ปริญญานิพนธ์ปีการศึกษา 2541

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การรับข้อมูลจากเครื่องจักรเพื่อนำมาวิเคราะห์

ผู้จัดทำ

1. นายอมร พิริยะแพทย์สม
2. นายอุษิณ วิโรจน์เตชะ
3. นายเอกรินทร์ ผิวเหลืองสวัสดิ์

วัน เดือน ปี -5.ต.ค.2541
เลขทะเบียน 038486
เลขเรียกหนังสือ T 400220 ปม ก

อาจารย์ที่ปรึกษา

(ผศ. พิชิต ลำยอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

038486

การรับข้อมูลจากเครื่องจักรเพื่อนำมาวิเคราะห์

นายอมร พิริยะแพทย์สม
นายอุษิณ วิโรจน์เตชะ
นายเอกรินทร์ ผิวเหลืองสวัสดิ์
ผศ. พิชิต ลำยอง อาจารย์ที่ปรึกษา
ปีการศึกษา 2541

บทคัดย่อ

โครงงานการรับข้อมูลจากเครื่องจักรเพื่อนำมาวิเคราะห์จึงสามารถนำมาใช้ประโยชน์ในการติดต่อสื่อสารกับเครื่องจักร สามารถควบคุมเครื่องจักร ในระยะทางไกล ๆ และ แสดงสถานะของเครื่องจักร ผ่านทางเครื่องคอมพิวเตอร์ ทำให้การดูแล และควบคุมเครื่องจักรในโรงงานอุตสาหกรรมเป็นเรื่องง่าย และสะดวกขึ้น

ในโครงงานจะใช้ เอซีมอเตอร์ (Ac motor) เป็นเครื่องจักร โดยควบคุมทางคอมพิวเตอร์ผ่านทางอินเวอตเตอร์ฟลักซ์เวกเตอร์ ซึ่งมีคุณสมบัติที่ไวต่อการตอบสนองของแรงบิด และความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA MONITORING

Mr. Amorn Phiriyaphetsom
Mr. Ausein Wirotacha
Mr. Akarin Pewluangsawat
Asst. Prof. Pichit Lumyong Advisor
1998

ABSTRACT

The propose of this project is to communicate with machine by using computer. To control and monitor machine in long distance. Display status through computer. It easy to take cares and controls machine in industrial factory

In this project used Ac motor that is machine controlled by computer through inverter flux vector. This is sensitively response for speed and torque.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
สารบัญรูป	III
สารบัญตาราง	IV
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีการทำงานของอินเวอตเตอร์ฟลักซ์เวกเตอร์ (Inverter flux vector)	
2.1 หลักการฟิลด์โอเรียนเตชั่น (Field orientation)	3
บทที่ 3 การการติดต่อระหว่างอินเวอตเตอร์กับคอมพิวเตอร์	
3.1 การทำงานของตัวการ์ด V-C2051 เบื้องต้น	6
3.2 การทำงานของ RS-485	6
3.3 คุณสมบัติของอินเวอตเตอร์ในการติดต่อสื่อสารแบบเครือข่าย	9
บทที่ 4 การใช้คำสั่ง พารามิเตอร์ ควบคุมอินเวอตเตอร์	
4.1 คำสั่งในหมวด 0	24
4.2 คำสั่งในหมวด 1	29
4.3 คำสั่งในหมวด 2	33
4.4 คำสั่งในหมวด 3	47
4.5 คำสั่งในหมวด 4	54
4.6 คำสั่งในหมวด 5	72
4.7 คำสั่งในหมวด 6	79
บทที่ 5 สรุปผล และวิจารณ์	85
ภาคผนวก	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1 แสดงการต่อของอุปกรณ์ทั้งหมด 1
รูปที่ 2.1 แสดงการเปรียบเทียบเวกเตอร์ฟลักซ์ และกระแส
ของมอเตอร์กระแสตรง และมอเตอร์เหนี่ยวนำ 5
รูปที่ 2.2 รูปแสดงหลักการฟิลด์โอเรียนเตชั่นสำหรับเวกเตอร์ไดรฟ์ 5
รูปที่ 3.1 Two wire network 8
รูปที่ 3.2 Four wire network 9
รูปที่ 3.3 แสดงพอร์ต RS-485 ในอินเวอตเตอร์ 9
รูปที่ 3.4 มาตรฐานการส่งข้อมูลในอินเวอตเตอร์ 10
รูปที่ 3.5 โปรโตคอลในคำสั่งอ่าน 10
รูปที่ 3.6 โปรโตคอลในคำสั่ง Successful 11
รูปที่ 3.7 โปรโตคอลในคำสั่ง Unsuccessful 11
รูปที่ 3.8 โปรโตคอลในคำสั่งเขียนใช้ในการเลือกเครื่อง 12
รูปที่ 3.9 โปรโตคอลในคำสั่งเขียนใช้ในการสั่งคำสั่ง 12
รูปที่ 4.1 รูปแสดงพอร์ตการต่อ Preset Speed 30
รูปที่ 4.2 รูปแสดงช่วงการหมุนของมอเตอร์ 32
รูปที่ 4.3 รูปแสดงกราฟความสัมพันธ์ระหว่าง ความเร็ว และเวลา 38
รูปที่ 4.4 แสดงพอร์ตในการต่อ RUN ENABLE 43
รูปที่ 4.5 แสดงกราฟความสัมพันธ์ระหว่าง ความเร็วกับเวลา และแรงบิดกับเวลา 49
รูปที่ 4.6 แสดงช่วงเวลาทำงานของ BCD และ “User” อินพุท 66
รูปที่ 5.1 รูปแสดงอินเวอตเตอร์รุ่น 1100 85
รูปที่ 5.2 รูปอินดักชั่นมอเตอร์ 86
รูปที่ 5.3 รูปแสดงชิ้นส่วนภายในของอุปกรณ์แปลงสัญญาณ 87
รูปที่ 5.4 รูปแสดงการต่อสายภายในของอุปกรณ์แปลงสัญญาณ 88
รูปที่ 5.5 รูปแสดงผลของโปรแกรมในการส่งข้อมูล 90
รูปที่ 5.6 รูปแสดงการทดสอบโปรแกรมในส่วนของกราฟฟิกเบื้องต้น 90
รูปที่ 5.7 รูปแสดงผลการทดสอบสีของตัวอักษรในเท็กซ์ โหมด 91
รูปที่ 5.8 รูปแสดงการทดสอบโปรแกรมในส่วนของอินเตอร์เฟส 91
รูปที่ 5.9 รูปแสดงการทดสอบการรับข้อมูลจาก Keyboard 92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.10 รูปแสดงการเก็บข้อมูลลงฮาร์ดดิสของโปรแกรม 93
รูปที่ 5.11 รูปแสดงรายการข้อมูลที่บันทึกไว้โดยโปรแกรม 93
รูปที่ 5.12 รูปแสดงการรวมโปรแกรมส่วนหลัก 94
รูปที่ 5.13 รูปแสดงการปรับคำสั่ง "ACCEL" และ "DECEL" ที่เวลา 1 วินาที 94
รูปที่ 5.14 รูปแสดงการปรับคำสั่ง "ACCEL" และ "DECEL" ที่เวลา 4 วินาที 95
รูปที่ 5.15 รูปแสดงการมอนิเตอร์ "TORQ" ตามคำสั่ง P201 95
รูปที่ 5.16 รูปแสดงผลการมอนิเตอร์ "COAST" ตามคำสั่งที่ P202 96
รูปที่ 5.17 รูปแสดงการมอนิเตอร์ "DC INJ" ตามคำสั่งที่ P202 96
รูปที่ 5.18 รูปแสดงการมอนิเตอร์ "DC HOLD" ตามคำสั่งที่ P202 97
รูปที่ 5.19 รูปแสดงผวลการมอนิเตอร์คำสั่ง P212 ที่ 0% 97
รูปที่ 5.20 รูปแสดงผลการมอนิเตอร์คำสั่ง P212 ที่ 100% 98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงความสามารถของมาตรฐาน RS-485 กับ RS-232 และ RS-422 7
ตารางที่ 4.1 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 0 15
ตารางที่ 4.2 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 1 16
ตารางที่ 4.3 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 2 18
ตารางที่ 4.4 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 3 19
ตารางที่ 4.5 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 4 21
ตารางที่ 4.6 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 5 22
ตารางที่ 4.7 ตารางแสดงคำสั่งทั้งหมดของหมวด 6 23
ตารางที่ 4.8 ตารางแสดงการตั้งค่า Preset Speed จาก 1-7 29
ตารางที่ 4.9 ตารางแสดงการต่อ TB1-16 , TB1-17 , TB1-18 30
ตารางที่ 4.10 แสดงสถานะของ BUS REG 42
ตารางที่ 4.11 แสดงค่าอัตราบอดของเครื่อง 51
ตารางที่ 4.12 แสดงค่าบิตข้อมูลในการส่ง 51
ตารางที่ 4.13 แสดงค่าพารามิเตอร์ในคำสั่ง "DIR CONTROL" ที่ Local Cmd 54
ตารางที่ 4.14 แสดงค่าพารามิเตอร์ของคำสั่ง "DIR CONTROL" ที่ Remote Cmd 54
ตารางที่ 4.15 แสดงคำสั่งเมื่อ "UVIN MODE" เป็น SCALE 61
ตารางที่ 4.16 แสดงคำสั่งเมื่อ "UVIN MODE" เป็น TRIM 61
ตารางที่ 4.17 แสดงค่าพารามิเตอร์ของคำสั่ง 67
ตารางที่ 4.18 แสดงค่าฟังก์ชั่นของคำสั่ง 68
ตารางที่ 4.19 แสดงความหมายของค่า ฟังก์ชั่น 70
ตารางที่ 4.20 แสดงค่าพารามิเตอร์ของคำสั่ง "INT" 75
ตารางที่ 4.21 แสดงความหมายของการเกิดฟอล์ต 76
ตารางที่ 4.22 แสดงการเปลี่ยนพารามิเตอร์ของคำสั่ง "INVERT" 81
ตารางที่ 4.23 แสดงการเปลี่ยนพารามิเตอร์ของคำสั่ง "INVERT" 81
ตารางที่ 4.24 แสดงการเปลี่ยนพารามิเตอร์ของคำสั่ง "INVERT" 82
ตารางที่ 4.25 แสดงค่าพารามิเตอร์ของคำสั่ง "H.MASK" 82
ตารางที่ 4.26 แสดงค่าพารามิเตอร์ของคำสั่ง "S MASK" 83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

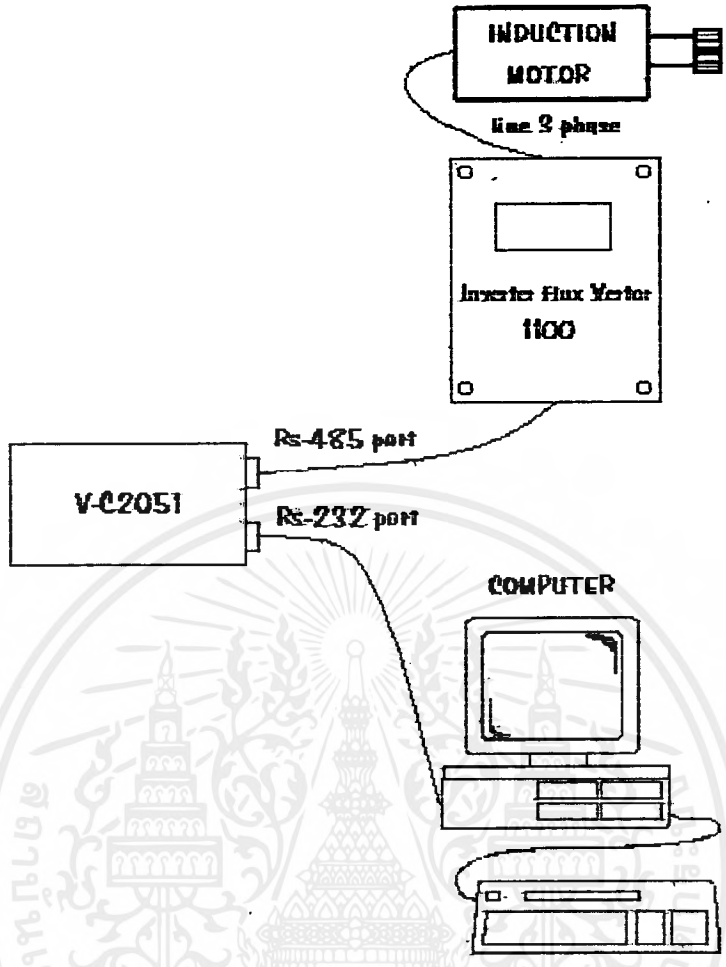
ปัจจุบันอุตสาหกรรมจะต้องแข่งขันกันเพื่อความอยู่รอด การใช้เครื่องจักรแทนมนุษย์ในบางอุตสาหกรรมที่มีอันตราย หรือ งานที่ซ้ำซากจำเจ จะมีการนำเอาคอมพิวเตอร์เข้ามาควบคุมและดูแลอุปกรณ์เครื่องจักรต่างๆ นอกจากจะประหยัด และไม่ต้องยุ่งยากกับปัญหาแรงงานแล้ว ยังเป็นการเพิ่มประสิทธิภาพของงานให้สูงขึ้นอีกด้วย ซึ่งเหมาะกับสภาวะเศรษฐกิจปัจจุบัน และอนาคต

ในการติดต่อกับ อินเวอตเตอร์ (inverter) จะใช้ โปรแกรม (program) Turbo C++ Version 3 ในการติดต่อหลักการพื้นฐานคือการเขียนโปรแกรมการส่งชุดคำสั่งโปรโตคอล (Protocol) ดังที่กล่าวไว้ข้างต้น ผ่านตัวการ์ด V-C2051 ไปยัง microprocessor ของ อินเวอตเตอร์ เมื่อ อินเวอตเตอร์ได้รับคำสั่งแล้วก็จะทำงานตามคำสั่งนั้นๆ โดย
เราสามารถควบคุมมอเตอร์ แสดงผลของกระแส , โวลต์ และ แรงบิด ทุกระยะเวลาที่ต้องการได้ทันที โดยผู้ใช้งานไม่จำเป็นต้องอยู่ในที่ๆ เดียวกับอุปกรณ์ ผู้ใช้จะสามารถควบคุม และ ดูการทำงานของมอเตอร์ในที่ไกล ๆ ได้โดยผ่านการสื่อสารแบบ โมเด็ม หมายความว่าไม่ว่าผู้ใช้งานจะอยู่ที่ใดก็ตามจะสามารถติดต่อกับเครื่องจักรได้ตลอดเวลา

โครงงานนี้จะประกอบด้วยอุปกรณ์ คือ

1. เครื่องคอมพิวเตอร์
2. ตัวการ์ด V-C2051
3. อินเวอตเตอร์ฟลักซ์เวกเตอร์ 1100
4. อินดักชั่น มอเตอร์ (Induction motor) กำลัง 3.5 kW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 แสดงการต่อของอุปกรณ์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีการทำงานของอินเวอตเตอร์ฟลักซ์เวกเตอร์ (Inverter flux vector)

2.1 หลักการฟิลด์โอเรียนเตชั่น [1]

เป็นขบวนการสร้างแรงบิดในมอเตอร์ ใช้หลักการเลียนแบบดีซีมอเตอร์ (Dc motor) โดยสนามแม่เหล็กที่ถูกสร้างโดยสเตเตอร์ (ขดฟิลด์) และโรเตอร์ (ขดอาร์เมเจอร์) จะตั้งฉากกัน และอยู่กับที่ (เป็นเงื่อนไขในการสร้างแรงบิดสูงสุด) ซึ่งเป็นผลมาจากการกระทำของคอมมิวเตเตอร์ และแปรงถ่าน การสร้างแรงบิดในลักษณะนี้จะเกิดขึ้นตลอดเวลา ไม่ว่าตำแหน่งของโรเตอร์จะเปลี่ยนไปอย่างไรก็ตาม เมื่อฟิลด์โอเรินเตชั่นเกิดขึ้น และถูกรักษาไว้ในลักษณะเช่นนี้ ดังนั้นเราสามาถควบคุมแรงบิดของดีซีมอเตอร์ได้ง่ายโดยอาศัยการควบคุมขนาดกระแสอาร์เมเจอร์ ซึ่งเป็นจุดเด่นของดีซีมอเตอร์ในการควบคุมแรงบิด

ในการที่จะควบคุมฟลักซ์เวกเตอร์ไดรฟ์ให้มีสมรรถนะสูง จะต้องใช้หลักการในลักษณะดังกล่าวคือ ง่าย , รวดเร็ว และควบคุมแรงบิดได้โดยตรง ตัวควบคุมฟลักซ์เวกเตอร์ไดรฟ์ (อินเวอร์เตอร์โดยใช้เวกเตอร์คอนโทรล) จะต้องสามารถสร้างปริมาณทางไฟฟ้า (แรงดัน/กระแส , ความถี่ และเฟส) ที่จำเป็นสำหรับทำให้เกิดฟิลด์โอเรียนเตชั่นได้อย่างถูกต้องในมอเตอร์เหนี่ยวนำเพื่อให้มีการควบคุมทอร์ก และฟลักซ์ได้อย่างมีอิสระ และแม่นยำ ไม่ว่าจะเป็นช่วงทรานเซียนต์ หรือสภาวะคงตัว อย่างไรก็ตามการควบคุมฟิลด์โอเรียนเต้นในฟลักซ์เวกเตอร์ไดรฟ์จะเป็นการควบคุมโดยอาศัยวิธีการทางอิเล็กทรอนิกส์ แทนที่จะใช้หลักทางกลเหมือนในดีซีมอเตอร์ที่มีคอมมิวเตเตอร์และแปรงถ่านเป็นตัวกระทำ หลักการสมมูลของการเกิดฟิลด์โอเรียนเตชั่นระหว่างดีซีไดรฟ์ และฟลักซ์เวกเตอร์ไดรฟ์ได้แสดงในรูปที่ 2-1 สามารถเปรียบเทียบได้ดังนี้

(1) Ψ_f ของดีซีมอเตอร์สมมูลย์กับ Ψ_R ของมอเตอร์เหนี่ยวนำ

(2) เวกเตอร์ฟลักซ์ของโรเตอร์ในมอเตอร์เหนี่ยวนำ ia (กระแสอาร์เมเจอร์) ของดีซีมอเตอร์สมมูลย์กับ ia เวกเตอร์กระแสสเตเตอร์ของมอเตอร์เหนี่ยวนำ

พิจารณาทอร์กในดีซีแมชชีน

$$T = K\, i_a \Psi_f \qquad (2.1)$$

$$T = K\, (i_a \times i) \qquad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{หรือ} \quad T = K\, i_a\, i_f \sin 90^\circ \qquad (2.3)$$

ในทำนองเดียวกัน

พิจารณาแรงบิดในมอเตอร์เหนี่ยวนำ

ค่า ia ไม่ได้ตั้งฉากกับ a ดังนั้น

$$T = K\,(i_s \times \Psi_R) \qquad (2.4)$$

$$= K\, i_s\, \Psi_R \sin\theta$$

ค่า θ = มุมระหว่าง i_s และ Ψ_R

จากรูปจะได้

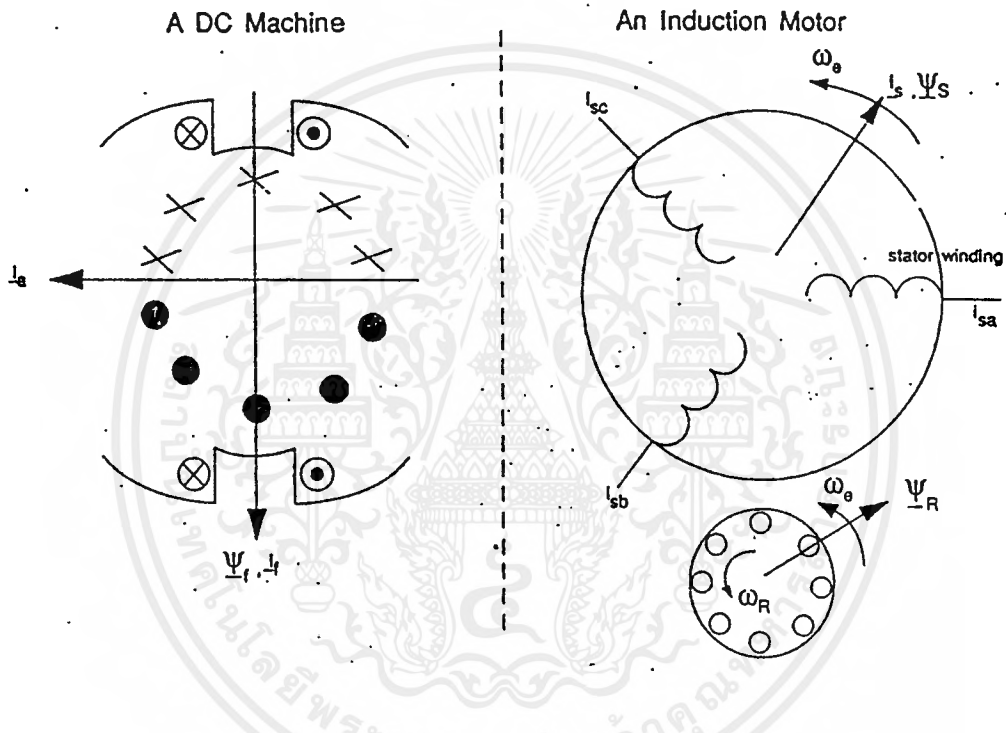
$$T = K i_s \Psi_R \sin\theta$$

$$= K\, i_{sq}\, \Psi_R$$

$$= K'\, i_{sd}\, i_{sq} \qquad (2.5)$$

ดังนั้นจะสามารถควบคุมฟลักซ์ได้จากกระแส i_{sd} และควบคุมแรงบิดได้จากกระแส i_{sq} ถ้ามีการบังคับแกน d ของเฟรม DQ ให้มีการหมุนด้วย ω_θ และวางทับบน Ψ_R ลักษณะเช่นนี้เรียกว่า "ฟิลด์โอเรียนเตชั่น" ดังนั้นฟลักซ์เวกเตอร์จะใช้ตำแหน่งมุมของโรเตอร์ฟลักซ์ (λ ตาม) รูป2-1 ในมอเตอร์เหนี่ยวนำซึ่งเป็นสิ่งสำคัญที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงการเปรียบเทียบเวกเตอร์ฟลักซ์ และกระแสของมอเตอร์กระแสตรง และมอเตอร์เหนี่ยวนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การการติดต่อระหว่างอินเวอตเตอร์กับคอมพิวเตอร์

3.1 การทำงานของตัวการ์ด V-C2051 เบื้องต้น

การ์ด V-C2051 จะมี Port ใช้งานได้ 2 แบบคือ RS-232 และ RS-485 โดยในกรณีของ RS-485 จะใช้ขา To ของ CPU เป็นตัวควบคุมทิศทางของการรับส่ง ถ้า To = 0 จะเป็นการรับข้อมูล และ To = 1 จะเป็นการส่งข้อมูล RS-485 จะมีแนวทางการใช้งานในทำนองเดียวกับ RS-232 แต่จะแตกต่างในด้านของระดับแรงไฟ คือจะใช้ที่ 5V โดยจะเป็น บวก และลบ สลับกันตามค่าของบิท 0 และ 1 ซึ่งจะยังผลให้มีการหักล้างกันเอง ทำให้ลดสัญญาณรบกวนได้เป็นอย่างดี และ RS-485 ยังมีการสื่อสารในแบบ Half Duplex โดยจะต้องรับ และส่งข้อมูลคนละจังหวะไม่สามารถสวนทางกันแบบ RS-232 ได้ จึงทำให้ต้องมีบิทเพื่อการควบคุมการรับ และส่งนั่นเอง และด้วยคุณสมบัตินี้เองจึงทำให้ RS-485 สามารถต่อเป็นระบบ เครือข่าย (Network) ได้ จุดสำคัญของระบบ เครือข่าย ก็คือ การทำโปรแกรมควบคุม (Protocal) นั่นเอง โดยจะต้องควบคุมทิศทาง และการสื่อสารข้อมูลแต่ละจุดในระบบให้ทำงานได้เป็นระเบียบและมีประสิทธิภาพ ซึ่งอธิบายการทำงานของ RS-485 ได้ดังต่อไปนี้

3.2 การทำงานของ RS-485

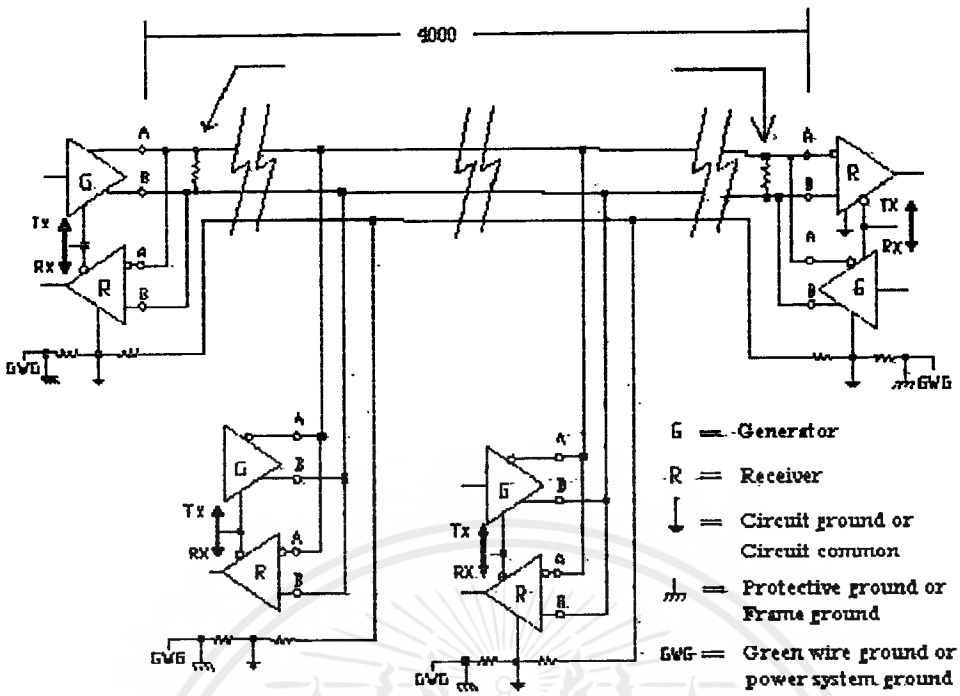
การส่งแบบ RS-485 จะใช้สายข้อมูลแค่ 4 เส้น มี Tx+ , Tx- , Rx+ , Rx- ซึ่งต่างจาก RS-232 ที่มีสาย RTS , CTS , DSR เป็น handchecking นอกเหนือจากสายส่งและรับ การควบคุม RS-485 ซึ่งต้องใช้การ interrupt แตกต่างจาก RS-232 โดยการส่งจะเป็นไปตามรูปที่ 3.1 จะมีสัญญาณที่ตัว G (Generate) และตัว R (Receive) เพื่อที่จะ enable ให้อุปกรณ์ทำหน้าที่ส่งหรือรับข้อมูล นอกจากนี้คุณสมบัติพิเศษของ RS-485 คือ สามารถทำให้อุปกรณ์ อยู่ในสภาวะ disable ได้ โดยจะสามารถตัด จุดรับส่ง ของ เครือข่าย ออกไปชั่วคราว หรือ ถาวรได้ โดยทำการ disable ขา enable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Characteristic	Rs232	Rs-422	Rs-485
Mode Maximum number	Single-end	Differential	Differential
= Driver = Receiver	1 1	1 10	32 32
Maximum cable length	15 m	1200m	1200m
Maximum data rate bit/s	20k	10M	10M
Transmit levels	± 5V min ±15V max	± 2V min	±1.5Vmin
Receive Sensitivity	± 3 V	±0.2V	±0.2V
Load impedance	3k to 7k	100 Ω (Minimum)	60 Ω (Minimum)
Output Current limit	500 mA to VCC or GND	150 mA to GND	150 mA to GND 250 mA to -8V or +12V
Driver load min (Power off)	300 Ω	60 kΩ	120 kΩ

ตารางที่ 3.1 แสดงความสามารถของมาตรฐาน RS-485 กับ RS-232 และ RS-422

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

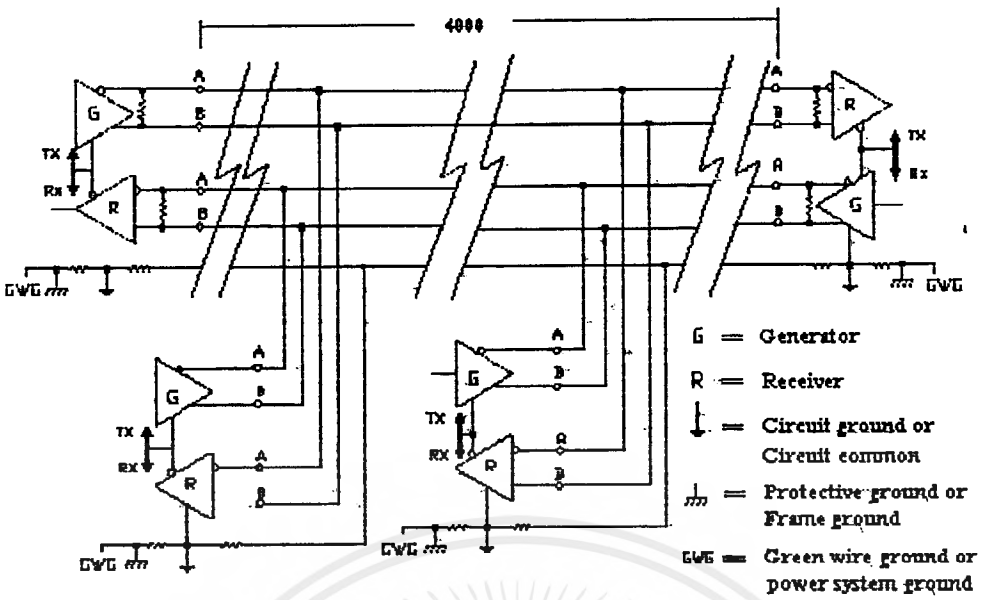


รูปที่ 3.1 Two wire network

ตามมาตรฐาน EIA RS-485 การส่งแบบ RS-485 สามารถเชื่อมต่ออุปกรณ์เป็น เครือข่าย ได้ทั้งหมด 32 ตัวรับและส่ง ในสายส่ง 2 สายเดียวกันได้ สามารถทนแรงดันที่จุดร่วมได้ในช่วง +12 ถึง -7 volt และตัวส่งสามารถ disconnect หรือ tristate จาก line ได้

RS-485 สามารถเชื่อมต่อได้อีกแบบ คือ แบบ four-wire ดูรูปที่ 3.2 ซึ่งการต่อแบบนี้ จะแบ่งเป็นตัว มาสเตอร์ (master) กับ สลาฟ (slave) ระบบ เครือข่าย ตัว มาสเตอร์ จะเป็นตัวติดต่อกับ สลาฟ และ สลาฟ จะติดต่อกับ มาสเตอร์ เท่านั้น ระบบ เครือข่าย แบบนี้มีประโยชน์เมื่อเป็นการติดต่อสื่อสาร โปรโตคอล แบบรวม (mixed protocol) เพราะตัว สลาฟ แต่ละตัวจะไม่ตอบสนองกับสลาฟ ตัวอื่นแต่จะตอบสนองกับ มาสเตอร์

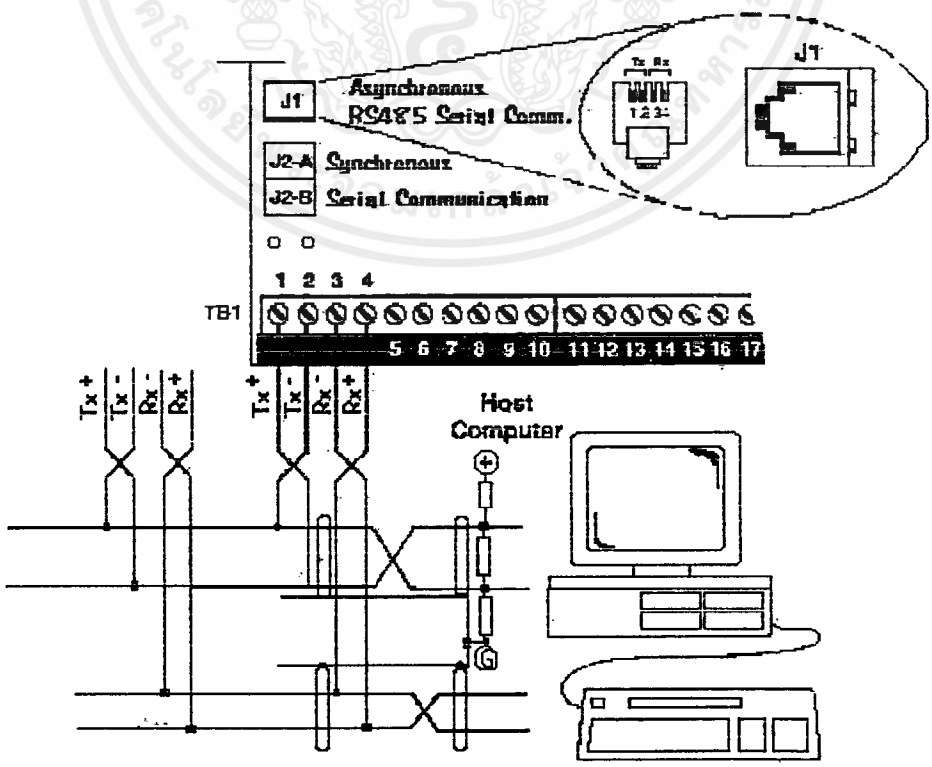
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 Four wire network

3.3 คุณสมบัติของอินเวอตเตอร์ในการติดต่อสื่อสารแบบเครือข่าย

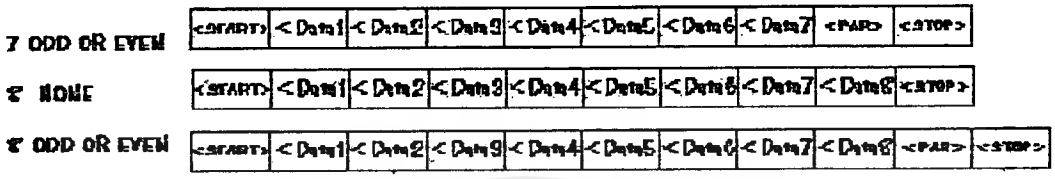
การต่อ อินเวอตเตอร์ กับ การ์ด V-C2051 สามารถต่อขยาย อินเวอตเตอร์ เพิ่มเติมได้สูงสุด 10 ตัว การต่อจะต่อได้ 2 ทาง คือตัวผ่าน terminal box หรือต่อผ่าน port RJ-11 ตามรูป



รูปที่ 3.3 แสดงพอร์ต RS-485 ในอินเวอตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลแบบ โปรโตคอล ที่ใช้ใน Inverter flux vector 1100 ที่จะติดต่อกับ computer จะใช้มาตรฐาน ANSI - x3.22-2.5-A4 ซึ่ง โปรโตคอล ตัวนี้มี 2 คำสั่งพื้นฐานคือ คำสั่ง Read ให้ คอมพิวเตอร์อ่านค่าจาก อินเวอตเตอร์ และ คำสั่ง Write ให้ คอมพิวเตอร์เขียนค่าลงใน อินเวอตเตอร์ รูปแบบของ โปรโตคอล ทั่วไปจะเป็นไปตามรูปที่ 3.4



รูปที่ 3.4 มาตรฐานการส่งข้อมูลในอินเวอตเตอร์

Start bit จะมี logic เป็น 0 ทำหน้าที่เป็นจุดเริ่มต้นของข้อมูลหนึ่ง ๆ บอกให้รู้ว่า bit หลังจากนี้จะเป็นข้อมูล

Data bit เป็นระหัส ASCII ของข้อมูล

Party bit เป็น คู่หรือคี่ เป็น bit check ความถูกต้องในการส่งข้อมูล

stop bit จะมี logic เป็น 1 ทำหน้าที่เป็นจุดสิ้นสุดของข้อมูลหนึ่ง ๆบอกให้รู้ว่าเป็นการจบของข้อมูลหนึ่ง ๆ

คำสั่ง Read การอ่านข้อมูลจาก อินเวอตเตอร์ จะมีตัวอักษรสูงสุด 9 ตัว และมีรูปแบบตามรูป

<EOT>	ADD1	ADD1	ADD2	ADD2	PAR1	PAR2	PAR3	<ENQ>

รูปที่ 3.5 โปรโตคอลในคำสั่งอ่าน

<EOT> เป็นการเริ่มต้นการส่ง เป็นสัญญาณบอกเริ่มต้นการส่ง (เท่ากับ ASCII 04h)

<ADD> เป็น address ของตัว อินเวอตเตอร์ ที่เราต้องการอ่าน มีได้ทั้งหมด 1-99 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ 0011 แทนเครื่องที่ 1 , 0033 แทนเครื่องที่ 3 , ฯลฯ

<PAR> ตัวเลข พารามิเตอร์ (parameter) คือคำสั่งที่สามารถควบคุม อินเวอตเตอร์

ได้ มีอยู่ 7 หมวด คือ P0xx ถึง P6xx ซึ่งจะกล่าวถึงในภายหลัง

<ENQ> End of message แทนการจบ โปรโตคอล (เท่ากับ ASCII 05h)

สัญญาณตอบสนองกลับ จะแบ่งออกเป็น

1. Succcssful อินเวอตเตอร์ จะตอบกลับเมื่อการติดต่อเสร็จสมบูรณ์โดยจะส่งเป็นชุด

โปรโตคอล ดังรูป

<STX>	PAR1	PAR2	PAR3	SPACE/-	VAL1	VAL2	VAL3	===	VAL15	<ETX>	<BCC>

รูปที่ 3.6 โปรโตคอลในคำสั่ง Successful

<STX> bit เริ่มของ โปรโตคอล (เท่ากับ ASCII 02h)

<VAL> เป็นค่าของ พารามิเตอร์ ที่ต้องการรู้

<ETX> bit แสดงว่าการส่งข้อมูลของ อินเวอตเตอร์ เรียบร้อยแล้ว

<BCC> Checksum เป็นการ check ข้อมูลที่ host คอมพิวเตอร์ได้รับว่ามี error ระหว่างการส่งหรือไม่

2. Unsuccessful อินเวอตเตอร์ จะส่ง โปรโตคอล นี้ออกมาเมื่อสัญญาณที่มันรับไม่ถูกต้องตามรูป

<STX>	PAR1	PAR2	PAR3	<EOT>

รูปที่ 3.7 โปรโตคอลในคำสั่ง Unsuccessful

<EOT> เป็นตัวบอกว่าจบ โปรโตคอล

คำสั่ง write เป็นการเขียนค่าให้ อินเวอตเตอร์ โดย อินเวอตเตอร์ จะนำค่าเหล่านั้นไปทำงาน คำสั่ง write ประกอบด้วย 2 ส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง write เป็นการเขียนค่าให้ อินเวอตเตอร์ โดย อินเวอตเตอร์ จะนำค่าเหล่านั้นไปทำงาน คำสั่ง write ประกอบด้วย 2 ส่วนคือ

1. ส่วนที่เลือกเครื่อง อินเวอตเตอร์ มีรูปแบบ โปรโตคอล ดังนี้

<EOT>	ADD1	ADD1	ADD2	ADD2

รูปที่ 3.8 โปรโตคอลในคำสั่งเขียนใช้ในการเลือกเครื่อง

<EOT> เป็นการเริ่มต้นการส่ง เป็นสัญญาณบอกเริ่มต้นการส่ง (เท่ากับ ASCII 04h)

ADD ส่งค่า address ของ อินเวอตเตอร์ ตัวที่ต้องการสั่งงาน ซึ่งมีค่าตั้งแต่ 1 ถึง 99

