว (SERVO AMP) และภาคขับเซอร์โว (SERVO DRIVE) 7
รูปที่ 2.4 เซอร์โวมอเตอร์พร้อมเฟือง 8
รูปที่ 2.5 ส่วนประกอบของเซอร์โวมอเตอร์ 8
รูปที่ 2.6 แสดงลักษณะของขาข้อต่ออย่างง่าย 9
รูปที่ 2.7 แผ่นลำตัวของหุ่นยนต์ 11
รูปที่ 2.8 ส่วนประกอบส่วนขาของหุ่นยนต์ 12
รูปที่ 2.9 ส่วนประกอบเฟืองของหุ่นยนต์ 12
รูปที่ 3.1แสดงการจัดเรียงขาสัญญาณของ dsPIC30F2010 16
รูปที่ 3.2 โครงสร้างบอร์ด ET-dsPIC30F2010 TRAININGT KIT V1.0 / EXP 17
รูปที่ 3.3 โครงสร้าง ET-EMPIC 18
รูปที่ 3.4 Moore Machine 27
รูปที่ 3.5 รูปแบบการเดินไปข้างหน้า 28
รูปที่ 3.6 รูปแบบการถอยหลัง 29
รูปที่ 3.7 รูปแบบการเดินไปข้างขวา 29
รูปที่ 3.8 รูปแบบการเดินไปข้างซ้าย 30
รูปที่ 3.9 แสดงสถานะขาหุ่นในสภาวะต่างๆ 30
รูปที่ 3.10 ตำแหน่งของ Infrared Sensor 32
รูปที่ 3.11 สเตจการเดินของหุ่นยนต์ 33
รูปที่ 4.1 วงจรควบคุมการทำงานหลัก 34
รูปที่ 4.2 วงจรตัวรับสัญญาณอินฟราเรด 35
รูปที่ 4.3 TSOP1738 IR Sensor Modules 36
รูปที่ 5.1 Flow Chart การทำงานของหุ่นยนต์ 38
รูปที่ 5.2 ขั้นตอนการทำงานสำหรับเลือกหา Stage (t) ที่เหมาะสมที่สุด 39
รูปที่ 6.1 สัญญาณพัลส์ 1.5 ms 44
รูปที่ 6.2 สัญญาณพัลส์ 1.25 ms 44

สารบัญรูป (ต่อ)

รูปที่ 6.3 สัญญาณพัลส์ 1.75 ms 45

รูปที่ 6.4 สัญญาณพัลส์ความถี่ 38 KHz 45

รูปที่ 6.5 สัญญาณเมื่อเซนเซอร์ไม่ตรวจพบวัตถุ 46

รูปที่ 6.6 สัญญาณเมื่อเซนเซอร์ตรวจพบวัตถุ 46

สารบัญตาราง

ตารางที่ 3.1 แสดงความแตกต่างของ dsPIC เบอร์ต่างๆ 15
ตารางที่ 3.2 ตารางแสดงค่าการเดินหน้าของหุ่นยนต์ 31
ตารางที่ 3.3 ตารางแสดงค่าการถอยหลังของหุ่นยนต์ 31
ตารางที่ 3.4 ตารางแสดงค่าการเดินไปทางขวาของหุ่นยนต์ 31
ตารางที่ 3.5 ตารางแสดงค่าการเดินไปทางซ้ายของหุ่นยนต์ 32
ตารางที่ 3.6 แสดงค่าอินพุตของ Sensor 33
ตารางที่ 5.1 Table การทำงานของหุ่นยนต์ 38
ตารางที่ 5.2 ตัวอย่างการเลือก Stage (t) ที่เหมาะสม 40

บทที่ 1
บทนำ

ปัจจุบันได้มีการสร้างและพัฒนา หุ่นยนต์ในรูปแบบต่างๆ มากยิ่งขึ้น เนื่องจากความต้องการหุ่นยนต์ที่สามารถทำการประมวลผลได้อย่างรวดเร็ว และมีความใกล้เคียงกับสิ่งมีชีวิต จึงได้มีการนำเอาเครือข่ายปัญญาประดิษฐ์มาประยุกต์ใช้งาน อาทิเช่นการประมวลผลสัญญาณ การสังเคราะห์เสียง การจดจำเสียง การจดจำรูปแบบ และระบบควบคุม เป็นต้น

จึงได้มีแนวคิดที่จะสร้างและพัฒนาหุ่นยนต์ ที่สามารถเคลื่อนที่โดยใช้ขา เป็นตัวขับเคลื่อน โดยได้เลือกรูปแบบขาที่ใช้งาน เพื่อใช้ในการพยุงและขับเคลื่อนตัวหุ่นยนต์ และควบคุมหุ่นยนต์ด้วยโปรแกรม โดยใช้หลัก State Transition Machine เพื่อจะสามารถพัฒนาในส่วนของโปรแกรมต่อไปได้โดยง่าย ซึ่งหุ่นยนต์ที่ขับเคลื่อนด้วยขานี้ สามารถที่จะพัฒนาให้เป็นหุ่นยนต์ที่ใช้งานแทนมนุษย์ได้ในจุดที่เป็นอันตรายและเสี่ยง ต่อการทำงาน เช่น หุ่นยนต์ทำความสะอาดกระจกบนตึกสูง หุ่นยนต์ตรวจสอบและตรวจซ่อมโครงสร้างสะพานในที่สูง หุ่นยนต์ตรวจสอบรอยรั่วบนถังก๊าซ ระบบส่งจ่ายก๊าซ หรือ ปิโตรเคมีเป็นต้น

ชื่อโครงงาน

หุ่นยนต์ 6 ขา (HEXRAPOD)

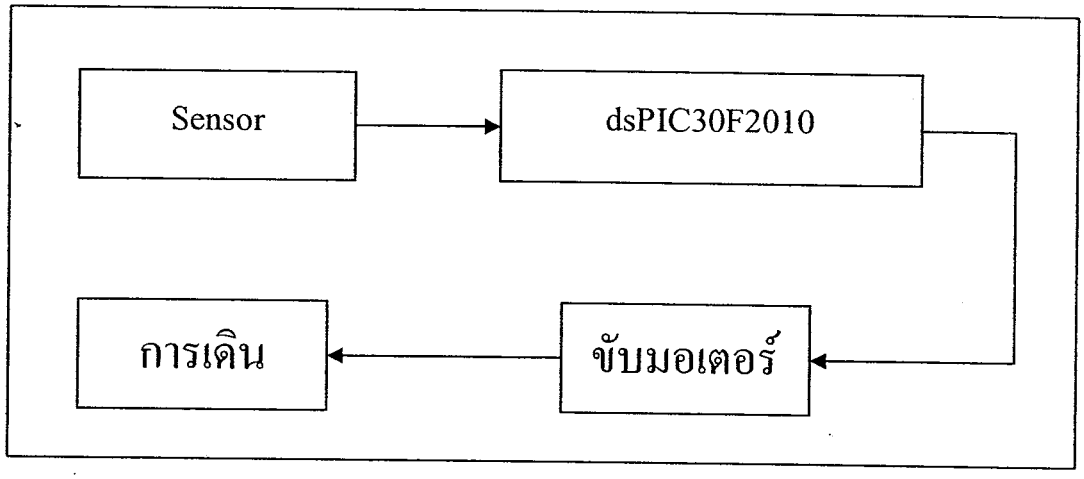
วัตถุประสงค์

- เพื่อการเรียนรู้หลักการทำงานของระบบควบคุมแบบ State Transition Machine
- เพื่อการเรียนรู้การทำงานของไมโครคอนโทรลเลอร์ ซึ่งใช้ภาษาซีเป็นคำสั่งการทำงาน
- เพื่อออกแบบและสร้างหุ่นยนต์ 6 ขาขนาดเล็กซึ่งสามารถติดตามวัตถุที่กำหนดได้ เพื่อเป็นหุ่นยนต์ต้นแบบ ในการพัฒนาหุ่นยนต์ต่อไป

คุณสมบัติ

โครงงานหุ่นยนต์เดิน 6 ขา ในส่วนควบคุมหุ่นยนต์ประกอบด้วย

- ส่วนควบคุมหรือสมองที่ใช้ไมโครคอนโทรลเลอร์เป็นหัวใจหลัก
- มีระบบควบคุมการทำงานเป็นแบบ State Transition Machine
- ส่วนโครงสร้างของหุ่นเป็นพลาสติกอะครีลิค
- ส่วนขับเคลื่อนโดย DC SERVO FUTABA – S3003 จำนวน 12 ตัว ในการควบคุมการเคลื่อนไหวของขาทั้ง 6 ขา
- Block Diagram ของระบบ



รูปที่ 1.1 แสดงภาคการทางานของหุ่นยนต์เดิน 6 ขา

ขอบเขตของการทำงาน

- สร้างหุ่นยนต์ที่สามารถเดิน 6 ขาได้
- หุ่นยนต์สามารถเดินตามวัตถุที่กำหนดได้

บทที่ 2
ทฤษฎีการออกแบบโครงสร้าง

2.1 เซอร์โวมอเตอร์

2.1.1 หลักการทำงานของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์ ประกอบด้วย มอเตอร์ความเร็วสูง ภายในมีเฟืองทดรอบ ให้หมุนช้าลง เพื่อจะได้มีกำลังแรงบิดที่สูงขึ้น นอกจากนี้ยังมีวงจรควบคุมมอเตอร์ ซึ่งวงจรนี้จะนำค่าแรงดันเฉลี่ยของพัลส์รูปสี่เหลี่ยม เข้าไปเปรียบเทียบกับค่าแรงดันค่าหนึ่งที่มีอยู่ในวงจร ถ้าค่าต่างกัน วงจรควบคุมจะสั่งให้มอเตอร์หมุนไปตามทิศทาง ซึ่งขึ้นอยู่กับขนาดความกว้างพัลส์ โดยที่แกนเฟืองทดรอบจะถูกพ่วงไปขับแกนของตัวต้านทานปรับค่าได้(Potentiometer) ซึ่งอยู่ในวงจรควบคุมมอเตอร์ ในขณะที่มอเตอร์หมุนตัวต้านทานปรับค่าได้ จะถูกปรับค่าทำให้ค่าแรงดันเปรียบเทียบของวงจรควบคุมมอเตอร์เปลี่ยนไปด้วย จนกระทั่งค่าเฉลี่ยของพัลส์ในวงจรควบคุมมอเตอร์ เท่ากับค่าเฉลี่ยของ พัลส์ที่เข้ามา จึงทำให้มอเตอร์หยุดหมุนได้

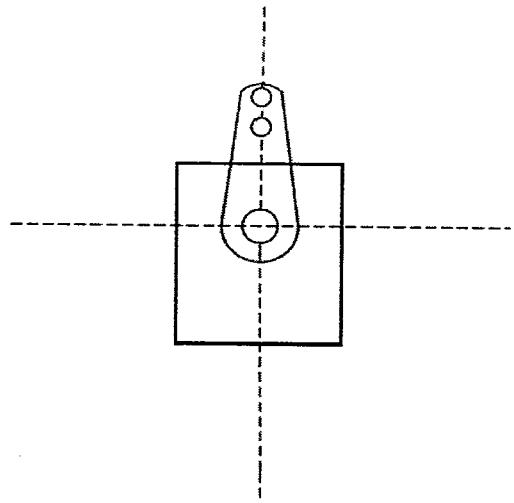
เซอร์โวมอเตอร์จะมีสายไฟ 3 เส้นคือ สายไฟเลี้ยง สายกราวด์ และสายสัญญาณพัลส์ควบคุม

ซึ่งลักษณะของสัญญาณ พัลส์ที่ใช้ควบคุมตำแหน่งของเซอร์โวมอเตอร์ จะเป็นการส่งพัลส์ที่มีความกว้างต่างกัน เพื่อควบคุมให้เซอร์โวมอเตอร์ หมุนไปยังตำแหน่งที่ต้องการ โดยที่มีความกว้างของพัลส์ จะเป็นตัวกำหนดขนาด และทิศทางของการหมุนแกนเซอร์โวมอเตอร์สำหรับคาบเวลา หรือระยะห่างระหว่างพัลส์แต่ละลูก จะเป็นตัวกำหนดแรงบิดของมอเตอร์

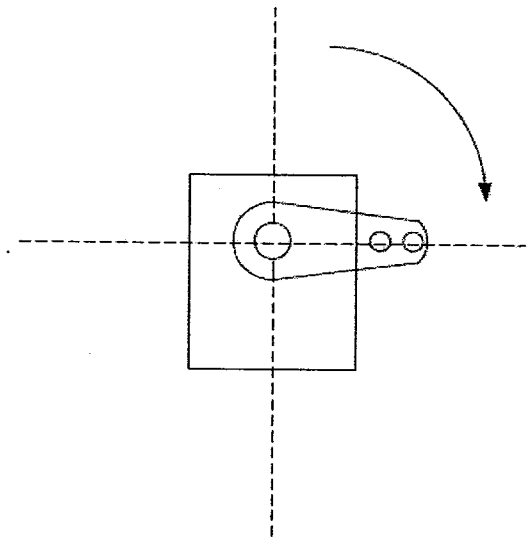
ถ้ากำหนดให้ในสภาวะปกติ เมื่อป้อนพัลส์ สี่เหลี่ยม ที่มีความกว้างขนาด 1.5 ms ให้กับเซอร์ โวมอเตอร์ แขนของเซอร์โวมอเตอร์จะอยู่ตำแหน่งกลาง

เมื่อป้อนพัลส์ สี่เหลี่ยม ที่มีความกว้างขนาด 1 ms ให้กับเซอร์โวมอเตอร์ แขนของเซอร์โวมอเตอร์จะหมุนตามเข็มนาฬิกา

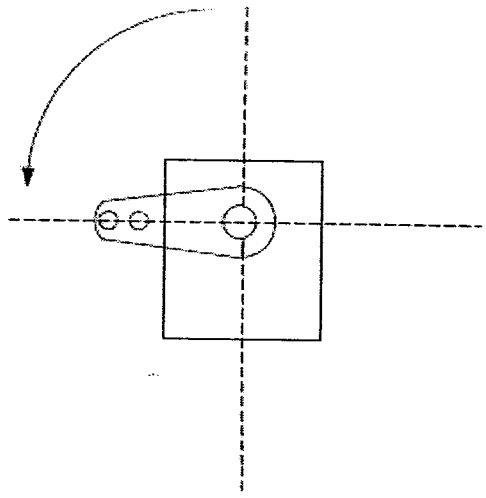
เมื่อป้อนพัลส์ สี่เหลี่ยม ที่มีความกว้างขนาด 2 ms ให้กับเซอร์โวมอเตอร์ แขนของเซอร์โวมอเตอร์จะหมุนทวนเข็มนาฬิกา ตามที่แสดงไว้ในรูปที่ 2.1 ดังต่อไปนี้



ก. แสดงการตอบสนองของเซอร์โวเมื่อจ่ายพัลส์ขนาด 1.4 ms



ข. แสดงการตอบสนองของเซอร์โวเมื่อจ่ายพัลส์ขนาด 1 ms



ค. แสดงการตอบสนองของเซอร์โวเมื่อจ่ายพัลส์ขนาด 2 ms

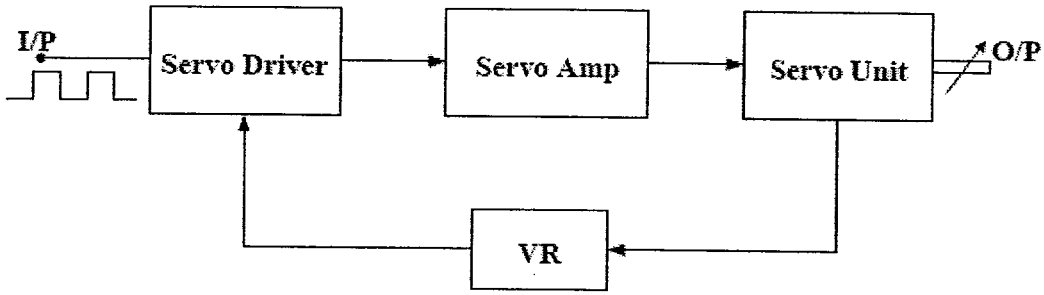
รูปที่ 2.1 แสดงการตอบสนองของเซอร์โวมอเตอร์ต่อสัญญาณพัลส์ในความถี่ที่ต่างกัน

ดังนั้นถ้าจ่ายสัญญาณพัลส์ที่มีความกว้างมากหรือน้อยกว่าความกว้างของพัลส์ 1.5 ms ก็ จะทำให้เซอร์โวมอเตอร์หมุนต่างทิศกัน ทั้งตามเข็มและทวนเข็มนาฬิกา โดยตำแหน่งของแขนหมุน เซอร์โวมอเตอร์จะเบี่ยงเบนออกจากจุดกึ่งกลางเป็นสัดส่วนกับความกว้างของพัลส์ที่จ่ายให้

