การออกแบบเพื่อควบคุมการเคลื่อนไหวของหุ่นยนต์โดยใช้ PC

DESIGNINIG FOR CONTROL MOBILE OF ROBOT WITH PC

โดย

นายณรงค์ เมตไตรพันธ์

นายบุญฤทธิ์ พงษ์ประเสริฐ

นายพิจารณ์ บูลสาร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

สาขาวิชาเทคโนโลยีโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034779

หัวข้อปริญญานิพนธ์ การออกแบบเพื่อควบคุมการเคลื่อนไหวของหุ่นยนต์โดยใช้ PC

ชื่อนักศึกษา นายณรงค์ เมตไตรพันธ์
นายบุญฤทธิ์ พงษ์ประเสริฐ
นายพิจารณ์ มูลสาร

อาจารย์ที่ปรึกษา อาจารย์มนชนก ศรีเสือขาม
อาจารย์ไพศาล สิทธิโยภาสกุล

ภาควิชา เทคนิคอุตสาหกรรม

ปีการศึกษา 2537

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ
( )

.....กรรมการ
(มนชนก ศรีเสือขาม)

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การออกแบบเพื่อควบคุมการเคลื่อนไหวของหุ่นยนต์โดยใช้ PC

โดย นายณรงค์ เมตไตรพันธ์ รหัส 36012051
นายบุญฤทธิ์ พงษ์ประเสริฐ รหัส 36012055
นายพิจารณ์ มูลสาร รหัส 36012058

อาจารย์ที่ปรึกษา อาจารย์มนชนก ศรีเสือขาม
อาจารย์ไพศาล สิทธิโยภาสกุล

ปีการศึกษา 2537

บทคัดย่อ

ปริญญานิพนธ์นี้ได้อธิบายถึง ทฤษฎีหุ่นยนต์เบื้องต้นและการใช้งานหุ่นยนต์ ET-18 และการพัฒนาหุ่นยนต์ทั้งทางด้านฮาร์ดแวร์และซอฟท์แวร์ โดยการนำเอา PC เข้ามาควบคุมการเคลื่อนไหวของแขนหุ่นยนต์ เพื่อวัตถุประสงค์ที่จะทำให้สามารถใช้ภาษาที่ง่ายขึ้นในการควบคุม แทนภาษาเครื่องซึ่งมีความยุ่งยากซับซ้อน ซึ่งทำให้ง่ายต่อการพัฒนาหุ่นยนต์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DESIGNING FOR CONTROL MOBILE OF ROBOT WITH PC

BY	Mr. NARONG METTHAIPUNT	NO. 36012.51
	Mr. BOONYARIT PONGPRASERT	NO. 36012055
	Mr. PHIJARN MOONSARN	NO. 36012058
ADVISER	Mr. PHAISARN SITTIYOPARTKOON	
	Miss. MONCHARNOK SRISEARKARM	
YEAR	1995	

ABSTRACT

This project describes the principle of mobile of robot , and application ET-18 mobile robot , and development of the mobile robot in hardware and software. The personal computer is used for control arm driving mobile for using easy. For the purpose of we can use more easy language to control instead of the machine language, which so difficult and intricate, and we will develop the robot in the future with ease.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของอาจารย์มนชนก ศรีเสือชาม และ อาจารย์ไพศาล สิทธิโยภาสกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้คำแนะนำและข้อคิดเห็นต่างๆ ของการวิจัยมาด้วยดีตลอด และเนื่องจากทุนการวิจัยครั้งนี้บางส่วนได้รับมาจากทุนอุดหนุนการวิจัยของบัณฑิตวิทยาลัย จึงขอขอบพระคุณคุณบัณฑิตวิทยาลัยมา ณ ที่นี้ด้วย

ท้ายนี้ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา-มารดา ซึ่งสนับสนุนในด้านการเงินและให้กำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

นายณรงค์ เมตไตรพันธ์
นายบุญฤทธิ์ พงษ์ประเสริฐ
นายพิจารณ์ มูลสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อภาษาไทย

บทคัดย่อภาษาอังกฤษ

กิตติกรรมประกาศ

บทที่	หน้าที่
1. บทนำ	1
2. ความรู้ทั่วไปเกี่ยวกับหุ่นยนต์	3
3. การใช้งานต่างๆ ของหุ่นยนต์ ET-18	12
4. ระบบต่างๆ ของหุ่นยนต์ ET-18	21
5. ทฤษฎีและการใช้งาน RS232 C	23
6. STEPPING MOTOR และการควบคุม	55
7. พื้นฐานแขนกล	62
8. การใช้ PERSONAL COMPUTER ควบคุมแขนกล	79
9. สรุปการวิจัยและข้อเสนอแนะ	91
เอกสารอ้างอิง	92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ประวัติความเป็นมา

เนื่องจากปัจจุบันนี้ในประเทศที่พัฒนาแล้ว ได้มีการนำหุ่นยนต์มาใช้แทนมนุษย์อย่างมากมาย เช่น ในงานที่เสี่ยงต่ออันตราย งานที่ต้องทำซ้ำๆ เป็นต้น เนื่องจากเทคโนโลยีทางด้านนี้มีราคาแพงมาก จึงเป็นการยากที่จะสรรหามาโดยเฉพาะประเทศที่กำลังพัฒนา ทางหนึ่งที่จะช่วยให้ก้าวทันเทคโนโลยี คือ การเร่งส่งเสริมการศึกษา การพัฒนาหุ่นยนต์ให้สามารถทำงานได้อย่างมีประสิทธิภาพตามนโยบายพึ่งตนเอง เพื่อเป็นแนวทางในการพัฒนาครั้งต่อไป

1.2 วัตถุประสงค์

1. เพื่อศึกษาระบบต่างๆ ของหุ่นยนต์
2. เพื่อศึกษาถึงการใช้งานภายนอกของหุ่นยนต์
3. เพื่อทำการพัฒนาทางด้านฮาร์ดแวร์และซอฟแวร์
4. ใช้ Personal Computer (PC) ควบคุมการเคลื่อนที่ของแขนหุ่นยนต์

1.3 ขอบเขตของการวิจัย

เนื้อหาของโครงงานวิจัย สามารถแบ่งออกได้เป็น 3 ส่วนใหญ่ๆ คือ

1.3.1 ทฤษฎีและการใช้งานเบื้องต้นของหุ่นยนต์

1.3.2 ฮาร์ดแวร์ คือ ระบบต่างๆ ของหุ่นยนต์และการอินเตอร์เฟส

1.3.3 ซอร์ฟแวร์ คือ โปรแกรมที่ใช้ควบคุมการเคลื่อนไหว

1.4 การดำเนินงาน

เทอม 1

- ศึกษาทฤษฎีเบื้องต้นของหุ่นยนต์
- ศึกษาการใช้งานปุ่มต่างๆ ของหุ่นยนต์ ET-18
- ศึกษาลักษณะโครงสร้างภายในของหุ่นยนต์ ET-18
- สามารถอธิบายลักษณะโครงสร้างและการทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ศึกษาถึงระบบต่างๆ ของหุ่นยนต์ ET-18
- ทำการประยุกต์ใช้งานโดยการนำ PC เข้ามาควบคุมการเคลื่อนไหว

1.5 ปัญหาที่พบ

ปัญหาที่พบในการวิจัยโครงงานนี้อันดับแรกพบว่า แบตเตอร์รี่ที่ใช้ในตัวหุ่นยนต์หมดสภาพใช้การไม่ได้ จึงทำให้แบตเตอร์รี่เก็บไฟไม่อยู่ ทำให้เวลาใช้งานหุ่นยนต์จะต้องใช้แหล่งจ่ายไฟจากภายนอกตลอดเวลา และประการที่สองก็คือการขาดแคลนข้อมูลเกี่ยวกับหุ่นยนต์ตัวนี้ทำให้การวิจัยเกิดความล่าช้า และประการสุดท้ายคือการขาดความรู้ความเข้าใจในระบบของหุ่นยนต์ ทำให้ต้องเสียเวลาในการเรียนรู้

1.6 ประโยชน์ที่ได้รับ

ประโยชน์ที่ได้รับขึ้นอยู่กับการนำไปประยุกต์ใช้งานโดยเฉพาะทางด้านอุตสาหกรรม จากการศึกษาทำให้เราทราบถึงการบังคับควบคุมการทำงานของหุ่นยนต์และการโปรแกรมให้หุ่นยนต์ทำงานตามที่ต้องการ และเข้าใจถึงหลักการนำเอาทฤษฎีมาใช้งานควบคุม stepping motor

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้ทั่วไปเกี่ยวกับหุ่นยนต์

ในขณะที่ประเทศไทยก้าวไปสู่ประเทศอุตสาหกรรมใหม่ ค่าจ้างแรงงานกำลังเพิ่มขึ้นทุกขณะทางออกของประเทศอุตสาหกรรมต่อปัญหานั้นคือ การทดแทนแรงงานโดยมนุษย์หุ่นยนต์หรือระบบอัตโนมัติต่างๆ โรงงานอุตสาหกรรมทั้งหลายจึงลดค่าใช้จ่ายเกี่ยวกับการจัดการด้านแรงงานได้อย่างมาก ขณะนี้เมืองไทยเริ่มมีผู้ที่นำเอาหุ่นยนต์เข้ามาใช้ในการผลิตหลักบ้างแล้ว เช่น โรงงานประกอบแผ่นวงจรพิมพ์ โรงงานประกอบรถยนต์ ฯลฯ

ดังนั้นผู้ที่มีส่วนเกี่ยวข้องกับอุตสาหกรรมหรือผู้ที่สนใจควรที่จะหาความรู้เกี่ยวกับเทคโนโลยีใหม่ๆ ที่ก้าวเข้ามาสู่บ้านเราส่วนผลกระทบที่เกิดขึ้นกับการจ้างงานก็ควรจะมาพิจารณาแก้ไขล่วงหน้าก่อนที่จะเกิดปัญหาต่างๆ ตามมา ดังที่เคยเกิดกับประเทศมหาอำนาจทางอุตสาหกรรมอย่างอเมริกาและญี่ปุ่นมาแล้ว

ความเป็นมาของหุ่นยนต์

เนื่องจากระบบคอมพิวเตอร์ได้รับการพัฒนาอย่างสูงในทศวรรษหลัง มีการประยุกต์ใช้เพื่อควบคุมเครื่องจักรกลให้สะดวกสบายมากขึ้น การเปลี่ยนรูปแบบการทำงานของเครื่องจักรสามารถทำเพียงแค่เปลี่ยนโปรแกรมของคอมพิวเตอร์ที่ควบคุมเท่านั้น โดยไม่ต้องยุ่งกับส่วนฮาร์ดแวร์เลย

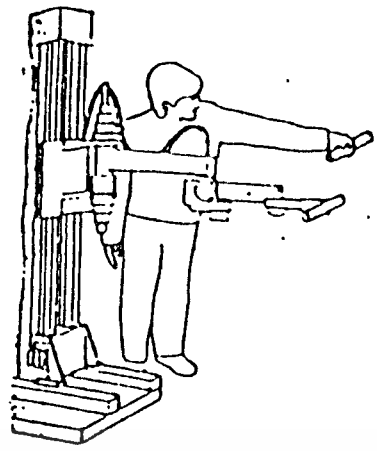
เครื่องจักรที่สามารถทำงานได้หลายอย่าง ผู้สร้างจะพยายามสร้างให้ใกล้เคียงกับมนุษย์ เพราะมนุษย์เปรียบเสมือนเครื่องจักรที่มีประสิทธิภาพสูงสุด บรรดาหุ่นยนต์หลายตัวจึงถูกสร้างให้มีลักษณะคล้ายแขนหรือมือของคน เพื่อความอิสระในการเคลื่อนไหวดังรูป

ลักษณะงานที่นำเอาหุ่นยนต์มาใช้มักจะเป็นงานที่ซ้ำๆ กัน หากใช้คนจะเกิดความเบื่อหน่ายจำเจจนก่อให้เกิดความผิดพลาดได้ง่าย เช่น การผลิตแผงวงจร การจัดเรียงชิ้นงานในชั้นวางของ เป็นต้น หรืออีกลักษณะหนึ่งก็คือ งานที่เป็นอันตราย ต่อสุขภาพ

ความแตกต่างของระบบอัตโนมัติกับหุ่นยนต์

ระบบอัตโนมัติ คือระบบที่สามารถสร้างงานด้วยตัวเอง โดยปราศจากการควบคุมของมนุษย์ในระหว่างการทำงานเพราะเพียงแต่กำหนดเป้าหมายเริ่มต้นของการทำงานโดยที่ลักษณะการทำงานไม่เปลี่ยนแปลง ตัวอย่างเช่น ระบบ PLC คอลโทรลเลอร์ PI,PID เป็นต้น ในขณะที่หุ่นยนต์ซึ่งเป็นระบบอัตโนมัติอย่างหนึ่งแต่สามารถเปลี่ยนแปลงลักษณะการทำงานได้ เช่นเปลี่ยนจากงานพ่นสีมาเป็นงานเชื่อมหรืองานประกอบแผ่นวงจรแบบต่าง ๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 1 ลักษณะการทำงานของหุ่นยนต์เปรียบเทียบกับมนุษย์

ลักษณะงานที่นำเอาหุ่นยนต์มาใช้มักจะเป็นงานที่ซ้ำๆ กัน หากใช้คนจะเกิดความเบื่อหน่ายจำเจจนก่อให้เกิดความผิดพลาดได้ง่าย เช่น การผลิตแผงวงจร การจัดเรียงชิ้นงานในชั้นวางของ เป็นต้น หรืออีกลักษณะหนึ่งก็คือ งานที่เป็นอันตราย ต่อสุขภาพ

<u>ความแตกต่างของระบบอัตโนมัติกับหุ่นยนต์</u>

ระบบอัตโนมัติ คือระบบที่สามารถสร้างงานด้วยตัวเอง โดยปราศจากการควบคุมของมนุษย์ในระหว่างการทำงานเพราะเพียงแต่กำหนดเป้าหมายเริ่มต้นของการทำงานโดยที่ลักษณะการทำงานไม่เปลี่ยนแปลง ตัวอย่างเช่น ระบบ PLC คอลโทรลเลอร์ PI,PID เป็นต้น ในขณะที่หุ่นยนต์ซึ่งเป็นระบบอัตโนมัติอย่างหนึ่งแต่สามารถเปลี่ยนแปลงลักษณะการทำงานได้ เช่นเปลี่ยนจากงานพ่นสีมาเป็นงานเชื่อมหรืองานประกอบแผ่นวงจรแบบต่าง ๆ กัน

แสดงการติดตั้งใช้งานของห่นยนต์ที่ใช้กับงานอุตสาหกรรมขนตอนการทำงานจะเริ่มจากการสร้างโปรแกรมในการควบคุม โดยผู้เชี่ยวชาญในการทำงานชิ้นนน ๆ หลังจากป้อนโปรแกรมที่ถูกต้องสมบูรณ์เข้าไปในคอมพิวเตอร์แล้ว ความต้องการด้านบุคลากรหรือผู้เชี่ยวชาญต่าง ๆ ก็หมดไป หน้าที่การดำเนินการผลิตและควบคุมเครื่องจักรต่าง ๆ จะตกอยู่กับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ลักษณะโครงสร้างของข้อต่อจะแตกต่างกันไปตามการใช้งาน การออกแบบข้อต่อนี้ต้องคำนึงความทนทาน ความคล่องตัว ความเที่ยงตรงของการใช้งาน ซึ่งแบ่งออกได้เป็น 3 ประเภท

1 ข้อต่อแบบพริสเมติก (Prismatic joint) เป็นข้อต่อที่ทำให้ท่อนแขนทงสองสามารถเคลื่อนที่ได้ในแนวเส้นตรง

2 ข้อต่อแบบหมุน (Revolute joint) เป็นข้อต่อที่ทำให้ท่อนแขนทงสองสามารถเคลื่อนที่ได้เป็นวงกลม

3 ข้อต่อแบบบอลและซ็อกเก็ต (Ball and Socket joint) เป็นข้อต่อที่ทำให้ท่อนแขนสามารถเคลื่อนที่ได้มากกว่า 3 แกน ซึ่งต่างจากสองแบบแรกที่เคลื่อนที่ได้เพียงแกนเดียว

ข้อต่อแบบพริสเมติกจะมีความทนทาน และสามารถรับแรงได้มากที่สุด เมื่อเปรียบเทียบกับข้อต่อแบบต่าง ๆ ทงหมด

ความสามารถในการเคลื่อนไหวของแขนกลนี้ สามารถกำหนดได้จากโครงสร้างของแขนกลเองและโปรแกรมคอมพิวเตอร์ที่ใช้ในการควบคุม

จากโครงสร้างของแขนกลที่เห็นได้ชัดจากภายนอกสามารถออกแบบให้จำนวนแกนของแขนกลที่สามารถเคลื่อนที่ได้โดยไม่ขึ้นกับแกนอื่น ความสามารถในการเคลื่อนไหวโดยมิได้ขึ้นอยู่กับแกนอื่นนี้ เรียกว่า องศาแห่งความอิสระ (degree of freedom)

ถ้าจำนวนองศาแห่งความอิสระของหุ่นยนต์มาก ความสามารถในการเคลื่อนไหวก็จะสูงตามไปด้วย

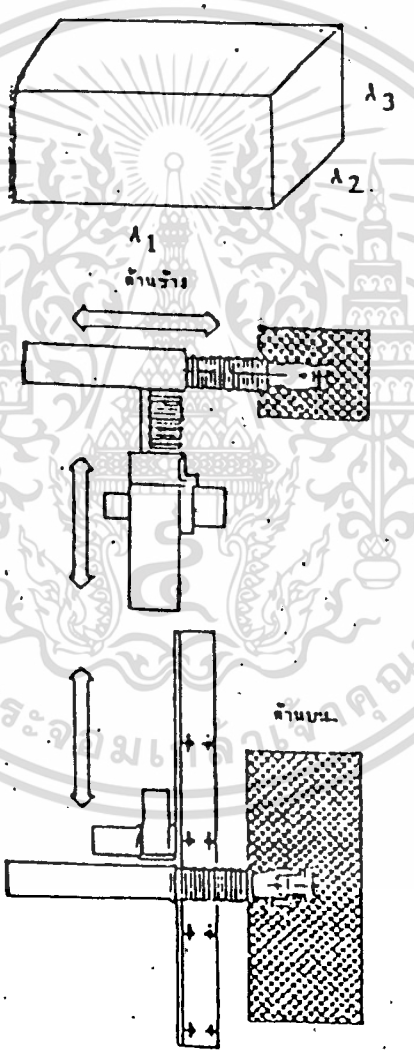
<u>ประเภทของแขนกล</u>

เนื่องจากแขนกลที่ประกอบด้วยข้อต่อที่แตกต่างกัน ทำให้การเคลื่อนที่มีลักษณะต่างกันออกไป จึงสามารถจำแนกประเภทของแขนกลด้วยความแตกต่างกันของพื้นที่ครอบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์แบบต่าง ๆ

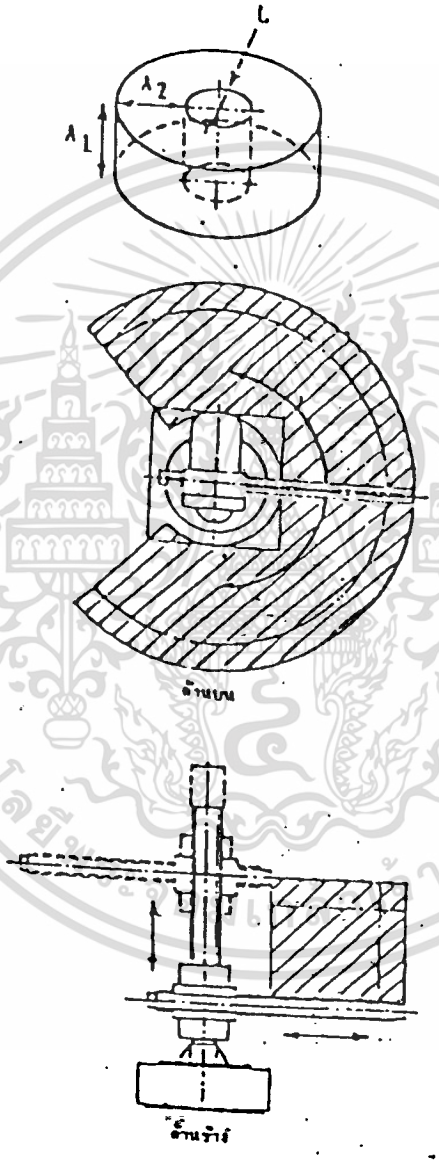
1.หุ่นยนต์แบบคาทีเซียล (Cartesian) เป็นหุ่นยนต์ที่มีการเคลื่อนที่ตามแกน 3 มิติ จึงมีพื้นที่ครอบคลุมการทำงานเป็นรูปลูกบาศก์ หุ่นยนต์แบบนี้ใช้ข้อต่อแบบพริสเมติกจึงมีความทนทานแข็งแรงสูงมาก



รูปที่ 2 แสดงการเคลื่อนที่ของหุ่นยนต์แบบ CARTESIAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

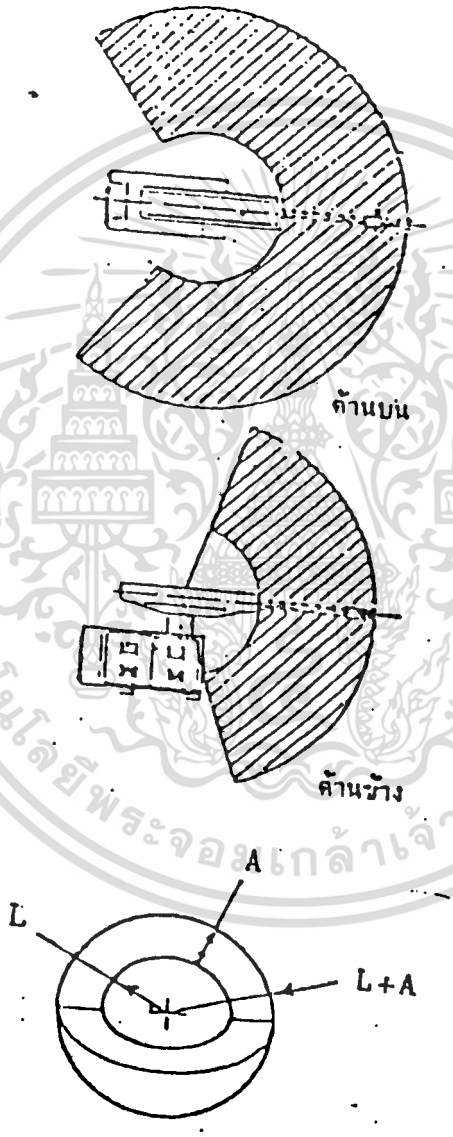
7 2.หุ่นยนต์แบบไซลินดริคัล (Cylindrical) จะมีฐานที่หมุนได้ และแขนที่เคลื่อนที่ได้ในแนวดิ่งและแนวนอน พื้นที่ครอบคลุมการทำงานจึงเป็นรูปทรงกระบอก



รูปที่ 3 แสดงการเคลื่อนที่ของหุ่นยนต์แบบ CYLINDRICAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.หุ่นยนต์แบบสเพียริคัล (Spherical) นอกจากจะมีฐานที่หมุนได้แล้ว ยังมีข้อต่อที่ควบคุมการก้มเงยของท่อนแขนเพิ่มขึ้นมา จึงมีพื้นที่ครอบคลุมการทำงานเป็นทรงกลม

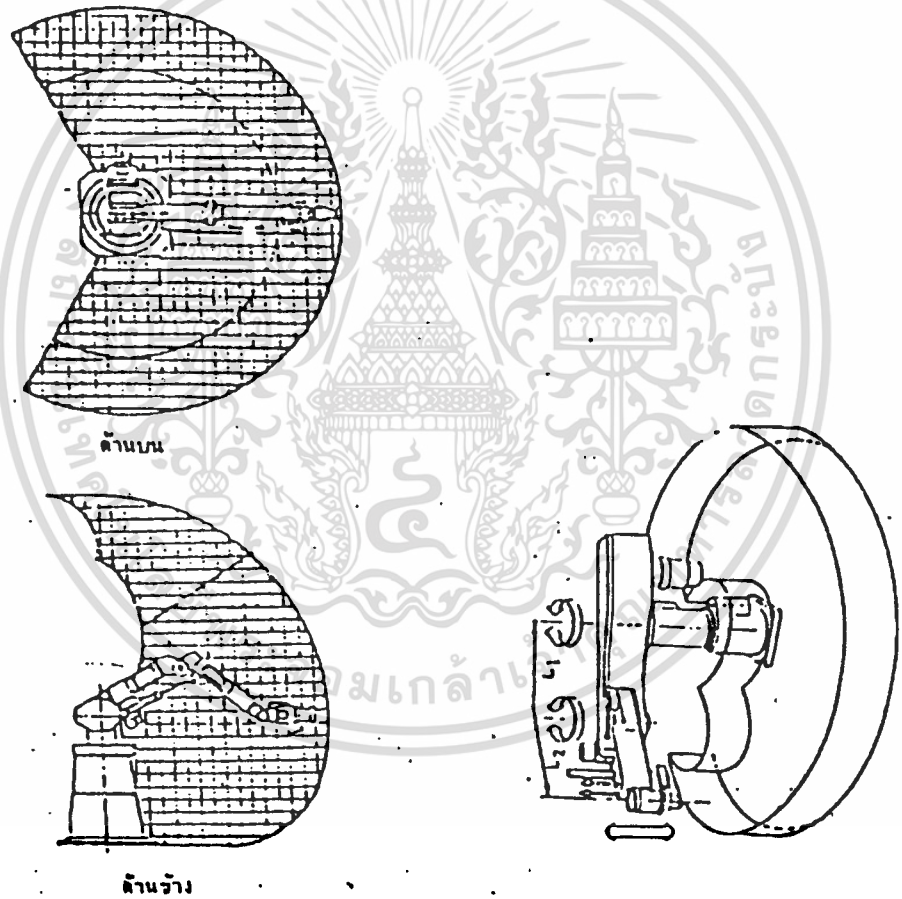


รูปที่ 4 แสดงการเคลื่อนที่ของหุ่นยนต์แบบ SPHERICAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

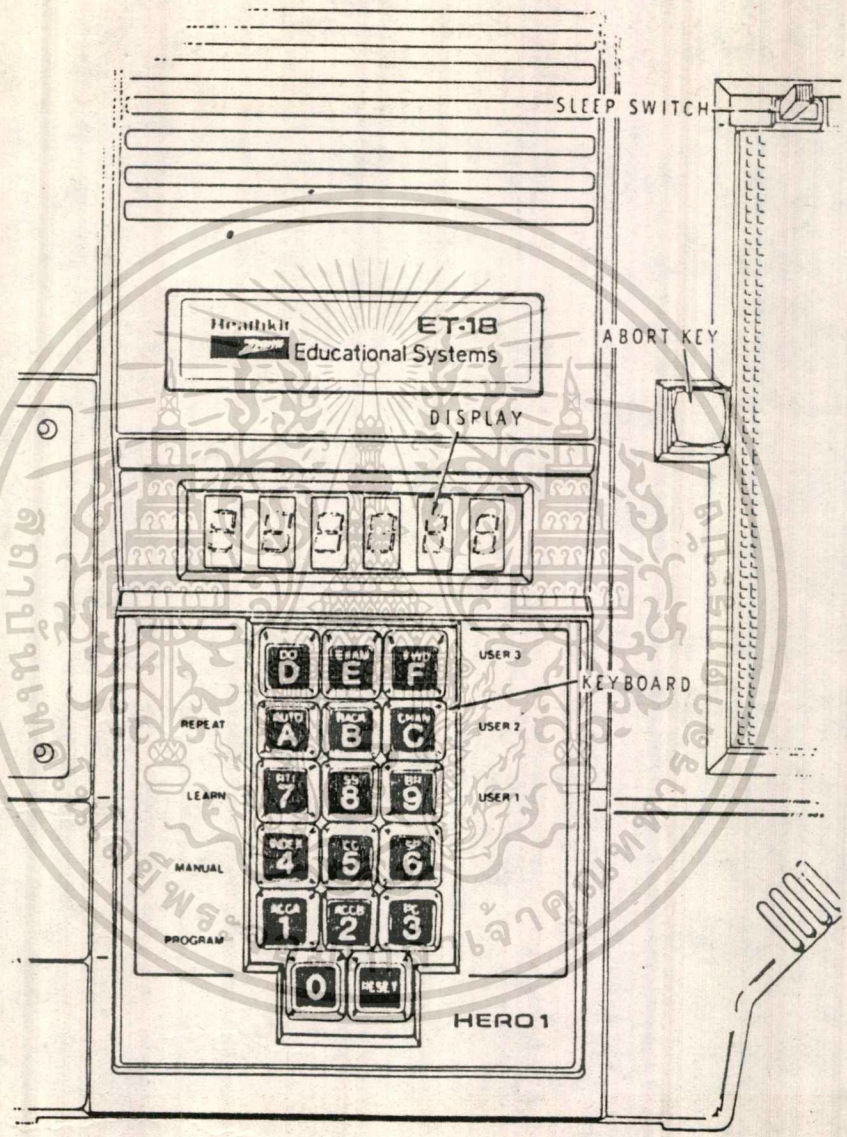
4.หุ่นยนต์แบบอาติคิวเลต (Articulated) เป็นหุ่นยนต์ที่มีแต่ข้อต่อแบบหมุนเพียงอย่างเดียว และยังแบ่งเป็นแนวตงหรือแนวนอน

หุ่นยนต์แบบสเพียร์คัลจะมีความคล้ายคลึงกับแขนมนุษย์มากที่สุด แต่จะไม่แข็งแรงเท่าหุ่นยนต์แบบคาร์ทีเซียน หุ่นยนต์แต่ละแบบจึงต้องคำนึงถึงภาระการทำงาน โมเมนต์ในการออกแบบและเลือกใช้ให้มีประสิทธิภาพสูงสุดและป้องกันการเกิดทรุดตัวของตำแหน่งหากภาระหนักจนเกินไป



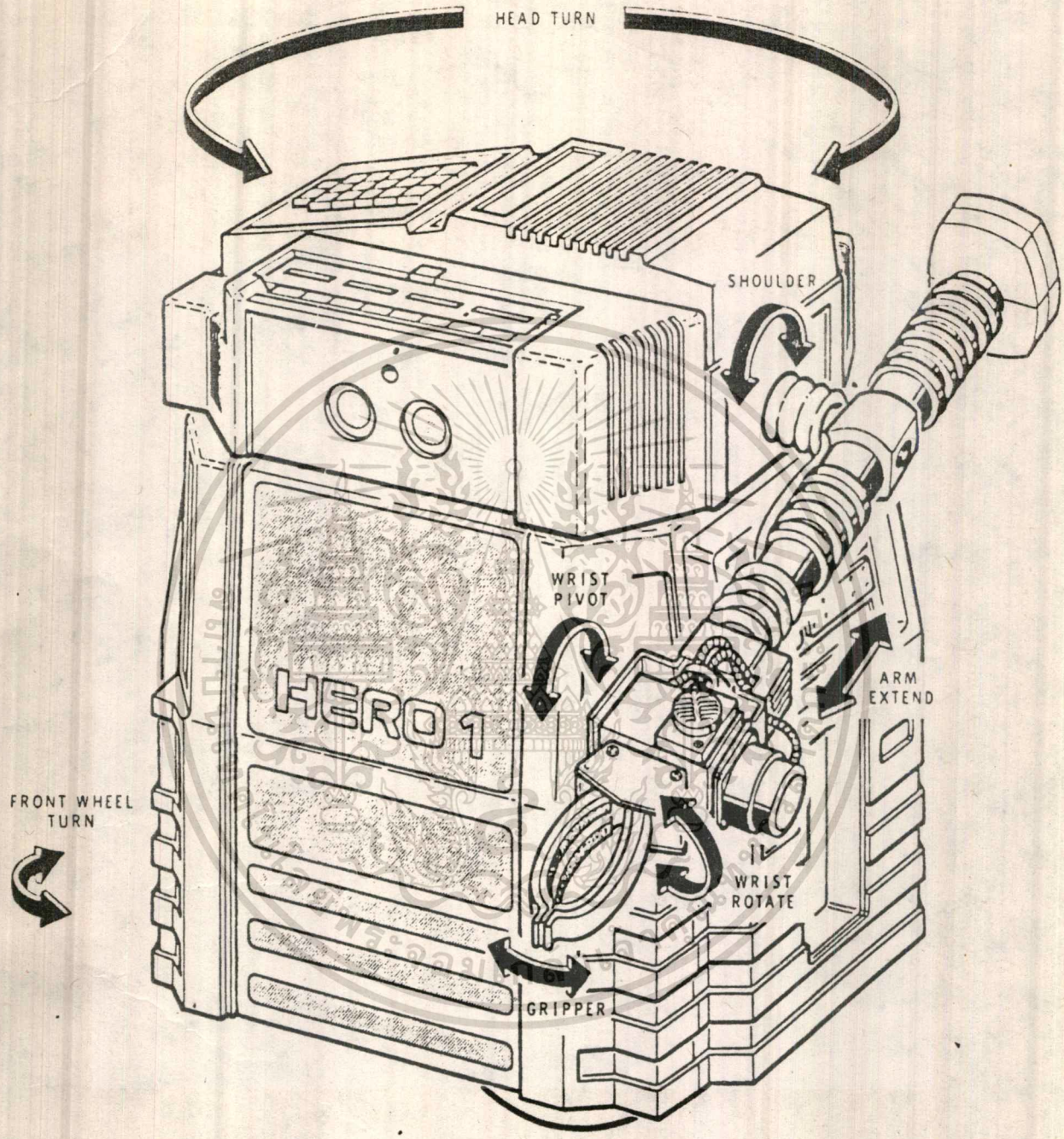
รูปที่ 5 แสดงการเคลื่อนที่ของหุ่นยนต์แบบ ARTICULATED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
034779



รูปที่ 7 แสดงปุ่มการใช้งานฟังก์ชั่นต่างๆ ของหุ่นยนต์ ET-18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 แสดงให้เห็นรูปร่างของหุ่นยนต์ ET-18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การใช้งานปุ่มต่างๆ ของหุ่นยนต์ ET-18

ปุ่มและจอภาพ

ปุ่มคือ วิธีการสำคัญของคุณในการที่จะติดต่อสื่อสารกับ computer และ Robat ปุ่มปุ่มบางอันจะมีเพียงจุดประสงค์เดียว ปุ่มอื่น ๆ ก็สามารถให้บริการตามจุดประสงค์อื่น ๆ ได้ขึ้นอยู่กับว่ากำลังอยู่ในสภาวะอย่างไรเมื่อกดแล้ว แสดงให้เห็นถึงปุ่มต่าง ๆ ดังต่อไปนี้

ปุ่ม O และ RESET

จะใช้เพียงจุดประสงค์เดียว ปุ่ม O จะใช้เป็นเลข O อย่างเดียวเท่านน นี่คือปุ่มคำสั่งหรือป้อน data หรือหมายเลขหน่วยความจำ ปุ่ม RESET จะเป็นการทำให้มันกลับไปเริ่มต้นใหม่ เมื่อพร้อมที่จะทำงานใน mode ที่เปลี่ยน