ตัวอย่างเช่น ถ้าต้องการสั่ง อินเวอตเตอร์ ตัวที่ 1 ทำงาน ค่า ADD จะเท่ากับ

0011 หรือ ถ้าต้องการสั่ง อินเวอตเตอร์ ตัวที่ 15 ทำงาน ต้องส่งค่า ADD เท่ากับ 1155 เป็นต้น

2. ส่วนของคำสั่งมีรูปแบบ โปรโตคอล ประกอบด้วยตัวอักษรสูงสุด 22 ตัว ตามรูป

<STX>	PAR1	PAR2	PAR3	SPACE	VAL1	VAL2	VAL3	===	VAL15	<ETX>	<BCC>

รูปที่ 3.9 โปรโตคอลในคำสั่งเขียนใช้ในการสั่งคำสั่ง

<STX> bit เริ่มของ โปรโตคอล (เท่ากับ ASCII 02h)

<PAR> ตัวเลข พารามิเตอร์ คือคำสั่งที่สามารถควบคุม อินเวอตเตอร์ ได้ มีอยู่ 7 หมวดคือ P0xx ถึง P6xx

<VAL> เป็นค่าของ พารามิเตอร์ ที่ต้องการสั่งงาน

<ETX> bit แสดงว่าการส่งข้อมูลของ อินเวอตเตอร์ เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณตอบสนองกลับ จะแบ่งออกเป็น

การตอบสนองกลับเมื่อการส่งสำเร็จ ถ้าคำสั่งที่ส่งไปยัง อินเวอตเตอร์ และ อินเวอตเตอร์ จะตอบกลับด้วยระหัส ASCII เป็น <ACK> (เท่ากับ 06h)

การตอบสนองกลับเมื่อการส่งไม่สำเร็จ ถ้าคำสั่งที่ส่งไปไม่ถูกต้องตามรูปแบบโปรโตคอล หรือ ตัว checksum ผิดพลาด (หมายถึงเกิด error ระหว่างการส่งข้อมูล) จะมีสัญญาณตอบกลับเป็นระหัส ASCII 15h ในกรณีนี้ข้อมูลที่ อินเวอตเตอร์ จะไม่เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การใช้คำสั่ง พารามิเตอร์ ควบคุมอินเวอตเตอร์

การใช้คำสั่ง พารามิเตอร์ ควบคุม อินเวอตเตอร์

ค่าของ พารามิเตอร์ ที่ป้อนเข้าไปใน โปรโตคอล จะมีความหมายแบ่งออกเป็น 7 หมวดดังนี้

1. P0xx = Main Parameter Menu Summary
2. P1xx = Preset Parameter Menu Summary
3. P2xx = Setup Parameter Menu Summary
4. P3xx = Drive Parameter Menu Summary
5. P4xx = Terminal Parameter Menu Summary
6. P5xx = Status Values Menu Summary
7. P6xx = Tune Parameter Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมวดที่ 0 Main Parameter Menu Summary (P0xx)

Parameter Number	Parameter Name	Default Value	Value Range Min / Max	Units	Run Edit
P000	SETPT :	0 RPM	P001 / P002	[ENG]	Yes
P001	MIN :	0	-2 x base / P002 - 1	[ENG]	Yes
P002	MAX :	1800	P001 + 1 / 2 x base	[ENG]	Yes
P003	ACCEL :	3	0.1 / 3200	S	Yes
P004	DECEL :	3.0	0.1 / 3200	S	Yes
P005	JOG :	100 RPM	P001 / P002	[ENG]	Yes
P006	JOG ACCEL :	3.0	0.1 / 999	S	Yes
P007	JOG DECEL :	3.0	0.1 / 999	S	Yes
P008	I LIMIT :	Rated overload	I Field / Rated overload	A_{rms}	Yes
P009	ENG CHAR 1 :	R	0x20 / 0x7e	None	Yes
P010	ENG CHAR 2 :	P	0x20 / 0x7e	None	Yes
P011	ENG CHAR 3 :	M	0x20 / 0x7e	None	Yes
P012	ENG DEC POINT :	0	0 / 3	None	Yes
P013	ENG SCALE :	1800	100 / 9999	[ENG]	None
P014	SECURITY :	OFF	OFF / ON	None	Yes
P015	DEFLT MENU :	No	No / Yes	None	None

ตารางที่ 4.1 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมวดที่ 1 Preset Parameter Menu Summary (P1xx)

Parameter Number	Parameter Name	Default Value	Value Range Min./Max.	Units	Run Edit
P100	PS1 :	0 RPM	P001 / P002	[ENG]	Yes
P101	PS1 ACCEL :	3.0	0.1 / 3200	S	Yes
P102	PS1 DECEL :	3.0	0.1 /3200	S	Yes
P103	PS2 :	0	P001 / P002	[ENG]	Yes
P104	PS2 ACCEL :	3.0	0.1 / 3200	S	Yes
P105	PS2 DECEL :	3.0	0.1 / 3200	S	Yes
P106	PS3 :	0 RPM	P001 / P002	[ENG]	Yes
P107	PS3 ACCEL :	3.0	0.1 / 3200	S	Yes
P108	PS3 DECEL :	3.0	0.1 / 3200	S	Yes
P109	PS4 :	0 RPM	P001 / P002	[ENG]	Yes
P110	PS4 ACCEL :	3.0	0.1 / 3200	S	Yes
P111	PS4 DECEL :	3.0	0.1 / 3200	S	Yes
P112	PS5 :	0 RPM	P001 / P002	[ENG]	Yes
P113	PS5 ACCEL :	3.0	0.1 / 3200	S	Yes
P114	PS5 DECEL :	3.0	0.1 /3200	S	Yes
P115	PS6 :		P001 / P002	[ENG]	Yes
P116	PS6 ACCEL :	3.0	0.1 / 3200	S	Yes
P117	PS6 DECEL :	3.0	0.1 / 3200	S	Yes
P118	PS7 :	0 RPM	P001 / P002	[ENG]	Yes
P119	PS7 ACCEL :	3.0	0.1 / 3200	S	Yes
P120	PS7 DECEL :	3.0	0.1 / 3200	S	Yes
P121	SKIP FREQ 1 :	10	0 / 120	Hz	Yes
P122	SKIP 1 BAND :	0	0 / 30	Hz	Yes
P123	SKIP FREQ 2 :	10	0 / 120	Hz	Yes
P124	SKIP 2 BAND :	0	0 / 30	Hz	Yes
P125	SKIP FREQ 3 :	10	0 / 120	Hz	Yes
P126	SKIP 3 BAND :	0	0 / 30	Hz	Yes
P127	DEFLT MENU :	NO	No / Yes	None	None

ตารางที่ 4.2 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมวดที่ 2 Drive Parameter Menu Summary (P2xx)

Parameter Number	Parameter Name	Default Value	Value Range Min / Max	Units	Run Edit
P200	START MODE :	MAN	MAN / LINE	None	Yes
P201	RUN MODE :	M SPEED	M SPEED S SPEED TORQ	None	Yes
P202	STOPMODE :	DECEL	DECEL COAST DC INJ DC HOLD	None	Yes
P203	SETPT SRC :	KEYPAD	KEYPAD / ANALOG	None	Yes
P204	KPAD ENABLE :	ENA	ENA (ble) DIS (able) DAL (disable all)	None	Yes
P205	SYN SER :	OFF	OFF SEND RECEIVE	None	No
P206	SLAVE SOURCE :	ANA	ANA (log) SCI	None	No
P207	ON DEL :	0.10	0.1 / 99.99	S	No
P208	AD DEL :	0	0.1 / 99.99	S	Yes
P209	RATIO NUM :	1	1 / 9999	None	Yes
P210	RATIO DEN :	1	1 / 9999	None	Yes
P211	SPINNING MOTOR :	D	D (isable) / E (nable)	None	Yes
P212	S RAMP :	0 %	0 / 100	%	Yes
P213	SRAMP M :	POSITION	POSITION / ALWAYS	None	Yes
P214	F R MODE :	NO REV	NO REV ALWAYS STOPPED JOG	None	Yes
P215	INJBRAKE TIME :	2	1 / 99	S	Yes
P216	INJBRAKE I :	30 %	0 / 100	%	Yes
P217	TRIP RESTARTS :	0	0 / 7	None	Yes
P218	RESTART TIME :	10	1 / 99	S	Yes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไปจำหน่าย หรือใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Parameter Number	Parameter Name	Default Value	Value Range Min / Max	Units	Run Edit
P219	BUS REG :	OFF	MID HIGH OFF LOW	None	Yes
P220	RUN ENABLE :	STOP	STOP / COAST	None	Yes
P221	START INIT :	ON	ON / OFF	None	Yes
P222	LANGUAGE :	ENGL	ENGL(ish) SPAN(ish) FREN(ch) GERM(an)	None	Yes
P223	SET USER DEFLT :	N	N(o) / Y(es)	None	No
P224	DEFLT MENU :	NO	NO / YES	None	No
P225	DEFLT ALL FACT :	N	N(o) / Y(es)	None	No
P226	DEFLT ALL USER :	N	N(o) / Y(es)	None	No

ตารางที่ 4.3 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมวดที่ 3 Terminal Parameter Menu Summary (P3xx)

Parameter Number	Parameter Name	Default Value	Value Range Min / Max	Units	Run Edit
P300	MTR AMPS :	Rated overload / 1.5	20 % / 83 %	Arms	No
P301	MOTOR VOLTS :	230 or 460 ,or 575	10 / 999	Vrms	No
P302	BASE FREQ :	60	10 / 120	Hz	No
P303	LIMIT RPM :	1800	1 / 7200	Rpm	Yes
P304	EXT AMPS :	105 %	10 / 105	%	No
P305	EXT FAULT :	TRIP	TRIP , FOLD	None	No
P306	MAX F TORQ :	100	0 / 100	%	Yes
P307	MAX R TORQ :	100	0 / 100	%	Yes
P308	MAX F BRAKE :	100	0 / 100	%	Yes
P309	MAX R BRAKE :	100	0 / 100	%	Yes
P310	NUM POLES :	4	2 , 4 , 6 , ... , 18	None	No
P311	TORQ SOURCE :	INT	INT / EXT	None	Yes
P312	INT TORQ :	%	0 / 100	%	Yes
P313	SERIAL BAUD :	96	96(00) 48(00) 12(00) 192(00)	None	Yes
P314	SERIAL ADDR :	1	0 / 99	None	Yes
P315	BITS PAR :	8 NONE	8 NONE 8 EVEN 8 ODD 7 EVEN 7 ODD	None	Yes
P316	CHECK SUM :	EN CHK	EN CHK DIS CHK	None	Yes
P317	MDROP MODE :	D	D , E , MD , ME	None	Yes
P318	DEFLT MENU :	NO	NO / YES	None	No

ตารางที่ 4.4 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการ
038486

หมวดที่ 4 Terminal Parameter Menu Summary (P4xx)

Parameter Number	Parameter Name	Default Value	Value Range Min / Max	Units	Run Edit
P400	SPARE :	DISABLE	DISABLE/ENABLE	None	Yes
P401	DIR CONTROL :	AUTO	AUTO UNIP(olar) BIP(olar)	None	Yes
P402	L HI :	1800 RPM	0 / 32767	[ENG]	Yes
P403	L LO :	0 RPM	-32767 / 32767	[ENG]	Yes
P404	L OFFSET :	NOMINAL	DONE Z SET P SET M SET INVERSE NOMINAL	None	Yes
P405	L OFFSETV :	0	-40000 / 40000	DAC	Yes
P406	L POS X :	32767	0 / 80000	DAC	Yes
P407	L NEG X :	32767	0 / 80000	DAC	Yes
	" Remote Cmd " Analog Scaling				
P408	R HI :	1800 RPM	1 / 32767	[ENG]	Yes
P409	R LO :	0 RPM	0 / 32767	[ENG]	Yes
P410	R OFFSET :	NOM4 20	DONE Z SET P SET INVERSE NOM4 20 (mA) NOM0 20 (mA)	None	Yes
P411	R OFFSETV :	6553	-40000 / 40000	DAC	Yes
P412	R POS X :	26213	0 / 80000	DAC	Yes
	" User Vin " Analog Scaling				
P413	USER VIN :	OFF	P000 , ... , P312	None	Yes
P414	USER HI :	100 %	-100 / 100	%	Yes
P415	USER LO:	0 %	-100 / 100	%	Yes
P416	UVIN MODE :	SCALE	SCALE , TRIM	None	Yes
P417	UVIN OS :	NOMINAL	DONE Z SET P SET	None	Yes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Parameter Number	Parameter Name	Default Value	Value Range Min / Max	Units	Run Edit
P418	UVIN OV :	0	-40000 / 40000	DAC	Yes
P419	UVIN PX :	32767	0 / 80000	DAC	Yes
" Local Vout " Analog Scale					
P420	LOCAL VOUT :	SPEED	SPEED TORQ PTR	None	Yes
P421	LVOUT HI :	100	-999999 / 8388607	%	Yes
P422	LVOUT LO:	0	-8388607 / 8388607	%	Yes
" User Vout " Analog Scale					
P423	USER VOUT :	P503	P500 , ... , P512	None	Yes
P424	UVOUT HI :	100	-999999 / 8388607	None	Yes
P425	UVOUT LO :	0	-8388607 / 8388607	None	Yes
P426	UIN :	NONE	NONE COAST CTL SLAVE CTL ZERO TORQ SPEED TORQ FORWARD FREEZE BCD	None	Yes
User Relay					
P427	USER RELAY :	P500	P500 , ... , P514	None	Yes
P428	UR SET :	1750	0 / 999999	None	Yes
P429	UR CLR :	1700	0 / 999999	None	Yes
User O. C. Out					
P430	USER OUT :	P500	P500 , ... , P514	None	Yes
P431	UO SET :	1750	0 / 999999	None	Yes
P432	UO CLR :	1700	0 / 999999	None	Yes
P433	DEFLT MENU :	NO	NO , YES	None	No
P434	LVOUT PTR :	17000	0 / 8388607	None	Yes
P435	UVOUT PTR :	17000	0 / 8388607	None	Yes

ตารางที่ 4.5 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมวดที่ 5 Status Values Menu Summary (P5xx)

เป็นคำสั่งใช้ในการดูค่า พารามิเตอร์ ต่าง ๆ ของ มอเตอร์ (Motor) และ ของ อินเวอตเตอร์ โดยจะแสดงผลค่าต่าง ๆ ของ มอเตอร์ เช่น ความเร็ว ฯลฯ และแสดงข้อมูลสถานะของ อินเวอตเตอร์ แต่ที่สำคัญหมวดนี้ไม่สามารถเขียน (Edit) ได้

Parameter Number	Parameter Name	Units
P500	SPEED : XXXXX RPM	[ENG]
P501	SET SP : XXXXX RPM	[ENG]
P502	SPEED ERR : XXX %	%
P503	MOTORTORQ : XXX %	%
P504	MOTOR I : XXX.XX	A
P505	MOTOR Hz : XXX	Hz
P506	MOTOR V : XXX	Vrms
P507	RMOTORTOR : XXX %	%
P508	IN WATTS : XXXX.XX	W
P509	INPUT PF : X.XX	None
P510	BUS VOLTS : XXXX	VDC
P511	IXT ACCUM : XXX %	%
P512	DB ACCUM : XXX %	%
P513	TOTAL KWH : XXXXXX	kW-h
P514	HOURS RUN : XXXXXX	h
P515	INS : XXXXXXXXXXXX	None
P516	LAST FAULT : XXX	None
P517	FAULT 2 : XXX	None
P518	FAULT 3 : XXX	None
P519	FLT VOLTS : XXXX	VDC
P520	FLT FREQ : XXX	Hz
P521	F MODE : XXXXXXXXX	None
P522	FLT CUR : XXX.XX	A
P523	FLT TIME : XXXXXX	h
P524	SW : XXXXXX.XXX	None
P525	DRIVE SIZE : XXX	hp

ตารางที่ 4.6 ตารางแสดงคำสั่งทั้งหมดของหมวดที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมวดที่ 6 Tune Parameter Menu Summary (P6xx)

เป็นคำสั่งปรับแต่งค่า พารามิเตอร์ ต่างๆ ใน อินเวอตเตอร์ เพื่อให้เหมาะสมกับ มอเตอร์ ที่นำมาต่อ ประกอบด้วยคำสั่งต่างๆ ตามตาราง

Parameter Number	Parameter Name	Defaul Value	Value Range Min / Max	Units	Run Edit
P600	AC TUNE	D	D (Disable) E (Enable)	None	No
P601	SYS WN:	10	1 / 200	None	Yes
P602	SYS GAIN :	0.000	0.000 / 999.000	None	Yes
P603	CAR FREQ :	6 kHz	6 / 8	None	No
P604	MOTOR PPR :	1024	0 / 9999	Pulses / rev	No
P605	STATOR R :	0.000	0.000 / 32.000	Ohm	Yes
P606	I FIELD :	0.27 Rated overload	0.00 / P300	A	Yes
P607	SLIP FREQ :	0.000	0.000 / 32.767	Hz	Yes
P608	LEAKAGE :	0.0000	0.0000 / 9.000	H	Yes
P609	KA CMD :	0.000	0.000 / 99999.000	None	Yes
P610	KV CMD :	0.000	0.000 / 99999.000	None	Yes
P611	KP FBK :	0.000	0.000 / 99999.000	None	Yes
P612	KD FBK :	0.000	0.000 / 99999.000	None	Yes
P613	INVERT :	PGN MTN 0	PGN MTN 0 PGN MTN 1 PGN MTN 2 PGN MTN 3	None	No
P614	SET T :	0.500	0 / 10.00	S	Yes
P615	H MASK :	383	0 / 511	None	Yes
P616	S MASK :	64751	0 / 65535	None	Yes
P617	LOWPASS FREQ :	50	0 / 9999	Hz	Yes
P618	NOTCH FREQ :	0	0 / 9999	Hz	Yes
P619	OPEN LOOP :	D	D (Disable) E (Enable)	None	No
P620	DZ STATE :	D	D (Disable) E (Enable)	None	Yes

ตารางที่ 4.7 ตารางแสดงคำสั่งทั้งหมดของหมวด 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 อธิบายคำสั่งในหมวดที่ 0

P000 SETPT

เป็นการตั้งค่า ความเร็ว ของตัวเครื่อง สามารถปรับได้จาก " User Vin " ซึ่งเป็นแรงดัน input (จาก terminal box TB2-3) ดูตามคำสั่ง P413 " USER VIN " ค่าสูงสุดไม่เกินค่า MAX ของคำสั่ง P002 ค่าต่ำสุดไม่น้อยกว่าค่า MIN ของคำสั่ง P001 ค่าของ " SETPT " จะเปลี่ยนถ้าค่า MIN และ MAX เปลี่ยนแปลง

P001 MIN

P002 MAX

ค่า MIN และ MAX เป็นการตั้งค่าต่ำสุดและสูงสุดที่ต้องการ ขณะที่ Jogging ใช้ BCD input และใช้ SETPT (P000) โดยสามารถตั้งค่า MAX และ MIN ได้จาก " Remote cmd " และ " Local cmd " ในขณะที่เป็น โหมด มาสเตอร์ และสามารถตั้งค่าได้จาก analog speed scaling ได้ (ขณะที่ " DIR CONTROL " (P401) ถูกตั้งเป็น AUTO)

MIN จะมีผลต่อ analog torque scaling เมื่อ " DIR CONTROL " ถูกตั้งเป็น AUTO โปรแกรมจะบังคับค่า MAX $\geq$ 0 และ $\geq$ MIN ถ้าจำเป็น ถ้าค่า MAX เปลี่ยน , MIN ควรจะเปลี่ยนอัตโนมัติโดยโปรแกรมด้วย

ค่าต่อไปนี้ควรจะต้อง $\leq$ MAX และ $\geq$ MIN ถ้าจำเป็นเมื่อ MIN หรือ MAX เปลี่ยน โปรแกรม จะเปลี่ยนค่าต่อไปนี้ด้วย

" SETPT " (P000)

" JOG " (P005)

" PS1 " (P100)

" PS2 " (P103)

" PS3 " (P106)

" PS4 " (P109)

" PS5 " (P112)

" PS6 " (P115)

" PS7 " (P118)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX จะตั้งค่า ความเร็ว ตาม limit ในขณะอยู่ โหมดแรงบิด ค่า MAX สามารถปรับได้โดยใช้ " User Vin " แรงดัน input (TB2-3) ตามคำสั่ง " User Vin " (P413)

การควบคุมจากภายนอก จะปรับได้เพียงค่า MAX ที่มีผลต่อการปรับ ความเร็ว ที่ limit ใน โหมดแรงบิด และมันจะไม่มีผลต่อ function อื่นๆ ที่ควบคุมโดย MAX

ACCEL P003

เป็นระยะเวลาเร่งจาก 0 ไปจนถึงค่า " BASE FREQ " (P302) ขณะทำงานใน โหมด มาสเตอร์

DECEL P004

เป็นระยะเวลาหน่วงจาก " BASE FREQ " (P302) ไปจนถึง 0 ขณะทำงานใน โหมด มาสเตอร์

JOG P005

เป็นค่า ความเร็ว ของ เครื่อง ขณะทำงานที่ jog mode ค่าดังกล่าวสามารถปรับจากภายนอกได้ทาง " User Vin " (TB2-3) ตามคำสั่ง " User Vin " (P413)

ค่าสูงสุดของ " JOG " จะต้องเท่ากับ " MAX " (P002) และค่าต่ำสุดจะต้องเป็นค่า " MIN " (P001) " JOG " สามารถเปลี่ยนโดยอัตโนมัติด้วย program ถ้า MIN หรือ MAX เปลี่ยน

JOG ACCEL P006

เวลาเร่งจาก 0 เป็น " BASE FREQ " ขณะอยู่ในคำสั่ง " JOG "

JOG DECEL P007

เวลาหน่วงจาก " BASE FREQ " ขณะอยู่ในคำสั่ง " JOG "

I LIMIT P008

ขอบเขตของกระแสของ เครื่อง อยู่ในรูปของ RMS (Root Means Square) Amperes ค่านี้จะถูกจำกัดตามค่า ดีฟอล์ต (default) ของพิกัดกระแสของ เครื่อง ค่าต่ำสุดของ " I LIMIT " คือ " I FIELD " (P606) ถ้าจำเป็น " I LIMIT " จะต้องเปลี่ยนโดยอัตโนมัติ โดย program เมื่อ " I FIELD " เปลี่ยนไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า " I.LIMIT " สามารถตั้งจากภายนอกได้โดยใช้ " Uscr Vin " แรงดัน input (TB2-3) ตามคำสั่ง (P413)

ENG CHAR1 P009

ENG CHAR2 P010

ENG CHAR3 P011

ค่าของตัวอักษรทั้ง 3 ตัว เป็นหน่วยของค่าที่ป้อนหรือค่าที่อ่านในรูปของ engineering unit ([ENG] ตามตารางค่า พารามิเตอร์ ข้างต้น) ค่า ดีฟอล์ต engineering unit คือ " RPM " ใช้สำหรับบอกความเร็ว

ในการเปลี่ยนแปลงการใช้งาน ค่าของตัวอักษรทั้ง 3 สามารถตั้งเองได้ สมมุติว่าต้องการให้แสดงค่าการไหลเป็น gallons ต่อนาที จะเปลี่ยนตัวอักษร 3 เป็น ' G ' , ' P ' และ ' M ' ตามลำดับ ค่าทุกอย่างที่เป็น " RPM " จะกลายเป็น " GPM "

ค่าต่อไปนี้อยู่ในหน่วย [ENG]

" SETPT " (P000)

"MIN" (P001)

"MAX" (P002)

"JOG" (P005)

"ENG SCALE" (P013)

"PS1" (P100)

"PS2" (P103)

"PS3" (P106)

"PS4" (P109)

"PS5" (P112)

"PS6" (P115)

"PS7" (P118)

"L HI" (P402)

"L LO" (P403)

"R HI" (P408)

"R LO" (P409)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"SPEED" (P500)

"SETSP" (P501)

ENG DEC POINT P012

เลือกตำแหน่งของจุดทศนิยม สำหรับค่าต่างๆ ในรูปของ engineering unit ค่าดังกล่าวจะขึ้นกับค่า " ENG SCALE " (P013) อ้วยเป็นการบอกความละเอียดแม่นยำของการวัดค่าต่างๆ ที่เกี่ยวข้อง เช่น ถ้า " ENG SCALE " คือ 9000 และ "ENG DEC POINT" เป็น 3 ดังนั้นค่าดังกล่าวจะเป็น 9.000 engineering unit

ENG SCALE P013

กำหนดค่าของ พารามิเตอร์ ทุกตัวให้เป็น engineering unit [ENG] ใส่ค่าของความเร็ว ของมอเตอร์ เป็น engineering unit เมื่อทำงานที่ " BASE FREQ " (P302) เช่น

engineering unit [ENG] = RPM

" BASE FREQ " (P302) = 60

" NUM POLES " (P310) = 4

ใช้สูตรหา ความเร็ว RPM ที่ " BASE FREQ " จะได้

base rpm = (" BASE FREQ " x 120) / (" NUM POLES ")

= (60 x 120) / 4

= 1800 RPM

จะใส่ค่า base rmp ใน " ENG SCALE "

หมายเหตุ ถ้าเราไม่ใช้ " BASE FREQ " ของ 60 หรือ 4 pole มอเตอร์ " ENG SCALE " ที่ได้จากการคำนวณจะไม่ถูกต้อง

SECURITY P014

กำหนดให้ Enable หรือ Disable ได้ เป็นการป้องกัน การเปลี่ยนแปลงของค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEFLT MENU P015

กำหนดให้เรียกค่า ดีฟอล์ต ของ พารามิเตอร์ ทั้งหมดในหมวด P0xx ให้เป็นค่าที่ผู้ผลิตตั้งไว้ต้องการเรียก ดีฟอล์ต เลือก " YES " หลังจากนั้นค่า พารามิเตอร์ จะกลับไปเป็น ดีฟอล์ต และ ค่าใน " DEFLT MENU " (P015) จะเปลี่ยนเป็น " NO " ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 อธิบายคำสั่งในหมวดที่ 1

PS1 P100

PS2 P103

PS3 P106

PS4 P109

PS5 P112

PS6 P115

PS7 P118

ค่าทั้งหมดนี้คือ การตั้งค่า ความเร็ว ล่วงหน้า ค่า ความเร็ว เหล่านี้ เครื่อง จะ run ตาม ความเร็ว ที่กำหนดไว้ ขณะอยู่ที่ โหมดความเร็ว การตั้งค่า PS1 ถึง PS7 จะใช้ binary input เป็นตัวตั้งค่าตั้งแต่ 1 ถึง 7 ตามตาราง

Terminal :		TB1-18	TB1-17	TB1-16
Input :		"BCD Speed 1 "	"BCD Speed 2 "	"BCD Speed 4 "
Speed :	PS1	Set	Clear	Clear
	PS2	Clear	Set	Clear
	PS3	Set	Set	Clear
	PS4	Clear	Clear	Set
	PS5	Set	Clear	Set
	PS6	Clear	Set	Set
	PS7	Set	Set	Set

ตารางที่ 4.8 ตารางแสดงการตั้งค่า Preset Speed จาก 1-7

หรือ อาจจะใช้การปรับจากภายนอกโดยใช้ " User Vin " แรงดัน input (TB2-3) ตามคำสั่ง P413

ค่าสูงสุดของ preset speed จะมีค่าเท่ากับ "MAX" ในคำสั่ง P002

ค่าต่ำสุดของ preset speed จะมีค่าเท่ากับ "MIN" ในคำสั่ง P001

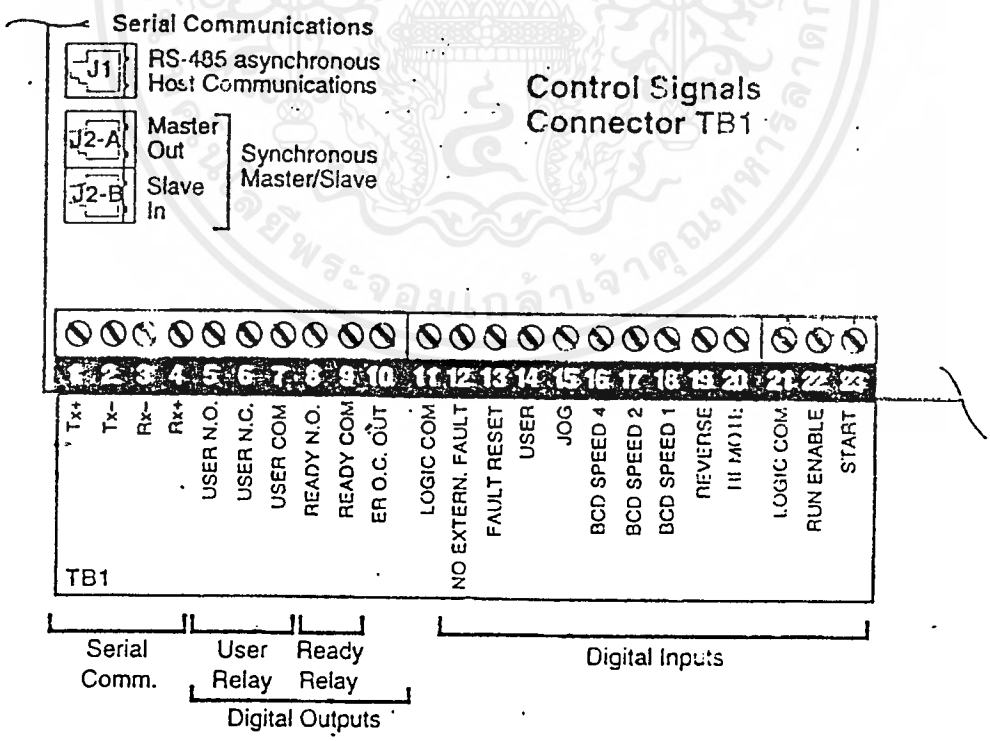
ค่า preset speed จะเปลี่ยนแปลงโดยอัตโนมัติโดย program ถ้าค่าของ "MIN" (P001) และ "MAX" (P002) เปลี่ยนไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ การต่อ BCD Speed จะต่อตามตารางข้างล่างนี้ โดยที่ LCOM จะเท่ากับการ Short สายเข้ากับ LOGIC COM ตามรูป

Terminal :	TB1-18	TB1-17	TB1-16
Input :	"BCD Speed 1"	"BCD Speed 2"	"BCD Speed 4"
Speed : PS1	LCOM	open	open
PS2	Open	LCOM	open
PS3	LCOM	LCOM	open
PS4	Open	open	LCOM
PS5	LCOM	open	LCOM
PS6	Open	LCOM	LCOM
PS7	LCOM	LCOM	LCOM

ตารางที่ 4.9 ตารางแสดงการต่อ TB1-16 , TB1-17 , TB1-18



รูปที่ 4.1 รูปแสดงพอร์ตการต่อ Preset Speed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PS1 ACCEL P101

PS2 ACCEL P104

PS3 ACCEL P107

PS4 ACCEL P110

PS5 ACCEL P113

PS6 ACCEL P116

PS7 ACCEL P119

เป็นระยะเวลาในการเร่งจาก ความเร็ว ที่ 0 ไปถึง "BASE FREQ" ตามการตั้ง PRESET SPEED ข้างต้น

PS1 DECEL P102

PS2 DECEL P105

PS3 DECEL P108

PS4 DECEL P111

PS5 DECEL P114

PS6 DECEL P117

PS7 DECEL P120

เป็นระยะเวลาในการหน่วงจาก ความเร็ว ที่ "BASE FREQ" ไปถึง 0 ตาม การตั้ง PRESET SPEED ข้างต้น

SKIP FREQ 1 P121

SKIP 1 BAND P122

SKIP FREQ 2 P123

SKIP 2 BAND P124

SKIP FREQ 3 P125

SKIP 3 BAND P126

ในการ run มอเตอร์ จะมีช่วง ความเร็ว ที่อาจจะเกิดการสั่นของ มอเตอร์ หรือเกิด resonance ขึ้นได้ซึ่งอาจจำเป็นต้องหลีกเลี่ยง ค่า ความเร็ว ดังกล่าว ตัว เครื่อง จะสามารถข้ามความถี่ดังกล่าวได้ 3 ช่วง

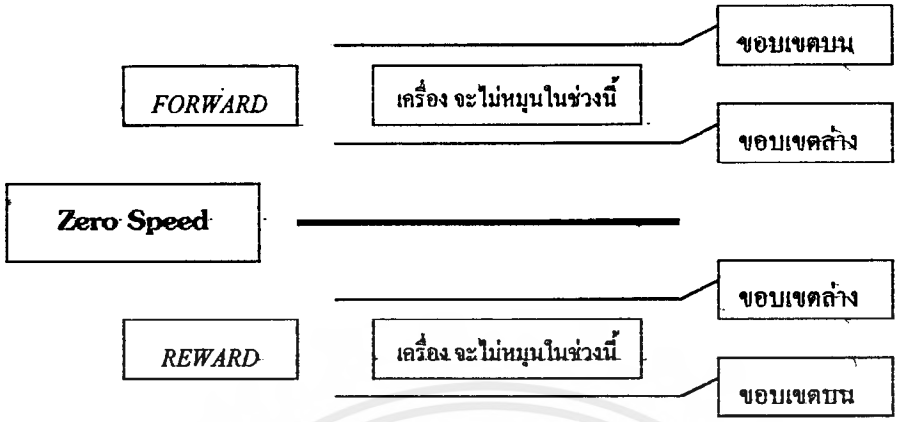
"SKIP FREQ 1" และ "SKIP 1 BAND" เป็นการตั้งค่าความถี่แรกของการข้าม

"SKIP FREQ 2" และ "SKIP 2 BAND" เป็นการตั้งค่าความถี่ที่สองของการข้าม

"SKIP FREQ 3" และ "SKIP 3 BAND" เป็นการตั้งค่าความถี่ที่สามของการข้าม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดูระยะในการตั้งค่าจากรูป



รูปที่ 4.2 รูปแสดงช่วงการหมุนของมอเตอร์

แต่ละความถี่ข้ามและช่วงของความถี่จะเป็นไปตามรูปข้างบน โดยแบ่งออกเป็น FORWARD และ REWARD จากรูปจะกำหนดช่วงของการข้ามได้จากสูตร

ถ้าเป็น FORWARD

ขอบเขตบนของการข้าม = (Skip Frequency + band) / 2

ขอบเขตล่างของการข้าม = (Skip Frequency – band) / 2

ถ้าเป็น REWARD

ขอบเขตล่างของการข้าม = - (Skip Erequency – band) / 2

ขอบเขตบนของการข้าม = - (Skip Frequency + band) / 2

เครื่อง จะไม่หมุนภายในช่วงของ Skip speed แต่ เครื่อง จะเร่งการหมุนไปที่จุดขอบเขตบนของช่วง และจะหน่วงไปที่จุดขอบเขตล่างของช่วง

ถ้าไม่ต้องการใช้ Skip mode ให้ตั้งค่า band เป็น 0

Skip Frequency ไม่ใช่ function ในโหมดสลาฟ (ใช้ใน โหมดสลาฟ ไม่ได้)

DEFLT MENU P127

กำหนดให้เรียกค่า ดีฟอล์ต ของ พารามิเตอร์ ทั้งหมดในหมวด P1xx ให้เป็นค่าที่ผู้ผลิตตั้งไว้ ต้องการเรียก ดีฟอล์ต เลือก " YES " หลังจากนั้นค่า พารามิเตอร์ จะกลับไปเป็น ดีฟอล์ต และ ค่าใน " DEFLT MENU " (P127) จะเปลี่ยนเป็น " NO " ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 อธิบายคำสั่งในหมวดที่ 2

START MODE P200

เป็นการเลือกโหมดการ start มี 2 แบบ คือ

MAN :

ถ้าเลือกโหมดนี้ ตัวเครื่อง จะไม่ทำงานทันทีที่เปิดระบบ (power – up)

LINE :

ถ้าเลือกโหมดนี้ ตัว เครื่อง จะทำงานเพื่อรอคำสั่งหมุนทันที ดังนั้น input ที่เข้า เครื่อง จะต้องไม่มีสัญญาณ เครื่อง จึงยังไม่ทำงาน

ถ้า " JOG " input (Terminal box , TB1-15) ปิด (short) ตัว เครื่อง จะ run ที่ ความเร็ว ที่กำหนดไว้

ถ้าตั้งค่า line และติด " WATCHDOG TIMER " ขณะเกิด ฟอล์ต (fault) ตอนเปิด เครื่อง ตัว เครื่อง จะพยายามที่จะ start อีกครั้ง ถ้าตั้งค่า " TRIP RESTART " เป็น 0 ตัว เครื่อง จะมีค่าเวลาหน่วง " RESTART TIME " ก่อนที่จะพยายาม start และถ้าตั้งค่า " RESTART TIME " > 0 ตัว เครื่อง จะพยายาม start ใหม่เป็นจำนวนที่ตั้งไว้

ถ้าตั้งค่า " START MODE " (P200) = LINE และเกิด " MEMORY CRC " , " MEMMORY OUT " หรือ " HW FAULT XXX " ตอนเปิด เครื่อง ตัว เครื่อง จะไม่พยายาม restart

ตัว เครื่อง จะแสดงค่าสถานะเป็น " line starting " ขณะเปิดเครื่อง

RUN MODE P201

เป็นการเลือกโหมดการทำงานพื้นฐานของ เครื่อง แบ่งเป็น

M SPEED :

ถ้าเลือก M SPEED , ตัว เครื่อง จะทำงานใน โหมด มาสเตอร์ ความเร็ว จะตั้งค่าได้จาก kcypad , digital I/O , analog input หรือจาก asynchronous serial port

S SPEED :

ถ้าเลือก S SPEED ตัว เครื่อง จะทำงานใน โหมดสลาฟ ความเร็ว จะตั้งค่าได้จาก keypad , digital I/O , analog input , จาก asynchronous serial port หรือ จาก synchronous serial port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TORQ :

ถ้าเลือก TORQ ตัว เครื่อง จะทำงานใน โหมดแรงบิด สามารถตั้งค่าได้จาก " INT TORQ " (P342) หรือจาก analog input ตามต้องการ ไม่ว่าจะเป็น TORQ mode หรือ โหมดความเร็ว ค่าที่ตั้งสามารถตั้งจาก " User " digital input (TB1-14) ตามคำสั่ง " UIN " (P426) โดยเลือก SPEED TORQ หรือ SLAVE CTL

STOP MODE P202

เลือก mode ในการหยุดตัว เครื่อง แบ่งเป็น

DECEL :

ถ้าเลือก DECEL ตัว เครื่อง จะหน่วงเวลาก่อนที่จะหยุดตามค่าหน่วงเวลาที่เหมาะสม (ขึ้นกับลักษณะการทำงานว่าเป็น setpoint (SET PT) , jog speed (JOG) , preset speed (PS1-PS7)หลังจากเวลาหน่วง " SET T " (P614) ตัว เครื่อง จะหยุดและปล่อยให้ มอเตอร์ หยุดเองได้แม้แต่ใน mode
" DECEL "
ถ้าเลือก โหมดแรงบิด หรือ ถ้าตั้ง " User " input (TB1-14) แล้ว " COAST CTL " ถูกเลือกตามคำสั่ง "UIN" (P426) นอกจากนี้แล้วถ้าเลือก " RUN ENABLE " (P220) เป็น COAST และ " Run Enable " input (TB1-22) ถูกตั้ง จะทำให้ตัว เครื่อง ปล่อยให้ มอเตอร์ หยุดเอง

COAST :

ถ้าเลือก COAST ตัว เครื่อง จะปล่อยให้ มอเตอร์ หยุดเอง

DC INJ :

ถ้าเลือก DC INJ ตัว เครื่อง จะใช้หลัก DC injection เพื่อที่จะหยุด มอเตอร์ กระแส DC จะถูกส่งไปที่ ac มอเตอร์ เพื่อให้เกิดแรงบิดต้านการหมุน ตัว เครื่อง จะหยุดหลังจากเวลา " INJ BRAKE " (P216)
ตัว เครื่อง จะปล่อย มอเตอร์ ให้หยุดเองแม้จะอยู่ใน mode "DC INJ" ถ้าตั้ง "User" input (TB1-14) แล้วเลือก COAST CTL ตามคำสั่ง "UIN" (P426) หรือเลือก COAST ตามคำสั่ง "RUN ENABLE" (P220) และตั้ง "Run Enable" input (TB1-22) เป็น open