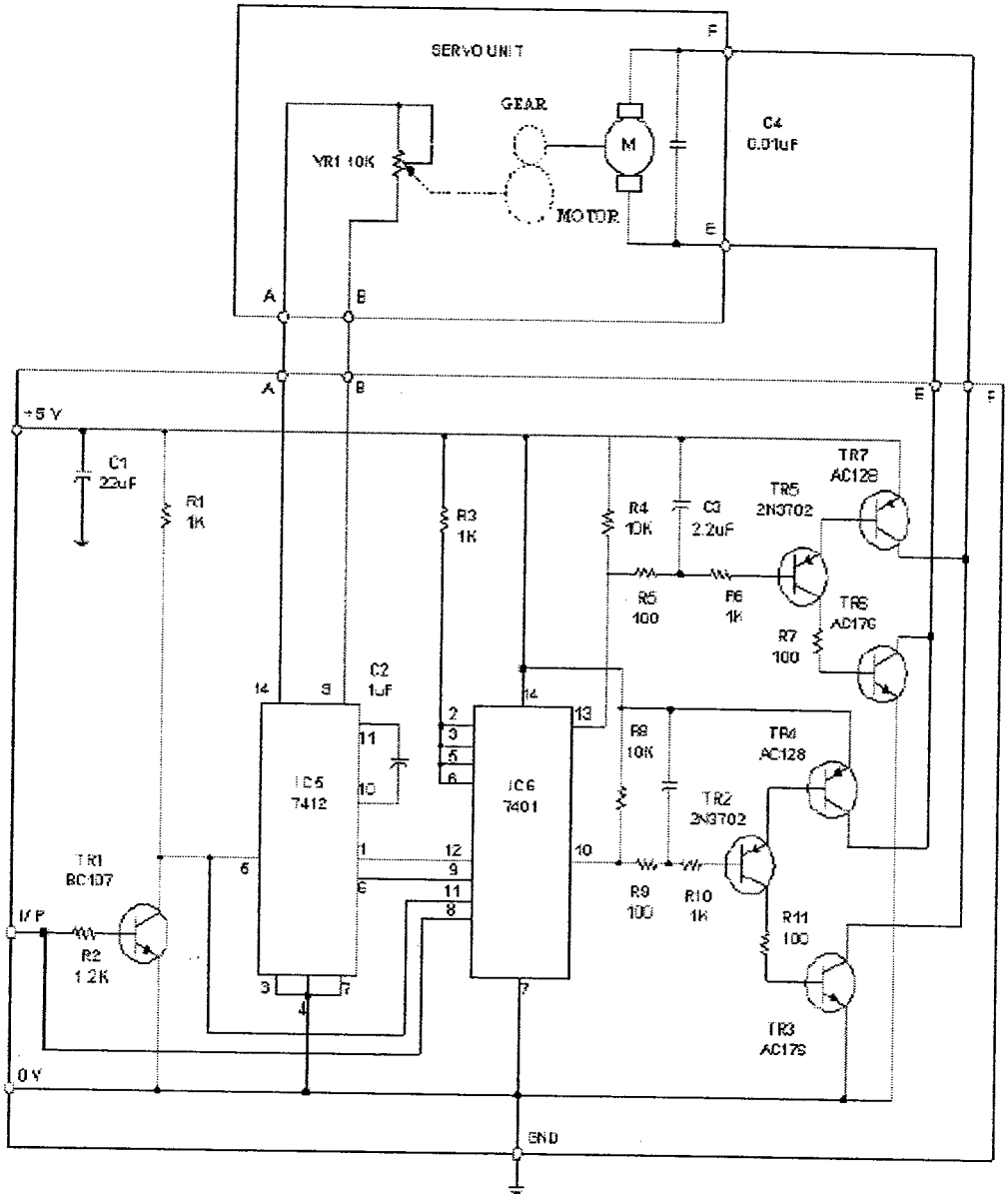
2.1.2 ภาคการทำงานของเซอร์โวมอเตอร์

ในเซอร์โวมอเตอร์หนึ่งตัวจะประกอบไปด้วย 3 ภาคการทำงานแต่ละภาคมีหน้าที่และ การทำงานดังนี้คือ ภาคขับเซอร์โว ประกอบด้วย วงจรสร้างสัญญาณพัลส์ และวงจรเปรียบเทียบสัญญาณพัลส์ที่สร้างขึ้น กับสัญญาณพัลส์ I/P ที่รับเข้ามา ภาคขยายเซอร์โว ประกอบด้วย วงจร RC Network ที่ช่วยหน่วงสัญญาณให้เซอร์โว สามารถทำงานได้ตลอดช่วงคาบเวลา จนกระทั่งมีสัญญาณลูกต่อไปมา รวมถึงวงจร กลับขั้วแรงดันไฟฟ้าควบคุมทิศทางการหมุนขอมอเตอร์ หน่วยเซอร์โว ประกอบด้วย มอเตอร์ความเร็วสูง เฟืองทดรอบ แกนหมุน อุปกรณ์ต่าง ๆ และตัวต้านทานปรับค่าได้ ทำหน้าที่ป้อนกลับตำแหน่ง (Position Feedback) ซึ่งในขณะที่มอเตอร์หมุนตัวต้านทานปรับค่าได้จะเปลี่ยนแปลงค่าซึ่งส่งผลให้ค่า ป้อนกลับเปลี่ยนแปลงไปจากนั้นนำค่าป้อนกลับมาปรับและเปรียบเทียบค่าความกว้างของพัลส์ ที่ภาคขับเซอร์โว เมื่อขนาดความกว้างของพัลส์มีค่าแรงดันเฉลี่ยเท่ากันมอเตอร์จะหยุดหมุน

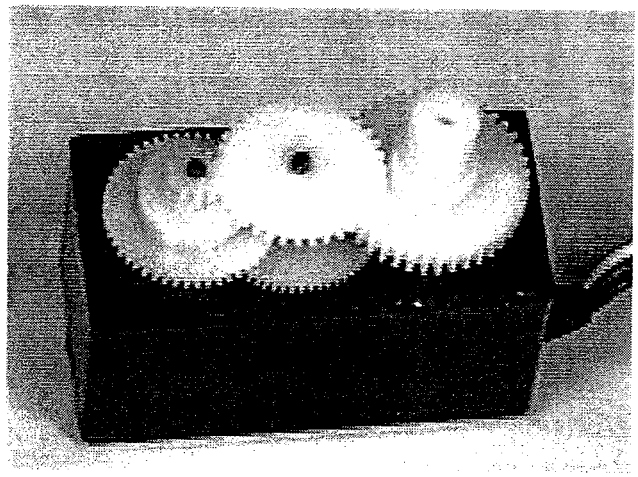
ทันที ซึ่งรูปที่ 2.2 ได้แสดงภาคการทำงานของเซอร์โวมอเตอร์ตามที่ได้กล่าวไว้ในข้างต้น และ ได้แสดงไว้ดังรูปต่อไปนี้



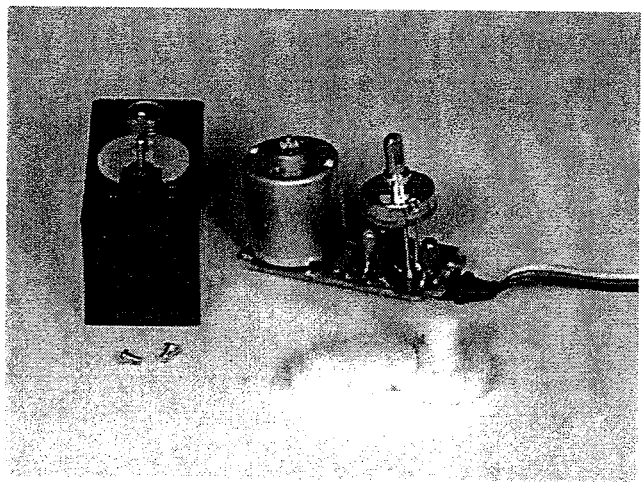
รูปที่ 2.2 แสดงภาคการทำงานของเซอร์โวมอเตอร์



รูปที่ 2.3 วงจรภาคขยายเซอร์โว (SERVO AMP) และภาคขับเซอร์โว (SERVO DRIVE)



รูปที่ 2.4 เซอร์โวมอเตอร์พร้อมเฟือง



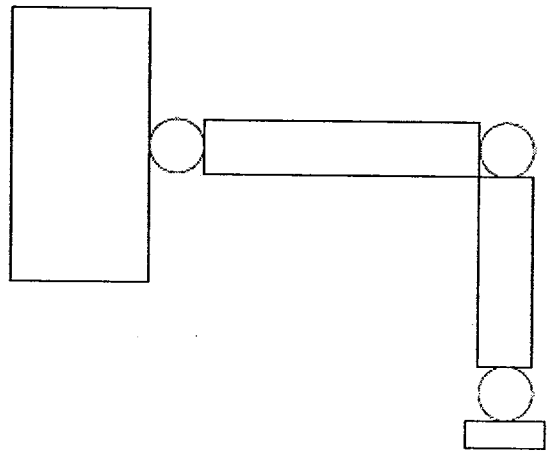
รูปที่ 2.5 ส่วนประกอบของเซอร์โวมอเตอร์

2.2 แบบขา

สำหรับการเลือกลักษณะขาของหุ่นยนต์ สิ่งที่สำคัญต่อการเลือกขาคือ จะต้องเลือกแบบ ที่มีลักษณะการเคลื่อนไหวมากที่สุด ให้เหมาะสมกับรูปร่างลักษณะโครงสร้างทางกายภาพของตัว หุ่นยนต์ ลักษณะการเดินของขาแต่ละแบบ จะขึ้นอยู่กับข้อจำกัดทางกายภาพของขาด้วย ซึ่งใน ปัจจุบันมีการออกแบบขาหุ่นยนต์อยู่หลายชนิด แต่ละชนิดจะมีคุณสมบัติเฉพาะแบบ ดังต่อไปนี้

2.2.1 ขาสองข้อต่ออย่างง่าย (Simple Two Link Leg)

ขาชนิดนี้จะมีลักษณะเป็นสองท่อน แต่ละท่อนจะต่อผ่านข้อต่อดังรูปที่ 2.6 ซึ่งสามารถ ควบคุมลักษณะการเดินได้ โดยการควบคุมของขาแต่ละท่อนซึ่งจะเป็นตัวกำหนดตำแหน่งปลายขา ของหุ่นยนต์ ส่วนของขาทั้งหมดจะต่อเข้ากับเดือยที่โคนขาเพื่อใช้ในการก้าวขา และยึด หดขา



รูปที่ 2.6 แสดงลักษณะของขาข้อต่ออย่างง่าย

วิธีการทำงานของข้อต่อ

มีหลายวิธีที่จะทำให้ข้อต่อทำงานได้ โดยใช้ลักษณะการขับเคลื่อนของข้อต่อสำหรับแบบ นี้ อาจติดตั้งมอเตอร์เข้าที่ข้อต่อโดยตรง หรือใช้โซ่ สายพาน สกรู และส่งกำลังจากมอเตอร์ที่ติด ตั้งอยู่ในลำตัวบริเวณโคนขา เพื่อกำหนดมุมที่ข้อต่อในการก้าวเดินของขาหุ่น

จุดด้อยประการสำคัญของขาแบบนี้คือ เราจำเป็นต้องใช้ตัวขับเคลื่อนอยู่ใกล้กับข้อต่อมาก ที่สุด การติดตั้งตัวขับเคลื่อนเข้าที่ข้อต่อเข่าทำให้เกิดผลประทบทางไดนามิคต่อขาหุ่น ซึ่งต้องมี การชดเชยโดยใช้ตัวควบคุม ซึ่งจะทำให้ต้องเพิ่มความซับซ้อนให้กับอัลกอริธึมในการเคลื่อนที่ของ ขา รวมทั้งยังต้องการมอเตอร์ที่มีกำลังสูงที่ข้อต่อส่วนสะโพก เพื่อใช้ในการเคลื่อนขาที่มีมวลมาก ซึ่งเราก็สามารถแก้ปัญหาเหล่านี้ได้โดยการติดตั้งตัวขับเคลื่อนที่ฐานของขาแต่จะเป็นการเพิ่มความ ซับซ้อนทางแมคคานิกส์

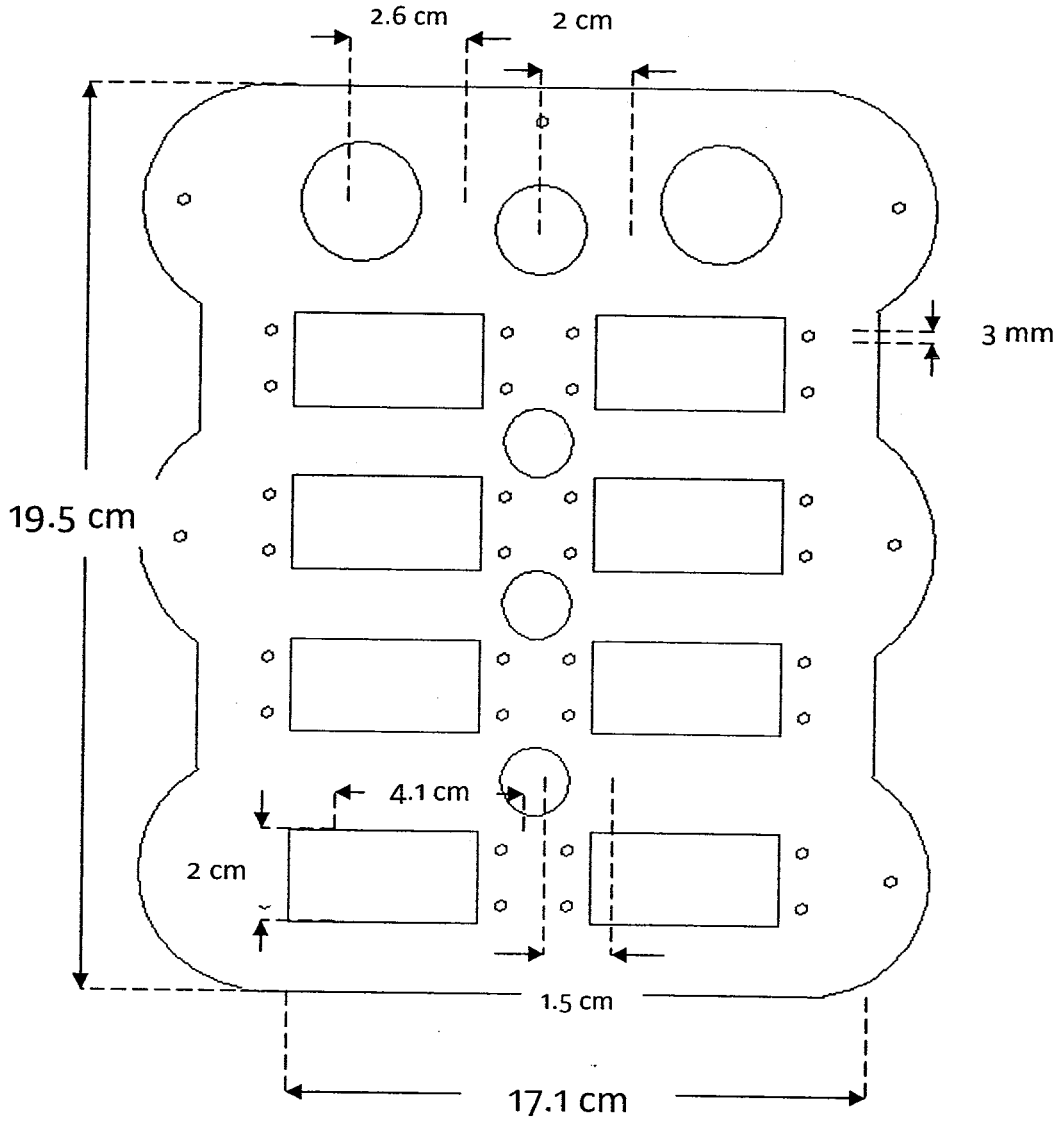
2.3 การออกแบบและการสร้าง

ในการออกแบบเลือกใช้เซอร์โวมอเตอร์ตัวหนึ่งทำหน้าที่ยกขาให้ลอยขึ้นและเซอร์โวมอเตอร์อีกตัวหนึ่งทำการหมุนขาเพื่อเคลื่อนย้ายตำแหน่งเท้าของหุ่นยนต์

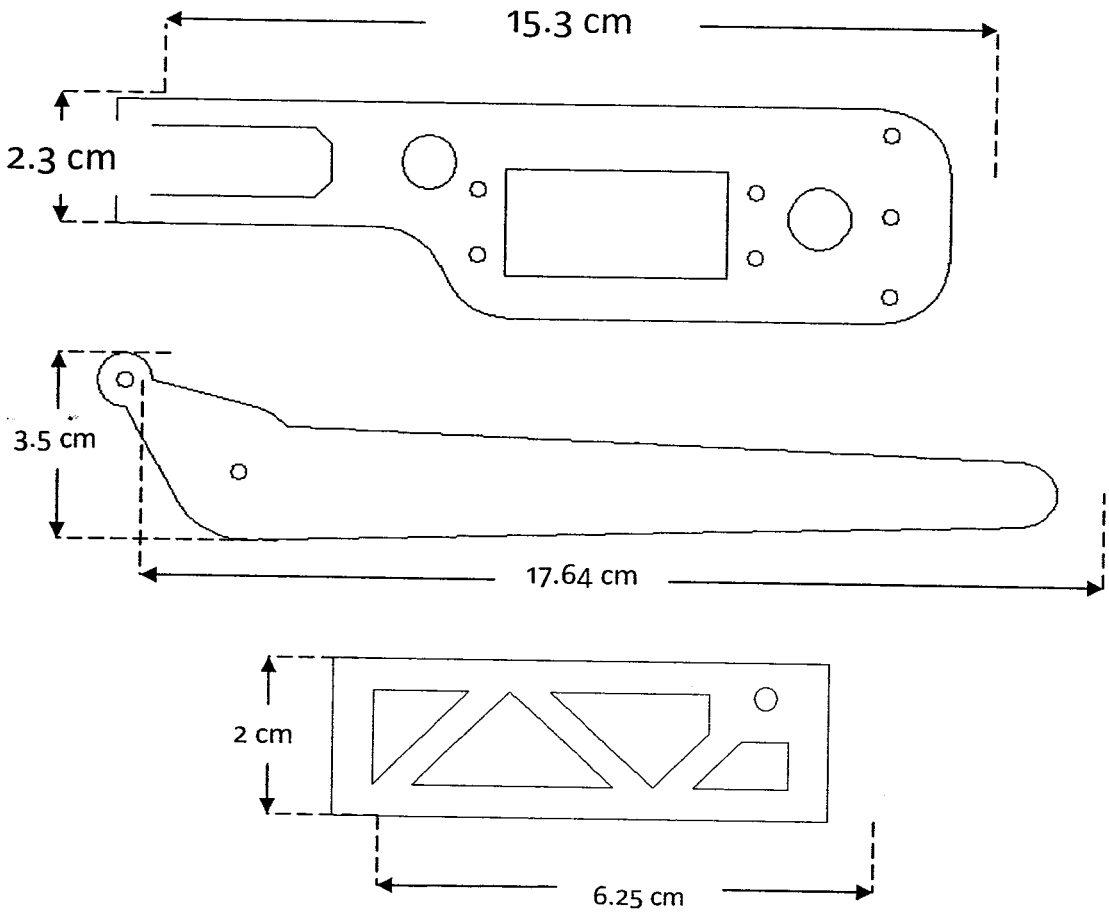
วัสดุที่ใช้สร้างแผ่นพลาสติกซิลิโคนจะนำมาสร้างเป็นตัวหุ่นและขาหุ่น เนื่องจากมีคุณสมบัติเหนียวทนทาน น้ำหนักเบา และตกแต่งง่าย ส่วนจุดหมุนเราใส่ลูกปืน แบริ่งขนาด 8 มิลลิเมตร เพื่อลดแรงเสียดทานของจุดหมุน

