

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อินเวอร์ทเพนดูลัม

INVERTED PENDULUM



เลขหมู่.....
เลขทะเบียน..... 62642
วัน,เดือน,ปี..... 21 ส.ค. 2549

| |
|-----------------|
| b..... 44625491 |
| i..... |

ปฏิญานិพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเวอร์ทเพนดูลัม

INVERTED PENDULUM



**ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงาน ปีการศึกษา 2548

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง อินเวอร์ทเพนดูลัม

INVERTED PENDULUM

ผู้จัดทำ

1. นาย กิตติพงษ์ วงศ์ชัย 46015254
2. นาย จิราวุธ อัมพัฒน์ 46015256

(.....) อาจารย์ที่ปรึกษา
ผศ.ดร. ยุทธนา คัดใจเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเวอร์ทเพนดูลัม

กิตติพงษ์ วงศ์ชัย 46015254
จิราวุธ อัมพัฒน์ 46015256
อ.ยุทธนา คิคใจเด็ว อาจารย์ที่ปรึกษา
ปีการศึกษา 2548

บทคัดย่อ

ในปฏิญานิพนธ์ฉบับนี้เป็นการศึกษา ระบบอินเวอร์ทเพนดูลัมบนรถ โดยนำเสนอการควบคุมการเหวี่ยงขึ้น และการรักษาเสถียรภาพของอินเวอร์ทเพนดูลัม โดยที่ตัวควบคุมฟัซซี่ (fuzzy control) ได้ถูกออกแบบเพื่อควบคุมการเหวี่ยงให้ก้านเพนดูลัมขึ้นจากจุดสมดุลต่าง แล้วพยายามรักษาเสถียรภาพ โดยจะใช้การชดเชยแรงแบบไม่เป็นเชิงเส้น(non-linear)ซึ่งจะเป็นการสร้างชิ้นงานระบบอินเวอร์ทเพนดูลัมบนรถ และสามารถควบคุมการเหวี่ยงขึ้นโดยผู้ควบคุมได้ แล้วนำชิ้นงานนี้ใช้ในการศึกษาระบบการควบคุมแบบต่างๆ เช่น ฟัซซี่ (fuzzy control) PD,PID control ตลอดจน นิวรอล(neural control) เพื่อให้เข้าใจการทำงานจริงของระบบ

INVERTED PENDULUM

Mr. KITTIPONG WONGCHAI

Mr. JIRAWUT IMPAT

Dr. YUTTANA KIDJAIDEAV(advisor)

Education Year 2005

ABSTRACT

This report presents the Inverted Pendulum on Cart. involveol in swinging up and stabilizing the inverted pendulum on cart system. Fuzzy control designed for swinging up the pendulum from the bottom and try to stabilize by nonlinear control. We setup hardware of inverted pendulum system and swinging up by manual. We use this search to study about .So we can understand the real system. The cart position control is employed various control system such as fuzzy control , PID control sol Neural Control . in swinging up procedure while the state feedback servo controller designed by pole placement technique is used to stabilize the pendulum at the upright position. The experimental results verify effectiveness of the proposed control scheme in attracting the pendulum angle to the upright position and maintaining it.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก อาจารย์
ยุทธนา คิทธิใจเดียว ซึ่งเป็นอาจารย์ผู้ควบคุมโครงงาน อาจารย์ประภากร สุวรรณะ พิสองเมือง
นันทะขว้าง ข้าพเจ้ารู้สึก ซาบซึ้ง ในความอนุเคราะห์จากท่านทั้งสามท่าน และขอขอบพระคุณ
เป็นอย่างสูง

ขอกราบพระคุณคณาจารย์ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบัน
เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมอิเล็กทรอนิกส์ สถาบันเทคโนโลยี
พระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็น
กำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำปริญญาานิพนธ์ฉบับนี้สำเร็จ
สุด่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากปริญญาานิพนธ์ฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุก
ท่าน

นาย กิตติพงษ์ วงศ์ชัย

นาย จิราวุธ อิมพัฒน์

สารบัญ

| เรื่อง | หน้า |
|---|------|
| บทคัดย่อ | I |
| ABSTRACT | II |
| กิตติกรรมประกาศ | III |
| สารบัญ | IV |
| สารบัญตาราง | V |
| สารบัญรูปภาพ | VI |
| บทที่ 1 บทนำ | 1 |
| บทที่ 2 ทฤษฎีและหลักการ | 3 |
| 2.1 หลักการในการควบคุม | 3 |
| 2.2 โครงสร้างของระบบ | 5 |
| 2.2.1 โครงสร้างและส่วนประกอบอินเวอร์ตเพนดูลัม | 5 |
| 2.2.2 ชุดควบคุมและเซนเซอร์ | 6 |
| 2.3 มอเตอร์ไฟฟ้ากระแสตรง | 7 |
| 2.3.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง | 7 |
| 2.3.2 พื้นฐานของระบบควบคุมมอเตอร์กระแสตรง | 8 |
| 2.4 แบบจำลองทางคณิตศาสตร์ของระบบอินเวอร์ตเพนดูลัมบนรถ | 9 |
| 2.5 การ interface แบบต่างๆ | 16 |
| บทที่ 3 รายละเอียดในการออกแบบและการสร้าง | 23 |
| 3.1. ส่วนของชิ้นงาน | 23 |
| 3.1.1. วิเคราะห์และออกแบบ | 23 |
| 3.1.2. จัดหาอุปกรณ์ | 23 |
| 3.1.3. สร้างชิ้นงาน | 23 |
| 3.1.4. การทดลอง | 23 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|--|----|
| 3.2 ส่วนของระบบควบคุม และโปรแกรม | 29 |
| 3.2.1 วงจรเชื่อมต่อระหว่างคอมพิวเตอร์และอุปกรณ์ภายนอก | 29 |
| 3.2.2 การใช้งานแบบเปิด-ปิด และปรับความกว้างพัลส์ (PWM) ด้วยไมโครคอนโทรลเลอร์ | 29 |
| 3.2.3 วงจรชุดขับมอเตอร์ | 30 |
| 3.2.4 วงจรคอมพาราเตอร์ | 32 |
| | |
| บทที่ 4 การวิเคราะห์และทดสอบ | 34 |
| 4.1 การทดสอบตัวรถและราง | 35 |
| 4.2 การทดลองเพื่อหาความเร็วของมอเตอร์ที่เหมาะสม | 36 |
| 4.3 ผลการทดลอง Interface ผ่าน printer port (parallel port) | 38 |
| 4.4 ผลการทดลอง Interface ผ่าน PCI card | 38 |
| | |
| บทที่ 5 สรุปผลและวิจารณ์ | 39 |
| 5.1 บทสรุป | 39 |
| 5.2 ปัญหาที่พบ | 40 |
| 5.3 แนวทางในการแก้ปัญหาและพัฒนา | 40 |
| 5.3.1 สร้างวงจรมัดจากภายนอก | 40 |
| 5.3.2 การส่งผ่านข้อมูลโดยการอินเตอร์รัพท์ | 40 |
| 5.3.3 การสร้างสัญญาณอินเตอร์รัพท์จากภายนอก | 40 |
| 5.3.4 สร้างตัวควบคุมระบบอินเวอร์ทเพนดูลัม | 40 |
| 5.3.5 ประยุกต์ นำระบบอินเวอร์ทเพนดูลัม ไปใช้งาน | 40 |

บรรณานุกรม

ภาคผนวก ก

ภาคผนวก ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

| ตารางที่ | หน้า |
|--|------|
| ตารางที่ 4.1 แสดงว่าก้านเพนคูล์มใช้เวลาจากจุดสมมูลบน | 35 |
| ตารางที่ 4.2 ผลการทดลอง Interface ผ่าน PCI card | 38 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

| | หน้า |
|--|------|
| รูปที่ 2.1 แสดงระบบอินเวอร์ทเพนคูล์มบนรถ | 4 |
| รูปที่ 2.2 แสดงบล็อกไดอะแกรมโดยรวมของระบบ | 4 |
| รูปที่ 2.3 รถอินเวอร์ทเพนคูล์ม | 5 |
| รูปที่ 2.4 ส่วนประกอบพื้นฐานของระบบควบคุมคิซิมอเตอร์ | 9 |
| รูปที่ 2.5 แบบจำลองของระบบอินเวอร์ทเพนคูล์มบนรถ | 9 |
| รูปที่ 2.6 Fuzzy Function ของระบบ | 11 |
| รูปที่ 2.7 Membership function ของ Input (1) | 12 |
| รูปที่ 2.8 Membership function ของ Input (2) | 13 |
| รูปที่ 2.9 Membership function ของ Output | 13 |
| รูปที่ 2.10 Rule Editor ของระบบ | 15 |
| รูปที่ 2.11 การ Interface ผ่าน Printer port | 16 |
| รูปที่ 2.12 รายละเอียดของ Printer port | 17 |
| รูปที่ 2.13 Address ที่ F300 HEX | 18 |
| รูปที่ 2.14 Address ที่ F400 HEX | 19 |
| รูปที่ 2.15 Address ที่ F500 HEX | 19 |
| รูปที่ 2.16 PCI Pin list | 21 |
| รูปที่ 2.17 Inport/Output | 21 |
| รูปที่ 2.18 รายละเอียดของ PCI Input/Output | 22 |
| รูปที่ 3.1 แสดงแนวคิดในการหาค่า พัลส์ เพื่อนำไปควบคุมมอเตอร์ | 24 |
| รูปที่ 3.2 แสดงด้านข้างของตัวรถ | 25 |
| รูปที่ 3.3 แสดงด้านหน้า | 25 |
| รูปที่ 3.4 แสดงด้านหลัง | 26 |
| รูปที่ 3.5 แสดงโครงด้านซ้าย | 26 |
| รูปที่ 3.6 แสดงโครงด้านขวา | 27 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|-------------|--|----|
| รูปที่ 3.7 | แสดงเฟืองและโซ่ | 27 |
| รูปที่ 3.8 | แสดงตัววาง | 28 |
| รูปที่ 3.9 | แสดงก้านเพนดูลัม | 28 |
| รูปที่ 3.10 | แสดงการใช้โปรแกรมควบคุมมอเตอร์ผ่าน MCS-51 2 ตัว | 29 |
| รูปที่ 3.11 | แสดงวงจรชุดขับมอเตอร์ | 30 |
| รูปที่ 3.12 | แสดงวงจรชุดขับมอเตอร์ | 31 |
| รูปที่ 3.13 | แสดงวงจรวงจรคอมพาราเตอร์ | 32 |
| รูปที่ 3.14 | แสดงวงจรวงจรคอมพาราเตอร์ | 33 |
| รูปที่ 4.1 | แสดงการปล่อยก้านเพนดูลัมจากจุดบน | 34 |
| รูปที่ 4.2 | แสดงรูปคลื่น | 36 |
| รูปที่ 4.3 | กราฟแสดงความสัมพันธ์ระหว่างค่าความถี่ที่เกิดแรงดันที่มอเตอร์ | 36 |
| รูปที่ 4.4 | แสดงการวิเคราะห์หุ้มในโหมบของพัลส์ | 37 |
| รูปที่ 4.4 | สัญญาณจากเอนโค้ดเดอร์ 1 | 37 |
| รูปที่ 5.1 | แสดงมุมในการแกว่งของก้านเพนดูลัม | 39 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

เมื่อก้าวถึงระบบที่ใช้ในการทดสอบประสิทธิภาพของตัวควบคุม จะเห็นว่าระบบอินเวอร์ทเพนดูลัมเป็นระบบหนึ่งที่นิยมใช้กัน เนื่องจากเป็นระบบที่มีข้อดีหลายข้อ โดยเฉพาะมีโครงสร้างที่ง่าย มีความไม่เป็นเชิงเส้นและความไม่แน่นอนของคุณลักษณะของระบบ และเมื่อก้าวถึงระบบอินเวอร์ทเพนดูลัมที่สามารถเหวี่ยงเพนดูลัมจากจุดสมดุลล่างขึ้นสู่จุดสมดุลบนและรักษาเสถียรภาพ ณ จุดสมดุลบนไว้ได้นั้น จะเห็นได้ว่าเป็นระบบที่น่าศึกษาว่าควรจะใช้ตัวควบคุมแบบใดที่สามารถควบคุมระบบอินเวอร์ทเพนดูลัมแบบธรรมดาที่ไม่สามารถเหวี่ยงเพนดูลัมขึ้นมาจากจุดสมดุลล่างเองได้อย่างมีประสิทธิภาพ มีตัวควบคุมต่างๆ ที่ถูกนำไปทดสอบกับระบบดังกล่าว เช่นตัวควบคุมแบบโครงข่ายประสาท (neural network controller) [1] ตัวควบคุมฟัซซี่ (fuzzy controller) [2] ตัวควบคุมโรบัสต์ (robust controller) [3] ตัวควบคุมแบบไม่เป็นเชิงเส้น (nonlinear controller) [4] และตัวควบคุมอาศัยหลักการพลังงาน (controller by energy-based methods) [5] ซึ่งหลักการออกแบบตัวควบคุมเหล่านี้จะมีหลักการค่อนข้างซับซ้อน โดยเฉพาะตัวควบคุมแบบไม่เป็นเชิงเส้น

รายงานฉบับนี้จึงนำเสนอหลักการออกแบบตัวควบคุมการเหวี่ยงขึ้นและรักษาเสถียรภาพของระบบอินเวอร์ทเพนดูลัมบนรูปทรงหนึ่งซึ่งเป็นหลักการที่ง่าย ไม่ยุ่งยาก และสามารถใช้งานได้ง่าย โดยตัวควบคุมในการควบคุมการเหวี่ยงขึ้นและรักษาเสถียรภาพของอินเวอร์ทเพนดูลัมถูกแบ่งออกเป็นสองส่วน ส่วนแรกเป็นตัวควบคุมการเหวี่ยงขึ้นของเพนดูลัมโดยใช้ตัวควบคุมแบบตัวควบคุมฟัซซี่ (fuzzy controller) และทำหน้าที่ในการควบคุมตำแหน่ง ตัวควบคุมนี้จะใช้หลักการของการควบคุมตำแหน่งให้เคลื่อนรถ ไปมาเป็นลำดับตามเงื่อนไขของตำแหน่งรถและความเร็วเชิงมุมของเพนดูลัม โดยที่รถรางจะเปลี่ยนตำแหน่งจากตำแหน่งลำดับที่หนึ่งไปยังตำแหน่งลำดับที่สอง ได้ก็ต่อเมื่อตัวรถรางอยู่ในตำแหน่งลำดับที่หนึ่งและความเร็วเชิงมุมของเพนดูลัมเท่ากับศูนย์ การเคลื่อนที่ของรถไปมาเป็นลำดับตามเงื่อนไขดังกล่าวนี้จะสร้างแรงเหวี่ยงจนเพนดูลัมขึ้นมาอยู่ ณ จุดสมดุลบนได้ และตัวควบคุมส่วนที่สองคือตัวควบคุมอินเวอร์ทเพนดูลัมให้มีเสถียรภาพ ณ จุดสมดุลบนโดยใช้ตัวควบคุมเซอร์โวป้อนกลับสถานะ (servo state feedback)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมทั้งสองจะทำงานร่วมกันดังนี้ โดยในลำดับแรกจะใช้ตัวควบคุมการเหวี่ยงขึ้นเพื่อเหวี่ยงเพนดูลัมมาอยู่บริเวณจุดสมดุลบนก่อน แล้วจึงเปลี่ยนมาใช้ตัวควบคุม เซอร์โวป้อนกลับสถานะเพื่อรักษาเสถียรภาพของเพนดูลัม ณ จุดสมดุลบนไว้

1.2 วัตถุประสงค์ของโครงการ

1. สร้างชุดทดลองระบบอินเวอร์ทเพนดูลัมบนรถสำหรับใช้ในการทดสอบประสิทธิภาพของตัวควบคุมการเหวี่ยงขึ้นและรักษาเสถียรภาพของอินเวอร์ทเพนดูลัมบนรถที่ทำการออกแบบ
2. ศึกษาการออกแบบตัวควบคุมสำหรับระบบอินเวอร์ทเพนดูลัมบนรถให้สามารถเหวี่ยงเพนดูลัมจากจุดสมดุลล่างขึ้นสู่จุดสมดุลบน และรักษาเสถียรภาพของเพนดูลัม ณ จุดสมดุลบนไว้
3. ออกแบบตัวควบคุม และนำไปควบคุมการเหวี่ยงขึ้นและรักษาเสถียรภาพของอินเวอร์ทเพนดูลัมบนรถให้สามารถเหวี่ยงเพนดูลัมจากจุดสมดุลล่างขึ้นสู่จุดสมดุลบน และรักษาเสถียรภาพของเพนดูลัม ณ จุดสมดุลบนไว้
4. ศึกษาถึงปัญหาที่เกิดขึ้นในโครงการ และแนวทางแก้ไข รวมถึงการพัฒนาระบบต่อไป

1.3 ขอบเขตของโครงการ

1. ออกแบบและสร้างชุดทดลองระบบอินเวอร์ทเพนดูลัมบนรถสำหรับใช้ในการทดสอบประสิทธิภาพของตัวควบคุมที่ทำการออกแบบ
2. ศึกษาโครงสร้าง ของระบบ อินเวอร์ทเพนดูลัมบนรถ โดย อาศัยฟัซซี่ (fuzzy controller) เพื่อใช้ในการ ออกแบบตัวควบคุม
3. ออกแบบและใช้ตัวควบคุมทำการทดลองการเหวี่ยงขึ้นและรักษาเสถียรภาพของระบบอินเวอร์ทเพนดูลัมบนรถให้สามารถเหวี่ยงเพนดูลัมจากจุดสมดุลล่างขึ้นสู่จุดสมดุลบน และรักษาเสถียรภาพของเพนดูลัม ณ จุดสมดุลบนไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีพื้นฐานที่ใช้ในโครงการ

ระบบอินเวอร์ทเพนดูลัมเป็นระบบที่ได้รับความนิยมในการนำมาทดสอบประสิทธิภาพของตัวควบคุมต่างๆ เพราะตัวระบบอินเวอร์ทเพนดูลัมมีโครงสร้างที่ง่ายไม่ซับซ้อนไม่มีเสถียรภาพ ณ จุดสมดุลบนและมีความไม่แน่นอนของคุณลักษณะของระบบ เป็นต้น ระบบอินเวอร์ทเพนดูลัมถูกแบ่งเป็นสองชนิดใหญ่ๆ คือ อินเวอร์ทเพนดูลัมแบบฐานหมุนและอินเวอร์ทเพนดูลัมบนรถ ในรายงานนี้เลือกใช้ระบบอินเวอร์ทเพนดูลัมบนรถในการศึกษาและทดลองตัวควบคุมที่ได้นำเสนอในรายงานฉบับนี้ เนื่องจากระบบอินเวอร์ทเพนดูลัมบนรถมีโครงสร้างที่ง่ายและไม่จำเป็นต้องใช้มอเตอร์ที่มีกำลังสูงและขนาดใหญ่มากนักในการขับเคลื่อนเมื่อเปรียบเทียบกับระบบอินเวอร์ทเพนดูลัมแบบฐานหมุน

การหาความเร็วของเพนดูลัมมีสูตรการคำนวณดังนี้

$$v = l\omega \quad (2.1)$$

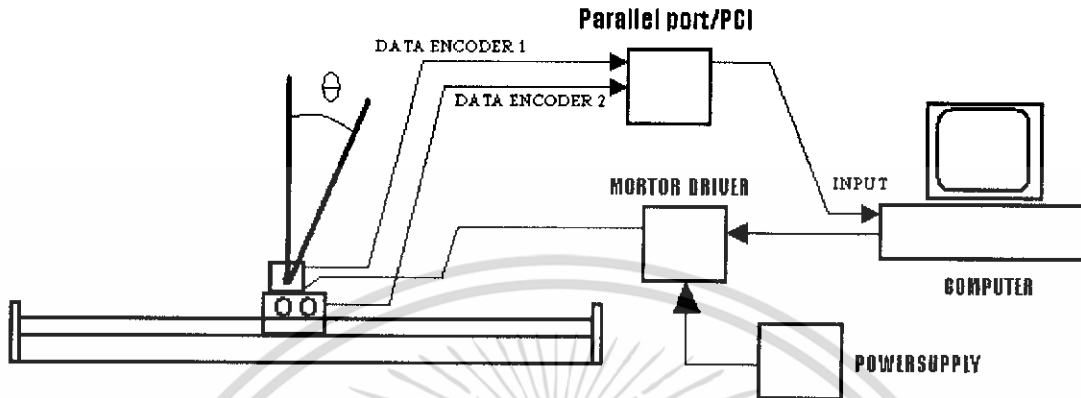
- เมื่อ ω คือ $d\theta / dt$
 v คือ ความเร็วของเพนดูลัม
 l คือ ความยาวของก้านเพนดูลัม
 ω คือ ความเร็วเชิงมุม

2.1 หลักการในการควบคุม

ในการควบคุมระบบการเคลื่อนที่ของรถอินเวอร์ทเพนดูลัมให้ได้ตามต้องการนั้น การควบคุมที่เป็นสิ่งสำคัญที่สุด คือ การควบคุมการหมุนของมอเตอร์ซึ่งเป็นตัวการสำคัญที่ทำให้รถเคลื่อนที่ไปด้วยความเร็วและทิศทางที่ถูกต้องและสัมพันธ์กับทิศทางการแกว่งตัวของก้านเพนดูลัม เพื่อให้ก้านเพนดูลัมอยู่ในตำแหน่งตั้งฉากตลอดเวลา

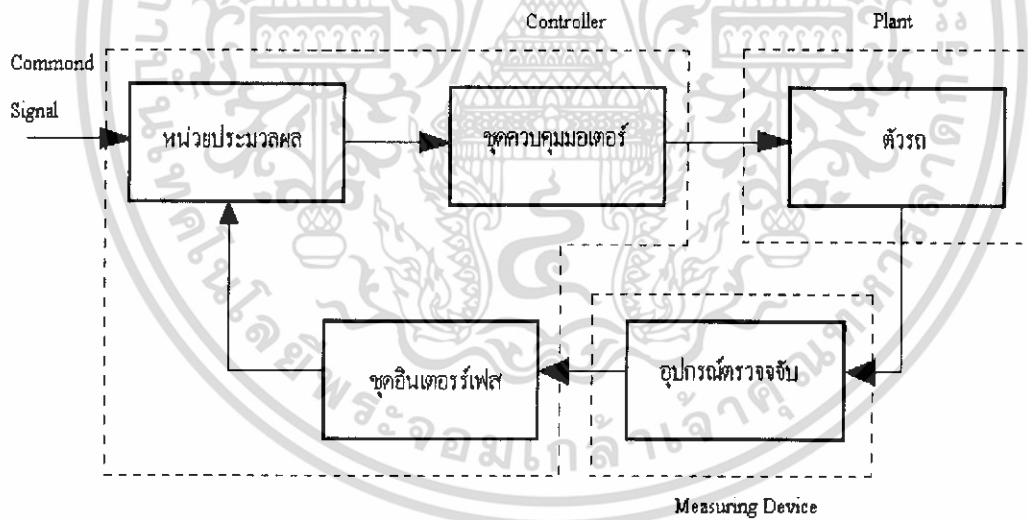
การควบคุมการทำงานของมอเตอร์เป็นการควบคุมแบบดิจิตอล (Digital Control) ซึ่งได้จากการทำงานร่วมกันของ คอมพิวเตอร์ ชุดควบคุม และเซนเซอร์ (Sensor) เซนเซอร์จะทำการวัดสัญญาณป้อนกลับ (Feedback Signal) ค่าตำแหน่งของตัวรถ และค่ามุมการแกว่งของตัวก้านอินเวอร์ทเพนดูลัม แล้วนำค่าที่ได้มาคำนวณหา ค่าสเตท(State) ต่างๆ และค่าสัญญาณควบคุม (Control signal) หรือสัญญาณความเร็วอ้างอิง (Velocity Reference Command) เพื่อส่งไปยังชุดควบคุมการหมุนของมอเตอร์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ระบบอินเวอร์ทเพนดูลัมบนรถ