ปุ่ม ABOUT

จุดมุ่งหมายอันที่ 3 ต่อไปคือ ปุ่ม ABOUT ตงอยู่บนแผงทดลองวงจร คุณสามารถใช้มันในกรณีที่ Robot เริ่มที่จะทำสิ่งใดสิ่งหนึ่งผิดหรือไม่เป็นดังที่คาดหมายไว้ เมื่อกดปุ่มนี้แล้ว computer จะหยุดไม่ว่าอะไรก็ตามที่กำลังทำอยู่จะหยุดหมด ปราศจากสิ่งที่เป็นอยู่ในเวลานี้ การเปลี่ยน program นนมันจะทำการ RUN เอง Robot จะหยุดการเคลื่อนไหว การพูดประสาทสัมผัสทงวหมดการรับรู้หรือการคำนวณจอภาพจะแสดงคิว่า abort 13 เป็นเวลา 1 วินาที และจะทำให้โปรแกรมนนต่อไป โดยปกติที่ตำแหน่งนี้ computer จะถูกแทนที่ด้วยคำสั่งบริการก่อนที่ปุ่ม abort จะถูกกด โปรมแกรมเมอร์ที่มีประสบการณ์สามารถที่จะใช้ตำแหน่งที่ debugging program ของ Robot

สวิทช์ SLEEP

sleep สวิทช์จะเป็นสวิทช์แบบเลื่อน ไม่ใช้ปุ่มกด แต่มันสามารถให้บริการได้เช่นเดียวกับหน้าที่การทำงานของป่มกด สวิทช์นี้ ในตำแหน่ง sleep โปรแกรม computer ของ Bobot จะเข้าไปในการทำงานแบบ sleep คือจะรับคำสั่งใน program หรือ คำสั่ง Reset จากแฟ้มพิมพ์สังเกตถ้าหากว่าสวิทช์อยู่ในตำแหน่ง sleep และคุณกดปุ่ม resel หรือเปิด Robot Robot ก็จะ sleep มันจะเป็นจะเป็นอย่างนั้นตลอดคือ จะปิด ในทางกลับกันถ้า Robot อยู่ในการทำงานปกติธรรมดาให้เคลื่อน sleep switch ไปที่ตำแหน่ง sleep Robot ก็จะเปิดการทำงานภายใน 10 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Multipurpose keys

ปุ่มที่เหลืออยู่ทงหมดคือ ปุ่ม Multipurpose keys จุดมุ่งหมายอันหนึ่ง พวกมันจะถูกเติมลงไปแทนที่หมายเลขสำหรับการ program computer พวกมันจะถูกใช้แทนที่หมายเลข O ตลอดถึง F เลขเหล่านี้จุลูกใช้แทนที่ คำสอนหรือข้อมูล ในขณะที่ทำการป้อนโปรแกรม หมายเลข ๆ เดียวจำเป็นมากสำหรับการโปรแกรม ปุ่มกดสามารถให้บริการอย่างอื่นได้ เมื่อยังไม่ได้ใช้การการโปรกรม computer จะมีหมายเลขโหมดการทำงานต่าง ๆ คนสามารถที่จะริเริ่มแต่ละ mode การทำงานนี้ได้โดยการกดปุ่มที่เกี่ยวกับ mode ตัวมันเอง ปุ่ม mode พื้นฐานสำคัญต่าง ๆ ใหญ่จะเป็นไปตามนน ๆ ถ้า computer อยู่ใน execlutive Mode เมื่อคุนกดปุ่มเหล่านี้ computer จะเปลี่ยน mode ความแตกต่างของ mode จะได้ตอบหลังจากส่วนนี้แล้ว

key 1 ถ้าคุณกด 1 computer จะเปลี่ยนเป็น Program Mode

key 3 ถ้าคุณกด 3 computer จะเปลี่ยนเป็น Mode ulility

key 4 ถ้าคุณกด 4 computer จะเปลี่ยนเป็น Manual Mode

key 7 ถ้าคุณกด 7 computer จะเปลี่ยนเป็น Learn Mode

key A ถ้าคุณกด A computer จะเปลี่ยนเป็น Mode ทำซ้ำ

key a, c, f (ผู้ใช้ 1, 2, 3) คือกำหนดผู้ใช้ปุ่มจุดมุ่งหมายพิเศษ

จะยอมให้เพียงปุ่มเดียวเข้าไปตัวโปรแกรม ว่าจะเป็นตำแหน่ง 0030, 0033 และ 0036 ตามลำดับ

- สังเกต ถ้าหากคุณกดปุ่มอื่นในขณะที่ computer อยู่ใน Mode บริการ มันจะไม่รู้จักว่าสมควรที่จะรับ Input ใด ๆ ถ้าคุณกดปุ่มใด ๆ ในขณะที่ compter อยู่ใน mode อื่น ปุ่มก็น่าจะให้บริการหน้าที่ที่ต่างกัน ปุ่มเหล่านี้จะยุ่งกว่าการทำงานปฏิบัติงานโต้ตอบในเทคนิควิธีของคู่มือ

Display

ปุ่มต่าง ๆ คือหนทางของคุณที่จะพูดคุยกับ computer จอภาพเป็นวิธีของ computer ที่จะใช้พูดคุยกับคุณ เช่น คุณเฉลียวฉลาดกว่าเกี่ยวกับการ prtgram คุณจะต้องอ่ายจอดภาพในการพิจารณาและแก้ไขโปรแกรมใน Bobot. พิจารณาคำแนะนำจากตัวตรวจจับ และตรงกันกับสิ่งที่คุณเป็นอยู่

Mode และการโปรแกรมสนทนาโต้ตอบในคู่มือ และส่วนที่เหมาะสมในคู่มือเทคนิค จะบอกคุณในสิ่งที่คาดหมายจากจอภาพและใช้คำแนะนำในจอภาพอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงาน

แต่ละครงที่เปลี่ยนความตุ้นเคยหรือประเภทของการทำงานเรียกว่า Mode ต่อไปคือสิ่งที่คุณต้องทำความคุ้นเคยกับคำแนะนำพื้นฐานเกี่ยวกับแต่ละ Mode แต่ละหัวข้อสำหรับโหมดต่าง ๆ จะแสดงในวงเล็บซึ่งก็คือปุ่มที่ต้องกด ถ้าคุณต้องการที่จะป้อน Mode นน

Utiltty mode (3)

Utility mode มีประโยชน์เหมือนกับอันอื่น ๆ มันทำให้เกิดการบริการที่ว่าคุณต้องการหรือจำเป็นที่จะใช้งานประจำตลอดเวลาทุควัน คุณก็ต้องป้อน Utility mode โดยการกด "3" การกด Initialiaing (31)

ในนนเร่มต้นจะเป็นประโยชน์ที่สำคัญ มันเป็นที่ส่งไปขับเคลื่อนล้อในแนวเส้นตรง และเท้า แขน ข้อมือ ฯลฯ ในสิ่งเหล่านี้ คือตำแหน่งเดิมของมัน ดังนBน computer จะรู้ตำแหน่งของพวกมัน บอกจากคุณมีคู่มือเคลื่อนไหวหัวหรือขับล้อหลังจาก Robrt ของคุณหยุดหมุน คุฯไม่จำเป็นต้องเริ่มใหม่ในแต่ละครBงที่คุณเปิดมัน อย่างไรก็ตามคุณควรกลับไปตBงต้นใหม่หลังจากที่ไม่ได้ใช้เป็นเวลานาน ๆ เพื่อให้แน่ใจว่าอยู่ในตำแหน่งเดิมของมัน คือในหน่วยความจำของ Robot การเริ่มต้นของ Robot จาก Ulility mode โดยการกด "1" (กด "31" เมื่อ start จาก Execlutive Mode)

เมื่อคุณเริ่มต้น หุ่นยนต์ สิ่งต่อไปนี้จะเกิดขึ้น

1.แขนจะไปทางที่จำกัดเต็มที่

2.หัวไหล่จะลงจำกัดที่ตำแหน่งเต็มที่

3.มือจะมหุนทวนเข็มและตามเข็มนาฬิกาจนสุด

4.ข้อมือจะหมุนขึ้น (ทางซ้ายของ Robot)

5.ตัวหนีบอยู่ในตำแหน่งที่ติดกัน

6.หัวจะหมุนทวนและตามเข็มนาฬิกาจนสุดของมัน

7.ล้อหน้าจะหันไปทางซ้ายสุดของมัน

8.มือจะหมุนกลับไปตำแหน่งตรงกลางของมัน

9.หัวจะหมุนกลับตำแหน่งกลาง

10.ล้อหน้าจะหันกลับมาตำแหน่งตรงกลางของมัน

11.computer จะกลับไปที่ Executive Mode

ถ้าหากชนิดส่วนกลไกเกิดมีแรงเคลื่อนที่ขึ้นมาครBงหนึ่งจากตำแหน่งเริ่มต้น Robot จะไม่รู้จักสิ่งนี้ และจะไม่แก้ไขตัวมัน สำหรับตัวอย่าง ล้อหน้าอาจจะผลักให้เลื่อนไปในขณะที่กำลังทำงานอยู่ ดังนBน Bobot จะไปทางเป็นวงกลมอย่างแน่นอน จนกระทั่งคุณเริ่มต้นมันใหม่ ดังนBนตรงข้างหน้างจุดมุ่งหมายจะต้องทำใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ARM HOMING (32)

ครั้งที่ 2 คำสั่ง homing คำสั่ง-ภายนอกใน Utilily mode ถ้า program มีแบบซ้าย หัว หรือ มุ่งวิถีในตำแหน่งอื่นใด มากกว่าตำแหน่งปกติ จะต้อง set เริ่มต้น Utility ประโยชน์ของ Homing จะมุนแขนกลับ หัว และมุ่งวิถีไปที่ตำแหน่งเดิมเริ่มแรก

ป้อน Homing Utility โดยการกด A (Utiliting) และ A Homing

EXECUTIVE MODE (RESET)

การเริ่มต้นจำเป็นต้องทำให้ชิ้นส่วนกลไกต่าง ๆ ของ Robot อยู่ในตำแหน่งเดิมเริ่มต้น Executive mode จะทำให้เกิดตำแหน่งเดิม ซึ่งเป็นกรรมวิธี Ingie exeeutive mode คือ จุดที่ต่อจากจุดซึ่งคุณป้อนโหมดอื่น ๆ และจุดที่คุณกลับไปเมื่อทำการออกจากโหมดอื่น

บ่อยครั้งที่ Robot ป้อน executive mode ด้วยตัวของมันเอง มันจะทำสิ่งนี้เมื่อคุณเปิดหรือเมื่อมันเสร็จสิ้นการทำงานประโยชน์ที่ต้องทำแล้ว บางครั้งคุณอาจต้องการที่จะเปลี่ยน mode ด้วยตัวคุณเอง คุณต้องเริ่มสิ่งนี้เป็นลำดับแรกในการที่จะกลับไป executive mode

ในการป้อน executive mode กด Rdset จะมีเสียงพูดว่า "neady" และจอภาพจะแสดงเป็นเวลา 10 วินาที และมันก็จะเปลี่ยนเป็นการเคลื่อนที่เล็กน้อย จะทำให้ Battery หมดเปลืองเล็กน้อย คุณสามารถที่จะป้อน mode ใด ๆ จาก executive mode โดยการกดปุ่มแก้ไขให้ถูกต้อง

ปุ่ม 1 เลือก Program mode

ปุ่ม 2 เลือก Utility mode

ปุ่ม 4 เลือก mamual mode

ปุ่ม 7 เลือก Learn mode

ปุ่ม A เลือก Repeat mode

ปุ่ม Reset กลับไปเริ่มการทำงานใหม่ ปุ่ม O ,C และ Fเป็นจุดมุ่งหมายพิเศษ ว่าปุ่มเดียวเข้าไปหาหน่วยความจำที่ตำแหน่ง 0030 , 0033 , 0036 โดยลำดับ

ปุ่ม D จะทำการ load program ใน ROM เพิ่มเติม ในกรณีที่ว่าปุ่มอื่นมีการแตกต่าง ของผลมันก็อธิบาย ดังนั้นจะต้องทำการเพิ่มเติม ROM ในคู่มือของคุณ

การกดปุ่มใด ๆ ในขณะที่กำลังทำงานใน Mode จะมีผลเหมือนกับการกดปุ่ม reset เลยทีเดียว

MAMUAL MODE (4)

Robot จะทำการสอบการใช้รีโมดคอนโทรลด้วยมือ ตรวจพิจารณาชื่อ และคุณจะเห็นว่า สี รหัส และสัญลักษณ์ ถูกทำขึ้นอย่างง่าย ๆ ที่เก็บตำแหน่ง สวิทซ์ทางขวา และรับมอเตอร์ในทิศทางขวา

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อคุณเลือก manual mode robot ก็จะรับคำสั่งจากการสกนของรีโมท คุณสามารถที่จะขับเคลื่อนหุ่นยนต์และควบคุมแขนและการเคลื่อนที่ของหัว สังเกตุว่าคุณสามารถทำเฉพาะเฉพาะการเคลื่อนไหวเดียวด้วยตัวควบคุมวันนี้ คุณไม่สามารถที่จะแทนที่การขับเคลื่อน Robot และหมุนห้าพรอ้มกันได้ อย่างไรก็ตามคุณสามารถที่จะขับเคลื่อน Robot และทำให้มันมุ่งไปในทางที่ถูกต้องได้

ป้อน manual mode โดยการกด key 4 ต่อข้อต่อของตัวบังคับที่ข้องต่ออันล่าง บนข้างหลังของ Robot

สิ่งนี้จะควบคุมการทำงานและอธิบายได้โดยส่วนต่อไปนี้

TRIGGER SWITCH

การทำงานจะแสดงให้เห็น เว้นแต่คุณดึง Trgger การเลือกควบคุมอื่น ซึ่งมอเตอร์ทำงานหรือซึ่งทิศทางของ Moter ทำงาน แต่แท้จริงแล้วสิ่งนี้เปิด Trigger

FUNCTIAN SWITCH

ใช้ SW นี้เลือกว่าคุณจะควบคุมแบบ (และหัว) หรือลำตัว (ขับเคลื่อน Robot) ถ้าคุณเลือกตำแหน่งแขน Retary switch จะเจาะจงการทำงานของแขนและการทำงานของหัว ถ้าคุณเลือกตำแหน่งลำตัว Retany switch จะเจาะจงการทำงานขับเคลื่อน Robot (ไปข้างหน้า 3 speet ไปข้างหลัง 3 speed และเป็นกลาง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROTARY SWITCH

ใช้สวิทซ์นี้เพื่อเลือก motor แขนหรือ motor หัวที่คุณต้องการให้มันทำงาน หรือเพิ่มหรือลดความเร็วตามที่คุณต้องการจะเคลื่อน robot ระมัดระวังเสมอในการสังเกตุว่า function sw ที่คุณทำการเลือกหรือ rotary sw นั้นคุณอาจจะคิดว่าคุณทำการเลือกควบคุมให้มือหนีบ แต่แท้จริงแล้วคุณอาจจะเลือกให้ robot เคลื่อนไปข้างหน้าแทน

MOTION SWITCH

คุณจำเป็นต้องเลือก motion switch นี้เพื่อเลือกว่าจะให้ motor ส่วนแขนหรือ motor ส่วนหัวทำงาน หรือคุณต้องการหมุนล้อไปข้างหน้า switch ตัวนี้จะควบคุมหน้าที่เหล่านี้ ควบคุมเพื่อที่จะช่วยให้ท่าทางการเคลื่อนที่ และใช้ rotary switch ร่วมกัน สังเกตุว่าคุณสามารถใช้มันเพื่อควบคุมการเคลื่อนที่ไปในวิถีทางใดๆ ได้ถ้าหาก robot ไม่เคลื่อนที่ ส่วนถ้า rotary sw อยู่ที่ตำแหน่ง "n" sw การเคลื่อนที่จะทำให้ล้อหน้านิ่ง นานพอกับที่คุณดึง trigger sw สิ่งนี้สามารถช่วยได้อย่างรวดเร็ว เมื่อ robot อยู่ในที่แคบ คุณสามารถหมุนล้อหน้าเป็นอันดับแรก เมื่อขับเคลื่อนทำให้ robot หมุนภายในเส้นผ่าศูนย์กลางของตัวมันเอง ในทำนองเดียวกันให้จดจำขณะที่ทำการเคลื่อนที่ว่าคุณจะเคลื่อนไปข้างหลังหรือข้างหน้าไม่ได้ แก้ไขได้ง่ายๆ โดย switch การเคลื่อนที่ของล้อหน้าให้กลับไปอยู่ในตำแหน่งข้างหน้า นานเท่ากับ trigger ถูกดึง

ตัวอย่างการทำงานของ manual (manual operation)

ตอนนี้คุณทราบแล้วว่าจะใช้ที่บังคับไปบังคับ robot เลือกพื้นที่อย่างน้อย 8 ฟุต และกว้าง 8 ฟุต ในการเคลื่อนที่ของ robot การหลบเครื่องกีดขวาง เช่น ขั้นบันได และคมของมุมกำแพง สุดท้ายเลือกพื้นผิวที่ราบเรียบสำหรับให้ robot เคลื่อนไป พรมน้ำมัน หรือลาดพื้นให้ดีที่สุดอย่าให้วิ่งบนพรมหนาๆ เพราะมันจะเป็นภาระที่หนักหน่วงมากสำหรับ robot ที่จะเคลื่อนไป และพวกเส้นใยไหมสามารถจับบนล้อ robot ได้

1. เปิด robot และเสียบปลั๊กที่บังคับ

2. ต้องแน่ใจว่า robot เริ่มเตรียมพร้อมแล้ว กดปุ่ม 4 บนแป้นกด เพื่อป้อนเป็น manual mode

3. set rotary switch ไปที่ตำแหน่ง "N" และ Function switch ไปที่ตำแหน่ง BODY

4. อย่าดึง trigger ในขณะที่คุณ set rotary switch ไปที่ตำแหน่งมือและข้อมือ

5. ดึง trigger และ robot จะเคลื่อนไปข้างหน้าอย่างช้าๆ ใช้ motion switch ไปทางซ้ายหรือขวาจับ switch ลงและ robot จะหันปล่อยมันและ robot ก็จะเริ่มตรงออกไป เพื่อไขปล่อย trigger เพื่อหยุด ทำก่หมุน rotary switch ตามเข็มนาฬิกาให้ไกลจะเลือกความเร็วไปข้างหน้า 2 ระดับ ทำการหมุน rotary switch ทวนเข็มนาฬิกา เพื่อทำให้เป็นกลาง และในทางกลับกันก็เดินไปข้างหน้าอย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ด้วยการปล่อย trigger และเปลี่ยน Function switch ไปที่ตำแหน่งแขน ดึง rotary switch ไปที่ตำแหน่งหัว กด trigger และใช้ motion switch ในบางกรณี motor อาจจะจบการทำงานของมันอยู่แล้ว ถ้าเป็นอย่างนี้ ทำการดัน motion switch เพื่อจำทิศที่จะเดินทาง จะไม่ทำให้เดินไปข้างหน้า

LEARN MODE

Learn mode นั้นจะเป็นทำนองเดียวกับ manual mode ใน learn mode robot จะจดจำทุกสิ่งทุกอย่างที่คุณทำกับมัน มันสามารถทำซ้ำในสิ่งที่ผ่านไปแล้ว ใน repeat mode ในการรวมกันคุณสามารถพิจารณาได้ง่ายหรือเปลี่ยนความยาวในการเคลื่อนไหวว่าจะให้ไปไกลแค่ไหน และการแสดงออกอะไรบางอย่างด้วยความรวดเร็วเมื่อกลับไปเล่นใน repeat mode โดยการใช้ learn mode คุณสามารถสอน robot ให้มันทำอะไรก็ได้ในสิ่งที่คุณต้องการให้มันทำอย่างเร็วกว่า และคุณสามารถให้มันทำด้วย manual mode สังเกตุว่าแขนและหัวสามารถแสดงออกได้มากกว่าการเคลื่อนที่ในแนววิถี

ทีนี้จะเป็นอย่างไรเมื่อใช้ learn mode

1. เริ่มต้นด้วยการเปิด robot โดยการเริ่มต้นใน executive mode และทำการเสียบปลั๊กของที่บังคับ

2. กดปุ่ม 7 เพื่อเริ่มต้น learn mode

3. จอภาพจะแสดง "7" แค่ 4 ครั้ง ป้อนตำแหน่ง address ที่ซึ่งคุณต้องการให้ robot เริ่มต้นซึ่งเป็นที่เก็บโปรแกรมการเคลื่อนไหวไว้

4. จอภาพแสดง "7" แค่ 4 ครั้ง อีกที ตอนนี้ป้อนตำแหน่ง address สูงสุดที่คุณจะยอมให้เพื่อโปรแกรมจะหยุดการ RUN โปรแกรมอื่นๆ ของคุณที่มีอยู่ใน memory ถ้าคุณใช้หน่วยความจำทั้งหมดที่คุณมี จอภาพจะแสดง "FULL" และ robot จะปฏิเสธคำสั่ง คุณต้องออกจาก learn mode โดยการกด reset ชนิดโปรแกรมสั้นของการเคลื่อนไหวสามารถที่จะเก็บไว้มรที่ว่างจาก 0100 ถึง 0200 ได้

หลังจากที่คุณป้อนตำแหน่ง address ครั้งที่ 2 จอภาพจะแสดง 7,F,XXXX, ที่ซึ่ง XXXX เท่ากับตำแหน่ง address บวก 8

นี่จะเป็นคำสั่งแรกของโปรแกรม จะใส่เข้าไปช่วยอัตโนมัติ และบอก robot ให้กลับไปตำแหน่งเริ่มต้นเดิมของมัน เมื่อโปรแกรมสั่งให้ย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ตอนนี้คุณพร้อมที่จะให้เริ่มต้นการทำงาน robot แล้ว ใช้เครื่องบังคับให้แสดงท่าทางต่างๆทั้งหมดที่คุณต้องการให้โปรแกรมลงไปในหน่วยความจำ เมื่อคุณทำเสร็จแล้ว ให้กดปุ่ม reset สิ่งนี้จะทำให้คุณกลับไปที่ executive mode

6. เล่นโปรแกรมป้อนกลับที่ใส่ลงไปในหน่วยความจำ กด A, D และ Address ในหน่วยความจำ ถ้าโปรแกรมของคุณมี robot จะเกิดการวิ่งเคลื่อนที่ ทำให้แน่ใจโดยการเริ่มจากแหน่งที่เหมือนกันเมื่อคุณเริ่มป้อน

โปรแกรม หรือ robot ของคุณอาจจบการเดินไปที่ตำแหน่งต่างๆ

Saving memory in Learn mode

แต่ละครั้งที่คุณดึง trigger บนที่บังคับการเคลื่อนไหว (การบอกว่าเลือก motor ตัวนี้และให้มันทำงานไกลแค่ไหนอย่างไร) มันจะเก็บไว้ในหน่วยความจำ หน่วยความจำถูกเก็บโดยการ RUN แต่ละการเคลื่อนไหวทั้งหมดของท่าทางที่ปราศจากการแก้ไขหรือ trigger ถ้าไปไกลคุณสามารถใช้การสลับสัญญาณที่สำคัญที่ถูกต้องในการเคลื่อนไหว ถ้าไปไกลไม่เพียงพอ ทำการดึง trigger 2 ครั้ง สำหรับ memory ว่าเป็น 1 การทำงาน (เพราะว่า robot จะเก็บคำแนะนำทุกครั้งที่แก้ไข trigger)

Back up Function

Robot มีความสามารถที่จะย้อนกลับในแขนของมันและการเคลื่อนไหวของหัวและหน่วยความจำของมันเอง ถ้าคุณป้อนการเคลื่อนไหวผิด คุณสามารถที่จะย้อนไปทำ step ที่ถูกต้องในโปรแกรม ลักษณะสำคัญนี้จะหาดูได้ง่ายมาก ถ้าคุณมีสิ่งที่กระทำได้ยาก การเคลื่อนไหวที่ยุ่ง และต้องการที่จะลดทอน robot ของคุณโดยการย้อนออกมาโดยปราศจากการเปลี่ยนโปรแกรม

19เริ่มต้นการย้อนกลับของแขนและหัวโดยกดปุ่ม "B" บนแป้นกด เมื่อคุณกด trigger robot จะเริ่มย้อนตลอดโปรแกรม คุณสามารถทำขึ้นได้ คุณสามารถย้อนตลอดเป็น step หรือตลอดทางที่จุดสิ้นสุดของแต่ละ step robot จะหยุดและ "B" จะสว่างที่จอภาพ ให้คุณแก้ไขที่ trigger robot จะกลับตลอดใน step ,ต่อไปและต่อไปตลอดทาง ทั้งหมดที่ทำการเริ่มต้นที่คุณต้องการ เช่นคุณต้องการย้อนกลับตลอดคุณจะเห็นหน่วยความจำที่จอภาพและตำแหน่ง motor และจอภาพจะนับถอยหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณอาจจะหยุดการป้อนกลับเมื่อใดก็ได้ที่คุณต้องการตลอด step หรือระหว่าง step จะไม่มีอะไรเปลี่ยนแปลงในหน่วยความจำจนกระทั่งคุณกด "F" (END) และป้อน step ใหม่ สิ่งนี้จะเปลี่ยนอย่างถาวร ที่โปรแกรมที่สาขาใหม่ที่คุณคิดขึ้น

ถ้าคุณจะใช้หน้าที่ย้อนกลับแนวการเคลื่อนที่ของคุณ และไม่ต้องการที่จะเปลี่ยนอะไรเลย คุณสามารถทำได้โดยไม่สูญเสียโปรแกรมเดิม ใช้ reset เพื่อออกจากโปรแกรมปราศจากการกดปุ่ม "F" ถ้าคุณย้อนกลับตลอด โปรแกรม และตัดสินว่าคุณต้องการต่อเชื่อมกับโปรแกรมซึ่ง อยู่ตอนนี้ คุณอาจจะต้องกด "F" และป้อนการเคลื่อนไหวบางอย่างที่ง่าย และจบตรงจุดนั้น

The Reverse The Instruction

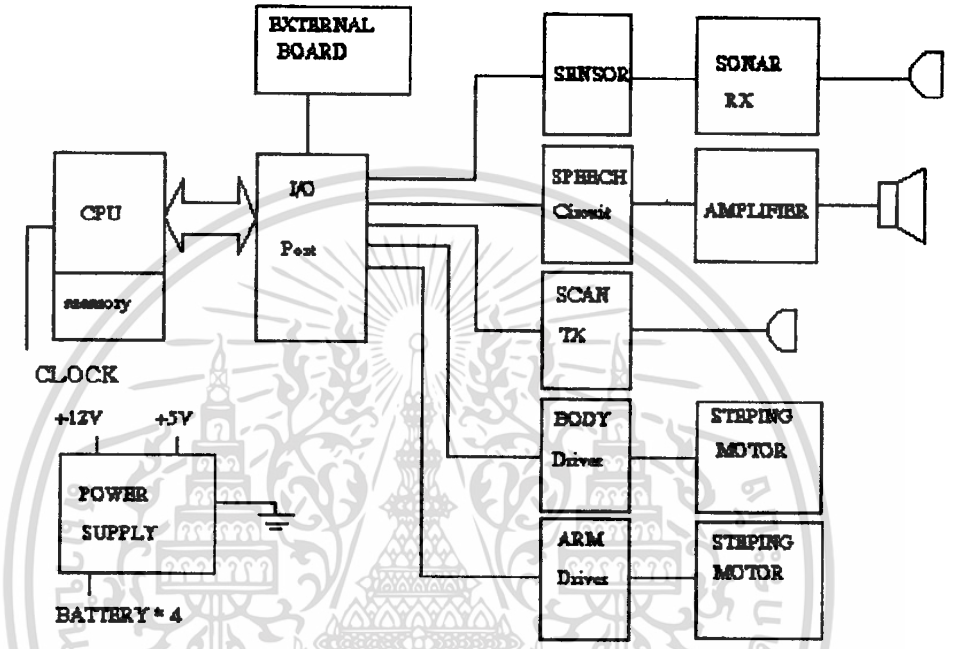
ถ้าขณะกำลังทำงานที่คุณต้องทำ robot ทำงานบางสิ่งบางอย่างอยู่ ย้อนกลับและทำในสิ่งที่เหมือนกัน ลักษณะสำคัญนี้สามารถที่จะใส่คำแนะนำการย้อนกลับที่ถูกต้องในโปรแกรมของคุณได้ (ลักษณะที่ย้อนกลับนี้ประยุกต์ใช้ได้เฉพาะแขนและหัวเท่านั้น)

เมื่อคุณเสร็จสิ้นการทำให้เดินหน้าของ step และปรารถนาที่จะให้คำแนะนำย้อนกลับกด RTI (ปุ่ม #7) "r" เพื่อย้อนกลับจะแสดงขึ้นที่จอภาพ เมื่อคุณดึง trigger บนเครื่องควบคุมแขนหรือหัวจะเริ่มกลับไปที่คำแนะนำสุดท้าย การทำงานจะดำเนินไปอย่างเหมาะสมเช่นเดียวกับการย้อนกลับ 1 step จะมี "r" สว่างขึ้น และหยุดระหว่าง step อย่างไรก็ตาม แต่ละ step เหล่านี้จะอยู่ในหน่วยความจำ เมื่อคุณต้องการที่จะหยุดการย้อนกลับของ memory กด "F" สำหรับไปข้างหน้าและป้อน step อื่นเข้าไป หรือง่ายกว่านั้นโดยการกดปุ่ม reset โปรแกรมจะทำ step ที่คุณป้อน ดังนั้นการย้อนกลับคุณต้องเลือกใช้ RTI และต่อเนื่องไปจนจบ

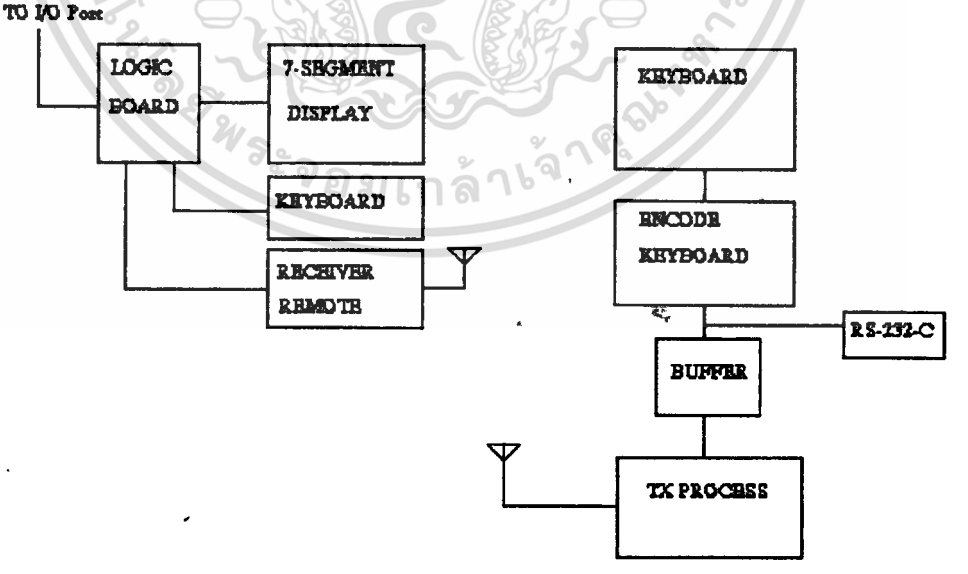
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ระบบการทำงานของหุ่นยนต์ ET-18



Block Diagram ของหุ่นยนต์ ET-18



Blockdiagram แสดงภาครับส่งสัญญาณของหุ่นยนต์ ET-18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. CPU คือหน่วยประมวลผลกลางข้อมูลของระบบและทำงานร่วมกับหน่วยอื่นอีก 2 หน่วยคือ หน่วยความจำ(memory),หน่วยรับส่งสัญญาณเข้าออก (input/output port)

2. หน่วยความจำ (MEMORY) เป็นส่วนเก็บคำสั่งและข้อมูลต่างๆ

3. หน่วยรับส่งสัญญาณเข้าออก (INPUT/OUTPUT PORT) ได้แก่ส่วนแสดงผลหรือใส่ข้อมูลที่เราใช้งาน บริการและจัดการกับอุปกรณ์รอบข้างที่เข้าออก เป็นจุดรวมของข้อมูลทั้งหมดก่อนส่งให้ CPU

4. EXTERNAL BOARD คือแผงโปรโตบอร์ดที่ใช้สำหรับต่อวงจรอิสระเพื่อใช้กับการอินเตอร์เฟส

5. LOGIC BOARD รับข้อมูลจาก keyboard และรับข้อมูลจากภาค receiver remote control และแสดงผลออกที่ LED 7-Segment

6. SONAR & SENSOR คืออุปกรณ์ตรวจจับสิ่งกีดขวางที่อยู่ข้างหน้าของหุ่นยนต์ทำให้หุ่นยนต์สามารถหลบหลีกสิ่งกีดขวางได้โดยอัตโนมัติ

7. SPEECH & AMPLIFIER วงจรสร้างเสียงและทำการขยายสัญญาณให้แรงขึ้นเพื่อขับออกทางลำโพง

8. SCAN ส่งสัญญาณออกไปข้างหน้าเมื่อพบสิ่งกีดขวางสัญญาณจะสะท้อนกลับมาเข้าที่เครื่องรับ

9. BODY DRIVER ส่วนควบคุมการขับเคลื่อนของหัวและล้อ ซึ่งจะส่งสัญญาณควบคุม stepping motor อีก 3 ตัว

10. ARM DRIVER ส่วนควบคุมการเคลื่อนที่ของแขน โดยจะส่งสัญญาณควบคุม stepping motor อีก 5 ตัว

11. CLOCK คือสัญญาณนาฬิกาเพื่อใช้เป็นฐานเวลาในการทำงานของระบบทั้งหมด

12. POWER SUPPLY แหล่งกำเนิดพลังงานของหุ่นยนต์ ใช้แบตเตอร์รี่แบบชาร์ตไฟได้

13. Keyboard Scan แป็นตัวอักษรต่างๆ และตัวเลข เพื่อใช้ป้อนข้อมูลให้หุ่นยนต์ทำงานตามต้องการ

14. Encode Keyboard คือวงจรถอดรหัสสัญญาณจาก keyboard

15. PORT PS-232-C เป็นคอนเน็คเตอร์ชนิดหนึ่งมีไว้สำหรับการอินเตอร์ เฟสกับภายนอก

16. TX PROCESS คือภาคส่งสัญญาณออกจาก REMOTE CONTROL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ทฤษฎีและการใช้งาน RS-232-C

ก่อนอื่นเราจะกล่าวถึงปัญหาที่เกิดขึ้นในการส่งข้อมูลระหว่างอุปกรณ์อิเล็กทรอนิกส์ การส่งข้อมูลขึ้นได้นั้นจะต้องมีแหล่งกำเนิดสัญญาณ(source) ,แหล่ง ,แหล่งรับสัญญาณ(distination) และการเชื่อมต่อระหว่างแหล่งสัญญาณสองแหล่งนี้ ในการส่งข้อมูลระหว่างชิฟไอซีสองตัวที่อยู่บนแผ่นวงจรพิมพ์ (printed circuit card) แผ่นเดียวกัน เช่น ในการอ่านข้อมูลจากหน่วยความจำซึ่งเป็นขบวนการที่เกิดขึ้นระหว่าง CPU และหน่วยความจำบางตัวบนแผ่นวงจรพิมพ์ ข้อมูลที่ส่งไปจะเดินทางไปตามลายวงจรที่อยู่บนแผ่นพลาสติก ซึ่งในระหว่างขบวนการนี้จะมีปัญหาบางอย่างเกิดขึ้นดังจะอธิบายต่อไป