แผ่นลำตัวของหุ่นยนต์มีลักษณะเป็นสี่เหลี่ยม ทำหน้าที่ยึดขาทั้งหมดเข้าด้วยกันโดยจัดวาง ขาให้สมมาตรดังรูปที่ 2.10 ด้านบนของลำตัวมีพื้นที่สำหรับจัดวางอุปกรณ์ควบคุม ลำตัว ของหุ่นมีความกว้าง 17.1 เซนติเมตรและความยาว 19.5 เซนติเมตร (ไม่รวมความยาวของขา) ซึ่งแผ่นลำตัวของหุ้นยนต์ทั้งด้านบนและด้านล่างจะมีลักษณะเดียวกัน

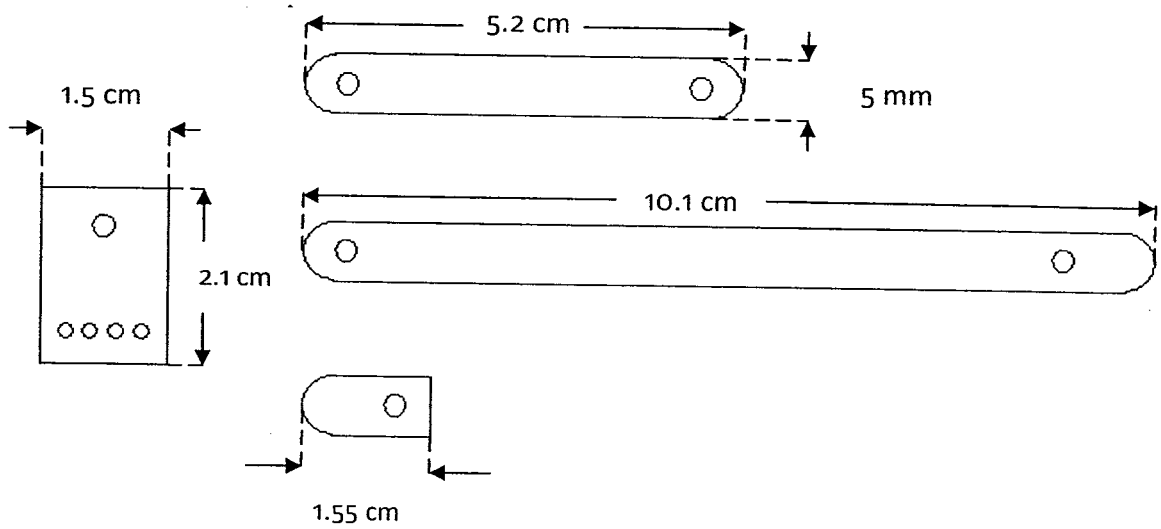
การยึดขาเข้ากับลำตัวหุ่นจะใช้แผ่นยึดลำตัวสองแผ่นประกบด้านบนและด้านล่างเพื่อ ยึดจุดหมุนส่วนขาให้มีความควบคุมคงที่ ช่องว่างกลาง ลำตัวระหว่างแผ่นยึดลำตัวทั้งสองสามารถวางแบตเตอรี่ เพื่อจ่ายกำลังงานไฟฟ้าให้กับส่วนควบคุม และส่วนขับเคลื่อนของหุ่นยนต์ได้



รูปที่ 2.7 แผ่นลำตัวของหุ่นยนต์



รูปที่ 2.8 ส่วนประกอบส่วนขาของหุ่นยนต์



รูปที่ 2.9 ส่วนประกอบเฟืองของหุ่นยนต์

บทที่ 3
ทฤษฎีและหลักการทำงาน

3.1 ทฤษฎีและบอร์ดควบคุมเซอร์โวมอเตอร์

3.1.1 บอร์ดควบคุมเซอร์โวมอเตอร์

เลือกใช้บอร์ดเบอร์ ET-dsPIC30F2010 TRAINING KIT V1.0 / EXP ซึ่งเป็นชุดทดลอง "dsPIC" ขนาดเล็ก โดยตัวบอร์ดได้รับการออกแบบให้สามารถรองรับการศึกษาเรียนรู้เกี่ยวกับสถาปัตยกรรมของไมโครคอนโทรลเลอร์ตระกูล dsPIC จากค่าย Microchips ในขั้นพื้นฐานได้อย่างครบถ้วน ภายใต้งบประมาณที่เรียกว่าประหยัดและคุ้มค่ามากที่สุด โดยตัวบอร์ดเลือกใช้ MCU เบอร์ dsPIC30F2010 เป็น CPU ประจำบอร์ด พร้อมทั้งจัดวงจรพื้นฐานอื่นๆที่จำเป็นรวบรวมไว้ภายในบอร์ด ไม่ว่าจะเป็นแหล่งจ่ายไฟ วงจรแสดงผลการทำงานของOutput แบบ LED ,วงจรสร้างสัญญาณ Input Logic แบบ Push-Button Switch ,วงจรสร้างสัญญาณ Input Analog แบบใช้ตัวต้านทาน ปรับค่าแรงดัน 0-5V ,วงจรกำเนิดเสียงด้วยลำโพงขนาดเล็ก ,วงจร LineDriver สำหรับพอร์ตสื่อสารอนุกรม RS232 และวงจร Download แบบ ICSP เป็นต้น

นอกจากนี้แล้วตัวบอร์ดยังได้ออกแบบให้มีพื้นที่สำหรับใช้ต่อทดลองวงจรขนาดเล็ก โดยมีทั้งส่วนที่เป็นลาย PCB แบบจุดไข่ปลา หรือ อาจติดตั้งแผง Photo Board ขนาด 360 จุดเพื่อใช้เป็นพื้นที่ต่อวงจรต่างเพิ่มเติมได้เองตามต้องการโดยใช้ชุด "ET-Hardware KIT" นอกจากนี้แล้วยังสามารถนำบอร์ด I/O ขนาดเล็ก ซึ่งทางทีมงาน อีทีที จัดเตรียมไว้เป็นทางเลือกและสนับสนุนการทดลอง ซึ่งเรียกว่าชุดโมดูลการทดลอง **"Mini I/O Module"** แบบต่างๆ ไม่ว่าจะเป็น ชุดทดลอง DC-Motor ,ชุดทดลอง Stepping Motor ,ชุดทดลอง I2C ต่างๆ ซึ่งมีทั้งชุดทดลอง I2C I/O PCF8574 ,ชุดทดลอง I2C EEPROM 24xx ,ชุดทดลอง I2C RTC เบอร์ DS1307 ,ชุดทดลอง I2C RTC เบอร์ PCF8583 ,ชุดทดลอง SPI โดยใช้ SPI Output เบอร์ 74HC595 ,ชุดทดลอง Matrix Keyboard ขนาด 4x4 ,ชุดทดลอง 7-Segment ฯลฯ

3.1.2 คุณสมบัติของบอร์ด ET-dsPIC30F2010 TRAINING KIT V1.0 EXP

บอร์ด ET-dsPIV30F2010 TRAINING KIT V1.0 EXP จะประกอบไปด้วย วงจรพื้นฐานที่จำเป็นสำหรับการศึกษาเรียนรู้และทดลองใช้งานทรัพยากรต่างๆของ MCU ตระกูล dsPIC โดยภายในบอร์ดได้จัดเตรียมวงจรใช้งานที่จำเป็นไว้ให้ใช้งานอย่างครบถ้วนได้แก่

- ใช้ dsPIC30F2010 เป็น MCU ประจำบอร์ด โดยขาสัญญาณต่างๆจาก MCU จะถูกจัดเรียงไว้อย่างเป็นระเบียบ ผ่าน "Pin Header" พร้อมลายพิมพ์ขาว แสดงชื่อขาสัญญาณอย่างชัดเจน เพื่อให้ง่ายและสะดวกต่อการต่อทดลองใช้งานขาสัญญาณต่างๆ โดยไม่ถูกจำกัดหน้าที่การใช้งานด้วยวงจรแบบใดแบบหนึ่ง ทำให้ผู้ใช้สามารถทดลองใช้งานฟังก์ชั่นภายในผ่านขาสัญญาณต่างๆ ของ MCU ได้โดยอิสระและครบถ้วนตามต้องการ
- วงจรแหล่งจ่ายไฟ แบบ Bridge Rectifier ขนาด 1A พร้อมวงจร Filter สามารถใช้กับแหล่งจ่ายไฟได้ทั้ง AC และ DC ขนาด 14-18V
- วงจร Regulate ขนาด +5V / 1A สำหรับใช้งานเป็นแหล่งจ่ายไฟเลี้ยงวงจรให้กับ MCU และอุปกรณ์การทดลองต่างๆ พร้อม LED แสดงสถานะสีแดง และจุด Connector เชื่อมต่อใช้งาน ทั้งตัวผู้และตัวเมีย
- วงจร Regulate ขนาด +13V / 100mA สำหรับใช้เป็นแรงดันโปรแกรม ในการ Download ข้อมูลแบบ HEX File ให้กับหน่วยความจำโปรแกรม (Flash) ของ MCU ภายในบอร์ด
- วงจร Download แบบ ICSP โดยใช้งานร่วมกับ "ET-CAP10P V2.0" สำหรับใช้พัฒนาโปรแกรมให้บอร์ด พร้อม LED แสดงสถานะสีเหลือง สามารถ Download HEX File จากคอมพิวเตอร์ PC ให้กับ MCU ในบอร์ดได้ทันที โดยไม่ต้องใช้เครื่องมือโปรแกรมอื่นๆอีก
- วงจร Line Driver สำหรับพอร์ตสื่อสารอนุกรม RS232 จำนวน 2 ช่อง โดยเชื่อมต่อกับสัญญาณ RC13(TXD) และ RC14(RXD) จำนวน 1 ช่อง ส่วนที่เหลืออีก 1 ช่อง จะปล่อยว่างไว้เป็นอิสระ เพื่อให้ผู้ใช้สามารถต่อทดลองการติดต่อสื่อสาร RS232 โดยใช้สัญญาณจาก Pin I/O อื่นๆของ dsPIC เช่น ต่อกับขาสัญญาณ RF2(RXD) และ RF3(TXD) ได้
- วงจร LED แสดงผลแบบ Sink Current ใช้ไฟเลี้ยง +5V โดยใช้ LED สีแดงขนาด 3mm. จำนวน 4 ชุด สำหรับใช้ในการทดสอบการทำงานของ Output ต่างๆ
- วงจรปรับแรงดัน 0-5V โดยใช้ตัวต้านทานปรับค่าได้แบบเกือกม้าแบบมีแกนปรับ จำนวน 4 ชุด สำหรับใช้ในการทดสอบการทำงานของ A/D
- วงจร Push-Button Switch จำนวน 4 ชุด สำหรับใช้ทดสอบการทำงานของ Input ต่างๆ

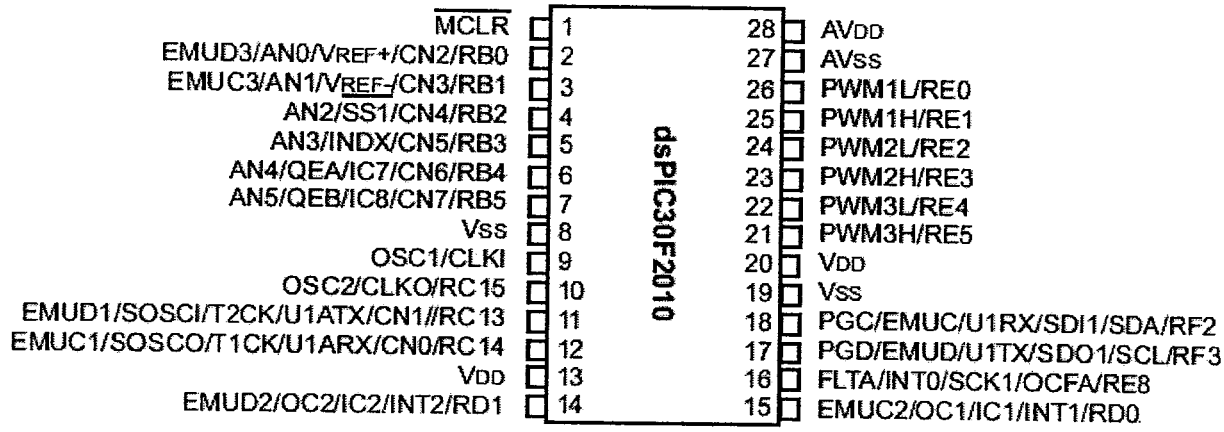
- วงจร Mini Speaker สำหรับใช้ทดสอบการกำเนิดเสียง Beep หรือเสียงอื่นๆ
- พื้นที่สำหรับบัดกรีวงจรเพิ่มเติมขนาด 8cm x 4.5cm หรือใช้เป็นพื้นที่ติดตั้ง Photo Board รุ่น AD100 ขนาด 360 จุด สำหรับต่อทดลองวงจรต่างๆ

3.1.3 คุณสมบัติของ Peripheral I/O

- ขาสัญญาณ I/O สามารถจ่ายกระแส(Source) และ รับกระแส (Sink) ได้มากถึง 25mA
- มี Timer ขนาด 16 บิต จำนวน 3 ชุด และสามารถโปรแกรมใช้งานเป็น Timer แบบ 32 บิต
- ได้โดยใช้ Timer 16 บิต 2 ช่องรวมกัน
- มี Input Capture ขนาด 16 บิต จำนวน 4 ช่อง
- มี Output Compare/PWM ขนาด 16 บิต จำนวน 2 ช่อง
- มีวงจรสื่อสารอนุกรมแบบ SPI จำนวน 1 ช่อง
- มีวงจรสื่อสารอนุกรมแบบ I2C จำนวน 1 ช่อง
- มีวงจรสื่อสารอนุกรมแบบ UART จำนวน 1 ช่อง
- มีวงจร DCPWM สำหรับใช้ควบคุมมอเตอร์ 3 ช่อง
- มีวงจรถอดรหัสแบบ QEIM ขนาด 16 บิต จำนวน 1 ช่อง
- มีวงจร A/D ขนาด 10บิต จำนวน 6 ช่อง

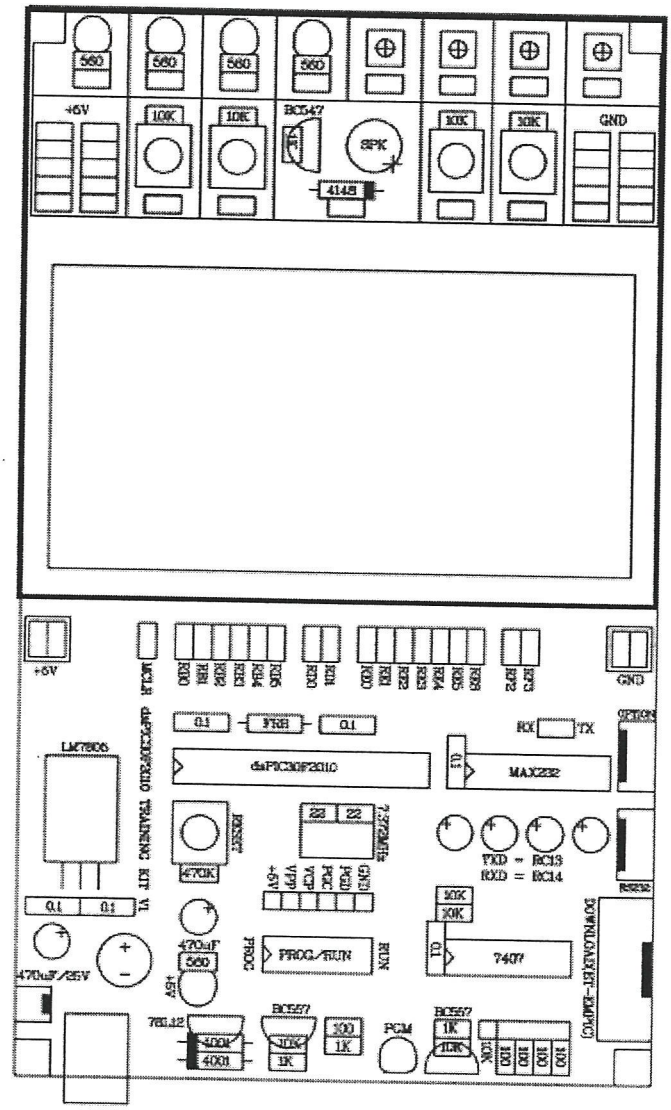
Device	Pins	Program Mem. Bytes/ Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Motor Control PWM	A/D 10-bit 500 Ksps	Quad Enc	UART	SPI™	I²C™	CAN
dsPIC30F2010	28	12K/4K	512	1024	3	4	2	6 ch	6 ch	Yes	1	1	1	–
dsPIC30F3010	28	24K/8K	1024	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	–
dsPIC30F4012	28	48K/16K	2048	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	1
dsPIC30F3011	40/44	24K/8K	1024	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	–
dsPIC30F4011	40/44	48K/16K	2048	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	1
dsPIC30F5015	64	66K/22K	2048	1024	5	4	4	8 ch	16 ch	Yes	1	2	1	1
dsPIC30F6010	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2