บล็อกไดอะแกรมของระบบแสดงได้ดังรูป



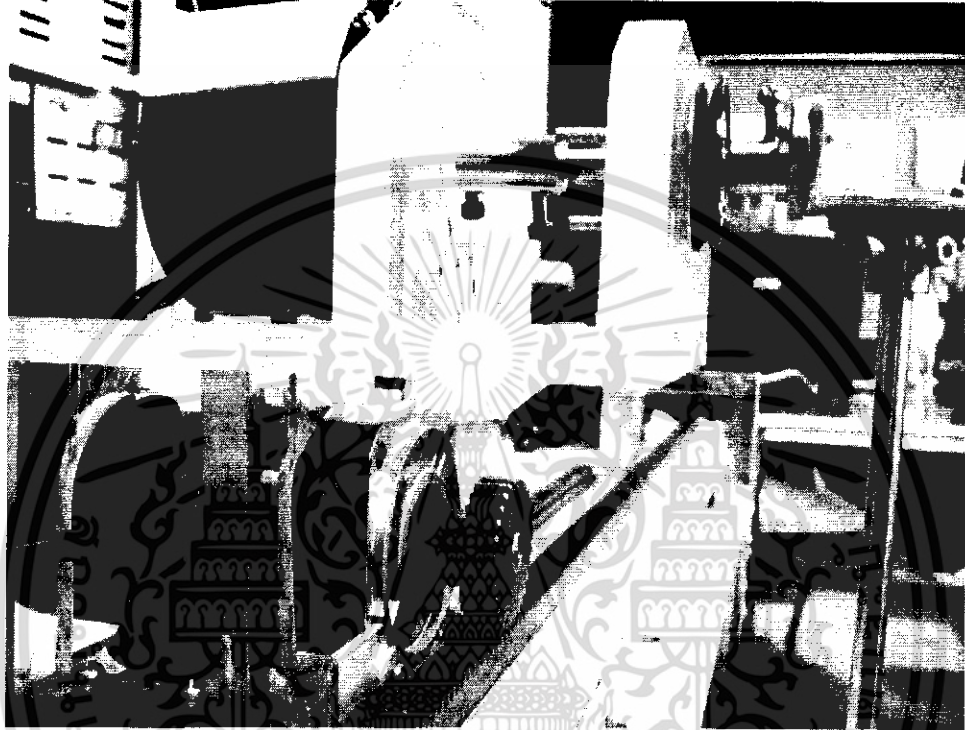
รูปที่ 2.2 แสดงบล็อกไดอะแกรมโดยรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โครงสร้างของระบบ

2.2.1 โครงสร้างและส่วนประกอบรถอินเวอร์ทเพนดูลัม

ตัวรถถูกออกแบบให้มีการเคลื่อนที่บนรางดังแสดงดังรูป



รูปที่ 2.3 รถอินเวอร์ทเพนดูลัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ชุดควบคุมและเซนเซอร์

สามารถแบ่งได้เป็น 3 ส่วนคือ

ส่วนที่ 1 ส่วนตรวจจับ (Sensor)

เซนเซอร์เป็นส่วนประกอบที่สำคัญในระบบคอนโทรลที่มีการป้อนกลับเพราะสัญญาณที่ป้อนกลับมาจากเซนเซอร์นั้นสามารถนำมาวิเคราะห์หาคุณสมบัติของระบบหรือสามารถบอกสถานะของระบบขณะนั้นได้ว่าเป็นอย่างไร ซึ่งจะได้นำมาหาสัญญาณควบคุมที่จะส่งไประบบควบคุม ดังนั้นถ้าสัญญาณป้อนกลับที่ได้มีความถูกต้องมากและมีสัญญาณรบกวนน้อยก็จะได้ค่าสัญญาณควบคุมที่สามารถควบคุมระบบให้เกิดประสิทธิภาพสูงสุดได้ สำหรับโครงการนี้ประกอบไปด้วยเซนเซอร์ 2 ตัว คือ เอนโคเดอร์ตัวที่ 1 (Encoder 1) ที่ติดกับมอเตอร์ใช้ในการตรวจจับทิศทาง การเคลื่อนที่และความเร็วของรถ และเอนโคเดอร์ตัวที่ 2 (Encoder 2) ใช้ในการวัดค่ามุมที่เบี่ยงเบนไปของก้านเพนดูลัม แล้วสัญญาณที่วัดได้ทั้งเอนโคเดอร์ตัวที่ 1 และ เอนโคเดอร์ตัวที่ 2 จะถูกส่งกลับไปยังหน่วยประมวลผลกลางและแสดงผล ซึ่งก็คือคอมพิวเตอร์ต่อไป

ส่วนที่ 2 หน่วยประมวลผลกลางและควบคุม

ได้แก่ คอมพิวเตอร์ส่วนบุคคล (Personal Computer, PC) และการอินเตอร์เฟสจาก Microcontroller เพราะสามารถนำมาต่อใช้งานร่วมกับอุปกรณ์ภายนอกต่างๆ ได้ง่าย โดยจะใช้ MCS-51 เป็นตัวอ่านค่าพัลส์จากเอนโคเดอร์ทำให้รู้ตำแหน่งและความเร็วของรถ คอมพิวเตอร์จะแสดงค่าที่รับมาจากเซนเซอร์ผ่านทางหน้าจอ และ คอมพิวเตอร์จะประมวลผลแล้วส่งสัญญาณควบคุมไปยังวงจรจับมอเตอร์ ต่อไป

ส่วนที่ 3 เป็นวงจรจับมอเตอร์

เป็นวงจรที่ควบคุมการหมุนมอเตอร์ซึ่งเป็นอุปกรณ์ที่ทำให้รถเคลื่อนที่ โดยรายละเอียดของวงจรจับมอเตอร์จะได้กล่าวถึงต่อไป

2.3 มอเตอร์ไฟฟ้ากระแสตรง

2.3.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงเป็นทรานส์ดิวเซอร์แรงบิด ซึ่งมีการออกแบบให้มีการออกแบบคุณลักษณะพิเศษ คือ แรงบิดของเพลามอเตอร์ไฟฟ้ากระแสตรงกับกระแสอาร์เมเจอร์ แรงบิดของเพลามอเตอร์จะได้จากผลระหว่างสนามแม่เหล็กและขดลวดตัวนำ หลักการในที่นี้กระแสที่ไหลในขดลวดตัวนำจะสร้างฟิลด์ที่ประกอบด้วยเส้นแรงแม่เหล็ก และขดลวดตัวนำเหล่านั้นอยู่ห่างจากจุดศูนย์กลางการหมุนเท่ากับ r ความสัมพันธ์ระหว่างแรงบิดของเพลากับกระแสเท่ากับ

$$T = K \phi I \quad (2.2)$$

เมื่อ T คือ แรงบิดของเพลามีหน่วยเป็น นิวตัน-เมตร
 ϕ คือ เส้นแรงแม่เหล็ก มีหน่วยเป็น เวเบอร์
 I คือ กระแส มีหน่วยเป็น แอมแปร์
 K คือ ค่าคงที่

ดังนั้น แรงบิดของเพลาก็จะเป็นสัดส่วนโดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแส เมื่อ ขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กก็จะทำให้เกิด โวลต์เตจตกคร่อมตัวมันเอง โวลต์เตจนี้จะเป็นสัดส่วนกับความเร็วของเพลามอเตอร์และด้านกรไหลของกระแส ความสัมพันธ์ระหว่างโวลต์เตจย้อนกลับนี้และความเร็วของเพลามอเตอร์ คือ

$$E = K \phi W \quad (2.3)$$

เมื่อ E คือ โวลต์เตจย้อนกลับ มีหน่วยเป็น โวลต์
 ϕ คือ เส้นแรงแม่เหล็ก มีหน่วยเป็น เวเบอร์
 W คือ ความเร็วของมอเตอร์ มีหน่วยเป็น เรเดียน/วินาที

2.3.2 พื้นฐานของระบบควบคุมมอเตอร์กระแสตรง

ส่วนประกอบพื้นฐานของระบบควบคุมมอเตอร์กระแสตรงประกอบด้วยบล็อกที่สำคัญ 4 บล็อก คือ

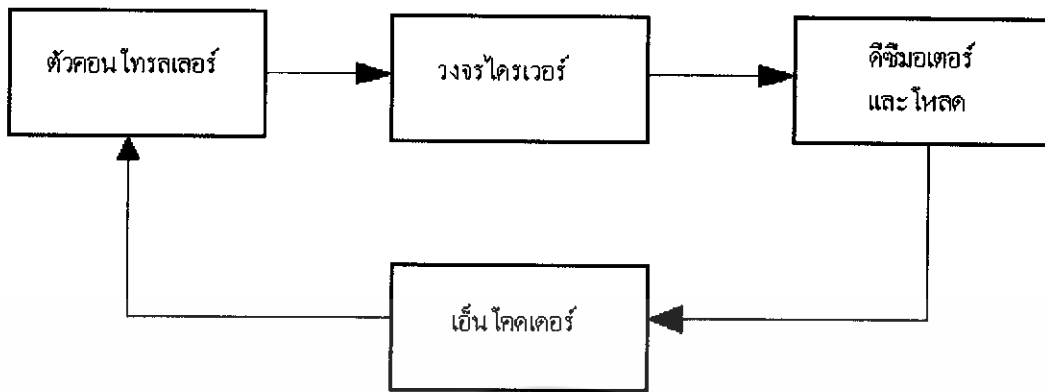
1. ตัวคอนโทรลเลอร์
2. วงจรไครเวอร์หรือเพาเวอร์แอมพลิไฟร์
3. ฟีดแบ็คทรานสดิวเซอร์หรือเอนโคเดอร์
4. ดีซีมอเตอร์และโหลด

ตัวคอนโทรล เป็นส่วนของระบบที่ทำให้เกิดสัญญาณคอนโทรลไปยังคัตติงมอเตอร์ และโหลดคอนโทรลเลอร์ที่ให้สัญญาณคอนโทรลเป็นสัญญาณอนาล็อกเราเรียกว่าอนาล็อกคอนโทรลเลอร์ ส่วนคอนโทรลเลอร์ที่ให้สัญญาณคอนโทรลเป็นสัญญาณดิจิทัล เราเรียกว่าดิจิทัลคอนโทรลเลอร์

วงจรไครเวอร์ เป็นส่วนประกอบของระบบที่อยู่ระหว่างตัวคอนโทรลเลอร์กับดีซีมอเตอร์และโหลดมีหน้าที่ปรับรูปและขยายสัญญาณให้เหมาะสมก่อนที่จะป้อนเข้าไปขับเคลื่อนมอเตอร์และโหลดวงจรไครเวอร์ส่วนใหญ่ ได้แก่เพาเวอร์แอมพลิไฟร์ซึ่งอาจแบ่งแยกเป็นลิเนียร์เพาเวอร์แอมพลิไฟร์และพัลส์วิดธมอดูเลชัน แอมพลิไฟร์

ฟีดแบ็คทรานสดิวเซอร์หรือเอนโคเดอร์ เป็นสิ่งประดิษฐ์ที่ใช้รับรู้หรือตรวจจับสัญญาณเอาท์พุทที่ต้องการ โดยไม่มีผลของโหลดคั้ง (loading) สัญญาณที่ตรวจจับได้นี้จะป้อนกลับไปเปรียบเทียบกับสัญญาณอ้างอิงทำให้ได้สัญญาณที่ผิดพลาด (error) ฟีดแบ็คทรานสดิวเซอร์แบ่งออกได้เป็น 2 แบบคืออนาล็อกทรานสดิวเซอร์คือสิ่งประดิษฐ์ใช้เปลี่ยนพลังงานรูปหนึ่งให้เป็นสัญญาณอนาล็อก ได้แก่ พวกลทาโคเจนเนอเรเตอร์ โพลเทนทิโอมิเตอร์และซิงโคร เป็นต้น ส่วนฟีดแบ็คทรานสดิวเซอร์อีกแบบหนึ่งคือ ดิจิตอลทรานสดิวเซอร์เป็นสิ่งประดิษฐ์ที่ใช้เปลี่ยนพลังงานรูปหนึ่งให้เป็นสัญญาณดิจิทัล ได้แก่ พวกรินคริเมนต์ เอนโคเดอร์ รีโซลเวอร์ แมกเนติกพิกอัพ เป็นต้น

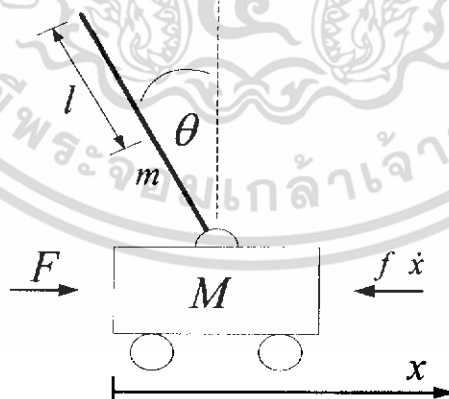
ดีซีมอเตอร์และโหลด คือระบบที่ถูกคอนโทรลหรือส่วนที่ออกแรงทำงานซึ่งจะเป็นเครื่องจักรกล (ดีซีมอเตอร์) หรืออะไรก็ตามที่ให้ตัวแปรดีซีมอเตอร์ในที่นี้เป็นแบบแม่เหล็กถาวรที่มีคุณสมบัติการทำงานสูง มีอาร์เมเจอร์อินดักแตนซ์และแรงเฉื่อยของโรเตอร์ต่ำ



รูปที่ 2.4 ส่วนประกอบพื้นฐานของระบบควบคุมดีซีมอเตอร์

2.4 แบบจำลองทางคณิตศาสตร์ของระบบอินเวอร์ทเพนดูลัมบนรถ

การหาแบบจำลองทางคณิตศาสตร์หรือสมการการเคลื่อนที่ของระบบอินเวอร์ทเพนดูลัมบนรถนั้น ใช้วิธีการของฟัซซี่ (fuzzy controller) ภายใต้สมมุติฐานที่ว่าแท่งเพนดูลัมแข็งไม่สามารถงอได้ และแรงเสียดทานสถิตมีค่าน้อยมากจนไม่มีอิทธิพลต่อระบบ ซึ่งวิธีการของฟัซซี่ (fuzzy controller) นี้เป็นวิธีที่สะดวกในการหาแบบจำลองทางคณิตศาสตร์ของระบบ ดังนั้น จึงไม่จำเป็นต้องพิจารณาทิศทางเคลื่อนที่และทิศทางของแรงในระบบมากนัก แบบจำลองของระบบอินเวอร์ทเพนดูลัมบนรถแสดงดังรูปที่ 2.3



รูปที่ 2.5 แบบจำลองของระบบอินเวอร์ทเพนดูลัมบนรถ

ในการแก้ปัญหาที่เราใช้ ฟัซซี่ (fuzzy controller) เป็นตัวควบคุมระบบ Invert Pendulum

โดยใช้ Differential Equation เป็นตัวอธิบายพฤติกรรมของระบบ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$-ml^2 \frac{d^2\theta}{dt^2} + (m l g) \sin \theta = \tau = u(t) \quad (2.4)$$

โดยที่ m คือ น้ำหนักของ Pendulum

L คือ ความยาวของ Pendulum

θ คือ การเปลี่ยนแปลงของมุมจากแนวตั้งตามเข็มนาฬิกา

$u(t)$ คือ torque หรือแรงที่ควบคุมทิศทางการเคลื่อนที่ของ pole ในทิศทางทวนเข็มนาฬิกา

t คือ เวลา

g คือ ความเร่งซึ่งเป็นค่าคงที่

สมมุติให้ $x_1 =$ มุม (θ) และ $x_2 =$ ความเร็วเชิงมุม ($d\theta/dt$)

ดังนั้นจากสมการที่ 1 จะได้ว่า

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = \left(\frac{g}{l}\right) \sin(x_1) - \left(\frac{1}{ml^2}\right) u(t)$$

เนื่องจากการเปลี่ยนแปลงของมุมมีค่าน้อย ดังนั้นจึงสามารถประมาณได้ว่า $\sin \theta = \theta$ โดยมี θ

ถูกวัดในหน่วยของ Radians

จะได้ว่า

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = \left(\frac{g}{l}\right) (x_1) - \left(\frac{1}{ml^2}\right) u(t) \quad (2.5)$$

โดยเลือก

$$l = g$$

$$m = \frac{180}{(\pi \cdot g^2)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ 2 สามารถแสดงได้ด้วย Matrix Differential Equation

$$x_1(k+1) = x_1(k) + x_2(k) \quad (2.6)$$

$$x_2(k+1) = x_1(k) + x_2(k) - u(t) \quad (2.7)$$

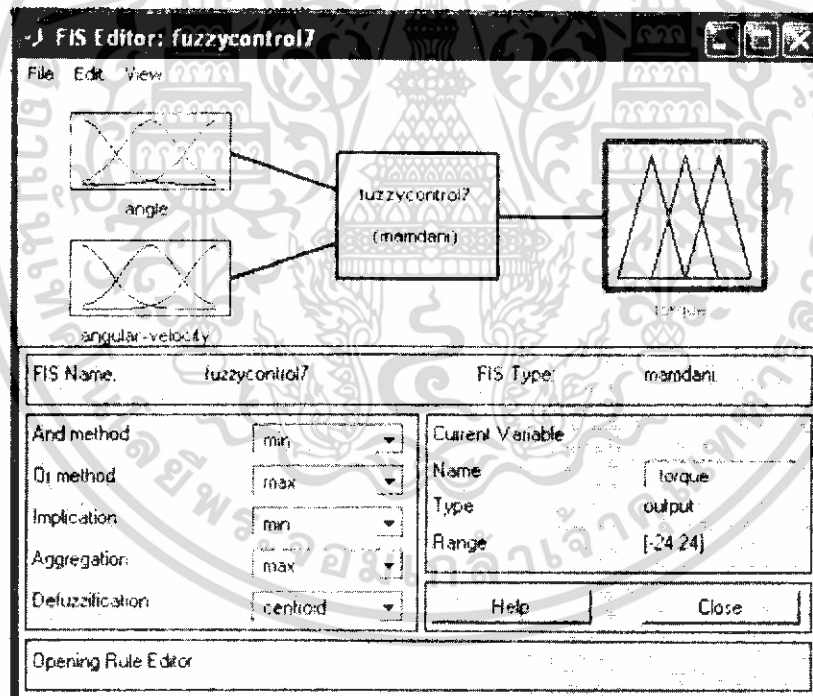
สำหรับในปัญหานี้เรากำหนดให้ Input ประกอบด้วย 2 ตัวแปร คือ

Input (1) มุมที่เปลี่ยนแปลงในแนวตั้ง มีขอบเขตระหว่าง[-4,4]

Input (2) ความเร็วเชิงมุม [-5,5] (หน่วยคือ Degree per second)

ขั้นตอนในการสร้างระบบควบคุม

2.4.1.สร้าง Fuzzy Function ของระบบด้วยคำสั่ง fuzzy ใน MATLAB จะได้ dialog ดังรูป



รูปที่ 2.6 Fuzzy Function ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วทำการกำหนดชื่อและค่าต่างๆที่ใช้งานจะได้ Fuzzy Function ที่มีคุณสมบัติดังนี้

Name : 'fuzzy control 17'

Type : 'mandani'

and Method : 'min'

or Method : 'max'

defuzzMethod : 'centroid'

impMethod : 'min'

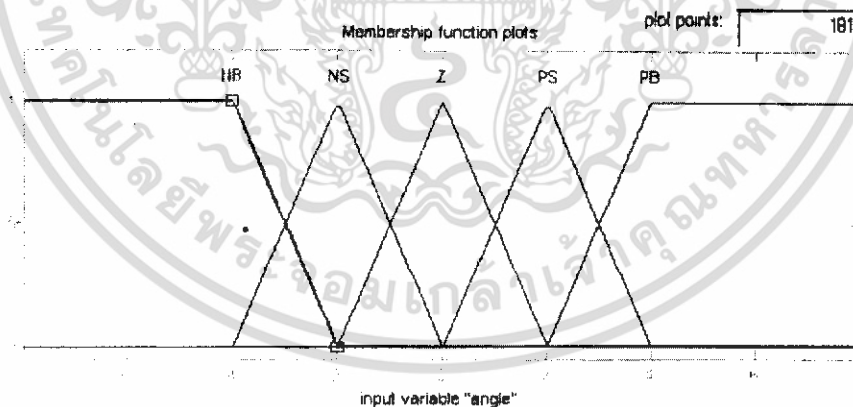
aggMethod : 'max'

input: [1x2 Struct]

output : [1x1 Struct]

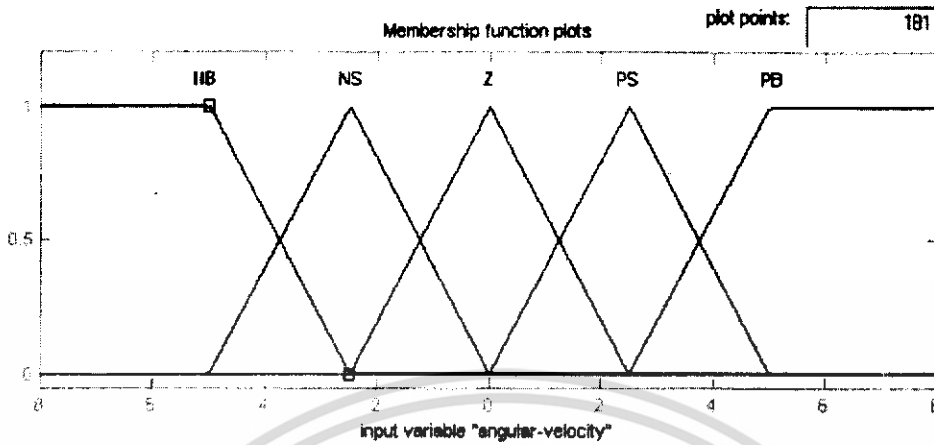
rule : [1x25 Struct]

2.4.2 กำหนด Membership function ของตัวแปร Input ทั้งสองตัวประกอบด้วย NB (Negative Big) , NS (Negative Small) , Z (Zero) , PS (Positive) , PB (Positive Big)