โดยทั่วๆ ไปข้อมูลจะถูกแทนด้วยเลขฐานสอง ระดับแรงดัน +5 V จะแทนด้วยลอจิก 1 และระดับแรงดัน 0 V จะแทนลอจิก 0 การแทนลอจิกด้วยระดับแรงดันเหล่านี้เรียกว่าการแทนระดับสัญญาณของอุปกรณ์ TTL (Transistor-Transistor Logic) ซึ่งการแทนระดับสัญญาณแบบนี้ ถูกใช้กันทั่วๆไปในการส่งข้อมูลระหว่างอุปกรณ์ชิ้นหนึ่งไปยังอีกชิ้นหนึ่งภายในเครื่องคอมพิเตอร์ แต่มันเป็นการไม่เหมาะสมที่จะกำหนดระดับแรงดันในการแทนลอจิก 0 หรือ 1 ที่จำเพาะเจาะจงลงไปเป็นค่าเฉพาะเพียงค่าเดียว ดังนั้นระดับแรงดันที่ใช้แทนลอจิก 0 และ 1 นั้น จึงถูกกำหนดเป็นพิสัย (range) พิสัยนี้แสดงไว้ดังรูปที่ 1 จากรูปจะเห็นได้ว่าพิสัยของแรงดันที่ส่งจากอุปกรณ์ที่เป็นตัวส่งจะแตกต่างจากพิสัยของแรงดันที่ถูกปรับโดยอุปกรณ์ที่เป็นตัวรับ (พิสัยที่แทนลอจิก 0 และ 1) นั่นก็คือ ตัวส่ง {sender หรือ transmitter แต่ในบางครั้งก็เรียกตัวสร้างสัญญาณ (generator) หรือตัวรับสัญญาณ (driver)} จะต้องจ่ายแรงดันที่มีระดับสัญญาณต่ำที่สุดเท่ากับ 2.4 V ในการส่งลอจิก 1 แต่ในภาครับสัญญาณ ตัวรับ (receiver หรือ transmittor) จะถือว่าระดับแรงดันที่มีระดับสัญญาณอยู่ระหว่าง 2.4-2.0 V เป็นลอจิกหนึ่งด้วย สำหรับสาเหตุที่ถือว่าระดับสัญญาณในช่วง 2.4-2.0 V เป็นลอจิก 1 ก็เนื่องจากมีการสูญเสีย (loss) ของสัญญาณระหว่างตัวส่งและตัวรับขึ้น ความคลาดเคลื่อนของระดับแรงดันที่เกิดขึ้นนี้เรียกว่า "noise margin" ซึ่งมีค่าเท่ากับ 0.4 V ไม่ว่าจะเป็นกรณีของลอจิก 0 หรือ 1 สำหรับลอจิก 0 ความคลาดเคลื่อนเกิดขึ้นจากสัญญาณรบกวน (noise) ที่ปนเข้ามา ในทางปฏิบัติความคลาดเคลื่อนนี้ถูกยอมรับให้ใช้ได้ในการใช้งานทั่วๆ ไป ซึ่งจะเห็นได้ในข้อกำหนดของคุณสมบัติทางไฟฟ้าของการอินเตอร์เฟสแบบมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

จากรูปที่ 1 จะเห็นว่ามีระดับแรงดันอยู่ช่วงหนึ่ง เรียกว่าช่วงการเปลี่ยนสถานะ สถานะซึ่งไม่อาจระบุได้ว่าสัญญาณในช่วงนั้นเป็นลอจิก 0 หรือ 1 ช่วงของสัญญาณนี้ทางด้านรับจะมีช่วงแคบกว่าทางด้านส่ง ดังนั้นระดับแรงดันทางด้านรับที่ใช้แทนลอจิก 0 หรือ 1 จะมีช่วงกว้างๆ กว่าทางด้านส่ง

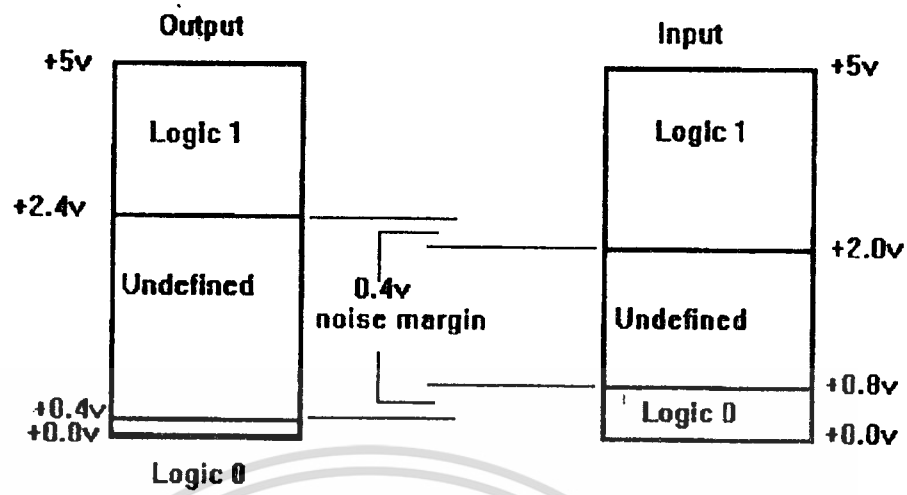
ในการส่งข้อมูลภายในเครื่องคอมพิวเตอร์ เราจะถือว่าระดับสัญญาณที่ส่งและรับ(ซึ่งใช้การแทนระดับสัญญาณแบบ TTL) เป็นแบบอุดมคติ เนื่องจากเหตุผลต่อไปนี้

1. กำลังงานที่ใช้ และการกระจายความร้อนมีค่าต่ำ

2. สัญญาณที่ใช้เป็นระดับสัญญาณลอจิกแบบ TTL ซึ่งสามารถจ่ายให้แก่ชิฟไอซีได้โดยตรง โดยไม่ต้องใช้ line driver และวงจรรับข้อมูลที่มีราคาแพง

3. การอินเตอร์เฟสระหว่างอุปกรณ์ TTL จะทำงานที่ความถี่สูง ซึ่งจำเป็นต้องใช้ในการส่งข้อมูลภายในคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 1 คุณสมบัติทางไฟฟ้าของการอินเตอร์เฟสที่ใช้การแทนระดับสัญญาณแบบ TTL

ต่อไปจะกล่าวถึงปัญหาที่เกิดขึ้นในการสื่อสารข้อมูลระหว่างอุปกรณ์แต่ละชิ้นที่ไม่ได้อยู่ในเครื่องๆ เดียวกัน (ไม่ได้อยู่บนแผ่นวงจรพิมพ์เดียวกัน) เช่น การติดต่อระหว่างเครื่องคอมพิวเตอร์กับเทอร์มินัล หรือการติดต่อระหว่างเครื่องคอมพิวเตอร์สองเครื่องในสำนักงานเดียวกัน เทคนิคที่ใช้ในการสื่อสารข้อมูลภายในเครื่องนั้นไม่เพียงพอที่จะนำไปประยุกต์ใช้กับการติดต่อระหว่างเครื่องหรืออุปกรณ์แต่ละชิ้นได้ เราต้องเพิ่มเทคนิคบางอย่างเข้าไปอีกเพื่อให้การติดต่อสื่อสารข้อมูลระหว่างเครื่องเป็นไปได้อย่างถูกต้อง ในปัจจุบันมีเทอร์มินัลบางตัวที่ใชระดับสัญญาณแบบ TTL กับคอมพิวเตอร์หลัก ซึ่งถ้าเราทำการสื่อสารข้อมูลด้วยระดับสัญญาณนี้ในระยะมากกว่า 2-3 ฟุต อาจจะมีปัญหาหรือข้อยุ่งยากเกิดขึ้นบางข้อเนื่องจาก

1. ระดับสัญญาณ TTL มักถูกเหนี่ยวนำจากสัญญาณรบกวนจากภายนอกได้ง่าย
2. การสูญเสีย (loss) ไปในสายนทำให้ระดับแรงดันของสัญญาณที่ส่งออกไปลดลงซึ่งมีผลกระทบที่สำคัญต่อระดับแรงดัน 0-5 V ของ TTL เพราะว่าการสูญเสียระดับแรงดันไปเพียง 2-3 V สามารถทำให้ลอจิกต่างๆที่ได้รับผิดพลาดไป

การสื่อสารข้อมูลโดยใช้ระดับสัญญาณ TTL นี้ มักใช้ในการสื่อสารข้อมูลแบบขนาน ซึ่งจะทำได้ยากถ้าระยะทางที่ใช้ในการส่งข้อมูลยาวเป็นสิบๆ ฟุต ดังเหตุผลต่อไปนี้ ความเชื่อถือได้และราคา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ความเชื่อถือได้ก็คือ สัญญาณจะต้องไม่ถูกหน่วงให้ช้าลง (delay) ซึ่งจะทำให้เฟสเปลี่ยนแปลงไป รวมทั้งปัญหาเกี่ยวกับ skewing

ราคาของสายเคเบิลและคอนเนคเตอร์ (connector) ที่ใช้ในการส่งข้อมูลแบบขนานนั้นมีราคาแพงกว่าแบบอนุกรม

ในต้นปี 1967 John Van Geen วิศวกรของ Standford Research Institute ได้ประสบความสำเร็จในการสร้างโมเด็มที่สามารถเคลื่อนย้ายได้ง่าย (portable) เป็นคนแรก และได้ผลิตออกจำหน่ายโดยบริษัท Anderson-Jacobson Corporation ในเดือนสิงหาคม 1967 ความสำเร็จของอุปกรณ์ชิ้นนี้คือ สามารถเคลื่อนย้ายได้สะดวก และค่าใช้จ่ายในการเชื่อมต่อกับเครือข่ายโทรศัพท์ที่ใช้กันทั่วโลก เพื่อทำการสื่อสารข้อมูลมีค่าต่ำ นอกจากนี้การแปลงสัญญาณไบนารี่ (0,1) ไปเป็นสัญญาณที่สามารถส่งไปตามเครือข่ายได้นั้น สามารถทำได้โดยใช้ acoustic modem แต่ปัญหาที่ตามมาก็คือ จะต้องมีการกำหนดมาตรฐานของการอินเตอร์เฟสระหว่างอุปกรณ์ดิจิตอลที่ใช้ในการสื่อสารข้อมูลกับโมเด็มเสียก่อน ในปี 1969 Electronics Industry Association (EIA) ได้กำหนดมาตรฐานขึ้น มาตรฐานนี้มีชื่อว่า RS-232-C (แม้ในปัจจุบันมาตรฐานนี้ยังถูกใช้ในระบบไมโครคอมพิวเตอร์ทั่วๆ ไป รวมทั้งระบบที่มีขนาดใหญ่กว่าด้วย)

Physical-Layer Communication Protocal

Physical-layer communications protocal คือกลุ่มของข้อกำหนดที่เกี่ยวข้องกับการเชื่อมต่อทางกล (mechanical connections) (เช่นขนาดของคอนเน็คเตอร์), คุณสมบัติทางสัญญาณไฟฟ้าและคุณสมบัติที่เกี่ยวกับหน้าที่ของสัญญาณเหล่านี้ ซึ่งข้อกำหนดเหล่านี้ต้องเป็นไปตามมาตรฐาน สาเหตุที่ต้องทำข้อกำหนดนี้ขึ้นก็เพื่อให้อุปกรณ์ที่ผลิตจากบริษัทแต่ละแห่งสามารถใช้งานร่วมกันได้

ในการสื่อสารข้อมูล ข้อมูลจะถูกส่งไปมาระหว่างอุปกรณ์อิเล็คทรอนิกส์ผ่านทางการอินเตอร์เฟส ซึ่งประกอบไปด้วยสัญญาณไฟฟ้า, สายเคเบิลที่เป็นตัวนำสัญญาณไฟฟ้าและคอนเน็คเตอร์(ตัวต่อสายเคเบิลเข้ากับตัวอุปกรณ์) ข้อมูลที่ส่งจะถูกแทนด้วยการเปลี่ยนแปลงของกระแสและแรงดันการสื่อสารข้อมูลจะเป็นไปได้ก็ต่อเมื่อ อุปกรณ์ต่างๆ ที่ใช้เป็นไปตามข้อกำหนดใน Physical-layer communication protocal โปรโตคอลนี้ได้กำหนดข้อกำหนดต่างๆ ไว้อย่างชัดเจนเพื่อหลีกเลี่ยงปัญหาบางอย่างที่อาจเกิดขึ้นได้ข้อกำหนดต่างๆ มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คุณสมบัติทางกล : ได้มีการกำหนดรายละเอียดเกี่ยวกับขนาดของคอนเน็คเตอร์, จำนวนขา (pin), ฟังก์ชั่นการทำงานของขาต่างๆ, เส้นผ่าศูนย์กลางของขาและซ็อกเก็ต (ปลั๊กตัวเมีย), ตำแหน่งของคอนเน็คเตอร์ และคุณสมบัติของสายเคเบิล เช่นความยาวและจำนวนลวดตัวนำ

- คุณสมบัติของสัญญาณไฟฟ้า : มีการกำหนดคุณสมบัติทางไฟฟ้าของสัญญาณที่ใช้ควบคุมการแลกเปลี่ยนข้อมูลและวงจรไฟฟ้าที่เกี่ยวข้อง รวมทั้งการกำหนดอัตราความเร็วสูงสุดในการส่งข้อมูล, ระดับกระแสและระดับแรงดันที่ใช้แทนลอจิกต่างๆ (0 หรือ 1, ON/OFF, MARK/SPACE) และคุณสมบัติของวงจรรับและส่ง

- รายละเอียดตามหน้าที่ของสัญญาณ : สัญญาณต่างๆ ที่ใช้ในการอินเตอร์เฟสมักถูกกำหนดชื่อตามลักษณะการทำงาน โดยดูจากทิศทางการส่งข้อมูลว่าถูกส่งจากตัวส่งหรือตัวรับ และดูจากความสัมพันธ์ของสัญญาณนี้กับสัญญาณตัวอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232-C

ลักษณะทั่วๆ ไป

ถ้าเราอ่านแคตตาลอก(catalogue) ของเทอร์มินัลหรือเครื่องพิมพ์ เราจะเห็นข้อความ "RS-232-C compatible" ข้อความนี้หมายความว่า อุปกรณ์ตัวนั้นสามารถต่อกับคอมพิวเตอร์ได้โดยใช้คอนเน็คเตอร์แบบ 25 ขา คุณสมบัติของสัญญาณไฟฟ้าและคุณสมบัติทางกลของการอินเตอร์เฟสต้องเป็นไปตามมาตรฐาน RS-232-C ซึ่งถูกกำหนดโดย EIA แต่แม้ว่าคุณสมบัติต่างๆ จะเป็นไปตามมาตรฐาน RS-232-C เราก็ยังไม่อาจมั่นใจได้ว่า เราจะใช้งานมันได้อย่างเต็มประสิทธิภาพ ผู้ผลิตเทอร์มินัลหรือเครื่องพิมพ์มักจะทำตามข้อกำหนดเพียงบางข้อเท่านั้น คือเราอาจใช้เซอร์กิตต่างๆ ตามมาตรฐาน RS-232-C ได้เพียงบางเซอร์กิตเท่านั้น ดังนั้นเราต้องอ่านข้อกำหนดที่ผู้ผลิตได้จัดทำไว้อย่างละเอียด และตัดสินใจว่าเซอร์กิตที่ผู้ผลิตได้จัดทำไว้ให้เหมาะกับความต้องการของเราหรือไม่

ต่อไปจะกล่าวถึงมาตรฐานของ RS232-C อย่างย่อๆ โดยเลือกเฉพาะบางส่วนจากเอกสารที่ได้รับจาก Electronic Industries Association (EIA)

เนื่องจากความต้องการในการสื่อสารข้อมูลผ่านทางเครือข่ายโทรศัพท์มีมากขึ้นเรื่อยๆ ดังนั้นจึงต้องมีการกำหนดมาตรฐานที่เรียกว่า RS-232-C ขึ้น เพื่อใช้เป็นมาตรฐานจากอุปอุปกรณ์ที่ถูกผลิตจากบริษัทต่างๆ ในสหรัฐอเมริกา Bell System operating telephone companies เป็นบริษัทหลักบริษัทแรกที่เป็นผู้ผลิตและติดตั้งระบบสื่อสารข้อมูล และเป็นผู้ผลิตอุปกรณ์ต่างๆ ที่ใช้ในการอินเตอร์เฟสอุปกรณ์ดิจิตอลกับเครือข่ายโทรศัพท์รายใหญ่ อุปกรณ์นี้ก็คือ Bell modem ซึ่งถูกพัฒนาโดย Bell Laborotories และถูกใช้เป็นมาตรฐานในงานอุตสาหกรรมจนถึงปัจจุบันนี้ (บริษัทต่างๆ มักจะลอกข้อกำหนดต่างๆของ Bell ไปใช้งาน เพื่อให้สินค้าของบริษัทนั้นๆ

.สามารถใช้กับอุปกรณ์ของ Bell ได้) ขณะที่อุปกรณ์เกี่ยวกับคอมพิวเตอร์ต่างๆ ได้ถูกพัฒนาขึ้นเรื่อยๆ โดนบริษัทต่างๆ เทอร์มินัลและอุปกรณ์อื่นๆ มักถูกออกแบบให้สามารถอินเตอร์เฟสกับ Bell modem ได้ ดังนั้นความต้องการข้อมูลเกี่ยวกับข้อกำหนดในการอินเตอร์เฟสกับโมเด็มจึงมีเพิ่มขึ้นเรื่อยๆ เพื่อตอบสนองต่อความต้องการนี้ EIA, Bell System และผู้ผลิตโมเด็มรายอื่นๆ จึงร่วมมือกันตั้งมาตรฐาน RS-232-C ขึ้น

มาตรฐาน RS-232-C ได้ถูกพิมพ์โดย EIA ในปี ค.ศ. 1969 ตัวอักษร RS แทน "Recommended Standard", 232 แทนหมายเลขของมาตรฐานส่วนอักษร C แสดงให้รู้ว่าได้รับการแก้ไขกี่ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามมาตรฐาน RS-232-C ที่ถูกตีพิมพ์โดย EIA ได้กล่าวถึงการสื่อสารมูลระหว่าง Data Terminal Equipment (DTE) และ Data Communication Equipment (DCE) ในปัจจุบันตัวย่อ DCE จะแทน data circuit terminating equipment) คำจำกัดความของ DCE และ DTE ซึ่งแสดงไว้ข้างล่าง ได้คัดมาจากคำแปลศัพท์ (glossary) ในหนังสือ "Technical Aspect of Data communication" ซึ่งเขียนโดย John McNamara (Digital Press, 1977) ดังนี้

DCE :อุปกรณ์ที่มีฟังก์ชั่นการทำงานต่างๆ ที่ทำให้เกิดการเชื่อมต่อ, ทำให้การเชื่อมต่อยังดำเนินต่อไป และยุติการเชื่อมต่อ นอกจากนี้ยังใช้เปลี่ยนลักษณะของสัญญาณและสร้างรหัสสัญญาณต่างๆ ที่จำเป็นต้องใช้ในการสื่อสารข้อมูลระหว่าง DTE (Data Terminal Equipment) และ data circuit โดย DCE อาจเป็นส่วนใดส่วนหนึ่งของคอมพิวเตอร์หรือไม่ก็ได้

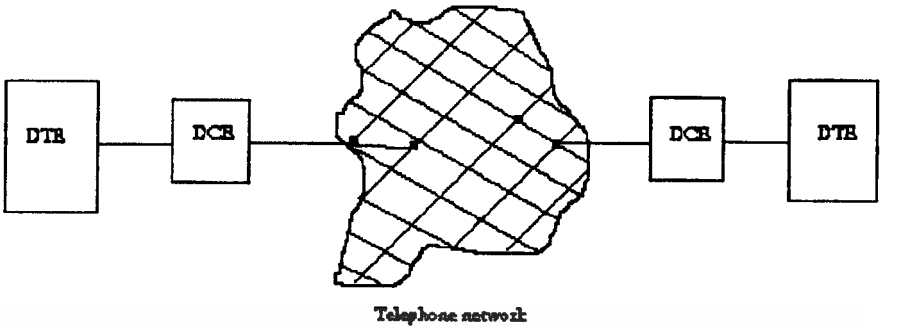
DTE :

1. เป็นอุปกรณ์ที่ประกอบไปด้วยตัวส่งข้อมูล (data source) หรือตัวรับข้อมูล (data sink) หรือเป็นทั้งตัวส่งและตัวรับข้อมูลก็ได้

2. เป็นอุปกรณ์ที่ประกอบด้วย function unit ต่อไปนี้ control logic, buffer store และอุปกรณ์อินพุทหรือเอาท์พุทจำนวนหนึ่งตัวหรือมากกว่าก็ได้ หรือรวมเครื่องคอมพิวเตอร์เข้าไปด้วยก็ได้ DTE อาจจะรวมส่วน error control, synchronization และความสามารถในการบ่งหรือระบุว่าต้องการเกี่ยวข้องกับอุปกรณ์ใด (station identification capability) เข้าไปด้วยก็ได้

ลักษณะของ DTE และ DCE ที่ใช้ในการสื่อสารข้อมูลได้แสดงไว้ในรูปที่ 2 ตามลักษณะการทำงานที่ได้อธิบายไว้ข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



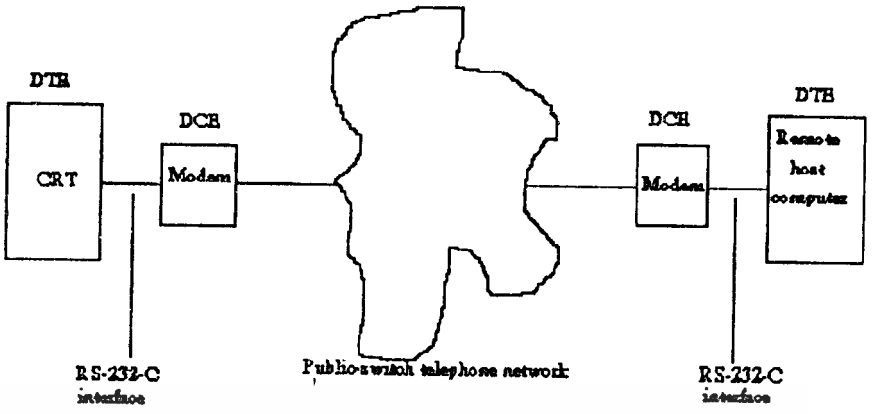
รูปที่ 2 ลักษณะของ DCE และ DTE ที่ใช้ในการสื่อสารข้อมูล

มูล

ในความเป็นจริงแล้ว DTE มักจะแทนแหล่งกำเนิดข้อมูลแหล่งแรก และ/หรืออุปกรณ์ที่เป็นแหล่งรับข้อมูลแหล่งสุดท้าย (ดังแสดงในรูปที่ 2) เช่นเครื่องพิมพ์ (เป็นอุปกรณ์ที่รับข้อมูลได้เพียงอย่างเดียว) จะเป็น DTE เพราะเป็นอุปกรณ์ที่รับข้อมูลเป็นตัวสุดท้าย หรือ CRT / คีย์บอร์ดเป็นทั้งตัวรับข้อมูลและตัวกำเนิดข้อมูลส่วน DCE เป็นอุปกรณ์ที่ทำให้การสื่อสารข้อมูลระหว่างแหล่งกำเนิดกับตัวรับข้อมูลที่ปลายทางทำได้สะดวกขึ้น ตัวอย่างหนึ่งของ DCE ก็คือ โมเด็ม เพื่อความเข้าใจเพิ่มขึ้นขอให้ดูตัวอย่างต่อไปนี้ประกอบ

ตัวอย่างที่ 1 จากรูปที่ 3 CRT ถูกต่อเข้ากับคอมพิวเตอร์ปลายทางหลัก (remote host computer) ซึ่งใช้เป็นระบบจัดสรรเวลา (time-sharing service) โดยการเชื่อมต่อผ่านสายโทรศัพท์ จากรูปที่ 3 นี้จะเห็นได้ว่ามี DCE และ DTE อยู่อย่างละ 2 ตัว DTE ตัวแรกเป็นเทอร์มินัล CRT, ส่วน DTE อีกตัวหนึ่งเป็นคอมพิวเตอร์ปลายทางหลัก เหตุที่ทั้ง 2 ตัวเป็น DTE ก็เพราะคอมพิวเตอร์ปลายทางหลักแนตัวส่งข้อมูล(แหล่งข้อมูล) แหล่งแรก(ต้นสุด) ส่วน CRT เป็นตัวรับข้อมูลตัวสุดท้าย(ปลายสุด) ส่วนโมเด็มทั้ง 2 ตัวที่ต่อที่ปลายสายโทรศัพท์ทั้ง 2 ด้านเป็น DCE เพราะโมเด็มทำหน้าที่เปลี่ยนลักษณะสัญญาณซึ่งทำให้สามารถทำการสื่อสารข้อมูลผ่านทางเครือข่ายโทรศัพท์ได้ ส่วนวงจรติดต่อ(circuit) ที่ใช้ในการสื่อสารข้อมูลระหว่าง DCE ทั้งสองดัง เราใช้การเชื่อมโยงทางโทรศัพท์(telephone link) ส่วนการเชื่อมโยงระหว่าง DCE และ DTE นั้นใช้ RS-232-C เป็นมาตรฐานหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 ลักษณะระบบที่ใช้ในตัวอย่างที่ 1

สำหรับเครือข่ายโทรศัพท์ที่ทำให้เกิดการเชื่อมต่อระหว่าง DCE ทั้งสองตัวนั้น เราจะไม่พูดถึง (เพราะว่ามันเป็นระบบมาตรฐานอยู่แล้ว) ดังนั้นเราจะถือว่าเครือข่ายโทรศัพท์เป็น "black box" เราไม่จำเป็นต้องเข้าไปยุ่งเกี่ยวกับการเดินทางของข้อมูลในเครือข่ายโทรศัพท์ แต่อย่างไรก็ตามข้อสำคัญประการหนึ่งที่ควรสนใจก็คือ คุณสมบัติของ black box นี้ (คุณสมบัติ นี้เป็นฟังก์ชั่นของทางเดินของข้อมูลที่ส่งออกไป) ตัวอย่างเช่น ค่าหน่วงเวลา (delay) ที่เกิดขึ้นในการส่งข้อมูลผ่านเครือข่ายโทรศัพท์เทียบกับค่าที่เกิดขึ้นในการส่งข้อมูลโดยใช้ดาวเทียมนั้นมีค่าแตกต่างกันมาก การเดินทางของข้อมูลผ่านทางสายโทรศัพท์จะใช้เวลาประมาณหนึ่งมิลลิวินาทีต่อระยะทาง 100 ไมล์ (โดยประมาณ) (แต่อย่างไรก็ตามการเชื่อมต่อของสายโทรศัพท์ผ่านทางชุมสายต่างๆ เป็นระยะทางไกลๆ สามารถทำให้ค่าหน่วงเวลา (delay) นี้มีค่าสูงมากถึง 20 มิลลิวินาทีได้) ส่วนในการส่งข้อมูลผ่านดาวเทียมนั้น ระยะทางจากพื้นโลกถึงดาวเทียมมีระยะทางเท่ากับ 22,000 ไมล์ แต่ละ "big hop" {big hop คือการเชื่อมโยงโดยดาวเทียม(Satellite link) } ที่ใช้ในการส่งข้อมูลนั้นกินเวลาอีก 700 มิลลิวินาที (จากพื้นโลกไปดาวเทียมแล้วกลับมายังพื้นโลกอีก) สำหรับทุกครั้งที่ใช้ในการส่งจากจุดหนึ่งไปยังอีกจุดหนึ่ง เวลาที่เพิ่มขึ้นนี้เป็นตัวตัดสินว่าเราควรจะเลือกใช้ตัวกลางที่ใช้ในการสื่อสารข้อมูลตัวไหนจึงจะเหมาะสม จากช่วงเวลาที่เสียไปตามที่กล่าวไว้ข้างต้นจะเห็นได้ว่าถ้าคอมพิวเตอร์ปลายทางหลักสะท้อน(echo) คาร์แรกเตอร์แต่ละตัวที่ได้รับจากคอมพิวเตอร์ต้นทางกลับมาที่คอมพิวเตอร์ต้นทางอีกครั้งหนึ่งจะกินเวลา (เดินทางไปกลับ) ถึง 1.4 วินาที (1400 มิลลิวินาที) ซึ่งค่าหน่วงเวลานี้(delay) มีค่ามากเกินกว่าที่จะนำไปใช้งานได้

ตัวอย่างที่ 2 จากรูปที่ 4 จะเห็นว่าเทอร์มินัล CRT ถูกต่อกับไมโครคอมพิวเตอร์ผ่านทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินเตอร์เฟสแบบ RS-232-C เรามีวิธีใดในการตัดสินว่าอุปกรณ์ตัวไหนเป็น DCE หรือ DTE จากที่ทราบมาแล้วว่า RS-232-C ใช้ในการสื่อสารข้อมูลระหว่าง DCE และ DTE ดังนั้นอุปกรณ์ตัวตัวใดตัวหนึ่งต้องเป็น DCE หรือ DTE เพื่อให้เข้าคู่กันได้ ตามปรกติ CRT และเทอร์มินัลที่เป็นตัวส่งหรือตัวรับข้อมูลจะถูกกำหนดเป็น DTE แต่ไมโครคอมพิวเตอร์สามารถเป็นได้ทั้ง DCE และ DTE ตัวอย่างเช่น พอร์ท I/O ของเครื่อง Cromemco เป็น DCE แต่พอร์ท I/O แบบอนุกรมของ TRS-80 เป็น DTE แสดงว่าคอมพิวเตอร์ของแต่ละบริษัทนั้นกำหนดลักษณะของพอร์ทแตกต่างจากกันออกไปสำหรับการตัดสินว่าพอร์ทใดเป็นแบบไหนจะอธิบายในหัวข้อการทดลอง

รูปที่ 4 ลักษณะของระบบที่ใช้ในตัวอย่างที่ 2

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของสัญญาณไฟฟ้า

เราจะอธิบายคุณสมบัติของสัญญาณไฟฟ้าของ RS-232-C ในหัวข้อนี้ โดยใช้รูปที่ 5 ประกอบ

1. สัญญาณที่ขาทุกขาที่คอนเน็คเตอร์ของ RS-232-C จะเป็นสภาวะ (status) ใดสภาวะหนึ่งในแต่ละคู่ของคู่ต่อไปนี้

MARK/SPACE

ON/OFF

logic 0/logic 1

ความสัมพันธ์ระหว่างสถานะของสัญญาณคู่ต่างๆ กับระดับแรงดันได้แสดงไว้ในตารางที่ 1 ขอให้สังเกตุด้วยว่า RS-232-C ใช้ลอจิกลบ(negative logic) แทนระดับแรงดันต่างๆ {ลอจิกลบ (negative logic)คือวิธีการเปรียบเทียบระดับแรงดันแบบหนึ่ง ถ้าระดับแรงดันหนึ่งค่าเป็นลบมากกว่าอีกระดับแรงดันหนึ่ง ระดับแรงดันที่มีค่าเป็นลบมากกว่าจะเป็นลอจิก "สูง" ดังนี้ 1 = -V, กราวด์ หรือ OFF ส่วน 0 = +V หรือ ON} โดยแรงดันของระดับสัญญาณต่างๆ จะถูกวัดเทียบกับเซอร์กิต Signal Ground ซึ่งจะกล่าวถึงในภายหลัง นอกจากนี้ช่วงของระดับแรงดันระหว่าง -3 ถึง +3 โวลต์ จะเป็นช่วงของการเปลี่ยนแปลงลอจิก ดังนั้นจึงไม่มีการระบุสถานะของสัญญาณในช่วงนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 แสดงความสัมพันธ์ระหว่างสถานะของสัญญาณคู่ต่างๆ กับระดับแรงดัน

Status	Signal Voltage	
	-25V<V1<-3V	3V<V1<25V
Binary logic		
state	1	0
Signal condition	MARK	SPACE
Function	OFF	ON

2. ในการแทนลอจิกหนึ่งหรือสถานะ MARK ตัวขับสัญญาณ (driver) ต้องจ่ายแรงดันระหว่าง -5 ถึง -15 โวลต์ ส่วนในการแทนลอจิก 0 หรือ SPACE ตัวขับสัญญาณต้องจ่ายแรงดันระหว่าง +5 ถึง +15 โวลต์

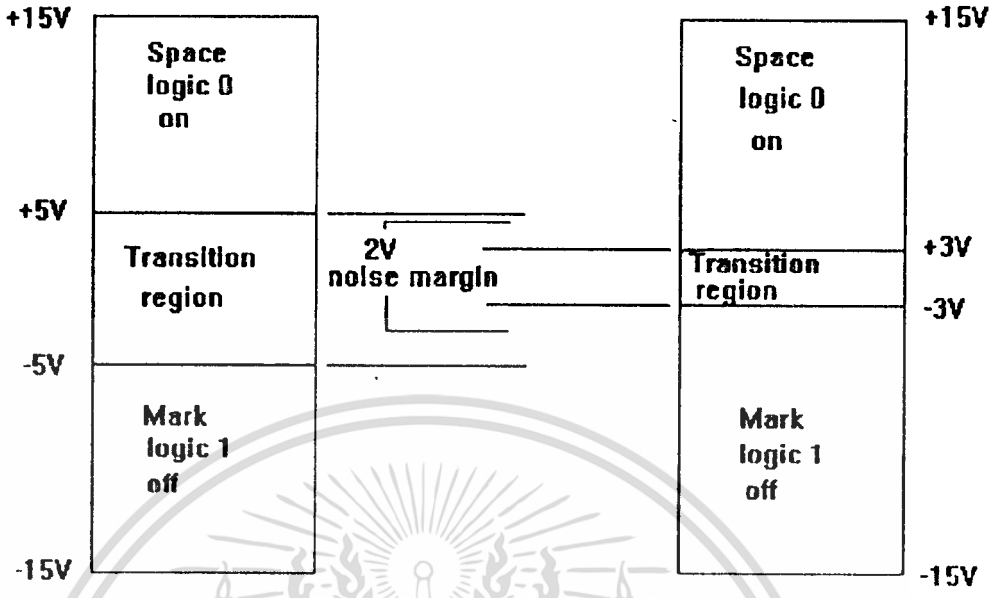
จากข้อ 1 และข้อ 2 แสดงว่า RS-232-C ยอมให้มี noise margin ได้ไม่เกิน 2 โวลต์ สำหรับความสัมพันธ์ระหว่างระดับแรงดันและสถานะของสัญญาณได้แสดงไว้ในรูปที่ 5 จากรูปจะเห็นได้ว่าถ้า line driver หรือตัวกำเนิดสัญญาณต้องการส่งลอจิก 0 line driver จะต้องจ่ายแรงดันระหว่าง +5 ถึง +15 โวลต์ ส่วน line receiver หรือตัวรับสัญญาณปลายทางจะถือว่าแรงดันที่อยู่ภายในช่วง +3 ถึง +15 โวลต์แทนลอจิก 0 จากการเปรียบเทียบระดับสัญญาณของตัวส่งและตัวรับจะเห็นว่า RS-232-C ยอมให้มีการ dropของสัญญาณในช่วง 2 โวลต์เกิดขึ้นได้ สำหรับในด้านการส่งลอจิก 1 ก็เป็นเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2 แสดงหน้าที่ของขาสัญญาณแต่ละขาของ connector แบบ RS-232-C