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC HOLD :

ถ้าเลือก DC HOLD ตัว เครื่อง จะหยุดโดยใช้เวลาหน่วงที่เหมาะสม (ตามลักษณะคำสั่งว่าเป็น setpoint (SET PT) , jog speed (JOG) หรือ preset speed (PS1-PS7)

กระแส DC จะถูกส่งไป AC มอเตอร์ สร้างแรงบิดต้านการหมุน และตัว เครื่อง จะหยุดหลังจากเวลา "INJ BRAKE" (P216) ตัว เครื่อง จะปล่อยให้ มอเตอร์ หยุดเอง แม้ว่าจะอยู่ใน mode "DC HOLD" ก็ตาม

ถ้าเลือก โหมดแรงบิด ก่อนที่ DC brake จะทำงานและหลังจาก DC BREAK ทำงาน DC HOLD จะยังอยู่แม้ว่าจะเลือก โหมดแรงบิด ก็ตาม ตัว เครื่อง จะปล่อย มอเตอร์ ให้หยุดเอง แม้ว่าจะอยู่ใน mode "DC HOLD" ถ้าตั้ง "User" input (TB-14) แล้วเลือก COAST CTL ตามคำสั่ง "UIN" (P426) หรือถ้าเลือก COAST ตามคำสั่ง "RUN ENABLE" (P220) แล้วตั้ง " Run Enable " input (TB1-22) เป็น open

SETPT SRC P203

เป็นการเลือคแหล่งสัญญาณอ้างอิง (local setpoint) เฉพาะใน โหมด มาสเตอร์ แบ่งเป็น

KEYPAD :

ถ้าเลือก keypad ตัว เครื่อง จะรับสัญญาณอ้างอิงจาก พารามิเตอร์ "SETPT" (P000)

ANALOG :

ถ้าเลือก analog ตัว เครื่อง จะรับสัญญาณอ้างอิงจาก "Local Command" แรงดัน input (TB2-2)

KEYPAD ENABLE P204

ตั้งค่า Enable หรือ disable ของปุ่มกดควบคุม เครื่อง

ENA(ble) :

ถ้าเลือก enable ปุ่มกด "JOG" , "Run Fwd" , "Run Rev" และปุ่ม "STOP" จะใช้ควบคุม เครื่อง ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIS(able) :

ถ้าเลือก disable ปุ่ม "JOG" , "Run Fwd" และ "Run Rev" จะไม่สามารถควบคุม เครื่อง ได้ และจะใช้สัญญาณ digital input จากภายนอก หรือ serial commands แทนแต่ว่าปุ่ม "Stop" ยังคงใช้หยุด มอเตอร์ ได้อยู่

DAL(disable all) :

เป็นการยกเลิกการใช้ปุ่มทั้งหมด (แม้แต่ปุ่ม Stop) และใช้การควบคุมจาก digital input (Terminal Box , TB1 ทั้งหมด) ภายนอกหรือ serial commands แทน

SYN SER P205

เป็นการตั้งค่าให้กับ synchronous serial เป็น OFF , SEND หรือ RECEIVE เฉพาะ มาสเตอร์ เครื่องเท่านั้นที่สามารถตั้งเป็น SEND ได้ใน synchronous serial network

SLAVE SOURCE P206

แหล่งของสัญญาณ ตั้งค่า ของตัว สลาฟ แบ่งเป็น

ANA(log) :

ถ้าตั้งเป็น analog และตัว เครื่อง เป็น สลาฟ สัญญาณ analog input ถูกเลือกโดย "Rcmotc" digital input (TB1-20) เพื่อเป็นสัญญาณ ตั้งค่า ที่สลาฟ

SCI :

ถ้าเลือก SCI และตัว เครื่อง เป็น สลาฟ จะใช้ synchronous serial input เป็นตัวรับสัญญาณ ตั้งค่า

ON DEL P207

AD DEL P208

"ON DEL"

เป็นค่าเวลาหน่วยต่ำสุดก่อนที่ตัว เครื่อง จะทำงานและทำการปรับความเร็ว ตามคำสั่งที่ป้อน ตัวเวลาหน่วงที่มีความสำคัญในการสร้าง flux ใน มอเตอร์ ให้เพียงพอก่อนที่ตัว มอเตอร์ จะทำการเปลี่ยน ความเร็ว ค่าดังกล่าวจะเป็นมาตรฐาน และปรับไม่ได้

"ON DEL" จะแปรผันกับคำสั่ง "MTR AMPS" P(300) , "SLIP FREQ" (P607)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"AD DEL"

เป็นค่าเวลาหน่วงที่ป้อนเข้าไปเพื่อไปรวมกับ ON DEL เป็นเวลาหน่วงก่อนที่ตัว เครื่อง จะทำการเปลี่ยน ความเร็ว

"AD DEL" จะมีประโยชน์ในการเปลี่ยน ความเร็ว ขณะที่เป็น synchronous serial master และต่อ สลาฟ และตัว เครื่อง ทุกตัว เปิด พร้อมๆ กัน (ตัว มาสเตอร์ ไม่ควรหมุน จนกว่า ตัว สลาฟ ทุกตัวจะพร้อม) เพื่อความปลอดภัยควรจะ

(1) ตั้ง "AD DEL" ในตัว สลาฟ ทุกตัวเป็น 0

(2) ตรวจสอบ "ON DEL" ใน สลาฟ ทุกตัวว่ามีค่าเท่าไร

(3) ตั้ง "AD DEL" ในตัว มาสเตอร์ เครื่อง เพื่อที่จะให้ "ON DEL" (ในตัว มาสเตอร์) + "AD DEL" (ในตัว มาสเตอร์) = เวลามากที่สุดของ "ON DEL" (ในตัว สลาฟ) และไม่ควรตั้ง "AD DEL" ใน มาสเตอร์ < 0

RATIO NUM P209

RATION DEN P216

"RATIO NUM" คือตัวตั้งของค่า overall speed ratio และ "RATIO DEN" เป็นค่าตัวหารของค่า overall speed ratio ซึ่งค่า overall speed ratio เป็นค่าอัตราส่วนของตัว สลาฟ ซึ่งจะกำหนดว่าตัว สลาฟ จะหมุนในอัตราส่วนเป็นเท่าไรของตัว มาสเตอร์ ถ้าไม่ต้องการใช้อัตราส่วนให้ตัวตั้ง "RATIO NUM" และตัวหาร "RATIO DEN" เป็น 1 ตามสูตรดังนี้

Overall Speed Ratio = "RATIO NUM" / "RATIO DEN"

ถ้าต้องการให้ตัว สลาฟ หมุนที่ ความเร็ว 1 ใน 3 ของตัว มาสเตอร์ จะต้องกำหนดค่า "RATIO NUM" เป็น 1 และ "RATIO DEN" เป็น 3

ค่าของ "RATIO NUM" สามารถปรับได้จากภายนอกโดยใช้ "User Vin" แรงดัน input (TB2-3) ตามคำสั่ง "USER VIN" (P413)

<u>หมายเหตุ</u>

- program ใช้ค่า overall speed ratio เพื่อที่จะเพิ่มความเที่ยงตรง
- ความเร็วในการหมุนอาจจะไม่เที่ยงตรงถ้าใช้สัญญาณ Analog input เนื่องจากธรรมชาติของสัญญาณ analog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความเร็วในการหมุนอาจไม่เที่ยงตรงถ้าใช้ synchronous scrial เพราะว่า ตัว มาสเตอร์ และ สลาฟ จะหมุนกันแบบ asynchronou

SPINNING MOTOR (P211)

แบ่งได้เป็น

D(isabled)

ถ้าเป็น disable ตัว เครื่อง จะมองเห็น มอเตอร์ พร้อมหมุนขณะที่เปิดทันที ซึ่งอาจทำให้ มอเตอร์ หมุนขณะเปิด และตัวเครื่อง อาจจะทำให้กระแสเดินพิดัดได้

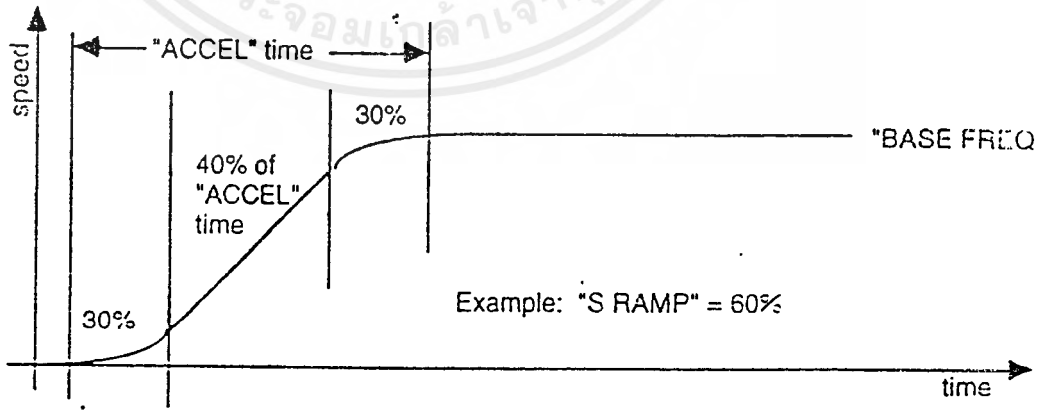
E(nabled)

ถ้าเลือก enable ตัว program จะเริ่มสั่งให้ มอเตอร์ หมุนที่ ความเร็ว ที่กำหนด มอเตอร์ จะค่อยๆ เร่งอย่างนุ่มนวลไปยัง ความเร็ว ที่ต้องการ

S RAMP P212

SRAMP M P213

"S RAMP" เป็นการตั้งค่าความเรียบในการเร่งหรือหน่วง (เรียกอีกอย่างว่า jerk limiting หรือขอบเขตของการกระตุก) ค่าดังกล่าวจะแสดงเป็น percent ของเวลาเร่งหรือหน่วง มีค่าระหว่าง 0 (linear) และ 100 (S-curve) ค่ายิ่งมาก ยิ่งทำให้การเร่งหรือหน่วงเรียบขึ้น แต่ถ้ามากเกินไปจะเป็นการเพิ่มค่าอัตราเร่ง/หน่วง ให้กับช่วงกลางของ curve ดูรูปประกอบ



รูปที่ 4.3 รูปแสดงกราฟความสัมพันธ์ระหว่าง ความเร็ว และเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NO REV :

ถ้าตั้งค่านี้ตัว เครื่อง จะไม่หมุน reverse และจะปฏิเสธคำสั่ง reverse

ALWAYS :

ถ้าตั้งค่านี้ตัว เครื่อง จะสนับสนุนการเปลี่ยนทิศทางการหมุนทุกเวลา และจะตอบสนองกลับทันที โดยจะไม่มีการหยุดก่อนเปลี่ยนทิศทาง

STOPPED :

ถ้าตั้งค่านี้ ตัว เครื่อง จะหยุด และเปลี่ยนทิศทางการหมุน ถ้าตัว เครื่อง ไม่หยุดมันจะยังคงไม่เปลี่ยนทิศทางการหมุน (ยังคงเก็บคำสั่งไว้ และทำการหมุนเปลี่ยนทิศทางหลังจากหยุด)

ถ้าตั้งค่านี้ ตัว เครื่อง จะหมุน reverse ในกรณีที่อยู่ในโหมด JOG ถ้าไม่ตัว เครื่อง จะไม่เปลี่ยนทิศการหมุน (แต่ยังคงเก็บคำสั่งไว้ และทำการหมุนเปลี่ยนทิศทาง หลังจากเข้า mode jog)

INJ BRAKE TIME P215

เป็นช่วงเวลาขณะที่ป้อนกระแส DC เข้าไปใน มอเตอร์ เมื่อตั้งค่า "DC INJ" ตามคำสั่ง "STOP MODE" (P202) กระแส dc จะหยุดหลังจากเวลาหน่วงนี้ ในขณะที่เกิด

ฟอล์ต หรือ ขณะที่ตัว เครื่อง restart

INJ BRAKE I (P216)

เป็นระดับของกระแส dc ที่ป้อนเข้า มอเตอร์ เมื่อตั้งค่า "DC INT" หรือ "DC HOLD" ตามคำสั่ง "STOP MODE" (P202) ค่ามากที่สุดที่ใส่เข้าไปได้คือ 100% มีค่าเท่ากับคำสั่ง "I LIMIT" (P008)

TRIP RESTART (P217)

เป็นจำนวนครั้งที่ตัว เครื่อง จะพยายาม restart และจะรอเวลา "RESTART TIME" (P218) ก่อนที่จะพยายาม restart แต่ละครั้ง ถ้าตั้งเป็น 0 การ restart จะถูกยกเลิก

ถ้า "TRIP RESTART" >0 ตัว เครื่อง ขณะเกิด ฟอล์ต จะ reset ฟอล์ต และพยายาม restart ใหม่ และถ้าตัว เครื่อง เกิด trip มันจะทำคำสั่ง "TRIP RESTARTS" และจะพยายาม restart เป็นจำนวนครั้งตามที่ตั้งไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า "TRIP RESTART" = 0 เกิด ฟอล์ต ขึ้นจะต้อง rcsct ฟอล์ต ก่อนที่ เครื่อง จะเริ่ม start

เมื่อตัวนับในการ restart > "TRIP RESTART" ตัว เครื่อง จะหยุดการ restart ค่าของตัวนับ restart จะถูกตั้งเป็น 0 ใหม่ เมื่อ ฟอล์ต ถูก reset , เมื่อตัว เครื่อง ปิด ขณะ ไม่เกิด ฟอล์ต หรือเมื่อหมด 10 นาทีหลังจากเกิด ฟอล์ต ครั้งที่แล้ว

ถ้า "START MODE" (P200) ถูกตั้งเป็น "LINE" และตัว "WATCHDOG TIMER" เกิด ฟอล์ต ขึ้นขณะเปิดเครื่อง ตัว เครื่อง จะพยายาม start ใหม่ 1 ครั้ง (ถ้าตั้ง "TRIP RESTART" = 0) และมีเวลาหน่วงในการเริ่ม start เท่ากับ "RESTART TIME" และตัว เครื่อง จะพยายาม rcstart เป็นจำนวนครั้งที่ตั้งถ้า "TRIP RESTART" > 0

ถ้า "START MODE" (P200) ตั้งเป็น "LINE" และ "MEMORY CRC" "MEMORY OUT" หรือ "HW FAULT XXX" ปรากฏขึ้นขณะเปิดเครื่อง ตัว เครื่อง จะ ไม่ทำการ restart

ถ้าแรงดันจากแหล่งจ่ายต่ำกว่ากำหนด ตัว เครื่อง จะหมุนแล้ว trip และจะทำการ restart อัตโนมัติ เมื่อแรงดันกลับเข้าสู่ปกติ

Auto restart จะยกเลิกเมื่อตัว เครื่อง ถูกสั่งให้หยุด

RESTART TIME P218

ช่วงเวลา (เป็นวินาที) ที่ตัว เครื่อง จะรอก่อนทำการ restart ถ้าค่า "TRIP RESTART" >0 หรือ ถ้า "START MODE" (P200) เป็น LINE และเกิด ฟอล์ต ขึ้นที่ "WATCHDOG TIMER" ขณะเดินเครื่อง

BUS REG P219

เป็นการกำหนดระดับแรงดัน เมื่อเกิดแรงดันไฟมากกว่าระดับที่กำหนด เช่นขณะทำการหน่วงตัว เครื่อง จะลดค่า torque limit ของมันเพื่อที่จะป้องกันการเกิด under voltage trip ให้ตั้งค่านี้เป็น OFF ถ้าไม่ต้องการใช้ bus voltage regulator สามารถตั้งค่าได้ 3 ระดับตามตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	230 V Unit	460 V Unit	575 VUnit
HIGH	375 V DC	750 V DC	938 V DC
MID	362 V DC	725 V DC	905 V DC
LOW	350 V DC	700 V DC	875 V DC

ตารางที่ 4.10 แสดงสถานะของ BUS REG

ถ้ามีการต่อ ตัวต้านทานสำหรับ dynamic break ตัว "BUS REG" จะต้องตั้งเป็น OFF เพื่อที่จะให้ตัวต้านทานเป็นตัวรับพลังงานที่เกินจาก bus

<u>หมายเหตุ</u>

เมื่อ "BUS REG" เป็น ON เวลาหน่วงของตัว เครื่อง อาจจะเพิ่มขึ้นเพื่อป้องกันการเกิด bus overvoltage fault แต่อาจเกิดกรณีที่ตัว เครื่อง กำลังจะหน่วงเพื่อที่จะหยุด แต่มันอาจจะดับก่อน ขณะที่กำลังจะเปลี่ยนสถานะการหมุน อันนี้เป็นเพราะตัว เครื่อง จะดับไม่ว่าตัว "BUS REG" จะขยายเวลาหน่วงหรือไม่ก็ตาม เพื่อเป็นการป้องกันการดับเช่นนี้จะต้องเพิ่มเวลาหน่วงในคำสั่ง "SET T" (P614)

RUN ENABLE P220

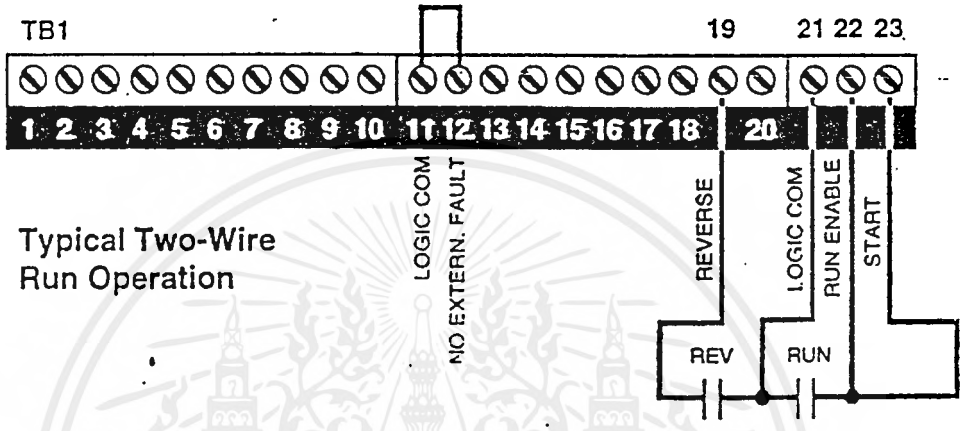
เมื่อ "RUN ENABLE" = STOP :

(1) "Start" input (TB1-22) จะปิดชั่วคราวเพื่อ start เครื่อง "Run Enable" input (TB1-22) จะเปิดชั่วคราวเพื่อ stop เครื่อง และตัว เครื่อง จะหยุดเองตามคำสั่ง "STOP MODE" (P202)

(2) "Jog" input (TB1-15) จะปิดเพื่อทำ jog และเปิดเพื่อหยุดตัว เครื่อง ถ้า "Run Enable" input เปิดตัว เครื่อง จะหยุด และตัว เครื่อง จะหยุดตามคำสั่ง "STOP MODE" (P202)

(3) ตัว เครื่อง จะสามารถทำ jog โดยใช้ "Start" input (ตามคำสั่ง "START INIT" (P221) = OFF) ใช้ปุ่มกดเพื่อเลือก JOG แต่ต้องตั้ง "Start" และ "Run enable" input ตามรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงพอร์ตในการต่อ RUN ENABLE

(4) "Run Enable" และ "Start" input สามารถต่อกันได้เพื่อที่จะควบคุมการ start / stop

(5) ตัว เครื่อง จะไม่สามารถ start จากปุ่มกดได้ถ้า "Run Enable" input ถูกเปิด เมื่อ "RUN ENABLE" =COAST

(1) เมื่อ "Run Enable" input (TB1-22) เปิดตัว เครื่อง จะปล่อยให้ มอเตอร์ ค่อยๆ หยุดเองไม่ว่าจะตั้งค่า "STOP MODE" (P202) หรือไม่ก็ตาม

(2) "Start" input (TB1-23) จะต้องรักษาค่าไว้เสมอเพื่อให้ตัว เครื่อง หมุนได้ ถ้าตัว TB1-23 เปิดตัว เครื่อง จะหยุดตามคำสั่ง "STIOP MODE" (P202) เมื่อ "Start" input ปิดตัว เอรื่อง จะเริ่ม start

(3) "Jog" input (TB1-15) จะต้องตั้งไว้เพื่อให้ เครื่อง ทำการ jogging เมื่อ TB1-15 เปิดตัว เครื่อง จะหยุดตามคำสั่ง "STOP MODE" (P202)

(4) ตัว เครื่อง จะไม่สามารถ start ได้จากภายนอกไม่ว่านะเป็นแบบ asynchronous หรือ synchronous) หรือ จากปุ่มกด ถ้า "Run Enable" input เปิดอยู่เมื่อ "Start" input เปิดตัว เครื่อง ก็ไม่สามารถทำงานได้ไม่ว่าจะเป็นสัญญาณจากภายนอก หรือ ปุ่มกด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(5) การใช้สาย 2 เส้น ("Run Enable" ต่อกับ "Start" เพื่อที่จะสั่งเครื่อง start / stop ไม่แนะนำเพราะว่า "Run Enable" จะต้องปิดก่อน "Start" input อย่างน้อย 0.01 วินาที เพื่อให้แน่ใจว่าตัว เครื่อง จะทำงาน

(6) แนะนำให้เลือก ON ในคำสั่ง "START INIT" (P221) เมื่อ "Run Enable" ถูกตั้งเป็น COAST

ถ้า "START INIT" = ON

เมื่อตัว เครื่อง ทำงาน ทุก function จะเริ่มทำงานตามสัญญาณ digital input (สัญญาณจาก terminal box , TB1)

ถ้า "START INIT" = OFF

เมื่อ "Start" input ปิด ตัว เครื่อง จะหมุนถ้า "Jog" input เปิด ตัว "Jog" input ควรจะเปิดตลอด (ไม่แนะนำให้ "Jog" input และ "start" input ปิดพร้อมกัน) ถ้า "Jog" input ปิดหลังจากที่ "Start" input ปิดอยู่ก่อนแล้ว ตัว เครื่อง จะไปทำ jog ทันที

START INIT P221

"Reverse" input (TB1-19) , "Jog" input (TB1-15) , "Remote" input (TB1-20) และ "User" input (TB1-14) input เหล่านี้ต้องทำงานสัมพันธ์กัน เช่น ถ้าเลือกไปที่ตัว "Reverse" input ขณะที่มันกำลังจะปิดตัว "Forward" input จะต้องกำลังจะเปิดออกพร้อมกัน ถ้า "Jog" input กำลังจะปิดตัว "notjog" จะต้องกำลังเปิดออก ถ้า เลือกไปที่ตัว "Remote" input ตัวของ "Remote" กำลังปิดตัว "Local" input จะต้องกำลังเปิดขึ้นสัมพันธ์กัน เป็นต้น ตัว เครื่อง จะอนุญาตให้ปุ่มกด และคำสั่งจาก port serial สามารถสั่งให้ตัว เครื่อง ทำงานที่สถานะตรงข้ามกับ input ที่กล่าวข้างต้นได้ตลอดเวลา ถ้าตั้งค่า **"START INIT" = ON**

ถ้า "START INIT" = ON (แนะนำให้เลือก) ทำให้

(1) เมื่อตัว เครื่อง หมุน โดยใช้ "Start" input (TB1-23) ตัว "forward" / "reverse" , "jog" / "not jog" , "local" / "remote" และ "User-in" / "not user in" จะเริ่มพร้อมที่จะทำงานตาม input ที่ตั้งไว้คือ "Reverse" input (TB1-19) "Jog" input (TB1-15) , "Remote" input (TB1-20) และ "User" input (TB1-14) input

(2) เมื่อตัว เครื่อง หมุน โดยใช้ "Jog" input (TB1-15) ตัว function "forward" /

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"Rcvcrsc" "local" / "rcmotc" และ "Uscr in" / "not uscr in" จะเริ่มพร้อมที่จะทำงานตาม input ที่ตั้งไว้คือ "Reverse" input (TB1-19) "Remote" input (TB1-20) และ "User" input (TB1-14) input

เมื่อ "START INIT" = OFF และ "RUN ENABLE" = COAST

การตั้งค่าแบบนี้ไม่แนะนำควรจะเลือก ON เมื่อเลือก "RUN ENABLE" เป็น COAST

เมื่อ "START INT" = OFF และ "RUN ENABLE" = COAST

การตั้งค่าแบบนี้มีประโยชน์เมื่อต้องการให้ บางfunction ควบคุมจากปุ่มกด และ scrial port และ function ที่เหลือควบคุมจากสัญญาณ digital input (TB1 ทั้งหมด) เช่น ปุ่มกดจะเป็นตัวเลือกว่า "jog" หรือ "not jog" แต่ สัญญาณ digital input จะเป็นตัวปิดและเปิด เครื่อง

LANGUAGE P222

กำหนดให้หน้าจอของตัว เครื่อง แสดงผลเป็นภาษา ENG(lish) , SPAN(ish) , FREN(ch) หรือ GERM(an)

SET USER DEFLT P223

เป็นการเอาค่าปัจจุบันที่ตั้งไว้มาเป็นค่า ดีฟอล์ต ของผู้ใช้ ให้เลือก "YES" หลังจากตั้งค่า ดีฟอล์ต แล้ว ค่าของคำสั่งนี้จะเปลี่ยนกลับเป็น "NO"

DEFLT MENU P224

กำหนดให้เรียกค่า ดีฟอล์ต ของ พารามิเตอร์ ทั้งหมดในหมวด P2xx ให้เป็นค่าที่ผู้ผลิตตั้งไว้ต้องการเรียก ดีฟอล์ต เลือก " YES " หลังจากนั้นค่า พารามิเตอร์ จะกลับ

ไปเป็น ดีฟอล์ต และ ค่าใน " DEFLT MENU " (P224) จะเปลี่ยนเป็น " NO " ทันที

DEFLT ALL FACT P225

เป็นการเรียกค่า ดีฟอล์ต ของ พารามิเตอร์ ทั้งหมดของทุกคำสั่งให้เป็นค่าที่ผู้ผลิตตั้งไว้ต้องการเรียก ดีฟอล์ต เลือก " YES " หลังจากนั้นค่า พารามิเตอร์ จะกลับไปเป็น ดีฟอล์ต และ ค่าจะเปลี่ยนเป็น " NO " ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEFLT ALL USER P226

เป็นการเรียกค่า ดีฟอล์ต ของ พารามิเตอร์ ทั้งหมดที่ผู้ใช้ตั้งขึ้นออกมาต้องการเรียก ดีฟอล์ต เลือก " YES " หลังจากนั้นค่า พารามิเตอร์ จะกลับไปเป็น ดีฟอล์ต และ ค่าจะเปลี่ยนเป็น " NO " ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 อธิบายคำสั่งในหมวดที่ 3

MTR AMPS P300

พิกัดของกระแสของตัว มอเตอร์ จาก nameplate ของ มอเตอร์ ค่าดังกล่าวใช้ใน กรณีป้องกันค่ากระแส overload

ค่าที่ตั้งควรจะเป็น 67 % ของค่า constant-torque ที่ overload ของตัว เครื่อง และเพื่อความปลอดภัยไม่ควรมากกว่า 83% ของค่า variable-torque ที่ overload ของตัว เครื่อง ค่าต่ำสุดที่ใช้ได้เป็น 20% ที่ overload ของตัว เครื่อง

ถ้า "MTR AMPS" เปลี่ยนไป จะต้องทำการ AC Tune test (คำสั่งที่ P600) ใหม่อีกเพื่อให้ค่าใน "SLIP FREQ" (P607) ถูกต้อง

ตัว program ค่าของ "I FIELD" (P606) จะต้องน้อยกว่า หรือเท่ากับ "MTR AMPS" เมื่อค่า "MTR AMPS" เปลี่ยนไปตัว program จะทำการเปลี่ยน "I FIELD" (ถ้าจำเป็น)

MOTOR VOLTS P301

แรงดันของ มอเตอร์ จาก nameplate ค่าดังกล่าวจะมีค่าให้เลือกตาม ดีฟอล์ต ของเครื่อง (คือ 230V , 460V , 575V) ค่านี้จะตั้งเมื่อ มอเตอร์ มีพิกัดแตกต่างจาก แหล่งจ่ายไฟค่า "MOTOR VOLTS" และ "BASE FREQ" (P302) ยังตั้งเพื่อปรับค่า Flux ใน มอเตอร์ ตามต้องการ

"MOTOR VOLTS" จะเป็นการตั้งค่า undervoltage trip และค่าที่ทำให้ relay เปิด (เป็นค่า ณ.จุดที่ซึ่ง DC bus Voltage น้อยกว่า "MOTOR VOLTS")

ถ้าค่า "MOTOR VOLTS" เปลี่ยนไปจะต้องหา AC Tune test ใหม่

BASE FREQ P302

ค่าความถี่ของ มอเตอร์ ที่พิกัด nameplate ค่าดังกล่าวเป็นค่าพื้นฐานสำหรับการทำงานของ มอเตอร์ ส่วนมากจะเป็น 60 หรือ 50 Hz

"MOTOR VOLTS" (P301) และ "BASE FREQ" เป็นการตั้งค่า flux ใน มอเตอร์ ให้เป็นไปตามที่ต้องการ (*E/f* แปรผันตาม flux)

ถ้าค่า "BASE FREQ" เปลี่ยนจะต้องทำ Ac Tune test ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LIMIT RPM P303

เป็นค่า ความเร็ว ค่าสูงสุดที่ให้ มอเตอร์ หมุนได้ใน 1 นาที (RPM) เป็นการป้องกัน มอเตอร์ และ load จากสภาวะความเร็วไม่ปลอดภัย

ถ้าความเร็ว มอเตอร์ เกิน 115% ของ "LIMIT RPM" จะเกิด ฟอล์ต ขึ้นได้

IXT AMPS P304

เป็นการตั้งค่าระดับของกระแสซึ่งเกี่ยวข้องกับค่า I^2t เป็นค่าความปลอดภัยในการหมุนของ มอเตอร์ ค่า I^2t เป็นค่าป้องกันตัว มอเตอร์ และ เครื่อง จากสภาวะ overcurrent นานๆ เมื่อ เครื่อง รับกระแสเกินพิกัด "IXT AMPS" ตัว เครื่อง จะทำการหาค่า I^2t และจะปรากฏ "IXT WARNING" ขึ้น (ลักษณะเป็นไฟกระพริบที่หน้าจอแสดงผลของตัว เครื่อง) ค่าต่ำสุดที่ตั้งได้คือ 10% และสูงสุดเท่ากับ 105% ค่าดังกล่าวสามารถปรับได้จาก "User Vin" input (TB2-3) ตามคำสั่ง "USER VIN" (P413)

IXT FAULT P305

ค่านี้ไม่สามารถตั้งได้

MAX F TORQ P306

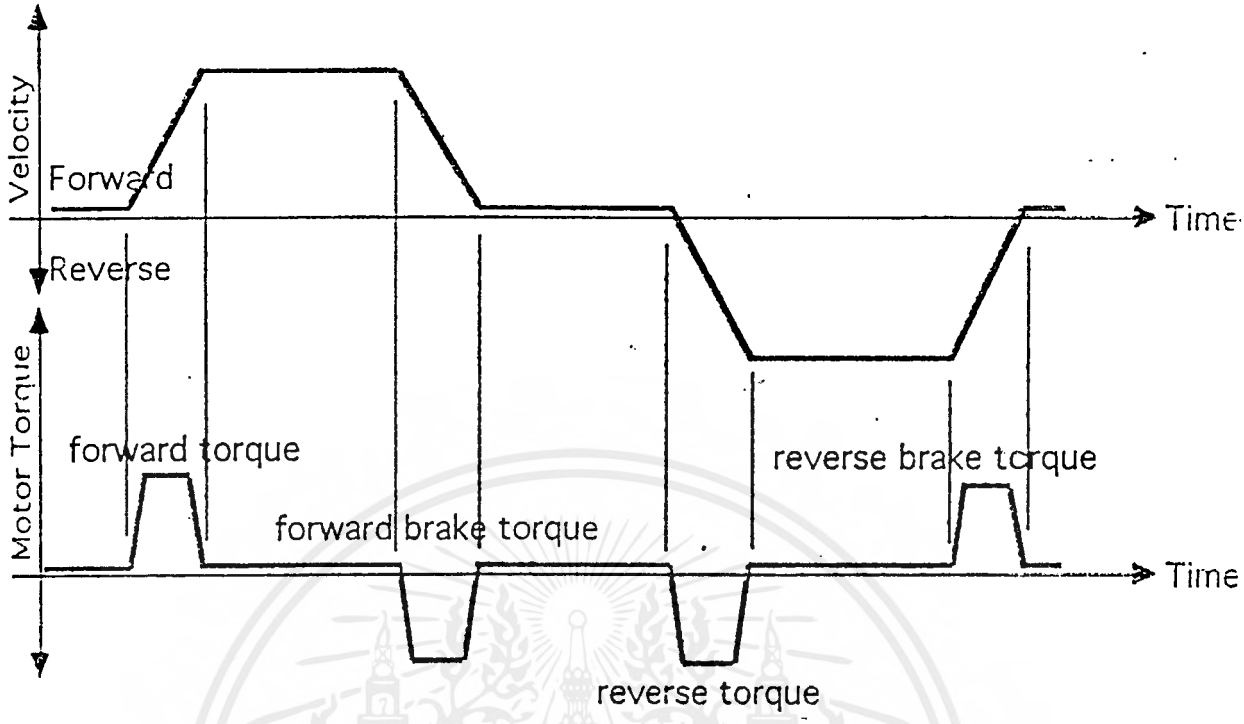
MAX R TORQ P307

MAX F BRAKE P308

MAX R BRAKE P309

"MAX F TORQ" , "MAX R TORQ" , "MAX F BRAKE" และ "MAX R BRAKE" เป็นการตั้งค่าแรงบิดสูงสุดของ forward และ reverse และอัตรา brake สูงสุดของ forward และ reverse ตามลำดับค่าสูงสุดที่ใส่ได้คือ 100% ซึ่งสัมพันธ์กับการตั้งค่าแรงบิดทั้งหมดตามคำสั่ง "I LIMIT" (P008) ค่าดังกล่าวสามารถปรับได้จาก ภายนอกโดยใช้ digital input คือ "User Vin" input ตามคำสั่ง (P413) ดูรูปประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงกราฟความสัมพันธ์ระหว่าง ความเร็วกับเวลา และแรงบิดกับเวลา

<u>หมายเหตุ</u>

ถ้าตั้งค่า แรงบิดและอัตรา brake ต่ำไปตัว เครื่อง จะสูญเสียการควบคุม ความเร็ว เป็นเพราะซอบเขตที่ตั้งนี้จะไม่อนุญาตให้แก้ไขแรงบิดเพื่อควบคุม ความเร็ว ให้ถูกต้องได้

NUM POLES P310

ตั้งค่าจำนวน Pole ของ มอเตอร์ ค่าดังกล่าวจะคำนวณได้จาก Ac Tune test โดยใช้สูตร

number of poles = (motor freq x 120) / (motor rpm)

ผลจะได้จำนวน pole ออกมา โดยกำหนดให้

- motor freq เป็นค่าที่ใส่ Ac Tune test โดย "mtr freq"
- motor rpm เป็นค่าที่ใส่ Ac Tune test โดย "mtr rpm"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TORQ SOURCE P311

เป็นค่าสำหรับใช้ใน โหมดแรงบิด แบ่งเป็น

INT :

ถ้าเลือก INT(ernal) ค่า แรงบิด ตั้งค่า จะถูกตั้งโดยคำสั่ง "INT TORQ" (P312)

EXT :

ถ้าเลือก EXT(ernal) ค่า แรงบิด ตั้งค่า จะถูกตั้งโดยใช้ local หรือ remote command analog input และคำสั่ง "INT TORQ" (P312)

INT TORQ P312

ถ้า "TORQ SOURCE" (P311) เป็น INT

"INT TORQ" เป็นค่าแรงบิดที่ใช้งานในขณะที่อยู่ที่ โหมดแรงบิด และ "TORQ SOURCE" (P311) เป็น INT ค่าสูงสุดที่ใส่ได้ในคำสั่งนี้คือ 100% ซึ่งสัมพันธ์กับการตั้งค่าแรงบิดทั้งหมดตามคำสั่ง "I LIMIT" (P008)

ถ้า "TORQ SOUCE" (P311) เป็น EXT

ถ้า P311 เป็น EXT ค่าแรงบิดที่ใช้งานในขณะอยู่ที่ โหมดแรงบิด จะเท่ากับ

ค่าแรงบิดทั้งหมดตาม "I LIMIT" x ("INT TORQ"/100) x (analog input/full analog input)

ค่าดังกล่าวสามารถปรับได้จากภายนอกทาง "User Vin" input (TB2-3) ตามคำสั่ง "USER VIN" P(413)

SERIAL BAUD P313

ตั้งค่า baud rate (อัตราการส่งข้อมูลตาม port.) ของ asynchronous serial port ซึ่งจะต้องตรงกับที่ตั้งไว้ใน host computer ค่าที่สามารถตั้งได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Value	Baud Rate
96	9600 bits/second
48	4800 bits/second
24	2400 bits/second
12	1200 bits/second
192	19200 bits/second

ตารางที่ 4.11 แสดงค่าอัตราบอดของเครื่อง

SERIAL ADDR P314

ตั้งค่า serial address ของตัว เครื่อง เพื่อการติดต่อผ่าน asynchronous serial port ซึ่งจะต้องตรงกับ host computer ถ้ามีมากกว่า 1 เครื่อง ต่อเข้ากับ host computer ตัว เครื่อง แต่ละตัวต้องมี "SERIAL ADDR" เฉพาะของตัวเอง

BIT PAR P315

ตั้งค่าจำนวน bit และ parity ของข้อมูลที่มาจาก asynchronous serial port ซึ่งจะต้องเลือกให้ตรงกับทาง host computer สามารถตั้งได้ดังนี้

Value	Data Bits	Parity
8 NONE	8 data bits	No parity
8 EVEN	8 data bits	Even parity
8 ODD	8 data bits	Odd parity
7 ODD	7 data bits	Odd parity
7 EVEN	7 data bits	Even parity

ตารางที่ 4.12 แสดงค่าบิตข้อมูลในการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHECK SUM P316

เป็นการตั้งค่าให้ตรวจ หรือ ไม่ตรวจ (Enable or disable) ค่าของ check sum เวลาเขียนข้อมูลที่ได้จาก asynchronous serial port เมื่อตั้งให้ enable จะต้องส่งค่า checksum (block สุดท้ายของข้อมูล) มากับข้อมูลเพื่อให้สมบูรณ์ และคำสั่งที่ส่งถูกต้อง

เมื่อ check sum เป็น disable จะส่งตัวอะไรก็ได้แทนใน block สุดท้ายของข้อมูล การใส่ check sum เป็นการป้องกันการผิดพลาดในการส่งข้อมูล

MDROP MODE P317

สามารถตั้งเป็น E , D , ME หรือ MD การตั้ง ME และ MD เป็นการใช้ monitor mode ที่ asynchronous serial port (ใช้สำหรับผู้ผลิตเท่านั้น , สำหรับผู้ใช้จะใช้ E หรือ D)

เมื่อ "MDROP MODE" = E :

(1) ตัว เครื่อง รุ่น 1100 สามารถต่อเข้าด้วยกันหลายๆ ตัวกับ host computer

(2) ตัว เครื่อง รุ่น 1100 จะมี 3 สถานะ คือ เป็นตัวส่ง , รับ และ disable (ขณะส่งจะไม่รับข้อมูล)

(3) monitor mode จะถูก disable

เมื่อ "MDROP MODE" = D :

(1) จะสามารถต่อ เครื่อง รุ่น 1100 ได้เพียง 1 ตัวใน เครือข่าย กับ host computer

(2) ตัว เครื่อง จะรักษาสถานะส่งตลอดเวลา

(3) monitor mode จะถูก disable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ "MDROP MODE" =ME :