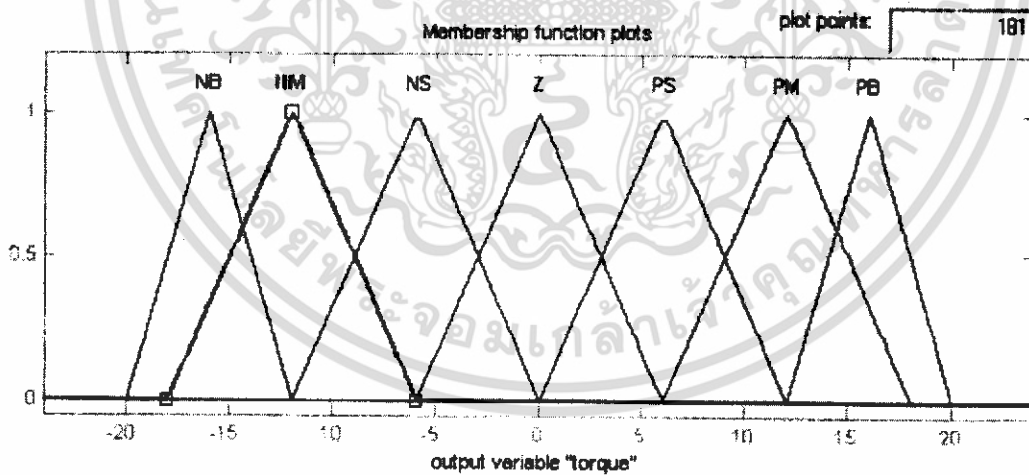
รูปที่ 2.7 Membership function ของ Input (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 Membership function ของ Input (2)

2.4.3 การกำหนดขอบเขต ของ Output สร้าง Membership function ของ torque ที่ใช้ควบคุมดังรูปที่ 2.9 ประกอบด้วย NB (Negative Big) , NS (Negative Small) , Z (Zero) , PS (Positive) , PB (Positive Big)



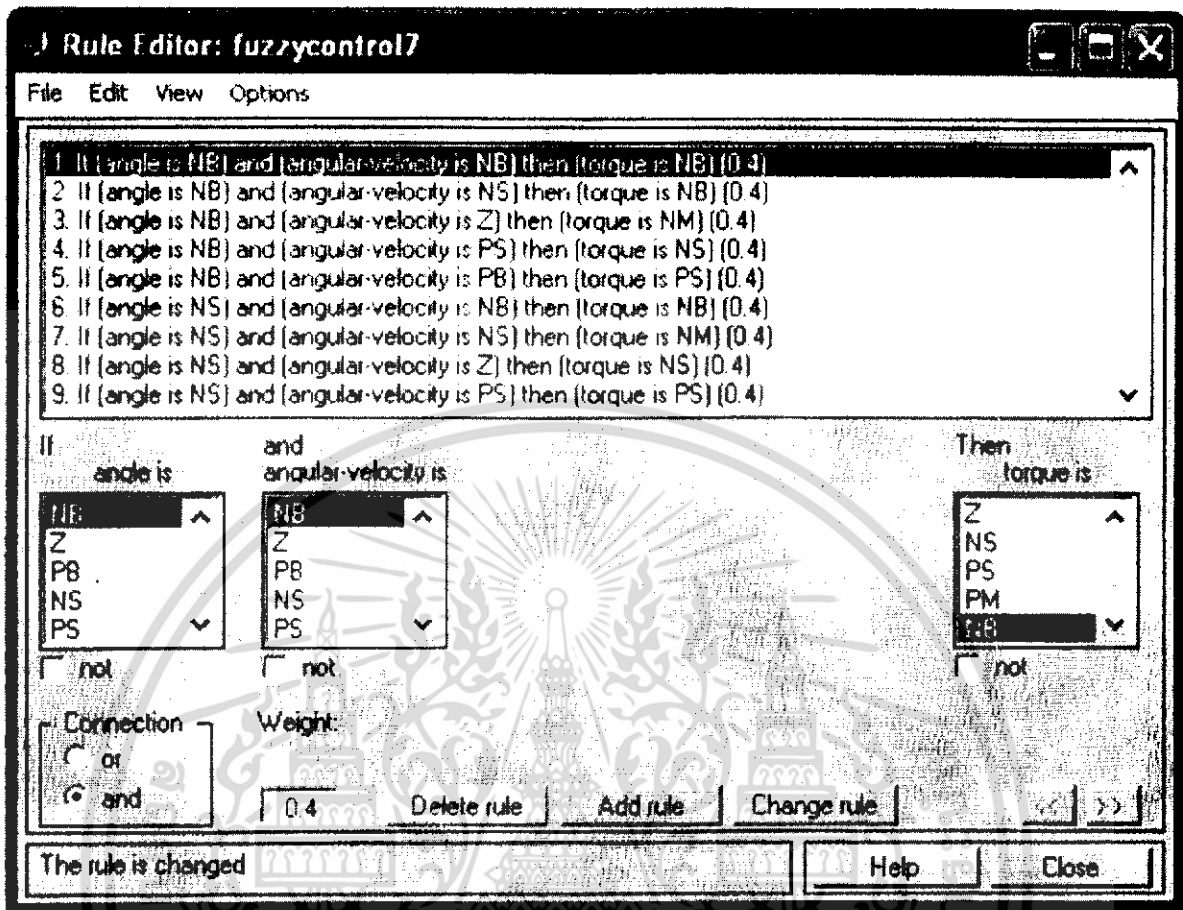
รูปที่ 2.9 Membership function ของ Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 การสร้างกฎให้กับระบบซึ่งประกอบไปด้วย 25 กฎ ดังนี้

1. If (angle is NB) and (angular – velocity is NB) then (torque is NB) (0.4)
2. If (angle is NB) and (angular – velocity is NS) then (torque is NB) (0.4)
3. If (angle is NB) and (angular – velocity is Z) then (torque is NM) (0.4)
4. If (angle is NB) and (angular – velocity is PS) then (torque is NS) (0.4)
5. If (angle is NB) and (angular – velocity is PB) then (torque is PS) (0.4)
6. If (angle is NS) and (angular – velocity is NB) then (torque is NB) (0.4)
7. If (angle is NS) and (angular – velocity is NS) then (torque is NM) (0.4)
8. If (angle is NS) and (angular – velocity is Z) then (torque is NS) (0.4)
9. If (angle is NS) and (angular – velocity is PS) then (torque is PS) (0.4)
10. If (angle is NS) and (angular – velocity is PB) then (torque is PS) (0.4)
11. If (angle is Z) and (angular – velocity is NB) then (torque is NM) (0.4)
12. If (angle is Z) and (angular – velocity is NS) then (torque is NS) (0.4)
13. If (angle is Z) and (angular – velocity is Z) then (torque is Z) (0.4)
14. If (angle is Z) and (angular – velocity is PS) then (torque is PS) (0.4)
15. If (angle is Z) and (angular – velocity is PB) then (torque is PM) (0.4)
16. If (angle is PS) and (angular – velocity is NB) then (torque is NS) (0.4)
17. If (angle is PS) and (angular – velocity is NS) then (torque is NS) (0.4)
18. If (angle is PS) and (angular – velocity is Z) then (torque is PS) (0.4)
19. If (angle is PS) and (angular – velocity is PS) then (torque is PM) (0.4)
20. If (angle is PS) and (angular – velocity is PB) then (torque is PB) (0.4)
21. If (angle is PB) and (angular – velocity is NB) then (torque is NS) (0.4)
22. If (angle is PB) and (angular – velocity is NS) then (torque is PS) (0.4)
23. If (angle is PB) and (angular – velocity is Z) then (torque is PM) (0.4)
24. If (angle is PB) and (angular – velocity is PS) then (torque is PB) (0.4)
25. If (angle is PB) and (angular – velocity is PB) then (torque is PB) (0.4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 Rule Editor ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Interface ระหว่างคอมพิวเตอร์ กับอุปกรณ์ต่างๆ

2.5.1. ทดลอง Interface ผ่าน printer port (parallel port)

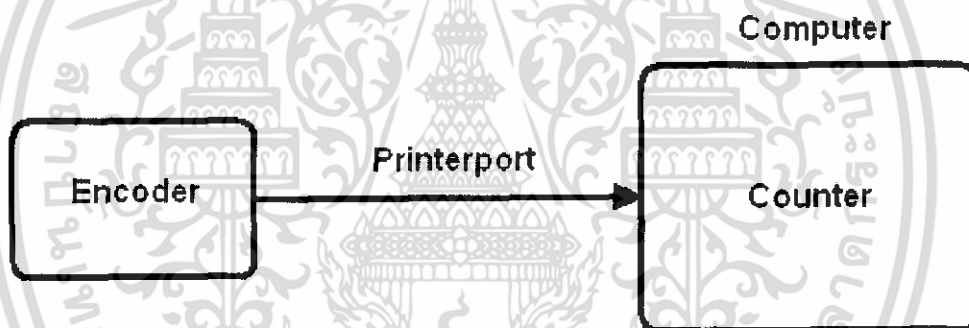
2.5.2. ทดลอง Interface ผ่าน PCI card.

2.5.1. ทดลอง Interface ผ่าน printer port (parallel port)

printer port หรือ parallel port มีด้วยกัน 25 pin มีแอดเดรส ดังนี้

in port = 0x379 5 bit

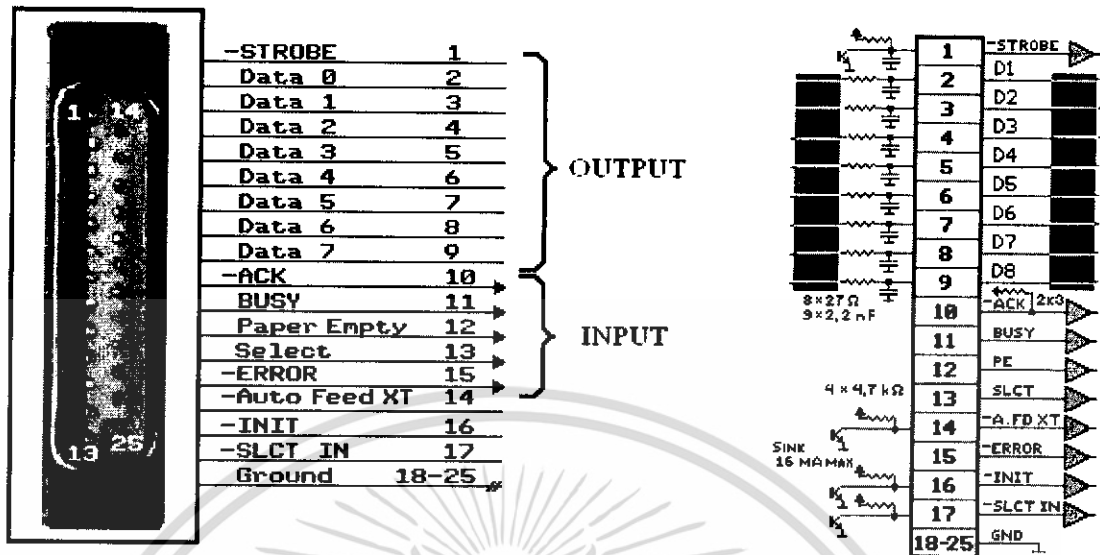
out port = 0x378 8 bit



รูปที่ 2.11 การ Interface ผ่าน Printer port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



เชื่อมต่อที่ PC 25 พิน ตัวเมีย

SINK 24 mA MAX
SOURCE 26 mA MAX

รูปที่ 2.12 รายละเอียดของ Printer port

2.5.2 ทดลอง Interface ผ่าน PCI card (Peripheral Component Interconnect)

ET-PCI16IO คือการ์ดต่อขยายระบบ คอมพิวเตอร์ PC เพื่อทำหน้าที่เป็นอุปกรณ์รับส่งสัญญาณดิจิทัลกับอุปกรณ์ภายนอกเนื่องจากปัจจุบัน ผู้ผลิตระบบคอมพิวเตอร์ส่วนมากได้ยกเลิกระบบบัสแบบ ISA กับ Main board รุ่นใหม่ บริษัท ETT ได้มองเห็นถึงความต้องการในการนำคอมพิวเตอร์ PC ไปใช้ในงานควบคุมยังมีอยู่ จึงได้พัฒนา ET-PCI16IO ซึ่งเป็น Card IO แบบ PCI Bu

การทำงานของ ET-PCI16IO

ET-PCI16IO การ์ด ประกอบด้วยสามส่วนคือ ส่วนของ 74F573, 74F574 ซึ่งทำหน้าที่เป็นอินพุต และ เอาท์พุต ส่วนของวงจร ดีโคเดออร์ทำหน้าที่ถอดรหัสโปรโตคอลของ PCI BUS ได้แก่ ไอซี PLD GAL22V10 ไอซีเปรียบเทียบลอจิก 74F521 เมื่อเกิดขบวนการอ่านเขียนข้อมูล ไอซี PLD จะทำการตรวจลำดับและรูปแบบของสัญญาณร่วมกับ ไอซีเปรียบเทียบลอจิก 74F521 จากนั้นจะส่งสัญญาณ Clock ให้กับ ไอซี 74F573 หรือ ไอซี 74F574 เป็นการถ่ายโอนข้อมูลระหว่าง PCI data bus กับ IO ภายนอก

ลักษณะโดยทั่วไปของ PCI BUS

PCI BUS เป็นสถาปัตยกรรมยุคใหม่ที่มีความสามารถสูงและสามารถถ่ายโอนข้อมูลได้มากมีความซับซ้อนมากกว่า ISA BUS แบบเก่า

อุปกรณ์ที่จะนำมาเชื่อมต่อกับ PCI จะต้องได้มาตรฐานตามคุณสมบัติที่ได้กำหนดไว้แล้ว แต่ การออกแบบอุปกรณ์ PCI ชนิด Target

only สำหรับนำมาใช้ในงานควบคุมที่ไม่ซับซ้อน นับเป็นทางเลือกอีกอย่างหนึ่ง และทางบริษัท อีทีที ได้ออกแบบให้ คุณสมบัติของการ์ด

ใกล้เคียงกับมาตรฐานของ PCI มากที่สุด

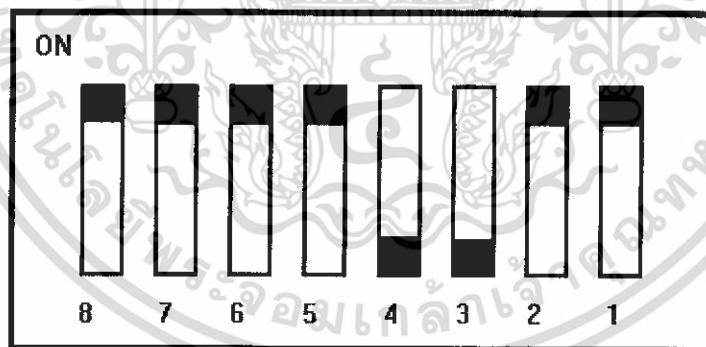
หลังจากตรวจสอบ Base address ของ ระบบแล้วในกรณี F300 H ไม่ว่าจะ หรือ ต้องการเปลี่ยน address ของ CARD ผู้ใช้งานต้องทำการ เลือกสวิตช์ใหม่ การเลือกสวิตช์ให้เลือกที่ DIPSW 2 เท่านั้น ส่วน แอดเดรสไบต์ล่างถูกกำหนดมาให้เป็น 00 จากฮาร์ดแวร์

แอดเดรส ที่แนะนำให้ใช้

DIP SW ON = 1 OFF = 0

2.5.1 F300 HEX

DIP SW 2



11110011

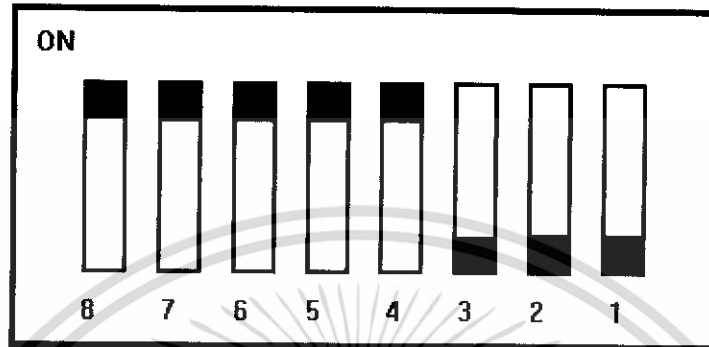
F | 3

รูปที่ 2.13 address ที่ F300 HEX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 F400 HEX

DIP SW 2



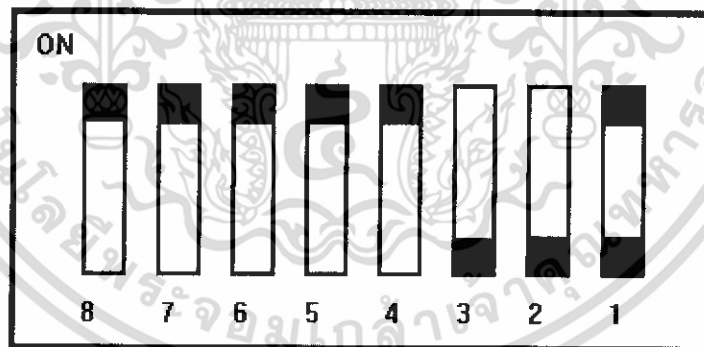
11111011

F|4

รูปที่ 2.14 Address ที่ F400 HEX

2.5.3 F500 HEX

DIP SW 2



11111001

F|5

รูปที่ 2.15 Address ที่ F500 HEX

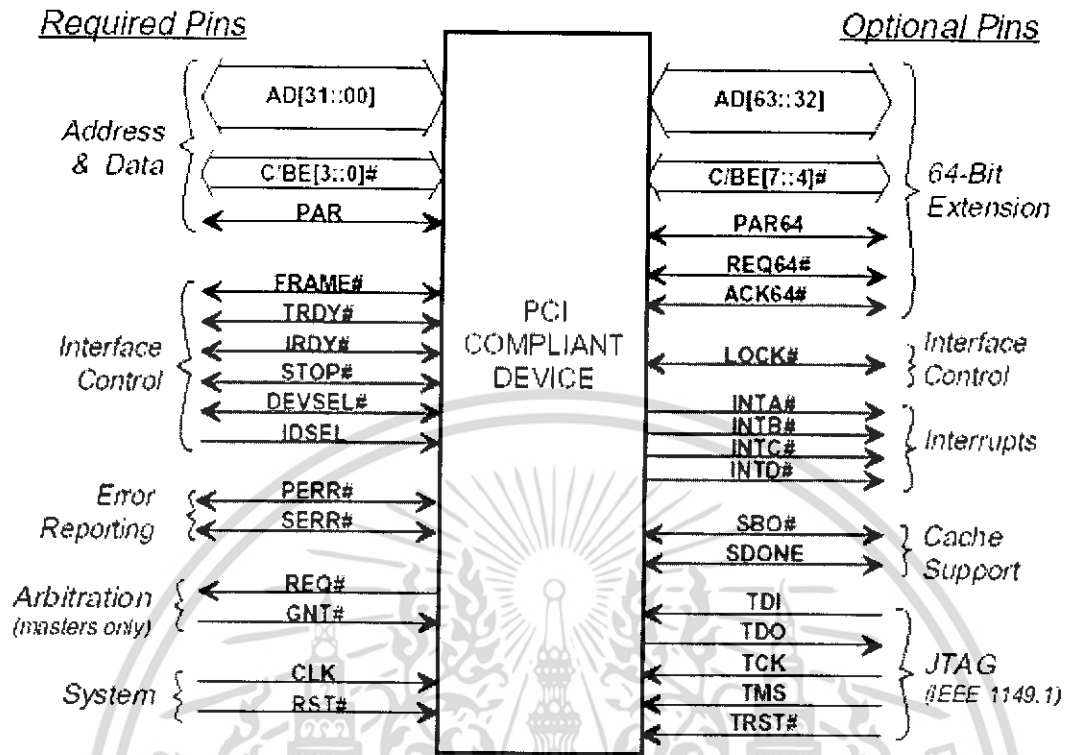
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ ET-PCI16IO

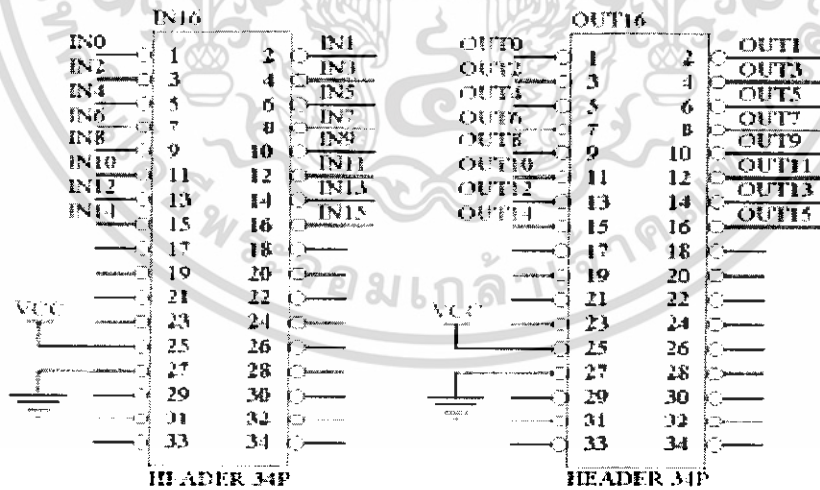
FET-PCI16IO การ์ด ประกอบด้วยสามส่วนคือ ส่วนของ74F573,74F574 ซึ่งทำหน้าที่เป็น อินพุตและเอาพุต ส่วนของวงจร ดีโคเดอร์ทำหน้าที่ถอดรหัสโปรโตคอลของ PCI BUS ได้แก่ ไอซี PLD GAL22V10 ไอซีเปรียบเทียบลอจิก 74F521 เมื่อเกิดขบวนการอ่านเขียนข้อมูล ไอซี PLD จะ ทำการตรวจลำดับและรูปแบบของสัญญาณร่วมกับ ไอซีเปรียบเทียบลอจิก 74F521จากนั้นจะส่ง สัญญาณ Clock ให้กับ ไอซี 74F573 หรือ ไอซี 74F574 เป็นการถ่ายโอนข้อมูลระหว่าง PCI data bus กับ IO ภายนอก

ลักษณะโดยทั่วไปของ PCI BUS

PCI BUS เป็นสถาปัตยกรรมยุคใหม่ที่มีความสามารถสูงและสามารถถ่ายโอนข้อมูลได้ มากมีความซับซ้อนมากกว่า ISA BUS แบบเก่าอุปกรณ์ที่จะนำมาเชื่อมต่อกับ PCI จะต้องได้ มาตรฐานตามคุณสมบัติที่ได้กำหนดไว้แล้ว แต่ การออกแบบอุปกรณ์ PCI ชนิด Target only สำหรับนำมาใช้ในงานควบคุมที่ไม่ซับซ้อน นับเป็นทางเลือกอีกอย่างหนึ่ง และทางบริษัท อีที ที ได้ออกแบบให้ คุณสมบัติของการ์ดใกล้เคียงกับมาตรฐานของ PCI มากที่สุด สัญญาณอินเทอร์เฟสของ PCI BUS