Interchange circuit	Connector pin assignment	Description	Gnd	Data		Control		Timing	
				From DCE	to DCE	From DCE	to DCE	From DCE	to DCE
AA	1	Protective Ground	X						
AB	7	Signal Ground/Common Return	X						
BA	2	Transmitted Data			X				
BB	3	Received Data		X					
	4	Request to Send					X		
	5	Clear to Send				X			
	6	Data Set Ready				X			
	20	Data Terminal Ready					X		
	22	Ring Indicator				X			
	8	Received Line Signal Detector				X			
	21	Signal Quality Detector				X			
	23	Data Signal Rate Selector (DTE)					X		
	23	Data Signal Rate Selector (DCE)				X			
DA	24	Transmitter Signal Element Timing (DTE)							X
DB	15	Transmitter Signal Element Timing (DCE)						X	
DD	17	Receiver Signal Element Timing (DCE)						X	
SBA	14	Secondary Transmitted Data			X				
SBB	16	Secondary Received Data		X					
SCA	19	Secondary Request to Send					X		
SCB	13	Secondary Clear to Send				X			
SCF	12	Secondary Received Line Signal Detector				X			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5 คุณสมบัติทางไฟฟ้าของอินเตอร์เฟสแบบ RS-232-C

จากที่ได้อธิบายมาอาจมีข้อสงสัยว่าทำไมไม่ใช้สถานะลอจิกแบบ TTL ซึ่งระดับของแรงดันมีค่าระหว่าง 0 ถึง +5 โวลต์ และทำไมถึงต้องใช้ระดับแรงดันระหว่าง -15 ถึง -3 และ +3 ถึง +15 โวลต์ด้วย

สาเหตุที่ไม่ใช้การแทนลอจิกแบบ TTL ก็เพราะสถานะลอจิกแบบ TTL ถูกรบกวนจากสัญญาณรบกวนต่างๆ ได้ง่าย นอกจากนี้ยังมีปัญหาเกี่ยวกับระยะทางที่สามารถทำการสื่อสารข้อมูลอีกด้วย สำหรับสาเหตุที่ต้องใช้แรงดันในช่วง -15 ถึง -3 และ +3 ถึง +15 ก็เพราะในขณะที่กำลังทำการพัฒนา RS-232-C ขึ้นนั้น ในวงจรคอมพิวเตอร์ต่างๆ โดยทั่วๆ ไปมีการใช้ระดับแรงดันในช่วงเหล่านี้อยู่ อนึ่งทรานซิสเตอร์ที่มีขายกันทั่วๆ ไปสามารถทำงานได้ในช่วงแรงดันเหล่านี้ และยังทนต่อสัญญาณรบกวนต่างๆ ที่มีเข้ามาได้ นอกจากนี้ยังสามารถทำงานที่ความถี่สูงๆ ได้ สูงถึง 20,000 บิท/วินาที (bps) ยิ่งกว่านั้นสถานะ MARK และสถานะ SPACE ยังถูกแทนด้วยการไหลของกระแสในทิศทางที่ตรงข้ามกันและความแตกต่างของแรงดันที่สถานะ MARK และ SPACE มีค่าสูงถึง 6 โวลต์เป็นอย่างน้อย ข้อดีต่างๆ ที่กล่าวมานี้ช่วยให้การส่งข้อมูลมีเสถียรภาพดี

3. ตัวเก็บประจุ C_L ที่ต่อขนานกับอุปกรณ์รับข้อมูลปลายทางจะต้องมีค่าไม่เกิน 2500 pF (พิโคฟารัด) โดนค่านี้ไม่รวมค่าความจุไฟฟ้าของสายเคเบิลเข้าไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

หมายเหตุ ตามข้อกำหนดข้อนี้ ระยะทางที่สามารถใช้ทำการสื่อสารข้อมูลได้ต้องไม่เกิน 50 ฟุต ซึ่งถูกกำหนดไว้ในมาตรฐาน RS-232-C

4. แรงดันขณะเปิดวงจรหรือขณะที่ไม่มีโหลด (V_O) จะต้องไม่เกิน 25 โวลต์

5. วงจรรับสัญญาณที่ใช้กับ RS-232-C ต้องสามารถทนต่อการลัดวงจรที่เกิดขึ้นได้(เช่นขา 2 ขาเกิดลัดวงจรโดยไม่ตั้งใจ) โดยไม่ทำให้เกิดความเสียหายต่อตัวมันเองหรืออุปกรณ์ที่เกี่ยวข้องด้วย เช่น เทอร์มินัล, โมเด็ม, พอร์ท I/O และอุปกรณ์ต่างๆ ที่ต่อเข้ากับเคเบิลที่ใช้ในการอินเตอร์เฟสแบบ RS-232-C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2 รายละอียดของขาต่างๆ พร้อมด้วยสัญญาณกำกับของคอนเน็คเตอร์

ขา	เซอร์กิต	ความหมายของเซอร์กิต
1	AA	Protective Ground
2	BA	Transmitted Data
3	BB	Received Data
4	CA	Request to Send
5	CB	Clear to Send
6	CC	Data set Ready
7	AB	Signal Ground
8	CF	Received Line Signal Detector
9/10	-	(Reserved for Data Set Testing)
11	-	Unassigned
12	SCF	Secondary Received Line Signal Dector
13	SCB	Secondary Clear to Send
14	SBA	Secondary Transmitted Data
15	DB	Transmit Signal Element Timing (DCE Source)
16	SBB	Secondary Received Data
17	DD	Receive Signal Element Timing
18	-	Unassigned
19	SCA	Secondary Request to Send
20	CD	Data Terminal Ready
21	CG	Signal Quality Detector
22	CE	Ring Indicator
23	CH/CI	Data Signal Rate Select (DTE/DCE Source)
24	DA	Transmit Signal Element Timing (DTE Source)
25	-	Unassigned

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติทางกลของการอินเตอร์เฟส

รายละเอียดของขาต่าง ๆ ของคอนเน็ตเตอร์ตามมาตราฐาน RS-232-C ได้แสดงไว้ในตาราง 3.2 ของให้สังเกตด้วยว่าตามมาตรฐาน RS-232 -C ไม่ได้กล่าวถึงปลั๊กตัวผู้ [plug] หรือปลั๊กตัวเมีย [socket] ของคอนเน็คเตอร์เลยว่าต้องมีรูปร่างลักษณะอย่างไร ในปัจจุบันเรามักจะใช้คอนเน็ตเตอร์แบบ DB-25 (บางทีเรียกแบบ D-type 25 pin connector) ในการอินเตอร์เฟสตมาตรฐาน PS-232-C คอนเน็คเอตร์แบบนี้เทียบเท่าแบบ ISO 1123 ซึ่งเป็นมาตรฐานท่ประกาศใช้โดย International Organization for Standardization [ISO] สำหรับรายละเอียดเพิ่มเติมของคอนเน็ตเตอร์แบบนี้หาได้จาก ISO /draft International Standard 2110. "Data Communication: 25-Pin DTE/DCE Interface Connector and Pin Assignments" [Revision of ISO 2110-1972], February 1979,.

ลักษณะการทำงานของเซอร์กิตต่าง ๆ

เซอร์กิจต่าง ๆ ได้แสดงไว้ในตารางที่2 ซึ่งสามารถแยกออกเป็นประเภทต่าง ๆ ได้ 5 ประเภทคือ

1. กราวนด์ หรือ Common Return [A]
2. เซอร์กิตข้อมูล [B]
3. เซอร์กิตควบคุม [C]
4. เซอร์กิตของสัญญาณฐานเวลา [timing circuit] [D]
5. เซอร์กิตของแชนแนลท่สอง [S]

จากตารางที่ 3 จะเห็นได้ว่า ตัวอักษรในวงเล็บที่ อยู่ท้ายเซอร์กิตประเภทต่าง ๆ จะเป็นตัวอักษรตัวแรกของกลุ่มตัวอักษร (ประกอบด้วยตัวอักษรสองหรือสามตัว) ซึ่งใช้กันทั่ว ๆ ไปในการอธิบายสัญญาณต่าง ๆ ที่ เกี่ยวข้องกับการใช้งาน RE-232-C จากตารางที่ 3 เซอร์กิตต่าง ๆ ของ RS- 232-C ถูกแบ่งออกเป็นประเภท ๆ โดยใช้กลุ่มของตัวอักษรดังอธิบายไว้ข้างต้น นากจากนี้ในตารางยังแสดงทิศการเคลื่อนที่ของข้อมูล สัญญาณควบคุมและสัญญาณเวลาว่าส่งจาก DCE หรือส่งจาก DTE รวมทั้งการกำหนดขาสัญญาณที่ใช้กำกับเซอร์กิตต่าง ๆ สำหรับลักษณะการทำงานของเซอร์กิตต่าง ๆ มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Circuit AA: Protective Ground

ลวดตัวนำของเซอร์กิตนี้จะถูกต่อเข้ากับตัวถังของอุปกรณ์เพื่อใช้เป็นสายดิน เมื่อเปรียบเทียบ Protective กับ Dignal Ground จะเห็นได้ว่า Signal Ground มีความสำคัญกว่ามาก ดังนั้น protective Ground จึงมักไม่ถูกต่อ การกระทำเช่นนี้ไม่เป็นการทำผิดข้อกำหนดในมาตรฐาน Ps-232-C เนื่องจากว่า Rs-232-C ได้กำหนดให้กรณีนี้เป็นกรณีเลือกใช้งาน [option]

Circuit AB: Signal หรือ Common Return

เซอร์กิตนี้ถูกใช้เป็นจุดอ้างอิงของสัญญาณท่ใช้ในเซอร์กิตต่าง ๆ ยกเว้นเซอร์กิต AA [Protective Ground] เซอร์กิตนี้เป็นเซอร์กิตเดียวที่ต้องถูกต่อไว้เสมอไม่ว่าเป็นการประยุกต์ใช้งานแบบใด

Circuit BB: Received Data

สัญญาณของเซอร์กิตนี้จะถูกส่งจาก DTE ไปยัง DCE.DTE จะทำให้เซอร์กิต BA [Transmitted Data] มีสถานะอลจิกเป็น 1 [MARK] ตลอดเวลาที่ไม่มีการส่งข้อมูล

ในระบบทุกระบบที่ใช้มาตรฐาน PS-232-C DTE จะไม่ทำการส่งข้อมูลนอกจากว่าเซอร์กิตต่อไปนี้มีสถานะลอจิกเป็น O [ON]

1. เซอร์กิต CA [Reguest to Send]
2. เซอร์กิต CB [clear to Send]
3. เซอร์กิต CC [Data Set Ready]
4. เซอร์กิต CD [Data Terminal Ready]

สำหรับการทำงานร่วมกันของเซอร์กิตเหล่านี้จะอธิบายในหัวข้อถัดไป

ในบางกรณีเราอาจต้องใช้ null malem ร่วมกับเซอร์กิตนี้ ตัวอย่างเช่น ในระบบคอมพิวเตอร์ระบบหนึ่ง เราทำการอินเตอร์เฟสคอมพิวเตอร์เฟสคอมพิวเตอร์ของเราเข้ากับเทอร์มินัลผ่านทาง RS-232-C ถ้าเทอร์มินัลเป็น DTE (ปกติมักจะเป็นเช่นนี้ด้วย) และคอมพิวเตอร์ทำตัวเป็น DCE คอมพิวเตอร์จะคอยรับสัญญาณที่ส่งจากเทอร์มินัลผ่านทางสาย Transmitted Data (การส่งนี้เราใช้ DTE เป็นตัวอ้างอิง ไม่ใช่ DCE) แต่ถ้าคอมพิวเตอร์ทำตัวเป็น DTE และส่งข้อมูลผ่านทางเซอร์กิตนี้ก็จะเกิดปัญหาขึ้นคือ อุปกรณ์ทั้งสองตัวจะส่งข้อมูลลงบนสายเส้นเดียวกัน ทำให้ข้อมูลเกิดการต้านกันจึงต้องใช้ null modem ช่วยแก้ปัญหานี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่สำคัญอีกข้อหนึ่งก็คือ ในเซอร์กิตบางเซอร์กิตซึ่งจะต้องอยู่ในสภาวะ ON ก่อนที่จะมีการส่งข้อมูลแต่กลับไม่อยู่ในสภาวะ ON สาเหตุนี้อาจเกิดจากการทำงานที่ผิดพลาดของตัวเทอร์มินัล. พอร์ท I/O ของคอมพิวเตอร์ หรือเกิดจากตัวสายเคเบิลที่ต่อระหว่างเทอร์มินัลกับคอมพิวเตอร์ ถ้ามีปัญหาเหล่านี้เกิดขึ้นในขณะที่เราส่งข้อมูลโดยใช้ RS-232-C ปัญหาของเราอาจเกิดจากสัญญาณใดสัญญาณหนึ่งข้างต้นเป็น OFF หรือเทอร์มินัลของเราทำการส่งข้อมูลบนสายเส้นเดียวกับที่คอมพิวเตอร์ใช้ส่งข้อมูล สำหรับการตรวจสอบข้อผิดพลาดเหล่านี้จะอธิบายในตอนท้ายของบทนี้

Circuit BB: Received Data

สัญญาณของเซอร์กิตนี้จะถูกส่งจาก DCE ไปยัง DTE เซอร์กิตนี้จะอยู่ในสถานะ MARK (ลอจิก1) ตลอดเวลาที่ไม่มีการส่งข้อมูล

ในการส่งข้อมูลแบบ half-duplex เมื่อ Reguest to send (เซอร์กิต CA) อยู่ในสภาวะ ON (ลอจิก O) Received Data (เซอร์กิต BB) จะมีสภาวะเป็น OFF (ลอจิก1) นอกจากนี้ Received Data จะคงอยู่ในสภาวะ OFF อีกช่วงระยะเวลาสั้น ๆ ระยะเวลาหนึ่งหลังจากที่ reguest to Send เปลี่ยนสภาวะจาก ON ไปเป็น OFF เมื่อการส่งข้อมูลเกิดขึ้นกลังจากที่ Reguest to Send เปลี่ยนสภาวะจาก ON ไปเป็น OFF เมื่อการส่งข้อมูลเกิดขึ้นเรียบร้อยแล้ว

Circuit CA: Request to Send

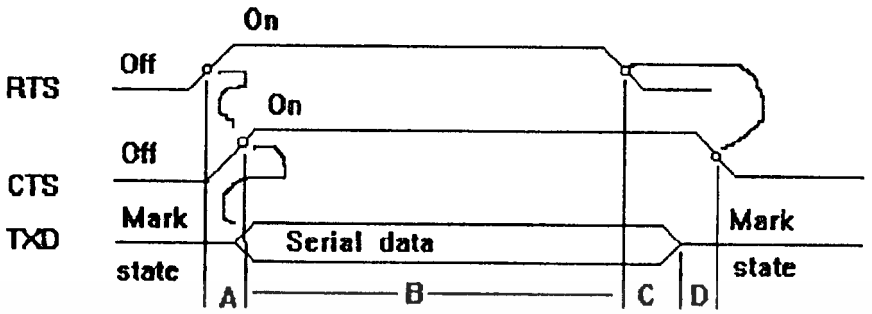
สัญญาณ REguest to Send [RTS] และสัญญาณ Clear to Send [CTS หรือเซอร์กิต CB] ซึ่งเกิดขึ้นในการส่งข้อมูลระหว่าง DTE และ DCE แสดงไว้ในรูปที่ 6 และ 7

ในการส่งข้อมูลแบบ simplex และ full-duplex เมื่อ reguest to Send มีสถานะของลอจิกเป็น O [ON] จะทำให้ DCE อยู่ในโหมดการส่งข้อมูล [transmit mode]. ในที่นี้การที่ DCE อยู่ในโหมดการส่งข้อมูลหมายความว่า DCE อยู่ในโหมดการส่งข้อมูล [communication link] ตัวอย่างเช่น ถ้า DCE ของเราเป็นโมเด็ม,เมื่อ DCE จะไม่อยู่ในโหมดการส่งข้อมูล (คือDCE จะไม่ส่งข้อมูลที่ได้รับจาก DTE ออกไปยังตัวกลางในการสื่อสารข้อมูล)

ในกรณี half-duplex เมื่อ Reguest to Send อยู่ในสถานะ ON,DCE จะอยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการส่งข้อมูล แต่ถ้า Reguest to Send อยู่ในสภาวะ OFF, DCE จะอยู่ในโหมดการรับข้อมูล (คือ DCE จะรับข้อมูลจากเครือข่ายการสื่อสารและส่งข้อมูลเหล่านี้ไปยัง DTE)



รูปที่ 6 การทำ handshake ของสัญญาณ RTS และ CTS ในช่วง A DTE จะป้อนสัญญาณ RTS แสดงให้ DCE ทราบว่า DTE ต้องการส่งข้อมูลซึ่งจะเกิดขั้นตอนเหล่านี้คือ DCE จะจัดตั้งช่องทางการสื่อสารและป้อนสัญญาณ CTS (เป็น ON) ซึ่งแสดงให้ DTE ทราบว่า DTE สามารถเริ่มต้นส่งข้อมูลได้แล้ว แต่ TXD จะยังอยู่ในสภาวะ MARK อยู่ในช่วง B ข้อมูลจะถูกส่งผ่านทางเซอร์กิต transmitted data เมื่อข้อมูลถูกส่งออกไปจนหมดแล้ว DTE จะ OFF สัญญาณ RTS เพื่อบอกให้ DCE ทราบว่า DTE ไม่ต้องการส่งข้อมูลอีกต่อไป ในช่วง C เมื่อ DCE ส่งข้อมูลทั้งหมดออกไปยัง communication link เสร็จแล้ว เซอร์กิต TXD จะกลับเข้าสู่สภาวะ MARK ในช่วง D DCE แจ้งให้ DTE ทราบว่า DCE พร้อมแล้วที่จะรับข้อมูลชุดใหม่เพื่อส่งออกไปโดยการ OFF สัญญาณ CTS

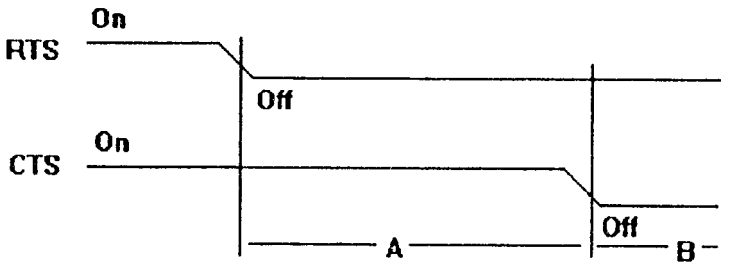
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 6 ขอให้ดูที่สภาวะการเปลี่ยนแปลงจาก ON ไปเป็น Off และ OFF ไป ไปเป็น ON ของสัญญาณ Reguest to Send การเปลี่ยนสภาวะของสัญญาณ Reguest to Send จาก OFF เป็น ON จะทริกก์ [trigger] DCE ให้อยู่ในโหมดการส่งข้อมูล และทำขั้นตอนต่าง ๆ ที่ให้การสื่อสารข้อมูลเกิดขึ้นได้ เช่น การต่อโทรศัพท์ไปยังคอมพิวเตอร์หลัก [host computer] (ถ้า DCE ตัวนั้นสามารถต่อโทรศัพท์โดยอัตโนมัติได้) ในระบบไมโครคอมพิวเตอร์ทั่ว ๆ ไป สาย Reguest to Send จะถูกต่อกับสาย Coear to Send โดยตรง ดังนั้นเมื่อไรก็ตามที่ DTE ป้อนสัญญาณ Reguest to Send, DTE ก็จะได้รับสัญญาณตัวนี้กลับมา โดยส่งมาทางสาย Clear to Send (รายละเอียดเพิ่มเติมของเรื่องนี้จะกล่าวในการทดลอง) เมื่อการทำขั้นตอนต่าง ๆ ที่ทำให้เกิดการสื่อสารข้อมูลได้เกิดขึ้นเรียบร้อยแล้ว DCE จะทำให้ Clear to Send (เซอร์กิต CB) มีมีค่าลอจิกเป็น 0 [ON] ซึ่งเป็นการบอกให้ DTE ทราบว่า สามารถส่งข้อมูลทางเซอร์กิต Transmit Data ข้ามจุดเชื่อมต่อ [infterface point] ได้แล้ว

การเปลี่ยนสถานะจาก ON ไปเป็น OFF ของสาย RTS เป็นการส่งให้ DCE ส่งข้อมูลที่ยังเหลืออยู่ที่จุดเชื่อมต่อ [interface point] ของเซอร์กิต Transmitted Data ออกไปยังช่องทางการสื่อสาร และออกจากโหมดการส่งข้อมูล (ในกรณีของ full-duplex หรือ simplex) หรือเข้าสู่โหมดการรับข้อมูล (ในกรณีของ half-duplex) DCE จะตอบสนองต่อสัญญาณนี้โดยทำให้สัญญาณ Coear to Send มีลอจิกเป็น 1 [OFF]

จากรูปที่ 6 เมื่อใดก็ตามที่ Reguest to Send เปลี่ยนสถานะจาก ON เป็น OFF Reugest to Send จะ ON ใหม่ได้อีกครั้งหนึ่งก็ต่อเมื่อ DCE สั่งให้ Clear to Send เปลี่ยนสถานะจาก ON เป็น OFF แล้ว การทำเช่นนี้เป็นการป้องกันไม่ให้เกิด overrun error ขึ้น ซึ่งก็คือ DTE ทำการส่งข้อมูลชุดใหม่มาอีก ในขณะที่ DCE ยังส่งข้อมูลชุดเก่าไม่เรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 ในช่วง (A) DTE ไม่สามารถ ON สัญญาณ RTS ใหม่อีกครั้งหนึ่งได้ DTE ต้องรอจน

กระทั่ง DCE ส่งข้อมูลที่เหลือออกไปจนหมด โดย DCE จะ OFF สัญญาณ CTS

เพื่อแสดงให้ DTE ทราบว่ามันพร้อมที่จะรับข้อมูลชุดใหม่แล้วในช่วง (B) DTE สา

มารถ ON สัญญาณ RTS ใหม่เมื่อใดก็ได้เนื่องจาก CTS มีสภาวะเป็น OFF

การทำ handshake โดยใช้ Reguest to Send กับ Clear to Send ที่อธิบายไปนั้นใช้ได้ทั้งในการส่งข้อมูลทีละคาร์แรกเตอร์ หรือทีละ block (เราใช้โปรโตคอลที่มีระดับสูงกว่า Physical-level protocol ในการกำหนดลักษณะของคาร์แรกเตอร์หรือบล็อก) ตัวอย่างเช่นสมมุติให้หนึ่งคาร์แรกเตอร์ประกอบด้วยบิทต่าง ๆ 10 บิท DTE จะป้อนสัญญาณลงในสาย Reguest to Send และคอยรับสัญญาณจาก DCE ทางสาย Clear to Send สำหรับการทำ handshake ในการส่งข้อมูลทีละบล็อกนั้น DTE จะส่งคาร์แรกเตอร์พิเศษที่บอกจุดนิ้นสุดของบล็อก [end of transmission character] เพิ่มเข้าไปด้วย และเมื่อถึงจุดสิ้นสุดของบล็อกข้อมูล DTE จะ OFF สัญญาณ Reguest to Send ในการตอบสนองต่อเหตุการณ์เหล่านี้ DCE จะ OFF สัญญาณ Clear to Send เมื่อคาร์แรกเตอร์พิเศษที่บอกจุดสิ้นสุดของข้อมูลได้ถูกส่งจาก DCE ออกไปยังเครือข่ายการสื่อสาร [communication network] เรียบร้อยแล้ว

จากที่อธิบายไว้ในเซอร์กิต BA, DTE จะส่งข้อมูลได้ก็ต่อเมื่อ Reguest to Send, Clear to Send, Data Set Ready และ Data Terminul Ready และ Data Set Ready ON เรียบร้อยแล้ว สัญญาณ Reguest to Send จะ ON ตามมาและ DCE จะตอบสนองต่อสัญญาณนี้ โดยการส่งสัญญาณ Coear to Send เป็น ON กลับมายัง DTE, DTE จะส่งข้อมูลออกทางเซอร์กิต Transmitted Data ได้ (เราสามารถ ON สัญญาณ Reguest to Send เมื่อไรก็ตามที่ Clear to Send เป็น OFF โดยไม่ต้องสนใจสภาวะของเซอร์กิตอื่น ๆ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่น่าสนใจอีกข้อหนึ่งคือ สัญญาณ Reguest to Send เป็นสัญญาณที่ใช้ระหว่าง DTE-DCE เท่านั้น เนื่องจากว่าการเชื่อมโยงโดยสายโทรศัพท์. ไมโครเวฟ หรือดาวเทียมจะแยก DCE ทางด้านผู้ส่ง [lecal] และผู้รับออกจากกัน [remote] ดังนั้นสัญญาณนี้จะไม่ถูกส่งไปยังอุปกรณ์ใด ๆ ที่อยู่ทางด้านที่เราจะติดต่อด้วย [remote] และสัญญาณนี้ก็ไม่ได้บอกสภาวะ [status] ของเครื่องทางด้านผู้รับด้วยเช่นกัน

การใช้งานเซอร์กิต Reguest to Send ยังเป็นตัวก่อให้เกิดปัญหาในการอินเตอร์เฟสขึ้นอีกด้วย โดยเฉพาะในหารใช้งานร่วมกับระบบไมโครคอมพิวเตอร์ที่ใช้อุปกรณ์พวก USART [Universal Synchronous/Asynchronous Receive/Tarnsmit] เป็นพอร์ท I/O แบบอนุกรม สาเหตุของปัญหาข้อนี้จะอธิบายในตอนท้ายของบทนี้

Circuit CB: Clear to Send

สัญญาณเป็นสัญญาณควบคุมซึ่งถูกส่งจาก DCE ไปยัง DTE เพื่อบอกให้ DTE ทราบว่า DCE พร้อมที่จะรับข้อมูลที่จะส่งมาจากDTE บนสาย Transmitted Data แล้ว เมื่อสัญญาณ Clear to Send อยู่ในสภาวะ ON รวมทั้งสัญญาณ Reguest to Send, Data Set Ready หรือ Data Terminal Ready มีสภาวะเป็น ON ด้วย การ ON ของสัญญาณเหล่านี้จะบอกให้ DTE ทราบว่าข้อมูลที่ส่งไปยัง DCE จะถูก DCE รับไว้และส่งต่อไปยัง Communication Channel เมื่อสัญญาณ Clear to Send อยู่ในสภาวะ OFF จะบอกให้ DTE ทราบว่า DCE ไม่พร้อมที่จะรับข้อมูล ดังนั้น DTE จะยังไม่ส่งข้อมูลออกมา (ต้องป้อนสัญญาณ RTS ใหม่)

Clear to Send จะอยู่ในสภาวะ ON ก็ต่อเมื่อสัญญาณ Reguest to send (เซอร์กิต CA) และ Data Set Ready (เซอร์กิต CC) จะอยู่ในสภาวะ ON ทั้งคู่ ถ้าไม่ใช้ขา Request to Send ให้ถือว่าสัญญาณ Request to Send เป็น ON ตลอดเวลา ดังนั้นสภาพสัญญาณของ Clear to Send จะเป็นอย่างไรจึงขึ้นอยู่กับสถานะของสัญญาณ Data set Ready ว่าเป็น ON หรือ OFF

ในการอินเตอร์เฟสตามมาตรฐานของ RS-232-C ซึ่งทำการอินเตอร์เฟสระหว่าง DTE และ DCE ในกรณีที่มีเครือข่ายสวิชชิ่งโทรศัพท์เข้าไปเกี่ยวพันด้วย เราจะต้องใช้เซอร์กิตต่อไปนี้เพิ่มเข้าไปด้วย

1. เซอร์กิต CC: Data Set Ready
2. เซอร์กิต CD: Data Terminal Ready
3. เซอร์กิต CE: Ring Indicator
4. เซอร์กิต CF: Received Line Signal Detector

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากการทำงานของเซอร์กิตต่อไปนี้เกี่ยวพันกับเครือข่ายโทรศพท์ ดังนั้นถ้าเรา เราทำการติดต่อในระยะสั้นๆ (ไม่ผ่านเครือข่ายโทรศัพท์) โดยใช้ RS-232-C เพียงอย่างเดียว สามารถตัดเซอร์กิตเหล่านี้ออกไปได้ การประยุกต์ใช้งานที่น่าสนใจมากที่สุดของเซอร์กิตเหล่า คือ การใช้งานร่วมกับอุปกรณ์ที่สามารถต่อโทรศัพท์ได้โดยอัตโนมัติ (auto dial) และอุปกรณ์ที่สามารถตอบรับต่อการเรียก (โทรศัพท์ที่มีเข้ามา) โดยอัตโนมัติ (auto answer)

หน้าที่สำคัญของสัญญาณ Data Set Ready และ Data Terminal Ready คือมัน เป็นตัวแสดงให้เราทราบว่าอุปกรณ์ของเราพร้อมที่จะทำการสื่อสารข้อมูล (Equipment Readiness) หรือไม่ สำหรับรายละเอียดของเรื่องนี้จะอธิบายในหัวข้อ "การส่งข้อมูลแบบ full duplex" ถ้า Data Set Ready อยู่ในสภาวะ ON จะบอกให้ DTE ทราบว่า DCE พร้อมจะส่งข้อมูลที่รับจาก DTE ออกไปยังเครือข่ายการสื่อสารแล้ว (เช่น เครือข่ายโทรศัพท์, ไมโครเวฟ) ในลักษณะเดียวกัน ถ้า Data Terminal Ready อยู่ในสภาวะ ON แสดงว่า DTE พร้อมที่จะส่งข้อมูลไปให้ DCE โดย ส่งข้อมูลออกทางเซอร์กิต Transmitted Data ดังที่ได้อธิบายไว้แล้วใน "Circuit BA: Transmitted Data" ทั้ง Data Set Ready และ Data Terminal Ready จะต้องอยู่ในสภาวะ ON ก่อนที่จะมีการส่ง ข้อมูลเกิดขึ้น แต่เนื่องจากสัญญาณเหล่านี้ไม่ค่อยถูกใช้ในกรณีที่ทำการอินเตอร์เฟสกับระบบคอม พิวเตอร์ ดังนั้นข้อมูลจะถูกส่งเมื่อใดจึงขึ้นอยู่กับสภาวะของสัญญาณ Request to Send และ Clear to Send

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Circuit CC: Data Set Ready

สัญญาณนี้เป็นสัญญาณควบคุมที่ส่งจาก DCE ไปยัง DTE ในกรณีที่ Data Set Ready อยู่ในสภาวะ ON แสดงว่า DCE ได้ถูกต่อกับ Communication Channel เรียบร้อยแล้ว ในในกรณีที่ DCE ของเราสามารถต่อโทรศัพท์ได้โดยอัตโนมัติ การที่ Data Set Ready เป็น ON หมายความว่า DCE ของเรา (local) ได้ต่อโทรศัพท์(เรียก) DCE ของเครื่องคอมพิวเตอร์ที่เราต้องการติดต่อด้วย(remote) ได้ตอบรับต่อการเรียกทำให้เกิดการเชื่อมต่อกันของ Communication Channel ขึ้นระหว่าง DCE ทั้งสองด้าน ทำให้สามารถทำการสื่อสารข้อมูลระหว่างกันได้ เมื่อ Communication Channel ถูกเชื่อมต่อแล้ว ระบบเข้าสู่โหมดการส่งข้อมูล (ไม่ใช่โหมดการส่งสัญญาณเสียง คือเราพูดสายไม่ได้)

Circuit CD: Data Terminal Ready

สัญญาณควบคุมตัวนี้จะถูกส่งจาก DTE ไปยัง DCE สัญญาณ Data Terminal Ready ต้องอยู่ในสภาวะ ON ก่อนที่ DCE จะ ON สัญญาณ Data Set Ready {ซึ่งบอกให้รู้ว่า DCE ถูกเชื่อมต่อเข้ากับช่องทางการสื่อสาร Communication Channel) แล้ว และสามารถส่งข้อมูลได้ } ขณะใดก็ตามที่ DCE ต่อกับช่องทางการสื่อสารแล้ว Data Terminal Ready เปลี่ยนสภาวะจาก ON เป็น OFF, DCE จะตัดการเชื่อมต่อระหว่าง DCE กับ Communication Channel ทิ้งในทันที

Circuit CE: Ring Indicator

สัญญาณนี้เป็นสัญญาณควบคุมที่ส่งจาก DCE ไปยัง DTE เมื่อสัญญาณนี้มีสภาวะเป็น ON แสดงว่า DCE กำลังได้รับสัญญาณเสียงวกริ่ง (เงียบและดังเป็นช่วงๆ) และในกรณีที่ DCE ไม่ได้รับสัญญาณเสียงกริ่ง สัญญาณ Ring Indicator จะมีสภาวะเป็น OFF เราใช้สัญญาณควบคุมตัวนี้ในกรณีที่ใช้โมเด็มที่สามารถตอบรับต่อการเรียกได้โดยอัตโนมัติ (autoanswer)

Circuit CF: Received Line Signal Detector

สัญญาณนี้ส่งจาก DCE ไปยัง DTE เมื่อ DCE ได้รับสัญญาณ Carrier (ซึ่งต้องเป็นไปตามข้อกำหนดของบริษัทผู้ผลิต DCE ด้วย) ที่ส่งมาจาก DCE อีกด้านหนึ่ง (remote side) สัญญาณ Received Line Signal Detector จะมีสภาวะเป็น ON แสดงว่า DCE จับสัญญาณใน Communication Channel ซึ่งจะถูกนำไป demodulate ได้แล้ว ในโมเด็มแบบต่างๆ สายเส้นนี้จะถูกต่อกับ LED เพื่อแสดงให้รู้ว่าขณะนี้มีสัญญาณ carrier เข้ามาหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ : สัญญาณนี้มีชื่อเรียกชื่อหนึ่งว่า Data Carrier Detector (DCD)

ในการใช้งานเทอร์มินอลที่มีจำหน่ายในท้องตลาด ส่วนมากเราต้องให้สัญญาณนี้มีสภาวะเป็น high ตลอดเวลาก่อนที่จะรับหรือส่งข้อมูล ตัวอย่างของเทอร์มินอลที่เราต้องทำเช่นนี้ได้แก่ Texas Instrument TI Silent 700 Series ซึ่งมี acoutic-coupled modem ติดตั้งไว้ภายในและมีพอร์ท RS-232-C ให้ใช้ในการต่อกับโมเด็มภายนอก Received Line Signal Detector จะมีสภาวะเป็น ON เมื่อโมเด็มภายในตรวจจับ Carrier ที่ส่งมาจากโมเด็มอีกด้านหนึ่ง (remote) ได้ (โมเด็มตัวนี้ต้องสามารถตอบรับการเรียกได้โดยอัตโนมัติ) ในอีกกรณีหนึ่งถ้าเราต้องการอินเตอร์เฟส Selent 700 เข้ากับไมโครคอมพิวเตอร์โดยตรง (ไม่ผ่านโมเด็ม) เรายังต้องทำให้สาย Received Line Signal Detector มีสภาวะเป็น ON เสมอ โดยทั่วๆ ไปมักจัดต่อขาสัญญาณนี้เข้ากับขาสัญญาณตัวอื่นที่มีอยู่บนคอนเน็คเตอร์ เช่นขา Data Terminal Ready เทคนิคนี้เรียกว่า "Jumping" สำหรับเซอร์กิตที่เหลือคือ