ตารางที่ 3.1 แสดงความแตกต่างของ dsPIC เบอร์ต่างๆ



รูปที่ 3.1แสดงการจัดเรียงขาสัญญาณของ dsPIC30F2010

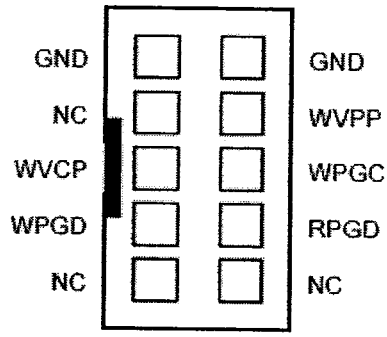
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 3.2 โครงสร้างบอร์ด ET-dsPIC30F2010 TRAININGT KIT V1.0 / EXP

104026

3.1.4 ขั้วต่อ DOWNLOAD (ET-EMPIC)



รูปที่ 3.3 โครงสร้าง ET-EMPIC

ขั้วต่อ **"DOWNLOAD"** นี้ จะใช้สำหรับเชื่อมต่อกับชุด **"ET-CAP10PIN V2.0"** ผ่านทางขั้วต่อของชุดที่ชื่อว่า **"ET-EMPIC"** โดยใช้สายแพร์ขนาด 10 Pin เป็นสายเชื่อมต่อสัญญาณระหว่างบอร์ดทั้งสองเพื่อใช้สำหรับสั่ง Download HEX File ให้กับหน่วยความจำโปรแกรม (Flash)ของ MCU ในบอร์ด "ET-dsPIC30F2010" ในขณะที่เลือกสวิตช์ **"PROG/RUN"** ไว้ทางด้าน**"PROG"** โดยสามารถใช้โปรแกรม **"WinPic800"** หรือ โปรแกรม **"Melabs Programmer"** เป็นโปรแกรมสำหรับสั่งงานก็ได้ตามต้องการ

3.2 คุณสมบัติเด่นโดยรวมของ dsPIC

dsPIC30F2010 เป็น MCU ซึ่งใช้การประมวลผลข้อมูลแบบ 16 บิต จากค่าย Microchips ซึ่งมีจุดเด่นในด้านของความสามารถในการประมวลผลข้อมูลสัญญาณแบบดิจิตอล สำหรับนำไปประยุกต์ใช้ในงานควบคุมต่างๆ โดยโครงสร้างภายในจะเป็นการผสมผสานระหว่างไมโครคอนโทรลเลอร์ (MCU) และวงจร DSP (Digital Signal Processing) รวมเข้าไว้ด้วยกัน หรืออาจเรียก MCU ตระกูล dsPIC ว่าเป็น DSC หรือ Digital Signal Controller ก็ได้

ซึ่งในปัจจุบัน MCU ในตระกูล dsPIC ของ Microchips นั้นจะมีการผลิตออกมาจำหน่ายให้ผู้ใช้งานได้เลือกใช้งานกันอยู่มากมายหลายเบอร์ตามความเหมาะสมของงาน โดยปัจจุบัน (ตุลาคม 2548) dsPIC จะแบ่งออกเป็น 5 กลุ่ม ใหญ่ๆด้วยกัน ได้แก่ dsPIC30F20xx ,dsPIC30F30xx

,dsPIC30F40xx,dsPIC30F50xx และ dsPIC30F60xx ซึ่งทุกเบอร์จะใช้โครงสร้างและสถาปัตยกรรมการประมวลผลแบบเดียวกันทั้งหมด แต่จะมีความแตกต่างกันในเรื่องของทรัพยากรภายใน เช่น ขนาดของหน่วยความจำใช้งาน จำนวนของ Peripheral I/O แบบต่างๆ ซึ่งอาจมีการบรรจุไว้ในแต่ละเบอร์ด้วยจำนวนที่ไม่เท่ากัน ซึ่งในที่นี้จะขอกล่าวถึงเพียงเฉพาะข้อมูลในส่วนที่เป็นของ dsPIC30F2010 เท่านั้น

3.2.1 คุณสมบัติเด่นของซีพียู

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูแบบ RISC
- ความเร็วในการทำงานสูงถึง 30 ล้านคำสั่งต่อวินาที
- มี 84 คำสั่งภาษาแอมเซมบีมาตรฐาน รองรับรูปแบบการอ้างแอดเดรสได้อย่างอิสระ
- ชุดคำสั่ง 24 บิต สามารถประมวลผลข้อมูลได้ 16 บิต
- มีหน่อยความจำโปรแกรมเป็นแบบแฟลช สามารถลบและเขียนใหม่ได้ไม่น้อยกว่า 100,000 ครั้ง
  สามารถป้องกันการอ่านได้ และสมามรถโปรแกรมตัวเอง โดยใช้กระบวนการทางซอฟต์แวร์
- มีหน่วยความจำข้อมูลอีอีพรอมที่สามารถลบและเขียนใหม่ได้ไม่น้อยกว่า 1,000,000 ครั้ง
- มีอินเตอร์รัปต์เวกเตอร์จำนวนมาก จึงรองรับการตอบสนองสัญญาณอินเตอร์รัปต์ได้ดี
- มีวงจรตรวจจับแรงดันไฟเลี้ยงต่ำกว่ากำหนดแบบโปรแกรมได้
- มีเพาเวอร์-ออนรีเซต, เพาเวอร์-อัปไทเมอร์ และออสซิลเลเตอร์สตาร์ต-อัปไทเมอร์
- มีวอตช์ด็อกไทเมอร์แบบโปรแกรมได้
- มีวงจรตรวจสอบการทำงานของวงจรกำเนิดสัญญาณนาฬิกา
- รองรับการโปรแกรมในวงจรแบบอนุกรม (ICSP : In-Circuit Serial Programming)
- สามารถเลือกโหมดการใช้พลังงานได้

3.2.2 คุณสมบัติด้านการประมวลสัญญาณดิจิตอล

- มีแอกคิวมูเลเตอร์ขนาด 40 บิต 2 ตัว รองรับการประมวลผลทางคณิตศาสตร์ได้เป็นอย่างดี
- มีหน่อยประมวลผลด้านการคูณและหารเลข 17 บิต ในรูปฮาร์ดแวร์ จึงทำให้สามารถคูณและหารเลขได้อย่างรวดเร็ว
- ทำการคูณเลข 16 บิต ได้ภายในสัญญาณนาฬิกาเพียง 1 ไซเกิล

- มีตัวเลื่อนข้อมูลบาร์เรล 40 สเตจ ช่วยให้การประมวลผลข้อมูลที่มีจำนวนบิตมากๆ สามารถทำได้อย่างรวดเร็ว
- มีวงจรเฟตช์ข้อมูลคู่ จึงทำให้สามารถประมวลผลข้อมูลได้อย่างรวดเร็ว

3.2.3 คุณสมบัติของโมดูลฟังก์ชั่นพิเศษ

- สามารถจ่ายกระแสออกทางขาพอร์ตได้ 25 mA ทั้งแบบกระแสซิงก์และซอร์ส
- ไทเมอร์/เคาน์เตอร์มีขนาด 16 บิต ไม่น้อยกว่า 3 ตัว ต่อใช้งานร่วมกันเป็นไทเมอร์ 32 บิต ได้
- มีโมดูลตรวจจับและเปรียบเทียบสัญญาณดิจิตอล
- มีส่วนเชื่อมต่ออุปกรณ์อนุกรมทั้งแบบ SPI และผ่านระบบบัส 12C
- มีโมดูลสื่อสารข้อมูลอนุกรม UART พร้อมบัฟเฟอร์แบบ FIFO
- มีวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล ความละเอียด 10 หรือ 12 บิต
- มีโมดูลสร้าง PWM สำหรับควบคุมมอเตอร์
- มีโมดูลเชื่อมต่อตัวเข้ารหัสแบบควอดราเจอร์

3.3 การพัฒนาโปรแกรมของบอร์ด

สำหรับการพัฒนาโปรแกรมของบอร์ด "ET-dsPIC30F2010 TRAINING KIT V1.0 / EXP" นั้น จะแบ่งออกเป็น 2 ส่วนด้วยกัน คือ ส่วนของการพัฒนาโปรแกรม และการ Download โปรแกรม โดยในส่วนของการพัฒนาโปรแกรมของบอร์ดนั้น จะเป็นการเขียนโปรแกรมเพื่อควบคุมและสั่งงานให้ MCU สามารถทำงานตามจุดประสงค์ที่ได้ออกแบบไว้ ซึ่งในส่วนนี้เป็นหน้าที่ของผู้พัฒนาโปรแกรม ว่าจะเลือกใช้ภาษาใดในการพัฒนาโปรแกรม รวมไปถึงการเลือกใช้โปรแกรมสำหรับทำหน้าที่แปลคำสั่งของภาษานั้นๆ ให้เป็นรหัสคำสั่งในรูปแบบของ HEX File สำหรับใช้ Download ให้กับหน่วยความจำโปรแกรม (Flash Memory) ของ MCU เพื่อสั่งงานให้ MCU ปฏิบัติตามคำสั่งในโปรแกรมที่ผู้พัฒนาโปรแกรมเขียนขึ้นมา

สำหรับส่วนของการ Download โปรแกรม หรือการ Download HEX File ที่ได้จากการพัฒนาโปรแกรมให้กับหน่วยความจำโปรแกรม (Flash Memory) ของ MCU นั้น จะต้องอาศัยเครื่องมือทางฮาร์ดแวร์เป็นส่วนประกอบในการพัฒนาโปรแกรมด้วย ซึ่งในกรณีของบอร์ด "ET-dsPIC30F2010" นั้น สามารถเลือกได้หลายแนวทาง ไม่ว่าจะเป็นการใช้เครื่องมือสำหรับโปรแกรมข้อมูลให้กับ MCU โดยใช้

เครื่อง Programmer ที่สนับสนุนการโปรแกรม MCU เบอร์ dsPIC30F2010 หรือใช้ชุดพัฒนาโปรแกรมจำพวก In-Circuit Debugger ที่สนับสนุนการใช้งานร่วมกับ MCU เบอร์ dsPIC30F2010 ซึ่งทั้ง 2 แนวทางข้างต้นที่กล่าวมาแล้วนั้น ทางผู้ใช้จะต้องเสียค่าใช้จ่ายในการจัดหาเครื่องมือมาใช้งานเอง แต่อย่างไรก็ตาม ทางทีมงาน อีทีที ได้ออกแบบวงจรของบอร์ด "ET-dsPIC30F2010" ให้สามารถทำการ Download Hex File จากคอมพิวเตอร์ PC โดยใช้สัญญาณควบคุมจากพอร์ตขนาน หรือ พอร์ต Printer (LPT) โดยใช้ชุดสาย ต่อสัญญาณรุ่น "ET-CAB10PIN V2.0" เป็นตัวกลางในการเชื่อมต่อได้ทันที โดยที่ผู้ใช้ไม่จำเป็นต้องจัดหาเครื่องมืออื่นใดมาใช้ให้สิ้นเปลืองอีก แต่อย่างไรก็ตามการพัฒนาโปรแกรมของบอร์ดด้วยวิธีการนี้ จะมีข้อจำกัดอยู่ 2 ประการ เมื่อเทียบกับการใช้ชุด "In-Circuit Debugger" ในการพัฒนาโปรแกรม ก็คือ

- ในปัจจุบัน โปรแกรม MPLAB ยังไม่ยอมให้มีการเชื่อมต่อเครื่องมือโปรแกรมอื่นๆ ที่ไม่ใช่เครื่องมือของ Microchips ผ่านทางเมนูคำสั่งของ MPLAB เอง ดังนั้นผู้ใช้จะต้องแยกการทำงานระหว่างการพัฒนาโปรแกรม (เขียน Source โปรแกรม และ แปลเป็น Hex File) และการ Download โปรแกรม (โปรแกรม Hex File ให้หน่วยความจำของ MCU)
- ไม่สามารถตรวจสอบข้อผิดพลาดของโปรแกรมในระหว่างการสั่งให้ MCU ทำงานอยู่ได้ แต่จะทำได้เพียงการสั่งโปรแกรม (Download Hex File ให้ MCU) และ ดูผลการทำงานของ MCU ในขณะที่ MCU ทำงานจริงๆ ซึ่งถ้าโปรแกรมที่เขียนขึ้นมาไม่ทำงานหรือทำงาน แล้วเกิดความผิดพลาดขึ้น ผู้ใช้จะต้องใช้ความสามารถ ในการ คาดเดา และตรวจสอบหาข้อผิดพลาดเหล่านั้นจาก Source Code ของโปรแกรมเอง ว่าน่าจะมีข้อผิดพลาดจากส่วนใดของโปรแกรมบ้าง แล้วจึงสั่งแก้ไขข้อผิดพลาดนั้นๆ พร้อมกับสั่งแปลโปรแกรมซ้ำใหม่ แล้วจึงสั่ง Download Hex File ให้ MCU ใหม่เพื่อดูผลการทำงาน

3.3.1 การเขียนโปรแกรมใช้งานกับบอร์ดโดยใช้ MPLAB C30

MPLAB C30 หรือ C30 Tools เป็นโปรแกรมภาษาซี สำหรับใช้แปลคำสั่งของ MCU ตระกูล dsPIC ซึ่งได้รับการพัฒนาขึ้นโดย Microchips เอง โดยข้อกำหนดและรายละเอียดของการเขียนโปรแกรมภาษาซีนั้น จะไม่กล่าวถึงในที่นี้ด้วย โดยถ้าผู้ใช้ต้องการพัฒนาโปรแกรมให้กับ dsPIC ด้วยภาษาซี แต่ยังไม่มีความรู้เรื่องการเขียนโปรแกรมภาษาซีเลยนั้นขอแนะนำให้ หาหนังสือที่อธิบายเกี่ยวกับการเขียนโปรแกรมภาษาซีในส่วนที่เป็นมาตรฐานตามข้อกำหนดของ "ANSI C" มาศึกษาให้เข้าใจเสียก่อน และสำหรับส่วนของข้อกำหนดปลีกย่อยอื่นๆที่เป็นของ MPLAB C30 เองก็สามารถอ่าน

เพิ่มเติมได้จากเอกสารและคู่มือการใช้งานของ MPLAB C30 ที่ทาง Microchips จัดทำไว้ได้ โดยสามารถ Download จาก Website ของ Microchips หรือจาก Folder ของ "..\pic30_tools\docs" ที่ทำการติดตั้งโปรแกรม MPLAB C30 ไว้ก็ได้โดยในที่นี้ จะขอกล่าวแนะนำถึงเฉพาะส่วนของการกำหนดค่าตัวเลือกในโปรแกรมเพื่อใช้งานร่วมกับบอร์ด "ET-dsPIC30F2010" เท่านั้น โดยในการที่จะใช้งานโปรแกรม MPLAB C30 ในการเขียนโปรแกรมนั้น ผู้ใช้จำเป็นต้องทำการติดตั้งโปรแกรมของ Microchips จำนวน 2 โปรแกรมดังนี้คือ

- MPLAB IDE ซึ่งเป็นโปรแกรม Text Editor ของ Microchips ซึ่งในปัจจุบัน (ตุลาคม 2548) จะเป็นรุ่น 7.21 แล้วสามารถ Download มาใช้งานได้ฟรีจาก Web ของ Microchips
- MPLAB C30 ซึ่งเป็นตัวแปลภาษาซี (C Complier) ให้เป็นรหัสคำสั่งของ dsPIC โดยในปัจจุบัน(ตุลาคม 2548) จะเป็น Version 1.33 ซึ่งตามปรกติแล้วโปรแกรมชุดนี้จะต้องซื้อมาใช้งานเอง แต่อย่างไรก็ตามทาง Microchips เองมีรุ่นทดลองใช้งานให้ผู้ใช้สามารถ Download มาใช้งานได้เช่นเดียวกันกับ MPLAB IDE