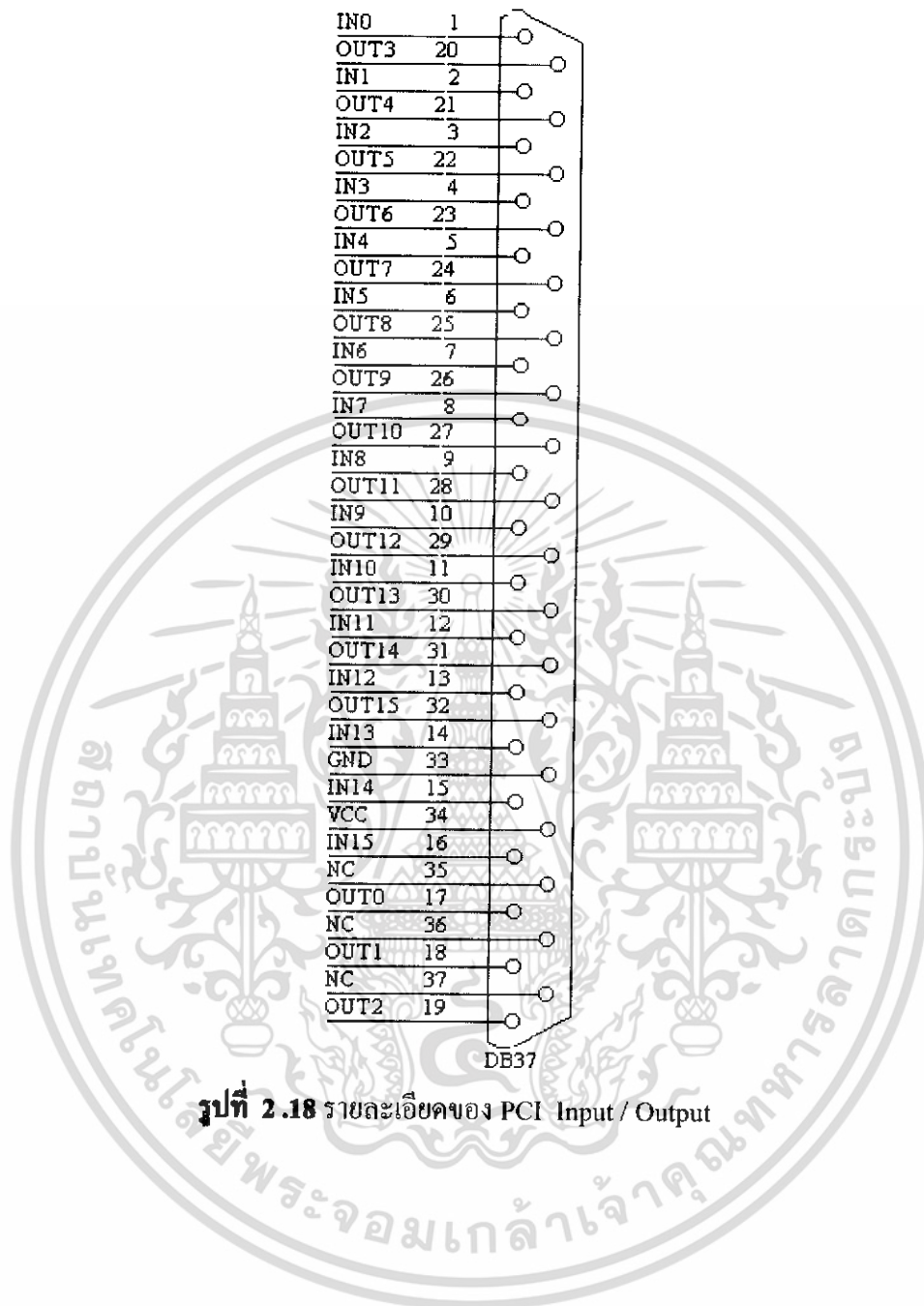


รูปที่ 2.16 PCI Pin list



รูปที่ 2.17 in port / out port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 รายละเอียดของ PCI Input / Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

รายละเอียดในการออกแบบและการสร้าง

แบ่งเป็น 2 ส่วนคือ

3.1. ส่วนของชิ้นงาน คือ รถ และ ราง

3.2. ส่วนของระบบควบคุม และ โปรแกรม

3.1. ส่วนของชิ้นงาน

มีขั้นตอนปฏิบัติดังนี้

3.1.1. วิเคราะห์และออกแบบ

3.1.2. จัดหาอุปกรณ์

ตัวรถและรางประกอบด้วย

3.1.2.1. เอนโคเดอร์ 2500 P/R 2 ตัว

3.1.2.2. ดีซีมอเตอร์ 24 Volts 1 ตัว

3.1.2.3. รางเลื่อน 92 ซม. 1 ชุด

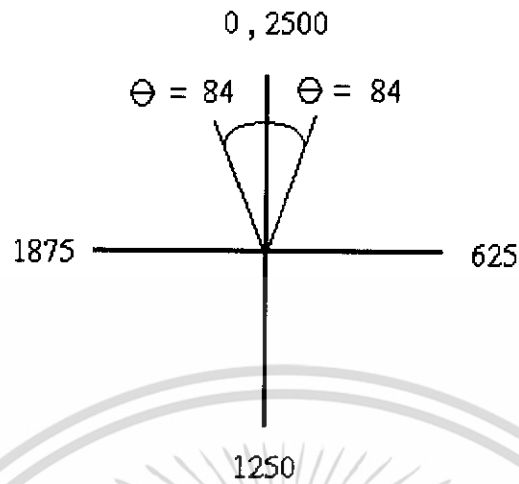
3.1.2.4. เฟือง 1 ตัว

3.1.2.5. อื่น ๆ

3.1.3. สร้างชิ้นงาน

3.1.4. การทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงแนวคิดในการหาค่า พัลส์ เพื่อนำไปควบคุมมอเตอร์

อิสระ

ขั้นตอน ที่ 1. ทำการทดลอง โดยการจับเวลาของก้านอินเวอร์ทเพนคูล์มขณะปล่อยอย่าง

ขั้นตอน ที่ 2. ทำการคำนวณหาค่าพัลส์โดยหาค่าได้ดังนี้

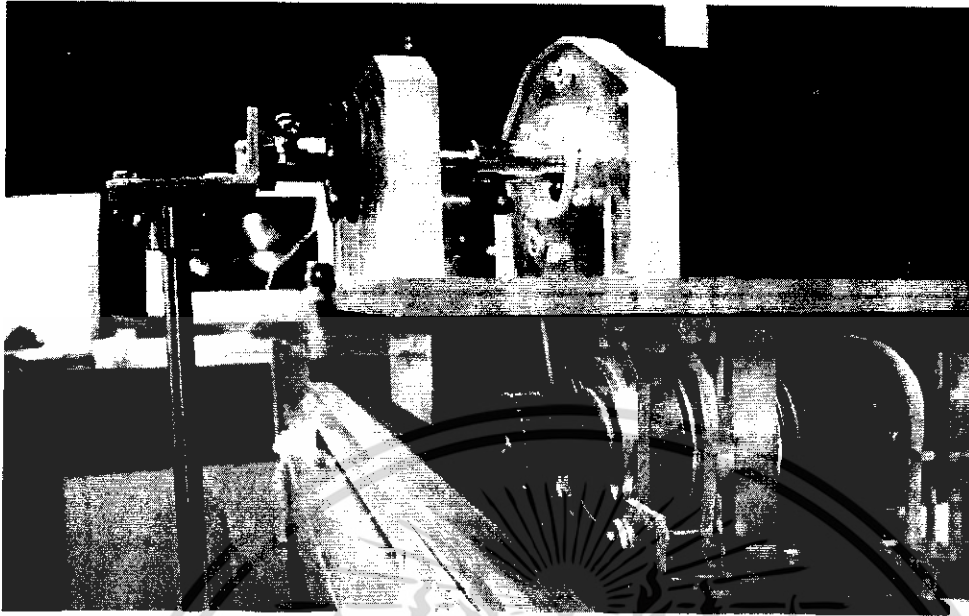
$$2500 \text{ pulse} = 360^\circ \text{ องศา}$$

$$84 = x \quad (\text{เทียบบัญญัติ})$$

$$x = \frac{360 * 84}{2500} = 12.096 \text{ องศา}$$

ขั้นตอน ที่ 3.ทำการประกอบตัวชิ้นงานดังแสดงดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

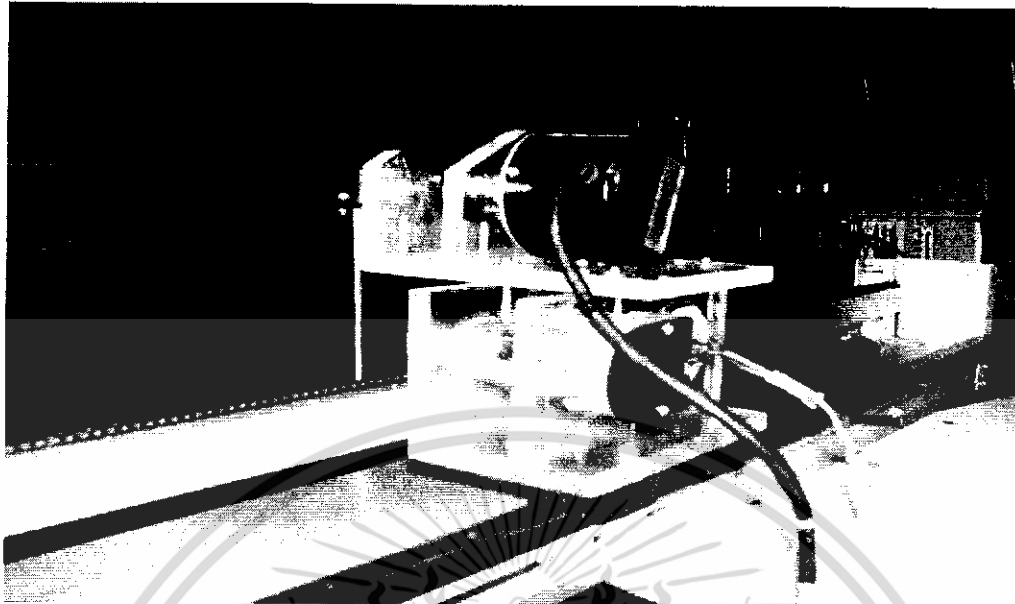


รูปที่ 3.2 แสดงด้านข้างของตัวรถ



รูปที่ 3.3 แสดงด้านหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงด้านหลัง



รูปที่ 3.5 แสดงโครงด้านซ้ายและยึดกับโต๊ะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

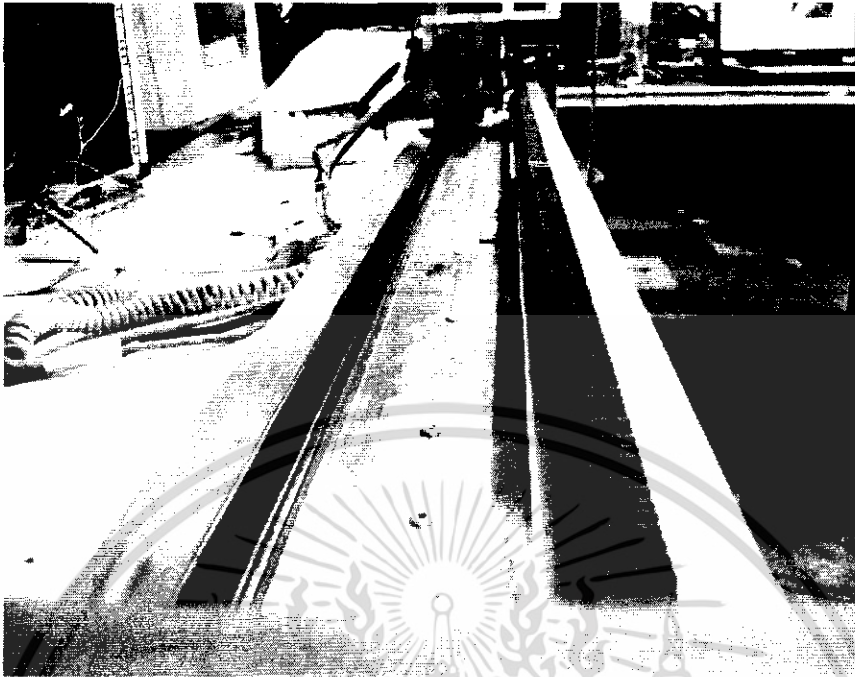


รูปที่ 3.6 แสดงโครงด้านขวาและยึดกับโต๊ะ



รูปที่ 3.7 แสดงเฟืองและรางแบบเฟือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงตัววาง



รูปที่ 3.9 แสดงก้านเพนดูลัม

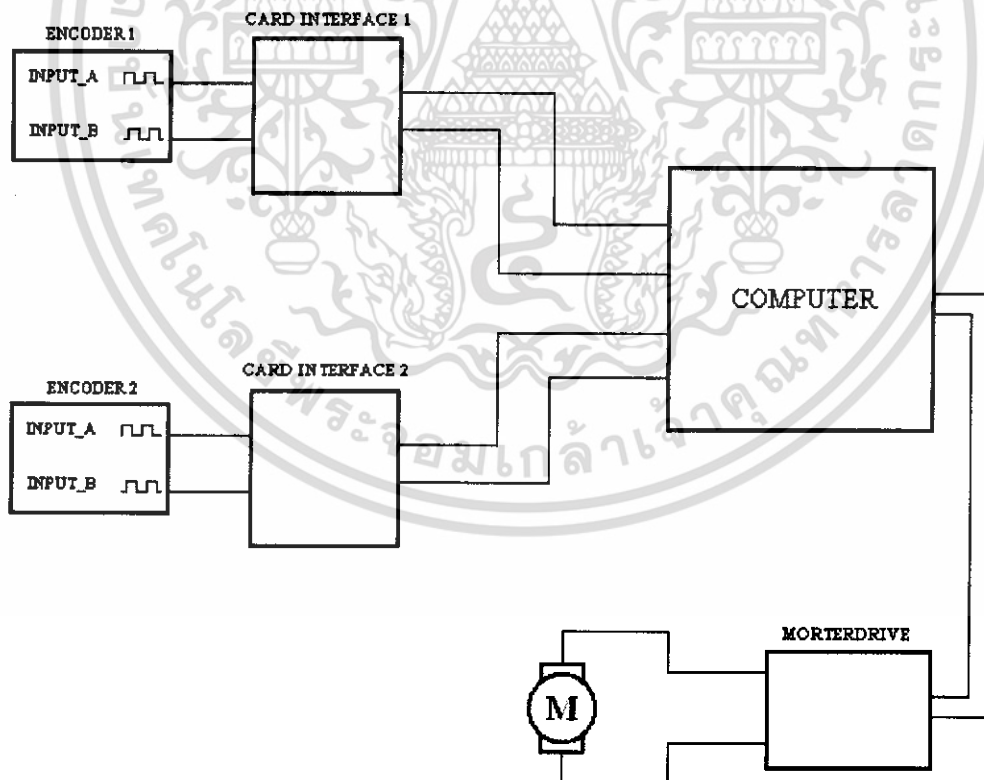
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2. ส่วนของระบบควบคุม และโปรแกรม

3.2.1 วงจรเชื่อมต่อระหว่างคอมพิวเตอร์และอุปกรณ์ภายนอก

การควบคุมอุปกรณ์ภายนอกโดยใช้เครื่องคอมพิวเตอร์นั้น จะทำให้เครื่องคอมพิวเตอร์ติดต่ออุปกรณ์ภายนอกให้ได้เสียก่อน การติดต่อสื่อสารระหว่างตัวรถกับคอมพิวเตอร์ในโครงงานนี้จะทำการอินเทอร์เฟส โดยใช้ไมโครคอนโทรลเลอร์เบอร์ MCS-51 เพื่อใช้เป็นตัวกำเนิด Pulse Width Modulation โดยวงจรเชื่อมต่อนี้ทำหน้าที่เป็นเหมือนพอร์ท รวมทั้งทำหน้าที่เป็นบัฟเฟอร์ของสัญญาณข้อมูล และสัญญาณควบคุมต่างๆ ด้วย ดังแสดงดังรูป

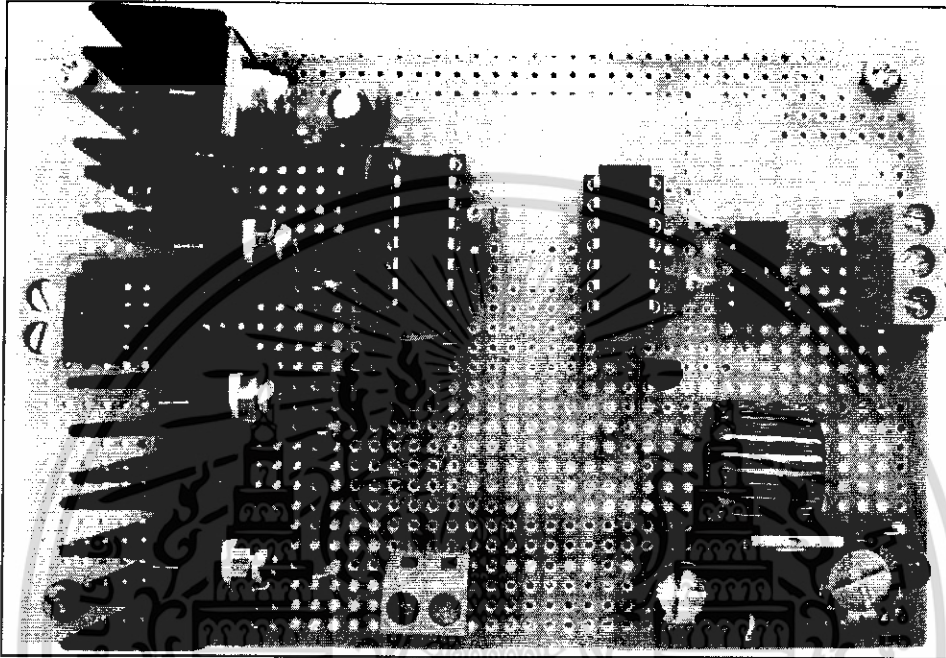
3.2.2 การใช้งานแบบเปิด-ปิด และปรับความกว้างพัลส์ (PWM) ด้วยไมโครคอนโทรลเลอร์



รูปที่ 3.10 แสดงการใช้โปรแกรมควบคุมมอเตอร์ผ่าน MCS-51 2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 วงจรชุดขับมอเตอร์



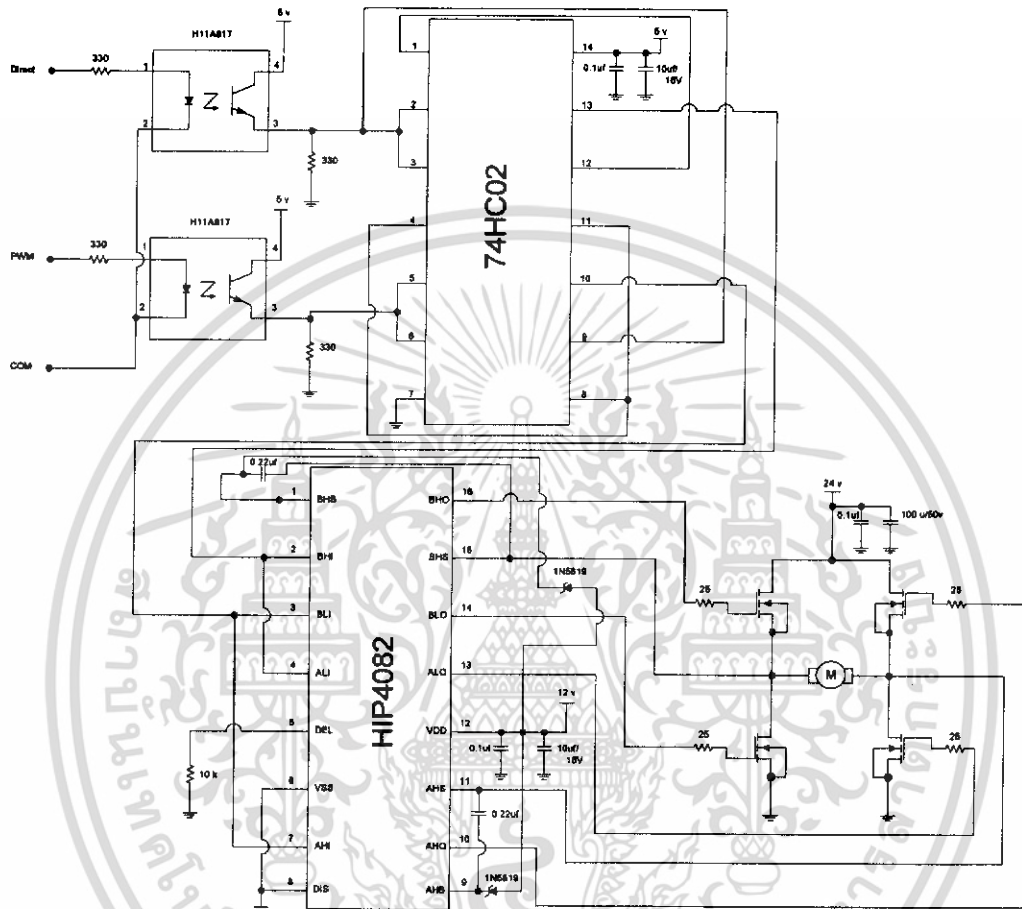
รูปที่ 3.11 แสดงวงจรชุดขับมอเตอร์

การทำงานของวงจร

วงจรขับมอเตอร์(Driver Motor) จะประกอบด้วยส่วนหลักอยู่ 3 ส่วนคือ ส่วน Part Logic , ส่วน Current H- Bridge FET Driver และแหล่งจ่ายแรงดัน (Power Supply)

ซึ่งการทำงานส่วนแรกจะได้จากการป้อนสัญญาณ PWM ผ่าน opto couple ให้มีสัญญาณออก เป็น 5 โวลต์ หรือ 0 โวลต์ ผ่านวงจร Logic gate เพื่อทำการตรวจสอบสัญญาณที่ออกมาว่า เป็นจริงหรือไม่หรือก็คือทำการกลับสัญญาณตรงข้ามกับสัญญาณอินพุตแล้วทำการส่งต่อไปยังส่วน Driver ซึ่งใช้ ไอซีเบอร์ HIP 4082 ซึ่งการทำงานจะทำงานเป็นคู่คือ จะมีอินพุต AHI ,ALI และ BHI , BLI ส่วนทางด้านเอาต์พุตก็เช่นกัน คือ จะมี ALO,AHO และ BHO, BLO ซึ่งจะส่งผลให้การทำงานของ FET ทำงานเป็นคู่โดยจะส่งผลให้มอเตอร์มีความเร็วมากหรือน้อยขึ้นอยู่กับจำนวนเปอร์เซ็นต์ duty cycle ของสัญญาณ PWM ที่ป้อนให้กับวงจร และส่วนแหล่งจ่าย จะใช้ ไอซีเบอร์ LM 2575 – 12 ให้แรงดันเอาต์พุตออก 12 โวลต์ และป้อนให้กับ ไอซีเบอร์ 78L05 ต่อไป.