จะเหมือนสถานะ E แต่จะสามารถใช้ monitor mode ได้

เมื่อ "MDROP MODE" =MD :

จะเหมือนสถานะ D แต่จะสามารถใช้ monitor mode ได้

DEFLT MENU P318

กำหนดให้เรียกค่า ดีฟอล์ต ของ พารามิเตอร์ ทั้งหมดในหมวด P3xx ให้เป็นค่าที่ผู้ผลิตตั้งไว้ต้องการเรียก ดีฟอล์ต เลือก " YES " หลังจากนั้นค่า พารามิเตอร์ จะกลับไปเป็น ดีฟอล์ต และ ค่าใน " DEFLT MENU " (P318) จะเปลี่ยนเป็น " NO " ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 อธิบายคำสั่งในหมวดที่ 4

SPARE P400

คำสั่งนี้ไม่ใช้งาน

DIR CONTROL P401

เป็นค่าที่ใช้ในการหาความเร็วเมื่อแหล่งของความเร็วมาจากอินพุท "Local cmd" (TB2-2) หรือ "Remote cmd" (TB2-4และ TB2-5) มีค่าตามตารางดังนี้

"DIR CONTROL"	Speed Setpoint Max. Positive	"Local Cmd" Zero / Max.Negative	Speed Setpoint Max. Positive	"Remote Cmd" zero
AUTO and "MIN" (P001) ≥ 0	"MAX" (P002)	"MIN" / "MIN"	"MAX" (P002)	"MIN" (P001)
AUTO and "MIN" (P001) < 0	"MAX" (P002)	0 / "MIN" (P001)	"MAX" (P002)	0
UNI(polar)	"L HI" (P402)	"L LO" / "L LO"	"R HI" (P408)	"R LO" (P409)
BIP(olor)	"L HI" (P402)	0 / "L LO" (P403)	"R HI" (P408)	"R LO" (P409)

ตารางที่ 4.13 แสดงค่าพารามิเตอร์ในคำสั่ง "DIR CONTROL" ที่ Local Cmd

หมายเหตุ

ในการเปลี่ยนค่า "DIR CONTROL" อาจทำให้ค่าพารามิเตอร์บางค่าเปลี่ยนแปลงโดยโปรแกรม ตามเงื่อนไขต่อไปนี้

(1) "MAX" ≥ 0 และ "MAX" > "MIN"

(2) "R LO" ≥ 0 และ "R HI" > "R LO"

(3) ถ้าเป็น UNI(polar) , "L LO" ≥ 0และ "L HI" > "L LO"

(4) ถ้าเป็น BIP(olar) , "L LO" ≤ 0 และ "L HI" > "L LO"

ค่าพารามิเตอร์ดังกล่าวใช้ในการหาค่าสั่งแรงบิด เมื่อแหล่งของแรงบิดจากอินพุท "Local Cmd" (TB2-4) หรือ "Remote Cmd" (TB2-4&5) มีค่าตามตารางดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"DIR CONTROL"	Torque Setpoint Max. Positive	"Local Cmd" Zero / Max.Negative	Torque Setpoint Max. Positive	"Remote Cmd" zero
AUTO and "MIN" (P001) ≥ 0	Max torque	0 / 0	Max torque	0
AUTO and "MIN" (P001) < 0	Max torque	0 / -max torque	Max torque	0
UNI(polar)	Max torque	0 / 0	Max torque	0
BIP(olor)	Max torque	0 / -max torque	Max torque	0

ตารางที่ 4.14 แสดงค่าพารามิเตอร์ของคำสั่ง "DIR CONTROL" ที่ Remote Cmd

หมายเหตุ

ค่า max torque = full torque x "INT TORQ" (P312)/100

L HI P402

ใช้ในการแบ่งสเกล (scaling) จาก "Local Cmd" (TB2-2) (แรงดันอะนาล็อกอินพุท) สัมพันธ์กับค่าสูงสุดของแรงดันอินพุท (ปกติเท่ากับ 10Vdc) ใช้ในโหมด Unipolar และ Bipolor เท่านั้น (ค่าใน "DIR CONTROL" (P401) ต้องเป็น UNIP หรือ BIP)

ค่านี้จะเปลี่ยนแปลงเมื่อ "DIR CONTROL" (P401) หรือ "L LO" (P411) มีค่าเปลี่ยนไป

L LO P403

ใช้ใน ในการแบ่งสเกล (scaling) จาก "Local Cmd" (TB2-2) (แรงดันอะนาล็อกอินพุท) สัมพันธ์กับค่าต่ำสุดของแรงดันอินพุท (ปกติ มีค่า 0 Vdc หรือ –10 Vdc) ใช้ในโหมด Unipolar และ Bipolor เท่านั้น (ค่าใน "DIR CONTROL" (P401) ต้องเป็น UNIP หรือ BIP) ค่านี้จะเปลี่ยนแปลงเมื่อ "DIR CONTROL" (P401) หรือ "L HI" (P411) มีค่าเปลี่ยนไป

L OFFSET P404

ค่าพารามิเตอร์นี้จะให้ผู้ใช้ปรับ แรงดันอะนาล็อกอินพุทที่ "Local Cmd" (TB2-2) เพื่อที่จะปรับสัญญาณที่ไม่อยู่ในช่วง 0 ถึง Vdc ออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้สามารถเลือกค่า 0 , สูงสุดและ ต่ำสุดของสัญญาณอินพุทและตัวเครื่องจะคำนวณค่าออฟเซต และเกน โดยอัตโนมัติค่าดีฟอล์ตของผู้ผลิตเป็น NOMINAL (แม้ว่าจะแสดงผลเป็น DONE) เป็นค่าตามปกติเมื่อไม่ต้องการความละเอียด เมื่อ DONE/Z SET/P SET/M ขึ้นให้ทำการปรับค่า "Local Cmd" แรงดันอินพุท ตามขั้นตอนต่อไปนี้

(1) ตั้งค่าใน potentiometer หรือแหล่งของสัญญาณอื่นที่ต้องการกำหนดให้เป็นแรงดัน 0

(2) ปรับ "L OFFSET" เป็น Z(ero) SET ตัวเครื่องจะรับค่าแรงดันอินพุทที่ป้อน และตั้งเป็นจุด 0 ค่าพารามิเตอร์จะเปลี่ยนเป็น DONE

(3) ตั้งค่าแรงดันบวกสูงสุด ที่ต้องการให้เป็นค่าแรงดันสูงสุด (ปกติเป็น 10 V)

(4) เปลี่ยนค่า "L OFESET" เป็น P(lus) SET ตัวเครื่องจะรับค่าแรงดันอินพุทที่ป้อนแล้วตั้งเป็นค่าสูงสุด ค่าพารามิเตอร์จะเปลี่ยนเป็น DONE

(5) ถ้าต้องการใช้ค่าลบของ แรงดันอินพุท "Local Cmd" (TB2-2) ให้ตั้งค่าลบสูงสุดของแรงดันที่ต้องการให้เป็นค่าแรงดันต่ำสุด (ปกติเป็น –10 V)

(6) เปลี่ยนค่า "L OFFSET" เป็น M(inus) SET ตัวเครื่องจะรับค่าแรงดันอินพุทที่ป้อน ค่าพารามิเตอร์จะเปลี่ยนเป็น DONE

NOMINAL :
ถ้าตั้งเป็น NOMINAL ค่าอะนาล็อคอินพุทที่ "Local Cmd" (TB2-2) จะมีสเกลระหว่าง 0 ถึง ± 10Vdc สัญญาณอินพุทที่เข้ามาจะอยู่ในช่วงนี้ และค่าพารามิเตอร์จะเปลี่ยนเป็น DONE

INVERSE และ DONE :
หลังจากปรับแรงดันอินพุตที่ "Local Cmd" (TB2-2) เปลี่ยนค่า "L OFFSET" เป็น INVERSE เพื่อเปลี่ยนอินพุทเป็นตรงข้าม ถ้าไม่ต้องการให้ "L OFFSET" เป็น DONE

INVERSE และ "DIR CONTROL" เป็น UNI(polar) :
ค่าแรงดันสูงสุด (ปกติเป็น 10V) จะถูกเลือกเป็นค่าความเร็วต่ำสุด (เช่น 0 rmp) หรือค่าแรงบิด 0 ที่โหมดแรงบิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติแล้วแรงดัน 0 V จะเป็นการเลือกความเร็วสูงสุด (เช่น 1800 rpm) หรือแรงบิดสูงสุดในโหมดแรงบิด

INVERSE และ "DIR CONTROL" เป็น BIP(olar) :

ค่าแรงดันสูงสุด (ปกติ 10 V) จะถูกเลือกเป็นค่าความเร็วต่ำสุด (เช่น –1800 rpm) หรือแรงบิดที่มีค่าเป็นลบสูงสุดในโหมดแรงบิด

ปกติแล้วค่าแรงดัน 0 เป็นการเลือกค่าความเร็ว 0 หรือแรงบิด 0 ในโหมดแรงบิด

ค่าแรงดันต่ำสุด (ปกติ –10V) เป็นการเลือกค่าความเร็วสูงสุด (เช่น 1800 rpm) หรือค่าแรงบิดสูงสุดในโหมดแรงบิด

L OFFSETV P405

ค่าแรงดันอินพุทที่ "Local Cmd" (TB2-2) ตามดัชนี 32767 มีค่าเป็น 10Vdc ค่าในคำสั่งนี้จะถูกคำนวณโดยอัตโนมัติเมื่อ "L OFFSET" มีค่าเป็น Z SET หรือจะป้อนค่าเองก็ได้

L POSX P406

เป็นค่าเกนบวก (positive gain) ของแรงดันอินพุทที่ "Local Cmd" (TB2-2) ตามดัชนี 32767 มีค่าเป็น 10 V (0 ถึง 10V) ค่าในคำสั่งนี้จะถูกคำนวณโดยอัตโนมัติเมื่อ "L OFFSET" มีค่าเป็น P SET หรือจะป้อนค่าเองก็ได้

L NEGX P407

เป็นค่าเกนลบ (negative gain) ของ แรงดันอินพุทที่ "Local Cmd" (TB2-2) ตามดัชนี 32767 มีค่าเป็น 10 V (0 ถึง -10V) ค่าในคำสั่งนี้จะถูกคำนวณโดยอัตโนมัติเมื่อ "L OFFSET" มีค่าเป็น M SET หรือจะป้อนค่าเองก็ได้

R HI P408

ใช้ในการแบ่งสเกลจากกระแสอะนาล็อกอินพุท "Remote Cmd" (TB2-4&5) ค่านี้จะสัมพันธ์กับค่าอินพุทสูงสุด (ปกติ 20 mA) และใช้ในโหมด unipolar หรือ bipolar (เมื่อ "DIR CONTORL" (P401) เป็น UNIP หรือ BIP) ค่าดังกล่าวจะได้รับผลกระทบถ้าเปลี่ยน "DIR CONTORL" (P401) หรือ "R LO" (P409)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R LO P409

ใช้ในการแบ่งสเกลจากกระแสอะนาล็อกอินพุท "Remote Cmd" (TB2-4&5) ค่านี้จะสัมพันธ์กับค่าอิดพุทต่ำสุด (ปกติ 0 mA หรือ 4 mA) ใช้ใน โหมด unipolar หรือ bipolar (เมื่อ "DIR CONTORL" (P401) เป็น UNIP หรือ BIP) ค่าดังกล่าวจะได้รับผลกระทบถ้าเปลี่ยน "DIR CONTORL" (P401) หรือ "R HI" (P409)

R OFFSET P410

ค่าพารามิเตอร์นี้ จะให้ผู้ใช้ปรับกระแสอะนาล็อกอินพุท ที่ "Remote Cmd" (TB2-4&5) เพื่อที่จะปรับสัญญาณที่ไม่อยู่ในช่วง 0 ถึง 20 mA หรือ 4 ถึง 20 mA ออกผู้ใช้สามารถเลือกค่า 0 , สูงสุด และต่ำสุด ของสัญญาณอินพุท และตัวเครื่องจะคำนวณค่า ออฟเซต และเกน โดยอัตโนมัติค่าดีฟอล์ตของผู้ผลิตเป็น NOM4 20 (แม้ว่าหน้าจอแสดงผลเป็น DONE) เป็นค่าตามปกติเมื่อ ไม่ต้องการความละเอียด เมื่อ DONE / Z SET / P SET / NOM4 20 / NOM0 20 ขึ้นให้ทำตามขั้นตอนดังนี้

(1) ตั้งค่าในแหล่งสัญญาณที่ต้องการ กำหนดให้เป็นกระแสต่ำสุด
(2) เปลี่ยน "R OFFSET" เป็น Z(ero) SET ตัวเครื่องจะรับค่ากระแสอินพุทที่ป้อน และตั้งเป็นจุด 0 แล้วค่าพารามิเตอร์จะเปลี่ยนเป็น DONE
(3) ป้อนกระแสบวกสูงสุดที่ต้องการ กำหนดให้เป็นกระแสสูงสุด (ปกติ 20 mA)
(4) เปลี่ยน "R OFFSET" เป็น P(lus) SET ตัวเครื่องจะรับค่ากระแสอินพุทที่ป้อนแล้วตั้งเป็นค่ากระแสบวกสูงสุด จากนั้นค่าพารามิเตอร์จะเปลี่ยนเป็น DONE

NOM4 20 (mA) :
ถ้าตั้งค่านี้ อะนาล็อกอินพุทที่ "Remote Cmd" จะมีสเกลอยู่ระหว่าง 4 ถึง 20 mA ค่าสัญญาณอินพุทจะอยู่ในช่วงนี้

NOM0 20 (mA) :
ถ้าตั้งค่านี้ อะนาล็อกอินพุทที่ "Remote Cmd" จะมีสเกลอยู่ระหว่าง 0 ถึง 20 mA ค่าสัญญาณอินพุทจะอยู่ในช่วงนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INVERSE และ DONE :

หลังจากปรับกระแสอินพุตที่ "Remote Cmd" (TB2-4&5) จะเปลี่ยนค่า "R OFFSET" เป็น INVERSE เพื่อเปลี่ยนอินพุตเป็นตรงข้าม ถ้าไม่ต้องการให้ "R OFFSET" เป็น DONE

เมื่ออินพุตถูกปรับเป็นตรงข้าม ค่ากระแสสูงสุด (ปกติ 20 mA) จะตรงกับค่าความเร็วต่ำสุด (เช่น 0 rpm) หรือแรงบิด 0 ในโหมดแรงบิด

โดยปกติแล้ว ค่ากระแสศูนย์ (0 mA หรือ 4mA) จะเลือกค่าความเร็วสูงสุด (เช่น 1800 rpm) หรือแรงบิดสูงสุดในโหมดแรงบิด

R OFFSETV P411

ค่ากระแสอินพุทที่ "Remote Cmd" (TB2-4&5) ตามดัชนี 32767 มีค่า 20 mA ค่าในคำสั่งนี้ จะถูกคำนวณโดยอัตโนมัติเมื่อ "R OFFSET" มีค่าเป็น Z SET หรือจะป้อนเองก็ได้

R POSX P412

เป็นค่าเกนบวกของกระแสอินพุทที่ "Remote cmd" (TB2-4&5) ตามดัชนี 32767 มีค่าเป็น 20 mA (0 ถึง 20 mA) ค่าในคำสั่งนี้ จะถูกคำนวณโดยอัตโนมัติเมื่อ "R OFFSET" มีค่าเป็น Z SET หรือจะป้อนเองก็ได้

USER VIN P413

ตั้งค่าพารามิเตอร์ให้สัญญาณอะนาล็อกอินพุทที่ "User Vin" (TB2-3) เปลี่ยนแปลง ค่านี้จะปรากฏอยู่ในคำสั่งต่างๆ เหล่านี้

OFF

"JOG" (P005)

"I LIMIT" (P008)

"PS1" (P100)

"PS2" (P103)

"PS3" (P106)

"PS4" (P109)

"PS5" (P112)

"PS6" (P115)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"PS7" (P118)

"RATIO NUM" (P209)

"MAX" (P002)

"EXT AMPS" (P304

"MAX F TORQ" (P306)

"MAX R TORQ" (P307)

"MAX F BRAKE" (P308)

"MAX R BRAKE" (P309)

"INT TORQ" (P312)

"SET PT" (P000)

LOCA

ROMO

เลือก OFF เมื่อไม่มีการเปลี่ยนแปลงค่าพารามิเตอร์

หมายเหตุ

(1) เมื่อ "USER VIN" เป็น LOCAL แรงดันอินพุทที่ "User Vin" จะเป็นตัวปรับค่าเกนของแรงดันอินพุทที่ "Local Cmd" (TB2-2) ค่าเกนปรับในโหมดมาสเตอร์ และโหมดสลาฟ ไม่มีผลกระทบในโหมดแรงบิด

(2) เมื่อ "USER VIN" เป็น REMO แรงดันอินพุทที่ "User Vin" เป็นตัวปรับค่าเกนของกระแสอินพุทที่ "Remote Cmd" (TB2-4&5) ค่าเกนปรับในโหมดมาสเตอร์ และโหมดสลาฟ ไม่มีผลกระทบในโหมดแรงบิด

UVIN HI P414

UVIN LO P415

UVIN MODE P416

"USER VIN" (P413) เป็นการตั้งค่าพารามิเตอร์ให้สัญญาณอะนาล็อกอินพุทที่ "User Vin" (TB2-3) เปลี่ยนแปลง "UVIN HI" , "UVIN LO" , "UVIN MODE" และ "UVIN OS" (P417) เป็นการเลือกวิธีที่ค่าพารามิเตอร์จะถูกเปลี่ยนแปลง ตามตารางข้างล่าง

เมื่อ "UVIN MODE" = SCALE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Setting of "User Vin" Analog Input (TB2-3)	Modified Value
Maximum voltage	*ค่า "USER VIN" x "UVIN HI" / 100*
Zero	*ค่า "USER VIN" x "UVIN LO" / 100*

ตารางที่ 4.15 แสดงคำสั่งเมื่อ "UVIN MODE" เป็น SCALE

เมื่อ "UVIN MODE" = TRIM

Setting of "User Vin" Analog Input (TB2-3)	Modified Value
Maximum voltage	*ค่า "USER VIN" x "UVIN HI" / 100 + ค่า "USER VIN"*
Zero	*ค่า "USER VIN" x "UVIN LO" / 100 + ค่า "USER VIN"*

ตารางที่ 4.16 แสดงคำสั่งเมื่อ "UVIN MODE" เป็น TRIM

หมายเหตุ

ตั้งค่าบวก "UVIN HI" และลบ "UVIN LO" เพื่อที่จะบวก และลบออกจากค่า "USER UIN" ขณะที่อะนาล็อกอินพุทที่ "User Vin" เปลี่ยนแปลง

UVIN OS P417

ค่าพารามิเตอร์นี้จะให้ผู้ใช้ ปรับค่าสัญญาณอะนาล็อกอินพุทที่ "User Vin" (TB2-3) เพื่อที่จะปรับ สัญญาณที่มีค่านอก 0 ถึง 10V ออก ผู้ใช้สามารถตั้งค่า 0 และจุดสูงสุดของสัญญาณอินพุท และเครื่องจะทำการคำนวณค่า ออฟเซต และเกน โดยอัตโนมัติ ค่าดีฟอล์ตของผู้ผลิตเป็น NOMINAL (แสดงผลเป็น DONE) เป็นค่าตามปกติเมื่อไม่ต้องการความละเอียด

เมื่อ DONE / Z SET / P SET / M ขึ้นให้ทำการปรับค่าอะนาล็อกอินพุทของผู้ใช้ตามขั้นตอนดังนี้

(1) ตั้งค่าในแหล่งจ่ายสัญญาณ ที่ต้องการกำหนดให้เป็นแรงดัน 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) เปลี่ยน "UVIN OS" เป็น Z(cro) SET ตัวเครื่องจะรับแรงดันอินพุทที่ป้อน และตั้งเป็นจุด 0 และค่าพารามิเตอร์จะเปลี่ยนเป็น DONE

(3) ป้อนแรงดันบวกสูงสุดที่ต้องการกำหนดให้เป็นแรงดันสูงสุด (ปกติ 10V)

(4) เปลี่ยน "L OFFSET" เป็น P(lus) SET ตัวเครื่องจะรับแรงดันอินพุทที่ป้อน และตั้งเป็นจุดแรงดันบวก และค่าพารามิเตอร์จะเปลี่ยนเป็น DONE

NOMINAL :

ถ้าตั้งเป็น NOMINAL ค่าสเกลอินพุทของผู้ใช้จะอยู่ในช่วง 0 ถึง 10 Vdc

UVIN OV P418

ค่าแรงดันอินพุทที่ "User Vin" (TB2-3) ตามดัชนี 32767 มีค่าเป็น 10 Vdc ค่าในคำสั่งนี้จะถูกคำนวณโดยอัตโนมัติ เมื่อ "UVIN OS" มีค่าเป็น Z SET หรือจะป้อนเองก็ได้

UVIN PX P419

เป็นค่าเกนบวกของแรงดันอินพุทที่ "User Vin" (TB2-3) ตามดัชนี 32767 มีค่าเป็น 10 V (0 ถึง 10Vdc) ค่าในคำสั่งนี้จะถูกคำนวณโดยอัตโนมัติ เมื่อ "UVIN OS" มีค่าเป็น P SET หรือจะป้อนเองก็ได้

LOCAL VOUT P420

LVOUT HI P421

LVOUT LO P422

"LOCAL VOUT" เป็นการกำหนดให้สัญญาณอะนาล็อกเอาท์พุท "Local Vout" (TB2-9) ทำงาน "LOCAL VOUT" อาจตั้งเป็น "SPEED" , "TORQ" หรือ "PTR" "VOUT LO" และ "LVOUT HI" ใช้ปรับสเกลของอะนาล็อกเอาท์พุท

SPEED :

ถ้าตั้งเป็น SPEED ค่าสัญญาณอะนาล็อกเอาท์พุทที่ "Local Vout" จะมีผลต่อค่าฟีดแบกของความเร็วในคำสั่ง "SPEED" (P500) ค่าอะนาล็อกเอาท์พุทมีสเกลดังนี้

10V = "MAX" (P002) x "LVOUT HI" / 100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0V = "MAX" (P002) x "LVOUT LO" /100

-10V = -"MAX" (P002) x "LVOUT HI" /100 + ("MAX" (P002) x "LVOUT LO" / 50

<u>หมายเหตุ</u>

ปกติแล้ว "LVOUT HI" จะมีค่า 100 และ "LVOUT LO" มีค่า 0 จะได้ว่า

10V = "MAX" (P002)

0V = 0

-10V = -"MAX" (P002)

TORQ :

ถ้าเลือก TORQ ค่าสัญญาณอะนาล็อกเอาท์พุทที่ "Loval Vout" จะมีผลต่อคำสั่งแรงบิดของมอเตอร์ "MOTORTORQ" (P503)

ค่าอะนาล็อกเอาท์พุทมีสเกลดังนี้

10V = 200% x "LVOUT HI" / 100

0V = 200^ x "LVOUT LO" /100

-10V = -200% "LVOUT HI" / 100 + (200% x "LVOUT LO" /50)

<u>หมายเหตุ</u>

ปกติแล้ว "LVOUT HI" จะมีค่า 100 และ "LVOUT LO" มีค่า 0 จะได้ว่า

10V = 200%

0V = 0

-10V = -200%

PTR :

สำหรับผู้ผลิตเท่านั้น

USER UOUT P423

UVOUT HI P424

UVOUT LO P425

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"USER VOUT" เป็นการกำหนดให้สัญญาณอะนาล็อกเอาท์พุทที่ "User Vout" (TB2-10) ทำงาน ค่าพารามิเตอร์ที่เลือกได้มีดังนี้ (เรียงตามลำดับที่ปรากฏในหน้าจอ)

"MOTORTORQ" (P503)

"MOTOR I" (P504)

"MOTOR V" (P506)

"RMOTORTOR" (P507)

"IN WATTS" (P508)

"INPUT PF" (P509)

"BUS VOLTS" (P510)

"EXT ACCUM" (P511)

"DB ACCUM" (P512)

(P599) (สำหรับผู้ผลิตเท่านั้น)

"SPEED" (P500)

"SET SP" (P501)

ค่าอะนาล็อก เอาท์พุทที่ "User Vout" มีสเกลดังนี้

สำหรับ "MOTORTORQ" และ "RMOTORTOR"

$$10V = 200\% \times \text{"LVOUT HI"} / 100$$

$$0V = 200\% \times \text{"LV OUT LO"} / 100$$

$$-10V = 200\% \times \text{"LVOUT HI"} / 100 + (200\% \times \text{"LVOUT LO"} / 50)$$

หมายเหตุ

ปกติแล้ว "LVOUT HI" จะมีค่า 100 และ "LVOUT LO" มีค่า 0 จะได้ว่า

10V = 200%

0V = 0

-10V = -200%

สำหรับทุกพารามิเตอร์

$$10V = \text{"LVOUT HI"}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0V = "LVOUT LO"

-10V = "LVOUT HI" + "LVOUT LO" x 2

หน่วย และทศนิยมสำหรับ "LVOUT HI" และ "LVOUT LO" จะเหมือนกับพารามิเตอร์ที่เลือก

UIN P426

เลือกผลตอบสนองเมื่อตั้งสัญญาณดิจิตอลอินพุท "User" (TB1-14) เป็นกราวด์ (ground)

NONE :

เมื่อไม่ใช้ดิจิตอลอินพุท่

COAST CTL :

เมื่อตั้ง (set) ดิจิตอลอินพุท "User" (TB1-14) ให้มอเตอร์หยุดหมุนเองเมื่อสั่งหยุด

เมื่อล้าง (clear) ดิจิตอลอินพุท (TB1-14) จะทำให้มอเตอร์หยุดหมุนตามคำสั่ง "STOP MODE" (P202)

SLAVE CTL :

เมื่อตั้งดิจิตอลอินพุท "Uscr" (TB1-14) ตัวเครื่องจะไม่ทำคำสั่ง "RUN MODE" (P201) ในสลาฟโหมด

เมื่อล้างดิจิตอลอินพุท (TB1-14) ตัวเครื่องจะไม่ทำคำสั่ง "RUN MODE" (P201) เมื่ออยู่ในโหมดมาสเตอร์

ZERO TORQ :

ถ้าอยู่ในโหมดแรงบิด และดิจิตอลอินพุท (TB1-14) ถูกเลือกแรงบิดของเครื่องจะกลายเป็น 0

SPEED TORQ :

เมื่อตั้งดิจิตอลอินพุท (TB1-14) ตัวเครื่องจะเข้าสู่โหมดแรงบิด

เมื่อล้างจะกลับเป็นโหมดวิ่งตามคำสั่ง "RUN MODE" (P201)

FORWARD :

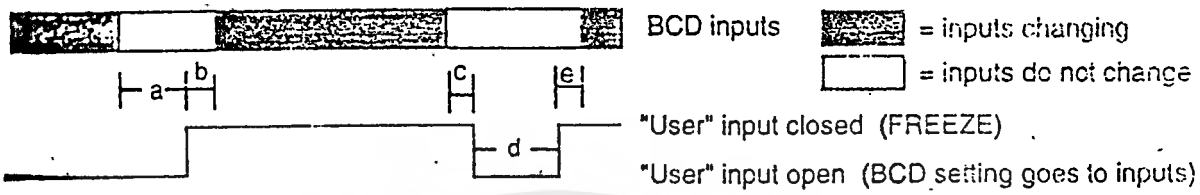
เมื่อเลือก FORWARD และดิจิตอลอินพุท (TB1-14) กลายเป็น ฟอร์เวอร์อินพุท (forward input) และ "FWD REV" และ "NO SPEED SELECT" ถูกกำหนดให้ทำงาน (enable)

ค่าดิจิตอลอินพุท (TB1-19) "Reverse" จะใช้สำหรับกลับทิศการหมุน (reverse)

FREEZE BCD :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อดิจิตอลอินพุท "User" (TB1-14) ถูกตั้งค่า BCD อินพุท (TB1-16,17,18) จะถูกแช่แข็ง (Frozen) ความคิดนี้เป็นการสลับกันทำงานเปลี่ยนค่า BCD ขณะอยู่ในช่วงของมัน และเปิด "User" อินพุท เมื่อต้องการเปลี่ยนค่า ดูภาพอธิบายต่อไปนี้



รูปที่ 4.6 แสดงช่วงเวลาทำงานของ BCD และ "User" อินพุต

a = หยุดการเปลี่ยนค่าที่ BCD อินพุทอย่างน้อย 0.05 วินาทีก่อนที่ "User" อินพุทจะปิด

b = BCD อินพุทคงที่เป็นเวลาอย่างน้อย 0.01 วินาทีหลัง "User" อินพุทปิด

c = หยุดเปลี่ยนค่าที่ BCD อินพุทเป็นเวลาอย่างน้อย 0.01 วินาทีก่อน "User" เปิด

d = "User" อินพุทเปิดอย่างน้อย 0.05 ค่าที่เปลี่ยนแปลงจะมีผลภายใน 0.02 ถึง 0.04 วินาที

e = แช่แข็งค่า BCD อินพุทเป็นเวลาอย่างน้อย 0.01 วินาที

USER RELAY P427

UR SET P428

UR CLR P429

USER OUT P430

UO SET P431

UO CLR P432

"USER RELAY" กำหนดค่าพารามิเตอร์ หรือฟังก์ชั่น ที่ควบคุมรีเลย์อินพุท (TB1-5 ,TB1-6 , TB1-7) เมื่อเป็นพารามิเตอร์ค่า "UR SET" จะควบคุมจุดปิด (output setpoint) (หน้าสัมผัสปิด N.C.) และ "UR CLR" ควบคุมจุดเปิด (output clearpoint) (หน้าสัมผัสเปิด N.O.) เมื่อเป็นฟังก์ชั่นค่าทั้งสองไม่ต้องทำอะไร ค่า parameter และ function ที่ตั้งได้อธิบายในตารางข้างล่าง ตามลำดับที่ปรากฏในหน้าจอ

"USER OUT" กำหนดค่าพารามิเตอร์ หรือฟังก์ชั่นที่ควบคุมเอาท์พุท TB1-10 (open collector output "User O.C.") จะควบคุมจุดปิด (output setpoint) และ "UO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLR" จะควบคุมจุดเปิด output clearpoint เมื่อเป็นฟังก์ชั่นค่าทั้งสองไม่ต้องทำอะไร ค่า parameter และ function ที่ตั้งได้อธิบายในตารางข้างล่าง ตามลำดับที่ปรากฏในหน้าจอ

Parameter Settings

User Setting	Parameter or Description	Setting type
P500	"SPEED"	Parameter
P501	"SET SP"	Parameter
P502	"SPEED ERR"	Parameter
P503	"MOTORTORQ"	Parameter
P504	"MOTOR I"	Parameter
P505	"MOTOR HZ"	Parameter
P506	"MOTOR V"	Parameter
P507	"RMOTORTOR"	Parameter
P508	"IN WATTS"	Parameter
P509	"INPUT PF"	Parameter
P510	"BUS VOLTS"	Parameter
P511	"IXT ACCUM"	Parameter
P512	"DB ACCUM"	Parameter
P513	"TOTAL KWH"	Parameter
P514	"HOURS RUN"	Parameter
FWDFD	Forward feedback	Parameter
REVFD	Reverse feedback	Parameter

ตารางที่ 4.17 แสดงค่าพารามิเตอร์ของคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function Settings

User Setting	Parameter or Description	Setting type
MTRON	Motor on	Function
FAULT	Fault	Function
FWD	Forward selected	Function
REV	Reverse selected	Function
FWDON	Forward and motor on	Function
REVON	Reverse and motor on	Function
TOLIM	Torque limit	Function
RUN	Run	Function
MSPED	Master speed mode	Function
SSPED	Slave speed mode	Function
TORQ	Torque mode	Function
LOCAL	Local selected	Function
REMOT	Remote selected	Function
JOG	Jog speed selected	Function
USEIN	User in selected	Function
ATSPD	At speed	Function
WDT	Watch dog timer	Function

ตารางที่ 4.18 แสดงค่าฟังก์ชั่นของคำสั่ง

พารามิเตอร์ที่ตั้งเป็น P500-P514 เอาท์พุทปิด หรือเปิด เมื่อค่าพารามิเตอร์เป็นดังนี้

เอาท์พุทปิดถ้า

ค่า ≥ "SET" หรือ ค่า ≤ –"SET"

เอาท์พุทเปิดถ้า

ค่า < "CLR" และ ค่า > -"CLR"

ถ้าค่าอยู่ระหว่าง "SET" และ "CLR" หรือ –"SET" และ –"CLR" ค่าเอาท์พุทจะอยู่ในสถานะก่อนหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์ที่ตั้งเป็น FWDFD

เอาท์พุทปิดหรือเปิดตามตาราง

เอาท์พุทปิดถ้า

ความเร็วย้อนกลับ (feedback velocity) ของเครื่อง > "SET" และตัวเครื่องทำงาน (มีกระแสในมอเตอร์)

เอาท์พุทเปิดถ้า

ความเร็วย้อนกลับ < "CLR" หรือตัวเครื่องปิด

ถ้าค่าอยู่ระหว่าง "SET" และ "CTR" และตัวเครื่องเปิดค่าเอาท์พุทอยู่ในสถานะก่อนหน้า

ค่า "SET" และ "CTR" มีหน่วยตาม [ENG]

พารามิเตอร์ที่ตั้งเป็น REVFD

เอาท์พุทปิด หรือเปิดตามตาราง

เอาท์พุทปิดถ้า

ความเร็วย้อนกลับของเครื่อง <- "SET" และตัวเครื่องทำงาน (มีกระแสในมอเตอร์)

เอาท์พุทเปิดถ้า

ความเร็วย้อนกลับ >- "CLR" หรือตัวเครื่องปิด

ถ้าค่าอยู่ระหว่าง –"SET" และ –"CTR" และตัวเครื่องเปิดเอาท์พุทจะอยู่ในสถานะก่อนหน้า หน่วยของ "SET" และ "CLR" เป็น [ENG]

สำหรับค่าพารามิเตอร์ทุกตัว

- ค่าของทศนิยมแสดงอัตโนมัติโดยค่าพารามิเตอร์ที่เลือกใน "USER RELAY" หรือ "USER OUT"
- ค่า "UR SET" ≥ "UR CLR" และเมื่อเปลี่ยนค่า "UR SET" โปรแกรมจะเปลี่ยน "UR CLR" โดยอัตโนมัติ หรือถ้าเปลี่ยนค่า "UR CLR" โปรแกรมจะเปลี่ยน "UR SET" โดยอัตโนมัติ
- ค่า "UO SET" ≥ "UO CLR" และเมื่อเปลี่ยนค่า "UO SET" โปรแกรมจะเปลี่ยน "UO CLR" โดยอัตโนมัติ หรือถ้าเปลี่ยนค่า "UO CLR" โปรแกรมจะเปลี่ยน "UO SET" โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.19 แสดงความหมายของค่า ฟังก์ชั่น

ค่าของฟังก์ชั่น	ความหมาย
MTRON	เอาท์พุทจะขึ้น(set) ทันทีเมื่อเครื่องทำงาน (กระแสจะไหลภายในเวลา 0.01 + "AD DEL" (P208) วินาที) เอาท์พุทลง(clear) เมื่อปิดเครื่อง (กระแสในมอเตอร์หยุดไหล)
FAULT	เอาท์พุทจะขึ้นเมื่อเครื่องเกิดฟอล์ต หรือทำการ auto restart
FWD	เอาท์พุทขึ้นเมื่อเลือก forward ตัวเครื่องยังสามารถหมุน reverse ได้
REV	เอาท์พุทขึ้นเมื่อเลือก reverse ตัวเครื่องยังสามารถหมุน forward ได้
FWDON	เอาท์พุทขึ้นเมื่อเลือก forward , ตัวเครื่องทำงาน และ เครื่องสั่งให้หมุนที่สปีด (speed) ≥ 0
REVON	เอาท์พุทขึ้นเมื่อเลือก reverse , ตัวเครื่องทำงาน และเครื่องสั่งให้หมุนที่สปีด (speed) ≤ 0
TQLIM	เอาท์พุทขึ้นเมื่อเครื่องทำงานอยู่ในลิมิต (limit) ของ "MAX F TORQ" (P306) , "MAX R TORQ" (P307) , "MAX F BRAKE (P308) หรือ "MAX R BRAKE (P309)"
RUN	เอาท์พุทขึ้นเมื่อตัวเครื่องหมุน เอาท์พุทลงเมื่อตัวเครื่องหยุด (หรือกำลังหน่วงเพื่อหยุด)
MSPED	เอาท์พุทขึ้นเมื่อตัวเครื่องอยู่ในโหมดมาสเตอร์สปีด (Master speed mode)
SSPED	เอาท์พุทขึ้นเมื่อตัวเครื่องอยู่ในโหมดสลาฟสปีด (Slave speed mode)
TORQ	เอาท์พุทขึ้นเมื่อตัวเครื่องอยู่ในโหมดแรงบิด (Torque mode)
LOCAL	เอาท์พุทขึ้นเมื่อเลือกแหล่งควบคุมความเร็วระยะใกล้
REMOT	เอาท์พุทขึ้นเมื่อเลือกแหล่งควบคุมความเร็วระยะไกล
JOG	เอาท์พุทขึ้นเมื่ออยู่ในโหมด Jog
USEIN	เอาท์พุทขึ้นเมื่อเลือกอินพุท "User"
ATSPD	เอาท์พุทขึ้นเมื่อเปิดเครื่อง และหมุนที่สปีดที่ตั้งไว้
WDT	เอาท์พุทจะขึ้นและลง (Toggle) ที่ความถี่ 25 Hertz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEFLT MENU P433

กำหนดให้เรียกค่า ดีฟอล์ต ของ พารามิเตอร์ ทั้งหมดในหมวด P3xx ให้เป็นค่าที่ผู้ผลิตตั้งไว้ต้องการเรียก ดีฟอล์ต เลือก " YES " หลังจากนั้นค่า พารามิเตอร์ จะกลับไปเป็น ดีฟอล์ต และ ค่าใน " DEFLT MENU " (P318) จะเปลี่ยนเป็น " NO " ทันที

LVOUT PTR P434

UVOUT PTR P435

สำหรับผู้ผลิตเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 อธิบายคำสั่งในหมวดที่ 5

พารามิเตอร์ในหมวดที่ 5 นี้ จะแสดงผลได้เพียงอย่างเดียว ไม่สามารถแก้ไขได้ ประกอบด้วยพารามิเตอร์ ดังนี้

SPEED P500

การแสดงค่า ความเร็ว ของ มอเตอร์ ขณะหมุน (ในหน่วยของ [ENG] ตามคำสั่ง P009-P013)

SET SP P501

แสดงค่าความผิดพลาดของ ความเร็ว โดยคำนวณ จากสูตร

"SPEED ERROR" = (P501 – P500) x100 / P501

ถ้า speed error > 999 หน้าจอจะแสดงผลแค่ 999 หรือ

ถ้า speed error < -999 หน้าจอจะแสดงผลแค่ –999

MOTORTORQ P503

แสดงค่าแรงบิดของ มอเตอร์ เป็น percent (ของ full load current จาก nameplate ตามคำสั่ง "MTR AMPS" (P300))

MOTOR I P504

กระแสที่ไหลใน มอเตอร์ หน่วยเป็น Amperes RMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOR HZ P505

ความเร็ว ของ มอเตอร์ ในหน่วย Hertz

MOTOR V P506

แรงดันของ มอเตอร์ หน่วย volts RMS

RMOTORTOQ P507

ขณะอยู่ใน โหมดแรงบิด

ค่าแรงบิดที่ป้อนเข้าจะแสดงผลเป็น % ของแรงบิดของ มอเตอร์ ไม่ว่าเครื่อง จะเปิดหรือปิดถ้า เครื่อง เปิด แรงบิดที่สั่งจะเท่ากับ แรงบิดที่ป้อนเข้า ถ้าตัว เครื่อง ยังทำงานในขอบเขตของ ความเร็ว ที่กำหนด