โดยโปรแกรมทั้ง 2 ชุดนี้ ทางอีทีที ได้ทำการ Download มาจัดเตรียมไว้ให้ในแผ่น CD-ROM ที่แถมไปกับบอร์ดของ "ET-dsPIC30F2010" ด้วยอยู่แล้ว โดยในการติดตั้งโปรแกรมนั้นขอแนะนำให้ผู้ใช้ทำการติดตั้งโปรแกรมในชุดของ MPLAB IDE ก่อนเป็นอันดับแรก โดยขอแนะนำให้ติดตั้งโปรแกรมของ MPLAB IDE ไว้ตามค่า Default ของโปรแกรมติดตั้งเลย คือ "C:\Program Files\Microchip\MPLAB IDE\" จะสะดวกต่อการใช้งานมากกว่า ซึ่งหลังจากทำการติดตั้งโปรแกรม MPLAB IDE เสร็จเรียบร้อยแล้วในครั้งแรก ก่อนการใช้งานนั้นต้องสั่ง Restart เครื่องคอมพิวเตอร์ก่อน หลังจากนั้นแล้ว MPLAB IDE จึงจะสามารถทำงานได้โดยไม่เกิดปัญหา จากนั้นจึงทำการติดตั้งโปรแกรม MPLAB C30 เป็นลำดับถัดไป โดยขอแนะนำให้ทำการติดตั้งโปรแกรมชุดนี้ไว้ตามค่า Default ของการติดตั้งโปรแกรม คือ "C:\PIC30_TOOSL\" จะเกิดความสะดวกต่อการใช้งานมากที่สุด โดยเฉพาะในขั้นตอนของการกำหนดการเชื่อมโยงการทำงานระหว่าง MPLAB IDE และ MPLAB C30 โดยในที่นี้จะขออธิบายโดยอ้างถึงตำแหน่งการติดตั้งโปรแกรมดังที่กล่าวไว้ แล้วข้างต้นเท่านั้น ซึ่งถ้าผู้ใช้ทำการสั่งติดตั้งโปรแกรมไว้ยังตำแหน่ง Folder ที่แตกต่างไปจากนี้แล้วข้อให้ทำความเข้าใจและดัดแปลงวิธีการกำหนดค่าเองตามที่ติดตั้งโปรแกรมไว้จริงๆด้วย

3.4 การใช้งานโมดูลเปรียบเทียบข้อมูลใน dsPIC เพื่อสร้างสัญญาณ PWM

โมดูลเปรียบเทียบข้อมูล (Output Compare: OC) เป็นโมดูลฟังก์ชั่นพิเศษที่มีบทบาทมากในการนำมาใช้ประมวลผลและสร้างสัญญาณทางดิจิตอลชิงไมโคลคอนโทรลเลอร์ dsPIC ใน dsPIC30F2010 มีโมดูล OC นี้ไว้ 2 ชุด โดยมีการใช้งานขาพอร์ตเอาต์พุตร่วมกับโมดูลตรวจจับสัญญาณอินพุตหรือ IC จึงทำให้งานจำเป็นต้องเลือกการทำงานระหว่างโมดูล IC และ OC เพราะทำงานพร้อมกันไม่ได้ แต่เนื่องจากใน dsPIC30F2010 มีให้ 2 ชุด การแยกใช้งานจึงเป็นทางออกของข้อจำกัดนี้

การทำงานหลักๆ คือเปรียบเทียบข้อมูลที่ค่าของฐานเวลาหนึ่งๆ กับข้อมูลในรีจิสเตอร์ หากเท่ากันเมื่อใดก็จะกำเนิดสัญญาณพัลส์เดี่ยวหรือขบวนสัญญาณพัลส์ออกมาทางเอาต์พุต ขึ้นอยู่กับการกำหนดโหมดการทำงาน ดังนั้นจึงมักนำโมดูลเปรียบเทียบข้อมูลมาใช้ในการสัญญาณพัลส์ โดยเฉพาะอย่างยิ่งกับการนำมาใช้ในสร้างสัญญาณ PWM (Pulse Width Modulation)

3.4.1 คุณสมบัติโดยสรุปของโมดูลเปรียบเทียบข้อมูล (Output Compare: OC)

- สามารถกำเนิดสัญญาณ PWM
- ใช้ไทเมอร์ 2 หรือ 3 ร่วมในการทำงาน
- สามารถกำหนดเงื่อนไขเมื่อถึงคาบเวลาการนับ (Match)
- อินเตอร์รัปต์เนื่องจากถึงคาบเวลาการนับ
- มีอินพุตป้องกันความผิดพลาดในการสร้างสัญญาณ PWM (Input Pin Fault Protection for PWM: ขา OCFA และ OCFB)

3.4.2 โหมดการทำงานของโมดูลเปรียบเทียบข้อมูล

มีด้วยกัน 3 โหมดหลัก ดังนี้

1. โหมดเปรียบเทียบข้อมูลตรงกันแบบเดี่ยว (Single Compare Match mode)
2. โหมดเปรียบเทียบข้อมูลตรงกันแบบคู่เพื่อสร้างสัญญาณ (Dual Compare Match mode generating) ซึ่งให้ผลลัพธ์เป็นสัญญาณพัลส์เดี่ยวหรือพัลส์ต่อเนื่อง
3. โหมดสร้างสัญญาณ PWM ซึ่งยังแบ่งได้อีก 2 ลักษณะการทำงานคือ
    3.1 แบบมีการตรวจสอบและป้องกันความผิดพลาดผ่านทางขา OCFA หรือ OCFB
    3.2 แบบไม่มีการตรวจสอบและป้องกันความผิดพลาดผ่านทางขา OCFA หรือ OCFB

3.4.3 รีจิสเตอร์หลักของโมดูลเปรียบเทียบข้อมูล

3.4.3.1 OCxR รีจิสเตอร์หลักสำหรับเก็บข้อมูลของโมดูลเปรียบเทียบข้อมูล

เป็นรีจิสเตอร์ขนาด 16 บิตที่ใช้เก็บข้อมูลที่ต้องการเปรียบเทียบสำหรับใน dsPIC30F2010 จะมี 2 ตัวคือOC1R สำหรับโมดูลเปรียบเทียบข้อมูลชุดที่ 1 (OC1) และ OC2R สำหรับโมดูลเปรียบเทียบข้อมูลชุดที่ 2 (OC2)

3.4.3.2 OCxRS รีจิสเตอร์หลักสำหรับเก็บข้อมูลของโมดูลเปรียบเทียบข้อมูลตัวที่สอง

เป็นรีจิสเตอร์ขนาด 16 บิตที่ใช้เก็บข้อมูลที่ต้องการเปรียบเทียบโดยได้รับการถ่ายทอดข้อมูลมาจากรีจิสเตอร์ OCxR สำหรับใน dsPIC30F2010 จะมี 2 ตัวคือOC1RS สำหรับโมดูลเปรียบเทียบข้อมูลชุดที่ 1 (OC1) และ OC2RSสำหรับโมดูลเปรียบเทียบข้อมูลชุดที่ 2 (OC2)

3.4.3.3 OCxCON เป็นรีจิสเตอร์ควบคุมการทำงานของโมดูลเปรียบเทียบข้อมูล

เป็นรีจิสเตอร์ 16 บิตที่ใช้กำหนดค่าเพื่อควบคุมการทำงานของโมดลเปรียบเทียบข้อมูล สำหรับในdsPIC30F2010 มี 2 ตัวเช่นกันคือ OC1CON สำหรับโมดูลเปรียบเทียบข้อมูลชุดที่ 1 (OC1) และ OC2CON สำหรับโมดูลเปรียบเทียบข้อมูลชุดที่ 2 (OC2) มีรายละเอียดโดยสรุปดังนี้

บิต 15	บิต 14	บิต 13	บิต 12	บิต 11	บิต 10	บิต 9	บิต 8
-	-	OSCIDL	-	-	-	-	-
U -0	U -0	R/W -0	U -0	U -0	U -0	U -0	U -0

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
-	-	-	OCFLT	OCTSEL	OCM2	OCM1	OCM0
U -0	U -0	U -0	R -0	R/W -0	R/W -0	R/W -0	R/W -0

บิต 15 และ 14 ไม่มีการใช้งาน กำหนดเป็น “0”

บิต 13 – OCSIDL (Step Output Compare in IDLE Mode Control bit): บิตเลือกการหยุดทำงานของโมดูลเปรียบเทียบข้อมูล เมื่อทำงานในโหมดไอเดิล

“0” = โมดูล OC ยังคงทำงานต่อไปแม้ไมโครคอนโทรลเลอร์จะเข้าสู่โหมดไอเดิลก็ตาม

"1" = โมดูล OC หยุดการทำงาน เมื่อไมโครคอนโทรลเลอร์เข้าสู่โหมดไอเดิล

บิต 12 – 5 ไม่มีการใช้งาน กำหนดเป็น "0"

บิต 4 – OCFLT (PWM Fault Condition Status bit): บิตแฟลกแสดงสถานะของการตรวจจับความผิดพลาด

"0" = ไม่มีความผิดพลาดใดๆ เกิดขึ้น

"1" = มีการแจ้งสถานะความผิดพลาด

บิตนี้สามารถอ่านได้เพียงอย่างเดียว

บิต 3 – OCTSEL (Output Compare Timer Select bit): บิตเลือกไทเมอร์ขนาด 16 บิตที่ต้องการทำงานร่วมด้วย

"0" = เลือกใช้ไทเมอร์ 2 (TMR2) เป็นแหล่งกำเนิดสัญญาณนาฬิกาสำหรับโมดูล OC

"1" = เลือกใช้ไทเมอร์ 3 (TMR3) เป็นแหล่งกำเนิดสัญญาณนาฬิกาสำหรับโมดูล OC

บิต 2 ถึง 0 – OCM2 ถึง OCM0 (Output Compare Mode Select bits): บิตเลือกโหมดการทำงานของโมดูล OC

"000" = ปิดการทำงานของโมดูล OC

"001" = กำหนดให้เป็นขา OCx เป็นลอจิกต่ำ เมื่อค่าเท่ากัน ทำให้ขานี้เป็น "1"

"010" = กำหนดให้เป็นขา OCx เป็นลอจิกสูง เมื่อค่าเท่ากัน ทำให้ขานี้เป็น "0"

"011" = กำหนดให้เอาต์พุตเกิดการกลับค่าลอจิก เมื่อค่าของการเปรียบเทียบเท่ากัน

"100" = กำหนดให้ขา OCx เป็นลอจิกต่ำ เมื่อค่าเท่ากัน จะเกิดสัญญาณพัลส์เดี่ยวออกมา

"101" = กำหนดให้ขา OCx เป็นลอจิกต่ำ เมื่อค่าเท่ากัน จะเกิดสัญญาณพัลส์ต่อเนื่องออกมา

"110" = ทำงานในโหมดสร้างสัญญาณ PWM แบบไม่ใช้ขาตรวจสอบ OCFA หรือ OCFB

"111" = ทำงานในโหมดสร้างสัญญาณ PWM แบบใช้ขาตรวจสอบ OCFA หรือ OCFB

3.4.4 การกำเนิดสัญญาณ PWM ของโมดูลเปรียบเทียบข้อมูล

จะเริ่มต้นด้วยการกำหนดบิต OCM2-OCM0 ในรีจิสเตอร์ OCxCON เป็น "110" หรือ "111" ขึ้นอยู่กับว่าต้องการใช้งานขาอินพุต OCFA หรือ OCFB เพื่อป้องกันความผิดพลาดหรือไม่ เนื่องจากใน dsPIC30F2010 มีขาพอร์ต 20 ขามีโมดูล OC 2ชุด จึงมีขาอินพุตเพื่อช่วยตรวจสอบความผิดพลาดเพียงขาเดียวคือ OCFA เนื่องจาก OCFA จะสามารถทำงานได้กับโมดูล OC1-OC4

ในโหมดกำหนดสัญญาณ PWM รีจิสเตอร์ OCxR จะถูกใช้ในการเก็บค่าดิวตี้ไซเกิล จึงสามารถอ่านได้เพียงอย่างเดียว ดังนั้นในการเขียนข้อมูลเพื่อนกำหนดค่าของดิวตี้ไซเกิลต้องกระทำผ่านรีจิสเตอร์

OCxRS แทน ทุกครั้งที่ค่าของรีจิสเตอร์คาบเวลา (period register) ซึ่งอ่านได้มาจากไทเมอร์ 2 หรือ 3 มีค่าตรงกับเงื่อนไขที่กำหนด ค่าดิวตี้ไซเกิลที่เก็บไว้ในรีจิสเตอร์ OCxR จะถูกโหลดมมายัง OCxRS เพื่อนำไปใช้สร้างสัญญาณ PWM ต่อไป พร้อมกันนั้นบิตแฟลก TyIF จะเซตเพื่อแจ้งสถานะที่เกิดขึ้น

สามารถสรุปขั้นตอนในการใช้งานโมดูล OC ในการสร้างสัญญาณ PWM ได้ดังนี้

1. กำหนดค่าของคาบสัญญาณ PWM แล้วเขียนลงในรีจิสเตอร์คาบเวลา PRy
2. กำหนดค่าดิวตี้ไซเกิลของสัญญาณ PWM แล้วเขียนลงในรีจิสเตอร์ OCxRS
3. จะเกิดกระบวนการเขียนข้อมูลลงในรีจิสเตอร์ OCxR ซึ่งเป็นกระบวนการภายใน จึงต้องรอชั่วขณะ
4. เอ็นเอเบิลอินเตอร์รัปต์ถ้าต้องการ แต่ถ้าหากเลือกใช้อินเตอร์รัปต์ จะต้องกำหนดให้ทำงานในโหมดตรวจสอบความผิดพลาดด้วยขา OCFA ด้วย
5. กำหนดโหมดการสร้างสัญญาณ PWM โดยเขียนข้อมูลไปยังบิต OCM2-OCM0 ซึ่งกำหนดได้ 2 โหมด แต่ถ้าเลือกใช้อินเตอร์รัปต์จะต้องกำหนดให้มีค่า "111" เท่านั้น
6. ตั้งค่าปรีสเกลเลอร์ในไทเมอร์ที่นำมาใช้งานร่วมด้วย แล้วเอ็นเอเบิลโดยการเซตบิต TON ซึ่งเป็นบิต 15 ในรีจิสเตอร์ TxCON ให้เป็น "1"

3.4.5 การคำนวณเกี่ยวกับคาบเวลาของสัญญาณ PWM (Tpwm)

สามารถคำนวณได้จากสมการดังต่อไปนี้

$$Tpwm = [(PRx)+1]*4*Tosc*(TMR*prescale\ value)$$

โดยที่ x = 2 หรือ 3 ขึ้นอยู่กับการเลือกไทเมอร์มาใช้งาน

PRx คือ รีจิสเตอร์เก็บค่าการนับของไทเมอร์ที่ถูกเลือก

Tosc คือ คาบเวลาของระบบ (1/Fosc)

TMRx prescale value คือ ค่าปรีสเกลเลอร์ของไทเมอร์ที่ถูกเลือก

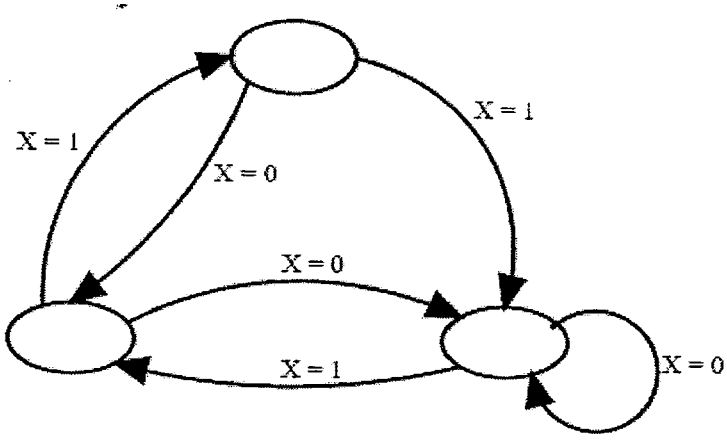
3.5 ทฤษฎี State Transition Machine

หุ่นยนต์แมลง 6 ขา ที่ควบคุมด้วยระบบโครงข่ายประสาทเทียมนี้แบ่งการทำงานออกเป็น 1 เซลล์ ซึ่งจะทำงานควบคุมขาแต่ละขาของหุ่นและทำงานเป็นอิสระต่อกัน โดยที่เซลล์ ได้รับอินพุทค่าเดียวกัน จากนั้นเซลล์จะนำค่าอินพุทและค่าเอาท์พุทเซลล์ของเข้า มาแล้วทำการเปรียบเทียบกับค่าในตารางซึ่งเป็นค่าที่เซลล์ได้เรียนรู้แล้ว (learned)

ถ้าเท่ากันก็จะส่งค่าเอาท์พุทออกมา ถ้าค่าที่ได้รับมาไม่เท่ากันก็จะส่งค่าที่ใกล้เคียงที่สุดออกมา ใน กรณีนี้หุ่นจะเดินไม่ถูกสเตท แต่หลังจากที่แต่ละเซลล์ให้เอาท์พุทออกมาแล้วก็จะทำให้ได้ค่าใหม่เป็นอินพุทซึ่งตอนนี้นี่เองที่ทำให้เมื่อนำค่าที่ได้มาเทียบตารางก็จะเป็นค่าที่ตรงกัน

การทำงานของหุ่นจะเป็นแบบ state transition machine คือทำงานทีละ state เป็นจังหวะที่ต่อเนื่องกันไปและใช้ state ที่แล้วมาเป็นตัวอ้างอิง กล่าวคือค่าเอาท์พุทที่ออกมาจะได้มาจากค่า เอาท์พุทของ state ขณะนั้นโดยพิจารณาว่าขณะนี้ตำแหน่งของขาอยู่ที่ไหน(ลักษณะ state เป็น อย่างไร) จากนั้นรับค่าอินพุทเข้ามาแล้วพิจารณาว่าตำแหน่งของขาเป็นอย่างนี้ รับอินพุทมาอย่างนี้ จะให้เอาท์พุทอย่างไร ซึ่งในโครงงานนี้จะใช้ การเขียน State Diagram แบบ Moore Machines ซึ่งจะประกอบไปด้วย รูปทรงกลมแทนสเตทต่างๆและเส้นเชื่อมต่อระหว่างสเตทที่มีลูกศรชี้ที่สเตท ถัดไปบนเส้นจะมีอินพุทกำกับไว้