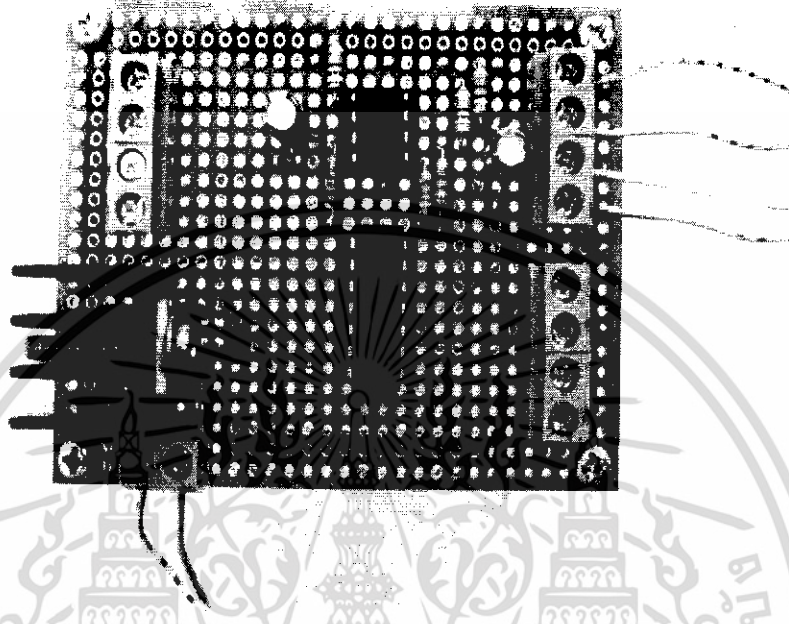
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงวงจรชุดขับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

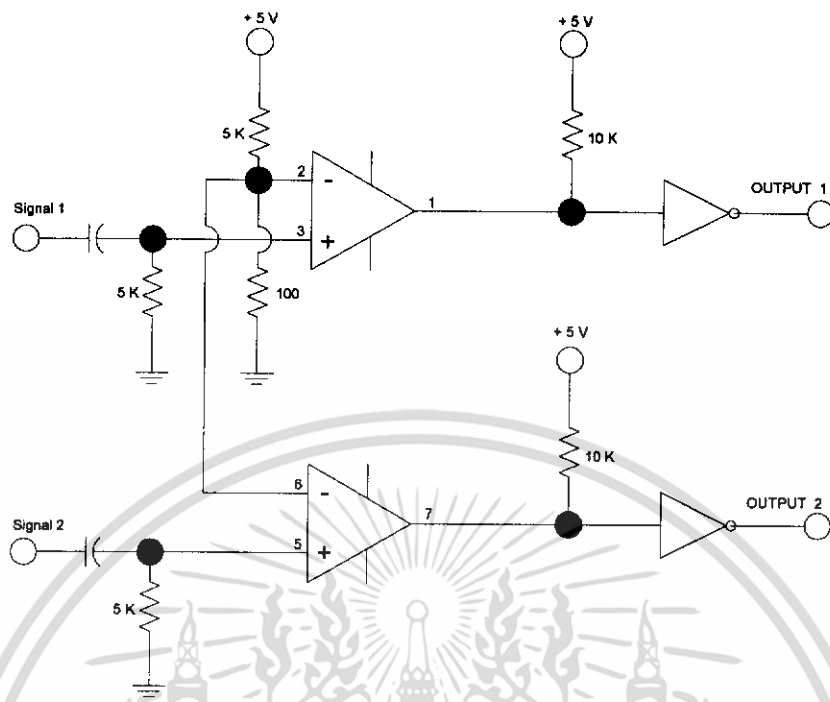
3.2.4 วงจรคอมพาราเตอร์



รูปที่ 3.13 แสดงวงจรวงจรถคอมพาราเตอร์

การทำงานของวงจร

ลักษณะการทำงานของวงจรถคอมพาราเตอร์ จะใช้ ไอซีเบอร์ LM 393 ซึ่งใช้แรงดันเปรียบเทียบ 2 อินพุตที่เป็นอิสระต่อกัน โดยถูกออกแบบให้ทำงานจากแหล่งจ่ายเดียวที่มีช่วงกว้างของแรงดัน ซึ่งสัญญาณอินพุตทั้ง 2 อินพุตจะเป็นสัญญาณ ซายน์เวฟเมื่อผ่านวงจรถคอมพาราเตอร์แล้วจะได้สัญญาณเป็นรูปสี่เหลี่ยมที่มีเฟสต่างกันอยู่ 90 องศา แล้วส่งผ่านไปยัง ไอซีเบอร์ 74HC04 ซึ่งทำการกลับสัญญาณจาก High เป็น Low หรือ Low เป็น High ซึ่งสัญญาณเอาต์พุตที่ได้ก็ยังมีเฟสต่างกันอยู่ 90 องศา .



รูปที่ 3.14 แสดงวงจรวงจรคอมพาราเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดสอบตัวรถและราง

ทำการทดสอบตัวรถและราง โดยการปล่อยก้อนเพนดูลัมจากจุดบนแล้วทำการจับเวลา



รูปที่ 4.1 แสดงการปล่อยก้อนเพนดูลัมจากจุดบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้คือ

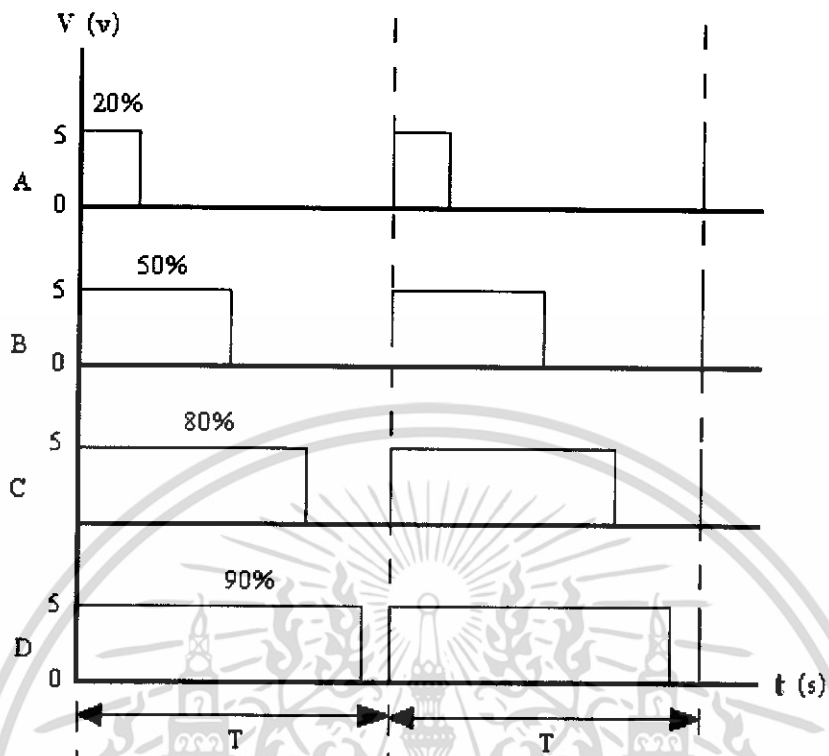
| ปล่อยครั้งที่ | เวลา(วินาที) |
|---------------|--------------|
| 1 | 55.2 |
| 2 | 55.3 |
| 3 | 55.2 |

ตารางที่ 4.1

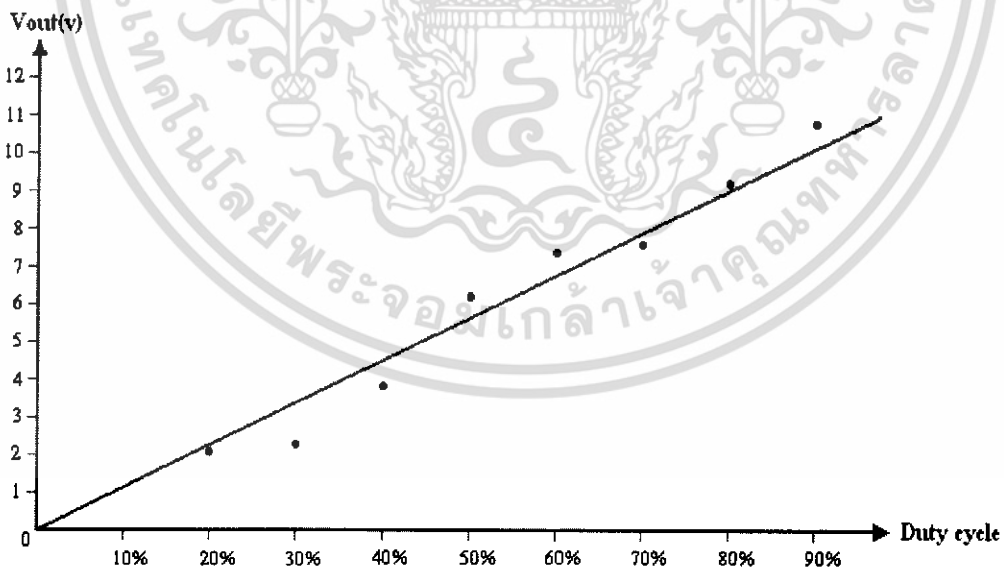
แสดงว่าก้านเพนดูลัมใช้เวลาจากจุดสมดุลบน แล้วลงจุดสมดุลล่าง ประมาณ 55 วินาที

4.2 การทดลองเพื่อหาความเร็วของมอเตอร์ที่เหมาะสม

การทดลองเพื่อหาความเร็วของมอเตอร์ที่เหมาะสม หมายถึง การควบคุมมอเตอร์จะใช้ การควบคุม โดย ความกว้างของพัลส์ โดย ที่ความถี่คงที่ $f = 500 \text{ Hz}$

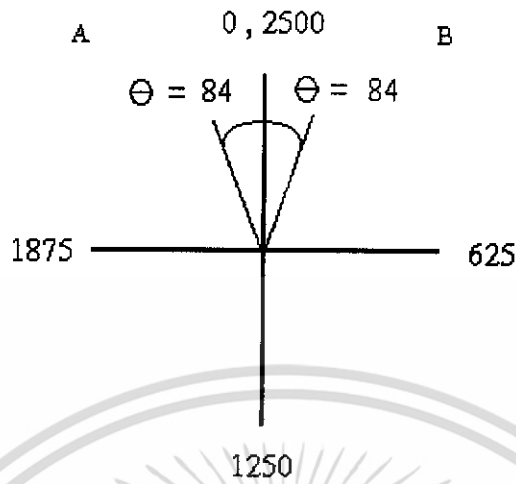


รูปที่ 4.2 แสดงรูปคลื่น



รูปที่ 4.3 กราฟแสดงความสัมพันธ์ระหว่าง ค่าตัวดี ไซเคิลแรงดันที่มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

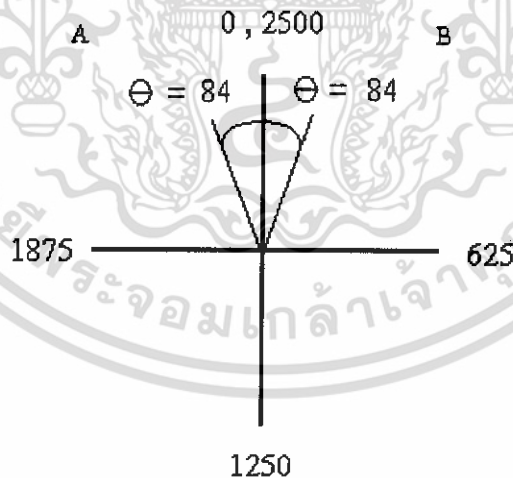


รูปที่ 4.4 แสดงการวิเคราะห์หมุมในโหมคของพัลส์

จากการวิเคราะห์หมุมในโหมคของพัลส์ได้

พัลส์สูงสุด = 84 ลูก ได้หมุม = 12 องศา

นำพัลส์สูงสุด 84 ลูก ไปทำการแปลงเป็นการหน่วงเวลาในการเขียนโปรแกรมโดยระบบฟัซซี่ (fuzzy system)



รูปที่ 4.4 สัญญาณจากเอ็น โคดเดอร์ 1

- If A θ = NM then P = D
- If B θ = PM then P = D
- If A θ = Nm then P = A
- If A θ = Pm then P = A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดลอง Interface ผ่าน printer port (parallel port)

ทดลอง Interface ผ่าน printer port (parallel port) จากความเร็วสูงสุดของก้านเพนดูลัม = 0.2 ms เนื่องจาก ตัวเอน โคคเตอร์ค่อนข้างละเอียดมาก คือหนึ่งรอบเอน โคคเตอร์ผลิตความถี่ได้ถึง 2500 พัลส์ ตัวนับจึงต้องมีความเร็วจะเห็นว่าไม่สามารถนับได้อย่างถูกต้องเนื่องจากการ sampling ความเร็วของก้านเพนดูลัม เข้าทาง

4.4 ผลการทดลอง Interface ผ่าน PCI card

| องศา | จำนวน พัลส์ นับได้(หมุนช้า) | จำนวน พัลส์ นับได้(หมุนเร็ว) |
|------|-----------------------------|------------------------------|
| 0 | 0 | 0 |
| 90 | 64 | 55 |
| 180 | 144 | 100 |
| 270 | 202 | 183 |
| 360 | 300 | 256 |

ทดลอง Interface ผ่าน PCI card จากความเร็วสูงสุดของก้าน เพนดูลัม = 0.2 ms เนื่องจาก ตัวเอน โคคเตอร์ค่อนข้างละเอียดมาก คือหนึ่งรอบ เอน โคคเตอร์ ผลิตความถี่ ได้ถึง 2500 พัลส์ ตัวนับจึงต้อง มีความเร็วเท่ากับหรือมากกว่า = 0.2 ms มากๆ จึงสามารถนับได้ถูกต้อง

บทที่ 5

สรุปผลวิจารณ์และสรุป

5.1 บทสรุป

จากการสร้างชิ้นงาน และทำการทดลองปรากฏว่า สามารถควบคุมการแกว่งขึ้นของก้าน เพนดูลัมได้โดยผู้ควบคุมแต่เมื่อทดสอบให้รับค่าจากเอ็น โคเดเดอร์ 1 แล้วให้ระบบทำการรักษา เสถียรภาพเองปรากฏว่า มีการแกว่งของก้านเพนดูลัมช่วงเวลาหนึ่งแล้วตกลง เข้าสู่จุดสมดุลล่างที่ 180 องศา



รูปที่ 5.1 แสดงมุมในการแกว่งของก้านเพนดูลัม

ระบบยังไม่สามารถรักษาเสถียรภาพเอง ได้สมมุติแนวทางการแก้ไขคือ จะต้องวิเคราะห์ ระบบให้ลึก พิจารณาค่าตัวแปรต่างๆ ให้ค่ามุม(input) และค่าตอบสนอง(output) ให้ สะเอียด มาก ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาที่พบ

ช่วงแรกโปรแกรมที่ใช้นับจำนวนพัลส์ไม่สามารถนับค่าได้อย่างถูกต้องทำให้ในการควบคุมไม่สามารถทำได้เนื่องจากการควบคุมต้องอาศัยข้อมูลที่ถูกต้องถึงจะสามารถนำไปวิเคราะห์และใช้งานได้อย่างมีประสิทธิภาพ การวิเคราะห์ปัญหาที่ไม่ถูกต้องทำให้ผลที่ได้เกิดความผิดพลาดทำให้ไม่สามารถออกแบบระบบที่ถูกต้องได้

5.3 แนวทางการแก้ปัญหาและพัฒนา

5.3.1 สร้างวงจรมับจากภายนอก

การสร้างวงจรมับจากภายนอกช่วยให้สามารถลดเวลาในการประมวลผลคือการที่ให้คอมพิวเตอร์เป็นตัวนับสัญญาณพัลส์ จะทำให้เสียเวลาหรือเพิ่มจำนวนคาบเวลาโดยเปล่าประโยชน์ ดังนั้นถ้าใช้วงจรมับจากภายนอกแล้วทำการอินเทอร์รัพท์ให้คอมพิวเตอร์รับค่าจากวงจรมับก็จะทำให้ช่วงการประมวลผลเร็วขึ้น

5.3.2 การส่งผ่านข้อมูลโดยการอินเทอร์รัพท์

การส่งผ่านข้อมูลโดยการอินเทอร์รัพท์ คอมพิวเตอร์จะรอรับข้อมูลจากวงจรมับ ทำให้ได้ข้อมูลที่เร็วข้อมูลที่ได้อีกก็จะมี ความถูกต้องทำให้ช่วงการประมวลผลเร็วขึ้น

5.3.3 การสร้างสัญญาณอินเทอร์รัพท์จากภายนอก

การสร้างสัญญาณอินเทอร์รัพท์จากภายนอก ทำให้สามารถกำหนดช่วงเวลาที่เหมาะสมในการดึงข้อมูลเข้ามาประมวลผลและส่งข้อมูลออกไปควบคุมมอเตอร์ ความถี่ของสัญญาณนั้นต้องมากกว่า 5kHz หรือ เร็วกว่า 0.2 ms

5.3.4 สร้างตัวควบคุมระบบอินเวอร์ทเพนดูลัม

สร้างตัวควบคุมระบบอินเวอร์ทเพนดูลัม ที่สามารถรักษาเสถียรภาพให้กับเพนดูลัมทรงตัวอยู่ได้โดยไม่ล้ม

5.3.5 ประยุกต์ นำระบบอินเวอร์ทเพนดูลัม ไปใช้งาน

สามารถนำระบบนี้ไปประยุกต์ใช้กับระบบต่างๆ ได้ เช่นควบคุมการทำงานของเครน การรักษามุมของหุ่นยนต์

บรรณานุกรม

1. Sigeru Omatu and Tatuya Ide “ Stabilization of Inverted Pendulum by Neuro – Control ”
Department of Information Science & Intelligent Systems University of Tokushima , Japan
2. L.A. Zadeh, “Fuzzy Sets” Information and Control , Vol.8, pp.338-353, 1965
- 3 T.Fukuda and T.Shibata, “Research Trends in Neuro - morphic Control ”, Journal of
Robotics and Mechatronics, Vol.2, No.4, pp1-15, 1991
4. E.H.Mamdani, “Application of Fuzzy Algorithms for Control of Simple Dynamic Plant ”,
Proc.of IEE, Vol .121 , No.12, pp 1585-1588 .1974
5. รศ.ดร.มนัส สัจจวิมล และ วรรัตน์ ภัทรอมรกุล “ คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์ ”
พิมพ์ครั้งที่ 1 เมษายน 2543 ISBN 974 – 8237 – 91 – 5
6. นายชุตินันต์ ชงศ์ประพัฒน์ , นาย ณรงค์พล ทองเหลือง , นายบัณฑิต คุณวุฒิ ” ทฤษฎี Fuzzy set
theory and logics” ภาคเรียนที่ 1 ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



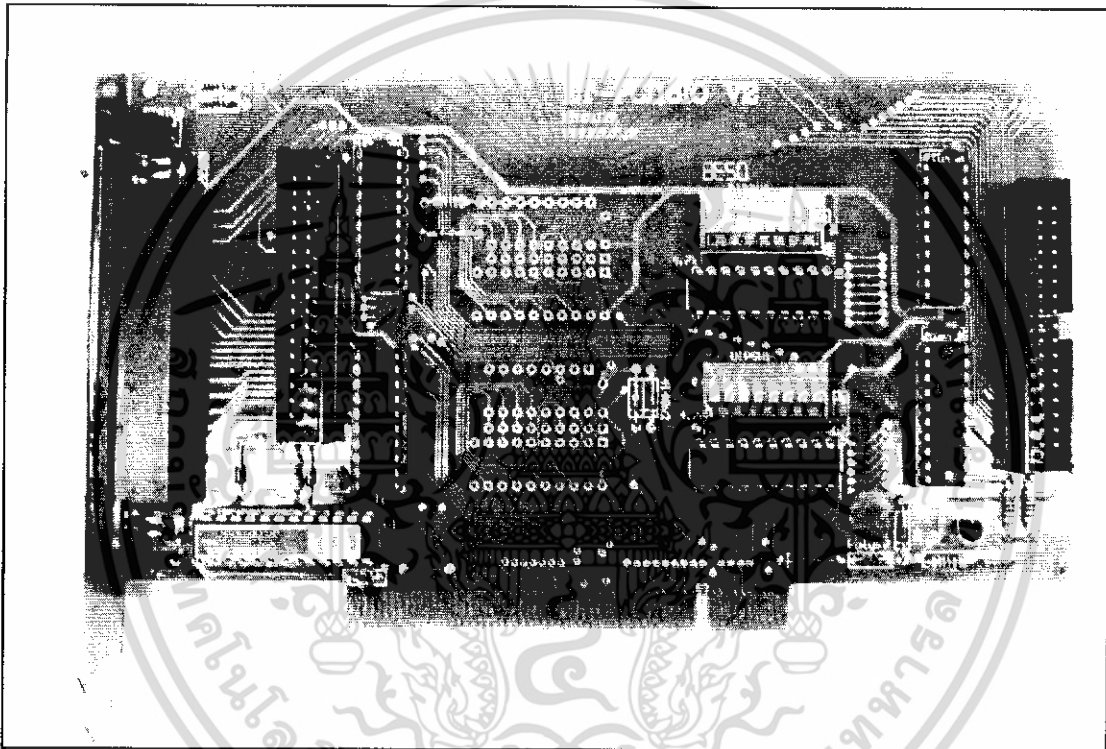
ภาคผนวก ก.
รายละเอียดของอุปกรณ์ที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.1 เครื่องคอมพิวเตอร์ส่วนบุคคล(PC)

เครื่องคอมพิวเตอร์ส่วนบุคคลทำหน้าที่ควบคุมการทำงานโดยรวม และทดลองโปรแกรมที่ใช้ นับสัญญาณ ตรวจสอบทิศทางรถ และควบคุมรถ เนื่องจากการ

ก.2 การ์ดอินเทอร์เฟซแบบ พีซีไอ(PCI)