Circuit CG: Signal Quality Detector

Circuit CH: Data Signal Rate Selector (DTE Source)

Circuit CI: Data Signal Rate Selector (DCE Source)

Circuit DA: Transmit Signal Timing (DTE Source)

Circuit DB: Transmit Signal Timing (DCE Source)

Circuit DD: Receiver Signal Timing (DCE Source)

Circuit SBA: Secondary Transmit Data

Circuit SBB: Secondary Received Data

Circuit SCA: Secondary Request to Send

Circuit SCB: Secondary Clear to Send

Circuit SCF: Secondary Receive Signal Detector

<u>โครงสร้างทั่วๆ ไปของมาตรฐาน RS-232-C</u>

ดังที่ได้กล่าวไว้ในตอนต้น มาตรฐาน RS-232-C ไม่ได้กำหนด ลักษณะโครงสร้างของคอนเน็คเตอร์ไว้เลย ดังนั้นเราสามารถเลือกใช้คอนเน็คเตอร์แบบใดก็ได้ อย่างไรก็ตาม EIA ได้ระบุไว้ว่าคอนเน็คเตอร์ที่เข้ามาตรฐานของทหารแบบ MIL-C-24308(MS-18275) สามารถใช้ในการอินเตอร์เฟสตามมาตรฐาน RS-232-C ได้เป็นอย่างดี (แต่ EIA ได้กล่าวย้ำไว้ด้วยว่ามาตรฐาน MIL นี้ไม่ได้เป็นส่วนใดส่วนหนึ่งในมาตรฐาน RS-232-C) การขาดมาตรฐานของคอนเน็คเตอร์นี้ได้ก่อให้เกิดปัญหาบางประการขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากมาตรฐาน RS-232-C เราจะเห็นว่ามีการใช้สัญญาณต่างๆ อยู่เป็นจำนวนมาก ดังนั้นในการใช้งานเราอาจสงสัยว่า ควรจะใช้สัญญาณเส้นใดบ้างจึงเหมาะกับระบบของเรา (จากทั้งหมด 21 เซอร์กิต) ข้อสงสัยข้อนี้ตอบได้ยากมาก เนื่องจากรูปแบบของกลุ่มสัญญาณที่ใช้ในในการสื่อสารข้อมูลมีแตกต่างกันออกไปมากมาย ขึ้นอยู่กับลักษณะของระบบที่ต้องการใช้ โครงสร้างของระบบอาจเป็นไปได้หลายแบบตั้งแต่การต่อเทอร์มินัลเข้ากับเครื่องคอมพิวเตอร์ ซึ่งซึ่งเป็นการอินเตอร์เฟสแบบง่ายๆ, ใช้เครื่องคอมพิวเตอร์ของเราร่วมกับเครื่องคอมพิวเตอร์เครื่องอื่นโดยอาศัยการมัลติเพล็กซ์ (multiplexed), ใช้ส่งและรับข้อมูลแบบซิงโครนัส(synchronous) หรือใช้กับสาย dedicated line (สายที่ไชในการส่งข้อมูลเพียงอย่างเดียว ใช้ส่งสัญญาณเสียงไม่ได้) ซึ่งเป็นการใช้ร่วมกับเทอร์มินัลปลายทางเครื่องอื่นๆ อย่างไรก็ตาม EIA ไแบ่งมาตรฐานของการใช้สายสัญญาณออกเป็นกลุ่มตามสภาพของระบบต่างๆ กลุ่มของสัญญาณที่ใช้ร่วมกับระบบไมโครคอมพิวเตอร์ถูกแบ่งออกเป็น 7 กลุ่มดังนี้

- ใช้ในการส่งข้อมูลอย่างเดียว (Transmit only)
- ใช้ในการส่งข้อมูลอย่างเดียวแต่ใช้สัญญาณ RTS ด้วย
- ใช้ในการรับข้อมูลอย่างเดียว (Receive only)
- ใช้ส่งและรับข้อมูลแบบฮาร์ฟดูเพล็ก (Half Duplex)
- ใช้ส่งและรับข้อมูลแบบฟูลดูเพล็ก (Full Duplex)
- ใช้ส่งและรับข้อมูลแบบฟูลดูเพล็ก แต่ใช้สัญญาณ RTS ด้วย
- แบบพิเศษ (Special)

การอินเตอร์เฟสแบบต่างๆ ที่กล่าวมานี้ได้แสดงไว้ในตารางที่ 4 ซึ่งได้ระบุสายสัญญาณใดบ้างที่ต้องต่อในการอินเตอร์เฟสแต่ละแบบ (เราได้ตัดสัญญาณที่ใช้ในการส่งข้อมูลแบบซิงโครนัสทิ้งไป เนื่องจากว่าในระบบไมโครคอมพิวเตอร์มักจะไม่ใช้การส่งข้อมูลแบบนี้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4 RS-232-C Standard Configurations

RS-232-C Interchange circuit Spacial		Transmit only	Terminal only with RTS	Receive only	Half duplex	Full duplex	Full Duplex with RTS
1 0	Protective Ground	-	-	-	-	-	-
7 x	Signal Ground	x	x	x	x	x	x
2 0	Transmitted Data	x	x		x	x	x
3 0	Received Data			x	x	x	x
4 0	Request to Send		x		x		x
5 0	Clear to Send	x	x		x	x	x
6 0	Data Set Ready	x	x	x	x	x	x
20 0	Data Terminal Ready	s	s	s	s	s	s
22 0	Ring Indicator	s	s	s	s	s	s
8 0	Received Line Signal Detector			x	x	x	x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

x = ใช้งานในทุกกรณี

s = ใช้ในกรณีที่ระบบของเราเดี่ยวพันกับเครือข่ายโทรศัพท์ด้วย

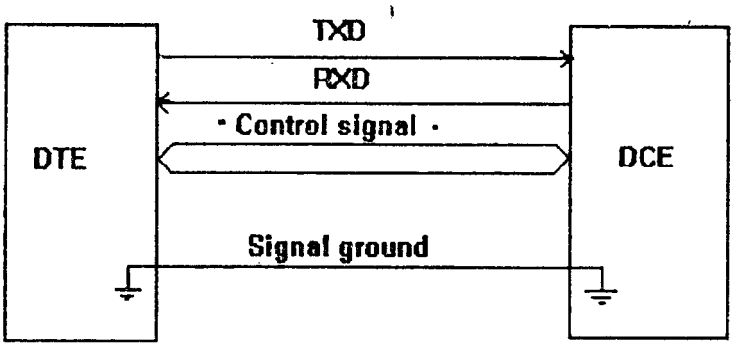
0 = เลือกใช้ตามลักษณะของระบบเรา

จากตารางที่ 4 จะเห็นว่ามีเซอร์กิตอยู่เซอร์กิตหนึ่งที่ต้องต่ออยู่เสมอคือ เซอร์กิต Signal Ground (ขา 7) ส่วนเซอร์กิตอื่นจะต่อหรือไม่ขึ้นกับการประยุกต์ใช้งานของเรา ให้เลือกจากวิธีการส่งข้อมูลของเราว่าเป็นแบบใดและต้องดูด้วยว่าเข้ากับระบบของเราได้หรือไม่

ในการส่งข้อมูลแบบ full-duplex นั้น ระบบไมโครคอมพิวเตอร์บางระบบอาจใช้เซอร์กิตต่างออกไปจากที่แสดงไว้ในตารางที่ 4 ในระบบไมโครคอมพิวเตอร์ ส่วนใหญ่มักใช้อุปกรณ์ที่ส่งข้อมูลในทิศทางเดียว เช่น ใช้รับข้อมูลอย่างเดียวซึ่งก็คือเครื่องพิมพ์ หรือส่งข้อมูลอย่างเดียว เช่น จอย สติกค์ (joystick) แต่เนื่องจากระบบเคเบิลแบบ full duplex ตามมาตรฐาน RS-232-C ถูกออกแบบให้ใช้ส่งข้อมูลได้ในสองทิศทาง ดังนั้นเราสามารถใช้สายเคเบิลกับอุปกรณ์ที่ใช้รับข้อมูลอย่างเดียว เช่น เครื่องพิมพ์ หรือจะนำไปใช้กับอุปกรณ์ตัวอื่นได้ในทันทีโดยไม่ต้องแก้ไขเพิ่มเติมอะไรอีก

การประยุกต์ใช้งานของการส่งข้อมูลแบบ full duplex ที่ใช้กันมากในระบบไมโครคอมพิวเตอร์คือ ใช้กับเทอร์มินัลที่ใช้ทำหน้าที่รับและส่งข้อมูลดังนี้ คาร์แรกเตอร์จะถูกส่งจากคีย์บอร์ดไปยังไมโครคอมพิวเตอร์และถูกสะท้อน (echo) ไปแสดงบนจอภาพ หรือพิมพ์ออกที่เครื่องพิมพ์ ในกรณีเช่นนี้ ข้อมูลจะเคลื่อนที่ในสองทิศทางพร้อมๆ กัน โดยถูกส่งและรับจาก DTE (คีย์บอร์ดและจอภาพ) กับ DCE (พอร์ทอินพุท/เอาท์พุท แบบอนุกรมของไมโครคอมพิวเตอร์) ดังแสดงในรูปที่ 8 ในตารางที่ 4 ได้แสดงการอินเตอร์เฟสแบบ full duplex ไว้สองแบบด้วยกันคือ แบบที่ใช้และไม่ใช้สาย Request to Send ซึ่งเป็นไปตามที่กำหนดไว้โดย EIA การใช้สายสัญญาณ Request to Send นั้นสามารถเลือกใช้ได้ แต่ถ้าพอร์ทอินพุท/เอาท์พุทของไมโครคอมพิวเตอร์ของเราเป็นแบบ USART เราต้องใช้สัญญาณ RTS เสมอ ดังนั้นในการประกอบสายเคเบิล เราควรต่อสาย RTS เผื่อไว้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8 ลักษณะพื้นฐานของการส่งข้อมูลแบบ full-duplex

ลักษณะการต่อ RS-232-C ที่ไม่ใช้แบบมาตรฐาน

ในรูปที่ 9 แสดงไดอะแกรมรูปภาพของการต่อ RS-232-C แบบต่างๆ ที่ไม่ใช่แบบมาตรฐาน ในระบบไมโครคอมพิวเตอร์มักจะใช้การต่อเคเบิล RS-232-C ตามรูปที่ 9 รูปที่ 9.a แสดงการต่อสายเคเบิลแบบ full-duplex ซึ่งถูกต้องตามมาตรฐานทุกอย่าง ลักษณะการทำงานของการต่อแบบนี้ได้อธิบายไว้แล้วในหัวข้อก่อนๆ สำหรับรูปที่ 9.b ถึง e แสดงการต่อที่ต่างออกไปไปจากมาตรฐาน โดยไดอะแกรมแต่ละรูปแทนคอนเน็คเตอร์ด้วยกล่องสี่เหลี่ยมผืนผ้าขอบมน และได้กำหนดไว้ด้วยว่าคอนเน็คเตอร์ตัวใดเป็น DTE หรือ DCE หมายเลขที่เขียนกำกับไว้ในคอนเน็คเตอร์แต่ละตัวแทนขาของคอนเน็คเตอร์ที่ใช้กำกับเซอร์กิตต่างๆ เซอร์กิตแต่ละตัวแทนด้วยเส้นตรงในแนวนอน พร้อมด้วยอักษรย่อของอักษรต่างๆ กำกับไว้ ลูกศรที่ปลายเส้นตรงแต่ละเส้นแสดงทิศทางการส่งสัญญาณ สำหรับเหตุผลของการเลือกใช้งานว่าควรจะเลือกใช้งานไดอะแกรมรูปใดจึงจะเหมาะสมกับระบบจะขออธิบายในหัวข้อถัดไป

- Three-wire Economy Model

จากไดอะแกรมรูปภาพในรูปที่ 9.b สายเคเบิลที่ใช้นั้นใช้เพียงสามเส้น คือใช้ที่ขา 2, 3 และ7 (ถ้าไม่ต่อขา 1 "เซอร์กิต Protective Ground" ซึ่งเป็นกรณีเลือกใช้งาน) การต่อสายเคเบิลแบบนี้ส่วนมากมักใช้ในการอินเตอร์เฟสกับระบบไมโครคอมพิวเตอร์ แต่มีบางระบบที่เราต้องต่อสายเคเบิลเพิ่มเข้าไปอีก ดังจะอธิบายต่อไป

การต่อสายเคเบิลแบบนี้ใช้ในการส่งข้อมูลแบบ full-duplex ซึ่งเป็นแบบที่ใช้เซอร์กิตน้อยที่สุด คือใช้เพียงขา 2 Transmitted Data, ขา 3 Received Data และขา 7 Signal Ground (เป็นตัวอ้างอิงของระดับแรงดันที่ขา 2 และขา 3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อผิดพลาดที่เกิดขึ้นบ่อยที่สุดในการใช้งานสายเคเบิลแบบ Three-Wire Economy model คืออุปกรณ์บางตัวของระบบไมโครคอมพิวเตอร์ใช้เซอร์กิต Request to Send และ Clear to Send ด้วย ดังนั้นอุปกรณ์เหล่านี้จะส่งข้อมูลได้ก็ต่อเมื่อมันได้รับสัญญาณ Clear to Send ก่อน ตัวอย่างของอุปกรณ์ที่ใช้งานในลักษณะนี้ได้แก่ IC พวก USART (Universal Synchronous/Asynchronous Receiver/Transmitter) ซึ่งใช้เป็นพอร์ท I/O แบบ USART ทำการส่งข้อมูล

Three-wire with Luxuary Loop-Back

จากรูปที่ 9.b และ 9.c จะเห็นได้ว่าไดอะแกรมในรูปที่ 9.c มีการแก้ไขเพิ่มเติมโดยทำ loop-back สายสัญญาณดังนี้

Request to Send ... ต่อเข้ากับ ... Clear to Send

Request to Send ... ต่อเข้ากับ ... Received Line Signal Detector

Data Terminal Ready ...ต่อเข้ากับ ... Data Set Ready

การต่อสายแบบนี้เป็นการแก้ปัญหาต่างๆ ที่กล่าวไว้ในหัวข้อก่อน อย่างไรก็ตามยังมีวิธีการอินเตอร์เฟสที่ต่างไปจากนี้ แต่การต่อสายแบบนั้นยังอาศัยหลักการต่อที่ได้อธิบายไว้ข้องต้น โดยเลือกต่อเฉพาะสายสัญญาณเพียงบางเส้นเท่านั้น (ขึ้นกับระบบของเรา) เช่น การต่อสาย Request to Send เข้ากับ Clear to Send ก็เพียงพอที่จะทำให้ระบบไมโครคอมพิวเตอร์ทำการสื่อสารข้อมูลได้

สำหรับหลักเหตุผลของการต่อแบบต่างๆ ดูได้จาก "State-transition diagram" ที่ได้อธิบายไปแล้วในหัวข้อ " การส่งข้อมูลแบบ full-duplex" สำหรับสาเหตุของการต่อ Data Terminal Ready เข้ากับ Data Set Ready นั้นเป็นการทำให้ระบบเข้าสู่ Eqoipment Readiness phase ทันทีที่ DTE ยื่อนสัญญาณ Data Terminal Ready ปกติสัญญาณนี้จะON เมื่อเราเปิดสวิทช์จ่ายไฟให้กับ DTE เมื่อ DTE ยื่อนสัญญาณ Request to Send ระบบจะเข้าสู่ Circuit Assurance phase ในทันที เนื่องจาก Request to Send ถูกต่อเข้ากับ Received Line Signal Detector (ปกติ Request to Send จะ ON เมื่อเราจ่ายไฟให้แก่ DTE) หรือจะกล่าวได้อีกทางหนึ่งก็คือเหตุการณ์ Request to Send ที่เกิดขึ้นจะเป็นตัวทริกก์(trigger) ให้เกิดเหตุการณ์ Data Carrier Detect นอกจากนี้ขา Request to Send ยังถูกต่อกับขา Clear to Send ดังนั้นเมื่อ Request to Send มีสภาวะเป็น ON จะทำให้เกิด Channel Readiness phase ขึ้นทันที ดังนั้นถ้า Data Terminal Ready และ Request to Send ยังอยู่ในสภาวะ ON เราจะเข้าสู่สภาวะ Data-Exchange stste ได้ในทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

อย่างไรก็ตามการต่อ (jump) สายเคเบิลแบบนี้ (ซึ่งช่วยให้เราส่งข้อมูลแบบ full-duplex ได้โดยแทนเหตุการณ์ด้วยการต่อสายเข้าด้วยกัน) ได้ตัดการตรวจสอบเหตุการณ์บางทิ้งไปเพื่อลดความซับซ้อนของระบบ เช่น การต่อขา Data Terminal Ready เข้ากับขา Data Set Ready โดยปกติ Data Set Ready จะ ON ได้ก็ต่อเมื่อเกิด connect sequence ขึ้นแล้ว และ (AND) Data Terminal Ready มีสภาวะเป็น ON แต่การต่อสายสองเส้นนี้เข้าด้วยกัน Data Set Ready จะ ON ตามสัญญาณ Data Terminal Ready เพียงอย่างเดียวจึงเป็นการตัดการตรวจสอบ Connect sequence ทิ้งไป สำหรับการต่อขาRequest to Send เข้ากับขา Data Carrier Detect จะตัดการตรวจสอบสัญญาณ carrier ที่ต้องส่งมาจาก DCE ทางด้านที่เราจะติดต่อด้วย (remote) ออกไป การต่อขา Request to Send เข้ากับขา Clear to Send จะตัดการทำ handshake ระหว่าง Request to Send กับ Clear to Send ทิ้ง ซึ่งอาจก่อให้เกิด overun error ขึ้น จากที่กล่าวมาขอให้สังเกตุด้วยว่า การต่อสายนี้ได้ตัดการตรวจสอบเหตุการณ์ที่เกี่ยวพันกับการเชื่อมโยงทางโทรศัพท์ออกไป ดังนั้นเวลาเลือกใช้งานควรพิจารณาให้เข้ากับระบบของเราด้วย

<u>เหตุผลของการทำ Loop-back และ Crossove</u>

ในหัวข้อนี้จะอธิบายสาเหตุของการทำ loop-back และการไขว้สายที่เซอร์กิตต่างๆ จุดประสงค์ของการไขว้สาย Transmitted Data และ Received Data คือต้องการแก้ปัญหาที่อุปกรณ์สองตัวส่งและรับข้อมูลผ่านทางเซอร์กิตเดียวกันบนสายเส้นเดียวกัน การแก้ปัญหานี้ทำได้โดยการไขว้สายให้อยู่ในลักษณะการทำงานของ null modem สำหรับการไขว้สาย Data Terminal Ready เข้ากับ Ring Indicator นั้นเมื่อใดที่ DTE เกิดเหตุการณ์ DTR ซึ่งเป็นการแสดงให้อุปกรณ์ตัวอื่นทราบว่า DTE พร้อมจะทำการแลกเปลี่ยนข้อมูลแล้ว DTE ทางด้านที่ต้องการติดต่อด้วย (remote) จะถูกกระตุ้นให้เกิดเหตุการณ์ RI ขึ้นทันที ซึ่งเป็นการต่อเพื่อหลอกให้ DTE ปลายทางเข้าใจว่ามีการเรียกเข้ามา นอกจากนี้ Ring Indicator ยังถูกต่อเข้ากับ Data Set Ready อีกด้วย ดังนั้นเมื่อใดก็ตามที่ DTE ปลายทางป้อนสัญญาณ Data Terminal Ready, DTE ปลายทางจะเข้าสู่ Equipment Readiness phase ทันที ซึ่งในกรณีของทางด้านเรียกก็เป็นเช่นเดียวกัน สำหรับการต่อ Request to Send เข้ากับ Received Line Signal Indicator นั้น การเกิดเหตุการณ์ RTS ที่ด้านเรียกหรือตอบรับต่อการเรียกจะทำให้ DTE ด้านตรงข้ามเข้าสู่ Circuit Assurance phase ในทันที สำหรับการต่อสายระหว่าง Request to Send กับ Clear to Send การเกิดเหตุการณ์ RTS จะทำให้เกิดเหตุการณ์ CTS โดยอัตโนมัติ

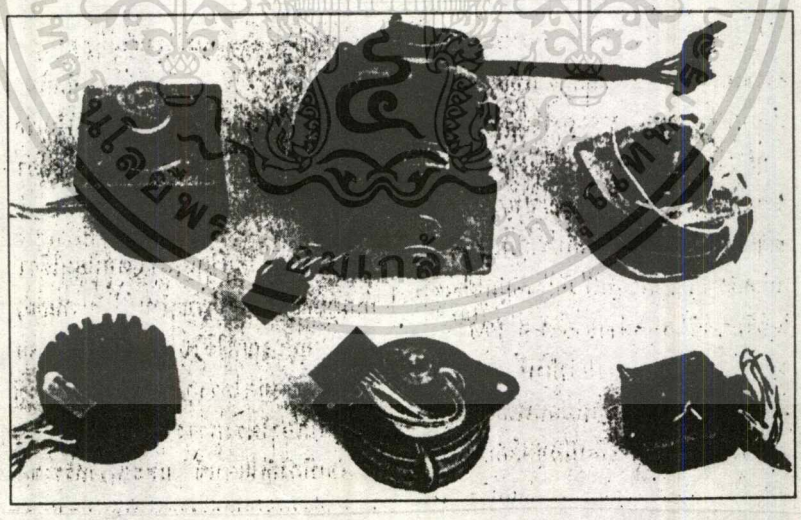
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

STEPPING MOTOR และการควบคุม

ในร่างกายของมนุษย์ ตัวเปลี่ยนพลังงานเคมีมาเป็นพลังงานกลก็คือ กล้ามเนื้อ แต่ถ้าเป็นในหุ่นยนต์ สเต็ปปิ้งมอเตอร์ คือตัวเปลี่ยนพลังงานไฟฟ้ามาเป็นพลังงานกลการที่หุ่นยนต์สามารถเคลื่อนไหวได้ตามการควบคุมก็เพราะอาศัยพลังงานขับเคลื่อนจากแหล่งพลังงานต่างๆ เช่น จากระบบไฮโดรลิกส์ ระบบนิวเมติก และระบบไฟฟ้า

ระบบไฟฟ้าจะเป็นที่นิยมมากเพราะค่าใช้จ่ายถูก ง่ายต่อการบำรุงรักษาและสามารถควบคุมตำแหน่งและความเร็วได้เที่ยงตรง โดยจะมีมอเตอร์เป็นตัวขับเคลื่อนให้แขนกลของหุ่นยนต์สามารถเคลื่อนไหวได้ตามการควบคุม จึงมีผู้นิยมหันมาใช้สเต็ปปิ้งมอเตอร์แทน เพราะมีคุณสมบัติที่ดีหลายอย่างคือ



รูปที่ 1 แสดงรูปร่างหน้าตาของสเตปเปอร์มอเตอร์ชนิดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไม่จำเป็นต้องมีสัญญาณป้อนกลับเพื่อควบคุมตำแหน่งและความเร็วนั่นคือ สามารถใช้การควบคุมแบบระบบเปิดได้ ลดความยุ่งยากทางวงจรควบคุมลง แต่ยังให้การควบคุมตำแหน่งและความเร็วที่เที่ยงตรงได้เช่นกัน

- สเต็ปปิ้งมอเตอร์เป็นอุปกรณ์ที่เปลี่ยนสัญญาณดิจิตอลไปเป็นการเคลื่อนที่ทางกล ดังนั้นการติดต่อกับอุปกรณ์ดิจิตอลก็เป็นไปได้โดยง่าย และวงจรขยายกำลังจากสัญญาณดิจิตอล (digital power amplifier) ที่ใช้ก็มีราคาถูกกว่าวงจรขยายกำลังเชิงเส้น (linear power amplifier) อีกด้วย

- การออกแบบวงจรควบคุมสเต็ปปิ้งมอเตอร์สามารถทำได้ง่ายกว่าวงจรควบคุมมอเตอร์แบบเซอร์โว และยังสามารถออกแบบวงจรให้สเต็ปปิ้งมอเตอร์ทำงานหรือหยุดได้แบบทันทีทันใด

ชนิดและการทำงานของสเต็ปปิ้งมอเตอร์

สเต็ปปิ้งมอเตอร์แบ่งออกได้หลายชนิดด้วยกัน เช่น Variable-reluctance, permanant magnet, hybrid, linear ฯลฯ แต่ในที่นี้จะกล่าวถึงเพียง 2 ชนิดเท่านั้นที่ใช้งานกันบ่อยๆ คือ แบบ Variable-reluctance และ permanant magnet

Variable-reluctance หรือเรียกสั้นๆ ว่า VR มอเตอร์ จะเป็นพื้นฐานสำคัญในการทำงานของสเต็ปปิ้งมอเตอร์ ซึ่งจะช่วยให้เข้าใจการทำงานของสเต็ปปิ้งมอเตอร์ชนิดอื่นๆ ได้ง่ายยิ่งขึ้น รูปที่ 6.1เป็นพื้นที่หน้าตัดและยังแสดงถึงการพันขดลวดของ VR มอเตอร์แบบ 3 เฟส มีขั้วเหนือและใต้อยู่ตรงข้ามกัน 3 คู่ โดยจะพันขดลวดแบบอนุกรมกันในแต่ละชุด ถ้ามีการกระตุ้นเฟสเกิดขึ้น ขั้ว I', I'', I''' จะเป็นขั้วใต้ และขั้ว I, II, III จะเป็นขั้วเหนือ ทั้งโรเตอร์และสเตเตอร์จะทำจากเหล็กผสมชนิดซิลิคอนซึ่งเป็นวัสดุที่มีค่าความซึมซับ (permeability) สูง สามารถให้เส้นแรงแม่เหล็กไหลผ่านได้มาก

การทำงานจะเริ่มจากการกระตุ้นที่เฟส I ก่อน (S_1 "ON") ซึ่งจะทำให้เส้นแรงแม่เหล็กเกิดขึ้นดังรูปที่ 6.2 ตัวโรเตอร์จะพยายามวางตำแหน่งตัวเองให้อยู่ในทิศทางที่ทำให้เกิดค่าความต้านทานแม่เหล็กน้อยที่สุด ในแนวขั้ว I-I'

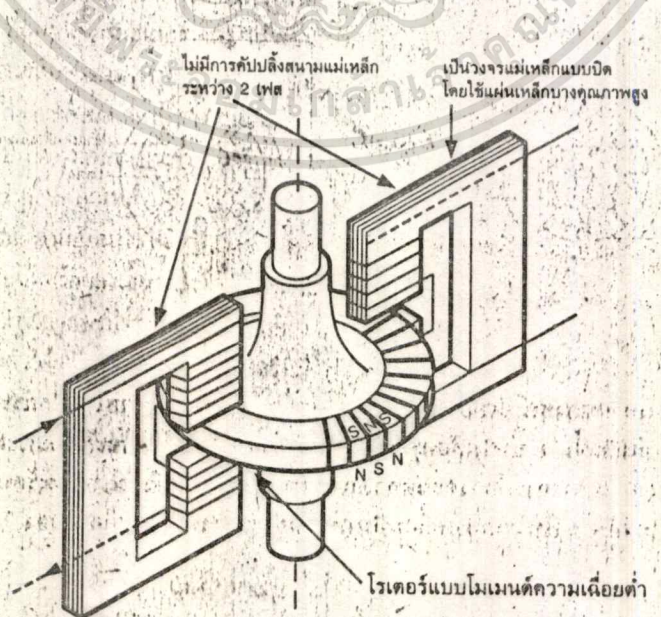
ในขณะเริ่มต้นที่จะกระตุ้นที่เฟส II (S_1 "OFF", S_2 "ON") ดังรูปที่ 6.3 เส้นแรงแม่เหล็กจะไม่อยู่ในแนวทางเดินที่สะดวก จึงทำให้ค่าความต้านทานแม่เหล็กมีค่าสูง ตัวโรเตอร์ก็จะพยายามปรับตัวเองเพื่อให้ค่าความต้านทานแม่เหล็กน้อยที่สุด ด้วยการหมุนในทิศทางทวนเข็มนาฬิกา ซึ่งแรงบิดที่ใช้หมุนเกิดจากแรงของเส้นแรงแม่เหล็ก แล้วจะไปหยุดที่ค่าความต้านทานแม่เหล็กน้อยที่สุด นั่นคือ จะหมุนไป 1 สเต็ป หรือ 30 องศานั่นเอง ความสัมพันธ์ระหว่างจำนวนสเต็ปของการหมุนโรเตอร์ไป 1 รอบ (S) มุมที่เปลี่ยนไปในหนึ่งสเต็ป (Q_s) จำนวนเฟสของสเตเตอร์ (m) และจำนวนฟันของโรเตอร์ (N_r) แสดงได้ดังสมการที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$S = 360/Q_s = mN_r$ ----------(1)

ตัวอย่างเช่น สเต็ปปิ้งมอเตอร์ตัวหนึ่งมี m = 3, N_r = 4 ก็จะได้ S = 3 x 4 = 12 สเต็ป และมุมในการหมุน Q_s =360/12 = 30 องศา ซึ่งจากสมการ (1) ยังทำให้ทราบอีกว่าถ้าจะลดค่า Q_s ให้น้อยลง อาจทำได้โดยการเพิ่มค่าของ m หรือและ N_r ให้สูงขึ้นและลดช่องว่างระหว่างโรเตอร์และสเตเตอร์ให้มีค่าน้อยๆ เพื่อให้เกิดแรงบิดสูงสุด และยังมีผลต่อความเที่ยงตรงของตำแหน่งมากยิ่งขึ้นด้วย

สำหรับสเต็ปปิ้งมอเตอร์ชนิด permanant magnet หรือเรียกสั้นๆ ว่า PM มอเตอร์จะมีข้อแตกต่างที่สำคัญจาก VR มอเตอร์ก็คือ โรเตอร์จะเป็นแม่เหล็กถาวร จึงทำให้การพันขดลวดที่สเตเตอร์ต้องแตกต่างไปด้วย ดังแสดงในรูปที่ 6.4 (ก) จะเห็นว่าสเตเตอร์ในแต่ละขั้วจะมีขดลวดพันอยู่ ซึ่งถือว่าแต่ละขั้วคือ 1 เฟส ดังนั้นจากรูปจึงมีทั้งหมด 4 เฟสด้วยกัน การต่อวงจรกระตุ้นเฟสจะเห็นว่าปลายขดลวด (C) ของทุกเฟสจะต่อร่วมกันถึงขั้วบวกของแหล่งจ่ายไฟดังนั้นเมื่อเกิดการกระตุ้นที่เฟสใดแล้วขั้วสเตเตอร์ที่เฟสนั้นก็จะกลายเป็นขั้วเหนือ รูปที่ 6.5 จะเป็นการแสดงตำแหน่งของโรเตอร์ในแต่ละสเต็ป หลังจากถูกกระตุ้นที่เฟส 1-2-3-4 ตามลำดับ และจะหมุนไปในทิศทางตามเข็มนาฬิกา ทุก 90 องศาต่อสเต็ป ถ้าต้องการจะให้มุมในละเอียดมากขึ้นจะต้องเพิ่มจำนวนเฟสของสเตเตอร์และจำนวนขั้วแม่เหล็กของโรเตอร์ให้มากขึ้น ข้อเสียของ PM มอเตอร์ก็คือ มีราคาแพง และความหนาแน่นของเส้นแรงแม่เหล็กจะถูกจำกัดโดยเส้นแรงแม่เหล็กภายใน (magnetic remanence) ของแม่เหล็กถาวรทำให้ไม่สามารถผลิตแรงบิดได้มาก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2 แสดงโครงสร้างภายในพื้นฐานของชนิดแรเอิร์ธเพอร์มาเนนต์แมกเน็ต สเต็ปเปอร์มอเตอร์

โหมดการทำงานของสเต็ปปิ้งมอเตอร์

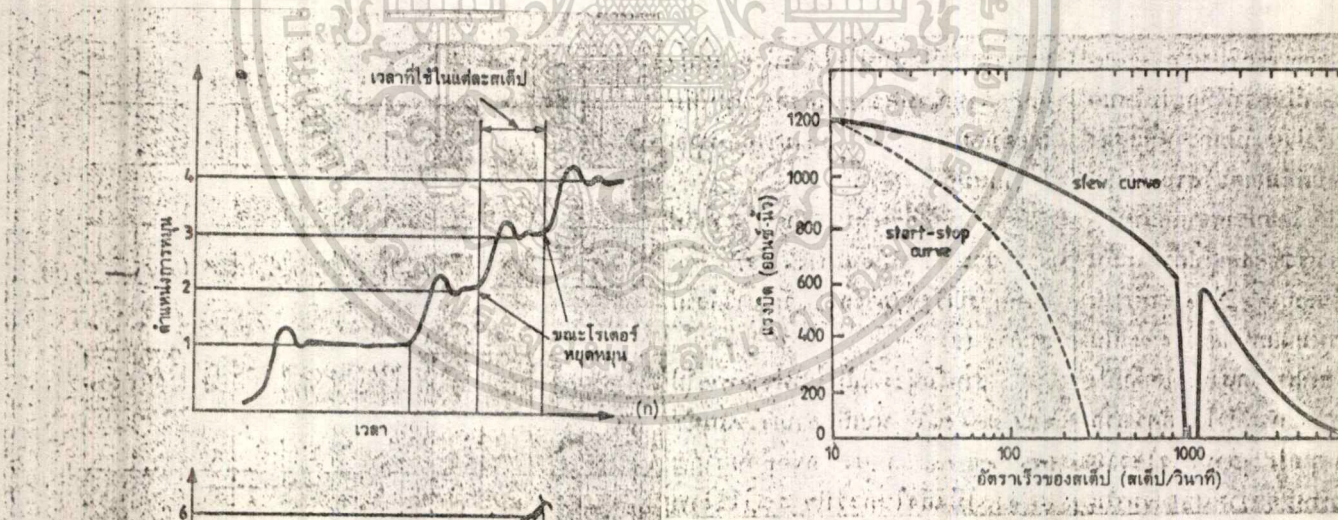
ถ้าจะแบ่งโหมดการทำงานของสเต็ปปิ้งมอเตอร์ตามอัตราเร็วของสเต็ปแต่ละสเต็ป จะแบ่งออกได้เป็น 2 โหมดคือ หมุนเป็นสเต็ป (descrete stepping mode) และหมุนแบบต่อเนื่อง (slewing mode)