ตัวอย่าง Moore Machines

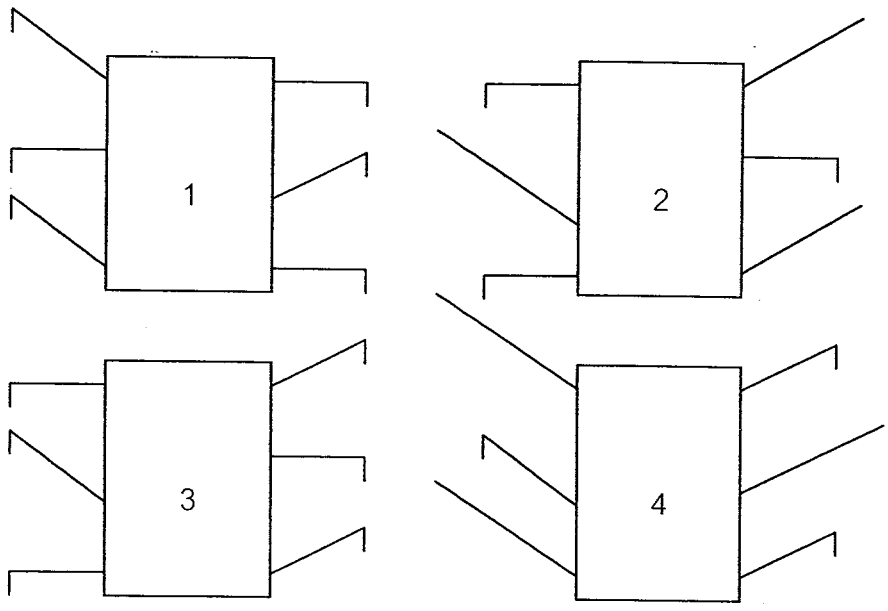


รูปที่ 3.4 Moor Machines

Moore Machines จะมีการเพิ่มเอาท์พุทเข้าไปในแต่ละสเตท หรือก็คือ เมื่อสเตทปัจจุบัน (PS) ได้รับอินพุทใดๆเข้ามาจะทำให้เกิดเอาท์พุทค่าหนึ่งๆ แล้วจึงเปลี่ยนสเตทไปที่สเตทถัดไป (NS) ตัวเลขที่อยู่ในวงกลมคือ สเตท/เอาท์พุท ตัวเลขที่อยู่บนเส้นคือ อินพุท จากหลักการนี้เริ่มต้นต้องมีการคิดรูปแบบการเดินของหุ่นออกมาเป็นสเตทการเดินแบบ ต่างๆ เช่น การเดินหน้า เลี้ยวซ้าย เลี้ยวขวา เป็นต้นแล้วเราก็จะพบว่าในการเดินแบบต่างๆของหุ่น มี ขาใดที่ทำงานบ้างแล้วนำมาเขียนแสดงเป็นตารางการทำงาน

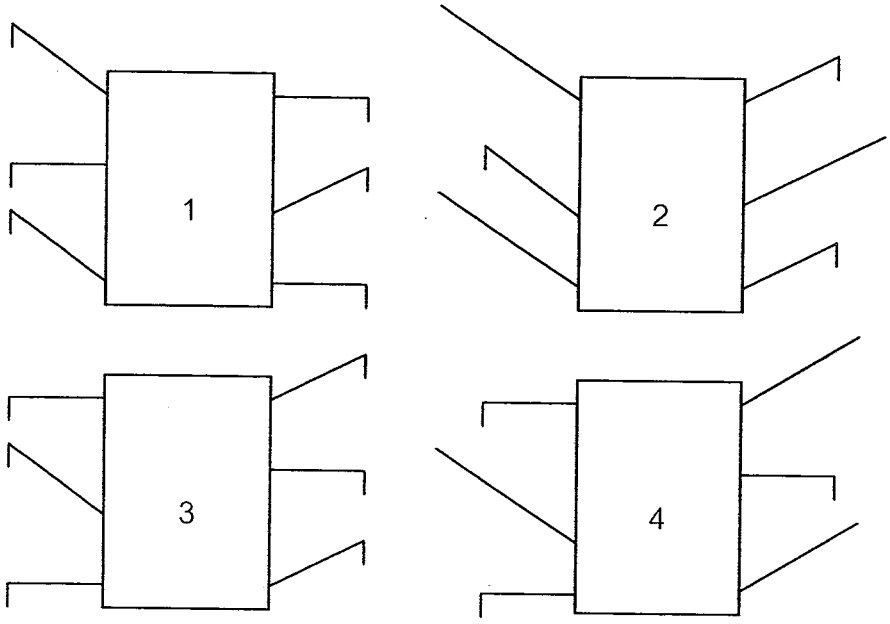
การทำงานของหุ่นยนต์เริ่มจากหุ่นยนต์เดินไปข้างหน้าจนกระทั่ง เมื่อตรวจจับอินพุทที่เป็นเซนเซอร์ชนิดต่างๆเช่น เซนเซอร์อินฟราเรดหรือเซนเซอร์ไพโรอิเล็กทริก ได้แล้วก็จะนำอินพุทนั้นเหล่านั้นมาเช็คว่าตรงกับเงื่อนไขการเดินแบบใด แล้วจึงทำการเปลี่ยนสเตท ยกตัวอย่างเช่น ตอนแรกหุ่นยนต์เดินไปข้างหน้าเรื่อยๆต่อมาสามารถตรวจจับได้ว่าพบสิ่งกีดขวางอยู่ทางซ้ายและพบคนอยู่ทางขวาเงื่อนไขของการเดินคือต้องหลบสิ่งกีดขวางแล้วเดินไปหาคน ดังนั้นหุ่นยนต์จึงเปลี่ยนจากสเตทเดินหน้าเป็นสเตทเลี้ยวขวา จนกระทั่งหลบสิ่งกีดขวางพ้น ถ้าไม่สามารถตรวจพบคนหรือสิ่งกีดขวางได้แล้ว หุ่นยนต์ก็จะกลับเข้าสู่สเตทเดินหน้าดังเดิม เป็นเช่นนี้ต่อไปเรื่อยๆ ทำให้หุ่นยนต์เคลื่อนที่แบบออโตเมติก (Automatic)

การเดินไปข้างหน้า



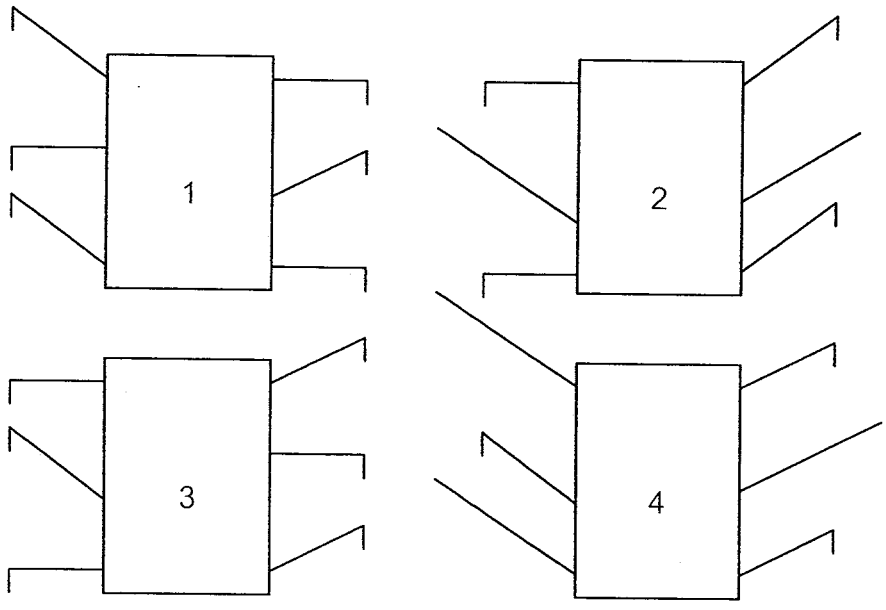
รูปที่ 3.5 รูปแบบการเดินไปข้างหน้า

การเดินไปถอยหลัง



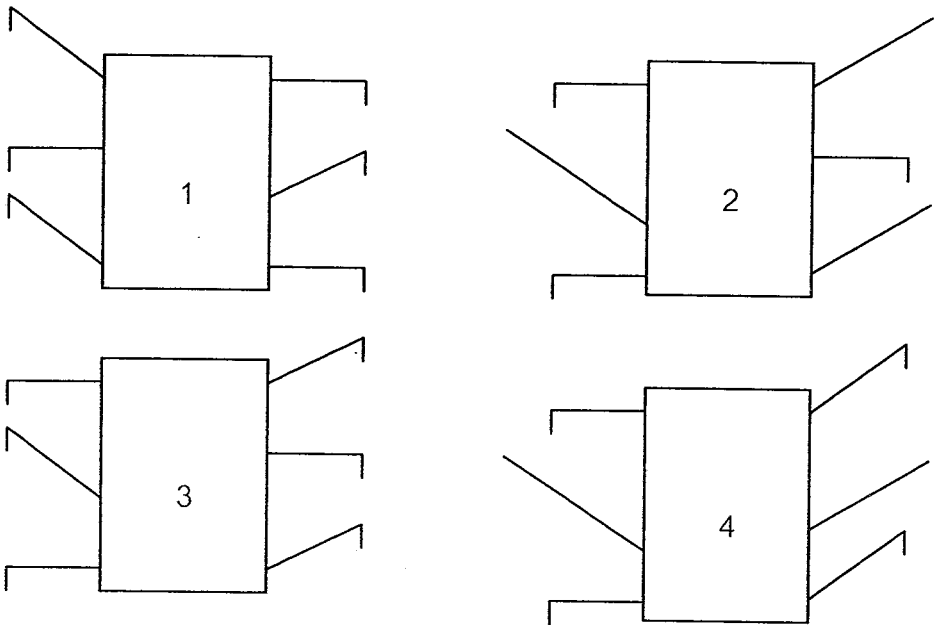
รูปที่ 3.6 รูปแบบการถอยหลัง

การเดินไปข้างขวา



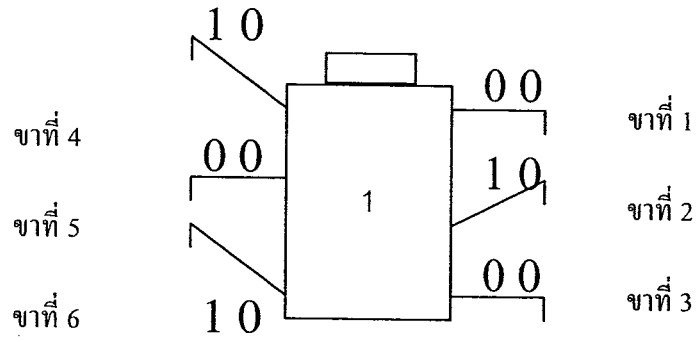
รูปที่ 3.7 รูปแบบการเดินไปข้างขวา

การเดินไปทางซ้าย



รูปที่ 3.8 รูปแบบการเดินไปข้างซ้าย

และเมื่อเราเขียนสเตทการเดินทุกแบบแล้วจะได้ลักษณะการก้าวขาและการยกขาของขาแต่ ละขาออกมา ดังนี้



รูปที่ 3.9 แสดงสถานะขาหุ่นในสภาวะต่างๆ

ตารางที่ 3.2 ตารางแสดงค่าการเดินหน้าของหุ่นยนต์

leg	1		2		3		4		5		6		รวม
Servo	1	2	3	4	5	6	7	8	9	10	11	12	
Step 1	0	0	1	0	0	0	1	0	0	0	1	0	001000100010
Step 2	1	1	0	0	1	1	0	0	1	1	0	0	110011001100
Step 3	1	0	0	0	1	0	0	0	1	0	0	0	100010001000
Step 4	1	0	1	1	1	0	1	1	1	0	1	1	101110111011

ตารางที่ 3.3 ตารางแสดงค่าการเดินถอยหลังของหุ่นยนต์

leg	1		2		3		4		5		6		รวม
Servo	1	2	3	4	5	6	7	8	9	10	11	12	
Step 1	0	0	1	0	0	0	1	0	0	0	1	0	001000100010
Step 2	1	0	1	1	1	0	1	1	1	0	1	1	101110111011
Step 3	1	0	0	0	1	0	0	0	1	0	0	0	100010001000
Step 4	1	1	0	0	1	1	0	0	1	1	0	0	110011001100

ตารางที่ 3.4 ตารางแสดงค่าการเดินไปทางขวาของหุ่นยนต์

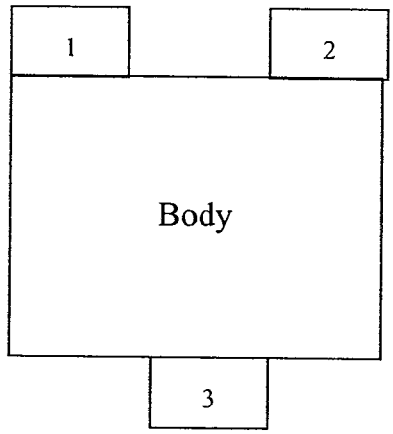
leg	1		2		3		4		5		6		รวม
Servo	1	2	3	4	5	6	7	8	9	10	11	12	
Step 1	0	0	1	0	0	0	1	0	0	0	1	0	001000100010
Step 2	1	0	1	1	1	0	0	0	1	1	0	0	101110001100
Step 3	1	0	0	0	1	0	0	0	1	0	0	0	100010001000
Step 4	1	0	1	1	1	0	1	1	1	0	1	1	101110111011

ตารางที่ 3.5 ตารางแสดงค่าการเดินไปทางซ้ายของหุ่นยนต์

leg	1		2		3		4		5		6		รวม
Servo	1	2	3	4	5	6	7	8	9	10	11	12	
Step 1	0	0	1	0	0	0	1	0	0	0	1	0	001000100010
Step 2	1	1	0	0	1	1	0	0	1	1	0	0	110011001100
Step 3	1	0	0	0	1	0	0	0	1	0	0	0	100010001000
Step 4	1	0	1	1	1	0	0	0	1	1	0	0	101110001100

3.6 ตารางอินพุท

เนื่องจากตำแหน่งของวัตถุที่เป็นไปได้มีทั้งหมด 4 แบบ จึงใช้รหัสอินพุทอย่างละ 1 บิตในการตรวจจับสิ่งกีดขวาง ซึ่งทำให้ความเป็นไปได้ของอินพุทที่เข้ามาซึ่งมีทั้งหมด 7 กรณี



รูปที่ 3.10 ตำแหน่งของ Infrared Sensor

ตารางที่ 3.6 แสดงค่าอินพุตของ Sensor

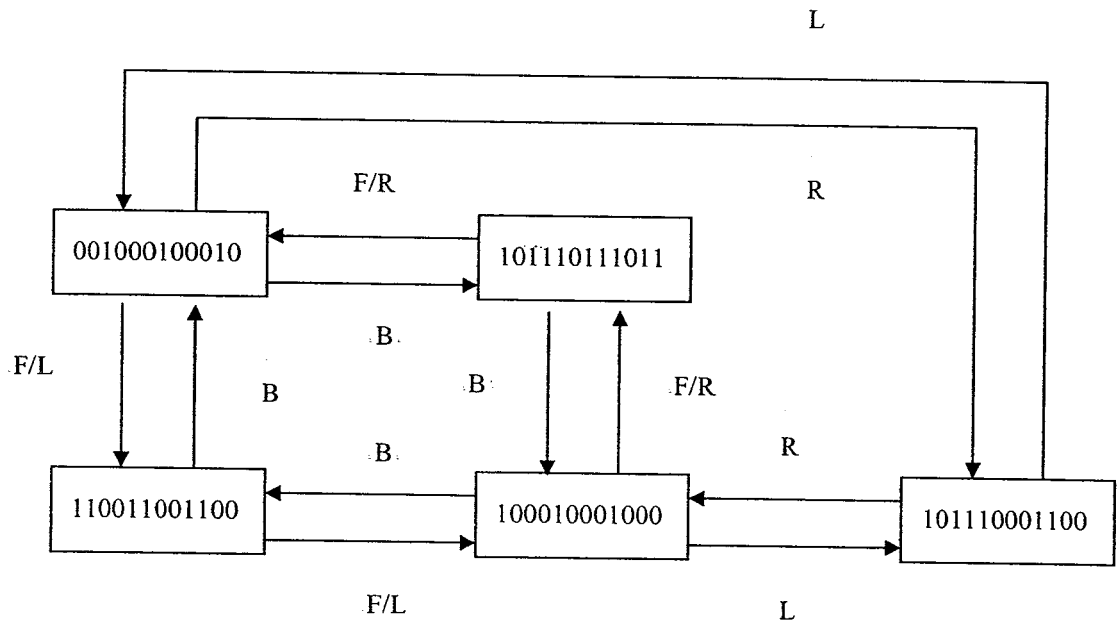
ตารางที่ 3.6 แสดงค่าอินพุตของ Sensor

ตำแหน่งของวัตถุ	Sensor ตัวที่1	Sensor ตัวที่2	Sensor ตัวที่3
ไม่พบ	1	1	1
ด้านหน้า	0	0	1
ด้านหลัง	1	1	0
ด้านซ้าย	0	1	1
ด้านขวา	1	0	1