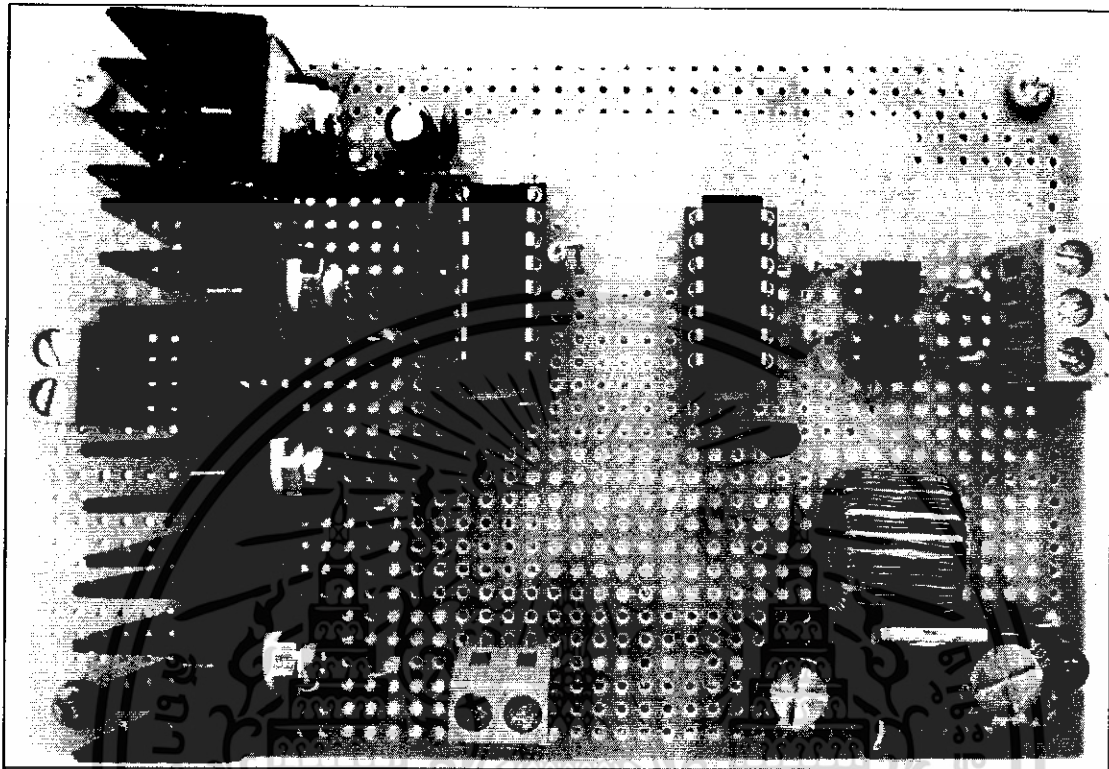


รูปที่ ก.1 การ์ดอินเทอร์เฟซ ET-PC16IO

FET-PCI16IO การ์ด ประกอบด้วยสามส่วนคือ ส่วนของ74F573,74F574 ซึ่งทำหน้าที่เป็นอินพุทและเอาพุท ส่วนของวงจร ดีโคเดอร์ทำหน้าที่ถอดรหัสโปรโตคอลของ PCI BUS ได้แก่ ไอซี PLD GAL22V10 ไอซีเปรียบเทียบลอจิก 74F521 เมื่อเกิดขบวนการอ่านเขียนข้อมูล ไอซี PLD จะทำการตรวจลำดับและรูปแบบของสัญญาณร่วมกับ ไอซีเปรียบเทียบลอจิก 74F521จากนั้นจะส่งสัญญาณ Clock ให้กับ ไอซี 74F573 หรือ ไอซี 74F574 เป็นการถ่ายโอนข้อมูลระหว่าง PCI data bus กับ IO ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.3 วงจรขับมอเตอร์



รูปที่ ก.2 วงจรขับมอเตอร์

การทำงานของวงจร

วงจรขับมอเตอร์(Driver Motor) จะประกอบด้วยส่วนหลักอยู่ 3 ส่วนคือ ส่วน Part Logic , ส่วน Current H- Bridge FET Driver และแหล่งจ่ายแรงดัน (Power Supply)

ซึ่งการทำงานส่วนแรกจะได้จากการป้อนสัญญาณ PWM ผ่าน opto couple ให้มีสัญญาณออก เป็น 5 โวลต์ หรือ 0 โวลต์ ผ่านวงจร Logic gate เพื่อทำการตรวจสอบสัญญาณที่ออกมาว่า เป็นจริงหรือไม่หรือก็คือทำการกลับสัญญาณตรงข้ามกับสัญญาณอินพุตแล้วทำการส่งต่อไปยัง ส่วน Driver ซึ่งใช้ ไอซีเบอร์ HIP 4082 ซึ่งการทำงานจะทำงานเป็นคู่คือ จะมีอินพุต AHI ,ALI และ BHI , BLI ส่วนทางด้านเอาต์พุตก็เช่นกัน คือ จะมี ALO,AHO และ BHO, BLO ซึ่งจะส่งผลให้การทำงานของ FET ทำงานเป็นคู่โดยจะส่งผลให้มอเตอร์มีความเร็วมากหรือน้อยขึ้นอยู่กับจำนวน เปอร์เซ็นต์ duty cycle ของสัญญาณ PWM ที่ป้อนให้กับวงจร และส่วนแหล่งจ่าย จะใช้ ไอซีเบอร์ LM 2575 – 12 ให้แรงดันเอาต์พุตออก 12 โวลต์ และป้อนให้กับ ไอซีเบอร์ 78L05 ต่อไป.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.เอนโคดเดอร์แบบหมุนสำหรับวัดตำแหน่งมุมและรอราง

เอนโคดเดอร์แบบหมุนสำหรับวัดตำแหน่งมุม(ตัวบน) เป็นเอนโคดเดอร์ที่สามารถกำเนิดสัญญาณในลักษณะอินครีเมนตัล(incremental encoder) จะมีความละเอียด 2500 พัลส์ ต่อ หนึ่งรอบการหมุน(360 องศา) เอนโคดเดอร์แบบหมุนสำหรับวัดตำแหน่งรอรางจะติดอยู่กับตัวมอเตอร์ สัญญาณเอาต์พุตเป็น ซายน์(sine) ขนาด 0.48 mv



รูปที่ ก.3 เอนโคดเดอร์แบบหมุนสำหรับวัดตำแหน่งมุมและรอราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.1 โปรแกรมที่ใช้ในการทดลองนับค่าด้วยคีย์บอร์ดเมื่อมุมของก้านเพนดูลัมแกว่งเกิน 13 องศา

องศา

```
#define PPORT 0xF300
#define Read inportb(PPORT)
#include<conio.h>
#include<stdio.h>
#include<dos.h>
#include<values.h>
void right()
{
    output(0xf300,0x1);
    delay(3);
    output(0xf300,0x3);
    delay(7);
}
void left()
{
    output(0xf300,0x0);
    delay(3);
    output(0xf300,0x2);
    delay(7);
}

int main(){
char ch='a';
long Counter=0;
clrscr();
do{
    while(Read==0xFC);
    while(Read==0xFD);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(Read==0xFE);
while(Read==0xFF){
    Counter++;
    if(Counter>90)
        left();
        break;
}
while(Read==0xFC);
while(Read==0xFF);
while(Read==0xFE);
while(Read==0xFD){
    Counter--;
    if(Counter<-90)
        right();
        break;
}
gotoxy(1,1);printf("Counter=%4d",Counter);
while(kbhit()){
    ch=getch();
    switch(ch){
        case 'n': Counter=0; break; // ตั้ง ค่า Counter ใหม่
        case 27 : break;
        case 'a': break;
        case 's': break;
    }
}
}while(ch!=27);
outport(PPORT,0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2 โปรแกรมทดสอบการเคลื่อนที่ให้กับรถ

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>

void left();
void right();
void wait();

char control,i;

main()
{ clrscr();
  printf("Press ""ESC"" Key To Exit");
  do{
    right();
    delay(200);

    wait();
    delay(500);

    right();
    delay(300);
    wait();

  }while(!kbhit());
  outport(0xf300,0x00);
}

void left()
{
  outport(0xf300,0x0);
  delay(2);
  outport(0xf300,0x2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delay(8);
}

void right()
{
    outport(0xf300,0x1);
    delay(2);
    outport(0xf300,0x3);
    delay(8);
}

void wait()
{
    outport(0xf300,0x0);
    delay(500);
}

```

ข.3 โปรแกรมทดลองการอินเทอร์รัพท์ ISA ที่ IRQ3

```

#include<stdio.h>
#include<conio.h>
#include<dos.h>
#include<stdlib.h>

void interrupt(oldIrq3)(void);
void interrupt countToggle(void);

    int i = 0;
    long j = 0;

#define IRQ3 0x0B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int main(void)
{
    window(5,5,50,75);
        clrscr();
        gotoxy(1,3);
        printf(" R U Man ? ");

        oldIrq3 = getvect(IQR3);
        setvect(IRQ3,countToggle);

        do{
            outputb(0x21,(inportb(0x21)&0xf7));
            j++;
            gotoxy(27,3);printf("%d\n",j);
        }while(!kbhit());
        setvect(IRQ3,oldIrq3);
        outputb(0x21,(inport(0x21)|0x08);
        printf("\nswitch presses i =%d\n",i);
        printf("j = %d\n",j);
        return 0;
    }

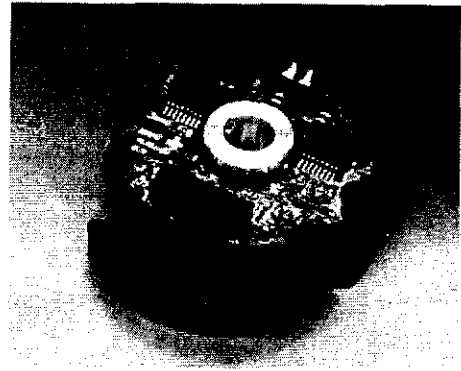
void interrupt countToggle(void)
{
    disable();
    i++;
    outport(0x20,0x20);
    enable();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INCREMENTAL MODULAR ENCODER

IRM3 Series



Outside Diameter: 35mm Length: 14mm Shaft Diameter: 6.0mm/8.0mm

Model Name

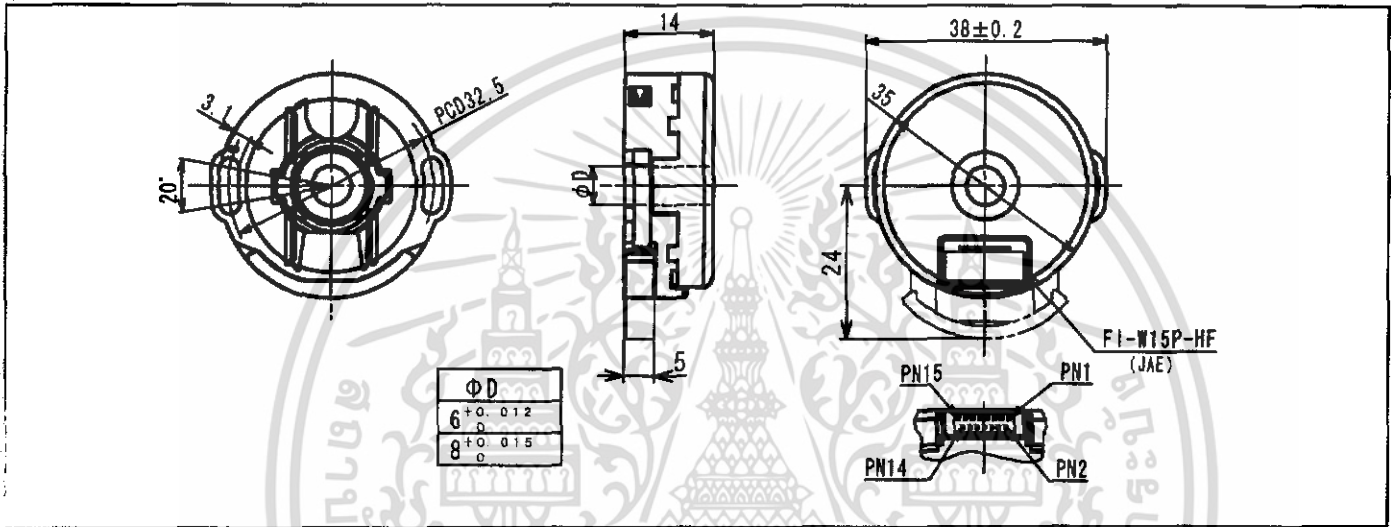
IRM3X0

Model

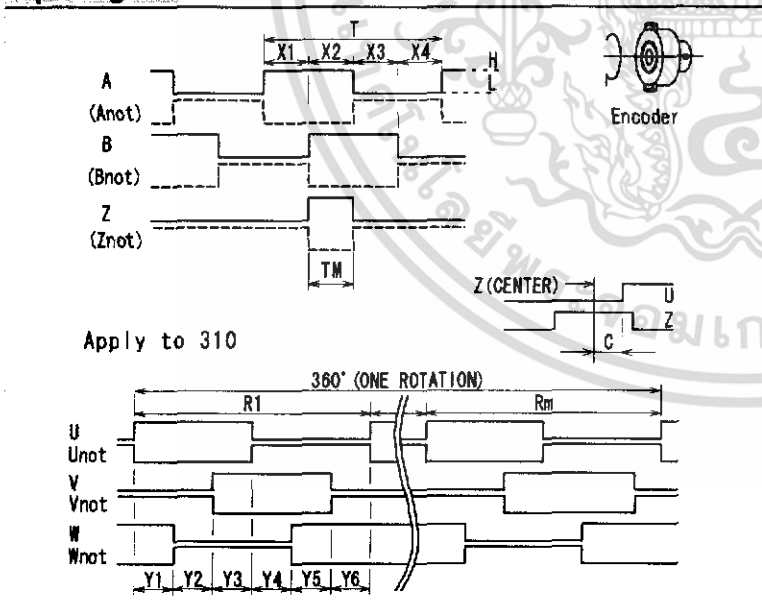
XXXX

Output Pulse (P/R)

Dimensions



Output Signals



Square-wave Accuracy

- $X1+X2=0.5T\pm0.2T$
 - $X2+X3=0.5T\pm0.2T$
 - $Xn\geq0.125T$ (n=1,2,3,4)
 - $TM=0.25T\pm0.125T$
 - $Yn=Rm\%6\pm2^\circ$ (n=1,2,3,4,5,6) (Mechanical Angle)
 - $C\leq\pm2^\circ$ (Mechanical Angle)
- Left-side figure shows positional relationship of A & B channels and Z channel.

Signal Accuracy

Pitch Error: $\pm0.1T$

* $T=360^\circ/Nl$ (Nl: P/R)

* $Rm=360^\circ/Ncs\pm2^\circ$

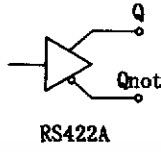
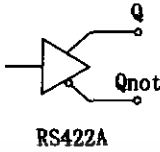
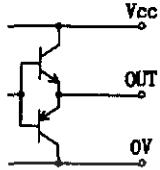
(Ncs: Commutation Signal P/R) (Mechanical Angle)

Wire Connection

| PIN NO. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---------|---|------|---|------|---|------|---|------|---|------|----|------|------|----|----|
| IRM310 | A | Anot | B | Bnot | Z | Znot | U | Unot | V | Vnot | W | Wnot | +5V | 0V | FG |
| IRM320 | A | Anot | B | Bnot | Z | Znot | / | / | / | / | / | / | +5V | 0V | FG |
| IRM360 | A | 0V | B | 0V | Z | 0V | / | / | / | / | / | / | +Vcc | 0V | FG |

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Specifications

| Item/Model | IRM310 | IRM320 | IRM360 | |
|---------------------------------|---|---|---|-----------|
| Output Pulse (P/R) | 500(2), 1000(2), 1000(4), 2000(4) | 250, 256, 500, 512, 1000, 1024, 2000, 2048, 4000, 4096, 8000, 8192 | | |
| Supply Voltage (Vcc) | 5V±0.5V | 5V±0.5V | 10.8V-26.4V | |
| Supply Current (No Load) | ≤70mA | ≤70mA | ≤70mA | |
| Output Type | Line Driver | Line Driver | Complementary | |
| Output Circuit |  |  |  | |
| Output Voltage | H | ≥2.4V | ≥2.4V | ≥Vcc-4.0V |
| | L | ≤0.5V | ≤0.5V | ≤2.0V |
| Output Current | ±10mA | ±10mA | ≤40mA | |
| Max. Applied Voltage | - | - | - | |
| Min. Load Resistance | - | - | - | |
| Rise Time | ≤100ns | ≤100ns | 350ns (Typ.) | |
| Fall Time | ≤100ns | ≤100ns | 50ns (Typ.) | |
| Response Frequency | 0-300kHz (X1) | 0-300kHz (X1) | 0-200kHz (X1) | |
| By-pass Capacitor (0V- Housing) | 0.22μF | 0.22μF | 0.22μF | |
| By-pass Capacitor (Vcc-Housing) | 0.22μF | 0.22μF | 0.22μF | |

Mechanical Specifications

| | | | |
|-------------------|--------------------------|--|-------------|
| Mech. Perm. Speed | 6000 | min ⁻¹ | |
| Moment of Inertia | 0.0839x 10 ⁻⁶ | kg·m ² | |
| Mounted Shaft | Axial Endplay | ±0.254 | mm |
| | Shaft Runout | 0.081 | mm (T.I.R.) |
| | Perpendicularity | 0.05 | mm (R18mm) |
| Max. Accel. Speed | 40000 | rad/s ² | |
| Connector | FI-W15P-HF (JAE) | Applicable Connector: Plug (FI-W15S), (Not Attached) Socket Contact (FI-C3-A1-15000) (Not Attached) | |
| Mass | Approx. 0.02 | kg | |

Environmental Specifications

| | | | |
|-----------------------|-------------|------------------|---|
| Operating Temperature | -30 to +115 | °C | Without Dewfall |
| Storage Temperature | -40 to +125 | °C | Without Dewfall |
| Vibration*1 | 100 | m/s ² | 25-2000Hz, X·Y·Z Each Direction 2h |
| Shock*1 | 1000 | m/s ² | 6ms, Half Sin Pulse, X·Y·Z Each Direction 2 Times |
| Protection Grade*1 | IP40 | Equivalent | With Plastic Cover |

*1 Test Condition

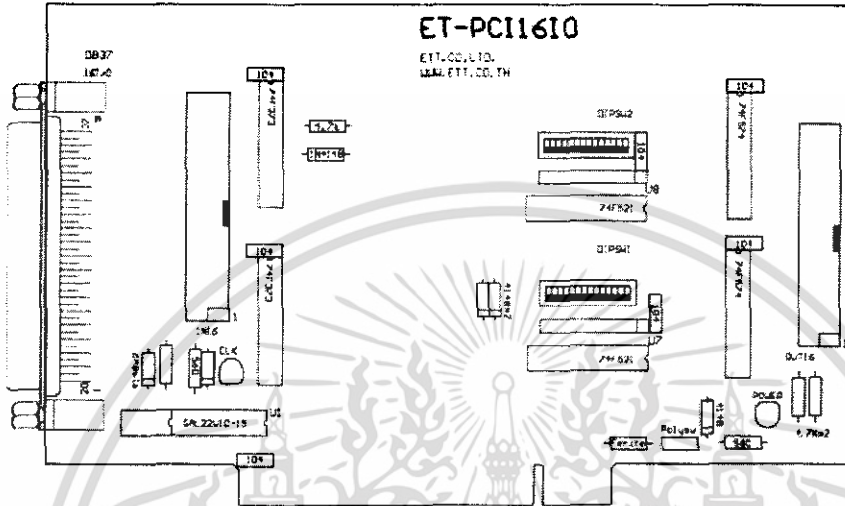
Note: Specifications may be altered without prior notice.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ET-PCI16IO PCI BUS INTERFACE



ET-PCI16IO คือการ์ดต่อขยายระบบคอมพิวเตอร์ PC เพื่อทำหน้าที่เป็นอุปกรณ์รับส่งสัญญาณดิจิทัลกับอุปกรณ์ภายนอก เนื่องจากปัจจุบัน ผู้ผลิตระบบคอมพิวเตอร์ส่วนมากได้ยกเลิกระบบบัสแบบ ISA กับ Mainboard รุ่นใหม่ บริษัท ETT ได้มองเห็นถึงความต้องการในการนำคอมพิวเตอร์ PC ไปใช้ในงานควบคุมยังมีอยู่ จึงได้พัฒนา ET-PCI16IO ซึ่งเป็น Card IO แบบ PCI Bus สามารถนำมาทดแทน ET-PC8255 ซึ่งเป็น CARD IO แบบ ISA Bus ได้



การติดตั้ง ET-PCI16IO กับเครื่องคอมพิวเตอร์ (ควรปฏิบัติตามขั้นตอน ตั้งแต่ ข้อ 1 ถึง ข้อ 6)

1. รันโปรแกรม PCView.EXE จาก แผ่น CDROM ที่ใหม่กับ ET-PCI16IO CARD เพื่อหา ADDRESS ที่ว่างสำหรับ ET-PCI16IO CARD ผู้ใช้ต้องทำการเลือกตำแหน่ง ADDRESS ของ ET-PCI16IO ไม่ให้ตรงกับอุปกรณ์อื่นๆที่อยู่ในระบบของ PC แอดเดรสที่แนะนำให้ใช้งานคือ F300H โปรแกรมสำหรับตรวจสอบ PCI BASE Adress ชื่อโปรแกรม PCView.EXE ผู้ใช้ Windows 2000/NT/XP จะไม่สามารถรันโปรแกรมนี้ได้ให้ข้ามขั้นตอนที่ 1 นี้ไปยังหัวข้อ การใช้งานโปรแกรม PCITREE หน้า 5 (เสร็จแล้วกลับมาต่อขั้นตอนที่ 2) โปรแกรมจะแสดงข้อมูลของ PCI ทั้งหมด ดังรูป