โดยถ้าการหมุนเป็นแบบสเต็ปและมีการหยุดนิ่งก่นที่จะเปลี่ยนเป็นสเต็ปถัดไป ก็จะเรียกการทำงานในโหมดนี้ว่าการหมุนเป็นสเต็ป ดังแสดงในรูปที่ 6.6 (ก) สำหรับตัวอย่างของเครื่องใช้ที่ทำงานในโหมดนี้คือ เครื่องเจาะบัตร การทำงานคร่าวๆ ก็คือ สเต็ปปิ้งมอเตอร์จะเป็นตัวส่งแถบกระดาษเข้าไปยังเครื่องปรุกระดาษ เพื่อบันทึกข้อมูลลงในแถบกระดาษ ซึ่งการหมุนของสเต็ปปิ้งมอเตอร์จะหมุนไปแล้วหยุดชั่วขณะ เพื่อปรุกระดาษให้เรียบร้อยก่อน แล้วจึงค่อยหมุนต่อไปยังตำแหน่งเจาะใหม่

ถ้าเพิ่มอัตราเร็วของในแต่ละสเต็ปให้เร็วขึ้น และเป็นไปอย่างต่อเนื่องไม่มีการหยุดนิ่งจะเรียกกำหนดการทำงานนี้ว่า การหมุนแบบต่อเนื่อง ดังแสดงในรูปที่ 6.6 (ข) ซึ่งจะสามารถหาความสัมพันธ์ของความเร็วรอบของมอเตอร์ (n) กับอัตราเร็วของสเต็ป (f) และจำนวนสเต็ปทั้งหมด (s) ได้ดังสมการ

$$n = 60\ f/s \qquad \text{----------(2)}$$

กราฟคุณลักษณะของสเต็ปปิ้งมอเตอร์



รูปที่ 7 กราฟแสดงความสัมพันธ์ระหว่างอัตราเร็วของสเต็ปกับแรงบิดของทั้ง 2 โหมดของสเต็ปปิ้งมอเตอร์

รูปที่ 6 กราฟแสดงสเต็ปของการหมุนในโหมดการทำงานแบบ
(ก) หมุนเป็นสเต็ป
(ข) หมุนต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟคุณลักษณะของสเต็ปปิ้งมอเตอร์จะเป็นการแสดงความสัมพันธ์ระหว่างอัตราเร็วของสเต็ปกับแรงบิดดังแสดงในรูปที่ 6.7 สำหรับกราฟเส้นประเรียกว่า start-stop curve หรือ single-step load curve เป็นกราฟที่อยู่ในโหมดการหมุนเป็นสเต็ป และเป็นกราฟที่แสดงย่านของแรงบิดที่มอเตอร์สามารถเริ่มและหยุดหมุนได้ โดยปราศจากความผิดพลาดแม้ที่อัตราเร็วต่างๆ กัน และกราฟอีกเส้นคือ slew curve ซึ่งทำงานอยู่ในโหมดการหมุนแบบต่อเนื่อง จะเป็นกราฟที่แสดงถึงค่าแรงบิดสูงสุดที่สเต็ปปิ้งมอเตอร์สามารถจะกระทำได้ที่อัตราเร็วของสเต็ปต่างๆ กัน ถ้ามีการใช้งานของสเต็ปปิ้งมอเตอร์เหนือกราฟนี้ก็อาจทำให้เกิดความผิดพลาดได้ ในทางตรงกันข้ามถ้าใช้งานอยู่ภายใต้กราฟนี้แม้จะควบคุมแบบระบบเปิดก็มั่นใจได้ว่าทั้งตำแหน่งและความเร็วมีความเที่ยงตรงแน่นอน โดยตำแหน่งของมอเตอร์สามารถคำนวณได้จากสมการ

มุมที่เปลี่ยนไป = มุมใน 1 สเต็ป x จำนวนพัลซ์ที่ป้อนให้ --------(3)

ส่วนความเร็วสามารถคำนวณหาได้จากสมการ (2)

สำหรับช่วงที่เส้นกราฟขาดหายไปของ slew curve นั้นเป็นย่านเรโซแนนซ์ (resonance region) ซึ่งเป็นจุดอ่อนสำคัญที่ต้องคำนึงถึงในการพิจารณาใช้งานเพราะช่วงนี้เป็นช่วงช่วงที่ไม่เสถียรและควบคุมไม่ได้

- วิธีการกระตุ้นเฟส

การที่จะทำให้สเต็ปปิ้งมอเตอร์หมุนได้อย่างต่อเนื่องเหมือนกับการหมุนของมอเตอร์ไฟฟ้ากระแสตรงนั้นจะต้องมีการจ่ายพัลซ์เป็นลำดับอย่างต่อเนื่อง วิธีการที่จะกระตุ้นเฟสมีด้วยด้วยกันหลายวิธี แต่จะอธิบายเพียง 3 วิธีเท่านั้น เริ่มจากแบบแรกคือ การกระตุ้นเฟสแบบเดี่ยว (single-phase excitation) เป็นการกระตุ้นเฟสเพียงเฟสเดียวเท่านั้นที่จังหวะสัญญาณนาฬิกาหนึ่งๆ แบบที่สองคือ การกระตุ้นแบบเฟสคู่ (two-phase excitation) ก็จะมีการกระตุ้นเฟสสองเฟสพร้อมกันในจังหวะสะญญาณนาฬิกาหนึ่งๆ สำหรับแบบสุดท้ายเป็นการกระตุ้นแบบกึ่งสเต็ป (half-step excitation) จะเป็นการรวมเอา 2 แบบแรกเข้าด้วยกัน โดยจะกระตุ้นเฟสแบบที่ 1 และแบบที่ 2 สลับกันไป การกระตุ้นเฟสของทั้ง 3 แบบดูได้จากตารางในรูปที่ 6.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการกระตุ้นเฟสแบบคู่จะมีทิศทางของเส้นแรงแม่เหล็กไม่เป็นเส้นตรงเหมือนกับแบบกระตุ้นแบบเฟสเดี่ยว ดังแสดงในรูปที่ 6.9 (ก) แต่ถึงกระนั้นค่ามุมที่เปลี่ยนไปในหนึ่งสเต็ปก็ยังคงมีค่าเท่าเดิมเหมือนกับแบบกระตุ้นแบบเฟสเดี่ยว สำหรับการกระตุ้นแบบกึ่งสเต็ป ค่ามุมที่เปลี่ยนไปใน 1 สเต็ปจะมีค่าลดลงครึ่งหนึ่ง การกระตุ้นแบบเฟสคู่จะเกิดแรงบิดได้มากกว่าการกระตุ้นแบบเหสเดี่ยว ขณะเดียวกันยังสามารถเข้าตำแหน่งในแต่ละสเต็ปได้เร็วกว่าด้วยดังแสดงในรูปที่ 6.9 (ข)

ข้อดีอย่างหนึ่งสำหรับการกระตุ้นเฟสแบบกึ่งสเต็ปคือ สามารถลดผลกระทบเนื่องจากย่านเรโซแนนซ์ได้ แต่ที่ความถี่ต่ำๆ จะมีค่าแรงบิดลดลง ดังแสดงในรูปที่ 6.9 (ค) และสำหรับทิศทางการหมุนของมอเตอร์ขึ้นอยู่กับทิศทางลำดับเฟสที่ถูกกระตุ้นด้วย

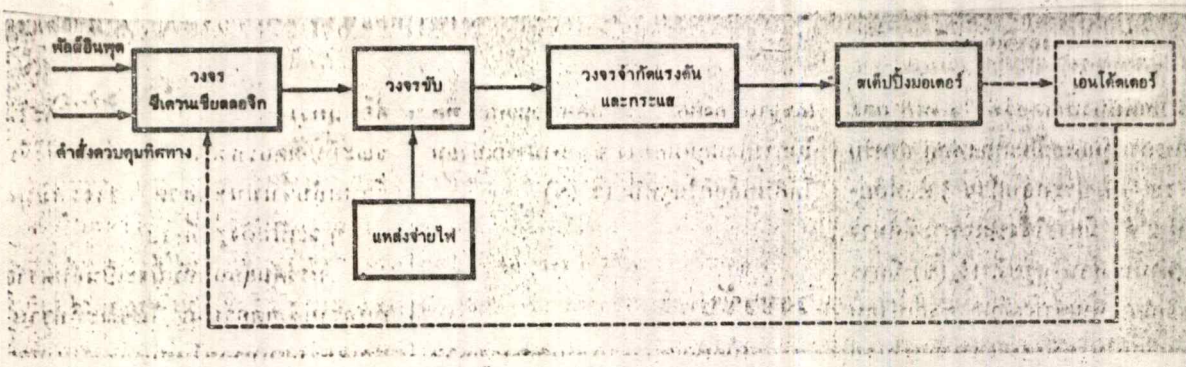
การควบคุมสเต็ปปิ้งมอเตอร์

บล็อคไดอะแกรมสำหรับการควบคุมสเต็ปปิ้งมอเตอร์ดังแสดงในรูปที่ 6.10 เป็นวงจรซีเควนเชียลลอจิกจะรับพัลซ์อินพุตและคำสั่งควบคุมทิศทาง แล้วจึงผลิตพัลซ์ที่ใช้ในการกระตุ้นเฟสของมอเตอร์ออกไป แต่จะมีระดับของสัญญาณต่ำจึงต้องนำสัญญาณนี้ไปผ่านวงจรขับเพื่อให้ระดับของสัญญาณสูงขึ้น และจะมีวงจรจำกัดกระแสและแรงดันทำหน้าที่เป็นบัฟเฟอร์ให้กับวงจรขับและมอเตอร์จากบล็อคไดอะแกรมจะเห็นได้ว่าไม่จำเป็นต้องต่อวงจรเอ็นโคดเดอร์ป้อนสัญญาณกลับเพื่อควบคุมตำแหน่งและความเร็ว แต่ในบางครั้งเช่นในกรณีที่ความสัมพันธ์ของการเคลื่อนที่ระหว่างแกนหมุนของมอเตอร์กับตำแหน่งของโหลดไม่ตรงกัน อันเกิดจากความผิดพลาดของเฟืองเกียร์ หรือในกรณีที่โหลดบางอย่างทำให้เกิดการผิดพลาดของสเต็ปขึ้นก็จำเป็นที่จะต้องมีการใช้สัญญาณป้อนกลับเพื่อควบคุมตำแหน่งและความเร็วให้ถูกต้อง

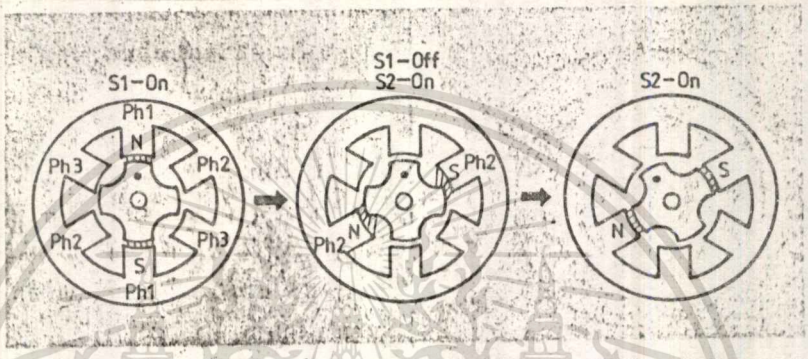
วงจรซีเควนเชียลลอจิก

การออกแบบวงจรซีเควนเชียลลอจิกนั้น ความแตกต่างของวงจรขึ้นอยู่กับว่าผู้ออกแบบจะเลือกใช้ไอซีประเภทใดมาใช้ในวงจร วิธีการกระตุ้นเฟสจำนวนเฟสของสเต็ปปิ้งมอเตอร์และยังรวมถึงจุดประสงของการใช้สเต็ปปิ้งมอเตอร์ว่าต้องการใช้งานเฉพาะทิศทางเดียวหรือสองทิศทาง ซึ่งในการใช้งานแบบสองทิศทางก็ยังสามารถแบ่งออกได้เป็น 2 แบบ ดังรูปที่ 6.11 จะเห็นว่าแบบแรกนั้นรับพัลซ์อินพุตเข้าทางหนึ่งและรับคำสั่งทิศทางอีกทางหนึ่ง สำหรับในแบบที่สองนั้นก็มีอินพุตอยู่สองทางเช่นกัน แต่จะรับพัลซ์อินพุตทั้งสองทางไม่พร้อมกัน ขึ้นอยู่กับว่าต้องการใช้งานในทิศทางไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



บล็อกไดอะแกรมของระบบควบคุมสเต็ปปิ้งมอเตอร์



แสดงขั้นตอนการหมุนเมื่อมีการกระตุ้นเฟสจาก เฟส 1 ไปยัง เฟส 2

(ก)

จังหวะสัญญาณนาฬิกา	R	1	2	3	4	5	6	7	8	9	10
เฟส 1											
เฟส 2											
เฟส 3											

(ข)

จังหวะสัญญาณนาฬิกา	R	1	2	3	4	5	6	7	8	9	10
เฟส 1											
เฟส 2											
เฟส 3											

(ค)

จังหวะสัญญาณนาฬิกา	R	1	2	3	4	5	6	7	8	9	10
เฟส 1											
เฟส 2											
เฟส 3											

8 ตารางแสดงการกระตุ้นเฟส

(ก) แบบเฟสเดี่ยว (single-phase) (ข) แบบเฟสคู่ (two-phase)

(ค) แบบกึ่งสเต็ป (half-phase)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

พื้นฐานของแขนกล

ในโรงงานอุตสาหกรรมขนาดใหญ่ที่ใช้ระบบการผลิตแบบอัตโนมัติ บางส่วนมีกาการนำเอาหุ่นยนต์เข้ามาช่วยในการทำงานซึ่งก็มีหลายรูปแบบด้วยกัน รูปแบบหนึ่งของหุ่นยนต์ที่มีใช้กันอยู่ คือแขนหุ่นยนต์หรือแขนกล

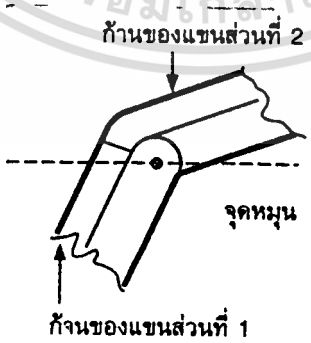
ความเป็นมา

ในปี ค.ศ. 1905 บริษัท ฟอร์ดมอเตอร์ จำกัด ซึ่งเป็นบริษัทผลิตชิ้นส่วนต่างๆ ของรถยนต์ได้นำระบบสายงานการผลิตเข้ามาใช้ในโรงงานเป็นบริษัทแรก เมื่อปริมาณการผลิตมีมากขึ้น ความรวดเร็วและแม่นยำในการทำงานจึงเป็นสิ่งที่ต้องการ จึงได้คิดค้นออกแบบเครื่องจักรช่วยในการผลิต โดยในยุคแรกๆเป็นเครื่องจักรที่ประกอบด้วยส่วนขับเคลื่อนที่ใช้ไฟฟ้าเรียกว่าเครื่องจักรอัตโนมัติ

เมื่อปริมาณและความซับซ้อนในการทำงานมีมากขึ้น ทำให้เครื่องจักรอัตโนมัติถูกพัฒนาให้มีความเหมาะสมกับงาน สามารถควบคุมได้ง่าย เรียกว่า Programmable mechanical manipulator ต่อมามีการใช้คอมพิวเตอร์เข้ามาควบคุมการทำงานต่างๆ เครื่องจักรอัตโนมัติจึงมีความยืดหยุ่นในการทำงานมากขึ้น เรียกว่าหุ่นยนต์ (robot) สำหรับคำว่าแขนกลหรือแขนหุ่นยนต์ (robot arm) ก็เป็นชิ้นส่วนหนึ่งของหุ่นยนต์คล้ายกับแขนมนุษย์ที่เป็นชิ้นส่วนหนึ่งของร่างกายมนุษย์เช่นกัน

การจัดประเภทของแขนกล

การจัดประเภทของแขนกลตามมาตรฐานโดยทั่วไป สามารถแบ่งเป็น 3 ลักษณะคือ



รูปที่ 1 การเคลื่อนที่ของแขนกลในแบบ REVOLUTTE

1. เทคโนโลยีการขับเคลื่อน

การจัดประเภทของแขนกลตามเทคโนโลยีการขับเคลื่อนข้อต่อต่างๆ ที่นิยมใช้กันอยู่มี 2 ระบบ คือระบบไฟฟ้าและระบบไฮโดรลิก สำหรับการขับเคลื่อนด้วยระบบไฟฟ้าที่ใช้กันอยู่จะเป็นดีซีเซอร์โวมอเตอร์ และดีซีสเตปเปอร์มอเตอร์ ซึ่งส่วนใหญ่จะใช้กับงานที่รับน้ำหนักไม่มากนัก

เอกสารนี้เ... ไม่ว่ากรณีใ...

ส่วนในกรณีที่แขนกลต้องทำงานที่รับน้ำหนักมากๆ เช่น งานจับและประกอบชิ้นส่วนในโรงงานประกอบและผลิตรถยนต์ ระบบไฮโดรลิกจะถูกนำมาใช้ในการขับเคลื่อน แต่กลไกแบบนี้ก็มีข้อเสียในเรื่องการรักษาความสะอาด เพราะไฮโดรลิกใช้น้ำมันในการขับดัน นอกจากการขับเคลื่อน 2 ระบบนี้ ปัจจุบันยังมีการขับเคลื่อนอีกระบบหนึ่ง ซึ่งใช้ลมในการขับดัน เรียกว่าระบบนิวเมติก ซึ่งแขนกลในปัจจุบันได้มีการนำเอาระบบขับเคลื่อนต่างๆ มาผสมผสานกัน เพื่อให้แขนกลทำงานได้ตามความต้องการ

2. ลักษณะการเคลื่อนที่ของข้อต่อ

เนื่องจากมือของแขนกลจะอยู่ที่ส่วนปลายของแขน ดังนั้นในการเคลื่อนที่เพื่อให้มือไปทำงานต่างๆ ข้อต่อต่างๆ ก็เคลื่อนที่ไปด้วย โดยลักษณะการเคลื่อนที่ของข้อต่อแบ่งออกเป็น 2 ลักษณะคือ

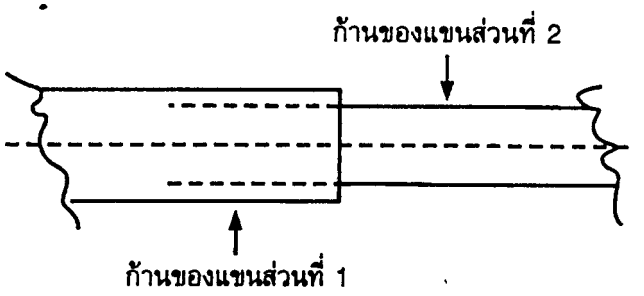
2.1 การเคลื่อนที่แบบ REVOLUTE สัญลักษณ์ที่ใช้คือ R เป็นการเคลื่อนที่ข้อต่อแบบวงกลมรอบจุดแกนหมุนดังแสดงดังในรูปที่ 1

2.1 การเคลื่อนที่แบบ PRISMATIC สัญลักษณ์ที่ใช้คือ P เป็นการเคลื่อนที่ของข้อต่อที่สไลด์ไปในทิศทางเดียวกับแกนคล้ายกับลูกสูบในกระบอกสูบของเครื่องยนต์ แสดงในรูปที่ 2 ในการทำงานจริงอาจจำเป็นต้องนำรูปแบบการเคลื่อนที่ทั้ง 2 ลักษณะมารวมกัน ขึ้นอยู่กับสภาพแวดล้อมในการทำงาน โดยส่วนใหญ่แขนกลพื้นฐานที่ใช้งานอยู่ต้องประกอบด้วยแกนอย่างน้อย 3 แกน ดังแสดงในตารางที่ 1

ตารางที่ 1ชนิดแขนกลและการเคลื่อนที่ในแต่ละแกน

ชนิดแขนกล	แกนที่ 1	แกนที่ 2	แกนที่ 3
CARTESIAN	P	P	P
CYLINDRICAL	R	P	P
SPHERICAL	R	R	P
SCARA	R	R	P
ARTICULATED	R	R	R

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 การเคลื่อนที่ของแขนกลในแบบ PRISMATIC

<u>คำศัพท์ที่ควรรู้เบื้องต้นเกี่ยวกับแขนหุ่นยนต์</u>

REVOLUTE คือการเคลื่อนที่ของข้อต่อแขนกลเป็นวงกลมรอบจุดหมุน

PRISMATIC คือการเคลื่อนที่ของข้อต่อที่มีทิศทางการทำงานเลื่อนเข้าออกในทิศทางเดียวกับแกน

CARTESIAN คือลักษณะขอบเขตการเคลื่อนที่ของแขนกลเป็นรูปสี่เหลี่ยม

CYLINDRICAL คือลักษณะขอบเขตการเคลื่อนที่ของแขนกลเป็นรูปทรงกระบอก

SPHERICAL คือลักษณะขอบเขตการเคลื่อนที่ของแขนกลเป็นรูปทรงกลม

SCARA คือลักษณะขอบเขตการเคลื่อนที่ของแขนกลเป็นรูปทรงกระบอกคล้ายกับแบบ CYLINDRICAL

ARTICULATED คือลักษณะขอบเขตการเคลื่อนที่ของแขนกลเป็นรูปครึ่งวงกลม

REACH คือระยะความยาวของรัศมีจากจุดศูนย์กลางเป็นแกนหลักของแขนกลไปยังส่วนปลายของมือ

STROKE คือระยะความยาวของมือของแขนกลสามารถเอื้อมถึงได้

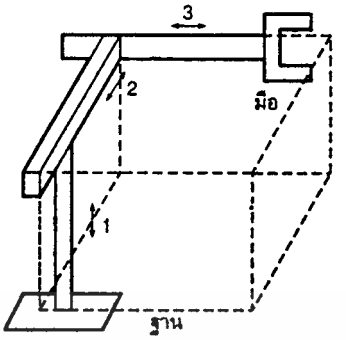
PITCH คือมุมในการหักข้อมือขึ้นลง

ROLL คือมุมในการหมุนข้อมือโดยรอบ

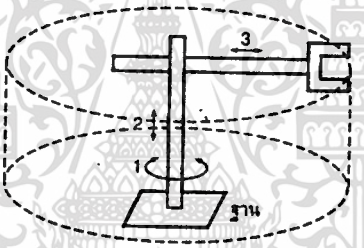
YAW คือมุมในการหักข้อมือไปทางซ้ายหรือขวา

TOLL คือมือจับที่อยู่ปลายสุดของแขนกล

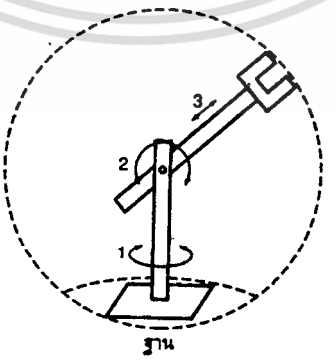
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 แขนกลชนิด CARTESIAN มีรูปขอบเขตการทำงานเป็นรูปทรงสี่เหลี่ยม



รูปที่ 4 แขนกลชนิด CYLINDRICAL มีรูปขอบเขตการทำงานเป็นรูปทรงกระบอก



รูปที่ 5 แขนกลชนิด SPHERICAL ขอบเขตการทำงานเป็นรูปทรงกลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

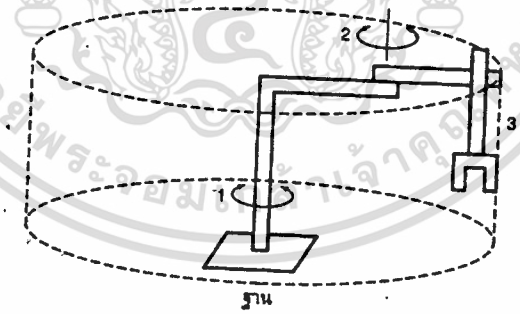
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 1 แขนกลชนิด CARTESIAN มีลักษณะการเคลื่อนที่ของข้อต่อเป็นแบบ PRISMATIC โดยจะเคลื่อนที่สไลด์ไปบนแกนซึ่งมีลักษณะตั้งฉากซึ่งกันและกันดังรูปที่ 3

ดังนั้น ขอบเขตการเคลื่อนที่ในการทำงานของแขนกลชนิด CARTESIAN จะเป็นรูปทรงสี่เหลี่ยม จากแกนที่ 1 ของแขนกลชนิด CARTESIAN ถ้าเปลี่ยนลักษณะการเคลื่อนที่จากแบบ P เป็นแบบ R จะได้แขนกลลักษณะใหม่คือแขนกลชนิด CYLINDRICAL แสดงดังรูปที่ 4

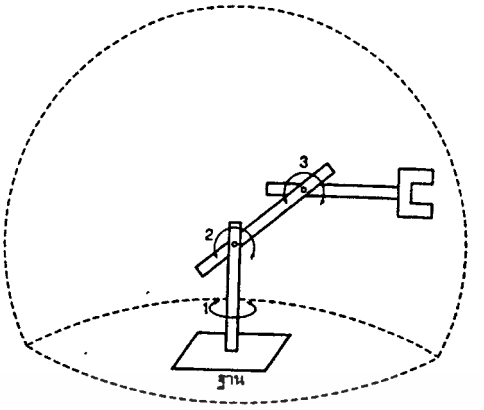
เมื่อเปลี่ยนลักษณะการเคลื่อนที่ของแกนที่ 1 ให้เป็นแบบ R ทำให้แขนกลชนิด CYLINDRICAL สามารถหมุนกลับมาทำงานที่อยู่ด้านหลังได้ ลักษณะของขอบเขตในการเคลื่อนที่ในการทำงานจึงเป็นรูปทรงกระบอก เมื่อเปลี่ยนลักษณะการเคลื่อนที่ในแนวที่ 2 ของแขนกลชนิด CYLINDRICAL จากแบบ P เป็นแบบ R จะได้แขนกลอีกลักษณะหนึ่ง ซึ่งมีขอบเขตการเคลื่อนที่ในการทำงานเป็นแบบครึ่งทรงกลม แขนกลลักษณะนี้เรียกว่า แขนกลชนิด SPHERICAL แสดงดังรูปที่ 5

มีแขนกลอีกลักษณะหนึ่งมีลักษณะการเคลื่อนที่ของข้อต่อคล้ายกับแขนกลชนิด SPHERICAL โดยมีแกนที่ 1 และ 2 เป็นการเคลื่อนที่แบบ R ส่วนแกนที่ 3 เป็นแบบ P แต่ลักษณะการเคลื่อนที่ระหว่างแกรที่ 1 กับแกนที่ 2 จะไม่ตั้งฉากซึ่งกันและกันเกมือนกับแขนกลชนิด SPHERICAL แต่จะขนานกัน แขนกลลักษณะนี้เรียกว่า แขนกลชนิด SCARA แสดงดงรูปที่ 6 แขนกลชนิด SCARA นี้มีขอบเขตการเคลื่อนที่ในการทำงานเป็นรูปทรงกระบอกคล้ายกับแขนกลชนิด CYLINDRICAL



รูปที่ 6 แขนกลชนิด SCARA ขอบเขตการทำงานรูปทรงกระบอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 แขนกลชนิด ARTICULATED ขอบเขตการทำงานเป็นรูปครึ่งทรงกลม

ตารางที่ 2 คุณสมบัติจำเพาะพื้นฐานของแขนกลแต่ละชนิด

รายการ	หน่วย
จำนวนแกน	-
ความสามารถในการรับภาระ	กิโลกรัม (kg)
ความเร็วสูงสุด	มิลลิเมตรต่อวินาที (mm/s)
ระยะ REACH และ STROKE	มิลลิเมตร (mm)
มุมในการหมุนมือ	องศา (deg)
ความสามารถในการทำซ้ำ	มิลลิเมตร (mm)
ความแม่นยำและความเที่ยงตรง	มิลลิเมตร (mm)

แขนกลชนิด SPHERICAL ถ้าทำการเปลี่ยนแกนที่ 3 ซึ่งมีลักษณะการเคลื่อนที่แบบ P ให้เป็นแบบ R แขนกลลักษณะนี้เรียกว่าแขนกลชนิด ARTICULATED แสดงดังรูปที่ 7 ซึ่งขอบเขตการเคลื่อนที่ในการทำงานจะเป็นรูปครึ่งทรงกลม

3. วิธีการควบคุมการเคลื่อนไหว

หลักการในการจัดประเภทของแขนกลอีกลักษณะหนึ่งคือ แบ่งตามวิธีการควบคุมการเคลื่อนไหวของแขนกล ซึ่งมี 2 ลักษณะด้วยกัน ลักษณะแรกเป็นการเคลื่อนไหวระหว่างจุดหรือ point to point หมายถึงการควบคุมโดยการกำหนดจุดปลายมือ ต้นทางและปลายทาง ภายในขอบเขตการเคลื่อนที่ในการทำงาน ซึ่งแขนกลจะเคลื่อนที่ไปตามจุดที่ได้กำหนดเท่านั้น วิธีการควบคุมลักษณะนี้ส่วนใหญ่ใช้ในโรงงานอุตสาหกรรม เช่น การเชื่อมแบบ spot การเคลื่อนย้ายวัตถุไปยังตำแหน่งต่างๆ (pick and place)

ส่วนลักษณะที่ 2 เป็นการควบคุมการเคลื่อนไหวแบบต่อเนื่อง (continuous path) หมายถึงการควบคุมให้ปลายมือเคลื่อนที่ไปตามเส้นทางที่ได้กำหนดอย่างต่อเนื่อง การควบคุมลักษณะนี้จะทำให้แขนกลสามารถเคลื่อนไหวได้ซับซ้อนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>คุณสมบัติจำเพาะของแขนกล</u>

ในการเลือกแขนกลใช้งาน สิ่งที่ควรคำนึงถึงอีกประการหนึ่งก็คือคุณสมบัติจำเพาะพื้นฐานของแขนกลที่จำเป็นต้องมี ซึ่งแสดงดังตารางที่ 2

จำนวนแกน คือจำนวนข้อต่อที่มีการเคลื่อนที่แบบ P หรือ R ดังที่กล่าวมาข้างต้น

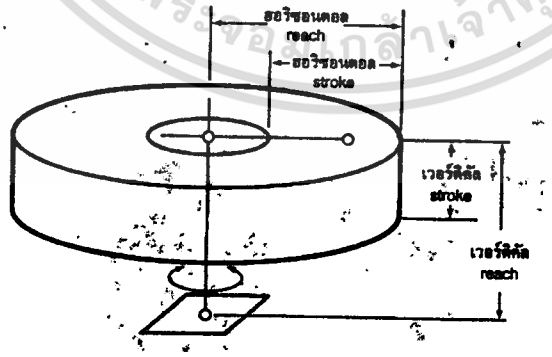
ความสามารถในการรับภาระ คือน้ำหนักสูงสุดของวัตถุที่แขนกลสามารถเคลื่อนย้ายได้

ความเร็วสูงสุด คือความเร็วในการหมุนข้อมือสูงสุดที่สามารถทำได้

ระยะ REACH และ STROKE เป็นส่วนที่ใช้วัดขอบเขตการเคลื่อนที่ในการทำงาน โดยประกอบด้วยระยะ HORIZONTAL REACH และ STROKE ระยะ VERTICAL REACH และ STROKE

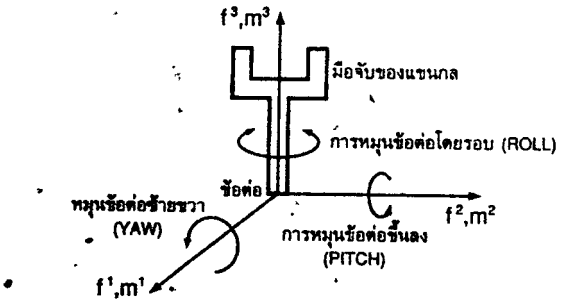
ระยะ HORIZONTAL REACH คือความยาวของรัศมีจากจุดศูนย์กลางแกนหลักของแขนกลไปยังปลายมือ ส่วนระยะ HORIZONTAL STROKE คือความยาวที่มือสามารถเอื้อมถึง แสดงดังรูปที่ 8 เป็นการแสดงระยะ REACH และ STROKE ของแขนกลชนิด CYLINDRICAL ซึ่งประกอบด้วยระยะในแนว HORIZONTAL และ VERTICAL สำหรับความหมายของระยะ REACH และ STROKE ในแนว VERTICAL ก็คล้ายกันเพียงแต่เปลี่ยนจากแนวนอนมาเป็นแนวตั้ง

มุมในการหมุนมือเป็นส่วนที่ใช้วัดขอบเขตในการหมุนของมือว่าสามารถหมุนได้สูงสุดเท่าไร สำหรับการหมุนของมือนั้นในลักษณะพื้นฐานจะประกอบด้วย 3 ส่วนคือ YAW PITCH และ ROLL โดยการหมุนของมือจะพิจารณาในพื้นที่ 3 มิติ แสดงดังรูปที่ 9 โดยความหมายถ้าเปรียบเทียบกับลักษณะของมือมนุษย์จะได้ดังนี้



รูปที่ 8 แสดงระยะ REACH และ STROKE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9 ทิศทางการหมุนของข้อต่อในแบบ YAW, PITCH และ ROLL

YAW คือการหักเหข้อมือไปทางซ้ายหรือขวา

PITCH คือการหักข้อมือขึ้นหรือลง

ROLL คือการหมุนข้อมือโดยรอบ โดยเปลี่ยนจากหน้ามือไปสู่หลังมือซึ่งมุมในการหมุนทั้ง 3 ลักษณะมีหน่วยเป็นองศา

ความสามารในการทำซ้ำ หมายถึงค่าผิดพลาดที่เกิดจากการทำงานซ้ำๆ กันหลายครั้ง มีหน่วนเป็นมิลลิเมตร ถ้าค่าผิดพลาดนี้มีค่าน้อยๆ แสดงว่าแขนกลชุดนี้มีความสามารถในการทำซ้ำสูง

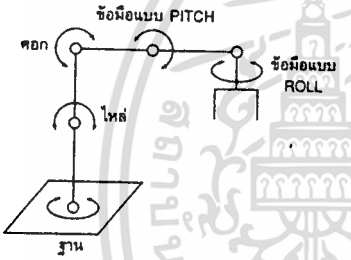
ความเที่ยงตรงเป็นความละเอียดที่แขนกลสามารถเคลื่อนที่ได้เล็กที่สุด เช่น แขนกลชุดหนึ่งมีความเที่ยงตรงเท่ากับ 1 มิลลิเมตร หมายความว่าแขนกลชุดนี้สามททารถเคลื่อนที่ได้ระยะสั้นที่สุดคือ 1 มิลลิเมตร ดังนั้นการเคลื่อนที่จะเป็นขั้นจาก 1, 2, 3, 4,.....มิลลิเมตร ตามต้องการ ถ้าต้องการสั่งให้แขนกลชุดนี้เคลื่อนที่ไป 1.5 มิลลิเมตร แขนกลชุดนี้ไม่สามารถทำได้ เพราะค่าที่เล็กที่สุดที่สามารถเคลื่อนที่ได้คือ 1 มิลลิเมตร ดังนั้นค่าต่อไปที่จะเคลื่อนได้คือ 2 มิลลิเมตร ไม่สามารถเคลื่อนไป 1.5 มิลลิเมตรได้เป็นต้น ความแม่นยำเป็นการวัดความสามารถของแขนกลในการเคลื่อนมือไปตำแหน่งต่างๆ ที่ต้องการ โดยมีค่าผิดพลาดไปจากที่กำหนดเท่าไร เช่น กำหนดตำแหน่ง X,Y,Z ต้นทางที่จะหยิบวัตถุในพื้นที่ 3 มิติ และตำแหน่ง X,Y,Z ปลายทางที่นำวัตถุไปวางเมื่อสั่งให้แขนกลทำงาน สังเกตตำแหน่งจริงที่แขนกลสามารถเคลื่อนที่ไปได้ทั้งตำแหน่งต้นทาง และปลายทางว่าผิดพลาดไปเท่าไรถ้าแขนกลชุดใดมีค่าความผิดพลาดในลักษณะนี้น้อยมากแสดงว่าแขนกลชุดนั้นมีความแม่นยำสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริทึมในการควบคุมแขนกล

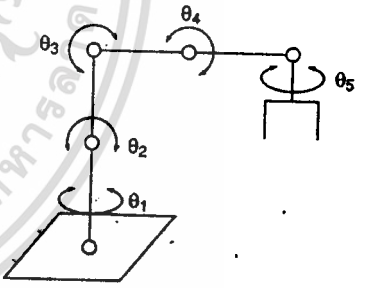
สิ่งหนึ่งที่บ่งบอกถึงประสิทธภาพในการทำงานของแขนกล คือความแม่นยำและเที่ยงตรงซึ่งการเคลื่อนไหวในการทำงานของแขนกลจะประกอบด้วยการเคลื่อนไหวของฐาน, ไหล่, ข้อศอก, ข้อมือ เป็นต้น การเคลื่อนไหวในส่วนต่างๆ นี้จะส่วผลให้ตำแหน่งของมือเคลื่อนที่ไปตามตำแหน่งต่างๆ ที่ปฏิบัติงาน ฉะนั้น ถ้าส่วนต่างๆ นี้มีการควบคุมที่ไม่ดีพอจะส่วผลถึงถึงความแม่นยำและเที่ยงตรงในการทำงานนั้น ลดน้อยลง

ดังนั้นมือของแขนกลจะเคลื่อนที่ไปตามตำแหน่งใดๆ ได้นั้นขึ้นอยู่กับข้อต่อต่างๆ จะเคลื่อนไปในตำแหน่งที่เหมาะสมและสอดคล้องกันด้วย ซึ่งการควบคุมก็ต้องควบคุมตำแหน่งของข้อต่อต่างๆ ให้เหมาะสมกันซึ่งสามารถแสดงได้ด้วยสมการคณิตรศาสตร์ สำหรับสมการที่จะกล่าวถึงในที่นี้มี 2 ลักษณะด้วยกันคือ สมการ direct kine matic และสมการ inverse kine matic ซึ่งสมการทั้ง 2 ลักษณะใช้แสดงความสัมพันธ์ของข้อต่อต่างๆ และตำแหน่งปลายมือ โดยรายละเอียดจะกล่าวต่อไป



รูปที่ 1 ไดอะแกรมการทำงานของแขนกลในรูปแบบ 5 ดีกรีออฟฟรีดอม

รูปที่ 2 บล็อกไดอะแกรมของสมการ direct kinematic



รูปที่ 3 ตำแหน่ง home ของแขนกล

ก่อนที่จะกล่าวถึงสมการของแขนกล มาทราบถึงศัพท์บางคำที่ใช้กันบ่อยในแขนกลหรือแขนหุ่นยนต์ คำแรกคือ ดีกรีออฟฟรีดอม (degree of freedom) หรือ DOF หมายถึงจำนวนข้อต่อหรือจุดต่อที่เคลื่อนที่เป็นอิสระต่อกัน ส่วนอีกคำหนึ่งคือก้าน (link) หมายถึงสิ่งที่เชื่อมต่อระหว่างข้อต่อกับข้อต่อ (ซึ่งข้อต่อนั้นอาจจะเป็นการเคลื่อนที่แบบ R หรือแบบ P ก็ได้ดังได้กล่าวมาแล้ว) ตัวอย่างเช่น แขนกลชนิด ARTICULATED ที่แสดงในรูปที่ 1 เป็นไดอะแกรมแสดงแขนกลที่มีการเคลื่อนที่แบบ 5 ดีกรีออฟฟรีดอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปในการควบคุมแขนกล ให้เคลื่อนที่ไปตามต้องการนั้น จะต้องมีตำแหน่งเริ่มต้นของแขน โดยเป็นตำแหน่งอ้างอิงในการเคลื่อนที่ เรียกว่าตำแหน่ง home ซึ่งก็ขึ้นอยู่กับชนิดของแขนกลและผู้ควบคุมว่าต้องการให้ตำแหน่งนี้อยู่ในลักษณะใด สำหรับบทความในตอนนี้จะอ้างอิงถึงตำแหน่ง home ในลักษณะดังรูปที่ 1 เป็นหลักรวมทั้งชนิดของแขนกลที่ใช้ในการอ้างอิงสมการหรือสูตรต่างๆ อ้างอิงเป็นชนิด ARTICULATED แบบ 5 ดีกรีออฟฟรีดอม

<u>สมการ Direct kinematic</u>

ความหมายของสมการ direct kinematic คือสมการของแขนกลที่แสดงความสัมพันธ์ของการเคลื่อนที่ของข้อต่อต่างๆโดยผลลัพท์ของสมการคือ ตำแหน่งปลายมือในพื้นที่ 3 มิติ (ตำแหน่ง X,Y,Z) บล็อคไดอะแกรมแสดงดังรูปที่ 2

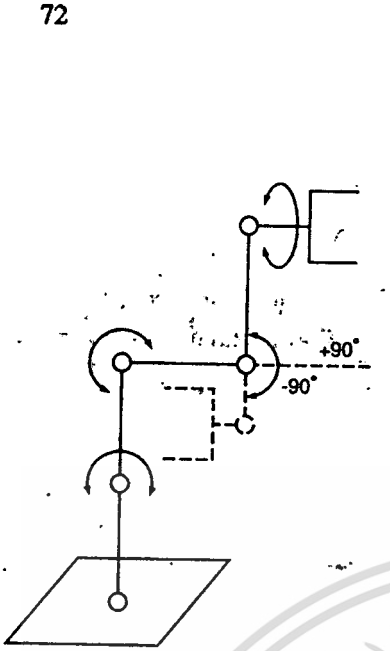
จากบล็อคไดอะแกรมจะพบว่าส่วนที่เป็นอินพุทของสมการคือ ค่าของมุมที่ข้อต่อต่างๆ ซึ่งจะขึ้นอยู่กับแขนกลว่ามีกี่ข้อต่อ ส่วนค่าพารามิเตอร์ที่เกี่ยวข้อง จะเป็นค่าคงที่ประจำตัวแขนกล เช่น ความยาวของแขนแต่ละก้านจากข้อต่อหนึ่งไปยังอีกข้อต่อหนึ่ง เป็นต้น และเมื่อผ่านสมการนี้ผลลัพท์ที่ได้ออกมาจะเป็นตำแหน่งของปลายมือที่แขนกลเคลื่อนที่ไปในพื้นที่ 3 มิติ (X,Y,Z)

<u>การอ้างอิงมุมของแขนกล</u>

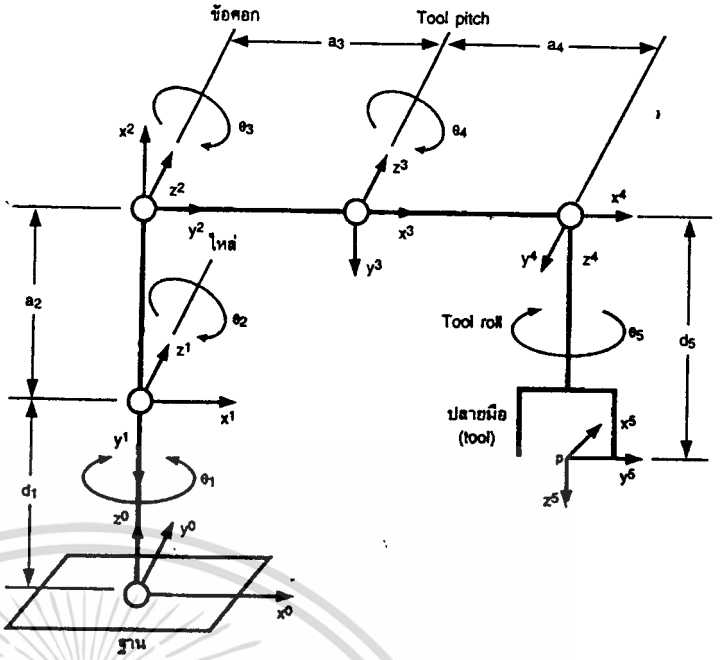
สำหรับค่าของมุมของแต่ละข้อต่อที่ป้อนให้กับสมการ direct kinmatic จะถูกอ้างอิงจากตำแหน่ง home ดังนั้น ในการป้อนค่ามุมของแต่ละข้อต่อจำเป็นต้องรู้ขนาดของทิศทางและมุมที่จะต้องป้อนด้วยโดยปกติจะกำหนดให้ที่ตำแหน่ง home ของทุกข้อต่อเป็น 0 องศา ตัวอย่างการอ้างอิงมุมดังรูปที่ 3 ซึ่งเป็นลักษณะตำแหน่ง home อีกรูปแบบหนึ่ง ค่าของมุมของทุกข้อต่อในตำแหน่งนี้จะเป็น 0 องศา Q1,Q2,Q3,Q4 และ Q5 กำหนดให้เป็น 0 องศา สมมติต้องการเคลื่อนเฉพาะ Q4 ไป +90 องศา รูปแขนกลที่ได้จะเป็นดังรูปที่ 4

จากรูปที่ 4 ค่าของมุมอื่นๆ ยังคงเป็น 0 องศามีเฉพาะค่า Q4 เท่านั้นที่เปลี่ยนแปลง ซึ่งถ้าเป็นค่าลบ มุมก็จะตรงข้ามกับทิศทางบวกโดนเริ่มนับจากตำแหน่งที่กำหนดให้เป็นตำแหน่งอ้างอิง ดังนั้นมุมที่ข้อต่ออื่นๆ ก็จะมีวิธีการอ้างอิงมุมในลักษณะเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 แสดงตัวอย่างในการอ้างอิงมุม



รูปที่ 5 ตำแหน่งด้านของแขนกลชนิด ARTICULATED แบบ 5 ดีกรีออฟฟรีดอม

ในรูปที่ 5 จะแสดงทิศทางของเวกเตอร์ในแต่ละข้อต่อ รวมทั้งแสดงค่าพารามิเตอร์ต่างๆ ที่เกี่ยวข้องซึ่งสามารถที่จะอธิบายแยกเป็นส่วนๆ ได้ดังนี้

Q1 คือมุมในการกวาดแขนไปมารอบฐาน

Q2 คือมุมในการหมุนไหล่ขึ้นลง

Q3 คือมุมในการหมุนข้อศอก

Q4 คือมุมในการหมุนข้อมือแบบหักขึ้นลง (toll pitch)

Q5 คือมุมในการหมุนข้อมือในลักษณะหน้ามือไปหลังมือ (toll roll)

จากตอนที่แล้วได้กล่าวถึงเรื่องการหมุนข้อมือ ว่ามีการหมุนอยู่ใน 3 ลักษณะ คือ YAW, PITCH, และ ROLL ซึ่งได้อธิบายความหมายไปแล้ว แต่สำหรับแขนกลที่ใช้อ้างอิงนี้เป็นแบบ 5 ดีกรีออฟฟรีดอม โดนตัดการเคลื่อนที่ของข้อมือแบบ YAW ไป ดังนั้น ถ้าแขนกลชนิด ARTICULATED แบบ 6 ดีกรีออฟฟรีดอม ก็จะมีการเคลื่อนที่แบบ YAW นี้ด้วย ซึ่งจะทำให้แขนกลมีลักษณะการเคลื่อนที่ของข้อต่อทุกข้อต่อคล้ายกับแขนของมนุษย์มากที่สุด

สำหรับความจำเป็นของการมีจำนวน ดีกรีออฟฟรีดอม เนื่องจากดีกรีออฟฟรีดอมมีการเคลื่อนที่อิสระของแต่ละข้อต่อ ถ้ามีจำนวนดีกรีออฟฟรีดอมมากแขนกลก็สามารถเคลื่อนไหวได้ซับซ้อนขึ้น แต่การสร้างและการควบคุมก็จะซับซ้อนเป็นเงาตามตัวด้วย ฉะนั้น ในการเลือกจำนวนดีกรีออฟฟรีดอมของแขนกลต้องขึ้นอยู่กับความเหมาะสมของงานที่ต้องการด้วย

สำหรับเวกเตอร์ X_6, Y_0, Z_0 จนถึง X_5, Y_5, Z_5 เป็นเวเตอร์ที่แสดงแนวการเคลื่อนที่ ส่วนค่าพารามิเตอร์ที่เป็นค่าคงที่ของแขนกลมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

d1 คือระยะห่างระหว่างฐานไปยังจุดหมุนที่ไหล่

a2 คือระยะห่างจากจุดหมุนที่ไหล่ไปยังจุดหมุนที่ศอก

a3 คือระยะห่างจากจุดหมุนที่ศอกไปยังจุดหมุนที่ข้อมือหมุนแบบ PITCH

a4 คือระยะห่างจากจุดหมุนที่ข้อมือแบบ PITCH ไปยังจุดหมุนข้อมือหมุนแบบ ROLL

d5 คือระยะห่างจากจุดหมุนที่ข้อมือหมุนแบบ ROLL ไปยังจุดปลายมือ

จากนั้นทำการป้อนค่ามุม Q_1, Q_2, Q_3, Q_4 และ Q_5 และค่าพารามิเตอร์ d1, a2, a3, a4, d5 ให้กับสมการ direct kinematic จะได้ตำแหน่ง P(X,Y,Z) ในพื้นที่ 3 มิติที่มือไปวางอยู่ เนื่องจากที่มาของสมการได้มาจากการวิเคราะห์ทางด้านเวกเตอร์ ฉะนั้นถ้าผู้ใดสนใจสามารถค้นคว้าเพิ่มเติมได้จากหนังสืออ้างอิง โดนสมการมีดังนี้

$$X = C_1 (a_2C_2 + a_3C_2 + a_4C_{234} + d_5S_{234})$$

$$Y = S_1 (a_2C_2 + a_3C_{23} + a_4C_{234} + d_5C_{234})$$

$$Z = d_1 - a_2S_2 - a_3S_{23} - a_4S_{234} - d_5C_{234}$$

จากสมการมีการใช้สัญญลักษณ์เพื่อย่อความหมายให้สมการดูง่ายขึ้น ความหมายของตัวย่อเหล่านี้คือ

S_k คือ sink

C_k คือ cosk

S_{ijk} คือ sine(i+j+k)

C_{ijk} คือ cos(i+j+k) โดยค่า i, j, k คือมุมค่าใดๆ

ตัวอย่างเช่น $C_1 = \cos Q_1$ หรือ $S_{234} = \sin (Q_2+Q_3+Q_4)$ เป็นต้น

รูปที่ 6 บล็อกไดอะแกรมของสมการ inverse kinematic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

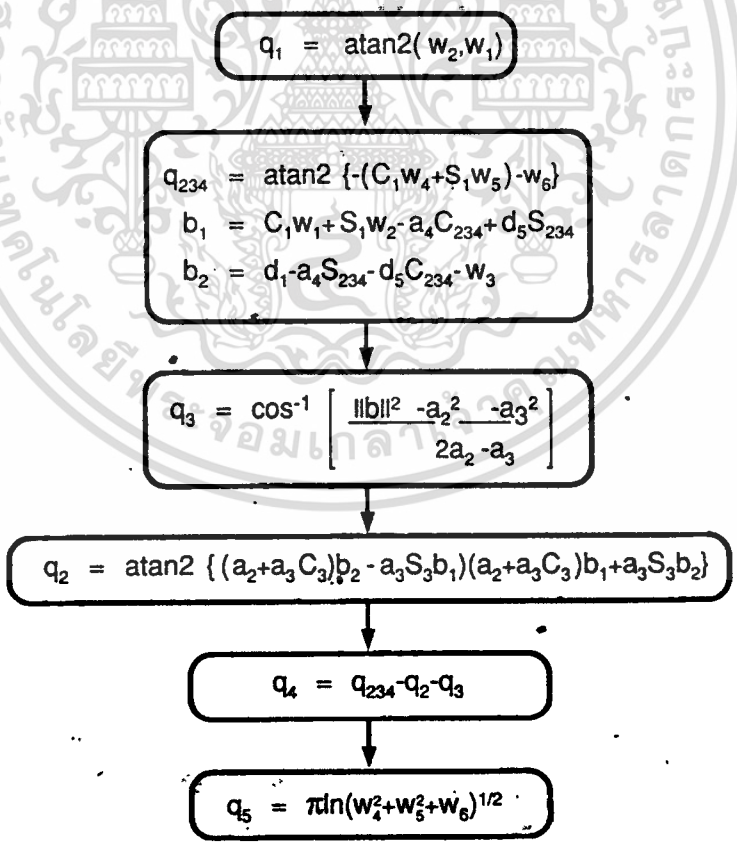
สำหรับแขนกลชนิดอื่นๆ ที่ไม่ใช้แขนกลชนิด ARTICULATED แบบ 5 ดีกรีออฟฟรีดอม ก็มีในสมการ direct kinematic เช่นกัน

สมการ Inverse Kinematic

จากที่ผ่านมาพอทราบความหมายของสมการ direct kinematic กันไปแล้วในต่อมาจะได้กล่าวถึงสมการอีกลักษณะหนึ่ง ซึ่งอาจกล่าวได้ว่าเป็นหัวใจในการควบคุมแขนกลแบบหนึ่งสมการชุดนี้คือสมการ inverse kinematic

ความหมายของสมการ inverse kinematic จะมีความหมายที่ตรงข้ามกับสมการ direct kinematic คือ จากเดิมที่ป้อนค่าของมุมที่ข้อต่อต่างๆ เข้าไปในสมการผลลัพท์ที่ได้ จะเป็นตำแหน่ง X, Y, Z ที่ปลายมือเคลื่อนไป แต่สมการนี้จะเปลี่ยนการป้อนตำแหน่ง X, Y, Z ของปลายมือเข้าไปแล้วจะได้ผลลัพท์เป็นมุมของแต่ละข้อต่อว่าจะต้องเลื่อนไปกี่องศาจากตำแหน่ง home ปลายมือจึงจะไปอยู่ตำแหน่งที่ต้องการ บล็อคไดอะแกรมการทำงานจะแสดงดังรูปที่ 6

จากรูปที่ 5 โดนค่าพารามิเตอร์ต่างๆ ยังคงใช้ค่าเหมือนเดิมทุกประการจากความหมายของสมการ X,Y,Z ที่ปลายมือและค่าพารามิเตอร์ที่เกี่ยวข้องผลลัพท์จะได้เป็นมุมของข้อต่อต่างๆ คือ Q_1 ถึง Q_5 โดยสามารถเขียนเป็นอัลกอริทึมของสมการ inverse kinematic ได้ดังรูปที่ 7



จากรูปที่ 7 แสดงลำดับการหาค่ามุมตามข้อต่อต่างๆ ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

q1 คือ Q1 ซึ่งเป็นมุมในการหมุนฐาน

atan2 คือ อาร์คแทน 4 ควอแดนท์

q2 คือ Q2 เป็นมุมที่ไหล่

q3 คือ Q3 เป็นมุมที่ศอก

q4 คือ Q4 เป็นมุมการหมุนข้อมือคน แบบ PITCH

q5 คือ Q5 เป็นมุมการหมุนข้อมือตนแบบ ROLL

q234 คือมุมการหมุนข้อมือแบบ PITCH โดยอ้างอิงจากระนาบพื้น (X_0,Y_0,Z_0)

w1 คือตำแหน่งปลายมือที่ X

w2 คือตำแหน่งปลายมือที่ Y

w3 คือตำแหน่งปลายมือที่ Z

w4,w5,w6 คือค่าในการหมุนข้อมือ

C1 คือ cos Q1

S1 คือ sin Q1

C3 คือ cos Q3

S3 คือ sin Q3

C23 คือ cos ของมุม q234

S234 คือ sin ของมุม q234

$\|b\| = b_1^2 + b_2^2$

สำหรับ q234 นั้นสามารถอธิบายรายละเอียด โดยดูรูปที่ 8 ประกอบ จากความความหมายของq234 เป็นมุมการหมุนข้อมือแบบ PITCH โดยอ้างอิงจากระนาบพื้น X0,Y0,Z0 การวัดมุมจะวัดเปรียบเทียบกับ Z0

สังเกตุได้ว่า q234 qเป็นมุมที่มือกระทำกับระนาบพื้น ในที่นี้คือ รูปแบบการวางมือในการเคลื่อนที่ไปยังตำแหน่งที่ต้องการ ซึ่งที่มาของ q234 มาจากเวกเตอร์ w4,w5,w6 ตัวอย่างเช่น ต้องการเคลื่อนมือไปจับวัตถุทางด้านข้าง แสดงดังรูปที่ 9 มุม q234 ซึ่งเป็นมุมในการว่างมือเมื่อเทียบกับระนาบพื้นของวัตถุจะมีค่าเป็น -90 องศา

โดยทั่วไปค่าอินพุทที่ป้อนให้กับอัลกอลิทึมของสมการ inverse kinematic จะเป็นค่า w1,w2,w3,q234 และ q5 (ตำแหน่งของวัตถุ, มุมในการวางมือ, มุมในการหมุนข้อมือแบบ ROLL) ตามลำดับ ดังนั้น ในการทำงานจริงๆ จะพบว่าอัลกอลิทึมของสมการ inverse kinematic ในรูปที่ 7 จะมีบางส่วนที่สามารถตัดออกไปได้เพื่อลดความซับซ้อนของสมการ เนื่องจากในการควบคุมแขนกลโดยใช้อัลกอลิทึมของสมการ inverse kinematic ค่า q234 ซึ่งเป็นมุมในการวางมือเมื่อเทียบกับระนาบของวัตถุและมุมในการหมุนข้อมือแบบ ROLL ผู้ควบคุมจะเป็นผู้กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลงไปเอง โดยไม่ต้องผ่านอัลกอลิทึมของสมการ inverse kinematic ซึ่งสมการส่วนที่สามารถตัดออกไปได้คือ

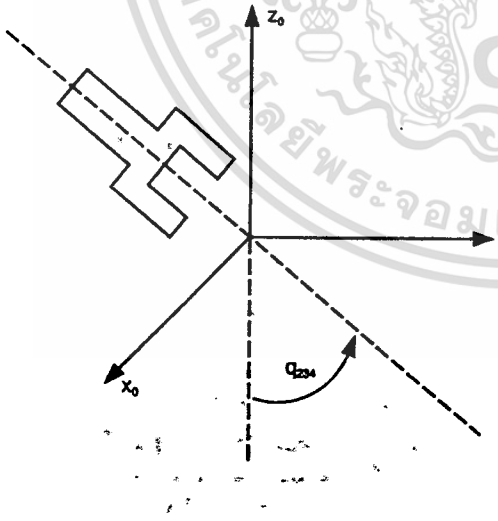
$$q234 = \text{atan}\,[(-C1w4 - S1w5\,) - w6]$$

และ $q5 = \ln(\,w42 + w52 + w62\,)$

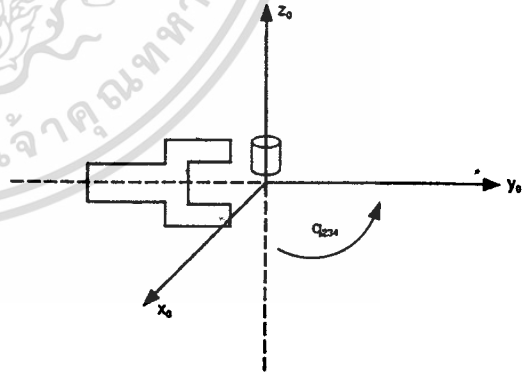
เอ้าท์พุทที่ได้จะเหลือเพียงมุม Q1, Q2, Q3 และ Q4 นอกจากอัลกอลิทึมของสมการ inxerse kinematic ของแขนกลชนิด ARTICULATED แบบ 5 ดีกรีออฟฟรีดอม นี้แล้วยังมีของแขนกลชนิดอื่นๆ อีก รวมทั้งที่มาของสมการนี้ด้วย สามารถค้นคว้าเพิ่มเติมได้จากหนังสืออ้างอิงท้ายเล่ม

<u>การประยุกต์ใช้งาน</u>

สำหรับการประยุกต์ใช้งานของอัลกอลิทึมของสมการ inverse kinematic อันดับแรกที่เห็นได้ชัดคือ ควบคุมการจับวัตถุที่ตำแหน่ง X,Y,Z ค้นทางให้นำไปวางที่ตำแหน่ง X,Y,Z, ปลายทาง โดยคอมพิวเตอร์จะเป็นตัวคำนวณค่ามุมของข้อต่อต่างๆ เมื่อคำนวณเสร็จคอมพิวเตอร์จะส่งค่ามุมต่างๆ ไปให้กับชุดขับเคลื่อน ชุดขับเคลื่อนก็จะทำการขับเคลื่อนข้อต่อต่างๆ ไปตามมุมที่ได้มาของอัลกอลิทึมของสมการ inverse kinematic ผลลัพท์ที่ได้ปลายมือจะไปอยู่ในตำแหน่งที่ต้องการ จากนั้นก็เป็นหน้าที่ของผู้เขียนโปรแกรมควบคุมว่าจะให้ทำอย่างไรต่อไป



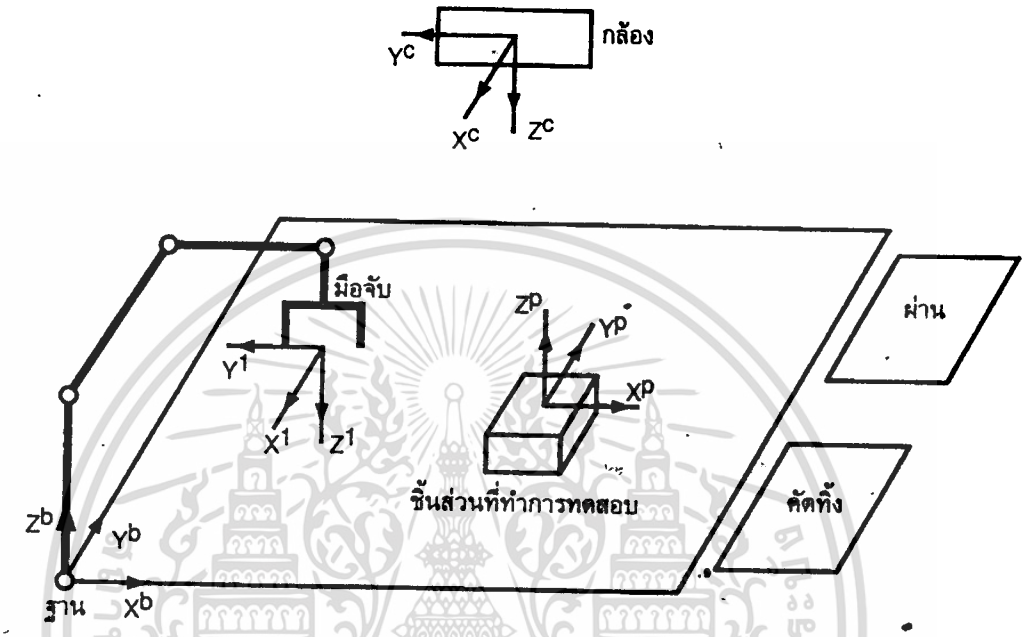
รูปที่ 8 แสดงการอ้างอิง q_{234}



รูปที่ 9 ตัวอย่างการอ้างอิง q_{234}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

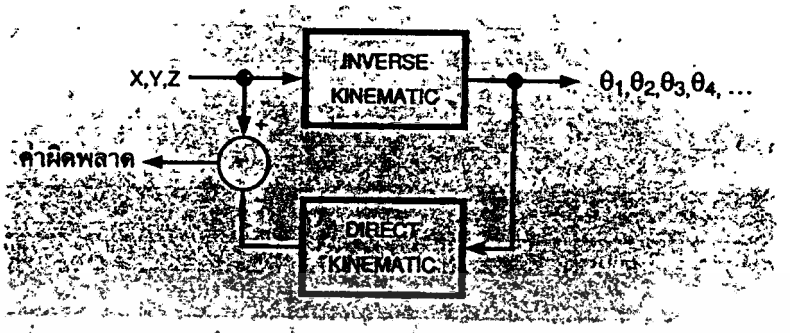
อัลกอลิทึมของสมการinverse kinematic เป็นเพียงส่วนที่ช่วยในการคำนวณค่ามุมที่แต่ละข้อต้องเคลื่อนไปเมื่อให้มืออยู่ในตำแหน่งที่ต้องการ ส่วนที่ 2 ของการประยุกต์ใช้งานอัลกอลิทึมของสมการ inverse kinematic โดยการใช้ควบคู่กับกล้องในการมองภาพดังรูปที่ 10 กล้อง



รูปที่ 10 แสดงตัวอย่างการนำแขนกลมาประยุกต์ใช้งานในการควบคุมกล้องตรวจจับในสายงานการตรวจสอบคุณภาพอุปกรณ์

จะทำหน้าที่ประมวลสัญญาณภาพที่รับเข้ามาเปลี่ยนให้เป็นตำแหน่ง X,Y,Z ของวัตถุ แล้วส่งค่า X,Y,Z ไปให้กับอัลกอลิทึมของสมการ inverse kinematic ซึ่งจะทำให้แขนกลสามารถทำงานได้เหมือนกับมีดวงตาในการมองเห็นวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11 บล็อกไดอะแกรมในการตรวจสอบตำแหน่งด้วยสมการ direct kinematic

สำหรับสมการ direct kinematic ที่ใช้งานอยู่ บางส่วนใช้ในการตรวจสอบตำแหแหน่งในการเขียนโปรแกรมจำลองหรือซิมูเลต การทำงานของแขนกล โดยจะใช้สมการ direct kinematic เป็นชุดป้อนกลับแสดงดังรูปที่ 11

จากบล็อคไดอะแกรมค่าเอ้าท์พุตของอัลกอลิทึมสมการ inverse kinematic จะถูกสมการ direct kinematic นำมาเป็นค่าอินพุตเพื่อป้อนกลับไปหาค่าผิดพลาดในการควบคุมตำแหน่งในการเคลื่อนที่ ค่าผิดพลาดที่ได้จะถูกส่งไปยังโปรแกรมอีกชุดหนึ่ง ซึ่งทำหน้าที่เป็นตัวควบคุม ถ้าเปรียบเทียบแขนกลจริงๆ กับโปรแกรมจำลองการทำงานนั้น สมการ direct kinematic ก็คือตัวตรวจจับตำแหน่งนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

การใช้ Personal Computer ควบคุมแขนกล

ในยุคปัจจุบันคอมพิวเตอร์ถูกนำมาใช้งานเกือบทุกสาขา และงานหนึ่งที่ได้รับความสนใจอย่างมากก็คือ เทคโนโลยีของหุ่นยนต์ซึ่งเป็นจักรกลที่ควบคุมด้วยคอมพิวเตอร์และและเครื่องกลอีกมากมายหลายชนิดหันมาใช้คอมพิวเตอร์ควบคุมแทนมนุษย์ ในหุ่นยนต์หรือเครื่องกลมักมีองค์ประกอบหลักก็คือมอเตอร์ประกอบอยู่ด้วยและเครื่องควบคุมด้วยคอมพิวเตอร์ส่วนใหญ่มักมี สเต็ปเปอร์มอเตอร์เป็นตัวขับเคลื่อน ดังนั้นเราจึงต้องทำความเข้าใจการทำงานของสเต็ปเปอร์มอเตอร์เป็นอันดับแรก

ในเครื่องคอมพิวเตอร์ IBM PC หรือเทียบเท่า พอร์ตเครื่องพิมพ์โดยปกติจะสามารถแลตช์ข้อมูลได้ เมื่อเขียนข้อมูลชุดหนึ่งไปยังพอร์ตข้อมูลชุดนี้จะถูกแลตช์จนกระทั่งมีข้อมูลชุดใหม่ส่งมาอีก แต่ในคอมพิวเตอร์บางเครื่องก็อาจจะไม่มีในส่วนนี้ การใช้วงจรแลตช์บนบอร์ดเครื่องพิมพ์อาจทำให้เกิดปัญหาในการส่งผ่านข้อมูล มากกว่าแบบไม่มีแลตช์บนพอร์ตเครื่องพิมพ์อย่างไรก็ตามการมีวงจรแลตช์ติดตั้วอยู่อาจทำให้เกิดประโยชน์ในด้านอื่นด้วย เช่น ช่วยในการกำหนดและเลือกตำแหน่งแอดเดรสให้กับวงจรภายนอกได้ง่าย

พอร์ตเครื่องพิมพ์หรือพอร์ตแบบขนาน (parallel port) 1 พอร์ต สามารถควบคุมบอร์ดได้พร้อมกัน 4 บอร์ด ซึ่งแต่ละบอร์ดจะมีตำแหน่งแอดเดรสของตนเองทำให้ควบคุมสเตปเปอร์มอเตอร์ได้ถึง 8 ตัวพร้อมๆ กัน บนบอร์ดควบคุมออกแบบให้มีวงจรแลตช์ด้วยเพื่อให้บอร์ดนี้สามารถนำไปใช้กับบอร์ดไมโครคอนโทรลเลอร์อื่นๆ ได้ นอกจากการใช้งานกับพอร์ตเครื่องพิมพ์เพียงอย่างเดียว

หลักการทำงานของวงจร

จากวงจรสมบูรณ์ของกล่องควบคุมการเคลื่อนที่ของแขนกลดังแสดงในรูป เริ่มต้นจาก IC1 ซึ่งเป็นตัวเชื่อมต่อระหว่างพอร์ตเครื่องพิมพ์และบอร์ดควบคุม IC1 เป็น ไอซีที่ทำหน้าที่แลตช์ข้อมูลซึ่งภายในเป็นฟลิปฟลอปแบบ D ฟลิปฟลอป จำนวน 8 ตัวอยู่ภายใน โดยใช้สัญญาณควบคุมแลตช์อีนาเบิลหรือสัญญาณนาฬิการ่วมกันที่ขา 11 ข้อมูลทั้ง 8 บิทจะถูกโหลดเข้าเมื่อขาสัญญาณแลตช์อีนาเบิลมีสถานะเป็นลอจิกไฮ และปรากฏทางขาเอาท์พุตทั้ง 8 คือขา 19, 2, 16, 5 และ 15, 6, 12, 9