ขณะอยู่ใน โหมดความเร็ว

แรงบิดที่สั่ง (เป็น % ของแรงบิดของ มอเตอร์) ค่าที่ 100% จะเป็นค่าแรงบิด ที่ "MTR AMPS" (P300) เหมือนกับค่าจากคำสั่ง "MOTORTORQ" (P503) เว้นแต่ค่าหลังจะเป็นค่าที่ผ่านการคำนวณมาแล้ว

IN WATTS P508

ค่า power (หน่วยเป็น watt) โดยประมาณ

INPUT PF P509

ค่า power factor โดยประมาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BUS VOLTS P510

ค่าแรงดันที่ DC bus

IXT ACCUM P511

ค่าของตัว I^2t แสดงเป็น % ของเวลาขณะเกิด overcurrent trip เมื่อค่านี้ = 100 ตัวเครื่อง จะ trip ทันที

DB ACCUM P512

เป็นค่าจากการคำนวณจักรการ break แบบ dynamic ค่าที่คำนวณได้จะนับขึ้นเมื่อค่าของวัฏจักรการ break มีค่า > 7% และจะนับลงถ้าค่าของวัฏจักร break < 7% (ค่าต่ำสุดเป็น 0) และถ้าเกิดการ break อย่างต่อเนื่อง ค่าที่คำนวณจะเป็น 100% ในเวลา 5 วินาที ถ้าตั้ง "EXCESSIVE DB" เป็น enable ตัวเครื่อง จะ trip เมื่อค่าที่คำนวณได้เป็น 100%

TOTAL kWh P513

ค่าในการใช้พลังงานโดยประมาณ (ในหน่วยของ kilowatt – hours) นับตั้งแต่ติดตั้ง หรือตั้งแต่ตั้งค่าให้หน่วยความจำทั้งหมดเป็น ดีฟอล์ต จากผู้ผลิตหรือผู้ใช้ การนับค่านี้จะเริ่มจาก 0 ถึง 999,999 kW-h และไม่สามารถ reset ได้

HOURS RUN P514

จำนวนชั่วโมงที่ตัวเครื่องทำการนับตั้งแต่ติดตั้ง มีค่าจาก 0 ถึง 999,999 ชั่งโมง และไม่สามารถ reset ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INT P515

เป็นสถานะของ digital input แต่ละ bit จะมีผลต่อ input ที่ใช้งานเรียงจากซ้ายไปขวาเช่น bit ที่ 1 จะเท่ากับ TB1-12 ตามตารางต่อไปนี้

Bit	Input Name	Terminal	Polarity
1	No external fault	TB1-12	(0=Open, 1=tied to LCOM)
2	Fault reset	TB1-13	(0=Open, 1=tied to LCOM)
3	User	TB1-14	(0=Open, 1=tied to LCOM)
4	Jog	TB1-15	(0=Open, 1=tied to LCOM)
5	BCD Speed 4	TB1-16	(0=Open, 1=tied to LCOM)
6	BCD Speed 2	TB1-17	(0=Open, 1=tied to LCOM)
7	BCD Speed 1	TB1-18	(0=Open, 1=tied to LCOM)
8	Reverse	TB1-19	(0=Open, 1=tied to LCOM)
9	Remote	TB1-20	(0=Open, 1=tied to LCOM)
10	-	-	-
11	Run Enable	TB1-22	(1=Open, 0=tied to LCOM)
12	Start	TB1-23	(0=Open, 1=tied to LCOM)

ตารางที่ 4.20 แสดงค่าพารามิเตอร์ของคำสั่ง "INT"

LAST FAULT P516

เป็นการแสดงสาเหตุการเกิดฟอลต์ครั้งสุดท้ายมีความหมายตามตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.21 แสดงความหมายของการเฉิดฟอล์ต

As Viewed	Description	Terminal	Polarity
POC	PEAK I LIMIT	ILL	I LOOP LOSS
AOT	AMBIENT OVERTEMP	OVR	OVERVOLTS REGEN
HSOT	HEATSINK TEMP	OVD	OVERVOLTAGE TRIP
IOC	IOC TRIP	EXDB	EXCESSIVE DB
PSF	LOGIC PS FAULT	MOMA	FBK MARKER FAULT
EXT	EXT FAULT TRIP	LEM	HW FAULT XXX
WDT	WATCHDOG TIMER	ZERO	NO SPEED SELECT
PGC	PC CARD FAULT	MOUT	MEMORY OUT
CRC	MEMORY CRC	UV	UNDERVOLT TRIP
IXTT	IXT CURRENT TRIP	SERR	SERIAL RECEIVE
RUN	MOTOR RUNAWAY	DIR	FWD REV
NOMO	MOTOR NOT WIRED		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAULT2 P517

เป็นการแสดงสาเหตุการเกิด ฟอล์ต ก่อน "LAST FAULT" (P516) ดูความหมาย จากตารางที่ 4.21

FAULT3 P518

เป็นการแสดงสาเหตุการเกิด ฟอล์ต ก่อน "FAULT2" (P517) ดูความหมายจาก ตารางที่ 4.21

FLT VOLTS P519

เป็นการแสดงค่า แรงดันที่ bus ขณะเกิด last fault trip จากสาเหตุตาม ตารางที่ 4.21

FLT FREQ P520

เป็นการแสดงค่าความถี่ขณะเกิด last fault trip

F MODE P521

เป็นการแสดงค่าการทำงานขณะเกิด last fault trip (เช่น "accel fwd" , "running" เป็นต้น)

FLT CUR P522

เป็นการแสดงค่ากระแสขณะเกิด last fault trip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELT TIME P523

เป็นการแสดงค่าเวลา (แสดงเป็นจำนวนชั่วโมงตามคำสั่ง "HOURS RUN" (P514) ขณะเกิด last fault trip

SW P524

แสดงรุ่นของ software ที่ควบคุม

DRIVE SIZE P525

แสดงแรงม้าของตัว เครื่อง ค่านี้จะเปลี่ยนแปลงขึ้นกับ "MTR AMPS" (P300) ถ้าค่า "MTR AMPS" ≤ ค่าของพิกัด constant—torque ของ เครื่อง จะแสดงผลเป็นพิกัด constant—torque ถ้า "MTR AMPS" > ค่าพิกัด constant—torque ของเครื่อง จะแสดงผลเป็นพิกัด variable - torque

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 อธิบายคำสั่งในหมวดที่ 6

AC TUNE P600

เป็นคำสั่งใช้ในการปรับค่าสำหรับ พารามิเตอร์ ให้เหมาะสมกับ สภาพการทำงาน ของอุปกรณ์ต่างๆ ที่ต่อเข้ากับเครื่อง

SYS P601

เป็นความถี่ธรรมชาติของระบบ ค่านี้จะใช้ในการคำนวณ การตอบสนองของ มอเตอร์ ค่าปกติจะอยู่ระหว่าง 2 ถึง 60 ค่ายิ่งสูง การตอบสนองจะเพิ่มขึ้น และการเปลี่ยนแปลงค่าจะทำให้ค่า ระบบควบคุมพีไอดี (PID controlcr) , "KV CMD" (P610) , "KP FBK" (P611) และ "KD FBK" (P612) เปลี่ยนแปลงโดยอัตโนมัติ และค่า "KA CMD" (P609) จะเป็น 0

CAR FREQ P603

เป็นค่าความถี่สวิตซ์ชิง (switching frequency) ใช้ในกรณีที่ต้องการให้การทำงาน เรียบ ลดเสียงรบกวน โดยการเพิ่มค่าความถี่สวิตซ์ชิง

MOTOR PPQ P604

ใช้ในกรณีที่ต้องการต่ออุปกรณ์แปลงสัญญาณ เช่น พัลส์เจนเนเรเตอร์ (Pulse generater) โดยการตั้งค่าเป็นขนาดของลูกคลื่นพัลส์ของเจนเนเรเตอร์ (Pulse) ต่อการหมุน 1 รอบ ถ้าต่อคับพัลส์เจนเนเรเตอร์ แล้วไม่มีสัญญาณพัลส์ ออกมาจะทำให้เกิดฟอล์ต หน้าจอจะขึ้น "PG CARD FAULT"
ถ้าไม่มีการต่ออุปกรณ์แปลงสัญญาณจะให้ค่านี้เป็น 0

STATOR R P605

ค่าความต้นทานของสเตเตอร์ (Stator) คำนวณโดยใช้ Ac tune test สำหรับทำงาน เมื่อไม่มีตัวถอดระหัส (encoder)

I FIELD P606

ค่ากระแสกระตุ้น (Field current) คำนวณโดยใช้ Ac tune test ถ้าใส่ค่าอื่นลงไป ควรจะทำการ Ac tune test เพื่อให้ค่า สลิป (slip) และค่าเกนระบบ (system gain) เป็นค่าที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SLIP FREQ P607

ค่าความถี่สลิป (slip frequency) คำนวณโดย Ac tune test ถ้าใส่ค่าอื่นจะต้องทำการ Ac tune test ใหม่เพื่อให้ค่าเกนระบบเหมาะสม

LEAKAGE P608

ค่าความเหนี่ยวนำรั่วไหลของมอเตอร์ (Motor leakage inductance) คำนวณจาก Ac tune test และไม่สามารถเปลี่ยนเองได้

KA CMD P609

ค่าอัตราเร่งของคำสั่งปรับค่าเกน (gain) ในระบบควบคุมพีไอดี ค่าเกนจะตั้งเป็น 0 และต้องใส่ทุกครั้งที่ทำการ Ac tune test หรือ เมื่อคำสั่ง "SYS WN" (P601) หรือ "SYS GAIN" (P602) เปลี่ยนแปลง

KV CMD P610

ค่าอัตราเร็วของคำสั่งปรับค่าเกนในระบบควบคุมพีไอดี ค่าดังกล่าวคำนวณโดย Ac tune test และจะเปลี่ยนแปลงทุกครั้งเมื่อคำสั่ง "SYS WN" (P601) หรือ "SYS GAIN" (P602) มีค่าเปลี่ยนไป

KP FBK P611

ค่าการป้อนกลับของอัตราเร็ว (Proportional feedback) ของเกนในระบบควบคุมพีไอดี คำนวณโดย Ac tune test และจะเปลี่ยนแปลงทุกครั้งเมื่อคำสั่ง "SYS WN" (P601) หรือ "SYS GAIN" (P602) มีค่าเปลี่ยนไป

KD FBK P612

ค่าการป้อนกลับของอัตราเร่ง (derivative feedback) ของเกนในระบบควบคุมพีไอดี คำนวณโดย Ac tune test และจะเปลี่ยนแปลงทุกครั้งเมื่อคำสั่ง "SYS WN" (P601) หรือ "SYS GAIN" (P602) มีค่าเปลี่ยนไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INVERT P613

เป็นค่าใช้ในการเปลี่ยนสัญญาณพัลส์ของเฟสเจนเนอเรเตอร์ และทิศทางการหมุนของมอเตอร์ ค่านี้จะคำนวณอัตโนมัติโดย Ac tune test ถ้าติดพัลส์เจนเนอรเตอร์ ถ้าไม่ค่า "INVERT" จะสามารถตั้งเพื่อกลับทิศการหมุนของโรเตอร์ (rotor) เองได้ ความหมายของคำสั่งเป็นดังนี้

PGN MTN 0 พัลส์เจนเนเรเตอร์ ไม่กลับทิศ / มอเตอร์ไม่กลับทิศ

PGN MTI 1 พัลส์เจนเนเรเตอร์ ไม่กลับทิศ / มอเตอร์กลับทิศ

PGI MTN2 พัลส์เจนเนเรเตอร์ กลับทิศ / มอเตอร์ไม่กลับทิศ

PGI MTI 3 พัลส์เจนเนเรเตอร์ กลับทิศ / มอเตอร์กลับทิศ

สำหรับการปรับทิศการหมุนของมอเตอร์เอง (โดยการใช้พัลส์เจนเนเรเตอร์)

ถ้า "INVERT" (P613) เป็น	ให้เปลี่ยนเป็น
PGN MTN 0	PGI MTI 3
PGI MTI 3	PGN MTN 0
PGN MTI 1	PGI MTN 2
PGI MTN2	PGN MTI 1

ตารางที่ 4.22 แสดงการเปลี่ยนพารามิเตอร์ของคำสั่ง "INVERT"

สำหรับการปรับเฟสพัลส์เจนเนเรเตอร์ และตั้งทิศทางของโรเตอร์ถ้าเฟสพัลส์เจนเนเรเตอร์ผิด ตัวเครื่อง จะหมุนช้าผิดปกติ และกินกระแสเต็มพิกัด จะต้องทำการปรับค่าให้ถูกต้องโดยสังเกตุ

(1) ดูทิศการหมุน ถ้าหมุนแล้วช้าผิดปกติให้รีบปิดเครื่องทันที

(2) ถ้าเครื่องหมุนถูกต้องแล้วให้กลังทิศของพัลส์เจนเนเรเตอร์เท่านั้น ตามตาราง

ตารางที่ 4.23 แสดงการเปลี่ยนพารามิเตอร์ของคำสั่ง "INVERT"

ถ้า "INVERT" (P613) เป็น	ให้เปลี่ยนเป็น
PGN MTN 0	PGI MTN 2
PGI MTN 2	PGN MTN 0
PGN MTI 1	PGI MTI 3
PGI MTI 3	PGN MTI 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าตัวเครื่องหมุนผิดทิศ ให้กลับทิศการหมุนมอเตอร์เท่านั้น ตามตาราง

ตารางที่ 4.24 แสดงการเปลี่ยนพารามิเตอร์ของคำสั่ง "INVERT"

ถ้า "INVERT" (P613) เป็น	ให้เปลี่ยนเป็น
PGN MTN 0	PGN MTI 1
PGN MTI 1	PGN MTN 0
PGI MTN 2	PGI MTI 3
PGI MTI 3	PGI MTN 2

SET T P614

จะใช้ค่านี้เมื่อ คำสั่ง "STOP MODE" (P202) เป็น DECEL ในโหมดสลาฟเป็นการตั้งค่าหน่วงเวลาหลังจากเครื่องถูกสั่งให้หยุด และก่อนที่จะปิดเครื่อง

H MASK P615

S MASK P616

"HMASK" และ "S MASK" เป็นการตั้งให้สามารถแสดงสาเหตุของความผิดพลาดของฮาร์ดแวร์ (Hardware) มีความหมายดังนี้

Mask Constant	Fault	Comments
0	-	ไม่ใช้
1	PEAK I LIMTI (POC)	
2	AMBIENT OVERTEMP (AOT)	
4	HEATSINK TEMP (HSOT)	
8	IOC TRIP (IOC)	รวมอยู่ในทุกคำสั่ง
16	LOGIC PS FAULT (PSF)	
32	EXT FAULT TRIP (EXT)	
64	WATCHDOG TIMER (WDT)	รวมอยู่ในทุกคำสั่ง
256	PG CARD FAULT (PGC)	

ตารางที่ 4.25 แสดงค่าพารามิเตอร์ของคำสั่ง "H MASK"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าตั้งค่าเป็น 383 จะเป็นการสั่งให้แสดงสาเหตุของการเกิดฟอลต์ในฮาร์ดแวร์ทั้งหมด

ค่าดีฟอล์ตของผู้ผลิตเป็นค่าที่เหมาะสมในการใช้งานตามปกติ

"S MASK" เป็นการตั้งให้สามารถแสดงสาเหตุของความผิดพลาดของละมุนภัณฑ์ (Software) มีความหมายดังนี้

Mask Constant	Fault	Comments
1	MEMORY CRC (CRC)	รวมอยู่ในทุกคำสั่ง
2	IXT CURRENT TRIP (IXTT)	
4	-	ไม่ใช้
8	MOTOR RUNAWAY (RUN)	
16	MOTOR NOT WIRED (NOMO)	
32	I LOOP LOSS (ILL)	
64	OVERVOLTS REGEN (OVR)	รวมอยู่ในทุกคำสั่ง
128	OVERVOLTAGE TRIP (OVD)	รวมอยู่ในทุกคำสั่ง
256	EXCESSIVE_DB (EXDB)	
512	FBK MARKER FAULT (MOMA)	
1024	HW FAULT XXX (LEM)	
2048	NO SPEED SELECT (ZER0)	
4096	MEMORY OUT (MOUT)	รวมอยู่ในทุกคำสั่ง
8192	UNDERVOLT TRIP (UV)	รวมอยู่ในทุกคำสั่ง
16384	SERIAL RECEIVE (SERR)	
32768	FWD REV (DIR)	

ตารางที่ 4.26 แสดงค่าพารามิเตอร์ของคำสั่ง "S MASK"

ถ้าตั้งค่าเป็น 64751 เป็นการสั่งให้แสดงสาเหตุของการเกิดฟอล์ตของ ซอฟต์แวร์ทั้งหมดยกเว้น "MOTOR NOT WIRED" , "EXCESSIVE DB" และ "FBK MARKER FAULT"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LOWPASS FREQ P617

NOTCH FREQ P618

OPEN LOOP P619

DZ STATE P620

คำสั่งทั้งหมดนี้สำหรับผู้ผลิตเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและวิจารณ์

เตรียมอุปกรณ์ การทดลอง

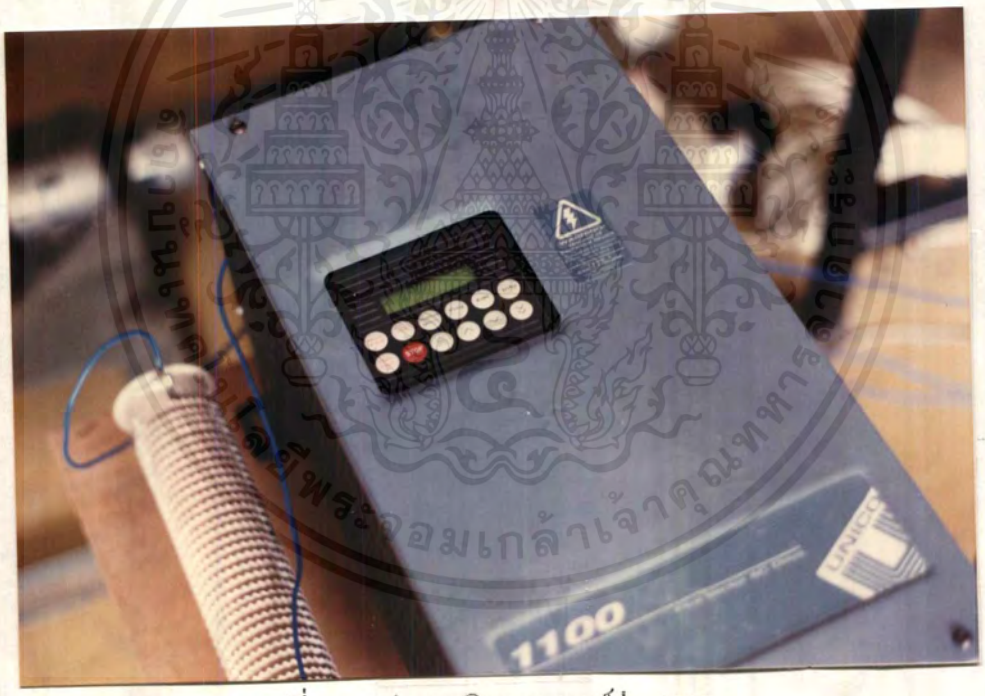
1. เครื่องคอมพิวเตอร์ เพนเทียม166MHz RAM 32 ระบบปฏิบัติการ WINDOWS95
2. ใช้โปรแกรมภาษา C++ ของ Borland version 3
3. อินเวอตเตอร์ แบบดิจิตอลเอซีฟิลด์โอเรียลเตชั่นฟลักซ์เวกเตอร์ไดรฟ์ ของ UNICO รุ่น 1100

   อินพุท : 0-460 Vac , 50-60 Hz , 3 เฟส (ปกติใช้ 460V 60 Hz)

   เอาท์พุท : 0-460 Vac , 20 แรงม้าที่วาริเอเบิลทอร์ก (variable-torque)

   กินกระแส 30 A , 15 แรงม้าที่คอนสแตนท์ทอร์ก (constant-torque)

   กินกระแส 24 A (โอเวอร์โหลดได้ 1 นาทีที่กระแส 36 A)

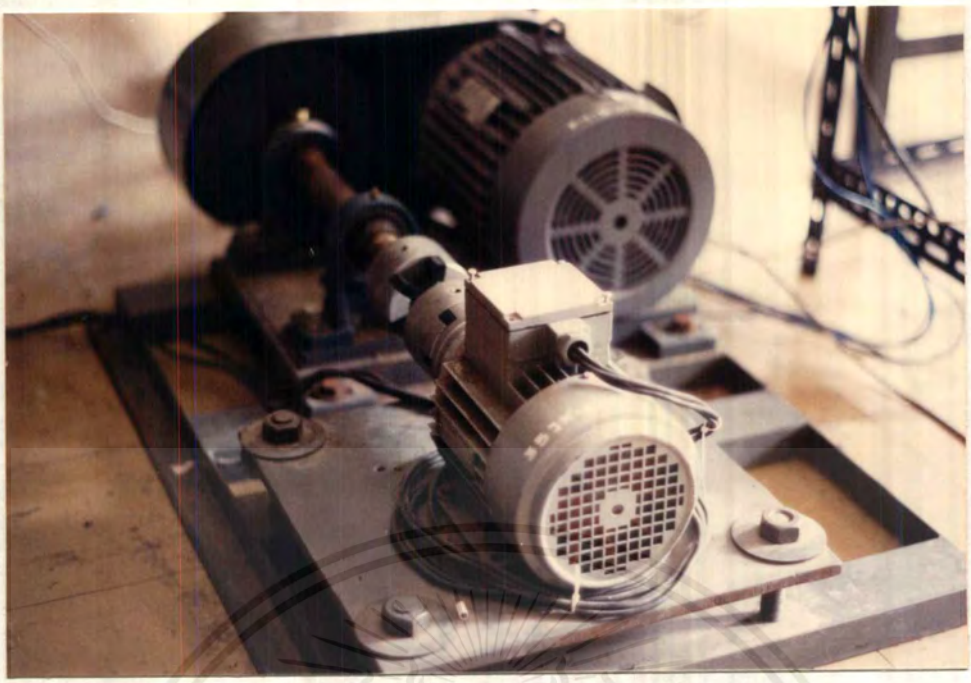


รูปที่ 5.1 รูปแสดงอินเวอตเตอร์รุ่น 1100

4. มอเตอร์ อินดักชั่นมอเตอร์ขนาดเอาท์พุท 37 kW 4 pole ยี่ห้อ HITACHI

380 V/ 50Hz	400V/ 50Hz	415V /50Hz	400V/ 60Hz	440V/ 60Hz
1440rpm	1440rpm	1440rpm	1720rpm	1730rpm
7.5A	7.2A	7.2A	7A	6.5A

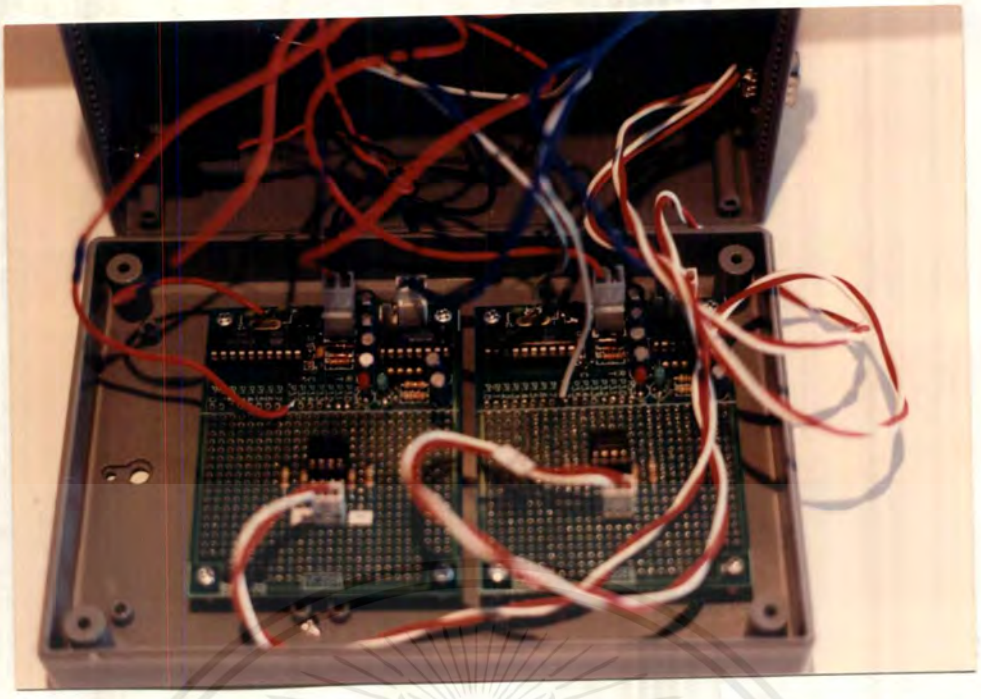
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 รูปอินดักชั่นมอเตอร์

5. อุปกรณ์แปลงสัญญาณการรับส่งข้อมูลจาก RS-232 เป็น RS-485 และ RS-485 เป็น RS-232 (RS-232 To RS-485 Converter) โดยใช้อุปกรณ์ต่างๆ ดังนี้
   - CPU 89C2051 Clock 11.0562Hz หน่วยความจำ Kbyte พอร์ต 15 บิท I/O on CPU
   - คอนเนคเตอร์ (connector) 3 pin RS-232 ใช้ชิป์ (chip) MAX232
   - คอนเนคเตอร์ 2 pin RS-485 ใช้ชิป์ 75176
   - ใช้เพาเวอร์ซัพพลาย (power supply) 5Vdc 20mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



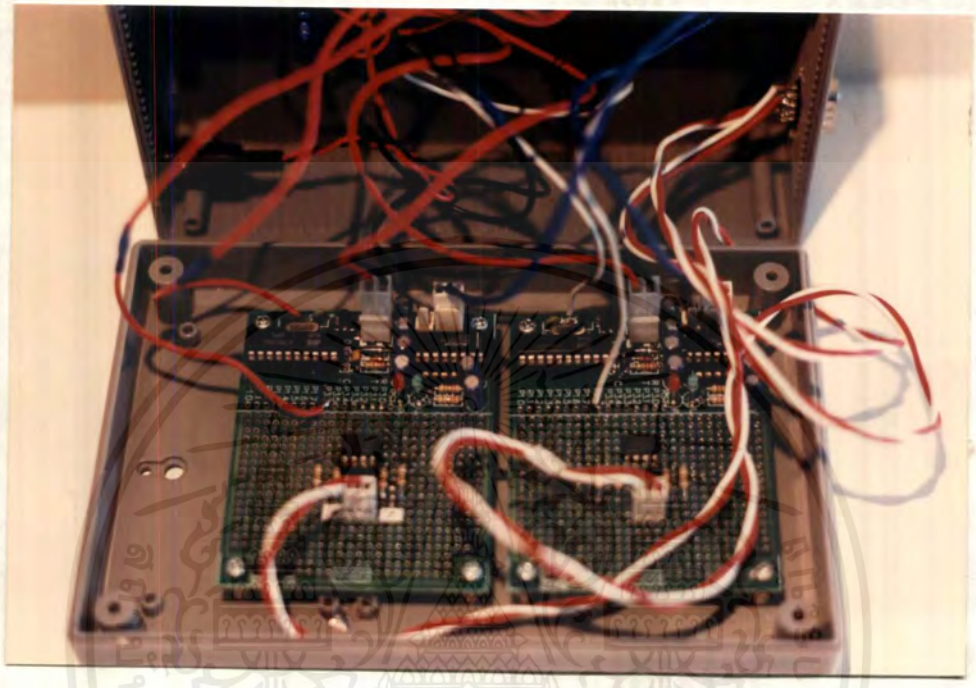
รูปที่ 5.3 รูปแสดงชิ้นส่วนภายในของอุปกรณ์แปลงสัญญาณ

การติดตั้งอุปกรณ์เพื่อทดลองโปรแกรม

1. ทำสายเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ กับชุดรับของอุปกรณ์แปลงสัญญาณ (แปลง RS-232 เป็น RS-485) โดยพอร์ตของคอมพิวเตอร์ที่ใช้เป็น RS-232 หัวต่อแบบ DB-9 โดยใช้สายสัญญาณทั้งหมด 2 เส้นคือเส้นส่งสัญญาณ (TXD) ใช้ขา 3 และเส้นซิกแนลกราวด์ (Signal ground) ใช้ขา 5 โดยมีการหลอกสัญญาณ Handshaking โดยเชื่อม DTR (Data Terminal Ready) กับ DSR (Data set Ready) และ RQS (Request To Send) เชื่อมกับ CTS (Clear To Send) เพื่อให้พร้อมรับส่งข้อมูลตลอดเวลา หลังจากนั้นนำเส้นส่งสัญญาณจากพอร์ต DB-9 ของคอมพิวเตอร์มาต่อเข้ากับขารับสัญญาณของชุดรับของอุปกรณ์แปลงสัญญาณ และเชื่อมซิกแนลกราวด์เข้าด้วยกัน
ทำสายเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ กับชุดส่งของอุปกรณ์แปลงสัญญาณ (แปลง RS-485 เป็น RS-232) โดยพอร์ตของคอมพิวเตอร์ที่ใช้เป็น RS-232 หัวต่อแบบ DB-9 โดยใช้สายสัญญาณทั้งหมด 2 เส้นคือเส้นรับสัญญาณ (RXD) ใช้ขา 2 และเส้นซิกแนลกราวด์ (Signal ground) ใช้ขา 5 โดยมีการหลอกสัญญาณ Handshaking โดยเชื่อม DTR (Data Terminal Ready) กับ DSR (Data set Ready) และ RQS (Request To Send) เชื่อมกับ CTS (Clear To Send) เพื่อให้พร้อมรับส่งข้อมูลตลอดเวลา หลังจากนั้นนำเส้นรับสัญญาณจากพอร์ต DB-9 ของคอมพิวเตอร์มาต่อเข้ากับขาส่งสัญญาณของชุดส่งของอุปกรณ์แปลงสัญญาณ และเชื่อมซิกแนลกราวด์เข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนสายที่เชื่อมระหว่าง อุปกรณ์แปลงสัญญาณกับอินเวอตเตอร์มี 4 เส้น โดย 2 เส้น เป็นส่วนรับข้อมูล Rx+ และ Rx- ต่อจากพอร์ตรับสัญญาณ RS-485 ส่วน 2 เส้นที่เหลือ เป็นส่วนส่งข้อมูล Tx+ และ Tx- ต่อจากพอร์ตส่งสัญญาณ RS-485 ของอุปกรณ์แปลงสัญญาณ



รูปที่ 5.4 รูปแสดงการต่อสายภายในของอุปกรณ์แปลงสัญญาณ

2. การตั้งค่าอินเวอตเตอร์เพื่อใช้กับโปรแกรม ดาต้ามอนิเตอร์ (Data Monitor)
   - ตั้งค่าแอดเดรสของอินเวอตเตอร์ ตั้งไว้ตั้งแต่ 01 ถึง 09 (กรณีต่อกับอินเวอตเตอร์เพียงตัวเดียวตั้งค่าที่ 01)
   - ตั้งค่าอัตราบอด (Baud rate) ของอินเวอตเตอร์ที่ใช้ให้ตรงกับอัตราบอดของโปรแกรมควบคุมคือ 19200 bps
   - ตั้งค่าจำนวนบิทของข้อมูลเท่ากับ 8 บิทส่วนพาริตี้บิท (parity bit) ตั้งค่าที่ none parity

3. การตั้งมอเตอร์เพื่อใช้กับอินเวอตเตอร์รุ่น 1100
   - ตั้งกระแสที่พิกัดของมอเตอร์จากเนมเพลท (nameplate) เลือกคำสั่งที่ P300 ซึ่งสำหรับโครงงานนี้เราใช้มอเตอร์ขนาดพิกัดกระแส 7.5 A ดังนั้นจึงควรตั้งที่ 7.5 A
   - ตั้งแรงดันที่พิกัดของมอเตอร์จากเนมเพลท เลือกคำสั่งที่ P301 ซึ่งสำหรับโครงงานนี้เราใช้มอเตอร์ขนาดพิกัดแรงดัน 380V จึงควรตั้งที่ 380V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตั้งความถี่ใช้งานของจากเนมเพลท เลือกคำสั่งที่ P302 ซึ่งสำหรับโครงงานนี้เราใช้มอเตอร์ขนาดพิกัดความถี่ 50 Hz จึงควรตั้งที่ 50 Hz
- ตั้งพิกัดความเร็วของมอเตอร์ พิจารณาจากเนมเพลท เลือกคำสั่งที่ P303 ซึ่งสำหรับโครงงานนี้เราใช้มอเตอร์ขนาดพิกัดความเร็ว 1440 rpm
- ทำ Ac tune test โดยเลือกคำสั่ง P600 จะได้ค่าพารามิเตอร์ออกมาดังนี้

  ค่า Estimate stator resistance

  ค่า Field Current

  ค่า slip frequency

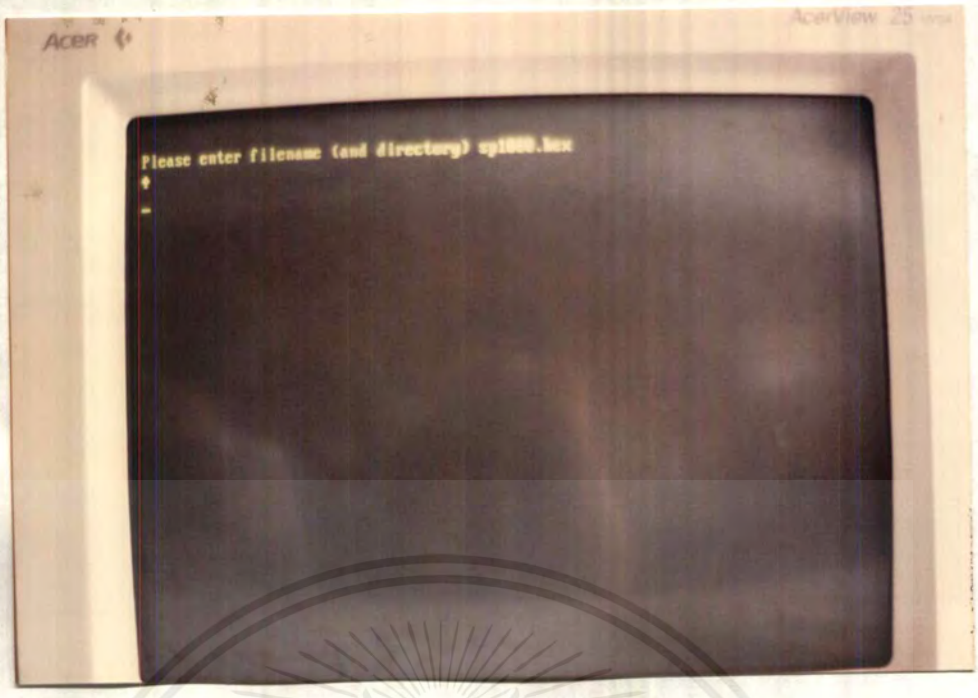
  ค่า system gain

ขั้นตอน และผลการทดลอง

1. ทดสอบ RS-232 converter (อุปกรณ์แปลงสัญญาณ) โดยใช้โปรแกรมเทอร์มินอล (Terminal) ของระบบปฏิบัติการวินโดวส์ 3.11 (เนื่องจากโปรแกรมนี้เป็นโปรแกรมพื้นฐานในการส่งข้อมูลออกพอร์ต) และทำการทดลองส่งคำสั่งตั้งค่าความเร็วให้อินเวอตเตอร์ (ใช้คำสั่ง 000 โดยเก็บเป็นไฟล์ในรูปแบบระหัสแอสกี้) เพื่อเช็คดูว่าอุปกรณ์แปลงสัญญาณได้เชื่อมต่ออย่างถูกต้องแล้วหรือไม่ ถ้าหลังจากส่งคำสั่งไปแล้วตัวอินเวอตเตอร์ตอบสนอง มอเตอร์จะหมุนด้วยความเร็วตามที่ตั้งไว้

2. เขียนโปรแกรมในส่วนของการรับส่งข้อมูลแบบอนุกรมออกพอร์ต RS-232 โดยทดลองส่งคำสั่งตั้งค่าความเร็วโดยใช้คำสั่ง 000 เช่นเดียวกับตอนทดลองส่งข้อมูลออก เทอร์มินอลเพื่อทดสอบว่าส่วนของโปรแกรมรับส่งข้อมูลแบบอนุกรมที่เขียนขึ้นสามารถทำงานได้เช่นเดียวกับโปรแกรมเทอร์มินอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 รูปแสดงผลของโปรแกรมในการส่งข้อมูล

3. ทดลองในส่วนของกราฟฟิกโหมด (Graphic mode) โดยเปลี่ยนจากเท็กซ์โหมด (text mode) เป็นกราฟฟิกโหมดโปรแกรมจะแสดงจุดขึ้นบนจอภาพ กรณีที่โหมดเปลี่ยนเป็นกราฟฟิกโหมดแล้วจะปรากฏจุดที่สั่งให้แสดงผล แต่ถ้าเป็นเท็กซ์โหมด จะไม่ปรากฏจุดให้เห็นในหน้าจอ

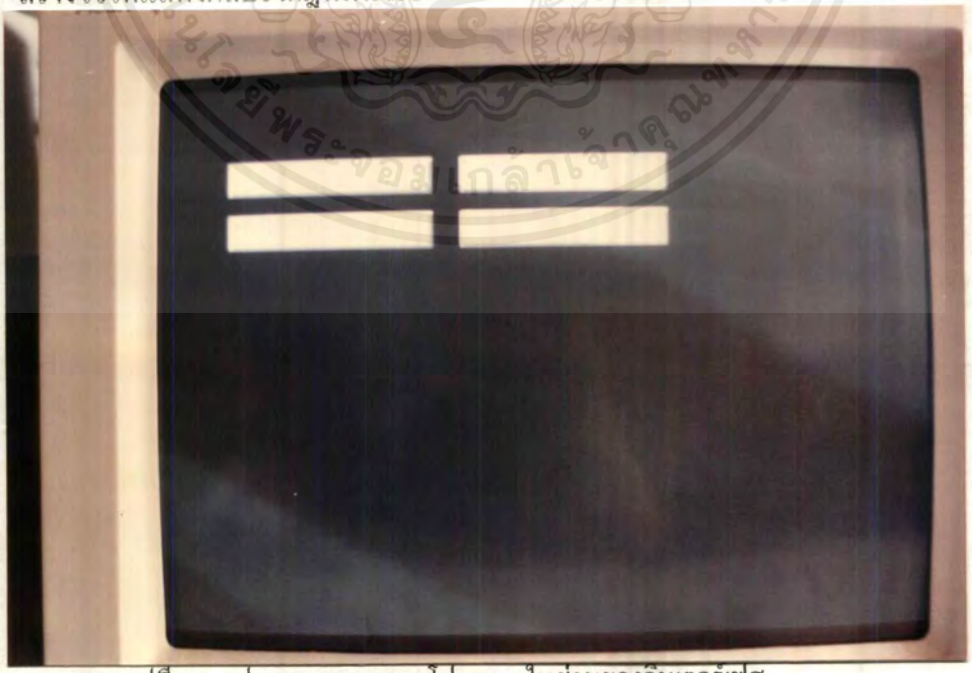
รูปที่ 5.6 รูปแสดงการทดสอบโปรแกรมในส่วนของกราฟฟิกโหมดเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทดสอบส่วนโปรแกรมแสดงผลในโหมดกราฟฟิก เนื่องจากในโหมดนี้ตัวอักษรสามารถแสดงเป็นสีได้ และคำสั่งที่ใช้ในเท็ กซ์โหมดไม่สามารถใช้ในกราฟฟิกได้