```

PCI View V2.0 Copyright 1999 by PCI Technology Corp.
Build Date: Mai 5 1999, PCI Config Mechanism : #1 BIOS version: 2.016
For help use Pcvview -H
PIC -> PCI Bridges Found : 1

```

| Bus | Slot | Func | Vandor | Device | Rev | Class | Crnd | IO Bar | MemBar |
|-----|------|------|--------|--------|-----|-------------|------|--------|--------|
| 0 | 0 | 0 | 1106h | 3128h | 00 | Host Bridge | 6 | 0 | |
| 0 | 9 | 0 | 11ECh | 8139h | 01 | Ether net | 7 | D000 | |
| 0 | 17 | 0 | 1106h | 3074 | 06 | Isa Bridge | 87 | 0 | |
| 0 | 17 | 0 | 1106h | 0571 | 00 | IDE | 7 | D400 | |

ตรวจสอบส่วนของ IO Bar ว่าไม่ซ้ำกับ Adress ของ ET-PCI16IO โดยทั่วไปแอดเดรสตั้งแต่ 000 H จนถึง 3FFH จะสงวนไว้สำหรับระบบไม่ควรจะใช้งาน และตั้งแต่ 3FFH ขึ้นไปสามารถใช้เป็นแอดเดรสของ PCI ได้แต่ทั้งนี้ต้องดู IO BAR ของ PCI CARD อื่นๆด้วยว่าไม่ซ้ำตำแหน่งกันและ Base Adress ที่เลือกควรจะห่างจาก Base Adress ของ Card อื่นอย่างน้อย 04H เพื่อป้องกันข้อผิดพลาดอื่นๆ

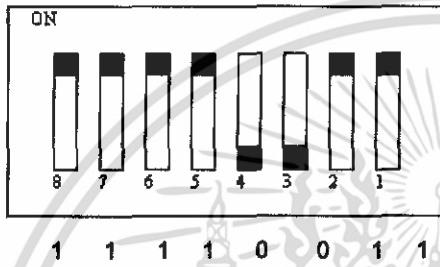
2. หลังจากตรวจสอบ Base Adress ของ ระบบแล้วในกรณีที F300 H ไม่ว่าง หรือ ต้องการเปลี่ยน Adress ของ CARD ผู้ใช้งานต้องทำการ เลือกดิพสวิทช์ใหม่ การเลือกดิพสวิทช์ให้เลือกที่ DIPSW 2 เท่านั้น ส่วน แอดเดรสไบต์ล่างถูกกำหนดมาให้เป็น 00 จากฮาร์ดแวร์

แอดเดรส ที่แนะนำให้ใช้

DIP SW ON = 1 OFF = 0

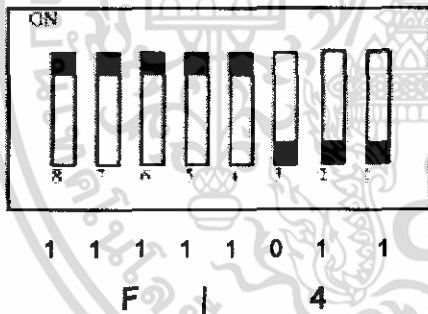
2.1 F300 HEX

DIP SW2



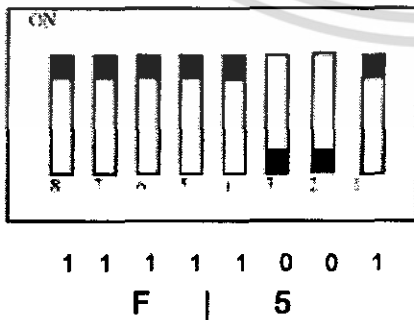
2.2 F400 HEX

DIP SW2



2.3 F500 HEX

DIP SW2



3. สำหรับเครื่องคอมพิวเตอร์รุ่นใหม่ ๆ บางรุ่น จะมีฟังก์ชันตรวจสอบ SLOT PCI ที่ว่างอยู่ และ จะไม่ส่งสัญญาณ CLOCK ไปยัง SLOT ที่ว่างนั้น ผู้ใช้ต้อง Disable ฟังก์ชันนี้ มิฉะนั้นจะไม่สามารถใช้งาน ET-PCI16IO ได้

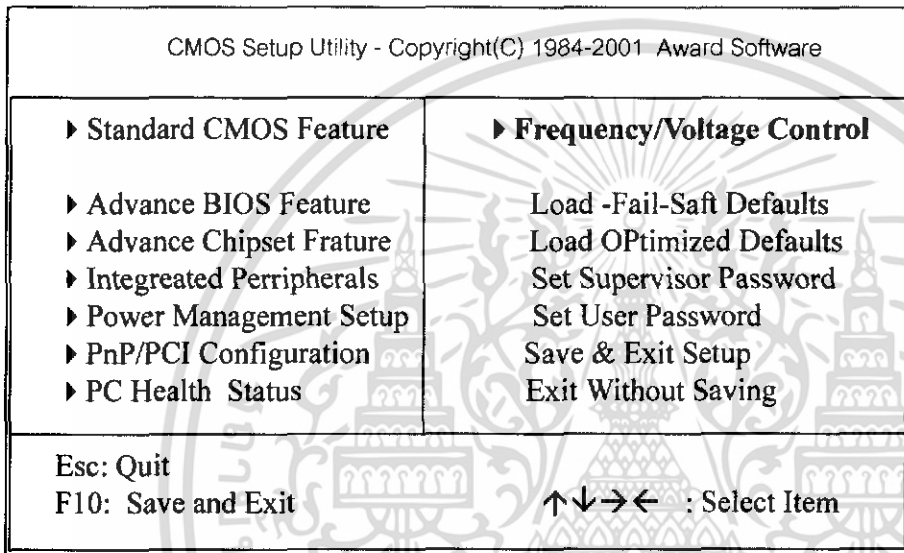
3.1 เข้าโหมด SETUP BIOS กด คีย์ DEL ขณะ BOOT เครื่อง (และอาจใช้คีย์อื่นๆ ก็ได้ในการเข้า SETUP BIOS ในเครื่องบางรุ่น)

3.2 เข้า MENU **CLOCK/ FREQUENCY CONTRL** หรือ **Auto Detect DIMM/PCI clock**

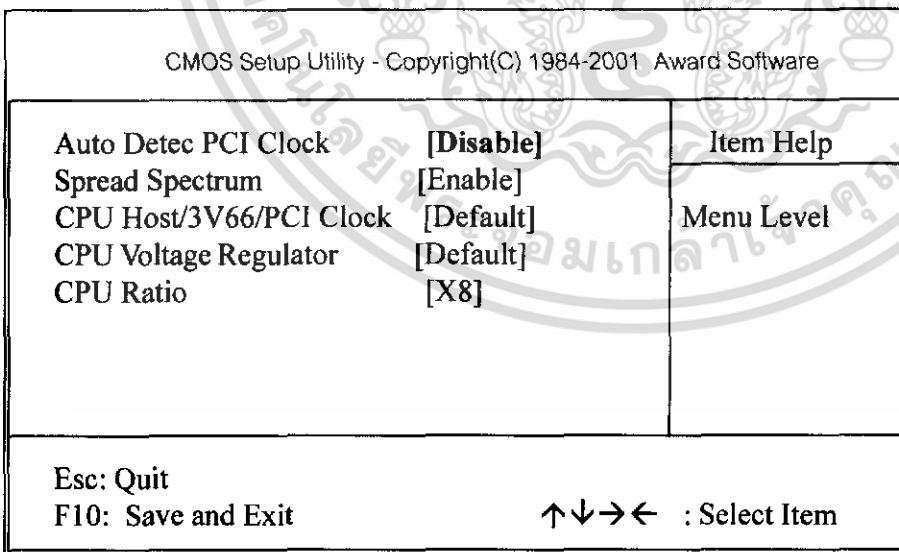
3.3 ทำการ Disable ฟังก์ชัน Empty slot detection

3.4 SAVE ค่า BIOS ที่แก้ไขแล้ว และ ออกจาก BIOS SETUP (SAVE CMOS AND EXIT) สำหรับ การ SETUP BIOS จะแตกต่างกันไปขึ้นอยู่กับ บริษัทผู้ผลิต และรุ่นของ BIOS

สำหรับคอมพิวเตอร์รุ่นใดที่ไม่มีฟังก์ชันนี้ให้ข้ามขั้นตอนในข้อนี้ไปได้



ตัวอย่าง Biosของ บริษัท Award



เลือกให้ฟังก์ชัน Auto Detec PCI Clock เป็น Disable

การติดตั้งและใช้งานโปรแกรม PCI TREE (สำหรับผู้ใช้ Windows2000,NT)

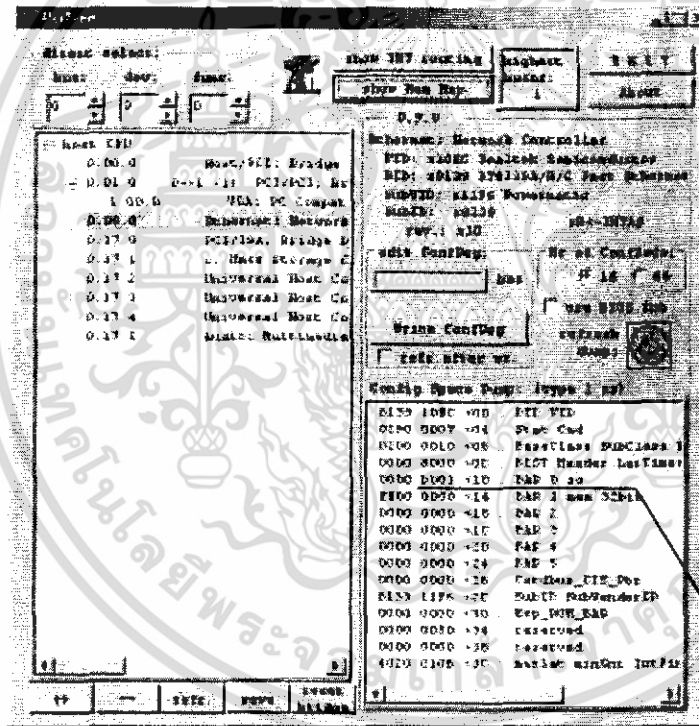
1. ตรวจสอบไฟล์ด้านล่างว่าครบถ้วนและอยู่ที่ Folder เดียวกันกับโปรแกรม PCITREE.EXE (ต้องทำการ ตั้งค่าใน Option ของ Windows Explorer ให้เป็น View All File หน้า 9)

- gbhlp.dll
- mfc42d.dll
- msvcrt.dll
- msvcrt.dll and
- hlp.vxd
- (pcidevs.txt) if present VIDs and DIDs are taken from this list

2. ทำการก๊อปปี้ไฟล์ hlp.sys ดังนี้

Win NT/2000 : hlp.sys must be copied to **WINNT\SYSTEM32\DRIVERS** directory.

3. หลังจากนั้น รันโปรแกรม PCITREE



IO Adress

4. ทำการจดบันทึก ค่า PCI I/O Adress ของแต่ละ PCI Device ทั้งหมด โดยคลิกเมาส์ไปที่แต่ละบรรทัดของ Dialog ด้านขวามือ หากบรรทัดใดมีเครื่องหมายบวกให้คลิกที่เครื่องหมายบวกด้วย จากนั้นสังเกต ข้อมูลของ Dialog ด้านขวามือจะเปลี่ยนไปตามข้อมูลของแต่ละ PCI Device

หมายเหตุ สำหรับ แต่ละ PCI Device ไม่จำเป็นต้องมี IO Adress ทุกตัวเสมอไป

5. เลือกแอดเดรสที่เหมาะสม โดยทั่วไป แอดเดรสตั้งแต่ 00 00 H จนถึง 03 FF H จะสงวนไว้สำหรับระบบไมโครจะใช้งานและ ตั้งแต่ 03 FF H ขึ้นไป สามารถใช้เป็นแอดเดรสของ PCI ได้แต่ทั้งนี้ต้องดู IO BAR ของ PCI CARD อื่นๆด้วยว่าไม่ซ้ำตำแหน่งกันและ Base Adress ที่เลือกควรห่างจาก Base Adress ของ Card อื่นอย่างน้อย 04H เพื่อป้องกันข้อผิดพลาดอื่นๆ

6. ไม่แนะนำให้ใช้งานฟังก์ชันอื่นๆของโปรแกรมนี เพราะอาจจะทำความเสียหายให้กับระบบคอมพิวเตอร์ได้

การทำงานของ ET-PCI16IO

ET-PCI16IO การ์ด ประกอบด้วยสามส่วนคือ ส่วนของ74F573,74F574 ซึ่งทำหน้าที่เป็นอินพุทและเอาพุท ส่วนของวงจร ดีโค้ดเดอร์ ทำหน้าที่ถอดรหัสโปรโตคอลของ PCI BUS ได้แก่ ไอซี PLD GAL22V10 ไอซีเปรียบเทียบลอจิก 74F521 เมื่อเกิดขบวนการอ่านเขียนข้อมูล ไอซี PLD จะทำการตรวจลำดับและรูปแบบของสัญญาณร่วมกับ ไอซีเปรียบเทียบลอจิก 74F521 จากนั้นจะส่งสัญญาณ Clock ให้กับ ไอซี 74F573 หรือ ไอซี 74F574 เป็นการถ่ายโอนข้อมูลระหว่าง PCI databus กับ IO ภายนอก

ลักษณะโดยทั่วไปของ PCI BUS

PCI BUS เป็นสถาปัตยกรรมยุคใหม่ที่มีความสามารถสูงและสามารถถ่ายโอนข้อมูลได้มากมีความซับซ้อนมากกว่า ISA BUS แบบเก่า อุปกรณ์ที่จะนำมาเชื่อมต่อกับ PCI จะต้องได้มาตรฐานตามคุณสมบัติที่ได้กำหนดไว้แล้ว แต่ การออกแบบอุปกรณ์ PCI ชนิด Target only สำหรับนำมาใช้ในงานควบคุมที่ไม่ซับซ้อน นับเป็นทางเลือกอีกอย่างหนึ่ง และทางบริษัท อีทีที ได้ออกแบบให้ คุณสมบัติของการ์ด ใกล้เคียงกับมาตรฐานของ PCI มากที่สุด

สัญญาณอินเทอร์เฟซของ PCI BUS

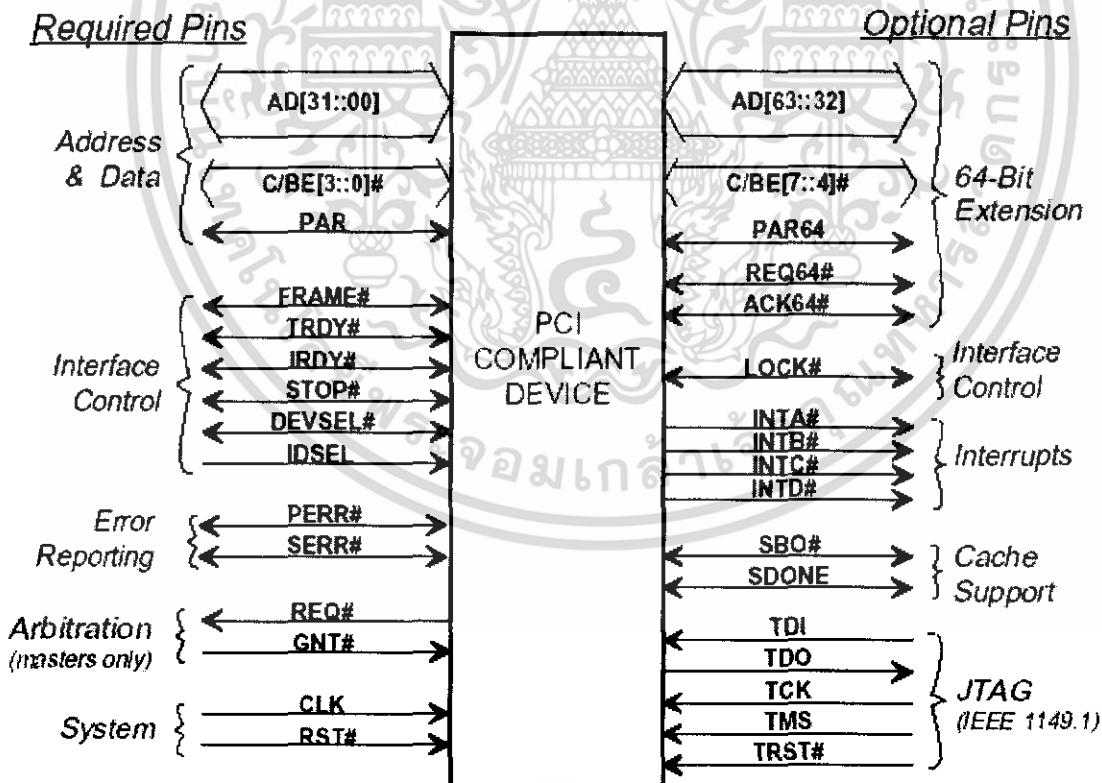


Figure 2-1: PCI Pin List

สัญญาณอินเทอร์เฟสของ PCI BUS สามารถแยกเป็นสองส่วนได้แก่ Option PIN และ Request PIN

Request PIN แบ่งได้ตามหน้าที่ได้ดังนี้

1. SYSTEM PIN

- 1.1 CLOCK Clock ของระบบ PCI Bus การเปลี่ยนแปลงของสัญญาณต่างๆจะมีผลที่ขอบขาขึ้นของ Clock
- 1.2 RESET สัญญาณ RESET

2. ADDRESS and DATA PIN

- 2.1 AD0-AD31 สัญญาณ Address และ Data ใช้งานร่วมกันทั้งหมด 32 บิต
- 2.2 C/BE[0::3] สัญญาณ ชุดคำสั่ง ของบัส เช่น อ่าน เขียน
- 2.3 PAR สัญญาณ ตรวจสอบ Parity

3. Interface Control Pin

- 3.1 FRAME สัญญาณ สำหรับบ่งชี้ว่าเริ่มต้น และ สิ้นสุด ขบวนการถ่ายโอนข้อมูล
- 3.2 IRDY สัญญาณแสดงความพร้อมของ Master
- 3.3 TRDY สัญญาณแสดงความพร้อมของ Target
- 3.4 DEVSEL สัญญาณเลือกอุปกรณ์

Option PIN เป็นสัญญาณสำหรับอุปกรณ์ PCI ความสามารถสูงจะไม่กล่าวถึงในที่นี้

การทำงานของ BUS แบบพื้นฐาน

หลักการถ่ายโอนข้อมูลของ PCI ประกอบด้วย 3 สัญญาณ คือ

- 1. FRAME เป็นสัญญาณจาก Master สำหรับบ่งชี้ว่าเริ่มต้น และ สิ้นสุด ขบวนการถ่ายโอนข้อมูล
- 2. IRDY Initiator Ready เป็นสัญญาณจาก Master สำหรับบ่งชี้ว่า พร้อมที่จะรับส่งข้อมูล
- 3. TRDY Target Ready เป็นสัญญาณจาก Target สำหรับบ่งชี้ว่า พร้อมที่จะรับส่งข้อมูล

เมื่อ Bus อยู่ในสภาวะ Idle สัญญาณ FRAME IRDY จะไม่ ACTIVE

สัญญาณ CLOCK ที่ขอบขาขึ้นลูกแรกเมื่อ สัญญาณ Frame เริ่ม Active เรียกสภาวะนี้ว่า Adress Phase และ สัญญาณ Adress,Command จะถูกถ่ายโอนในสัญญาณ Clock ที่ขอบขาขึ้นนี้ สัญญาณที่ขอบขาขึ้น Clock ถัดไปเรียกสภาวะนี้ว่า Data Phase ในสภาวะนี้ สัญญาณ IRDY TRDY จะ Active สภาวะ Wait Cycle อาจเกิดขึ้นได้ เมื่อสัญญาณ IRDY หรือ TRDY ไม่ Active ขบวนการถ่ายโอนข้อมูลจะสิ้นสุดเมื่อ สัญญาณ FRAME ไม่ Active

ขบวนการ เขียนข้อมูล

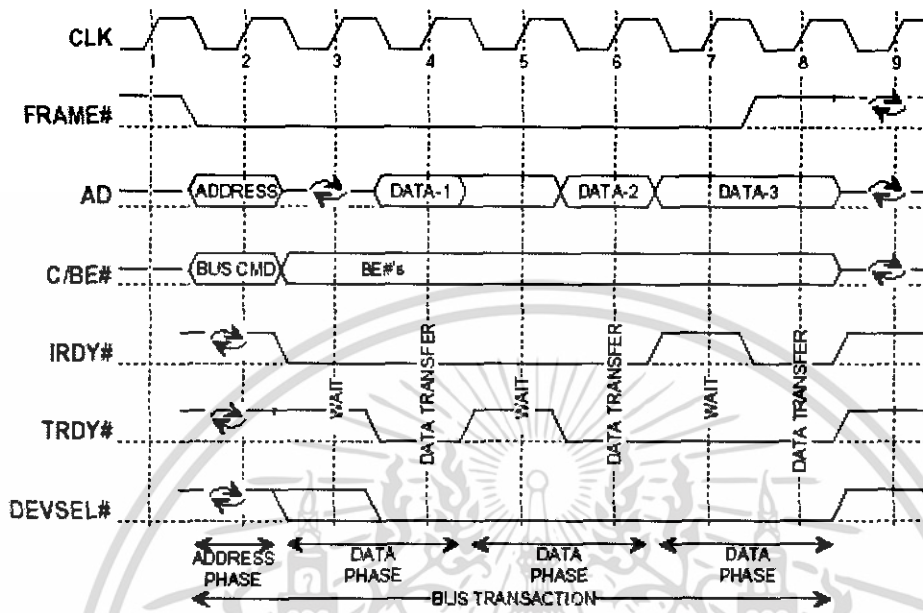


Figure 3-1: Basic Read Operation

ขบวนการอ่านข้อมูล

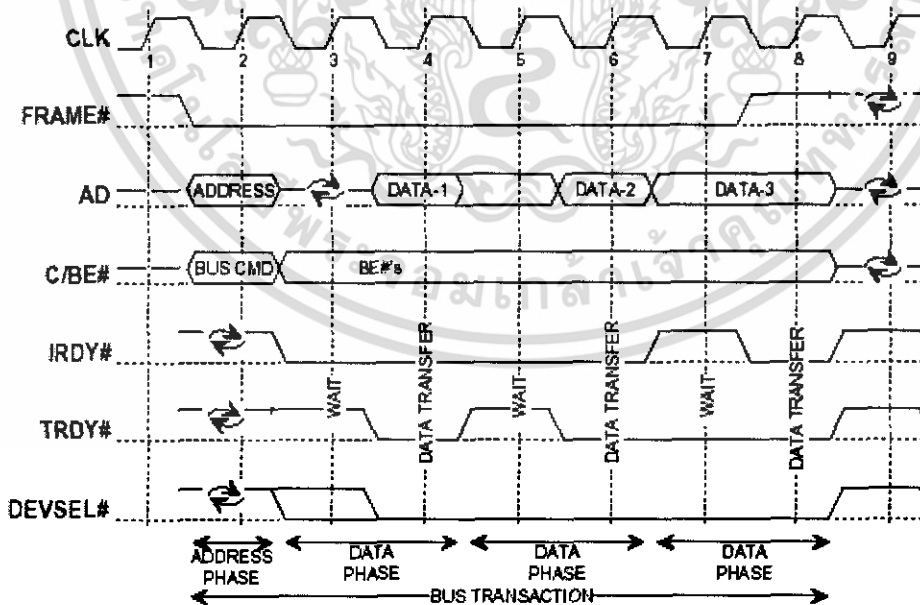


Figure 3-1: Basic Read Operation

การโปรแกรมเพื่อติดต่อกับ ET-PCI16IO

ผู้ใช้งานสามารถสร้าง APPLICATION หรือโปรแกรมใดๆ ที่มีฟังก์ชันสำหรับอินเทอร์พอร์ต ติดต่อกับ ET-PCI16IO ได้โดยตรง แต่ปัจจุบันโปรแกรมสำหรับสร้าง Application สำหรับรันบน Windows ส่วนหนึ่ง ไม่มีฟังก์ชันสำหรับอินเทอร์พอร์ต และ ถึงแม้ว่าบางโปรแกรมจะมีฟังก์ชันสำหรับอินเทอร์พอร์ต ก็ไม่สามารถรัน APPLICATION ที่สร้างขึ้นนั้น กับระบบปฏิบัติการรุ่นใหม่ เช่น Windows NT, windows2000, WindowsXP เนื่องจากข้อจำกัดที่ว่า ระบบปฏิบัติการรุ่นใหม่ ไม่อนุญาตให้ใช้คำสั่งอินเทอร์พอร์ตได้โดยตรงจาก USER Mode แต่สามารถใช้คำสั่งอินเทอร์พอร์ตได้โดยผ่าน Driver Mode เท่านั้น ทั้งนี้ การพัฒนาโปรแกรมในส่วนของ Driver Mode มีความยุ่งยากและซับซ้อน