1 แสดงสถานะไม่พบวัตถุ

0 แสดงสถานะพบวัตถุ

3.7 สเตจการเดินของหุ่นยนต์



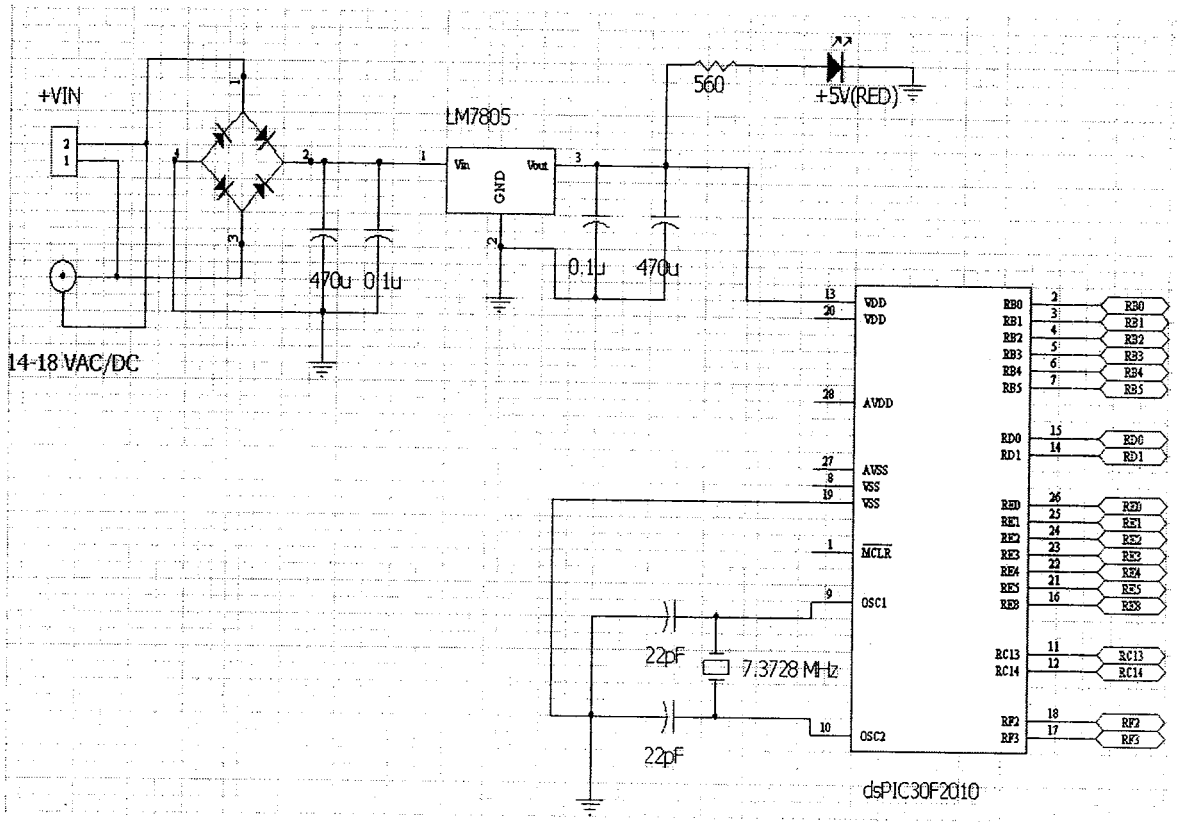
F = Forward, B = Backward, R = Right, L = Left

รูปที่ 3.11 สเตจการเดินของหุ่นยนต์

บทที่ 4

วงจรอิเล็กทรอนิกส์และการทำงาน

4.1 ส่วนของวงจรทำงานหลัก



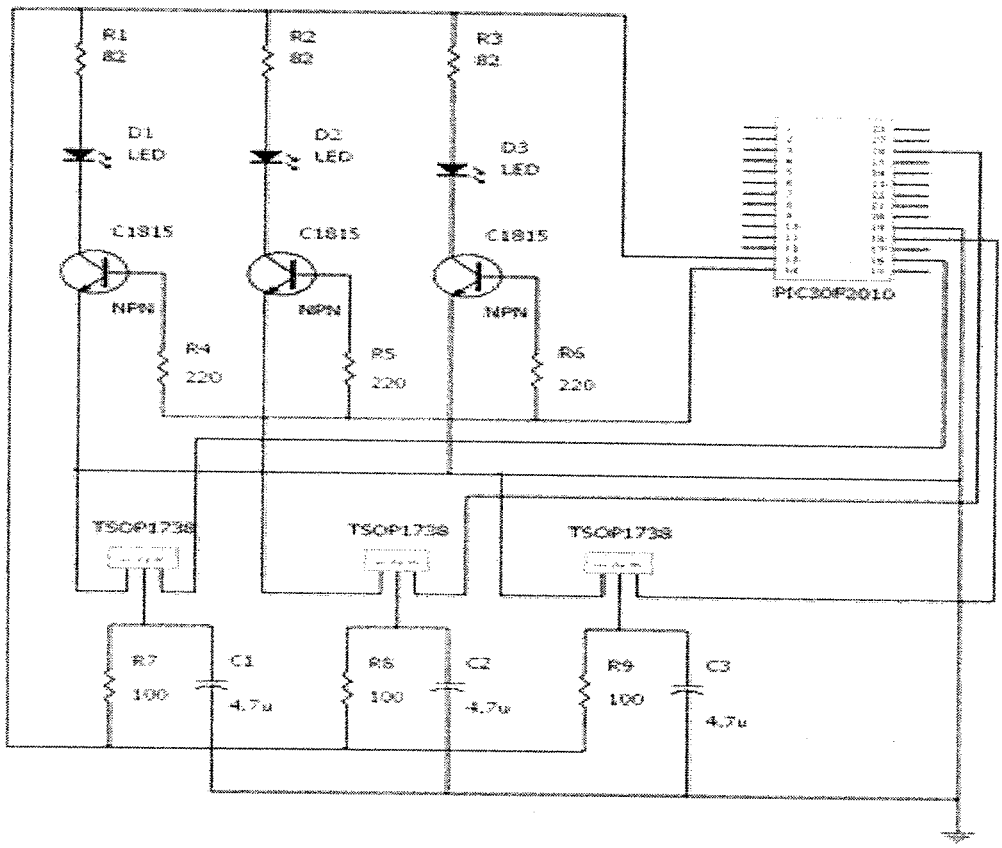
รูปที่ 4.1 วงจรควบคุมการทำงานหลัก

วงจรควบคุมประกอบด้วยไมโครคอนโทรลเลอร์ (dsPIC30F2010) ตัวรับสัญญาณจากอินฟราเรดส่วนส่งสัญญาณไปควบคุมเซอร์โวมอเตอร์และหน่วยความจำ (RAM) ขนาด 8Kเบอร์ HT6264-70-28DIP การควบคุมการทำงานทั้งหมดจะถูกสั่งการจาก ไมโครคอนโทรลเลอร์ (dsPIC30F2010) ซึ่งเป็นการประมวลผลบนชิพเพื่อเป็นการลดขนาดของอุปกรณ์ที่ใช้

ไมโครคอนโทรลเลอร์(dsPIC30F2010) ใช้แหล่งกำเนิดสัญญาณนาฬิกาขนาด 7.3728 MHz และปฏิบัติการที่แรงดันไฟฟ้ากระแสตรงขนาด 5 โวลท์ซึ่งได้จาก IC 7805 โวทเตจเรกูเลเตอร์ (Voltage regulator) โดยมีแหล่งจ่าย แบตเตอรี่ ขนาด 14 โวลท์

พอร์ต RE0 – RE5 และพอร์ต RB0 – RB5 จะทำหน้าที่เป็นเอาต์พุตสัญญาณพัลส์ขนาด 1.25, 1.5 และ1.75 มิลลิวินาที เพื่อขับมอเตอร์ให้ทำงานและพอร์ต RF2, RD0 และ RE8 จะทำหน้าที่เป็นอินพุตรับสัญญาณจากเซนเซอร์ทั้ง 3 ตัว

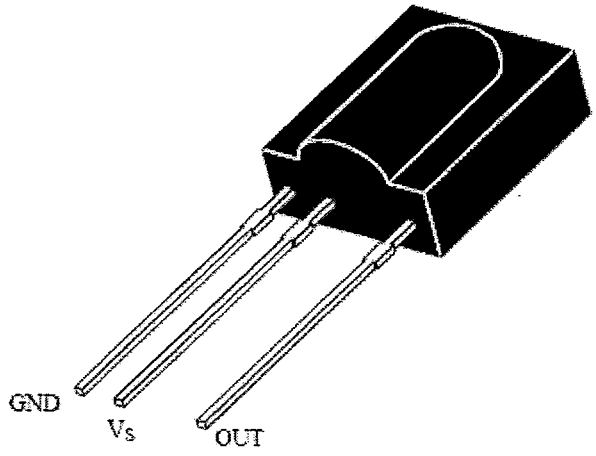
4.2 วงจรหลบหลีกสิ่งกีดขวาง



รูปที่ 4.2 วงจรตัวรับสัญญาณอินฟราเรด

วงจรตัวรับสัญญาณอินฟราเรด ประกอบด้วย LED อินฟราเรด 3 ตัว และตัวเซนเซอร์ (TSOP1738 IR Sensor Modules) ตัวรับและตัวส่งสัญญาณอินฟราเรดถูกติดตั้งบริเวณส่วนหน้าและด้านหลังของหุ่นยนต์เพื่อเพิ่มขีดความสามารถในการประมวลผลลักษณะของสภาพแวดล้อมข้างหน้าเพื่อเคลื่อนที่หลบหลีกสิ่งกีดขวางในขณะเดิน

TSOP1738 IR ถูกออกแบบมาให้รับได้เฉพาะสัญญาณอินฟราเรดที่มีความถี่ 38 KHz และจะตัดสัญญาณจากแหล่งกำเนิดแสงอื่นๆ ซึ่งจะทำให้มีความสามารถในการทำงานร่วมกับแสงในชีวิตประจำวัน ซึ่งมีความสามารถในการรับสัญญาณที่อยู่ห่างออกไปเป็นระยะทางประมาณ 10 เซนติเมตร



รูปที่ 4.3 TSOP1738 IR Sensor Modules

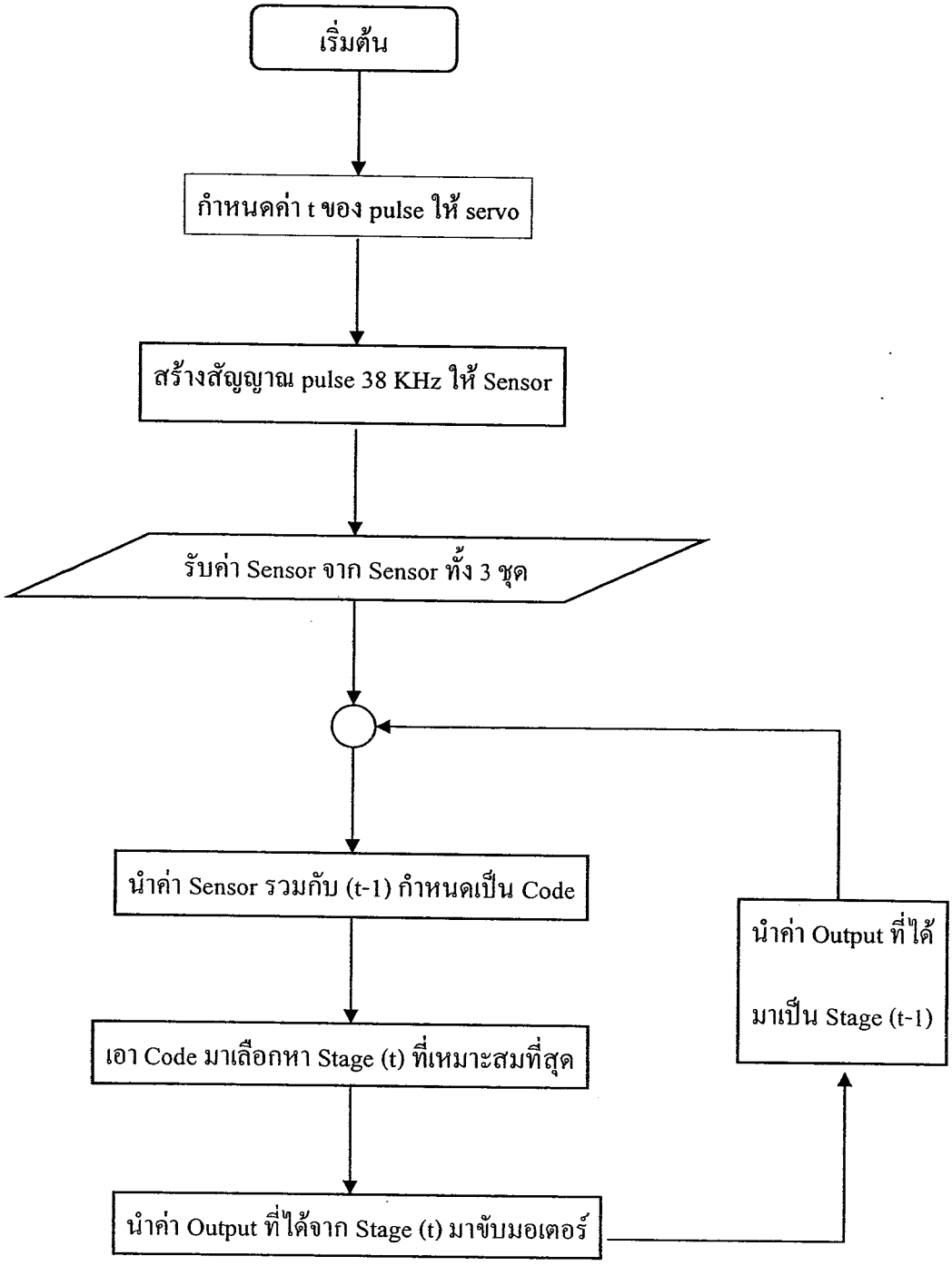
บทที่ 5
ขั้นตอนและโปรแกรมการทำงาน

5.1 Table การทำงานของหุ่นยนต์

ตารางที่ 5.1 Table การทำงานของหุ่นยนต์

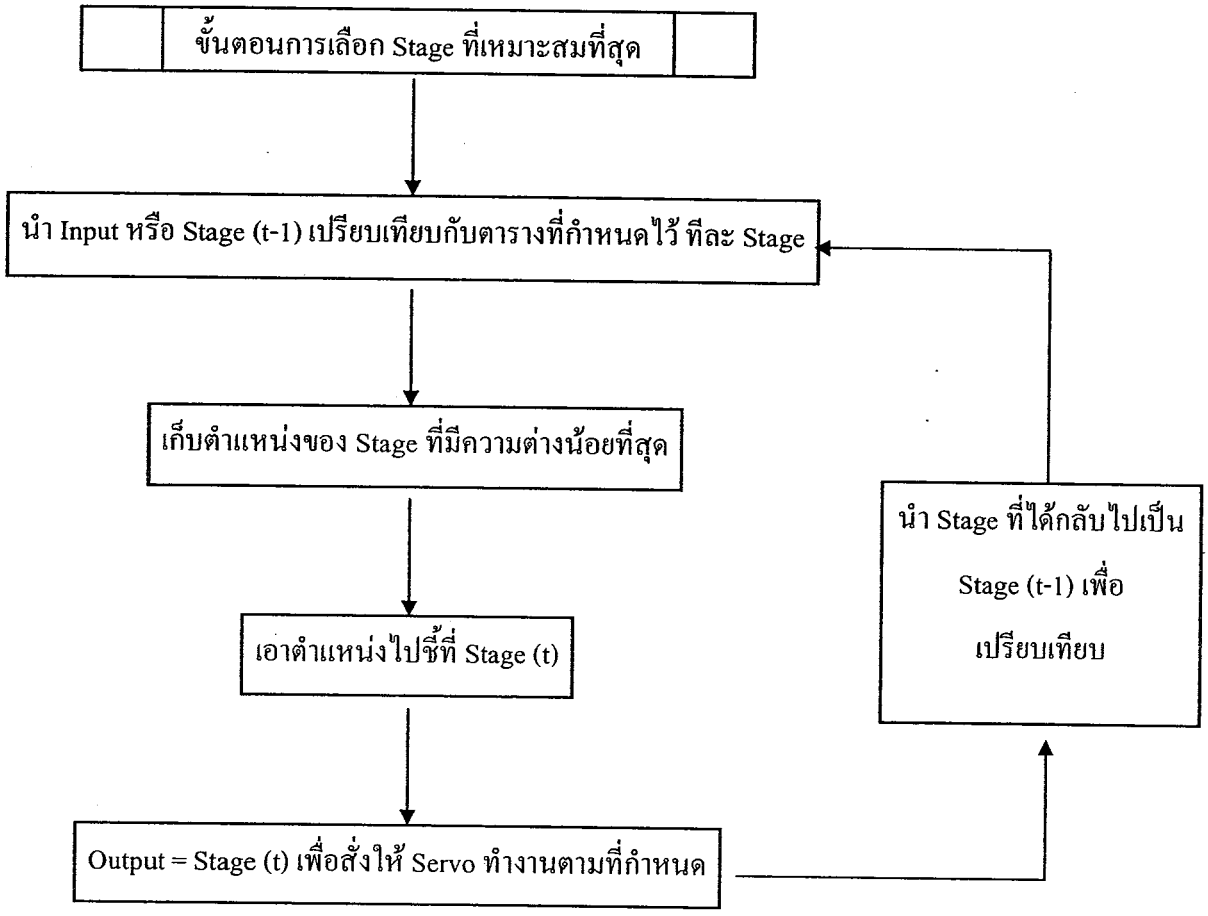
Input				Output
Sensor 1	Sensor 2	Sensor 3	t - 1	
11	11	1	101110111011	001000100010
11	11	1	001000100010	110011001100
11	11	1	110011001100	100010001000
11	11	1	100010001000	101110111011
11	11	0	101110111011	001000100010
11	11	0	001000100010	110011001100
11	11	0	110011001100	100010001000
11	11	0	100010001000	101110111011
00	00	1	110011001100	001000100010
00	00	1	001000100010	101110111011
00	00	1	101110111011	100010001000
00	00	1	100010001000	110011001100
00	11	1	101110111011	001000100010
00	11	1	001000100010	101110001100
00	11	1	101110001100	100010001000
00	11	1	100010001000	101110111011
11	00	1	101110001100	001000100010
11	00	1	001000100010	110011001100
11	00	1	110011001100	100010001000
11	00	1	100010001000	101110001100