รูปที่ 5.7 รูปแสดงผลการทดสอบสีของตัวอักษรในเท็กซ์โหมด

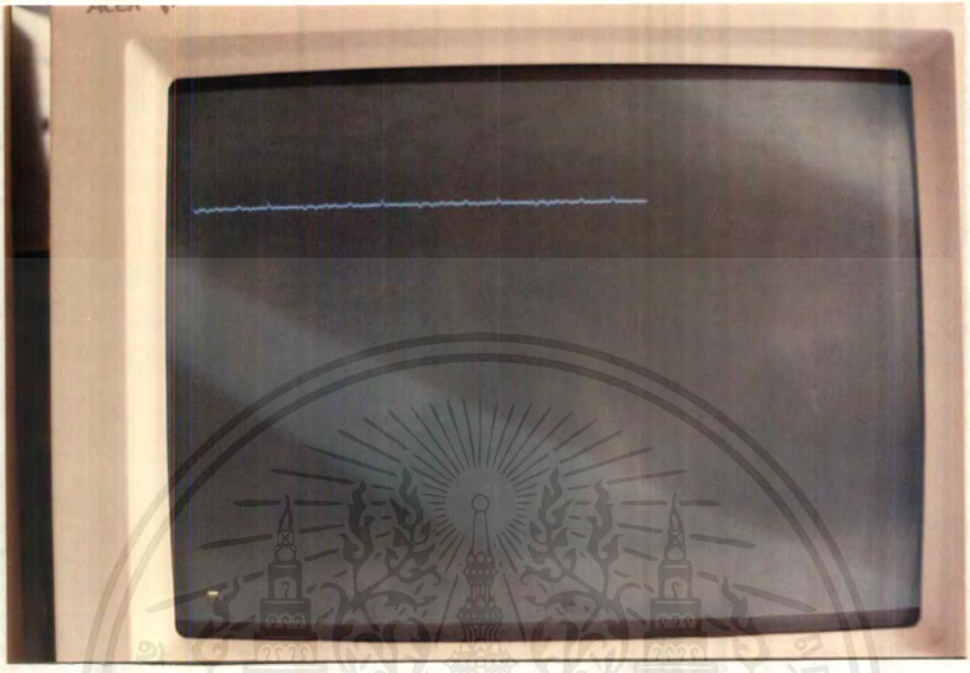
5. ทดสอบโปรแกรมเปิดไฟล์ภาพเพื่อใช้ในส่วนอินเตอร์เฟส (interface) เนื่องจากปุ่มต่างๆ และพื้นหลังของโปรแกรมจะเป็นไฟล์ภาพ โดยโปรแกรมนี้จะเปิดไฟล์ภาพที่สร้างไว้ให้แสดงผลบ่ รากฏที่หน้าจอ



รูปที่ 5.8 รูปแสดงการทดสอบโปรแกรมในส่วนของอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

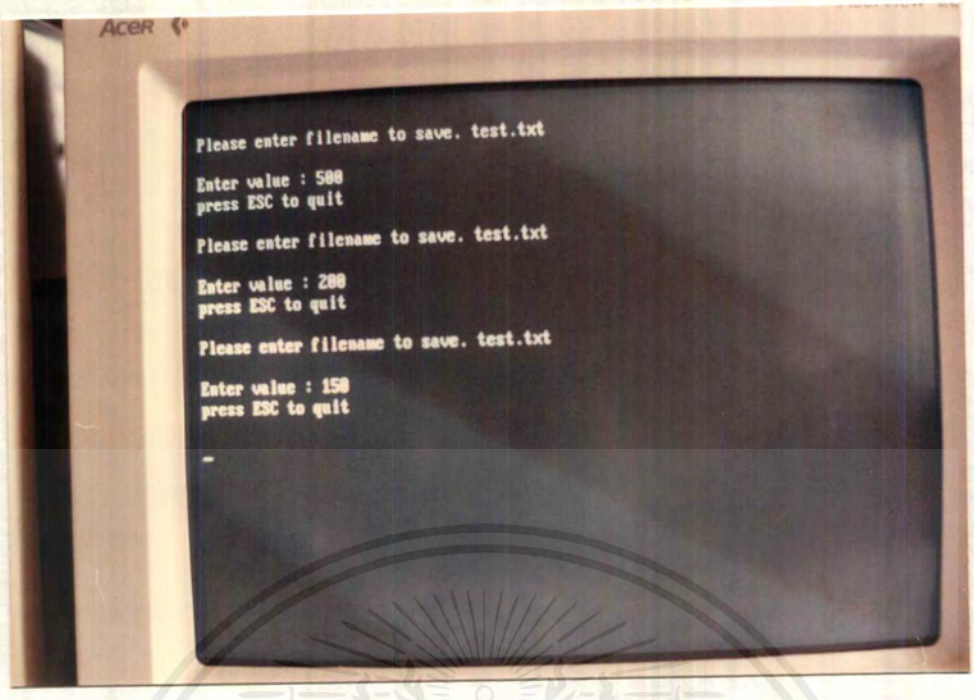
6. ทดลองรับข้อมูลจากคีย์บอร์ด (kcyboard) แล้วนำมาพล็อต (plot) กราฟโดยใช้หลักที่ว่า ข้อมูล 1 ค่าจะได้จุด 1 จุด โดยค่าต่างๆ ที่รับมาจะถูกพล็อตไปเรื่อยๆ โดยตำแหน่งจะเลื่อนไปทางซ้าย 1 พิกเซล (pixel) เมื่อมีค่าใหม่เพิ่มเข้ามาอีก 1 ค่า



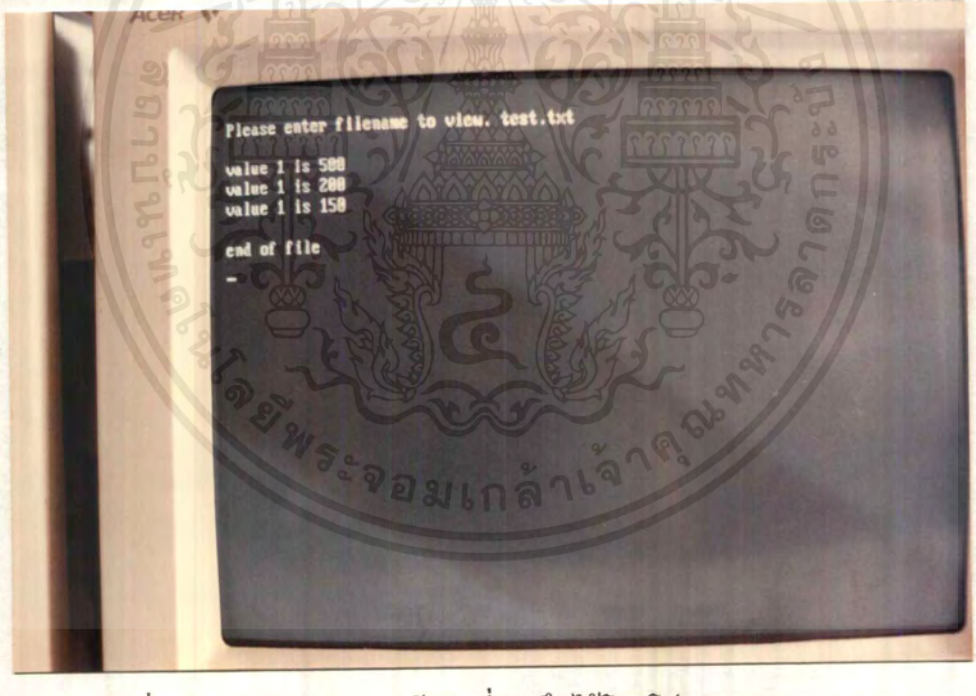
รูปที่ 5.9 รูปแสดงการทดสอบการรับข้อมูลจาก Keyboard

7. ทดสอบในส่วนของการเก็บข้อมูลเป็นไฟล์ลงฮาร์ดดิส (Hardisk) โดยให้โปรแกรมอ่านค่าข้อมูลจากการพิมพ์ตัวเลขด้วยคีย์บอร์ด ตรวจสอบว่าโปรแกรมสามารถอ่านค่าข้อมูลนั้นแล้วนำมาบันทึกเป็นไฟล์ข้อมูลเก็บไว้ในฮาร์ดดิส และสามารถเรียกไฟล์ข้อมูลดังกล่าวออกมาแสดงผลได้ตรงกับค่าที่ปรากฏที่หน้าจอหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



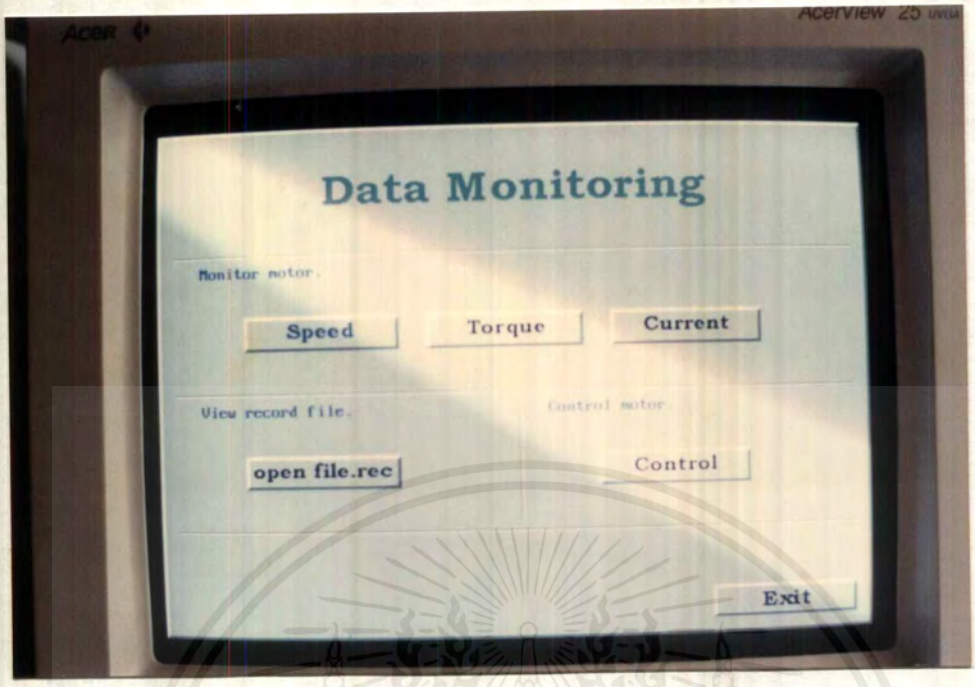
รูปที่ 5.10 รูปแสดงการเก็บข้อมูลลงฮาร์ดดิสของโปรแกรม



รูปที่ 5.11 รูปแสดงรายการข้อมูลที่บันทึกไว้โดยโปรแกรม

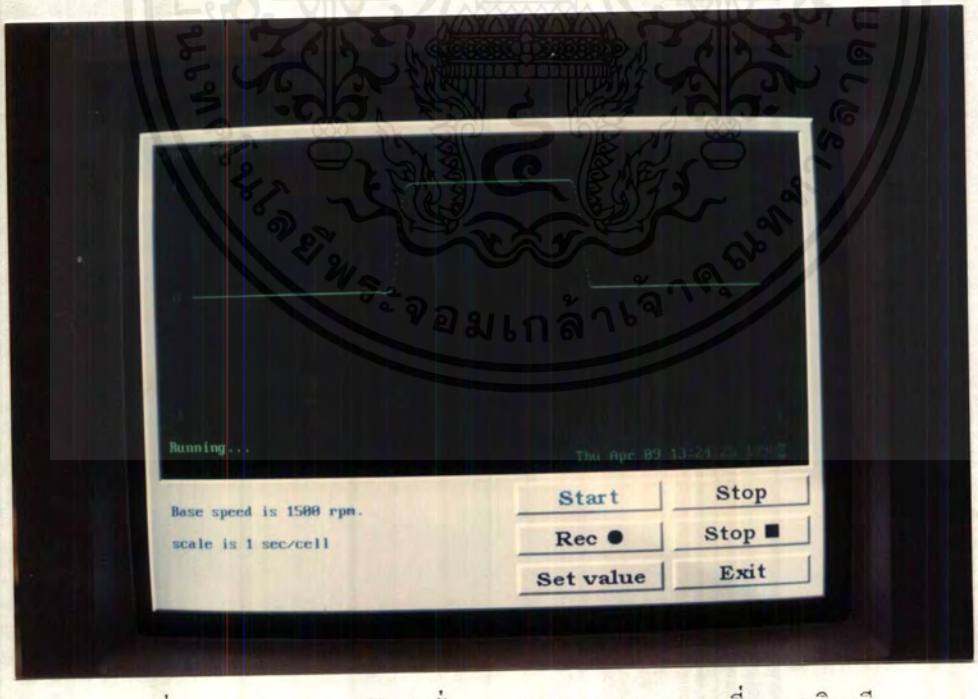
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. นำโปรแกรมย่อยแต่ละส่วนข้างต้นมารวมกันเป็นโปรแกรมที่ใช้งาน



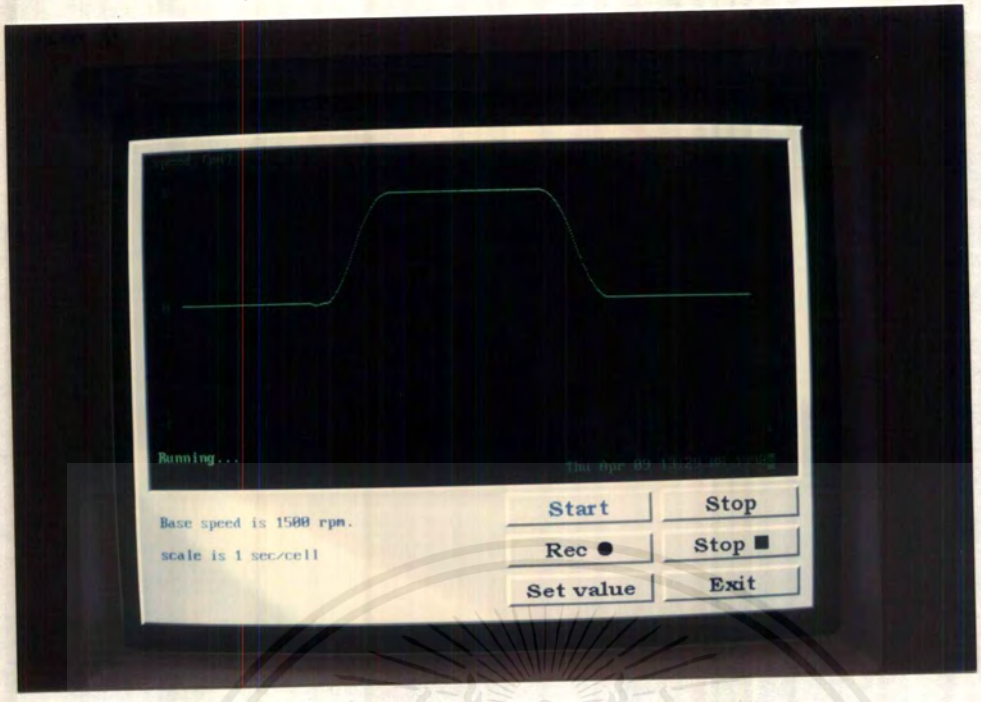
รูปที่ 5.12 รูปแสดงการรวมโปรแกรมส่วนหลัก

9. นำโปรแกรมที่ใช้งานมามอนิเตอร์เพื่อปรับค่าคำสั่งพารามิเตอร์ต่างๆ

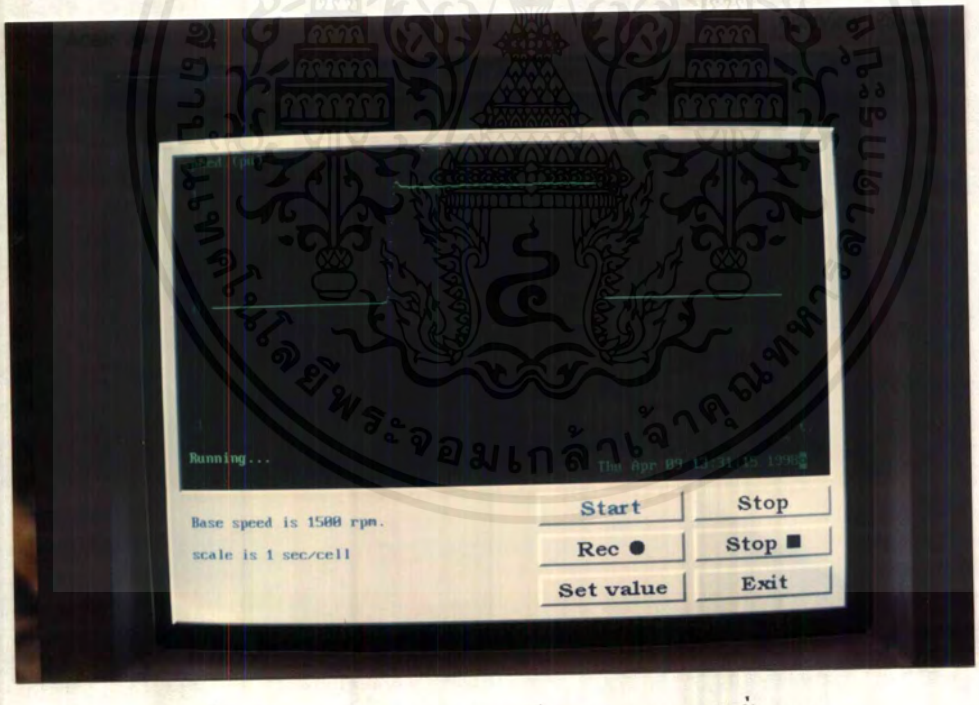


รูปที่ 5.13 รูปแสดงการปรับคำสั่ง "ACCEL" และ "DECEL" ที่เวลา 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

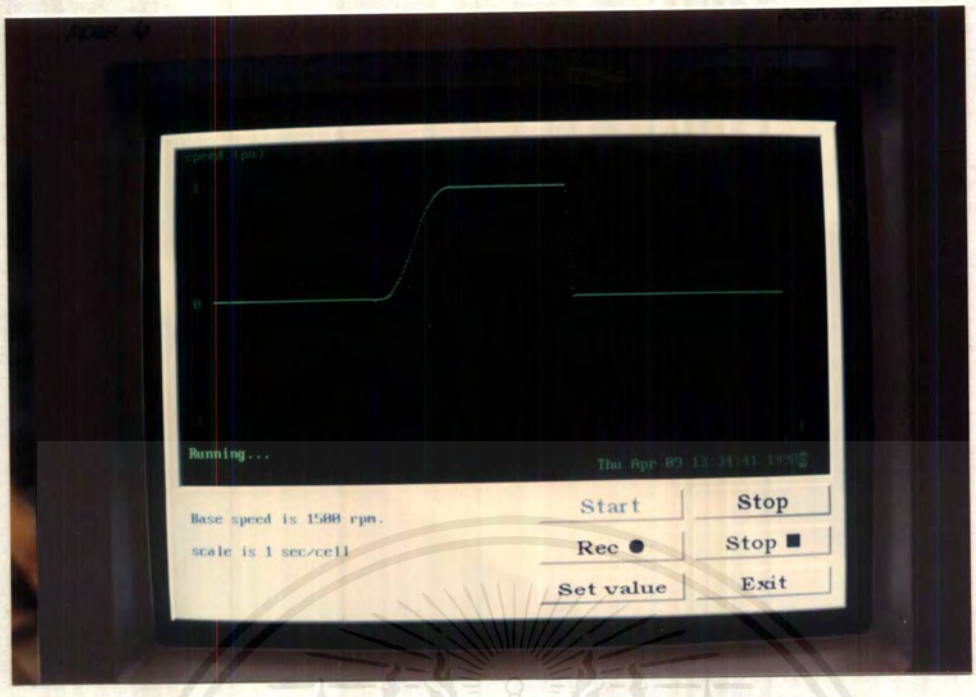


รูปที่ 5.14 รูปแสดงการปรับคำสั่ง "ACCEL" และ "DECEL" ที่เวลา 4 วินาที

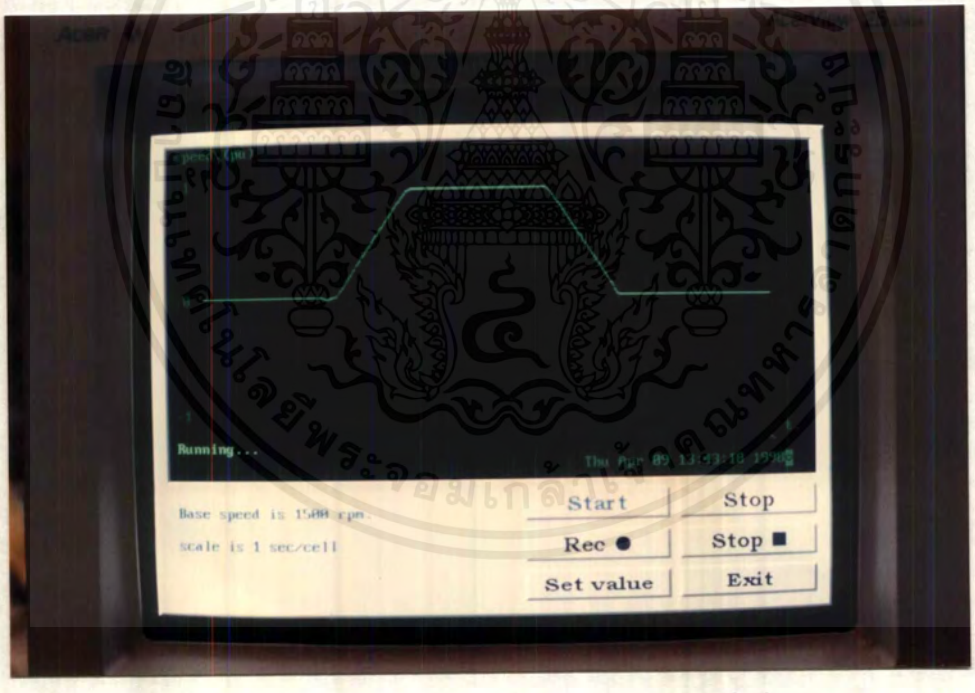


รูปที่ 5.15 รูปแสดงการมอนิเตอร์ "TORQ" ตามคำสั่ง P201

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



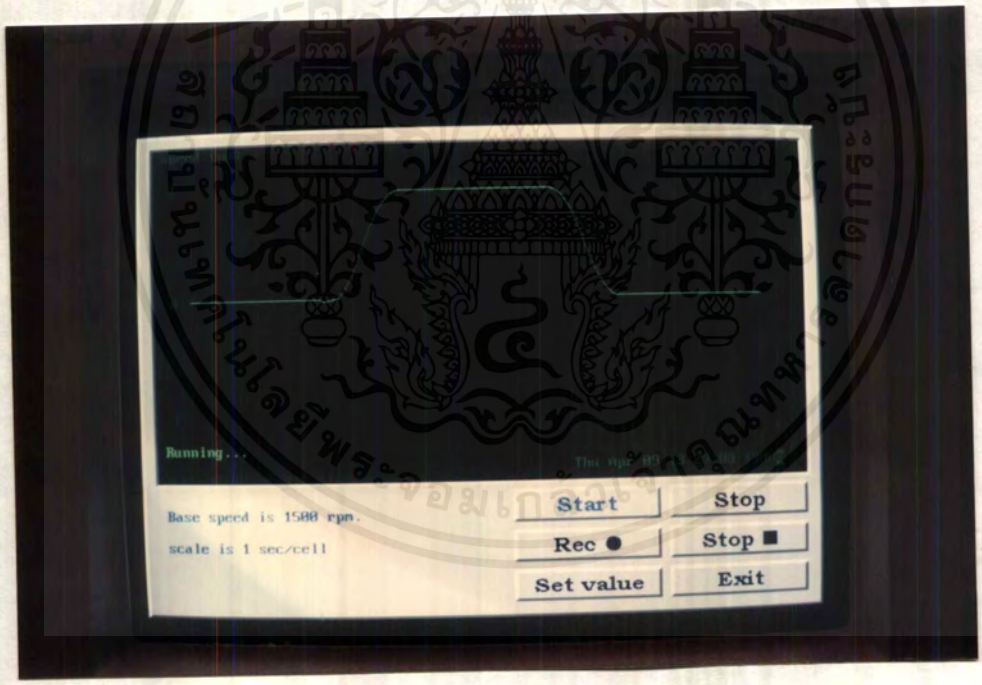
รูปที่ 5.16 รูปแสดงผลการมอนิเตอร์ “COAST” ตามคำสั่งที่ P202



รูปที่ 5.17 รูปแสดงการมอนิเตอร์ “DC INJ” ตามคำสั่งที่ P202

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.18 รูปแสดงการมอนิเตอร์ "DC HOLD" ตามคำสั่งที่ P202



รูปที่ 5.19 รูปแสดงผลการมอนิเตอร์คำสั่ง P212 ที่ 0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.20 รูปแสดงผลการมอนิเตอร์คำสั่ง P212 ที่ 100%

<u>สรุป และวิจารณ์ผลการทดลอง</u>

1. อุปกรณ์แต่ละอย่างอาจจะมีรูปแบบการส่งข้อมูลแตกต่างกันไปตามชนิด และรุ่นของอุปกรณ์นั้นๆ กล่าวคืออุปกรณ์ที่ใช้ระยะทางในการควบคุมไม่ไกลนัก และต้องการการรับส่งข้อมูลที่ความเร็วสูงมักใช้การส่งข้อมูลแบบขนาน แต่อุปกรณ์ที่ต้องใช้ระยะทางในการควบคุมไกลๆ และความเร็วในการส่งข้อมูลไม่จำเป็นต้องสูงนัก จะใช้การส่งข้อมูลแบบอนุกรมซึ่งในการส่งข้อมูลแบบอนุกรมนั้นยังมีอีกหลายรูปแบบเนื่องจากการพัฒนาให้ได้ระยะทาง และความเร็วในการส่งข้อมูลสูงขึ้น
2. ขณะโปรแกรมมหกำลังทำงาน หากมีการกดคีย์บอร์ดคอมพิวเตอร์ หรือคีย์แพดของอินเวอตเตอร์บ่อยๆ จะทำให้กราฟข้อมูลกับเวลาคลาดเคลื่อนไปบ้าง เนื่องจากจะทำให้เกิดสัญญาณอินเทอร์รัพไปยังหน่วยประมวลผล โดยส่วนของการประมวลผลจะประมวลผลว่าปุ่มที่กดมีผลต่อการทำงานของอุปกรณ์หรือไม่ ถ้ามีอุปกรณ์จะทำตามคำสั่งนั้น แต่ถ้าไม่มีผลจะกลับมาทำงานต่อ ซึ่งจะเห็นว่าหน่วยประมวลผลต้องใช้เวลาในการประมวลผล ส่งผลให้เวลาที่ใช้ในการอ่านข้อมูล ใช้เวลานานขึ้นกว่าปกติ ทำให้กราฟข้อมูลกับเวลาที่แสดงผลคลาดเคลื่อนไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนา

สามารถแสดงผลข้อมูล และบันทึกข้อมูลหลายๆ ส่วน (เช่น กราฟของทอร์ก , สปีด , กระแส) พร้อมกัน

สามารถใช้เครื่องควบคุมเครื่องเดียวในการควบคุม และแสดงผลการทำงานของอินเวอตเตอร์หลายๆ ตัวพร้อมๆ กัน

สามารถใช้ควบคุมการทำงานอินเวอตเตอร์หลายๆ แบบซึ่งมีพารามิเตอร์ และมีรูปแบบในการรับส่งข้อมูลแตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ และโครงงานทั้งหมดนี้สำเร็จได้ เพราะมีอาจารย์ที่ปรึกษา เพื่อนๆ คอยช่วยเหลือด้วยดีตลอดจึงขอขอบคุณในความช่วยเหลือ และขอให้ความดีทั้งหลายในวิทยานิพนธ์เล่มนี้จงบังเกิดสุขให้กับบุคคลที่กล่าวถึงตลอดไปเทอญ...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. วิจิตร กิณเรศ , " เจาะลึกฟลักซ์เวกเตอร์ไดรฟ์ , 148 , เทคนิค , กรุงเทพฯ , 2540
2. "" RS-422 and RS-485 Application Note "" , B&B Electronic , Ottawa , 1992
3. " คู่มือการใช้งาน Inverter Flux Vector 1100 "
4. " คู่มือการใช้งานบอร์ด V-C2051 "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include"c:\tc\project\headfile\vesa.h"
#include"c:\tc\project\headfile\graph.h"
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

#define ESC 0x1B
#define DOWN 0x50
#define UP 0x48
#define LEFT 0x4b
#define RIGHT 0x4d
#define ENTER 0X0d

void ButtonSelect(int menu, int del)
{
  switch(menu)
  {
    case 0: if(del)
                DisplayButton(80,176,0,0xb2,"speed.mlt");
            else
                DisplayButton(80,176,0,0x21,"speed.mlt");
            break;
    case 1: if(del)
                DisplayButton(240,176,0,0xb2,"torque.mlt");
            else
                DisplayButton(240,176,0,0x21,"torque.mlt");
            break;
    case 2: if(del)
                DisplayButton(400,176,0,0xb2,"current.mlt");
            else
                DisplayButton(400,176,0,0x21,"current.mlt");
            break;
    case 3: if(del)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DisplayButton(80,304,0,0xb2,"open.mlt");
            else
                DisplayButton(80,304,0,0x21,"open.mlt");
            break;
        case 4: if(del)
                DisplayButton(388,304,0,0xb2,"control.mlt");
            else
                DisplayButton(388,304,0,0x21,"control.mlt");
            break;
        case 5: if(del)
                DisplayButton(488,432,0,0xb2,"exit.mlt");
            else
                DisplayButton(488,432,0,0x21,"exit.mlt");
            break;
        default: break;
    }
}

void main(void)
{
  int c, select = 0;

  if(!OpenGraphic(0x101))
  {
    fprintf(stderr,"Error opening VESA mode...\n");
    exit(0);
  }

  FIRSTMENU:

  DisplayButton(0,0,0,0xac,"toparea.mlt");
  DisplayFileMLT(0,128,"midarea1.mlt");
  DisplayFileMLT(0,256,"midarea2.mlt");
  DisplayFileMLT(0,384,"downarea.mlt");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PutString(5,8,"Monitor motor.",0x21,0x1c);
if(select == 0)
  DisplayButton(80,176,0,0x21,"speed.mlt");
else
  DisplayButton(80,176,0,0xb2,"speed.mlt");
if(select == 1)
  DisplayButton(240,176,0,0x21,"torque.mlt");
else
  DisplayButton(240,176,0,0xb2,"torque.mlt");
if(select == 2)
  DisplayButton(400,176,0,0x21,"current.mlt");
else
  DisplayButton(400,176,0,0xb2,"current.mlt");

PutString(5,16,"View record file.",0x21,0x1c);
if(select == 3)
  DisplayButton(80,304,0,0x21,"open.mlt");
else
  DisplayButton(80,304,0,0xb2,"open.mlt");

PutString(43,16,"Control motor.",0x21,0x1c);
if(select == 4)
  DisplayButton(388,304,0,0x21,"control.mlt");
else
  DisplayButton(388,304,0,0xb2,"control.mlt");

DisplayButton(488,432,0,0xb2,"exit.mlt");

do{
  if (kbhit())
  {
        c=getch();
        switch (c)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case UP: if((select == 0)||(select == 1)||(select == 2))
                break;
        else
        {
                ButtonSelect(select,1);
                switch(select)
                {
                  case 3: select = 0;
                          break;
                  case 4: select = 2;
                          break;
                  case 5: select = 4;
                          break;
                }
                ButtonSelect(select,0);
        }
        break;
case DOWN: if(select == 5)
                break;
          else
          {
                ButtonSelect(select,1);
                switch(select)
                {
                  case 0: select = 3;
                          break;
                  case 1: select = 4;
                          break;
                  case 2: select = 4;
                          break;
                  case 3: select = 5;
                          break;
                  case 4: select = 5;
                          break;
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ButtonSelect(select,0);
            }
            break;
case LEFT: if((select == 0)||(select == 3)||(select == 5))
                break;
            else
            {
                ButtonSelect(select,1);
                switch(select)
                {
                  case 0: select = 0;
                          break;
                  case 1: select = 0;
                          break;
                  case 2: select = 1;
                          break;
                  case 3: select = 3;
                          break;
                  case 4: select = 3;
                          break;
                  case 5: select = 5;
                          break;
                }
                ButtonSelect(select,0);
            }
            break;
case RIGHT: if((select == 2)||(select == 4)||(select == 5))
                 break;
               else
               {
                 ButtonSelect(select,1);
                 switch(select)
                 {
                   case 0: select = 1;
                           break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 1: select = 2;
                        break;
                case 2: select = 2;
                        break;
                case 3: select = 4;
                        break;
                case 4: select = 4;
                        break;
                case 5: select = 5;
                        break;
              }
              ButtonSelect(select,0);
            }
            break;
case ENTER: switch(select)
            {
              case 0: DisplayButton(80,176,1,0x21,"speed.mlt");
                      DisplayButton(80,176,0,0x21,"speed.mlt");
                      spawnl(P_WAIT,"speed.exe","speed.exe",NULL);
                      goto FIRSTMENU;
              case 1: DisplayButton(240,176,1,0x21,"torque.mlt");
                      DisplayButton(240,176,0,0x21,"torque.mlt");
                      spawnl(P_WAIT,"torque.exe","torque.exe",NULL);
                      goto FIRSTMENU;
              case 2: DisplayButton(400,176,1,0x21,"current.mlt");
                      DisplayButton(400,176,0,0x21,"current.mlt");
                      spawnl(P_WAIT,"current.exe","current.exe",NULL);
                      goto FIRSTMENU;
              case 3: DisplayButton(80,304,1,0x21,"open.mlt");
                      DisplayButton(80,304,0,0x21,"open.mlt");
                      spawnl(P_WAIT,"viewrec.exe","viewrec.exe",NULL);
                      goto FIRSTMENU;
              case 4: DisplayButton(388,304,1,0x21,"control.mlt");
                      DisplayButton(388,304,0,0x21,"control.mlt");
                      spawnl(P_WAIT,"control.exe","control.exe",NULL);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    goto FIRSTMENU;
                case 5: DisplayButton(488,432,1,0x21,"exit.mlt");
                    DisplayButton(488,432,0,0xb2,"exit.mlt");
                    goto EXIT;
                default: break;
            }
        default: break;
      }
    }
  }while(1);

  EXIT:

  CloseGraphic();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "c:\tc\project\headfile\serial.h"
#include"c:\tc\project\headfile\inverter.h"
#include"c:\tc\project\headfile\vesa.h"
#include"c:\tc\project\headfile\graph.h"
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

#define FALSE 0
#define TRUE (!FALSE)
#define OFF 0
#define ON 1

#define NOERROR     0          /* No error */
#define BUFOVFL     1          /* Buffer overflowed */

#define ESC 0x1B
#define DOWN 0x50
#define UP 0x48
#define LEFT 0x4b
#define RIGHT 0x4d
#define ENTER 0X0d
#define ASCII 0x007F           /* Mask ASCII characters */
#define SBUFSIZ0x4000          /* Serial buffer size */

int SError = NOERROR;
int portbase = 0;
void interrupt (*oldvects[2])();

static char ccbuf[SBUFSIZ];
unsigned int startbuf = 0;
unsigned int endbuf = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FILE *testfile;
char filename[80],text;

/* Handle communications interrupts and put them in ccbuf */
void interrupt com_int(void)
{
  disable();
  if ((inportb(portbase + IIR) & RX_MASK) == RX_ID)
  {
    if (((endbuf + 1) & SBUFSIZ - 1) == startbuf) SError = BUFOVFL;
    ccbuf[endbuf++] = inportb(portbase + RXR);
    endbuf &= SBUFSIZ - 1;
  }

  /* Signal end of hardware interrupt */
  outportb(ICR, EOI);
  enable();
}

/* Output a character to the serial port */
int SerialOut(char x)
{
  long int timeout = 0x0000FFFFL;

  outportb(portbase + MCR , MC_INT | DTR | RTS);

  /* Wait for Clear To Send from modem */
  while ((inportb(portbase + MSR) & CTS) == 0)
  if (!(--timeout))
  return (-1);

  timeout = 0x0000FFFFL;

  /* Wait for transmitter to clear */
  while ((inportb(portbase + LSR) & XMTRDY) == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (!(--timeout))
    return (-1);

    disable();
    outportb(portbase + TXR, x);
    enable();

    return (0);
}

/* Output a string to the serial port */
void SerialString(char *string)
{
  int i;

  i = 0;
  while(string[i] != '\0')
  {
    SerialOut(string[i]);
    i++;
  }

}

/* This routine returns the current value in the buffer */
int getccb(void)
{
  int res;

  if (endbuf == startbuf)
  return (-1);

  res = (int) ccbuf[startbuf++];
  startbuf %= SBUFSIZ;
  return (res);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

/* Install our functions to handle communications */
void setvects(void)
{
  if (portbase == COM1BASE)
  {
    oldvects[1] = getvect(0x0C);
    setvect(0x0C, com_int);
  }
  if (portbase == COM2BASE)
  {
  oldvects[0] = getvect(0x0B);
  setvect(0x0B, com_int);
  }
}

/* Uninstall our vectors before exiting the program */
void resvects(void)
{
  if (portbase == COM1BASE)
    setvect(0x0C, oldvects[1]);
  if (portbase == COM2BASE)
    setvect(0x0B, oldvects[0]);
}

/* Turn on communications interrupts */
void i_enable(int pnum)
{
  int c;

  disable();
  c = inportb(portbase + MCR) | MC_INT;
  outportb(portbase + MCR , c);
  outportb(portbase + IER , RX_INT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    c = inportb(IMR) & (((pnum == COM1) || (pnum == COM1)) ? IRQ4 : IRQ3);
    outportb(IMR, c);
    enable();
}

/* Turn off communications interrupts */
void i_disable(void)
{
    int c;

    disable();
    if (portbase == COM1BASE)
        c = inportb(IMR) | ~IRQ4;
    if (portbase == COM2BASE)
        c = inportb(IMR) | ~IRQ3;
    outportb(IMR, c);
    outportb(portbase + IER, 0);
    c = inportb(portbase + MCR) & ~MC_INT;
    outportb(portbase + MCR, c);
    enable();
}

/* Tell modem that we're ready to go */
void comm_on(void)
{
    int c , pnum;

    pnum = (portbase == COM1BASE ? COM1 : COM2);
    i_enable(pnum);
    c = inportb(portbase + MCR) | DTR | RTS;
    outportb(portbase + MCR, c);
}

/* Go off-line */
void comm_off(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  i_disable();
  outportb(portbase + MCR, 0);
}

void initserial(void)
{
  endbuf = startbuf = 0;
  setvects();
  comm_on();
}

void closeserial(void)
{
  comm_off();
  resvects();
}

/* Set the port number to use */
int SetPort(int Port)
{
  int Offset , far *RS232_Addr;

  switch (Port)
  {
  /* Sort out the base address */
  case COM1 : Offset = 0x0000; break;
  case COM2 : Offset = 0x0002; break;
  default       : return (-1);
  }

  RS232_Addr = MK_FP(0x0040 , Offset);  /* Find out where the port is.        */
  if (*RS232_Addr == NULL) return (-1); /* If NULL, then port not used. */
  portbase = *RS232_Addr; /* Otherwise, set portbase. */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  return (0);
}