ดังนั้นทางเลือกอย่างอื่นที่ง่ายกว่าสำหรับผู้ใช้ คือ การเรียกใช้ LIBRARY ที่เขียนขึ้นมาสำหรับอินเทอร์พอร์ต โดยทั่วไปแล้ว LIBRARY เหล่านี้จะถูกพัฒนาจากภาษา C และชุด DDK (Driver Development KIT) จากโปรแกรมเมอร์ที่มีความชำนาญ และข้อดีอีกอย่างหนึ่งของการเรียกใช้ LIBRARY (DLL)คือ ผู้ใช้สามารถใช้โปรแกรมภาษาที่ตนเองถนัดเรียกใช้ DLL LIBRARY เหล่านั้นได้ สำหรับ Library ที่ให้มาพร้อมกับ ET-PCI16IO นี้ เป็น Library ตัวหนึ่งที่ใช้งานได้ดีพอสมควร หรือผู้ใช้อาจจะจัดหา Library ตัวอื่นๆมาทดลองใช้งานกับ ET-PCI16IO ก็ไม่เป็นปัญหาแต่ประการใด

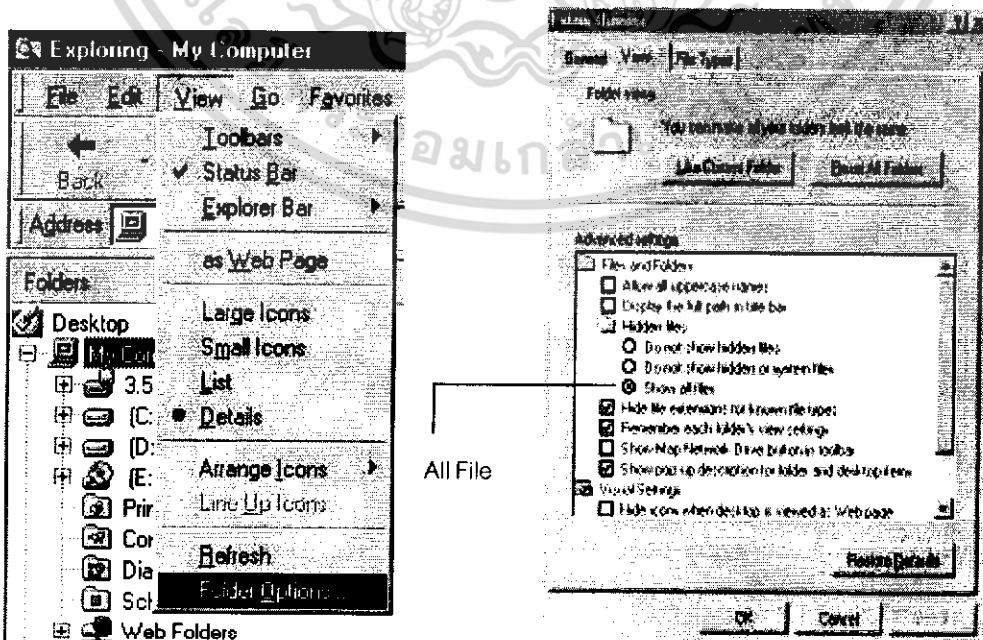
การติดตั้งไดรเวอร์และ การใช้งาน LIBRARY

1. Library ที่มีอยู่บนแผ่น CDROM มีทั้งหมด 3 ไฟล์ เนื่องจากทั้ง 3 ไฟล์เป็น SYSTEM FILE ผู้ใช้ต้องทำการตั้งค่าให้ Windows Exploror มองเห็นไฟล์ SYSTEM จึงจะสามารถ COPY ไฟล์จากแผ่นCDROMได้

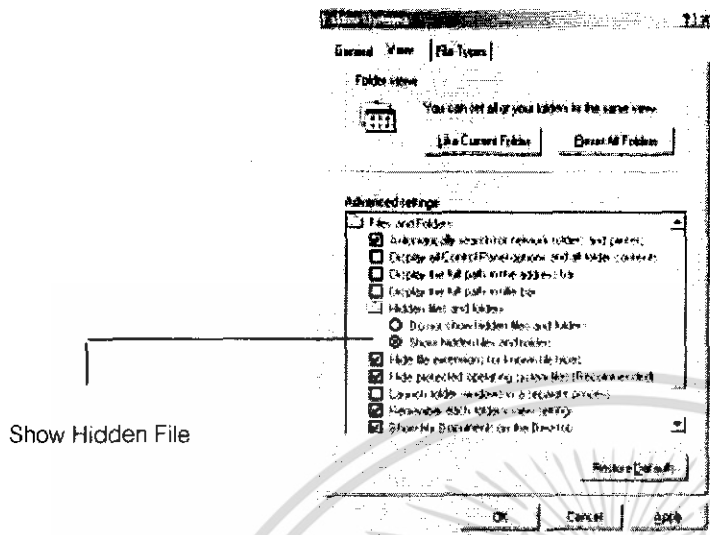
- Klibdrv.SYS ไฟล์สำหรับระบบปฏิบัติการ windows95,98,me
- Klibdrv.VXD ไฟล์สำหรับระบบปฏิบัติการ windowsNT4,2000,XP
- Klibdrv.DLL ไฟล์สำหรับ Compiler ของแต่ละโปรแกรมตัวอย่าง (Visual C , Visual Basic , Delphi)

ควรเลือกใช้เฉพาะ ไฟล์ที่ตรงกับระบบปฏิบัติการที่จะใช้งาน

การ SET ค่า ให้มองเห็น System File สำหรับ windows95,98



การ SET ค่า ให้มองเห็น System File สำหรับ windowsme



2. ทำการ Copy ไฟล์ System Directory ซึ่งจะถูกกำหนดไว้ตอนติดตั้ง Windows

สำหรับ windows 95, windows 98, windows Me

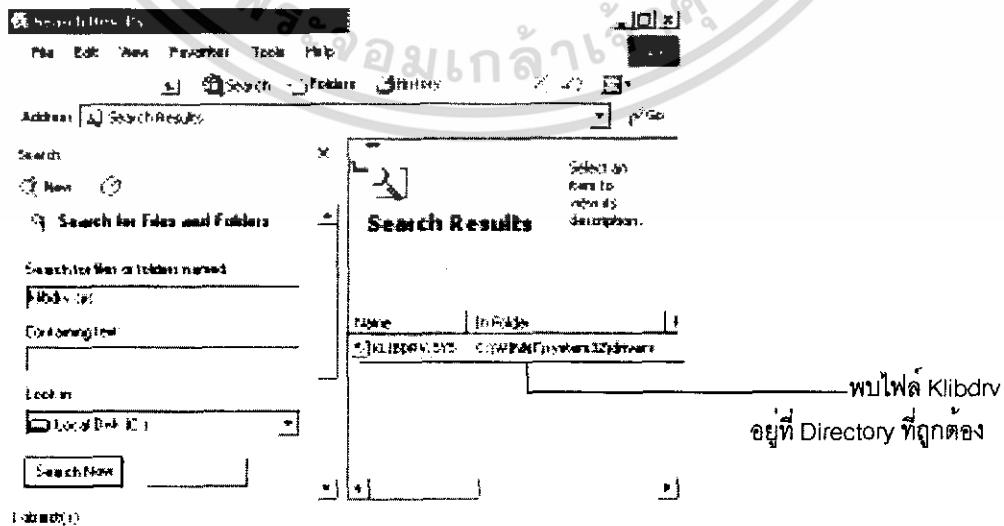
ทำการ Copy ไฟล์ Klibdrv.VXD ไปที่ System Directory ในกรณีติดตั้งที่ไว้ที่ Drive C
C:\Windows\ME\System\

สำหรับ Windows NT4, Windows2000 , Windows XP

ทำการ Copy ไฟล์ Klibdrv.SYS ไปที่ Driver Directory C:\Windows\System32\Driver\

ในกรณีที่เครื่อง PC ที่ใช้งานอยู่มี OS มากกว่า 1 เช่น เครื่องที่สามารถ boot ได้ทั้ง windows XP และ windows 98 จะต้องตรวจสอบ System Directory ของแต่ละระบบให้ถูกต้องว่า ระบบที่ boot นั้นมี System Directory อยู่ที่ Drive และ Path ไດ ก่อนทำการ Copy ไฟล์

3. ทดสอบการว่า copy File ได้ถูกต้อง โดยการ Search File Name Klibdrv.sys หรือ Klibdrv.vxd

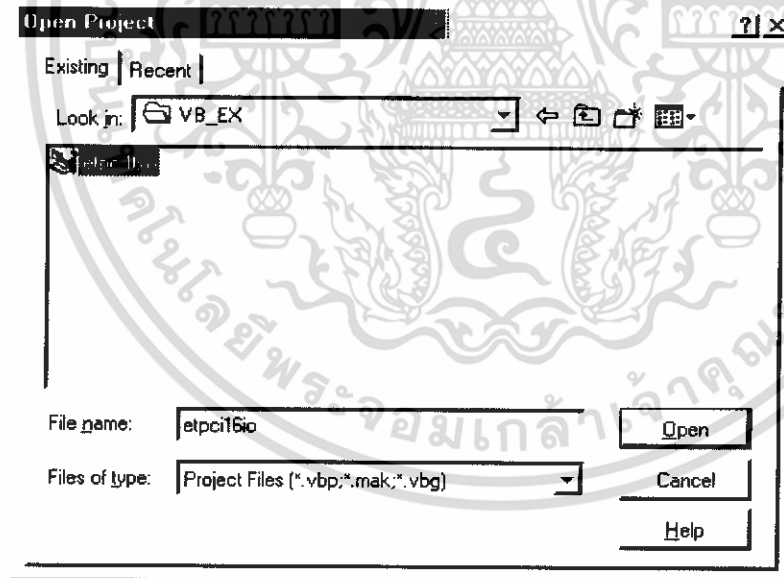


การใช้งานตัวอย่างการโปรแกรมของ Visual Basic Version 6.0

ก่อน copy ตัวอย่างโปรแกรมจากแผ่น CDROM จะต้องตั้งค่าใน Windows Explorer ให้เป็น Show All File ตามตัวอย่างหน้า 9 ไฟล์ตัวอย่างมีทั้งหมด 8 ไฟล์

| Name | Size | Type |
|-------------|-------|--------------------------------|
| etpci16io | 20 KB | Application |
| etpci16io | 1 KB | Visual Basic Project |
| etpci16io | 1 KB | Visual Basic Project Workspace |
| Form1 | 4 KB | Visual Basic Form File |
| KlibDrv.dll | 60 KB | Application Extension |
| KLIBDRV.sys | 24 KB | System file |
| KLIBDRV.VXD | 18 KB | Virtual device driver |
| KlibHWT | 10 KB | Visual Basic Module |

1. Run Application Visual Basic Version 6.0
2. open project et-pci16.vbp

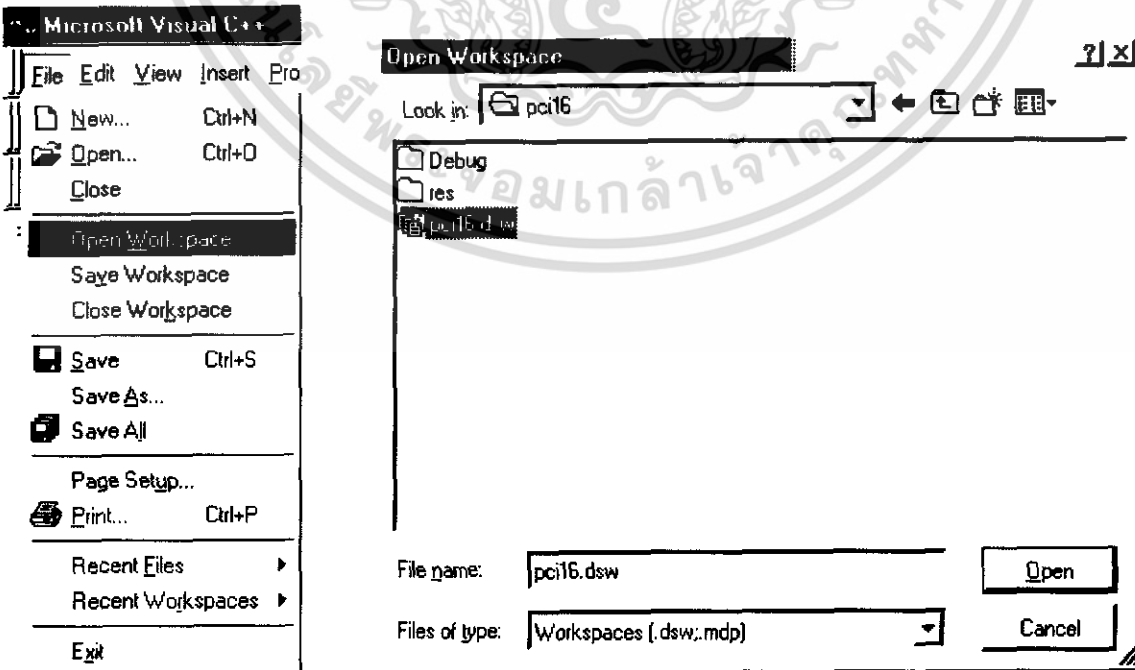


การใช้งานตัวอย่างการโปรแกรมของ Visual C 6.0

ก่อน copy ตัวอย่างโปรแกรมจากแผ่น CDROM จะต้องตั้งค่าใน Windows Explorer ให้เป็น Show All File ตามตัวอย่างหน้า 9 ไฟล์ตัวอย่างมีทั้งหมด 20 ไฟล์ 2 โฟลด์เดอร์

| Name | Size | Type |
|--------------|-------|-----------------------|
| Debug | | File Folder |
| res | | File Folder |
| KlibDrv.dll | 60 KB | Application Extension |
| KlibDrv | 18 KB | LIB File |
| KLIBDRV.SYS | 25 KB | System file |
| KLIBDRV.VXD | 18 KB | Virtual device driver |
| Klibdrv32.h | 11 KB | C Header file |
| pci16.aps | 35 KB | APS File |
| pci16.clw | 2 KB | CLW File |
| pci16.cpp | 2 KB | C++ Source file |
| pci16.dsp | 5 KB | Project File |
| pci16.dsw | 1 KB | Project Workspace |
| pci16.h | 2 KB | C Header file |
| pci16.ncb | 81 KB | NCB File |
| pci16.opt | 48 KB | OPT File |
| pci16 | 2 KB | HTML Document |
| pci16.rc | 6 KB | Resource Template |
| pci16Dlg.cpp | 4 KB | C++ Source file |
| pci16Dlg.h | 2 KB | C Header file |
| resource.h | 1 KB | C Header file |
| StdAfx.cpp | 1 KB | C++ Source file |
| StdAfx.h | 1 KB | C Header file |

1. Run Application Visual C++ 6
2. คลิกที่ File .. Open Work Space
3. คลิกที่ pci16.dsw



การใช้งานตัวอย่างการโปรแกรมของ Delphi 5.0

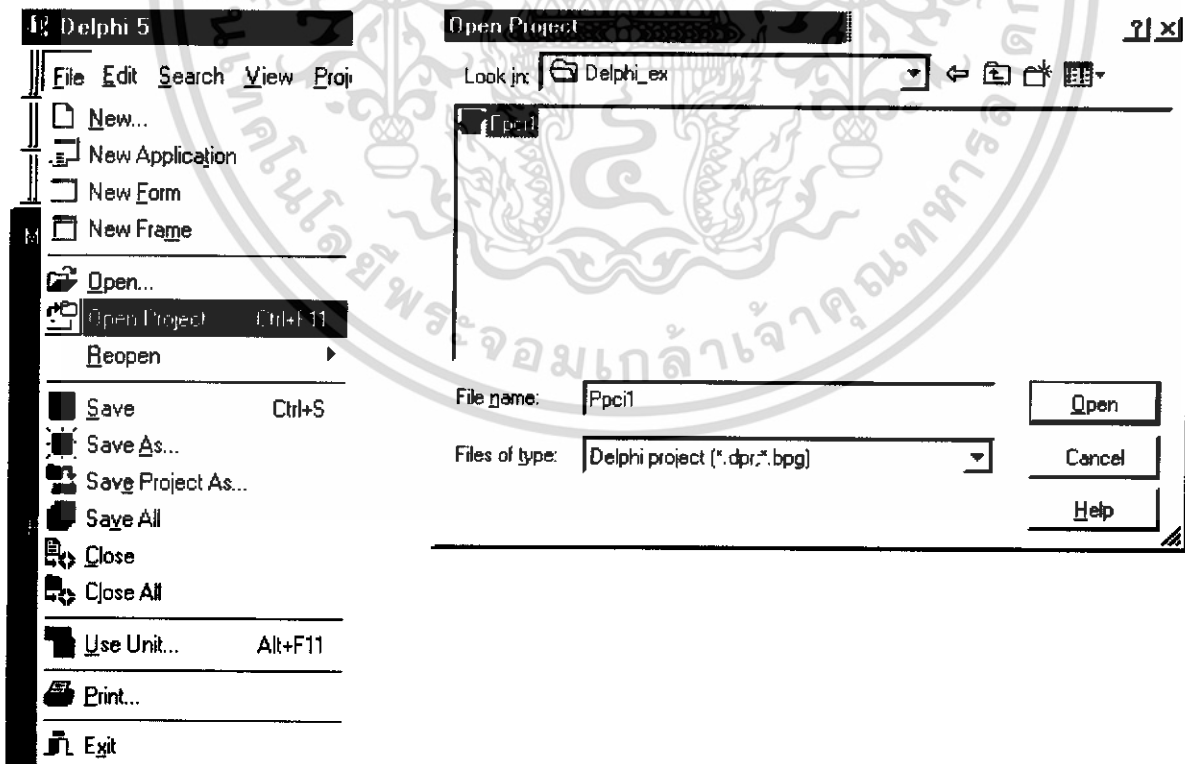
ก่อน copy ตัวอย่างโปรแกรมจากแผ่น CDROM จะต้องตั้งค่าใน Windows Explorer ให้เป็น Show All File ตามตัวอย่างหน้า 9 ไฟล์ตัวอย่างมีทั้งหมด 14 ไฟล์

| Name | Size | Type |
|--------------|--------|-----------------------|
| HW_Types.dcu | 5 KB | DCU File |
| KlibDrv.dcu | 8 KB | DCU File |
| KlibDrv.dll | 100 KB | Application Extension |
| KlibDrv | 19 KB | Delphi Source File |
| KLIBDRV.SYS | 25 KB | System file |
| KLIBDRV.VXD | 19 KB | Virtual device driver |
| pci1.dcu | 7 KB | DCU File |
| pci1 | 3 KB | Delphi Form |
| pci1 | 3 KB | Delphi Source File |
| Ppci1.cfg | 1 KB | CFG File |
| Ppci1.dof | 2 KB | DOF File |
| Ppci1 | 1 KB | Delphi Project |
| Ppci1 | 323 KB | Application |
| Ppci1.res | 1 KB | RES File |

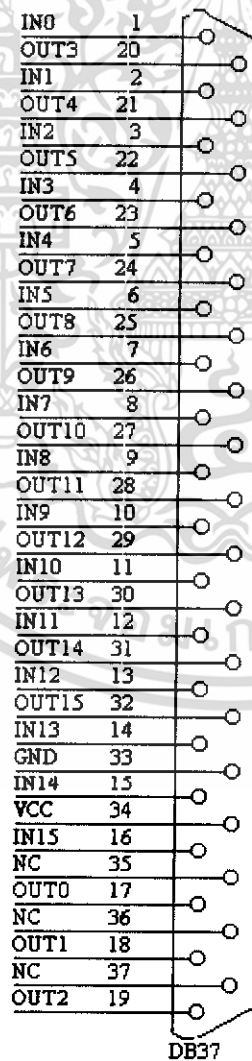
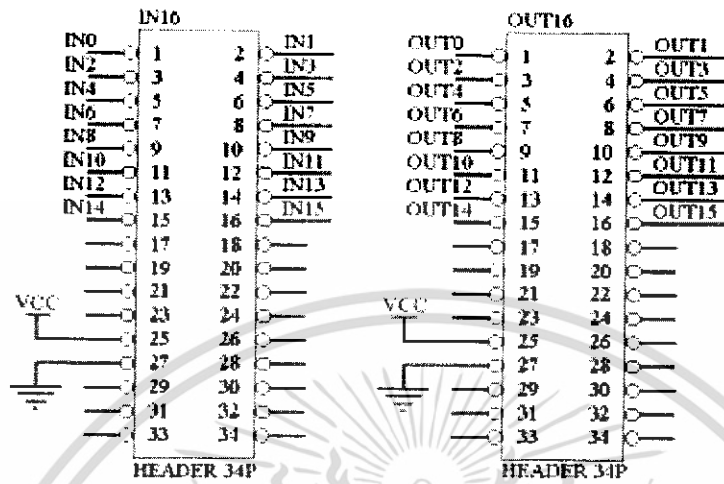
1. Run Application Delphi 5.0

2. คลิกที่ File .. Open Project

3. คลิกที่ Ppci1.dpr



คอนเน็กเตอร์ของ ET-PCI16IO



PCI CONNECTOR

| Pin | 5V System Environment | |
|-----|-----------------------|-----------|
| | Side B | Side A |
| 1 | -12V | TRST# |
| 2 | TCK | +12V |
| 3 | Ground | TMS |
| 4 | TDO | TDI |
| 5 | +5V | +5V |
| 6 | +5V | INTA# |
| 7 | INTB# | INTC# |
| 8 | INTD# | +5V |
| 9 | PRSNT1# | Reserved |
| 10 | Reserved | +5V (I/O) |
| 11 | PRSNT2# | Reserved |
| 12 | Ground | Ground |
| 13 | Ground | Ground |
| 14 | Reserved | Reserved |
| 15 | Ground | RST# |
| 16 | CLK | +5V (I/O) |
| 17 | Ground | GNT# |
| 18 | REQ# | Ground |
| 19 | +5V (I/O) | Reserved |
| 20 | AD[31] | AD[30] |
| 21 | AD[29] | +3.3V |
| 22 | Ground | AD[28] |
| 23 | AD[27] | AD[26] |
| 24 | AD[25] | Ground |
| 25 | +3.3V | AD[24] |
| 26 | C/BE[3]# | IDSEL |
| 27 | AD[23] | +3.3V |
| 28 | Ground | AD[22] |
| 29 | AD[21] | AD[20] |
| 30 | AD[19] | Ground |
| 31 | +3.3V | AD[18] |
| 32 | AD[17] | AD[16] |
| 33 | C/BE[2]# | +3.3V |
| 34 | Ground | FRAME# |
| 35 | IRDY# | Ground |
| 36 | +3.3V | TRDY# |
| 37 | DEVSEL# | Ground |
| 38 | Ground | STOP# |
| 39 | LOCK# | +3.3V |
| 40 | PERR# | SDONE |
| 41 | +3.3V | SBO# |
| 42 | SERR# | Ground |
| 43 | +3.3V | PAR |
| 44 | C/BE[1]# | AD[15] |
| 45 | AD[14] | +3.3V |
| 46 | Ground | AD[13] |
| 47 | AD[12] | AD[11] |
| 48 | AD[10] | Ground |
| 49 | Ground | AD[09] |
| 50 | CONNECTOR KEY | |
| 51 | CONNECTOR KEY | |
| 52 | AD[08] | C/BE[0]# |
| 53 | AD[07] | +3.3V |
| 54 | +3.3V | AD[06] |
| 55 | AD[05] | AD[04] |
| 56 | AD[03] | Ground |
| 57 | Ground | AD[02] |
| 58 | AD[01] | AD[00] |
| 59 | +5V (I/O) | +5V (I/O) |
| 60 | ACK64# | REQ64# |
| 61 | +5V | +5V |
| 62 | +5V | +5V |