5.2 Flow Chart การทำงานของหุ่นยนต์



รูปที่ 5.1 Flow Chart การทำงานของหุ่นยนต์

5.3 ขั้นตอนการทำงานสำหรับเลือกหา Stage (t) ที่เหมาะสมที่สุด



รูปที่ 5.2 ขั้นตอนการทำงานสำหรับเลือกหา Stage (t) ที่เหมาะสมที่สุด

5.4 ตัวอย่าง การเลือก Stage (t) ที่เหมาะสม

เมื่อ Stage (t-1) ที่รับเข้ามามีค่าเท่ากับ 110101110111011 จะได้ค่าดังตาราง

ตาราง ที่ 5.2 ตัวอย่าง การเลือก Stage (t) ที่เหมาะสม

	Input				Output	ตำแหน่ง	ความต่าง bit
	S1	S2	S3	t - 1	t		
เดินหน้า	11	11	1	101110111011	001000100010	1	1
	11	11	1	001000100010	110011001100	2	7
	11	11	1	110011001100	100010001000	3	10
	11	11	1	100010001000	101110111011	4	7
เดินหน้า	11	11	0	101110111011	001000100010	5	**0**
	11	11	0	001000100010	110011001100	6	6
	11	11	0	110011001100	100010001000	7	9
	11	11	0	100010001000	101110111011	8	6
ถอยหลัง	00	00	1	110011001100	001000100010	9	3
	00	00	1	001000100010	101110111011	10	9
	00	00	1	101110111011	100010001000	11	12
	00	00	1	100010001000	110011001100	12	9
เลี้ยวขวา	00	11	1	101110111011	001000100010	13	2
	00	11	1	001000100010	101110001100	14	8
	00	11	1	101110001100	100010001000	15	8
	00	11	1	100010001000	101110111011	16	8
เลี้ยวซ้าย	11	00	1	101110001100	001000100010	17	8
	11	00	1	001000100010	110011001100	18	8
	11	00	1	110011001100	100010001000	19	11
	11	00	1	100010001000	101110001100	20	8

S1 = Sensor 1, S2 = Sensor 2, S3 =Sensor 3

จะได้ว่าที่ตำแหน่งที่ 5 มีความต่างบิทน้อยที่สุด จึงเลือกเอา stage(t) = 001000100010 ไปสั่งให้มอเตอร์ทำงานต่อไป

บทที่ 6

การทดลองและผลการทดลอง

6.1 การทดลองจ่ายพัลส์ให้กับมอเตอร์

6.1.1 จากการทดลองจ่ายพัลส์แบบต่างๆให้กับมอเตอร์ที่ความถี่คงที่ 50-60Hz โดยใช้พัลส์วิดมอดูเลตจะได้ลักษณะการหมุนของมอเตอร์ที่แตกต่างกันดังนี้

สำหรับมอเตอร์ตัวที่ 1

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา
-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา
-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

สำหรับมอเตอร์ตัวที่ 2

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา
-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา
-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

สำหรับมอเตอร์ตัวที่ 3

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา
-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา
-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

สำหรับมอเตอร์ตัวที่ 4

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา
-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา
-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

สำหรับมอเตอร์ตัวที่ 5

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา
-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา
-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

สำหรับมอเตอร์ตัวที่ 6

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา

-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา

-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

สำหรับมอเตอร์ตัวที่ 7

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา

-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา

-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

สำหรับมอเตอร์ตัวที่ 8

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา

-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา

-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

สำหรับมอเตอร์ตัวที่ 9

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา

-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา

-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

สำหรับมอเตอร์ตัวที่ 10

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา

-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา

-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

สำหรับมอเตอร์ตัวที่ 11

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา

-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา

-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

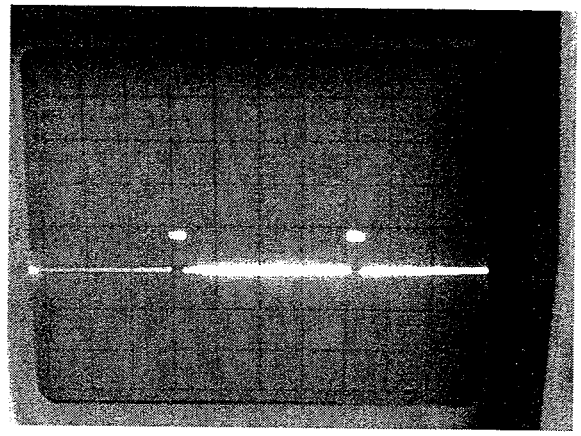
สำหรับมอเตอร์ตัวที่ 12

- เมื่อป้อนพัลส์ที่ 1.75 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ 45 องศา

-เมื่อป้อนพัลส์ที่ 1.5 ms. มอเตอร์กลางจะอยู่ในสถานะกลางที่ 0 องศา

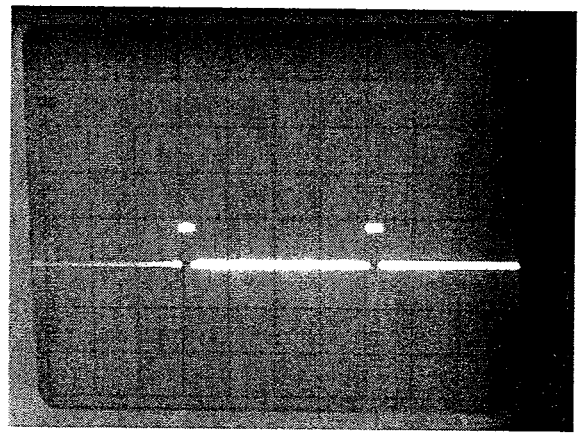
-เมื่อป้อนพัลส์ที่ 1.25 ms. มอเตอร์กลางจะหมุนตามเข็มที่ประมาณ -45 องศา

การป้อนสัญญาณพัลส์ให้เซอร์โวมอเตอร์ (5 ms / DIV)



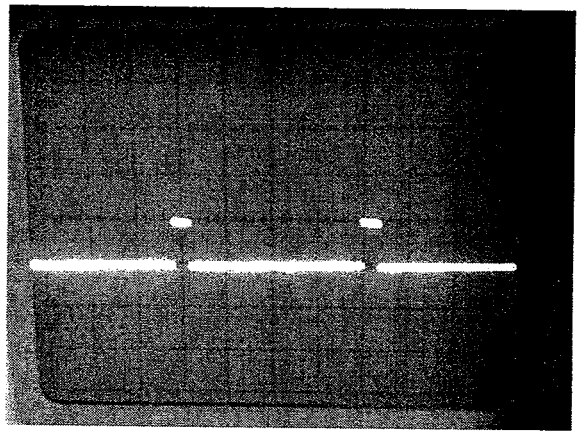
เมื่อป้อนสัญญาณพัลส์ 1.5 ms. โดยที่เซอร์โวมอเตอร์จะอยู่ที่ 0 องศา คาบเวลา 20 ms

การป้อนสัญญาณพัลส์ให้เซอร์โวมอเตอร์ (5 ms / DIV)



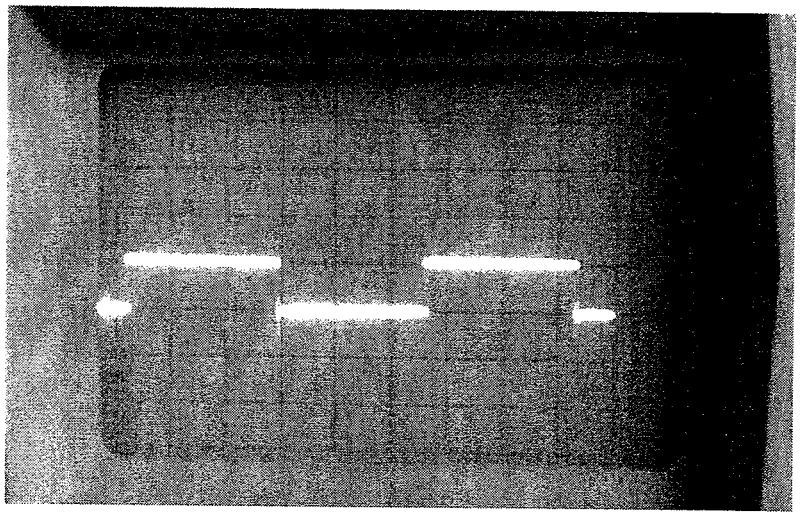
เมื่อป้อนสัญญาณพัลส์ 1.25 ms. โดยที่เซอร์โวมอเตอร์จะอยู่ที่ -45 องศา คาบเวลา 20 ms

การป้อนสัญญาณพัลส์ให้เซอร์โวมอเตอร์ (5 ms / DIV)



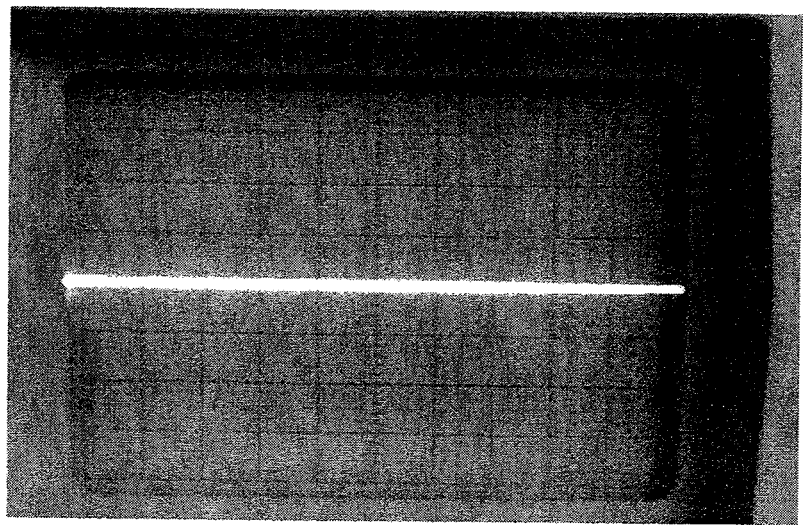
เมื่อป้อนสัญญาณพัลส์ 1.75 ms. โดยที่เซอร์โวมอเตอร์จะอยู่ที่ 45 องศา คาบเวลา 20 ms

6.2 การทดลองจ่ายสัญญาณพัลส์ให้กับเซนเซอร์



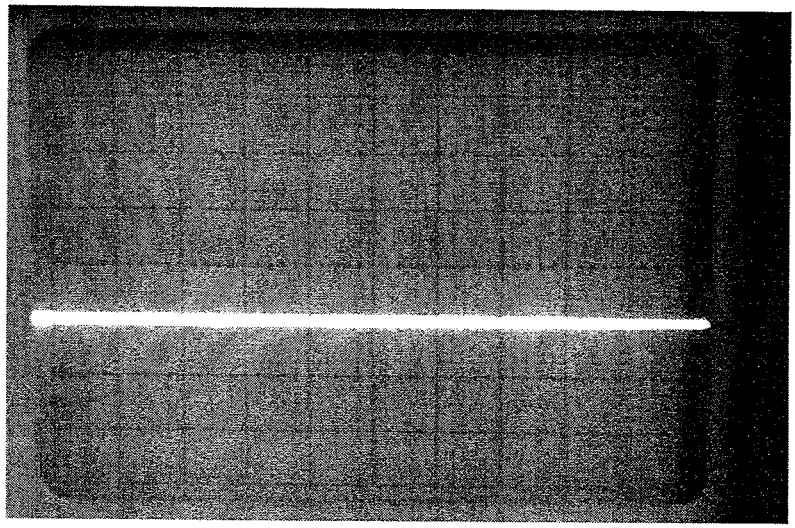
สร้างสัญญาณพัลส์ขนาด 5 โวลต์ ความถี่ 38 KHz จากวงจรไมโครคอลโทลเลอร์ เพื่อป้อนให้กับวงจร Sensor (5 us/DIV)

สัญญาณ Input ที่วัดได้จาก Sensor เมื่อไม่พบวัตถุ



เมื่อ Sensor ไม่พบวัตถุจะได้สัญญาณขนาด 5โวลต์ ดังรูป ป้อนเข้าสู่วงจรไมโครคอลโทลเลอร์ (5 V/DIV)

สัญญาณ Input ที่วัดได้จาก Sensor เมื่อพบวัตถุ



เมื่อ Sensor พบวัตถุจะได้สัญญาณขนาด 0 โวลต์ ดังรูป ป้อนเข้าสู่วงจรไมโครคอลโทลเลอร์ (5 V/DIV)

บทที่ 7

วิเคราะห์และสรุปผลการดำเนินโครงงาน

ผลการดำเนินโครงงาน

การดำเนินโครงงานสำเร็จตามวัตถุประสงค์ที่ได้กำหนดไว้คือ

1. สามารถสร้างหุ่นยนต์แมลงตามที่ได้ออกแบบไว้ได้
2. หุ่นยนต์แมลงสามารถเดินหน้า ถอยหลัง เลี้ยวซ้ายและเลี้ยวขวาได้โดยระบบออโตเมติก
3. หุ่นยนต์สามารถตรวจจับสิ่งกีดขวางได้

รายละเอียดของหุ่นยนต์แมลง

ลักษณะ	หุ่นยนต์เดิน 6 ขา
ลักษณะการเดิน	ประยุกต์จากการเดินแบบ 3 จุด
โครงสร้าง	อะครีลิค
ต้นกำลัง	เซอร์โวมอเตอร์
ระบบควบคุม	ไมโครคอนโทรลเลอร์แบบ 8 บิต
ลักษณะการควบคุม	ระบบออโตเมติก
รายละเอียดการเดิน	สามารถเดินหน้า ถอยหลัง เลี้ยวซ้าย และเลี้ยวขวาได้

วงจรตัวรับสัญญาณอินฟราเรด

วงจรตัวรับสัญญาณอินฟราเรด มีการพัฒนาโดย เพิ่มทรานซิสเตอร์ (2SC1815) เพื่อให้สามารถส่งสัญญาณอินฟราเรดได้ในระยะทางที่ไกลมากขึ้น ทำให้หุ่นยนต์สามารถหลบหลีกสิ่งกีดขวางได้ดีขึ้น โดยความถี่ของสัญญาณอินฟราเรดที่ใช้นั้นอยู่ในช่วง 38 kHz ระบบเซนเซอร์ยังมีความผิดพลาดอยู่บ้าง เนื่องจากอุปกรณ์ที่ใช้อยู่ยังไม่มีประสิทธิภาพเท่าที่ควรแก่ความต้องการ เซนเซอร์แบบใช้แสงอินฟราเรดนั้นมีทั้งข้อดีและข้อเสีย ในส่วนของข้อดีคือ วงจรควบคุมไม่ซับซ้อน โมดูลทั้งตัวรับและตัวส่งมีราคาถูกและหาได้ง่ายในท้องตลาด แต่ก็มีข้อเสียตรงที่เมื่อนำมาใช้ตรวจจับวัตถุที่เป็นสิ่งกีดขวางจะมีความผิดพลาดอยู่บ้างในกรณีที่วัตถุมีขนาดเล็กและแคบ เช่นขาโต๊ะ หรือไม่ก็อาจมีปัญหาที่สัญญาณรบกวน (Noise) ซึ่งมีความถี่ในย่านอินฟราเรดพอดีเข้ามาที่โมดูลตัวรับ ซึ่งอาจทำให้การทำงานของหุ่นยนต์ผิดพลาดได้

สรุปผลการดำเนินโครงงาน

วงจรที่ใช้ในโครงงานนี้ประกอบด้วย 2 วงจร คือ วงจรรวม (Main board) และวงจรรับสัญญาณอินฟราเรด ซึ่งวงจรทั้งสองสามารถทำงานร่วมกันได้เป็นอย่างดีกล่าวคือ สัญญาณอินฟราเรดที่ใช้ตรวจจับสิ่งกีดขวางจากวงจรรับ สามารถส่งผ่านกลับมายังวงจรรวม (Main board) เพื่อที่จะส่งเข้าไมโครคอนโทรลเลอร์ (dsPIC30F2010) เพื่อประมวลผลและสั่งรูปแบบการเคลื่อนที่ของหุ่นยนต์

เทคโนโลยีด้านหุ่นยนต์เดินยังเป็นเทคโนโลยีที่ยังคงจำกัดอยู่ในวงแคบซึ่งมักพบการพัฒนาในเฉพาะต่างประเทศเท่านั้น โดยเฉพาะในประเทศไทยยังคงไม่มีการพัฒนาอย่างจริงจัง แต่อาจพบได้บ้างประเภทหุ่นยนต์เดินประกอบขนาดเล็ก (Kitrobot) ซึ่งมีการจัดการแข่งขันตามสถาบันต่างๆในการทำโครงงานนี้ผู้จัดทำได้เริ่มจากการหาข้อมูลของหุ่นยนต์แบบต่างๆและศึกษาพฤติกรรมการเดินของสัตว์เพื่อที่จะนำมสร้างหุ่นยนต์ต้นแบบโดยใช้ความรู้ด้านเครื่องกล สร้างโครงสร้างและระบบกลไกที่ใช้ขับรวมทั้งได้นำอุปกรณ์ควบคุมมาควบคุมกลไกต่างๆให้เคลื่อนที่ตามที่ได้เขียนโปรแกรมไว้โดยใช้อุปกรณ์ตรวจจับคอยตรวจสอบการเคลื่อนที่ของกลไก ทำให้หุ่นยนต์สามารถเดินได้ตามที่ออกแบบไว้โดยอัตโนมัติ