ข้อมูลที่จะส่งไปยังกล่องควบคุมการเคลื่อนที่ของแขนกลทำได้โดยคอมพิวเตอร์ส่งข้อมูลขนาด 1 ไบท์ (8 บิท) ออกทางพอร์ตเครื่องพิมพ์แล้วส่งสัญญาณทำให้ขา 11 ของ IC1 เป็นลอจิก "1" ข้อมูลนี้จะปรากฏที่เอาต์พุตของไอซี 1 ทันที สัญญาณเอาต์พุตที่ได้ป้อนให้กับ IC2 และ IC3 ซึ่งเป็นไอซีทำหน้าที่ดีโค้ดเดอร์สายสัญญาณจาก 2 อินพุต ออกเป็น 4 เอาต์พุต 2 วงจรในตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ นอกจากนี้ยังไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ขา 5 และ 6 ของ IC2 ถูกนำมาใช้งานซึ่งเป็นเอาต์พุตของดีโคดเดอร์ตัวแรกโดยที่อีก 2 ขาเอาต์พุตไม่นำมาใช้งาน และที่ขา 10 และ 11 ก็ถูกนำมาใช้งานเช่นกันสำหรับดีโคดเดอร์ตัวที่ 2 ไอซี 6 ทำหน้าที่กลับสถานะลอจิกจากเอาต์พุตของดีโคดเดอร์เป็นตรงกันข้าม จากแอคตีฟลอจิก "0" เป็นแอคตีฟลอจิก "1" สำหรับป้อนให้วงจรขับ

<u>การทดสอบ</u>

เมื่อลงอุปกรณ์เป็นที่เรียบร้อยแล้วให้ทำการตรวจสอบความถูกต้องทั้งหมดอีกครั้งหนึ่งก่อนการทดสอบเครื่อง เมื่อแน่ใจแล้วให้ป้อนแรงดันไฟ 12 โวลต์ให้กับบอร์ดควบคุม ตรวจสอบแรงดันไฟเลี้ยง +5 โวลต์ ที่ขา 14 ของ IC4-IC7, ที่ขา 16 ของ IC 2 IC3 และที่ขา 20 ของ IC1 ทดลองจับที่ตัวถังของทรานซิสเตอร์ทุกตัว ถ้าเกิดความร้อนขึ้นสูงผิดปกติให้ปลดแหล่งจ่ายไฟออกทันที แล้วตรวจสอบว่ามีความผิดพลาดเกิดขึ้นที่ส่วนใด

ถ้าทุกอย่างเรียบร้อยให้ต่อบอร์ดควบคุมเข้ากับพอร์ตเครื่องพิมพ์แล้วเปิดเครื่องคอมพิวเตอร์ ต่อไปให้รันโปรแกรมทดสอบที่อยู่ในแผ่นดิสก์ 2 โปรแกรมได้แก่ STEPTEST และ STEPS ตามลำดับ แต่ก่อนจะทำการทดสอบโดยใช้โปรแกรมทั้งสองนี้ให้พิมพ์ไฟล์เอกสาร READ.ME มาศึกษาดูก่อนทดสอบ

โปรแกรมทดสอบ STEPTEST ทำหน้าที่ทดสอบการขับสเต็ปเปอร์มอเตอร์โดยแบ่งเป็นทีละเฟสในการหมุน เพื่อทดสอบการทำงานของวงจรบริดจ์ให้ป้อนแรงดันจากแหล่งจ่ายไฟ 12 โวลต์เข้าสู่บอร์ดควบคุมโดยไม่ต้องต่อสเต็ปเปอร์มอเตอร์เข้ากับบอร์ดควบคุม หลังจากนั้นก็ให้รันโปรแกรมทดสอบ บนจอภาพจะปรากฏข้อความให้เติมข้อมูลต่างๆ และให้ทำการวัดแรงดันที่ขาต่างๆ ของมอเตอร์ให้เป็นไปตามที่โปรแกรมระบุไว้จริงๆ แล้วถ้าต่อมอเตอร์เข้ากับบอร์ดควบคุมด้วยจะสังเกตเห็นมอเตอร์หมุนไป แต่อาจสังเกตได้ไม่ชัดเจนนัก

เพื่อให้สังเกตได้ง่ายขึ้นทำการทดสอบต่อไปโดยการรันโปรแกรม STEPS บ้างให้ต่อสเต็ปเปอร์มอเตอร์เข้ากับบอร์ดควบคุมโดยตรวจสอบขาของแต่ละเฟสภายในที่ต่อออกจากมอเตอร์ การหาขาหรือสายไฟที่ต่อกับขดลวดภายในแต่ละเฟสทำได้โดยสังเกตสีของสายไฟดังแสดงในรูปที่ 12 แต่อย่างไรก็ตามมอเตอร์บางรุ่นก็ไม่ได้ใช้สายไฟตามนี้ จึงควรใช้มัลติมิเตอร์ตรวจสอบให้แน่ใจเสียก่อน หลังจากนั้นให้รันโปรแกรมทดสอบ โปรแกรมจะให้ใส่จำนวนสเต็ปที่ต้องการหมุนและทิศทางการหมุน หลังจากกดแป้น ENTER แล้วให้สังเกตมุมของเพลาที่หมุนไปตามที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

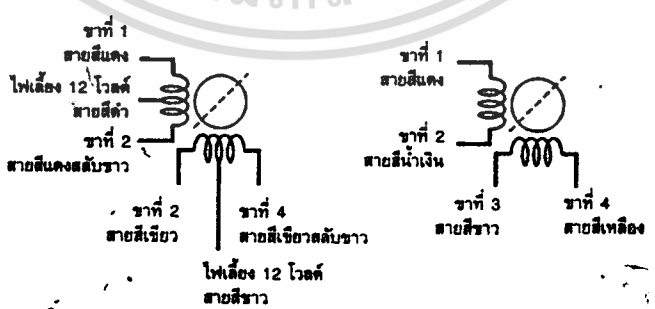
ตารางที่ 6 แสดงแรงดันตกคร่อมเอาท์พุทที่ขดลวดของมอเตอร์ที่ขาต่างๆและเฟสที่ถูกกระตุ้นให้ทำงานโดยสัญญาณข้อมูลทางอินพุท

เฟสที่ถูกกระตุ้น	สัญญาณควบคุมมอเตอร์ตัวที่ 1	สัญญาณควบคุมมอเตอร์ตัวที่ 2	แรงดันตกคร่อมเอาต์พุตที่ขาต่าง ๆ ของมอเตอร์	
			แรงดันไฟบวก	กราวด์
1	D0	D4	ขาที่ 1	ขาที่ 2
2	D1	D5	ขาที่ 3	ขาที่ 4
3	D2	D6	ขาที่ 2	ขาที่ 1
4	D3	D7	ขาที่ 4	ขาที่ 3

ตารางที่ 7 แสดงรหัสที่ใช้ควบคุมการส่งสัญญาณ อีนาเบิล ไปยังบอร์ดควบคุม โดยใช้สัญญาณจากพอร์ทเครื่องพิมพ์ต่างๆ

บอร์ดควบคุมที่ถูกเลือก	สัญญาณควบคุมจากพอร์ตเครื่องพิมพ์	รหัสที่กำหนดเพื่อส่งสัญญาณอีนาเบิล
ไม่มีบอร์ดใดทำงาน	-	04H
บอร์ดควบคุมที่ 1	$\overline{\text{STROBE}}$	05H
บอร์ดควบคุมที่ 2	$\overline{\text{AUTOFEED}}$	06H
บอร์ดควบคุมที่ 3	INIT	00H
บอร์ดควบคุมที่ 4	$\overline{\text{SELECT}}$	0CH

รูปที่ 12 แสดงการดูสีของสายไฟที่ต่อจากมอเตอร์ เพื่อทราบถึงการเชื่อมต่อภายในตัวมอเตอร์และทำการเชื่อมต่อกับบอร์ดควบคุมได้



ตารางที่ 4 แสดงรหัสควบคุมของมอเตอร์ทั้ง 2 ตัว

พิกัดกระแสของสเต็ปเปอร์มอเตอร์	ค่าตัวต้านทานจำกัดกระแส
500 มิลลิแอมป์	15 โอห์ม
800 มิลลิแอมป์	8.2 โอห์ม
1 แอมป์	6.8 โอห์ม
1.5 แอมป์	4.7 โอห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบการทำงานของวงจรควบคุมแขนกลนั้นสามารถทำได้โดยวิธีการเขียนโปรแกรมดีบัก (debug) ทำการโหลดรหัสที่ควบคุมมอเตอร์ให้ทำงานในแต่ละเฟสไปยังตำแหน่งแอดเดรสฐานของการ์ดเครื่องพิมพ์ (โดยปกติอยู่ที่ตำแหน่ง 378H) แล้วส่งข้อมูลลอจิก "1" ไปยังขาอีนาเบิลที่เซ็ทไว้บนบอร์ดควบคุมมอเตอร์เป็นค่าอื่นบ้างลงสู่บอร์ดควบคุม ดังแสดงในตารางที่ 5 การเขียนโปรแกรมดีบักทำได้ดังนี้

ให้เริ่มจากเรียกไฟล์เอ็กซ์ซีคิวชื่อ DEBUG จากแผ่นดิสก์ที่มีดอสอยู่ กลังจากนั้นจะปรากฏเป็น "-" ที่บรรทัดต่อมาให้คีย์โปรแกรมเข้าไป ซึ่งในที่นี้เป็นตัวอย่างสำหรับทำการทดสอบบอร์ดควบคุมที่เซ็ทให้ใช้สัญญาณอีนาเบิลที่สัญญาณควบคุม STROBE เท่านั้น ถ้าจะเปลี่ยนใหม่ให้ดูจากตารางที่ 7 ประกอบในขณะทำการทดสอบไม่ต้องต่อสเตปปิ้งมอเตอร์เข้ากับบอร์ดควบคุม โดยเมื่อกด Enter ในแต่ละบรรทัดให้นำมัลติมิเตอร์ตรวจสอบไปทีละจุดตามโปรแกรมที่เขียนขึ้นด้วยแล้วสังเกตผล โปรแกรมมีรูปแบบในการเขียนดังนี้

- O 378 (รหัสควบคุมมอเตอร์) ; สั่งให้โหลดรหัสควบคุมมอเตอร์ไปยังพอร์ตเครื่องพิมพ์

- O 37A 05 ; ส่งสัญญาณลอจิก "0" ไปที่ขาอีนาเบิลบอร์ดควบคุมที่ 1

- O 37A 04 ; เซ็ทสัญญาณที่ขาอีนาเบิลเป็นลอจิก "1" เพื่อโหลดข้อมูลลงสู่แลตช์บนบอร์ดควบคุม

- q ; ออกจากการใช้โปรแกรมดีบัก

รหัสควบคุมมอเตอร์ คือรหัสเลขฐาน 16 ที่ใช้ควบคุมให้ขดลวดเฟสใดภายในตัวมอเตอร์ให้ทำงาน ซึ่งมีทั้งหมด 4 เฟส รหัสควบคุมของมอเตอร์ตัวที่ 1 และตัวที่ 2 แสดงดังในตารางที่ 5

หลังจากประมวลผลโปรแกรมแต่ละครั้งหรือทุกครั้งที่เปลี่ยนรหัสควบคุมใหม่ให้ตรวจสอบแรงดันที่ตกคร่อมเอาต์พุตแต่ละเฟสของมอเตอร์ซึ่งเชื่อมต่อกับบอร์ดว่าทำงานตรงตามที่ต้องการหรือไม่ โดยจะมีแรงดันตกคร่อม 12 โวลต์ แต่ละเฟสที่ถูกกระตุ้นให้ทำงานดังในตารางที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      REM test program to drive motor1 on stepper motor controller board

porta = &H378:      REM address of port A for LPT1     (LPT2= &H278)
portc = &H37A:      REM address of port C for LPT1     (LPT2= &H27A)
card = &H5:         REM card1=&H05  card2=&H06  card3=&H00 card4=&H0C
ontime = 30:        REM  specifies phase on time
offtime = 5:      REM  specifies delay between steps
steps = 1:          REM initialise variable to be valid into the while loop
state = 1:          REM initialise state to a known state
PRINT
PRINT "Stepper motor example driver program written in QBASIC"
PRINT
WHILE (steps > 0)
PRINT "number of steps to move (zero ends program):";
INPUT steps
IF (steps = 0) THEN END
DO
        PRINT "direction (0=clockwise 1=anticlockwise):";
        INPUT dir
LOOP WHILE (dir < 0 OR dir > 1):  REM unsure valid input otherwise repeat
PRINT "steps: ", steps, "direction: ", dir
FOR x = 1 TO steps:
     IF (dir < 1) THEN
       IF (state = 8) THEN state = 1 ELSE state = state * 2
     ELSE
       IF (state = 1) THEN state = 8 ELSE state = state / 2
     END IF

PRINT "state ", state
OUT portc, 4:                    REM make all card enables initially high
OUT porta, state:                REM write the data to the latch
OUT portc, card:                 REM make selected card enable go low
                                 REM card2=6, card3=0, card4=12
OUT portc, 0:                    REM load the data into the latch
FOR d = 1 TO ontime: NEXT:       REM on time delay

OUT porta, 0:                    REM turn the phases off data
OUT portc, card:                 REM make selected card enable go low
OUT portc, 4:                    REM load the new data into the latch
                                 REM on the rising edge of the enable line
FOR d = 1 TO offtime: NEXT:      REM off time delay
NEXT x:                          REM end of for loop

WEND:                            REM end of while loop
END:                             REM end of program r
```

โปรแกรมที่ 1 โปรแกรมทดสอบการทำงานของสเตปเปอร์มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปแกรมที่ 2 โปรแกรมควบคุมการเคลื่อนไหวของแขนกล

```
DECLARE SUB firstscreen ()
DECLARE SUB motor (mt, dir)
'//-------------------------------------------------------
'//  PROJECT/N : ROBOT ARM
'//  PROJECT/MA: Boonyalit Pongprasert
'//  PRG-NAME  : motorsp.bas
'//  DETAIL    : For control stepping motor
'//  DETAIL    :
'//  Copyright (c) 1990-1995, KECS GROUP SOFTWARE CO.,LTD.
'//  All rights reserved.
'//  COMPILER  : Qbasic from MS DOS
'//  LINK      :
'//  EXECUTE   : motorsp  <press Enter>
'//  AUTHOR    : Boonyalit Pongprasert
'//  DATE      : 20/04/1995
'//  LASTUPDATE:
'//  WHY UPDATE:
'//-------------------------------------------------------

COMMON SHARED porta
COMMON SHARED portc
COMMON SHARED card1
COMMON SHARED card2
COMMON SHARED card3
COMMON SHARED ontime
COMMON SHARED offtim
COMMON SHARED state1
COMMON SHARED state2
COMMON SHARED state3
COMMON SHARED state4
COMMON SHARED state5
COMMON SHARED state6
COMMON SHARED Forout

porta = &H378:        REM address of port A for LPT1    (LPT2= &H278)
portc = &H37A:        REM address of port C for LPT1    (LPT2= &H27A)
card1 = &H5:          REM card1=&H05  card2=&H06  card3=&H00 card4=&H0C
card2 = &H6:
card3 = &HC:
ontime = 100:          REM specifies phase on time
offtime = 5:          REM specifies delay between steps
state1 = &H1:           REM initialise state to a known state
state2 = &H10:
state3 = &H1:
state4 = &H10:
state5 = &H1:
state6 = &H10:
Forout = &H0:

CALL firstscreen
DO
    keypress$ = INKEY$
    SELECT CASE keypress$
           CASE "w"
                CALL motor(1, 1)
           CASE "s"
                CALL motor(1, 0)
           CASE "e"
                CALL motor(2, 1)
           CASE "d"
                CALL motor(2, 0)
           CASE "r"
                CALL motor(3, 1)
           CASE "f"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CALL motor(3, 0)
        CASE "t"
            CALL motor(4, 1)
        CASE "g"
            CALL motor(4, 0)
        CASE "y"
            CALL motor(5, 1)
        CASE "h"
            CALL motor(5, 0)
        CASE "u"
            CALL motor(6, 1)
        CASE "j"
            CALL motor(6, 0)

    END SELECT
LOOP UNTIL keypress$ = CHR$(27)


UB firstscreen
    CLS
    LOCATE 2, 20
    PRINT "ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿": LOCATE 3, 20
    PRINT "³  Program for control stepping motor   ³": LOCATE 4, 20
    PRINT "³              ROBOT ARM                ³": LOCATE 5, 20
    PRINT "ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ"
    LOCATE 7, 30: PRINT "ESC for Exit"
    LOCATE 12, 1: PRINT "Motor 1:"
    LOCATE 14, 1: PRINT "Motor 2:"
    LOCATE 16, 1: PRINT "Motor 3:"
    LOCATE 18, 1: PRINT "Motor 4:"
    LOCATE 20, 1: PRINT "Motor 5:"
    LOCATE 22, 1: PRINT "Motor 6:"
ND SUB

UB motor (mt, dir)
    OUT portc, 4:                   REM make all card enables initially high
    SELECT CASE mt
        CASE 1
        IF (dir = 1) THEN
           IF (state1 = 8) THEN state1 = 1 ELSE state1 = state1 * 2
        ELSE
           IF (state1 = 1) THEN state1 = 8 ELSE state1 = state1 / 2
        END IF
           Forout = state1 OR state2
           OUT porta, Forout:           REM write the data to the latch

           LOCATE 12, 40
           PRINT "forout  "; HEX$(Forout)
           LOCATE 12, 12
           PRINT state1

           card = &H5

        CASE 2

        IF (dir = 1) THEN
           IF (state2 = &H80) THEN state2 = &H10 ELSE state2 = state2 *
        ELSE
           IF (state2 = &H10) THEN state2 = &H80 ELSE state2 = state2 /
        END IF
           Forout = state1 OR state2
           OUT porta, Forout:           REM write the data to the latch

           LOCATE 12, 40
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    PRINT "forout  "; HEX$(Forout)
    LOCATE 14, 12
    PRINT HEX$(state2)

    card = &H5

CASE 3

IF (dir = 1) THEN
   IF (state3 = 8) THEN state3 = 1 ELSE state3 = state3 * 2
ELSE
   IF (state3 = 1) THEN state3 = 8 ELSE state3 = state3 / 2
END IF
   Forout = state3 OR state4
   OUT porta, Forout:              REM write the data to the latch

   LOCATE 12, 40
   PRINT "forout  "; HEX$(Forout)
   LOCATE 16, 12
   PRINT state3
   card = &H6

CASE 4
IF (dir = 1) THEN
   IF (state4 = &H80) THEN state4 = &H10 ELSE state4 = state4 * 2
ELSE
   IF (state4 = &H10) THEN state4 = &H80 ELSE state4 = state4 / 2
END IF
   Forout = state3 OR state4
   OUT porta, Forout:              REM write the data to the latch

   LOCATE 12, 40
   PRINT "forout  "; HEX$(Forout)
   LOCATE 18, 12
   PRINT HEX$(state4)

card = &H6

CASE 5

IF (dir = 1) THEN
   IF (state5 = 8) THEN state5 = 1 ELSE state5 = state5 * 2
ELSE
   IF (state5 = 1) THEN state5 = 8 ELSE state5 = state5 / 2
END IF
   Forout = state5 OR state6
   OUT porta, Forout:              REM write the data to the latch

   LOCATE 12, 40
   PRINT "forout  "; HEX$(Forout)
   LOCATE 20, 12
   PRINT state5

card = &HC
CASE 6
IF (dir = 1) THEN
   IF (state6 = &H80) THEN state6 = &H10 ELSE state6 = state6 * 2
ELSE
   IF (state6 = &H10) THEN state6 = &H80 ELSE state6 = state6 / 2
END IF
   Forout = state5 OR state6
   OUT porta, Forout:              REM write the data to the latch
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PRINT "forout  "; HEX$(Forout)
```

วิธีการใช้งานโปรแกรมที่ 1

โปรแกรมนี้เป็นโปรแกรมใช้ทดสอบการควบคุม stepper motor โดยการ RUN ไฟล์เอ็กซีคิ้วชื่อ STEPS ดังนี้

C:\ STEPS

Enter

C:\ Stepper motor example driver program written in C

C:\ Number of steps to move (zero ends program) :_

ต่อไปให้ทำการป้อนตัวเลขค่าเท่าไรก็ได้ลงไป เช่น 1000 เป็นต้น

Enter

C:\Direction (0:clockwise 1:anticlockwise) :_

ให้เราทำการป้อนค่า "1" เมื่อต้องการให้มอเตอร์หมุนตามเข็มนาฬิกาและป้อนค่า "0" เมื่อต้องการให้มอเตอร์หมุนทวนเข็มนาฬิกา หลังจากนั้นโปรแกรมก็จะทำงานทันทีแล้วจะจบที่ประโยคเดิมคือ

C:\ Number of steps to move (zero ends program) :_

ถ้าต้องการจบการทำงานให้กดปุ่ม "0" แล้วกด Enter เป็นอันจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

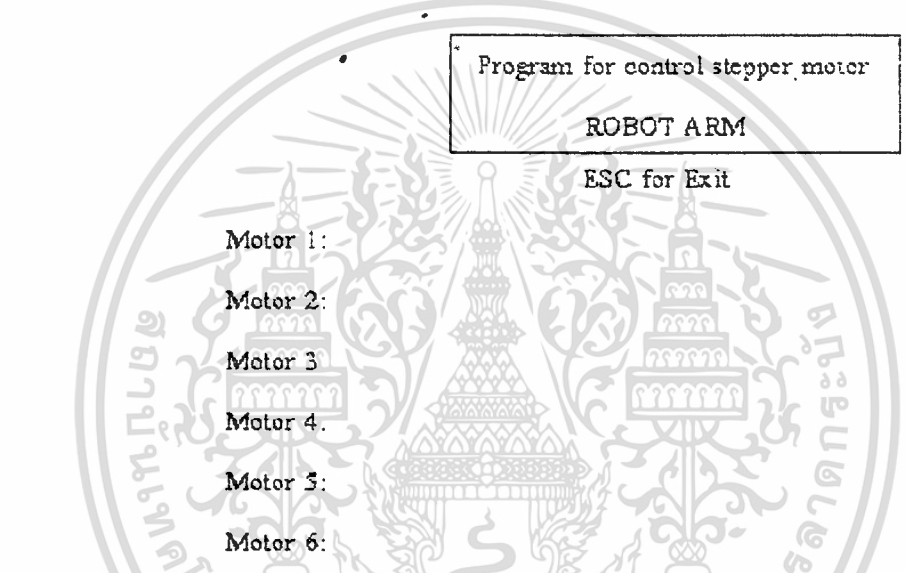
<u>วิธีการใช้งานโปรแกรมที่ 2</u>

โปรแกรมที่ 2 คือโปรแกรมใช้ควบคุมการเคลื่อนไหวของแขนกลโดยการควบคุมการหมุนของ stepper motor ที่อยู่ตามข้อต่อต่างๆ ของแขนกลซึ่งมีทั้งหมด 5 ตัว การใช้งานทำได้โดยการกดคำว่า "GO" แล้วกด Enter โปรแกรมจะเข้าสู่การทำงานโดยอัตโนมัติดังนี้

C:\ GO

Enter

basic/run motorsp



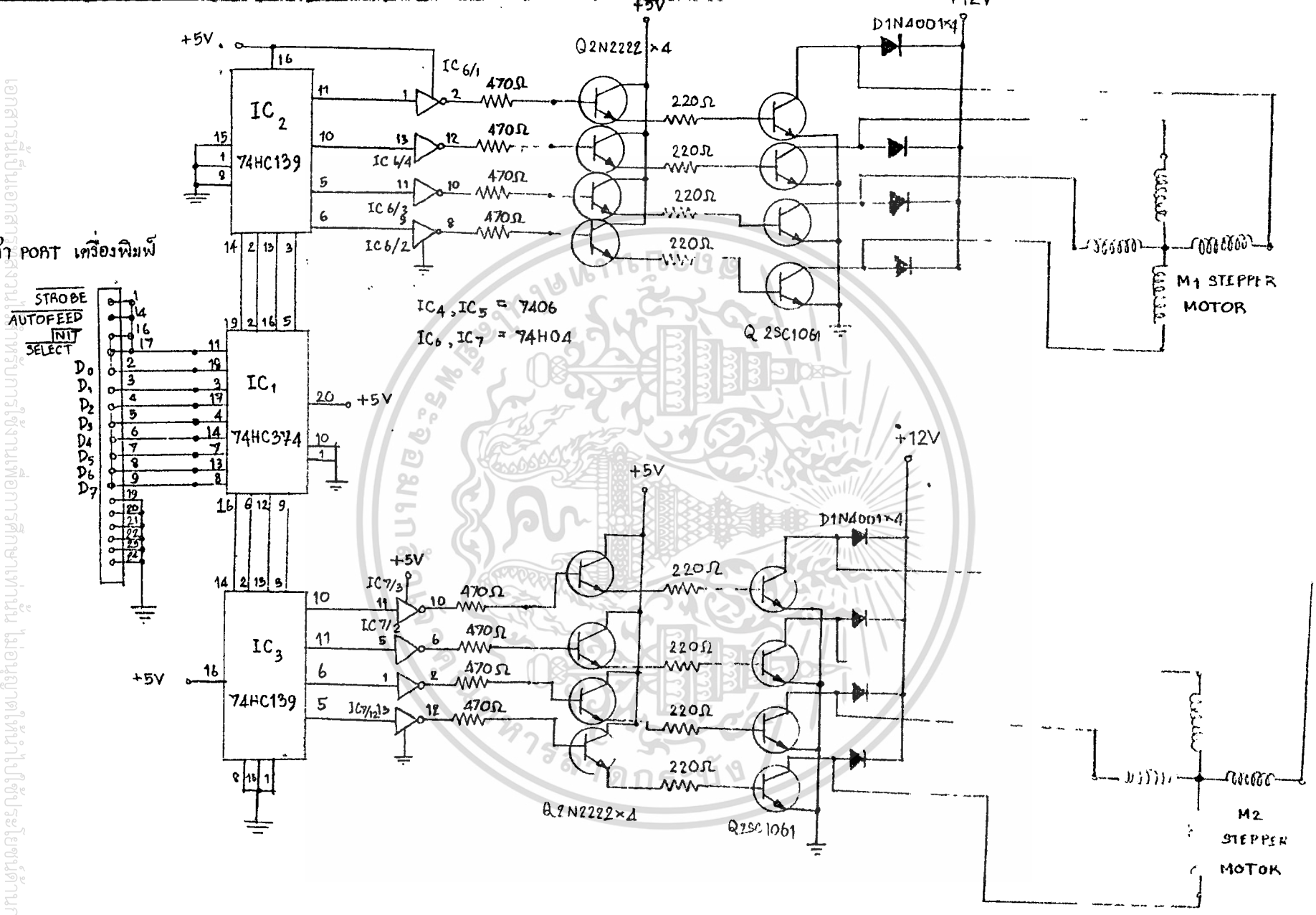
เมื่อ RUN โปรแกรมแล้วหน้าจอจะแสดงผลดังรูปข้างบน ถ้าต้องการบังคับให้แขนกลเกิดการเคลื่อนไหวสามารถทำได้โดยกดปุ่มบนแป้นพิมพ์เพื่อควบคุม stepper motor แต่ละตัวตามลำดับดังต่อไปนี้

W, E, R, T, Y, U : คือตำแหน่งของ Motor ตัวที่ 1,2,3,4,5,6 ตามลำดับ และเป็นการหมุนแบบตามเข็มนาฬิกา

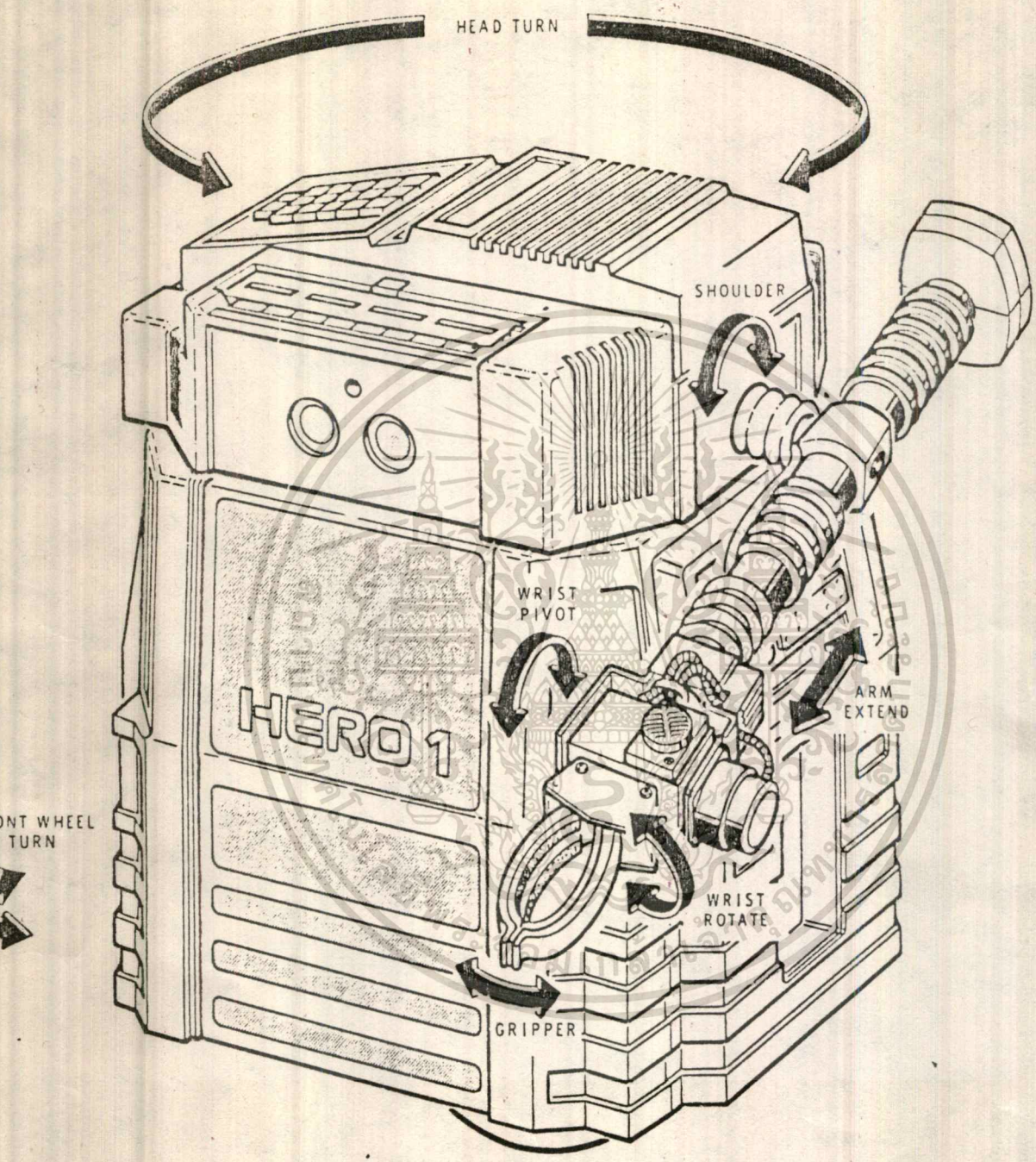
S, D, F, G, H, J : คือตำแหน่งของ Motor ตัวที่ 1,2,3,4,5,6 ตามลำดับ และเป็นการหมุนแบบทวนเข็มนาฬิกา

ถ้าต้องการออกจากโปรแกรมให้กดปุ่ม ESC เป็นอันจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13 วงจรสมบูรณ์ของวงจรควบคุมสเตปเปอร์มอเตอร์ผ่าน Personal Computer
(วงจรนี้สามารถควบคุมสเตปเปอร์มอเตอร์ได้ 2 ตัว)



รูปที่ 14 แสดงให้เห็นถึงขอบเขตการเคลื่อนที่ของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9
สรุปการวิจัยและข้อเสนอแนะ

สรุป

1. การควบคุมหุ่นยนต์ทำได้เฉพาะแมนนวลเท่านั้น ส่วนการทำให้หุ่นยนต์สามารจำได้ยังไม่สามารถทำได้

2. การพัฒนาซอร์ฟแวร์ครั้งต่อไปควรจะศึกษาฮาร์ดแวร์ให้ดีแล้วดึงซอร์ฟแวร์ส่วนที่เป็นปรแกรมย่อยที่ดีของซอร์ฟแวร์เดิมมาใช้บ้างเพื่อประหยัดเวลา

3. แบตเตอร์รี่หากไม่ได้รับการบำรุงรักษาที่ดีพอ จะทำให้อายุการใช้งานสั้น หรืออาจเสียหายจนใช้งานไม่ได้

4. ไม่ควรเอียงตัวหุ่นยนต์เพราะจะทำให้น้ำกรดของแบตเตอร์รี่ไหลออกมาได้

5. เนื่องจากล้อหน้าและล้อหลังมีความฝืดมากทำให้หุ่นยนต์เคลื่อนที่ไปไม่ตรงพอและหมุนตัวกลับไม่ได้ศูนย์ จึงควรเปลี่ยนใหม่

6. วัสดุที่ใช้ทำแขนยังไม่เบาพอ ถ้าเป็นไปได้ควรจะเปลี่ยนใหม่

แนวทางในการพัฒนาต่อไป

การพัฒนา คือการก้าวทันเทคโนโลยี ในอนาคตหุ่นยนต์ย่อมมีประสิทธิภาพสูงกว่าที่เป็นอยู่อย่างแน่นอน จึงขอเสนอแนวทางในการพัฒนาต่อไปดังนี้

1. พัฒนาทางซอร์ฟแวร์ที่ใช้ควบคุมการเคลื่อนที่ของแแขน และส่วนอื่นให้ดีขึ้น โดยอาศัยหลักการที่ได้เตรียมไว้แล้วหรือพัฒนาวงจรควบคุมให้สามารถทำงานได้เร็วขึ้น และมีแรงบิดมากขึ้นเพื่อใช้หยิบสิ่งของได้ดีขึ้น

2. การพัฒนาให้ส่วนแขนและมอเตอร์มีน้ำหนักเบา อันจะทำให้การเคลื่อนที่ของแขนรวดเร็ว และยกของได้น้ำหนักมากขึ้น

3. พัฒนาความจำของหุ่นยนต์ให้หุ่นยนต์สามารถจำบริเวณที่เคยผ่านไปได้ เช่น ถ้าหุ่นยนต์เดินชนกำแพงด้านหนึ่ง มันจะจำไว้ว่ามีกำแพง และจะไม่เดินชนกำแพงด้านนี้อีก

4. พัฒนาให้หุ่นยนต์มีตา ซึ่งเป็นพวกกล้องโทรทัศน์ ทำให้หุ่นยนต์สามารถแยกแยะสิ่งของได้มากขึ้น

5. พัฒนาให้หุ่นยนต์มีความเป็นตัวของตัวเองมากขึ้น มีความคิดและตัดสินใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ยืน ภู่สุารรณ และ วัฒนา เชียงกูล, " ไมโครโปรเซสเซอร์ ไมโครคอมพิวเตอร์ " พิมพ์ครั้งที่ 1 พ.ศ. 2527
2. "คู่มือ/ เทียบเบอร์ ไอซี" บริษัทซีเอ็ดยูเคชั่น จำกัด พิมพ์ครั้งที่ 1 พ.ศ. 2522
3. "คู่มือ/ เทียบเบอร์ ทรานซิสเตอร์" บริษัทซีเอ็ดยูเคชั่น จำกัด พ.ศ. 2521
4. "การคำนวณวงจรทรานซิสเตอร์" วันชัย คุณากรวงศ์, บัณฑิต บัวบูชา, รัฐวุฒิ ประทุมราช, รุ่งแสง เครือไวศยวรรณ
5. "วิเคราะห์ และออกแบบระบบการควบคุมมอเตอร์" ผศ. โยธิน เปรมปรีณีรัชต์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง
6. Ichiro Kato, "Mechanical Hand Illustrated" Copyright 1982 Servey Japan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้