/* This routine sets the speed; will accept funny baud rates. */
/* Setting the speed requires that the DLAB be set on. */
int SetSpeed(int Speed)
{
  char  c;
  int divisor;

  if (Speed == 0) /* Avoid divide by zero */
  return (-1);
  else
  divisor = (int) (115200L/Speed);

  if (portbase == 0)
  return (-1);

  disable();
  c = inportb(portbase + LCR);
  outportb(portbase + LCR, (c | 0x80)); /* Set DLAB  */
  outportb(portbase + DLL, (divisor & 0x00FF));
  outportb(portbase + DLH, ((divisor >> 8) & 0x00FF));
  outportb(portbase + LCR, c);     /* Reset DLAB */
  enable();

  return (0);
}

/* Set other communications parameters */
int SetOthers(int Parity , int Bits , int StopBit)
{
  int setting;

  if (portbase == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return(-1);
    if (Bits &5 || Bits > 8)
    return(-1);
    if (StopBit != 1 && StopBit != 2)
    return (-1);
    if (Parity != NO_PARITY && Parity != ODD_PARITY && Parity != EVEN_PARITY)
    return (-1);

    setting  = Bits-5;
    setting |= ((StopBit == 1) ? 0x00 : 0x04);
    setting |= Parity;

    disable();
    outportb(portbase + LCR, setting);
    enable();

    return (0);
}

/* Set up the port */
int SetSerial(int Port , int Speed , int Parity , int Bits , int StopBit)
{
  if (SetPort(Port)) return (-1);
  if (SetSpeed(Speed))    return (-1);
  if (SetOthers(Parity , Bits , StopBit)) return (-1);

  return (0);
}

/* Control-Break interrupt handler */
int c_break(void)
{
  i_disable();
  fprintf(stderr, "\nStill online.\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return(0);
}

void ButtonSelect(int menu, int del)
{
  switch(menu)
  {
    case 0: if(del)
                DisplayButton(344,352,0,0xb2,"start.mlt");
            else
                DisplayButton(344,352,0,0x21,"start.mlt");
            break;
    case 1: if(del)
                DisplayButton(488,352,0,0xb2,"stop.mlt");
            else
                DisplayButton(488,352,0,0x21,"stop.mlt");
            break;
    case 2: if(del)
                DisplayButton(344,392,0,0xb2,"record.mlt");
            else
                DisplayButton(344,392,0,0x21,"record.mlt");
            break;
    case 3: if(del)
                DisplayButton(488,392,0,0xb2,"stoprec.mlt");
            else
                DisplayButton(488,392,0,0x21,"stoprec.mlt");
            break;
    case 4: if(del)
                DisplayButton(344,432,0,0xb2,"setvalue.mlt");
            else
                DisplayButton(344,432,0,0x21,"setvalue.mlt");
            break;
    case 5: if(del)
                DisplayButton(488,432,0,0xb2,"exit.mlt");
            else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    DisplayButton(488,432,0,0x21,"exit.mlt");
                break;
      default: break;
    }
  }


main()
{
  /* Communications parameters */
  int port = COM2;
  int speed = 19200;
  int parity = NO_PARITY;
  int bits = 8;
  int stopbits = 1;

  int c, numberpar, number, basecurrent;
  int recordflag = OFF, startflag = OFF, select = 0;

  char command[28], value[16], base[5], stringbase[20], headstring[] = "current ";

  int queue_display, queue_erase, i, x, y;
  int oldpixel, newpixel, pixel[528];

  time_t timer;
  struct tm *tblock;
  char stringtime[25];

  FILE *recfile;
  char filename[15], rename[30];

  if (SetSerial(port, speed, parity, bits, stopbits) != 0)
  {
    fprintf(stderr, "Serial Port setup error.\n");
    return (99);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

initserial();

ctrlbrk(c_break);

if(!OpenGraphic(0x101))
{
  fprintf(stderr,"Error opening VESA mode...\n");
  exit(0);
}

/* display background program */
DisplayFileMLT(0,0,"screen.mlt");
DisplayFileMLT(0,352,"but_area.mlt");
DisplayButton(344,352,0,0x21,"start.mlt");
DisplayButton(488,352,0,0xb2,"stop.mlt");
DisplayButton(344,392,0,0xb2,"record.mlt");
DisplayButton(488,392,0,0xb2,"stoprec.mlt");
DisplayButton(344,432,0,0xb2,"setvalue.mlt");
DisplayButton(488,432,0,0xb2,"exit.mlt");
DisplayScreen(51,294,1);

/* define command */
command[0] = EOT; command[1] = 0x30; command[2] = 0x30;
command[3] = 0x31; command[4] = 0x31; command[5] = 0x35;
command[6] = 0x30; command[7] = 0x34; command[8] = ENQ;

/* write scale */
PutString(3,1,"current.(pu)",0x78,0x00);
PutString(74,18,"t.",0x78,0x00);
PutString(4,3,"1",0x78,0x00);
PutString(4,10,"0",0x78,0x00);
PutString(3,17,"-1",0x78,0x00);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sprintf(base,"10");
basecurrent = atoi(base);
sprintf(stringbase,"Base current is %s rpm.",base);
PutString(3,23,stringbase,0x21,0x1c);
sprintf(stringbase,"Scale is 1 sec/cell.");
PutString(3,25,stringbase,0x21,0x1c);

/* define initial value */
queue_display = 0;
queue_erase = 528;
for(i = 0;i < 528;i++)
  pixel[i] = 0;

do{
  STARTMENU:
  if (kbhit())
  {
        switch (c=getch())
        {
          case UP: if((select == 0)||(select == 1))
                        break;
                   else
                   {
                        ButtonSelect(select,1);
                        select = select - 2;
                        ButtonSelect(select,0);
                   }
                   break;
          case DOWN: if((select == 4)||(select == 5))
                        break;
                     else
                     {
                        ButtonSelect(select,1);
                        select = select + 2;
                        ButtonSelect(select,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            break;
case LEFT: if((select == 0)||(select == 2)||(select == 4))
                break;
            else
            {
                ButtonSelect(select,1);
                select = select - 1;
                ButtonSelect(select,0);
            }
            break;
case RIGHT: if((select == 1)||(select == 3)||(select == 5))
                 break;
               else
               {
                ButtonSelect(select,1);
                select = select + 1;
                ButtonSelect(select,0);
               }
               break;
case ENTER: switch(select)
               {
                case 0: DisplayButton(344,352,1,0x21,"start.mlt");
                        DisplayButton(344,352,0,0x21,"start.mlt");
                        PutString(3,19,"Running...",0x30,0x00);
                        startflag = ON;
                        goto CONTINUEGRAPH;
                case 1: DisplayButton(488,352,1,0x21,"stop.mlt");
                        DisplayButton(488,352,0,0x21,"stop.mlt");
                        if(!startflag)
                          break;
                        startflag = OFF;
                        if(recordflag)
                        {
                          recordflag = OFF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fclose(recfile);
  }
  /* display screen */
  DisplayFileMLT(0,0,"screen.mlt");
  DisplayScreen(51,294,1);
  PutString(3,1,"current.(pu)",0x78,0x00);
  PutString(74,18,"t.",0x78,0x00);
  PutString(4,3,"1",0x78,0x00);
  PutString(4,10,"0",0x78,0x00);
  PutString(3,17,"-1",0x78,0x00);
  /* define initial value */
  queue_display = 0;
  queue_erase = 528;
  for(i = 0;i < 528;i++)
    pixel[i] = 0;
  goto STARTMENU;
case 2: DisplayButton(344,392,1,0x21,"record.mlt");
  DisplayButton(344,392,0,0x21,"record.mlt");
  if(!startflag)
  {
   if(recordflag)
        break;
   break;
  }
  PutString(3,19,"          ",0x00,0x00);
  PutString(3,28,"Enter file name        .rec",0x1c,0x00);
  filename[0] = 8;
  GetString(19,28, filename,0x1c,0x00);
  strcat(filename,".rec");
  PutString(3,28,"                          ",0x1c,0x1c);
  PutString(3,20,"Recording...",0x28,0x00);
  PutString(3,19,"Running...",0x30,0x00);
  /* open file record */
  recfile = fopen(filename,"w+b");
  fwrite(headstring,(8*sizeof(char)),1,recfile);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                /* time when record graph */
                timer = time(NULL);
                fwrite(&timer,sizeof(time_t),1,recfile);
                recordflag = ON;
                goto CONTINUEGRAPH;
        case 3: DisplayButton(488,392,1,0x21,"stoprec.mlt");
                DisplayButton(488,392,0,0x21,"stoprec.mlt");
                if(recordflag)
                {
                  PutString(3,20,"          ",0x00,0x00);
                  recordflag = OFF;
                  fclose(recfile);
                  goto CONTINUEGRAPH;
                }
                break;
        case 4: DisplayButton(344,432,1,0x21,"setvalue.mlt");
                DisplayButton(344,432,0,0x21,"setvalue.mlt");
                if(startflag)
                  break;
                PutString(3,23,"Base current is    Amp.",0x1c,0x00);
                base[0] = 2;
                GetString(19,23, base,0x1c,0x00);
                sprintf(stringbase,"Base current is %s Amp. ",base);
                PutString(3,23,stringbase,0x21,0x1c);
                basecurrent = atoi(base);
                /* display screen */
                DisplayFileMLT(0,0,"screen.mlt");
                DisplayScreen(51,294,1);
                PutString(3,1,"current.(pu)",0x78,0x00);
                PutString(74,18,"t.",0x78,0x00);
                PutString(4,3,"1",0x78,0x00);
                PutString(4,10,"0",0x78,0x00);
                PutString(3,17,"-1",0x78,0x00);
                break;
        case 5: DisplayButton(488,432,1,0x21,"exit.mlt");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    DisplayButton(488,432,0,0x21,"exit.mlt");
                                    goto EXITPROGRAM;
                                }
            default: break;
          }
        }

        if(!startflag)
            goto STARTMENU;

        CONTINUEGRAPH:
        /* display grid cell */
        DisplayScreen(51,294,1);
        delay(13);

        #define sbase basecurrent

        /* display current time */
        timer = time(NULL);
        tblock = localtime(&timer);
        putVESAchar(72,19,0x5E,0x78,0x00);
        PutString(50,20,asctime(tblock),0x78,0x00);

        if(!(166-(int)(((float)oldpixel*112)/sbase) < 40))
            WritePixel(51+0,166-(int)(((float)oldpixel*112)/sbase),0x00); /* deleate oldpixel */
        if(!(166-(int)(((float)pixel[queue_display]*112)/sbase) < 40))
        WritePixel(51+0,166-(int)(((float)pixel[queue_display]*112)/sbase),0x30);
        queue_display++;
        if(queue_display >= 528)
            queue_display = 0;
        queue_erase++;
        if(queue_erase >= 528)
            queue_erase = 0;

        for(i = 1;i < 528;i++) /* loop plot the other pixel */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    if(!(166-(int)(((float)pixel[queue_erase]*112)/sbase) < 40))
      WritePixel(51+i,166-(int)(((float)pixel[queue_erase]*112)/sbase),0x00);
    if(!(166-(int)(((float)pixel[queue_display]*112)/sbase) < 40))
    WritePixel(51+i,166-(int)(((float)pixel[queue_display]*112)/sbase),0x30);
    queue_display++;
    if(queue_display >= 528)
      queue_display = 0;
    queue_erase++;
    if(queue_erase >= 528)
      queue_erase = 0;
}

do{
    SerialString(command);

    STARTLOOP:
    /* initial flag  check routine */
    numberpar = OFF;
    number = OFF;

    do {
      if ((c = getccb()) != -1)
      {
        if(c == STX)
        {
            goto LOOP2;
        }
      }
    }while(!SError);

    LOOP2:
    do {
      if ((c = getccb()) != -1)
      {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    numberpar++;

    if(numberpar >2)
    {
        goto LOOP3;
    }
  }
}while(!SError);

LOOP3:
do {
  if ((c = getccb()) != -1)
  {
    switch(c)
    {
        case EOT: goto STARTLOOP;
        case ETX: value[number] = '\0';
                  goto OUTLOOP;
        default: value[number] = c;
                  number++;
                  break;
    }
  }
}while(!SError);

OUTLOOP:
newpixel = atoi(value); /* use atoi change string to int */
if(recordflag)
  fwrite(&newpixel,sizeof(int),1,recfile);
oldpixel = pixel[queue_display]; /* keep oldpixel */
pixel[queue_display] = newpixel; /* put newpixel in loop array */
queue_display++;
if(queue_display >= 528)
  queue_display = 0;
queue_erase++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(queue_erase >= 528)
              queue_erase = 0;
            goto OUTGETNEW;
      }while (1);
      OUTGETNEW:
    }while (!SError);


    EXITPROGRAM:

    CloseGraphic();
    return (0);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "c:\tc\project\headfile\serial.h"
#include"c:\tc\project\headfile\inverter.h"
#include"c:\tc\project\headfile\vesa.h"
#include"c:\tc\project\headfile\graph.h"
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

#define FALSE 0
#define TRUE (!FALSE)
#define OFF 0
#define ON 1

#define NOERROR     0       /* No error */
#define BUFOVFL     1       /* Buffer overflowed */

#define ESC 0x1B
#define DOWN 0x50
#define UP 0x48
#define LEFT 0x4b
#define RIGHT 0x4d
#define ENTER 0X0d
#define ASCII 0x007F                /* Mask ASCII characters */
#define SBUFSIZ 0x4000              /* Serial buffer size */

int SError = NOERROR;
int portbase = 0;
void interrupt (*oldvects[2])();

static char ccbuf[SBUFSIZ];
unsigned int startbuf = 0;
unsigned int endbuf = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FILE *testfile;
char filename[80],text;

/* Handle communications interrupts and put them in ccbuf */
void interrupt com_int(void)
{
  disable();
  if ((inportb(portbase + IIR) & RX_MASK) == RX_ID)
  {
    if (((endbuf + 1) & SBUFSIZ - 1) == startbuf) SError = BUFOVFL;
    ccbuf[endbuf++] = inportb(portbase + RXR);
    endbuf &= SBUFSIZ - 1;
  }

  /* Signal end of hardware interrupt */
  outportb(ICR, EOI);
  enable();
}

/* Output a character to the serial port */
int SerialOut(char x)
{
  long int timeout = 0x0000FFFFL;

  outportb(portbase + MCR , MC_INT | DTR | RTS);

  /* Wait for Clear To Send from modem */
  while ((inportb(portbase + MSR) & CTS) == 0)
  if (!(--timeout))
  return (-1);

  timeout = 0x0000FFFFL;

  /* Wait for transmitter to clear */
  while ((inportb(portbase + LSR) & XMTRDY) == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (!(--timeout))
    return (-1);

    disable();
    outportb(portbase + TXR, x);
    enable();

    return (0);
}

/* Output a string to the serial port */
void SerialString(char *string)
{
  int i;

  i = 0;
  while(string[i] != '\0')
  {
    SerialOut(string[i]);
    i++;
  }

}

/* This routine returns the current value in the buffer */
int getccb(void)
{
  int res;

  if (endbuf == startbuf)
  return (-1);

  res = (int) ccbuf[startbuf++];
  startbuf %= SBUFSIZ;
  return (res);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

/* Install our functions to handle communications */
void setvects(void)
{
  if (portbase == COM1BASE)
  {
    oldvects[1] = getvect(0x0C);
    setvect(0x0C, com_int);
  }
  if (portbase == COM2BASE)
  {
  oldvects[0] = getvect(0x0B);
  setvect(0x0B, com_int);
  }
}

/* Uninstall our vectors before exiting the program */
void resvects(void)
{
  if (portbase == COM1BASE)
    setvect(0x0C, oldvects[1]);
  if (portbase == COM2BASE)
    setvect(0x0B, oldvects[0]);
}

/* Turn on communications interrupts */
void i_enable(int pnum)
{
  int c;

  disable();
  c = inportb(portbase + MCR) | MC_INT;
  outportb(portbase + MCR , c);
  outportb(portbase + IER , RX_INT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    c = inportb(IMR) & (((pnum == COM1) || (pnum == COM1)) ? IRQ4 : IRQ3);
    outportb(IMR, c);
    enable();
  }

  /* Turn off communications interrupts */
  void i_disable(void)
  {
    int c;

    disable();
    if (portbase == COM1BASE)
      c = inportb(IMR) | ~IRQ4;
    if (portbase == COM2BASE)
      c = inportb(IMR) | ~IRQ3;
    outportb(IMR, c);
    outportb(portbase + IER, 0);
    c = inportb(portbase + MCR) & ~MC_INT;
    outportb(portbase + MCR, c);
    enable();
  }

  /* Tell modem that we're ready to go */
  void comm_on(void)
  {
    int c , pnum;

    pnum = (portbase == COM1BASE ? COM1 : COM2);
    i_enable(pnum);
    c = inportb(portbase + MCR) | DTR | RTS;
    outportb(portbase + MCR, c);
  }

  /* Go off-line */
  void comm_off(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  i_disable();
  outportb(portbase + MCR, 0);
}

void initserial(void)
{
  endbuf = startbuf = 0;
  setvects();
  comm_on();
}

void closeserial(void)
{
  comm_off();
  resvects();
}

/* Set the port number to use */
int SetPort(int Port)
{
  int Offset , far *RS232_Addr;

  switch (Port)
  {
  /* Sort out the base address */
  case COM1 : Offset = 0x0000; break;
  case COM2 : Offset = 0x0002; break;
  default   : return (-1);
  }

  RS232_Addr = MK_FP(0x0040 , Offset);  /* Find out where the port is.        */
  if (*RS232_Addr == NULL) return (-1); /* If NULL, then port not used. */
  portbase = *RS232_Addr; /* Otherwise, set portbase. */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return (0);
    }

    /* This routine sets the speed; will accept funny baud rates. */
    /* Setting the speed requires that the DLAB be set on. */
    int SetSpeed(int Speed)
    {
      char  c;
      int divisor;

      if (Speed == 0) /* Avoid divide by zero */
      return (-1);
      else
      divisor = (int) (115200L/Speed);

      if (portbase == 0)
      return (-1);

      disable();
      c = inportb(portbase + LCR);
      outportb(portbase + LCR, (c | 0x80)); /* Set DLAB  */
      outportb(portbase + DLL, (divisor & 0x00FF));
      outportb(portbase + DLH, ((divisor >> 8) & 0x00FF));
      outportb(portbase + LCR, c);    /* Reset DLAB */
      enable();

     return (0);
}

/* Set other communications parameters */
int SetOthers(int Parity , int Bits , int StopBit)
{
  int setting;

  if (portbase == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return(-1);
    if (Bits &5 || Bits > 8)
    return(-1);
    if (StopBit != 1 && StopBit != 2)
    return (-1);
    if (Parity != NO_PARITY && Parity != ODD_PARITY && Parity != EVEN_PARITY)
    return (-1);

    setting  = Bits-5;
    setting |= ((StopBit == 1) ? 0x00 : 0x04);
    setting |= Parity;

    disable();
    outportb(portbase + LCR, setting);
    enable();

    return (0);
}

/* Set up the port */
int SetSerial(int Port , int Speed , int Parity , int Bits , int StopBit)
{
    if (SetPort(Port)) return (-1);
    if (SetSpeed(Speed))    return (-1);
    if (SetOthers(Parity , Bits , StopBit)) return (-1);

    return (0);
}

/* Control-Break interrupt handler */
int c_break(void)
{
    i_disable();
    fprintf(stderr, "\nStill online.\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  return(0);
}

void ButtonSelect(int menu, int del)
{
  switch(menu)
  {
    case 0: if(del)
                DisplayButton(344,352,0,0xb2,"start.mlt");
            else
                DisplayButton(344,352,0,0x21,"start.mlt");
            break;
    case 1: if(del)
                DisplayButton(488,352,0,0xb2,"stop.mlt");
            else
                DisplayButton(488,352,0,0x21,"stop.mlt");
            break;
    case 2: if(del)
                DisplayButton(344,392,0,0xb2,"record.mlt");
            else
                DisplayButton(344,392,0,0x21,"record.mlt");
            break;
    case 3: if(del)
                DisplayButton(488,392,0,0xb2,"stoprec.mlt");
            else
                DisplayButton(488,392,0,0x21,"stoprec.mlt");
            break;
    case 4: if(del)
                DisplayButton(344,432,0,0xb2,"setvalue.mlt");
            else
                DisplayButton(344,432,0,0x21,"setvalue.mlt");
            break;
    case 5: if(del)
                DisplayButton(488,432,0,0xb2,"exit.mlt");
            else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    DisplayButton(488,432,0,0x21,"exit.mlt");
                break;
        default: break;
      }
    }


    main()
    {
      /* Communications parameters */
      int port = COM2;
      int speed = 19200;
      int parity = NO_PARITY;
      int bits = 8;
      int stopbits = 1;

      int c, numberpar, number, basetorque;
      int recordflag = OFF, startflag = OFF, select = 0;

      char command[28], value[16], base[5], stringbase[20], headstring[] = "torque ";

      int queue_display, queue_erase, i, x, y;
      int oldpixel, newpixel, pixel[528];

      time_t timer;
      struct tm *tblock;
      char stringtime[25];

      FILE *recfile;
      char filename[15], rename[30];

      if (SetSerial(port, speed, parity, bits, stopbits) != 0)
      {
        fprintf(stderr, "Serial Port setup error.\n");
        return (99);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

initserial();

ctrlbrk(c_break);

if(!OpenGraphic(0x101))
{
  fprintf(stderr,"Error opening VESA mode...\n");
  exit(0);
}

/* display background program */
DisplayFileMLT(0,0,"screen.mlt");
DisplayFileMLT(0,352,"but_area.mlt");
DisplayButton(344,352,0,0x21,"start.mlt");
DisplayButton(488,352,0,0xb2,"stop.mlt");
DisplayButton(344,392,0,0xb2,"record.mlt");
DisplayButton(488,392,0,0xb2,"stoprec.mlt");
DisplayButton(344,432,0,0xb2,"setvalue.mlt");
DisplayButton(488,432,0,0xb2,"exit.mlt");
DisplayScreen(51,294,1);

/* define command */
command[0] = EOT; command[1] = 0x30; command[2] = 0x30;
command[3] = 0x31; command[4] = 0x31; command[5] = 0x35;
command[6] = 0x30; command[7] = 0x33; command[8] = ENQ;

/* write scale */
PutString(3,1,"torque.(pu)",0x78,0x00);
PutString(74,18,"t.",0x78,0x00);
PutString(4,3,"1",0x78,0x00);
PutString(4,10,"0",0x78,0x00);
PutString(3,17,"-1",0x78,0x00);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sprintf(base,"100");
basetorque = atoi(base);
sprintf(stringbase,"Base torque is %s %",base);
PutString(3,23,stringbase,0x21,0x1c);
sprintf(stringbase,"Scale is 1 sec/cell.");
PutString(3,25,stringbase,0x21,0x1c);

/* define initial value */
queue_display = 0;
queue_erase = 528;
for(i = 0;i < 528;i++)
  pixel[i] = 0;

do{
  STARTMENU:
  if (kbhit())
  {
        switch (c=getch())
        {
          case UP: if((select == 0)||(select == 1))
                        break;
                   else
                   {
                        ButtonSelect(select,1);
                        select = select - 2;
                        ButtonSelect(select,0);
                   }
                   break;
          case DOWN: if((select == 4)||(select == 5))
                        break;
                    else
                    {
                        ButtonSelect(select,1);
                        select = select + 2;
                        ButtonSelect(select,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            break;
case LEFT: if((select == 0)||(select == 2)||(select == 4))
                break;
            else
            {
                ButtonSelect(select,1);
                select = select - 1;
                ButtonSelect(select,0);
            }
            break;
case RIGHT: if((select == 1)||(select == 3)||(select == 5))
                break;
            else
            {
                ButtonSelect(select,1);
                select = select + 1;
                ButtonSelect(select,0);
            }
            break;
case ENTER: switch(select)
            {
            case 0: DisplayButton(344,352,1,0x21,"start.mlt");
                    DisplayButton(344,352,0,0x21,"start.mlt");
                    PutString(3,19,"Running...",0x30,0x00);
                    startflag = ON;
                    goto CONTINUEGRAPH;
            case 1: DisplayButton(488,352,1,0x21,"stop.mlt");
                    DisplayButton(488,352,0,0x21,"stop.mlt");
                    if(!startflag)
                      break;
                    startflag = OFF;
                    if(recordflag)
                    {
                      recordflag = OFF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fclose(recfile);
      }
      /* display screen */
      DisplayFileMLT(0,0,"screen.mlt");
      DisplayScreen(51,294,1);
      PutString(3,1,"torque.(pu)",0x78,0x00);
      PutString(74,18,"t.",0x78,0x00);
      PutString(4,3,"1",0x78,0x00);
      PutString(4,10,"0",0x78,0x00);
      PutString(3,17,"-1",0x78,0x00);
      /* define initial value */
      queue_display = 0;
      queue_erase = 528;
      for(i = 0;i < 528;i++)
        pixel[i] = 0;
      goto STARTMENU;
case 2: DisplayButton(344,392,1,0x21,"record.mlt");
      DisplayButton(344,392,0,0x21,"record.mlt");
      if(!startflag)
      {
        if(recordflag)
            break;
        break;
      }
      PutString(3,19,"          ",0x00,0x00);
      PutString(3,28,"Enter file name        .rec",0x1c,0x00);
      filename[0] = 8;
      GetString(19,28, filename,0x1c,0x00);
      strcat(filename,".rec");
      PutString(3,28,"                          ",0x1c,0x1c);
      PutString(3,20,"Recording...",0x28,0x00);
      PutString(3,19,"Running...",0x30,0x00);
      /* open file record */
      recfile = fopen(filename,"w+b");
      fwrite(headstring,(8*sizeof(char)),1,recfile);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                /* time when record graph */
                timer = time(NULL);
                fwrite(&timer,sizeof(time_t),1,recfile);
                recordflag = ON;
                goto CONTINUEGRAPH;
        case 3: DisplayButton(488,392,1,0x21,"stoprec.mlt");
                DisplayButton(488,392,0,0x21,"stoprec.mlt");
                if(recordflag)
                {
                  PutString(3,20,"          ",0x00,0x00);
                  recordflag = OFF;
                  fclose(recfile);
                  goto CONTINUEGRAPH;
                }
                break;
        case 4: DisplayButton(344,432,1,0x21,"setvalue.mlt");
                DisplayButton(344,432,0,0x21,"setvalue.mlt");
                if(startflag)
                  break;
                PutString(3,23,"Base torque is    %",0x1c,0x00);
                base[0] = 3;
                GetString(18,23, base,0x1c,0x00);
                sprintf(stringbase,"Base torque is %s % ",base);
                PutString(3,23,stringbase,0x21,0x1c);
                basetorque = atoi(base);
                /* display screen */
                DisplayFileMLT(0,0,"screen.mlt");
                DisplayScreen(51,294,1);
                PutString(3,1,"torque.(pu)",0x78,0x00);
                PutString(74,18,"t.",0x78,0x00);
                PutString(4,3,"1",0x78,0x00);
                PutString(4,10,"0",0x78,0x00);
                PutString(3,17,"-1",0x78,0x00);
                break;
        case 5: DisplayButton(488,432,1,0x21,"exit.mlt");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        DisplayButton(488,432,0,0x21,"exit.mlt");
                                        goto EXITPROGRAM;
                                }
            default: break;
        }
    }

    if(!startflag)
        goto STARTMENU;

    CONTINUEGRAPH:
    /* display grid cell */
    DisplayScreen(51,294,1);
    delay(13);

    #define sbase basetorque

    /* display current time */
    timer = time(NULL);
    tblock = localtime(&timer);
    putVESAchar(72,19,0x5E,0x78,0x00);
    PutString(50,20,asctime(tblock),0x78,0x00);

    if(!(166-(int)(((float)oldpixel*112)/sbase) < 40))
        WritePixel(51+0,166-(int)(((float)oldpixel*112)/sbase),0x00); /* deleate oldpixel */
    if(!(166-(int)(((float)pixel[queue_display]*112)/sbase) < 40))
    WritePixel(51+0,166-(int)(((float)pixel[queue_display]*112)/sbase),0x30);
    queue_display++;
    if(queue_display >= 528)
        queue_display = 0;
    queue_erase++;
    if(queue_erase >= 528)
        queue_erase = 0;

    for(i = 1;i < 528;i++) /* loop plot the other pixel */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    if(!(166-(int)(((float)pixel[queue_erase]*112)/sbase) < 40))
      WritePixel(51+i,166-(int)(((float)pixel[queue_erase]*112)/sbase),0x00);
    if(!(166-(int)(((float)pixel[queue_display]*112)/sbase) < 40))
    WritePixel(51+i,166-(int)(((float)pixel[queue_display]*112)/sbase),0x30);
    queue_display++;
    if(queue_display >= 528)
      queue_display = 0;
    queue_erase++;
    if(queue_erase >= 528)
      queue_erase = 0;
}

do{
    SerialString(command);

    STARTLOOP:
    /* initial flag  check routine */
    numberpar = OFF;
    number = OFF;

    do {
      if ((c = getccb()) != -1)
      {
        if(c == STX)
        {
            goto LOOP2;
        }
      }
    }while(!SError);

    LOOP2:
    do {
      if ((c = getccb()) != -1)
      {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        numberpar++;

        if(numberpar >2)
        {
            goto LOOP3;
        }
      }
    }while(!SError);

    LOOP3:
    do {
      if ((c = getccb()) != -1)
      {
        switch(c)
        {
            case EOT: goto STARTLOOP;
            case ETX: value[number] = '\0';
                goto OUTLOOP;
            default: value[number] = c;
                number++;
                break;
        }
      }
    }while(!SError);

    OUTLOOP:
    newpixel = atoi(value); /* use atoi change string to int */
    if(recordflag)
      fwrite(&newpixel,sizeof(int),1,recfile);
    oldpixel = pixel[queue_display]; /* keep oldpixel */
    pixel[queue_display] = newpixel; /* put newpixel in loop array */
    queue_display++;
    if(queue_display >= 528)
      queue_display = 0;
    queue_erase++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(queue_erase >= 528)
              queue_erase = 0;
            goto OUTGETNEW;
        }while (1);
        OUTGETNEW:
    }while (!SError);


    EXITPROGRAM:

    CloseGraphic();
    return (0);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "c:\tc\project\headfile\serial.h"
#include"c:\tc\project\headfile\inverter.h"
#include"c:\tc\project\headfile\vesa.h"
#include"c:\tc\project\headfile\graph.h"
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

#define FALSE 0
#define TRUE (!FALSE)
#define OFF 0
#define ON 1

#define NOERROR     0          /* No error */
#define BUFOVFL     1          /* Buffer overflowed */

#define ESC 0x1B
#define DOWN 0x50
#define UP 0x48
#define LEFT 0x4b
#define RIGHT 0x4d
#define ENTER 0X0d
#define ASCII 0x007F                /* Mask ASCII characters */
#define SBUFSIZ0x4000                /* Serial buffer size */

int SError = NOERROR;
int portbase = 0;
void interrupt (*oldvects[2])();

static char ccbuf[SBUFSIZ];
unsigned int startbuf = 0;
unsigned int endbuf = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FILE *testfile;
char filename[80],text;

/* Handle communications interrupts and put them in ccbuf */
void interrupt com_int(void)
{
  disable();
  if ((inportb(portbase + IIR) & RX_MASK) == RX_ID)
  {
    if (((endbuf + 1) & SBUFSIZ - 1) == startbuf) SError = BUFOVFL;
    ccbuf[endbuf++] = inportb(portbase + RXR);
    endbuf &= SBUFSIZ - 1;
  }

  /* Signal end of hardware interrupt */
  outportb(ICR, EOI);
  enable();
}

/* Output a character to the serial port */
int SerialOut(char x)
{
  long int timeout = 0x0000FFFFL;

  outportb(portbase + MCR , MC_INT | DTR | RTS);

  /* Wait for Clear To Send from modem */
  while ((inportb(portbase + MSR) & CTS) == 0)
  if (!(--timeout))
  return (-1);

  timeout = 0x0000FFFFL;

  /* Wait for transmitter to clear */
  while ((inportb(portbase + LSR) & XMTRDY) == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if (!(--timeout))
  return (-1);

  disable();
  outportb(portbase + TXR, x);
  enable();

  return (0);
}

/* Output a string to the serial port */
void SerialString(char *string)
{
  int i;

  i = 0;
  while(string[i] != '\0')
  {
    SerialOut(string[i]);
    i++;
  }

}

/* This routine returns the current value in the buffer */
int getccb(void)
{
  int res;

  if (endbuf == startbuf)
  return (-1);

  res = (int) ccbuf[startbuf++];
  startbuf %= SBUFSIZ;
  return (res);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

/* Install our functions to handle communications */
void setvects(void)
{
  if (portbase == COM1BASE)
  {
    oldvects[1] = getvect(0x0C);
    setvect(0x0C, com_int);
  }
  if (portbase == COM2BASE)
  {
  oldvects[0] = getvect(0x0B);
  setvect(0x0B, com_int);
  }
}

/* Uninstall our vectors before exiting the program */
void resvects(void)
{
  if (portbase == COM1BASE)
    setvect(0x0C, oldvects[1]);
  if (portbase == COM2BASE)
    setvect(0x0B, oldvects[0]);
}

/* Turn on communications interrupts */
void i_enable(int pnum)
{
  int c;

  disable();
  c = inportb(portbase + MCR) | MC_INT;
  outportb(portbase + MCR , c);
  outportb(portbase + IER , RX_INT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  c = inportb(IMR) & (((pnum == COM1) || (pnum == COM1)) ? IRQ4 : IRQ3);
  outportb(IMR, c);
  enable();
}

/* Turn off communications interrupts */
void i_disable(void)
{
  int c;

  disable();
  if (portbase == COM1BASE)
    c = inportb(IMR) | ~IRQ4;
  if (portbase == COM2BASE)
    c = inportb(IMR) | ~IRQ3;
  outportb(IMR, c);
  outportb(portbase + IER, 0);
  c = inportb(portbase + MCR) & ~MC_INT;
  outportb(portbase + MCR, c);
  enable();
}

/* Tell modem that we're ready to go */
void comm_on(void)
{
  int c , pnum;

  pnum = (portbase == COM1BASE ? COM1 : COM2);
  i_enable(pnum);
  c = inportb(portbase + MCR) | DTR | RTS;
  outportb(portbase + MCR, c);
}

/* Go off-line */
void comm_off(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  i_disable();
  outportb(portbase + MCR, 0);
}

void initserial(void)
{
  endbuf = startbuf = 0;
  setvects();
  comm_on();
}

void closeserial(void)
{
  comm_off();
  resvects();
}

/* Set the port number to use */
int SetPort(int Port)
{
  int Offset , far *RS232_Addr;

  switch (Port)
  {
  /* Sort out the base address */
  case COM1 : Offset = 0x0000; break;
  case COM2 : Offset = 0x0002; break;
  default           : return (-1);
  }

  RS232_Addr = MK_FP(0x0040 , Offset);  /* Find out where the port is.          */
  if (*RS232_Addr == NULL) return (-1); /* If NULL, then port not used. */
  portbase = *RS232_Addr; /* Otherwise, set portbase. */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  return (0);
}

/* This routine sets the speed; will accept funny baud rates. */
/* Setting the speed requires that the DLAB be set on. */
int SetSpeed(int Speed)
{
  char  c;
  int divisor;

  if (Speed == 0) /* Avoid divide by zero */
  return (-1);
  else
  divisor = (int) (115200L/Speed);

  if (portbase == 0)
  return (-1);

  disable();
  c = inportb(portbase + LCR);
  outportb(portbase + LCR, (c | 0x80)); /* Set DLAB  */
  outportb(portbase + DLL, (divisor & 0x00FF));
  outportb(portbase + DLH, ((divisor >> 8) & 0x00FF));
  outportb(portbase + LCR, c);     /* Reset DLAB */
  enable();

  return (0);
}

/* Set other communications parameters */
int SetOthers(int Parity , int Bits , int StopBit)
{
  int setting;

  if (portbase == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return(-1);
    if (Bits &5 || Bits > 8)
    return(-1);
    if (StopBit != 1 && StopBit != 2)
    return (-1);
    if (Parity != NO_PARITY && Parity != ODD_PARITY && Parity != EVEN_PARITY)
    return (-1);

    setting  = Bits-5;
    setting |= ((StopBit == 1) ? 0x00 : 0x04);
    setting |= Parity;

    disable();
    outportb(portbase + LCR, setting);
    enable();

    return (0);
}

/* Set up the port */
int SetSerial(int Port , int Speed , int Parity , int Bits , int StopBit)
{
    if (SetPort(Port)) return (-1);
    if (SetSpeed(Speed))     return (-1);
    if (SetOthers(Parity , Bits , StopBit)) return (-1);

    return (0);
}

/* Control-Break interrupt handler */
int c_break(void)
{
    i_disable();
    fprintf(stderr, "\nStill online.\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return(0);
}

void ButtonSelect(int menu, int del)
{
    switch(menu)
    {
        case 0: if(del)
                    DisplayButton(344,352,0,0xb2,"start.mlt");
                else
                    DisplayButton(344,352,0,0x21,"start.mlt");
                break;
        case 1: if(del)
                    DisplayButton(488,352,0,0xb2,"stop.mlt");
                else
                    DisplayButton(488,352,0,0x21,"stop.mlt");
                break;
        case 2: if(del)
                    DisplayButton(344,392,0,0xb2,"record.mlt");
                else
                    DisplayButton(344,392,0,0x21,"record.mlt");
                break;
        case 3: if(del)
                    DisplayButton(488,392,0,0xb2,"stoprec.mlt");
                else
                    DisplayButton(488,392,0,0x21,"stoprec.mlt");
                break;
        case 4: if(del)
                    DisplayButton(344,432,0,0xb2,"setvalue.mlt");
                else
                    DisplayButton(344,432,0,0x21,"setvalue.mlt");
                break;
        case 5: if(del)
                    DisplayButton(488,432,0,0xb2,"exit.mlt");
                else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    DisplayButton(488,432,0,0x21,"exit.mlt");
                break;
            default: break;
          }
        }


    main()
    {
      /* Communications parameters */
      int port = COM2;
      int speed = 19200;
      int parity = NO_PARITY;
      int bits = 8;
      int stopbits = 1;

      int c, numberpar, number, basecurrent;
      int recordflag = OFF, startflag = OFF, select = 0;

      char command[28], value[16], base[5], stringbase[20], headstring[] = "current ";

      int queue_display, queue_erase, i, x, y;
      int oldpixel, newpixel, pixel[528];

      time_t timer;
      struct tm *tblock;
      char stringtime[25];

      FILE *recfile;
      char filename[15], rename[30];

    if (SetSerial(port, speed, parity, bits, stopbits) != 0)
    {
      fprintf(stderr, "Serial Port setup error.\n");
      return (99);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

    initserial();

    ctrlbrk(c_break);

    if(!OpenGraphic(0x101))
    {
      fprintf(stderr,"Error opening VESA mode...\n");
      exit(0);
    }

    /* display background program */
    DisplayFileMLT(0,0,"screen.mlt");
    DisplayFileMLT(0,352,"but_area.mlt");
    DisplayButton(344,352,0,0x21,"start.mlt");
    DisplayButton(488,352,0,0xb2,"stop.mlt");
    DisplayButton(344,392,0,0xb2,"record.mlt");
    DisplayButton(488,392,0,0xb2,"stoprec.mlt");
    DisplayButton(344,432,0,0xb2,"setvalue.mlt");
    DisplayButton(488,432,0,0xb2,"exit.mlt");
    DisplayScreen(51,294,1);

    /* define command */
    command[0] = EOT; command[1] = 0x30; command[2] = 0x30;
    command[3] = 0x31; command[4] = 0x31; command[5] = 0x35;
    command[6] = 0x30; command[7] = 0x34; command[8] = ENQ;

    /* write scale */
    PutString(3,1,"current.(pu)",0x78,0x00);
    PutString(74,18,"t.",0x78,0x00);
    PutString(4,3,"1",0x78,0x00);
    PutString(4,10,"0",0x78,0x00);
    PutString(3,17,"-1",0x78,0x00);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sprintf(base,"10");
basecurrent = atoi(base);
sprintf(stringbase,"Base current is %s rpm.",base);
PutString(3,23,stringbase,0x21,0x1c);
sprintf(stringbase,"Scale is 1 sec/cell.");
PutString(3,25,stringbase,0x21,0x1c);

/* define initial value */
queue_display = 0;
queue_erase = 528;
for(i = 0;i < 528;i++)
  pixel[i] = 0;

do{
  STARTMENU:
  if (kbhit())
  {
        switch (c=getch())
        {
          case UP: if((select == 0)||(select == 1))
                        break;
                   else
                   {
                        ButtonSelect(select,1);
                        select = select - 2;
                        ButtonSelect(select,0);
                   }
                   break;
          case DOWN: if((select == 4)||(select == 5))
                        break;
                     else
                     {
                        ButtonSelect(select,1);
                        select = select + 2;
                        ButtonSelect(select,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                break;
        case LEFT: if((select == 0)||(select == 2)||(select == 4))
                        break;
                else
                {
                        ButtonSelect(select,1);
                        select = select - 1;
                        ButtonSelect(select,0);
                }
                break;
        case RIGHT: if((select == 1)||(select == 3)||(select == 5))
                        break;
                    else
                    {
                     ButtonSelect(select,1);
                     select = select + 1;
                     ButtonSelect(select,0);
                    }
                    break;
        case ENTER: switch(select)
                    {
                     case 0: DisplayButton(344,352,1,0x21,"start.mlt");
                             DisplayButton(344,352,0,0x21,"start.mlt");
                             PutString(3,19,"Running...",0x30,0x00);
                             startflag = ON;
                             goto CONTINUEGRAPH;
                     case 1: DisplayButton(488,352,1,0x21,"stop.mlt");
                             DisplayButton(488,352,0,0x21,"stop.mlt");
                             if(!startflag)
                               break;
                             startflag = OFF;
                             if(recordflag)
                             {
                               recordflag = OFF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fclose(recfile);
      }
      /* display screen */
      DisplayFileMLT(0,0,"screen.mlt");
      DisplayScreen(51,294,1);
      PutString(3,1,"current.(pu)",0x78,0x00);
      PutString(74,18,"t.",0x78,0x00);
      PutString(4,3,"1",0x78,0x00);
      PutString(4,10,"0",0x78,0x00);
      PutString(3,17,"-1",0x78,0x00);
      /* define initial value */
      queue_display = 0;
      queue_erase = 528;
      for(i = 0;i < 528;i++)
        pixel[i] = 0;
      goto STARTMENU;
    case 2: DisplayButton(344,392,1,0x21,"record.mlt");
      DisplayButton(344,392,0,0x21,"record.mlt");
      if(!startflag)
      {
        if(recordflag)
          break;
        break;
      }
      PutString(3,19,"          ",0x00,0x00);
      PutString(3,28,"Enter file name        .rec",0x1c,0x00);
      filename[0] = 8;
      GetString(19,28, filename,0x1c,0x00);
      strcat(filename,".rec");
      PutString(3,28,"                          ",0x1c,0x1c);
      PutString(3,20,"Recording...",0x28,0x00);
      PutString(3,19,"Running...",0x30,0x00);
      /* open file record */
      recfile = fopen(filename,"w+b");
      fwrite(headstring,(8*sizeof(char)),1,recfile);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                /* time when record graph */
                timer = time(NULL);
                fwrite(&timer,sizeof(time_t),1,recfile);
                recordflag = ON;
                goto CONTINUEGRAPH;
        case 3: DisplayButton(488,392,1,0x21,"stoprec.mlt");
                DisplayButton(488,392,0,0x21,"stoprec.mlt");
                if(recordflag)
                {
                  PutString(3,20,"          ",0x00,0x00);
                  recordflag = OFF;
                  fclose(recfile);
                  goto CONTINUEGRAPH;
                }
                break;
        case 4: DisplayButton(344,432,1,0x21,"setvalue.mlt");
                DisplayButton(344,432,0,0x21,"setvalue.mlt");
                if(startflag)
                  break;
                PutString(3,23,"Base current is    Amp.",0x1c,0x00);
                base[0] = 2;
                GetString(19,23, base,0x1c,0x00);
                sprintf(stringbase,"Base current is %s Amp. ",base);
                PutString(3,23,stringbase,0x21,0x1c);
                basecurrent = atoi(base);
                /* display screen */
                DisplayFileMLT(0,0,"screen.mlt");
                DisplayScreen(51,294,1);
                PutString(3,1,"current.(pu)",0x78,0x00);
                PutString(74,18,"t.",0x78,0x00);
                PutString(4,3,"1",0x78,0x00);
                PutString(4,10,"0",0x78,0x00);
                PutString(3,17,"-1",0x78,0x00);
                break;
        case 5: DisplayButton(488,432,1,0x21,"exit.mlt");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        DisplayButton(488,432,0,0x21,"exit.mlt");
                                        goto EXITPROGRAM;
                                }
            default: break;
        }
    }

    if(!startflag)
        goto STARTMENU;

    CONTINUEGRAPH:
    /* display grid cell */
    DisplayScreen(51,294,1);
    delay(13);

    #define sbase basecurrent

    /* display current time */
    timer = time(NULL);
    tblock = localtime(&timer);
    putVESAchar(72,19,0x5E,0x78,0x00);
    PutString(50,20,asctime(tblock),0x78,0x00);

    if(!(166-(int)(((float)oldpixel*112)/sbase) < 40))
        WritePixel(51+0,166-(int)(((float)oldpixel*112)/sbase),0x00); /* deleate oldpixel */
    if(!(166-(int)(((float)pixel[queue_display]*112)/sbase) < 40))
    WritePixel(51+0,166-(int)(((float)pixel[queue_display]*112)/sbase),0x30);
    queue_display++;
    if(queue_display >= 528)
        queue_display = 0;
    queue_erase++;
    if(queue_erase >= 528)
        queue_erase = 0;

    for(i = 1;i < 528;i++) /* loop plot the other pixel */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        if(!(166-(int)(((float)pixel[queue_erase]*112)/sbase) < 40))
          WritePixel(51+i,166-(int)(((float)pixel[queue_erase]*112)/sbase),0x00);
        if(!(166-(int)(((float)pixel[queue_display]*112)/sbase) < 40))
        WritePixel(51+i,166-(int)(((float)pixel[queue_display]*112)/sbase),0x30);
        queue_display++;
        if(queue_display >= 528)
          queue_display = 0;
        queue_erase++;
        if(queue_erase >= 528)
          queue_erase = 0;
}

do{
        SerialString(command);

        STARTLOOP:
        /* initial flag  check routine */
        numberpar = OFF;
        number = OFF;

        do {
          if ((c = getccb()) != -1)
          {
            if(c == STX)
            {
                  goto LOOP2;
            }
          }
        }while(!SError);

        LOOP2:
        do {
          if ((c = getccb()) != -1)
          {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        numberpar++;

        if(numberpar >2)
        {
              goto LOOP3;
        }
      }
    }while(!SError);

    LOOP3:
    do {
      if ((c = getccb()) != -1)
      {
        switch(c)
        {
              case EOT: goto STARTLOOP;
              case ETX: value[number] = '\0';
                    goto OUTLOOP;
              default: value[number] = c;
                    number++;
                    break;
        }
      }
    }while(!SError);

    OUTLOOP:
    newpixel = atoi(value); /* use atoi change string to int */
    if(recordflag)
      fwrite(&newpixel,sizeof(int),1,recfile);
    oldpixel = pixel[queue_display]; /* keep oldpixel */
    pixel[queue_display] = newpixel; /* put newpixel in loop array */
    queue_display++;
    if(queue_display >= 528)
      queue_display = 0;
    queue_erase++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(queue_erase >= 528)
              queue_erase = 0;
            goto OUTGETNEW;
        }while (1);
        OUTGETNEW:
    }while (!SError);


    EXITPROGRAM:

    CloseGraphic();
    return (0);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include"c:\tc\project\headfile\vesa.h"
#include"c:\tc\project\headfile\graph.h"
#include <conio.h>
#include <stdio.h>
#include <time.h>

#define OFF 0
#define ON 1

#define ESC 0x1B
#define DOWN 0x50
#define UP 0x48
#define LEFT 0x4b
#define RIGHT 0x4d
#define ENTER 0X0d

void ButtonSelect(int menu, int del)
{
 switch(menu)
 {
  case 0: if(del)
                DisplayButton(344,352,0,0xb2,"eject.mlt");
          else
                DisplayButton(344,352,0,0x21,"eject.mlt");
          break;
  case 1: if(del)
                DisplayButton(440,352,0,0xb2,"play.mlt");
          else
                DisplayButton(440,352,0,0x21,"play.mlt");
          break;
  case 2: if(del)
                DisplayButton(536,352,0,0xb2,"stopplay.mlt");
          else
                DisplayButton(536,352,0,0x21,"stopplay.mlt");
          break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 3: if(del)
                    DisplayButton(344,392,0,0xb2,"pause.mlt");
                else
                    DisplayButton(344,392,0,0x21,"pause.mlt");
                break;
        case 4: if(del)
                    DisplayButton(440,392,0,0xb2,"previous.mlt");
                else
                    DisplayButton(440,392,0,0x21,"previous.mlt");
                break;
        case 5: if(del)
                    DisplayButton(536,392,0,0xb2,"next.mlt");
                else
                    DisplayButton(536,392,0,0x21,"next.mlt");
                break;
        case 6: if(del)
                    DisplayButton(344,432,0,0xb2,"setvalue.mlt");
                else
                    DisplayButton(344,432,0,0x21,"setvalue.mlt");
                break;
        case 7: if(del)
                    DisplayButton(488,432,0,0xb2,"exit.mlt");
                else
                    DisplayButton(488,432,0,0x21,"exit.mlt");
                break;
        default: break;
    }
}

void main(void)
{
  FILE *recfile;
  char filename[15];

  time_t timer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    struct tm *tblock;
    char stringtime[25];

    int i, select, sbase, loadflag, startflag, pauseflag, counterflag;
    char c;

    char rectype[9], rectime[25], base[5], stringbase[20];
    int pixel[528], delpixel[528];
    long firstdata, enddata, pointerdata, readpointer;

   if(!OpenGraphic(0x101))
   {
     fprintf(stderr,"Error opening VESA mode...\n");
     exit(0);
   }

   /* display background program */
   DisplayFileMLT(0,0,"screen.mlt");
   DisplayFileMLT(0,352,"but_area.mlt");
   DisplayButton(344,352,0,0x21,"eject.mlt");
   DisplayButton(440,352,0,0xb2,"play.mlt");
   DisplayButton(536,352,0,0xb2,"stopplay.mlt");
   DisplayButton(344,392,0,0xb2,"pause.mlt");
   DisplayButton(440,392,0,0xb2,"previous.mlt");
   DisplayButton(536,392,0,0xb2,"next.mlt");
   DisplayButton(344,432,0,0xb2,"setvalue.mlt");
   DisplayButton(488,432,0,0xb2,"exit.mlt");

   /* write scale */
   PutString(74,18,"t.",0x78,0x00);
   PutString(4,3,"1",0x78,0x00);
   PutString(4,10,"0",0x78,0x00);
   PutString(3,17,"-1",0x78,0x00);

   /* initial data */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
select = 0;
sbase = 0;
loadflag = OFF;
startflag = OFF;
pauseflag = OFF;

do{
  DisplayScreen(51,294,1);

  STARTLOOP:
  if (kbhit())
  {
        c=getch();
        switch (c)
        {
          case UP: if((select == 0)||(select == 1)||(select == 2))
                        break;
                   else
                   {
                    ButtonSelect(select,1);
                    switch(select)
                    {
                      case 3: select = 0;
                              break;
                      case 4: select = 1;
                              break;
                      case 5: select = 2;
                              break;
                      case 6: select = 3;
                              break;
                      case 7: select = 5;
                              break;
                      default: break;
                    }
                    ButtonSelect(select,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            break;
        case DOWN: if((select == 6)||(select == 7))
                    break;
                else
                {
                    ButtonSelect(select,1);
                    switch(select)
                    {
                      case 0: select = 3;
                              break;
                      case 1: select = 4;
                              break;
                      case 2: select = 5;
                              break;
                      case 3: select = 6;
                              break;
                      case 4: select = 7;
                              break;
                      case 5: select = 7;
                              break;
                      default: break;
                    }
                    ButtonSelect(select,0);
            }
            break;
        case LEFT: if((select == 0)||(select == 3)||(select == 6))
                    break;
            else
            {
                ButtonSelect(select,1);
                switch(select)
                {
                  case 1: select = 0;
                          break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    case 2: select = 1;
                            break;
                    case 4: select = 3;
                            break;
                    case 5: select = 4;
                            break;
                    case 7: select = 6;
                            break;
                    default: break;
                  }
                  ButtonSelect(select,0);
                }
                break;
case RIGHT: if((select == 2)||(select == 5)||(select == 7))
                    break;
                  else
                  {
                    ButtonSelect(select,1);
                    switch(select)
                    {
                      case 0: select = 1;
                              break;
                      case 1: select = 2;
                              break;
                      case 3: select = 4;
                              break;
                      case 4: select = 5;
                              break;
                      case 6: select = 7;
                              break;
                      default: break;
                    }
                    ButtonSelect(select,0);
                  }
                  break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case ENTER: switch(select)
            {
              case 0: DisplayButton(344,352,1,0x21,"eject.mlt");
                      DisplayButton(344,352,0,0x21,"eject.mlt");
                      if(loadflag)
                        goto STARTLOOP;

                      AGAIN:
                      PutString(3,19,"          ",0x00,0x00);
                      PutString(3,28,"Enter file name          ",0x1c,0x00);
                      filename[0] = 12;
                      GetString(19,28, filename,0x1c,0x00);
                      PutString(3,28,"                         ",0x1c,0x1c);

                      /* open file */
                      recfile = fopen(filename,"rb");

                      /* read record type */
                      fread(rectype, sizeof(char), 8, recfile);
                      rectype[8] = '\0';

                      /* condition define sbase */
                      if(strcmp(rectype,"speed   ") == 0)
                      {
                        sbase = 1500;
                        PutString(3,23,"Base speed is 1500 rpm ",0x21,0x1c);
                      }
                      if(strcmp(rectype,"torque  ") == 0)
                      {
                        sbase = 100;
                        PutString(3,23,"Base torque is 100 %   ",0x21,0x1c);
                      }
                      if(strcmp(rectype,"current ") == 0)
                      {
                        sbase = 10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PutString(3,23,"Base current is 10 Amp.",0x21,0x1c);
              }

              if((strcmp(rectype,"speed   ") != 0)&&(strcmp(rectype,"torque  ")
!= 0)
               &&(strcmp(rectype,"current ") != 0))
              {
                PutString(3,28,"File error try again.      ",0x1c,0x00);
                delay(3000);
                PutString(3,28,"                           ",0x1c,0x00);
                goto AGAIN;
              }

              PutString(3,1,rectype,0x78,0x00); /* display type of record */
              sprintf(stringbase,"Scale is 1 sec/cell.");
              PutString(3,25,stringbase,0x21,0x1c);

              /* read time start type */
              fread(&timer, sizeof(time_t), 1, recfile);
              rectime[24] = '\0';

              /* find pointer first data */
              fseek(recfile,12L,SEEK_SET);
              firstdata = ftell(recfile);

              /* find pointer end data */
              fseek(recfile,0L,SEEK_END);
              enddata = ftell(recfile);

              pointerdata = firstdata; /* set pointer check end of file*/
              readpointer = firstdata; /* set read pointer */
              fseek(recfile,12L,SEEK_SET); /* go to first data */

              /* first time read data */
              for(i = 0;i < 528;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                  if(pointerdata == enddata)
                        pixel[i] = 0x8000;
                  else
                        fread(&pixel[i], sizeof(int), 1, recfile);
                  pointerdata = ftell(recfile);
                }

                /* first time initial data to delpixel */
                for(i = 0;i < 528;i++)
                delpixel[i] = 0x8000;
                loadflag = ON;
                break;
        case 1: DisplayButton(440,352,1,0x21,"play.mlt");
                DisplayButton(440,352,0,0x21,"play.mlt");
                if(!loadflag)
                  goto STARTLOOP;
                if(!pauseflag)
                {
                  startflag = ON;
                  counterflag = 0;
                  tblock = localtime(&timer);
                  putVESAchar(6,19,0x5E,0x78,0x00);
                  PutString(3,20,asctime(tblock),0x78,0x00);
                }
                else
                  break;
                break;
        case 2: DisplayButton(536,352,1,0x21,"stopplay.mlt");
                DisplayButton(536,352,0,0x21,"stopplay.mlt");
                if(!loadflag)
                  goto STARTLOOP;
                fclose(recfile);

                DisplayFileMLT(0,0,"screen.mlt");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PutString(74,18,"t.",0x78,0x00);
                PutString(4,3,"1",0x78,0x00);
                PutString(4,10,"0",0x78,0x00);
                PutString(3,17,"-1",0x78,0x00);
                PutString(3,23,"                ",0x1c,0x1c);
                PutString(3,25,"                ",0x1c,0x1c);

                loadflag = OFF;
                startflag = OFF;
                pauseflag = OFF;
                break;
        case 3: DisplayButton(344,392,1,0x21,"pause.mlt");
                DisplayButton(344,392,0,0x21,"pause.mlt");
                if(!pauseflag)
                {
                  if(!startflag)
                      break;
                  else
                  {
                      startflag = OFF;
                      pauseflag = ON;
                  }
                }
                else
                {
                  startflag = ON;
                  pauseflag = OFF;
                }
                break;
        case 4: DisplayButton(440,392,1,0x21,"previous.mlt");
                DisplayButton(440,392,0,0x21,"previous.mlt");
                if(!pauseflag)
                  goto STARTLOOP;
                if(readpointer == firstdata)
                  goto STARTLOOP;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    for(i = 0;i < 528;i++)
                      delpixel[i] = pixel[i];
                    readpointer = readpointer - 2L;
                    fseek(recfile,readpointer,SEEK_SET);
                    for(i = 0;i < 528;i++)
                    {
                      if(pointerdata == enddata)
                            pixel[i] = 0x8000;
                      else
                            fread(&pixel[i], sizeof(int), 1, recfile);
                      pointerdata = ftell(recfile);
                    }
                    for(i = 0;i < 528;i++)
                    {
                      if(!(166-(int)(((float)delpixel[i]*112)/sbase) < 40))
                            WritePixel(51+i,166-
(int)(((float)delpixel[i]*112)/sbase),0x00);
                      if(!(166-(int)(((float)pixel[i]*112)/sbase) < 40))
                            WritePixel(51+i,166-
(int)(((float)pixel[i]*112)/sbase),0x30);
                    }
                    if(counterflag == 0)
                      {
                            counterflag = 15;
                            timer--;
                            tblock = localtime(&timer);
                            putVESAchar(6,19,0x5E,0x78,0x00);
                            PutString(3,20,asctime(tblock),0x78,0x00);
                      }
                      else
                            counterflag--;
                    break;
              case 5: DisplayButton(536,392,1,0x21,"next.mlt");
                    DisplayButton(536,392,0,0x21,"next.mlt");
                    if(!pauseflag)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                goto STARTLOOP;
            if(readpointer == enddata)
                goto STARTLOOP;
            if(pointerdata != enddata)
            {
                for(i = 0;i < 528;i++)
                        delpixel[i] = pixel[i];
                readpointer = readpointer + 2L;
                fseek(recfile,readpointer,SEEK_SET);
                for(i = 0;i < 528;i++)
                {
                        if(pointerdata == enddata)
                            goto STARTLOOP;
                        else
                            fread(&pixel[i], sizeof(int), 1, recfile);
                        pointerdata = ftell(recfile);
                }
                for(i = 0;i < 528;i++)
                {
                        if(!(166-(int)(((float)delpixel[i]*112)/sbase) < 40))
                            WritePixel(51+i,166-
(int)(((float)delpixel[i]*112)/sbase),0x00);
                        if(!(166-(int)(((float)pixel[i]*112)/sbase) < 40))
                            WritePixel(51+i,166-
(int)(((float)pixel[i]*112)/sbase),0x30);
                }
                if(counterflag == 15)
                {
                        counterflag = 0;
                        timer++;
                        tblock = localtime(&timer);
                        putVESAchar(6,19,0x5E,0x78,0x00);
                        PutString(3,20,asctime(tblock),0x78,0x00);
                }
                else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                counterflag++;
            }
            break;
        case 6: DisplayButton(344,432,1,0x21,"setvalue.mlt");
            DisplayButton(344,432,0,0x21,"setvalue.mlt");

            if(strcmp(rectype,"speed   ") == 0)
            {
              PutString(3,23,"Base speed is     rpm",0x1c,0x00);
              base[0] = 4;
              GetString(17,23, base,0x1c,0x00);
              sprintf(stringbase,"Base speed is %s rpm.",base);
              PutString(3,23,stringbase,0x21,0x1c);
              sbase = atoi(base);
            }
            if(strcmp(rectype,"speed   ") == 0)
            {
              PutString(3,23,"Base torque is    %",0x1c,0x00);
              base[0] = 3;
              GetString(18,23, base,0x1c,0x00);
              sprintf(stringbase,"Base torque is %s % ",base);
              PutString(3,23,stringbase,0x21,0x1c);
              sbase = atoi(base);
            }
            if(strcmp(rectype,"current ") == 0)
            {
              PutString(3,23,"Base current is   Amp.",0x1c,0x00);
              base[0] = 2;
              GetString(19,23, base,0x1c,0x00);
              sprintf(stringbase,"Base current is %s Amp. ",base);
              PutString(3,23,stringbase,0x21,0x1c);
              sbase = atoi(base);
            }

            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        case 7: DisplayButton(488,432,1,0x21,"exit.mit");
                                DisplayButton(488,432,0,0x21,"exit.mit");
                                goto EXIT;
                        default: break;
                      }
          default: break;
        }
    }
    if(startflag)
    {
        for(i = 0;i < 528;i++)
          delpixel[i] = pixel[i];
        readpointer = readpointer + 2L;
        fseek(recfile,readpointer,SEEK_SET);
        for(i = 0;i < 528;i++)
        {
          if(pointerdata == enddata)
          {
            startflag = OFF;
            pauseflag = ON;
            goto STARTLOOP;
          }
          else
            fread(&pixel[i], sizeof(int), 1, recfile);
          pointerdata = ftell(recfile);
        }
        for(i = 0;i < 528;i++)
        {
          if(!(166-(int)(((float)delpixel[i]*112)/sbase) < 40))
            WritePixel(51+i,166-(int)(((float)delpixel[i]*112)/sbase),0x00);
          if(!(166-(int)(((float)pixel[i]*112)/sbase) < 40))
            WritePixel(51+i,166-(int)(((float)pixel[i]*112)/sbase),0x30);
        }
        if(counterflag == 15)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            counterflag = 0;
            timer++;
            tblock = localtime(&timer);
            putVESAchar(6,19,0x5E,0x78,0x00);
            PutString(3,20,asctime(tblock),0x78,0x00);
          }
          else
            counterflag++;
      }
    }while(1);

    EXIT:

    CloseGraphic();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "c:\tc\project\headfile\serial.h"
#include"c:\tc\project\headfile\vesa.h"
#include"c:\tc\project\headfile\graph.h"
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

#define FALSE 0
#define TRUE (!FALSE)
#define OFF 0
#define ON 1

#define NOERROR     0           /* No error */
#define BUFOVFL     1           /* Buffer overflowed */

#define ESC 0x1B
#define DOWN 0x50
#define UP 0x48
#define LEFT 0x4b
#define RIGHT 0x4d
#define ENTER 0X0d
#define ASCII 0x007F            /* Mask ASCII characters */
#define SBUFSIZ0x4000           /* Serial buffer size */

#define EOT 0X04
#define ENQ 0X05
#define STX 0X02
#define ETX 0X03
#define SPACE 0X20
#define SUCC 0X06
#define UNSUC 0X15

int SError = NOERROR;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int portbase = 0;
void interrupt (*oldvects[2])();

static char ccbuf[SBUFSIZ];
unsigned int startbuf = 0;
unsigned int endbuf = 0;

FILE *testfile;
char filename[80],text;

/* Handle communications interrupts and put them in ccbuf */
void interrupt com_int(void)
{
  disable();
  if ((inportb(portbase + IIR) & RX_MASK) == RX_ID)
  {
    if (((endbuf + 1) & SBUFSIZ - 1) == startbuf) SError = BUFOVFL;
    ccbuf[endbuf++] = inportb(portbase + RXR);
    endbuf &= SBUFSIZ - 1;
  }

  /* Signal end of hardware interrupt */
  outportb(ICR, EOI);
  enable();
}

/* Output a character to the serial port */
int SerialOut(char x)
{
  long int timeout = 0x0000FFFFL;

  outportb(portbase + MCR , MC_INT | DTR | RTS);

  /* Wait for Clear To Send from modem */
  while ((inportb(portbase + MSR) & CTS) == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (!(--timeout))
    return (-1);

    timeout = 0x0000FFFFL;

    /* Wait for transmitter to clear */
    while ((inportb(portbase + LSR) & XMTRDY) == 0)
    if (!(--timeout))
    return (-1);

    disable();
    outportb(portbase + TXR, x);
    enable();

    return (0);
}

/* Output a string to the serial port */
void SerialString(char *string)
{
  int i;

  i = 0;
  while(string[i] != '\0')
  {
    SerialOut(string[i]);
    i++;
  }

}

/* This routine returns the current value in the buffer */
int getccb(void)
{
  int res;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (endbuf == startbuf)
    return (-1);

    res = (int) ccbuf[startbuf++];
    startbuf %= SBUFSIZ;
    return (res);
  }

  /* Install our functions to handle communications */
  void setvects(void)
  {
    if (portbase == COM1BASE)
    {
      oldvects[1] = getvect(0x0C);
      setvect(0x0C, com_int);
    }
    if (portbase == COM2BASE)
    {
    oldvects[0] = getvect(0x0B);
    setvect(0x0B, com_int);
    }
  }

  /* Uninstall our vectors before exiting the program */
  void resvects(void)
  {
    if (portbase == COM1BASE)
      setvect(0x0C, oldvects[1]);
    if (portbase == COM2BASE)
      setvect(0x0B, oldvects[0]);
  }

  /* Turn on communications interrupts */
  void i_enable(int pnum)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  int c;

  disable();
  c = inportb(portbase + MCR) | MC_INT;
  outportb(portbase + MCR , c);
  outportb(portbase + IER , RX_INT);
  c = inportb(IMR) & (((pnum == COM1) || (pnum == COM1)) ? IRQ4 : IRQ3);
  outportb(IMR, c);
  enable();
}

/* Turn off communications interrupts */
void i_disable(void)
{
  int c;

  disable();
  if (portbase == COM1BASE)
    c = inportb(IMR) | ~IRQ4;
  if (portbase == COM2BASE)
    c = inportb(IMR) | ~IRQ3;
  outportb(IMR, c);
  outportb(portbase + IER, 0);
  c = inportb(portbase + MCR) & ~MC_INT;
  outportb(portbase + MCR, c);
  enable();
}

/* Tell modem that we're ready to go */
void comm_on(void)
{
  int c , pnum;

  pnum = (portbase == COM1BASE ? COM1 : COM2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  i_enable(pnum);
  c = inportb(portbase + MCR) | DTR | RTS;
  outportb(portbase + MCR, c);
}

/* Go off-line */
void comm_off(void)
{
  i_disable();
  outportb(portbase + MCR, 0);
}

void initserial(void)
{
  endbuf = startbuf = 0;
  setvects();
  comm_on();
}

void closeserial(void)
{
  comm_off();
  resvects();
}

/* Set the port number to use */
int SetPort(int Port)
{
  int Offset , far *RS232_Addr;

  switch (Port)
  {
  /* Sort out the base address */
  case COM1 : Offset = 0x0000; break;
  case COM2 : Offset = 0x0002; break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  default          : return (-1);
  }

  RS232_Addr = MK_FP(0x0040 , Offset);  /* Find out where the port is.          */
  if (*RS232_Addr == NULL) return (-1); /* If NULL, then port not used. */
  portbase = *RS232_Addr; /* Otherwise, set portbase. */

  return (0);
}

/* This routine sets the speed; will accept funny baud rates. */
/* Setting the speed requires that the DLAB be set on. */
int SetSpeed(int Speed)
{
  char  c;
  int divisor;

  if (Speed == 0) /* Avoid divide by zero */
  return (-1);
  else
  divisor = (int) (115200L/Speed);

  if (portbase == 0)
  return (-1);

  disable();
  c = inportb(portbase + LCR);
  outportb(portbase + LCR, (c | 0x80)); /* Set DLAB   */
  outportb(portbase + DLL, (divisor & 0x00FF));
  outportb(portbase + DLH, ((divisor >> 8) & 0x00FF));
  outportb(portbase + LCR, c);     /* Reset DLAB */
  enable();

  return (0);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Set other communications parameters */
int SetOthers(int Parity , int Bits , int StopBit)
{
  int setting;

  if (portbase == 0)
  return(-1);
  if (Bits &5 || Bits > 8)
  return(-1);
  if (StopBit != 1 && StopBit != 2)
  return (-1);
  if (Parity != NO_PARITY && Parity != ODD_PARITY && Parity != EVEN_PARITY)
  return (-1);

  setting  = Bits-5;
  setting |= ((StopBit == 1) ? 0x00 : 0x04);
  setting |= Parity;

  disable();
  outportb(portbase + LCR, setting);
  enable();

  return (0);
}

/* Set up the port */
int SetSerial(int Port , int Speed , int Parity , int Bits , int StopBit)
{
  if (SetPort(Port)) return (-1);
  if (SetSpeed(Speed))    return (-1);
  if (SetOthers(Parity , Bits , StopBit)) return (-1);

  return (0);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Control-Break interrupt handler */
int c_break(void)
{
  i_disable();
  fprintf(stderr, "\nStill online.\n");

  return(0);
}

void ButtonSelect(int menu, int del)
{
  switch(menu)
  {
    case 0: if(del)
              DisplayButton(448,160,0,0xb2,"writecom.mlt");
            else
              DisplayButton(448,160,0,0x21,"writecom.mlt");
            break;
    case 1: if(del)
              DisplayButton(448,352,0,0xb2,"readcom.mlt");
            else
              DisplayButton(448,352,0,0x21,"readcom.mlt");
            break;
    case 2: if(del)
              DisplayButton(488,432,0,0xb2,"exit.mlt");
            else
              DisplayButton(488,432,0,0x21,"exit.mlt");
            break;
    default: break;
  }
}

char CalBlockChecksum(char *string)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    char bcc = 0X00;
    int count = 0;

    do{
      bcc = bcc^string[count];
      if(string[count]==ETX) goto OutLoop;
      ++count;
    }while(1);
    OutLoop:
    if(bcc < 0x20)
      bcc = bcc+0x20;
    return(bcc);
}

void GetParameter(char *string, int flag)
{
    GetPARStart:
    string[0] = 3;
    if(flag)
      GetString(15,18, string,0x1c,0x00);
    else
      GetString(15,6, string,0x1c,0x00);
    switch(string[0])
    {
      case '0': switch(string[1])
                {
                  case '0': if((string[2]<0x30)||(string[2]>0x39))
                                goto GetPARError;
                            else
                                goto GetPAREnd;
                  case '1': if((string[2]<0x30)||(string[2]>0x35))
                                goto GetPARError;
                            else
                                goto GetPAREnd;
                  default : goto GetPARError;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
        case '1': switch(string[1])
                {
                  case '0': if((string[2]<0x30)||(string[2]>0x39))
                                goto GetPARError;
                            else
                                goto GetPAREnd;
                  case '1': if((string[2]<0x30)||(string[2]>0x39))
                                goto GetPARError;
                            else
                                goto GetPAREnd;
                  case '2': if((string[2]<0x30)||(string[2]>0x37))
                                goto GetPARError;
                            else
                                goto GetPAREnd;
                  default : goto GetPARError;
                }
        case '2': switch(string[1])
                {
                  case '0': if((string[2]<0x30)||(string[2]>0x39))
                                goto GetPARError;
                            else
                                goto GetPAREnd;
                  case '1': if((string[2]<0x30)||(string[2]>0x39))
                                goto GetPARError;
                            else
                                goto GetPAREnd;
                  case '2': if((string[2]<0x30)||(string[2]>0x36))
                                goto GetPARError;
                            else
                                goto GetPAREnd;
                  default : goto GetPARError;
                }
        case '3': switch(string[1])
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case '0': if((string[2]<0x30)||(string[2]>0x39))
                                goto GetPARError;
                          else
                                goto GetPAREnd;
                case '1': if((string[2]<0x30)||(string[2]>0x38))
                                goto GetPARError;
                          else
                                goto GetPAREnd;
                default : goto GetPARError;
                }
        case '4': switch(string[1])
                {
                case '0': if((string[2]<0x30)||(string[2]>0x39))
                                goto GetPARError;
                          else
                                goto GetPAREnd;
                case '1': if((string[2]<0x30)||(string[2]>0x39))
                                goto GetPARError;
                          else
                                goto GetPAREnd;
                case '2': if((string[2]<0x30)||(string[2]>0x39))
                                goto GetPARError;
                          else
                                goto GetPAREnd;
                case '3': if((string[2]<0x30)||(string[2]>0x35))
                                goto GetPARError;
                          else
                                goto GetPAREnd;
                default : goto GetPARError;
                }
        case '5': switch(string[1])
                {
                case '0': if((string[2]<0x30)||(string[2]>0x39))
                                goto GetPARError;
                          else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                goto GetPAREnd;
            case '1': if((string[2]<0x30)||(string[2]>0x39))
                goto GetPARError;
            else
                goto GetPAREnd;
            case '2': if((string[2]<0x30)||(string[2]>0x35))
                goto GetPARError;
            else
                goto GetPAREnd;
            default : goto GetPARError;
        }
    case '6': switch(string[1])
        {
            case '0': if((string[2]<0x30)||(string[2]>0x39))
                goto GetPARError;
            else
                goto GetPAREnd;
            case '1': if((string[2]<0x30)||(string[2]>0x39))
                goto GetPARError;
            else
                goto GetPAREnd;
            case '2': if(string[2]!=0x30)
                goto GetPARError;
            else
                goto GetPAREnd;
            default : goto GetPARError;
        }
    case '9': switch(string[1])
        {
            case '0': if((string[2]<0x30)||(string[2]>0x35))
                goto GetPARError;
            else
                goto GetPAREnd;
            case '1': if(string[2]!=0x30)
                goto GetPARError;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    else
                        goto GetPAREnd;
                case '2': if(string[2]!=0x33)
                        goto GetPARError;
                    else
                        goto GetPAREnd;
                default : goto GetPARError;
                }
        default : goto GetPARError;
    }

    GetPARError:
    if(flag)
        PutString(15,21,"Parameter not correct",0x1c,0x00);
    else
        PutString(15,9,"Parameter not correct",0x1c,0x00);
    goto GetPARStart;

    GetPAREnd:
}

void ReadCommand(char *string)
{
    char rest[4];

    /* Read Log-On: */
    /* Initialize the Communication Link EOT */
    string[0] = EOT;
    /* Address drive */
    string[1] = 0x30;
    string[2] = 0x30;
    string[3] = 0x31;
    string[4] = 0x31;
    string[5] = '\0';
    /* Get command parameter */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    GetParameter(rest,1);
    /* Send the Parameter Number */
    strcat(string,rest);
    /* End the Message ENQ */
    string[8] = ENQ;
    string[9] = '\0';
}

void WriteCommand(char *string)
{
    char rest[17];
    int counter = 0, temp;

    /* Write Log-On: */
    /* Initialize the Communication Link EOT */
    string[0] = EOT;
    string[1] = 0x30;
    string[2] = 0x30;
    string[3] = 0x31;
    string[4] = 0x31;
    /* Write Command : */
    /* Start the Massage STX */
    string[5] = STX;
    string[6] = '\0';
    /* Get command parameter */
    GetParameter(rest,0);
    /* Send the Parameter Number */
    strcat(string,rest);
    /* Get value of command VAL */
    rest[0] = 16;
    PutString(29,6,"          ",0x00,0x00);
    GetString(29,6, rest,0x1c,0x00);
    do{
      if(rest[counter]=='\0')
            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ++counter;
  }while(1);
  if(rest[0] != '-')
  {
    string[9] = SPACE;
    string[10] = '\0';
  }
  strcat(string,rest);
  /* Indicate the End of the Data */
  if(rest[0] != '-')
  {
    string[10+counter] = ETX;
    string[11+counter] = '\0';
    string[11+counter] = CalBlockChecksum(&string[6]);
    string[12+counter] = '\0';
  }
  else
  {
    string[9+counter] = ETX;
    string[10+counter] = '\0';
    string[10+counter] = CalBlockChecksum(&string[6]);
    string[11+counter] = '\0';
  }
}


main()
{
  /* Communications parameters */
  int port = COM2;
  int speed = 19200;
  int parity = NO_PARITY;
  int bits = 8;
  int stopbits = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int c, select, number, numberpar;

char command[28], parameter[4], value[16];

if (SetSerial(port, speed, parity, bits, stopbits) != 0)
{
  fprintf(stderr, "Serial Port setup error.\n");
  return (99);
}

initserial();

ctrlbrk(c_break);

if(!OpenGraphic(0x101))
{
  fprintf(stderr,"Error opening VESA mode...\n");
  exit(0);
}

/* display background program */
DisplayFileMLT(0,0,"write.mlt");
DisplayFileMLT(0,208,"read.mlt");
DisplayFileMLT(0,400,"other.mlt");
PutString(5,1,"Write Command.",0x21,0x1c);
PutString(5,6,"Parameter",0x21,0x1c);
PutString(23,6,"Value",0x21,0x1c);
PutString(5,9,"Response",0x21,0x1c);
PutString(5,13,"Read Command.",0x21,0x1c);
PutString(5,18,"Parameter",0x21,0x1c);
PutString(23,18,"Value",0x21,0x1c);
PutString(5,21,"Response",0x21,0x1c);
/* display button */
DisplayButton(448,160,0,0x21,"writecom.mlt");
DisplayButton(448,352,0,0xb2,"readcom.mlt");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DisplayButton(488,432,0,0xb2,"exit.mlt");

/* initial value */
select = 0;

LOOPPROGRAM:
do{
  if (kbhit())
  {
        c=getch();
        switch (c)
        {
          case UP: if(select == 0)
                        break;
                   else
                   {
                        ButtonSelect(select,1);
                        select--;
                        ButtonSelect(select,0);
                   }
                   break;
          case DOWN: if(select == 2)
                        break;
                     else
                     {
                          ButtonSelect(select,1);
                          select++;
                          ButtonSelect(select,0);
                     }
                     break;
           case ENTER: switch(select)
                       {
                         case 0: DisplayButton(448,160,1,0x21,"writecom.mlt");
                                 DisplayButton(448,160,0,0x21,"writecom.mlt");
                                 WriteCommand(command);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                SerialString(command);
                                goto ANSWERWRITE;
                            case 1: DisplayButton(448,352,1,0x21,"readcom.mlt");
                                DisplayButton(448,352,0,0x21,"readcom.mlt");
                                ReadCommand(command);
                                SerialString(command);
                                goto ANSWERREAD;
                            case 2: DisplayButton(488,432,1,0x21,"exit.mlt");
                                DisplayButton(488,432,0,0x21,"exit.mlt");
                                goto EXIT;
                            default: break;
                          }
          default: break;
        }
    }
  }while(1);

  ANSWERWRITE:
  do {
    if ((c = getccb()) != -1)
    {
        switch(c)
        {
          case SUCC: PutString(15,9,"Command ok          ",0x1c,0x00);
                goto LOOPPROGRAM;
          case UNSUC: PutString(15,9,"Drive not response command",0x1c,0x00);
                goto LOOPPROGRAM;
          default: break;
        }
    }
  }while(!SError);

  ANSWERREAD:
  number = 0;
  numberpar = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
do {
  if ((c = getccb()) != -1)
  {
        if(c == STX)
        {
          goto LOOP2;
        }
  }
}while(!SError);

LOOP2:
do {
  if ((c = getccb()) != -1)
  {
        parameter[numberpar] = c;
        if(numberpar >2)
        {
          parameter[numberpar] = c;
          parameter[3] = '\0';
          goto LOOP3;
        }
        numberpar++;
  }
}while(!SError);

LOOP3:
do {
  if ((c = getccb()) != -1)
  {
        switch(c)
        {
          case EOT: goto COMMANDERROR;
          case ETX: value[number] = '\0';
                    goto COMMANDOK;
          default: value[number] = c;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    number++;
                    break;
                }
            }
        }while(!SError);

        COMMANDOK:
        PutString(29,18,value,0x1c,0x00);
        PutString(15,21,"Command ok              ",0x1c,0x00);
        goto LOOPPROGRAM;

        COMMANDERROR:
        PutString(15,21,"Drive not response command",0x1c,0x00);
        goto LOOPPROGRAM;

        EXIT:

        CloseGraphic();
        return (0);
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้